Digital Forensics and Incident Response

Second Edition

Incident response techniques and procedures to respond to modern cyber threats

Packt

www.packt.com

Gerard Johansen

Digital Forensics and Incident Response
Second Edition

Incident response techniques and procedures to respond to modern cyber threats

Gerard Johansen

Packt

BIRMINGHAM - MUMBAI

Digital Forensics and Incident Response
Second Edition

Commissioning Editor: Vijin Barucha
Acquisition Editor: Rahul Nair
Content Development Editor: Ronn Kurien
Senior Editor: Richard Brookes-Bland
Technical Editor: Dinesh Pawar
Copy Editor: Safis Editing
Project Coordinator: Anish Daniel
Proofreader: Safis Editing
Indexer: Tejal Daruwale Soni
Production Designer: Arvindkumar Gupta

First published: July 2017
Second edition: January 2020

Production reference: 1280120

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-83864-900-5

www.packt.com

Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Contributors

About the author

Gerard Johansen is an incident response professional with over 15 years' experience in areas like penetration testing, vulnerability management, threat assessment modeling, and incident response. Beginning his information security career as a cyber crime investigator, he has built on that experience while working as a consultant and security analyst for clients and organizations ranging from healthcare to finance. Gerard is a graduate of Norwich University's Master of Science in Information Assurance program and a certified information systems security professional.

He is currently employed as a senior incident response consultant with a large technology company, focusing on incident detection, response, and threat intelligence integration.

I would like to thank my family for their support in this endeavor. Thank you also to my teammates, from whom I have learned a great deal. Finally, thank you to the staff at Packt Publishing for their tireless efforts in publishing this volume.

About the reviewer

Kyle Anderson is a graduate of the Joint Cyber Analysis Course (JCAC), and holds a Master of Science (M.S.) degree in digital forensics from Champlain College and a Bachelor of Arts degree in theater from Idaho State University. Kyle is currently serving in the United States Navy his focus being incident response, digital forensics and malware analyst. As a DF and IR team lead, he has guided analysis of multiple incidents, including cases involving sensitive data spillage, insider threats, and malicious compromise. He was responsible for creating and providing forensics and malware analysis training to a wide variety of audiences, including Navy red team members, junior forensic and malware analysts, and other government employees.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Preface

Digital Forensics and Incident Response – Second Edition provides an overview of the various topics surrounding the various technical and operational aspects of incident response and digital forensics. This will start with an examination of the proactive actions to take to ensure that an organization is ready for an incident. Next, the integration of digital forensic concepts and techniques and how they relate to incident response is addressed. Moving from concepts to actual techniques, you will be shown how to acquire evidence from a variety of sources including disk, memory, and networks. You will then be guided through examining those sources of evidence for indicators of compromise or attack. Next, you will examine the role of reporting your findings and how to configure reports for the various entities that require insight into an incident. To round out the skill set, the roles of malware analysis, threat intelligence, and threat hunting are discussed. By the end of this book, you will have a solid foundation in the forensic techniques and methodologies of incident response, as well as the experience required to bring these techniques into your own organization to better prepare for a potential security incident.

Who this book is for

This book is for the information security professional, digital forensic practitioner, and students with knowledge and experience in the use of software applications and basic command-line usage. This book will also help information security professionals who are new to an incident response, digital forensics, or threat hunting role within their organization.

What this book covers

`Chapter 1`, *Understanding Incident Response*, addresses the incident response process at a high level and explains how to craft an incident response framework within an enterprise. This framework allows the detailed and orderly investigation of an incident's root cause, the containment of the incident to lessen the impact, and finally, the remediation of damage to bring the enterprise back to a normal state.

`Chapter 2`, *Managing Cyber Incidents*, discusses the incident management framework, which provides a strategic construct for incident response. In this chapter, you will be guided through managing the incident. This includes tactical-level issues such as incident escalation, configuring an incident war room, crisis communication, and the technical aspects of bringing an organization back to normal.

`Chapter 3`, *Fundamentals of Digital Forensics,* focuses on the fundamental aspects of digital forensics. This includes an examination of the history of digital forensics, the basic elements of forensic science, and how these techniques are integrated into the incident response framework.

`Chapter 4`, *Collecting Network Evidence*, focuses on the acquisition of network-based evidence. This includes log files from network devices such as firewalls, routers, switches, proxy servers, and other network-layer devices. Other types of evidence such as packet captures will also be explored.

`Chapter 5`, *Acquiring Host-Based Evidence*, explains that compromised hosts are often the target of attacks, either as the direct target or as a pivot point into other areas of the network. Evidence from these systems is critical in determining root causes. This chapter focuses on the tools and techniques used to capture the volatile memory, log files, and other pertinent evidence.

`Chapter 6`, *Forensic Imaging*, explains that physical disk drives from compromised systems are a significant source of evidence. In order to ensure that this evidence is sound, it has to be acquired properly. This chapter focuses on the proper methods to image a suspect **hard disk drives** (**HDDs**).

`Chapter 7`, *Analyzing Network Evidence*, shows how to use open source tools such as tcpdump, Wireshark, and Moloch. You will be guided through the analysis of network evidence to identify command and control channels or data exfiltration. This evidence will be further correlated with other network evidence, such as a network proxy or firewall logs and packet captures.

`Chapter 8`, *Analyzing System Memory,* through the use of several industry-standard tools, shows various methods for identifying malicious activity contained within the system memory. These include methods for identifying malicious processes, network connections, and other indicators associated with malware running on an infected system.

`Chapter 9`, *Analyzing System Storage*, is an overview of the tools and techniques available for extracting evidence from previously imaged HDDs. An overview of some of the methods available to examine a system's storage is explored, but it should be noted that due to the depth of this topic, this chapter will only highlight certain aspects.

`Chapter 10`, *Analyzing Log Files*, explores the various Windows OS logs that are created during legitimate and adversarial behavior. You will be shown methods to analyze log files with open source tools to examine security, system or application event logs, and to identify potential indicators of compromise.

`Chapter 11`, *Writing the Incident Report*, discusses crafting a written document that captures the actions of responders and their analysis, which is as critical as the investigation itself. This chapter focuses on preparing reports for key internal and external stakeholders, including potential legal entities. The end goal is to prepare a report that stands up to the scrutiny of a court of law.

`Chapter 12`, *Malware Analysis for Incident Response*, provides an overview of some of the tools and techniques that are deployed when examining malicious code. This includes static analysis techniques to identify key indicators, as well as dynamic analysis where the behavior of the malware is explored.

`Chapter 13`, *Leveraging Threat Intelligence*, explains that threat intelligence has become more and more important to incident response by providing details of the wider context of adversarial tactics, techniques, and procedures. This chapter will give you an understanding of threat intelligence and how it can be applied to the incident response process.

`Chapter 14`, *Hunting for Threats*, introduces a methodology that integrates digital forensics tools and techniques with threat intelligence to determine whether a network has been compromised. This chapter explores the methodology of threat hunting and how threat intelligence can facilitate hunting through the crafting of a threat hunt hypothesis and indicators to hunt for.

`Chapter 15`, *Appendix*, includes the most critical events that pertain to security and incident investigations and have been provided as a reference. There is a significant number of Windows Event Log types available to IT and security professionals.

To get the most out of this book

Readers should be familiar with the Windows OS and have the ability to download and run applications as well as to use the Windows command line. Familiarity with the Linux command line is also helpful. An understanding of the basic network protocols and various types of network traffic is required as well. It's not required, but it is helpful to have access to a virtualization software platform and a Windows OS in which to run specific tools. Finally, incident response and digital forensics is a growing field. You will get the most out of this book by continuing to research and try new tools and techniques.

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `http://www.packtpub.com/sites/default/files/downloads/9781838649005_ColorImages.pdf`.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Once in Command Prompt, navigate to the folder containing the `RawCap.exe` file."

A block of code is set as follows:

```
meta:
 description = "Stuxnet Sample - file ~WTR4141.tmp"
 author = "Florian Roth"
 reference = "Internal Research"
 date = "2016-07-09"
```

Any command-line input or output is written as follows:

```
dfir@ubuntu:~$ tcpdump -h
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Click on **File** and then on **Capture Memory**."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at `customercare@packtpub.com`.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit `packt.com`.

1
Section 1: Foundations of Incident Response and Digital Forensics

Section one of this book lays the foundations of incident response and digital forensics. These foundational elements, such as the IR process, forensic principles, and incident management, will tie in directly with subsequent parts of the book.

This section comprises the following chapters:

- `Chapter 1`, *Understanding Incident Response*
- `Chapter 2`, *Managing Cyber Incidents*
- `Chapter 3`, *Fundamentals of Digital Forensics*

1
Understanding Incident Response

When examining the threats to today's information technology, it can seem overwhelming. From simple script kiddies using off-the-shelf code to nation state adversary tools, it is critical to be prepared. For example, an internal employee can download a single instance of ransomware and can have a significant impact on an organization. More complex attacks such as a network exploitation attempt or targeted data breach increases the chaos that a security incident causes. Technical personnel will have their hands full attempting to determine the systems that have been impacted and how they are being manipulated. They will also have to contend with addressing the possible loss of data through compromised systems. Adding to this chaotic situation are senior managers haranguing them for updates and an answer to the all-important questions: *How did this happen?* and *How bad is it?*

Having the ability to properly respond to security incidents in an orderly and efficient manner allows organizations to both limit the damage of a potential cyber attack, but also recover from the associated damage that is caused. To facilitate this orderly response, organizations of all sizes have looked at adding an incident response capability to their existing policies, procedures, and processes.

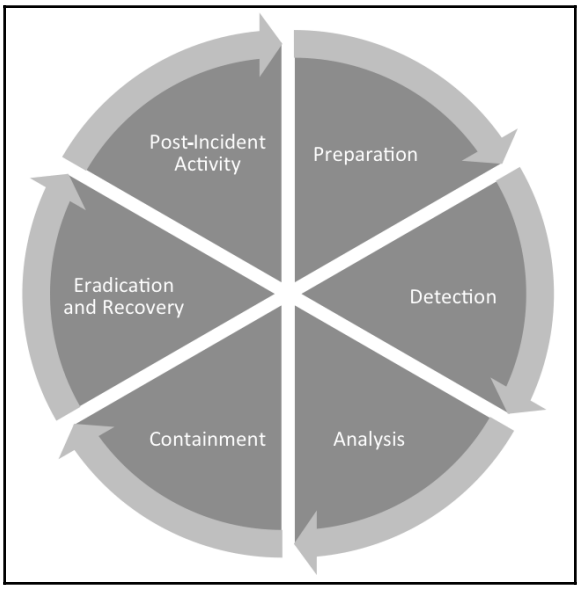
In order to build this capability within the organization, several key components must be addressed. First, organizations need to have a working knowledge of the incident response process. This process outlines the general flow of an incident and the general actions that are taken at each stage. Second, organizations need to have access to personnel who form the nucleus of any incident response capability. Once a team is organized, a formalized plan and associated processes need to be created. This written plan and processes form the orderly structure that an organization can follow during an incident. Finally, with this framework in place, the plan must be continually evaluated, tested, and improved as new threats immerge. Utilizing this framework will position organizations to be prepared for the unfortunate reality that many organizations have already faced, an incident that compromises their security.

We will be covering the following topics in this chapter:

- The incident response process
- The incident response framework
- The incident response plan
- The incident response playbook
- Testing the incident response framework

The incident response process

There is a general path that cyber security incidents follow during their lifetime. If the organization has a mature incident response capability, they will have taken measures to ensure they are prepared to address an incident at each stage of the process. Each incident starts with the first time the organization becomes aware of an event or series of events indicative of malicious activity. This detection can come in the form of a security control alert or external party informing the organization of a potential security issue. Once alerted, the organization moves through analyzing the incident through containment measures to bring the information system back to normal operations. The following diagram shows how these flow in a cycle with **Preparation** as the starting point. Closer examination reveals that every incident is used to better prepare the organization for future incidents as the **Post-Incident Activity**, and is utilized in the preparation for the next incident:



The incident response process can be broken down into six distinct phases, each with a set of actions the organization can take to address the incident:

- **Preparation**: Without good preparation, any subsequent incident response is going to be disorganized and has the potential to make the incident worse. One of the critical components of preparation is the creation of an incident response plan. Once a plan is in place with the necessary staffing, ensure that personnel detailed with incident response duties are properly trained. This includes processes, procedures, and any additional tools necessary for the investigation of an incident. In addition to the plan, tools such as forensics hardware and software should be acquired and incorporated into the overall process. Finally, regular exercises should be conducted to ensure that the organization is trained and familiar with the process.
- **Detection**: The detection of potential incidents is a complex endeavor. Depending on the size of the organization, they may have over 100 million separate events per day. These events can be records of legitimate actions taken during the normal course of business or be indicators of potentially malicious activity. Couple this mountain of event data with other security controls constantly alerting to activity and you have a situation where analysts are inundated with data and must subsequently sift out the valuable pieces of signal from the vastness of network noise. Even today's cutting-edge **Security Incident and Event Management** (**SIEM**) tools lose their effectiveness if they are not properly maintained with regular updates of rule sets that identify what events qualify as a potential incident. The detection phase is that part of the incident response process where the organization first becomes aware of a set of events that possibly indicates malicious activity. This event, or events, that have been detected and are indicative of malicious behavior are then classified as an incident. For example, a security analyst may receive an alert that a specific administrator account was in use during the time where the administrator was on vacation. Detection may also come from external sources. An ISP or law enforcement agency may detect malicious activity originating in an organization's network and contact them and advise them of the situation.

In other instances, users may be the first to indicate a potential security incident. This may be as simple as an employee contacting the help desk and informing a help desk technician that they received an Excel spreadsheet from an unknown source and opened it. They are now complaining that their files on the local system are being encrypted. In each case, an organization would have to escalate each of these events to the level of an incident (which we will cover a little later in this chapter) and begin the reactive process to investigate and remediate.

- **Analysis**: Once an incident has been detected, personnel from the organization or a trusted third party will begin the analysis phase. In this phase, personnel begin the task of collecting evidence from systems such as running memory, log files, network connections, and running software processes. Depending on the type of incident, this collection can take as little as a few hours to several days.

  Once the evidence is collected, it then needs be examined. There are a variety of tools to conduct this analysis, many of which are explored in this book. With these tools, analysts are attempting to ascertain what happened, what it affected, whether any other systems were involved, and whether any confidential data was removed. The ultimate goal of the analysis is to determine the root cause of the incident and reconstruct the actions of the threat actor from initial compromise to detection.

- **Containment**: Once there is a solid understanding of what the incident is and what systems are involved, organizations can then move into the containment phase. In this phase, organizations take measures to limit the ability for threat actors to continue compromising other network resources, communicating with command and control infrastructures, or exfiltrating confidential data. Containment strategies can range from locking down ports and IP addresses on a firewall to simply removing the network cable from the back of an infected machine. Each type of incident involves its own containment strategy, but having several options allows personnel to stop the bleeding at the source if they are able to detect a security incident before or during the time when threat actors are pilfering data.

- **Eradication and recovery**: During the eradication phase, the organization removes the threat actor from the impacted network. In the case of a malware infection, the organization may run an enhanced anti-malware solution. Other times, infected machines must be wiped and reimaged. Other activities include removing or changing compromised user accounts. If an organization has identified a vulnerability that was exploited, vendor patches are applied, or software updates are made. Recovery activities are very closely aligned with those that may be found in an organization's *business continuity or disaster recovery* plans. In this phase of the process, organizations reinstall fresh operating systems or applications. They will also restore data on local systems from backups. As a due diligence step, organizations will also audit their existing user and administrator accounts to ensure that there are no accounts that have been enabled by threat actors. Finally, a comprehensive vulnerability scan is conducted so that the organization is confident that any exploitable vulnerabilities have been removed.
- **Post-incident activity**: At the conclusion of the incident process is a complete review of the incident with all the principle stakeholders. Post-incident activity includes a complete review of all the actions taken during the incident. What worked, and more importantly, what did not work, are important topics for discussion. These reviews are important because they may highlight specific tasks and actions that had either a positive or negative impact on the outcome of the incident response. It is during this phase of the process that a written report is completed. Documenting the actions taken during the incident is critical to capture both what occurred and whether the incident will ever see the inside of a courtroom. For documentation to be effective, it should be detailed and show a clear chain of events with a focus on the root cause, if it was determined. Personnel involved in the preparation of this report should realize that stakeholders outside of information technology might read this report. As a result, technical jargon or concepts should be explained.

Finally, the organizational personnel should update their own incident response processes with any new information developed during the post-incident debrief and reporting. This incorporation of *lessons learned* is important as it makes future responses to incidents more effective.

The role of digital forensics

There is a misconception that is often held by people unfamiliar with the realm of incident response. This misconception is that incident response is merely a digital forensics issue. As a result, they will often conflate the two terms. While digital forensics is a critical component to incident response (and for this reason we have included a number of chapters in this book to address digital forensics), there is more to addressing an incident than examining hard drives. It is best to think of forensics as a supporting function of the overall incident response process. Digital forensics serves as the mechanism for understanding the technical aspects of the incident, potentially identifying the root cause, and discovering unidentified access or other malicious activity. For example, some incidents such as **Denial of Service** (**DoS**) attacks will require little to no forensic work. On the other hand, a network intrusion involving the compromise of an internal server and **Command and Control** (**C2**) traffic leaving the network will require extensive examination of logs, traffic analysis, and examination of memory. From this analysis may be derived the root cause. In both cases, the impacted organization would be able to connect with the incident, but forensics played a much more important role in the latter case.

Incident response is an information security function that uses the methodologies, tools, and techniques of digital forensics but goes beyond what digital forensics provides by addressing additional elements. These elements include containing possible malware or other exploits, identifying and remediating vulnerabilities, and managing various technical and non-technical personnel. Some incidents may require the analysis of host-based evidence or memory, others may only require a firewall log review but, in each, the responders will follow the incident response process.

The incident response framework

Responding to a data breach, ransomware attack, or other security incident should never be an ad hoc process. Undefined processes or procedures will leave an organization unable to both identify the extent of the incident and be able to stop the bleeding in sufficient time to limit damage. Further, attempting to craft plans during an incident may in fact destroy critical evidence, or worse, create more problems.

Having a solid understanding of the incident response process is just the first step to building this capability within an organization. What organizations need is a framework that puts that processes to work utilizing the organization's available resources. The incident response framework describes the components of a functional incident response capability within an organization. This framework is made up of elements such as personnel, policies, and procedures. It is through these elements that an organization builds its capability to respond to incidents.

The incident response charter

The first step to building this capability is the decision by senior leadership that the risk to the organization is too significant not to address the possibility of a potential security incident. Once that point is reached, a senior member of the organization will serve as a project sponsor and craft the incident response charter. This charter outlines key elements that will drive the creation of a **Computer Security Incident Response Team** (**CSIRT**).

While there are several titles for incident response teams, the term CERT is often associated with the US-CERT through the United States Department of Homeland Security or the **Computer Emergency Response Team Coordination Center** (**CERT/CC**), through the Carnegie Mellon Software Engineering Institute. For our purposes, we will use the more generic CSIRT.

The incident response charter should be a written document that addresses the following:

- **Obtain senior leadership support**: In order to be a viable part of the organization, the CSIRT requires the support of the senior leadership within the organization. In a private sector institution, it may be difficult to obtain the necessary support and funding, as the CSIRT itself does not provide value in the same way marketing or sales does. What should be understood is that the CSIRT acts as an insurance policy in the event the worse happens. In this manner, a CSRIT can justify its existence by reducing the impact of incidents and thereby reducing the costs associated with a security breach or other malicious activity.
- **Define the constituency**: The constituency clearly defines which organizational elements and domains the CSIRT has responsibility for. Some organizations have several divisions or subsidiaries that for whatever reason may not be part of the CSIRT's responsibility. The constituency can be defined either as a domain such as `local.example.com` or an organization name such as ACME Inc. and associated subsidiary organizations.

- **Create a mission statement**: Mission creep or the gradual expansion of the CSIRT's responsibilities can occur without clear definition of what the defined purpose of the CSIRT is. In order to counter this, a clearly defined mission statement should be included with the written information security plan. For example, *the mission of the ACME Inc. CSIRT is to provide timely analysis and actions to security incidents that impact the confidentiality, integrity, and availability of ACME Inc. information systems and personnel.*
- **Determine service delivery**: Along with a mission statement, a clearly defined list of services can also counter the risk of mission creep of the CSIRT. Services are usually divided into two separate categories, proactive and reactive services:
    - **Proactive services:** These includes providing training for non-CSIRT staff, providing summaries on emerging security threats, testing and deployment of security tools such as endpoint detection and response tools, and assisting security operations with crafting IDS/IPS alerting rules.
    - **Reactive services**: These primarily revolve around responding to incidents as they occur. For the most part, reactive services address the entire incident response process. This includes the acquisition and examination of evidence, assisting in containment, eradication, and recovery efforts, and finally documenting the incident.

Another critical benefit of an expressly stated charter is to socialize the CSIRT with the entire organization. This is done to remove any rumors or innuendo about the purpose of the team. Employees of the organization may hear words such as digital investigations or incident response team and believe the organization is preparing a *secret police* specifically designed to ferret out employee misconduct. To counter this, a short statement that includes the mission statement of the CSIRT can be made available to all employees. The CSIRT can also provide periodic updates to senior leadership on incidents handled to demonstrate the purpose of the team.

CSIRT

Once the incident response charter is completed, the next stage is to start staffing the CSIRT. Larger organizations with sufficient resources may be able to task personnel with incident response duties full-time. More often than not though, organizations will have to utilize personnel who have other duties outside incident response. Personnel who comprise the internal CSIRT can be divided into three categories: core team, technical support, and organizational support. Each individual within the CSIRT fulfills a specific task. Building this capability into an organization takes more than just assigning personnel and creating a policy and procedure document. Like any major project initiative, there is a good deal of effort required in order to create a functional CSIRT.

For each of the CSIRT categories, there are specific roles and responsibilities. This wide range of personnel is designed to provide guidance and support through a wide range of incidents ranging from the minor to the catastrophic.

CSIRT core team

The CSIRT core team consists of personnel who have incident response duties as their full-time job or assume incident response activities when needed. In many instances, the core team is often made up of personnel assigned to the information security team. Other organizations can leverage personnel with expertise in incident response activities. The following are some of the roles that can be incorporated into the core team:

- **Incident response coordinator**: This is a critical component of any CSIRT. Without clear leadership, the response to a potential incident may be disorganized or with multiple individuals vying for control during an incident, a chaotic situation that can make the incident worse. In many instances, the incident response coordinator is often the **Chief Security Officer (CSO**), **Chief Information Security Officer** (**CISO**), or the **Information Security Officer** (**ISO**) as that individual often has overall responsibility for the security of the organization's information. Other organizations may name a single individual who serves as the incident response coordinator.

The incident response coordinator is responsible for management of the CSIRT prior to, during, and after an incident. In terms of preparation, the incident response coordinator will ensure that any plans or policies concerning the CSIRT are reviewed periodically and updated as needed. In addition, the incident response coordinator is responsible for ensuring that the CSIRT team is appropriately trained and oversees testing and training for CSIRT personnel. During an incident, the incident response coordinator is responsible for ensuring the proper response and remediation of an incident and guides the team through the entire incident response process. One of the most important of these tasks during an incident is coordination of the CSIRT with senior leadership. With the stakes of a data breach being high, senior leadership such as the **Chief Executive Officer** (**CEO**) will want to be informed of the critical information concerning an incident. It is the responsibility of the incident response coordinator to ensure that the senior leadership is fully informed of the activities associated with an incident using clear and concise language. One stumbling block is that senior leaders within an organization may not have the acumen to understand the technical aspects of an incident, so it is important to speak in language they will understand.

Finally, at the conclusion of an incident, the incident response coordinator is responsible for ensuring that the incident is properly documented and that reports of the CSIRT activity are delivered to the appropriate internal and external stakeholders. In addition, a full debrief of all CSIRT activities is conducted and lessons learned are incorporated into the CSIRT plan.

- **CSIRT senior analyst(s)**: CSIRT senior analysts are personnel with extensive training and experience in incident response and associated skills such as digital forensics or network data examination. They often have several years of experience conducting incident response activities as either a consultant or as part of an enterprise CSIRT.

During the preparation phase of the incident response process, they are involved in ensuring that they have the necessary skills and training to address their specific role in the CSIRT. They are also often directed to assist in the incident response plan review and modification. Finally, senior analysts will often take part in training junior members of the team.

Once an incident has been identified, the senior analysts will engage with other CSIRT members to acquire and analyze evidence, direct containment activities, and assist other personnel with remediation.

At the conclusion of an incident, the senior analysts will ensure that both they and other personnel appropriately document the incident. This will include the preparation of reports to internal and external stakeholders. They will also ensure that any evidence is appropriately archived or destroyed depending on the incident response plan.

- **CSIRT analyst(s)**: The CSIRT analysts are personnel with CSIRT responsibilities that have less exposure or experience in incident response activities. Oftentimes, they have only one or two years' experience of responding to incidents. As a result, they can perform a variety of activities with some of those under the direction of senior analysts.

In terms of preparation phase activities, analysts will develop their skills via training and exercises. They may also take part in reviews and updates to the incident response plan. During an incident, they will be tasked with gathering evidence from potentially compromised hosts, from network devices, or from various log sources. Analysts will also take part in the analysis of evidence and assist other team members in remediation activities.

- **Security operations center analyst**: Larger enterprises may have an in-house or contracted 24/7 **Security Operations Center** (**SOC**) monitoring capability. Analysts assigned to the SOC will often serve as the point person when it comes to incident detection and alerting. As a result, having an SOC analyst as part of the team allows them to be trained in incident identification and response techniques and serve as an almost immediate response to a potential security incident.
- **IT security engineer/analyst(s)**: Depending on the size of the organization, there may be personnel specifically tasked with the deployment, maintenance, and monitoring of security-related software such as anti-virus or hardware such as firewalls or SIEM systems. Having direct access to these devices is critical when an incident has been identified. The personnel assigned these duties will often have a direct role in the entire incident response process.

The IT security engineer or analyst will often have a large piece of the preparation component of the incident response process. They will be the primary resource to ensure that security applications and devices are properly configured to alert to possible incidents and to ensure that the devices properly log events so that a reconstruction of events can take place.

During an incident, they will be tasked with monitoring security systems for other indicators of malicious behavior. They will also assist the other CSIRT personnel with obtaining evidence from the security devices. Finally, after an incident, these personnel will be tasked with configuring security devices to monitor for suspected behavior to ensure that remediation activities have eradicated the malicious activity on impacted systems.

Technical support personnel

Technical support personnel are those individuals within the organization who do not have CSIRT activities as part of their day-to-day operations, but rather have expertise or access to systems and processes that may be affected by an incident. For example, the CSIRT may need to engage a server administrator to assist the core team with acquiring evidence from servers such as memory captures, acquiring virtual systems, or offloading log files. Once completed, the server administrator's role is completed and they may have no further involvement in the incident. The following are some of the personnel that can be of assistance to the CSIRT during an incident:

- **Network architect/administrator**: Often, incidents involve the network infrastructure. This includes attacks on routers, switches, and other network hardware and software. The network architect or administrator is vital for insight into what is normal and abnormal behavior for these devices as well as identifying anomalous network traffic. In incidents where the network infrastructure is involved, these support personnel can assist with obtaining network evidence such as access logs or packet captures.
- **Server administrator**: Threat actors often target systems within the network where critical or sensitive data is stored. These high-value targets often include domain controllers, file servers, or database servers. Server administrators can aid in acquiring log files from these systems. If the server administrator(s) are also responsible for the maintenance of the active directory structure, they may be able to assist with identifying new user accounts or changes to existing user or administrator accounts.
- **Application support**: Web applications are a prime target for threat actors. Flaws in coding that allow for attacks such as SQL injection or security misconfigurations are responsible for some security breaches. As a result, having application support personnel as part of the CSIRT allows for direct information related to application attacks. These individuals will often be able to identify code changes or to confirm vulnerabilities discovered during an investigation into a potential attack against an application.

- **Desktop support**: Desktop support personnel are often involved in maintaining controls such as data loss prevention and anti-virus on desktop systems. In the event of an incident, they can assist in providing the CSIRT with log files and other evidence. They may also be responsible for cleaning up infected systems during the remediation phase of an incident.
- **Help desk**: Depending on the organization, help desk personnel are the proverbial *canary in the coal mine* when it comes to identifying an incident. They are often the first individuals contacted when a user experiences the first signs of a malware infection or other malicious activity. Thus, help desk personnel should be involved in training of the CSIRT responses and their role in the incident identification and escalation procedures. They may also assist with identifying additional affected personnel in the event of a widespread incident.

Organizational support personnel

Outside of the technical realm, there are still other organizational members that should be included within the CSIRT. Organizational personnel can assist with a variety of non-technical issues that fall outside those that are addressed by the CSIRT core and technical support personnel. These include navigating the internal and external legal environment, assisting with customer communications, or supporting CSIRT personnel while onsite.

The following are some of the organizational support personnel that should be included in a CSIRT plan:

- **Legal**: Data breaches and other incidents carry a variety of legal issues along with them. Many countries now have breach notification laws where organizations are required to notify customers that their information was put at risk. Other compliance requirements such as HIPAA and the PCI DSS require the impacted organization to contact various external bodies and notify them of a suspected breach. Including legal representation early in the incident response process will ensure that these notifications and any other legal requirements are addressed in a timely fashion. In the event that a breach has been caused by an internal source such as an employee or contractor, the impacted organization may want to recoup losses through civil action. Including legal representation early in the process will allow for a more informed decision as to what legal process should be followed.

- **Human resources**: A good deal of incidents that occur in organizations are perpetrated by employees or contractors. The investigation of actions such as fraud all the way to massive data theft may have to be investigated by the CSIRT. In the event that the target of the investigation is an employee or contractor, the human resources department can assist with ensuring that the CSIRT's actions are in compliance with applicable labor laws and company policies. If an employee or contractor is to be terminated, the CSIRT can coordinate with the human resources personnel so that all proper documentation concerning the incident is complete to reduce the potential of a wrongful termination suit.
- **Marketing/communications**: If external clients or customers may be adversely impacted by an incident such as a DoS attack or data breach, the marketing or communications department can assist in crafting the appropriate message to assuage fears and ensure that those external entities are receiving the best information possible. When looking back at past data breaches where organizations attempted to keep the details to themselves and customers were not informed, there was a backlash against those organizations. Having a solid communications plan that is put into action early will go a long way to soothing any potential customer or client adverse reactions.
- **Facilities**: The CSIRT may need access to areas after hours or for a prolonged time. The facilities department can assist the CSIRT in obtaining the necessary access in a timely manner. Facilities also may have access to additional meeting spaces for the CSIRT to utilize in the event of a prolonged incident that requires dedicated workspace and infrastructure.
- **Corporate security**: The CSIRT may be called in to deal with the theft of network resources or other technology from the organization. Laptop and digital media theft are very common. Corporate security will often have access to surveillance footage from entrances and exits. They may also maintain access badges and visitor logs for the CSIRT to track movement of employees and other personnel within the facility. This can allow for a reconstruction of events leading up to a theft or other circumstances that led up to the incident.

External resources

Many industries have professional organizations where practitioners, regardless of their employer, can come together to share information. CSIRT personnel may also be tasked with interfacing with law enforcement and government agencies at times, especially if they are targeted as part of a larger attack perpetrated against a number of similar organizations. Having relationships with external organizations and agencies can assist the CSIRT with intelligence sharing and resources in the event of an incident. These resources include the following:

- **High Technology Crime Investigation Association (HTCIA)**: The HTCIA is an international group of professionals and students with a focus on high-tech crime. Resources include everything from digital forensic techniques to wider enterprise-level information that could aid CSIRT personnel with new techniques and methods. For more information, visit the official website: `https://htcia.org/`.
- **InfraGard**: For those CSIRT and information security practitioners in the United States, the Federal Bureau of Investigation has created a private-public partnership geared toward networking and information sharing. This partnership allows CSIRT members to share information about trends or discuss past investigations. We can find more information on the website: `https://www.infragard.org/`.
- **Law enforcement**: Law enforcement has seen an explosive growth in cyber-related criminal activity. In response, a great many law enforcement organizations have increased their capacity to investigate cyber crime. CSIRT leadership should cultivate a relationship with agencies that have cyber crime investigative capabilities. Law enforcement agencies can provide insight into specific threats or crimes being committed and provide CSIRTs with any specific information that concerns them.
- **Vendors**: External vendors can be leveraged in the event of an incident and what they can provide is often dependent on the specific line of business the organization has engaged them in. For example, an organization's IPS/IDS solution provider could assist with crafting custom alerting and blocking rules to assist in the detection and containment of malicious activity. Vendors with threat intelligence capability can also provide guidance on malicious activity indicators. Finally, some organizations will need to engage vendors who have a specialized incident response specialty such as reverse engineering malware when those skills fall outside an organization's capability.

Depending on the size of the organization, it is easy to see how the CSIRT can involve a number of people. It is critical to putting together the entire CSIRT that each member is aware of their roles and responsibilities. Each member should also be asked for specific guidance on what expertise can be leveraged during the entire incident response process. This becomes more important in the next part of the incident response framework, which is the creation of an incident response plan.

The incident response plan

With the incident response charter written and the CSIRT formed, the next step is to craft the incident response plan. The incident response plan is the document that outlines the high-level structure of an organization's response capability. This is a high-level document that serves as the foundation of the CSIRT. The major components to the incident response plan are as follows:

- **Incident response charter**: The incident response plan should include the mission statement and constituency from the incident response charter. This gives the plan continuity between the inception of the incident response capability and the incident response plan.
- **Expanded services catalog**: The initial incident response charter had general service categories with no real detail. The incident response plan should include specific details of what services the CSIRT will be offering. For example, if forensic services are listed as part of the service offering, the incident response plan may state that forensic services include the evidence recovery from hard drives, memory forensics, and reverse engineering potentially malicious code in support of an incident. This allows for the CSIRT to clearly delineate between a normal request, say for the searching of a hard drive for an accidentally deleted document not related to an incident, and the imaging of a hard drive in connection with a declared incident.
- **CSIRT personnel**: As outlined before, there are a great many individuals who comprise the CSIRT. The incident response plan will clearly define these roles and responsibilities. Organizations should expand out from just a name and title and define exactly the roles and responsibilities of each individual. It is not advisable to have a turf war during an incident, and having the roles and responsibilities of the CSIRT personnel clearly defined goes a long way to reducing this possibility.

- **Contact list**: An up-to-date contact list should be part of the incident response plan. Depending on the organization, the CSIRT may have to respond to an incident 24 hours a day. In this case, the Incident Response Plan should have primary and secondary contact information. Organizations can also make use of a rotating *on-call* CSIRT member who could serve as the first contact in the event of an incident.
- **Internal communication plan**: Incidents can produce a good deal of chaos as personnel attempt to ascertain what is happening, what resources they need, and who to engage to address the incident. The incident response plan internal communication guidance can address this chaos. This portion of the plan addresses the flow of information upward and downward between senior leadership and the CSIRT. Communications sideways between the CSIRT core and support personnel should also be addressed. This limits the individuals who are communicating with each other and cuts down on potentially conflicting instructions.
- **Training**: The incident response plan should also indicate the frequency of training for CSIRT personnel. At a minimum, the entire CSIRT should be put through a tabletop exercise at least annually. In the event that an incident post-mortem analysis indicates a gap in training, that should also be addressed within a reasonable time after conclusion of the incident.
- **Maintenance**: Organizations of every size continually change. This can include changes to infrastructure, threats, and personnel. The incident response plan should address the frequency of reviews and updates to the incident response plan. For example, if the organization acquires another organization, the CSIRT may have to adjust service offerings or incorporate specific individuals and their roles. At a minimum, the incident response plan should be updated at least annually. Individual team members should also supplement their skills through individual training and certifications through such organizations as SANS or on specific digital forensic tools. Organizations can incorporate lessons learned from any exercises conducted into this update.

Incident classification

Not all incidents are equal in their severity and threat to the organization. For example, a virus that infects several computers in a support area of the organization will dictate a different level of response than an active compromise of a critical server. Treating each incident the same will quickly burn out a CSIRT as they will have to respond the same way to even minor incidents. As a result, it is important to define within the incident response plan an incident classification schema. By classifying incidents and gauging the response, organizations make better use of the CSIRT and ensure that they are not all engaged in minor issues. The following is a sample classification schema:

- **High-level incident**: A high-level incident is an incident that is expected to cause significant damage, corruption, or loss of critical and/or strategic company or customer information. A high-level incident may involve widespread or extended loss of system or network resources. The event can have potential damage and liability to the organization and to the corporate public image. Examples of high-level incidents include, but are not limited to, the following:
    - Network intrusion
    - Physical compromise of information systems
    - Compromise of critical information
    - Loss of computer system or removable media containing un-encrypted confidential information
    - Widespread and growing malware infection (more than 25% of hosts)
    - Targeted attacks against the IT infrastructure
    - Phishing attacks using the organization's domain and branding
- **Moderate-level incident**: A moderate-level incident is an incident that may cause damage, corruption, or loss of replaceable information without compromise (there has been no misuse of sensitive customer information). A moderate-level event may involve significant disruption to a system or network resource. It also may have an impact on the mission of a business unit within the corporation:
    - Anticipated or ongoing DoS attack.
    - Loss of computer system or removable media containing un-encrypted confidential information.
    - Misuse or abuse of authorized access.
    - Automated intrusion.
    - Confined malware infection.
    - Unusual system performance or behavior.
    - Installation of malicious software.

    - Suspicious changes or computer activity.
    - Playbooks can be configured in a number of ways. For example, a written document can be added to the incident response plan for specific types of incidents. Other times, organizations can use a flow diagram utilizing software such as iStudio or Visio. Depending on how the organization chooses to document the playbook, they should create 10-20 that address the range of potential incidents.
- **Low-level incident**: A low-level incident is an incident that causes inconvenience and/or unintentional damage or loss of recoverable information. The incident will have little impact to the corporation:
    - Policy or procedural violations detected through compliance reviews or log reviews
    - Lost or stolen laptop or other mobile equipment containing encrypted confidential information
    - Installation of unauthorized software
    - Malware infection of a single PC
- **Incident tracking**: Tracking incidents is a critical responsibility of the CSIRT. During an incident, all actions taken by the CSIRT and other personnel during an incident should be noted. These actions should be recorded under a unique incident identifier.

For organizations that have limited resources and experience a limited number of incidents per year, most IT ticketing systems are sufficient for tracking incidents. The drawback to this method is that these systems generally lack an incident response focus and do not have additional features that are designed to support incident response activities. Larger organizations that have a higher frequency of incidents may be best served by implementing a purpose-designed incident response tracking system. These systems allow for integration of evidence collection and incident playbooks.

The incident response playbook

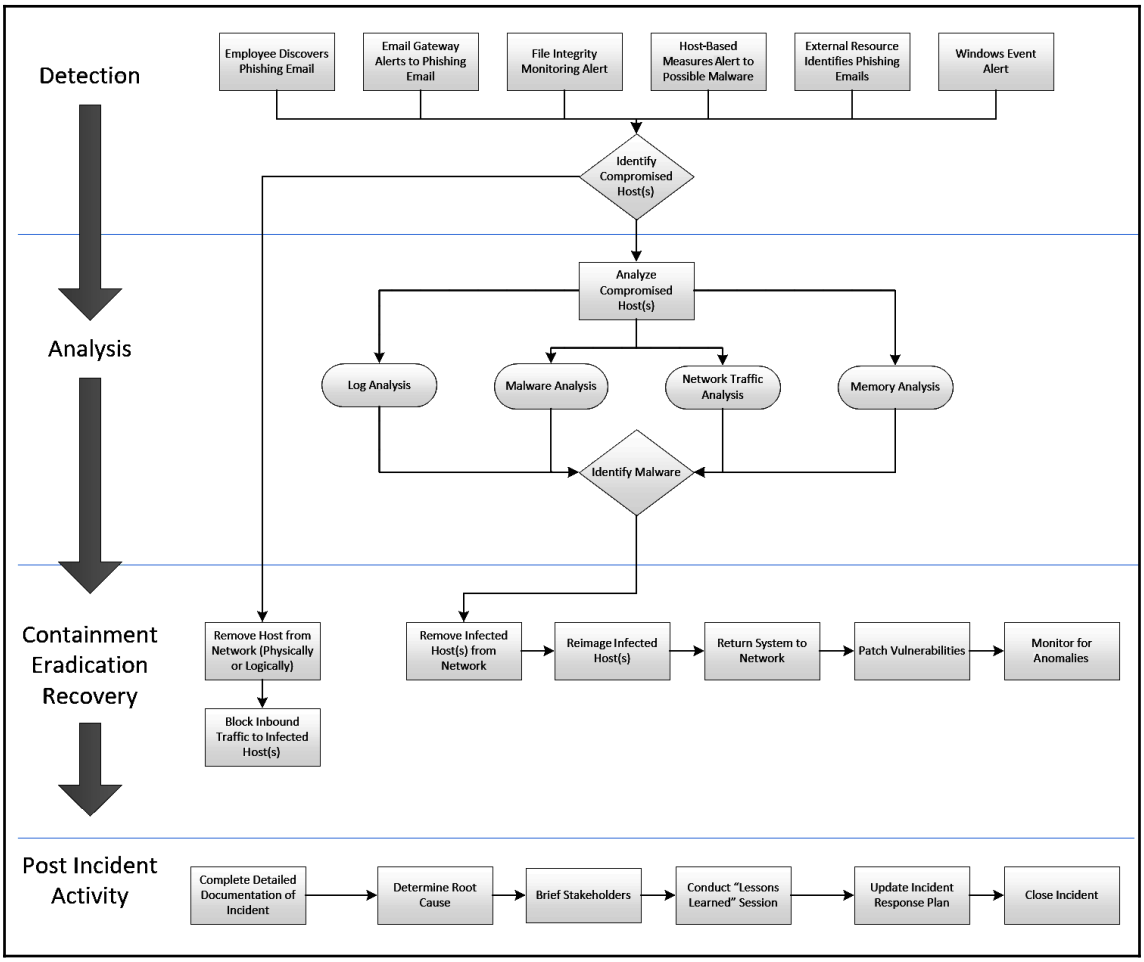
One key aspect of the incident response plan is the use of playbooks. An incident response playbook is a set of instructions and actions to be performed at every step in the incident response process. The playbooks are created to give organizations a clear path through the process, but with a degree of flexibility in the event that the incident under investigation does not fit neatly into the box.

A good indicator of which playbooks are critical is the organization's risk assessment. Examining the risk assessment for any threat rated critical or high will indicate which scenarios need to be addressed via an incident response playbook. Most organizations would identify a number of threats, such as a network intrusion via a zero-day exploit, ransomware, or phishing as critical, requiring preventive and detective controls. As the risk assessment has identified those as critical risks, it is best to start the playbooks with those threats.

For example, let's examine the breakdown of a playbook for a common threat, social engineering. For this playbook, we are going to divide it out into the incident response process that was previously discussed.

- **Preparation**: In this section, the organization will highlight the preparation that is undertaken. In the case of phishing, this can include employee awareness to identify potential phishing email or the use of an email appliance that scans attachments for malware.
- **Detection**: For phishing attacks, organizations are often alerted by aware employees or through email security controls. Organizations should also plan on receiving alerts via malware prevention or **Host Intrusion Prevention System** (**HIPS**) controls.
- **Analysis**: If an event is detected, analyzing any evidence available will be critical to classifying and appropriately responding to an incident. In this case, analysis may include examining the compromised host's memory, examining event logs for suspicious entries, and reviewing any network traffic going to and from the host.
- **Containment**: If a host has been identified as compromised, it should be isolated from the network.
- **Eradication**: In the event that malware has been identified, it should be removed. If not, the playbook should have an alternative such as reimaging with a known good image.
- **Recovery**: The recovery stage includes scanning the host for potential vulnerabilities and monitoring the system for any anomalous traffic.
- **Post-incident activity**: The playbook should also give guidance on what actions should take place after an incident. Many of these actions will be the same across the catalog of playbooks, but are important to include, ensuring that they are completed in full.

The following diagram is a sample playbook for a phishing attack. Note that each phase of the incident response cycle is addressed as well as specific actions that should be taken as part of the response. Additionally, organizations can break specific actions down, such as through log analysis for a certain playbook, for greater detail:



Playbooks are designed to give the CSIRT and any other personnel a set of instructions to follow in an incident. This allows for less time wasted if a course of action is planned out. Playbooks serve as a guide and they should be updated regularly, especially if they are used in an incident and any key pieces or steps are identified. It should be noted that playbooks are not written in stone and are not a checklist. CSIRT personnel are not bound to the playbook in terms of actions and should be free to undertake additional actions if the incident requires it.

Escalation procedures

A critical component of the incident response plan is the escalation procedures. Escalation procedures outline who is responsible from moving an event or series of events from just anomalies in the information system to an incident. The CSIRT will become burned out if they are sent to investigate too many false positives. The escalation procedures ensure that the CSIRT is effectively utilized and that personnel are only contacted if their expertise is required.

The procedures start with the parties who are most likely to observe anomalies or events in the system that may be indicative of a larger incident. For example, the help desk may receive a number of calls that indicate a potential malware infection. The escalation procedures may indicate that if malware is detected and cannot be removed via malware prevention controls, they are to contact the CSIRT member on call. That CSIRT member will then take control. If they are able to contain the malware to that single system and identify the infection vector, they will attempt to remove the malware and, barring that, have the system reimaged and redeployed. At that point, the incident has been successfully concluded. The CSIRT member can document the incident and close it out without having to engage any other resources.

Another example where the escalation moves farther up into an all-out CSIRT response can start very simply with an audit of active directory credentials. In this case, a server administrator with access management responsibilities is conducting a semi-annual audit of administrator credentials. During the audit, they identify three new administrator user accounts that do not tie to any known access rights. After further digging, they determine that these user accounts were created within several hours of each other and were created over a weekend. The server administrator contacts the CSIRT for investigation.

The CSIRT analyst looks at the situation and determines that a compromise may have happened. The CSIRT member directs the server administrator to check event logs for any logins using those administrator accounts. The server administrator identifies two logins, one on a database server and another on a web server in the DMZ. The CSIRT analyst then directs the network administrator assigned to the CSIRT to examine network traffic between the SQL database and the web server. Also, based on the circumstances, the CSIRT analyst escalates this to the CSIRT coordinator and informs them of the situation. The CSIRT coordinator then begins the process of engaging other CSIRT core team and technical support members to assist.

After examining the network traffic, it is determined that an external threat actor has compromised both systems and is in the process of exfiltrating the customer database from the internal network. At this point, the CSIRT coordinator identifies this as a high-level incident and begins the process of bringing support personnel into a briefing. As this incident has involved the compromise of customer data, the CSIRT support personnel such as marketing or communications and legal need to become involved. If more resources are required, the CSIRT coordinator will take the lead on making that decision.

The escalation procedures are created to ensure that the appropriate individuals have the proper authority and training to call upon resources when needed. The escalation procedures should also address the involvement of other personnel outside the core CSIRT members based on the severity of the incident. One of the critical functions of the escalation procedures is to clearly define what individuals have the authority to declare anomalous activity an incident. The escalation procedures should also address the involvement of other personnel outside the core CSIRT members, based on the severity of the incident.

Testing the incident response framework

So far, there have been a number of areas that have been addressed in terms of preparing for an incident. From an initial understanding of the process involved in incident response, we moved through the creation of an incident response plan and associated playbooks.

Once the capability has been created, it should be run through a table-top exercise to flush out any gaps or deficiencies. This exercise should include a high-level incident scenario that involves the entire team and one of the associated playbooks. A report that details the results of the table-top exercise and any gaps, corrections, or modifications should also be prepared and forwarded to the senior leadership. Once leadership has been informed and acknowledges that the CSIRT is ready to deploy, it is now operational.

As the CSIRT becomes comfortable executing the plan under a structured scenario, they may want to try more complex testing measures. Another option that is available is the Red/Blue or Purple Team Exercise. This is where the CSIRT is tasked with responding to an authorized penetration test. Here, the team is able to execute against a live adversary and test the plans and playbooks. This significantly increases the value of the penetration test as it provides both insight into the security of the infrastructure as well as the ability for the organization to respond appropriately.

Regardless of the makeup of the team, another key component of CSIRT deployment is the inclusion of regular training. For CSIRT core members, specific training on emerging threats, forensic techniques, and tools should be ongoing. This can be facilitated through third-party training providers or, if available, in-house training. The technical support members of the CSIRT should receive regular training on techniques and tools available. This is especially important if these members may be called upon during an incident to assist with evidence collection or remediation activities. Finally, the other support members should be included in the annual test of the incident response plan. Just as with the inaugural test, the organization should pick a high-level incident and work through it using a tabletop exercise. Another option for the organization is to marry up the test of their incident response plan with a penetration test. If the organization is able to detect the presence of the penetration test, they have the ability to run through the first phases of the incident and craft a tabletop for the remaining portions.

One final component to the ongoing maintenance of the incident response plan is a complete annual review. This annual review is conducted to ensure that any changes in personnel, constituency, or mission that may impact other components of the plan are addressed. In addition to a review of the plan, a complete review of the playbooks is conducted as well. As threats change, it may be necessary to change existing playbooks or add new ones. The CSIRT personnel should also feel free to create a new playbook in the event that a new threat emerges. In this way, the CSIRT will be in a better position to address incidents that may impact their organization. Any major changes or additions should also trigger another table-top exercise to validate the additional plans and playbooks.

Summary

Benjamin Franklin is quoted as saying, *"By failing to prepare, you are preparing to fail."* In many ways, this sentiment is quite accurate when it comes to organizations and the threat of cyber attacks. Preparing for a cyber attack is a critical function that must be taken as seriously as any other aspect of cyber security. Having a solid understanding of the incident response process to build on with an incident response capability can provide organizations with a measure of preparation, so that in the event of an incident, they can respond. Keep in mind as we move forward that forensic techniques, threat intelligence, and reverse engineering are there to assist an organization to get to the end, that is, back up and running.

This chapter explored some of the preparation that goes into building an incident response capability. Selecting a team, creating a plan, and building the playbooks and the escalation procedures allows a CSIRT to effectively address an incident. The CSIRT and associated plans give structure to the digital forensic techniques to be discussed. This discussion begins with the next chapter, where proper evidence handling and documentation is the critical first step in investigating an incident.

Questions

1. A table-top exercise should be conducted after changes are made to the incident response plan and/or playbooks.

   A) True
   B) False

2. Which of the following roles would not be a member of the CSIRT core team?

   A) Incident coordinator
   B) CSIRT analyst
   C) Legal

3. It is not important to have technical resources available as part of the incident response framework to aid during an incident.

   A) True
   B) False

4. The risk assessment is a valid data source for identifying high-risk incidents for playbook creation.

   A) True
   B) False

Further reading

- *Computer Security Incident Handling Guide*, NIST SP 800-61 Rev 2 : `https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final`
- ENISA *Incident Handling in Live Role Play Handbook*: `https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/documents/incident-handling-in-live-role-playing-handbook/view`
- *Incident Handler's Handbook* by Patrick Kral, SANS Reading Room: `https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901`

2
Managing Cyber Incidents

The incident response framework detailed in the previous chapter provided the specific structure of a **Computer Security Incident Response Team** (**CSIRT**), and how the CSIRT will engage with other business units. The chapter further expanded on the necessary planning and preparation an organization should undertake to address cyber incidents. Unfortunately, planning and preparation cannot address all the variables and uncertainties inherent in cyber incidents.

As the boxer Mike Tyson said:

> *"Everyone has a plan until they get hit in the face."*

This chapter will focus on executing those plans and frameworks detailed in `Chapter 1`, *Understanding Incident Response*, to properly manage a cyber incident. A solid foundation in and an understanding of cyber incident management allows organizations to put their plans into action more efficiently, communicate with key stakeholders in a timely manner and, most importantly, lessen the potential damage or downtime of a cyber incident.

This chapter will address how to manage a cyber incident, examining the following topics:

- Engaging the incident response team
- Incorporating crisis communications
- Investigating incidents
- Incorporating containment strategies
- Getting back to normal: eradication and recovery

Engaging the incident response team

A CSIRT functions in much the same way as an urban or rural fire department. A fire department has specifically trained professionals who are tasked with responding to emergency situations with specialized equipment to contain and eradicate a fire. In order to engage a fire department, a citizen must contact emergency services and provide key information, such as the nature of the emergency, the location, and if there are any lives in danger. From here, that information is passed on to the fire department, which dispatches resources to the emergency.

The process of engaging a CSIRT is very similar to engaging a fire department. Internal or external personnel need to escalate indications of a cyber security incident to the appropriate personnel. From here, resources are dispatched to the appropriate location/s, where those on the ground will take the lead in containing the incident, and eradicating or limiting potential downtime or loss of data. To make this process as efficient as possible, the following are critical components of the engagement process:

- CSIRT models provide a framework that places the CSIRT and the associated escalation procedures within the organizational structure.
- A war room describes the location from which the CSIRT manages the incident.
- Communications address the ability of the CSIRT to communicate properly.
- Staff rotation examines the need to rest personnel during a prolonged incident.

Engaging a CSIRT, much like a fire department, requires a set path of escalation. In the following sections, there are three CSIRT models that describe some options when looking at a proper escalation.

CSIRT models

How an organization escalates incidents to a CSIRT is largely dependent on how it is structured. Organizations configure their individual CSIRT to best fit their structure and resources. The following three basic structures can serve as a guide for placing the CSIRT within the most suitable part of the organization to facilitate a speedy escalation, as well as capture as many details of the incident as possible, in order for a proper investigation to take place.

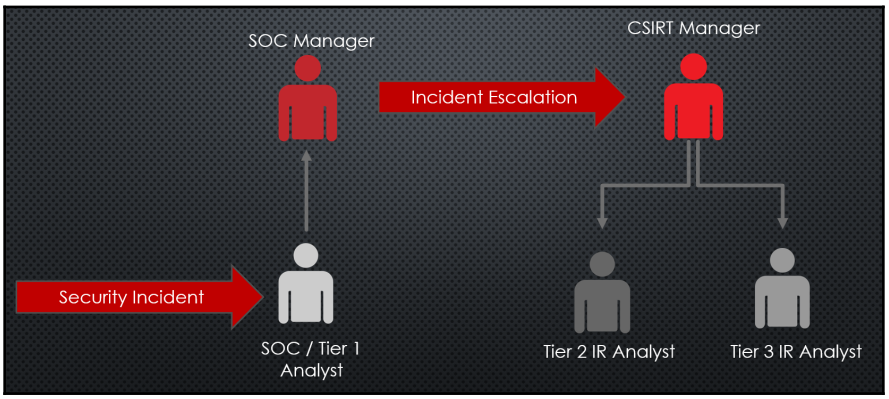
Security Operations Center escalation

In this organizational model, the **Security Operations Center** (**SOC**) is responsible for handling the initial incident detection or investigation. In general, the SOC is responsible for the management of the security tools that monitor the network infrastructure. It has direct access to event management, intrusion prevention and detection, and antivirus systems. From here, it is able to view events, receive and review alerts, and process other security-related data.

SOC escalation is a common model among organizations that have a dedicated SOC, either through in-house personnel or through a third-party **Managed Security Service Provider** (**MSSP**). In this model, there are clearly defined steps, from the initial notification to the escalation, as follows:

1. An alert is received by the SOC or Tier 1 analyst.
2. The SOC or Tier 1 analyst then determines whether the alert meets the criteria for an incident.
3. When a potential incident has been identified, the analyst performs an initial investigation.
4. If warranted, the analyst will then escalate the incident to the SOC manager.
5. After a review by the SOC manager, the incident is escalated to the CSIRT manager to address the incident.

The following diagram shows the flow of incident escalation from the **SOC manager** to the **CSIRT manager**:



In this model, there are several issues of concern that need to be addressed by the CSIRT and SOC personnel, as follows:

- First, engaging the CSIRT in this manner creates a situation where there are several individuals handling an incident before the CSIRT is fully engaged.
- Second, if the incident escalation is not properly documented, the CSIRT manager would have to engage the SOC manager for clarification or additional information, thereby increasing the time taken to properly address an incident.
- Third, the SOC personnel require training to determine which observed events constitute an incident and which may be false positives. The CSIRT may suffer from burnout and become weary of the SOC chasing up false incidents.
- Finally, communication between the SOC and the CSIRT needs to be clear and concise. Any gap in their ability to share information in real time will cause additional confusion.

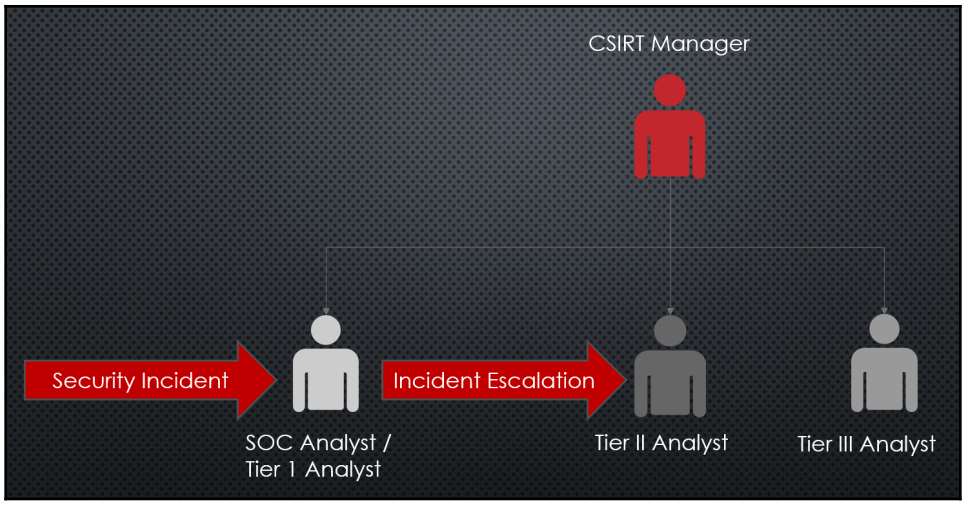
Another variation of this model, common within organizations without a dedicated SOC, is where an initial security incident is received by either a helpdesk or a network operations center. This adds further complexity in terms of engaging the CSIRT in a timely manner, as such personnel are often not trained to address incidents of this nature.

The best practice in a case like this is to have several of the personnel on these teams trained in cyber security analysis, to address initial triage and a proper escalation.

SOC and CSIRT combined

To limit some of the drawbacks with the SOC escalation model, some organizations embed the SOC within the overall CSIRT team. Placing the SOC in such a structure may prove to be a more efficient fit since the SOC has responsibility for the initial alerting and triaging function, which is directly related to the CSIRT.

In this model, the SOC analyst serves as the first tier. As previously discussed, they have the first view of security events or security control alerts. After processing and triaging the alert, they have the ability to immediately escalate the incident to the Tier 2 analyst, without having to engage a manager who would then escalate it to the CSIRT manager. This process is highlighted in the following diagram:

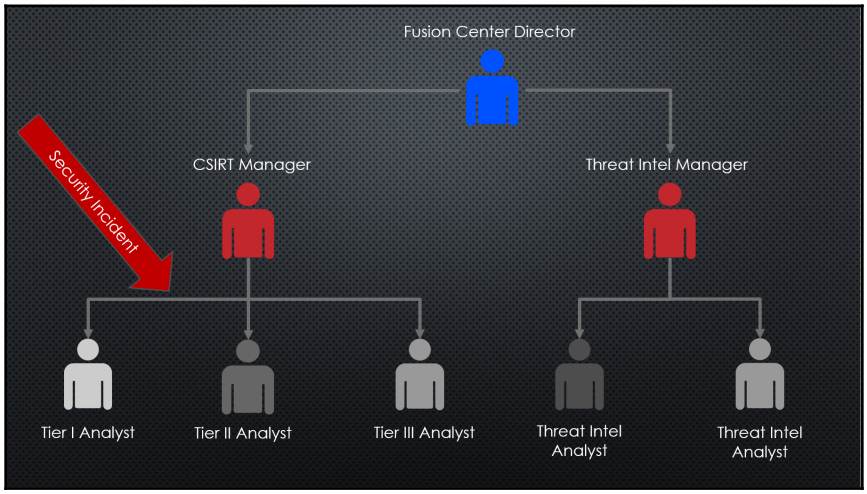


This model has some distinct advantages over the previous one. First, the CSIRT has a greater degree of visibility into what the SOC is seeing and doing. Further, having the SOC embedded within the CSIRT allows the CSIRT manager and their team to craft more efficient policies and procedures related to incidents. A second, distinct, advantage of this approach is that the incident escalation is completed much faster and, more likely, with greater precision. With the SOC analyst having a direct escalation to the next tier of CSIRT personnel, the entire process is much faster, and a more detailed analysis is performed as a result.

This approach works well in organizations with a dedicated SOC center that is in-house and not outsourced. For organizations making use of a network operations center or a helpdesk, and without a dedicated SOC, this approach is not realistic, as those functions are often managed outside of the CSIRT, or even network security teams. One other issue is that, depending on the size of the SOC and CSIRT teams, additional CSIRT managers may be required, in order to address the day-to-day workload of both the SOC and the CSIRT.

CSIRT fusion center

As threat intelligence becomes an increasing part of daily security operations, one organizational structure that addresses this trend is the CSIRT fusion center. In this case, the CSIRT analysts, SOC analysts, and threat intelligence analysts are teamed up together, within a single team structure. This merges the elements of an SOC- and CSIRT-combined structure with dedicated threat intelligence analysts. In such a scenario, the threat intelligence analysts would be responsible for augmenting incident investigations with external and internal resources related to the incident. They could also be leveraged for detailed analysis in other areas related to the incident. The following diagram shows the workflow from the **Fusion Center Director** to the various personnel responsible for incident management:



As organizations continue to develop threat intelligence resources within their security operations, this model allows the CSIRT to make use of that capability, without having to create new processes. `Chapter 13`, *Leveraging Threat Intelligence,* will discuss threat intelligence in depth, and how this capability may enhance incident investigations.

The CSIRT fusion center is not widely deployed, largely because threat intelligence integration is a relatively new methodology, as well as being resource-intensive. Very few organizations have the resources in either technology or personnel to make this type of structure effective. Pulling in full-time threat intelligence analysts, as well as various paid and open source feeds (and the technology to support them), is often cost-prohibitive. As a result of this, there are not many organizations that can leverage a full-time threat intelligence analyst as part of their CSIRT capability.

The war room

Another consideration when engaging a CSIRT is the need to have a single location from which the CSIRT can operate. There are several terms in use for the physical location a CSIRT can operate from, such as a SOC or a crisis suite, but a simple term is a war room. A war room can be set up as is necessary, or, in some instances, a dedicated war room is set aside. In the former case, an existing meeting room is purposed as the war room for the duration of the entire incident. This is often the preferred option for those organizations that do not have a high enough number of incidents to necessitate a dedicated war room. For those organizations that experience a higher number of incidents or more complex incidents, there may be a need to create a dedicated war room.

The war room should have the following capabilities to facilitate a more orderly response to incidents:

- **Workspaces**: Each member of the CSIRT core team should have a dedicated workspace in which to perform analysis. Workspaces should include network connectivity, power, and monitors.
- **Team displays**: One of the frustrations CSIRT team members may encounter during an incident is the inability to share the output of the analysis. An overhead projector or a large screen can facilitate better sharing of data across the entire team.
- **Note sharing**: Along the lines of sharing data through team displays, there may also be the need to share information among teams that are geographically dispersed. This may also be facilitated through the use of collaboration tools such as OneNote, SharePoint, or a wiki created for the incident.
- **Whiteboards**: There is a good deal of information flowing in and out of a war room. Data related to assignments and running lists of compromised systems are best left on a whiteboard so that they are clear to everyone.
- **Limited access**: The CSIRT should limit access to a war room to only those personnel who have a legitimate need to enter. Limiting access to this area prevents information from falling into the wrong hands.

Communications

One area of consideration that is often overlooked by organizations is how to communicate within a larger organization during an incident. With email, instant messaging, and voice calls, it may seem as though organizations already have the necessary tools to appropriately communicate internally. These communications platforms may need to be set aside in the event of an incident impacting user credentials, email systems, or other cloud-based collaboration platforms. For example, a common attack being observed is compromises of the Office 365 cloud-based email. If attackers have gained access to the email system, they may have also compromised associated instant messaging applications, such as Skype. Given this, relying on such applications during an incident may, in fact, be providing the attackers with an insight into the actions of the CSIRT.

If it is suspected that these applications have been compromised, it is critical to have a secondary—and even tertiary—communications option. Commercially acquired cell phones are often a safe alternative. Furthermore, CSIRT members may leverage free or low-cost collaboration tools, for a limited time. These can be leveraged until such time that the usual communications platforms are deemed safe for use.

Staff rotation

Prolonged incident investigations can begin to take their toll on CSIRT personnel, both physically and mentally. While it may seem prudent at the time to engage a team until an incident has been addressed, this can have a detrimental impact on the team's ability to function. Studies have shown the negative cognitive effects of prolonged work with little rest. As a result, it is imperative that the **incident commander** (**IC**) places responders on shifts after a period of time has passed.

For example, approximately 24 hours after an incident investigation has been started, it will become necessary to start rotating personnel so that they have a rest period of 8 hours. This also includes the IC. During a prolonged incident, an alternative IC should be named, to ensure continuity and that each of the ICs gets the appropriate amount of rest.

Another strategy is to engage support elements during a period of inactivity in an incident. These periods of inactivity generally occur when an incident has been contained and potential **command and control** (**C2**) traffic has been addressed. Support personnel can be leveraged to monitor the network for any changes, giving the CSIRT time to rest.

Incorporating crisis communications

The notion that serious security incidents can be kept secret has long passed. High-profile security incidents, such as those that impacted Target and TJX, have been made very public. Adding to this lack of secrecy are the new breach notification laws that impact organizations across the globe. The **General Data Protection Regulation** (**GDPR**) Article 33 has a 72-hour breach notification requirement. Other regulatory or compliance frameworks, such as the **Health Insurance Portability and Accountability Act (HIPAA)** Rule 45 CFR § § 164.400-414, stipulate notifications to be made in the event of a data breach. Compounding legal and regulatory communication pressures are communications that need to be made to internal business units and external stakeholders. While it may seem that crafting and deploying a communications plan during an incident is a waste of resources, it has become a necessity in today's legal and regulatory environment. When examining crisis communications, there are three focus areas that need to be addressed:

- Internal communications
- External communications
- Public notification

As each of these represents a specific audience, each requires different content and messaging.

Internal communications

Internal communications are those communications that are limited to the business or organization's internal personnel and reporting structure. There are several business units that need to be part of communications. The legal department will need to be kept abreast of the incident, as they will often have to determine reporting requirements and any additional regulatory requirements. Marketing and communications can be leveraged, for crafting communications to external parties. This can best be facilitated by including them as early as possible in the process so that they have a full understanding of the incident and its impact. If the incident impacts any internal employees, Human Resources should also be included as part of internal communications.

One of the critical groups that are going to want to be informed as the incident unfolds is the C-suite and, more specifically, the CEO. A CSIRT will often fly well below the line of sight of senior leadership until there is a critical incident. At that point, the CEO will become very interested in the workings of the CSIRT and how they are addressing the incident.

With all of these parties needing to be kept in the loop, it is critical to ensure orderly communications and to limit misinformation. To limit confusion, the IC or CSIRT team lead should serve as a single point of contact. This way, for example, the legal department does not contact a CSIRT analyst and receive information about the investigation that is, at that time, speculative. Reliance on that type of information can lead to serious legal consequences. To keep everyone informed, the CSIRT team lead or IC should conduct periodic updates throughout each day of the incident. The cadence of such communications is dependent on the incident type and severity, but having a cadence of every 4 hours, with a conference call during the working period of 6 a.m. to 10 p.m., will ensure that everyone is kept up to date.

In addition to a regular conference call, the CSIRT team lead or the IC should prepare a daily status report, to be sent to senior leadership. This daily status report does not have to be as comprehensive and detailed as a digital forensics report but should capture significant actions taken, any incident-related data that has been obtained, and any potential factors that may limit the ability of the CSIRT to function. At a minimum, a daily status meeting, in conjunction with this report, should be conducted with senior leadership and any other personnel that are required to be in attendance during the course of the incident.

External communications

Incidents may have a downstream impact on other external entities outside the organization that is suffering the incident. Some of these external entities may include suppliers, customers, transaction processing facilities, or service providers. If any of these organizations have a direct link—such as a **virtual private network** (**VPN**)—to the impacted organization, external partners need to be informed sooner rather than later. This is to limit any possibility that an attacker has leveraged this connection to compromise other organizations.

A significant area of concern when addressing incident management and external communications for **managed service providers** (**MSPs**) is the trend of attackers targeting MSPs first, with the intent of using them as a jumping-off point into other organizations through established VPNs. One perfect example of this is the Target breach, where attackers compromised a **heating, ventilation, and air conditioning** (**HVAC**) vendor as the initial point of entry. Attackers are using this tried-and-true method of attacking MSPs using ransomware, now with the intent of compromising more than one organization per attack.

At a minimum, an organization should inform external parties that they are dealing with an incident and, as a precaution, the connection will be blocked until the incident has been addressed. This can then be followed up with additional information. Much like internal communications, setting a regular cadence may go a long way to smoothing out any damage to the relationship as a result of the incident. In some cases, well-trusted external parties may be made part of regular daily status updates.

Public notification

As discussed previously, there are several legal and compliance requirements that need to be taken into consideration when discussing the notification of customers or the general public about an incident. Organizations may have to ride a fine line in terms of complying with the requirements of regulations such as HIPAA, without disclosing operational details of an incident still under investigation. Compounding this pressure are the possible implications on stock value or the potential for lost business. With all these pressures, it is critical to craft a message that is within the legal or compliance requirements but that also limits the damage to the organization's reputation, revenue, or stock value.

While directly related to the incident at hand, the CSIRT should not be responsible for crafting a public notification statement. Rather, the CSIRT should be available to provide insight into the incident investigation and to answer any questions. The two best business units that should be involved in crafting a message are the legal and marketing departments. The marketing department would be tasked to craft a message to limit the fear of backlash from customers. The legal department would be tasked to craft a message that meets legal or regulatory requirements. The CSIRT should advise as far as possible, but these two business units should serve as the point of contact for any media or public questions.

Investigating incidents

The lion's share of this volume addresses the various methods that can be leveraged when investigating an incident. The primary goal of the CSIRT is to utilize methods that follow a systems analysis to address the following key facets of an incident:

- **Identifying the scope**: In some incidents, the actual scope may not be clearly defined at the initial detection stage. For example, an organization may be contacted by a law enforcement agency that has indicated a C2 server has been taken down. During an analysis of that system, the external IP address of the organization has been identified. From this data point, the scope is first defined as the entire network. From here, the CSIRT would analyze data from the firewall or web proxy, to identify the internal systems that were found to be communicating with the C2 server. From this data, they would narrow down the initial scope of the incident to those systems that had been impacted.

  When attempting to identify the scope of the incident, there is a drive to find the patient zero or the first system that was compromised. In some incidents, this may be easy to discover. A phishing email containing a PDF document that, when opened, executes malware can be easily identified by the user or by security control. Other attacks may not be so obvious. While finding patient zero does provide a good deal of data for root-cause analysis, it is more important to identify the scope of the incident first, rather than looking for a single system.

- **Identifying the impact**: Another key consideration is determining the impact of the incident. Those that have been exposed to the fundamental concepts of information security are well familiar with the CIA triad. The CIA triad represents the elements of security within an information system: confidentiality, integrity, and availability. Any breach or violation of security will have an impact on one or more of these elements. For example, a ransomware incident that impacts 15 production servers impacts the availability of the data on those systems. Impacts against availability related to the incident, either as a direct result or through adversary actions or the time necessary to respond and remediate, are important factors in determining the incident's impact. Other incidents, such as the theft of intellectual property, impact the confidentially of data. Finally, incidents involving unauthorized manipulation to source code or other data impact the integrity of that data. The following diagram highlights the CIA triad:

Understanding the potential impact an incident may have is important in making decisions concerning the resources that are allocated for a response. A **distributed denial-of-service** (**DDoS**) attack against a non-critical service on the web will not necessitate the same type of response resulting from credit card-harvesting malware within a retail payment infrastructure. The impact also has a direct bearing on compliance with laws and other regulations. Understanding the potential impact of an incident on compliance is critical in ensuring that a proper response is conducted.

- **Identifying the root cause**: The central question that IT professionals and managers will ask during, and especially after, an incident is: "How did this happen?" Organizations spend a great deal of money and resources to protect their infrastructure. If an incident has occurred that causes an impact, there will be a need to understand how it happened. A goal of an incident investigation is to determine what sequence of events, vulnerabilities, or other conditions was present that led to the incident and the impact. Often, the root cause of an incident is not a simple vulnerability, but a sequence of events and conditions that allowed an adversary to penetrate the security systems and conduct their attack. Through an investigation, these events and conditions can be identified so that they are corrected, or otherwise controlled.

- **Incident attribution**: One area of debate involved in an incident investigation is incident attribution. With attribution, the CSIRT or investigative body attempts to determine which organization was behind the attack. Incidents may be attributed to nation-state actors, criminal groups, or other cyber adversaries. While there is some importance to attribution from a threat intelligence perspective (`Chapter 13`, *Leveraging Threat Intelligence*, will address attribution, as it relates to incident response), resources are better off investigating or containing an incident. Attempting to ascertain the group or groups responsible for an attack is time-consuming, with few positive returns. If the organization's leadership is adamant about determining attribution, the best approach is to comprehensively document the incident and pass off the data to a third party that specifically addresses attribution. Such organizations often combine data from several incident investigations, to build a dossier on groups. If the data supplied matches these, they may be able to provide some context in terms of attribution.

Incorporating containment strategies

Containment strategies are the actions taken during an incident to limit damage to specific systems or areas of the network. It is critical for organizations to have prepared these in the event of an incident. The rise of ransomware that combines elements of viruses and worms that can quickly spread through an organization highlights the need to rapidly contain an outbreak before it impacts a great many systems. Compounding the challenge with containment is that many enterprise IT systems utilize a "flat" topology, whereby the bulk of systems can communicate with each other. In this type of environment, ransomware and other worms can quickly propagate via legitimate protocols, such as **Remote Desktop Services** (**RDS**) or through the **Server Message Block** (**SMB**), that were popular during the WannaCry ransomware campaign, which leveraged the EternalBlue vulnerability in the Windows OS SMB installation. For more information, visit `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144`.

In order to address containment, an organization should have a clear idea of the network topology. This type of network awareness can be achieved through outputs of network discovery tools, up-to-date network diagrams, system inventories, and vulnerability scans. This data should be shared with the CSIRT so that an overall view of the network can be achieved. From here, the CSIRT should coordinate containment plans with network operations personnel so that an overall containment strategy can be crafted, and the potential damage of an incident limited. Having network operations personnel as part of the technical support personnel goes a long way in ensuring this process is streamlined and that containment is achieved as quickly as possible.

One other aspect of how infrastructure is managed that has a direct impact on incident management is that of change management. Mature IT infrastructures usually have a well-documented and governed change management process in place. During an incident, though, the CSIRT and support personnel cannot wait for a change management authorization and a proper change window to implement changes. When exercising containment strategies, IT and organizational leadership should fully understand that changes are going to be made based on the incident. This does not absolve the CSIRT and IT personnel from exercising due care and ensuring that changes are well documented.

In terms of containing a malware outbreak such as a ransomware attack, there are several strategies that can be employed. Ideally, organizations should have some ability to isolate segments of the network from each other, but in the event that this is not possible, CSIRT and IT personnel can take one or more of the following measures:

- **Physical containment**: In this case, the physical connection to the network is removed from the system. This can be as simple as unplugging the network cable, disabling wireless access, or disabling the connection through the operating system. While this sounds simple, there are several factors that may make this strategy challenging for even the smallest organization. First, is the ability to physically locate the systems impacted? This may be a simpler task inside a data center where the impacted systems are in the same rack, but attempting to physically locate 20 to 30 desktops in a fairly corporate environment takes a great deal of effort. In the time that it may take to remove 20 systems from the network, the malware could have easily spread across to other systems. Further compounding the difficulty of physical containment is the challenge of addressing geographically diverse systems. Having a data center or other operation an hour's drive away would necessitate having an individual on that site to perform the physical containment. As can be seen, physically containing a malware outbreak or another incident can be very difficult if the scope of the incident is beyond the capability of the CSIRT to perform. Physical containment should be reserved for those incidents where the scope is limited and where the CSIRT personnel can immediately remove the systems from the network.

- **Network containment**: The network containment strategy relies heavily on the expertise of network engineers or architects. It is for this reason that they are often included as part of the technical support personnel within the CSIRT, and should be involved in any containment strategy planning. With this containment strategy, the network administrator(s) will be tasked with modifying switch configurations, to limit the traffic from infected systems on a subnet to other portions of the network. This containment strategy may require modification of configurations on individual switches or through the use of the management console. One aspect of this approach that needs to be addressed is how the organization handles change control. In many organizations, it is common practice to review any switch configuration changes as part of the normal change control process. There needs to be an exception written into that process to facilitate the rapid deployment of switch configuration changes during a declared incident. Network administrators should also ensure that any changes that are made are properly documented so that they can be reversed or otherwise modified during the recovery phase of an incident.
- **Perimeter containment**: The perimeter firewall is an asset well suited for containment. In some circumstances, the perimeter firewall can be utilized in conjunction with network containment in a Russian nesting-doll approach, where the CSIRT contains network traffic at the perimeter and works its way to the specific subnets containing the impacted systems. For example, malware will often download additional code or other packages via tools such as PowerShell. In the event that the CSIRT has identified the external IP address that is utilized by the malware to download additional packages, it can be blocked at the firewall, thereby preventing additional damage. From here, the CSIRT can then work backward from the perimeter to the impacted systems. The organization can then leave the rule in place until such time that it is deemed no longer necessary. As with network containment, it is important to address any change control issues that may arise from making changes to the firewall ruleset.

- **Virtual containment**: With the advent of cloud computing and virtualization, many organizations have at least partially moved systems such as servers from physical systems to virtualized systems. Virtualization provides a great deal of flexibility to organizations during normal operations but also has some advantages, in the event that an incident may need to be contained. First, hypervisor software such as VMware's ESXi platform can be utilized to remove the network connection to multiple systems at once. Organizations may also make use of virtual switching in much the same way as physical switches, in terms of containment. Finally, virtualization software allows for the pausing of systems during an incident. This is the preferred method, as suspending or pausing a virtual machine during an incident preserves a good deal of evidence that can be examined later.

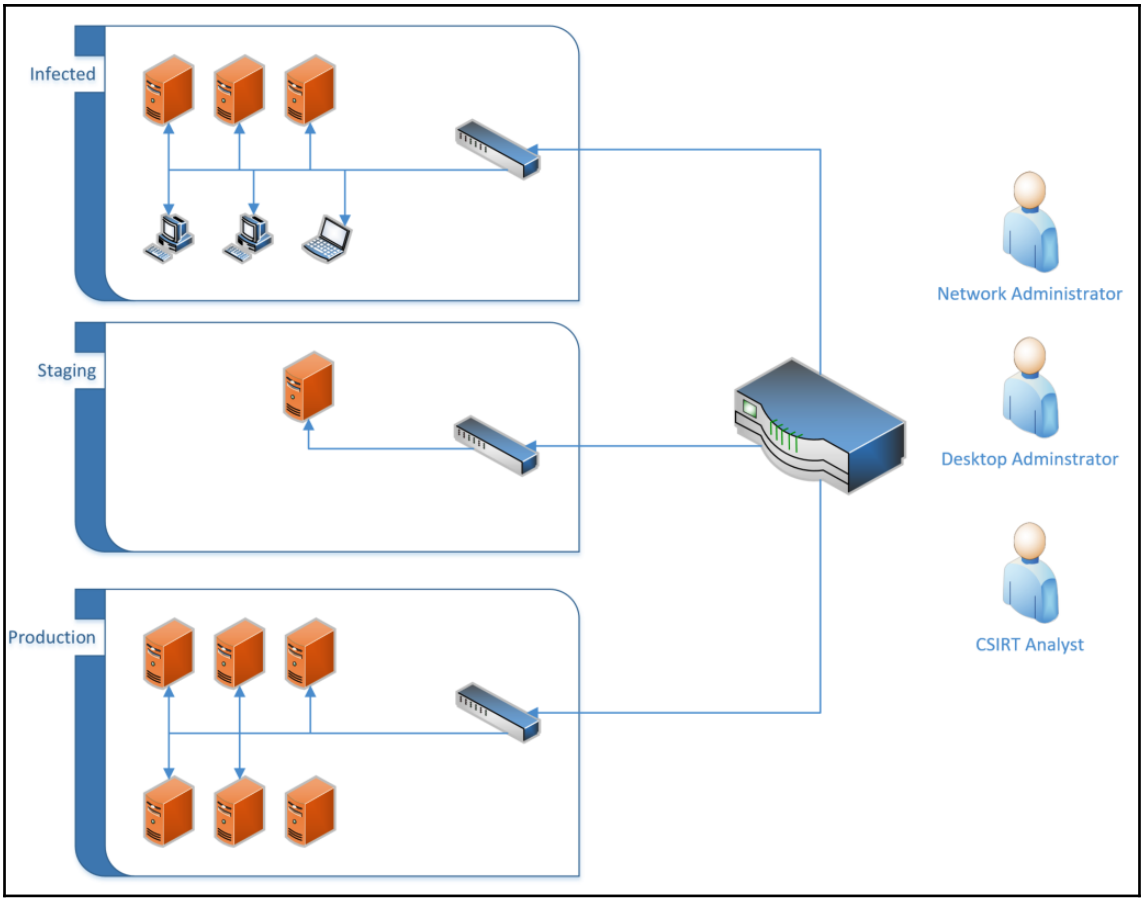
Getting back to normal – eradication and recovery

Once an incident has been properly and comprehensively investigated, it is time to move into the eradication and recovery phase. There may be a good deal of haste in getting to this stage, as there is a strong desire to return to normal operations. While there may be business drivers at play here, rushing eradication and recovery may reintroduce an unidentified compromised system that has been overlooked. In other scenarios, it could be possible to miss the patching of previously compromised systems, leaving them open to the same exploits that previously compromised them or, worse, placing a still-infected system back on the network. For this reason, both eradication and recovery strategies are addressed here.

Eradication strategies

The unfortunate reality with modern malware is that there is no surefire way to be 100 percent sure that all malicious code has been removed. In the past, organizations could simply scan the system with an antivirus program to have the offending malicious code. Now, with malware techniques such as process injection or DLL hijacking, even if the original code is removed, there is still a chance that the system is still infected. There is also the possibility that additional code that has been downloaded is also installed and will go undetected. As a result, most eradication strategies rely on taking infected machines and reimaging them with a known good image or reverting to a known good backup.

A strategy that is often employed in the cases of malware and ransomware is to make use of three separate **virtual LAN** (**VLAN**) segments and reimage the infected machines. First, all the infected machines are placed onto their own separate VLAN. From here, the CSIRT or system administrator will move one of the infected systems onto a secondary staging VLAN. The system is then reimaged with a known good image, or a known good backup is utilized. From here, once the system has been reimaged or has the backup installed, it is then moved to a production VLAN, where additional monitoring is conducted to ensure that there is no remaining infection or compromise. The following diagram shows a simple network structure that facilitates this recovery strategy:



While this method may be time-consuming, it is an excellent way to ensure that all systems that have been impacted have been addressed.

In the case of virtual systems, if the containment strategy previously discussed has been employed, the infected virtual systems have no network connectivity. From here, the most straightforward eradication strategy is to revert systems to the last-known good snapshot. Once the system has been restarted, it should be connected to the VLAN with enhanced monitoring. It is important that in the case of reverting to snapshots, the CSIRT has a great deal of confidence in the timeline. If the CSIRT is unsure about the timeline of an attack, there is the possibility that the snapshot may be compromised as well. This is especially true in organizations that conduct snapshots regularly.

Recovery strategies

In terms of recovery, there are several tasks that the CSIRT will need to manage to bring operations back to normal. The first of these is to ensure that all systems—not only those that have been through the eradication phase, but all systems—are properly patched with the most up-to-date patches. This is critical in those instances where the attacker has taken advantage of a zero-day exploit or a relatively new vulnerability. In cases where a patch is not forthcoming from the manufacturer, the CSIRT should recommend additional security controls to mitigate any residual risk.

A second piece of the recovery phase is for the CSIRT to work with IT and information security personnel in crafting additional detection and prevention alerts. During the examination of the evidence when determining the root cause, or in the containment phase, the CSIRT may have provided data for detection and prevention controls. The CSIRT should work with other personnel to augment those with additional detective and preventive rules. These additional rules may be specific to the incident or may pertain to specific vulnerabilities identified.

Third, any changes that were made to the infrastructure should be reviewed. These changes can be initially reviewed by the CSIRT and IT personnel, to determine if they are still required or can be removed. If changes are required in the long term, they should be evaluated by the organization's change control, and approved according to the change control process.

Fourth, before the incident can be closed out, it is good practice to conduct a full vulnerability scan of all systems. This is critical to ensure that any systems that have been compromised have been addressed. Additionally, this step will also address that any other systems that may not have been impacted by the security incident are nonetheless patched for any security vulnerabilities.

Finally, at the conclusion of an incident, it is important to conduct an after-action review. This review goes over the entire incident, from start to finish. All actions taken by the CSIRT personnel are reviewed. In addition, the plans and playbooks that were utilized are also reviewed in light of the incident actions. Any deficiencies, such as a lack of specific tools, training, or processes, should be brought up so that they may be corrected. The output of this after-action review should be documented as part of the overall incident documentation.

Summary

Planning for an incident is critical. Equally critical is properly managing an incident. This involves several elements that each CSIRT must address during the life of an incident. Proper logistics provides the necessary elements for the CSIRT to function. Having strategies to communicate incident information to leadership, third parties, and customers keeps those stakeholders informed, lessens speculation, and ensures compliance requirements are met. Incident investigation allows the CSIRT to properly identify the attack, identify the scope, and limit damage via a proper containment strategy. Finally, these elements are all part of eradicating an adversary's ability to access a network and helping an organization to return to normal. As was stated at the beginning: *Everyone has a plan until they get hit in the face*. The real value of a CSIRT to an organization is not in the plans and playbooks, but in how well they execute when an incident occurs. The following chapter will expand on the incident investigation portion of incident management, by providing the digital forensics framework to which CSIRT personnel adhere.

Questions

1. Which of the following containment strategies is the most difficult to perform?

   A) Physical
   B) Network
   C) Perimeter
   D) Virtual

2. A cyber security breach can have an impact on which of the following?

   A) Confidentiality
   B) Integrity
   C) Availability
   D) All of the above

3. Attribution is critical and has to be completed for a successful incident investigation.

   A) True
   B) False

Further reading

- NIST SP 800-61 Rev 2, *Computer Security Incident Handling Guide, at* `https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final`
- ENISA *Incident Handling in Live Role Playing Handbook*, at `https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/documents/incident-handling-in-live-role-playing-handbook/view`
- *Incident Handler's Handbook* by Patrick Kral, SANS Reading Room, at `https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901`
- MITRE *Ten Strategies of a World-Class Cybersecurity Operations Center*, at `https://www.mitre.org/sites/default/files/publications/pr-13-1028-mitre-10-strategies-cyber-ops-center.pdf`

3
Fundamentals of Digital Forensics

Forensic science can be defined as the application of scientific principles to legal matters. In an incident, **CSIRT** (short for **computer security incident response team**) members may be called upon to perform analysis on digital evidence acquired during the incident, utilizing digital forensics tools, techniques, and knowledge. To make certain that the evidence is processed correctly and can subsequently be admitted in a courtroom, digital forensics examiners need to understand the legal issues, along with the fine points, of the digital forensics process.

In this chapter, we will examine the legal statutes that impact the CSIRT and digital forensics examiners, as well as the rules that govern how evidence is admitted in court. To provide context to actions taken, we will also explore the digital forensics process and, finally, address the infrastructure necessary to incorporate a digital forensics capability within a CSIRT.

We will be covering the following topics in this chapter:

- Legal aspects
- Digital forensics fundamentals

Legal aspects

As we saw in `Chapter 1`, *Understanding Incident Response*, a proper incident response involves key individuals from a variety of disciplines. This highlights one frequently held misconception held: incident response is strictly a technology matter. One realm into which incident response falls heavily is the legal arena. There are a wide range of laws and regulations that directly impact an organization's incident response capability, ranging from breach notification to privacy. These laws provide a framework for governments to prosecute offenders, as well as providing strict rules concerning topics such as how evidence is handled and presented in court.

Laws and regulations

In the mid-1980s, as computer crime started to become more prevalent, jurisdictions began crafting laws to address ever-increasing instances of cybercrime. In the United States, for example, federal criminal law has specific statutes that deal directly with criminal activity when utilizing a computer, as follows:

- **18 USC § 1029—Fraud and related activity in connection with access devices**: This statute addresses the use of a computer to commit fraud. This is most often utilized by prosecutors in connection with cases where cybercriminals use a computer, or computers, to commit identity theft or other fraud-related activities.
- **18 USC § 1030—Computer Fraud and Abuse Act** (**CFAA**): Among the number of provisions within this law, the one most commonly associated with incident response is that of unauthorized access to a computer system. This law also addresses the illegality of **denial-of-service** (**DoS**) attacks.
- **Electronic Communications Privacy Act** (**ECPA**): This amendment to the Federal Wiretap Statute was enacted in 1986. It makes illegal the unauthorized interception of communications through electronic means, such as telecommunications and the internet. The ECPA was further amended by the **Communications Assistance for Law Enforcement Act** (**CALEA**). CALEA imposed the requirement on ISPs to ensure that their networks could be made available to law enforcement agencies, in order to conduct lawfully authorized surveillance.

Being familiar with the ECPA is critical for those organizations that have a presence in the United States. Provisions of the law make it a crime for an organization to conduct surveillance and capture traffic on networks, even those under their control, if the users have a reasonable expectation of privacy. This can lead to an organization being held liable for sniffing traffic on its own network if, in fact, its users have a reasonable expectation of privacy. For CSIRT members that , this creates potential legal problems if they access network resources or other systems. This can be easily remedied, by having all system users acknowledge that they understand their communications can be monitored by the organization and that they have no reasonable expectation of privacy in their communications when using computer and network resources provided by the organization.

- **Economic Espionage Act of 1996** (**EEA**): This law contains several provisions found in 18 USC § 1831-1839, and makes economic espionage and the theft of trade secrets a crime. This Act goes further than previous espionage legislation, as it deals directly with commercial enterprises and not just national security or government information.

Rules of evidence

Federal rules of evidence serve as the basis by which evidence can be admitted or excluded during a criminal or civil proceeding. Having knowledge of the following rules is important for CSIRT members so that any evidence collected is handled in a manner that prevents contamination and the possibility of the evidence being barred from being seen in court:

- **Rule 402—Test for Relevant Evidence**: This rule has two parts. First, the evidence to be admitted into the proceedings must have a tendency to make a fact more or less probable than it would be without the evidence. Second, the evidence (or the facts the evidence proves) is of consequence to the proceedings. This makes clear that not only should the evidence be relevant to the proceedings, but also it should prove or disapprove a facet of the case.

- **Rule 502—Attorney-Client Privilege and Work Product**: One of the most sacrosanct tenets of modern law is the relationship between a client and his/her attorney. One of the provisions of the attorney-client privilege is that what is said between the two is not admissible in court. This not only applies to spoken communications, but to written communications as well. In the world of digital forensics, reports are often written concerning actions taken and information obtained. Oftentimes, incident responders will be working directly for attorneys on behalf of their clients. As a result, these reports prepared in conjunction with an incident may fall under attorney work product rules. It is important to understand this when you work under the auspices of an attorney, and when these rules may apply to your work.
- **Rule 702—Testimony by Expert Witnesses**: Through the acquisition of experience and knowledge in digital forensics, an analyst may be allowed to testify as an expert witness. This rule of evidence outlines the specifics concerning expert witness testimony.
- **Rule 902—Evidence that is Self-Authenticating**: This rule has recently undergone a revision, as it relates to digital forensics. A new subpart has been added, as of December 1, 2017. This new subpart allows the verification of digital evidence integrity through hashing (we will discuss the role that hashing has in later chapters). Furthermore, this rule requires that a qualified person presents the evidence and that the evidence being presented has been collected according to best practices.
- **Rule 1002—Best Evidence Rule**: In civil or criminal proceedings, the original writings, recordings, or photographs need to be offered up as evidence, unless a reasonable exception can be made. In the physical realm, it is fairly easy to produce physical evidence. Parties to a case can easily present a knife used in an assault. It becomes a bit more complex when the evidence is essentially magnetic polarity on a hard drive, or log files that came from a router. In this case, courts have held that a forensically sound image of a hard drive is a reasonable substitute for the actual hard drive that was examined.
- **Rule 1003—Admissibility of Duplicates**: One of the most critical steps when conducting a forensic examination of digital media is to make an image or forensic copy of the media. This rule of evidence allows for such an image to be admitted into court. It is important to note that, if an image or forensic copy is to be admitted, the analyst who performed that action will most likely have to testify to having performed the action correctly.

Next, we will have a look at the fundamentals of Digital forensics.

Digital forensics fundamentals

As it was stated in the previous chapter, digital forensics is an important component of incident response. It is often the application of digital forensics methods that allows incident responders to gain a clear understanding of the chain of events that led to a malicious action, such as a compromised server or other data breach. For other incidents, such as internal fraud or malicious insider activity, digital forensics may provide the proverbial smoking gun that points to the guilty party. Before a detailed examination of the tools and techniques available to incident responders, it is critical to address the foundational elements of digital forensics. These elements provide not only context for specific actions, but also a method to ensure that evidence made part of an incident investigation is usable.

A brief history

Law enforcement first started to pay attention to the role that computers play in criminal activity in the mid-1980s. Prior to this, existing laws and law enforcement techniques were not adept at identifying and prosecuting computer criminals. As the use of computers by criminals began to gain more prominence, agencies such as the United States **Federal Bureau of Investigation** (**FBI)** decided to incorporate a dedicated digital and forensic investigation capability. This led to the creation of the FBI **Computer Analysis and Response Team** (**CART**). Other agencies, such as the Metropolitan Police Service, started to build a capability for investigating cybercrime.

An excellent historical document that addresses the FBI's CART is a short article in the United States Department of Justice *Crime Laboratory Digest*, dated January 1992: `https://www.ncjrs.gov/pdffiles1/Digitization/137561NCJRS.pdf`.

Two other seminal events brought the need for cyber investigations and forensics into the minds of many. The first was hacker Markus Hess broke into the Lawrence Berkeley National Laboratory. This break-in might have gone undetected had it not been for the efforts of Clifford Stoll, who hatched a plan to trap the attacker long enough to trace the connection. These efforts paid off, and Stoll, along with other authorities, was able to trace the hacker and eventually prosecute him for espionage. This entire episode is recorded in Stoll's book, *The Cuckoo's Egg*.

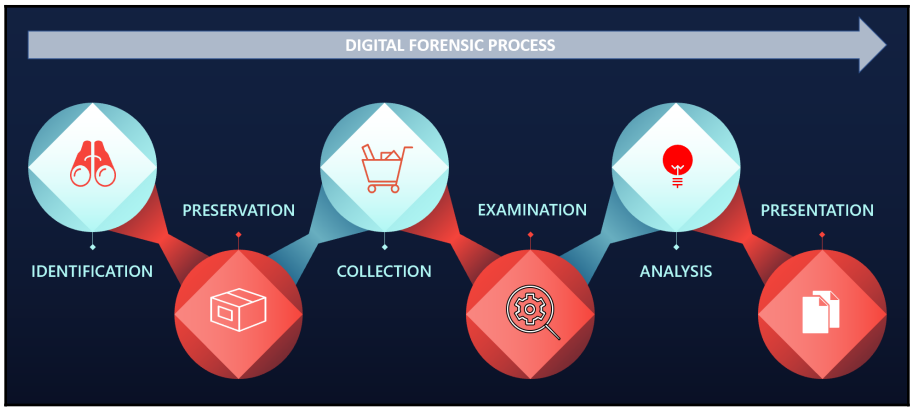
The second high-profile event was the Morris worm that was unleashed on the fledgling internet in 1988. The worm, created and released by Robert Tappan Morris, caused denial of service on a number of systems, subsequently causing damage in excess of $100,000. A post-incident investigation by a number of individuals, including Clifford Stoll, found at least 6,000 systems were infected. The rapid spread of the worm and the damage associated with it led to the creation of the Carnegie Mellon **CERT Coordination Center** (**CERT/CC**).

Throughout the 1990s, as more law enforcement agencies began to incorporate digital forensics into their investigative capabilities, the need for standardization of forensic processes became more apparent. In 1993, an international conference was held, to specifically address the role of computer evidence. Shortly thereafter, in 1995, the **International Organization on Computer Evidence** (**IOCE**) was formed. This body was created to develop guidelines and standards around the various phases of the digital forensics examination process. In 1998, in conjunction with the IOCE, the federal crime laboratory directors created the **Scientific Working Group on Digital Evidence** (**SWGDE**). This group represented the United States component of the IOCE's attempt to standardize digital forensics practices.

As organizations continued to standardize practices, law enforcement agencies continued to implement digital forensics in their overall forensic capabilities. In 2000, the FBI established the first **Regional Computer Forensic Laboratory** (**RCFL**). These laboratories were established to serve law enforcement at various levels, in a number of cybercriminal investigations. The RCFL capability has grown over the last two decades, with 17 separate RCFLs spread across the United States. In addition, other federal, state, and local police agencies have formed task forces and standalone digital forensics capabilities. With ever-increasing instances of computer-related crime, these agencies will continue to perform their critical work.

The digital forensics process

Much like the incident response process, the digital forensics process defines the flow of digital evidence related to an incident, from when it is first identified to when it is presented to either senior leadership or to a trier of fact, such as a civil or criminal court. There are several schemas that define this process and, for the most part, they generally follow a similar path. Here, we will be utilizing the **Digital Forensics Research Workshop** (**DFRWS**) digital investigation framework. This framework is depicted in the following diagram:



The framework contains six elements:

- Identification
- Preservation
- Collection
- Examination
- Analysis
- Presentation

From an incident response standpoint, personnel will not normally seize network components or critical systems and take them offline unless there is a compelling reason to do so. This is one of the balancing acts inherent in digital forensics and incident response. A purely digital forensics approach will take all relevant evidence, secure it, and process it. This process can take months, depending on the type of incident. This approach, while thorough and detailed, can leave an organization without critical components for some time. The CSIRT may be able to tell the leadership after a month-long analysis which chain of events led to a breach, but that would be pointless if a month's revenue has been lost. The examiners assigned to a CSIRT must be ready to balance the need for thoroughness against the need to resume or continue normal operations.

Identification

One principle that is often discussed in forensic science is **Locard's exchange principle**. This principle postulates that, when two objects come into contact, they leave a trace on each other. For example, if you walk into a house with carpeting, dirt from your shoes is left on the carpet, and the carpet leaves fibers on the soles of your shoes.

These traces that are exchanged form the basis of what is termed **trace evidence** in the physical forensics world. In the digital world, there is often very similar trace evidence left when two systems come into contact with each other. For example, if an individual browses a website, the web server or web application firewall may record the individual's IP address within a collection log. The website may also deposit a cookie on the individual's laptop. Just as in the physical world, evidence exchanged in this manner may be temporary, and our ability to observe it may be limited to the tools and knowledge we currently have.

This principle can guide the identification of potential sources of evidence during an incident. For example, if a CSIRT is attempting to determine the root cause of a malware infection on a system, it will start by analyzing the infected system. As some malware requires access to a C2 server, analysts can search firewall connection or proxy logs for any outbound traffic from the infected system to external IP addresses. A review of those connection IP addresses may reveal the C2 server and, potentially, more details about the particular malware variant that has infected the system.

It should be noted, though, that threat actors very easily manipulate digital evidence, so reliance on a single piece of digital evidence without other corroborating evidence should always be tempered with caution; it should be verified before it can be trusted.

Preservation

Once evidence is identified, it is important to safeguard it from any type of modification or deletion. For evidence such as log files, it may become necessary to enable controls that protect log files from removal or modification. In terms of host systems such as desktops, it may become necessary to isolate the system from the rest of the network, through either physical or logical controls, network access controls, or perimeter controls. It is also critical that any users are not allowed to access a suspect system. This ensures that users do not deliberately or inadvertently taint the evidence. Another facet of preservation measures has been increased reliance on virtual platforms. Preservation of these systems can be achieved through snapshotting systems, and by saving virtual machines on non-volatile storage.

Collection

The collection element is where digital forensics examiners begin the process of acquiring digital evidence. When examining digital evidence, it is important to understand the volatile nature of some of the evidence an examiner will want to look at. Volatile evidence is evidence that can be lost when a system is powered down. For network equipment, this could include active connections or log data that is stored on the device. For laptops and desktops, volatile data includes running memory or the **Address Resolution Protocol** (**ARP**) cache.

The **Internet Engineering Task Force** (**IETF**) has put together a document titled *Guidelines for Evidence Collection and Archiving* (RFC 3227) that addresses the order of volatility of digital evidence, as follows:

- Registers and cache
- Routing table, ARP cache, process table, kernel statistics, memory (RAM)
- Temporary filesystems
- Disk
- Remote logging and monitoring data
- Physical configuration, network topology
- Archival media

It is imperative that digital forensics examiners take this volatility into account when starting the process of evidence collection. Methods should be employed whereby volatile evidence is collected and moved to a non-volatile medium, such as an external hard drive.

Proper evidence handling

Proper handling and securing of evidence are critical. Mistakes in how evidence is acquired can lead to that evidence being tainted and, subsequently, not forensically sound. In addition, if an incident involves potential legal issues, critical evidence can be excluded from being admitted in a criminal or a civil proceeding. There are several key tenets for evidence handling that need to be followed, as listed here:

- **Altering the original evidence**: Actions taken by digital forensics examiners should not alter the original evidence. For example, a forensic analyst should not access a running system if they do not have to. It should be noted that some of the tasks that will be explored have the potential to alter some of the evidence. By incorporating proper documentation and having a justifiable reason, digital forensics examiners can reduce the chance that evidence will be deemed tainted.
- **Document**: One central theme you will often hear in law enforcement is the phrase: "If you didn't write it down, it didn't happen." This is especially true when discussing digital forensics. Every action that is taken should be documented in one way or another. This includes detailed notes and diagrams. Another way to document is through photographs. Proper documentation allows examiners to reconstruct the chain of events if ever the integrity of evidence is called into question.

There is a wide range of resources available from various law enforcement agencies on proper evidence handling in the field. You should become familiar with these procedures. The following guides are utilized by law enforcement agencies:

- `http://www.crime-scene-investigator.net/SeizingElectronicEvidence.pdf`
- `https://www.ncjrs.gov/pdffiles1/nij/219941.pdf`
- `https://www.iacpcybercenter.org/wp-content/uploads/2015/04/digitalevidence-booklet-051215.pdf`

Chain of custody

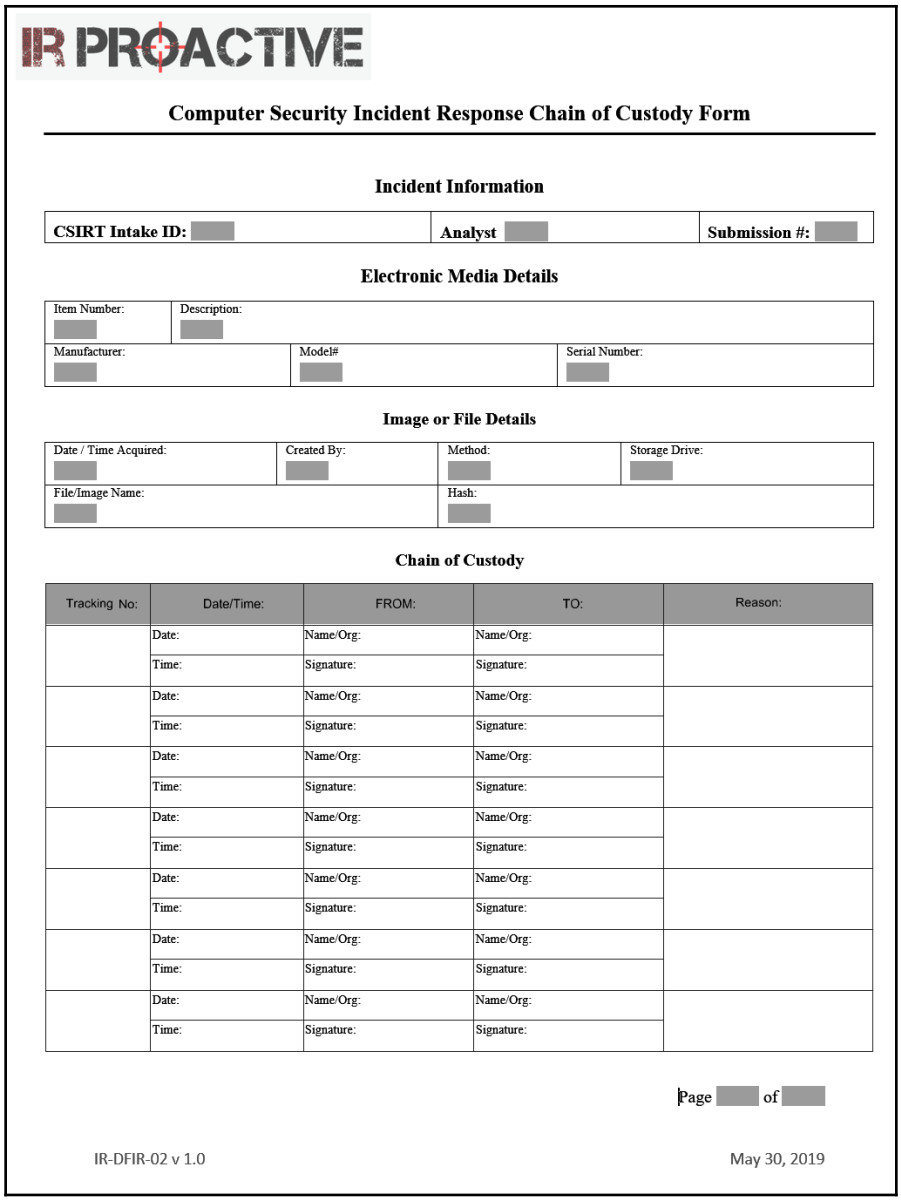
Chain of custody describes the documentation of a piece of evidence through its life cycle. This life cycle begins when an individual first takes custody of the piece of evidence, to when the incident is finally disposed of and the evidence can either be returned or destroyed. Maintaining a proper chain of custody is critical. In the event that a piece of evidence has to be brought into a courtroom, any break in the chain of custody can lead to the piece of evidence being excluded from ever being admitted into the proceedings. It is critical, therefore, to ensure that the entire life cycle of the piece of evidence is recorded.

There are two primary ways that a CSIRT can record and maintain the chain of custody of a piece of evidence.

The first is **electronically**. There are manufacturers that provide organizations such as forensic laboratories or law enforcement agencies with hardware and software that automates the chain of custody process for evidence. These systems utilize unique barcoded stickers for each piece of evidence. A scanner then creates an electronic trail as it reads these barcodes.

The second method for creating and maintaining a chain of custody is a **paper and pen method**. This method makes use of paper forms that contain the necessary information to start and maintain a chain of custody. While the paper and pen method can be a bit cumbersome and requires more due diligence to ensure that the form is safeguarded from destruction or manipulation, it is a much more cost-effective solution for smaller CSIRTs that may not have the resources necessary to implement an automated solution.

In terms of what a proper chain of custody form contains, there are several sections, each with its own details that need to be provided. The following screenshot shows a template chain of custody form (an editable chain of custody form is available from NIST at `https://www.nist.gov/document/sample-chain-custody-formdocx`.):



IR PROACTIVE

Computer Security Incident Response Chain of Custody Form

Incident Information

CSIRT Intake ID:	**Analyst**	**Submission #:**

Electronic Media Details

Item Number:	Description:	
Manufacturer:	Model#	Serial Number:

Image or File Details

Date / Time Acquired:	Created By:	Method:	Storage Drive:
File/Image Name:		Hash:	

Chain of Custody

Tracking No:	Date/Time:	FROM:	TO:	Reason:
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	

Page of

IR-DFIR-02 v 1.0 May 30, 2019

The first of these sections is a detailed description of the item. It may seem redundant to include a number of different elements, but digital forensics is about details. Having the information recorded leaves no doubt as to its authenticity. This description should contain the following elements:

- **Item Number**: A unique item number should be included on the form. In the cases of multiple pieces of evidence, a separate chain of custody form will be completed.
- **Description**: This should be a general description of the item. This can be a simple statement, such as 500 GB SATA HDD.
- **Manufacturer**: This detail assists in the case of multiple pieces of evidence with potentially different manufacturers.
- **Model**: As there is a wide variety of model numbers for components, recording this provides further details about the item.
- **Serial Number**: This is critical, in the event that an incident involves a number of systems with exactly the same configuration. Imagine attempting to reconstruct which chain of custody goes with which HDD if six were all seized together, and they had the same make and model number.

A completed first section for the chain of custody form will look like this:

Electronic Media Details

Item Number:	Description:		
1	Western Digital WD01EURS Hard Drive		
Manufacturer:		Model#	Serial Number:
Western Digital		WD01EURS	WMAV1234567

An alternate section can be used in circumstances where the evidence may be log files or images captured during the investigation. These include the following elements:

- **Date/Time Acquired**: It is important to be precise about the date and time at which specific files are acquired.
- **Description**: A brief description of the media that was acquired is useful.
- **Method**: If a software application or forensic tool is utilized to acquire the evidence, it should be noted. In other circumstances such as log files, it might simply be a copy to an external hard drive.

- **Storage Drive**: In a later section, there will be a discussion on the importance of having external media available for the storage of files. The exact drive used should be recorded on the chain of custody form.
- **File/Image Name**: The unique filename for the file or image is inserted here.
- **Hash**: For each individual file that is acquired a unique hash value should be calculated.

A completed **Image or File Details** section of the chain of custody form will look like this:

Image or File Details

Date / Time Acquired:	Created By:	Method:	Storage Drive:
5/30/19 1224 UTC	Gerard Johansen	Wireshark	USB Drive 1
File/Image Name:		Hash:	
EdgeFirewallCapture.PCAP / Packet Capture		1ceaa2393357d2ed88f81bec1e647af0	

The next section details the specific steps that the piece of evidence took in its life cycle. For each stage, the following details should be captured:

- **Tracking No**: This number indicates the step in the life cycle that the piece of evidence took.
- **Date/Time**: This is a critical piece of information in any chain of custody, and applies equally to each step the evidence took. This allows anyone who views the chain of custody to be able to reconstruct, down to the minute, each step in the chain of custody life cycle.
- **FROM and TO**: These fields can either be a person or a storage place. For example, if an analyst has seized a hard drive and is moving it to a secure storage locker, they would note that as the *to* location. It is critical to have individuals named within the chain of custody sign the form (when applicable), to enforce accountability.
- **Reason**: Moving a piece of evidence should never be done without a reason. In this portion of the chain of custody, the reason is described.

The following screenshot is a sample of the movement of the hard drive recorded in the previous screenshot. Each move for each individual piece of evidence is recorded here. The first move is the actual seizure of the drive from the system. In this case, there is no individual custodian, as the drive has been taken from the data center. What is critical is that the author is the custodian of the drive until he is able to transfer it to David Michell of ACME Forensics for analysis. The details are as follows:

Chain of Custody

Tracking No:	Date/Time:	FROM:	TO:	Reason:
1	Date: 5/30/19	Name/Org: Carol Davies Global Services Corp.	Name/Org: Gerard Johansen IRProactive	Evidence Acquisition
	Time: 1224 UTC	Signature: *Carol Davies*	Signature: *Gerard T Johansen*	
2	Date: 5/30/19	Name/Org: G Johansen IRProactive	Name/Org: David Michell ACME Forensics	Analysis
	Time: 1305 UTC	Signature: *Gerard T Johansen*	Signature: *David Michell*	

The chain of custody is maintained throughout the life of the piece of evidence. Even when the evidence is destroyed or returned, an entry is made in the chain of custody form. These forms should be maintained with any other material generated by the incident and made part of any subsequent report that is created.

Examination

The examination phase details the specific tools and forensic techniques that are utilized, to discover and extract data from the evidence that is seized as part of an incident. For example, in a case where malware is suspected of infecting a desktop system as part of a larger attack, the extraction of specific information from an acquired memory image would take part at this stage. In other cases, digital forensics examiners may need to extract **Secure Shell** (**SSH**) traffic from a network capture. The examination of digital evidence also continues the process of proper preservation, in that examiners maintain the utmost care with the evidence during the examination. If the digital forensics examiner does not take care to preserve the evidence at this stage, there is the possibility of contamination that would result in the evidence being unreliable or unusable.

Analysis

Once the examination phase has extracted potentially relevant pieces of data, the digital forensic examiner then analyzes the data in light of any other relevant data obtained. For example, if the digital forensic analyst has discovered that a compromised host has an open connection to an external IP address, they would then correlate that information with an analysis of a packet capture taken from the network. Using the IP address as a starting point, the analyst would be able to isolate that particular traffic. From here, the analyst may be able to determine that the compromised host is sending out a beacon to a C2 server. From here, using additional sources, the analyst may be able to determine which particular attack vector is linked with that IP address.

Presentation

The reporting of facts related to digital forensics needs to be clear, concise, and unbiased. In nearly all instances, a forensic examiner will be required to prepare a detailed written report, that addresses every action and captures the critical data required. This report should be thorough, accurate, and without opinion or bias. This report will often be made part of a larger incident investigation, and aids in determining the root cause of an incident.

Another aspect of presentation is the role that a forensic examiner might play in a criminal or civil proceeding. Testifying in court may be required if the incident under investigation has yielded a suspect or other responsible party. It is during this testimony that the forensic examiner will be required to present the facts of the forensic examination, in much the same dispassionate manner as the report. The examiner will be required to present facts and conclusions without bias and may be limited as to what opinions they testify to. How an examiner will be allowed to testify is often dependent on their training and experience. Some may be limited to presenting the facts of the examination. Other times, as an examiner acquires skills and has been deemed an expert witness, they may be able to offer an opinion.

Digital forensic lab

Digital forensics is an exacting process that involves the use of proper tools, techniques, and knowledge in order to extract potential evidence from systems. It is imperative that forensic examiners have a location that is separate from normal business operations. The best approach to achieving this separation is to provide CSIRT members directly involved in the examination of digital evidence with a location that is completely separate from the rest of the organization. A digital forensics lab should have several key features, to ensure that examiners have the necessary privacy but also to ensure the integrity of the evidence while it is being examined.

Physical security

Access to the forensic lab needs to be strictly controlled. In order to maintain a chain of custody, only those with a justifiable need should be allowed access to the lab. This limitation is necessary to remove any chance that the evidence can be tampered with or destroyed. The lab, therefore, should be locked at all times. Ideally, access should be granted via access cards or fobs, with a central management system granting access. This allows for a complete reconstruction of all personnel who access the laboratory within a specific time period.

The laboratory should also contain evidence lockers so that evidence can be properly stored while not being examined. Lockers should be secured, either through an onboard lock or through the use of a combination lock. The keys to these lockers should be secured within the laboratory, and access given to examiners. If the organization has adequate resources, each specific incident should have its own locker, with all the evidence contained within a single locker. This reduces the chance of digital evidence becoming commingled.

The climate and humidity should be controlled in much the same way as in any data center and should be set to the appropriate levels.

Tools

Depending on the specific examinations to be performed, it may become necessary to remove screws or cut wires. Having a small set of hand tools will be convenient for examiners. The laboratory should also be stocked with boxes for securing evidence. If examiners may have to process smartphones or tablets, faraday bags should be available. These bags allow examiners to isolate a smartphone or tablet from the cellular network, while still maintaining a power source.

Hardware

The laboratory should have sufficient computers and other hardware to perform a variety of necessary functions. Examiners will be tasked with imaging hard drives and processing gigabytes of data. As a result, a forensic computer with sufficient RAM is necessary. While there are personal preferences for the amount, a minimum of 32 GB of RAM is recommended. In addition to memory and processing power, examiners will often be looking at a large amount of data. Forensic workstations should have a primary OS drive that can contain forensic software, and a secondary drive to hold evidence. The secondary drive should contain 2 TB, or greater, of storage.

In addition to a forensic workstation, the examiner should also be provided with an internet-connected computer. The forensic workstation should have no internet connection to maintain security, but also to guard against possible corruption of evidence during an examination. A secondary machine should be used to conduct research or write reports.

Another piece of critical information is a physical write blocker. This device allows for a connection between a hard drive seized as evidence and the forensic imaging machine. The critical difference between this physical write blocker and a USB or Thunderbolt connection is that the digital forensic examiner can be sure that there is no data written to the evidence drive.

The following shows the Tableau eSATA Forensic Bridge physical write blocker:

For digital forensic laboratories that conduct a higher number of imaging tasks, there is the option of including a dedicated forensic imaging station. This allows for quicker imaging of evidence drives and does not tie up a forensic workstation. The drawback is its expense: if the CSIRT member does not see a performance drop without it, it may be hard to justify such an expense.

The CSIRT should also invest in a number of high-capacity external USB drives. These are much easier to work with and use in the imaging process than traditional SATA or IDE drives. These drives are utilized to store an evidence drive image for further analysis. The CSIRT member should have at least six of these high-capacity drives available. Drives that have 2 to 3 TB of storage space can possibly store several images at a time. Smaller USB drives are also useful to have on hand, to capture log files and memory images for later processing. With any of these USB drives, having the latest 3.0 version allows for faster processing as well.

Finally, digital forensic examiners that support a CSIRT should have a durable case to transport all of the necessary hardware, in the event they have to conduct an off-site examination. Many of these tools are fragile and would not stand the pounding delivered by baggage handlers at the local airport. The CSIRT should invest in at least two hard-sided cases, such as those used for electronic or photographic equipment. One case can transport hardware such as external hard drives, and the second can transport a forensics laptop and minimize potential damage through rough handling.

Software

There are a number of software tools on the commercial and freeware market today. The digital forensics laboratory should have access to several tools to perform similar functions. At a minimum, the lab should have software that can perform imaging of evidence drives, examine images, analyze memory captures, and report findings.

There are several different types of forensic software that a digital forensic analyst can utilize. The first of these is forensic applications. These applications are purpose-designed, to perform a variety of digital forensic tasks. They are often commercially available and are in wide use in the law enforcement and government communities, as well as in private industry. The following four forensic applications are the most common and widely deployed:

- **Autopsy**: This open source software, developed by Brian Carrier, provides a feature-rich application that automates key digital forensic tasks. As an open source project, Autopsy also has open source modules that provide a great deal of additional functionality. Autopsy will be covered in greater depth in later chapters.
- **EnCase**: Developed by OpenText, EnCase is a full-spectrum digital forensic application, performing the entire gamut of tasks in the examination of digital evidence, primarily from hard drives and other storage media. Besides analyzing digital evidence, EnCase has a reporting capability that allows examiners to output case data in an easy-to-digest format. EnCase is widely deployed in government and law enforcement agencies. One drawback is the cost associated with the application. Some CSIRTs and forensic examiners on a limited budget will have trouble justifying this cost.

- **Forensic Toolkit (FTK)**: This is another full-service forensic application that is in wide use by government and law enforcement agencies. With many of the same features as EnCase, this may be an alternative that digital forensic analysts will want to explore.
- **X-Ways Forensics**: Another option is application X-Ways Forensics application. With similar functionality to FTK and EnCase, this is a great lower-cost option for CSIRTs who do not need functionality such as network access or remote capture.

Linux forensic tools

There is also a wide range of Linux distributions that have been created for digital forensic purposes. These distributions, often provided for free, provide tools that can aid a digital forensics investigator. These tools are divided into two main types. The first of these is distributions that are intended as boot CD/DVD or USBs. These are useful for conducting triage or to obtain access to files, without having to image the drive. These distributions can be placed onto a CD/DVD or, more commonly now, a USB device. The examiner then boots the system under investigation into the Linux distribution. There are a number of these distributions available.

The following are two that are popular with digital forensic examiners:

- **Digital Evidence and Forensic Toolkit (DEFT) Zero**: This is based upon the GNU Linux platform. DEFT can be booted from a USB or CD/DVD. Once booted, the DEFT platform includes a wide range of tools that can be utilized by a digital forensic examiner to perform such functions as the acquisition of mass storage, for example, the hard drive on the system from which it is being booted. DEFT minimizes the risk of altering data on the system by not booting into the swap partition and does not use automated mounting scripts, thereby ensuring the integrity of the system's storage. DEFT can be seen in the following screenshot:

- **Paladin**: Paladin is another live Linux distribution, based on the Ubuntu OS. Paladin has a number of tools that aid digital forensic tasks such as malware analysis, hashing, and imaging. The forensic toolset includes a number of packages that can be utilized for a wide range of different operating systems. Paladin can be seen in the following screenshot:

Another category of Linux distributions is those designed as platforms for conducting an examination of evidence such as RAM captures and network evidence. There are several distributions available:

- **The SANS Investigate Forensic Toolkit (SIFT)**: This is a comprehensive forensic toolset, based upon the Ubuntu 16.04 Base OS. Tools are included for imaging, memory analysis, timeline creation, and a host of other digital forensics tasks. SIFT is provided for free by the SANS Institute as a standalone virtual machine, available at `https://digital-forensics.sans.org/community/downloads`. Alternatively, the SIFT can be installed onto an existing Ubuntu 14.04 installation.

  Once Ubuntu has been fully installed, run the following command:

  ```
  wget --quiet -O - https://raw.github.com/sans-dfir/sift-
  bootstrap/master/bootstrap.sh | sudo bash -s -- -i -s -y
  ```

  Once installed, there is a desktop, based upon the Ubuntu distribution, with additional tools that are run from the command line or through a GUI, as can be seen in the following screenshot:

- **Computer Aided INvestigative Environment (CAINE)**: This is another forensic distribution that will be put to further use in this book. CAINE is a GNU/Linux platform that includes a number of tools to assist digital forensics examiners. CAINE can be seen in the following screenshot:

- **Linux Forensics Tools Repository** (**LiFTeR**): LiFTeR is a collection of digital forensics tools for the Fedora and Red Hat Linux operating systems. This tool repository is maintained by the Carnegie Mellon University Software Engineering Institute and contains a wide range of tools for intrusion analysis and digital forensics. The package is available from: `https://forensics.cert.org/`.

- **REMnux**: REMnux is a specialized tool that has aggregated a number of malware reverse engineering tools into an Ubuntu Linux-based toolkit. There are a number of tools available in REMnux, such as those specifically designed for analyzing Windows and Linux malware and for examining suspicious documents, and it also has the ability to intercept potential malicious network traffic in an isolated container. REMnux can be seen in the following screenshot:

REMnux can be downloaded as a virtual machine from `https://remnux.org` for a standalone virtual system. REMnux can also be added to either the SIFT workstation or CAINE by utilizing the following command:

```
wget --quiet -O - https://remnux.org/get-remnux.sh | sudo bash
```

When incorporating different tools into a CSIRT digital forensics capability, it is important to keep several factors in mind. First, tools that have been developed by outsiders should absolutely be tested for efficacy. This can be done through the use of test data, commonly available on the internet. Second, open source tools such as Linux distributions are sometimes not adequately maintained. Digital forensics analysts should ensure that tools such as SIFT, CAINE, and REMnux are updated as new versions of both the tools and underlying operating systems become available. Finally, some tools that we will explore in this book are derived from network monitoring tools, but can also serve as tools in incident response. When using these tools, it is critical to document their use and their justification. If ever are called into question the efficacy and reliability of the evidence obtained or analyzed with these tools, proper documentation can lessen the chances of their use being seen as forensically unsound.

The **National Institute of Standards and Technology** (**NIST**) has provided guidance on the proper testing of forensic tools through the **Computer Forensics Tool Testing** (**CFTT**) program, found at `http://www.cftt.nist.gov/`. In addition to specific guidance on testing, there are a number of reports on different forensic hardware and software products. Having this information available for the tools you use provides validation, in the event that their use is ever challenged in a courtroom.

Jump kits

One facet of incident response that can present a challenge to CSIRT team members is the possibility that they may have to respond to incidents outside their own location. Off-site response is quite common in larger enterprises and is even the norm in CSIRTs that consult for other organizations. As a result, CSIRTs may often have to perform the entire response at another location, without the support of a digital forensics laboratory. With this challenge in mind, CSIRTs should prepare several jump kits. These kits are preconfigured and contain the hardware and software necessary to perform the tasks a CSIRT would be called upon to carry out during an incident. These kits should be able to sustain an incident investigation throughout the process, with the CSIRT identifying secure areas at the incident location in which to store and analyze evidence.

Jump kits should be portable and can be configured to fit within a secure hard-sided case, and should be ready to be deployed at any time. CSIRTs should ensure that, after each incident, the jump kit is restocked with any items that were utilized in the last incident, and that hardware and software are properly configured so that, during an incident, analysts can be confident in their availability. An example of a jump kit can be seen in the following:

At a minimum, a jump kit should contain the following:

- **Forensic laptop**: This laptop should contain enough RAM (32 GB) to image a hard drive in a reasonable amount of time. The laptop should also contain a forensic software platform (as previously discussed). If possible, the laptop should also contain at least one Linux forensic OS, such as CAINE or SIFT.
- **Networking cables**: Having several CAT5 cables of varying lengths is useful in the event that the CSIRT team has to access a network or patch into any network hardware, such as a router or a switch.
- **Physical write blocker**: Each kit should have a physical write blocker that can be used to image any hard drives that CSIRT personnel may encounter.
- **External USB hard drives**: The jump kit should contain several 1 TB or 2 TB USB hard drives. These will be used for imaging hard drives on potentially compromised systems.
- **External USB devices**: It is not forensically sound to store evidence collected from log sources or RAM captures on a potentially compromised system. The jump kit should contain several large-capacity (64 GB) USBs for offloading log files, RAM captures, or other information obtained from command-line outputs.
- **Bootable USB or CD/DVD**: While not utilized in every case, having several bootable Linux distributions can be useful in the event that the forensic laptop is currently performing another task.
- **Evidence bags or boxes**: It may become necessary to seize a piece of evidence and transport it off-site while an incident is ongoing. There should be the capability to secure evidence on-site without having to search around for a proper container.
- **Anti-static bags**: In the event that hard drives are seized as evidence, they should be transported in anti-static bags.
- **Chain of custody forms**: As was previously discussed, having a chain of custody form for each piece of evidence is critical. Having a dozen blank forms available saves the trouble of trying to find a system and printer to print out new copies.
- **Toolkit**: A small toolkit that contains screwdrivers, pliers, and a flashlight comes in handy when hard drives have to be removed, connections are cut, or the analyst has to access a dark corner of the data center.

- **Notepad and writing instrument**: Proper documentation is critical; handwritten notes in pen may seem old-fashioned, but they are the best way to reconstruct events as an incident continues to develop. Having several steno notebooks and pens as part of the kit ensure that CSIRT personnel do not have to hunt down these items while a critical event has just occurred. Jump kits should be inventoried at least monthly so that they are fully stocked and prepared for deployment. They should also be secured and accessible by CSIRT personnel only. Left in public view, these kits are often raided by other personnel in search of a screwdriver, network cable, or flashlight. For CSIRTs that support geographically dispersed organizations, with several kits at key locations, such as major office headquarters, data centers, or other off-site locations, it may be pertinent to have several of these jump kits pre-staged for use. This avoids having to cart the kit through an airport. An example of some items to be stocked in a jump kit can be seen in the following:

Summary

Incident response spans a wide range of disciplines, from legal to scientific. CSIRT members responsible for conducting digital forensics examinations should be very familiar with the legal and technical aspects of digital forensics. In addition, they should be familiar with the wide variety of tools and equipment necessary to acquire, examine, and present data discovered during an examination. The proper application of forensic techniques is critical, to provide insight into the chain of events that led to the deployment of the CSIRT to investigate an incident. This chapter initially delved into the various legal aspects of digital forensics, such as the rules of evidence and laws pertaining to cybercrime. Next, the science of digital forensics was discussed, providing an understanding of how techniques should be applied to investigations. To enhance this knowledge, we looked at how these techniques fit into a framework of digital investigations. This led to an overview of the various tools available for digital forensics examiners.

In the next chapter, the focus will be on jumping *onto the wire*, with a discussion of network forensics.

Questions

1. What is not a federal rule of evidence?

   A) Test for relevant evidence
   B) Locard's principle
   C) Testimony by an expert witness
   D) Best evidence rule

2. A proper chain of custody should be maintained, to ensure the integrity of digital evidence.

   A) True
   B) False

3. Which items should be included as part of a digital forensics jump kit?

   A) Physical write blocker
   B) Notepad and pen
   C) Networking cables
   D) All of the above

4. What is NOT a portion of the forensic process?

   A) Identification
   B) Courtroom testimony
   C) Collection
   D) Analysis

Further reading

- Digital Forensics Research Workshop: `https://www.dfrws.org`
- ISACA *Overview of Digital Forensics*: `http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/overview-of-digital-forensics.aspx`
- Historical background on the FBI CART: `https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=137561`

2
Section 2: Evidence Acquisition

This section focuses on the technical aspects of digital evidence acquisition. This includes detailed examination of the tools and techniques that can be leveraged for proper evidence acquisition.

This section comprises the following chapters:

- `Chapter 4`, *Collecting Network Evidence*
- `Chapter 5`, *Acquiring Host-Based Evidence*
- `Chapter 6`, *Forensic Imaging*

4
Collecting Network Evidence

The traditional focus of digital forensics has been on locating evidence on the suspect host's hard drive. Law enforcement officers interested in criminal activity such as fraud or child exploitation can find the evidence required for prosecution on a single hard drive. In the realm of incident response, though, it is critical that the focus goes far beyond a suspected compromised system. For example, there is a wealth of information that can be obtained within the hardware and software along with the flow of traffic from a compromised host to an external **Command and Control** (**C2**) server.

This chapter focuses on the preparation, identification, and collection of evidence that is commonly found among network devices and along traffic routes within an internal network. This collection is critical during incidents where an external threat source is in the process of commanding internal systems or stealing data out of the network. Network-based evidence is also useful when examining host evidence as it provides a second source of event corroboration, which is extremely useful in determining the root cause of an incident.

We will cover the following topics in this chapter:

- An overview of network evidence
- Firewalls and proxy logs
- NetFlow
- `tcpdump` packet capture
- Wireshark packet capture
- Evidence collection

An overview of network evidence

There are network log sources that can provide CSIRT personnel and incident responders with good information. A range of manufacturers provides each of these with network devices. As a preparation task, CSIRT personnel should become familiar with how to access these devices in order to obtain the necessary evidence or should have existing communication structures in place to engage IT personnel to assist with the proper response techniques during an incident.

Network devices such as switches, routers, and firewalls also have their own internal logs that maintain data on who accessed the device and made changes. Incident responders should become familiar with the types of network device on their organization's network and should be able to access these logs in the event of an incident:

- **Switches**: These are spread throughout a network through a combination of core switches that handle traffic from a range of network segments and edge switches that handle traffic for individual segments. As a result, traffic that originates on a host and travels out of the internal network will traverse a number of switches. Switches have two key points of evidence that should be addressed by incident responders. First is the **Content Addressable Memory** (**CAM**) table. This CAM table maps the physical ports on the switch to the **Network Interface Card** (**NIC**) on each device connected to the switch. Incident responders tracing connections to specific network jacks can utilize this information. This can aid the identification of possible rogue devices such as wireless access points or systems connected to the internal network by an adversary. The second way in which switches can aid an incident investigation is through facilitating network traffic capture.
- **Routers**: Routers allow organizations to connect multiple LANs into either a **Metropolitan Area Network** (**MAN**) or a **Wide Area Network** (**WAN**). As a result, they handle an extensive amount of traffic. The key piece of evidentiary information that routers contain is the routing table. This table holds the information for specific physical ports that map to the networks. Routers can also be configured to deny specific traffic between networks and maintain logs on allowed traffic and data flows. Another significant source of evidence that routers can provide is NetFlow data. NetFlow provides data on IP addresses, ports, and protocols of network traffic. This data can be utilized to determine the flow of traffic from various segments of the network (NetFlow will be covered in greater detail later in this chapter).

- **Firewalls**: Firewalls have changed significantly since the days when they were just considered to be a different type of router. Next-generation firewalls contain a wide variety of features such as intrusion detection and prevention, web filtering, data loss prevention, and detailed logs about allowed and denied traffic. Often, firewalls serve as a detection mechanism that alerts security personnel to potential incidents. This can include alerts from features such as IDS/IPS systems, blacklists of known bad URLs or IP addresses, or alerts flagging configuration changes to the firewall without the knowledge of IT personnel. Incident responders should have as much visibility of how their organization's firewalls function and what data can be obtained prior to an incident.
- **Network intrusion detection and prevention systems**: These systems were purposefully designed to provide security personnel and incident responders with information concerning potential malicious activity on the network infrastructure. These systems utilize a combination of network monitoring and rulesets to determine whether there is any malicious activity. **Intrusion Detection System** (**IDS**) are often configured to alert you to a specific malicious activity while an **Intrusion Prevention System** (**IPS**) can detect, but also block, potential malicious activity. In either case, both types of platform log are an excellent place for incident responders to locate specific evidence on malicious activity.
- **Web proxy servers**: Organizations often utilize web proxy servers to control how users interact with websites and other internet-based resources. As a result, these devices can give an enterprise-wide picture of web traffic that both originates and is destined for internal hosts. Web proxies also have additional features such as alerting security personnel to connections to known malware C2 servers or websites that serve up malware. A review of web proxy logs in conjunction with a possibly compromised host may identify a source of malicious traffic or a C2 server exerting control over the host.
- **Domain controllers or authentication servers**: Serving the entire network domain, authentication servers are the primary location that incident responders can leverage for details on successful or unsuccessful logins, credential manipulation, or other credential uses.
- **DHCP server**: Maintaining a list of assigned IP addresses for workstations or laptops within the organization requires an inordinate amount of upkeep. The use of **Dynamic Host Configuration Protocol** (**DHCP**) allows for the dynamic assignment of IP addresses to systems on the LAN. DHCP servers often contain logs on the assignment of IP addresses mapped to the MAC address of the host's NIC. This becomes important if an incident responder has to track down a specific workstation or laptop that was connected to the network at a specific date and time.

- **Application servers**: A wide range of applications from email to web applications is housed on network servers. Each of these can provide logs that are specific to the type of application. Also of interest during an incident investigation are any logs pertaining to remote connections. Adversaries will often pivot from a compromised system to servers in order to gain access to confidential data or for other follow-up activities.

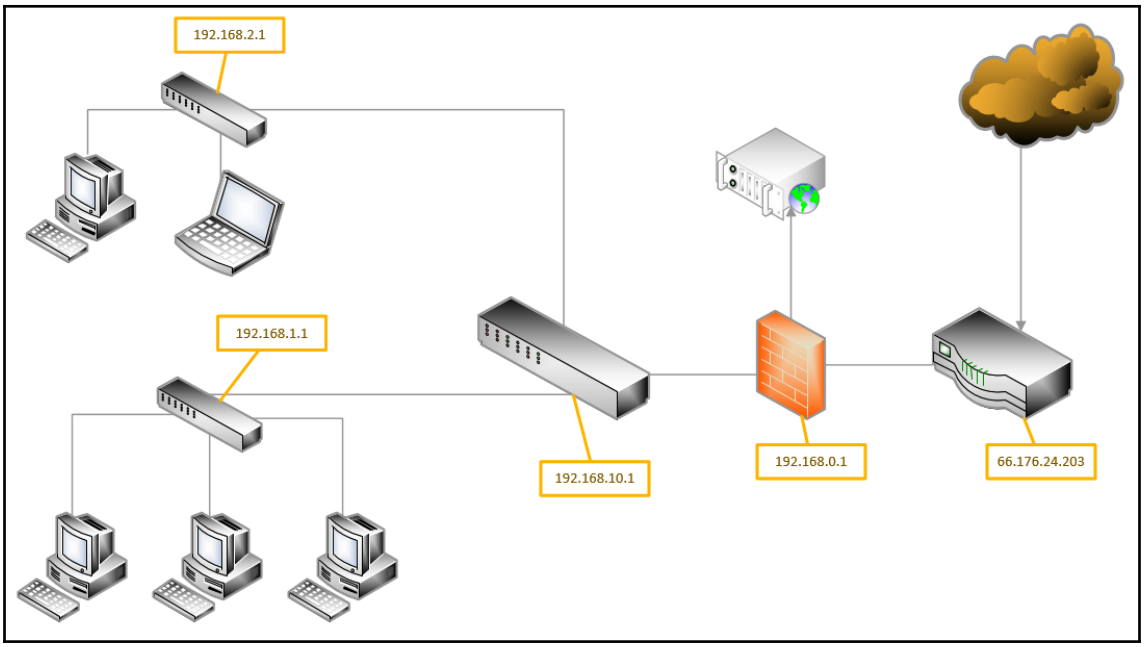
Preparation

The ability to acquire network-based evidence is largely dependent on the preparations that are undertaken by an organization prior to an incident. Without some critical components of a proper infrastructure security program, key pieces of evidence will not be available for incident responders in a timely manner. The result is that evidence may be lost as CSIRT members hunt down critical pieces of information. In terms of preparation, organizations can aid the CSIRT by having proper network documentation, up-to-date configurations of network devices, and a central log management solution in place.

Aside from the technical preparation for network evidence collection, CSIRT personnel need to be aware of any legal or regulatory issues with regard to collecting network evidence. Additionally, CSIRT personnel need to be aware that capturing network traffic can be considered an invasion of privacy if there is no policy clearly stating that network monitoring takes place. Therefore, the legal representative of the CSIRT should ensure that all employees of the organization understand that their use of the information system will be monitored. This should be expressly stated in policies prior to any evidence collection that may take place.

Network diagram

To identify potential sources of evidence, incident responders need to have a solid understanding of what the internal network infrastructure looks like. One method that can be employed by organizations is to create and maintain an up-to-date network diagram. This diagram should be detailed enough so that incident responders can identify individual network components such as switches, routers, or wireless access points. This diagram should also contain internal IP addresses so that incident responders can immediately access those systems through remote methods. For instance, examine the following simple network diagram:



This diagram allows for the quick identification of potential evidence sources. For example, suppose that the laptop connected to the switch at `192.168.2.1` is identified as communicating with a known malware C2 server. A CSIRT analyst could examine the network diagram and ascertain that the C2 traffic would have to traverse several network hardware components on its way out of the internal network. For example, there would be traffic traversing the switch at `192.168.10.1`, through the firewall at `192.168.0.1`, and, finally, from the router out to the internet.

Configuration

Determining whether an attacker has made modifications to a network device such as a switch or a router can be made easier if the CSIRT has a standard configuration immediately available. Organizations should already have configurations for network devices stored for disaster recovery purposes, but they should have these available for CSIRT members in the event that there is an incident.

Firewalls and proxy logs

Two main sources of evidence available while investigating an incident are ingress/egress points into the network from the internet. Modern malware and other exploits will often require the ability to reach internet-based resources. This may be for the purpose of downloading additional malware or to exploit code. Other attacks that involve data exfiltration will require access to the internet. Finally, adversaries will often have to establish C2 over compromised systems. In all of these cases, traffic from various protocols will traverse the perimeter of the victim network. Depending on the victim, this traffic will have to traverse a firewall, internet proxy, or both. As a result, both of these technologies provide incident response personnel with a major source of evidence.

Firewalls

Firewalls have evolved from a simplified routing and blocking technology into platforms that provide a significant insight into the traffic coming into and leaving the network. Next-generation firewalls often combine the deny/allow ruleset with IDS or IPS, as well as controlling network access to applications. This creates a significant source of evidence that can be leveraged during an incident.

Acquiring evidence from firewalls is largely dependent on the manufacturer and the specific model that is used. Incident responders should thoroughly understand the feature set and specific data that can be obtained as part of their preparation. Although features differ from vendor and model, there are some key evidence points that are near-universal:

- **Connection log**: The connection log provides the source and destination IP addresses and protocols of connections between internal and external systems. This is critical to determining whether any internal systems may have contacted an adversary-controlled system or are possibly being controlled. In addition to allowed connections, the logs may also provide an insight into connections that were denied. One technique that is often used by adversaries is to use tools to attempt to connect to well-known ports that are commonly in use. If these ports are closed to external connections, there will be a *deny* entry in the logs. Successive denies across a range of ports are indicative of reconnaissance activity.

- **Remote access logs:** Firewalls often serve as the **Virtual Private Network** (**VPN**) concentrator for remote access. If a remote user becomes infected via malware, they can introduce that infection into the internal network through the VPN. Remote access logs will show systems that are connected and what time they connected. This may allow incident responders to correlate activities and determine whether a remote user was the source of the infection.

Web proxy server

Adversaries often make use of scripting such as Microsoft Visual Basic or PowerShell to download secondary exploit packages or malware. These scripts will often contain a URL that points to the exploit or malware. Adversaries make use of URLs as opposed to IP addresses as the IP addresses can be easily changed via domain name registration, allowing them to change their infrastructure without having to change their scripts.

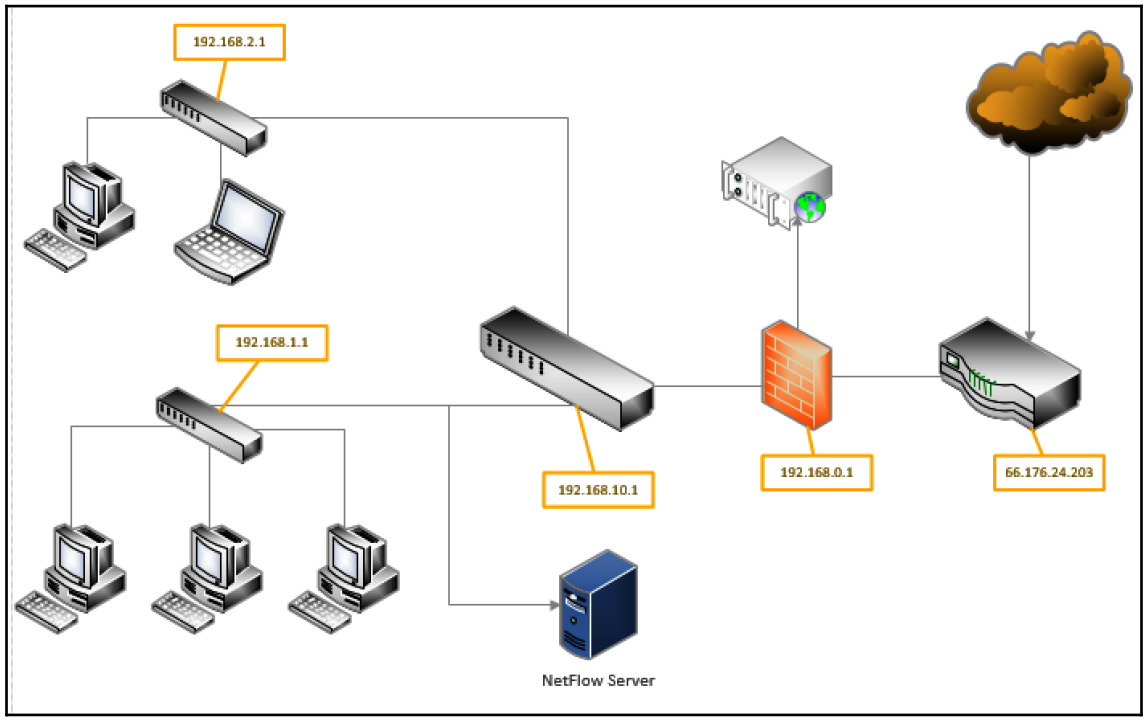
Organizations that make use of web proxy servers for HTTP and HTTPS requests will have a record of any system on the internal network that reached out to an external site. From here, they may be able to identify the location and, potentially, the malware or exploit that has been downloaded. Additional insight may be gained from C2 traffic that makes use of similar tactics as malware.

As detecting attacks often takes months, it is imperative that incident responders can view the history of an activity that has happened over weeks or even months. Given the relatively small size of proxy requests, even just the date, time, requesting system, and the URL that was visited can provide a significant piece of evidence that might not otherwise be available.

NetFlow

First designed by Cisco Systems 1996, NetFlow is a feature found in network devices such as switches and routers that allows network administrators to monitor traffic within the network. NetFlow is not strictly a security tool, but it does provide a good deal of data to incident responders in the event of an incident. NetFlow is sent by network devices via the UDP protocol to a central collection point, often called the NetFlow Collector.

In a security context, NetFlow provides deep insights into the internal traffic of systems as they communicate with each other. This is often referred to as east-west traffic as opposed to the north-south traffic, which is used to describe internal systems communicating with external systems through the perimeter firewall. For example, the following diagram shows a simple network. In a real-world scenario, an attacker may compromise a system on the `10.10.2.0/24` subnet. From there, they may attempt to pivot to a file server on the `10.10.1.0/24` subnet. Once there, they can acquire confidential data and move it back to the compromised system for exfiltration. The switches forward the NetFlow data to the collector, which includes the IP addresses, protocols, and data size. This data is critical to providing incident response analysts with details that they may not normally otherwise acquire:



Configuring NetFlow is dependent on the type and manufacturer of the network components. Moreover, there is a wide range of collectors and analysis tools that can be leveraged depending on budgetary and other resources. One of the advantages from including NetFlow analysis in the overall network operations is that it not only provides data to the incident response team, but it is also highly useful in day-to-day network operations in terms of hunting down latency or other communication issues. This dual purpose makes including it as part of the overall network operations easier to justify.

Packet captures

Capturing network traffic is critical to having a full understanding of an incident. Being able to identify potential C2 IP address traffic may provide further information about the type of malware that might have infected a host. In other types of incidents, CSIRT members may be able to identify potential exfiltration methods that an external threat actor is utilizing.

One method is to set up what is referred to as a network tap. A network tap is a system that is in line with the compromised host and the switch. For example, in the network diagram, if the host that is compromised is on the `192.168.1.0/24` subnet, the tap should be placed in between the host and the switch. This often involves placing a system in between the host and the switch.

Another option is to configure a **Switched Port Analyzer** (**SPAN**) port. In this configuration, the switch closest to the compromised host will have port mirroring enabled. This then sends the traffic from the entire segment the switch is on to the system that is on the mirrored port.

Finally, some network devices have built-in applications such as `tcpdump` that can be utilized to capture traffic for further analysis. This may be the quickest option as it does not require physical access to the network or the switch and can be set up remotely. The drawback to this method is that storage on the switch may not support a large capture file and the added strain may increase the chances of some packets not being captured.

tcpdump

`tcpdump` is a command-line tool specifically designed for packet capture. `tcpdump` is often included with Linux distributions and is found on many network devices. For many of these devices, `tcpdump` has to be run as a root user or with root privileges as it will be monitoring network traffic. The documentation is available at `http://www.tcpdump.org/` . To perform a packet capture with tcpdump, the following process can be used:

1. To access the basic help menu, type the following into a Command Prompt:

```
dfir@ubuntu:~$ tcpdump -h
```

The output of the preceding command is as follows:

```
File  Edit  View  Search  Terminal  Help
dfir@ubuntu:~$ tcpdump -h
tcpdump version 4.9.2
libpcap version 1.8.1
OpenSSL 1.1.1  11 Sep 2018
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqStuUvxX#] [ -B size ] [ -c count ]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
                [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
                [ --immediate-mode ] [ -T type ] [ --version ] [ -V file ]
                [ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z postrotate
-command ]
                _[ -Z user ] [ expression ]
```

The default `tcpdump` setting is to capture traffic on all available interfaces. Running the following command produces a list of all the interfaces that tcpdump can capture traffic on:

```
dfir@ubuntu:~$ tcpdump -D
```

The following screenshot shows that the `ens33` (Ethernet) and `lo` (loopback) interfaces are available for capturing traffic:

```
File  Edit  View  Search  Terminal  Help
dfir@ubuntu:~$ tcpdump -D
1.ens33 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth0 (Bluetooth adapter number 0)
5.nflog (Linux netfilter log (NFLOG) interface)
6.nfqueue (Linux netfilter queue (NFQUEUE) interface)
7.usbmon1 (USB bus number 1)
8.usbmon2 (USB bus number 2)
```

2. To configure a basic capture on the Ethernet interface located at `ens33` with normal verbosity, type the following command:

```
dfir@ubuntu:~$ sudo tcpdump -i ens33 -v
```

The `-i` switch tells `tcpdump` which interface to perform the packet capture on. In this case, it is on the following Ethernet interface: `ens33`. The `-v` switch sets the verbosity of the packet capture. In this case, the verbosity is set rather low. For additional data, the switch can be set to `-vvv` for a more detailed look at the packets. The following screenshot shows what information is displayed by the command:

```
File  Edit  View  Search  Terminal  Help
(1), length 84)
    ubuntu > dns.google: ICMP echo request, id 40024, seq 47, length 64
08:31:08.437340 IP (tos 0x0, ttl 128, id 43477, offset 0, flags [none], proto IC
MP (1), length 84)
    dns.google > ubuntu: ICMP echo reply, id 40024, seq 47, length 64
08:31:09.420894 IP (tos 0x0, ttl 64, id 54227, offset 0, flags [DF], proto ICMP
(1), length 84)
    ubuntu > dns.google: ICMP echo request, id 40024, seq 48, length 64
08:31:09.440265 IP (tos 0x0, ttl 128, id 43478, offset 0, flags [none], proto IC
MP (1), length 84)
    dns.google > ubuntu: ICMP echo reply, id 40024, seq 48, length 64
08:31:10.423250 IP (tos 0x0, ttl 64, id 54439, offset 0, flags [DF], proto ICMP
(1), length 84)
    ubuntu > dns.google: ICMP echo request, id 40024, seq 49, length 64
08:31:10.443728 IP (tos 0x0, ttl 128, id 43479, offset 0, flags [none], proto IC
MP (1), length 84)
    dns.google > ubuntu: ICMP echo reply, id 40024, seq 49, length 64
08:31:11.424959 IP (tos 0x0, ttl 64, id 54559, offset 0, flags [DF], proto ICMP
(1), length 84)
```

While this method determines whether traffic is traversing that interface, the individual packet information is useless to an analyst due to the speed with which the individual packets appear on the screen. For the packet capture to be of any use, it is recommended that you output the file so that a later examination can be performed with a packet analysis tool such as Wireshark. Wireshark will be reviewed later on in this chapter and with greater detail in `Chapter 7`, *Analyzing Network Evidence*.

3. To configure `tcpdump` to output the packet capture to a file, the following command is used:

```
dfir@ubuntu:~$ sudo tcpdump -i ens33 -vvv -w ping_capture
```

PING (short for **Packet Internet Groper**) is the utility that uses the ICMP packets being sent in this section. PING is used to determine whether there is network connectivity to various systems. In this case, a check of connectivity to the Google DNS at `8.8.8.8` is being performed.

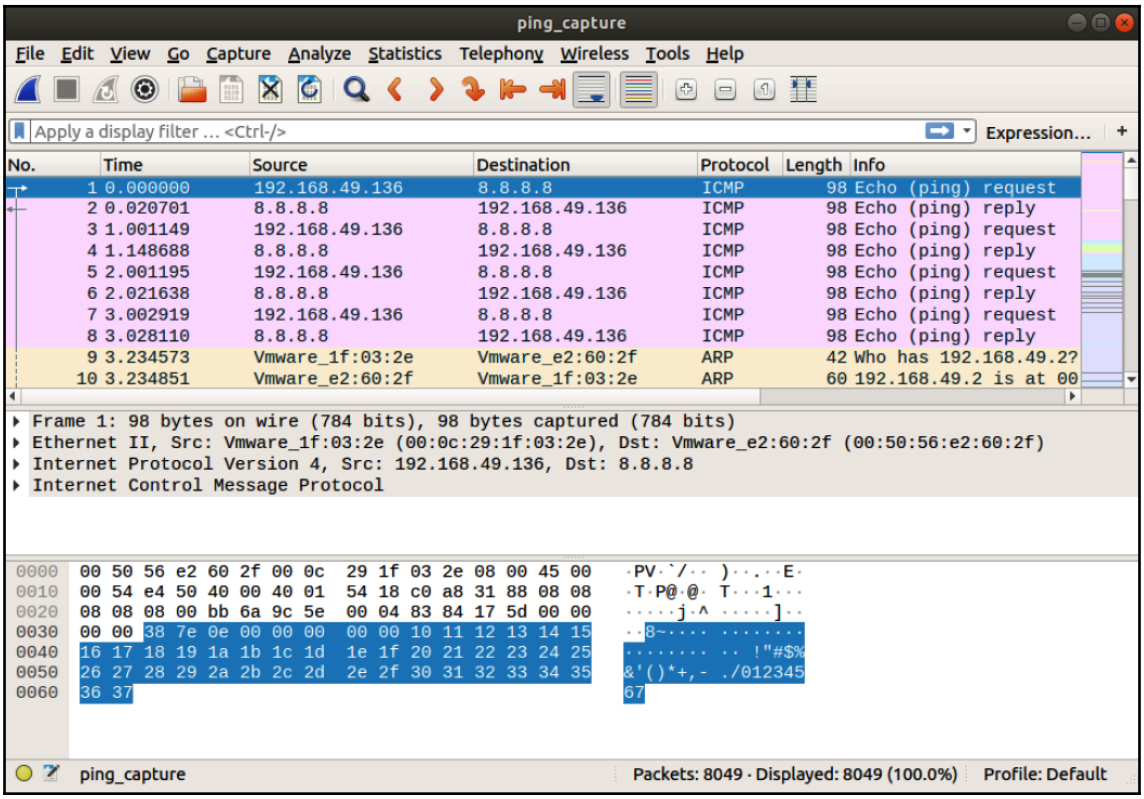
The command tells `tcpdump` to capture network traffic and write the file out to capture. Unlike the previous capture, there is no traffic indicated on the screen.

4. To stop the capture, type *Ctrl + C*, which produces the following information:

```
dfir@ubuntu:~$ sudo tcpdump -i ens33 -vvv -w ping_capture
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 by
tes
^C4331 packets captured
4333 packets received by filter
0 packets dropped by kernel
```

The previous screenshot indicates that a total of 4,333 packets were received and recorded in the capture file.

5. After navigating to the root directory, the file can then be opened via Wireshark:



`tcpdump` can also be configured to focus the capture on specific sources or destination IP addresses and ports. For example, if an incident response analyst needs to collect packets leaving a specific host at the `192.168.10.54` IP address, the following `tcpdump` command will produce the desired results:

```
dfir@ubuntu:~$ sudo tcpdump –i ens33 src host 192.168.10.54
```

Packets going to a destination such as a known C2 server at the IP address can also be separated out from the background network traffic with the following command:

```
dfir@ubuntu:~$ sudo tcpdump –i ens33 dst host 162.4.5.23
```

`tcpdump` is a powerful tool and has plenty of options. Incident response analysts would be advised to examine and incorporate its various features into their toolkit.

WinPcap and RawCap

During an incident, it may become necessary to obtain a packet capture from a Windows system. In incidents such as the compromising of a web server or application server, a Windows system will not have a native application with which to conduct a packet capture. There are several packet capture tools available on Windows systems. The first tool that can be utilized is WinPcap. This tool is generally recognized as the standard for packet capture on Windows systems and is available as a free download at `https://www.winpcap.org/`. The drawback to this tool from a forensics perspective is that it has to be installed on the system. This can complicate a forensic analysis as any changes to the system have to be thoroughly documented. For this reason, it is a good preparatory step to ensure that high-risk systems such as web servers, file servers, and application servers have WinPcap installed.

Another option available to incident response analysts is the use of tools such as RawCap. RawCap has the same basic capability as WinPcap without the need to install it on the local system. RawCap can be easily run from a USB device attached to the system. To perform a packet capture with RawCap, the following process is used:

1. Start the Windows Command Prompt as an administrator.
2. In the Command Prompt, navigate to the folder containing the `RawCap.exe` file. For a list of options, type the following:

```
D:\>RawCap.exe –help
```

The command will produce the following output:

```
D:\>RawCap.exe --help
NETRESEC RawCap version 0.1.5.0
http://www.netresec.com

Usage: RawCap.exe [OPTIONS] <interface_nr> <target_pcap_file>

OPTIONS:
 -f             Flush data to file after each packet (no buffer)
 -c <count>     Stop sniffing after receiving <count> packets
 -s <sec>       Stop sniffing after <sec> seconds

INTERFACES:
 0.     IP         : 169.254.166.101
        NIC Name   : Ethernet
        NIC Type   : Ethernet

 1.     IP         : 169.254.172.194
        NIC Name   : Npcap Loopback Adapter
        NIC Type   : Ethernet

 2.     IP         : 169.254.180.113
        NIC Name   : Local Area Connection* 2
        NIC Type   : Wireless80211

 3.     IP         : 192.168.80.1
        NIC Name   : VMware Network Adapter VMnet1
        NIC Type   : Ethernet

 4.     IP         : 192.168.49.1
        NIC Name   : VMware Network Adapter VMnet8
        NIC Type   : Ethernet

 5.     IP         : 192.168.0.30
        NIC Name   : Wi-Fi
        NIC Type   : Wireless80211
```

The output produces a list of interfaces. One of the advantages of RawCap in that, even from a USB device, the incident response analyst can perform a packet capture on each of the interfaces. In this example, the capture will be performed on wireless interface number `5`.

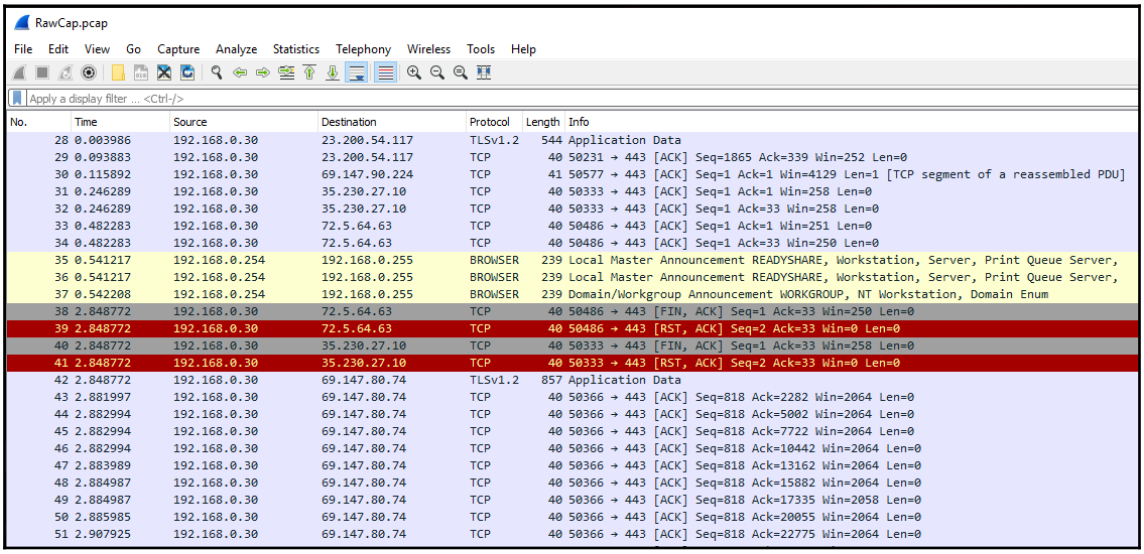
3. To start the packet capture, RawCap requires the network interface where the traffic should be captured, and an output file to output the packet capture. To capture the traffic on the wireless interface and output it to a file called `RawCap.pcap`, the following command is used:

```
D:\>RawCap.exe 5 RawCap.pcap
```

The command produces the following output:

```
Sniffing IP : 192.168.0.30
File        : RawCap.pcap
Packets     : 4508
```

4. Typing *Ctrl* + *C* will stop the capture. The capture file, `RawCap.pcap`, is saved to the same directory as the `RawCap.exe` file. This file can then be opened with tools such as Wireshark for further analysis:

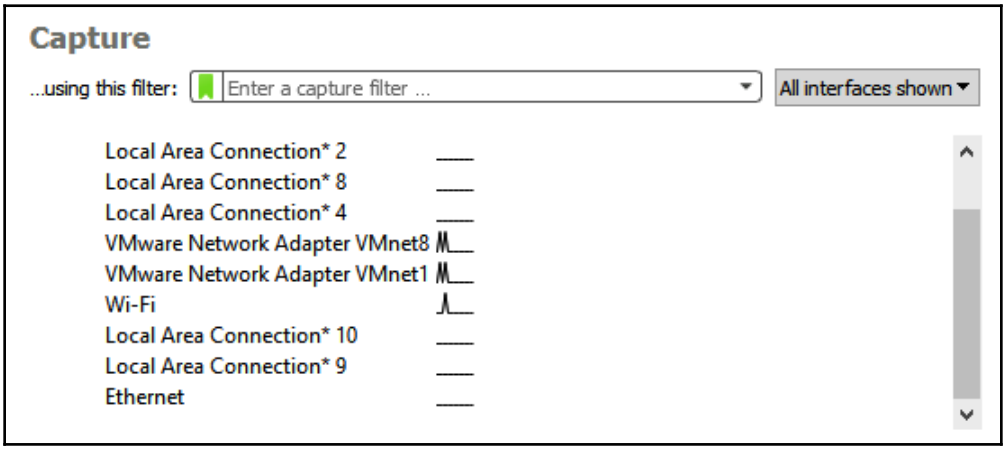


Next, we will learn about the Wireshark tool.

Wireshark

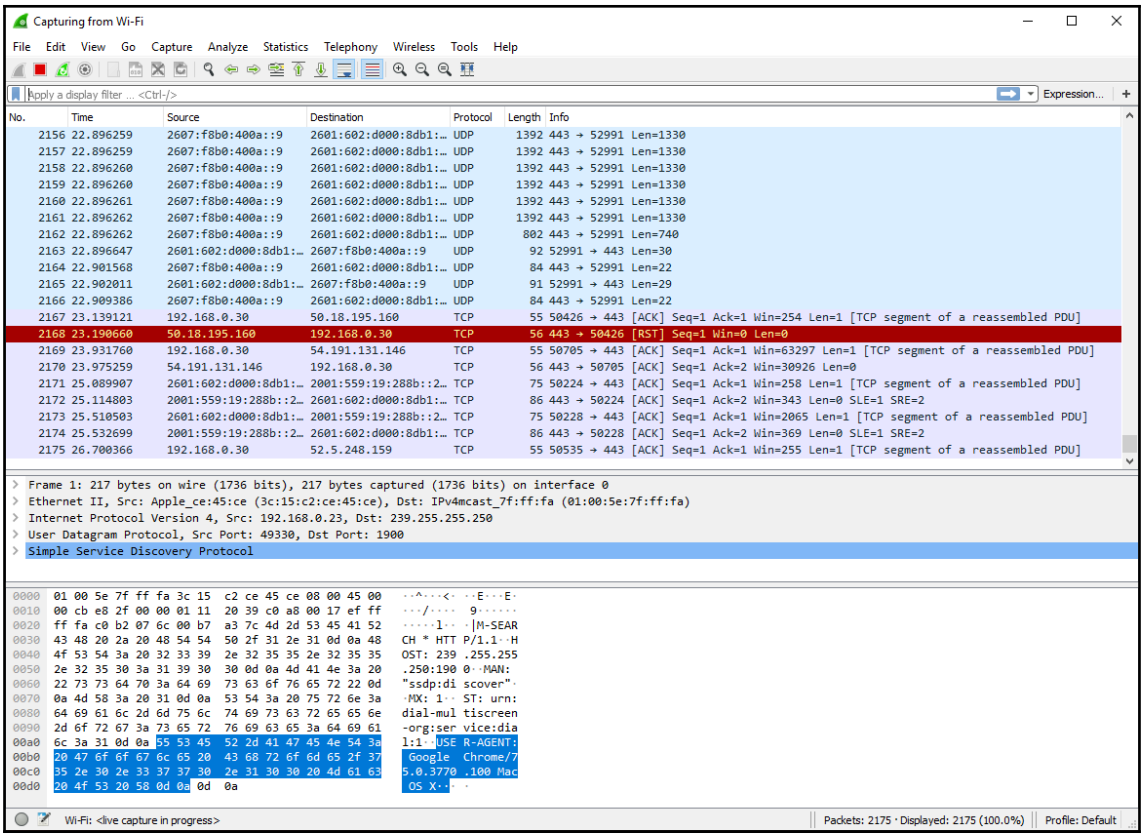
Wireshark is a Unix or Windows packet capture and analysis tool. Unlike `tcpdump` or tools such as RawCap, Wireshark is a GUI-based tool and has a number of not only packet capture, but also analysis features. As a result, Wireshark may be difficult to deploy rapidly during an incident as the program has to be installed. Furthermore, the tool is only supported on Windows or macOS. To install Wireshark on a Linux system requires a bit more effort. The one distinct advantage that Wireshark has over command-line options is that incident response analysts can perform a detailed inspection of the traffic as it is being captured. Wireshark can be run on the system itself or on a USB drive. Once installed, it has to be run as an administrator. To perform a packet capture with Wireshark, the following process is used:

1. The first step is to select an interface that Wireshark will capture traffic on:



   In the previous screenshot, there are three interfaces that appear to be handling traffic. The one we will focus on is the Wi-Fi interface.

2. Double-clicking on the interface will start a packet capture. As was stated before, unlike `tcpdump` or RawCap, the actual capture is outputted to the screen for immediate analysis:



3. To stop the capture, hit the red box in the upper-left corner of the pane. The file can then be saved for further analysis.

Another tool that is included with Wireshark, and is useful during evidence acquisition, is mergecap. Mergecap is a command-line tool that allows incident response analysts to combine multiple packet capture files from Wireshark, `tcpdump`, or RawCap. This is extremely useful in situations where incident response analysts obtain packet captures from several sources but want to check for traffic to a specific host. To access the menu for mergecap, type the following into a Command Prompt:

```
dfir@ubuntu:~$mergecap -help
```

That command produces the following help information:

```
File  Edit  View  Search  Terminal  Help
dfir@ubuntu:~$ mergecap -help
Mergecap (Wireshark) 2.6.8 (Git v2.6.8 packaged as 2.6.8-1~ubuntu18.04.0)
Merge two or more capture files into one.
See https://www.wireshark.org for more information.

Usage: mergecap [options] -w <outfile>|- <infile> [<infile> ...]

Output:
  -a                 concatenate rather than merge files.
                     default is to merge based on frame timestamps.
  -s <snaplen>       truncate packets to <snaplen> bytes of data.
  -w <outfile>|-     set the output filename to <outfile> or '-' for stdout.
  -F <capture type> set the output file type; default is pcapng.
                     an empty "-F" option will list the file types.
  -I <IDB merge mode> set the merge mode for Interface Description Blocks; defau
lt is 'all'.
                     an empty "-I" option will list the merge modes.

Miscellaneous:
  -h                 display this help and exit.
  -v                 verbose output.
```

To merge several packet capture files, the following command is used:

```
dfir@ubuntu:~$mergecap –w switches.pcap switch1.pcap switch2.pcap
switch3.pcap
```

By combining the output of three packet captures to one file, the incident response analyst has the ability to examine a wider range of activities across multiple network paths. If, for example, the analyst is searching for traffic coming from an unknown host to an external C2 server, they would be able to combine captures over the entire span of the network and then search for that particular IP rather than individually picking through each packet capture.

Evidence collection

In order to conduct a proper examination of log files and other network data such as packet captures, they often have to be moved from the log source and examined offline. As with any source of evidence, log files or packet captures have to be handled with due care to ensure that they are not corrupted or modified during the transfer. One simple solution is to transfer the evidence immediately to a USB drive or similar removable medium. From there, a hash can be created for the evidence prior to any examination.

The acquisition of network evidence such as a packet capture or log file should be thoroughly documented. Incident response personnel may be acquiring log files and packet captures from a number of sources over the entire network. As a result, they should ensure that they can trace back every separate piece of evidence to its source as well as the date and time that the evidence was collected. This can be recorded in a network evidence log sheet and entries completed for each piece of evidence. For example, the following is a sheet with an entry:

File Name	Description	Location	Date	Time	Collected By	MD5 Hash
Ping_capture	Packet Capture of Ping activity	192.168.2.1	6/26/19	1642	GTJ	7e559dc8eeeb66115566d93f96e7dfb8

The log entry captures the following necessary information:

- **File Name**: Each log file or packet capture should have its own unique name. Within the procedures in use by the CSIRT, there should be a naming convention for different types of evidence file.
- **Description**: A brief description of the file. There does not need to be too much detail unless it is a unique file and a detailed description is called for.
- **Location**: The location is important. In this case, the packet capture was obtained on the switch located at `192.168.2.1`.
- **Date and Time**: Record the date and time the file was transferred to the medium.

Prior to an incident, it is important to identify what time zone will be in use. From an evidentiary standpoint, the time zone does not really matter as long as it is consistent throughout the entire incident investigation.

- **Collected By**: Initials are sufficient for the log file.
- **MD5 Hash**: A comprehensive overview of hashing will be covered in later chapters. For now, suffice it to say that a hash is a one-way algorithm that is utilized to provide a digital fingerprint for a file. This hash will be recorded at the collection phase and after analysis to demonstrate that the file was not modified during the analysis phase. There are several ways to compute the hash. In this case, the MD5 hash can be computed using the installed hashing program: `md5sum` on Ubuntu. `md5sum` has several different options that can be accessed via the command line. For the help menu, type the following:

```
dfir@ubuntu:~$md5sum --help
```

That produces the following help menu:

```
File  Edit  View  Search  Terminal  Help
dfir@ubuntu:~$ md5sum --help
Usage: md5sum [OPTION]... [FILE]...
Print or check MD5 (128-bit) checksums.

With no FILE, or when FILE is -, read standard input.

  -b, --binary         read in binary mode
  -c, --check          read MD5 sums from the FILEs and check them
      --tag            create a BSD-style checksum
  -t, --text           read in text mode (default)

The following five options are useful only when verifying checksums:
      --ignore-missing  don't fail or report status for missing files
      --quiet          don't print OK for each successfully verified file
      --status         don't output anything, status code shows success
      --strict         exit non-zero for improperly formatted checksum lines
  -w, --warn           warn about improperly formatted checksum lines

      --help     display this help and exit
      --version  output version information and exit

The sums are computed as described in RFC 1321.  When checking, the input
should be a former output of this program.  The default mode is to print a
line with checksum, a space, a character indicating input mode ('*' for binary,
' ' for text or where binary is insignificant), and name for each FILE.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/md5sum>
or available locally via: info '(coreutils) md5sum invocation'
```

The MD5 hash can be calculated for the packet capture from the switch by simply entering the following command:

```
dfir@ubuntu:~$md5sum ping_capture
```

This produces the following output:

```
dfir@ubuntu: ~
File  Edit  View  Search  Terminal  Help
dfir@ubuntu:~$ md5sum ping_capture
7e559dc8eeeb66115566d93f96e7dfb8  ping_capture
```

Log files and packet captures should be transferred to a storage device as soon as possible. Once the collection is complete, a chain of custody forms should also be filled out for the external medium that contains the evidence files. From here, the files can be analyzed.

Summary

Evidence that is pertinent to incident responders is not just located on the hard drive of a compromised host. There is a wealth of information available from network devices spread throughout the environment. With proper preparation, a CSIRT may be able to leverage the evidence provided by these devices through solutions such as a SIEM. CSIRT personnel also have the ability to capture network traffic for later analysis through a variety of methods and tools. Behind all of these techniques, though, are legal and policy implications that CSIRT personnel and the organization at large need to navigate. By preparing for the legal and technical challenges of network evidence collection, CSIRT members can leverage this evidence and move closer to the goal of determining the root cause of an incident and bringing the organization back up to full operation.

This chapter discussed several sources of evidence available to incident response analysts. Logs from network devices, whether they report to a SIEM or through other methods, can give you an insight into what has transpired in the network. Packet captures provide details about the exact nature of network traffic. Finally, analysts must be prepared to acquire these sources of evidence in a forensically sound manner.

In the next chapter, the focus will shift from network evidence acquisition to acquiring volatile data from host-based systems.

Questions

1. What items are potential sources of network evidence?

   A) Switches
   B) Routers
   C) Firewalls
   D) All of the above

2. Network diagrams are important in identifying potential areas where network evidence can be acquired.

   A) True
   B) False

3. Which of the following is not a network forensic evidence capture tool?

   A) RawCap
   B) Wireshark
   C) WinPcap
   D) LogBeat

4. When conducting evidence acquisition, it is not important to record the hash value of the file.

   A) True
   B) False

Further reading

- Wireshark training: `https://www.chappell-university.com/`
- Introduction to Cisco IOS NetFlow – A Technical Overview: `https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html`

5
Acquiring Host-Based Evidence

Host systems are the targets of malicious actions far too often. Host systems represent a possible initial target so that someone can gain a foothold in the network or a pivot point for additional attacks, or create the end goal of threat actors. As a result, incident response analysts should be prepared to investigate these systems. Modern operating systems such as Microsoft Windows create a variety of evidentiary artifacts during the execution of an application, when changes to files are made, or when user accounts are added. All of these changes leave traces of activity that can be evaluated by incident response analysts. The amount of data that's available to incident response analysis is increasing as storage and memory in even the lowest-cost consumer systems continues to expand. Commonly available systems are routinely manufactured with extensive memory and storage in terabytes; there is a great deal of data that could assist incident responders with determining a root cause. As a result, incident response analysts should be prepared to acquire different types of evidence from systems for further analysis.

We will cover the following topics in this chapter:

- Preparation
- Order of volatility
- Evidence acquisition
- Acquiring volatile memory
- Acquiring non-volatile evidence

Preparation

In terms of preparation, incident response analysts should have the necessary tools at their disposal to acquire host-based evidence. The techniques that will be discussed within this chapter do not rely on any highly specialized technology, but rather on tools that can be acquired for little or no cost. It is critical that the tools that are selected for the acquisition of evidence are those that are provided by reputable sources, have been proven effective by other CSIRT personnel, and have been validated for efficacy prior to use. Outside of software, the only additional hardware that is required is external hard drives and common desktop computers.

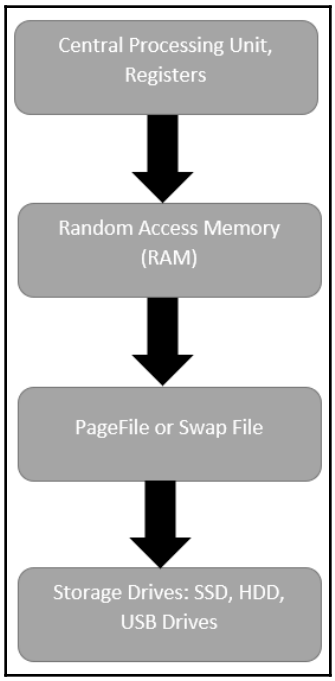
When supporting an enterprise environment, it is a good idea for incident response personnel to have a solid understanding of the types of systems that are commonly deployed. For example, in an enterprise that utilizes strictly Microsoft operating systems, the tools that are available should have the ability to support a wide range of versions of the Microsoft OS. In other circumstances, incident response personnel may support an enterprise where there is an 80/20 split between Microsoft and Linux systems; incident response personnel should be prepared with tools and techniques that support the acquisition of evidence.

Many of the tools and techniques that will be discussed in this chapter require administrator privileges. Incident responders should be provided with the necessary credentials to perform these tasks. It should be noted that analysts should only use existing accounts and that adding accounts to a possibly compromised system may make evidence inadmissible in a judicial proceeding. One technique is for incident response analysts to be given individual credentials that are enabled only during an incident. This allows the organization to separate the legitimate use of credentials from possible malicious ones. This also allows the incident response team to recreate their actions. It is worth noting that highly technical adversaries will often monitor the network they are attacking during an active compromise to determine whether they are being detected. Therefore, these credentials should not indicate that they are tied to the incident response analysts or other personnel investigating a possible breach.

Order of volatility

Not all evidence on a host system is the same. Volatility is used to describe how data on a host system is maintained after changes such as log-offs or power shutdowns. Data that will be lost if the system is powered down is referred to as volatile data. Volatile data can be data in the CPU, routing table, or ARP cache. One of the most critical pieces of volatile evidence is the memory currently running on the system. When investigating incidents such as malware infections, the memory in a live system is of critical importance. Malware leaves a number of key pieces of evidence within the memory of a system and, if lost, can leave the incident response analyst with little or no room to investigate. This can include such artifacts are registry data, command history, and network connections.

Non-volatile data is the data that is stored on a hard drive and will usually persist after shutting down. Non-volatile data includes **Master File Table** (**MFT**) entries, registry information, and the actual files on the hard drive. While malware creates evidence in memory, there are still items of evidentiary value in non-volatile memory. The following diagram shows the different levels of volatility of digital evidence that should be taken into account when determining the order of acquisition.



In the next section, we will learn how to collect evidence.

Evidence acquisition

There are a variety of methods that are used to not only access a potential evidence source but determine the type of acquisition that can be undertaken. To define these methods, it is important to have a clear understanding of the manner and type of acquisition that can be utilized:

- **Local**: Having access to the system under investigation is often a luxury for most enterprises. Even so, there are many times where incident response analysts or other personnel have direct physical access to the system.
- **Remote**: In a remote acquisition, incident response analysts leverage tools and network connections to acquire evidence. Remote acquisition is an obvious choice if the incident response analysts are dealing with geographical challenges. This can also be useful if incident response analysts cannot be onsite immediately.
- **Live acquisition**: A live acquisition of evidence occurs when the incident response analyst acquires the evidence from a system that is currently powered on and running. Some of the techniques that will be demonstrated in this chapter have to be deployed on a live system (for example, running memory). Completely acquiring digital evidence from a live system may be a technique that's necessary in high-availability environments where a suspected system cannot be taken offline. These techniques allow incident response analysts to acquire and analyze evidence to determine whether a system is indeed compromised.
- **Offline acquisition**: The offline acquisition method is the one that's often used by law enforcement agencies to preserve digital evidence on the hard drive. This technique requires that the system be powered down and the hard drive removed. Once the drive is accessed, specialized tools are utilized to acquire the hard drive evidence. There are some drawbacks to focusing strictly on offline acquisition. First is the loss of any volatile memory. Second, it may be time-consuming to acquire a suspect system's hard drive, image it, and process the image for investigation. This may create a situation where incident responders do not have any idea of what has transpired for more than 24 hours.

Depending on the type of incident and any constraints in time or geography, incident response analysts should be prepared to perform any of these types of acquisitions. The best-case scenario is for a CSIRT to have the ability to perform both live and offline acquisition on any suspect system. This provides the greatest amount of evidence that can be analyzed. In terms of preparation, analysts should have the necessary tools and experience to conduct evidence acquisition through any of these methods.

To perform local acquisition, incident response analysts require an external hard drive or USB drive with sufficient space for the capture of at least the running memory of the system or systems that are being investigated, along with other files if deemed necessary. In order to ensure the integrity of the evidence being collected, it is advisable to configure the USB drive into two partitions. The first partition should contain the necessary tools to perform the evidence acquisition, while the second should act as a repository for the evidence. This also allows the incident response analyst to move evidence to a more permanent form of storage and subsequently wipe the evidence partition without having to reinstall all the tools.

Evidence collection procedures

There are a number of parallels between digital forensics and other forensic disciplines such as trace evidence. The key parallel is that organizations acquiring evidence need to have a procedure that is sound, reproducible, and well documented. The following are some guidelines for proper collection of digital evidence:

- Photograph the system and the general scene. One of the key pieces of equipment that can save time is a small digital camera. While it may seem overkill to photograph a system in place, in the event that actions that have been taken by incident responders ever see the inside of a courtroom, having photos will allow for proper reconstruction of the events. One word of caution, though is to make sure that you utilize a separate digital camera. Utilizing a cell phone may expose the device to discovery in the event of a lawsuit or criminal proceeding. The best method is to snap all of the photos necessary and at a convenient time and place and transfer them to permanent storage.
- Determine whether the system is powered up. If the system is powered on, leave it on. If the system is powered off, do not power it on. A number of changes take place when turning a system on or off. In the event that the system is powered on, the volatile memory will be available for capture. In addition, in the case of Full Disk Encryption, leaving the system on will allow the responder to still acquire the logical disk volumes. If the system is turned off, preserving this state ensures that any evidence in the non-volatile memory is preserved. In the event that incident response personnel feel that the system may be a danger to other systems, simply remove the network connection to isolate it.
- Acquire the running memory. This is a critical piece of evidence that can produce a wealth of data concerning running processes, DLLs in use, and network connections. Due to this, procedures for acquiring memory will be covered extensively in this chapter.

- Acquire registry and log files. While these files are non-volatile in nature, having near-immediate access is beneficial, especially when investigating malware or other exploitation means.
- Unplug the power from the back of the system. In the event that the system is a laptop, remove the battery as well. This preserves the state of the system.
- Photograph the back or bottom of the system to capture the model and serial number. This procedure allows the incident response analyst to capture any information that's necessary for the chain of custody.
- Remove the cover to the system and photograph the hard drive to capture the model and serial number. Again, this aids in the reconstruction of the chain of custody.
- Remove the hard drive from the system and package it in an anti-static bag.
- Secure the drive in a sealable envelope or box. Anti-static bags will protect the hard drive, and the packaging should ensure that any attempt to open it will be evident. This can be facilitated through purpose-designed evidence bags or simple mailing envelopes that can be sealed with tape. The seizing analyst should sign on any seals. Furthermore, indicate the incident number, evidence number, date, time, and seizing analyst somewhere on the exterior of the packaging.
- Document all actions. Ensure that dates and times are recorded, as well as which incident response analyst performed the action. Incident reporting is often the last stage of any response. As a result, hours or even days can pass before analysts are able to record their actions. Due to this, pictures and notes that are taken during the initial seizure are invaluable when it comes to reconstructing the sequence of events.

In the next section, we will look at acquiring volatile memory.

Acquiring volatile memory

Traditional digital forensics, or what is often referred to now as **dead box forensics**, focuses on the hard disk drive that's been taken from a shut-down system acting as the primary source of evidence. This approach works well when addressing criminal activity such as fraud or child exploitation where image files, word processing documents, and spreadsheets can be discovered in a forensically sound manner. The issue with this approach is that, to properly acquire this evidence, the system has to be powered off, thereby destroying any potential evidence that could be found within the volatile memory.

As opposed to traditional criminal activity, incident responders will find that a great deal of evidence for a security incident is contained within the memory of a potentially compromised system. This is especially true when examining systems that have been infected with malware or exploited by utilizing a common platform such as Metasploit. Trace evidence is often found within the running memory of the compromised system. As a result, before powering down the system and removing the hard drive, it is critical that the running memory is acquired for processing.

There are several free and commercial tools that can be leveraged by incident response analysts to acquire the running memory. Which tool is used will often be dependent on the techniques and tools that will be used during the analysis phase. Two popular frameworks for a detailed analysis of memory images are rekall and volatility.

Running memory can be acquired in two ways. First, memory can be acquired locally via a USB device or other writable medium that is directly connected to the suspect system. The other method of acquiring memory is through a remote connection. This can be facilitated through the use of specialized software that performs the acquisition over a network connection.

Local acquisition

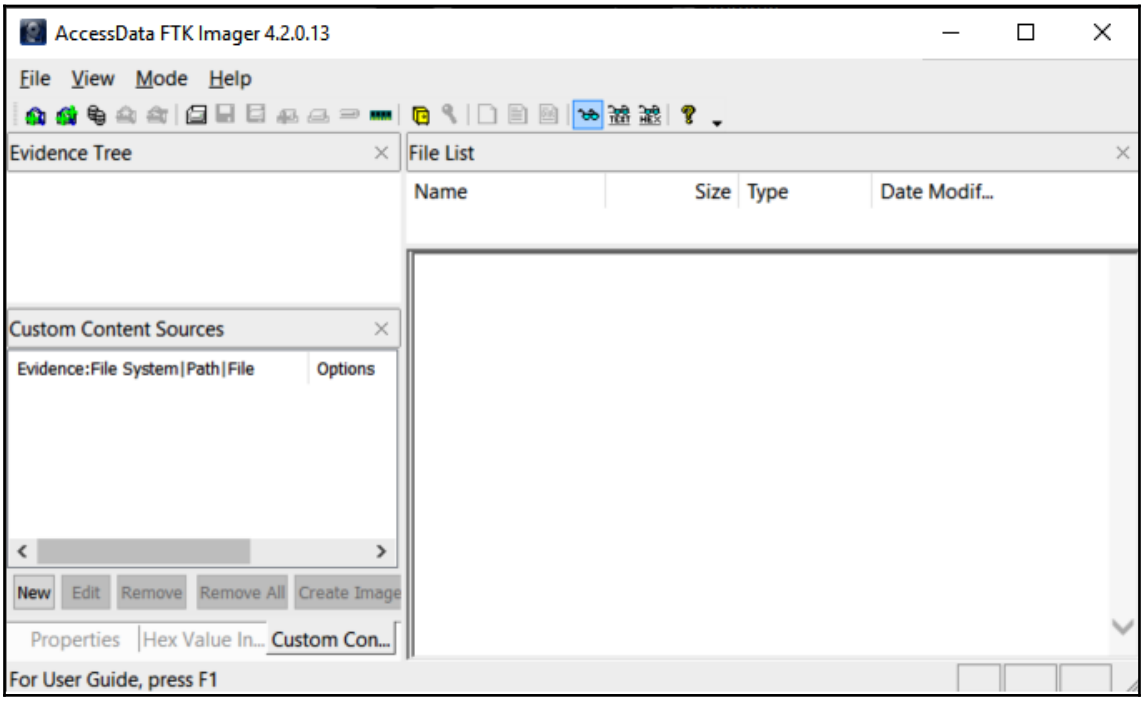
If an incident response analyst has physical access to a potentially compromised system, they have the option of acquiring the memory and other evidence locally. This involves the use of tools that are run from a USB device or a similar removable medium that is connected to the potentially compromised system. From there, the tools are run and the evidence is collected. Local acquisition is often conducted in conjunction with seizing the hard drive and other evidence from the system. There are several tools that are available for local acquisition. For the purposes of this book, three such tools – Access Data's FTK Imager, Google's WinPmem, and Belkasoft's RamCapturer – will be discussed.

When acquiring memory in this fashion, it is advisable to utilize an external drive with sufficient capacity for multiple files. Incident response analysts should make use of a USB device with two partitions. The first of these partitions contains the tools that are necessary to perform the memory acquisition, while the second partition will contain the evidence files. This way, incident response analysts can be sure that the evidence does not become co-mingled with their tools.

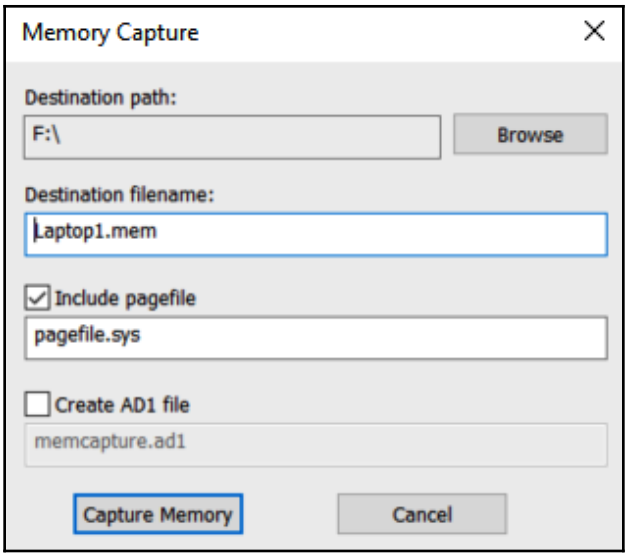
FTK Imager

Access Data's FTK Imager is a Windows software platform that performs a variety of imaging tasks, including acquiring the running memory of a system. The software can be downloaded at `https://accessdata.com/product-download`. Let's take a look at this platform:

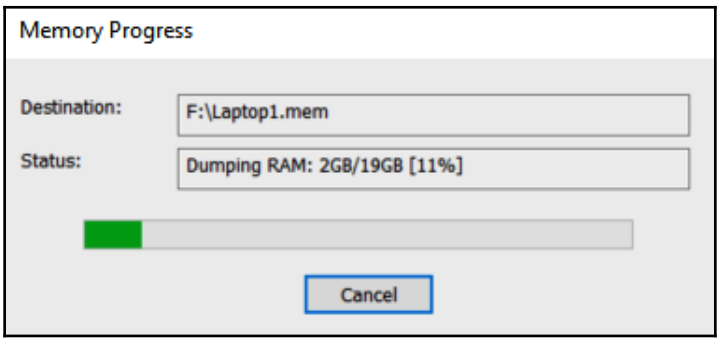
1. Once downloaded, install the executable in the **Tools** partition of the USB drive.
2. Open the `FTK Imager` folder and run the executable as administrator. (FTK Imager requires the use of drivers and, as a result, requires administrator privileges.) The following window will appear:



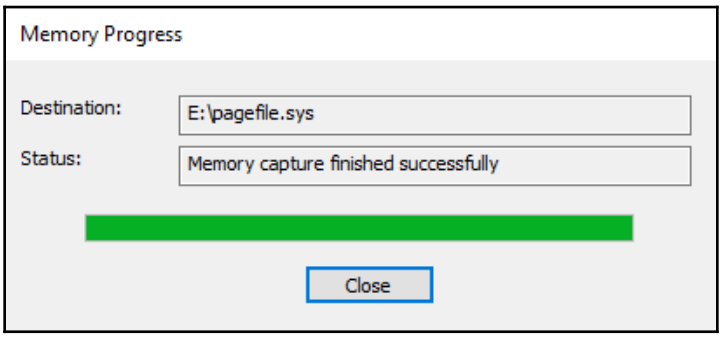
3. Click on **File** and then on **Capture Memory**. This opens the following window:



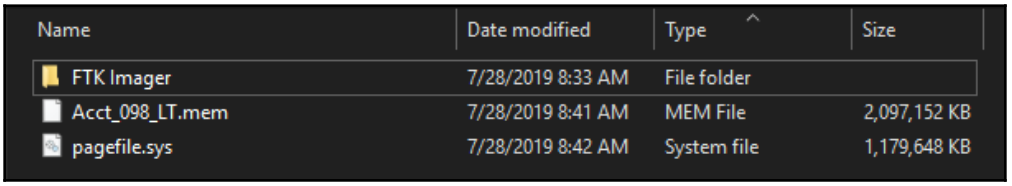
4. Browse to the **Evidence** partition of the USB drive attached to the system and provide a name for the capture file. This name should be a unique identifier such as `Laptop1` or `Evidence Item 1`. Also, check the **Include Pagefile** checkbox. There might not be any information of evidentiary value within the Pagefile, but it may become important later on during the investigation (the Pagefile will be discussed later on, in `Chapter 9`, *Analyzing System Storage*).
5. Finally, there is the option to create an AD1 file, that is, Access Data's proprietary file format. This file is for the analysis of this image using the FTK analysis program. For the purposes of this book, the standard output is sufficient for the analysis that will be performed.
6. Once the configuration details have been set, click on **Capture Memory** and the following screen will appear:



After running this, FTK Imager will indicate whether the memory capture was successful or not:



Examining the evidence partition reveals the two files, as shown in the following screenshot:



Note that the `.mem` file is approximately 2 GB. The RAM on the system that was utilized in this demonstration has 16 GB of memory. This situation is common where the `.mem` file is not the exact size of the entire RAM space.

Winpmem

As we mentioned previously, some memory acquisition tools work better with different memory analysis tools. In the case of the rekall memory analysis tool, there are several memory acquisition tools provided by the same organization that created it. The PMEM tools that are available are used to capture raw memory from Linux, macOS, and Windows systems. These tools are available at the rekall website: `http://releases.rekall-forensic.com/`.

In the following demonstration, the target system is the same one that was utilized in the FTK Imager demonstration. As a result, the WinPmem tool, which is specifically designed to capture the memory of Windows systems, will be utilized.

Starting with version 2.0.1, the default output for the WinPmem tool is the **Advanced Forensic Framework 4** (**AFF4**) file format. This format was created to allow for a number of separate data sources and workflows. This open source format is utilized for digital forensics evidence and other associated data. Let's get started:

1. To acquire the physical memory of the target system, open up Windows Command Prompt as an administrator. Typing `D:\winpmem-2.1.exe -h` will produce the following help menu:

```
D:\>winpmem-2.1.exe -h

USAGE:

   winpmem-2.1.exe  [-l] [-u] [--write-mode] [--mode <MmMapIoSpace,
                    PhysicalMemory, PTERemapping>] [--driver <Path to
                    driver.>] [--format <map, elf, raw>] [-m] [-p
                    </path/to/pagefile>] ...  [-V] [-d] [-v] [-t] [-i
                    </path/to/file/or/device>] ...  [-e <string>] [-o
                    </path/to/file>] [-c <zlib, snappy, none>] [--]
                    [--version] [-h] </path/to/aff4/volume> ...
```

More information about the AFF4 file format is available at `http://www.aff4.org/`.

2. Next, configure WinPmem to acquire the memory of the system by typing the following:

```
D:\winpmem-2.1.exe --format raw -o e:\Laptop1
```

This command tells WinPmem to acquire the raw memory and output it to a folder that will be created on the evidence partition of the USB drive in use. The preceding command will produce the following output:

```
D:\>winpmem-2.1.exe --format raw -o e:\Laptop1
Driver Unloaded.
CR3: 0x00001AA000
 7 memory ranges:
Start 0x00001000 - Length 0x0009C000
Start 0x00100000 - Length 0x00002000
Start 0x00103000 - Length 0xBE2FE000
Start 0xBE889000 - Length 0x1BD7E000
Start 0xDA770000 - Length 0x00775000
Start 0xDBAFF000 - Length 0x00001000
Start 0x100000000 - Length 0x31E800000
```

WinPmem then runs through the entire memory structure. During this process, it will produce the following output:

```
Creating output AFF4 Directory structure.
Dumping Range 0 (Starts at 1000, length 9c000)
Dumping Range 1 (Starts at 100000, length 2000)
Dumping Range 2 (Starts at 103000, length be2fe000)
Dumping Range 3 (Starts at be889000, length 1bd7e000)
Dumping Range 4 (Starts at da770000, length 775000)
Dumping Range 5 (Starts at dbaff000, length 1000)
Dumping Range 6 (Starts at 100000000, length 31e800000)
 Reading 0x8000  0MiB / 16272MiB 0MiB/s
 Reading 0x4398000  67MiB / 16272MiB 255MiB/s
 Reading 0x89b0000  137MiB / 16272MiB 275MiB/s
 Reading 0xd288000  210MiB / 16272MiB 276MiB/s
 Reading 0x11858000  280MiB / 16272MiB 274MiB/s
 Reading 0x15f48000  351MiB / 16272MiB 283MiB/s
 Reading 0x1a998000  425MiB / 16272MiB 295MiB/s
 Reading 0x1f3f0000  499MiB / 16272MiB 296MiB/s
 Reading 0x23cb8000  572MiB / 16272MiB 289MiB/s
 Reading 0x283c8000  643MiB / 16272MiB 283MiB/s
 Reading 0x2cb68000  715MiB / 16272MiB 285MiB/s
 Reading 0x310d0000  784MiB / 16272MiB 276MiB/s
 Reading 0x346f8000  838MiB / 16272MiB 206MiB/s
 Reading 0x38c70000  908MiB / 16272MiB 276MiB/s
 Reading 0x3cbe8000  971MiB / 16272MiB 252MiB/s
 Reading 0x41240000  1042MiB / 16272MiB 280MiB/s
 Reading 0x45580000  1109MiB / 16272MiB 267MiB/s
```

In conclusion, a review of the output file reveals that the entire physical memory is contained within a single file, along with other files as part of the AFF4 file container.

WinPmem is an easy tool to use, but one of the other great advantages is that there are versions for Linux and macOS systems, as well as Microsoft OS. This allows incident response analysts to become familiar with one tool across all the operating systems they might encounter in their organization.

The one drawback of the use of WinPmem is that the AFF4 format must be converted into a raw memory image if the analyst or incident responder wants to examine the memory capture with volatility. To convert an AFF4 file into RAW format, the following command needs to be run with WinPmem:

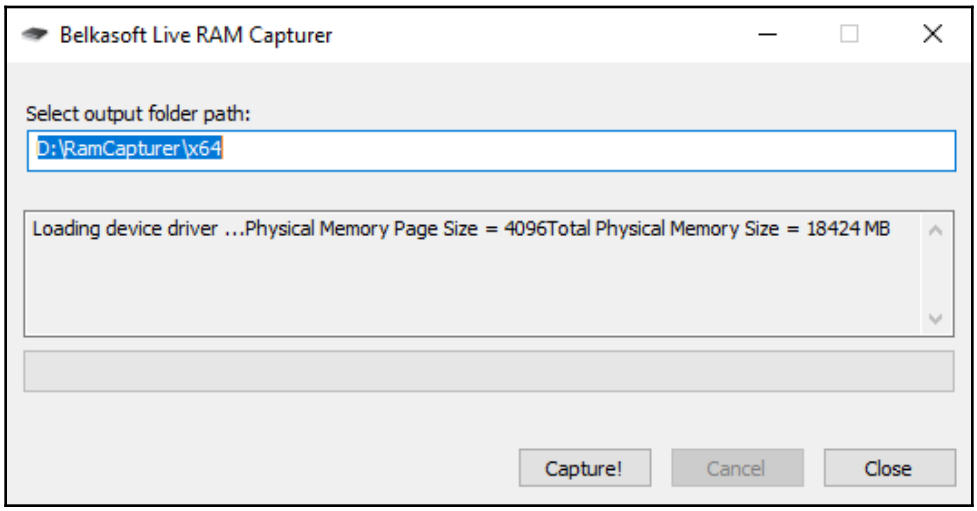
```
D:\winpmem-2.1.exe D:\Laptop1.aff4 -e PhysicalMemory -o Laptop1.raw
```

In the event the analyst will be using volatility as the primary platform for memory analysis, it is preferable to use FTK Imager. Another option would be to use a tool such as RAM Capturer.

RAM Capturer

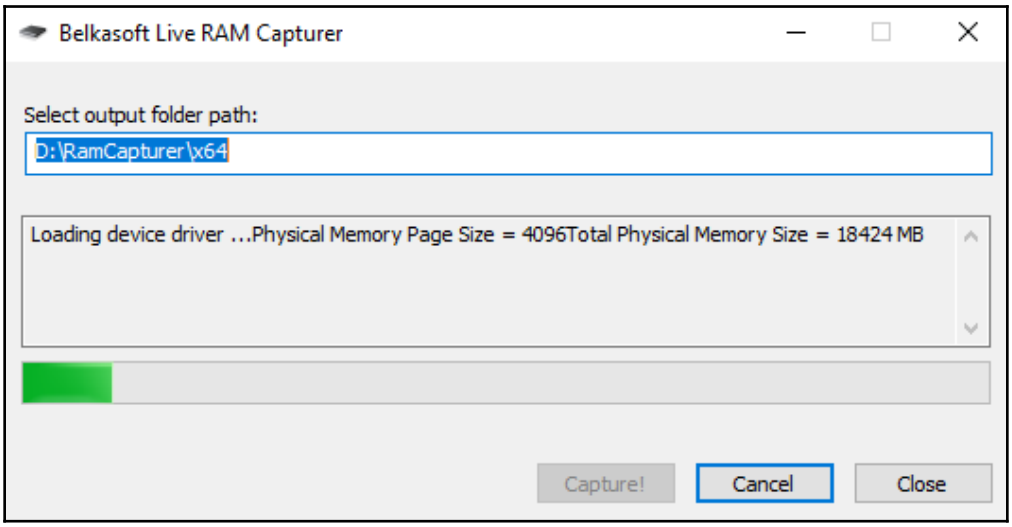
RAM Capturer is a free tool provided by the software company Belkasoft. RAM Capturer is a simple tool to utilize and, like FTK Imager and WinPmem, it can be run from a USB. Let's take a look:

1. Running the application will cause the following window to appear:

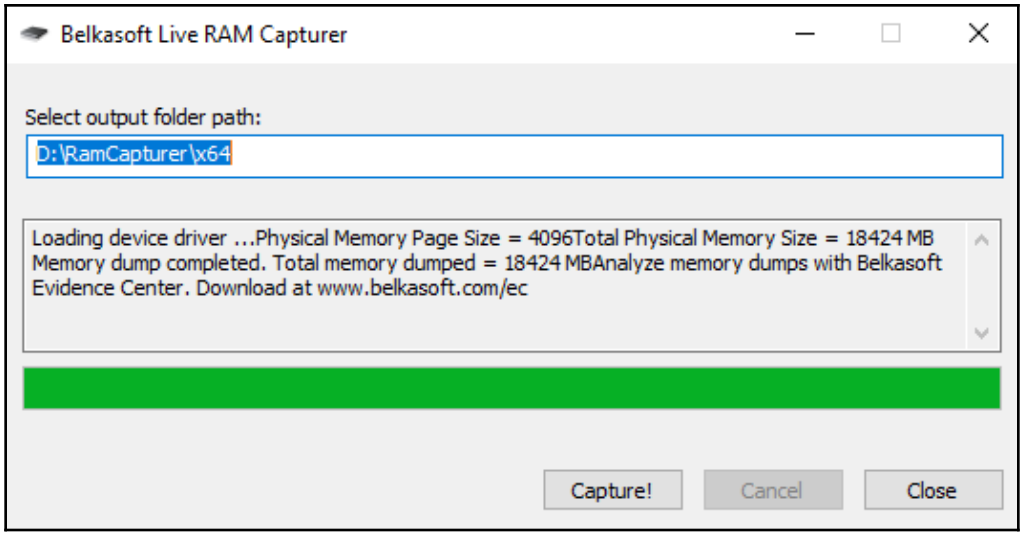


The only input that's required for acquisition is to place the path of the folder where the memory image should be placed.

2. Once the output has been set, click on the **Capture!** button and let it run:



3. Once RAM Capturer completes, the following message will appear:



When looking at memory acquisition tools, the best approach is to capture as much data as possible as efficiently as possible. Tools such as FTK Imager are highly reliable and allow for acquiring not just memory but also other key pieces of evidence. However, at times, this may not be possible, and responders will have to use a USB key with a lightweight tool such as RAM Capturer. The best option is to determine the type of forensic tools that will be used to examine the evidence and then select the appropriate tool to acquire memory.

When looking at acquiring memory, another key factor with these tools that makes them useful is that they can also be leveraged in the event that responders do not have physical access to the suspect system.

Remote acquisition

The preferred method for the acquisition of memory is through direct contact with the suspect system. This allows for adaptability by incident response analysts in the event that a tool or technique does not work. This method is also faster at obtaining the necessary files since it doesn't depend on a stable network connection. Although this is the preferred method, there may be geographical constraints, especially with larger organizations where the incident response analysts are a plane ride away from the location containing the evidence.

In the case of remote acquisition, incident response analysts can leverage the same tools that are utilized in local acquisition. The one change is that incident response analysts are required to utilize a remote technology to access the suspect systems and perform the capture. As with any method that is utilized, incident response analysts should ensure that they document any use of remote technology. This will allow for proper identification of legitimate versus suspect connections later on.

Winpmem

Winpmem can be deployed on remote systems through native applications such as Remote Desktop or PsExec. Once installed on the remote system, the output of WinPmem can be piped to another system utilizing NetCat. For example, suppose that the incident response analyst is utilizing a system located at `192.168.0.56`. If the analyst is able to access the compromised host via PSExec or RDS, they can establish a NetCat connection back to their machine by using the following command:

```
C:/winpmem-2.1.exe - | nc 192.168.0.56 4455
```

The preceding command tells the system to perform the capture and send the output via NetCat to the incident response analyst workstation over port `4455`. The drawback of this technique is that it requires access to Command Prompt, as well as the installation of both NetCat and WinPmem. This may not be the best option if the incident response analyst is dealing with a system that is suspected of being compromised.

Virtual machines

Other systems that incident response analysts should prepare to address are virtual machines. The one distinct advantage that virtual systems have over physical systems is their ability to maintain the current state by either performing a snapshot of the system or simply pausing. This allows incident response analysts to simply copy the entire file over to an evidence drive for later analysis. It is recommended that analysts ensure that they conduct a hash of each component of the virtual machine pre and post copy to ensure the integrity of the evidence.

One key feature of popular virtualization software such as VMware is that the virtual machine utilizes two files for the running memory. The first of these is the **Virtual Memory** (**VMEM**) file. The VMEM file is the RAM or physical memory of the virtual machine. The second file is the **VMware Suspended State** (**VMSS**) file. The VMSS file contains the files that are saved as part of the suspended state of the virtual machine. Let's take a look at this:

1. To acquire the running memory from a VMware virtual machine, pause the system.
2. Second, transfer the VMSS and VMEM files to a removable media source such as a USB. VMWare software will often include the `Vmss2Core.exe` application as part of the installation process. This application combines the VMSS and VMEM files into a single `.dmp` file that can be analyzed with forensic tools. Both these files are required to create a complete memory capture.
3. To create the `.dmp` file, run the following command:

```
C:\Program Files (x86)\VMware\VMware Workstation>vmss2core.exe
suspect.vmss suspect.vmem
```

From here, the responder will have the necessary `.dmp` file to conduct analysis.

Acquiring non-volatile evidence

Although there is a great deal of data running in memory, it is still important to acquire the hard drive from a potentially compromised system. There is a great deal of evidence on these devices, even in the case of malware or other exploitation. Hard drive evidence becomes even more important when examining potential incidents such as internal malicious action or data loss. To ensure that this evidence is available and can be utilized in a court of law, incident responders should be well versed in the procedures we've discussed in this chapter.

In certain circumstances, incident responders may want to acquire two key pieces of data from suspected compromised systems before shutting down a running system. While not volatile in nature, the registry keys and event log files can aid analysts during their investigation. Acquiring these files from an imaged hard drive is largely dependent on the time that's needed to image and then process the entire hard disk drive. As a result, there are a few techniques that can be leveraged to acquire these key pieces of evidence.

In the event that analysts have access to the system, they can utilize the command line to access the log files by running the following command:

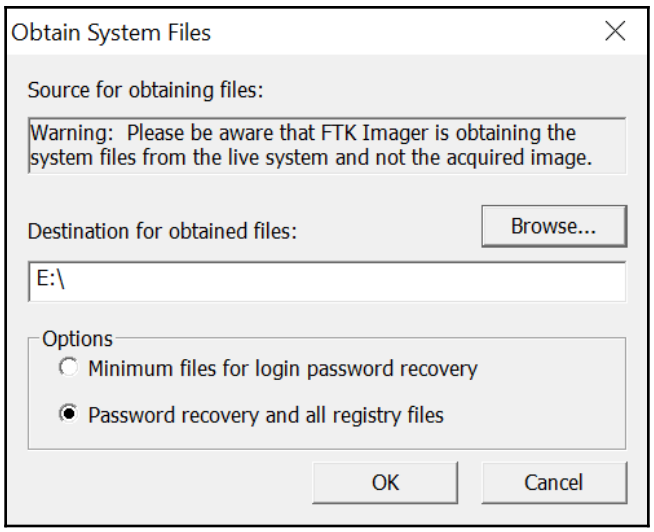
```
C:\wevtutil epl<Log Type> E:\<FileName>.evtx
```

This command can be repeated for security, application, and system logs.

FTK Imager also allows for the capture of registry key settings and other information that can aid in an investigation. Let's take a look:

1. Open FTK Imager and navigate to the **File** tab.

2. Click on **Obtain Protected Files**. The following dialog box will appear:



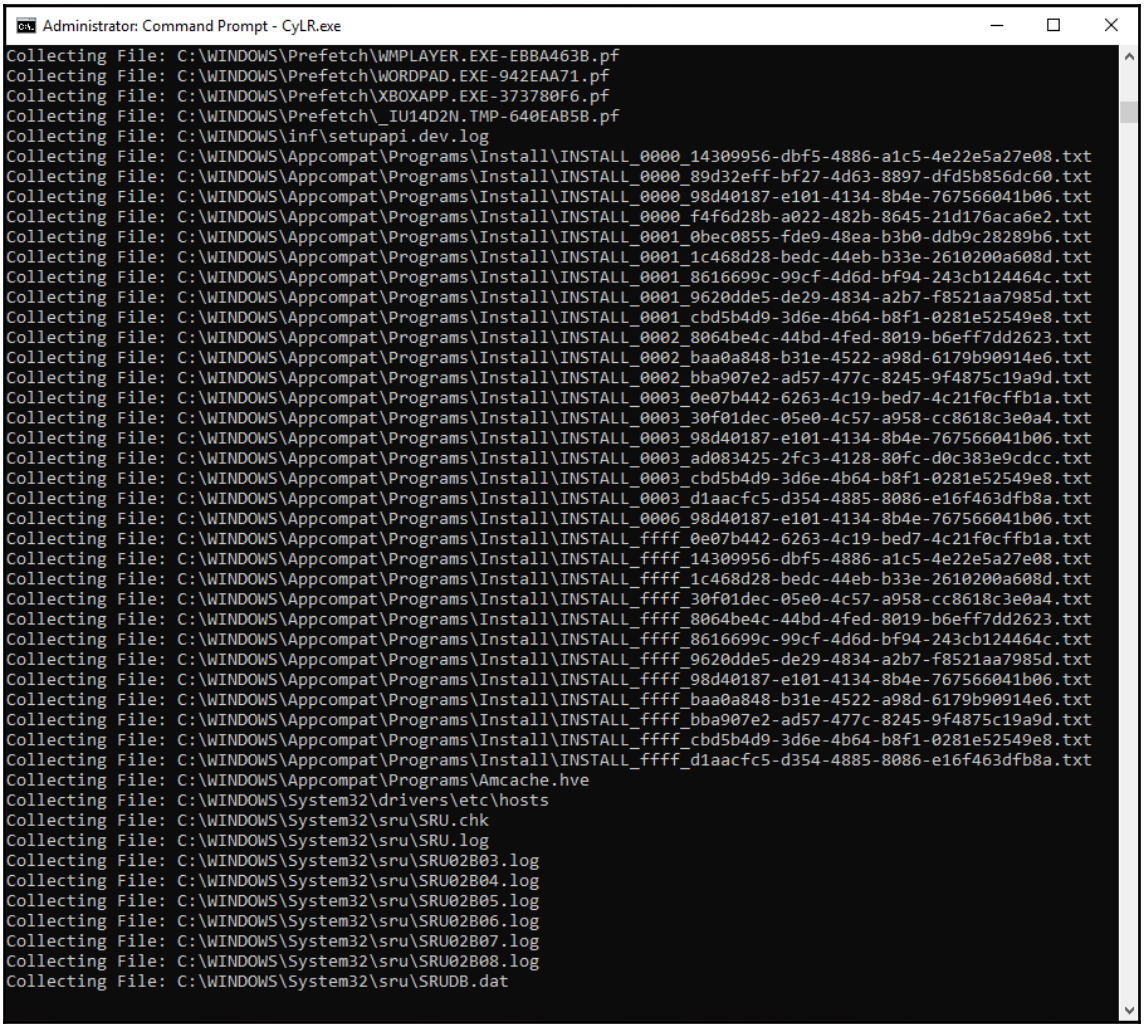
3. Click on **Browse...** and navigate to the evidence file location.
4. Next, click the radio button for **Password recovery and all registry files** and click **OK**. Once the tool completes, the registry and password data will be transferred to the evidence folder. This command directs FTK Imager so that it obtains the necessary registry files to recover the system passwords. These include the user, system, SAM, and `NTUSER.DAT` files. From here, analysis can take place before the imaging process. This allows for a more rapid response to an incident.

CyLR.exe

An open source tool that aids responders in this type of acquisition is the `CyLR.exe` application. This standalone executable, available at `https://github.com/orlikoski/CyLR/releases`, can be run from a USB or on the system. It is a small application but can acquire a great deal of evidence that can be leveraged as part of the initial investigation or possibly triage. Another key feature of `CyLR.exe` is its ability to send the data that's been acquired to a remote system either for storage or processing, as we will demonstrate in `Chapter 10`, *Analyzing Log Files*.

To acquire the non-volatile log files and other protected files, navigate to the `CyLR.exe` executable via Command Prompt and run it as administrator. The output directory containing the evidence files will be placed in the same directory as the `CyLR.exe` application. This is handy if you're running this from a USB device as the output will land directly on the USB.

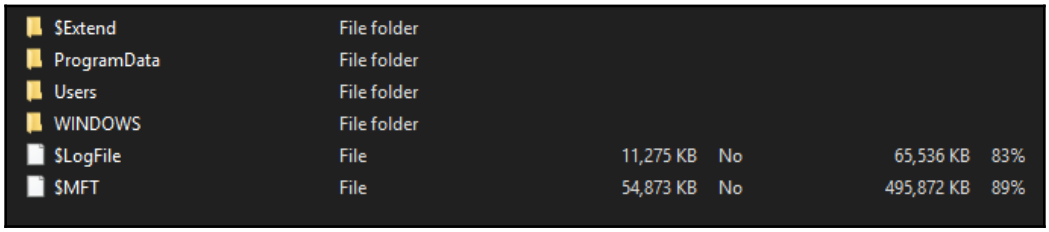
While the `CyLR.exe` application is running, the responder will be able to see the individual files that are being acquired:



Depending on the processor and available RAM, `CyLR.exe` can be expected to run for a few minutes. Afterward, the following message will appear:

```
Extraction complete. 0:09:14.4347905 elapsed

C:\Users\IRProactive-WKST\Desktop>
```

Finally, a check of the directory where `CyLR.exe` was run will reveal a compressed file with the name of the system as the filename. Uncompressing the file reveals the extensive evidence that was collected:

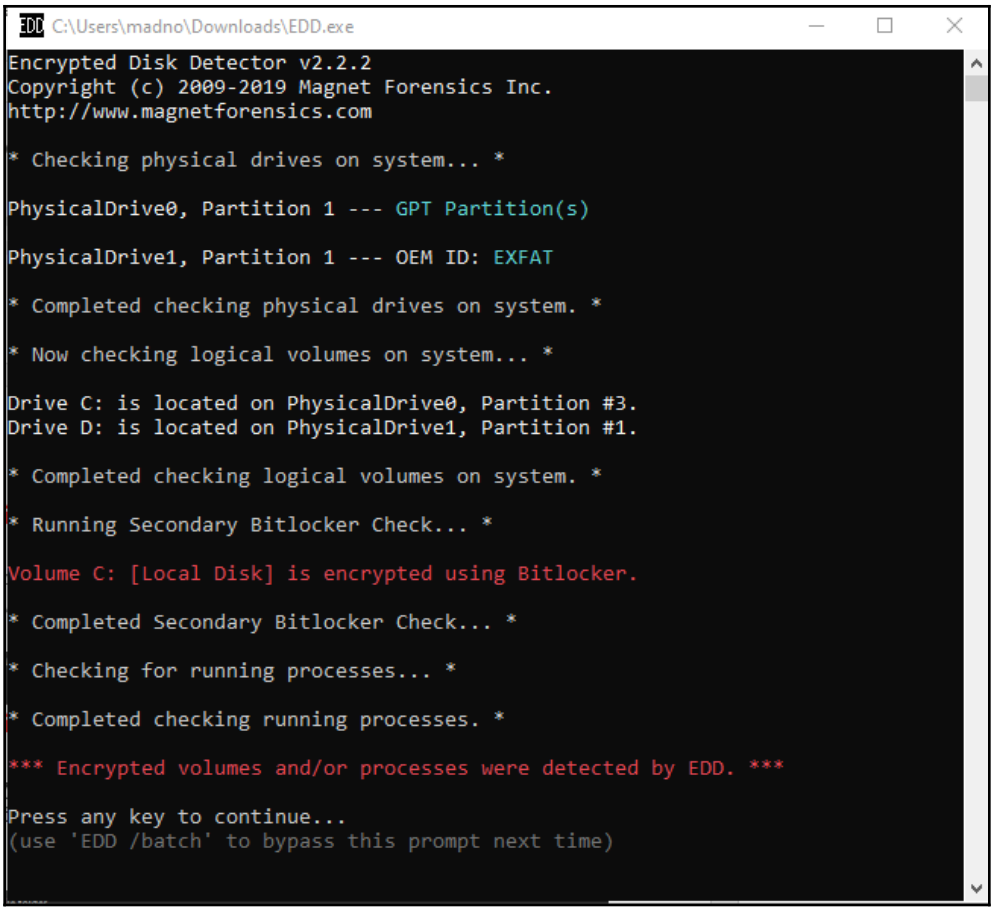


The output contains the log files, registry files, and master file table, which will be important in later chapters. The ability to acquire this data from a simple tool is a major advantage of using `CyLR.exe` to acquire evidence before a system is shut down.

Checking for encryption

One area of concern for responders is the use of **Full Disk Encryption** (**FDE**). Windows systems administrators are likely to be familiar with FDE tools such as Bitlocker, which is now part of the Windows OS. Outside of that, there are a number of tools such as VeraCrypt that allow users to encrypt individual files all the way to entire volumes. As part of the acquisition process, responders should check for any encryption.

Tools such as Endpoint Disk Detector from Magnet Forensics determine whether there are any encrypted volumes. This tool, available for free at `https://www.magnetforensics.com/resources/encrypted-disk-detector/`, can determine what, if any, encryption is being used. Simply running the executable as administrator starts the application. The following data is then presented:



By doing this, the responder has determined that the `Volume C` is encrypted using `BitLocker`. As a check, the responder should ensure that the appropriate BitLocker key is available if necessary. If an encryption key cannot be obtained, then the responder should acquire the logical volume while the system is still running.

Summary

Proper evidence handling starts the overall process that aims to determine the root cause of an incident and potentially identify the responsible party. For evidence to be of any use in an incident investigation, it has to be acquired in a sound manner. Incident responders should have a solid foundation in understanding the various types of acquisition and the tools and techniques that are available and apply those tools and techniques to the various situations that may arise. By applying solid techniques and properly documenting their actions, incident responders will be in a position to utilize the evidence to not only determine the root cause of an incident but also to back up their actions in a courtroom if necessary.

The next chapter will look at capturing non-volatile data, that is, data contained on the disk drive.

Questions

1. When looking at the order of volatility, which of the following evidence categories should be acquired first?

   A) Random Access Memory
   B) PageFile or swap file
   C) Central Processing Unit, registers
   D) Storage drive

2. It is good practice to acquire the PageFile with RAM if you're using FTK Imager.

   A) True
   B) False

3. Remote acquisition of digital evidence cannot be achieved using what?

   A) Remote desktop services
   B) PsExec
   C) USB
   D) NetCat

4. When recreating the memory from a virtual system, responders should acquire both the VMSS and VMEM file.

   A) True
   B) False

Further reading

- CFR and Order of Volatility: `https://blogs.getcertifiedgetahead.com/cfr-and-order-of-volatility/`
- *The Advanced Data Acquisition Model (ADAM): A process model for digital forensic practice*: `https://researchrepository.murdoch.edu.au/id/eprint/14422/`
- *Best Practices In Digital Evidence Collection*: `https://digital-forensics.sans.org/blog/2009/09/12/best-practices-in-digital-evidence-collection/`

6
Forensic Imaging

One critical task that incident response analysts often have to perform is imaging digital evidence. As we discussed in prior chapters, a great deal of evidence related to an incident can be found within log files, memory, and other areas that can be acquired relatively quickly. In some incidents, such as internal malicious activity (for example, fraud, industrial espionage, or data leakage), a more detailed search for evidence may be required. This evidence includes master file table entries, files, and specific user data that is contained on the hard drive of a suspect system. In the event that incident response analysts encounter such circumstances, they will be required to obtain an image of a suspect drive. As with any aspect of digital forensics, obtaining a usable and court-defensible image depends on the appropriate tools, techniques, and documentation.

This chapter will explore the fundamental concepts of digital imaging and the preparation and tools that are needed to acquire a forensically sound image of a physical drive or other logical volume. More specifically, we will cover the following topics:

- **Understanding digital imaging**: Imaging a storage drive is a process where details matter. This section provides a solid foundation on forensic imaging, how it is accomplished, the various types of digital imaging process, and the various proprietary file formats.
- **Tools for imaging**: Like much of the previous material we covered, there are several tools available to the responder for imaging drives. Having an understanding of these tools provides responders with knowledge about which tool to apply to the appropriate incident.
- **Preparing a stage drive**: Just as important as learning how to handle the evidence drive, having a forensically sound stage drive to which the evidence will be imaged is critical. Responders will be walked through how to prepare this item.
- **Write blockers**: Write blockers are critical components and ensure that evidence is not tainted during the imaging process. In this section, responders will be exposed to physical and software write blockers.

- **Imaging techniques**: The main part of this chapter will focus on techniques that are available to responders who are called upon to image an evidence drive.

While this chapter presents some very technical and process-driven material, it is important that responders understand imaging. This process is critical to producing images that can be relied on for root-cause analysis and potential courtroom use.

Understanding forensic imaging

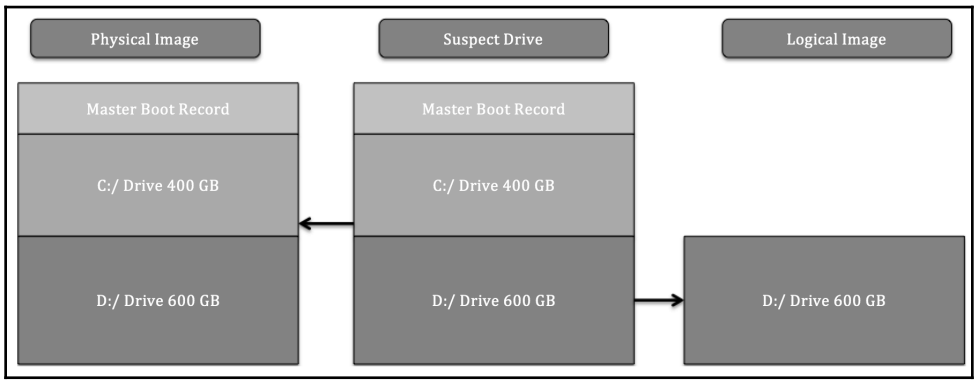
Having a solid understanding of the facets of forensic imaging is important for incident response analysts. Having an understanding of the tools, techniques, and procedures involved ensures that evidence is handled properly and that analysts have confidence in the evidence they've acquired. In addition, understanding the necessary terminology allows analysts to accurately prepare reports and testify as to their findings if the need arises.

One of the first concepts that should be understood is the difference between forensic imaging and copying. Copying files from a suspect hard drive or other medium only provides analysts with the actual data associated with that file. Imaging, on the other hand, allows the analyst to capture the entire drive. This includes areas such as slack space, unallocated space, and possibly access deleted files. Imaging also maintains metadata on the volume, including file timestamps. This becomes critical in the event that a timeline analysis is conducted to determine when specific files were accessed or deleted.

Often, the terms cloning and imaging are utilized in place of each other. This is a common improper use of terminology in the Information Technology realm. When cloning a drive, a one-to-one copy of the drive is made. This means that the drive can then be inserted into a system and booted. Cloning a drive is often done to make a fully functional backup of a critical drive. While a cloned drive contains all the necessary files, it is cumbersome to work with, especially with forensic tools. As a result, an image file is taken. An image of a drive contains all the necessary files; its configuration permits a detailed examination while utilizing forensic tools.

The second concept that needs to be understood is the types of volume that can be imaged. Volumes can be separated into physical or logical volumes. Physical volumes can be thought of as containing the entirety of a hard drive. This includes any partitions, as well as the master boot record. When imaging a physical volume, the analyst captures all of this data. In contrast, a logical volume is a part of the overall hard drive. For example, in a hard drive that is divided into the master boot record and two partitions, a logical volume would be the `D` drive. When imaging a logical volume, the analyst would only capture data contained within the `D` drive.

The following diagram illustrates data that is captured while imaging either a physical or logical volume:



The type of incident that is being investigated largely dictates the type of imaging that is conducted. For example, if an analyst is able to identify a potentially malicious file being executed from the `D` drive and is intent on only capturing that data, it might be faster to acquire a logical image of only that volume. Furthermore, logical acquisition may be necessary in cases where **Full Disk Encryption** (**FDE**) is being used. Without the encryption key, logically acquiring files while the system is running is often the only option that's available.

The one key drawback that a logical image has is that it will not capture unallocated data or data that is not part of the filesystem. Deleted files and other trace evidence will not be part of a logical image. In cases where activities such as employee misconduct are suspected, the analyst will need to trace as much activity as possible, so a full image of the physical volume will be conducted. Time isn't a necessary factor here.

In `Chapter 4`, *Collecting Network Evidence*, we discussed the acquisition of evidence such as log files and running memory from a live or powered up system. In much the same way, incident response analysts have the capability to obtain a logical volume from a running system. This technique is referred to as live imaging. Live imaging may be the best option if a potentially compromised system cannot be taken offline, say, in a high-availability production server, and potential evidence is located within a logical volume.

Dead imaging is performed on a system that has been powered down and the hard drive removed. In this type of imaging, the analyst is able to capture the entire disk, including all the volumes and the master book record. This may become necessary in incidents where analysts want ensure that they capture the entirety of the source evidence so that there is no location that hasn't been examined.

One final aspect of forensic imaging that an analyst should have knowledge of is the types of image files that can be created and leveraged during an investigation. There are several types of image file, some of which are very specialized, but for the purposes of this book, we will focus on the two most common types of evidence file that analysts will most likely create and work with during an incident:

- **Raw images**: A raw image file contains only the data from the imaged volume. No additional data is provided in this type of image, although some imaging tools such as FTK Imager include a separate file with imaging information. Raw image outputs include the `.raw`, `.img`, and `.dd` extensions. Some software, such as the Linux `dd` command, provides a flexible option when speed and compatibility with forensic tools may be an issue.
- **EnCase evidence files**: The EnCase evidence file, or E01 or EX01 file, is a proprietary file format that was developed by OpenText as part of their EnCase forensic tools in 1998. This format was based on the **Expert Witness Format** (**EWF**), which was found in the ASR Data's Expert Witness Compression Format. The E01 file contains metadata about the image. The metadata that is contained in both the header and footer captures and stores information about the drive type, operating system, and timestamps. Another key feature of the E01 file is the inclusion of a **Cyclical Redundancy Check** (**CRC**). This CRC is a file integrity verification that takes place after every 64 KB of data is written to the image file. This CRC ensures the integrity of the preceding block of data over the entire image file. Finally, the E01 file contains the MD5 hash within the footer of the file. The following diagram illustrates which components of the E01 file are created during the imaging process:

An E01 file is the preferred output for law enforcement and legal entities as it combines the ability to verify evidence integrity with software features such as compression. The E01 file also works with a variety of forensic software.

The information that has been presented here is really just an overview of some of the core concepts of imaging. As a result, there are a great many details concerning forensic imaging that couldn't be included in this book. Having a detailed understanding of forensic imaging will allow an incident response analyst to prepare an accurate report, while also being able to describe how their actions produced the output that served as the foundation of their analysis.

While this book focuses on the RAW and EnCase evidence file formats, there are two other common file formats that responders may encounter, especially if they are working with other organizations:

- **Advanced Forensic File Format (AFF4)**: The AFF4 image file format is an open source project that was developed for the storage of digital evidence and other data. This format is used primarily with the Pmem volatile memory acquisition toolset.
- **Access Data Evidence File (AD1)**: Here, the owner of the **Forensic Tool Kit** (**FTK**) suite of tools uses AD1 as their proprietary evidence file.

With the overall concept of forensic imaging addressed, the next step is to look at the tools available to capture a forensically sound image.

Imaging tools

While there is no court or legal body that certifies digital forensic imaging tools, there are several methods and associated tools that represent best practices when acquiring disk evidence. Let's go over these now:

- **FTK Imager**: FTK Imager is provided as a free software application by Access Data. This GUI-based application allows for the forensically sound acquisition of logical and physical volumes, memory, and other protected files and outputs those images in a variety of formats. In addition, FTK Imager Lite is a self-contained application that can be run on removable media for the acquisition of digital evidence from running systems (this will be covered in detail later in this chapter).
- **EnCase Imager**: Provided by Guidance Software, EnCase Imager is another forensic application that allows responders to acquire digital evidence from a variety of systems. Similar to FTK Imager, EnCase Imager can also be run on an external drive for the acquisition of running systems.
- **AFF4 Imager**: The AFF4 Imager is a Command-Line executable that serves as the basis for tools such as WinPmem. AFF4 Imager can be used to acquire logical and physical disks such as EnCase or FTK Imager. One advantage of AFF4 Imager is that it can be used to carve out files based on time creation, and to slit volumes and decrease imaging time with compression.

- **dd**: An old Linux standby. In some instances, the Linux `dd` command which is used to copy files and volumes, can be used to image drives or volumes. Responders will most likely use the `dd` command when using Linux-based forensic platforms for evidence acquisition.
- **Virtualization tools**: With the wide adoption of virtualization, responders are most likely going to have to acquire at least a portion of their evidence from virtual systems. There is an advantage to this, though: the entire system can be offloaded for analysis. Depending on the virtualization software, acquisition can be accomplished by pausing the system and offloading the entire directory containing the system. This can also be accomplished using the snapshot feature of many virtualization software platforms.

The imaging tools you decide to use will depend on the organization, your training and experience, and what other forensic tools are in use. For example, if an organization uses the Forensic Tool Kit for analysis, it may best to use FTK Imager as part of the process. With any imaging tool, it is good practice to ensure that the tool functions properly and that responders have been adequately trained in its use.

Once an imaging tool is selected, the next step is to ensure that the other hardware is ready. This includes ensuring that the destination of stored media is correctly prepared.

Preparing a stage drive

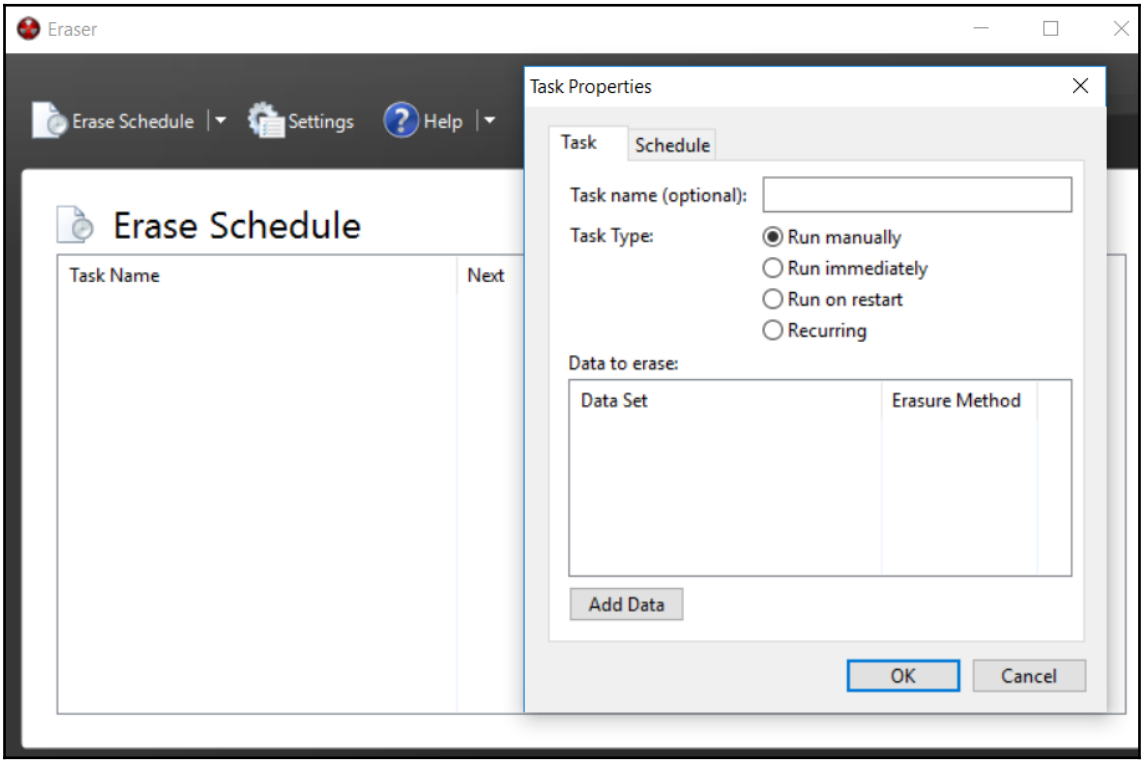
Beyond having the necessary hardware and software to perform forensic imaging, it is critical to prestage a location to hold the image or evidence file. For incident response teams, the best thing to utilize as an evidence repository is an external USB or FireWire disk drive. This allows for a degree of portability as incident responders may have to investigate an incident offsite or at a variety of locations without the benefit of a forensic laboratory.

There are two tasks that need to be performed on evidence drives prior to their use. The first is to ensure that the repository is free of any data. Incident response teams should have a policy and procedure that dictate that an evidence drive be wiped prior to each use. This includes drives that are new in box. This is due to the fact that a number of manufacturers ship drives with backup software or other data that needs to be removed prior to use. Wiping further ensures that previously utilized drives are free of any trace data from another incident. This ensures that the evidence collected on to a properly wiped drive is not contaminated with unrelated data.

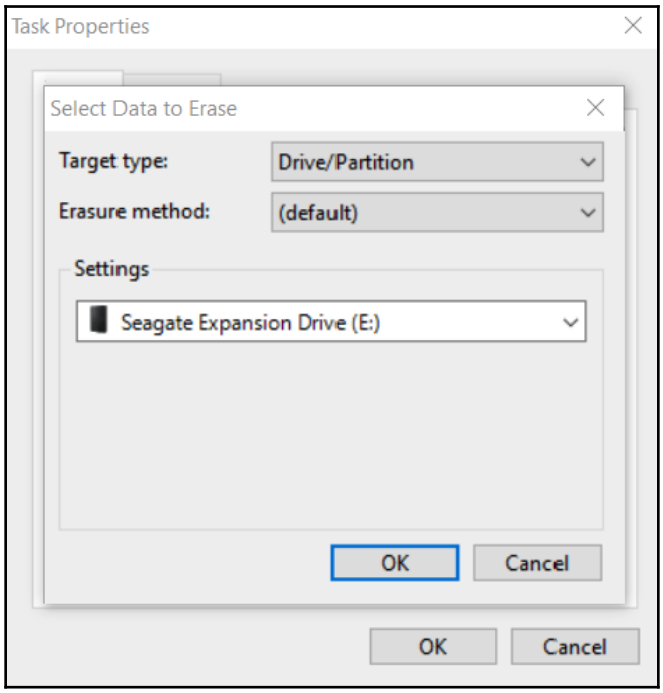
This is easily accomplished through a wiping program. There are a number of programs, both free and commercial, that can be utilized for this. For example, the Eraser program from Heidi Computers is a freeware wiping utility that can be utilized for both file and volume wiping (Eraser can be downloaded at `https://eraser.heidi.ie/`).

In the following example, a 2 TB external hard drive will be erased and prepared for use as an evidence drive. The following sequence should be repeated every time a drive is going to be placed into a state that can be utilized for incident investigation:

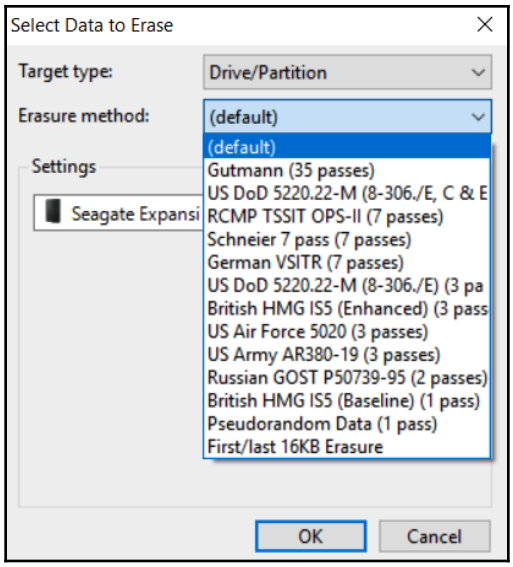
1. Start the Eraser application. In the GUI, click **Erase Schedule** and then **New Task**:



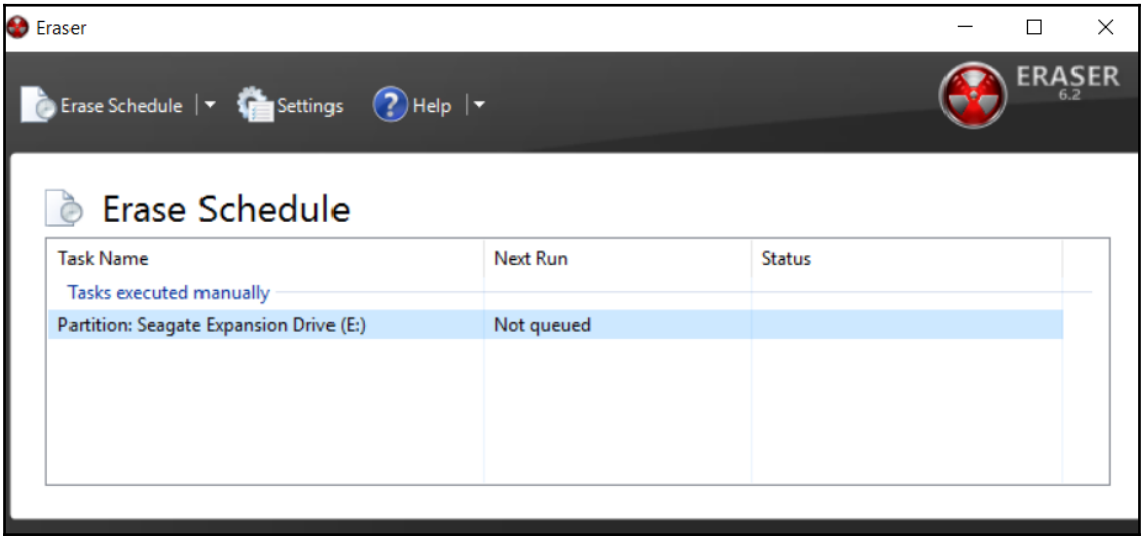
2. Now, a task name can be assigned. This is helpful when you wish to properly document the erasure of the evidence drive. Click the **Add Data** button. This will open another window:



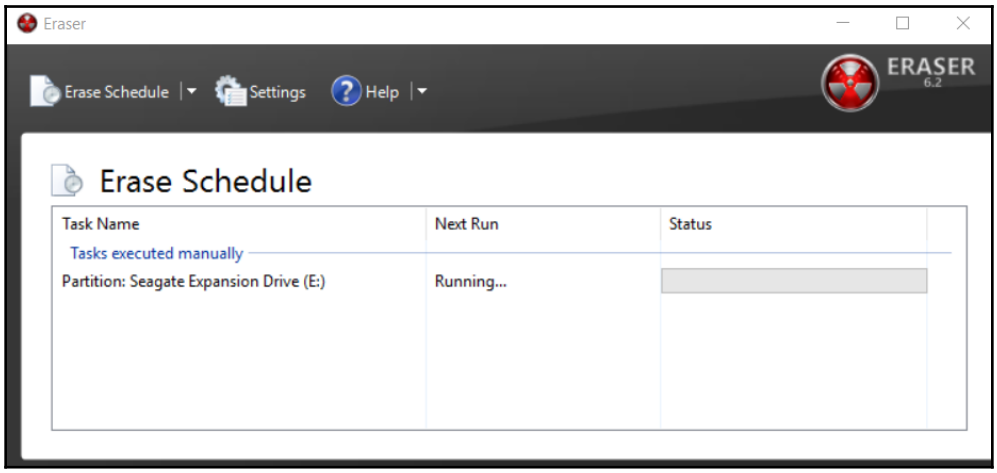
3. For **Target type**, select **Drive/Partition**. In the **Settings** area, there will be a drop-down list of partitions and drive letters. Pay very close attention to the drive letters that are assigned to the various drives and ensure that the external drive that requires wiping is selected. In this case, a new Seagate external HDD is being utilized. Finally, select an **Erasure method**. There are several different options for wiping drives. In this case, the **US DoD 5220.22-M (8-306./E) (3 Pass)** wiping option is selected:



4. Click **OK**. Now the wiping task will be listed in the **Erase Schedule**:



5. Right-click the **Partition: Seagate Expansion Drive (E:)** task and click **Run Now**. This will start the wiping process. As we mentioned previously, ensure that the correct evidence drive is being wiped:



Depending on the size of the drive and the system that is performing the wipe, this process can take hours or even days. Once completed, the incident response analyst should capture any wiping information that verifies that the evidence drive has been properly wiped. This is important information to include in a written forensic analysis report as it demonstrates that the incident response analyst took appropriate measures to ensure that any evidence files were free from corruption or co-mingling with other files on the evidence drive.

It is recommended that incident response analysts have several drives available and that these drives be pre-wiped before any incident. This will allow incident response analysts to immediately utilize a wiped drive instead of having to wipe a drive onsite, which wastes time that would be better spent on incident-related activities.

A second preparation step that can be undertaken is to encrypt the evidence drive. Software such as VeraCrypt or another disk encryption platform can be utilized to encrypt the partition of the evidence drive that contains the evidence files. Incident response analysts dealing with confidential information such as credit cards or medical records should encrypt the evidence drive, regardless of whether it leaves the facility or not.

There are two methods that can be leveraged to encrypt the evidence drive. The first is to utilize encryption software on the forensic workstation that is utilized in the imaging process. This approach is limited to imaging on drives that have been removed from the system and imaged on dedicated systems that have the encryption software installed. A second option is to include the encryption software on the evidence drive. In the previous section, an evidence drive was divided into two partitions. One partition is set aside for evidence files, while the second partition is utilized for tools such as those used for dumping memory files or imaging. In this scenario, the encryption software can be loaded in the tools partition and the drive can be encrypted during the evidence imaging process. This limits the number of changes that are made to the system under investigation.

Once a drive is prepared, another layer of protection is needed to ensure that no changes are made to the suspect system during the imaging process. To ensure that no changes are made, responders should be familiar and know how to use write blockers.

Using write blockers

A key tenet of digital forensics is to ensure that no changes are made to digital evidence while processing and examining it. Any change, no matter how slight, has the potential to bring the entire examination into question. There is a distinct possibility that the evidence may even be excluded from legal proceedings if the responder is unable to articulate how they ensured that the evidence was not tainted during the examination. As a result, it is important to understand how write blockers maintain the integrity of digital evidence.

Write blockers come in two different types. The first of these is a software write blocker. This software sits between the operating system and the evidence. These are often part of any digital forensic tools that are used during the examination phase. They ensure that there is read-only access to the evidence file and that, during the examination, no changes have been made to the evidence. For example, the FTK Imager tool, which will be explored extensively in this chapter, ensures that the acquisition of digital evidence is done without any writes to the disk.

Another type of write blocker is a physical or hardware write blocker. As its name indicates, this is a physical piece of hardware that sits between the evidence drive and the system performing the acquisition. Data is allowed to pass from the evidence disk to the analysis system but not the other way around. The use of this device allows responders to clearly demonstrate that no evidence was altered during the acquisition phase.

Which type of write blocker is used is largely dependent on the type of acquisition that is being conducted. Ideally, responders should choose tools and techniques that clearly demonstrate that they took every reasonable precaution to ensure that the evidence has not been altered. Doing so significantly decreases the risk that the evidence will be excluded from any legal proceedings, and also affords the responder the ability to rely on the evidence while making a root-cause determination.

With a properly staged drive and write blocker in place, responders are now able to move on and image evidence drives.

Imaging techniques

Once a proper repository has been configured for the image file, the incident response analyst is ready to acquire the necessary evidence. Responders will encounter suspect systems that are either powered on or have been shut down. Based on the state that responders find the suspect system, they will have to utilize one of the following techniques. In any incident, no matter what technique is utilized, incident responders should be prepared to properly document their actions for any subsequent forensic report.

Dead imaging

Dead imaging is conducted on media that is not powered on and, in the case of hard drives, removed from the potentially compromised system. In terms of evidence preparation, this method is most comprehensive as it allows for the complete preservation and analysis of a physical volume. There are several methods and tools available, both commercial and freeware, that allow for proper imaging. In addition to software, often incident response analysts will make use of a hardware write blocker. These devices ensure that no changes are made to the suspect media. As we discussed in `Chapter 1`, *Understanding Incident Response*, it is critical to be able to demonstrate to a court of law that no changes were made to the original evidence.

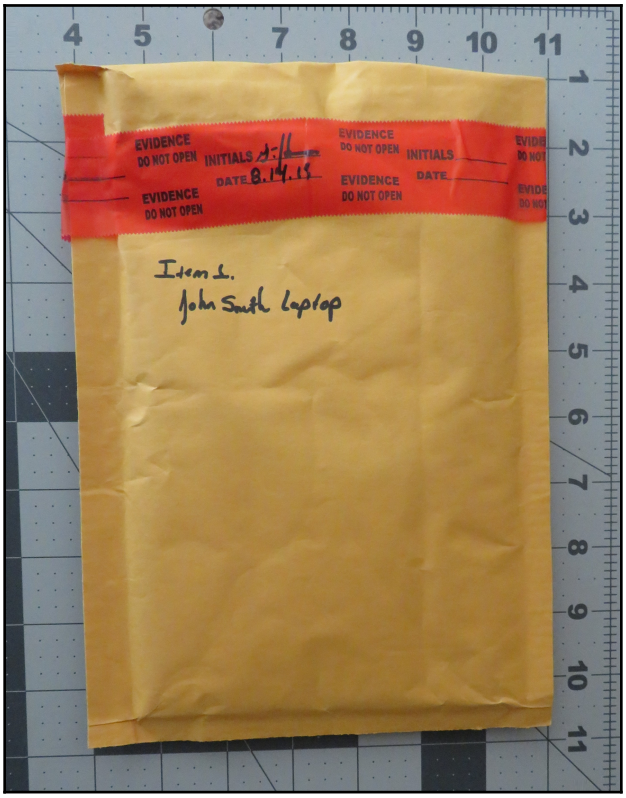
One advantage of imaging a hard drive or other digital media in this manner is that the process can be predefined and repeatable. Having a predefined process that is formalized as part of incident response planning and the procedure itself ensures that evidence is handled in a forensically sound manner.

One tool that is extremely useful in Dead Imaging is FTK Imager. This tool, provided by Access Data, is a forensically sound platform for acquiring a disk image.

Imaging using FTK Imager

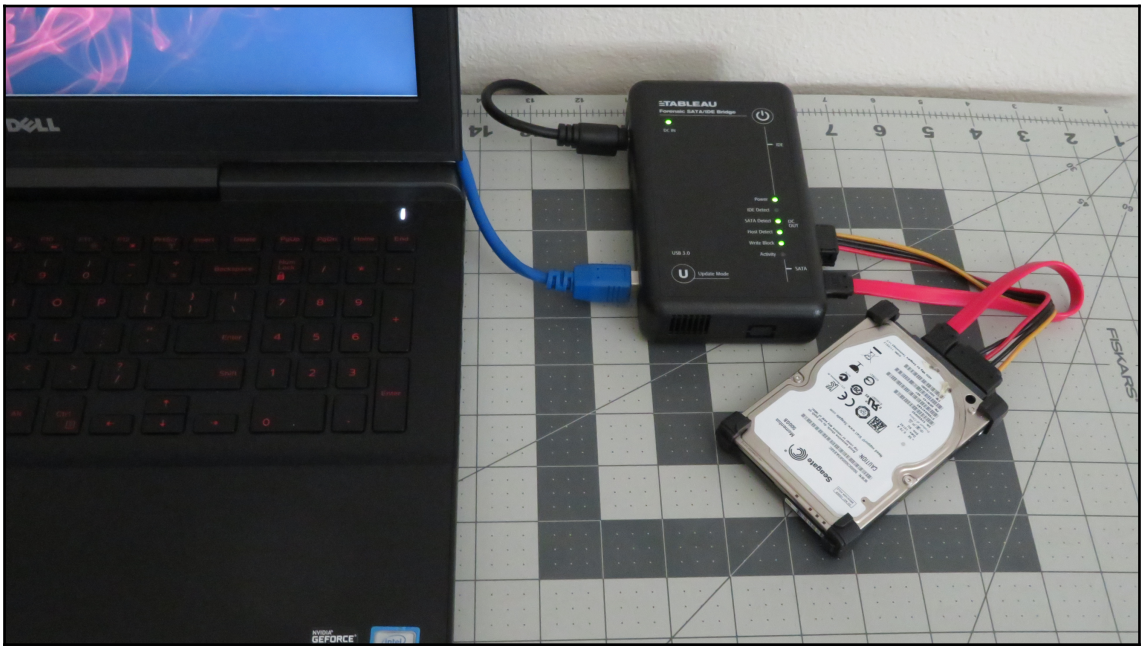
The following process uses a hard drive and FTK Imager to produce a forensically sound image for analysis. Rushing or deviating from these steps may create a situation where the responder may not be able to rely on the evidence's integrity, thereby making potential evidence unreliable.

1. The first step is to physically inspect the evidence. There are two primary focal points that should be inspected. The first is the chain of custody form. Any time that you are taking custody of the evidence, you should have access to the form, ensure that all steps are properly documented, and complete the entry with your information.
2. Then, you need to inspect the evidence packaging to ensure that any seals have not been breached. One quick way to document this is to take a photo of the evidence in the original packaging:

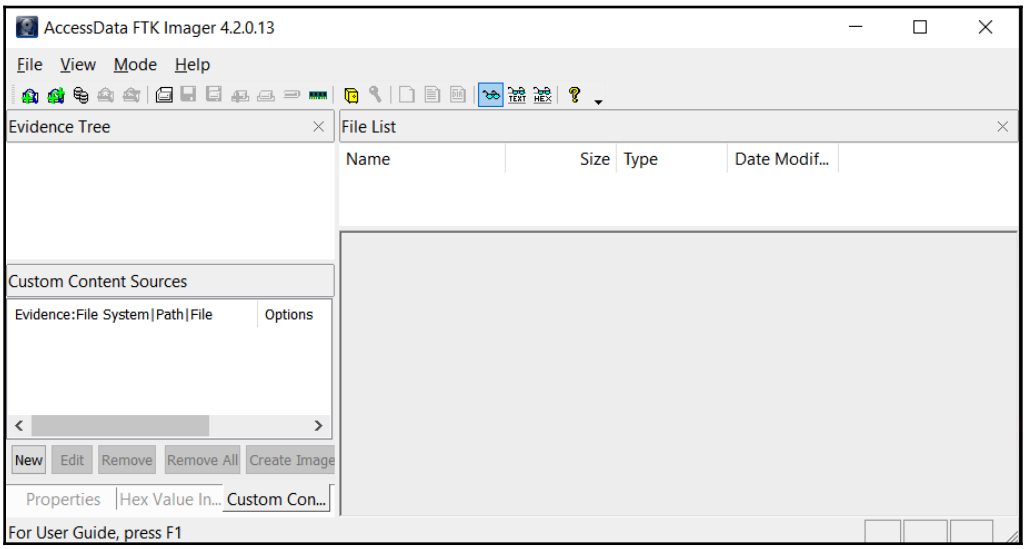


3. In the preceding photo, we have captured all the information concerning the piece of evidence and demonstrated that, prior to imaging, the integrity of the evidence has been maintained. After the seal has been broken, you need to take another photo of the contents of the packaging:

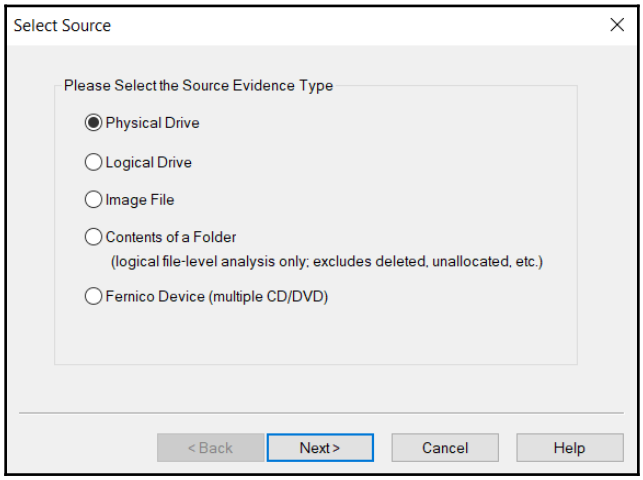
4. Once a photo of the piece of evidence has been taken, you should ensure that it matches the chain of custody form. Errors can occur in an incident, and this is one way to ensure that mistakes in the chain of custody are corrected as early as possible. By confirming the chain of custody, any mixups can be rectified. The next step is to configure the physical write blocker. In this case, a Tableau TK35u USB 3.0 Forensic IDE/SATA Bridge Kit is utilized as a physical write blocker. The suspect drive is attached via the included SATA drive adapter and a FireWire connection is made to the imaging laptop. When utilizing a physical write blocker, ensure that the device indicates proper functioning:



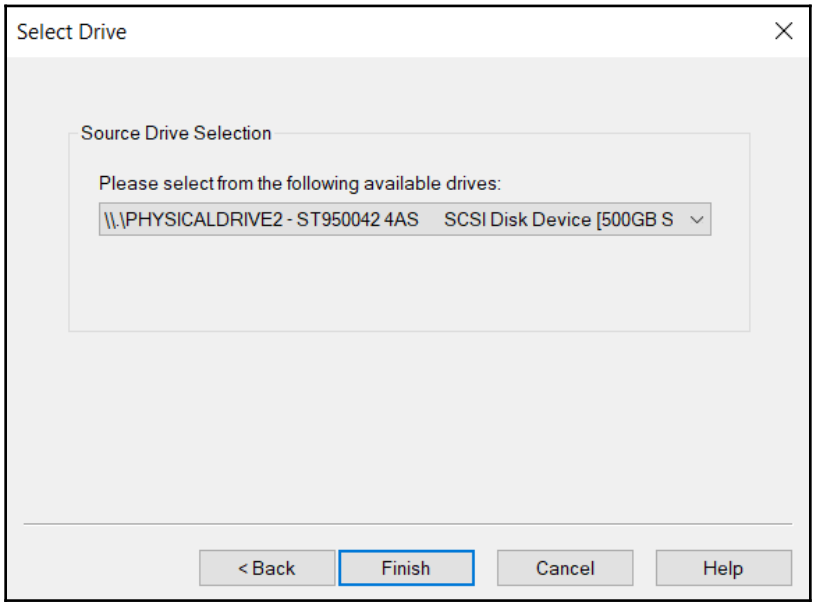
5. With the physical write blocker in place, the suspect drive is now ready for imaging. In this example, the FTK Imager freeware application will be used. FTK Imager requires administrator privileges to run. Open the executable; the following screen will appear:



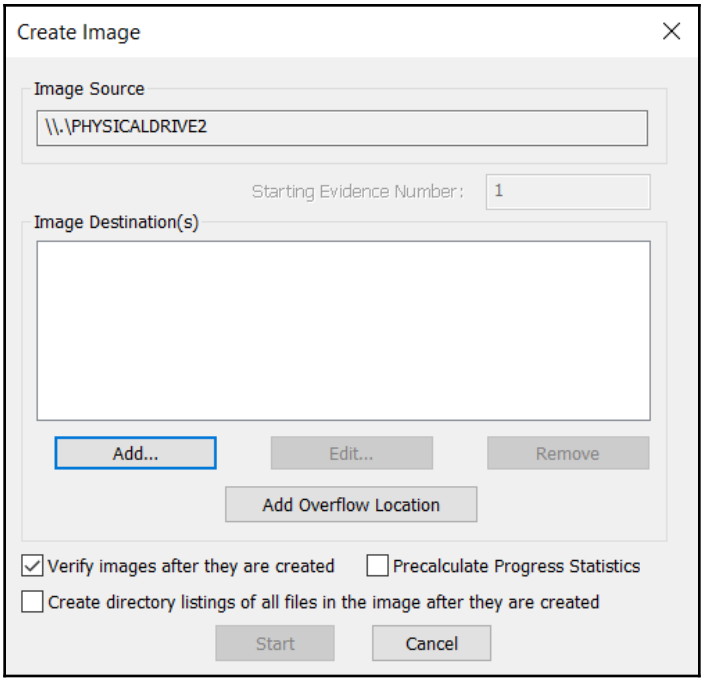
6. Click on **File** and then **Create Disk Image**. This will open a window where you can select the media source. In this case, select **Physical Drive** so that the entire drive, including the master boot record, will be captured for further analysis. Then, click **Next**:



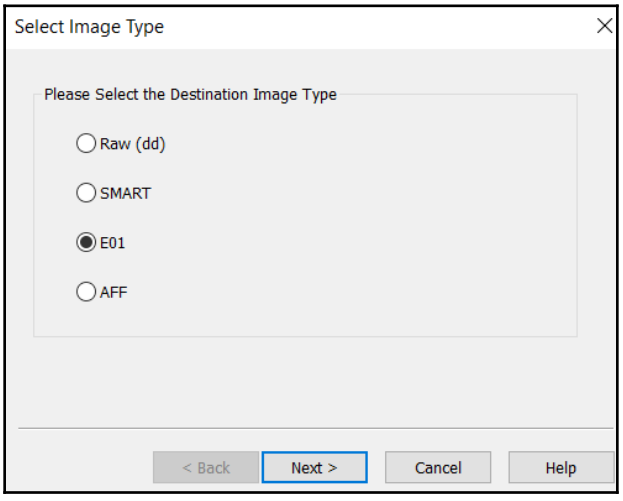
7. The next window allows the analyst to select which drive will be imaged. Incident response analysts should pay close attention to ensure that they are imaging the correct device since all devices that are visible to the operating system are listed. Here, you need to pay attention to the storage space of the drives to differentiate between the suspect and image drives. In this case, four separate drives are listed. Two are drives contained within the imaging laptop. Another drive is the destination drive. In this case, the third drive, labeled `\\.\PHYSICALDRIVE2`, is the correct suspect drive. Highlight this drive and click **Finish**:



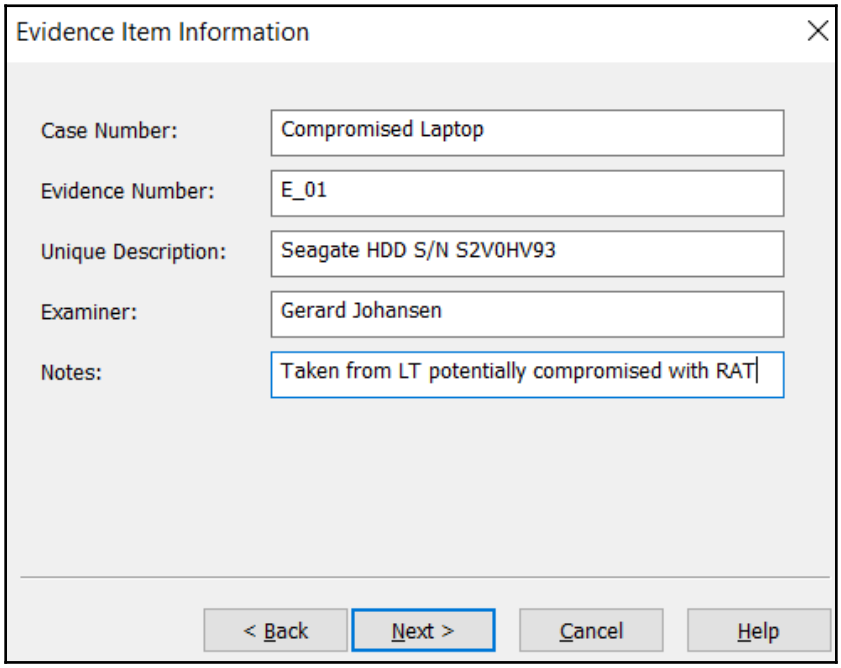
8. Once the suspect drive has been selected, set the destination drive. Click **Add...**:



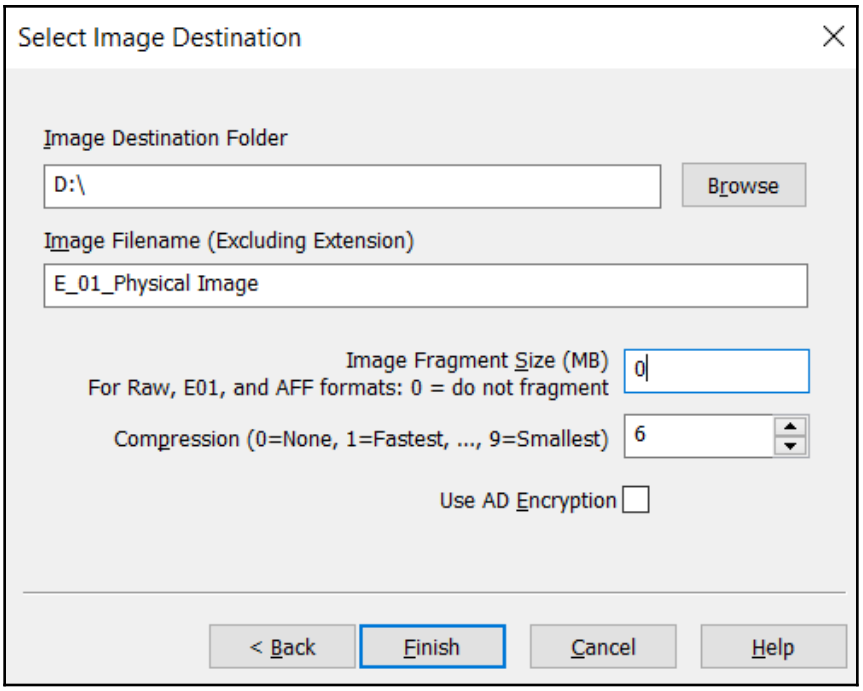
9. At this point, choose the type of image file you want to create. Four options are available. In this case, **E01** needs to be selected. Click **Next**:



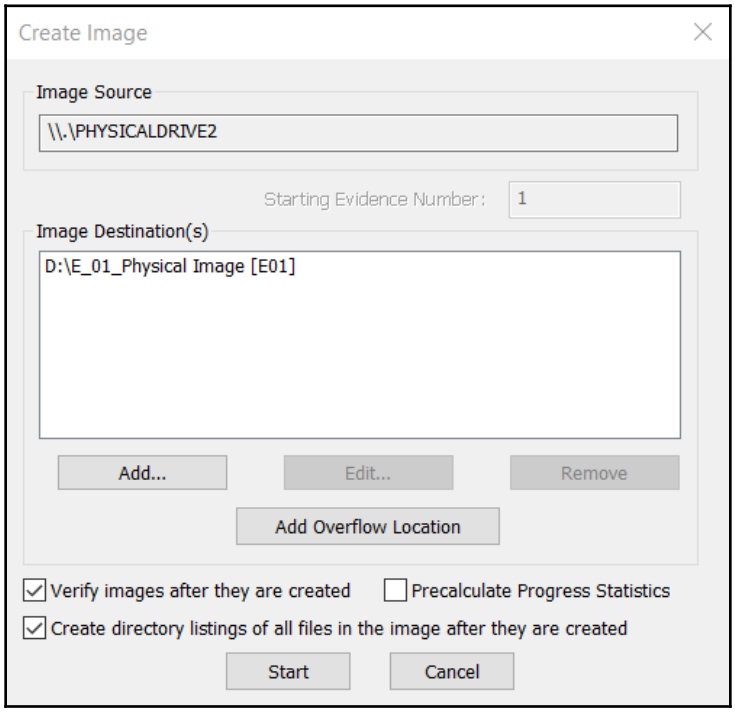
10. In the next window, enter information specific to the image. We will discuss reporting in `Chapter 11`, *Writing the Incident Report*. For now, the analyst should complete the fields with as much detail as possible since this information will be included in the forensic report. Once the fields have been filled in, click **Next**:



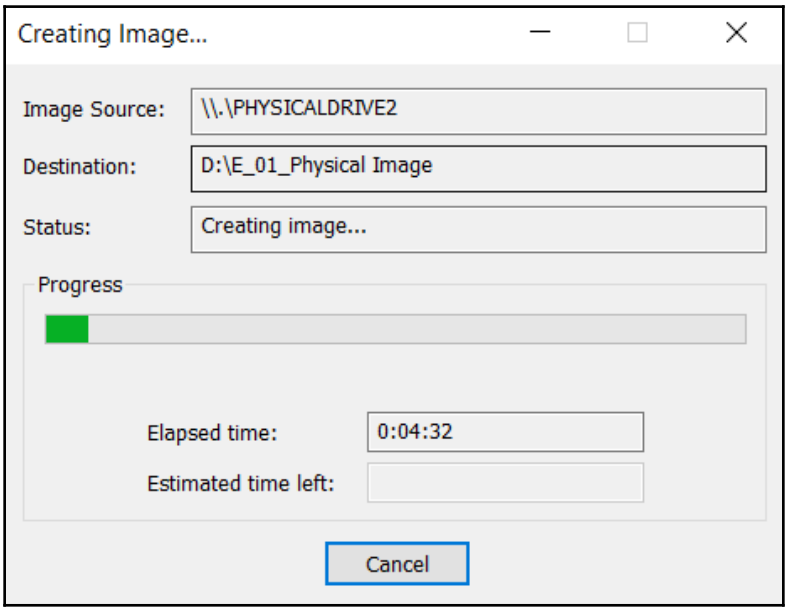
11. In the next window, verify that the image destination and filenames are correct. In addition to this, you'll be able to set the image fragmentation size and compression. The fragmentation size can be set to 0 since this is where the entire disk image will be contained within a single file. For now, the defaults will be utilized since mounting a disk image that is fragmented is not an issue. Once the information you've entered has been verified as correct, click **Finish**:



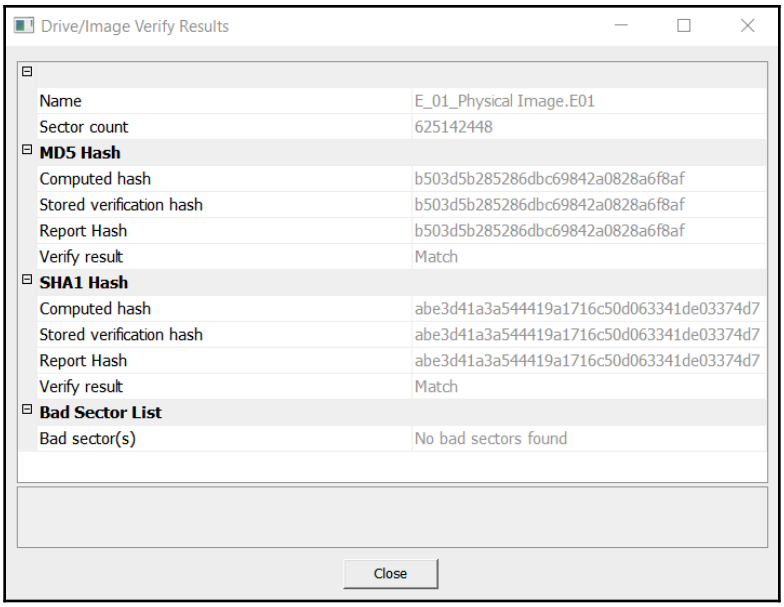
12. Now, the **Create Image** window will open. This is the final stage and is where the analyst can cancel creation of the image file. There are also two options that the analyst should enable, depending on the use case. The first of these involves FTK Imager verifying the image after it's been created. In this feature, FTK Imager will verify that no changes have been made and that the image file is complete without errors. Second, FTK Imager can create a list of all files on the image. This may be handy for the analyst in the event that a specific file(s) has evidentiary value. The analyst will be able to determine whether the file is on this system. This can save time if several drives have to be examined. Once all the settings have been verified, click **Start**:



13. FTK Imager will then begin the process of imaging the drive. This can take several hours or even days, depending on the size of the drive being imaged, the available processing speed of the imaging system, and the type of connection (FireWire, USB, and so on) to the imaging system. While this is happening, the following window will appear:



14. Once FTK Imager has completed the imaging process, a window will open. In this window, FTK Imager will provide the incident response analyst with detailed information. Something that should be of concern to analysts is the hashes that have been computed for both the drive and the image. In this case, both the MD5 and SHA1 hashes match, indicating that the imaging process captured the drive properly and that no changes have been made to the evidence that was taken from the suspect drive. It's good practice to include this information as part of the forensic report:



Navigate to the evidence drive. Here, the entire image can be located. Depending on how FTK has been configured with regard to fragment size, there may be several or a single evidence file. In addition to the evidence files, FTK Imager provides a full list of all the files on the hard drive.

Finally, FTK Imager provides a text file with detailed information concerning the imaging process. This information should be captured and included with any subsequent forensic reporting. At this point, the imaging process has been completed and the evidence drive needs to be returned to security storage.

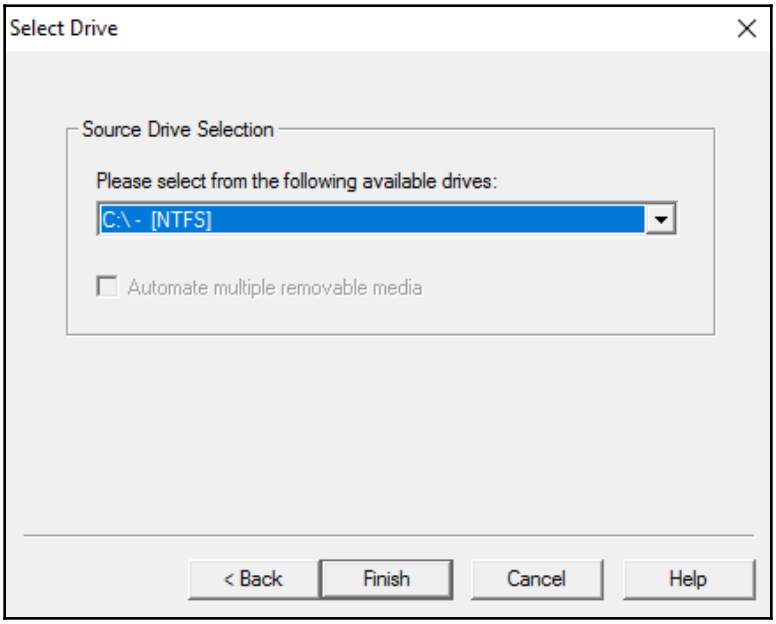
Dead Imaging provides the most forensically sound acquisition, however there may be times when responders will have to image a system that is powered on. This would necessitate the responder to perform a Live Imaging of a suspect system.

Live imaging

A logical image can be captured from a running system utilizing FTK Imager in much the same way. In this case, the one major difference is that FTK Imager will be run from a USB device connected to the system. This allows the incident response analyst to image the drive without changing the system. While certain files and registry settings will be updated, imaging in this fashion will not change system files in the same way that installing FTK Imager would on a potentially compromised system.

In terms of preparation, the analyst should have a preconfigured USB drive with separate tools and evidence partitions. As we discussed previously, the evidence partition should be wiped prior to any use. Also, the full-featured FTK Imager often has issues with DLL files not being in the correct place when attempting to run them from a USB drive. To counter this, Access Data provides a light version called FTK Imager Lite. This can be downloaded at `http://marketing.accessdata.com/ftkimagerlite3.1.1`.

To image a drive, connect the USB to the target machine and simply start FTK Imager Lite. Follow the same process that we outlined previously. The imager will create the image in the same way. As opposed to the previous example, where an entire disk is imaged, live imaging can focus directly on a partition of the drive. For example, in this case, the incident response analyst is only concerned with capturing the `C` drive of the target system. When the **Source Drive Selection** is made, the analyst will need to select `C:\ - [NTFS]:`



The remaining steps are the same for a live image as they were for a dead image in that the analyst will select the destination drive that was previously configured for evidence files. Once the imaging process has been completed, the responder will be provided with the same information that was provided in the previous imaging process.

Remote memory acquisition

The preferred method for the acquisition of memory is through direct contact with the suspect system. This allows incident response analysts to adapt in the event that a tool or technique does not work. This method is also faster at obtaining the necessary files since it doesn't depend on a stable network connection. Although this is the preferred method to use, there may be geographical constraints, especially with larger organizations where the incident response analysts are a plane-ride away from the location containing the evidence.

In the case of remote acquisition, incident response analysts can leverage the same tools that are utilized in local acquisition. The one change is that incident response analysts are required to utilize remote technology to access the suspect systems and perform the capture. In the following sections, two options will be discussed: the first is the open source tool, WinPmem, and the commercial option, F-Response. As with any method that is utilized, incident response analysts should ensure that they document any use of remote technology. This will allow for the proper identification of legitimate as opposed to suspect connections later.

WinPmem

WinPmem can be deployed on remote systems through native applications such as Remote Desktop or PSExec. Once installed on the remote system, the WinPmem output can be piped to another system utilizing NetCat. For example, suppose that the incident response analyst is utilizing a system located at `192.168.0.56`. If the analyst is able to access the compromised host via PSExec or RDS, they can establish a NetCat connection back to their machine by utilizing the following command:

```
C:/winpmem-2.1.exe - | nc 192.168.0.56 4455
```

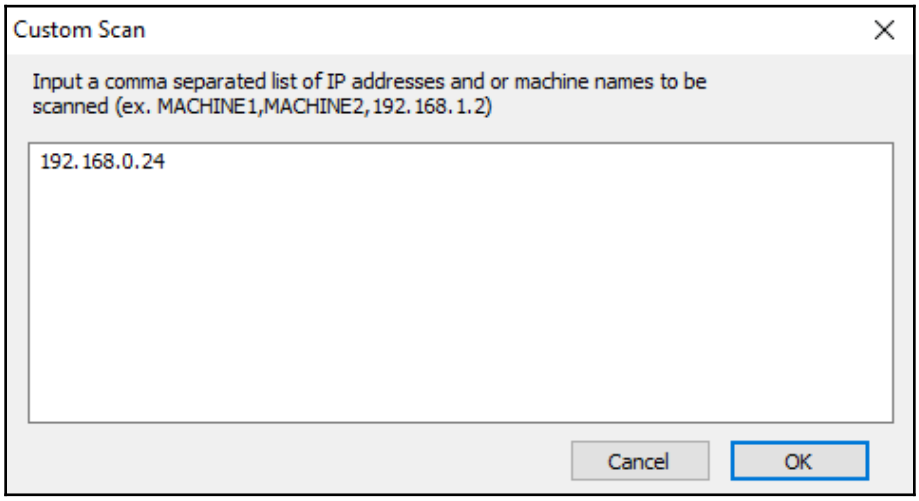
The preceding command tells the system to perform the capture and send the output via NetCat to the incident response analyst workstation over port `4455`. The drawback of this technique is that it requires access to the command prompt, as well as the installation of both NetCat and WinPmem. This may not be the best option if the incident response analyst is dealing with a system that is already suspected of being compromised.

F-Response

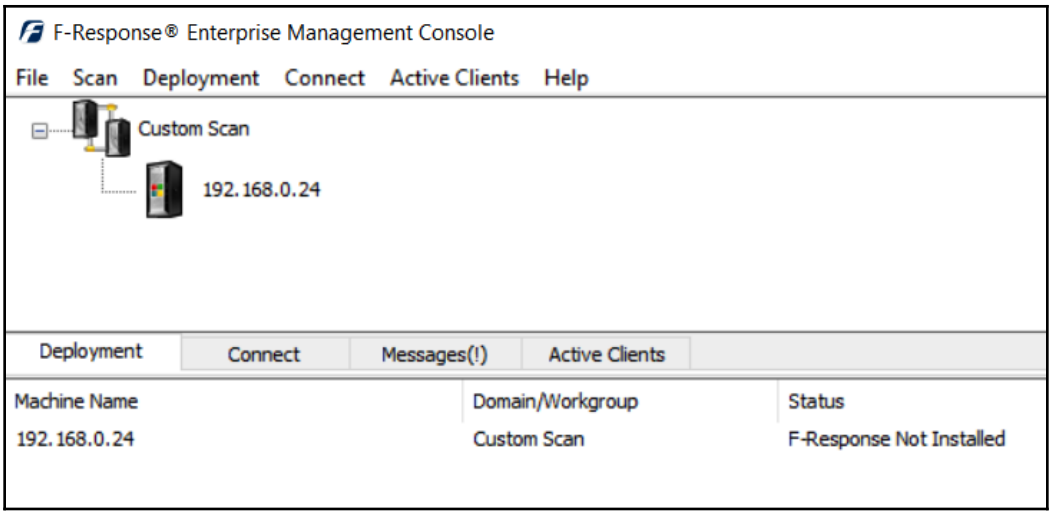
Another option that is available to incident response analysts is the use of the F-Response tool. F-Response is a software platform that allows incident response analysts to perform remote acquisition of evidence over a network. One advantage of utilizing F-Response is that it does not require direct access to the remote system via SSH or RDS. Another key feature of F-Response is that the tool is designed to establish a connection while allowing the incident response analyst to utilize their preferred tools to perform the acquisition. Information on the F-Response product is available at `https://www.f-response.com/buyfresponse/software`.

In the following example, F-Response is being utilized to connect to a system that is suspected of being compromised over a network whereby the incident response analyst can utilize FTK Imager to acquire the memory of the suspect system. Let's take a look at the steps we need to follow in such a situation:

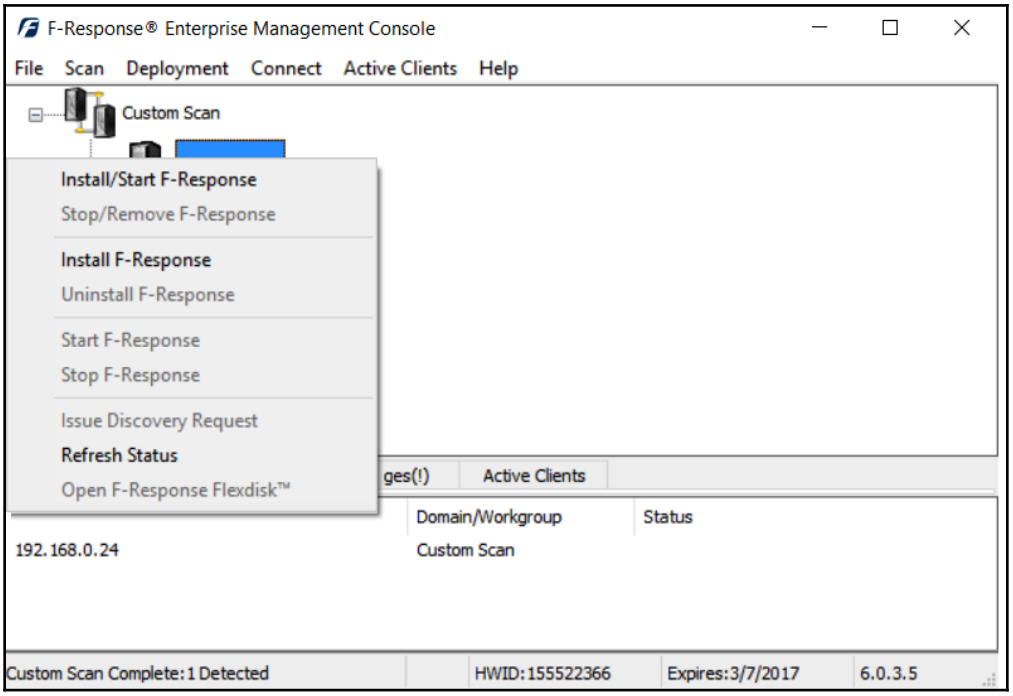
1. After installing F-Response Enterprise, navigate to the **Scan** menu and click on **Custom Scan**. From there, you can enter the suspect system's IP address. In this case, F-Response will be utilized to capture the memory from a system on the local network at the following IP address: `192.168.0.24`:



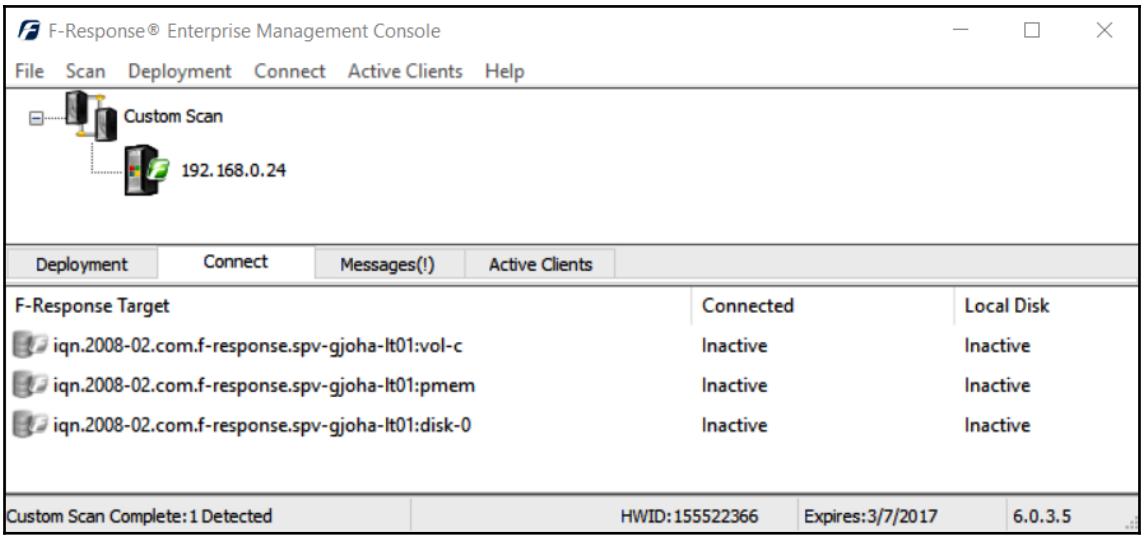
2. After inputting the target IP address, click **OK**. At this point, F-Response attempts to connect to the target system. If F-Response is able to connect to the target system, the system will appear in the upper pane as an icon. In this case, it appears with the Windows logo on the system. In the bottom pane, the target system indicates that it does not have F-Response installed:



3. In order for F-Response to acquire the necessary evidence, an agent has to be installed. You can do this by right-clicking on the system and choosing **Install/Start F-Response**:

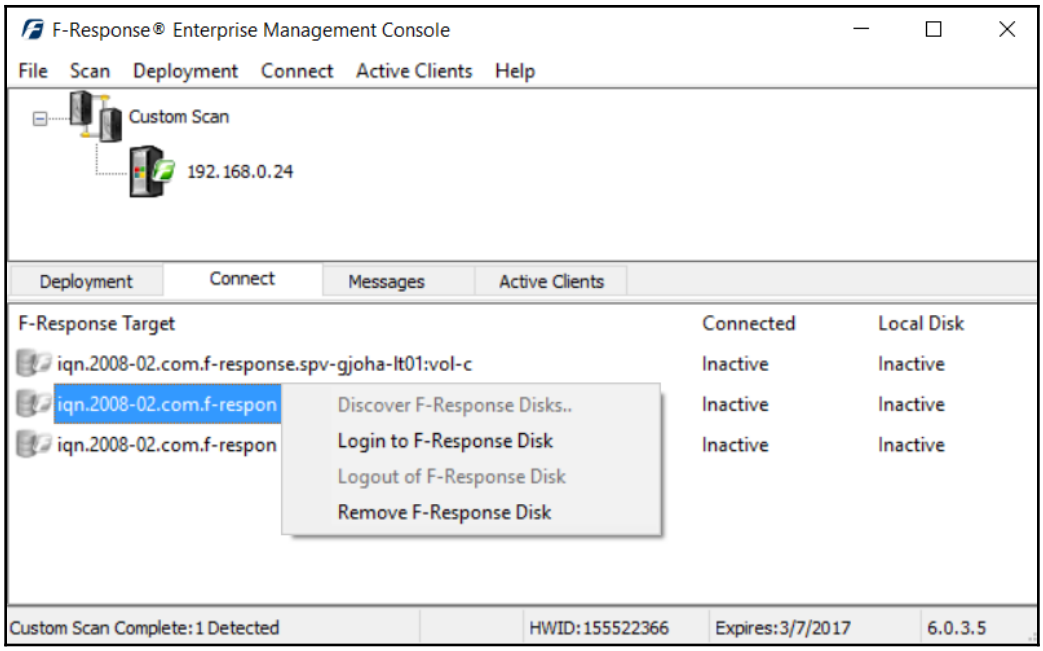


4. Once F-Response has been installed on the remote system, two indicators will be visible. First, a green F-Response icon will appear on the system icon in the top pane. In the bottom pane, a list of the system's available targets will appear:

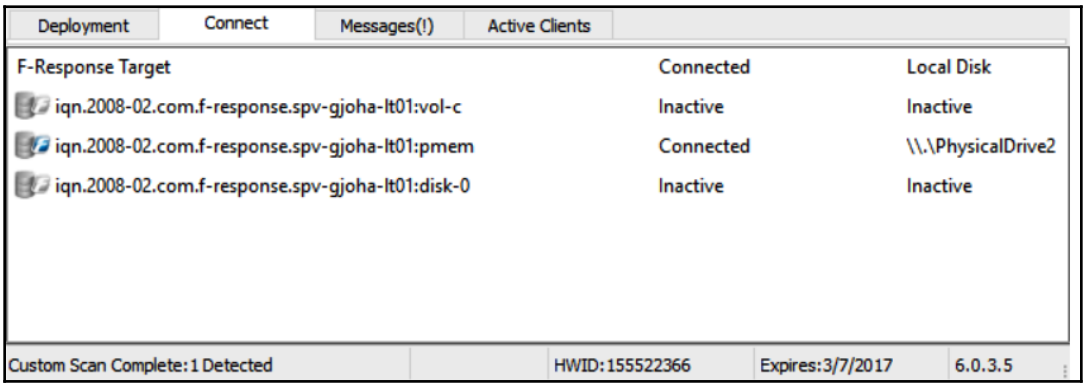


In the preceding screenshot, F-Response indicates that the target system has both a physical drive, indicated by the target ending in `disk-0`, and a logical drive, indicated by the target ending in `vol-c`. In addition to those drives, there is also the memory target, which ends in `pmem`. In this case, the memory is the target.

5. To acquire the memory, F-Response has to be configured so that it can mount the target. Right-click on the target and select **Login to F-Response Disk**:

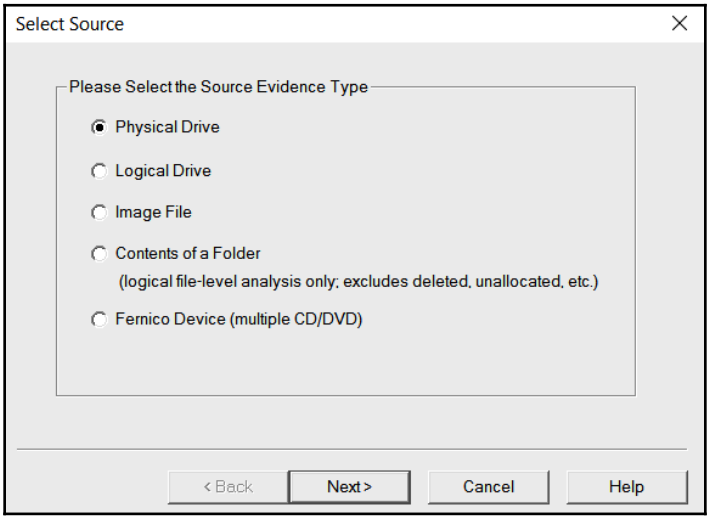


6. Once F-Response has logged in, the bottom pane will indicate which disk is active:

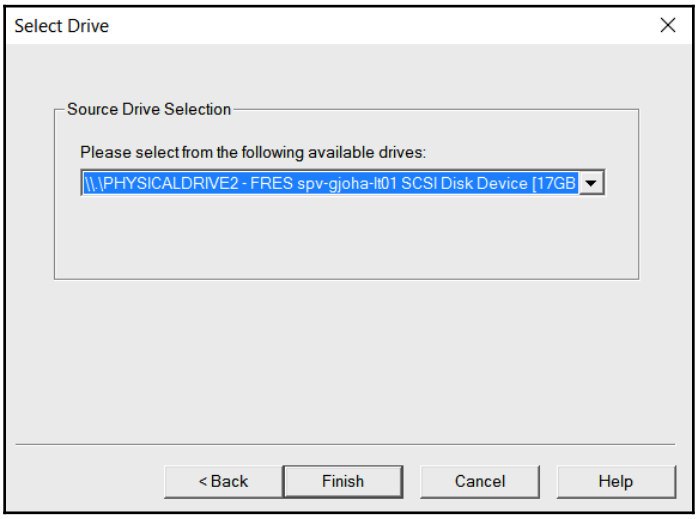


In this case, the memory target has been activated and can be mounted as a physical drive, as indicated by `\\.\PhysicalDrive2` under local disk. From here, the memory can be acquired by utilizing any number of tools. In this case, FTK Imager will be used to acquire the memory as a RAW file.

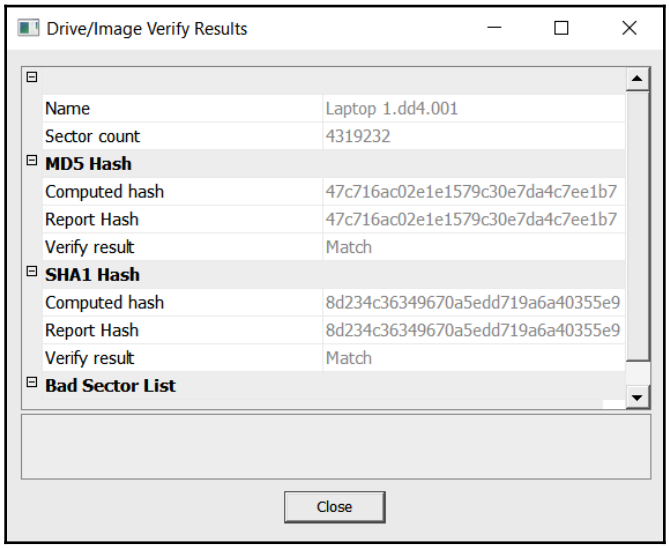
7. Open FTK Imager, navigate to **File**, and select **Create Disk Image**. (Be careful not to select Capture Memory as this will capture the memory on the system running FTK Imager.) A new window will open. Select **Physical Drive** as the source evidence type:



8. Click **Next**. This will open a window where the analyst will select a drive for imaging. In the drop-down menu, locate the drive that F-Response has indicated. In this case, this is `\\.\PHYSICALDRIVE2`:



The remainder of this process is exactly the same as it was for Dead Box forensics. What the F-Response tool does is provide a mechanism whereby FTK Imager can recognize the remote drive as a source for imaging, as can be seen in the preceding screenshot. To FTK, it doesn't make any difference whether the drive is local or remote; it only cares about whether it can be recognized for imaging. As with the other imaging methods, when this process has finished, FTK Imager will provide the following data, all of which should be included in a forensic report:



Next, let's look at Virtual machines.

Virtual machines

Responders will often encounter virtual servers and even workstations as part of an investigation. Virtualized systems can be acquired by simply exporting a paused virtual machine to a removable drive. In other instances, responders can make use of the snapshot feature of a virtual system. This creates a separate file that can be analyzed at the date and time a snapshot is taken. In either case, responders should make sure that the drive has been sanitized properly and that the proper documentation has been addressed.

To acquire the virtual machine, simply pause the system and then move the entire directory to the external media. (In some instances, this can even be accomplished remotely.) In Windows virtual platforms such as VMWare, there are several files that make up the virtual image:

- `.vmdk`: This is the virtual disk image file. This is the logical volume where the virtual operating system and files reside. Obtaining this file is much like imaging the `C` drive on a physical system.
- `.vmem`: The `.vmem` file is the virtual memory file. This is the storage area for the virtual RAM or physical memory. This file can be exported and combined with an additional file for analysis using the methods that will be discussed in `Chapter 8`, *Analyzing System Memory*
- `.vmss`: The VMWare suspended state file saves the running configuration of a suspended virtual machine. This includes process and network connection data. This file is combined with the `.vmem` file to provide the system memory.
- `.vmsn`: This is the virtual snapshot state file. This file contains the state of the system when the snapshots were taken.

Incident responders can use these files in several ways. First, the `.vmdk` file can be mounted the same way as an image file can in various digital forensic software platforms. These will be discussed in `Chapter 9`, *Analyzing System Storage*. Second, the `.vmsn` file can be used to reconstruct the system by simply copying the file and working with the facsimile. From here, responders can look at the behavior of the system or extract evidence without impacting the original `.vmsn` file.

Finally, the running memory that is captured through the `.vmem` and `.vmss` files can be analyzed in much the same way you would analyze other memory captures. To obtain the proper forensic data, the two files must be combined. This can be done by utilizing the `vmss2core.exe` tool, which is included as part of the VMWare suite of tools. To combine these files, the following command syntax needs to be used:

```
C:\VirtualTools\vmss2core.exe –W "InfectedServer.vmss"
"InfectedServer.vmem"
```

The preceding command will produce a memory dump in the directory containing the two files.

Although virtualization is common in large enterprises, it should not represent a significant challenge. In some ways, the ability to simply pause a system and extract all the necessary files makes extracting the necessary evidence faster.

Thus far, the focus has been on Windows tools for imaging. Another option available to incident responders is the use of Linux imaging tools. There are a variety of tools that provide write-blocking and imaging capabilities that are often open source.

Linux imaging

`Chapter 3`, *Fundamentals of Digital Forensics*, provided an overview of various forensic tools that are available to the incident response analyst. Some of these tools include Linux distributions that can be leveraged during an incident for various digital forensic tasks. The following example will demonstrate how a Linux distribution with forensics applications can be deployed to capture a forensically sound image of a potentially compromised computer.

The combination of a Linux distribution and a bootable USB device is an option you can use to conduct forensic imaging of potentially compromised systems. Incident response analysts may find themselves in a situation where multiple systems need to be imaged and the analysts have only one write blocker. A great deal of time will be wasted if the analyst must image each one of these systems in sequence. In this situation, the analyst can avoid this by creating a bootable USB drive for each system and imaging each one at the same time. All the analyst needs is an evidence drive and a bootable USB drive for each source of evidence. Utilizing this technique will allow the analyst to image each system at the same time, saving time that is better spent on other activities.

In this scenario, the **Computer Aided INvestigative Environment Live** (**CAINE**) Linux distribution will be utilized to image the hard drive from a potentially compromised system. First, the system is powered off and the bootable USB device containing the CAINE OS is installed. The suspect system is then powered on. Incident response analysts should be aware of how to change the boot order of a system to ensure that it boots to the USB device. Analysts should also be prepared to immediately power down the system if it attempts to boot into the native OS and not the USB device. Let's get started:

1. Shut down the suspect system and power it back on. Once the device boots up, insert the evidence drive into another available USB interface.

If the evidence drive does not have a bootable OS, it may cause issues as the boot sequence may try to find a valid OS on that drive. That is why it is necessary to wait until the Linux OS that is being utilized boots up.

2. After inserting the evidence drive, open a Terminal and type the following:

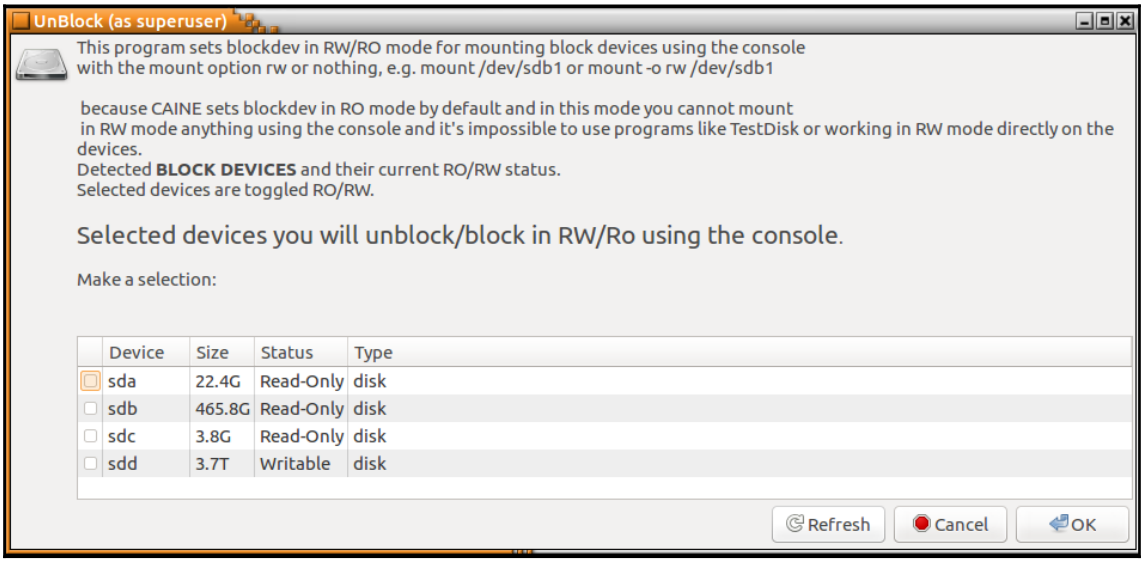
```
caine@caine:~$fdisk -l
```

The `fdisk -l` command lists all partitions that are visible to the CAINE OS. The abridged output will look similar to this:

```
Disk /dev/sdb: 465.8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x345601e6

Device     Boot     Start       End   Sectors   Size Id Type
/dev/sdb1  *         2048   1026047   1024000   500M  7 HPFS/NTFS/exFAT
/dev/sdb2         1026048 975847423 974821376 464.9G  7 HPFS/NTFS/exFAT
/dev/sdb3       975847424 976769023    921600   450M 27 Hidden NTFS WinRE


Disk /dev/sdc: 3.8 GiB, 4060086272 bytes, 7929856 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000f1d04

Device     Boot Start     End Sectors  Size Id Type
/dev/sdc1  *     2048 7929855 7927808  3.8G  c W95 FAT32 (LBA)


Disk /dev/sdd: 3.7 TiB, 4000787029504 bytes, 7814037167 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 33553920 bytes
Disklabel type: gpt
Disk identifier: 30B0BF34-42D8-41E5-A90C-E5735893CFB6

Device      Start        End    Sectors  Size Type
/dev/sdd1      34     262177     262144  128M Microsoft reserved
/dev/sdd2  264192 7814035455 7813771264  3.7T Microsoft basic data
```

In the preceding screenshot, there are three separate disks, each with its own partitions. The disk labeled `/dev/sdc` is the USB drive that contains the CAINE OS that the system has been booted from. The `/dev/sdd` disk is the evidence drive that the system will be imaged to. Finally, the target system is labeled as `/dev/sdb`. It is important to identify the separate disks that appear to ensure that the right target drive is being imaged. By examining `/dev/sdb` more closely, the analyst can see the three separate partitions that make up the entire physical volume. CAINE indicates the boot volume, `/dev/sdb1` in the entries, with an asterisk. This information can be valuable as CAINE can be leveraged to image either the physical volume, as in this demonstration, or specific logical volumes.

3. After identifying the proper target drive of the system, it is critical that the imaging being performed does not change any of the target system's data. The CAINE OS has a built-in software write blocker. On the desktop, you will find the application block on/off. This opens the software write blocker that will be utilized. While examining the list of devices, you will see that the only one that is writable is `sdd`, which we previously identified as the evidence drive. The other drives are set to read-only. This assures the incident response analysts that the imaging process will not alter the target drive (it is a good idea for analysts to take a screenshot of such information for subsequent reporting):



4. After verifying that the evidence drive is in place and that the target system has been set to read-only, the analyst will configure the evidence drive so that it is mounted properly. First, a directory called `EvidenceDrive1` needs to be made an `mnt` directory. Do this by entering the following command:

```
caine@caine:~$ sudo mkdir /mnt/EvidenceDrive1
```

5. Next, mount the `sdd` disk on that newly created mount directory by entering the following command:

```
caine@caine:~$sudomount /dev/sdd2 /mnt/EvidenceDrive1/
```

Now, the evidence drive has been mounted on the mount point that was created.

6. Next, change the directory to the evidence drive by using the following command:

```
caine@caine:~$sudomount /dev/sdd2 /mnt/EvidenceDrive1/
```

7. The next step is to make a directory that will contain the image file. First, change to the `EvidenceDrive1` directory by entering the following command:

```
caine@caine:~$ cd /mnt/EvidenceDrive1/
```

8. Next, make the directory. In this case, the directory will contain the case number, `Case2017-01`, as the directory. It is a good idea to make this directory tie indirectly with the incident in some fashion. The following command will create the proper directory:

```
caine@caine :/ mnt /EvidenceDrive1$ mkdir Case2017-01
```

9. The final step is to navigate to the new directory by entering the following command:

```
caine@caine :/ mnt /EvidenceDrive1$ cd Case2017-01/
```

10. Now that you're in the proper directory, all you need to do is image the suspect drive. There are several tools available for doing this. In this example, the Dc3dd tool will be used. This tool was developed by the Department of Defense Cyber Crime Center forensic specialist, Jesse Kornblum. This application has additional features that aren't found in the `dd` Linux imaging application. These include error reporting and multiple hashing algorithms that can be leveraged on the fly. To start the imaging process, the following commands need to be entered:

```
caine@caine:/mnt/EvidenceDrive1/Case2017-01$ dc3dd
if=dev/sdbof=ideapad.img hash=md5 log=dc3ddlog.txt
```

The preceding command contains `dc3dd`. Start by imaging the disk at `sdb` to the evidence drive under the `ideapad.img` file name, and by hashing the output with MD5. Finally, the application will create a log file called `dcddlog.txt` that can be utilized for reporting purposes. This produces the following output:

```
caine@caine:/mnt/EvidenceDrive1/Case2017-01$ sudo dc3dd if=/dev/sdb of=ideapad.img hash=md5 log
=dc3ddlog.txt

dc3dd 7.2.641 started at 2017-04-02 19:18:35 +0100
compiled options:
command line: dc3dd if=/dev/sdb of=ideapad.img hash=md5 log=dc3ddlog.txt
device size: 976773168 sectors (probed),   500,107,862,016 bytes
sector size: 512 bytes (probed)
 6376849408 bytes ( 5.9 G ) copied (  1% ),   58 s, 105 M/s
```

11. Depending on the size of the drive, this process can take hours. During this time, the analyst can keep track of any progress that is made. Upon completion, the application will produce some output indicating how many sectors were utilized for input and how many sectors were used as output to the image file. Ideally, these should be the same. Finally, an MD5 hash of the image file is calculated and utilized as part of the output:

```
dc3dd 7.2.641 started at 2017-04-02 19:18:35 +0100
compiled options:
command line: dc3dd if=/dev/sdb of=ideapad.img hash=md5 log=dc3ddlog.txt
device size: 976773168 sectors (probed),   500,107,862,016 bytes
sector size: 512 bytes (probed)
500107862016 bytes ( 466 G ) copied ( 100% ), 5854 s, 81 M/s

input results for device `/dev/sdb':
   976773168 sectors in
   0 bad sectors replaced by zeros
   d48a7ccafaead6fab7d284b4be300bd8 (md5)

output results for file `ideapad.img':
   976773168 sectors out

dc3dd completed at 2017-04-02 20:56:09 +0100
```

12. Examining the evidence drive from a Windows system reveals the image and log file that were created with the application:

13. Examining the log file reveals the following information, all of which should be incorporated into any subsequent reporting:

```
dc3dd 7.2.641 started at 2017-04-02 19:18:35 +0100
compiled options:
command line: dc3dd if=/dev/sdb of=ideapad.img hash=md5
log=dc3ddlog.txt
device size: 976773168 sectors (probed), 500,107,862,016 bytes
sector size: 512 bytes (probed)
500107862016 bytes ( 466 G ) copied ( 100% ), 5854.43 s, 81 M/s
input results for device `/dev/sdb': 976773168 sectors in
0 bad sectors replaced by zeros d48a7ccafaead6fab7d284b4be300bd8
(md5)
output results for file `ideapad.img': 976773168 sectors out
dc3dd completed at 2017-04-02 20:56:09 +0100
```

Linux is a viable option when it comes to acquiring disk evidence. One significant advantage it has is that it is easy to scale. In the event that multiple systems have to be acquired, responders can use several USB storage drives and Linux USB devices and acquire them in parallel, rather than waiting for software to become available. CAINE is an excellent option for this as the included write blocker also affords a measure of evidence integrity in the process.

Imaging is a critical process for responders to understand. The incident will often dictate which technique should be used. In any incident though, responders should ensure that the process is conducted in a sound manner as subsequent investigation will often rely on data acquired from these systems.

Summary

Not every incident may dictate the need to obtain an image from a potentially compromised hard drive or other volume. Regardless, incident response analysts should be familiar with, and able to perform, this function when called upon. The evidence that's found on a hard drive may be critical to determining a sequence of events or to obtaining actual files that can aid in determining the root cause of an incident. This is the central reason why responders need to understand the fundamentals of imaging, the tools and processes involved, how to create a stage drive, using write blockers, and executing any of the imaging techniques we mentioned in this chapter. As with any process that's performed in a forensic discipline, imaging should be conducted in a systematic manner in which all the steps are followed and properly documented. This will ensure that any evidence that's obtained will be sound and admissible in a courtroom.

In the next chapter, we will discuss examining network-based evidence in relation to the network activity which is associated with an incident.

Questions

1. What are the two types of write blockers?

   A) Hardware
   B) Digital
   C) Software
   D) Court approved

2. Responders should ensure that any storage drive that's used for imaging is properly sanitized before each use.

   A) True
   B) False

3. What type of imaging is used to acquire the entire physical volume of a drive?

   A) Dead imaging
   B) Live imaging
   C) Remote imaging
   D) Hardware imaging

4. What imaging application is found only on Linux systems?

   A) FTK Imager
   B) EnCase Imager
   C) AFF4
   D) dd

Further reading

- *FTK Imager Guide*: `https://ad-pdf.s3.amazonaws.com/Imager/3_4_3/FTKImager_UG.pdf`
- NIST *Computer Forensic Tools & Techniques Catalog*: `https://toolcatalog.nist.gov/search/index.php?ff_id=1`
- *An Overview of Disk Imaging Tool in Computer Forensics*: `https://www.sans.org/reading-room/whitepapers/incident/overview-disk-imaging-tool-computer-forensics-643`

3
Section 3: Analyzing Evidence

Having completed the acquisition of digital evidence in section two, section three focuses on proper analysis techniques in digital forensics. This section will focus on the appropriate tools and techniques to determine the root cause of an incident.

This section comprises the following chapters:

- `Chapter 7`, *Analyzing Network Evidence*
- `Chapter 8`, *Analyzing System Memory*
- `Chapter 9`, *Analyzing System Storage*
- `Chapter 10`, *Analyzing Log Files*
- `Chapter 11`, *Writing the Incident Report*

7
Analyzing Network Evidence

`Chapter 4`, *Collecting Network Evidence*, explored how incident responders and security analysts are able to acquire network-based evidence for later evaluation. That chapter focused on two primary sources of that evidence, network log files and network packet captures. This chapter will show which tools and techniques are available to examine the evidence acquired. Incorporating these techniques into an incident response investigation can provide incident response analysts with insight into the network activity of possible threats. In this chapter, the following main topics will be addressed:

- **Network evidence overview**: Adversaries are bound to the same network protocols that govern normal network traffic. Adversarial techniques that can be identified with the proper analysis of network data are addressed.
- **Analyzing firewall and proxy logs**: Adversaries need to make initial and continued connections to their infrastructure. Network devices such as firewalls and proxies may provide a source of evidence from log files.
- **NetFlow**: NetFlow describes the data about connections between devices in the network. Used primarily to troubleshoot connectivity and bandwidth issues, NetFlow can be used by responders to gain insight into the movement of data in relation to an incident.
- **Packet captures**: One of the best sources of evidence during an incident is packet captures. Dissecting them can uncover data exfiltration, exploits, and command and control.

Network evidence overview

In `Chapter 4`, *Collecting Network Evidence*, there was a focus on the various sources of evidence that network devices produce. Most of this evidence is contained within the variety of log files produced by switches, routers, and firewalls. Depending on the type of environment that responders find themselves in, this evidence source can be augmented with NetFlow data and full packet captures.

Once the various sources are understood, it is important to then focus on what logs, NetFlow, and packet captures can tell us about an incident. The following are several areas of focus where proper logging and evidence collection may provide additional context surrounding an incident, as well as potential data points when deriving root cause:

- **Reconnaissance and scanning behavior**: There are a plethora of tools available to adversaries to automate scanning of perimeter devices such as firewalls and routers. These scanners attempt to ascertain open ports, vulnerabilities, or authentication protocols such as **Secure Shell** (**SSH**) that can be exploited. These scans do in fact leave a trace as they will often require connections to the devices. Depending on the level of logging and the retention period, responders may be able to identify the external infrastructure that is attempting to compromise the perimeter systems.
- **Initial infection**: Adversaries have become very sophisticated in compromising systems. They will often make use of multi-stage exploits and malware. The first stage will call out to an external infrastructure through a URL and download additional exploits. Web proxies and firewalls may have connection data contained within the log files that record this activity.
- **Lateral movement**: Once inside a network, adversaries will often attempt to conduct reconnaissance, exploit other systems, and move data around. NetFlow logs provide insight into this type of behavior.
- **Command and control**: Once a foothold is established in the network, adversaries require the ability to maintain control over compromised systems. Logs, packet captures, and NetFlow data may be leveraged to identify this type of behavior.
- **Data exfiltration**: One of the goals of an adversary may be the compromise and exfiltration of data. Proxy logs may identify the destination of such data. NetFlow may show the flow of data from the internal systems to any external systems. Finally, packet captures may be leveraged to identify the exfiltrated files, the source of the data, and the destination.

In `Chapter 4`, *Collecting Network Evidence*, there was a discussion on the three main types of network evidence that can be leveraged in an incident. It is often hard for responders that do not have knowledge about network traffic to understand the various aspects. Think about network traffic as a letter that is sent from one individual to another. Log data records the sender and receiver's address and mailbox number at a central location, such as the local post office. This is akin to the source and destination IP address and ports. NetFlow records much of the same information about the letter but can also tell the individual the weight or relative size of the letter, along with the sender and receiver's address and mailbox number. Finally, a packet capture tells us all the same information obtained through logs and NetFlow, but will also tell the individual the contents of the letter, including (as long as it is not encrypted) the actual data contained.

Identifying a root cause with network evidence is largely dependent on the evidence itself. One major drawback to evidence such as packet captures and log files is the sheer volume of data that normal network operations create. Often, an incident is identified days or even weeks after it is has occurred. During the intervening period, these log files and packet captures have become unavailable. It is therefore incumbent on responders to understand fully what their organization's capabilities are in regard to network evidence.

Analyzing firewall and proxy logs

`Chapter 4`, *Collecting Network Evidence*, contained a good deal of information concerning the acquisition of network-based evidence and the types of log files that are of importance to an incident responder or security analyst. Aside from the previously covered packet capture, there was a good deal focused on the acquisition of log files from a variety of sources. These log files can provide some insight into the potential indicators of compromise that can aid in an incident investigation. The main challenge for analysts, though, is sifting through all of the irrelevant logs to find those that have some evidential value.

Log file analysis can be performed in a variety of ways. The specific method that is used may often depend on the type of incident, the tools available, and the amount of log data that has to be analyzed. The following are some of the methods that can be utilized:

- **Manual log review**: In a manual log review, raw log files are dumped into a tool such as a text editor. From there, the analyst will review the logs line by line. This is a low-cost solution, but it is only useful with a limited amount of data. For example, an analyst would not be able to perform this type of analysis on a large enterprise firewall connection log. Rather, it may be useful to determine which users logged into a seldom-used web application on a particular day.

- **Filtered log review**: Log review tools allow analysts to filter out log files along specific parameters. This can include showing a list of any known malicious activity. The one drawback is that logs may not immediately indicate known malicious activity, but rather are innocuous at the onset.
- **Log file searching**: Another key feature in most log analysis tools is the ability to search log files for specific expressions. Tools for searching can utilize both regex and Boolean expressions and allow the analyst to limit logs to a specific time period, source IP address, or other specific condition. This allows analysts to quickly isolate specific log files. Depending on the search terms, this may return a good deal of information that has to then be reviewed manually.
- **Log file correlation**: Separate log activity can be correlated with other logs based upon either preconfigured rules or algorithms. Log correlation is often made part of log management tools or **Security Information and Event Management** (**SIEM**) platforms with rulesets that have been created. This method is very powerful, as it automates the process, but it does require a good deal of upfront labor to configure and tune to the specific environment.
- **Log file data mining**: The next step up from correlation is the ability to mine log files and extract meaning from these. This gives greater context and insight into the specific activity. Currently, there are several tools, such as Elasticsearch and Logstash, which have been integrated into a platform for more useful information.

The quantity of logs that are produced in a network over a month or so can be staggering. This quantity only increases with the addition of new sources. Sorting through these manually is near impossible. In terms of log review, it is better to have a solution that provides some measure of automation, even in small networks. These tools give analysts the ability to sort through the proverbial stack of hay for that critical needle.

DNS blacklists

One technique that performs a combination of filtering and manual log review is utilizing scripting languages such as Python. These scripts can parse through firewall logs or other inputs to highlight specific areas of focus for the analyst. One such script is DNS blacklists, which is available at `https://bitbucket.org/ethanr/dns-blacklists/`. This script takes a text file created by the log source or analyst and compares it to lists of IP addresses and domains that have been blacklisted.

The folder containing the script contains two other folders that are compared against each other. One folder contains the text files of IP and domain blacklists. These blacklists can be obtained from open sources or threat intelligence providers. (`Chapter 13`, *Leveraging Threat Intelligence* will address how threat intelligence sources can be leveraged for incident response.) The script runs the suspicious log files or IP addresses against the blacklists to determine whether there are any matches.

In the following example, a list of known Emotet URLs and IP addresses are going to be compared to a raw firewall log that has been obtained. Once the data is placed into the appropriate folders, the following command is entered into the Terminal:

```
dfir@ubuntu:~/python dns_blacklists.py bad_lists/ traffic_directory/
```

This command runs the script with the Emotet blacklists contained in the `Bad Lists` folder against the log files or IP addresses in the `Traffic Directory` folder. The command produces the following output:

```
Note: DNS resolution and reverse resolution is currently not supported.
Parsing blacklist files...
--------------------------------------------------------------------------------
EmotetIOC_01_17_19.txt
EmotetIOC_04_2019.txt
EmotetIOC_08_2019.txt

Parsing check files...
--------------------------------------------------------------------------------
Firewall Logs.txt

================================================================================
The following hostnames were found in the blacklists:
================================================================================
rozhan-hse.com
================================================================================
The following IPs were found in the blacklists:
================================================================================
```

The output indicates that the `rozhan-hse.com` URL was found on one of the Emotet IOC blacklists. `DNS_Blacklists` is a good tool to perform an initial triage of log files. The efficacy of the results, though, is largely dependent on what data is placed within the `Blacklist` folder. The more up to date and accurate those are, the better the results will be. Positive results should be followed up via additional searching.

SIEM tools

In `Chapter 4`, *Collecting Network Evidence*, there was also discussion of the use of SIEM platforms. These platforms not only serve as an aggregation point for log files from network devices, they also allow analysts to perform queries on the logs that have been aggregated. For example, there were IP addresses associated with potential malicious activity discovered during the analysis of the packet capture file. This file was limited to a single host on the internal network. One question that analysts would like to answer is, how many other hosts could possibly be infected? If the SIEM aggregates connection log files from devices such as the exterior facing firewall and web proxy, the analyst would be able to determine if any other internal hosts connected to those suspect IP addresses.

There are a wide variety of SIEM platforms available, from freeware solutions to enterprise security management platforms. Most of these platforms allow analysts to conduct filtered, searching, and correlation log reviews. Many of the more robust commercial platforms provide rulesets for detecting specific types of attacks and updates to these rulesets as new attacks become known. Analysts could also query the SIEM for connection logs for the host IP address to any other systems. This would normally be the behavior seen in an incident where malware has infected a machine and an attacker is attempting to compromise other machines.

In organizations where incident response personnel are separate from those that have responsibility for the maintenance of the SIEM, it is a good idea to review the communications structure so that incident response analysts have access to these platforms. The wealth of information and data that is available can be leveraged to determine what activity on the internal network is connected to a possible incident, as well as evidence that can be utilized to determine the root cause.

The Elastic Stack

Alongside SIEM technology, incident response analysts can also leverage a bundle of applications for log analysis. This bundle, referred to as the Elastic Stack, combines three tools together that allow for the analysis of large sets of data. The first of these is Elasticsearch. Elasticsearch is a log-searching tool that allows for near real-time searching of log data. This is accomplished through full-text searching, powered by Lucene. This allows analysts to perform queries against log files for such elements as user IDs, IP addresses, or log entry numbers. Another key feature of Elasticsearch is the ability for the platform to expand the solution as the enterprise grows larger and gains more data sources. This is useful for organizations that may want to test this capability and then add data sources and log files incrementally.

The next component in the Elastic Stack is Logstash. Logstash is the mechanism that handles the intake of log files from the sources across the network, processes log entries, and finally, allows for their output through a visualization platform. Logstash can be configured and deployed easily. The integration of Logstash with Elasticsearch provides the incident response analyst the ability to conduct fast queries against a large amount of log data.

The final component of the Elastic Stack is Kibana. Kibana serves as the visual interface or dashboard of the Elastic Stack. This platform allows analysts to gain insight into the data through the use of dashboards. Kibana also allows analysts to drill down into specific key data points for detailed analysis. Incident response analysts can customize the dashboards so that the most critical information, such as intrusion detection logs or connection logs, are immediately available for review.

For example, the Kibana dashboard utilizes a number of pie charts to display log activity. Utilizing these allows for an overview of what information is available to an analyst.

The Elastic Stack has become a powerful tool for security professionals and incident responders. It is recommended that analysts and incident response professionals consult more resources to become familiar with this technology, as they will most assuredly see it again.

Analyzing NetFlow

NetFlow is a feature that was first introduced by Cisco Systems in the 1990s. NetFlow collects specific data about packets as they enter or exit an interface of a router or switch. This data is then sent to a NetFlow Collector via a NetFlow exporter, which is often made part of switches or routers. The NetFlow Collector then aggregates and stores the flow data for analysis. This data is often leveraged by network and systems administrators to troubleshoot bandwidth issues, identify network congestion, and to observe the flow of data.

A sample NetFlow output is included next. What is included with flow data can vary from network device manufacturer as there are several versions in the commercial market. The following screenshot shows some of the basic information that is captured as part of a NetFlow dataset:

Src Addr	Dst Addr	Sport	Dport	Proto	Packets	Bytes	Flows
192.168.1.7	192.168.2.56	5734	22	tcp	42	3028	1
192.168.1.5	192.168.2.45	3687	22	tcp	52	2564	1
192.168.1.7	192.168.2.55	4675	22	tcp	1	1240	1
192.168.1.6	192.168.2.34	6897	22	tcp	46	4056	1
192.168.1.6	192.168.2.56	3657	445	tcp	325	56798	1

The following components of a NetFlow record are found in the preceding screenshot:

- `Src Addr`: This is the source address that has initiated the connection or is sending traffic.
- `Dst Addr`: The destination address for the connection.
- `Sport`: This is the source port for the source address.
- `Dpost`: This is the destination port. In terms of analyzing NetFlow as part of an incident investigation, this is one of the key data points to focus in on as this often tells responders the service the source address is connecting to.
- `Proto`: This is the protocol in use.
- `Packets`: The number of packets that are made as part of the flow.
- `Bytes`: The total number of bytes.
- `Flows`: Indicates how many flows have been recorded.

When examining the NetFlow data of the preceding example, there are two significant data points that may be important. The first are the number of SSH connections between devices. Secure Shell is a common way for systems to communicate with each other, but if this is outside the bounds of normal network behavior, it warrants a follow-up. In addition, connections via SMB (port `445`) are commonly abused by adversaries to access other systems, deliver ransomware, or to access file shares. Even in this short example, it becomes very clear that responders gain a great deal of insight by just having visibility of the connections that occur on the internal network.

There are a wide variety of commercial tools that are in use to view NetFlow. The use of NetFlow data is also largely dependent on the organization. Configuring NetFlow is not something that can readily be accomplished during an incident without access to significant resources from both commercial providers and internal operations personnel. Regardless, responders that do have access to NetFlow would be well served to acquaint themselves with the technology, as it does provide a significant insight as to how data moves through the network.

Analyzing packet captures

A great deal of `Chapter 4`, *Collecting Network Evidence* covered the various methods to obtain packet captures from a range of sources and from a variety of locations. Packet captures contain a great deal of information that is potentially valuable to incident response analysts. Some of this information includes source and destination IP addresses, domains and ports, and the content of communications between hosts. In some instances, incident response analysts are able to reconstruct actual files, such as text documents and images, in these packet captures.

This chapter makes reference to several preconfigured packet captures that are examined. These packet captures are taken directly from `http://malware-traffic-analysis.net/` by permission of the author. This site has a number of packet capture exercises, where incident response analysts can practice locating indicators of compromise. It should be noted, though, that these captures may contain malware. Readers should only examine the live packet captures in a properly configured sandbox (see `Chapter 12`, *Malware Analysis for Incident Response*) or other system not connected to a production environment.

Command-line tools

There are several command-line tools that can be utilized during the analysis of network packet captures. During more in-depth or lengthy incident response engagements, analysts may gather several packet captures files. It may be beneficial to combine these multiple packet captures into one single file to make analysis easier. The application Mergecap does just that by combining several packet capture files. Mergecap is offered as part of the CAINE OS and can be executed utilizing the following command:

```
caine@caine:~$ mergecap -w mergedpacketcapture.pcap packetcapture1.pcap
packetcapture2.pcap
```

Another command-line tool that is useful in analyzing packet captures is the tool Editcap. Editcap allows analysts to manipulate the packet capture files into smaller segments for easier review. For example, an analyst may only want to look at captures that are broken up into 50,000 packet segments. This would be helpful if an analyst has a large packet capture and dividing would make searching easier. To do this, the analyst would type the following into the command line:

```
caine@caine:~$ editcap -F pcap -c evidence.pcap split.pcap
```

In the preceding command, Editcap took the `evidence.pcap` evidence file and divided it out into 50,000 packet segments. Another technique that Editcap can be leveraged for is to divide a larger packet capture into time segments. For example, if analysts want to divide a packet capture into 10-minute segments, they type in the following:

```
caine@caine:~$ editcap -F pcap-t+600 evidence.pcap split.pcap
```

Analysts may also find that, in some circumstances, they may want to isolate domain name registration traffic. This is due in large part to a variety of adversarial actions such as C2 traffic, data exfiltration, and the possible redirection to compromised websites, often leveraging vulnerabilities in the DNS system. The application Dnstop parses packet capture files and ascertains the sources and count of DNS queries from internal hosts. To install on a Linux system, the following command is used:

```
dfir@ubuntu:~$ sudo apt-get install dnstop
```

This command will download and install Dnstop. In the following example, the following packet capture was taken from the Malware Traffic Analysis site located at `https://www.malware-traffic-analysis.net/2019/03/13/index.html`. If an incident response analyst wants to determine whether any IP addresses were sending outbound DNS queries for packet capture, they simply execute the following command:

```
dfir@ubuntu:~/Documents/Packet Captures$ dnstop 2019-03-13-Emotet-with-
Trickbot.pcap
```

The output of the preceding command is as follows:

```
Queries: 10 new, 10 total, EOF

Sources          Count      %   cum%
---------- --------- ------ ------
10.3.13.101         10  100.0  100.0
```

The output indicates that only one host in the packet capture is the source of DNS queries, having made a total of 10. While this was a simple example, incident response analysts can utilize the preceding technique of combining multiple packet capture files and then utilizing DNStop in order to gain a better sense of what DNS traffic is leaving the internal network, and if that is something that warrants further investigation.

Moloch

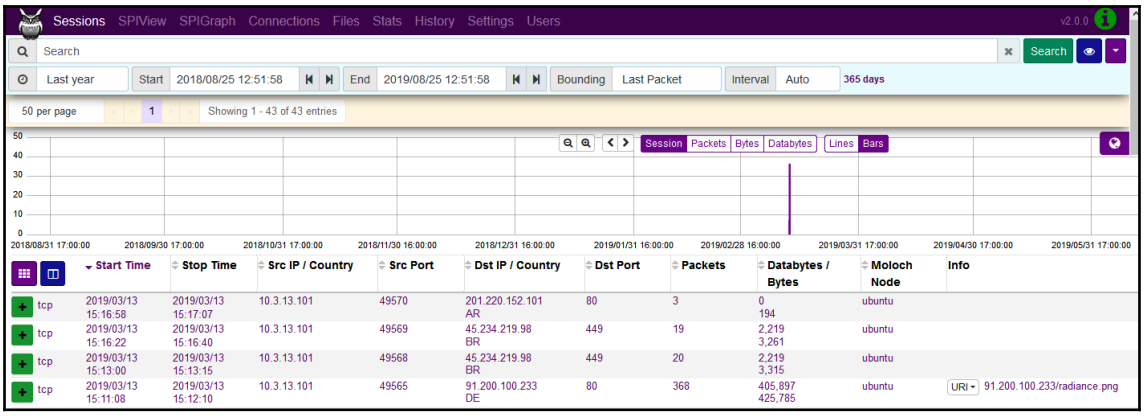
Moloch is an open source packet capture and search system that allows analysts and responders to examine large network packet captures. By default, Moloch organizes the packet captures into the various sessions contained within the capture. Moloch can be utilized as a network monitoring system that can be leveraged through importing packets into the Elasticsearch infrastructure. From here, responders can examine network activity in near real time. Another method that Moloch can be leveraged through is loading offline packet captures for indexing.

Installation instructions for Moloch can be found at `https://molo.ch/#download`. Moloch can be installed on a variety of Linux desktop or server platforms. The server option provides larger teams with the ability to share data concerning packet captures as well as evaluate running captures. Desktop installations are an option for responders that will be handling offline data and who do not need to share the results.

For the purposes of this chapter, Moloch will be used to examine an offline packet capture obtained from Malware Traffic Analysis at `https://www.malware-traffic-analysis.net/2019/07/22/index.html`. The packet capture needs to be transferred to the Moloch system first. This can be done via any Secure File Transfer Protocol client directly to the Moloch directory: `/data/moloch/raw`. From here, execute the following command to have Moloch ingest the packet capture:

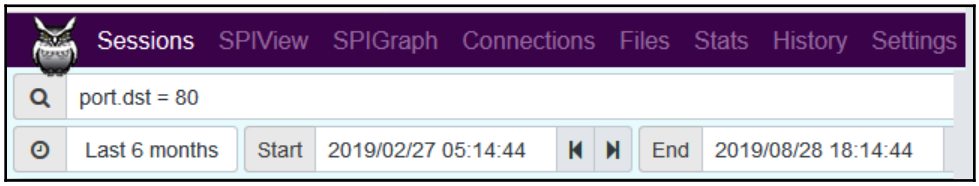
```
dfir@ubuntu:~/data/moloch/bi/moloch-capture -r /data/moloch/raw/
2019-07-22-Amadey-infection-with-Pony-and-Ursnif-and-Cobalt-Strike.pcap
```

This will take the offline packet capture and process it. Once completed, open a web browser and navigate to the IP address of the server or workstation with the port `8005`. This will open the Moloch interface. Once there, the following view will appear:



Moloch is a feature-rich platform. The following steps provide an overview of some of the features available in examining offline packet captures:

1. An examination of the packet capture from the dashboard identifies several different sessions where the internal system at `10.3.13.101` is communicating with external IP addresses. To narrow down the search results to internet traffic over HTTP, the following search query should be entered into the search bar:



2. A good way to determine the presence of files within a packet capture is to identify the number of packets per session. In this case, click on the down arrow next to the column header, `Packets`. This will sort the number of packets from largest to smallest:

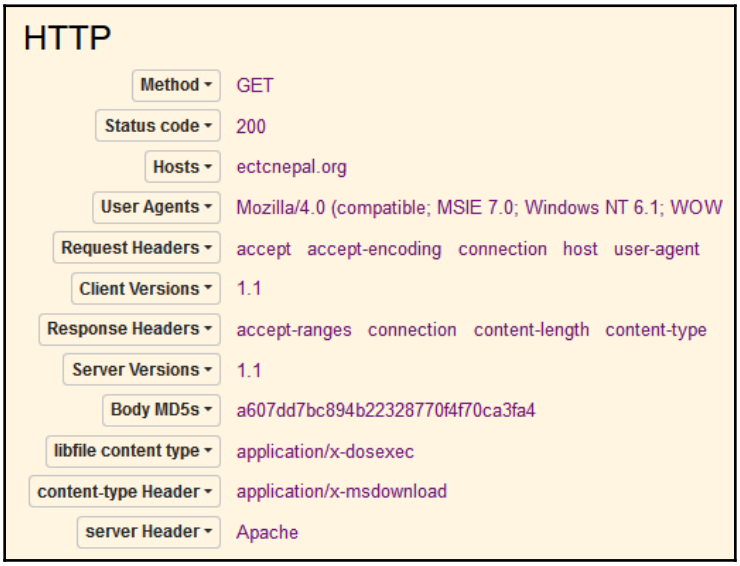
Dst IP / Country	Dst Port	Packets	Databytes / Bytes
83.220.141.232 DE	80	524	506,630 534,942
129.226.63.136 SG	80	291	246,151 261,881
31.44.184.33 RU	80	230	211,253 223,689
129.226.63.136 SG	80	221	195,452 207,402
31.44.184.33 RU	80	218	211,265 223,053

3. The far right of the dashboard contains URLs and associated information concerning the sessions. An analysis of the queries thus far has indicated that several executable files, including an executable named `a22`, appear to have been accessed by the system under analysis:

URI neu.x-sait.de/wp-content/plugins/mce-table-buttons/pp.exe
neu.x-sait.de/wp-content/plugins/mce-table-buttons/4.exe
URI
cd.pranahat.at/webstore/zSR1Z6AnDl0PfpsiGN_2F9W/7oYFSY47cH/9hfkJMvimgZEx3TlC/hR5_2BF3YKL1/WqZUwlQoia
O/9WGtTrqojm1Fpc/w4JGS_2BqOVVP9F5qP_2B/QcnCwcoZRFssnyQ8/6pxNROtgxS0JSSG/j_2B9TBJ5hwNSZU9f7/1ud
mNsp5a/TjPoAB3gEfW2yvZBcgGC/o1Z1ifT9HbS1Dg4Stts/AFZuGjnzhJzwDKC84R_2Fu/ZcUMga13Z/eNW1ffKY/VxS
URI 31.44.184.33/uDaB
URI
x1.narutik.at/webstore/Wlpdq7f64iZDVKdP/nNlPLUwRF2LoqfY/E5R_2FJjib_2Ba2k97/bclDnPA_2/BkBxN61i7f8UEj0rGwAa
/dPKTXlMVsfT1EHFq3EY/G9gCfV7T5wEjN4HML4X7pG/NOM985YvHHdzP/5Lcb8zEq/NMGJwwOWXxbU6a6_2FuMdbR/V
av0v2m3j2/4GXc_2FXtmWGU6mJ_/2Fn_2BLwcjuW/sYrlBoHcbMb/Xpu043fnlxCp/Uay
URI 31.44.184.33/H7mp
URI cd.pranahat.at/jvassets/o1/s64.dat
URI 31.44.184.33/visit.js
URI ectcnepal.org/wp-includes/customize/a22.exe

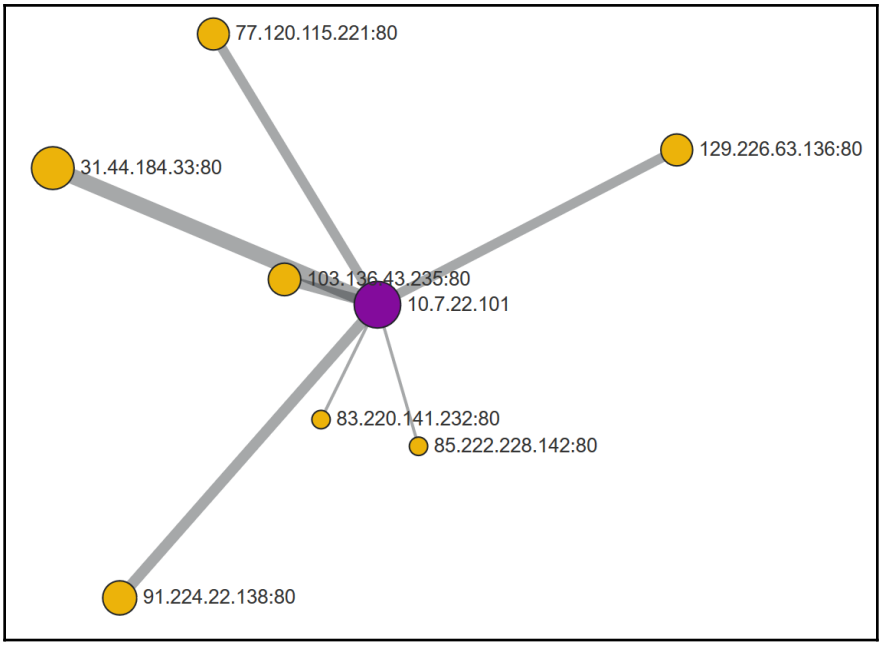
4. Moloch provides additional information for the session. In the same session row as the info URL, `ectcnepal[.]org/wp-includes/customize/a22.exe`, click the green box at the far left. This opens the following:

5. Further down, under the heading **HTTP**, is valuable data concerning the connection:



An analysis of this HTTP information indicates that the `a22.exe` file was downloaded from the site indicated. From here, a responder can use additional tools to extract the file for further analysis.

Another feature that is useful with Moloch is the ability to visualize connections. At the top of the Moloch web application is **connection**. Click on **connection** and the following appears:



Next, let's have a look at the Wireshark tool.

Wireshark

Wireshark is one of the most popular packet capture analysis tools available to incident response analysts. In addition to the ability to capture packets, there are a great many other features that are available. As entire volumes and training courses are built around this platform, it is impossible to identify every feature. Therefore, this chapter will focus on some of the key features of Wireshark that are most applicable to an incident investigation.

Arguably, Wireshark is the packet analyzer of choice for IT and security professionals. Due to the ubiquity of the application, there are a wide variety of resources available for additional training on Wireshark and its capability. The Wireshark site at `https://www.wireshark.org/` contains a great deal of information. Furthermore, the site at `https://www.chappell-university.com/` contains exercises and training packet captures to hone skills around analysis.

Because Wireshark is a feature-rich tool, there are some settings that lend themselves more to network traffic analysis that are outside incident response activities. As a result, there are some changes to be made to better assist the incident response analyst with performing packet capture analysis in relation to an incident investigation:

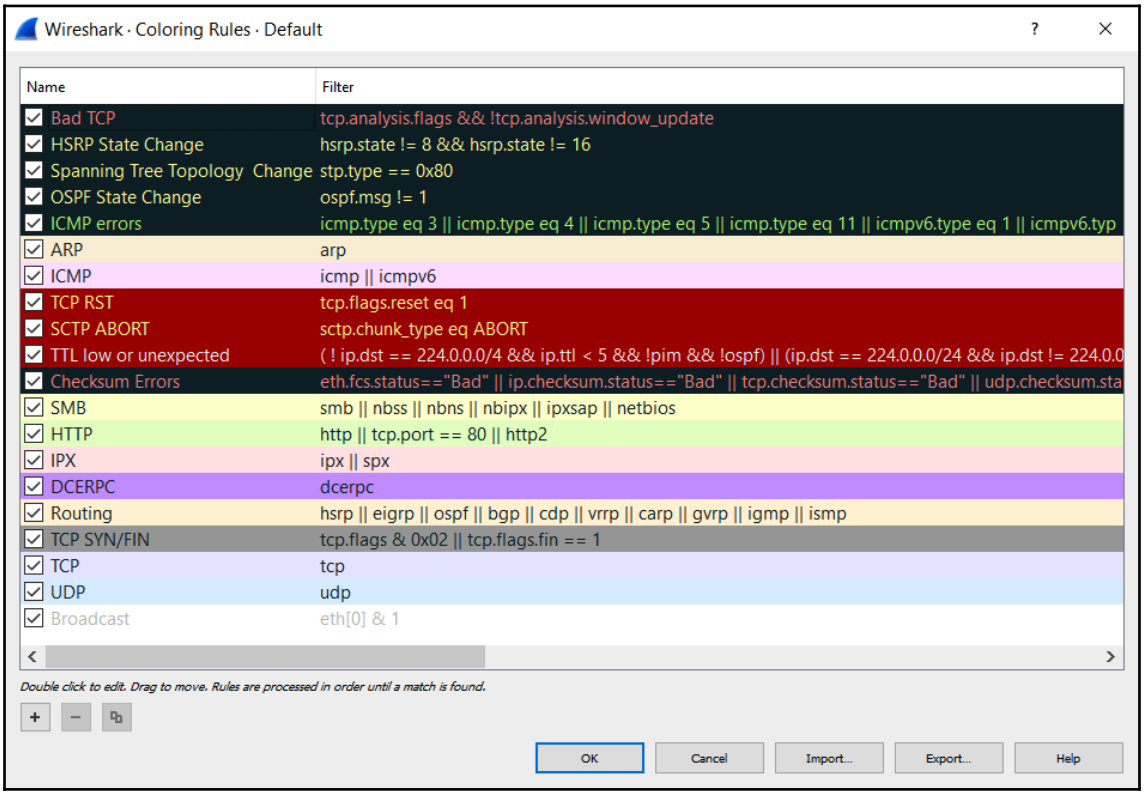
- **Time**: The time setting in Wireshark allows for several options. These include the time of the packet since 1/1/1970 or since the start of the packet capture. One of these options, which can be useful in an incident investigation, is the date and time that the individual packets have been captured. This allows analysts to correlate the date and time of other suspicious or malicious activity with the date and time of specific traffic within the packet capture. To enable this, navigate to **View** and then to **Time Display Format**. From there, choose one of the time options such as **Date and Time of Day** or **Time of Day**. Another option to consider is utilizing the UTC time options as well. This is very useful if the internal network utilizes UTC rather than local time. Also, the time can be set all the way to nanoseconds.
- **Name resolution**: The name resolution setting allows analysts to toggle between seeing the IP address of source and destination hosts and hostname resolution. This is useful if an analyst is examining a packet capture and wants to determine if there are any suspicious hostnames found. For example, if the packet capture is opened, the following shows the IP addresses:

No.	Time	Source	Destination	Protocol
1	0.000000	10.3.13.101	10.3.13.1	DNS
2	0.030692	10.3.13.1	10.3.13.101	DNS
3	0.556303	10.3.13.101	88.198.14.102	TCP
4	0.739115	88.198.14.102	10.3.13.101	TCP
5	0.739762	10.3.13.101	88.198.14.102	TCP
6	0.739961	10.3.13.101	88.198.14.102	HTTP
7	0.740053	88.198.14.102	10.3.13.101	TCP
8	2.084099	88.198.14.102	10.3.13.101	TCP
9	2.084368	88.198.14.102	10.3.13.101	TCP
10	2.084401	88.198.14.102	10.3.13.101	TCP
11	2.084670	88.198.14.102	10.3.13.101	TCP
12	2.084701	10.3.13.101	88.198.14.102	TCP
13	2.084715	88.198.14.102	10.3.13.101	TCP
14	2.084740	88.198.14.102	10.3.13.101	TCP
15	2.084990	10.3.13.101	88.198.14.102	TCP
16	2.085834	88.198.14.102	10.3.13.101	TCP
17	2.085905	88.198.14.102	10.3.13.101	TCP
18	2.085924	88.198.14.102	10.3.13.101	TCP
19	2.085941	88.198.14.102	10.3.13.101	TCP

To determine the hostnames, navigate to **View** and then **Name Resolution**. Click on **Resolve Network Addresses**. Wireshark will then resolve the IP addresses to hostnames:

No.	Time	Source	Destination	Protocol
1	0.000000	10.3.13.101	10.3.13.1	DNS
2	0.030692	10.3.13.1	10.3.13.101	DNS
3	0.556303	10.3.13.101	rozhan-hse.com	TCP
4	0.739115	rozhan-hse.com	10.3.13.101	TCP
5	0.739762	10.3.13.101	rozhan-hse.com	TCP
6	0.739961	10.3.13.101	rozhan-hse.com	HTTP
7	0.740053	rozhan-hse.com	10.3.13.101	TCP
8	2.084099	rozhan-hse.com	10.3.13.101	TCP
9	2.084368	rozhan-hse.com	10.3.13.101	TCP
10	2.084401	rozhan-hse.com	10.3.13.101	TCP
11	2.084670	rozhan-hse.com	10.3.13.101	TCP
12	2.084701	10.3.13.101	rozhan-hse.com	TCP
13	2.084715	rozhan-hse.com	10.3.13.101	TCP
14	2.084740	rozhan-hse.com	10.3.13.101	TCP
15	2.084990	10.3.13.101	rozhan-hse.com	TCP
16	2.085834	rozhan-hse.com	10.3.13.101	TCP
17	2.085905	rozhan-hse.com	10.3.13.101	TCP
18	2.085924	rozhan-hse.com	10.3.13.101	TCP
19	2.085941	rozhan-hse.com	10.3.13.101	TCP

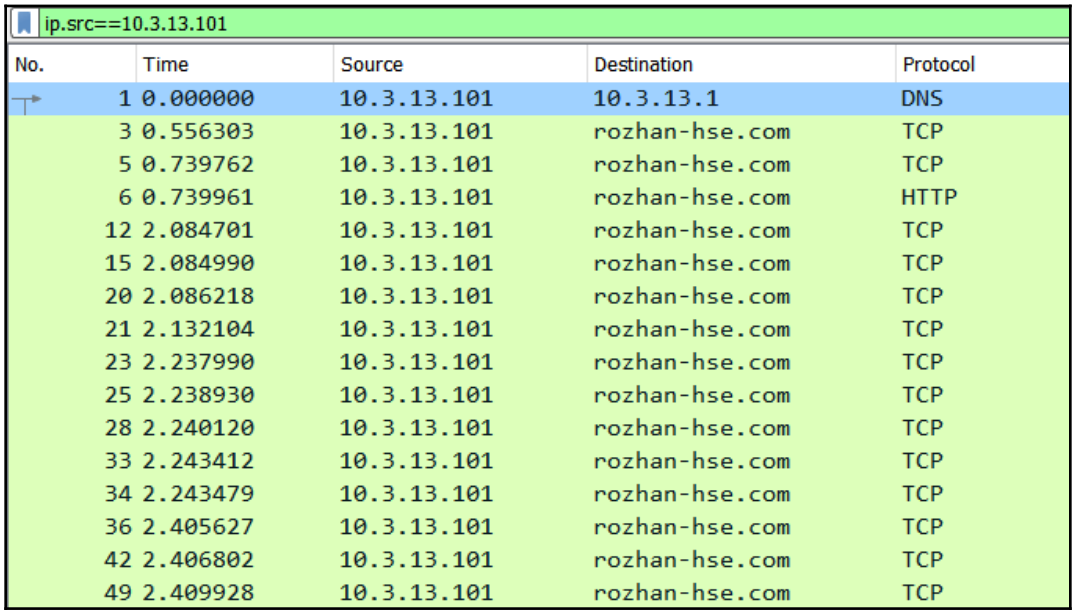
- **Colorize packet list**: This feature allows analysts to toggle between a blank background of the packet list or to allow Wireshark to color-code the packets:



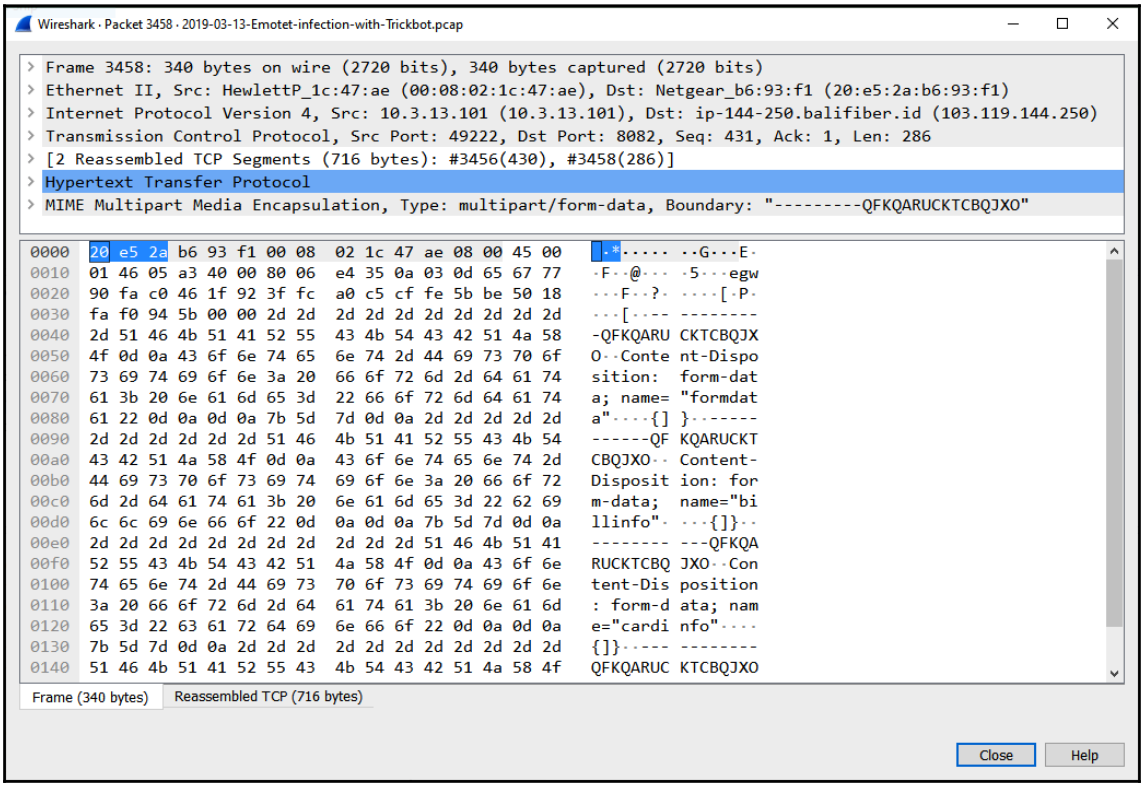
For the purposes of this chapter, an exploration of Wireshark will be done utilizing a packet capture found on Malware Traffic Analysis at `https://www.malware-traffic-analysis.net/2019/03/13/index.html`. This packet capture is provided along with a scenario involving a user that downloads a crypto locker malware strain contained within a Word document. For the purposes of this chapter, several key elements of the packet capture will be identified. Prior to examining the packet capture, Wireshark was configured so that date and time are visible, as well as the hostnames identified.

The following are some of the features in Wireshark that provide key pieces of information from the packet capture:

- **Display filters**: One of the most important features is the ability to filter packet captures on a wide range of services and ports. Filters can also be utilized on the source and destination IP addresses. For example, an incident response analyst would like to filter traffic on the source IP address of `10.3.13.101`. By right-clicking on the IP address in the packet capture window and navigating to **Apply as Filter** and then **Selected**, the analyst can select the IP address as a filter. This filter then appears in the filter bar with the syntax `ip.src==10.3.13.101`:



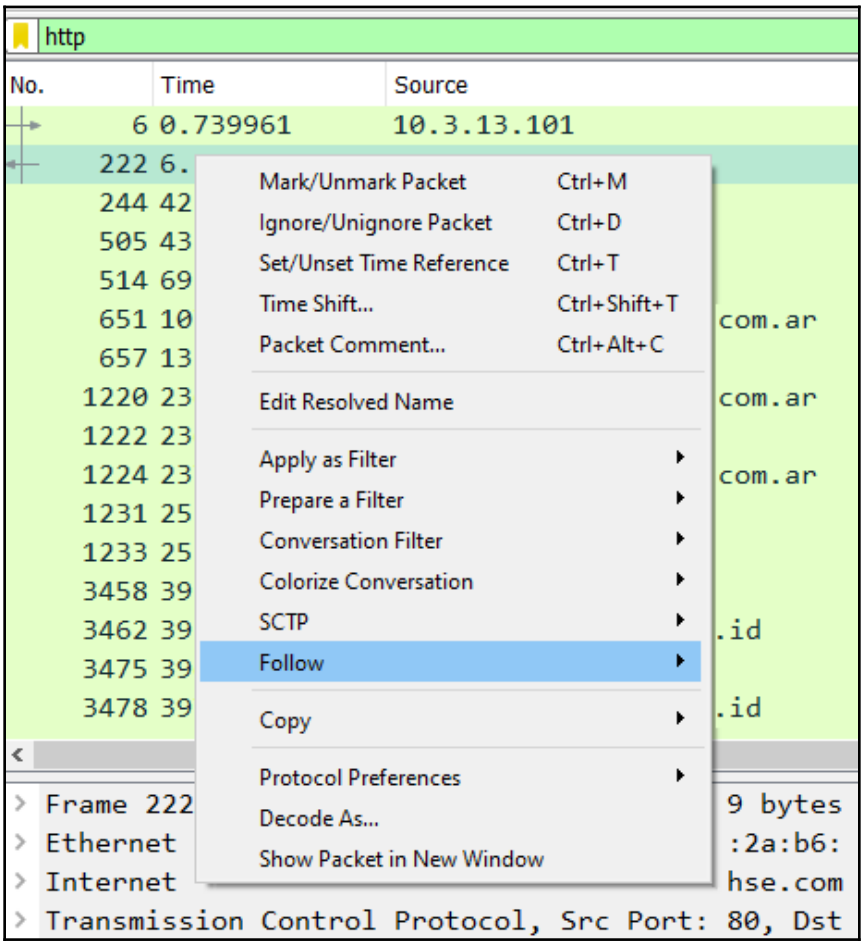
- **Host identification**: Another key aspect to the analysis of packet captures is to identify the localhost, if applicable. Considering that this packet capture is from a single host, identifying the hostname, IP address, and MAC address is straightforward. By double-clicking on the individual packet, a great deal of information is found:



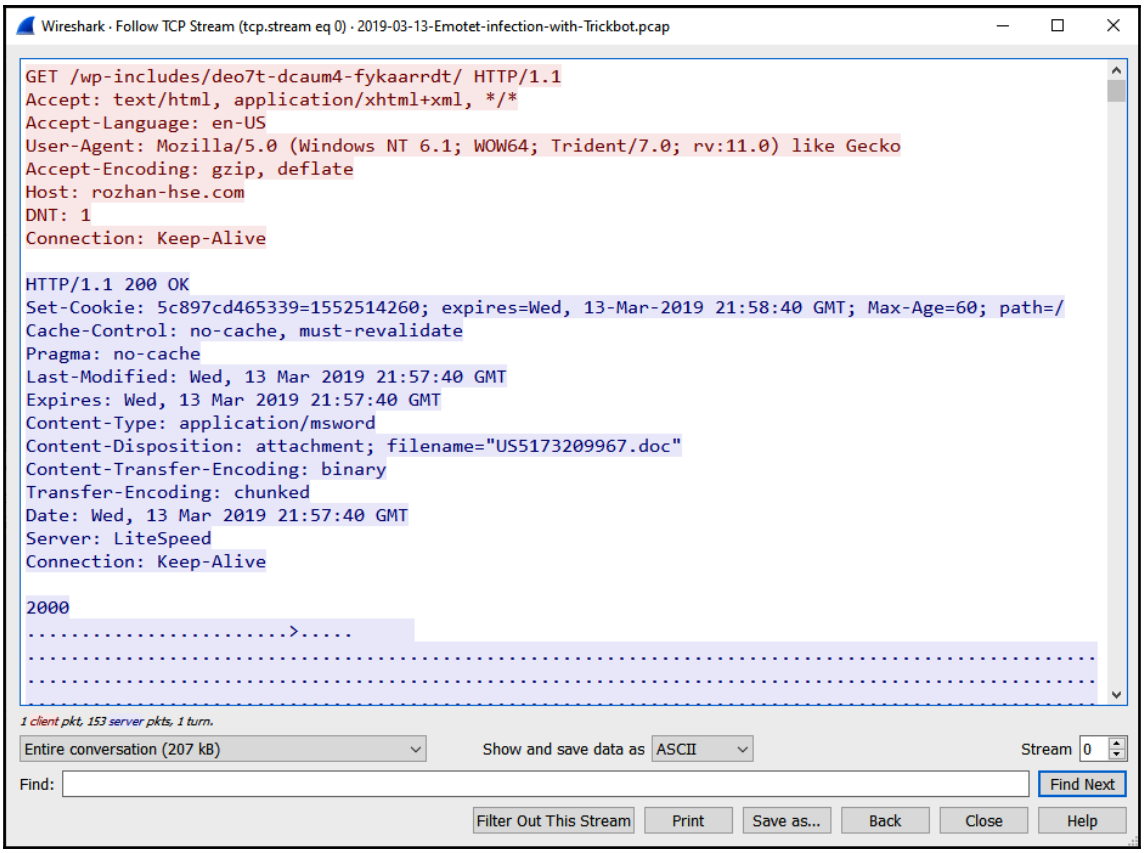
- **Physical connection identification**: In this packet, the analyst can identify the source of the traffic from the **Ethernet II** and **Internet Protocol Version 4** (**IPV4**) lines. In this case, the source of the traffic is the Hewlett Packard device located at `10.3.13.101` and the destination located at `103.119.144.250`. By examining the **Ethernet II** line, an analyst would be able to identify the physical connections for both systems. Finally, an analysis of the preceding data reveals that although this is an HTTP packet, it is over a non-standard HTTP port, as the destination port is `8082`. While this may be benign, it could also be something that should be followed up.
- **Protocol identification**: In this case, there was a good deal of HTTP connections, due to the activity of the user. As a result, the primary transmission of the malware was quite possibly through an HTTP connection. Wireshark has a number of filters that allow analysts to limit the packet capture results with specific parameters. In the top green dialog box, enter `http`. Pay attention while entering in the filter, as there will be several different filters available. Once the filter is typed in, click the right-facing arrow located at the far right of the dialog box. Wireshark will now limit the view of packets to those that are utilizing the HTTP protocol:

No.	Time	Source	Destination	Protocol
6	0.739961	10.3.13.101	rozhan-hse.com	HTTP
222	6.111115	rozhan-hse.com	10.3.13.101	HTTP
244	42.716145	10.3.13.101	aliyev.org	HTTP
505	43.744089	aliyev.org	10.3.13.101	HTTP
514	69.467806	10.3.13.101	101.152.220.201.itc.com.ar	HTTP
651	107.988909	101.152.220.201.itc.com.ar	10.3.13.101	HTTP
657	133.250172	10.3.13.101	101.152.220.201.itc.com.ar	HTTP
1220	236.160800	101.152.220.201.itc.com.ar	10.3.13.101	HTTP
1222	236.190844	10.3.13.101	101.152.220.201.itc.com.ar	HTTP
1224	237.025219	101.152.220.201.itc.com.ar	10.3.13.101	HTTP
1231	251.073395	10.3.13.101	ip.anysrc.net	HTTP
1233	251.219323	ip.anysrc.net	10.3.13.101	HTTP
3458	393.641008	10.3.13.101	ip-144-250.balifiber.id	HTTP
3462	394.534185	ip-144-250.balifiber.id	10.3.13.101	HTTP
3475	396.180268	10.3.13.101	ip-144-250.balifiber.id	HTTP
3478	397.048160	ip-144-250.balifiber.id	10.3.13.101	HTTP

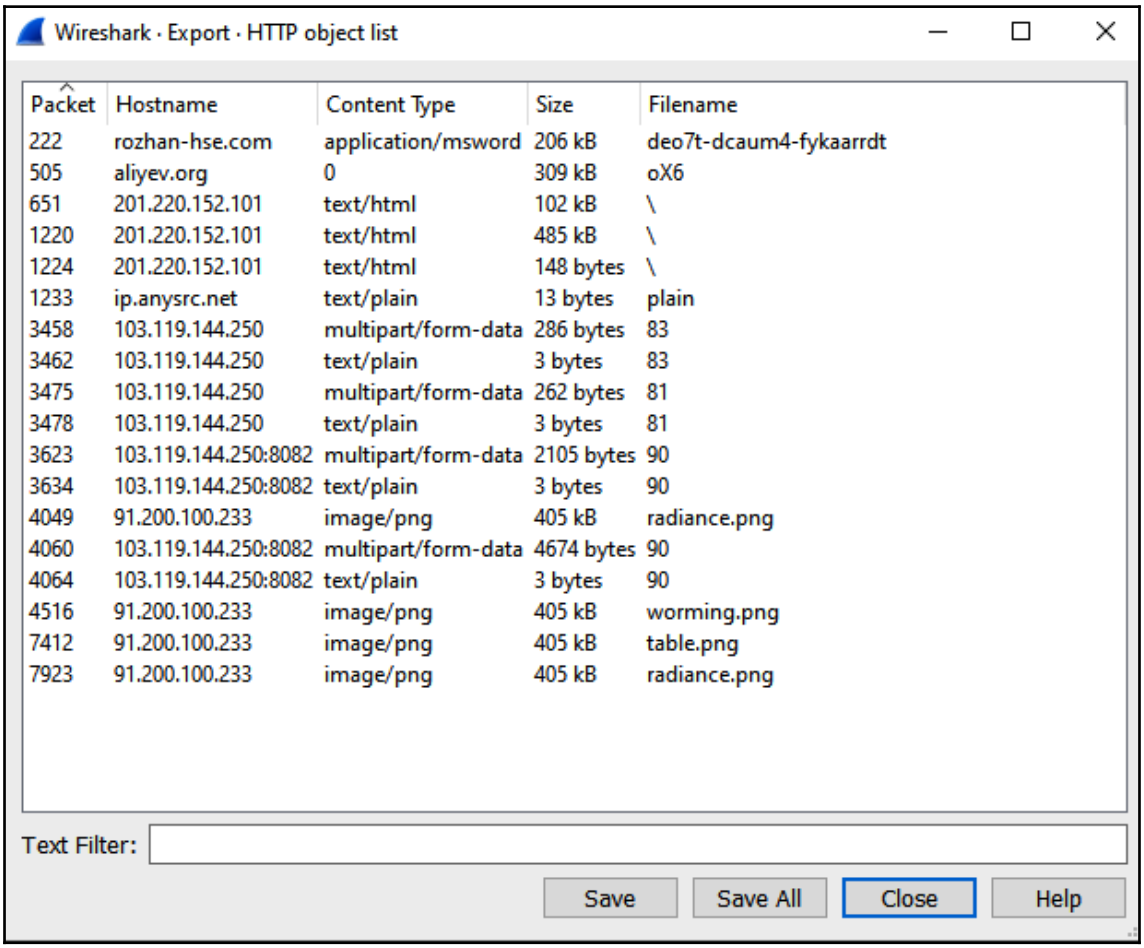
- **Hostname identification**: Parsing through the packet capture source and destination hostnames, one hostname appears to be suspicious. This host, `rozhan-hse.com`, may be a suspect URL. Another feature of Wireshark is the ability to follow the TCP or HTTP stream of communication between the source and destination hosts. Right-click on the hostname `rozhan-hse.com` and the following appears:



A second window will appear; click on **HTTP Stream** and a third window appears. This window contains the HTTP packets in a format that can be read. The incident response analyst can review this output to determine what types of files may have been sent or received:



- **Packet stream examination**: An examination of the **Follow TCP Stream** output indicates that an HTTP `GET` command is reaching out to the `deo7t-dcaum4-fykaarrdt` file. An analyst may want to extract this file for analysis. Click on **File** and then **Export Objects**, and then **HTTP**, and a window will appear listing all of the files associated with the HTTP connections. The list can be sorted on any of the fields at the top of the window. In this case, select the hostname and scroll down until the suspected URL is located:



From here, the analyst can click on the file and save it onto the local system for later analysis. `Chapter 12`, *Malware Analysis for Incident Response*, will take select files and evaluate them for malicious code.

Wireshark is a powerful tool for conducting detailed analysis of packet captures. The ability to drill down to individual packets and dissect them allows analysts to gain a very detailed sense of what is contained within the traffic running to and from external hosts, as well as to and from internal hosts. This visibility can afford the analyst possible insight into how an infected host communicates with an external host, or even identify other hosts that may have become compromised.

Summary

Security incidents not only produce trace evidence on host systems, but also leave traces throughout the devices and traffic flows within a network. The ability to analyze this trace evidence will allow incident response analysts to have a better understanding of what type of incident they are investigating, as well as potential actions that can be taken. This chapter addressed how to evaluate log files through the rapid process of blacklist comparison or DNS analysis to log analysis utilizing the Elastic Stack or other SIEM. Augmenting this primary method of network evidence evaluation was the inclusion of NetFlow analysis, and examining packet captures with Moloch and Wireshark. Network evidence is a critical component of incident investigation. This trace evidence, taken in conjunction with evidence obtained from potentially compromised websites, goes a long way in allowing analysts to reconstruct the events of an incident.

The next chapter will move the focus from network traffic to the host, and memory analysis will be explored.

Questions

1. A filtered log review is one where the responder or analyst filters out specific logs based on a set parameter.

   A) True
   B) False

2. What is not a component of the Elastic Stack?

   A) Elasticsearch
   B) Log forwarder
   C) Logstash
   D) Kibana

3. Which packet analysis tool places the packet capture into sessions as the default view?

   A) Wireshark
   B) NetFlow
   C) Elastic Stack
   D) Moloch

4. Wireshark does not allow for DNS name resolution.

   A) True
   B) False

Further reading

- *Elasticsearch 7.0 Cookbook - Fourth Edition*: `https://www.packtpub.com/big-data-and-business-intelligence/elasticsearch-70-cookbook-fourth-edition`
- Malware traffic analysis: `https://www.malware-traffic-analysis.net`
- Moloch: `https://molo.ch/`
- Chappell University: `https://www.chappell-university.com/`
- Cisco IOS NetFlow: `https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html`

8
Analyzing System Memory

For a long time, law enforcement and other organizations performing digital forensic tasks associated with incident investigations often relied on methodologies that focused on evidence contained within the hard drive of a machine. Procedures dictated that the system should be powered down and the hard drive removed for imaging. While this methodology and the associated procedures were effective at ensuring the integrity of the evidence, this overlooked the wealth of information that was contained within the **Random Access Memory** (**RAM**), or memory for short, of the targeted system. As a result, incident response analysts began to focus a great deal of attention on ensuring that appropriate methods were employed that maintained the integrity of this evidence, as well as giving them a platform from which to obtain information of evidentiary value.

This chapter will focus on the types of evidence that can be located within the memory of a system, the tools and techniques available to incident response analysts, and, finally, how to analyze this information to obtain a clear understanding of how the system was compromised. In addition, these techniques can also be integrated into the analysis of other evidence, such as network log files and files located on the targeted system.

In this chapter, the main topic areas will be addressed:

- **Memory analysis overview**: This section addresses the critical data points that can be discovered through proper memory analysis.
- **Memory analysis methodology**: A structured approach is important to ensure that responders are able to extract the necessary data.
- **Memory analysis with Redline**: The first tool that will be reviewed is Mandiant Redline, a GUI-based tool that allows responders to examine memory captures.
- **Memory analysis with Volatility**: Often thought of as the gold standard of memory analysis, this command-line tool has extensive features for data acquisition and analysis.
- **Memory analysis with Strings**: A simple but effective tool that affords responders the ability to cull data from those areas of memory that other tools may miss.

At the end of this chapter, the responder will have both an understanding of the methodology and the tools necessary for finding data points, analyzing them, and extracting other evidence for follow-up analysis.

Memory analysis overview

When discussing analyzing the memory of a system, there are two terms that are used interchangeably. The terms RAM and memory are used to describe the portion of the computer internal systems where the operating system places data utilized by applications and the system hardware while that application or hardware is in use. What makes RAM or memory different from storage is the volatile nature of the data. Often, if the system is shut down, the data will be lost.

One change in operating systems that has had a direct impact on memory analysis is the advent of the 64-bit OS. The use of a 64-bit register allows the OS to reference a total of 17,179,869,184 GB of memory. When compared to the 32-bit OS, this is several million more times the amount of data previously available. As a result, there is a good deal of data contained within RAM at the time a system is running that is valuable in incident investigation. These include the following:

- Running processes
- Loaded **Dynamic Link Libraries** (**DLL**)
- Loaded device drivers
- Open registry keys
- Network connections
- Command history

As the necessity for analyzing the memory of systems has increased, there are several tools that analysts have at their disposal. This chapter will focus on three such tools; all of them are either open source or freeware and can be deployed easily. These tools allow analysts to gain critical insight into the activity of exploits and malware that have impacted a system.

Throughout this chapter, two memory captures will be utilized. The first memory capture is from a Windows system that has been infected by the Stuxnet virus. The memory image can be downloaded from the following site: `jonrajewski.com/data/Malware/stuxnet.vmem.zip`. The second is another Windows system infected with the Cridex banking trojan and can be downloaded from the following site: `http://files.sempersecurus.org/dumps/cridex_memdump.zip`. While both of the malware infections are relatively old, they are useful for highlighting specific features of the toolsets we are going to examine.

Memory analysis methodology

When examining system memory, it is advisable for analysts to follow a methodology. This ensures that all potential evidence is uncovered and can be utilized in an incident investigation. There are a variety of methodologies that can be leveraged. Which specific methodology that is used can often be dependent on the type of incident. For example, a methodology that is geared towards identifying indicators of compromise around a malware infection may yield a great deal of information but may not be the best approach if the analyst has evidence from other network sources of a suspect IP address.

One of the chief aims of memory analysis is to identify potentially malicious processes or executables that can be extracted and examined. Much of the material that is present in this chapter will carry over into `Chapter 12`, *Malware Analysis for Incident Response*, where the extracted data will be further analyzed.

SANS six-part methodology

The SANS institution makes use of a six-part methodology for the analysis of memory images. This process is designed to start from an overall view of what is running to identifying and accessing the malicious software. The SANS methodology follows the following steps:

1. **Identify rogue processes**: Malware often hides its behavior behind processes that on the surface may seem legitimate. Uncovering these involves identifying what processes are running, finding the location in the operating system they are running from, and verifying that only legitimate processes are in use. Sometimes processes are hidden in plain sight, and adversaries change a single letter in a process name. Other times, they will attempt to execute a process from an illegitimate source.
2. **Analyze process DLLs and handles**: Once a process or multiple processes have been identified as rogue, the next step is to examine the DLL files associated with the process as well as other factors such as account information. DLL files are often leveraged by malware coders to hide their activity. Techniques for using DLL files to compromise a system include techniques where malware coders insert their own malicious DLL files as part of the malware. Other techniques include DLL injection, where a path is written in the process to one of the malicious DLLs.

3. **Review network artifacts**: Malware, especially multi-stage malware, requires a connection to the internet. Even systems that are fully compromised often beacon out to C2 servers. Active and listening network connections are contained within the memory of these systems. Identifying external host IP addresses may give some insight into what type of compromise has taken place.
4. **Look for evidence of code injection**: Techniques such as process hollowing, and unmapped sections of the memory are often used by advanced malware coders. Memory analysis tools help analysts to find evidence of these techniques.
5. **Check for signs of a rootkit**: Achieving persistence is a goal with many external threat actors. If they are able to achieve the initial compromise of the system, it is critical that they maintain that. As a result, adversaries might use a Rootkit or malware that imbeds itself deep within the operating system. This malware allows the adversary to have continuous and often elevated access to the system while remaining undetected.
6. **Dump suspicious process and drivers**: After locating any suspicious processes or executables, analysts need to be able to acquire them for later analysis with additional tools.

Next, we will look at the network connections methodology.

Network connections methodology

In many incidents, the first indication that a system has been compromised is attempted or completed connections to external hosts. Detection mechanisms such as firewalls or web proxies may indicate that a system or systems are attempting to communicate with suspect external hosts. From this starting position, it may be possible to identify potential malware on a system:

- **Suspicious network connections**: Conducting a review of network connections on hosts that have been associated with external connections will often provide the process that is attempting to communicate.
- **Process name**: Examining the process from the network connections allows analysts to perform similar actions found within the SANS methodology. It is advisable for the analyst to also determine whether the identified process is one that often requires a network connection.
- **Parent process ID**: Further insight into the parent process is useful for determining whether the process is legitimate and has a legitimate need to communicate via a network connection.
- **Associated entities**: Finally, examining the associated DLLs and other artifacts brings us to the stage where they can be acquired and analyzed.

Let's now look at some memory analysis tools.

Memory analysis tools

There are several tools available to analysts for the review of memory images. Some tools provide a GUI for ease of use, while others operate via the command line, making them useful for scripting. For the purposes of this chapter, three tools will be examined. The first of these, Mandiant Redline, is a GUI-based memory analysis tool that examines memory images for signs of rogue processes and scores them based upon several factors. The second of these tools is Volatility, a command-line tool that allows analysts to drill into the details of the memory image and identify potentially malicious code. The final tool that will be examined is the Strings utility available in Linux. Strings allows keyword searching through GREP, which allows the responder to identify IOCs that may not be readily visible with the other tools.

Memory analysis with Redline

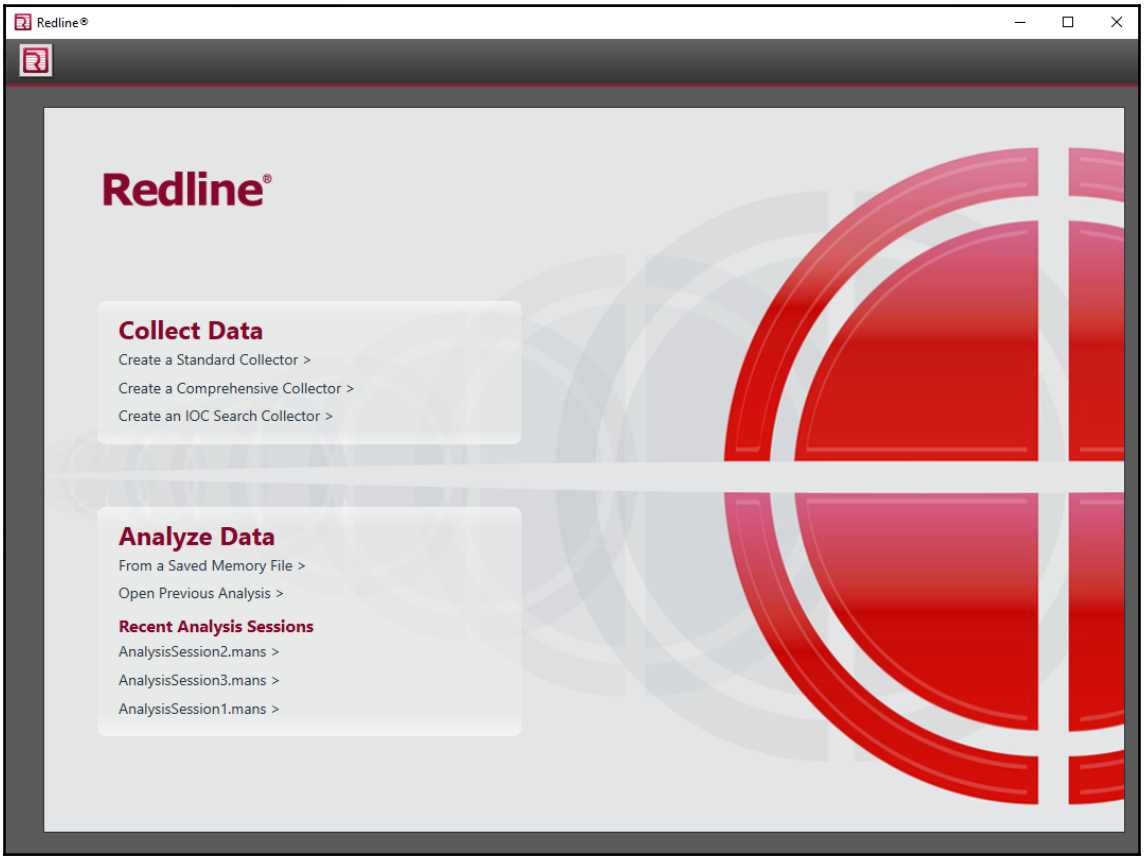
One powerful tool that analysts should include in their toolkits is Mandiant Redline. This Microsoft Windows application provides a feature-rich platform for analyzing memory images. These features include the ability to create a memory collector, although the tool will work with memory captures that have been performed via tools previously discussed. There is also the ability to utilize previously discovered **Indicators of Compromise** (**IOCs**) to aid in the examination. The tool can be downloaded at `https://www.fireeye.com/services/freeware/redline.html`. The download package includes a Microsoft Self Installer.

Redline analysis process

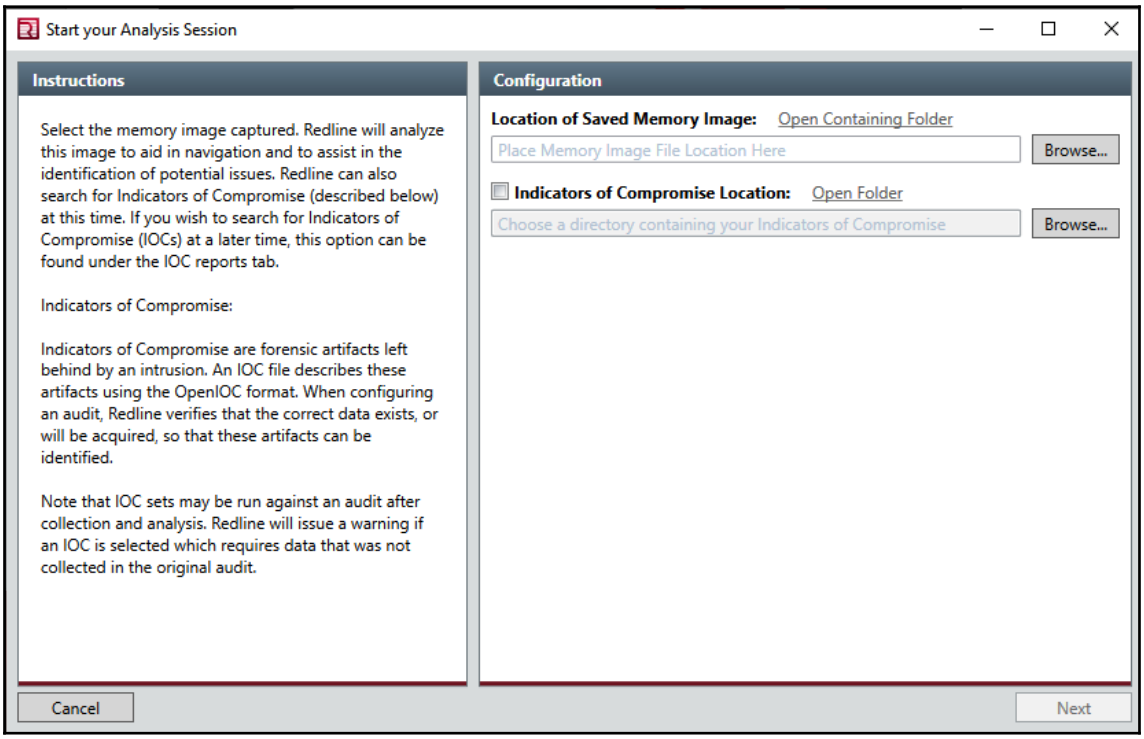
To demonstrate some of the key features of Redline, the Stuxnet memory capture will be used. To conduct an analysis, follow these steps:

1. Install Redline via the Microsoft Self Installer.

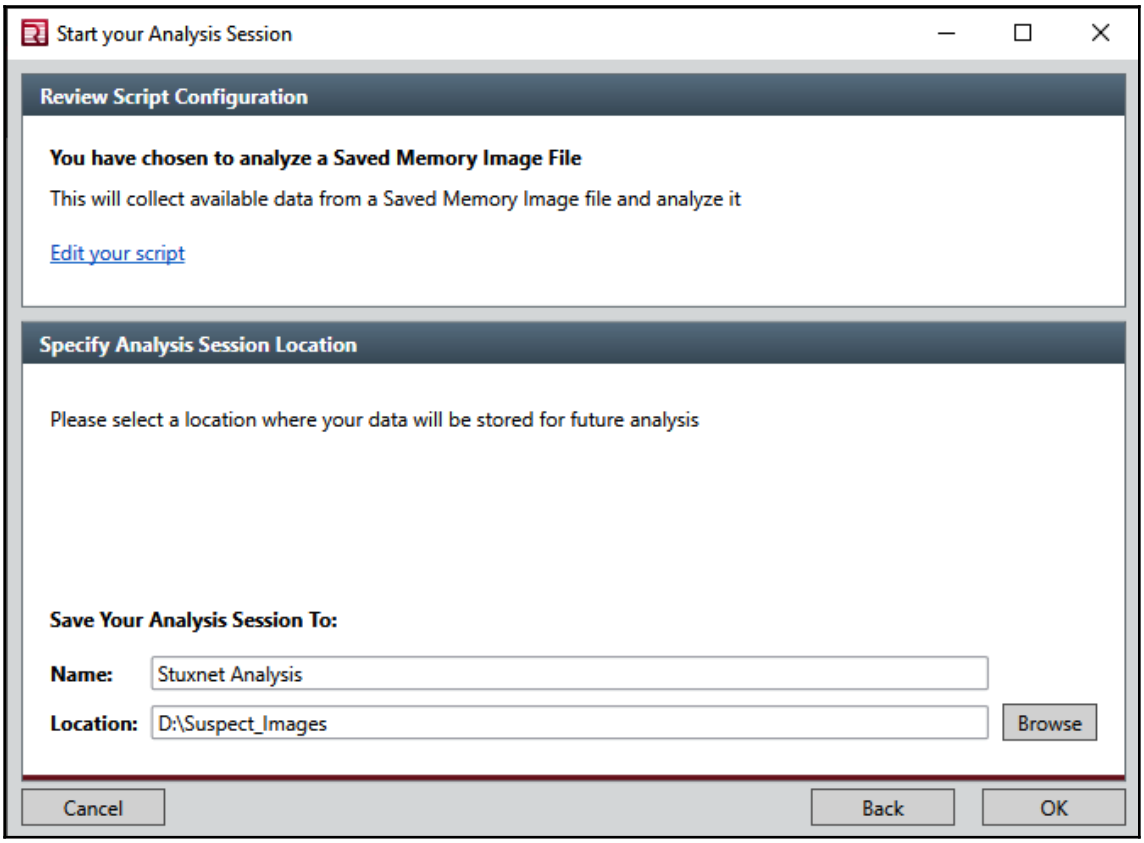
2. Once installed, double-click on the icon and the following screen will appear. There are a number of options broken down into two categories: **Collect Data** and **Analyze Data**. In this case, the Stuxnet memory capture will be analyzed.



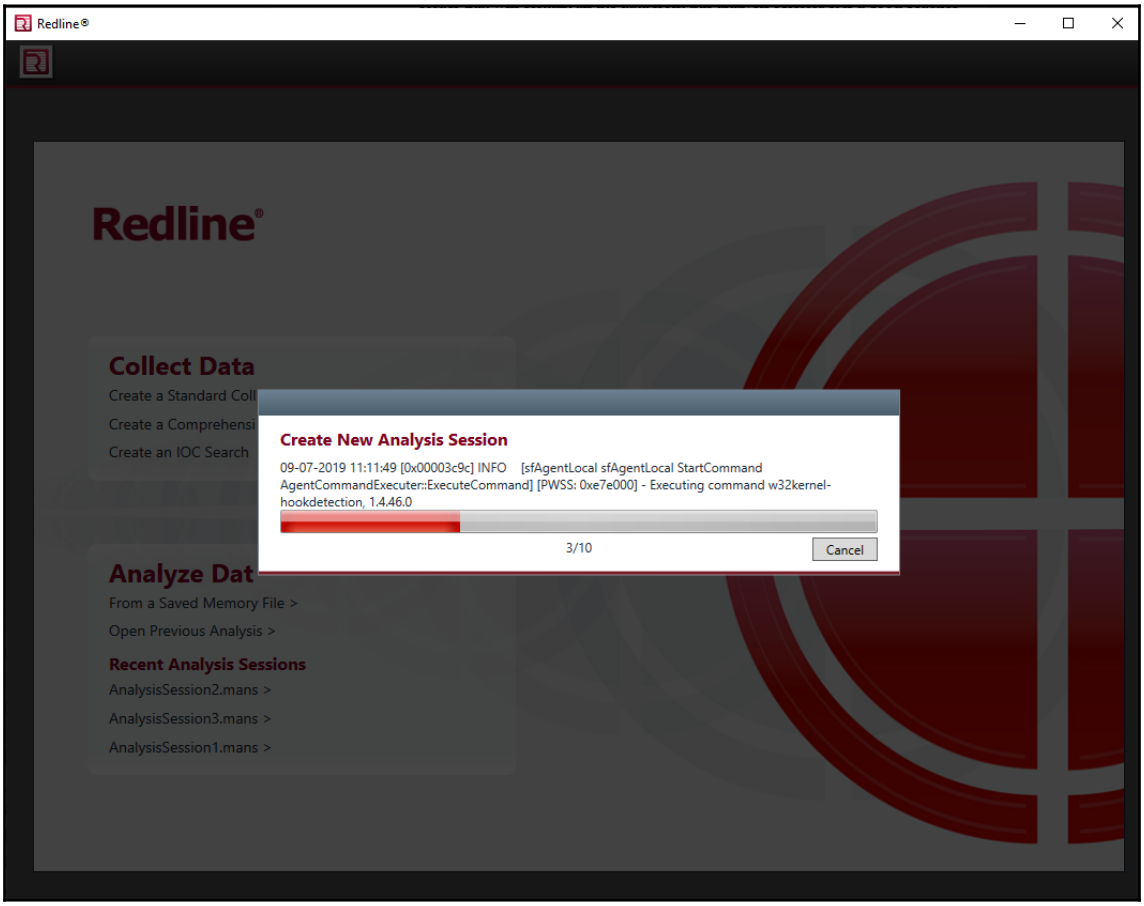
3. Click on **From a Saved Memory File** in the **Analyze Data** category. This will open a second window. Under **Location of Saved Memory Image**, navigate to the location of the memory file and select it. Click **Next**:



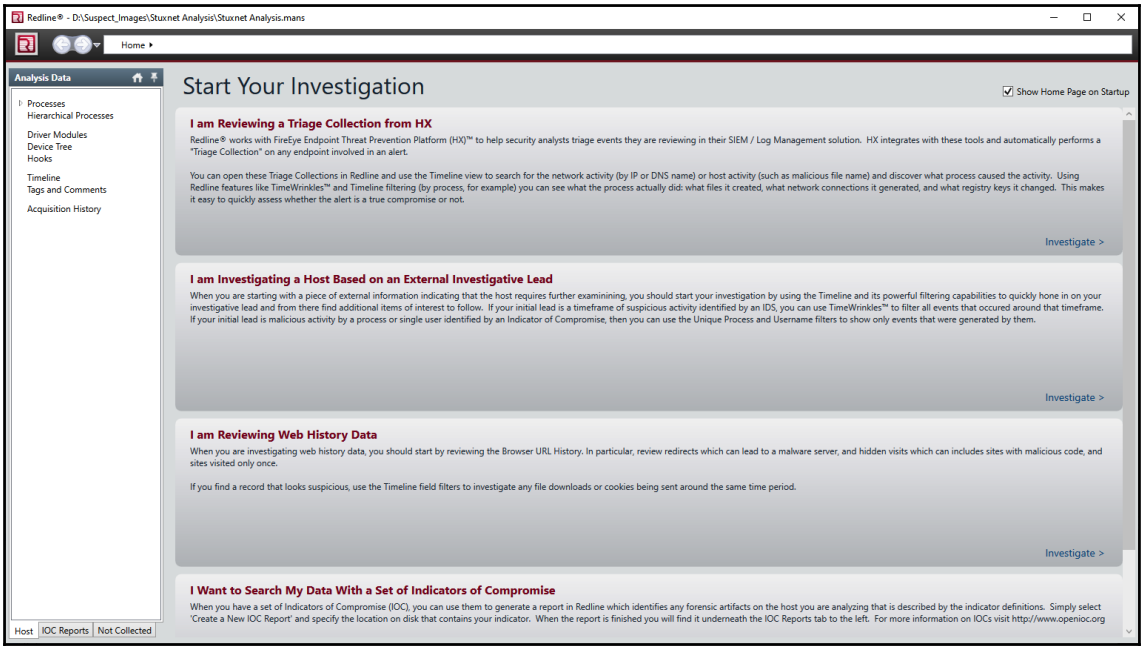
4. Once the memory file is loaded, the next screen will require a name for the session file that will be created by Redline. In this case, the filename `Stuxnet Analysis` will be utilized. Furthermore, select a folder that will contain all the data from this analysis session. It is a good practice to have separate folders for each session to ensure that each analysis is segregated. In the event that several systems are examined, this reduces the risk of commingling evidence. Once those parameters are set, click **OK**:



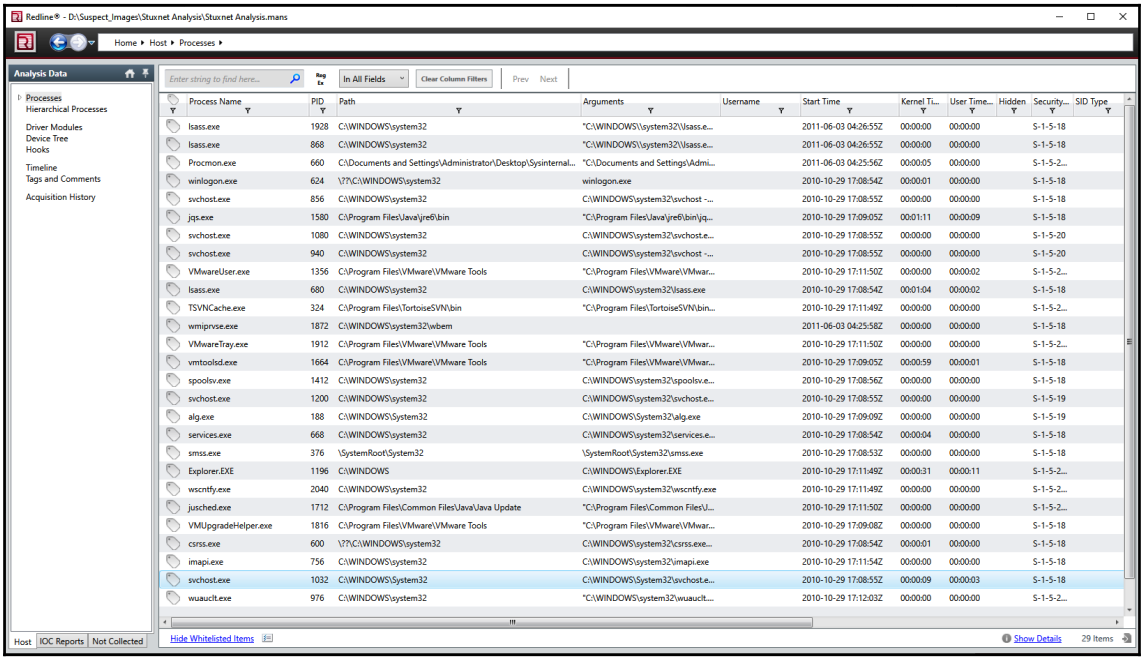
5. Redline will then begin the process of putting the data into a format for analysis. Depending on the size of the image, this may take several minutes:



6. After creating the analysis session, the following window will appear. For memory images that do not contain any other information, click on the section titled **I am Investigating a Host Based on an External Investigative Lead**:



7. The next window will appear that details the results of the analysis:



Next, we will look at Redline process analysis.

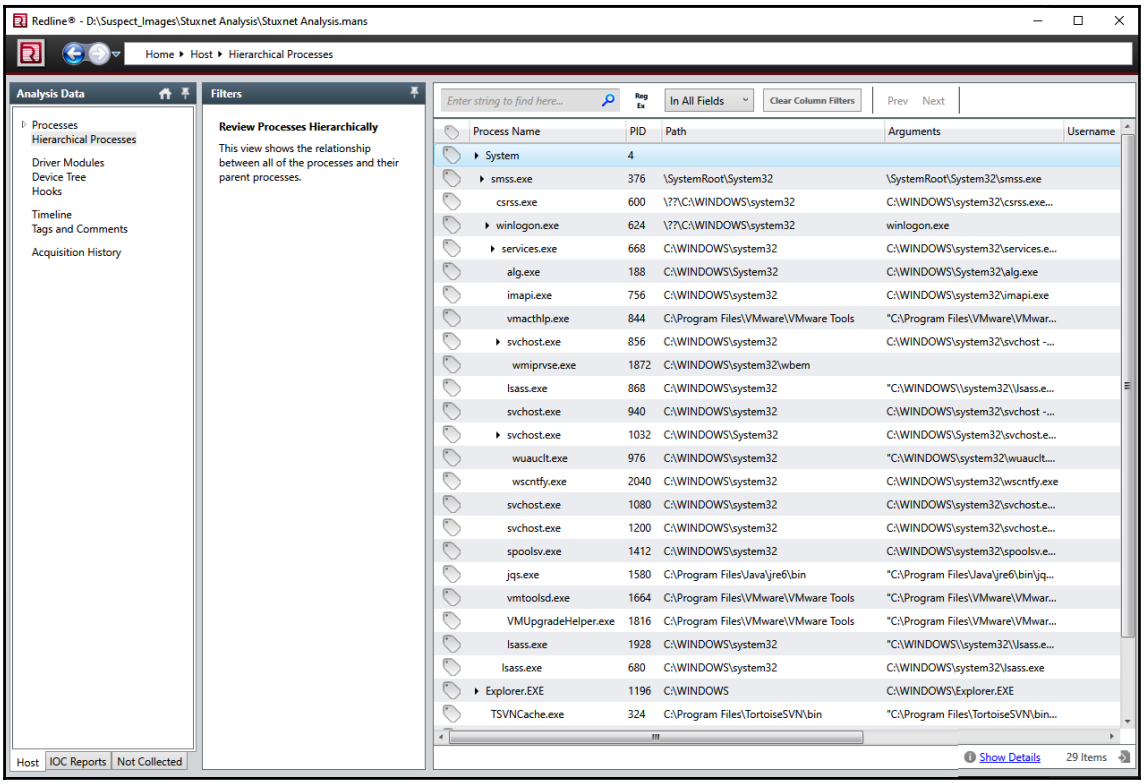
Redline process analysis

Once the memory image is done processing, the main page provides a comprehensive list of the running processes at the time of acquisition. This includes the full path of the process, the process name, and the **Process Identification** (**PID**) number.

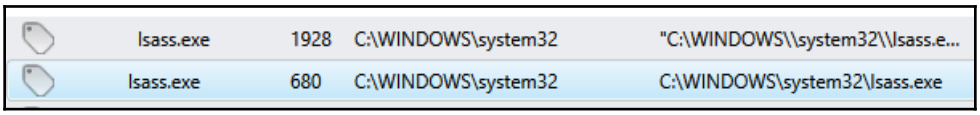
A review of the current processes indicates that there are three `lsass.exe` processes with PIDs of **1928**, **868**, and **680**. This is suspicious as multiple `lsass.exe` entries are indicative of malware behavior:

lsass.exe	1928	C:\WINDOWS\system32	"C:\WINDOWS\\system32\\lsass.exe"
lsass.exe	868	C:\WINDOWS\system32	"C:\WINDOWS\\system32\\lsass.exe"
Procmon.exe	660	C:\Documents and Settings\Administrator\Desktop\Sysinternal...	"C:\Documents and Settings\Administrator\Deskto...
winlogon.exe	624	\??\C:\WINDOWS\system32	winlogon.exe
svchost.exe	856	C:\WINDOWS\system32	C:\WINDOWS\system32\svchost -k DcomLaunch
jqs.exe	1580	C:\Program Files\Java\jre6\bin	"C:\Program Files\Java\jre6\bin\jqs.exe" -service -c...
svchost.exe	1080	C:\WINDOWS\system32	C:\WINDOWS\system32\svchost.exe -k NetworkSe...
svchost.exe	940	C:\WINDOWS\system32	C:\WINDOWS\system32\svchost -k rpcss
VMwareUser.exe	1356	C:\Program Files\VMware\VMware Tools	"C:\Program Files\VMware\VMware Tools\VMware...
lsass.exe	680	C:\WINDOWS\system32	C:\WINDOWS\system32\lsass.exe

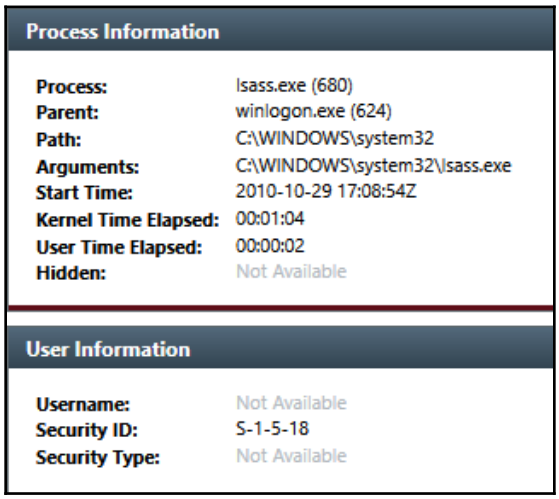
Another feature of Redline is the ability of responders to examine the processes in relation to their parent processes. The hierarchical process provides a view into the path that processes were executed. This is useful when examining memory for evidence of illegitimate or rogue processes. Click on **Hierarchical Processes** in the left pane under **Analysis Data**. This will change the view to show the processes in relation to their parent process:



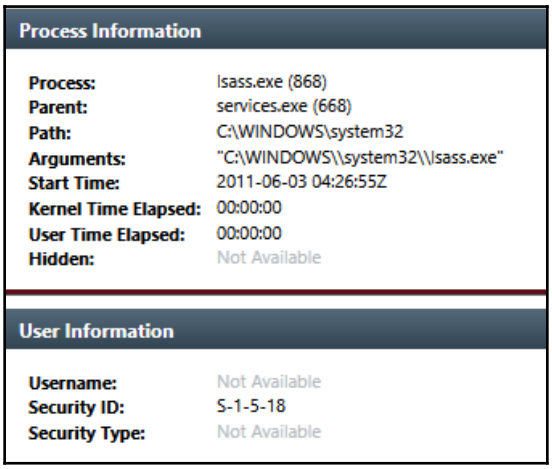
An examination of the processes shows two instances of `lsass.exe` with the process IDs of **680**, **868**, and **1928**. Having a single `lsass.exe` process is expected, but having three is suspicious because `lsass.exe` can be spoofed and is often abused by malware coders:



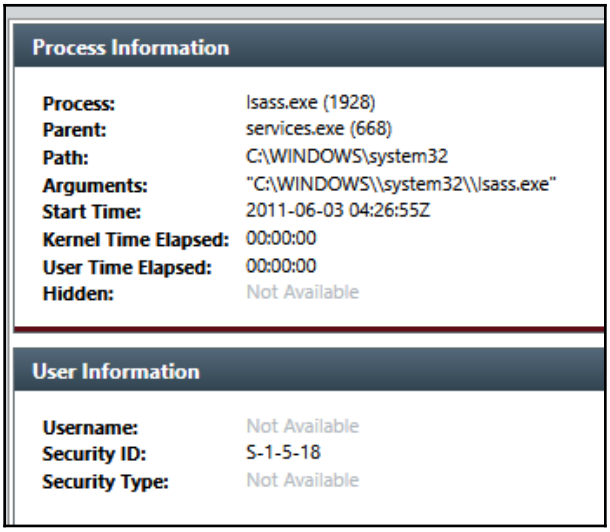
From here, the `lsass.exe` process with the ID **680** can be investigated further. If you double-click on the process, the following window appears:



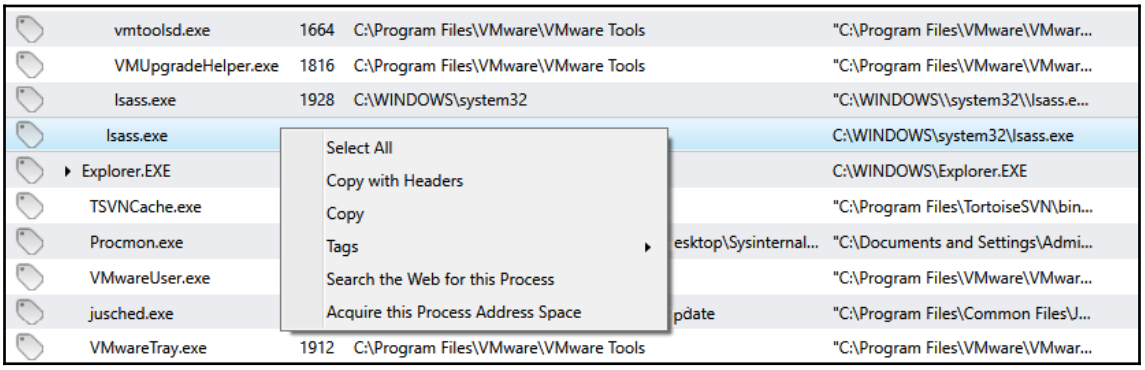
This expanded process information shows that this process was spawned by the `winlogon.exe` process from the `System32` folder. This may not be suspicious behavior, as this is expected. Since there should only be one `lsass.exe` process, the other two `lsass.exe` processes should be examined further. The `lsass.exe` with the PID of **868** appears to have been executed with the parent process of `services.exe`:



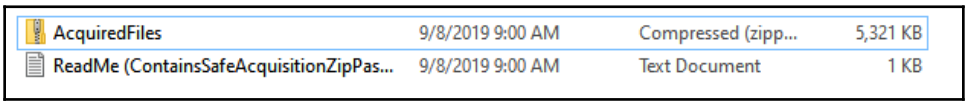
The same results are found with the `lsass.exe` with PID **1928**:



At this stage, it appears that this system has multiple `lsass.exe` processes, which in and of itself is suspicious. The next stage in this process is to dump the address space for all three processes for further examination. To dump the address space, right-click on the process and the following window will open:



Click on **Acquire this Process Address Space**. From here, Redline will acquire and dump the files associated with the files into a separate zip folder in the directory containing the analysis data:



As all three `lsass.exe` processes are suspect, they should all be acquired. From here, the files associated with the address spaces can be examined. This will be addressed in `Chapter 12`, *Malware Analysis for Incident Response*, where the files will be examined.

Redline will send the acquired files to a local system directory. There is the potential that the files may contain malware so a file exclusion for that directory should be set up in the system's antivirus so that files are not quarantined. Further, as there is the potential for malicious files to be placed on the system, great care should be taken with using this process to ensure the system does not become infected.

This chapter was only able to scratch the surface with the feature set of Redline. In addition to being able to identify processes and extract the data associated with it, Redline also has the ability to identify other key data points, including network connections, DLL files associated with processes, and the ability to see processes in a timeline view. The next section will demonstrate how to extract some of these same data points using the open source command-line tool Volatility.

Memory analysis with Volatility

Volatility is an advanced open source memory forensics framework. The primary tool within the framework is the Volatility Python script, which utilizes a wide array of plugins to perform the analysis of memory images. As a result, Volatility can be run on any operating system that supports Python. In addition, Volatility can be utilized against memory image files from most of the commonly distributed operating systems, including Windows for Windows XP to Windows Server 2016, macOS, and finally, common Linux distributions.

There is a range of plugins available for Volatility with more being developed. For the purposes of examining system memory, several plugins will be examined to ensure that the responder has sufficient information to conduct a proper analysis. It is recommended though that, prior to using Volatility, the analyst ensures that software is up to date and that any new plugins are explored to determine their applicability to the current incident investigation.

Installing Volatility

Volatility is available for Linux, Windows, and macOS. Information on installing on the various OSes is available at `https://www.volatilityfoundation.org/releases`. For this chapter, Volatility was installed on the Linux Ubuntu subsystem available on the Windows 10 OS. The following command will install Volatility in the Ubuntu subsystem, as well as other Linux OSes:

```
dfir@Desktop-SFARF6G~$ sudo apt-get install volatility
```

Once installed, the Volatility application can be run from any location.

Working with Volatility

Volatility has a basic syntax with individual commands. The first portion of the command is the memory image that is under analysis. Second is the profile of the memory image. OSes each have their own specific methods of memory addressing. The profile points Volatility to the specific areas of memory in which to find the appropriate data. Third, the command syntax will have a plugin that dictates the information the responder would like Volatility to access. Here is an example of Volatility command syntax:

```
dfir@Desktop-SFARF6G~$ volatility -f memoryimage.mem -profile=Win7SP1x64
plugin
```

There are several other options available, such as pointing Volatility to a PID or to output the results to a text file. Information on the various plugins is available on the Volatility GitHub page at `https://github.com/volatilityfoundation/volatility/wiki/Command-Reference`.

The Volatility section will work with the Cridex memory image discussed earlier. While this is a known infected image, it will provide a known context for understanding the data that can be obtained with Volatility. In this case, for ease of use, the memory image was renamed `cridex_laptop.mem` in the following examples.

Volatility image information

One of the key preliminary steps that must be completed prior to conducting a detailed examination of the system memory is to determine the exact OS of the system under investigation. Even if the analyst is certain of the OS, it is still a good practice to run the memory images against Volatility's `imageinfo` plugin. The output of this plugin identifies the potential profile of the memory image that becomes critical to utilizing the other plugins available. The following command is used:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem imageinfo
```

Volatility will output the following, which indicates that the profile most likely to produce the best results is WinXPSP2x86:

```
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/mnt/d/Suspect_Images/cridex_laptop.mem)
                      PAE type : PAE
                           DTB : 0x2fe000L
                          KDBG : 0x80545ae0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2012-07-22 02:45:08 UTC+0000
     Image local date and time : 2012-07-21 22:45:08 -0400
```

Next, let's look at the Volatility process analysis

Volatility process analysis

As was done in the Redline section, the first plugins that will be discussed are those that provide data around the processes running on the system at the time of the memory capture. The aim here is to identify those processes that appear suspicious and to identify any related data associated with them.

Process list

The first of these will be `pslist` plugin. The `pslist` command lists the current processes running in memory. This plugin outputs the offset, process name, **PID**, the number of threads and handles, and the date and time the process started and exited. Because the `pslist` plugin walks the doubly-linked list indicated by `PsActiveProcessHead`, it does not have the ability to detect hidden or unlinked process. To execute the plugin, enter the following into Command Prompt:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
pslist
```

The preceding command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01 UTC+0000
```

From the output, there does not appear to be anything suspicious right away. What is interesting is the `reader_sl.exe` executable. This stands out as a different file naming convention from the other processes. While there is no concrete data indicating that the file is malicious, it may be something to examine further.

Process scan

`psscan` is a useful plugin that allows the analyst to examine processes that have been terminated. As was previously discussed, `pslist` only shows active processes. `psscan` can provide data about the possibility of a rootkit through the examination of those processes that have been unlinked or hidden. The following command will execute the plugin:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
psscan
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID   PPID PDB        Time created
------------------ ---------------- ------ ------ ---------- ------------------------------
0x0000000002029ab8 svchost.exe         908    652 0x079400e0 2012-07-22 02:42:33 UTC+0000
0x000000000202a3b8 lsass.exe           664    608 0x079400a0 2012-07-22 02:42:32 UTC+0000
0x000000000202ab28 services.exe        652    608 0x07940080 2012-07-22 02:42:32 UTC+0000
0x000000000207bda0 reader_sl.exe      1640   1484 0x079401e0 2012-07-22 02:42:36 UTC+0000
0x00000000020b17b8 spoolsv.exe        1512    652 0x079401c0 2012-07-22 02:42:36 UTC+0000
0x000000000225bda0 wuauclt.exe        1588   1004 0x07940200 2012-07-22 02:44:01 UTC+0000
0x00000000022e8da0 alg.exe             788    652 0x07940140 2012-07-22 02:43:01 UTC+0000
0x00000000023dea70 explorer.exe       1484   1464 0x079401a0 2012-07-22 02:42:36 UTC+0000
0x00000000023dfda0 svchost.exe        1056    652 0x07940120 2012-07-22 02:42:33 UTC+0000
0x00000000023fcda0 wuauclt.exe        1136   1004 0x07940180 2012-07-22 02:43:46 UTC+0000
0x0000000002495650 svchost.exe        1220    652 0x07940160 2012-07-22 02:42:35 UTC+0000
0x0000000002498700 winlogon.exe        608    368 0x07940060 2012-07-22 02:42:32 UTC+0000
0x00000000024a0598 csrss.exe           584    368 0x07940040 2012-07-22 02:42:32 UTC+0000
0x00000000024f1020 smss.exe            368      4 0x07940020 2012-07-22 02:42:31 UTC+0000
0x00000000025001d0 svchost.exe        1004    652 0x07940100 2012-07-22 02:42:33 UTC+0000
0x0000000002511360 svchost.exe         824    652 0x079400c0 2012-07-22 02:42:33 UTC+0000
0x00000000025c89c8 System                4      0 0x002fe000
```

From the output of this plugin, it does not appear that any additional processes have exited. The responder can then start to look at the existing processes for any that may appear to be malicious.

Process tree

As was shown in the Redline section, it is necessary for responders to see what parent processes child processes are executed under. One indicator of a system being compromised is the identification of a process executed outside the normal parent process. The `pstree` plugin provides examiners with a tree-like structure that identifies the parent process that is executing a potential suspect process. The Cridex image is run with this plugin, utilizing the following command:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
pstreee
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Name                                               Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x823c89c8:System                                   4      0     53    240 1970-01-01 00:00:00 UTC+0000
. 0x822f1020:smss.exe                              368      4      3     19 2012-07-22 02:42:31 UTC+0000
.. 0x82298700:winlogon.exe                         608    368     23    519 2012-07-22 02:42:32 UTC+0000
... 0x81e2ab28:services.exe                        652    608     16    243 2012-07-22 02:42:32 UTC+0000
.... 0x821dfda0:svchost.exe                       1056    652      5     60 2012-07-22 02:42:33 UTC+0000
.... 0x81eb17b8:spoolsv.exe                       1512    652     14    113 2012-07-22 02:42:36 UTC+0000
.... 0x81e29ab8:svchost.exe                        908    652      9    226 2012-07-22 02:42:33 UTC+0000
.... 0x823001d0:svchost.exe                       1004    652     64   1118 2012-07-22 02:42:33 UTC+0000
..... 0x8205bda0:wuauclt.exe                      1588   1004      5    132 2012-07-22 02:44:01 UTC+0000
..... 0x821fcda0:wuauclt.exe                      1136   1004      8    173 2012-07-22 02:43:46 UTC+0000
.... 0x82311360:svchost.exe                        824    652     20    194 2012-07-22 02:42:33 UTC+0000
.... 0x820e8da0:alg.exe                            788    652      7    104 2012-07-22 02:43:01 UTC+0000
.... 0x82295650:svchost.exe                       1220    652     15    197 2012-07-22 02:42:35 UTC+0000
... 0x81e2a3b8:lsass.exe                           664    608     24    330 2012-07-22 02:42:32 UTC+0000
.. 0x822a0598:csrss.exe                            584    368      9    326 2012-07-22 02:42:32 UTC+0000
 0x821dea70:explorer.exe                          1484   1464     17    415 2012-07-22 02:42:36 UTC+0000
. 0x81e7bda0:reader_sl.exe                        1640   1484      5     39 2012-07-22 02:42:36 UTC+0000
```

An analysis of the results from the three plugins shows an interesting entry. PID **1640** is associated with the `reader_sl.exe` executable. The responder may focus on this due to the fact that it may not look like an application that should run. Further, the parent PID indicates that it was run via Windows Explorer:

```
0x821dea70:explorer.exe                          1484   1464
 0x81e7bda0:reader_sl.exe                        1640   1484
```

From here, the responder can supplement the existing process data with additional data, such as which DLLs are loaded and other ancillary data.

DLL list

Responders can also check the loaded DLL files associated with a process. This allows the analyst to determine whether a suspect process accessed these files when it was executed. For example, if a responder would like to examine the DLL files associated with one of the suspect processes, PID **1640**, the following command is run:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
-p 1640 dlllist
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
************************************************************************
reader_sl.exe pid:   1640
Command line : "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"
Service Pack 3

Base             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x00400000    0xa000     0xffff                                 C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe
0x7c900000   0xaf000     0xffff                                 C:\WINDOWS\system32\ntdll.dll
0x7c800000   0xf6000     0xffff                                 C:\WINDOWS\system32\kernel32.dll
0x7e410000   0x91000     0xffff                                 C:\WINDOWS\system32\USER32.dll
0x77f10000   0x49000     0xffff                                 C:\WINDOWS\system32\GDI32.dll
0x77dd0000   0x9b000     0xffff                                 C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000   0x92000     0xffff                                 C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000   0x11000     0xffff                                 C:\WINDOWS\system32\Secur32.dll
0x7c9c0000  0x817000     0xffff                                 C:\WINDOWS\system32\SHELL32.dll
0x77c10000   0x58000     0xffff                                 C:\WINDOWS\system32\msvcrt.dll
0x77f60000   0x76000     0xffff                                 C:\WINDOWS\system32\SHLWAPI.dll
0x7c420000   0x87000     0xffff                                 C:\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_6b128700\MSVCP80.dll
0x78130000   0x9b000     0xffff                                 C:\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_6b128700\MSVCR80.dll
0x773d0000  0x103000        0x1                                 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
0x5d090000   0x9a000        0x1                                 C:\WINDOWS\system32\comctl32.dll
0x5ad70000   0x38000        0x2                                 C:\WINDOWS\system32\uxtheme.dll
0x71ab0000   0x17000        0x1                                 C:\WINDOWS\system32\WS2_32.dll
0x71aa0000    0x8000        0x1                                 C:\WINDOWS\system32\WS2HELP.dll
```

The output indicates that there are several DLL files that are loaded as part of the `reader_sl.exe` process. Later in this chapter, these DLL files will be acquired for further examination.

Handles plugin

The `handles` plugin allows analysts to view what type of handles are open in an existing process. These handles are references to resources that are managed by the operating system. This data provides to the responder an understanding of the specific blocks of memory an application or process is using. This includes a wide variety of information, including registry keys and files associated with that process. To identify the open handles for PID **1640** that was previously identified, the following command is used:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
-p 1640 handles
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- --------------- -------
0xe10096e0   1640        0x4    0xf0003 KeyedEvent      CritSecOutOfMemoryEvent
0xe159c978   1640        0x8        0x3 Directory       KnownDlls
0x82211678   1640        0xc   0x100020 File            \Device\HarddiskVolume1\Documents and Settings\Robert
0x82210208   1640       0x10   0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_6b128700
0xe14916d0   1640       0x14    0xf000f Directory       Windows
0xe1c6a588   1640       0x18  0x21f0001 Port
0x82319610   1640       0x1c  0x21f0003 Event
0x8205a2a0   1640       0x20    0xf037f WindowStation   WinSta0
0x822f8168   1640       0x24    0xf01ff Desktop         Default
0x8205a2a0   1640       0x28    0xf037f WindowStation   WinSta0
0x82311280   1640       0x2c   0x100003 Semaphore
0x82234dd0   1640       0x30   0x100003 Semaphore
0xe1c042d0   1640       0x34  0x20f003f Key             MACHINE
0xe16ce308   1640       0x38    0x2000f Directory       BaseNamedObjects
0x8213d0e0   1640       0x3c   0x1f0003 Semaphore       shell.{A48F1A32-A340-11D1-BC6B-00A0C90312E1}
0xe1835648   1640       0x40  0x20f003f Key             USER\S-1-5-21-789336058-261478967-1417001333-1003
0x820d2f28   1640       0x44   0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
0xe1c72300   1640       0x48   0x1f0001 Port
0xe17d3938   1640       0x4c        0x4 Section
0x81de10c8   1640       0x50   0x1f0003 Event
0x822924c8   1640       0x54   0x1f03ff Thread          TID 1648 PID 1640
0x821dd728   1640       0x58   0x1f0003 Event
0x82196418   1640       0x5c   0x1f0003 Event
0x820022e0   1640       0x60   0x1f0003 Event
0x82002a18   1640       0x64   0x1f0003 Event
0x822924c8   1640       0x68   0x1f03ff Thread          TID 1648 PID 1640
0x821dc270   1640       0x6c   0x100001 File            \Device\KsecDD
0xe1c5cfb8   1640       0x70       0x10 Key             USER\S-1-5-21-789336058-261478967-1417001333-1003\SOFTWARE\MICROSOFT\WSH\8149A9A8
0xe1c6c030   1640       0x74       0x18 Token
0x81de1e68   1640       0x78   0x1f0003 Event
0x81dd2e08   1640       0x7c   0x1f0003 IoCompletion
0x81de3c70   1640       0x80   0x1f0003 IoCompletion
0x81dd2e08   1640       0x84   0x1f0003 IoCompletion
0x822fdb00   1640       0x88   0x1f0001 Mutant          XMM00000668
0x822d0d98   1640       0x8c   0x1f0003 Event           XME00000668
0xe154db20   1640       0x90       0x10 Key             USER\S-1-5-21-789336058-261478967-1417001333-1003\SOFTWARE\MICROSOFT\WSH\9DBBCFAD
0x820fd260   1640       0x94   0x1f0003 Semaphore       shell.{210A4BA0-3AEA-1069-A2D9-08002B30309D}
0x81e9d708   1640       0x98   0x1f0001 Mutant          XMR8149A9A8
0x81e1d3c0   1640       0x9c   0x1f0003 Event
```

As the output indicates, the suspect process has several open handle processes, threads, and registry keys. These may become important data points moving forward and give some indication of the behavior of the `reader_sl.exe` executable.

LDR modules

A common practice with malware coders is attempting to hide the activities of the malware. One technique is to attempt to hide the DLL files associated with the malicious code. This can be accomplished by unlinking the suspect DLL from the **Process Environment Block** (**PEB**). While this may provide some obfuscation on the surface, there is still trace evidence of the DLL's existence contained within the **Virtual Address Descriptor** (**VAD**). The VAD is a mechanism that identifies a DLL file's base address and full path. The `ldrmodules` plugin compares the list of processes and determines if they are in the PEB. The following command runs the `ldrmodules` against the Cridex image file:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
-p 1640 ldrmodules
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Pid      Process              Base       InLoad InInit InMem MappedPath
-------- -------------------- ---------- ------ ------ ----- ----------
    1640 reader_sl.exe        0x00400000 True   False  True  \Program Files\Adobe\Reader 9.0\Reader\reader_sl.exe
    1640 reader_sl.exe        0x7c800000 True   True   True  \WINDOWS\system32\kernel32.dll
    1640 reader_sl.exe        0x773d0000 True   True   True  \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6
    1640 reader_sl.exe        0x7c420000 True   True   True  \WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b
    1640 reader_sl.exe        0x5d090000 True   True   True  \WINDOWS\system32\comctl32.dll
    1640 reader_sl.exe        0x77f60000 True   True   True  \WINDOWS\system32\shlwapi.dll
    1640 reader_sl.exe        0x77f10000 True   True   True  \WINDOWS\system32\gdi32.dll
    1640 reader_sl.exe        0x78130000 True   True   True  \WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b
    1640 reader_sl.exe        0x71aa0000 True   True   True  \WINDOWS\system32\ws2help.dll
    1640 reader_sl.exe        0x77e70000 True   True   True  \WINDOWS\system32\rpcrt4.dll
    1640 reader_sl.exe        0x71ab0000 True   True   True  \WINDOWS\system32\ws2_32.dll
    1640 reader_sl.exe        0x7c9c0000 True   True   True  \WINDOWS\system32\shell32.dll
    1640 reader_sl.exe        0x77dd0000 True   True   True  \WINDOWS\system32\advapi32.dll
    1640 reader_sl.exe        0x77fe0000 True   True   True  \WINDOWS\system32\secur32.dll
    1640 reader_sl.exe        0x7e410000 True   True   True  \WINDOWS\system32\user32.dll
    1640 reader_sl.exe        0x7c900000 True   True   True  \WINDOWS\system32\ntdll.dll
    1640 reader_sl.exe        0x77c10000 True   True   True  \WINDOWS\system32\msvcrt.dll
    1640 reader_sl.exe        0x5ad70000 True   True   True  \WINDOWS\system32\uxtheme.dll
```

A review of the output reveals an interesting entry at the top line:

```
0x00400000 True   False  True  \Program Files\Adobe\Reader 9.0\Reader\reader_sl.exe
```

From this output, the `reader_sl.exe` process does appear to have an issue associated with the DLL file. The indicator that this process is suspect is the `False` indicator in the `InInit` column for the first entry. This indicates that the executable has de-linked the DLL files and the `reader_sl.exe` file warrants further investigation.

Process xview

Another good plugin that aids in discovering hidden processes is the `psxview` plugin. This plugin compares the active process indicated within `psActiveProcessHead` with any other possible sources within the memory image. To run the plugin, type the following command:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
psxview
```

The command produces the following:

```
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
0x02498700 winlogon.exe            608 True   True   True     True   True  True    True
0x02511360 svchost.exe             824 True   True   True     True   True  True    True
0x022e8da0 alg.exe                 788 True   True   True     True   True  True    True
0x020b17b8 spoolsv.exe            1512 True   True   True     True   True  True    True
0x0202ab28 services.exe            652 True   True   True     True   True  True    True
0x02495650 svchost.exe            1220 True   True   True     True   True  True    True
0x0207bda0 reader_sl.exe          1640 True   True   True     True   True  True    True
0x025001d0 svchost.exe            1004 True   True   True     True   True  True    True
0x02029ab8 svchost.exe             908 True   True   True     True   True  True    True
0x023fcda0 wuauclt.exe            1136 True   True   True     True   True  True    True
0x0225bda0 wuauclt.exe            1588 True   True   True     True   True  True    True
0x0202a3b8 lsass.exe               664 True   True   True     True   True  True    True
0x023dea70 explorer.exe           1484 True   True   True     True   True  True    True
0x023dfda0 svchost.exe            1056 True   True   True     True   True  True    True
0x024f1020 smss.exe                368 True   True   True     True   False False   False
0x025c89c8 System                    4 True   True   True     True   False False   False
0x024a0598 csrss.exe               584 True   True   True     True   False True    True
```

A `False` within the column indicates that the process is not found in that area. This allows the analyst to review that list and determine whether there is a legitimate reason that the process may not be there, or if it is indicative of an attempt to hide the process.

Volatility network analysis

In the *Network connection methodology* section, there was a discussion regarding beginning the process of analysis with a URL or IP address associated with malicious activity. Volatility has the ability to pull out of the memory image existing and even exited network connections that were resident at the time of acquisition.

The `netscan` plugin scans the memory image for network artifacts. The plugin will find TCP and UDP endpoints and listeners as well as provide the local and foreign IP addresses. `netscan` will only work with 32-bit and 64-bit Windows Vista, Windows 7, Windows 10, and Windows 2008 Server or newer. One key feature that is of help to incident response analysts with the `netscan` plugin is that, for the network connections, the process owner is indicated in the output. This is useful in determining whether a connection is utilizing Internet Explorer or another process, such as Remote Desktop Services or SMB.

connscan

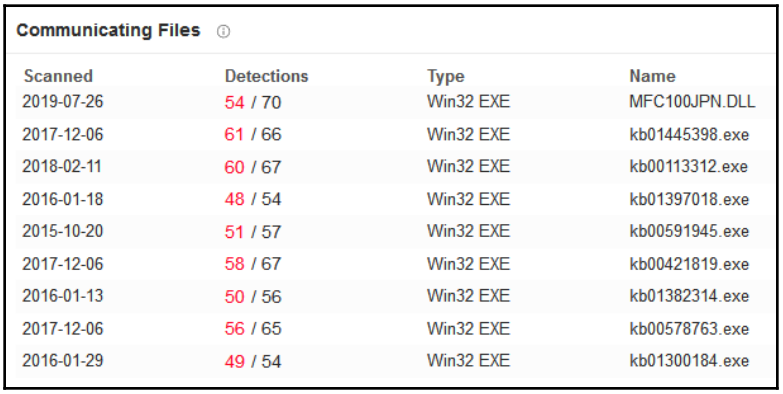
For earlier versions of Windows, such as Windows XP and earlier, the `connscan` plugin performs the same function as the `netscan` plugin. The `connscan` plugin finds the `_TCPT_OBJECT` and is able to find both existing and exited connections. This provides responders with data concerning connections in relation to processes that were running. To determine the network connections, run the following command against the Cridex image:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
connscan
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x02087620 172.16.112.128:1038      41.168.5.140:8080         1484
0x023a8008 172.16.112.128:1037      125.19.103.198:8080       1484
```

The output indicates that the PID of `1484`, which is `Explorer.exe`, the parent process of `Reader_sl.exe` was communicating with two external IP addresses. A review of the threat intelligence resource VirusTotal indicated that the IP address `41.168.5.140` was associated with several URLs that were communicating with malicious executables. The following image shows the various malicious files, their detections, and the filenames associated with the IP address `41.168.5.140`:



Communicating Files

Scanned	Detections	Type	Name
2019-07-26	54 / 70	Win32 EXE	MFC100JPN.DLL
2017-12-06	61 / 66	Win32 EXE	kb01445398.exe
2018-02-11	60 / 67	Win32 EXE	kb00113312.exe
2016-01-18	48 / 54	Win32 EXE	kb01397018.exe
2015-10-20	51 / 57	Win32 EXE	kb00591945.exe
2017-12-06	58 / 67	Win32 EXE	kb00421819.exe
2016-01-13	50 / 56	Win32 EXE	kb01382314.exe
2017-12-06	56 / 65	Win32 EXE	kb00578763.exe
2016-01-29	49 / 54	Win32 EXE	kb01300184.exe

Taking the data that was derived from the process analysis in conjunction with the IP address taken from the network connections, there is enough reason to believe one, or both, of the `Explorer.exe` and `Reader_sl.exe` processes are associated with malicious code. In the next section, `Reader_sl.exe` will be extracted along with its associated DLL files for analysis.

Volatility evidence extraction

As was stated previously, one of the central goals of memory analysis is to determine whether there are any suspicious data points indicative of malware. In the event data points such as those from the Cridex memory image are located, they can be acquired for further analysis.

Memory dump

During the course of the analysis, it may become necessary to dump the memory-resident pages associated with a process. In this case, the `memdump` plugin is run against the memory image, with the output directed to a folder, via the following command:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem -profile=WinXPSP2x86
-p 1640 memdump --dump-dir /mnt/d/Suspicious_Process_PID_1640
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing reader_sl.exe [  1640] to 1640.dmp
```

The DMP file will be written to the location selected.

It is a good practice to develop a naming convention for folders associated with memory analysis. This way, files are kept in the appropriate location. In this case, the author is using the following filename for the acquisition: `Suspect_Process_PID_1640`. It should be noted that the acquisition may contain malware and should be done on an appropriate system.

DLL file dump

In the event that an analyst is able to identify a suspect process within the memory image, the `dlldump` plugin can be utilized to dump the contents of those DLL files to the local system. This allows the analysts to examine the contents of the DLL files and compare them to legitimate files to determine whether they are malicious. For example, the process that has been identified, `Reader_sl.exe` with the PID of `1640`, was identified as potentially malicious in several sections of this chapter. To acquire the DLL files and have them accessible to the local system, type the following:

```
dfir@Desktop-SFARF6G~$ volatility -f cridex_laptop.mem --
profile=WinXPSP2x86 -p 1640 dlldump --dump-dir
/mnt/d/Suspicious_Process_PID_1640/
```

The command produces the following:

```
Volatility Foundation Volatility Framework 2.6
Process(V) Name                 Module Base Module Name          Result
---------- -------------------- ----------- -------------------- ------
0x81e7bda0 reader_sl.exe        0x000400000 Reader_sl.exe        OK: module.1640.207bda0.400000.dll
0x81e7bda0 reader_sl.exe        0x07c900000 ntdll.dll            OK: module.1640.207bda0.7c900000.dll
0x81e7bda0 reader_sl.exe        0x078130000 MSVCR80.dll          OK: module.1640.207bda0.78130000.dll
0x81e7bda0 reader_sl.exe        0x07c420000 MSVCP80.dll          OK: module.1640.207bda0.7c420000.dll
0x81e7bda0 reader_sl.exe        0x077f10000 GDI32.dll            OK: module.1640.207bda0.77f10000.dll
0x81e7bda0 reader_sl.exe        0x077f60000 SHLWAPI.dll          OK: module.1640.207bda0.77f60000.dll
0x81e7bda0 reader_sl.exe        0x05ad70000 uxtheme.dll          OK: module.1640.207bda0.5ad70000.dll
0x81e7bda0 reader_sl.exe        0x077e70000 RPCRT4.dll           OK: module.1640.207bda0.77e70000.dll
0x81e7bda0 reader_sl.exe        0x05d090000 comctl32.dll         OK: module.1640.207bda0.5d090000.dll
0x81e7bda0 reader_sl.exe        0x071aa0000 WS2HELP.dll          OK: module.1640.207bda0.71aa0000.dll
0x81e7bda0 reader_sl.exe        0x071ab0000 WS2_32.dll           OK: module.1640.207bda0.71ab0000.dll
0x81e7bda0 reader_sl.exe        0x077c10000 msvcrt.dll           OK: module.1640.207bda0.77c10000.dll
0x81e7bda0 reader_sl.exe        0x07c9c0000 SHELL32.dll          OK: module.1640.207bda0.7c9c0000.dll
0x81e7bda0 reader_sl.exe        0x0773d0000 comctl32.dll         OK: module.1640.207bda0.773d0000.dll
0x81e7bda0 reader_sl.exe        0x077fe0000 Secur32.dll          OK: module.1640.207bda0.77fe0000.dll
0x81e7bda0 reader_sl.exe        0x07c800000 kernel32.dll         OK: module.1640.207bda0.7c800000.dll
0x81e7bda0 reader_sl.exe        0x07e410000 USER32.dll           OK: module.1640.207bda0.7e410000.dll
0x81e7bda0 reader_sl.exe        0x077dd0000 ADVAPI32.dll         OK: module.1640.207bda0.77dd0000.dll
```

Next, we will be looking at the executable dump.

Executable dump

A review of the results from a variety of sources has indicated that the process 1640 and the associated executable `Reader_sl.exe` are suspected of containing malware. While the data thus far is very useful, it is often necessary to obtain confirmation from external sources that the executable in question is malicious. This can include something as simple as checking the hash of the executable against third-party sources all the way to forwarding the executable to a malware reverse engineering team.

To acquire the executable from the memory image, utilize the `procdump` plugin. The following command will dump the executable to the selected folder:

```
volatility -f cridex_laptop.mem --profile=WinXPSP2x86 -p 1640 procdump --
dump-dir /mnt/d/Suspicious_Process_PID_1640/
```

The command produces the following output:

```
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x81e7bda0 0x00400000 reader_sl.exe        OK: executable.1640.exe
```

A check of the folder where the files were dumped to reveals that the executable, memory space, and the DLL files have all been successfully acquired:

1640.dmp	9/8/2019 2:43 PM	DMP File	75,396 KB
executable.1640	9/8/2019 2:48 PM	Application	29 KB
module.1640.207bda0.5ad70000.dll	9/8/2019 2:46 PM	Application exten...	214 KB
module.1640.207bda0.5d090000.dll	9/8/2019 2:46 PM	Application exten...	603 KB
module.1640.207bda0.7c9c0000.dll	9/8/2019 2:46 PM	Application exten...	8,263 KB
module.1640.207bda0.7c420000.dll	9/8/2019 2:46 PM	Application exten...	536 KB
module.1640.207bda0.7c800000.dll	9/8/2019 2:46 PM	Application exten...	967 KB
module.1640.207bda0.7c900000.dll	9/8/2019 2:46 PM	Application exten...	690 KB
module.1640.207bda0.7e410000.dll	9/8/2019 2:46 PM	Application exten...	565 KB
module.1640.207bda0.71aa0000.dll	9/8/2019 2:46 PM	Application exten...	20 KB
module.1640.207bda0.71ab0000.dll	9/8/2019 2:46 PM	Application exten...	81 KB
module.1640.207bda0.77c10000.dll	9/8/2019 2:46 PM	Application exten...	335 KB
module.1640.207bda0.77dd0000.dll	9/8/2019 2:46 PM	Application exten...	603 KB
module.1640.207bda0.77e70000.dll	9/8/2019 2:46 PM	Application exten...	571 KB
module.1640.207bda0.77f10000.dll	9/8/2019 2:46 PM	Application exten...	279 KB
module.1640.207bda0.77f60000.dll	9/8/2019 2:46 PM	Application exten...	463 KB
module.1640.207bda0.77fe0000.dll	9/8/2019 2:46 PM	Application exten...	55 KB
module.1640.207bda0.773d0000.dll	9/8/2019 2:46 PM	Application exten...	1,030 KB
module.1640.207bda0.400000.dll	9/8/2019 2:46 PM	Application exten...	29 KB
module.1640.207bda0.78130000.dll	9/8/2019 2:46 PM	Application exten...	612 KB

Once the files have been acquired, they can then be analyzed for malware, either by the incident response team or through a separate malware analysis team. These files will make up a significant portion of the analysis in `Chapter 12`, *Malware Analysis for Incident Response*.

Memory analysis with strings

In the previous sections, the Redline and Volatility tools focused on those areas of the memory image that are mapped. In the event that data is not properly mapped, these tools would be unable to extract the data and present it properly. This is one of the drawbacks of these tools for memory analysis. There is a good deal of data that will become unstructured and invisible to these tools. This could be the case when network connections are shut down or processes exited. Even though they may not show up when the RAM is examined via Redline or Volatility, trace evidence will often still be present.

One tool that is useful for extracting these traces is the Strings command present in many of the Linux and Windows OSes. Strings allows a responder to search for human-readable strings of characters. Given a set of keywords or **GREP** (short for **Global Regular Expression Print**) commands, the responder may be able to extract additional relative data, even from RAM captures that may have been corrupted via malware or improper acquisitions.

Installing Strings

Strings will often come preinstalled in many Linux distributions. Windows has a standalone executable for string searches available at `https://docs.microsoft.com/en-us/sysinternals/downloads/strings`. If Strings is not installed on the Linux platform of choice for the responder, the following command will install it:

```
dfir@Desktop-SFARF6G~$ sudo apt install binutils
```

For a rather simple tool, Strings is a powerful way to search through bulk data for specific keyword-based strings. For the purposes of this book, the focus will be on extracting specific data points with the following Strings syntax:

```
dfir@Desktop-SFARF6G~$ strings cridex_laptop.mem | grep <Regular
Expression>
```

In the preceding command, Strings has been set to search the `cridex_laptop.mem` memory image for the regular expression following the grep command. In the regular expression portion, this can be any regular expression, including IP addresses, URLs, commands, and potentially scripts that have been run on the system.

IP address search

In the previous section on Volatility, the IP address `41.168.5.140` was identified by using the `connscan` plugin. The drawback of that process of identifying IP addresses is that, if the connection has been closed and there is no activity, it may not be visible with Volatility. In that case, a way to expand the search for IP addresses resident in memory is to conduct the following Strings search:

```
strings cridex_laptop.mem | grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b"
```

This Strings search will look for any matching patters of an IP address. When examining the Cridex memory capture, several IP addresses show up. This includes internal IP address ranges and broadcast IP addresses. An examination of the results revealed that the IP address `188.40.0.138` was at one time located within memory:

```
192.168.10.102
192.168.10.102
188.40.0.138
1.7.1.1
3.14.3.2
```

From here, a second search for the IP address can be performed using the following command:

```
strings cridex_laptop.mem | grep 188.40.0.138
```

The command produces the following output:

```
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
```

Next, we will look at HTTP search.

HTTP Search

Often, adversaries will use an IP address as part of the URL that is used as a delivery mechanism. The previous screenshot indicates that the adversary may be using an IP address as part of the URL. The next command will search the memory image for any HTTP entries in memory:

```
strings cridex_laptop.mem | grep "http://"
```

That command produces the following output:

```
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
<!-- BEGIN Global Navigation table --><table cellspacing="0" cellpaddin
://chaseonline.chase.com/images//ChaseNew.gif" alt="Chase Online Logo"
//www.chase.com/';" class="globalnavlinks">Chase.com</a>  </td>
<!--Footer--><table border="0" cellspacing="0" cellpadding="0" class="f
cal-align:top"> </td><td align="center" width="40%" valign="top"><span
/shared/assets/page/security_measures';" onBlur="window.status='';retur
Security</a> | <!-- mp_trans_remove_start --><a id="TermsLink" href="Ja
us='';return true" onMouseOver="window.status='';return true" onFocus="
dd<a id="TermsLink" href="JavaScript:document.location.href='https://ww
us='';return true" onFocus="window.status='';return true" onMouseOut="w
></table><div class="printable"><table border="0" cellspacing="0" cellp
```

Examining the output shows some interesting data. First, there is the URL `http://chaseonline.com/`. In addition, there appears to be website coding associated with this hit. A search of various sites for data on Cridex reveals that it is often used to steal banking credentials by hooking various APIs in the web browser and redirecting that traffic for various sites, including Chase Bank. (Microsoft has a write-up on Cridex available at `https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?name=win32%2Fcridex`.)

This short example shows how Strings can be utilized by responders to gather more evidence and provide additional context to an incident investigation. Strings is only limited to what keywords can be used.

Summary

This chapter discussed two major topic areas of memory analysis. First is the data points available and the methodology that can be followed. In addition, several tools, such as Redline, Volatility, and Strings have been explored. In addition to an overview of these tools, several of their features have been explored. This only scratches the surface of the number of features each of these tools has to offer the incident response analyst. These tools, taken in conjunction with a methodology for analyzing system RAM, can give the analyst a powerful tool for determining if a system has been compromised. With malware becoming more advanced, including malware that executes entirely in RAM, it is critical that analysts incorporate memory analysis into their capability. Marrying these techniques with network evidence collection can provide analysts and their organizations with a powerful tool to identify and remediate an incident.

In the next chapter, the analysis will move from the volatile evidence captured to examining the system's hard drive. Here, an analyst can gain more insight into the events surrounding an incident.

Questions

1. What are some of the data points that can be found via memory analysis?

   A) Running processes
   B) Network connection
   C) Command history
   D) All of the above

2. What is not part of the network connections methodology?

   A) Process name
   B) Parent process ID
   C) Check for signs of a Rootkit
   D) Associated entities

3. Dumping files associated with a process will never introduce malware to a responder's system.

   A) True
   B) False

4. One of the primary goals of memory analysis is to acquire malicious processes or executables for further analysis.

   A) True
   B) False

Further reading

- *SANS Memory Forensics Cheat Sheet*: `https://digital-forensics.sans.org/blog/2017/12/11/updated-memory-forensics-cheat-sheet`
- *Redline User Guide*: `https://www.fireeye.com/content/dam/fireeye-www/services/freeware/ug-redline.pdf`

9
Analyzing System Storage

So far, the evidence that has been analyzed has focused on those elements that are obtained from the network or the system's memory. Even though incident root cause may be ferreted out from these evidence sources, it is important to understand how to obtain evidentiary material from a system's storage, whether that is removable storage such as USB devices or the larger connected disk drives. In these containers is a massive amount of data that may be leveraged by incident response analysts to determine a root cause. It should be noted that this chapter will only be able to scratch the surface as entire volumes have been devoted to the depth of forensic evidence that's available.

To provide a better understanding of analyzing system storage, this chapter will focus on the following topics:

- **Forensic platforms**: There are a variety of commercial and open source platforms that we can use to conduct system storage analysis. This section will address the key features and potential options we have.
- **Autopsy**: To provide you with an open source platform that can be leveraged in system storage analysis, the majority of this chapter will use the Autopsy tool. Some of its features will be highlighted by utilizing a test image.
- **Master File Table (MFT) analysis**: Containing a comprehensive list of all the files on the system, the MFT is a key source of data for responders. This section addresses the extraction and analysis of the Master File Table.
- **Registry analysis**: A favorite target of malware coders and other exploits, responders should become familiar with registry analysis. An overview of the extraction and analysis of the registry will be addressed in this section.

System storage analysis is a complex process. The depth and breadth of it cannot be explored in a single chapter; due to this, we hope that this chapter provides some concrete areas of focus with the understanding that responders will gain a better sense of some of the tools that can be employed, as well as an understanding of some of the critical data that can be leveraged.

Forensic platforms

Over the past 15 years, there has been an increase in the power of disk forensic platforms. For the incident response analyst, there are options as to what type of platform can be leveraged for conducting an examination of the disk drives. Often, the limiting factor in utilizing these platforms is the cost of more robust systems, when a lower-cost alternative will be just as effective for an incident response team.

There are several factors that should be addressed when examining software for disk analysis. First, has the platform been tested? There are several organizations that test platforms for efficacy, such as the National Institute of Standards and Technology Computer Forensic Tools Testing Program (`https://www.cftt.nist.gov/`). Second is an examination of the tool's use in criminal and civil proceedings. There is no single court-accepted standard, but tools should conform to the rules of evidence. The use of a platform that has not been tested or does not conform to the rules of evidence may lead to the evidence being excluded from legal proceedings. In other, more disastrous consequences, it may lead to an analyst arriving at the wrong conclusion.

An example of an untested and forensically unsound toolset that was used in a criminal proceeding was in the case of The State of Connecticut versus Amero. In this case, a law enforcement agency utilized unsound forensic methods and tools to convict a woman for allegedly allowing children to see sexually explicit pop-up ads. A subsequent review of the methods and facts of the case indicated that there were significant deficiencies with the forensic examination. An excellent examination of this case is available from the Journal of Digital Forensics, Security, and Law at `https://commons.erau.edu/cgi/viewcontent.cgi?article=1120 context=jdfsl`.

One final consideration is how the tool fits into the overall incident response planning. For example, commercial disk forensic tools are excellent at locating images and web artifacts. They are also excellent at carving out data from the suspect drive. This is often due to the fact that forensic software is utilized by law enforcement agencies as a tool to investigate child exploitation crimes. As a result, this capability is paramount to bringing a criminal case against such suspects. While these are excellent capabilities to have, incident responders may be more interested in tools that can be utilized for keyword searches and timeline analysis so that they can reconstruct a series of events prior to, during, and after an incident.

While most commercial and free forensic platforms have a variety of features, there are several common ones that can be of use to incident response personnel:

- **File structure view**: It is often very important to be able to view the file structure of the disk under examination. Forensic platforms should have the ability to view the file structure and allow responders to quickly review files with known locations on a suspect system.
- **Hex viewer**: Having the ability to view files in hexadecimal allows responders to have a granular look at the files under examination. This may be beneficial in cases involving malware or other custom exploits.
- **Web artifacts**: With a great deal of data stored on the drive associated with web searching, forensic platforms should have the ability to examine these pieces of data. This is very handy when examining social engineering attacks where users navigate to a malicious website.
- **Email carving**: Incident responders may be called into cases where malicious employees are involved in illegal activities or have committed policy violations. Often, evidence of this type of conduct is contained within emails on the suspect system. Having a platform that can pull this data out for immediate view assists the analyst in viewing communication between the suspect system and others.
- **Image viewer**: Often, it is necessary to view the images that are saved on systems. As we mentioned previously, law enforcement may utilize this feature to determine whether there is evidence of child exploitation on a system. Incident responders can utilize these features to determine whether there has been a policy violation.
- **Metadata**: Key pieces of data about files such as date and time created, file hashes, and the location of a suspect file on the disk are useful when examining a system associated with an incident. For example, the time an application is run, taken in conjunction with a piece of malware, may be correlated with network activity, allowing the analyst to determine the actual executable run.

In terms of commercial options, the following three platforms are generally accepted as sound and are in use by commercial and government entities all over the world. Each uses the features we described previously, among other, more specialized, tools:

- **OpenText EnCase**: Arguably the preeminent forensics platform, EnCase has a long history as being the platform that's used in major criminal investigations, such as the BTK Killer. EnCase is a feature-rich platform that makes it a powerful tool in the hands of a trained analyst. In addition to disk forensics, EnCase also has integrated features for mobile devices. This is a powerful capability for organizations that may have to analyze not only disks, but also mobile devices, in connection with an incident.

- **AccessData Forensic Toolkit**: In `Chapter 6`, *Forensic Imaging*, the FTK Imager tool was utilized to acquire disk and memory evidence. This tool is part of a suite of tools provided by Access Data that have been specifically tailored for disk forensics. In addition to the imager, Access Data has a full-featured forensic platform that allows responders to perform a range of tasks associated with an incident. FTK is in use by law enforcement agencies such as the Federal Bureau of Investigation and has proven to be more than effective in assisting responders with incident investigations.
- **X-Ways Forensics**: One drawback of FTK and EnCase is cost. These platforms can cost several thousands of dollars per year. For larger organizations, such as government agencies and large enterprises, the trade-off of cost versus features may not be an issue. For smaller organizations, these platforms may be cost-prohibitive. An alternative, feature-rich forensic platform is X-Ways. This platform has the ability to perform a variety of tasks but at a fraction of the cost. Another great benefit of X-Ways is that it is less resource-intensive and can be run off a USB device, making it an alternative platform, especially for incident response.

Each of these platforms has a rich feature set and provides responders with a powerful tool for conducting a wide range of forensic tasks. The specific tools in each of these platforms are outside the scope of this book. As such, it is recommended that responders are trained on how to use these platforms to ensure that they fully understand these tools' capabilities.

Autopsy

One alternative to the commercial forensics programs is Autopsy. Autopsy is a GUI-based forensic platform based upon the open source SleuthKit toolset. This open source platform has features that are commonly found in commercial platforms. This includes timeline analysis, keyword searching, web and email artifacts, and the ability to filter results on known bad file hashes. One of its key features is its ease of use. This allows incident responders to have a light platform that focuses on critical tasks and obtain the critical evidence that's needed.

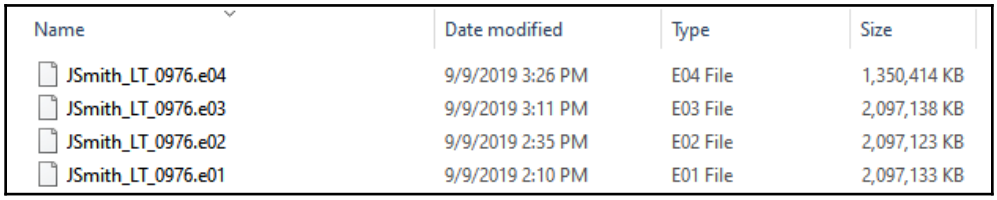
Installing Autopsy

Several of the Linux distributions we discussed previously have Autopsy preinstalled. It is good practice for responders to ensure that the platform they are using is up to date. For the Windows operating system, download the Microsoft self-installer file located at `https://www.sleuthkit.org/autopsy/download.php`. Once downloaded, execute the MSI file and choose an install location. Once you've done this, the application will be ready to use.

Opening a case

Once Autopsy has been installed, the analyst can open a case with very little preconfiguration. The following steps will discuss the process of opening a new case:

1. To begin an analysis, ensure that the entire disk image is located in a single directory. This allows the entire image to be utilized during the analysis:



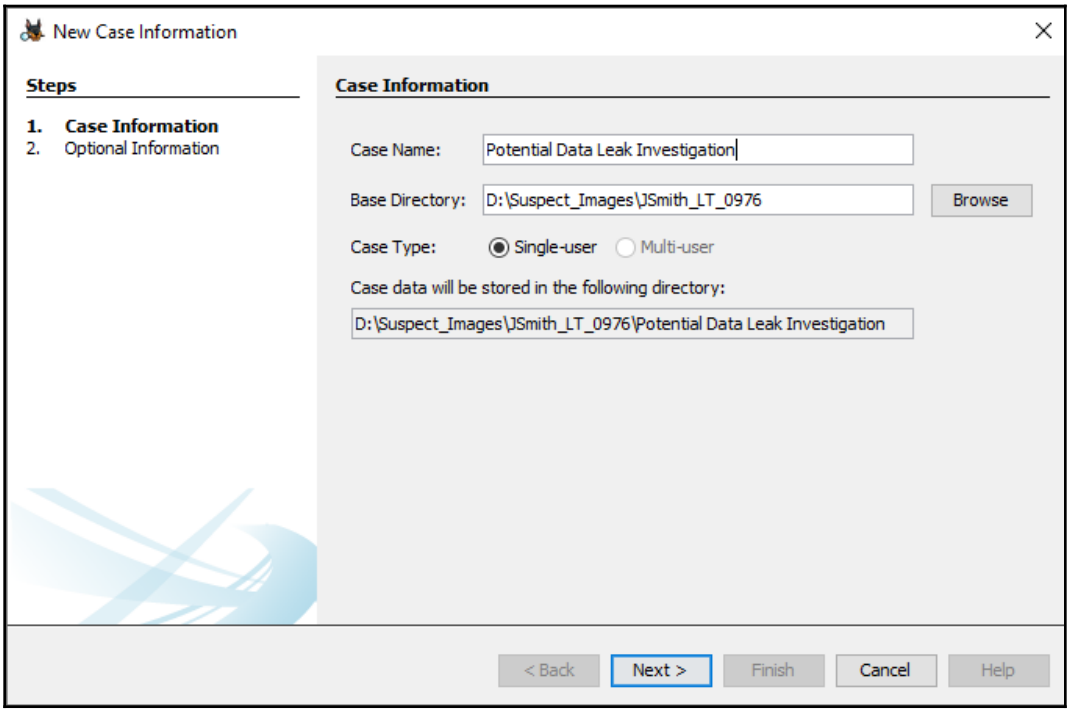
Name	Date modified	Type	Size
JSmith_LT_0976.e04	9/9/2019 3:26 PM	E04 File	1,350,414 KB
JSmith_LT_0976.e03	9/9/2019 3:11 PM	E03 File	2,097,138 KB
JSmith_LT_0976.e02	9/9/2019 2:35 PM	E02 File	2,097,123 KB
JSmith_LT_0976.e01	9/9/2019 2:10 PM	E01 File	2,097,133 KB

In the preceding screenshot, an image file has been taken from a suspect system. The image has been divided into four 2 GB files. Autopsy will be able to take the four files and reconstruct the entire volume that has been imaged.

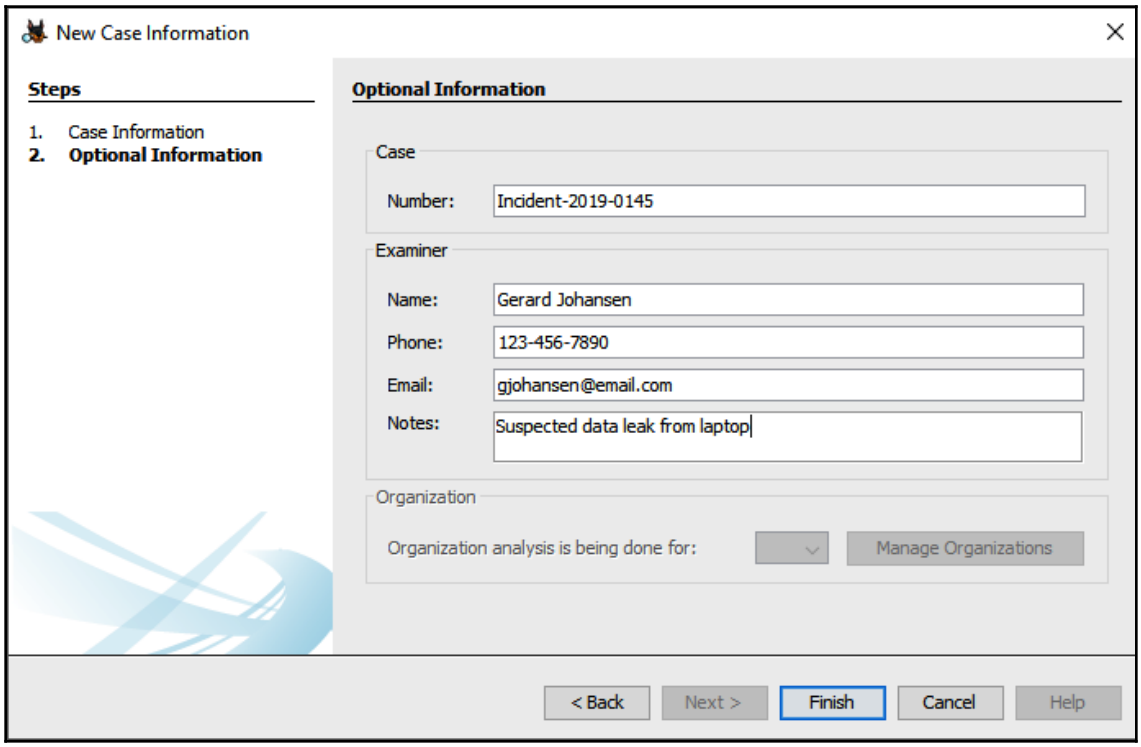
In this chapter, a memory image from the Computer Forensic Reference Data Sets is being utilized. The entire memory image can be downloaded from `https://www.cfreds.nist.gov/data_leakage_case/data-leakage-case.html` (the EnCase image).

2. Open Autopsy. The following window will appear. Choose **New Case**:

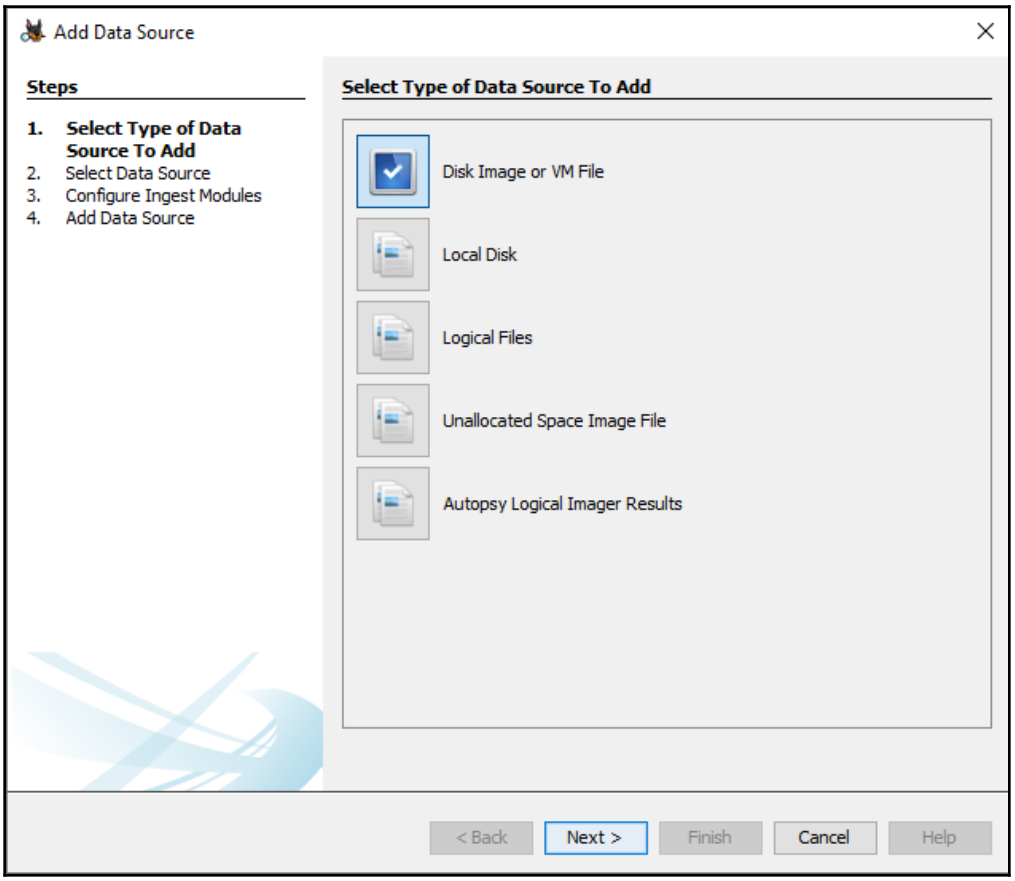
3. A second window will appear where the analyst will input the case title. In addition, the path to Autopsy that will store the files associated with the case can also be set. This is useful when circumstances dictate that the analyst has to place the files in a secure container. Once done, click **Next**:



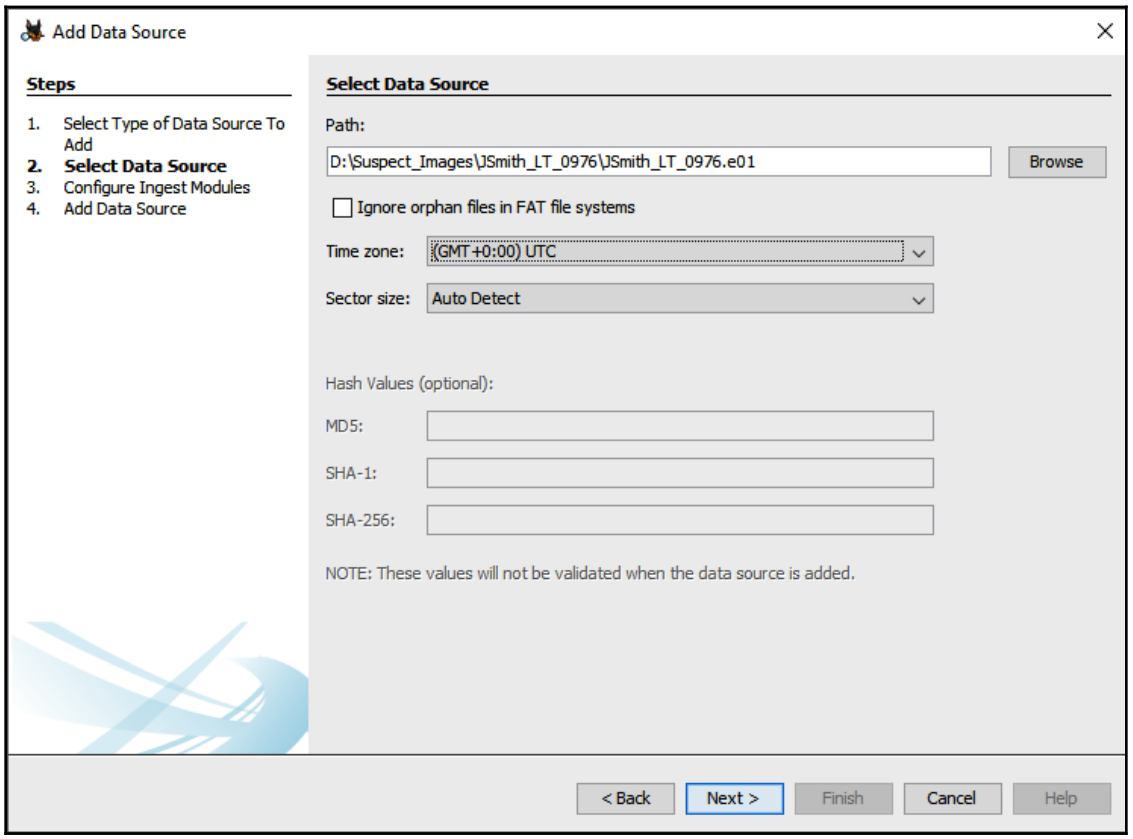
4. On the next window, the responder should input the case number, their name, their contact information, and a brief description of the case in **Notes**. Click **Finish**:



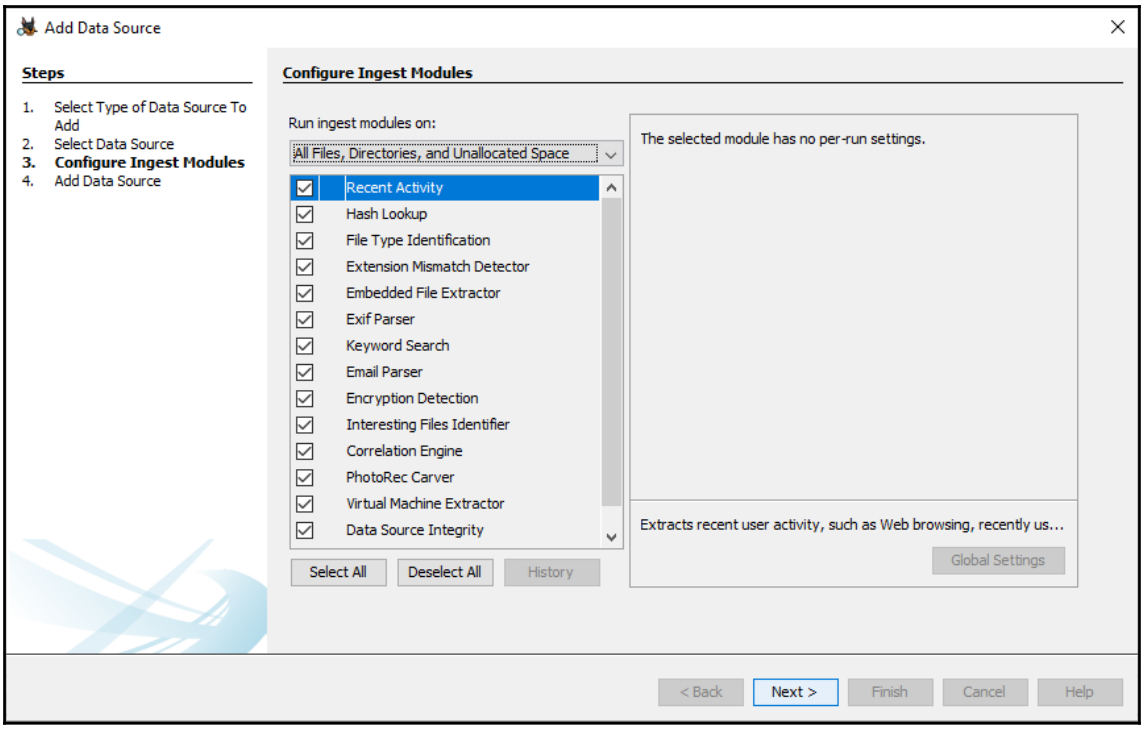
5. Once the case details have been entered, the analyst will need to load the image file that was created previously. Select the appropriate data source type. In this case, the examination will be conducted against an image file that was forensically acquired. Autopsy can also conduct an examination against a `.vmdk` file. This is a handy feature in environments where virtualization is utilized for systems. This feature allows the analyst to conduct an examination against a VM file, without having to acquire it via tools such as FTK Imager:



6. Once the file type has been selected, browse to the image location. This folder contains a number of image files; select the file that ends in `.E01`. Loading this file will include all the subsequent image files located in that folder. Next, select the appropriate time zone. Once done, click **Next**:



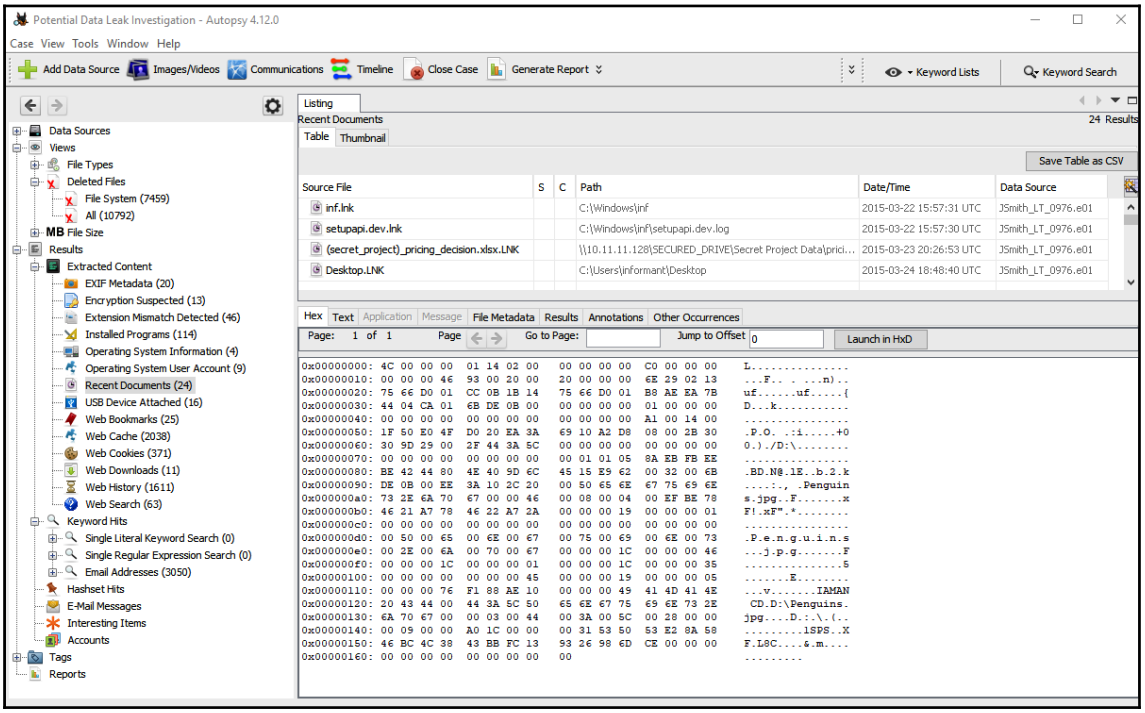
7. The next screen allows the analyst to tailor the modules in use. Depending on the type of investigation, some of these options can go unchecked. At the beginning, though, the analyst should select all of them to ensure that all the necessary information is available for examination.
   One other option is to process unallocated space (this is important!). This captures all the information in the space that's not currently allocated on the hard drive for data. There are methods where unallocated space can be utilized to hide information. Once done, click **Next**:



8. On the next screen, verify that the data source has been loaded and click **Finish**.
9. Autopsy will now go through the process of analyzing the files from the image. Depending on the size of the image, this will take between several minutes and a couple of hours. The process bar in the lower-right corner of the screen will show its progress. How long this process takes is often dependent on the processing speed of the computer, as well as the size of the image file(s).

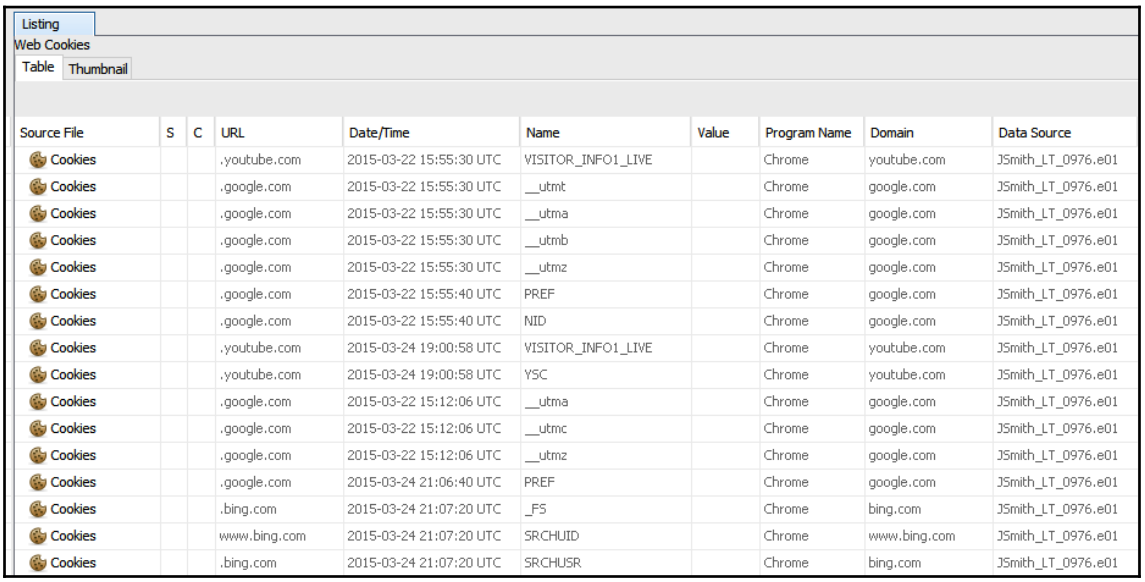
Navigating Autopsy

The Autopsy GUI is divided into three main sections. These sections display details relating to the system and specific files. When Autopsy has completed processing a new case or opening an existing case, the analyst will see the following window:



As shown in the previous screenshot, Autopsy is divided into three main panes. The first of these is the left-hand pane, which contains the data sources and file structure, as well as search results. Clicking on the plus (+) sign expands the results, while clicking on the minus (-) sign collapses them. This allows the analyst to access the system at a high level, and also to drill down to specific elements.

The center pane contains directory listings or results from searches. For example, the following screenshot shows web cookies that were located on the system:



Listing
Web Cookies
Table | Thumbnail

Source File	S	C	URL	Date/Time	Name	Value	Program Name	Domain	Data Source
Cookies			.youtube.com	2015-03-22 15:55:30 UTC	VISITOR_INFO1_LIVE		Chrome	youtube.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utmt		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utma		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utmb		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utmz		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:40 UTC	PREF		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:40 UTC	NID		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.youtube.com	2015-03-24 19:00:58 UTC	VISITOR_INFO1_LIVE		Chrome	youtube.com	JSmith_LT_0976.e01
Cookies			.youtube.com	2015-03-24 19:00:58 UTC	YSC		Chrome	youtube.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:12:06 UTC	__utma		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:12:06 UTC	__utmc		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:12:06 UTC	__utmz		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-24 21:06:40 UTC	PREF		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.bing.com	2015-03-24 21:07:20 UTC	_FS		Chrome	bing.com	JSmith_LT_0976.e01
Cookies			www.bing.com	2015-03-24 21:07:20 UTC	SRCHUID		Chrome	www.bing.com	JSmith_LT_0976.e01
Cookies			.bing.com	2015-03-24 21:07:20 UTC	SRCHUSR		Chrome	bing.com	JSmith_LT_0976.e01

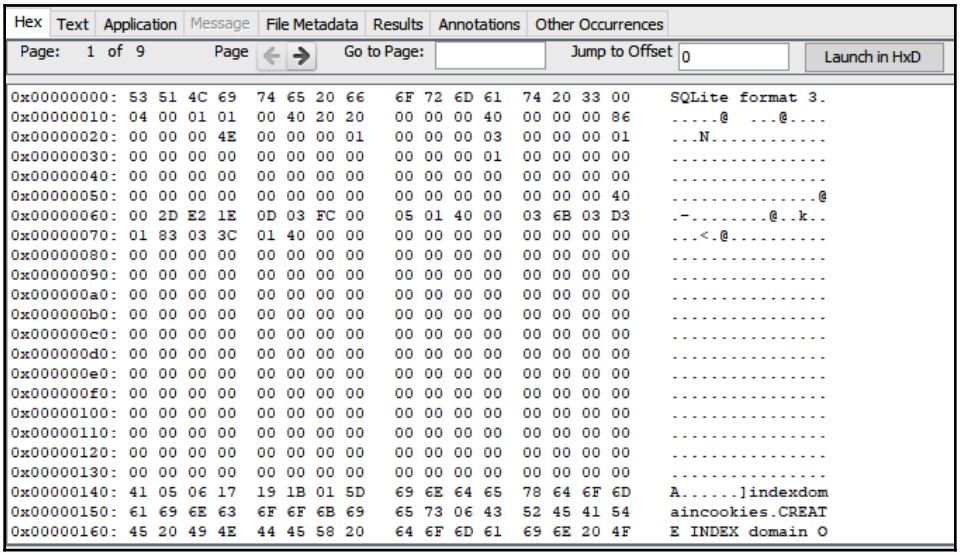
Finally, the bottom pane contains the metadata and other information about individual files contained in the center pane. For example, if the `.youtube.com` cookie is selected, the following data appears when the **Results** tab is selected:

Type	Value
URL	.youtube.com
Date/Time	2015-03-24 19:00:58
Name	YSC
Value	
Program Name	Chrome
Domain	youtube.com
Source File Path	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppData/Local/Google/Chrome/User Data/Default/Cookies
Artifact ID	-9223372036854775654

Clicking the **File Metadata** tab will produce information specific to that file. This includes the timestamps for the file, as well as an MD5 hash:

Name	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppData/Local/Google/Chrome/User Data/Default/Cookies
Type	File System
MIME Type	application/x-sqlite3
Size	137216
File Name Allocation	Allocated
Metadata Allocation	Allocated
Modified	2015-03-24 21:07:21 UTC
Accessed	2015-03-22 15:11:57 UTC
Created	2015-03-22 15:11:57 UTC
Changed	2015-03-24 21:07:21 UTC
MD5	7a247be5ff943b90262c755dfdefeca7
Hash Lookup Results	UNKNOWN
Internal ID	9952

Finally, the file's hexadecimal content can be viewed by clicking on the **Hex** tab:

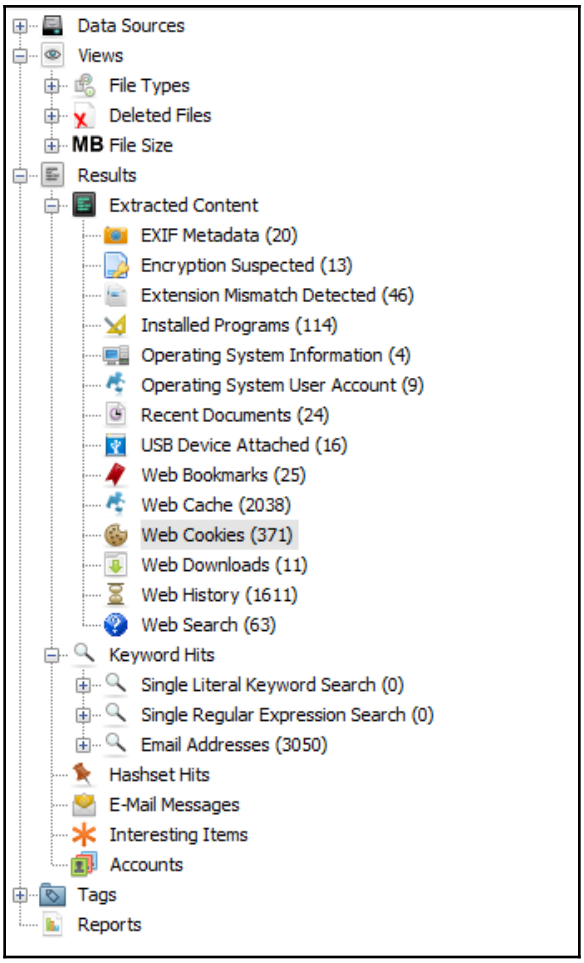


This view is excellent if an analyst wants to inspect an application or another file that is suspected of being malware.

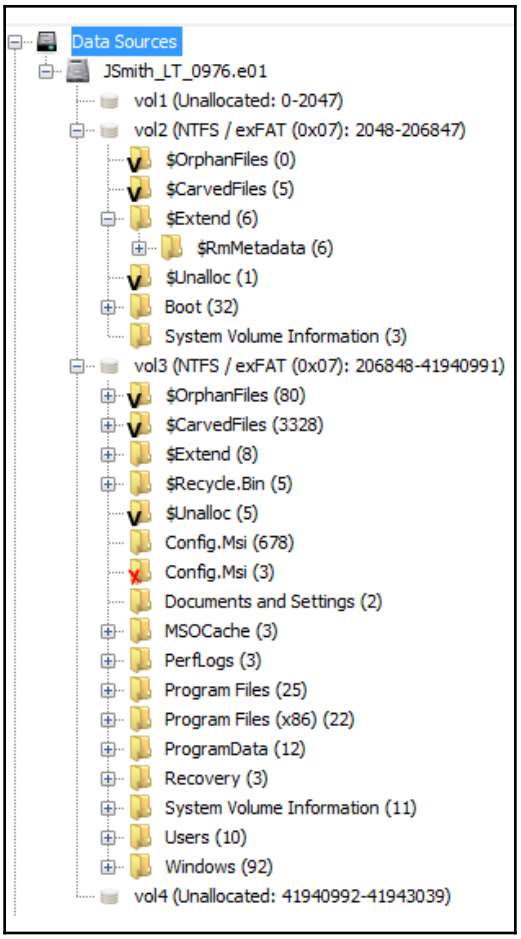
What Autopsy offers is the ability to perform some of the actions and analysis that can be found in other commercial platforms. However, it should be noted that in the case of more complex investigations, it may become necessary to utilize more sophisticated platforms. Autopsy also provides responders that are new to disk forensics withe a more user-friendly platform so that they can gain experience with one before they move on to a more sophisticated commercial solution.

Examining a case

Once the case has been processed, the left-hand pane will be populated with the number of artifacts located on the system:



In the previous screenshot, there are several items listed under the **Extracted Content** portion. These include looking at programs that have been installed, the operating system's information, and recent documents. Another key feature of Autopsy is the ability to examine the entire folder structure of the image file. Clicking on the plus (+) sign next to **Data Sources** expands the entire folder structure. This is useful if, through other sources, an analyst is able to identify the location of a suspect file:



There are different data points that can be examined by utilizing Autopsy. What to search for and how to search for it is often dictated by the type of incident or examination under investigation. For example, a malware infection that originates from a compromised website may involve examining the system for URLs that the user may have typed in or otherwise accessed via a browser. Furthermore, the actual file may be located by utilizing information that's been obtained by examining the system memory, which we covered in the previous chapter. For example, if an analyst was able to locate a suspect process via Volatility or Redline and was subsequently able to also locate the executable, they may utilize Autopsy to find the last time the executable was launched. This can provide responders with a time so that they can examine other systems for evidence of compromise.

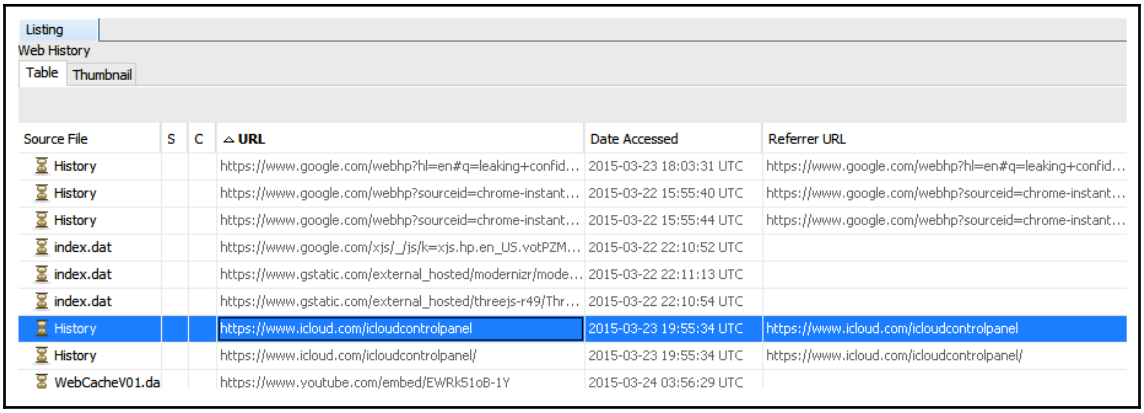
In another scenario, responders may be tasked with identifying whether an employee accessed confidential files so that they could pass them on to a competitor. This may involve examining the system for the times and dates when files were accessed, email addresses that may have been used, external cloud storage sites that were accessed, or USB storage that was connected to the system. Finally, a full list of these files may provide insight into the confidential documents that were moved.

Web artifacts

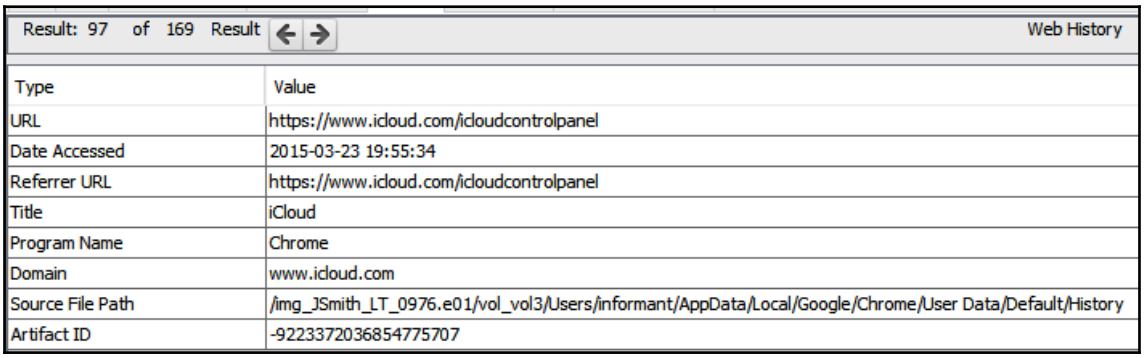
There are several types of incidents where it may be necessary to examine a system for evidence of malicious activity that's been conducted by a user. Previously, we mentioned accessing cloud-based storage where a malicious insider has uploaded confidential documents. In other circumstances, social engineering attacks may have an unsuspecting employee navigate to a compromised website that subsequently downloads malicious software. In either case, Autopsy provides us with the ability to examine several areas of web artifacts.

The first of these web artifacts is the web history. In the event of a social engineering attack that involves a user navigating to a malware delivery site, this data may provide some insight into the specific URL that was navigated to. This URL can then be extracted and compared with known malicious website lists from internal or external sources. In other cases, where an insider has accessed an external cloud storage site, the web history may provide evidence of this activity. Let's take a look at this case in detail:

1. Clicking on the **Web History** section in the left-hand pane opens the center pane and shows detailed information concerning a URL that was accessed by the system:



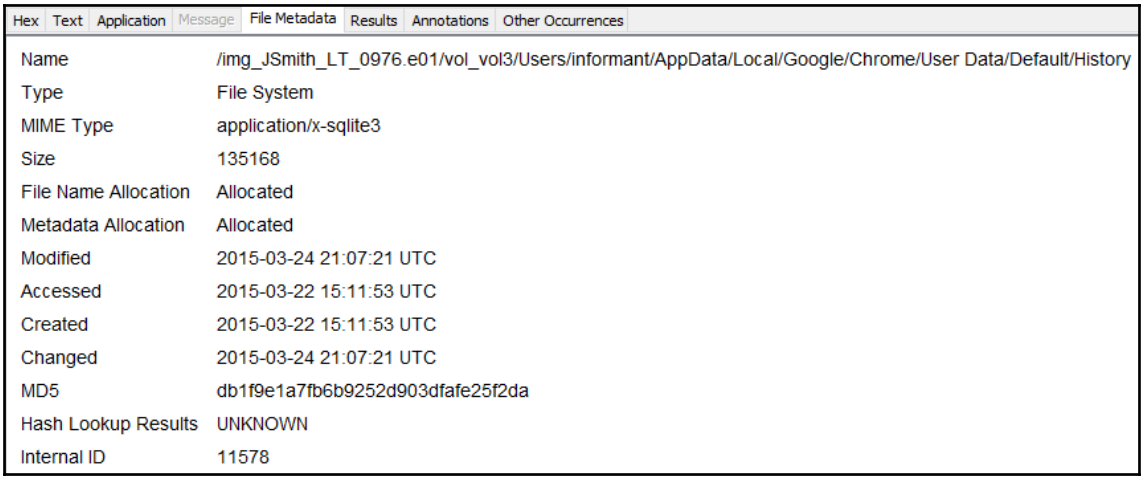
2. In the preceding screenshot, Autopsy indicates that the iCloud service was accessed by this system. Further information provided by Autopsy allows the analyst to evaluate other information, such as the location of the artifact and what type of browser was used. This information can be accessed via the **Results** tab in the lower pane:



Result: 97 of 169 Result Web History

Type	Value
URL	https://www.icloud.com/icloudcontrolpanel
Date Accessed	2015-03-23 19:55:34
Referrer URL	https://www.icloud.com/icloudcontrolpanel
Title	iCloud
Program Name	Chrome
Domain	www.icloud.com
Source File Path	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppData/Local/Google/Chrome/User Data/Default/History
Artifact ID	-9223372036854775707

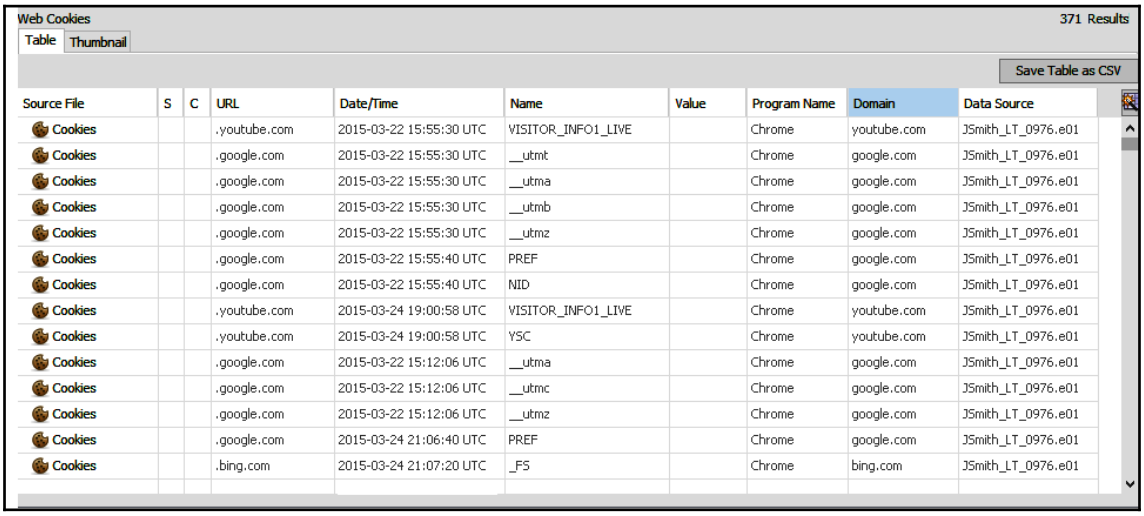
3. In addition, Autopsy provides the metadata of the specific file under examination. Clicking on the **File Metadata** tab produces the following data:



Hex | Text | Application | Message | File Metadata | Results | Annotations | Other Occurrences

Name	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppData/Local/Google/Chrome/User Data/Default/History
Type	File System
MIME Type	application/x-sqlite3
Size	135168
File Name Allocation	Allocated
Metadata Allocation	Allocated
Modified	2015-03-24 21:07:21 UTC
Accessed	2015-03-22 15:11:53 UTC
Created	2015-03-22 15:11:53 UTC
Changed	2015-03-24 21:07:21 UTC
MD5	db1f9e1a7fb6b9252d903dfafe25f2da
Hash Lookup Results	UNKNOWN
Internal ID	11578

4. As the preceding screenshot shows, there are some more details concerning that file. For example, the analyst can gather time information, file location, and an MD5 hash, which can be utilized to compare any extracted files that are examined further. In some circumstances, a suspect may decide to delete the browsing history from the system in an effort to hide any malicious activity. Another location that may provide evidence of sites that have been accessed by a malicious insider is web cookies. These can be accessed in the left-hand pane under **Web Cookies**. Clicking on this produces a list of the cookies that are still on the system:



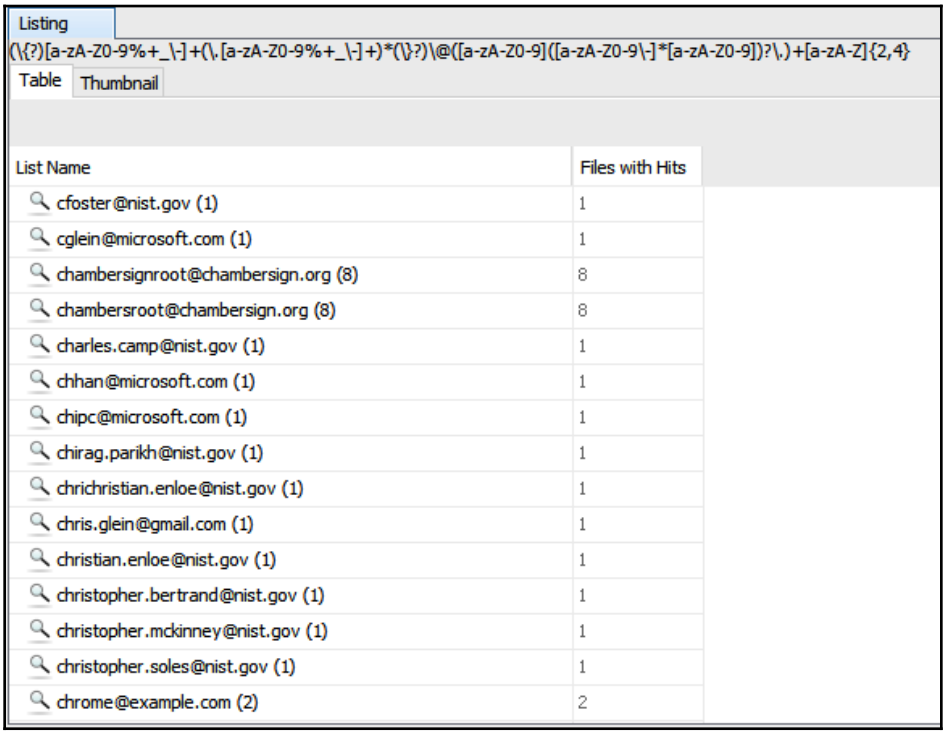
Source File	S	C	URL	Date/Time	Name	Value	Program Name	Domain	Data Source
Cookies			.youtube.com	2015-03-22 15:55:30 UTC	VISITOR_INFO1_LIVE		Chrome	youtube.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utmt		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utma		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utmb		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:30 UTC	__utmz		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:40 UTC	PREF		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:55:40 UTC	NID		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.youtube.com	2015-03-24 19:00:58 UTC	VISITOR_INFO1_LIVE		Chrome	youtube.com	JSmith_LT_0976.e01
Cookies			.youtube.com	2015-03-24 19:00:58 UTC	YSC		Chrome	youtube.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:12:06 UTC	__utma		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:12:06 UTC	__utmc		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-22 15:12:06 UTC	__utmz		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.google.com	2015-03-24 21:06:40 UTC	PREF		Chrome	google.com	JSmith_LT_0976.e01
Cookies			.bing.com	2015-03-24 21:07:20 UTC	_FS		Chrome	bing.com	JSmith_LT_0976.e01

Depending on the type of incident, web artifacts can play an important role. Autopsy has some functionality for this, but responders may find that other commercial solutions provide a much more robust platform. Evidence Finder by Magnet Forensics (`www.magnetforensics.com`) scours the entire system for internet artifacts and then presents them in a way that is easy for the analyst to view. Another key advantage of commercial solutions such as this is that their functionality is updated continuously. Depending on the frequency of internet and web artifact searching, the inclusion of tools such as this may be beneficial.

Email

Locating suspect emails continues to be a task that incident responders often engage in. This can include externally caused incidents such as social engineering, where responders may be tasked with locating a suspect email that had malware attached to it. In other circumstances, malicious insiders may have sent or received communication that was inappropriate or violated company policy. In those cases, responders may be tasked with recovering those emails so that they can be included in termination proceedings or in legal action.

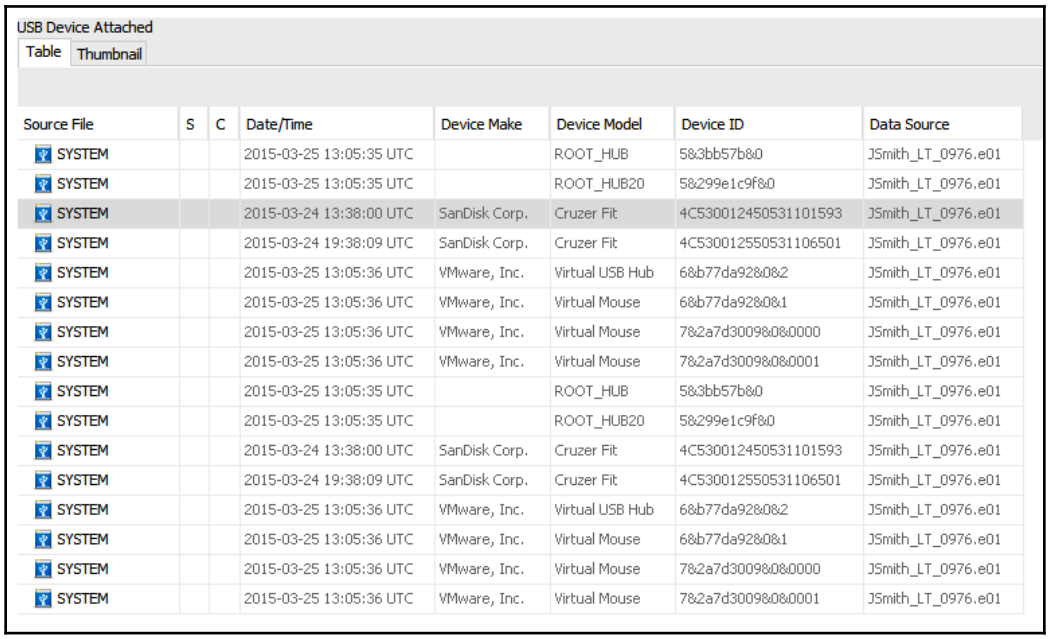
Autopsy has the ability to locate emails contained on the system. From these emails, they may be able to identify one or more suspicious emails and domains that can be further researched to see if they are associated with social engineering or other malicious activity. Simply click on the **Email Addresses** tab in the left-hand pane. From there, the analyst can see the email addresses that are located on the system:



Next, let's look at the type of attached devices.

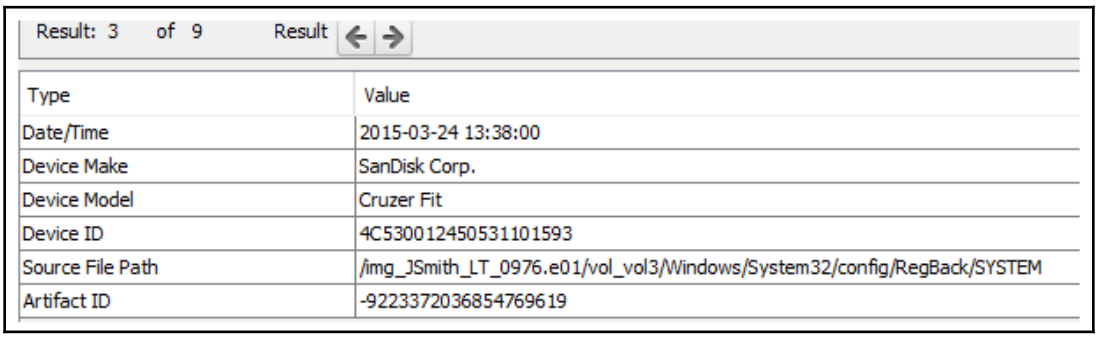
Attached devices

Another key piece of evidence that may be useful to an analyst is data about when specific devices were attached to the system. In the scenario of a malicious insider attempting to steal confidential documents, knowing whether they utilized a USB device would be helpful. Autopsy utilizes the registry settings located on the system to identify the types of devices attached and the last time that they were used. In this case, the output of clicking **Devices Attached** in the left-hand pane produces the following results:



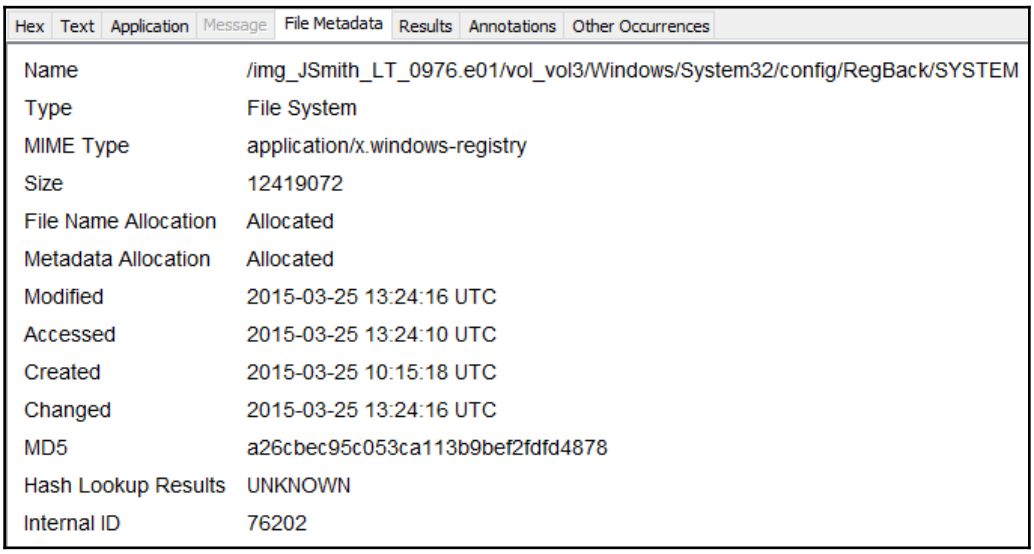
Source File	S	C	Date/Time	Device Make	Device Model	Device ID	Data Source
SYSTEM			2015-03-25 13:05:35 UTC		ROOT_HUB	5&3bb57b8&0	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:35 UTC		ROOT_HUB20	5&299e1c9f&0	JSmith_LT_0976.e01
SYSTEM			2015-03-24 13:38:00 UTC	SanDisk Corp.	Cruzer Fit	4C530012450531101593	JSmith_LT_0976.e01
SYSTEM			2015-03-24 19:38:09 UTC	SanDisk Corp.	Cruzer Fit	4C530012550531106501	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual USB Hub	68b77da928&0&2	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual Mouse	68b77da928&0&1	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual Mouse	78a2a7d3009&0&0000	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual Mouse	78a2a7d3009&0&0001	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:35 UTC		ROOT_HUB	5&3bb57b8&0	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:35 UTC		ROOT_HUB20	5&299e1c9f&0	JSmith_LT_0976.e01
SYSTEM			2015-03-24 13:38:00 UTC	SanDisk Corp.	Cruzer Fit	4C530012450531101593	JSmith_LT_0976.e01
SYSTEM			2015-03-24 19:38:09 UTC	SanDisk Corp.	Cruzer Fit	4C530012550531106501	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual USB Hub	68b77da928&0&2	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual Mouse	68b77da928&0&1	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual Mouse	78a2a7d3009&0&0000	JSmith_LT_0976.e01
SYSTEM			2015-03-25 13:05:36 UTC	VMware, Inc.	Virtual Mouse	78a2a7d3009&0&0001	JSmith_LT_0976.e01

Drilling down into the **Results** tab, the analyst would be able to identify the type of device and the date and time that the USB device was attached:



Type	Value
Date/Time	2015-03-24 13:38:00
Device Make	SanDisk Corp.
Device Model	Cruzer Fit
Device ID	4C530012450531101593
Source File Path	/img_JSmith_LT_0976.e01/vol_vol3/Windows/System32/config/RegBack/SYSTEM
Artifact ID	-9223372036854769619

Finally, a more detailed examination of the **File Metadata** would show additional data that can be utilized to reconstruct the time that the USB device was accessed on the system:



Next, let's look at the deleted files.

Deleted files

Files that have been deleted can also be reconstructed, either partially or completely. The Windows operating system will not delete files when the user selects deletion. The operating system will mark the space a deleted file takes up in the Master File Table as available to write new files to. As a result, responders may be able to view deleted files that have not been overwritten.

One challenge that is facing forensic analysts is the use of **solid state drives** (**SSDs**) in tablets and computers. Deleted files can often be recovered from traditional platter hard drives, even after a system is powered down. With SSDs, the operating system will often remove deleted files to make the storage of files more efficient. The following website has an excellent breakdown of this if you want to find out more: `https://www.datanarro.com/the-impact-of-ssds-on-digital-forensics/`.

To view the deleted files on a system, click on the **Deleted Files** tab in the left-hand pane. From here, the analyst can see all of the files that have been marked for deletion:

Name	S	C	Location	Modified Time	Change Time	Access Time
System.Data.Entity.dll			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Re...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
nb.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Ap...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
ko.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Bo...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
it.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
es.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
ru.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
ColorSync.resources			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
libdispatch.dll			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
es.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
pthreadVC2.dll			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
fi.lproj			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
AuditResultView.js			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
buildSystemOnly.js			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
ConsoleView.js			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
DOMAgent.js			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
ElementsPanelDescriptor.js			/img_JSmith_LT_0976.e01/vol_vol3/Program Files (x86)/Co...	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00

From here, the analyst can search through deleted files. These files may hold evidentiary value. For example, in the case of malicious insider activity, if several sensitive files are found in the deleted files, all of which have been deleted within the same time period, it may be indicative of the insider attempting to cover their tracks by deleting suspicious files.

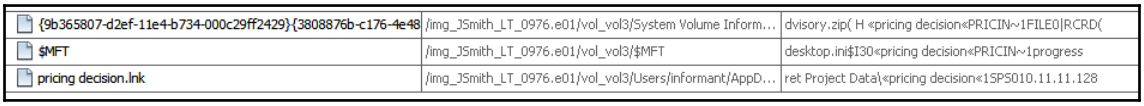
Keyword searches

One key advantage that forensic applications have is the ability to perform keyword searches. This is especially advantageous as disk drives have gotten larger and responders would have to parse through an overwhelming quantity of data. Keywords are often derived from other elements of the investigation or by using external sources. For example, if an analyst is investigating a malware incident, they may use a suspicious DLL or executable name from the analysis of the memory image. In other instances, such as a malicious insider being suspected of accessing confidential information, keywords in those documents, either secret or confidential, can be used to see if the suspect had used the system to access those files.

Autopsy has the ability to perform keyword searches while utilizing an exact or a substring match. For example, an analyst is tasked with determining whether a system was used to access a particular file titled `pricing decision` (16). The analyst is tasked with locating any trace evidence that would indicate that the system accessed it and determining which user accessed the file.

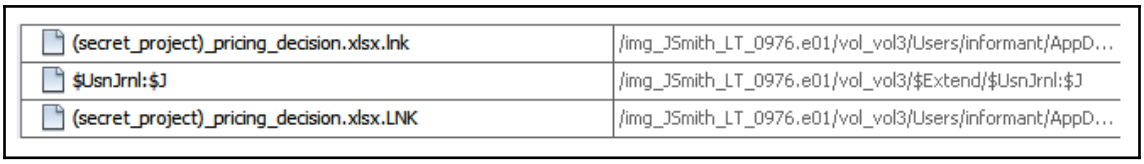
The analyst would navigate to the top-right corner and input the following text pricing decision in the field. In this case, an exact match will be utilized. Once selected, they would click the **Search** button. The left-hand pane will indicate whether there were any hits on that keyword. In this case, pricing decision has 19 hits.

In the center pane will be a list of the files that contained the hits. The first file that stands out is MFTentry. This indicates that there was at least an entry on the MFT:



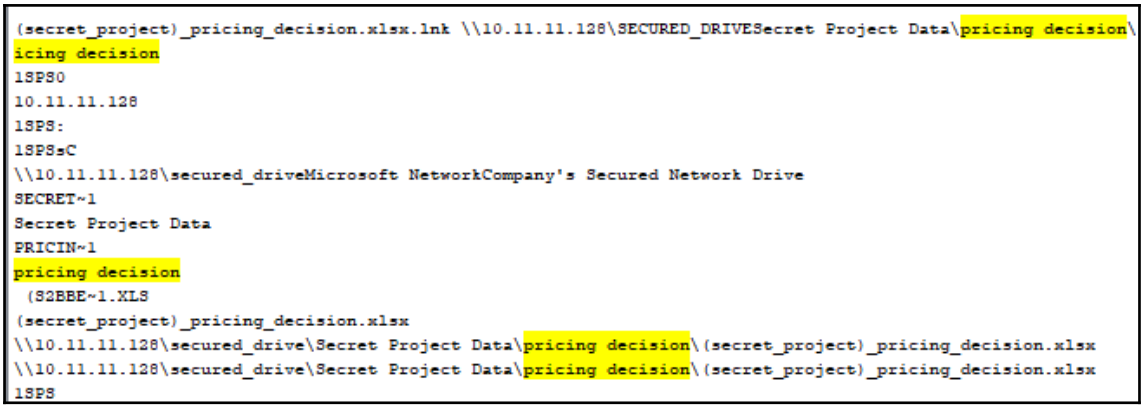
{9b365807-d2ef-11e4-b734-000c29ff2429}{3808876b-c176-4e48	/img_JSmith_LT_0976.e01/vol_vol3/System Volume Inform...	dvisory.zip(H «pricing decision«PRICIN~1FILE0|RCRD(
$MFT	/img_JSmith_LT_0976.e01/vol_vol3/$MFT	desktop.ini$I30«pricing decision«PRICIN~1progress
pricing decision.lnk	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppD...	ret Project Data\«pricing decision«1SPS010.11.11.128

Further review of the results indicates that there are two link files associated with the pricing decision spreadsheet. A link file is a shortcut file and is created when a document is opened. At that point, the Windows OS creates a link file with the `.LNK` extension within the MFT. This new entry is also placed in the `Recents` folder and allows the user to access recent documents. In this case, the link file shows that the pricing decision spreadsheet was opened:



(secret_project)_pricing_decision.xlsx.lnk	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppD...
$UsnJrnl:$J	/img_JSmith_LT_0976.e01/vol_vol3/$Extend/$UsnJrnl:$J
(secret_project)_pricing_decision.xlsx.LNK	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppD...

By selecting the first link file in the upper pane, additional details can be uncovered in the lower pane. For instance, if the analyst clicks on indexed text, there will be information concerning when the file was accessed:



```
(secret_project)_pricing_decision.xlsx.lnk \\10.11.11.128\SECURED_DRIVESecret Project Data\pricing decision\
icing decision
1SPS0
10.11.11.128
1SPS:
1SPSsC
\\10.11.11.128\secured_driveMicrosoft NetworkCompany's Secured Network Drive
SECRET~1
Secret Project Data
PRICIN~1
pricing decision
 (S2BBE~1.XLS
(secret_project)_pricing_decision.xlsx
\\10.11.11.128\secured_drive\Secret Project Data\pricing decision\(secret_project)_pricing_decision.xlsx
\\10.11.11.128\secured_drive\Secret Project Data\pricing decision\(secret_project)_pricing_decision.xlsx
1SPS
```

By analyzing this data, the analyst can infer that the system did, in fact, access a shared computer at `10.11.11.128` and accessed the pricing decision spreadsheet. Further details are also available by clicking on the **File Metadata** tab:

Name	/img_JSmith_LT_0976.e01/vol_vol3/Users/informant/AppData/Roaming/Microsoft/Windows/Recent/(secret_project)_pricing_decision.xlsx.lnk
Type	File System
MIME Type	application/octet-stream
Size	1952
File Name Allocation	Allocated
Metadata Allocation	Allocated
Modified	2015-03-23 20:26:53 UTC
Accessed	2015-03-23 20:26:53 UTC
Created	2015-03-23 20:26:53 UTC
Changed	2015-03-23 20:26:53 UTC
MD5	a9a4d030a0e6124ef8610617ee9125fc

This data indicates not only the time that the file was accessed, but also that the informant is clearly the account accessing the file. This is evident through examining the metadata, which shows that the link file was created within the MFT.

By clicking **Operating System User Account** in the left-hand pane, the analyst will be able to find data concerning the suspect user account:

Type	Value
Username	informant
User ID	S-1-5-21-2425377081-3129163575-2985601102-1000

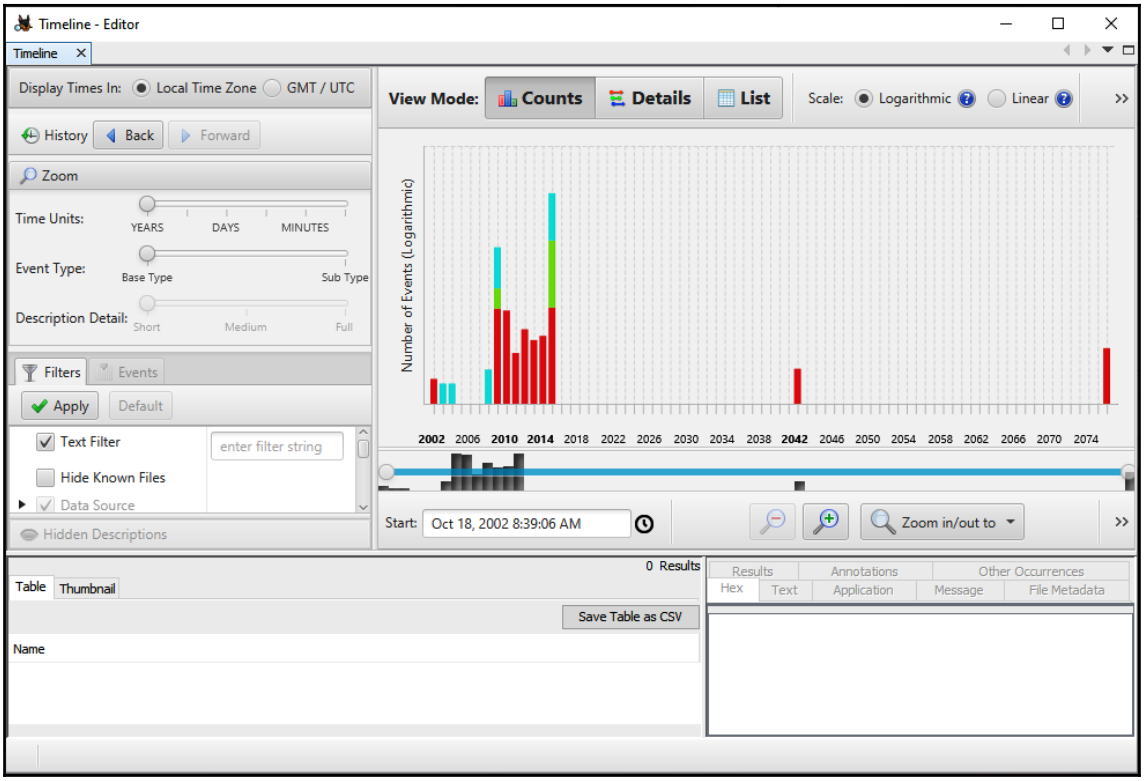
The ability to search for keywords is one aspect of disk forensic platforms that truly makes it worth the investment. With the ability to zero in on specific aspects of the incident, responders can quickly sift through the mountains of data to the key pieces of data they need to reconstruct the sequence of events.

Timeline analysis

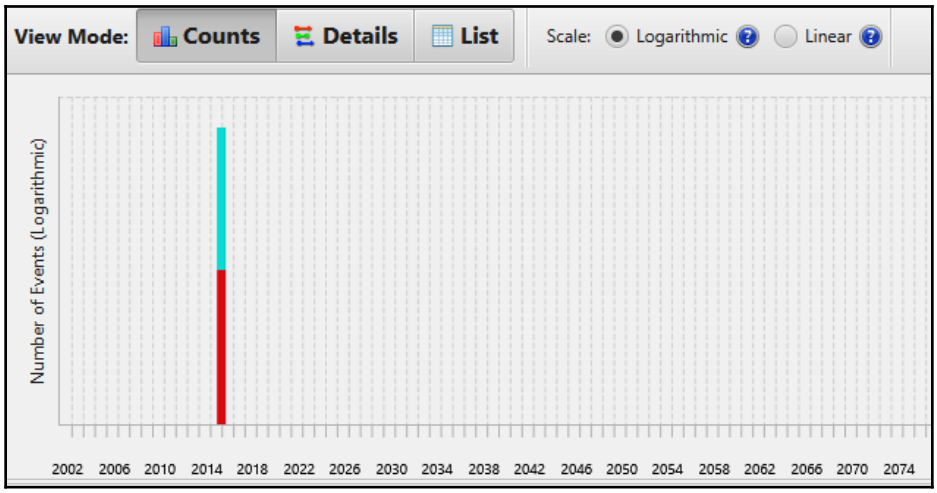
When investigating an incident, it is critical to have an idea of when applications or files were executed. Timestamps can sometimes be found in other aspects of the investigation, such as when examining memory images. Also, identifying specific DLL files or executable files in the memory image can be compared to the date and time they were accessed in order to correlate other activity that's been observed on the system.

One aspect of digital forensics that bears repeating is to ensure that all the systems are using the same time zone. With network systems, this is usually accomplished with the **Network Time Protocol** (**NTP**). There are times where systems do not have normalized time through NTP. Responders should take great care in understanding what time zone and synchronization should be used. The best practice in regard to time is to set all the systems to UTC. This is critical if an organization is geographically diverse.

Autopsy has functionality specifically for timeline analysis. Simply clicking on the **Timeline** button at the top of the window will make Autopsy begin the process of parsing out timeline data. Depending on the size of the image file being analyzed, it may take a few minutes. Once completed, the following window will open:



From here, the analyst can utilize several features. First is the text filter on the left-hand side of the screen. From here, the analyst can search for specific text in files. For example, the analyst has already identified that the spreadsheet named `pricing decision` had been accessed by the system under investigation. If the analyst would like to know whether that file was accessed at any other times, they could enter pricing into the **Text Filter** box and click **Apply**, which would produce the following results:



From this graph, the analyst can further drill down into the specific times the file was accessed by clicking on the colored bars.

The responders can now see that the file was only accessed on one particular date and time from this system.

MFT analysis

Another technique that can be leveraged for timeline analysis is utilizing external tools to analyze the MFT. Autopsy allows the analyst to export the MFT for analysis using third-party tools. In this case, the analyst will utilize a Python script called `analyzeMFT`, which is available on GitHub at `https://github.com/dkovar/analyzeMFT`. Utilizing this script, the analyst will produce an Excel spreadsheet with date and time information. This script can also be combined with other scripts, such as `Log2Timeline`, to create files so that timestamps can be reviewed.

Follow these steps to create the spreadsheet while utilizing `analyzeMFT`:

1. Download and install analyzeMFT from `https://github.com/dkovar/analyzeMFT`.
2. Extract the MFT from the compromised system. In Autopsy, look in the Volume 3 file structure for the filenamed `$MFT`. Extract the file by right-clicking on the filename and selecting **Extract File(s)**. Save the file to a storage folder. It's good practice to change the filename to something other than `$MFT`. In this case, `JSmith_LT_0976` is used.
3. Once the file has been saved, navigate to the folder containing the extracted file and type in the following command:

```
dfir@DESKTOP-SFARF6G: /mnt/d/analyzeMFT-master$ python
analyzeMFT.py -f JSmith_LT_0976_\$MFT -c JSMith_LT_0976_Timeline
```

   This command tells `analyzeMFT` to parse the MFT and output the results to a CSV file, which can then be viewed by Microsoft Excel.

4. Navigate to the folder containing the MFT file; the CSV file should be there.

The analyst may have to add the `.csv` extension to the file for it to open properly.

From there, they can view a spreadsheet that's utilizing the program of choice (in this case, Microsoft Excel is being utilized):

2010-11-21|07:06:15.881506|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-alg.resources_31bf3856ad364e35_6.1.
2009-07-14|03:20:30.316587|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-alg_31bf3856ad364e35_6.1.7600.1638
2010-11-21|07:06:26.337038|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-alttab.resources_31bf3856ad364e35_
2009-07-14|05:30:18.216305|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-alttab_31bf3856ad364e35_6.1.7600.16
2015-03-23|20:26:53.986593|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Users/informant/AppData/Roaming/Microsoft/Windows/Recent/(secret_project)_
2015-03-23|20:26:54.002193|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Users/informant/AppData/Roaming/Microsoft/Windows/Recent/pricing decision.ln
2009-07-14|05:30:10.993492|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-appcompat-adm_31bf3856ad364e35_
2010-11-21|07:06:19.819912|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-appid.resources_31bf3856ad364e35_
2010-11-21|03:17:30.539425|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-appid_31bf3856ad364e35_6.1.7601.17
2010-11-21|07:06:15.881506|TZ|...B|FILE|NTFS	$MFT\|$FN	[...B]	time|user|host|/Windows/winsxs/amd64_microsoft-windows-appwiz.resources_31bf3856ad364e35_

From here, the analyst can review the entire timeline, and also use any of the searching tools available in the spreadsheet program. There are other scripts and tools that are available to responders so that they can parse out this information even further. One such example is the use of `Log2Timeline` and `analyzeMFT`, both of which can be found on GitHub at `https://github.com/log2timeline/`.

Registry analysis

There is a great deal of activity that occurs under the hood on the Windows operating system. One place that this activity occurs and is documented is in the Windows Registry. The Windows Registry is a database that stores the low-level system settings for the Windows operating system. This includes settings for devices, security, services, and the storage of user account security settings in the **Security Accounts Manager** (**SAM**).

The registry is made up of two elements. The first is the key. The key is a container that holds the second element – the values. These values hold specific settings information. The highest-level key is called the root key and the Windows operating system has five root keys, all of which are stored on the disk in the registry hives. These registry hives are located in the `%SystemRoot%\system32\config` folder on the Windows file structure:

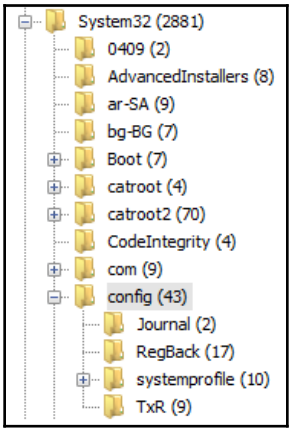
- `HKEY_CURRENT_USER`
- `HKEY_USERS`
- `HKEY_CLASSES_ROOT`
- `HKEY_LOCAL_MACHINE`
- `HKEY_CURRENT_CONFIG`

Of the five root keys, the most valuable during an incident investigation is the `HKEY_LOCAL_MACHINE` or HKLM key. This key contains the following subkeys (these are the ones that are the most interesting during an investigation):

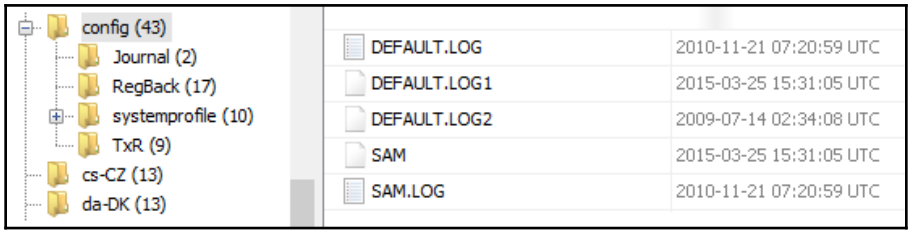
- **SAM**: This is the location where the Windows OS stores the user's passwords in the LM or NTLM hash form. The main purpose of the SAM subkey is to maintain the Windows account passwords.
- **Security**: This subkey contains the security information of the domain that the system is connected to.
- **Software**: The software subkey is the repository for software and Windows settings. This subkey is often modified by software or system installers. This is a good location to check for additions or modifications that have been made to software by malware.
- **System**: This subkey stores information about the Windows system configuration. One key piece of evidence that is also included within the system subkey is the currently mounted devices within a filesystem.

Another source of data that can be critical to an incident investigation is the `HKEY_CURRENT_USER` key. Attackers may make changes to a user account or profile as part of a privilege escalation attack. Changes that have been made to the user's data are recorded in that user's `NTUSER.dat` file. An `NTUSER.dat` file is created for every user account on the system and is located at `C:\Users\*UserName*`. This file contains the user's profile settings and may provide additional data on the systems that are connected, network connections, or other settings. Data contained within the `HKEY_CURRENT_USER` key may be of benefit in some incidents where user activity or user account modification of the system is suspected.

Responders can access the various registry hives using Autopsy. Simply navigate to the `vol3/Windows/System32/config` folder from the file structure in the left-hand pane:



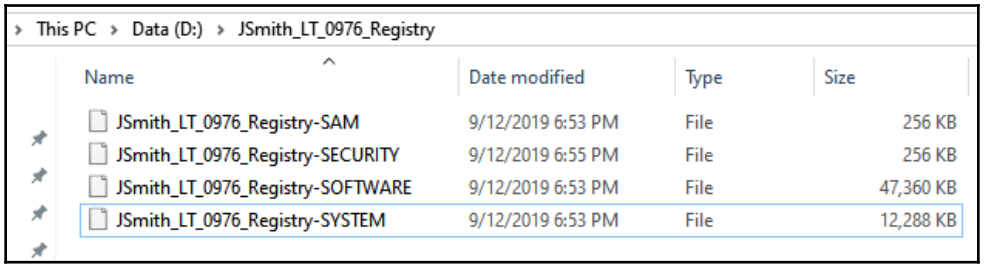
The `SAM` registry file is located in the center pane:



The actual examination and evidentiary value of registry key settings is, like many aspects of digital forensics, very detailed. While it is impossible to cover all of the aspects of registry forensics in this chapter, or even in this book, it is important for responders to be able to acquire the registry keys for evaluation, and also to have some familiarity with tools that can allow responders to gain some hands-on experience with evaluating registry settings.

In this case, the system, SAM, security, and software registry keys will be acquired for analysis. For this, the analyst can use Autopsy to acquire the proper keys and then examine them with a third-party tool. Let's take a look at how to do this:

1. First, navigate to the proper folder, `/System32/config`, on the third volume of the system image.
2. Next, select the four registry keys using the right mouse button and the *Ctrl* key. Right-click on one of the files and select **Export File(s)**.
3. Select a folder to output the registry keys to. In this case, a separate file folder was created to contain the keys. Select **Save**.
4. Verify that the registry keys have been saved:



The preceding screenshot shows the four registry files that have been acquired.

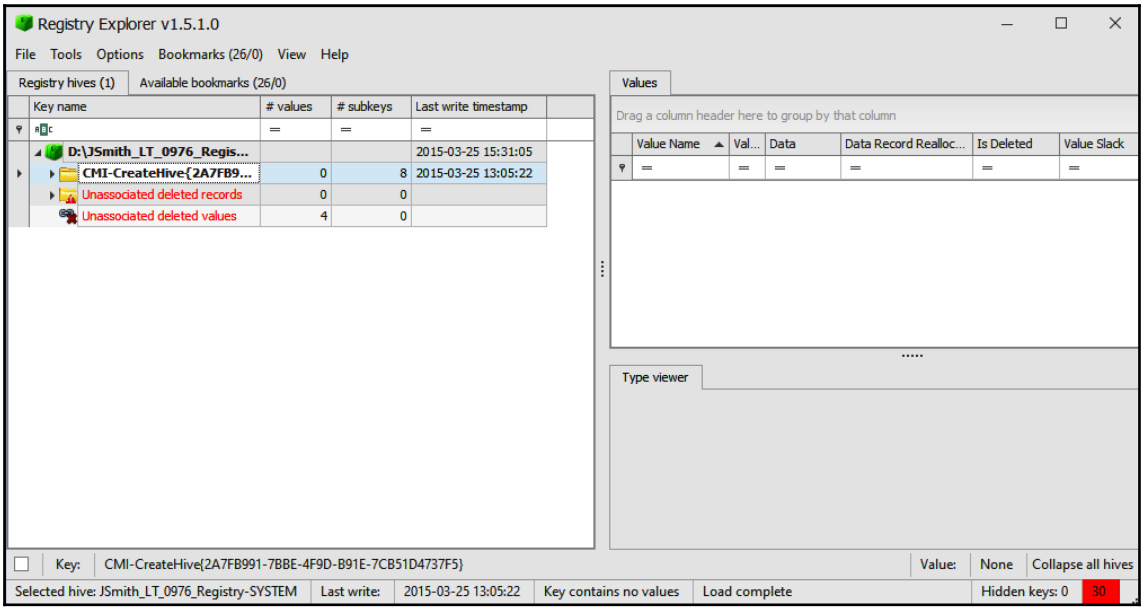
Now that the suspect image's registry files have been saved, the analyst can use a third-party tool to examine the registry. In this case, the Registry Explorer/RECmd Version 1.5.1.0 tool, which was developed by Eric Zimmerman, will be used to analyze the registry keys. This freeware application can be downloaded from `https://ericzimmerman.github.io/#!index.md`. Unzip the file to a safe location and execute the application.

Now that progress has been made in the analysis of the image, the analyst has identified that potential data loss has occurred via a USB device that was attached to the system at some point. While Autopsy has provided us with some information on this, it may be necessary to find out what registry key settings have been changed as a result of the USB being connected. The best location for additional information is contained within the system registry hive.

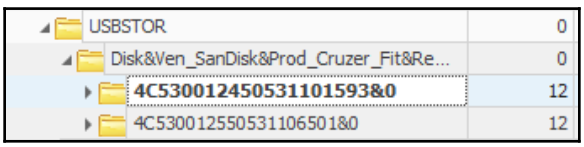
The Windows operating system records and maintains artifacts of when USB devices such as mass storage, iOS devices, digital cameras, and other USB devices are connected. This is due to the Plug and Play manager, which is part of the Windows operating system. The PnP receives notification that a USB has been connected and queries the device for information so that it can load the proper device driver. Upon completion, the Windows operating system will make an entry for the device within the registry settings.

To determine what USB devices were connected, follow these steps:

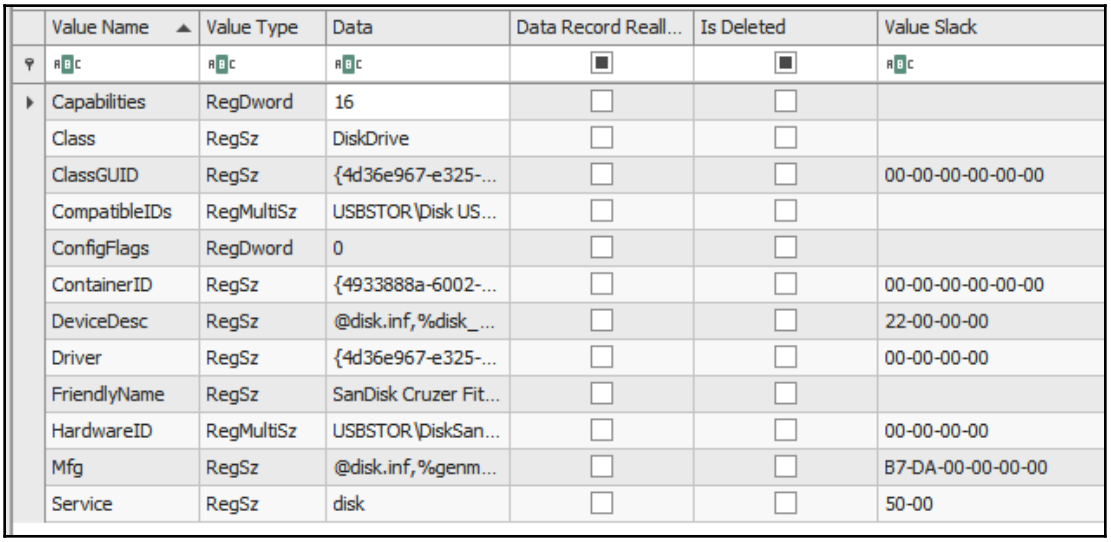
1. Open Registry Explorer.
2. Click **File** and then **Load Hive**.
3. Navigate to the system registry hive.
4. Once loaded, the following window will appear:



From here, navigate to the proper USB registry location at `ControlSet001\Enum\USBSTOR\`:

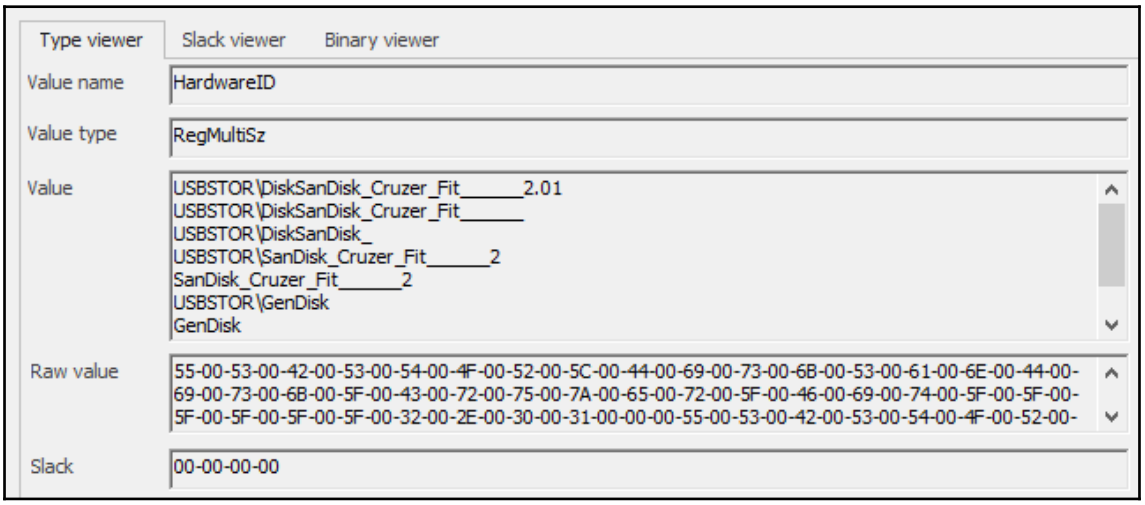


5. Click on the first registry value, `4C530012450531101593&0`. The following information will appear in the upper-right pane:



Value Name	Value Type	Data	Data Record Reall...	Is Deleted	Value Slack
Capabilities	RegDword	16	☐	☐	
Class	RegSz	DiskDrive	☐	☐	
ClassGUID	RegSz	{4d36e967-e325-...	☐	☐	00-00-00-00-00-00
CompatibleIDs	RegMultiSz	USBSTOR\Disk US...	☐	☐	
ConfigFlags	RegDword	0	☐	☐	
ContainerID	RegSz	{4933888a-6002-...	☐	☐	00-00-00-00-00-00
DeviceDesc	RegSz	@disk.inf,%disk_...	☐	☐	22-00-00-00
Driver	RegSz	{4d36e967-e325-...	☐	☐	00-00-00-00
FriendlyName	RegSz	SanDisk Cruzer Fit...	☐	☐	
HardwareID	RegMultiSz	USBSTOR\DiskSan...	☐	☐	00-00-00-00
Mfg	RegSz	@disk.inf,%genm...	☐	☐	B7-DA-00-00-00-00
Service	RegSz	disk	☐	☐	50-00

From here, the analyst has a lot of information they need to review. Of particular importance is the `HardwareID`. Clicking on that section of the output produces the following in the lower-right window:



Type viewer | Slack viewer | Binary viewer

Value name: HardwareID

Value type: RegMultiSz

Value:
USBSTOR\DiskSanDisk_Cruzer_Fit_____2.01
USBSTOR\DiskSanDisk_Cruzer_Fit_____
USBSTOR\DiskSanDisk_
USBSTOR\SanDisk_Cruzer_Fit_____2
SanDisk_Cruzer_Fit_____2
USBSTOR\GenDisk
GenDisk

Raw value:
55-00-53-00-42-00-53-00-54-00-4F-00-52-00-5C-00-44-00-69-00-73-00-6B-00-53-00-61-00-6E-00-44-00-
69-00-73-00-6B-00-5F-00-43-00-72-00-75-00-7A-00-65-00-72-00-5F-00-46-00-69-00-74-00-5F-00-5F-00-
5F-00-5F-00-5F-00-5F-00-32-00-2E-00-30-00-31-00-00-00-55-00-53-00-42-00-53-00-54-00-4F-00-52-00-

Slack: 00-00-00-00

What the analyst has been able to uncover by evaluating the date and time is that a SanDisk Cruzer Fit was connected to the system. The analyst was able to ascertain that it was connected at 13:38:00 on 03/24/2015. This is critical compared to the date and time that the confidential files were accessed.

As we mentioned previously, registry analysis is a deep subset of digital forensics in and of itself. Whole volumes have been written on the evidentiary value present in the settings and entries in registry hives. At a minimum, responders should be prepared to acquire this evidence for others for further examination. That being said, as responders gain more and more experience and skill, the registry should be an area that can be leveraged for evidence when examining a disk image.

Summary

In many ways, this chapter just scratches the surface of what information can be found by leveraging disk forensic tools. The exploration of a disk image by Autopsy demonstrated some of the features that are available to responders. From here, extracting other data stores such as the Windows Registry and MFT were explored to provide responders with an idea of what data is available during an incident analysis.

Specific tools and techniques are largely dependent on the tool that's utilized. What's important to understand is that modern operating systems leave traces of their activity all over the disk, from file change evidence in the MFT to registry key settings when new user accounts are added. Incident responders should have expertise in understanding how modern operating systems store data and how to leverage commercial or freeware tools to find this data. Taken in concert with other pieces of evidence that's obtained from network sources and in memory, disk evidence may provide more clarity on an incident and aid in determining its root cause. One area of focus when it comes to system storage analysis is the extraction and examination of log files. Log files are a critical data point that provides responders with a great deal of information.

The next chapter will carry on from the work that was done here and address how log files can be utilized in an incident investigation.

Questions

1. What are some of the features that are available with commercial and open source forensic platforms?

   A) Hex viewer
   B) Email carving
   C) Metadata viewer
   D) All of the above

2. In what registry hive could an incident responder find USBs that have been connected to the system?

   A) SAM
   B) Security
   C) System
   D) User profile

3. Web history may provide data on a phishing URL that's been accessed by the system.

   A) True
   B) False

4. Which of the following is not a Windows registry hive?

   A) System
   B) SAM
   C) Storage
   D) Software

Further reading

- Autopsy GitHub: `https://github.com/sleuthkit/autopsy`
- Eric Zimmerman Tools: `https://ericzimmerman.github.io/#!index.md`
- Eric Zimmerman Tools Cheat Sheet: `https://digital-forensics.sans.org/media/EricZimmermanCommandLineToolsCheatSheet-v1.0.pdf`
- *Registry Analysis with FTK Registry Viewer*: `https://subscription.packtpub.com/book/networking_and_servers/9781784390495/6/ch06lvl1sec37/registry-analysis-with-ftk-registry-viewer`
- *Windows Registry Analysis 101*: `https://articles.forensicfocus.com/2019/04/05/windows-registry-analysis-101/`

10
Analyzing Log Files

The Sherlock Holmes of France, Dr. Edmond Locard, was a pioneer in the field of forensic science. A criminologist and teacher, Locard developed techniques and methodologies that still inform forensic science today. One principle for which he is well known is **Locard's exchange principle**. This principle states that when a suspect interacts with a crime scene, they leave traces behind. In the physical world, this can include hair, fibers from clothing, blood, or skin, which is left on the various surfaces and objects within the crime scene. The crime scene itself also leaves traces on the suspect. Fibers from the carpet, dust, metal fragments, or glass from a broken window may make its way onto the suspect. Forensic science has developed more and more sophisticated practices and technology to find more and more minute traces.

Locard's work was centered on the physical world, well before computers were even a reality. Having said this, the principle that every action by the actor at a crime scene leaves traces is just as applicable in digital forensics as it is in the physical world. For example, an adversary or adversaries compromise a system and configure a command-and-control infrastructure on a web server. In doing so, they will leave trace evidence, in the form of firewall log entries. The execution of malware on the web server may leave traces in the running memory, event log entries, and malicious code on the storage device. Throughout the chain of events, the adversary will leave traces of their presence on the various devices with which they come into contact.

Previous chapters have discussed the various locations and techniques that can be leveraged by responders in uncovering these traces from memory, hard drives, and network traffic. One location that provides a wealth of data that can be leveraged is that of log files. Actions are logged across a wide range of hardware and software. What is needed is for responders to understand how to acquire these logs, how to examine them, and what they detail. In doing so, they may be able to ascertain a good deal about the root cause of an incident.

In this chapter, the discussion will focus on logs and log management, the use of log aggregation tools such as a security information and event management system, the Windows event logs, and—finally— analyzing Windows event logs. It is hoped that, through a discussion of some of these techniques, responders will be able to articulate how logs are critical to an incident investigation, while also being able to examine them as part of a larger incident investigation.

We will cover the following topics in this chapter:

- Logs and log management
- Security information and event management
- Windows event logs
- Windows event log analysis

Logging and log management

The lifeblood of a good incident investigation is evidence from a wide range of sources. Even something like a malware infection on a host system requires corroboration from a variety of sources. One common challenge with incident response, especially in smaller networks, is how the organization handles log management. For a comprehensive investigation, incident response analysts need access to as much network data as possible. All too often, organizations do not dedicate the proper resources to enabling the collection of comprehensive logs from network devices and other systems.

Prior to any incident, it is critical to clearly define how and what an organization will log, as well as how it will maintain those logs. This should be established within a log management policy and associated procedure. The **Computer Security Incident Response Team** (**CSIRT**) personnel should be involved in any discussion as to which logs are necessary or not, as they will often have insight into the value of one log source over another.

The **National Institute of Standards and Technology** (**NIST**) has published a short guide to log management, available at `http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf`.

Aside from the technical issues regarding log management, there are legal issues that must be addressed. The following are some issues that should be addressed by the CSIRT and its legal support prior to any incident:

- **Establish logging as a normal business practice**: Depending on the type of business and the jurisdiction, users may have a reasonable expectation of privacy absent from any expressly stated monitoring policy. In addition, if logs are enabled strictly to determine a user's potential malicious activity, there may be legal issues. As a result, the logging policy should establish that logging of network activity is part of the normal business activity and that users do not have a reasonable expectation of privacy.
- **Logging close to the event**: This is not so much an issue with automated logging, as logs are often created almost as the event occurs. From an evidentiary standpoint, logs that are not created close to the event lose their value as evidence in a courtroom.
- **Knowledgeable personnel**: The value of logs is often dependent on who created the entry, and whether or not they were knowledgeable about the event. In the case of logs from network devices, the logging software addresses this issue. As long as the software can be demonstrated to be functioning properly, there should be no issue.
- **Comprehensive logging**: Enterprise logging should be configured for as much of the enterprise as possible. In addition, logging should be consistent. A pattern of logging that is random will have less value in a court than a consistent pattern of logging across the entire enterprise.
- **Qualified custodian**: The logging policy should name a data custodian. This individual would speak for the logging procedure and the types of software utilized to create the logs. They would also be responsible for testifying to the accuracy of the logs and the logging software used.
- **Document failures**: Prolonged failures, or a history of failures in the logging of events, may diminish their value in a courtroom. It is imperative that any logging failure should be documented, and a reason associated with the failure.

- **Log file discovery**: Organizations should be made aware that logs utilized within a courtroom proceeding are going to be made available to the opposing legal counsel.
- **Logs from compromised systems**: Logs that originate from a known compromised system are suspect. In the event that these logs are to be introduced as evidence, the custodian or incident responder will often have to testify at length concerning the veracity of the data contained within the logs.
- **Original copies are preferred**: Log files can be copied from the log source to storage media. As a further step, any logs should be archived off the system as well. Incident responders should establish a chain of custody for each log file used throughout the incident, and these logs should be maintained as part of the case until an order from the court is obtained, allowing their destruction.

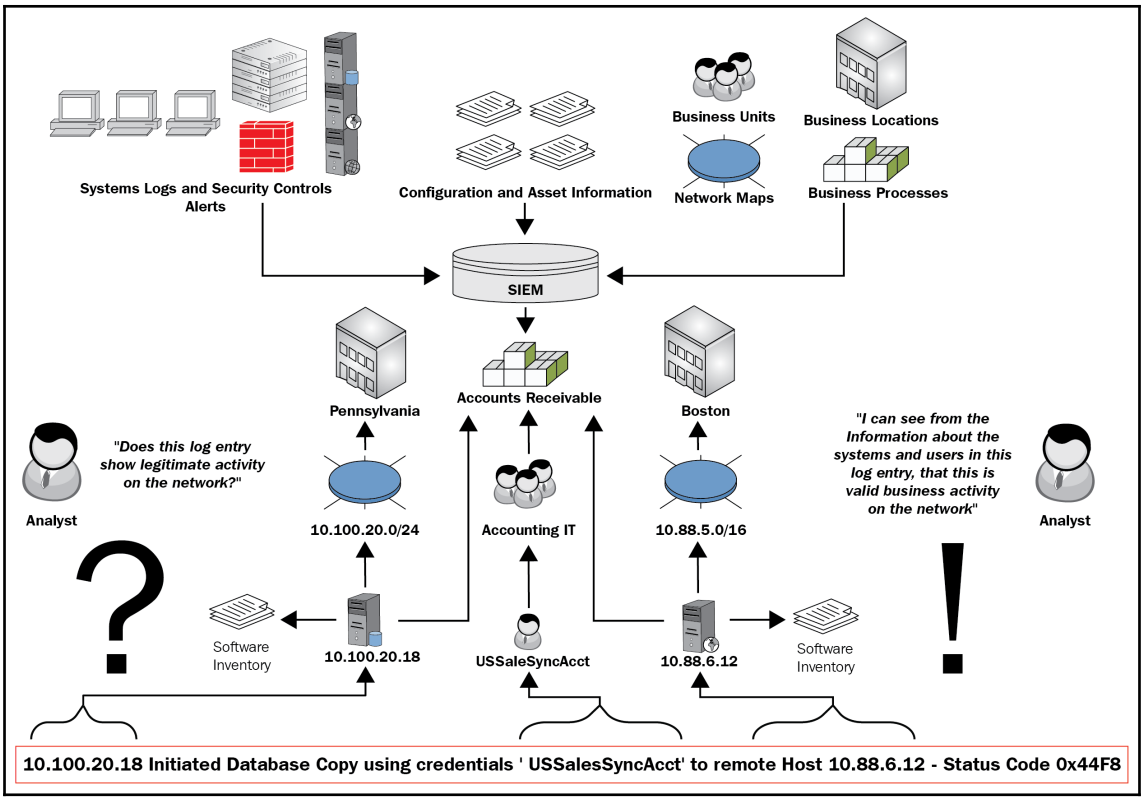
A log management process addresses the foundational elements required to identify those events that an organization deems necessary. From here, the next major component to a proper log management strategy is the technology that is leveraged for aggregation and review. This involves the integration of a **security information and event management** (**SIEM**) system as part of the overall structure of the log management process.

Working with event management systems

A significant challenge that a great many organizations have is the nature of logging on network devices. With limited space, log files are often *rolled over,* whereby new log files are written over older log files. The result is that, in some cases, an organization may only have a few days, or even a few hours, of important logs. If a potential incident happened several weeks ago, the incident response personnel will be without critical pieces of evidence.

One tool that has been embraced by a number of enterprises is a SIEM system. These appliances have the ability to aggregate log and event data from network sources and combine them into a single location. This allows the CSIRT and other security personnel to observe activity across the entire network, without having to examine individual systems.

The following screenshot illustrates how a SIEM integrates into the overall network:



A variety of sources, from security controls to SQL databases, are configured to send logs to the SIEM. In this case, the SQL database located at `10.100.20.18` indicates that the **USSalesSyncAcct** user account was utilized to copy a database to the remote host, located at `10.88.6.12`. The SIEM allows for a quick examination of this type of activity. For example, if it is determined that the **USSalesSyncAcct** account has been compromised, CSIRT analysts can quickly query the SIEM for any usage of that account. From there, they would be able to see the log entry that indicated a copy of a database to the remote host. Without that SIEM, CSIRT analysts would have to search each individual system that might have been accessed, a process that may be prohibitive.

From the SIEM platform, security and network analysts have the ability to perform a number of different tasks related to incident response, as follows:

- **Log aggregation**: Typical enterprises have several thousand devices within the internal network, each with their own logs; the SIEM can be deployed to aggregate these logs in a central location.

- **Log retention**: Another key feature that SIEM platforms provide is a platform to retain logs. Compliance frameworks, such as the **Payment Card Industry Data Security Standard** (**PCI-DSS**), stipulate that logs should be maintained for a period of 1 year, with 90 days immediately available. SIEM platforms can aid with log management, by providing a system that archives logs in an orderly fashion and allows for their immediate retrieval.
- **Routine analysis**: It is advisable when using a SIEM platform to conduct periodic reviews of the information. SIEM platforms often provide a dashboard that highlights key elements, such as the number of connections, data flow, and any critical alerts. SIEMs also allow for reporting, so that stakeholders can keep informed of activity.
- **Alerting**: SIEM platforms have the ability to alert to specific conditions that may indicate malicious activity. This can include alerting from security controls such as antivirus, intrusion prevention, or detection systems. Another key feature of SIEM platforms is event correlation. This technique examines the log files and determines whether there is a link or any commonality between the events. The SIEM then has the capability to alert to these types of events. For example, if a user account attempts multiple logins across a number of systems in the enterprise, the SIEM can identify that activity and alert the relevant parties to it.
- **Incident response**: As the SIEM becomes the single point for log aggregation and analysis, CSIRT analysts will often make use of the SIEM during an incident. CSIRT analysis will often make queries on the platform, as well as download logs for offline analysis. Because of the centralization of log files, the time to conduct searches and event collection is significantly reduced. For example, a CSIRT analysis has indicated a user account has been compromised. Without a SIEM, the CSIRT analyst would have to check various systems for any activity pertaining to that user account. With a SIEM in place, the analyst simply conducts a search of that user account on the SIEM platform, which has aggregated user account activity in logs from systems all over the enterprise. The result is that the analyst has a clear idea of the user account activity, in a fraction of the time it would have taken to examine logs from various systems throughout the enterprise.

SIEM platforms do entail a good deal of time and money to purchase and implement. Added to that cost is the constant upkeep, maintenance, and modification to rules that are necessary. From an incident response perspective, though, a properly configured and maintained SIEM is vital to gathering network-based evidence in a timely manner. In addition, the features and capability of SIEM platforms can significantly reduce the time it takes to determine the root cause of an incident once it has been detected.

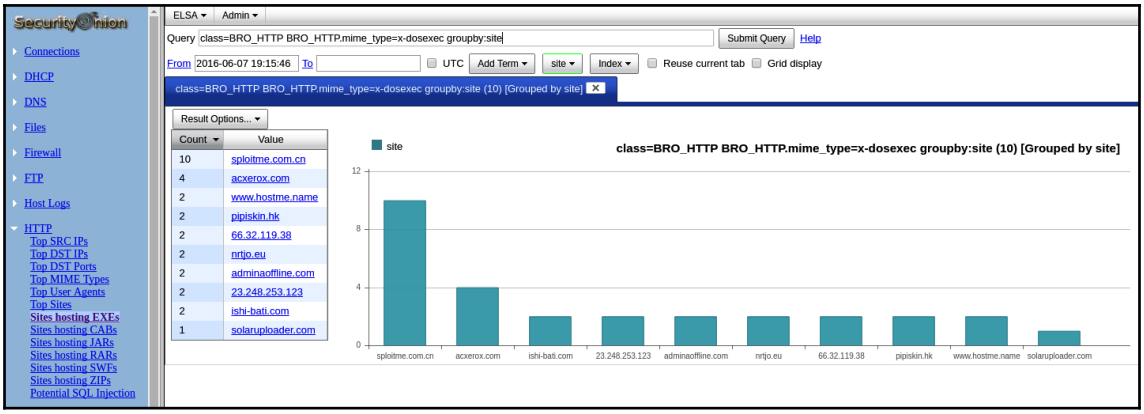
The following article has an excellent breakdown of use cases of SIEM platforms in enterprise environments, at `https://gbhackers.com/security-information-and-event-management-siem-a-detailed-explanation`.

There are several commercial and open source options for SIEM platforms. Each of these have a variety of features and capabilities. The following two open source options, Security Onion and Elastic Stack, are full featured tools that can aid responders in their analysis of log files.

Security Onion

Full-featured SIEM platforms may be cost-prohibitive for some organizations. One option that is available is the open source platform Security Onion. Security Onion ties a wide range of security tools—such as OSSEC, Suricata, and Snort—into a single platform. Security Onion also has features such as dashboards and tools for deep analysis of log files.

For example, the following screenshot shows the level of detail available:

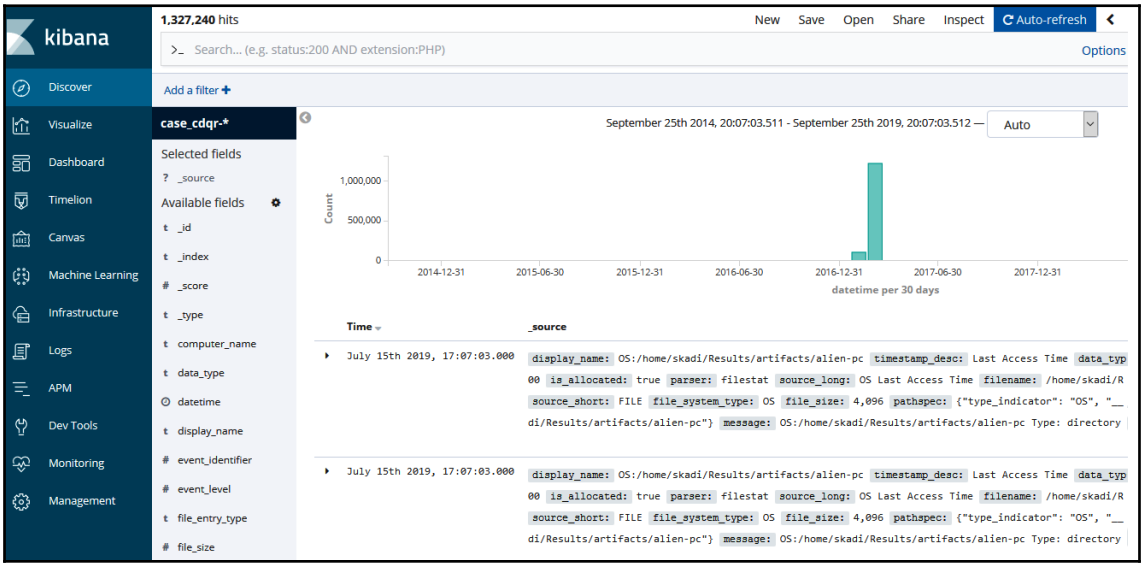


Although installing and deploying the Security Onion platform may require some resources in terms of time, it is a powerful, low-cost alternative, providing a solution to organizations that cannot deploy a full-featured SIEM solution (the Security Onion platform and associated documentation are available at `https://securityonion.net/`).

Elastic Stack

Another open source option for a SIEM is the Elastic Stack (or the ELK Stack, as it is commonly known). The Elastic Stack is a combination of three tools in one. The open source tools Elasticsearch, Logstash, and Kibana are combined to provide threat hunters an open source platform that ingests data and then transforms it into a format that can be viewed and analyzed via the Kibana GUI. This provides the ability for threat hunters to visualize log data from multiple systems at once. The Elastic Stack is built into a number of different open source security tools, including the aforementioned Security Onion. The Elastic Stack can also be configured as a standalone SIEM solution, with tools such as Winlogbeat, which forwards Windows event logs to the Elastic Stack.

The following is the most visible portion of the Elastic Stack, and that is the Kibana interface. This interface allows for data visualization and searching, as can be seen here:



SIEM platforms are an excellent way for responders to examine a wide range of logs from a large number of systems. One facet where this becomes critical is examining Windows Event Logs. The next section will examine the variety of Windows Event Logs and the insight they can provide responders into account and application usage.

Understanding Windows logs

The most prevalent endpoint operating system that responders will have to examine related to an incident is by far the Windows OS. Due to the overwhelming market share that Microsoft has, the vast majority of enterprise endpoints will be Microsoft desktop/laptop, server, or virtual systems. As a result, it is critical that responders have a solid understanding of how to leverage the Windows event logs for incident analysis.

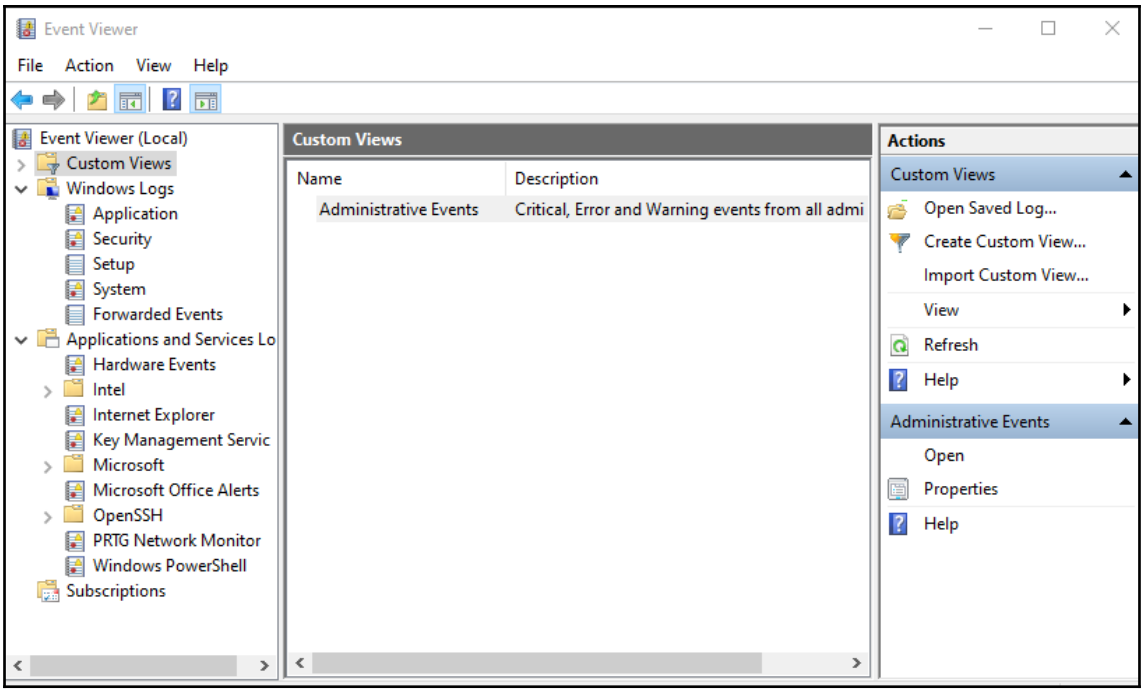
The Windows event logs provide extensive data on the actions of the operating systems, connections from other systems, and credential use, along with the use of PowerShell. Adversarial tactics from initial compromise using malware or other exploits, credential accessing, and elevation and lateral movement using the Windows operating system internal tools are often captured via the Windows event logs.

The specific logs that are captured during the operating system's activities are largely dependent on how the organization has configured them. Most enterprises utilize the Group Policy settings to configure which actions the system logs, as well as the storage space allocated for log files. Depending on the organization's log management practices, the Windows OS can be configured to log the use of PowerShell, **Server Message Block** (**SMB**) usage, application activity, DHCP client administration, and Task Scheduler maintenance.

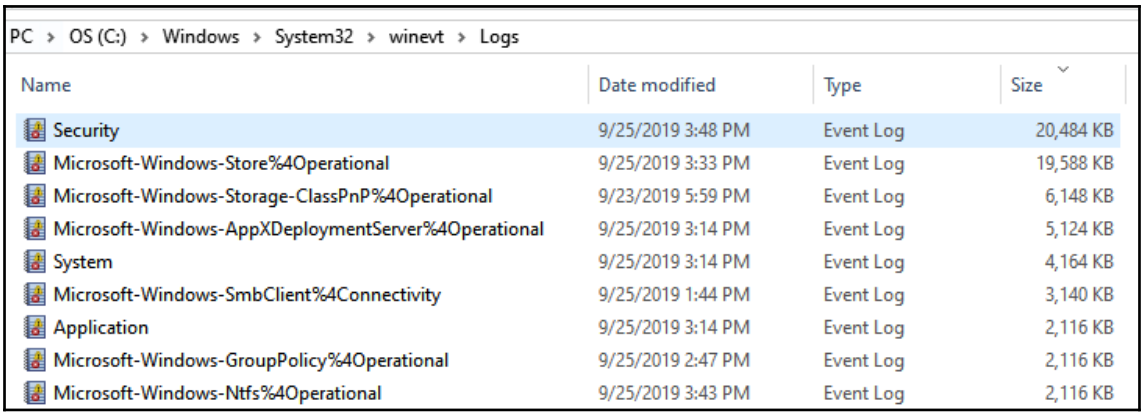
Most often, log management configurations are managed via the Windows Group Policy. Here, administrators have the ability to manage a wide range of systems via one policy.

To determine which logs are available on a local system, proceed as follows:

1. Search for `Event Viewer` in the **Search** field located in the Windows taskbar. Click on **Event Viewer**, and the following window will open:



2. From this viewer, the responder can get a good sense of what is being logged, and can even search for specific log entries. To access the logs directly for offline analysis, navigate to the default file path for log storage, at `C:\Windows\System32\winevt\logs`. This will show the variety of events that can be logged, as follows:



As previously stated, there are Windows event logs for a wide range of activities performed by the operating system. For the purposes of this chapter, the focus will be on three of the more pertinent Windows event log types. These types cover a wide range of activities and are useful in determining which actions have taken place on a potentially compromised system, and are detailed as follows:

- **Security logs**: These logs contain data entries concerning the security of the system. This includes logons, logoffs, security group membership, and program execution.
- **Application logs**: Application developers determine which types of activity applications will log. These are aggregated in the application log file.
- **System logs**: Often utilized to troubleshoot non-malicious activity, the system logs maintain data that the Windows OS creates.

There are several hundred Windows event log types. Depending on how the operating system is used, some of these are seldom—if ever—observed on the system. Others can be very common and are seen constantly in use, even in normal circumstances. The following are some of the more useful Windows event log types for responders:

- **4624 and 4634—logon and logoff**: These event log entries show the use of credentials on a potentially compromised system. In addition, the 4624 event IDs can show whether the logon was performed on the local system or through a remote connection, which is critical to finding lateral movement using the Windows SMB protocol.
- **4625—account failed logon**: One or two of these entries may not mean much. A few entries of this nature may indicate a fat-fingered logon here and there, but an excessive amount of these log entries is indicative of an adversary attempting to brute-force credentials.
- **4672—special privileges assigned to new logon**: This is the Windows OS equivalent of a user account attempting to elevate to root- or administrator-level privileges. This can be used to determine if an adversary is escalating privileges with a compromised account.
- **4688—a new process has been created**: This log entry documents every time a program is run. While there may be a lot to sift through in the logs, in terms of how many executables are run, threat hunters can focus on well-known abused programs, such as PsExec, `CMD.EXE`, or `Whami.exe`, to zero in on potentially malicious behavior.
- **4768-4773—Kerberos service**: There are several well-known exploits used by adversaries where the Kerberos Ticket Granting Ticket is utilized for elevated privileges. This attack—often referred to as Kerberoasting—is particularly devastating, as it allows attackers to run through the network with valid credentials.
- **5140—a network share object was accessed**: This activity is logged when a user account first logs on to a network share. Anomalies in time or user activity may be indicative of an adversary attempting to obtain confidential data, or ransomware attempting to infect network shares.
- **7045—a new service was installed:** This log entry occurs when a new service was installed by the user indicated within the log entry. Some strains of malware will install themselves as a service. A review of these log entries may indicate the presence of malicious code.

For more information please check the list of event logs in the *Appendix*.

As previously stated, there are over a hundred specific Windows event types available. The specific ones in use are often determined by the organization and have to be weighed against the amount of storage space that is available, and against the usefulness of the specific log entries during an investigation.

There are a number of resources that can be leveraged, to better understand Windows event logs. The first of these is the site `ultimatewindowssecurity.com`. This site provides a searchable database of the various Windows event log types by event ID. This is very useful in those circumstances where responders may come across a more obscure event ID. The MITRE Corporation also provides the ATT&CK knowledge database. This knowledge base can be searched for Windows event log IDs that may be pertinent to an investigation. For example, a responder is examining a system for indications that the system has been infected with the Carbanak malware. From the ATT&CK knowledge database, the responder is able to determine that Carbanak has created an account, and the Windows event ID for that is 4720. From here, the responder would be able to search systems for that specific event ID, and determine if there were any additional accounts that appeared suspicious.

As can be seen, the Windows Operating System has a significant number of log event types and IDs. The following section will provide the responder with a method to collect and analyze these log files.

Analyzing Windows event logs

Analyzing Windows event logs is a detailed process. One challenge that is often encountered by responders is the sheer number of logs that they may have to potentially analyze during an incident. In the case of multiple systems, the responder may have to contend with millions of separate event log entries. Cutting them down requires the use of specialized tools and processes, starting with acquisition, moving into triage, and then, finally, focusing on analyzing the key event logs that are pertinent to the incident investigation.

Acquisition

There are several methods that a responder can utilize in the acquisition of the Windows event logs. Ideally, log files should be sent to a SIEM, to allow the responders to search log entries across the enterprise. Unfortunately, many organizations face a significant hurdle in terms of storage costs with commercial, or even open source, platforms. The result is that they often must trade off the cost of aggregating these logs, by allowing the local systems to handle storage.

Since most of these logs are on the local system, responders will need to use techniques to gather them. The first of these techniques is to simply copy the event logs from the local system to some type of removable media. Simply navigate to the default directory `C:\Windows\System32\winevt\Logs`, and copy the pertinent logs. This method does require local access and a good deal of interaction with the local system. It is incumbent on the responder to document every action they took on the system, for proper reporting.

Responders also have the option of scripting the acquisition of log files through simple batch scripts. The acquisition can take place along with other actions to acquire evidence from a local system. For example, the following screenshot shows the acquisition of four Windows event log types from a local system:

```
echo Log Files
wevtutil epl Setup > \%COMPUTERNAME%\Logs\%COMPUTERNAME%_Setup.evtx
wevtutil epl System > \%COMPUTERNAME%\Logs\%COMPUTERNAME%_System.evtx
wevtutil epl Security > \%COMPUTERNAME%\Logs\%COMPUTERNAME%_Security.evtx
wevtutil epl Application > \%COMPUTERNAME%\Logs\%COMPUTERNAME%_Application.evtx
```

These types of scripts can be run from a USB device or through remote sessions, thereby reducing the amount of interaction with the system.

`Chapter 5`, *Acquiring Host-Based Evidence*, introduced the tool `CyLR.exe` for the local acquisition of evidence. One of the key sets of evidence that `CyLR.exe` acquires is the Windows event logs. As was previously indicated, these log files can be acquired from the local system and exported to a USB. Another option that will be explored in this section is the use of `CyLR.exe` to acquire Windows event logs and forward them to the Skadi log review platform. Skadi will be addressed later on in this section, but first, `CyLR.exe` will be run against a system, and the output sent to the Skadi server.

To acquire the log files from a local system and send them to a Skadi instance, proceed as follows:

1. Open the Windows Command Prompt as administrator.
2. Navigate to the directory where the `CyLR.exe` file is located.
3. Enter the following command into the Command Prompt:

```
C:\Users\JSmith\Desktop>CyLR.exe -s 192.168.207.130:22 -u admin -p
password
```

   In the previous command, the `-s` is the IP address or domain name of the remote system where the `CyLR.exe` output is sent. In this case, this compressed evidence file will be sent to the system `192.168.207.130` via SFTP. The `-u` is the username of the account utilized to access the remote system, and, finally, `-p` is the password for the account related to the remote system.

4. Just as with a local acquisition, `CyLR.exe` will run, and the following will be visible in the Command Prompt:

```
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-Adminless%4Operational.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-Audit-Configuration-Client%4Operational.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-EnterpriseData-FileRevocationManager%4Operational.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-LessPrivilegedAppContainer%4Operational.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-Mitigations%4KernelMode.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-Mitigations%4UserMode.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-Netlogon%4Operational.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-SPP-UX-GenuineCenter-Logging%4Operational.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-Security-UserConsentVerifier%4Audit.evtx
Collecting File: C:\WINDOWS\System32\winevt\logs\Microsoft-Windows-SecurityMitigationsBroker%4Operational.evtx
```

This remote capture technique can be accomplished via any remote access tool available. The one distinct advantage of this method is the ability to acquire log data along with the other evidence that `CyLR.exe` captures, and automatically forward it to a central repository. This central repository can be the Skadi instance, or simply an SFTP server that is configured to accept this data.

Depending on the incident, there may be a significant amount of data. In fact, it may be too much for a responder to examine manually. In those cases, it is necessary to triage that data to determine what log entries are most important.

Triage

As discussed previously, depending on the incident, responders may be examining multiple Windows systems. Each of these systems may contain several thousand, or even a hundred thousand, event log entries. There is no possible way for a responder or team of responders to be able to examine that many individual entries. This equates to the often-used saying *it's like finding a needle in a haystack*. To address the large datasets that are often encountered in Windows event log analysis, responders can utilize the DeepBlueCLI tool. This PowerShell script, developed by Eric Conrad, detects suspicious Windows event log entries, such as service creation, account creation, a high number of logon failures, and malicious PowerShell usage. By focusing on these more critical event types, responders will be able to analyze more log files and potentially identify suspicious activity.

To run DeepBlueCLI, proceed as follows:

1. Download the PowerShell script from the GitHub site: `https://github.com/sans-blue-team/DeepBlueCLI`.
2. Open PowerShell, and navigate to the directory containing `DeepBlue.ps1`.
3. Execute the `DeepBlue.ps1` PowerShell script by pointing it to a specific Windows event log file—in this case, the Windows security event log, as shown here:

```
PS C:\Users\IRProactive-WKST\Desktop\DeepBlueCLI-master> .\DeepBlue.ps1 C:\Users\IRProactive-WKST\Desktop\evtx\Security.evtx
```

4. Enable the script, if necessary, by entering `R`, as shown here:

```
Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run C:\Users\IRProactive-WKST\Desktop\DeepBlueCLI-master\DeepBlue.ps1?
[D] Do not run  [R] Run once  [S] Suspend  [?] Help (default is "D"): R
```

After the script has run, it will produce a text output in the PowerShell window. In this case, the `DeepBlueCLI` script was run against Windows security event logs from a potentially compromised system. A review of the output indicated some potential signs that the system was attacked by an adversary, as can be seen in the following screenshot:

```
Date    : 2/15/2017 11:05:20 PM
Log     : Security
EventID : 4625
Message : High number of logon failures for one account
Results : Username: War_Machine
          Total logon failures: 2287
Command :
Decoded :

Date    : 2/15/2017 11:05:20 PM
Log     : Security
EventID : 4625
Message : High number of logon failures for one account
Results : Username: IIS_WPG
          Total logon failures: 2290
Command :
Decoded :

Date    : 2/15/2017 11:05:20 PM
Log     : Security
EventID : 4625
Message : High number of total logon failures for multiple accounts
Results : Total accounts: 7
          Total logon failures: 10006
```

From the preceding screenshot, there are three indicators of potential brute-forcing of credentials. The first is that the username `War_Machine` had attempted a total of 2287 logons that were failures. Second, the username `IIS_WPG` executed a total of 2290 logon failures. Finally, the output indicated there were a total of seven accounts that had a high number of failures. This data allows responders to focus their attention on the event ID 4625 and those accounts, for potential evidence of a compromise.

`DeepBlueCLI` also provides data associated with account creations. As previously discussed, the event ID 4720 indicates that an account has been created on the system. For example, if the responder has indications that the system under examination has been infected with Carbanak, they may be able to see if any new accounts had been created on the system. The following is a sample output:

```
Date    : 2/22/2017 6:35:08 PM
Log     : Security
EventID : 4720
Message : New User Created
Results : Username: MBadegain
          User SID: S-1-5-21-2865824651-146060924-1132756725-1019

Command :
Decoded :

Date    : 2/22/2017 6:34:55 PM
Log     : Security
EventID : 4720
Message : New User Created
Results : Username: BFernandez
          User SID: S-1-5-21-2865824651-146060924-1132756725-1018

Command :
Decoded :

Date    : 2/22/2017 6:34:27 PM
Log     : Security
EventID : 4720
Message : New User Created
Results : Username: MMartin
          User SID: S-1-5-21-2865824651-146060924-1132756725-1017
```

From the output, there are three new accounts that have been created. Each follows a very similar naming convention. From here, the responder can check with the access management team or system administrator if these names are legitimate. In the event that there is an indication that one or more of these names are not legitimate, the responder now has a data point from which to pivot and conduct other searches.

The one major drawback with Windows event log triage tools such as DeepBlueCLI is that they often focus on finding the *low-hanging fruit*, or those event logs that are often clearly indicative of abuse. They do make it easier for responders to triage a larger dataset, but responders should be aware that they are not foolproof, and there is a chance that an event log entry or entries may be missed.

Once a responder has been able to narrow down the entries that are important, there are several tools and techniques that can be leveraged to analyze the important log files.

Analysis

As was highlighted, the use of triage tools is a useful first step, but any incident investigation where event logs are available will require the use of specialized tools to dig deeper into the data that they provide. The Windows operating system has a native event log viewer. In the experience of many responders, that viewer is more suited to limited troubleshooting than to a deep analysis of the event logs. There are several tools, either open source or commercial, that can be leveraged for event log analysis. SIEM tools provide one of the best types of tools, especially if they have the ability to analyze offline event logs or those logs that have been acquired through scripts or other tools. In this chapter, two tools will be discussed: Event Log Explorer and Skadi. Each of these tools is useful for event log analysis but has its own unique features that make it suited for different aspects of event log analysis.

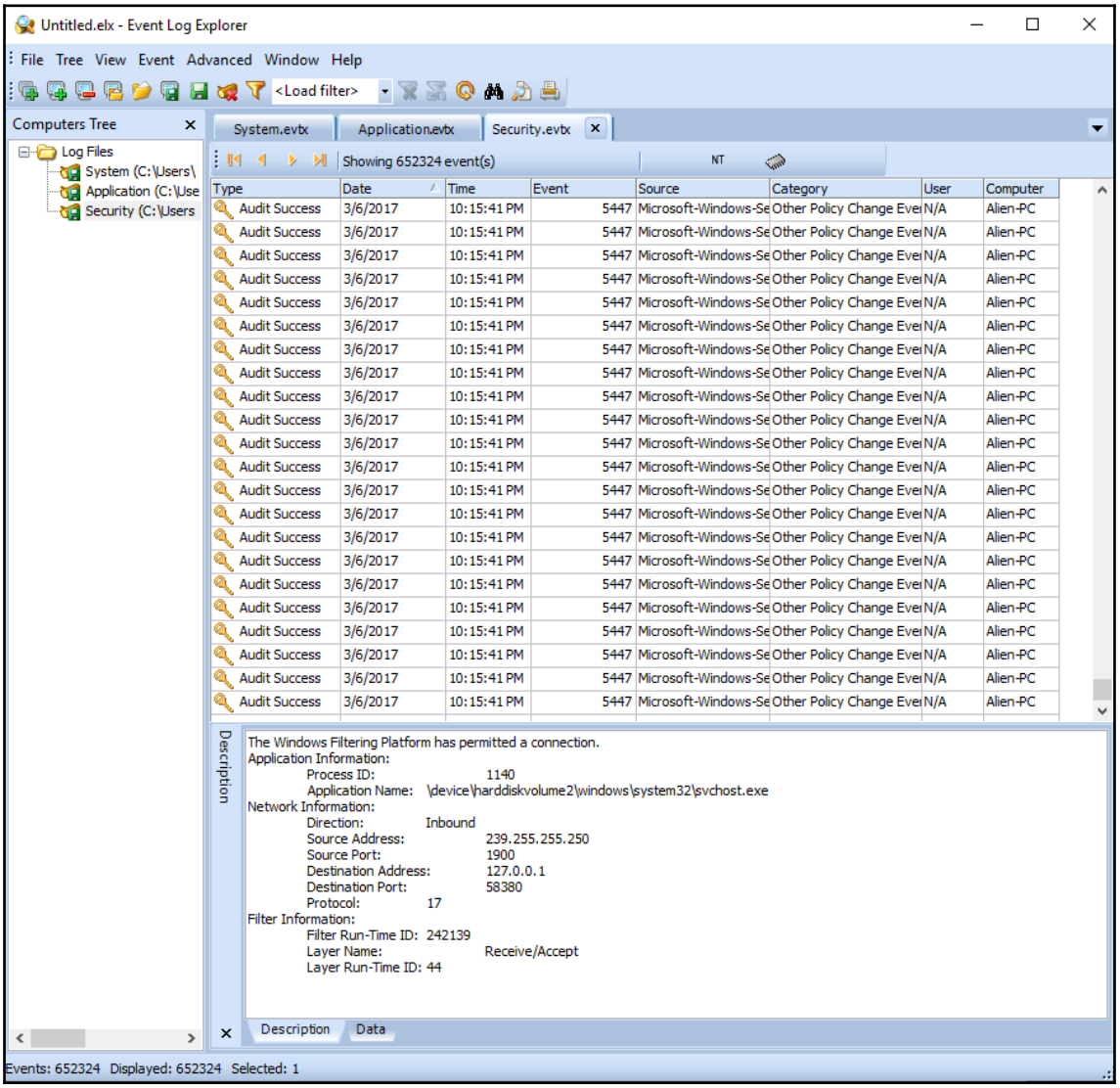
For example, Event Log Explorer allows for better filtering of results, along with its string searching ability. Event Log Explorer also has the ability to combine multiple sources. Other tools, such as Skadi, allow for the remote acquisition of log files and also combine log entries with other data, such as master file table entries and registry key settings. The one drawback with Skadi is the time necessary to ingest and process the data for review. It is therefore up to the responder to choose which tool best fits the incident under investigation.

Event Log Explorer

Event Log Explorer is an event log analysis tool that has more features and has an easy-to-navigate GUI. Available as a commercial tool, the creators of Event Log Explorer, FSPRo Labs, provide a 30-day trial period in which to test the tool. The tool can be downloaded from the website at `https://eventlogxp.com/` and can be installed on the Windows OS.

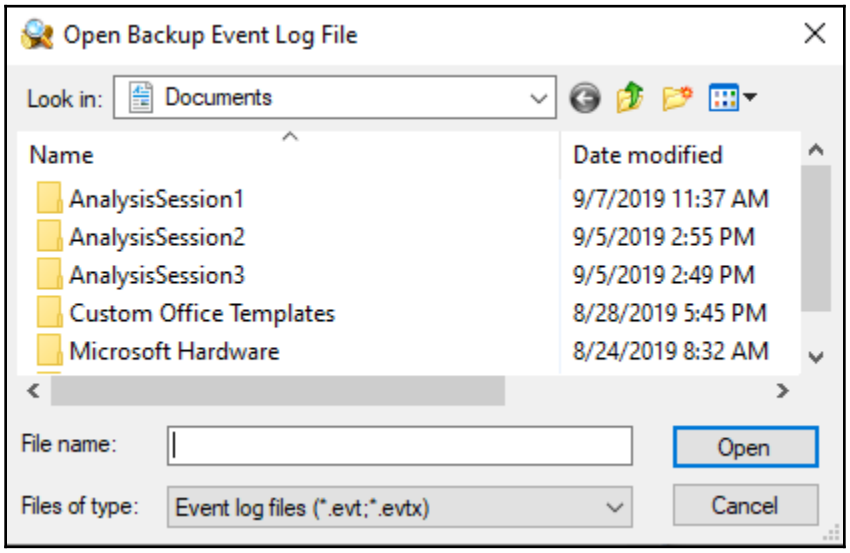
To begin an analysis of offline event logs, proceed as follows:

1. Open Event Log Explorer. The following window will appear:



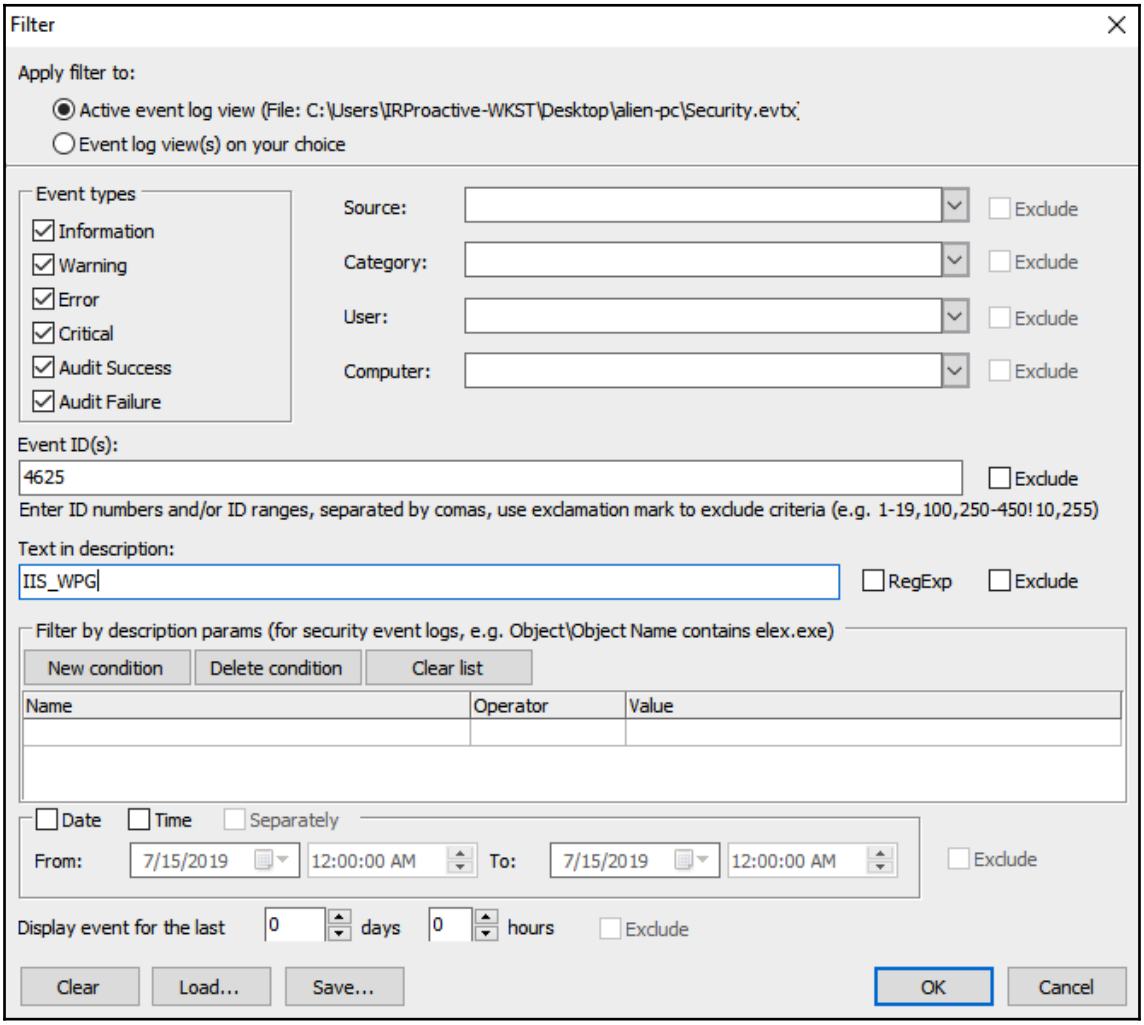
The GUI has three main areas. The center pane contains the individual log entries that are contained within the Windows event log type. The lower pane contains the details contained within each log entry. Finally, the left-hand pane includes the Windows event log types that are under analysis.

2. Event Log Explorer will automatically import the localhost's Windows event logs. To remove these logs, right-click on the computer name, and click **Remove Computer**. Click **YES**. This will remove the existing Windows event logs.
3. To import an event log file or files, click on **File | Open Log File | Standard**. From here, load the log file from a directory and click **Open**, as shown here:

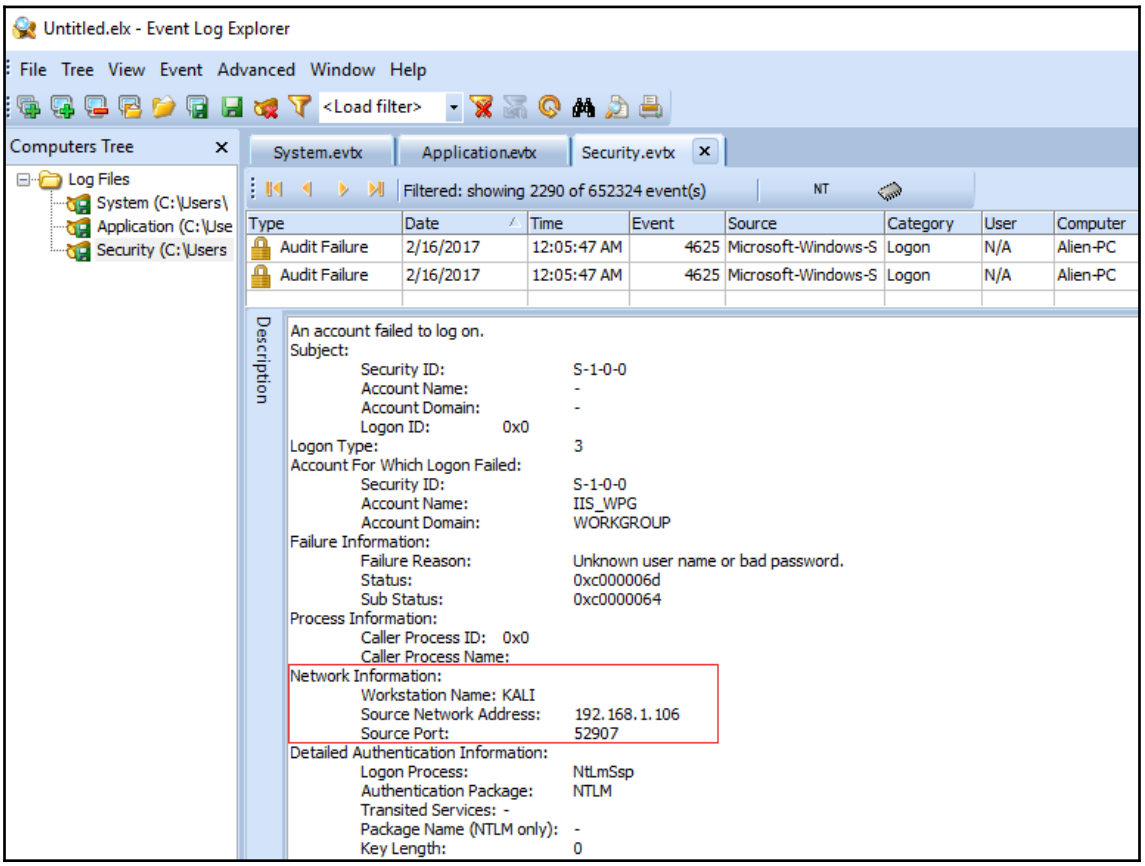


4. Once the log file or files have been loaded, the responder can utilize the filter to focus on specific data contained within the log files. To open the filter, look for the filter icon on the taskbar:

5. The filter screen will then open. From here, the responder can filter the event logs on a variety of specific attributes. This includes the event types, event ID, and even keyword searching in the text of the log file entry. In this case, the responder will examine the log entries for the failed logins, event ID 4625, that include the username `IIS_WPG` that had been identified with the `DeepBlueCLI` triage script. The responder enters the event ID as `4625`, and in the **Text in description** field, the plaintext of the account name, `IIS_WPG`, as follows:



6. The output, after clicking **OK**, shows the event log entries that match the filters that were entered. An examination of the details of the event log entries indicates that the workstation that was attempting to use those credentials was named `Kali`, with the IP address `192.168.1.106`, as shown here:



In this brief scenario, the responder was able to take data from the Windows security event logs and extract two very important data points: first, the hostname, Kali (a well-known penetration testing platform), as well as the IP address of the attacking system. They can now use this information to examine other log sources for that IP address and hostname, to discover any additional information concerning that system's activity.

Event Log Explorer has a great deal of functionality that cannot be addressed in this volume. Some of the other features include building custom views, filtering on specific data points, and finding text within log entries across multiple event log files. Even with these features, Event Log Explorer does have some minor limitations. First, responders have to gather the logs onto the system for analysis and load them manually. The second is that depending on the file size, Event Log Explorer may have performance issues, including freezing. Responders should ensure they do not overload the application. Regardless, Event Log Explorer is an excellent tool for responders to include in their toolkit.

Analyzing logs with Skadi

Incidents often involve multiple systems across an enterprise network. Correlating this activity is often very difficult without analyzing the event logs from multiple systems. This is where the previously discussed SIEM appliances are really helpful. Another option, if the SIEM is not preconfigured to ingest and analyze event logs, is the Skadi platform. This open source platform, available from GitHub at `https://github.com/orlikoski/Skadi`, is a group of applications and forensics installed on an Ubuntu 16.04 LTS server base image.

The primary tool that this chapter will focus on is the Elastic Stack that is included as part of the Skadi platform. The other major feature that Skadi offers is the ability to ingest logs and other forensic data that is acquired through `CyLR.exe`. As previously discussed, `CyLR.exe` can be configured to send its output via SFTP to a remote system. Skadi combines an additional tool with `CyLR.exe`, to produce a dataset that is ingestible by the Elastic Stack on Skadi. This feature allows responders to run `CyLR.exe` on several different systems and have it sent directly to Skadi, where it can then be processed, indexed, searched, and correlated.

For example, an initial investigation of an incident has identified a system called `alien-pc` that appears to have been compromised. Responders deploy `CyLR.exe` to the system, and run the following command:

```
C:\Users\alien-pc\Desktop>CyLR.exe -s 192.168.49.132:22 -u skadi -p skadi
```

In the previous command, the `-s` is the IP address for the Skadi server. The default username and password for the Skadi server is `skadi`. After `CyLR.exe` has completed, the responder will then log in to the Skadi console. From here, the **CDQR** (short for **Cold Disk Quick Response**) tool will be run, to convert the data acquired into a format that can be ingested by the Elasticsearch tool. The following command starts the processing with CDQR:

```
skadi@skadi:~$cdqr in:alien-pc.zip out:Results -z -max_cpu
```

The command produces the following output:

```
skadi@skadi:~$ cdqr in:alien-pc.zip out:Results -z --max_cpu
docker run   -v /etc/hosts:/etc/hosts:ro   --network host -v /home/skadi/alien-pc.zip:/home/skadi/al
ien-pc.zip -v /home/skadi/Results:/home/skadi/Results aorlikoski/cdqr:4.4.0 -y /home/skadi/alien-pc.
zip /home/skadi/Results -z --max_cpu
```

CDQR produces a file called `alien-pc.plaso` in the `Results` directory. The results of the `CyLR.exe` file have not been converted into a format for ingestion by Elasticsearch. CDQR is used again in the next step, with the following command:

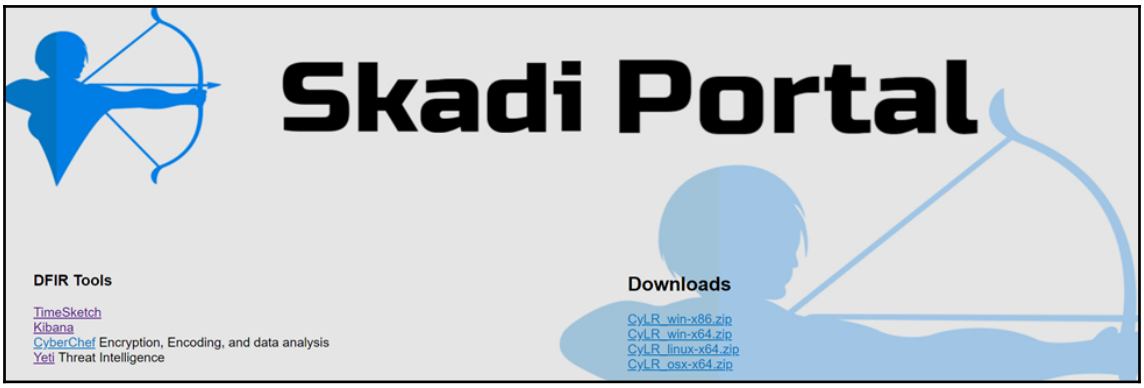
```
skadi@skadi:~$cdqr in:Results/alien-pc.plaso -plaso_db -es_kb winevt
```

Upon completion, the `CyLR.exe` file that had been converted to the `.plaso` file has been processed and sent to the Elasticsearch portion of the Elastic Stack, as can be seen here:

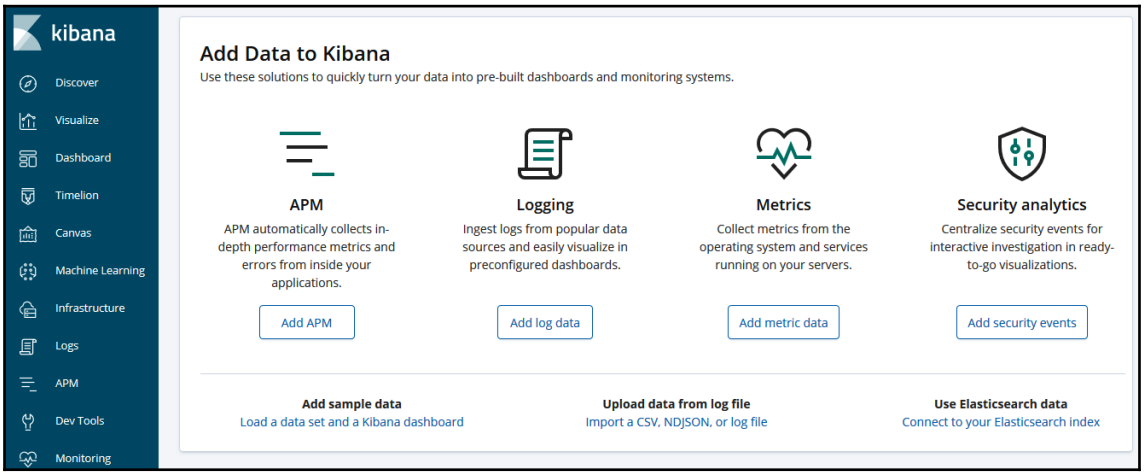
```
skadi@skadi:~$ cdqr in:Results/alien-pc.plaso --plaso_db --es_kb winevt
docker run   -v /etc/hosts:/etc/hosts:ro   --network host -v /home/skadi/Results/alien-pc.plaso:/hom
e/skadi/Results/alien-pc.plaso aorlikoski/cdqr:4.4.0 -y /home/skadi/Results/alien-pc.plaso --plaso_d
b --es_kb winevt
CDQR Version: 4.4
Plaso Version: 20190131
WARNING!! Known compatible version of Plaso NOT detected. Attempting to use default parser list. Try
 using the --no_dependencies_check if Plaso dependancies are the issue.
Using parser: win
Number of cpu cores to use: 1
Destination Folder: Results
Source data: /home/skadi/Results/alien-pc.plaso
Log File: Results/alien-pc.plaso.log
Database File: Results//home/skadi/Results/alien-pc.plaso

Total start time was: 2019-07-16 01:51:32.335461
WARNING: File must be plaso database file otherwise it will not work.  Example: artifact.plaso (from
 CDQR)

Process to export to ElasticSearch started
Exporting results in Kibana format to the ElasticSearch server
"psort.py" "-o" "elastic" "--status_view" "linear" "--index_name" "case_cdqr-winevt" "--server" "127
.0.0.1" "--port" "9200" "/home/skadi/Results/alien-pc.plaso"
All entries have been inserted into database with case: case_cdqr-winevt

Process to export to ElasticSearch completed
ElasticSearch export process duration was: 0:13:03.303455
```

After the process has completed, the results can be viewed in the Kibana GUI, as follows:

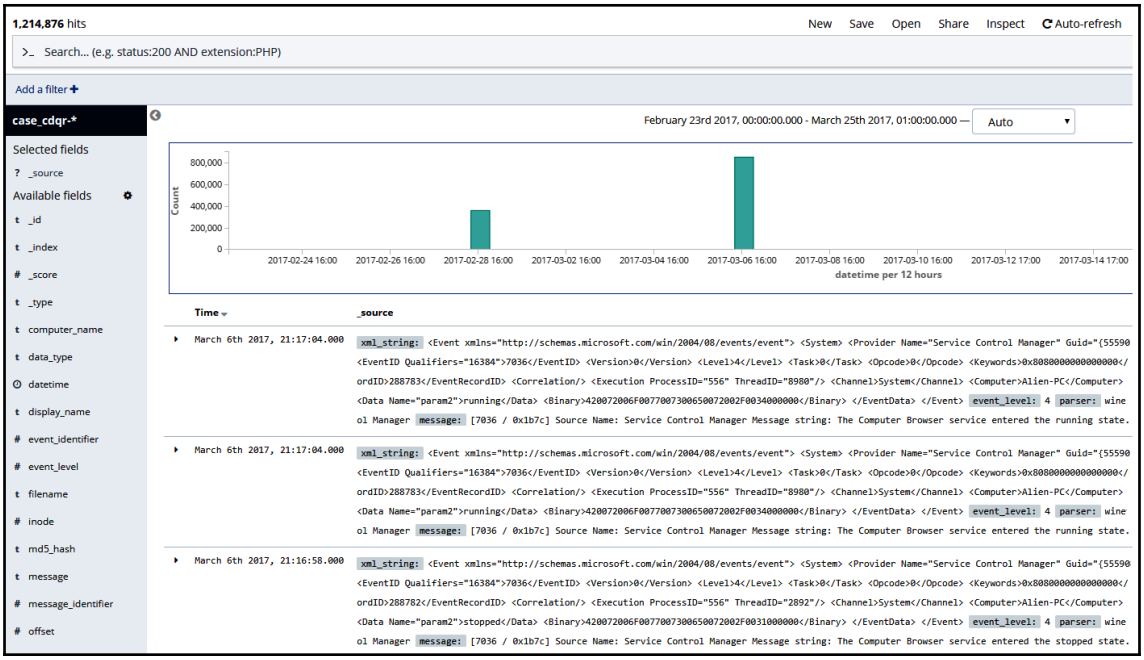
1. Navigate to the IP address of the Skadi server, and the portal shown in the following screenshot will appear:



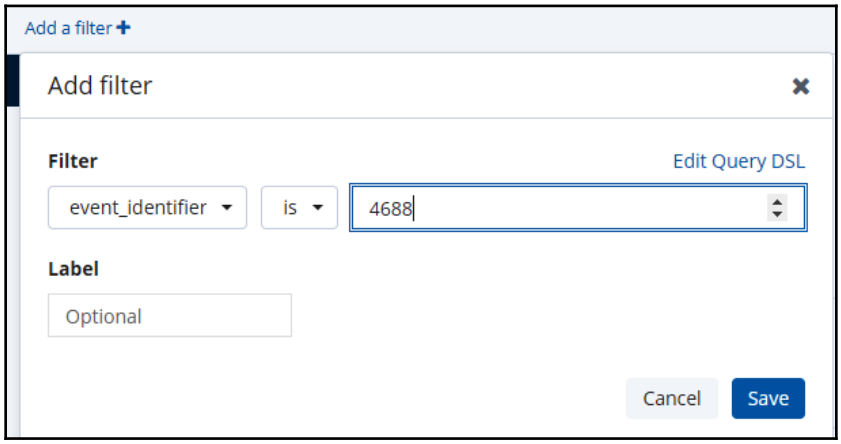
2. Click on **Kibana**, and the following screen appears:



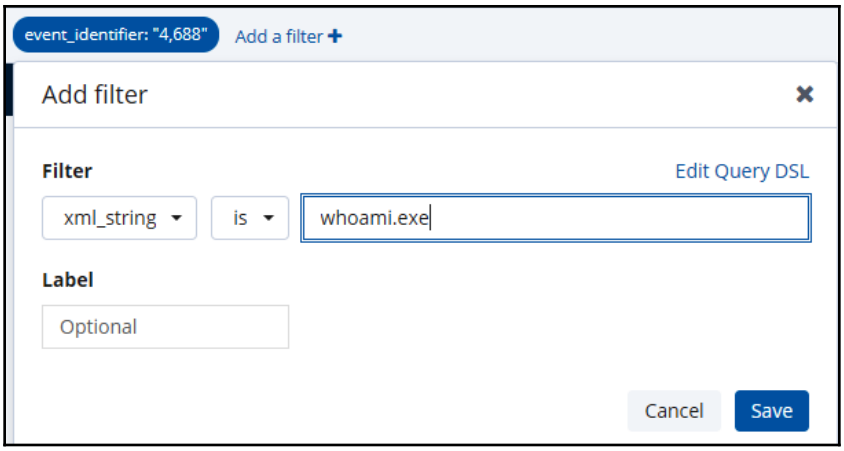
3. From here, click **Discover**.
4. In the top right-hand corner, set the date to an appropriate time period. Kibana defaults to the last 15 minutes of data. For offline data, set the time period that is applicable or simply click **5 years**, as follows:



5. Kibana is feature-rich and provides a wide range of options in terms of analyzing data. This includes the use of customer queries, event IDs, keywords, and XML strings. In this case, the responder will focus on event ID 4688, which indicates a new process has run to identify potentially malicious code or other behavior. The **Add a filter** feature in the left-hand pane allows responders to filter on specific data points, as shown in the following screenshot:



6. Once the filter is set on the event ID `4688`, the responder clicks **Save** and then refreshes Kibana by clicking on **Refresh** in the upper-right corner. This filters on that specific event ID, taking the results of almost 1.3 million events to 784.
7. From here, the responder begins a cursory search of the results and identifies the `whoami.exe` executable . This is not a malicious executable but is suspicious, as it may indicate that an adversary has compromised the system and then attempted to ascertain the name of the system using the Windows `whoami.exe` file. The filter can then be further enhanced by including this keyword in the filter, as follows:



8. This additional filter produces a log entry that can be analyzed by the responder, as follows:

```
t  xml_string  <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
                 <System>
                   <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}"/>
                   <EventID>4688</EventID>
                   <Version>1</Version>
                   <Level>0</Level>
                   <Task>13312</Task>
                   <Opcode>0</Opcode>
                   <Keywords>0x8020000000000000</Keywords>
                   <TimeCreated SystemTime="2017-03-07T05:11:01.568600000Z"/>
                   <EventRecordID>9505938</EventRecordID>
                   <Correlation/>
                   <Execution ProcessID="4" ThreadID="68"/>
                   <Channel>Security</Channel>
                   <Computer>Alien-PC</Computer>
                   <Security/>
                 </System>
                 <EventData>
                   <Data Name="SubjectUserSid">S-1-5-21-2865824651-146060924-1132756725-500</Data>
                   <Data Name="SubjectUserName">Administrator</Data>
                   <Data Name="SubjectDomainName">Alien-PC</Data>
                   <Data Name="SubjectLogonId">0x000000000162e98c</Data>
                   <Data Name="NewProcessId">0x0000000000001a68</Data>
                   <Data Name="NewProcessName">C:\Windows\System32\whoami.exe</Data>
                   <Data Name="TokenElevationType">%%1936</Data>
                   <Data Name="ProcessId">0x0000000000001a0c</Data>
                   <Data Name="CommandLine"/>
                 </EventData>
               </Event>
```

9. From here, the responder can determine the date, time, and the account that ran the command. There may be a legitimate reason for an administrator to run this command in their normal day-to-day activities, but it is possible that this was part of a larger compromise, and the responder should verify the use of this tool.
10. The combination of `CyLR.exe` and Skadi provides responders the ability to remotely access log files and automate their ingestion into a platform that allows for a more detailed analysis. Skadi can also be stood up as necessary, as it can be configured as a virtual machine. From here, responders can forward all of the necessary log files, and correlate activity across multiple systems in a short time. This combination is powerful and comes at a low cost.

Skadi, combined with CyLR.exe, provides the responder with the ability to acquire and analyze log files from a number of systems involved in the incident. The ability to pivot off of specific Event IDs or keywords makes Skadi a powerful tool to zero in on specific log entries that are important to identifying additional evidence important in a incident investigation.

Summary

At the heart of log analysis is the assumption that actions by an adversary will leave a trace. Just as in the physical world, responders' ability to see these traces is based upon the tools and techniques that are used. This chapter explored the foundational elements of logs and log management, provided tools such as SIEM to aggregate and review these logs, and finally, looked at the tools and techniques to examine the most prevalent logs that originate from the Windows OS. This chapter has really only scratched the surface with regard to how logs play an integral part in an incident investigation.

In keeping with the theme of understanding the traces of an adversary attack, the next chapter will examine the role that malware analysis plays in incident response.

Questions

1. For effective log management, an organization should establish logging as a normal business practice.

   A) True
   B) False

2. Which is not one of the functions of a SIEM?

   A) Log retention
   B) Automated response
   C) Alerting
   D) Log aggregation

3. Which of these is not part of the Elastic Stack?

   A) Kibana
   B) Elasticsearch
   C) Log response
   D) Logstash

4. Locard's exchange principle basically states that when two objects come into contact with each other, they leave traces.

   A) True
   B) False

Further reading

- Windows Security Log Events, at `https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/`
- Graylog, at `https://github.com/Graylog2`
- Skadi, at `https://github.com/orlikoski/Skadi`

11
Writing the Incident Report

An incident response team functions in much the same way that a fire department does. Both teams take time to prepare themselves with training on their respective techniques, tools, and practices, and they can respond at a moment's notice to a fire or an incident. During their response to a fire, the firefighters take notes and record their actions, ensuring that critical decisions are documented and that individual contributions are noted. Once the fire is out, they sift through the debris to determine what the causes and origins of the fire were. Once the proper documentation has been prepared, the fire department conducts an after-action review in order to critique their performance and find avenues for improvement. Other reports allow fire departments and safety experts to update building codes and improve the survival of structures should a fire break out.

Incident response teams utilize much of the same workflow. During an incident, notes are taken and actions recorded. Evidence is obtained from systems and maintained in a forensically sound manner. A root cause analysis is conducted utilizing the notes, observations, and evidence obtained during the incident. This root cause analysis is utilized by information technology personnel to patch up vulnerabilities and further harden systems. Finally, the team conducts its own after-action review where the series of events is laid out and critiqued so that the team may improve their processes, their tools, and their techniques, as well as make any corrections to the incident response plan.

To maximize the benefits of the root cause analysis and after-action brief, incident responders will need to ensure that all of their actions are recorded in the proper fashion. They will also be required to prepare several documents that senior leaders and decision makers will use when considering the future state of the IT infrastructure. To better prepare responders to craft the necessary documentation, the following topics will be addressed:

- **Documentation overview**: This overview will cover the various elements of preparing reports, including what data to capture, the audience that will review the reports, and the sources that responders can draw upon in crafting incident documentation.

- **Incident tracking**: For organizations that conduct a routine response to incidents, tracking software is useful in capturing actions and relevent data. In this case, the **Fast Incident Response** (**FIR**) tracking system will be explored.
- **Written reports**: Depending on the severity or complexity of an incident, a written report will be prepared. By crafting a well-written and thoughtful report to senior managers and external stakeholders, incident responders can provide these key decision makers with an accurate picture of what happened and how to prevent it in the future.

Documentation overview

The documentation associated with an incident takes several forms. The length of any documentation is often dictated by the type of incident. Simple incidents that take very little time to investigate and have a limited impact may be documented informally in an existing ticketing system. However, in more complex incident investigations, such as a data breach that has led to the disclosure of confidential information (such as medical records or credit card information), you may require extensive written reports and supporting evidence.

What to document

When looking at documenting an incident, it is not very difficult to ascertain what should be documented. Following the five *Ws (Who, What, Where, When, and Why)*, and sometimes *How?*, is an excellent foundation when considering what to document during an incident. Another good piece of wisdom when discussing documentation, especially when discussing the legal implications of security incidents, is the axiom that if you didn't write it down, it didn't happen. This statement is used to drive home the point that proper documentation is often comprised of as much detail that the incident response analyst can bring. Analysts may be involved in an incident that ends up in a civil proceeding. The wheels of justice often move slowly, and an analyst may be called to the witness stand after 18 months, during which 10 other incidents may have transpired. Having as much detail available in the incident reporting will allow analysts to be able to reconstruct the events in the proper manner.

An excellent example of using these five Ws (and one H) structure in your documentation is when looking at a digital forensics task, such as imaging a hard drive. In `Chapter 6`, *Forensic Imaging*, proper documentation was partially addressed when we looked at the practice of taking photos of the suspect drive. The following is a more detailed record of the event:

- **Who**: This is the easiest detail to make a note of. Simply, who was involved in the process? For example, the person involved was analyst Jane Smith.
- **When**: Record the date and time that the imaging began and when it ended. For example, the imaging process was started at 21:08 UTC on August 16, 2019, and ended at 22:15 UTC on August 16, 2019. Times are critical, and you should ensure that a standard time zone is utilized and indicated in the report.
- **Where**: This should be a detailed location, such as an office.
- **What**: The action that was performed; for example, acquiring memory or firewall logs or imaging a drive.
- **Why**: Having a justification for the action helps in understanding the reason why the action was performed.
- **How**: A description of how an action is performed should be included. Additionally, if an incident response team utilizes playbooks or standard operating procedures as part of their plan, this should be included. Any departure from the standard operating procedures should also be similarly recorded.

Putting all this information together, the following sample language can be entered into the report:

On August 16, 2019, analyst Jane Smith arrived at office 217 of the Corporate Office Park located at 123 Maple St., Anytown, US, as part of the investigation. Upon arrival, Smith took control of the Dell laptop, asset tag #AccLT009, serial #7895693-862. An alert from the firewall IDS/OPS indicated that the laptop had communicated with a known Command and Control server. The laptop was to be imaged in an attempt to ascertain whether it had been infected with malware. At 21:08 UTC, Smith imaged the drive utilizing the live imaging technique in accordance with the Standard Operating Procedure IR-002. The process was completed at 22:15 UTC on August 16, 2019.

This entry provides sufficient detail to reconstruct the events that transpired. Taken together with other documentation, such as photographs and the chain of custody, the analyst has a clear picture of the process and the outcome.

Types of documentation

There is no one standard that dictates how an incident is documented, but there are a few distinct categories. As was previously stated, the depth of the documentation will often depend on the type, scale, and scope of an incident; however, in general, the following categories apply:

- **Trouble ticketing system**: Most enterprise organizations have an existing ticketing system utilized to track system outages and other problems that normally arise in today's network infrastructure. These systems capture a good deal of data associated with an incident. An entry usually captures the start and stop date and time, the original reporting person, and the action performed, and also provides an area for notes. The one major drawback to ticketing systems is that they were originally designed to support the general operations of enterprise infrastructures. As a result, more complex incidents will require much more documentation than is possible in these systems. Due to this, they are often reserved for minor incidents, such as isolated malware infections or other such minor incidents that are disposed of quickly.
- **Incident response orchestration**: Some organizations have seen the need for a dedicated incident response platform and have come up with applications and other types of infrastructure that support incident response teams. These incident response orchestration platforms allow analysts to input data, attach evidence files, and collaborate with other team members, as well as pull in outside resources, such as malware reverse engineering and threat intelligence feeds.

  There are several of these platforms available both commercially and as freeware. The main advantage of these platforms is that they automate the capture of information, such as the date, time, and analyst's actions.

  Another distinct advantage is that they can limit who is able see the information to a select group. With ticketing systems, there is the possibility that someone without authorization will observe details that the organization may want to keep confidential. Having an orchestration system can provide a certain level of confidentiality. Another key advantage is the ability for team members to see what actions are taken and what information is obtained. This cuts down on the number of calls made and the possibility of miscommunication.

- **Written reports**: Even with automated platforms in use, some incidents require extensive written reporting. In general, these written reports can be divided into three main types. Each of the following types will be expanded on later in this chapter:
    - **Executive summary**: The executive summary is a one- to two-page report that is meant to outline the high-level bullet points of the incident for the senior management. A brief synopsis of the events, a root cause, if it can be determined, and remediation recommendations are often sufficient for this list.
    - **Incident report**: This is the detailed report that is seen by a variety of individuals within the organization. This report includes the details of the investigation, a detailed root cause analysis, and thorough recommendations on preventing the incident from occurring again.
    - **Forensic report**: The most detailed report that is created is the forensics report. This report is generated when a forensic examination is conducted against the log files, captured memory, or disk images. These reports can be very technical, as they are often reviewed by other forensic personnel. These reports can be lengthy, as outputs from tools and portions of evidence, such as log files, are often included.

Having an understanding of the various categories that comprise an incident report allows responders to properly organize their material. Even smaller incidents create documentation, meaning that responders can become overwhelmed. Coupled with the high number of data sources, the reporting process can become a chore. To make the process flow better, responders should be prepared to address the various categories at the onset of an incident and organize their documentation accordingly.

Sources

When preparing reports, there are several sources of data that are included within the documentation, whether the incident is small, requiring only a single entry into a ticketing system, all the way to a complex data breach that requires extensive incident and forensic reporting. Some sources include the following:

- **Personal observations**: Users may have some information that is pertinent to the case. For example, they might have clicked on a file in an email that appeared to come from a legitimate address. Other times, analysts may observe behavior in a system and make a note of it.

- **Applications**: Some applications produce log files or other data that it may be necessary to include in a report.
- **Network/host devices**: A great deal of this book deals with acquiring and analyzing evidence from a host of systems in an enterprise environment. Many of these systems also allow for outputting reports that can be included with the overall incident or forensic reporting.
- **Forensic tools**: Forensic tools often have automated reporting functions. This can be as simple as an overview of some of the actions, as was addressed in the previous chapters, or the actual outputs, such as file hashes, that can be included within a forensic report.

Wherever the material comes from, a good rule to follow is to capture and include as much as possible in the report. It is better to have more information than less.

Audience

One final consideration to bear in mind when preparing your documentation is who will read an incident report versus a detailed forensic report. In general, the following are some of the personnel, both internal and external to an organization, that may read the reports associated with an incident:

- **Executives**: High-profile incidents may be brought to the attention of the CEO or CFO, especially if they involve the media. The executive summary may suffice, but do not be surprised if the senior leadership requires a more detailed report and briefing during and at the conclusion of an incident.
- **Information technology personnel**: These individuals may be the most interested in what the incident response analysts have found. Most likely, they will review the root cause analysis and remediation recommendations very seriously.
- **Legal**: In the event that a lawsuit or other legal action is anticipated, the legal department will examine the incident report to determine whether there are any gaps in security or the relevant procedures for clarification. Do not be surprised if revisions have to be made.
- **Marketing**: Marketing may need to review either the executive summary or the incident report to craft a message to customers in the event of an external data breach.

- **Regulators**: In regulated industries, such as healthcare and financial institutions, regulators will often review an incident report to determine whether there is a potential liability on the part of the organization. Fines may be assessed based upon the number of confidential records that have been breached, or if it appears that the organization was negligent.
- **Law enforcement**: Some incidents require law enforcement to become involved. In these cases, law enforcement agencies may require copies of incident and forensics reports for review.
- **Outside support**: There are some instances where the need to bring in outside forensics or incident response support becomes necessary. In these cases, the existing reports would go a long way in bringing these individuals up to speed.

Understanding the audience gives incident response analysts an idea of who will be reading them. Understand that the report needs to be clear and concise. In addition to this, technical details may require some clarification for those in the audience that do not have the requisite knowledge or experience.

Incident tracking

Many organizations utilize an IT trouble ticket -tracking system for incidents such as ServiceNow and Jira. While tracking incidents in this manner works when there are few incidents and there is no need to automate incident response processes. Organizations that have a more robust incident response capability may need to utilize an incident response platform to track and assist with the proper execution and documentation of the incident response process.

Commercial solutions for tracking cybersecurity incidents generally fall into the category of **Security Orchestration Automation Response** (**SOAR**) platforms. Some of these solutions can integrate with other toolsets so that a good deal of the process is automated. They may also incorporate playbooks into the platform so that incident response teams have immediate access and can work through the process, all the while recording their actions on the platform itself. One of the most significant advantages of a dedicated platform is the ability to track incidents over time and gain a sense of what types of attacks the organization is dealing with.

Fast Incident Response

The Société Genéralé CERT has put together a platform titled FIR as a freeware tool to aid incident response teams with the creation, tracking, and reporting of incidents.

This web application can allow anyone within an organization to create incidents, make notes, and track incidents to completion. This tool provides a good deal of features that may make it a very handy tool for incident response teams that have budgetary considerations.

The tool utilizes a MySQL backend database and a combination of Python, Django, and Bootstrap to create a web application environment where analysts and other users can input data, as well as perform queries. Another key feature of the FIR application is the ability to customize fields to fit the organization. FIR can be installed either in a Docker container or installed on a Linux system, such as Ubuntu.

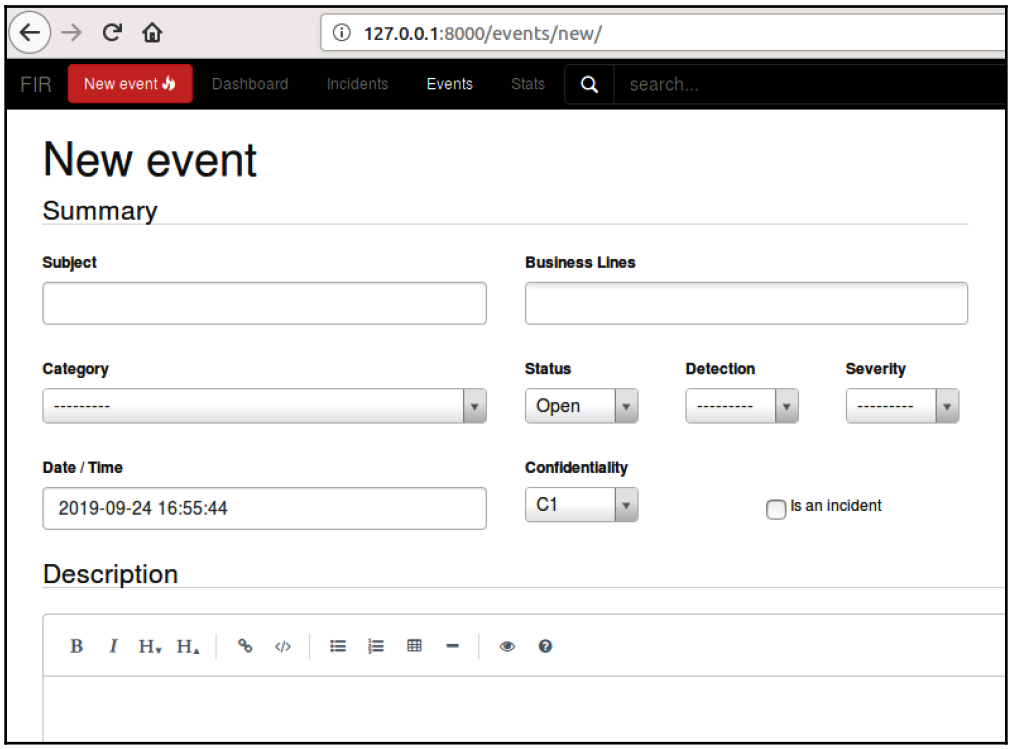
FIR is available for both a development and production environment. The installation of either option is based on the size of the organization and how often data will be put into the system. A complete build guide is available at `https://github.com/certsocietegenerale/FIR/wiki/Setting-up-a-development-environment`.

For the purposes of this book, we will give you a review of how to customize the platform and how to record an incident. There are other options that can be utilized to tailor the FIR platform for a particular CSIRT. The FIR site at `https://github.com/certsocietegenerale/FIR/wiki/User-manual` has additional information that is useful.

To create a new incident without any modifications, go through the following steps:

1. Once FIR is installed, navigate to the login screen by entering `http://localhost:8000` in the URL bar in a web browser. The sign-in form will appear. For the development environment, sign in using `admin/admin`:

2. After logging in, the dashboard will be empty as there are no incidents to record. Click on the **New event** button in the upper-left corner. The following window will appear:

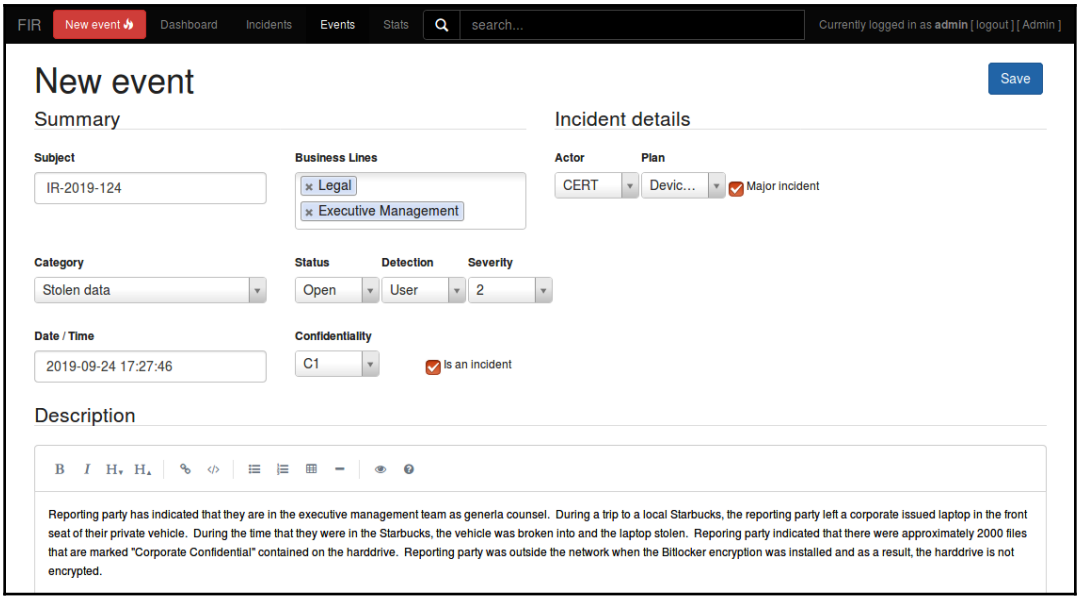


3. Within the form, there are several fields that can be utilized to record an incident:

   - **Subject**: This is a free text field that can take any plain text. For best practices, this would best be utilized for the individual incident number.
   - **Business lines**: This is one of the preconfigured fields that can be modified. Depending on the size of the organization, separating out incidents by business lines may show decision makers the security vulnerabilities within that department.
   - **Category**: FIR has a good number of incident categories preconfigured that cover the wide range of attacks that an incident response team could see. There is also the ability to add additional categories.

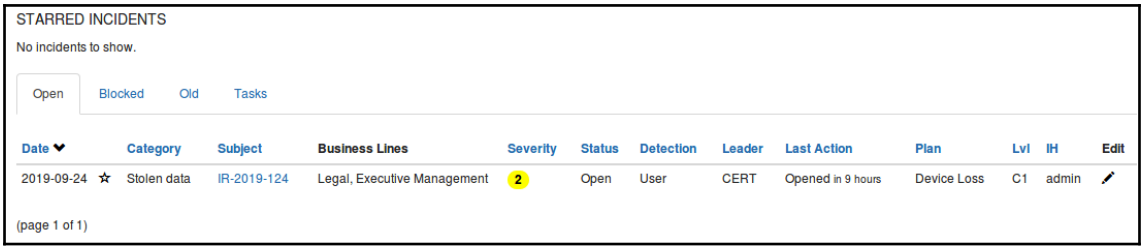
- **Status**: This indicates whether the incident is still open.
- **Detection**: This shows who or what the first entity to detect the incident was.
- **Severity**: FIR comes preconfigured with severity levels set from **1** to **4**.
- **Date/Time**: FIR automatically sets a date and timestamp for actions performed within the application. During configuration, you may need to modify the settings within the platform to change the time zone. The FIR installation instructions can assist with making that modification.
- **Confidentiality**: For organizations that have certain levels of confidentiality, this allows for a gradation from 0 to 3.

4. Create the incident by entering information into the specific fields. In this case, a laptop has been stolen and reported by the user to the incident response team (**CERT**). In this case, the reporting party has indicated that there are approximately 2,000 confidential files stored on an unencrypted hard drive:



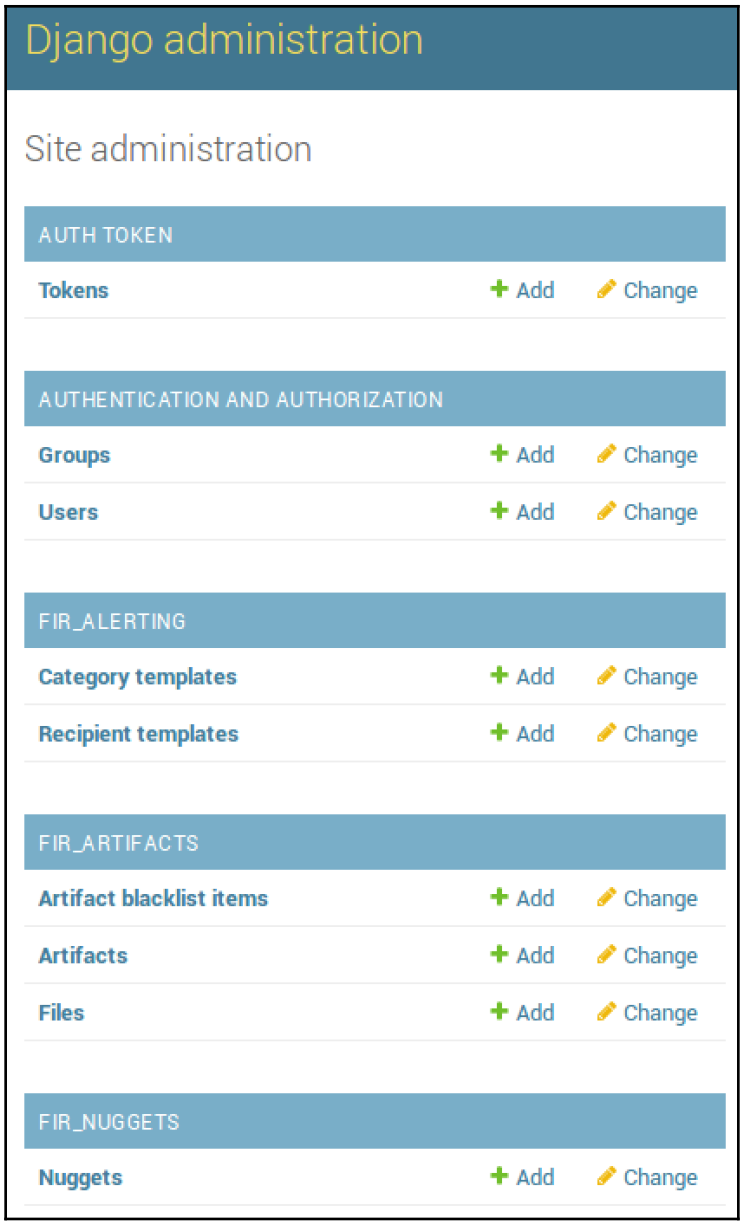
5. When the box for an incident is checked, two additional fields, **Actor** and **Plan**, appear. These are selections that can be modified to fit the organization. In this case, the actor is the **CERT** team and the plan will be the **Device Loss Playbook**. Once the fields are completed, click on **Save**.
6. FIR then opens another window with the incident information. Click on **Add** and **To-Do** in the lower portion of the window. This will open up the following:

7. In the **Task** field, enter in `Execute Device Loss Playbook` and select **CERT** under **Accountable**. Once done, click on the plus icon. This adds a task into the FIR system for a follow-up. Click on **Dashboard** and the incident will appear:

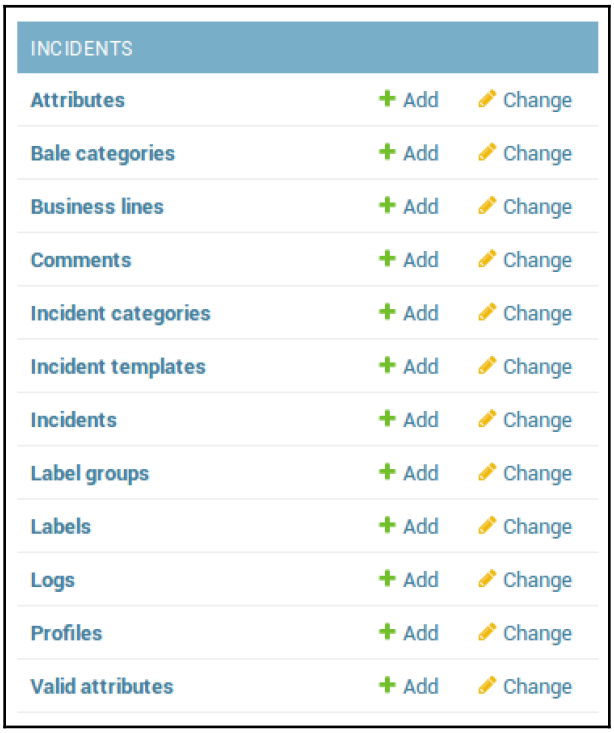


8. Click on **Tasks** and the task that was assigned to the **CERT** team can be seen:

Through the use of the FIR platform, an incident response team can have a single repository for the incident data, as well as the ability to assign specific tasks to individuals. To further enhance this capability, FIR allows the administrator of the system the ability to make modifications to fields such as the business units or actions. To access this, click on the **Admin** icon in the top-right corner of the window. This will open the configuration menu:

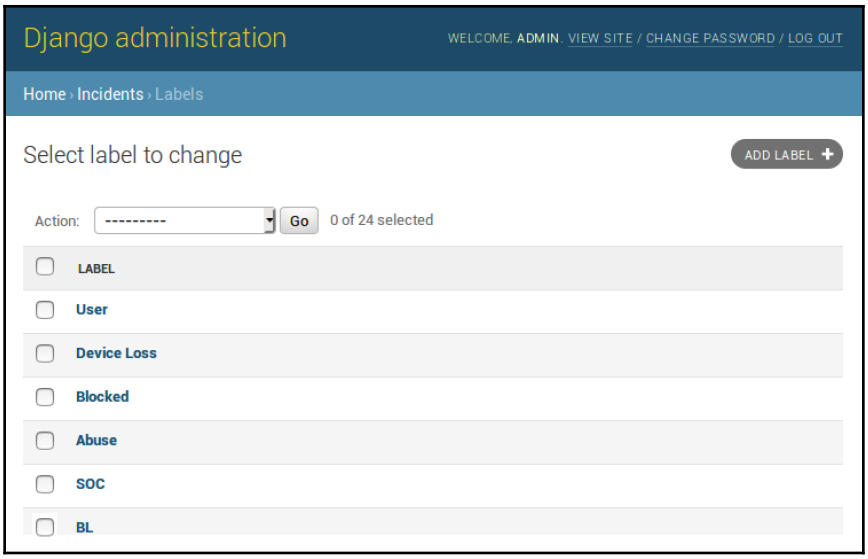

Django administration
Site administration
AUTH TOKEN
Tokens
Add
Change
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
FIR_ALERTING
Category templates
Add
Change
Recipient templates
Add
Change
FIR_ARTIFACTS
Artifact blacklist items
Add
Change
Artifacts
Add
Change
Files
Add
Change
FIR_NUGGETS
Nuggets
Add
Change

Many of these fields have not been configured yet, allowing the administrator to set specific types of alerting and artifacts. One area that the administrator may want to configure prior to utilizing is the incident information. Scrolling down, the following fields for incidents can be modified by the administrator:

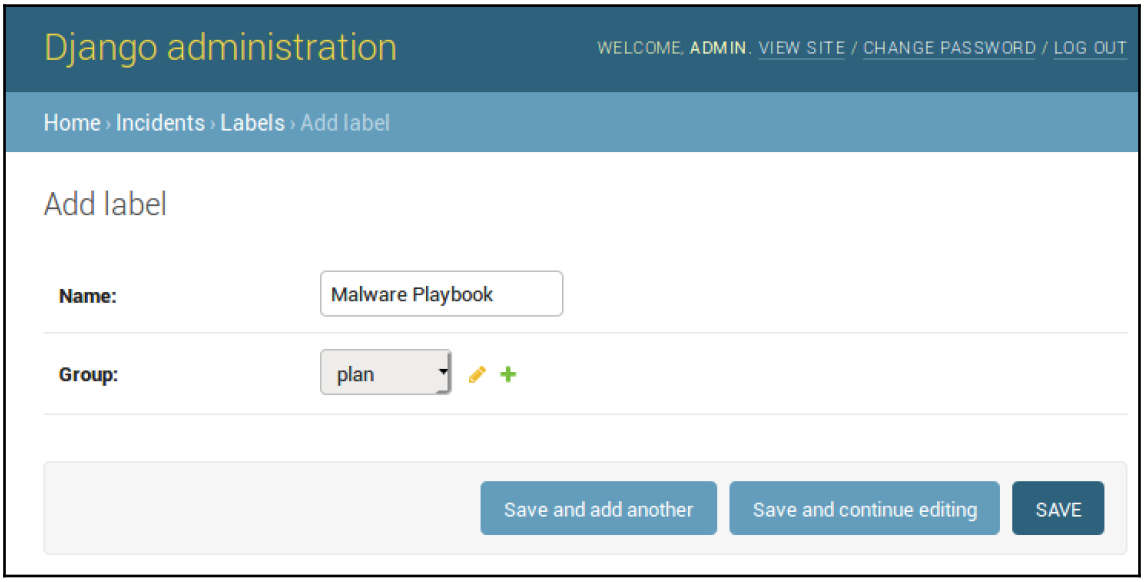


For example, suppose the administrator wants to add `malware playbook` to the **Plan** drop-down menu. This addition would immediately alert other CSIRT personnel that the playbook should be executed:

1. Click on **Labels** and the following window will appear:



2. Click on **Add Label**. In the text field, enter `Malware Playbook`. For the drop-down menu, select **Plan**. Finally, click on **Save**:



3. Navigate back to the home screen and click on **New Event**. Click on the **Is an incident** checkbox. Under **Plan**, there should be a selection titled `Malware Playbook`:

This is an example of the many options that are available for modification so that the incident response analysts and team members can tailor the FIR to their own operational needs. The FIR application and other applications like it allow incident response teams to track incidents from detection to conclusion while also providing a central storage place for case information. This becomes crucial when it is time to wrap up the investigation and prepare the documentation necessary for stakeholders and key decision makers.

Written reports

How the written report is structured is often dictated by several factors. There are many aspects of an incident response, such as the personnel involved, the type and depth of the investigation conducted, the number of resources involved, and how many individuals from the organization were involved not only in investigating the incident, but also those who have a stake in the outcome. As a result, some organizations may combine the core elements of the incident report, executive summary, and the forensic report into a single document. Others may find that separating out the documentation may better serve those with differing involvement and stakes in the incident investigation. The following are some of the key pieces of information that should be captured and reported during an investigation.

Executive summary

As was previously discussed, the executive summary captures the macro-level view of the incident. This includes a summary of the events, a description of the root cause, and what recommendations are being made to remediate and prevent such an occurrence from happening again. In regulated industries, such as financial institutions or hospitals that have mandatory reporting requirements, it is good practice to state whether the notification was necessary, and, if it was necessary, how many confidential records were exposed. This allows senior management to understand the depth of the incident and ensure that the appropriate legal and customer communication steps are addressed.

Incident report

The incident report has perhaps the widest audience within, and external to, the organization. Even though there are individuals with limited technical skills who will be reviewing this report, it is important to have the proper terminology and associated data. There will always be time to explain technical details to those that may be confused.

The following are some of the key pieces of data that should be captured and incorporated into the report:

- **Background**: The background is the overview of the incident from detection to final disposition. A background of the incident should include how the CSIRT first became aware of the incident and what initial information was made available. Next, it should draw conclusions about the type and extent of the incident. The report should also include the impact on systems and what confidential information may have been compromised. Finally, it should include an overview of what containment strategy was utilized and how the systems were brought back to normal operation.
- **Events timeline**: As the report moves from the background section to the events timeline, there is an increased focus on detail. The events timeline is best configured in a table format. For each action performed, an entry should be made in the timeline. The following table shows the level of detail that should be included:

Date	Time	Description	Performed by
6/17/19	19:08	SOC alerted CSIRT on-call about attempted C2 traffic from an internal host.	John Q. Examiner
6/17/19	19:10	Examined firewall log and determined that host 10.25.4.5 had connected to a known malware C2 server.	John Q Examiner

6/17/19	19:14	Contacted the network security CSIRT member to administratively down the port connecting host 10.25.4.5 on switch 009.	John Q. Examiner
6/17/19	19:25	Removed connectivity to the internal network from host 10.25.4.5 from the network switch 009.	Dale Mitchell

This log may include several pages of entries, but it is critical to understand the sequence of events and how long it took to perform certain actions. This information can be utilized to recreate the sequence of events, but it can also be utilized to improve the incident response process by examining response and process times.

- **Network infrastructure overview**: In the event that an incident has occurred that involves multiple systems across a network, it is good practice to include both a network diagram of the impacted systems and an overview of how systems are connected and how they communicate with each other. Other information, such as firewall rules that have a direct bearing on the incident, should also be included.
- **Forensic analysis overview**: Incidents that include the forensic analysis of logs, memory, or disk drives, an overview of the process, and the results should be included in the incident report. This allows stakeholders to understand what types of analyses were performed, as well as the results of that analysis, without having to navigate the very technical aspects of digital forensics. Analysts should ensure that conclusions reached via forensic analysis are included within this section. If the incident response team made extensive use of forensic techniques, these can be recorded in a separate report covered later in this chapter.
- **Containment actions**: One of the key tasks of an incident response team is to limit the amount of damage to other systems when an incident has been detected. This portion of the report will state what types of containment actions were undertaken, such as powering off a system, removing its connectivity to the network, or limiting its access to the internet. Analysts should also ensure that the effectiveness of these measures is incorporated into the report. If, for example, it was difficult to administratively remove network access via accessing the switch, and a manual process had to be undertaken, knowledge of this fact will help the CSIRT create new processes that streamline this action and limit the ability of a compromised host accessing the other portions of the network.

- **Findings/root cause analysis**: The meat of the report that is of most use to senior leadership and information technology personnel is the findings and, if it has been discovered, the root cause. This portion of the report should be comprehensive and incorporate elements of the timeline of events. Specific factors within hosts, software, hardware, and users that contributed to either a negative or positive outcome within the incident should be called out. If the specific exploit used by the attacker, or a vulnerability that was exploited, has been determined, then this should also be included. The overall goal with this portion of the report is to describe how the threat was able to compromise the infrastructure, and lend credence to the remediation and recommendations that follow.
- **Remediation**: If steps were taken during the incident to remediate vulnerabilities or other deficiencies, they should be included. This allows the CSIRT to fully brief other IT personnel to the changes that were made to limit damage to the rest of the network so that they can then be placed into the normal change control procedures and vetted. This ensures that these changes do not have an adverse impact on other systems in the future.
- **Final recommendations**: Any recommendations for improvements to the infrastructure, patching of vulnerabilities, or additional controls should be included in this section of the report. However, any recommendations should be based upon observations and a thorough analysis of the root cause.
- **Definitions**: Any specific definitions that would aid technical personnel in understanding the incident should be included within the report. Technical terms, such as **Server Message Block** (**SMB**), should be included if, in particular, an exploit was made against vulnerabilities within the SMB protocol on a specific system.

It is critical to understand that this report is the most likely to make its way to various entities within, and external to, the organization. The report should also make its way through at least one quality control review to make sure that it is free of errors and omissions and can be read by the target audience.

Forensic report

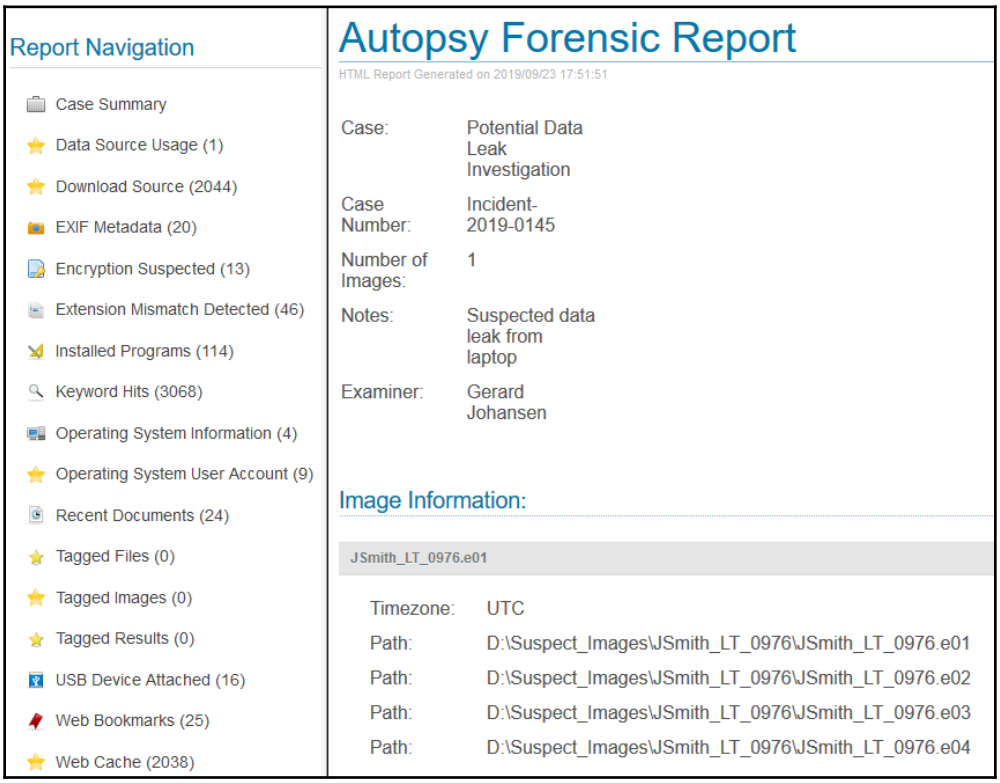
Forensic reports are the most technically complex of the three main report types. Analysts should be free to be as technically accurate as possible and to not dumb down the reporting for those that may be nontechnical. Analysts should also be aware that the forensic report will be critical to the overall incident reporting if it was able to determine a specific individual, such as a malicious insider.

In cases where a perpetrator has been identified, or where the incident may incur legal ramifications, the forensic report will undergo a great deal of scrutiny. It, therefore, behooves the analyst to take great pains to complete it accurately and thoroughly:

- **Examiner bio/background**: For audience members such as legal or external auditors, it is critical to have an idea of the background and qualifications of the forensic analysts. This background should include formal education, training, experience, and an overview of an analyst's courtroom experience, which should include whether they were identified as an expert witness. A complete CV can be attached to the forensic report, especially if it is anticipated that the report will be used as part of a court case.
- **Tools utilized**: The report should include a complete list of hardware and software tools that were utilized in the analysis of evidence. This information should include the make, model, and serial number of hardware, such as a physical write blocker, or the software name and version utilized for any software used. A further detail that can be included in the report is that all tools were up to date prior to use.
- **Evidence items**: A comprehensive list of the evidence items should include any disk images, memory captures, or log files that were acquired by the analysts during the incident. The date, time, location, and analyst who acquired the evidence should also be included. It may be necessary to include as an attachment the chain of custody forms for physical pieces of evidence. If there are a number of evidence items, this portion of the report can be included as an addendum to allow for a better flow of reading for the reader.
- **Forensic analysis**: This is where analysts will be very specific with the actions that were taken during the investigation. Details such as dates and times are critical, as well as detailed descriptions of the types of actions that were taken.
- **Tool output**: During the previous chapters, there have been a great many tools that have been leveraged for investigating an incident. Some of these tools, such as Volatility or Rekall, do not have the ability to generate reports. It is, therefore, incumbent upon the analyst to capture the output of these tools. Analysts can include screen captures or text output from these command-line tools and should incorporate them within the report. This is critical if these tools produce an output that is pertinent to the incident.

Other tools, such as Autopsy, have the ability to output reports for inclusion in the forensic analysis report. For example, to run the report from the analysis conducted in the previous chapter, perform the following steps:

1. Open the case in **Autopsy**.
2. Navigate to **Tools** and then to **Generate Report**.
3. Select **Results - HTML**. Click on **Next** and then **All Results**.
4. This produces an HTML report that will open in the default browser:



From here, the analyst can review the information. Other techniques, such as printing to a PDF file, allow analysts to attach the output directly to the report. Analysts should become familiar with their toolset, as having the ability to export a report directly from the tool will reduce errors and can stand up better under scrutiny.

- **Conclusions**: Conclusions that are derived from the evidence can be included in the report. For example, if an analyst determines that a specific executable is malware by matching the hash with a known strain, and that this malware steals credentials, they are well within their bounds to make that conclusion. However, analysts should be cautious about supposition and making conclusions without the proper evidence to support it. Responders should be careful to never make assumptions or include opinions in the report.
- **Definitions**: As the forensic report is very technical, it is important to include the necessary definitions. Internal stakeholders, such as legal representatives, will often review the report in the event that legal action is anticipated. They may need further clarification on some of the technical details.
- **Exhibits**: Output from tools that are too long to include within the body of the report can be included as an addendum. For example, if the output of a Volatility command is several pages long, but the most pertinent data is a single line, the analyst can pull that single line out and include it in the forensic analysis portion while making it clear that the entire output is located as an addendum. It is important to include the entire output of a tool as part of this report to ensure that it will stand up to scrutiny.

One of the key factors of the forensic report is to have a peer-review process before it is issued as part of the incident documentation. This is to ensure that the actions that have been performed, the analysis, and the conclusions match the evidence. This is one of the reasons that analysts should include as much data as possible from the output of tools or through the review. In the event that a forensic report does go to court, understand that an equally or even more qualified forensic analyst may be reviewing the report and critiquing the work. Another responder or analyst should be able to review the report, review the descriptions of the responder's work, and come to the same conclusion. Knowing this may make analysts more focused on preparing their reports.

Whether or not an organization chooses to separate the documentation or prepare a master report, there is certain data that should be captured within the report. Having an idea of what this data is comprised of allows incident response personnel to ensure that they take the proper notes and record their observations while the incident investigation is in progress. Failure to do so may mean that any actions taken, or observations made, are not captured in the report. Furthermore, if the case is going to see the inside of a courtroom, evidence may be excluded. It is better to overdocument than under document.

Summary

Incident response teams put a great deal of effort into preparing for and executing the tasks necessary to properly handle an incident. Of equal importance is properly documenting the incident so that decision makers and the incident response team itself have a clear understanding of the actions taken and how the incident occurred. It is through the use of this documentation and analyzing a root cause that organizations can improve their security and reduce the risk of similar events taking place in the future. One area of major concern to incident responders and forensic analysts is the role that malware plays in incidents.

The next chapter will discuss some of the techniques available to analysts in addressing these types of incidents.

Questions

1. When preparing an incident report, it is necessary to take into account the audience that will read it.

   A) True
   B) False

2. Which of these is a data source that can be leveraged in preparing an incident report?

   A) Applications
   B) Network/host devices
   C) Forensic tools
   D) All of the above

3. Incident responders should never include a root cause analysis as part of the incident report.

   A) True
   B) False

4. What is not part of a forensic report?

   A) Tools utilized
   B) Examiner biography / CV
   C) Opinion
   D) Exhibit list

Further reading

- Intro to Report Writing for Digital Forensics: `https://digital-forensics.sans.org/blog/2010/08/25/intro-report-writing-digital-forensics/`
- Understanding a Digital Forensics Report: `http://www.legalexecutiveinstitute.com/understanding-digital-forensics-report/`
- Digital forensics report, Ryan Nye: `http://rnyte-cyber.com/uploads/9/8/5/9/98595764/exampledigiforensicsrprt_by_ryan_nye.pdf`

4
Section 4: Specialist Topics

To supplement the first three sections of this book, this section delves into several of the specialized aspects of incident response and digital forensics that have a direct impact on the successful investigation of incidents. These topics include the analysis of malicious code, the integration of threat intelligence, and how to integrate various digital forensic techniques into the practice of threat hunting

This section comprises the following chapters:

- `Chapter 12`, *Malware Analysis for Incident Response*
- `Chapter 13`, *Leveraging Threat Intelligence*
- `Chapter 14`, *Hunting for Threats*

12
Malware Analysis for Incident Response

Malicious software continues to be an ever-evolving scourge on enterprise and consumer systems. As soon as defenses are created, malware coders create a new strain that has the power to corrupt or destroy a system. Malware is even being utilized as a weapon against nation states and global organizations. A great many of the data breaches that have made the news have some component, either in whole or in part, that involves the use of malware to achieve some goal. Organizations in every sector of the economy have faced the threat of malware. With the addition of ransomware attacks such as WannaCry and Petya, organizations have had to spring into action to address these attacks.

With malware an ever-present risk, it is critical that incident response analysts have some knowledge of the methods and tools utilized in the analysis of malicious code. It would be impossible to address the complexities of malware analysis in a single chapter. Therefore, this chapter will focus on the foundational elements of malware analysis, while examining some of the tools that are utilized. This will give an analyst a solid understanding of these methods, and they will then be better able to see the results of such an analysis in the context of an incident.

In this discussion of malware analysis, the following topics will be addressed:

- Malware classifications
- Malware analysis overview
- Analyzing malware
- Tools for analysis
- Sandbox tools and techniques

Malware classifications

Malicious software, or malware, is an all-encompassing term for any software that has been created to damage, disable, or produce an unwanted condition within a computer system. This definition, while functional, is also very broad in its categorization of malware. There is malware that is coded specifically to steal credit card numbers from payment systems, while other malware is utilized to take control of a system, allowing an attacker to remotely control that system. Analysts who observe these specific behaviors—such as how a compromised system sends communications out to the internet after infection, or what actions are taken on an infected system—may be able to determine the type of the malware, and what the end goal of the attacker may be.

In general, when discussing malware, the following are some of the more specific categories:

- **Virus**: For a time, the term *virus* was used as the term for any malicious code that had a detrimental impact on a computer system. As the types of malware increased, the term *virus* was relegated to mean any code that has an intentionally malicious impact on a system.
- **Worm**: Often part of a virus, a worm can not only have an impact on a system but is also able to self-replicate and impact other systems connected to it. One of the most famous worms was the Morris worm that spread worldwide, causing **denial-of-service** (**DoS**) attacks across the internet in 1988.
- **Trojan**: The Trojan horse of mythology is the inspiration for this class of malware. Trojan malware is often hidden within a legitimate application or file. When an unsuspecting user opens the file, the malware infects the system. This type of malware often leverages a social engineering attack to infect a system.
- **Keylogger**: This specific malware hides in the background of a running system and captures the keystrokes of the user. It then takes this information and sends it to a controller for review. Coders who write keyloggers are often interested in obtaining credentials.
- **Rootkit**: Rootkits are utilized to conceal other malicious code such as a **Remote Access Trojan** (**RAT**), which allows an attacker to take remote command of an infected system.
- **Information-stealing malware**: Often coded for a single purpose, this type of malware is used to capture information such as credit card numbers or banking credentials, such as the Shylock malware that was created specifically to capture banking logins.

- **Backdoor**: Another variation of remote access, this type of malware infects a system, and then allows the attacker to take control of the infected system.
- **Downloader**: As defenses have become more sophisticated, so have the malware writers. A downloader is part of a *multi-stage* malware program. The downloader often infects a system, and then reaches out to a remote server for the rest of the code. This method is often utilized to bypass security controls and is useful for enabling malware coders to utilize larger and more sophisticated malware.
- **Botnet**: A botnet is a series of computers, all controlled through a central system on the internet called a **botnet controller**. First, the botnet malware infects a system. As the number of infected systems grows, the malware writers can then utilize this botnet to conduct **distributed denial-of-service** (**DDoS**) attacks against a single target.
- **Ransomware**: A relatively new type of malware, ransomware encrypts a victim's files. The malware then solicits a payment, often in the form of a cryptocurrency such as Bitcoin, from the victim for the decryption key.
- **File wipers**: A file wiper either destroys the files or is able to infect the **Master Boot Record** (**MBR**) and modify records so that files are no longer accessible to the system.

Many of the variants are used together in a chain. For example, a malware coder may conduct an initial infection of a system, with a RAT disguised as a legitimate application. When an unsuspecting user opens the application, the code executes itself. It then downloads a second payload and further infects the system, allowing the coder remote access. Finally, with remote access, the attack continues, with the attacker identifying a payment system. From there, they load a second piece of malware onto the payment system and capture cleartext credit card numbers.

Another key aspect of malware is how it has evolved over time. There has been an explosion in how many variants of malware there are and the sheer amount of malicious code there is currently in the wild. Malware is evolving every day, with new techniques of encoding and delivery—as well as execution—changing rapidly. Analysts would be well advised to make a point of keeping abreast of these changes as they are happening so that they are prepared for the latest, and more damaging, code.

Malware analysis overview

Malware analysis, or malware reverse engineering, is a highly technical and specialized field in forensics. Anti-virus and threat intelligence utilizes a highly trained cadre of programmers and forensic personnel who acquire malware from the wild, and then rip it open to determine what it does, how it does it, and who may be responsible for it. This is done utilizing two types of analysis: static and dynamic. Like much of digital forensics, each type of analysis affords some advantages, and incident response analysts should be familiar with both.

An excellent treatment of malware analysis conducted against actual malware found in the wild can be found in Kim Zetter's book *Countdown to Zero Day*. Comprehensively researched, this book delves deep into the Stuxnet virus, as various research teams attempt to understand what the malware is doing.

An excellent malware analysis methodology was created by Lenny Zeltser, a malware analysis professional who has an excellent array of resources on his website at `https://Zeltser.com`. This methodology comprises the following seven steps that aid analysts in their process:

1. Create a controlled laboratory environment where examinations can be conducted.
2. Examine the behavior of the suspected malware as it interacts with the **Operating System** (**OS**) environment.
3. Examine the suspicious application's code, to gain a sense of the inner workings.
4. Perform a dynamic analysis, to determine what actions to take that could not be identified in the static analysis.
5. Determine if the malware is *packed*, and unpack as necessary.
6. Continue the process, until the analysis objectives have been completed.
7. Prepare a supplement to the forensics reporting and return the laboratory to the state prior to the analysis.

Let's look at static analysis.

Static analysis

Static analysis is an examination of the actual malware code without executing it on a system. For malware researches, the code may be obtained from systems that are left out to be deliberately infected, or from production systems that have been impacted by the malware.

In this case, incident response analysts can obtain the actual source code or executable through a combination of memory analysis and acquiring the actual executable during an analysis of the hard drive. Static analysis often comprises several different techniques, as follows:

- **Fingerprinting**: One of the most basic techniques is obtaining a cryptographical hash of the code. These hashes can then be compared to other known hashes, to determine if the code has been seen before.
- **Anti-virus scanning**: Anti-virus vendors often do not catch every virus. For example, some vendors may have done an analysis of the code and deployed a signature for their own product. Other vendors may not have had access to the code or deployed their own signature. A good step is to use multiple different anti-virus vendors to scan a file.
- **String extraction**: Malware coders will often include IP addresses, error messages, or other data encoded within the malware in cleartext. Finding these strings may allow the analysts to identify a **Command and Control** (**C2**) server or other data that may indicate the purpose of the malware.
- **File format**: With any executable, legitimate or not, there is metadata associated with it. Malware analysts can view the compilation time, functions, strings, menus, and icons of portable executable-format applications.
- **Packer analysis**: To bypass anti-virus programs, malware coders make use of packers. These packers use compression or encryption so that they do not leave a tell-tale file hash. There are some tools available but, often, conducting a static analysis against packed malware is difficult.
- **Disassembly**: Reversing the code through the use of specialized software allows malware analysts to view the assembly code. From here, the analyst may be able to determine which actions the malware is attempting to perform.

When compared to dynamic analysis, static analysis may seem a bit more laborious. While there is a lot of searching and analysis done by hand, there are some advantages. First, it is safer to examine the code without having to execute it. This is especially true in organizations where a comprehensive sandbox solution is not in place. Also, it provides a more comprehensive analysis and a better understanding of what the malware coder's intentions might be.

There are several disadvantages to static analysis as well. This technique requires the malware code in its entirety, for best results. Another key disadvantage is the time necessary to conduct the analysis. With malware becoming increasingly more complex, the time required for a static analysis may be longer than an organization can afford.

This is even more of an issue during an incident where the incident response team may be better off with an analysis that covers most of their issues now, rather than having to wait for the most comprehensive analysis.

Dynamic analysis

In static analysis, the focus is on examining the potential malware in a controlled environment. The focus is on examining the actual code, or to look for specific file attributes that could be compared to other sources. In dynamic analysis, the focus is on allowing the potential malware to execute within a controlled environment, and to observe the behaviors that the program exhibits.

There are several advantages that dynamic analysis affords malware researchers and incident responders. First, allowing the code to execute fully will remove barriers such as encryption, or other obfuscation techniques that are utilized by malware coders. Second, there are several automated tools that can be leveraged for dynamic analysis. This removes the manual process, which can be very labor-intensive as malware continues to increase in complexity. Finally, dynamic analysis is often much faster, as a researcher can monitor in real time how a piece of potential malware works on a system.

There are two broad categories of dynamic malware analysis that can be utilized, as follows:

- **Defined point analysis**: In this method, a test OS such as Windows 7 is configured in a live production state. Analysts make a recording of various registry key settings, processes, and network connections. Once these are recorded, the suspected malware is executed on the system. Once the analysts are confident that the malware is executed completely, they will then compare the two points of the system, such as comparing the running processes or identifying changes. This type of analysis can make use of some of the forensic techniques addressed in previous chapters. For example, analysts can take a freshly installed OS and perform a memory capture. This memory capture, and a subsequent one that is taken from the infected machine, gives the analysts a point of comparison, to identify specific behaviors of the malware.

- **Runtime behavior analysis**: In this method, analysts utilize tools such as Process Explorer and other utilities to observe the behavior of the suspected malware while it is executing. There are also tools that automate a good deal of this process, to give analysts a good understanding of how the malware is executing.

While there are distinct advantages to dynamic analysis, incident responders should understand some of the concerns that need to be addressed prior to detonating suspected malware on a system. First, a controlled environment must be configured.

Suspected malware should never be executed in a production environment. Researchers and incident responders should ensure that any test or analysis environment is completely separated from the production environment.

Another concern is the number of resources that are required to create a proper environment for dynamic analysis. Malware researches and incident responders make use of a sandbox environment for the analysis of malware. A sandbox is simply a controlled environment where suspect malware is executed, and the associated analysis can take place. For organizations that research malware, this sandbox can become quite large, as copies of the various OSes and their patch levels should be maintained. For example, for an organization to test a malware sample that impacts the Windows OS, they will often have to have instances of Windows XP, Windows 7, Windows 8, and—finally—Windows 10, with the various patch levels. This allows them to zero in on the specific OSes that are impacted by the malware. In addition to the OSes, analysts will also need to have images of the memory.

Analyzing malware

The tools for analyzing malware range from simple hex editors and interactive disassemblers to GUI-based tools that integrate online searching and analysis. Each incident will often dictate the specific tools or techniques utilized. A possible infection through a social engineering email that is in the process of infecting network systems may require analysts to work rapidly to identify the malware's behavior and craft a solution to remove it. In other circumstances, a security control may have identified a file that it deems suspicious. With no active incident at hand, the incident response analysts may want to completely rip apart the code, to determine if it had a specific purpose. In either case, tools described in the next section are useful in assisting in the process, but the list is by no means all-inclusive.

There are several sites that provide sample malware for training and research. For example, in this chapter, two such samples were taken from the website `http://malware-traffic-analysis.net/`. These files can be downloaded in ZIP format. As a general rule, any malware sample will be password protected with the word *infected*.
Before taking on the task of analyzing, ensure that the system utilized is properly isolated from the network, and anti-virus is turned off. One technique is to utilize a virtual machine that can be snapshotted before use and returned to that state after analysis.

Static analysis

There are several tools that can be leveraged for static analysis. The combination of automated and manual tools allows an analyst to identify components of the malware that should have additional focus, as well as identifying specific elements within the application that are indicative of malware.

ClamAV

One first step in conducting a static analysis is to determine if the potential malware under analysis has been previously identified. A single sample's hash can be uploaded to sites such as VirusTotal, but if a responder has acquired a number of files through their analysis, they will need to be able to determine if there are any that warrant further examination. One technique is to use a commercial anti-virus scanner to scan the directory. In this case, a free, open source tool that can be leveraged is ClamAV.

ClamAV is a command-line utility that allows responders to scan a directory with a variety of suspicious file formats. From here, suspicious files that are identified can be further analyzed by the responder. To get started with ClamAV, proceed as follows:

1. Navigate to the ClamAV downloads page at `https://www.clamav.net/downloads`.
2. Download the applicable OS version. (For this volume, the Windows executable available at `https://www.clamav.net/downloads/production/ClamAV-0.102.1.exe` will be utilized.)

3. Follow the directions for configuration and updating the signature file at `https://www.clamav.net/documents/installing-clamav-on-windows`.
4. Open a Command Prompt or Windows PowerShell terminal.
5. For example, several files from the site Malware Traffic Analysis will be reviewed. The files are available at `https://www.malware-traffic-analysis.net/2019/09/04/index.html`. In the terminal, type the following code:

```
PS C:\Program Files\ClamAV>.\clamav.exe -m D:\Malware
Samples\2019-09-04-malware-from-Ursnif-and-Trickbot-infection
```

6. This command executes the ClamAV scanner against all the files contained in the folder `2019-09-04-malware-from-Ursnif-and-Trickbot-Infection`. Hit *Enter*, which produces the following results:

```
2019-09-04-Windows-registry-updates-caused-by-Ursnif.txt: OK
2019-09-04-Word-doc-from-password-protected-zip-archive.doc: Doc.Malware.Sagent-7159046-0 FOUND
```

7. ClamAV has indicated that the `.doc` file is associated with the `Doc.Malware.Sagent-7159046` malicious file signature.

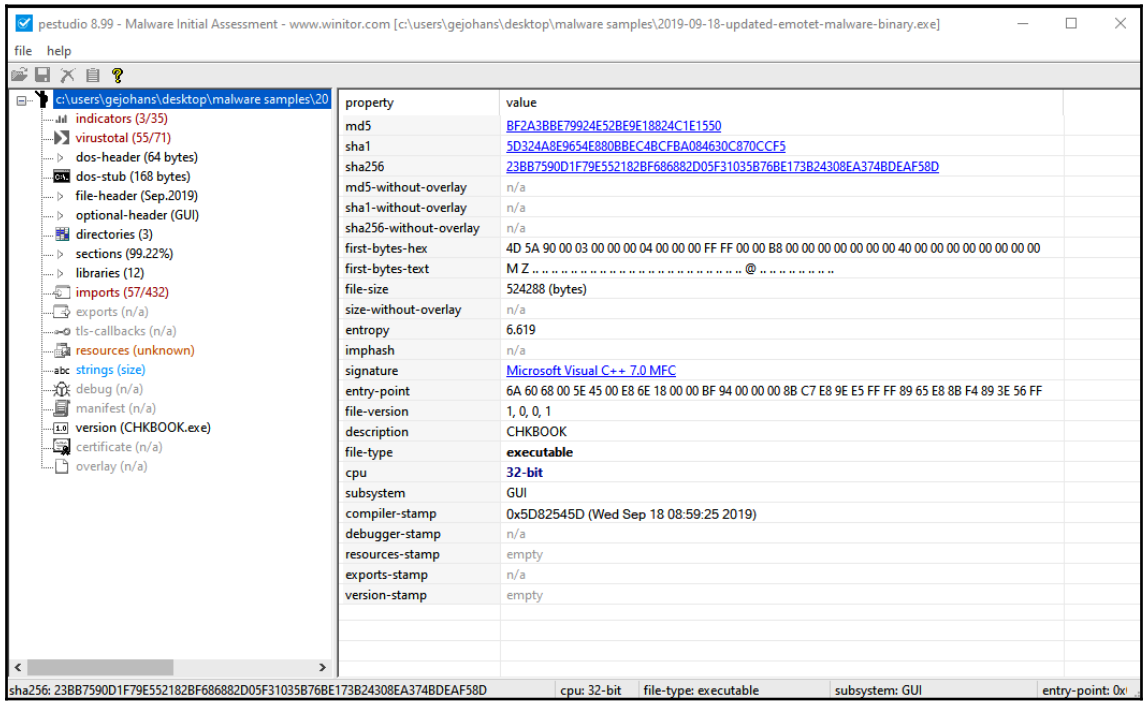
The efficacy of ClamAV is largely dependent on the signatures that are included as part of the scanning package. There are some malware variants that may not have a corresponding signature available and, as a result, will go undetected. Understanding that, ClamAV is a useful way to examine a large number of potential malware files, and to identify those that are already known.

PeStudio

A good place to begin a static analysis of a single file is with PeStudio. `Chapter 8`, *Analyzing System Memory*, introduced this application when examining suspect malicious software obtained through the analysis of a memory image. In this case, an actual piece of malware will be analyzed, using PeStudio. This tool allows analysts to focus on specific attributes of the malware, for further analysis.

In this scenario, a live piece of malware will be examined. The malware sample is an **Emotet infection with TrickBot**. This sample was taken from `https://www.malware-traffic-analysis.net/2019/09/18/index.html`. Ensure that the proper preconfiguration is completed prior to downloading any malware, as any anti-virus program will quarantine the malware, making any analysis impossible. Once downloaded into a folder, the file is ready for analysis. Proceed as follows:

1. Open PeStudio. Click the folder icon in the upper left-hand corner, and navigate to the malware sample with the filename `2019-09-18-updated-Emotet-malware-binary.exe`.
2. Once loaded, the following window will appear. In the pane to the left are several elements that PeStudio examines against the malware. Several of these elements (in red) indicate that the code is suspected of containing malware, as shown here:



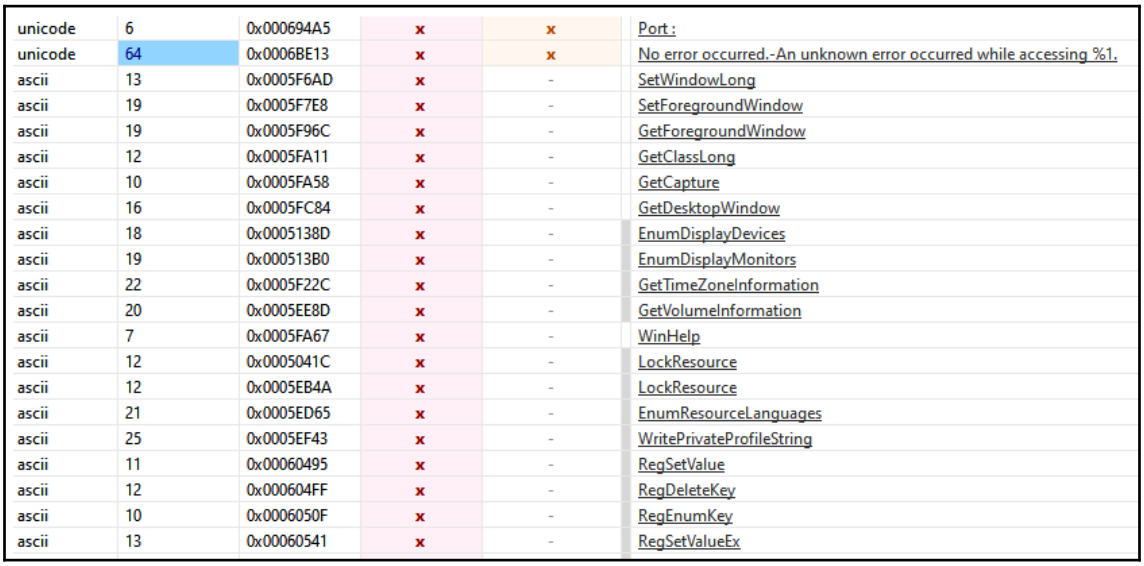
3. Click on **indicators** first, to get an overview of the specific components of the malware that have been identified as malicious. PeStudio has identified three specific indicators—the file has 55/71 hits on VirusTotal, there are several blacklisted strings, and, finally, the file imports symbols that are blacklisted, as can be seen in the following screenshot:

1430	The file references string(s) tagged as blacklist	count: 85	1
1120	The file is scored by virustotal	score: 55/71	1
1266	The file imports symbol(s) tagged as blacklist	count: 57	1

4. Click on **imports**. From here, the analyst can see the imported library files that are blacklisted, as shown in the following screenshot:

name (432)	group (14)	MITRE-Technique (5)	type (1)	anonymous (6)	blacklist (57)
GetCapture	windowing	-	implicit	-	x
GetClassLongA	windowing	-	implicit	-	x
GetForegroundWindow	windowing	-	implicit	-	x
SetForegroundWindow	windowing	-	implicit	-	x
SetWindowLongA	windowing	-	implicit	-	x
GetDesktopWindow	windowing	-	implicit	-	x
GetTimeZoneInformation	system-information	-	implicit	-	x
GetVolumeInformationA	storage	-	implicit	-	x
WinHelpA	shell	-	implicit	-	x
EnumResourceLanguagesA	resource	-	implicit	-	x
LockResource	resource	-	implicit	-	x
WritePrivateProfileStringA	registry	-	implicit	-	x
RegDeleteValueA	registry	T1112	implicit	-	x
RegSetValueA	registry	T1112	implicit	-	x
RegDeleteKeyA	registry	T1112	implicit	-	x
RegEnumKeyA	registry	T1012	implicit	-	x
RegCreateKeyA	registry	-	implicit	-	x
RegSetValueExA	registry	T1112	implicit	-	x
VirtualProtect	memory	-	implicit	-	x
GetKeyState	keyboard-and-mouse	-	implicit	-	x
SetWindowsHookExA	hooking	T1179	implicit	-	x
CallNextHookEx	hooking	T1179	implicit	-	x
UnhookWindowsHookEx	hooking	T1179	implicit	-	x
GetShortPathNameA	file	-	implicit	-	x
FindFirstFileA	file	-	implicit	-	x
FindClose	file	-	implicit	-	x
UnlockFile	file	-	implicit	-	x

5. Click on **strings**. This gives the analysts a clear understanding of the various strings within the malware that are suspect. From here, the analyst can focus on those strings when conducting a deeper analysis, as shown in the following screenshot:



unicode	6	0x000694A5	x	x	Port :
unicode	64	0x0006BE13	x	x	No error occurred.-An unknown error occurred while accessing %1.
ascii	13	0x0005F6AD	x	-	SetWindowLong
ascii	19	0x0005F7E8	x	-	SetForegroundWindow
ascii	19	0x0005F96C	x	-	GetForegroundWindow
ascii	12	0x0005FA11	x	-	GetClassLong
ascii	10	0x0005FA58	x	-	GetCapture
ascii	16	0x0005FC84	x	-	GetDesktopWindow
ascii	18	0x0005138D	x	-	EnumDisplayDevices
ascii	19	0x000513B0	x	-	EnumDisplayMonitors
ascii	22	0x0005F22C	x	-	GetTimeZoneInformation
ascii	20	0x0005EE8D	x	-	GetVolumeInformation
ascii	7	0x0005FA67	x	-	WinHelp
ascii	12	0x0005041C	x	-	LockResource
ascii	12	0x0005EB4A	x	-	LockResource
ascii	21	0x0005ED65	x	-	EnumResourceLanguages
ascii	25	0x0005EF43	x	-	WritePrivateProfileString
ascii	11	0x00060495	x	-	RegSetValue
ascii	12	0x000604FF	x	-	RegDeleteKey
ascii	10	0x0006050F	x	-	RegEnumKey
ascii	13	0x00060541	x	-	RegSetValueEx

PeStudio allows incident responders to get a 10,000-foot overview over suspected malware. Often, this may be enough for incident responders to work from, in terms of identifying other infections. In other incidents, it may be necessary to perform a deeper analysis. This is where other tools come into play.

REMnux

REMnux is a freeware command line-based utility for conducting malware analysis. Developed and maintained by Lenny Zeltser, REMnux has a variety of tools that allow analysts to examine suspicious documents, JavaScript, and other artifacts associated with malware. Further, there are tools such as Wireshark that can be utilized to not only analyze the malware but to identify network connections or traffic.

Information on REMnux is located on the site `https://remnux.org/`and can be downloaded in an OVA file format from `https://docs.google.com/uc?id=0B6fULLT_NpxMampUWlBCQXVJZzAexport=download`.

Once downloaded, the file can be converted by the analyst's selected virtualization software. On the desktop are two links to `.html` files, and a PDF document that contains all of the necessary information for an analyst to conduct an examination. To start an examination, click on the Terminal icon, and an icon window will appear. For most commands, the convention is as follows:

```
remnux@remnux:~$ <Command><Malware File>
```

In the following example, a malware executable will be analyzed. In this case, the file will be a binary associated with the Ursnif doc. The file can be found at `https://www.malware-traffic-analysis.net/2019/09/04/index.html`. Again, ensure that proper precautions are taken when working with malware. One advantage of REMnux as a Linux platform is that there is little risk in downloading a Windows-based malware sample directly onto that system.

Once the file is properly situationed, pescanner can be run against the file. pescanner is a tool that statically examines a PE-formatted file. To run the tool, the following command is used:

```
remnux@remnux:~$ pescanner 2019-09-04-initial-Ursnif-binary.exe
```

The command produces a good deal of information. It first provides the metadata, including a file hash, size of the binary, and the architecture for which the binary has been coded, as shown in the following screenshot:

```
remnux@remnux:~/MalwareSamples$ pescanner 2019-09-04-initial-Ursnif-binary.exe
################################################################################
[0] File: 2019-09-04-initial-Ursnif-binary.exe
################################################################################

Meta-data
================================================================================
Size            : 300032 bytes
Type            : PE32 executable (GUI) Intel 80386, for MS Windows
Architecture    : 32 Bits binary
MD5             : b2490c2f4f8d22ddb34b4cbecd3c69b3
SHA1            : 59154cb6a203e00fbe0431281b2bb33e1b00061a
ssdeep          : 6144:TJ8mth3sLtIAqj3FVzpe5ZFzbLXLe86HGrHnQ2Jx:uWJsIY5ZFzPy86HOHH
imphash         : 0e1c43d49561655b09b5f1bc6792fa38
Date            : 0x4AA0FBD5 [Fri Sep  4 11:36:53 2009 UTC]
Language        : ENGLISH
CRC:    (Claimed) : 0x0, (Actual): 0x54247 [SUSPICIOUS]
Entry Point     : 0x4207ae .text 0/5
```

Another key focus point is that REMnux identified that the **CRC** (short for **Cyclic Redundancy Check**) does not match, indicating suspicious behavior. A CRC check is normally used to detect errors or changes in raw data contained within software code, through the use of a hashing function. In this case, the executable's metadata indicates that the CRC is 0x0, and in actuality is 0x54247. This anomaly is another indicator that may point toward the file being malicious.

Other information includes the instructions that the malware is programmed to perform, shown in the following screenshot:

```
Offset | Instructions
----------------------------------------
0       call 0x423e7c
5       jmp 0x420631
10      push byte 0xc
12      push dword 0x43b3e0
17      call 0x4210f0
22      and dword [ebp-0x1c],0x0
26      mov esi,[ebp+0x8]
29      cmp esi,[0x44e2d0]
35      ja 0x4207f5
37      push byte 0x4
39      call 0x42408e
44      pop ecx
45      and dword [ebp-0x4],0x0
49      push esi
50      call 0x4248a0
55      pop ecx
56      mov [ebp-0x1c],eax
59      mov dword [ebp-0x4],0xfffffffe
66      call 0x4207fe
71      mov eax,[ebp-0x1c]
74      call 0x421135
79      ret
80      push byte 0x4
82      call 0x423fb4
87      pop ecx
88      ret
89      mov edi,edi
91      push ebp
92      mov ebp,esp
94      push esi
95      mov esi,[ebp+0x8]
98      cmp esi,0x0
```

Next, pescanner has identified the sections of memory in use by the piece of malware. There is also a section that details the entropy of specific sections. Higher entropy indicates that a file is compressed or encrypted. In this case, pescanner has indicated that there is no specific memory section deemed suspicious, due to its being outside the norm for entropy, as shown in the following screenshot:

```
Sections
=============================================================================================
Name       VirtAddr     VirtSize     RawSize    MD5                                Entropy
---------------------------------------------------------------------------------------------
.text      0x1000       0x2e9c7      0x2ea00    ea80d5c83da498d1b76c537bdfc80370 6.717635
.rdata     0x30000      0xdc66       0xde00     0f5ff6da63a54786b653ce46fe4c1830 5.805477
.data      0x3e000      0x1041c      0x4c00     a5297d66be916d217b1f5f18812916a5 5.463237
.rsrc      0x4f000      0x530        0x600      15f94e54fda75a4b78243579f4c48870 3.658907
.reloc     0x50000      0x742a       0x7600     8e489f790699d5a7eda31642e180e611 2.852141
```

Moving down the results, pescanner indicates which **dynamic-link library** (**DLL**) files are imported as part of this malware. From here, the analyst can possibly determine more of the malicious file's behavior, by examining these files, a list of which can be seen in the following screenshot:

```
Imports
================================================
[1] KERNEL32.dll
[2] USER32.dll
[3] WINSPOOL.DRV
[4] COMCTL32.dll
[5] ole32.dll
[6] OLEAUT32.dll
[7] SHLWAPI.dll
[8] ADVAPI32.dll
[9] CLUSAPI.dll
[10] OLEACC.dll
[11] GDI32.dll
```

Finally, pescanner will indicate which suspicious **Import Address Table** (**IAT**) entries are being called by the malicious software. This allows analysts to determine what behavior the malicious code is exhibiting and, possibly, which actions it is performing on the infected system by examining these entries, as shown in the following screenshot:

```
Suspicious IAT alerts
============================================================
[1] CreateDirectoryW
[2] CreateFileA
[3] CreateFileW
[4] FindResourceW
[5] GetCommandLineW
[6] GetModuleFileNameA
[7] GetModuleFileNameW
[8] GetModuleHandleA
[9] GetModuleHandleW
[10] GetProcAddress
[11] GetStartupInfoA
[12] GetStartupInfoW
[13] GetTickCount
```

Boasting a wide range of tools, REMnux is an excellent resource to conduct a wide range of tasks associated with file examination. Further, REMnux includes other tools, such as Rekall and Volatility, so that analysts can perform a panoply of tasks from memory image analysis, in conjunction with malware analysis.

YARA

Another tool that responders should be familiar with when it comes to malware analysis is the pattern-matching tool YARA. **YARA** (short for **Yet Another Ridiculous Acronym**) is a schema to identify and classify malware, through the creation of text-based rules. YARA rules are a combination of strings and Boolean expressions that are married with a condition, to determine if a specific file has any of the attributes contained within the YARA rule. For example, the following is a YARA rule created by Florian Roth for the Stuxnet malware and is available at `https://github.com/Yara-Rules/rules/blob/master/malware/APT_Stuxnet.yar`. This rule has been written to examine a suspect file for full-word strings that are associated with the Stuxnet malware, and the code for this can be seen here:

```
rule Stuxnet_Malware_3
{
  meta:
    description = "Stuxnet Sample - file ~WTR4141.tmp"
    author = "Florian Roth"
    reference = "Internal Research"
    date = "2016-07-09"
    hash1 =
"6bcf88251c876ef00b2f32cf97456a3e306c2a263d487b0a50216c6e3cc07c6a"
    hash2 =
"70f8789b03e38d07584f57581363afa848dd5c3a197f2483c6dfa4f3e7f78b9b"
  strings:
    $x1 = "SHELL32.DLL.ASLR." fullword wide
    $s1 = "~WTR4141.tmp" fullword wide
    $s2 = "~WTR4132.tmp" fullword wide
    $s3 = "totalcmd.exe" fullword wide
    $s4 = "wincmd.exe" fullword wide
    $s5 = "http://www.realtek.com0" fullword ascii
    $s6 = "{%08x-%08x-%08x-%08x}" fullword wide
  condition:
    ( uint16(0) == 0x5a4d and filesize < 150KB and ( $x1 or 3 of ($s*) ) )
or ( 5 of them )
}
```

The previous rule has been written to trigger if the file size is less than 150 KB, and if either the full-word string `SHELL32.DLL.ASLR` or three of the `$S` strings are found within the file. YARA rules can be written in a standard text editor as a single rule, or multiple rules within the same YARA file.

YARA rules can be utilized in several different ways. First, they can be written based upon the static analysis of malware or exploits and used to scan memory images through Volatility's `yarascan` plugin. This allows responders to determine if there are traces of a particular malware in a memory image. Second, multiple rules can be run against files and disk images, to detect the presence of malware or exploits. There are also scanners such as Loki, available at `https://github.com/Neo23x0/Loki`, that use multiple YARA rules to scan live disks for **indicators of compromise** (**IoCs**).

Writing YARA rules can sometimes present an issue during an incident. This is due to the fact that other techniques of malware analysis, such as PeStudio or pescanner, are needed to extract strings and other indicators. Another option with regard to YARA rules is the tool yarGen, by Florian Roth. This Python script, available at `https://github.com/Neo23x0/yarGen`, allows responders to create YARA rules for specific files that have been identified during the course of an incident.

For this example, a file associated with the `2019-09-04-malware-from-Ursnif-and-Trickbot-infection` file set previously examined contained a Microsoft Word document. Indicators extracted from that file may be useful in analyzing other systems. yarGen can be utilized to create a rule from that file. Proceed as follows:

1. Download yarGen from GitHub at `https://github.com/Neo23x0/yarGen`.
2. Uncompress the ZIP file.
3. Navigate to the directory containing the `yargen.py` script. It is a good practice to put samples in a single directory, as yarGen works against files contained within a directory.
4. Use the following command syntax to create the YARA rule. In this case, the `-m` directs yarGen to scan the directory `Malware` for the files for which the rule will be created. The `-o` is the output file, generally a text file with the extension `.yar`:

```
remnux@remnux:~/Desktop/yarGen-master$python yarGen.py -m Malware -
o sample.yar
```

The script will run, and then produce a YARA rule such as this:

```
/*
   Yara Rule Set
   Author: yarGen Rule Generator
   Date: 2019-10-12
   Identifier: Malware
   Reference: https://github.com/Neo23x0/yarGen
*/

/* Rule Set ----------------------------------------------------------------- */

rule Malware_sample {
   meta:
      description = "Malware - file sample.doc"
      author = "yarGen Rule Generator"
      reference = "https://github.com/Neo23x0/yarGen"
      date = "2019-10-12"
      hash1 = "4e59d5e27d1109b3e2cb46b5dbe8d5c04aa82978177f4d29e3e986b88ff68894"
   strings:
      $x1 = "*\\G{0D452EE1-E08F-101A-852E-02608C4D0BB4}#2.0#0#C:\\WINDOWS\\system32\\FM20.DLL#Mic
      $s2 = "*\\G{00020430-0000-0000-C000-000000000046}#2.0#0#C:\\Windows\\System32\\stdole2.tlb#
      $s3 = "var rXvzpRtxp = function(){return \"\" + \"R\" + \"u\" + \"\\x6e\";};var pzLDkTdpjP
      $s4 = "*\\G{1ABB45AB-A1DB-4268-86A8-DAA4B48BD521}#2.0#0#C:\\Users\\Willy\\AppData\\Local\\T
      $s5 = "TdpjP(\"Ws\" + \"cript.Shell\")[[\"nbHqBJBJXBKzPPXJjLBtz\", rXvzpRtxp()][1]](\"C:\\\
      $s6 = "*\\G{2DF8D04C-5BFA-101B-BDE5-00AA0044DE52}#2.8#0#C:\\Program Files\\Common Files\\Mi
      $s7 = "22222222222222222222222222222222222222222222222222" ascii /* hex encoded string '""
      $s8 = "\\MSO.DLL" fullword ascii
      $s9 = "*\\G{000204EF-0000-0000-C000-000000000046}#4.2#9#C:\\PROGRA~1\\COMMON~1\\MICROS~1\\V
      $s10 = "OAC4AZABlAGYAbABBAFQAZQBgAFMAVABSAEUAQQBtACgAWwBzAHkAUwB0AGUAbQAuAEkATwAuAE0ARQBNAE
```

yarGen has taken the various strings associated with this particular file and created the rule. yarGen is a good option for responders, in the event that they are dealing with a number of systems or files, and they do not have the necessary time to conduct a full static analysis to extract strings.

This has been a very limited exploration of YARA and its capabilities. YARA rules are a powerful tool that can be leveraged during an incident. As rules can be crafted not only for malware but for any file, responders have the ability to scan other systems for IoCs related to exploits or scripts that an adversary has utilized, as part of their attack, that may have bypassed anti-virus or other detective controls.

Dynamic analysis

Dynamic malware analysis is the detonation of the malware in a controlled environment or malware sandbox. During the execution of the malware, the incident responder is able to see the various processes that are created, the network connections that are established, additional packages that are downloaded, and if the malware performs any actions that allow it to maintain persistence. From this analysis, responders gain a better sense of the IoCs associated with the malware and are better able to identify other systems that have been impacted.

While gaining a sense of the actions malware takes when it executes, dynamic analysis has the advantage of not being as time-intensive as static analysis. Responders often do not need to understand the full depth of complexity of the malware in question, but rather have the ability to identify the IoCs associated with the malware. With these, they can craft additional detective and preventive controls, to contain the incident and lessen the damage.

Malware sandbox

Dynamic analysis is often facilitated using a sandbox. A sandbox is a system that has been created for the controlled execution of malware. One method in crafting a sandbox is to have a virtual appliance, with the proper OS and toolset installed. Once the sandbox is properly configured, a snapshot of the virtual system can be taken, and the malware then executed. From here, the analyst can monitor network traffic, examine processes, and compare them to normal baseline behavior. Deviations from this baseline can be examined in more detail, to determine if they are legitimate or tied to a malicious executable.

Responders can also conduct a range of digital forensic tasks on the sandbox, to extract available IoCs. The preferred method is to gain an understanding of the malware's behavior using real-time monitoring, but in cases where there is additional time, it is also possible for a detailed examination of the memory artifacts, log files, and registry settings that were changed as part of the malware execution. A full understanding of these IoCs can provide responders with a deeper understanding of the malware and its relationship to the incident.

One significant advantage of working with virtual systems is that provided a snapshot has been taken, responders can use the sandbox as much as needed. Once the analysis of a piece of malware is completed, the system can be reverted to the original snapshot, allowing the responder to repeat the process without having to build a new system every time.

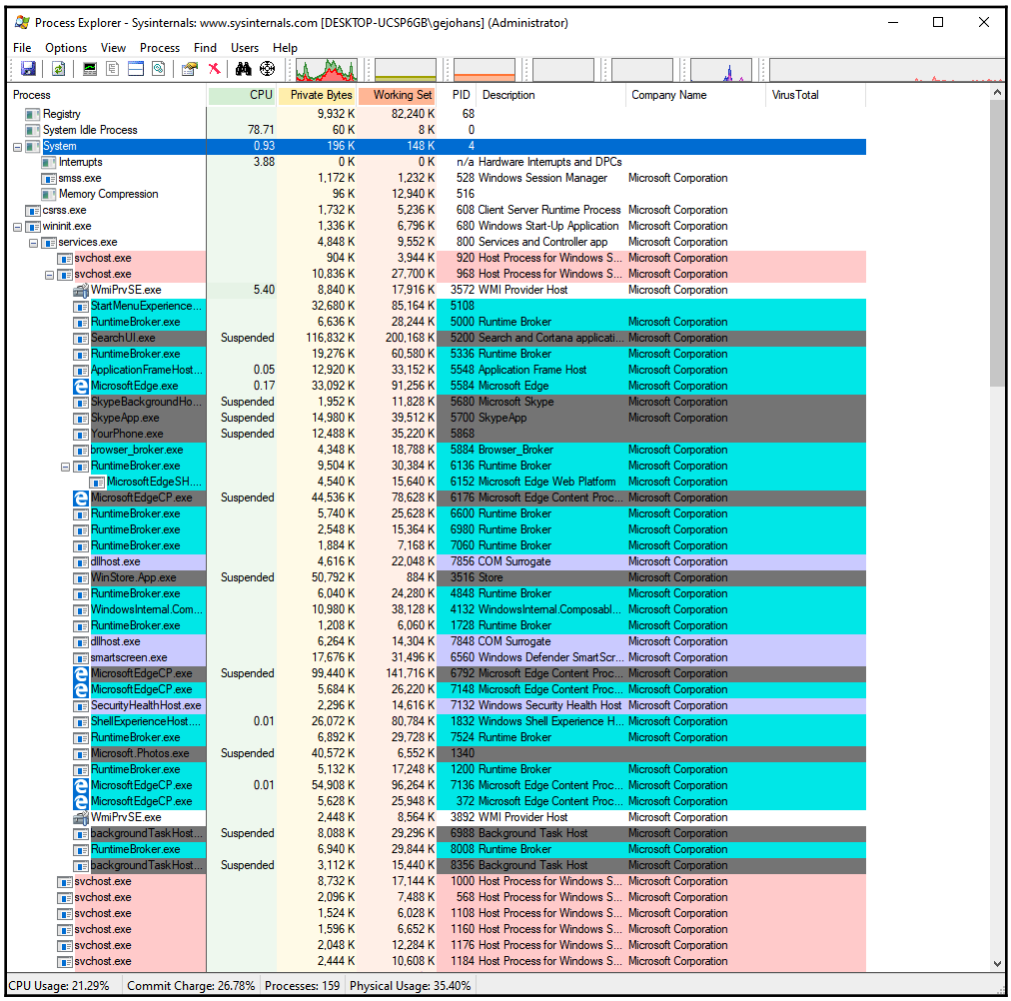
An excellent resource for building a preconfigured sandbox is the **FLARE** (short for **FireEye Labs Advanced Reverse Engineering**) sandbox. This sandbox configuration is a set of tools for dynamic analysis, built on a Windows OS platform. FireEye provides a preconfigured virtual machine, available at `https://github.com/fireeye/flare-vm`.

In addition to utilizing virtual machines, there are several automated tools that recreate the sandbox environment, for analysts to execute live applications that are suspected of containing malicious code. These tools can be configured and deployed within the enterprise environment, or there is the ability to upload the potential malware to a cloud-based service that can perform the functions of a malware sandbox.

Process Explorer

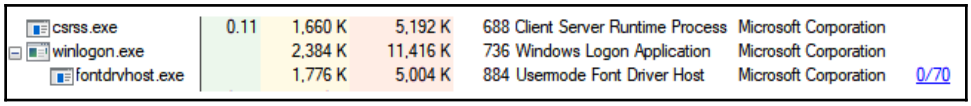
One of the key tools that allow for a detailed examination of malware as it is executing is Process Explorer. This tool is made as part of the Windows Sysinternals suite of tools and provides a no-cost platform for analysts to gain a sense of what each process is running and their parent process, as well as examining CPU usage. Simply download the application from the following site: `https://technet.microsoft.com/en-us/sysinternals/processexplorer.aspx`.

Extract the contents, and then double-click the version of Process Explorer (32-bit or 64-bit version) that is applicable. The following window will appear:



As can be seen, there are several key pieces of information available to the analyst. The major advantage of this tool is the visual representation. As opposed to attempting to utilize either native Windows tools or other memory analysis tools after capture, analysts can quickly see if any processes look suspicious.

Analysts have the ability to send a process and associated data to `https://www.virustotal.com/gui/home/upload`. If a suspicious process is identified, Process Explorer will send the information off to the site, for analysis and comparison. If a process is identified, click on it in the window. Navigate to **Process**, and then **Check VirusTotal**. The results will be indicated by a number out of 70, as can be seen in the following screenshot:



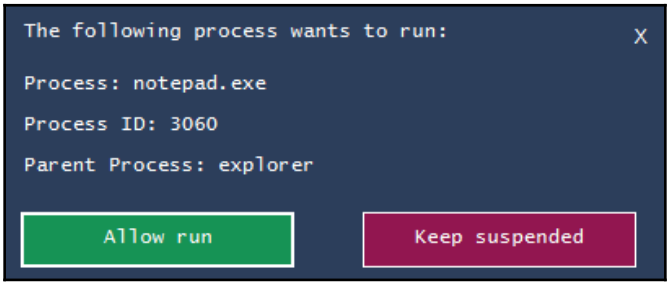
Another key feature that Process Explorer can provide is the ability to dump the process contents in much the same way that Volatility is able to. The major difference is that the analyst is able to conduct the dump without having to acquire a memory image. To dump the memory, click on the process, and navigate to **Process**, and then **Create Dump**. The analyst has the option to choose from a mini-dump or a full dump. As a standard practice, it is advisable to capture a full dump. This dump can then be saved to a directory of choice.

Process Spawn Control

One technique that can be leveraged in examining malware is to create a virtual machine with the appropriate Windows OS. It is best to start with a bare-bones OS, with the Microsoft Office suite installed. Other third-party programs can be installed later if it appears that the malicious code leverages a vulnerability in those applications. A tool that is useful in this type of examination is Process Spawn Control. This PowerShell script, available at `https://github.com/felixweyne/ProcessSpawnControl`, allows responders to control the execution of malware and observe what actions are taken in Process Explorer. To conduct this type of analysis, take the following steps:

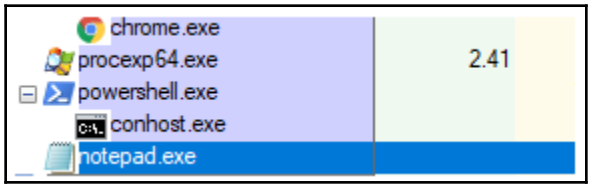
1. Start Process Explorer and let it run for a few seconds.
2. In the PowerShell terminal, execute the `ProcessSpawnControl.ps1` script. Select **Run Once**, if prompted.

3. Process Spawn Control will pause all executables, not just potential malware. Once it is running, open the Windows executable `notepad.exe`. The following window should appear:



4. In the Process Explorer window, the `notepad.exe` process will appear to be suspended, as shown in the following screenshot:

5. Click on **Allow run** in the PowerShell dialog box, and the `notepad.exe` process will then execute, as follows:



Using these tools in combination allows the responder to understand how a potential malware executable functions, and what execution path it may take. This data, combined with other artifacts obtained through memory or log file analysis, can provide additional context to how malware has compromised a system.

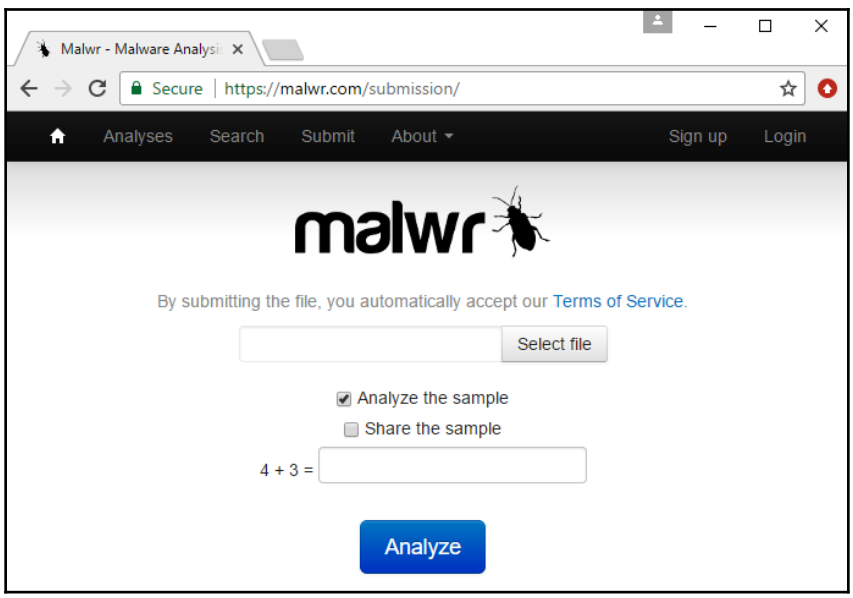
Cuckoo Sandbox

Cuckoo Sandbox is a malware analysis system that automates many of the tasks associated with malware analysis. This open source application has the ability to analyze a variety of suspected malicious files such as Windows executables, documents, and Java applets, all within a virtualized environment. This analysis includes network traffic and memory analysis, utilizing Volatility.

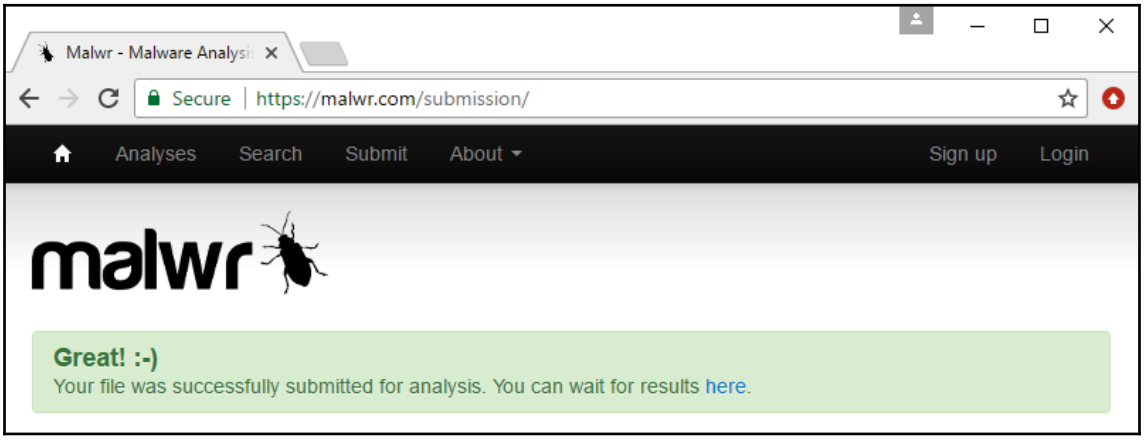
Installing Cuckoo Sandbox can take some time and effort. An excellent resource on installing the local sandbox can be found at `https://bdavis-cybersecurity.blogspot.com/2016/11/cuckoo-sandbox-installation-part-1.html`.

In addition to a local version of Cuckoo Sandbox, analysts can make use of a web-based version. The site `https://malwr.com/` is a free service that allows analysts to upload a copy of the malware and have the site conduct a dynamic analysis. From here, the site will produce a report that can be reviewed. In the following example, `http://malwr.com/` will be utilized to conduct a review of the LokiBot malspam that can be obtained from `http://www.malware-traffic-analysis.net/2017/06/12/index.html`:

1. Navigate to the site `http://malwr.com`, and click **Submit** in the upper left-hand corner. This will open the following window:

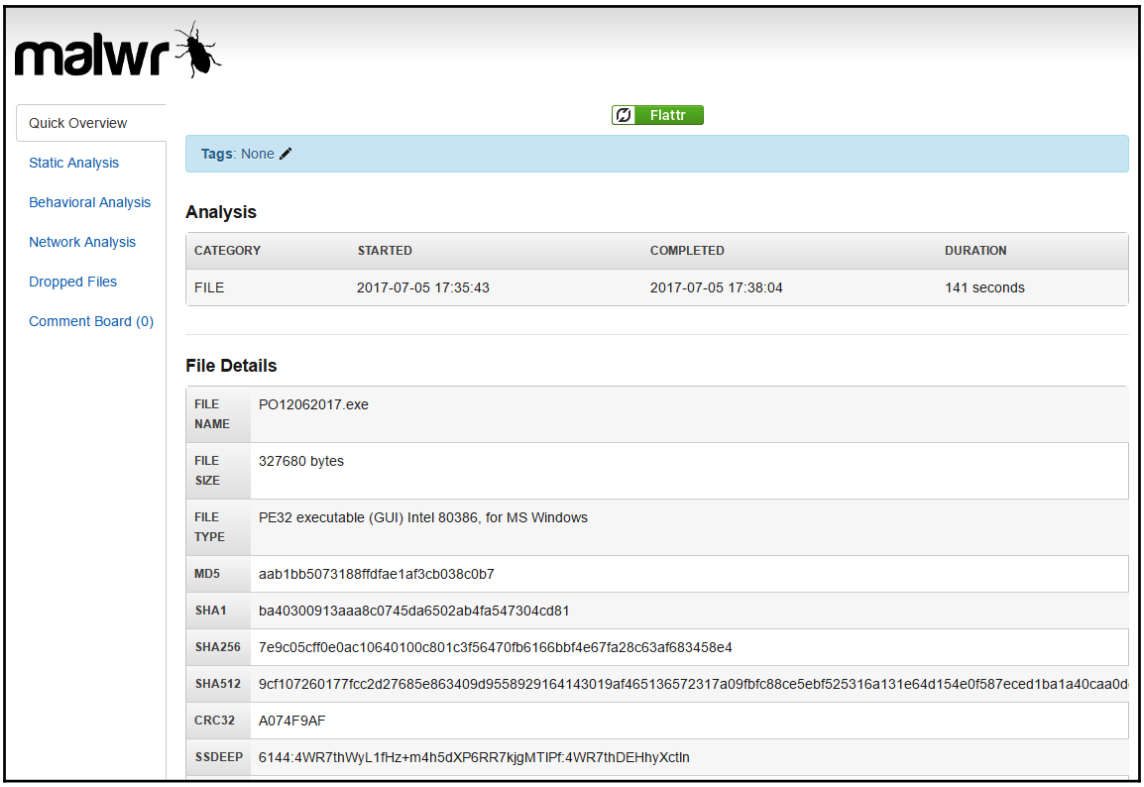


2. Click **Select file**, and then navigate to the malware file to be analyzed. Malwr allows the analyst to share the sample of malware with the community, or not. In this case, as the malware being tested is known, this is not selected. Finally, complete the equation, and click **Analyze**. The following window will appear:



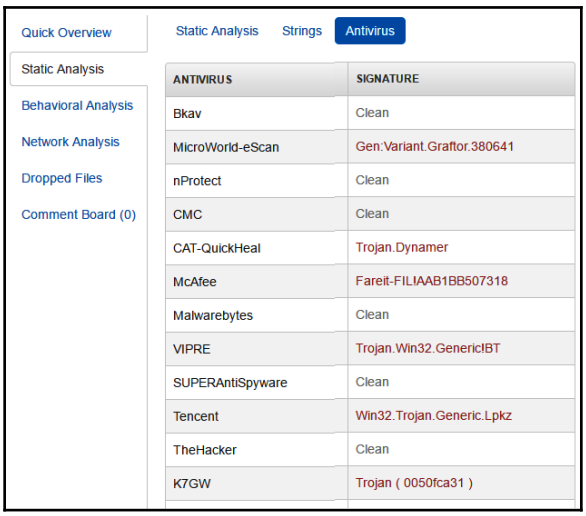
3. Depending on the type of malware and its size, it may take a few minutes for Malwr to analyze. During that time, the following window will appear:

4. Once the analysis is complete, a window will open with the analysis results. The analysis results include static and dynamic analysis elements such as behavioral and network elements for review, as shown in the following screenshot:



5. Click on **Static Analysis**. From here, the analyst can view specific elements, including strings and which elements are imported as part of the DLL file, which, in this case, is `MSVBVM60.dll`, as shown in the following screenshot:

6. While in the **Static Analysis** section, click on **Antivirus**. This provides the analysts with a breakdown of VirusTotal results for the sample uploaded, as shown in the following screenshot:



7. Next, click **Behavioral Analysis**. From here, specific file behaviors are outlined. There are charts that break down the sequence of events that transpired after the malware was executed. This allows analysts to view the specific elements in greater detail, as can be seen in the following screenshot:

8. Often, the malware drops other files as part of the infection. Malwr also allows the analyst to see these files as well. Click on **Dropped Files**. Malwr indicates that there were two files that were dropped via this malware, as can be seen in the following screenshot:

FILE NAME	**filename.vbs**
FILE SIZE	384 bytes
FILE TYPE	ASCII text, with CRLF line terminators
MD5	899dbb13af252b6cd89a6de23048cf8c
SHA1	857745ec21332c7e3a2f6f44af1113ecd9ec3f6a
SHA256	bf01560f94fd75d02a21b8cd133cab3d6e181f7eb4d41b6d415b65fe63665e96
CRC32	DA5FB4F1
SSDEEP	3:j+qAHmFEm86oQ/FERMQsNC2xA+KdlH1MARm5iRMQbm34MkWJJFHrLL:j+q9Nht6G9KdEARm5Mm34M9
YARA	• embedded_win_api - A non-Windows executable contains win32 API functions names

FILE NAME	**filename.exe**
FILE SIZE	327680 bytes
FILE TYPE	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	1f54f93c28df730aa1c13d0090c7eeb5
SHA1	c152e04d928a5397943d9285af8373813affed25
SHA256	75d49db3c12349a5be3da567ed9ca169e05d0ced8e6a15846bf9b884ed954f0c
CRC32	9E4A0D9D
SSDEEP	6144:9WR7thWyL1fHz+m4h5dXP6RR7kjgMTlPv:9WR7thDEHhyXctlX
YARA	None matched

There is a good deal more information that can be obtained via Malwr, including examining network activity and comments provided by the Malwr community. One key consideration does need to be made when examining this platform against a local solution. Malware coders do pay attention to community boards and VirusTotal to see if a hash—or the actual—file has been uploaded. If the malware is specific to a single organization such as a government entity or large retailer, they will know that incident response analysts have discovered their creation. Incident response teams need to balance the speed and ease of this technique with the possibility that their efforts might be discovered.

Summary

This chapter addressed the various elements of malware analysis for the incident responder. First, having an understanding of malware, in general, is necessary, as it is by far the most prevalent threat available to adversaries. Second, the techniques of malware analysis—static and dynamic—provide responders with tools and techniques, to extract key data points. Finally, the use of sandboxing systems allows responders to gain insight into malware behavior and attributes quickly, and in a controlled manner.

In many ways, this chapter has merely scratched the surface in regard to malware analysis. It should become apparent that, even with tools for static and dynamic analysis, incident response analysts still have a great deal of skill-building ahead of them if they want to master this highly specialized subset of digital forensics. Although it may be difficult, it is important to have at least a functional knowledge of this type of analysis as cybercriminals and nation states continue to utilize more sophisticated malware. This chapter delved into malware analysis, by examining the types of malware currently being seen. An overview of the two primary methods of analysis—static and dynamic—gave some context to the tools available. The tools discussed allow an analyst to identify behaviors in malware that can be used to identify them. Finally, actually executing malware can provide further details. The next chapter will tie the use of threat intelligence into malware analysis, to allow analysts an opportunity to tie their observations into what is happening to other organizations.

Questions

1. Which of the following is not a type of malware?

   A) Trojan
   B) Keylogger
   C) Rootkit
   D) Webshell

2. Responders should create a controlled environment in which to conduct malware analysis.

   A) True
   B) False

3. Which of the following is a type of static analysis?

   A) Runtime behavior
   B) String extraction
   C) Memory addressing
   D) Malware coding

4. Which of the following is a type of dynamic analysis?

   A) Disassembly
   B) Defined point
   C) Packer analysis
   D) Artifact extraction

Further reading

- A source for pcap files and malware samples: `https://www.malware-traffic-analysis.net/index.html`
- Malware Unicorn: `https://malwareunicorn.org/#/`
- MalwareJake: `http://malwarejake.blogspot.com/`
- Florian Roth's GitHub account: `https://github.com/Neo23x0/`

13
Leveraging Threat Intelligence

One area of incident response that has had a significant impact on an organization's ability to respond to cyberattacks is the use of cyber threat intelligence or, simply, threat intelligence. The term **cyber threat intelligence** covers a wide range of information, data points, and techniques that allow analysts to identify attack types in their network, adequately respond to them, and prepare for future attacks. To be able to properly leverage this capability, information security analysts should have a solid foundation of the various terminologies, methodologies, and tools that can be utilized in conjunction with threat intelligence. If analysts are able to utilize this data, they will be in a better position to take proactive security measures and, in the event of a security incident, be more efficient in their response.

In this chapter's discussion of cyber threat intelligence, the following key topics will be discussed:

- **Understanding threat intelligence**: Cyber threat intelligence is an amalgamation of existing and emerging disciplines in intelligence. The overview provides a level set of the various topics that are part of cyber threat intelligence.
- **Threat intelligence methodology**: Cyber threat intelligence generation and integration is a process-driven endeavor. This section provides an overview of the cyber threat intelligence methodology.
- **Threat intelligence sources**: There are several sources where responders can gain access to cyber threat intelligence. This portion examines some of the key sources available.
- **Threat intelligence platforms**: Threat intelligence provides an extensive amount of data to responders. An examination of a threat intelligence platform will provide responders with an option to deal with this potential data overload.
- **Using threat intelligence**: Cyber threat intelligence is meant to be used either proactive or reactive. This section provides some key tools and techniques to do just that.

In many ways, this chapter merely scratches the surface of the tools, techniques, and methodologies of cyber threat intelligence. It is hoped that this overview provides a starting point for responders to integrate threat intelligence into their operations.

Understanding threat intelligence

Like some terms in information security and incident response, threat intelligence is a bit nebulous. Various organizations such as the government and academics produce information and data that is often touted as threat intelligence. Various commercial providers also have information available, either through free or paid subscriptions, that is touted as threat intelligence. This often results in difficulty when determining what threat intelligence is and what, simply, data or information is.

A good starting point to determine what comprises threat intelligence is to utilize a definition. Here is the Gartner research company's definition of threat intelligence:

> *"Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications, and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard."*

When examining this definition, there are several key elements that need to be present for data or information to be considered threat intelligence:

- **Evidence-based**: This chapter will examine how evidence obtained through other processes, such as malware analysis, produces threat intelligence. For any intelligence product to be useful, it must first be obtained through proper evidence collection methods. In this way, analysts that rely on it can be sure of its validity.
- **Utility**: For threat intelligence to have a positive impact on a security incident's outcome or an organization's security posture, it has to have some utility. The intelligence must provide clarity, in terms of context and data, about specific behaviors or methods to determine whether an analyst is evaluating an incident against other incidents of a similar nature.
- **Actionable**: The key element that separates data or information from threat intelligence is action. Intelligence should drive action, whether that is a specific sequence of events or a specific focus area of an incident, or whether or not a specific security control is implemented in the face of intelligence about what cyber threats the organization is most likely to face.

To see how this plays together, imagine a scenario where an incident response team at a healthcare institution is attempting to ascertain what types of attacks are most likely to occur against their infrastructure. Vague data about cybercriminals wanting to steal data is not useful. There is no specific context or information in that dataset and the end result is that the organization cannot put that information into action.

On the other hand, say that the incident response team leverages a third-party threat intelligence provider. This third party outlines a specific criminal group by name. The provider also indicates that these groups are currently utilizing PDF files sent via email to hospital employees. The PDF files contain a remote access Trojan that is controlled from C2 servers, which are spread out in Europe. The third party also provides the team with MD5 file hashes of malware, the IP and domain addresses of the C2 servers, and, finally, the filenames most associated with the PDF document.

With this information, the incident response team can align their security controls to prevent PDF attachments from opening in emails. They can also utilize tools to search their infrastructure to determine whether an infection has already occurred. Finally, they may be able to configure their event management solution in order to alert the team if any host within the network attempts to communicate with the C2 server.

The major difference between these two scenarios is that the latter scenario drives actions within the organization. In the first scenario, the information was so vague and useless that the organization was left no better off. In the second scenario, the team could execute specific actions to either prevent an adverse condition or be better prepared to respond to one.

Threat intelligence is a response to the increased complexity and technical skill of cyber threat actors. The focus of threat intelligence is on the following threat actor groups:

- **Cybercriminals**: Organized and technically skilled, cybercriminals have been responsible for a variety of financial crimes against banking, retail, and other organizations. The motive for these groups is purely mercenary and their ultimate goal is to acquire data that can be monetized. For example, attacks against retailers such as Home Depot and Target involved the theft of credit card data with the intent of selling numbers on the dark web or other black markets.
- **Hacktivism**: Groups such as **Anonymous** and the **Idlib Martyrs' Brigade** are hacker groups that take on large businesses, governments, and even religious institutions to further a political cause. Penetrating networks to obtain confidential data for disclosure or conducting denial-of-service attacks are done as part of an overall political versus monetary objective.

- **Cyber espionage**: Nation-states such as the United States, Russia, China, Iran, and North Korea continually engage in espionage activities involving penetrating networks and obtaining intelligence. One of the most well-known cyberattacks, the Stuxnet virus, was reportedly perpetrated by the United States and Israel.

Another key element to understanding threat intelligence is the concept of **Advanced Persistent Threat** (**APT**). The term APT has been around for approximately a decade, and it is used to describe a cyber threat actor whose capability and motivation go far beyond that of a cybercriminal or cyber vandal. APT groups often target organizations for an intended purpose with a clear objective in mind and over a long period of time. As the term APT describes, these groups have the following characteristics:

- **Advanced**: APT threat actors have advanced skills. These skills often involve intelligence gathering skills that exceed what can be obtained through open source methods. This includes such sources as **Imagery Intelligence** (**IMINT**), which includes pictures available through sites such as Google Earth. **Signals Intelligence** (**SIGINT**) is intelligence gathered through the compromise of voice and data communications that use telephone infrastructure, cellular data, or radio signals. Finally, APT groups have the ability to leverage **Human Intelligence** (**HUMINT**) or gather intelligence from interacting with human sources. Further, these groups can not only utilize advanced network penetration tools, but they are also adept at finding zero-day vulnerabilities and crafting custom malware and exploits that specifically target these vulnerabilities.
- **Persistent**: APT threat actors are focused on a clearly defined objective and will often forgo other opportunities to get closer to achieving their objective. APT threat actors will often go months or even years to achieve an objective through the intelligent leveraging of vulnerabilities and continuing a pace that allows them to bypass detection mechanisms. One of the key differentiators between APT threat actors and others is the intention to stay within the target network for a long period of time. Whereas a cybercriminal group will stay long enough to download a database full of credit card numbers, an APT group will maintain access within a network for as long as possible.
- **Threat**: To organizations that face APT groups, they are most definitely a threat. APT threat actors conduct their attacks with a specific objective and have the necessary infrastructure and skillset to attack targets such as large corporations, the military, and government organizations.

Threat intelligence is a wide field of study with many elements that are tied together. In the end, threat intelligence should drive action within an organization. What that action may be is often decided after careful evaluation of the threat intelligence. This involves understanding the type of threat intelligence being reviewed and what advantage each of those types provides the organization.

Threat intelligence types

When discussing the wide variety of information types and datasets that constitute threat intelligence, they often fall into one of three main categories:

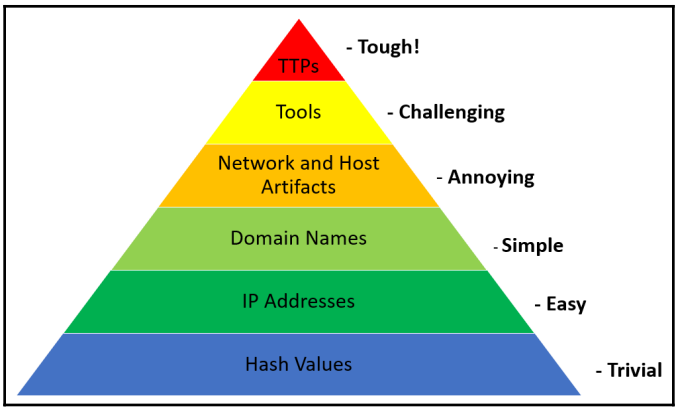
- **Tactical threat intelligence**: This is the most granular of the three threat intelligence categories. Information in this category involves either **Indicators of Compromise** (**IOCs**), **Indicators of Attacks** (**IOAs**), or **Tactics, Techniques, and Procedures** (**TTPs**):
    - **IOCs**: An IOC is an artifact observed on a system that is indicative of a compromise of some sort. For example, a C2 IP address or an MD5 hash of a malicious file are both IOCs.
    - **IOAs**: An IOA is an artifact observed on a system that is indicative of an attack or an attempted attack. This can be differentiated from an IOC, as an IOA does not indicate that a system was compromised, but rather attacked, due to indicators left by an adversary attacking a system. An example may be connection attempts left in a firewall log that are indicative of an automated port scan utilizing Nmap or another network scanning tool.
    - **TTPs**: Humans are creatures of habit and, as a result, cyber attackers often develop a unique methodology to how they attack a network. For example, a cybercriminal group may favor a social engineering email that has an Excel spreadsheet that executes a remote access Trojan. From there, they may attempt to access the credit card **point of sale** (**POS**) device and infect it with another piece of malware. How this group executes such an attack is considered to be their TTPs.

- **Operational threat intelligence**: The past decade has seen more and more coordinated attacks that do not just target one organization but may target an entire industry, region, or country. Operational threat intelligence is data and information about the wider goal of cyberattacks and cyber threat actors. This often involves not just examining the incident response team's own organization, but examining how cyber threat actors are attacking the larger industry. For example, in returning to a previous example where incident responders at a healthcare institution were preparing for an attack, a wider knowledge of what types of attacks are occurring at similar sized and staffed healthcare institutions would be helpful in aligning their own security controls to the prevalent threats.
- **Strategic threat intelligence**: Senior leadership such as the CIO or CISO often must concern themselves with the strategic goals of the organization alongside the necessary controls to ensure that the organization is addressing the cyber threat landscape. Strategic threat intelligence examines trends in cyberattacks, what cyber threat actors are prevalent, and what industries are major targets. Other key data points are changes in technology that a threat actor or group may leverage in an attack.

The best use of threat intelligence is to understand that each one of these types can be integrated into an overall strategy. Leveraging internal and external threat intelligence of all three types provides key decision makers with an understanding of the threat landscape; managers with the ability to implement appropriate security controls and procedures; and analysts the ability to search for ongoing security issues or to prepare their own response to a cyberattack.

Pyramid of pain

A useful construct for describing the various types of IOCs and IOAs that an adversary can leverage and their ability to modify them during an attack is the pyramid of pain. This construct, developed by David Bianco, describes the relationship between the IOCs, IOAs, and TTPs that an attacker makes available through observations by the defender and the attacker's ability to change those indicators. The following diagram shows the relationship to the various indicators and the work effort necessary to modify them in order to bypass security controls:



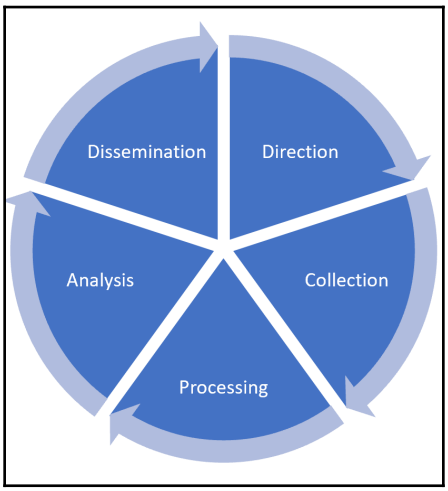
For example, an attacker may have crafted a piece of malware that spreads through lateral movement via the Windows SMB protocol. To bypass traditional signature-based malware prevention tools, the attacker uses a polymorphic virus that changes its hash every time it is installed. This change allows the piece of malware to bypass this control. Other indicators such as network or host artifacts are harder to change for the attacker and, as a result, responders have a greater chance of successfully stopping an attack by aligning their security controls at the top layers of the pyramid.

From a threat intelligence perspective, the pyramid of pain allows responders to align threat intelligence requirements with what would be useful from a long-term strategy. Having detailed information and intelligence about the TTPs in use by threat actors will provide more insight into how the threat actor operates. Lower-level indicators such as the IP addresses of C2 servers are useful, but responders do need to understand that these can easily be changed by the adversary.

Threat intelligence methodology

Threat intelligence goes through a feedback cycle in order to keep pace with an ever-changing environment. While there are several methodologies that can place context around this challenge, one that is often utilized is the cycle of intelligence that is used by the U.S. Department of Defense.

This cycle provides the framework and a starting point for organizations to incorporate threat intelligence into their operations:



The phases are explained as follows:

- **Direction**: Decision makers such as the CISO, information security personnel, or incident response analysts set down what threat intelligence is required. In determining the requirements for intelligence, it is a good practice to identify the users of each of the types of threat intelligence previously discussed. For example, a CISO might want threat intelligence about what trends in cyberattacks against hospitals are anticipated in the next year. An incident response analyst may require intelligence on what individual IOCs of malware are being seen in other healthcare institutions. The organization may also start by looking at what critical systems and applications are in use, as well as the critical data they are trying to protect. Another good starting point is if an organization already has some information about what types of cyber threats they may face.
- **Collection**: In the collection stage, the organization obtains the data and information from its sources. In terms of cyber threat intelligence, this can come from government organizations such as government-sponsored CERTs or through third-party organizations that sell threat intelligence. Finally, there are a great many **Open Source Intelligence** (**OSINT**) feeds that an organization can leverage.
- **Processing**: The sheer amount of intelligence that an organization may obtain can be staggering. During the processing stage, the organization takes the raw data, evaluates it, determines the relevance and reliability of the data, and then collates it for the next step.

- **Analysis**: During the analysis stage, the organization evaluates the data that has been processed and combines it with other data from other sources. From here, it is interpreted, and the finished product can be deemed *curated* or properly evaluated threat intelligence.
- **Dissemination**: The newly curated threat intelligence is then sent to the various users within the organization for use.

The cyclical nature of this methodology ensures that feedback is part of the process. Those analysts involved in the collection and processing should make sure that they receive feedback on the relevancy and veracity of the intelligence that is disseminated. From here, they would be able to tune the intelligence product over time. This ensures the highest level of relevancy and fidelity of intelligence consumed by end users.

Threat intelligence direction

There is a great deal of information, and data points, available to an organization in terms of threat intelligence. One of the major hurdles that organizations will have to jump over is in determining what their threat intelligence requirements are. With the depth of threat intelligence available, it is necessary to sift out the noise and focus only on threat intelligence that drives action within the organization.

In determining an organization's threat intelligence requirements, it is advisable to examine what actors pose a threat to the organization and how those threat actors would conduct an attack against the organization's network. To determine this, there are resources such as the MITRE ATT&CK methodology, the cyber kill chain, and the diamond model that can aid an organization in determining what their threat landscape is like and better align their threat intelligence utilization to meet that threat.

Cyber kill chain

The cyber kill chain is a concept that was first authored by three researchers at Lockheed-Martin.
(`https://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf`). The cyber kill chain outlines the stages of a network penetration that an attacker would have to go through to reach their ultimate goal. From here, organizations can extrapolate the various methods and IOCs that the organization may observe using detection capabilities enhanced with threat intelligence.

The cyber kill chain breaks down a network attack into seven steps that the attacker will progress through:

- **Reconnaissance**: Attackers often spend a considerable amount of time reviewing open source intelligence such as social media, corporate websites, and domain registrations to map the externally facing network of a target organization. Other reconnaissance methods include using network mapping and scanning tools such as Nmap and Netcat to determine open ports or enabled services. Reconnaissance activities are often very difficult to detect as threat actors can conduct such attacks with no direct action or tune scanning so as to hide their efforts behind normal network traffic.
- **Weaponization**: After conducting their reconnaissance, threat actors will then craft their tools for actual penetration. For example, this can be a multistage malware payload that compromises a system. From an examination of the tools utilized in an attack, specific data points such as how the malware is packed or what exploits are used, can be combined to create a mosaic that is unique to the adversary, almost creating a DNA profile to compare against.
- **Delivery**: Threat actors need a vector to deliver their malware or exploit payload. They may make use of VPN connections or deliver malware attached to a Word document emailed to an employee of the target organization.
- **Exploitation**: In this stage, a threat actor either leverages a vulnerability within the target network or the functionality of toolsets such as PowerShell.
- **Installation**: To gain more than a temporary foothold in the target organization, the threat actor will install their exploit or malware. This can even include the modification of settings or other functions on a compromised system.
- **C2**: To control the system once the installation has been successful, the threat actor has to configure a remote C2 channel back to a central server. From here, they are able to maintain control, load additional exploits or malware, and observe the target organization's actions.
- **Actions on the objective**: Once the previous six steps have been completed, the threat actor moves onto accomplishing the objective of the penetration. For retail targets, this may mean infecting POS devices with malware and obtaining credit card numbers. In government organizations, it may be acquiring a database of confidential data to sell.

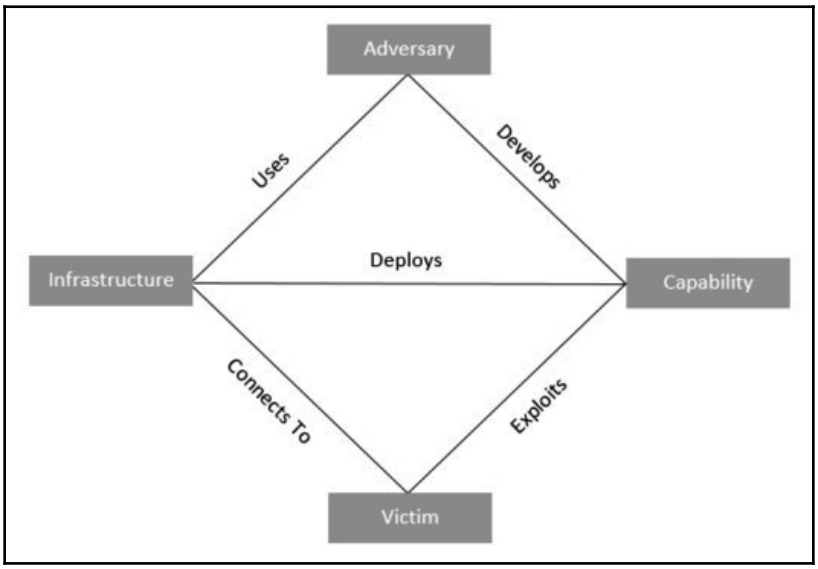
By working through these various steps, an organization can see where individual IOCs and more general TTPs about threat actors can be obtained. One technique that is often utilized is to determine what threats are applicable to an organization, and then map them out at each stage to the individual IOCs that they will need specific threat intelligence to address.

For example, they may have a report about a cybercriminal group that targets POS devices. From here, they realize that they would need to understand what the IOCs would be for the initial tools configured in the weaponization stage. Next, they would examine the TTPs surrounding how the threat actor delivers the exploit or malware. The organization would then need to understand how the threat actor exploits the network either through vulnerabilities or utilizing native utilities. The installation of an exploit or malware will produce IOCs in running memory and the registry settings of a compromised system. Having access to the specific IOCs in those areas would assist the organization with developing additional detective capabilities or the ability to find these IOCs during an incident investigation.

Diamond model

The diamond model of intrusion analysis is a methodology used to describe the process of differentiating APT threats from their specific attributes. The diamond is comprised of four components: **Adversary**, **Infrastructure**, **Capabilities**, and **Victim**.

The model attempts to determine the interplay between each of these four groups:



For example, take a simple malware attack. The **Adversary** is going to use a custom piece of malware. Their ability to develop custom malware indicates their **Capability**. The **Adversary** then utilizes their capability to deploy the malware via a compromised web server or infrastructure. This connects to the **Victim** where the capability exploits a social engineering vulnerability.

This simple example highlights just a small sliver of how the diamond model can be utilized to categorize attacks. Therefore, it is recommended that a deeper exploration be undertaken by reviewing the diamond model paper, which can be downloaded at `http://www.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf`. By integrating this model, an organization can have a better understanding of the threats they face and how those threats interact during an attack against their infrastructure. From here, they will be able to align their threat intelligence requirements to better fit their unique challenges.

One key reference in determining threat intelligence requirements is the MITRE ATT&CK wiki located at `https://attack.mitre.org/wiki/Main_Page`. The **Adversarial Tactics, Techniques, & Common Knowledge** (**ATT&CK**) is an extensive collection of tactics and techniques in use by adversaries. The tactics include each stage of the *kill chain* and includes an in-depth analysis of each technique.

ATT&CK also includes detailed information about the various APT groups that have been identified by various information security and incident response research organizations. Entries in the ATT&CK platform are also thoroughly documented and footnoted to allow analysts to view both a digest and a comprehensive report.

The value of the ATT&CK wiki is that it allows analysts to have detailed information about threat groups, their techniques, and their tactics. This can better inform the other models such as the cyber kill chain and the diamond model. This allows organizations to fully understand what threats they face and align their threat intelligence requirements to fulfill that need.

Threat intelligence sources

There are three primary sources of threat intelligence that an organization can leverage. Threat intelligence can be produced by the organization in an internal process, acquired through open source methods, or, finally, through third-party threat intelligence vendors. Each organization can utilize their own internal processes to determine what their needs are and what sources to leverage.

Internally developed sources

The most complex threat intelligence sources are those that an organization internally develops. This is due to the infrastructure that is needed to obtain the individual IOCs from malware campaigns and TTPs from threat actors. To obtain IOCs, the organization can make use of honeypots or other deliberately vulnerable systems to acquire unique malware samples. They will also need to have the expertise and systems available to not only evaluate suspected malware but reverse engineer it. From there, they would be able to extract the individual IOCs that can then be utilized.

Other systems such as SIEM platforms can be utilized to track an attacker's TTPs as they attempt to penetrate a network. From here, a **Security Operations Center** (**SOC**) analyst can record how different attackers go about their penetration attempts. With this information, the organization can build a profile of specific groups. This can aid in the alignment of security controls to better prevent or detect network intrusions.

Developing threat intelligence internally requires expertise in areas such as malware analysis, network, and host-based forensics. Furthermore, the infrastructure required is often cost-prohibitive. As a result, organizations are often forced to rely on third-party providers on what is shared openly among other organizations.

Commercial sourcing

An alternative to internal sourcing is to contract with a threat intelligence vendor. These organizations utilize their own personnel and infrastructure to acquire malware, analyze attacks, and conduct research on various threat groups. Commercial threat intelligence providers will often process the threat intelligence so that it is tailored to the individual client organization.

Often, commercial vendors provide SIEM and SOC services for a variety of clients utilizing a common SIEM platform. From here, they can aggregate samples of malware and attacks across various enterprises that span the entire world. This allows them to offer a comprehensive product to their clients. This is one of the distinct advantages of utilizing a commercial service. This is in addition to the cost savings that come from transferring the cost to a third party.

Open source

One sourcing area that has become quite popular with organizations of every size is the OSINT providers. Community groups, and even commercial enterprises, make threat intelligence available to the general public free of charge. Groups such as SANS and US-CERT provide specific information about threats and vulnerabilities. Commercial providers such as AlienVault provide an **Open Threat Exchange** (**OTX**) that allows a user community to share threat intelligence such as IOCs and TTPs. Other commercial organizations will provide whitepapers and reports on APT groups or strategic threat intelligence on emerging trends within the information security industry. Depending on the organization, OSINT is often very useful and provides a low-cost alternative to commercial services.

The widespread use of OSINT has led to various organizations creating methods to share threat intelligence across organizations. Depending on the source, the actual way that an organization can obtain threat intelligence is dependent on how it is configured.

While not a completely exhaustive list, the following are some of the formats of cyber threat OSINT that is available:

- **OpenIOC**: OpenIOC was first developed so that Mandiant products, such as the Redline application utilized in `Chapter 6`, *Forensic Imaging*, could ingest threat intelligence and utilize it to search for evidence of compromise on the systems analyzed. It has evolved into an XML schema that describes the technical IOCs that an incident responder can use in determining whether a system has been compromised.
- **STIX**: The **Structured Threat Information Exchange** (**STIX**) is a product of the OASIS consortium. This machine-readable format allows organizations to share threat intelligence across various commercial and freeware threat intelligence aggregation platforms.
- **TAXII**: The **Trusted Automated Exchange of Intelligence Information** (**TAXII**) is an application layer protocol that shares threat intelligence over HTTPS. TAXII defines an API that can be utilized to share threat intelligence in the STIX format.
- **VERIS**: The **Vocabulary for Event Recording and Incident Sharing** (**VERIS**) is a comprehensive schema for standardizing the language of cybersecurity incidents. The one key problem that the VERIS schema attempts to solve is the lack of a standard way to document security incidents. VERIS provides a structure in which organizations have a defined way to categorize the variety of attacks that may occur. The VERIS schema also serves as the collection point of data provided by organizations that is incorporated into the Verizon Data Breach Study.

With a variety of intelligence sources available, one challenge that presents itself is the ability for organizations to aggregate, organize, and utilize threat intelligence. In the next section, a discussion of threat intelligence platforms will provide an insight into solving these issues.

Threat intelligence platforms

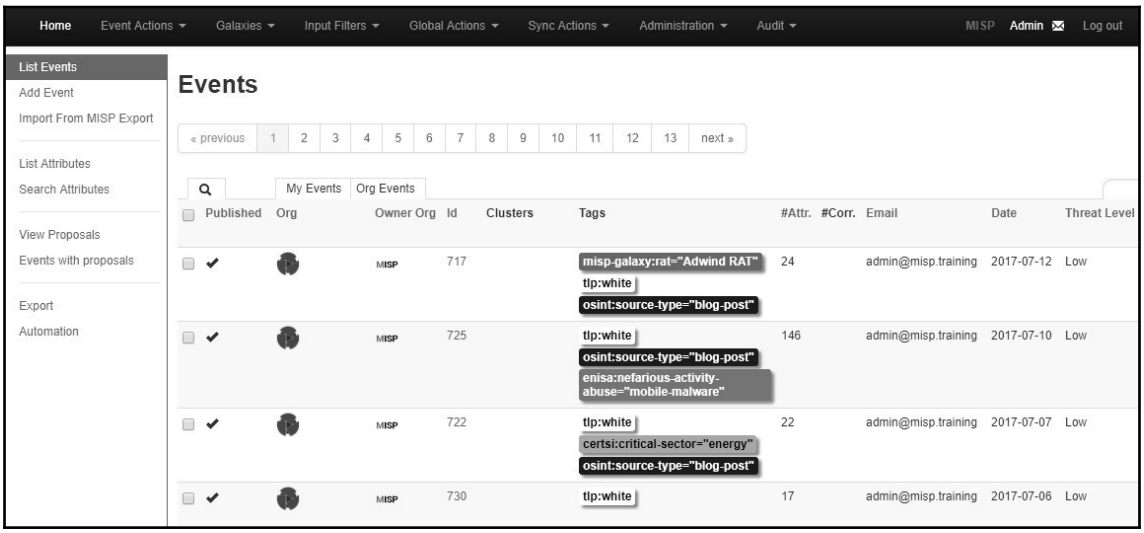
As organizations begin to aggregate threat intelligence, whether it is created internally or externally sourced, they will need a platform in which to aggregate it. Any platform should be able to store threat intelligence indicators as well as provide analysts with an easily searchable database that allows them to connect IOCs from an incident to the intelligence available. There are several commercial platforms available as well as freeware versions that provide analysts with this capability. It is up to the individual organization to determine which platform fits their needs.

MISP threat sharing

One freeware platform that is available is the **Malware Information Sharing Platform** (**MISP**). This community project has produced a software platform that can be used by analysts to store data about malware and other exploits. From here, they can share this intelligence within their team or other personnel. MISP is a feature-rich application with such functionality as searching, a correlation engine, the ability to export and import IOCs, and community support where users can share data.

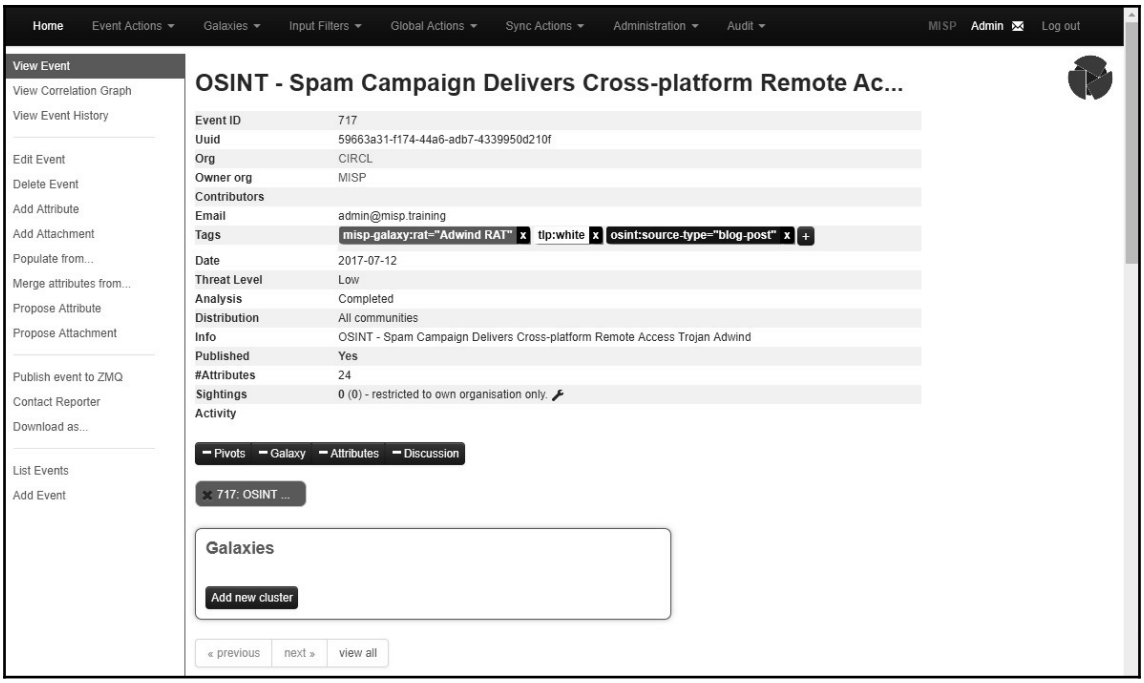
Installing MISP is dependent on the type of operating system platform in use. Complete directions are available at `https://github.com/MISP/MISP/tree/2.4/INSTALL`. The creators of MISP also have provided users with a complete installation on an **Open Virtualization Format** (**OVA**) virtual machine that can be downloaded for testing. The OVA file is available at `https://www.circl.lu/assets/files/misp-training/MISP_v2.4.77.ova`. This is a good option as it allows analysts to test the functionality of the application without having to populate the database.

For the following demonstration, the training version of MISP will be utilized. Once logged in, the following window will appear:

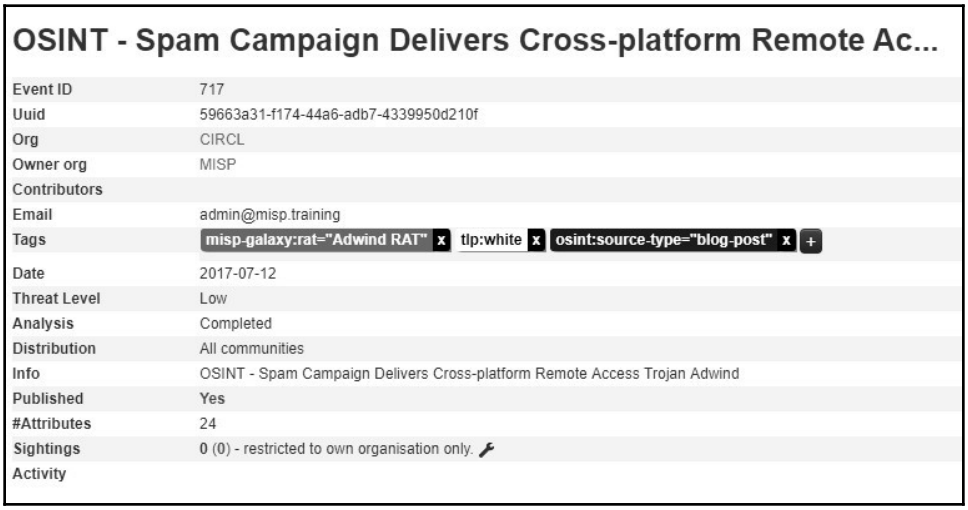


The window contains all of the events that the MISP database has on file. There is a good deal of data on this page, including tags that identify the classifications of events, the date that they were added, and basic information that allows analysts to quickly sort through the various entries.

Clicking on an **Event ID** or the **View** icon to the far right of an event brings up another window:



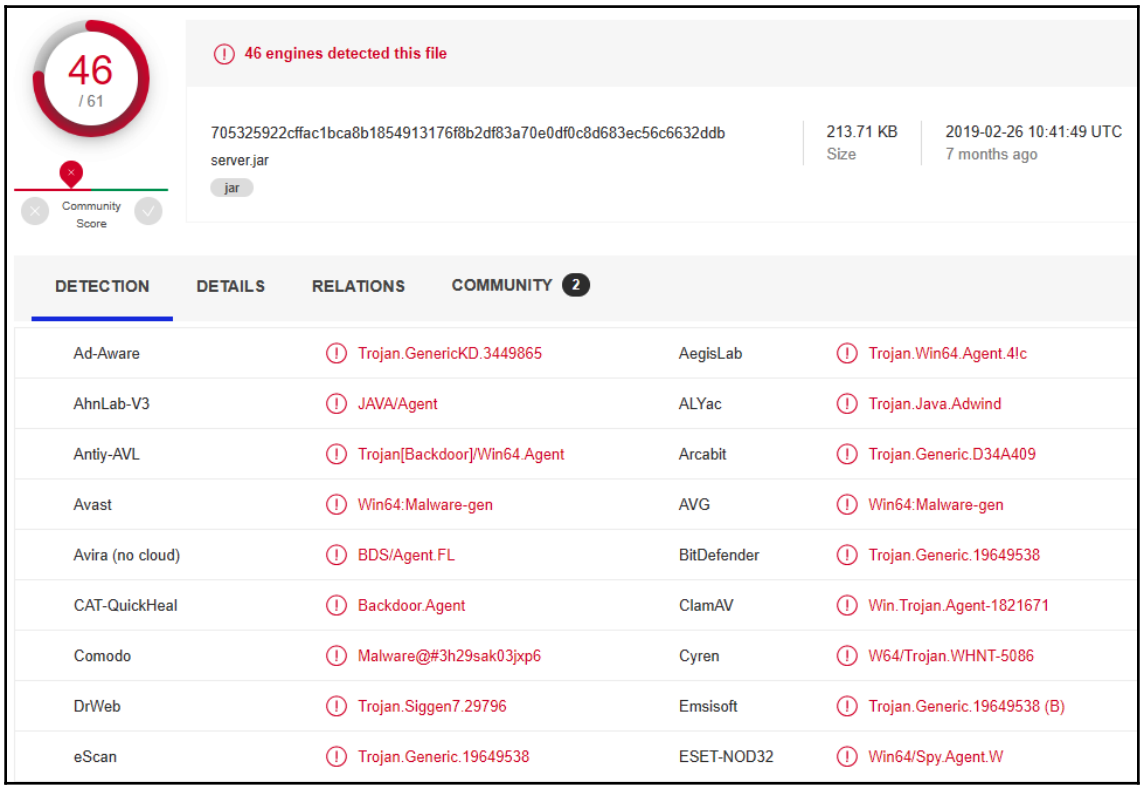
In this window, analysts will be provided with a good deal of intelligence regarding the specific event. First is the event data, which is an overview of the attributes of the IOCs contained within the event:



This data gives the analysts an overview of the event information that was available in the overall home window. Further down, the window reveals the specific elements of the event:

2017-07-12	External analysis	link	https://www.virustotal.com/file/705325922cffac1bca8b1854913176f8b2df83a70e0df0c8d683ec56c6632ddb/analysis/1499247775/	BKDR64_AGENT.TYUCT - Xchecked via VT: 705325922cffac1bca8b1854913176f8b2df83a70e0df0c8d683ec56c6632ddb
2017-07-12	External analysis	link	https://www.virustotal.com/file/97d585b6aff62fb4e43e7e6a5f816dcd7a14be11a88b109a9ba9e8cd4c456eb9/analysis/1499851519/	JAVA_ADWIND.AUJC - Xchecked via VT: 97d585b6aff62fb4e43e7e6a5f816dcd7a14be11a88b109a9ba9e8cd4c456eb9

Here, the specific Trojan indicated in the background information has been evaluated by VirusTotal and following the link indicates that 46 out of 61 antivirus providers have detected that this event is linked with a Trojan virus:



46 / 61

Community Score

46 engines detected this file

705325922cffac1bca8b1854913176f8b2df83a70e0df0c8d683ec56c6632ddb
server.jar
jar

213.71 KB
Size

2019-02-26 10:41:49 UTC
7 months ago

DETECTION DETAILS RELATIONS COMMUNITY 2

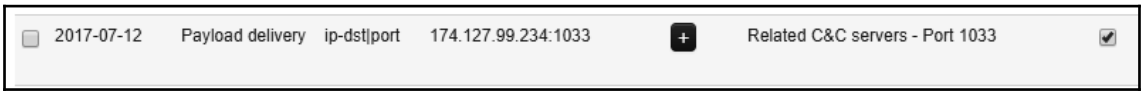
Ad-Aware	Trojan.GenericKD.3449865	AegisLab	Trojan.Win64.Agent.4!c
AhnLab-V3	JAVA/Agent	ALYac	Trojan.Java.Adwind
Antiy-AVL	Trojan[Backdoor]/Win64.Agent	Arcabit	Trojan.Generic.D34A409
Avast	Win64:Malware-gen	AVG	Win64:Malware-gen
Avira (no cloud)	BDS/Agent.FL	BitDefender	Trojan.Generic.19649538
CAT-QuickHeal	Backdoor.Agent	ClamAV	Win.Trojan.Agent-1821671
Comodo	Malware@#3h29sak03jxp6	Cyren	W64/Trojan.WHNT-5086
DrWeb	Trojan.Siggen7.29796	Emsisoft	Trojan.Generic.19649538 (B)
eScan	Trojan.Generic.19649538	ESET-NOD32	Win64/Spy.Agent.W

The real value of the MISP platform is the IOCs that are associated with this event. Navigating down the window, the analyst can then view the individual IOCs associated with this malware addressed in this event:

2017-07-12	Network activity	url	https://nup.pw/Qcaq5e.jar	+	Files and URLs related to Adwind/jRAT
2017-07-12	Network activity	url	http://vacanzaimmobiliare.it/testla/WebPanel/post.php	+	Related C&C servers:
2017-07-12	Network activity	url	http://ccb-ba.adv.br/wp-admin/network/ok/index.php	+	Files and URLs related to Adwind/jRAT
2017-07-12	Network activity	url	https://nup.pw/e2BXtK.exe	+	Files and URLs related to Adwind/jRAT

From here, the analyst is able to identify specific URLs that are associated with either C2 communications or if the malware is part of a multistaged attack. The analyst can then either correlate these URLs with logs on the firewall or block those URLs through a web proxy to contain a possible malware outbreak.

Further down the window is more specific network information:



2017-07-12	Payload delivery	ip-dst|port	174.127.99.234:1033	+	Related C&C servers - Port 1033	☑

This intelligence allows analysts to drill down to a specific IP address and either block or craft an alert on the network exit point to determine whether there are any additional systems that have been compromised.

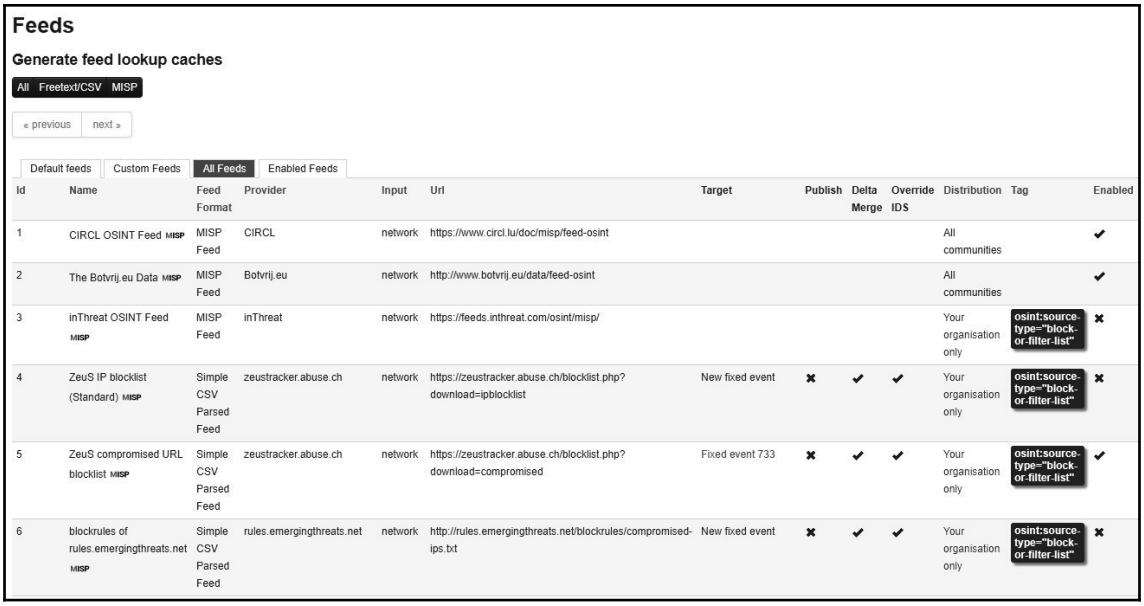
MISP also allows analysts to see specific files associated with the event:

2017-07-12	Payload delivery	sha1	9ce4518ebcb5be6d1f0b5477fa00c26860fe9a68	+	JAVA_ADWIND.AUJC - Xchecked via VT: 97d585b6aff62fb4e43e7e6a5f816dcd7a14be11a88b109a9ba9e8cd4c456eb9	☑
2017-07-12	Payload delivery	md5	781fb531354d6f291f1ccab48da6d39f	+	JAVA_ADWIND.AUJC - Xchecked via VT: 97d585b6aff62fb4e43e7e6a5f816dcd7a14be11a88b109a9ba9e8cd4c456eb9	☑
2017-07-12	Payload delivery	sha256	705325922cffac1bca8b1854913176f8b2df83a70e0df0c8d683ec56c6632ddb	+	BKDR64_AGENT.TYUCT	☑

From here, analysts can utilize memory or disk images and search for matching hashes. This allows response analysts to drill down on the specific files that need to be addressed by any analysis and recovery activity.

Threat intelligence needs to be timely in order for it to be effective. In the realm of cyberattacks, intelligence goes stale very quickly. Any platform that aggregates threat intelligence should have the ability to be updated. MISP has the ability to integrate various threat intelligence feeds and add those records to the event view:

1. To demonstrate, navigate to the **Sync Actions** tab and then click on **List Feeds**. The following window will open:



Feeds

Generate feed lookup caches

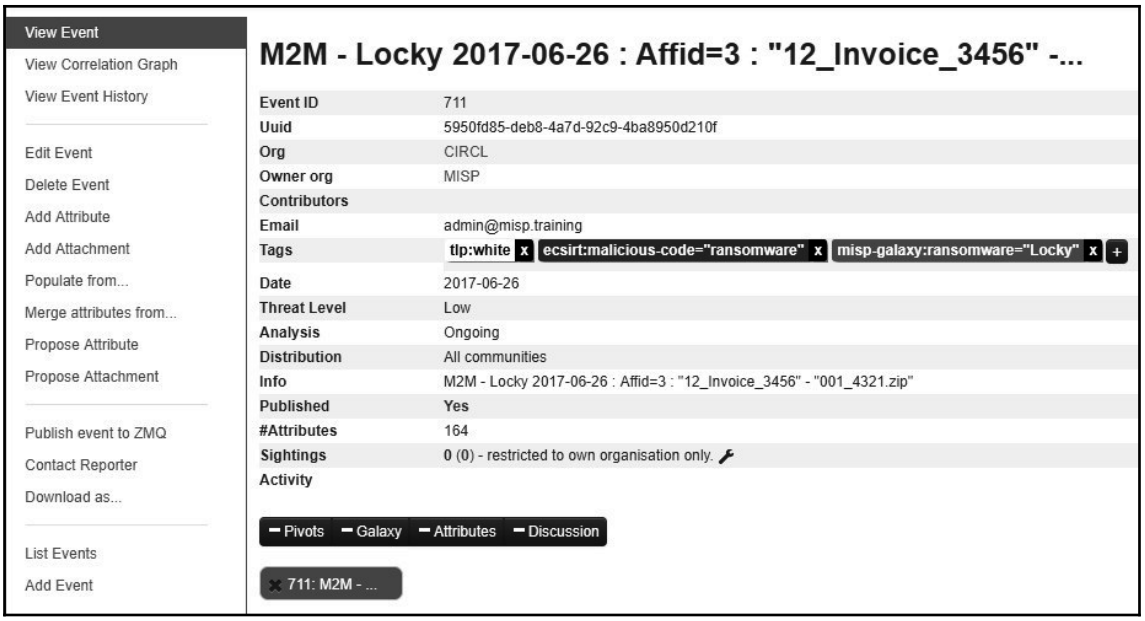
All Freetext/CSV MISP

« previous next »

Default feeds Custom Feeds All Feeds Enabled Feeds

Id	Name	Feed Format	Provider	Input	Url	Target	Publish	Delta Merge	Override IDS	Distribution	Tag	Enabled
1	CIRCL OSINT Feed MISP	MISP Feed	CIRCL	network	https://www.circl.lu/doc/misp/feed-osint					All communities		✔
2	The Botvrij.eu Data MISP	MISP Feed	Botvrij.eu	network	http://www.botvrij.eu/data/feed-osint					All communities		✔
3	inThreat OSINT Feed MISP	MISP Feed	inThreat	network	https://feeds.inthreat.com/osint/misp/					Your organisation only	osint:source-type="block-or-filter-list"	✖
4	ZeuS IP blocklist (Standard) MISP	Simple CSV Parsed Feed	zeustracker.abuse.ch	network	https://zeustracker.abuse.ch/blocklist.php?download=ipblocklist	New fixed event	✖	✔	✔	Your organisation only	osint:source-type="block-or-filter-list"	✖
5	ZeuS compromised URL blocklist MISP	Simple CSV Parsed Feed	zeustracker.abuse.ch	network	https://zeustracker.abuse.ch/blocklist.php?download=compromised	Fixed event 733	✖	✔	✔	Your organisation only	osint:source-type="block-or-filter-list"	✔
6	blockrules of rules.emergingthreats.net MISP	Simple CSV Parsed Feed	rules.emergingthreats.net	network	http://rules.emergingthreats.net/blockrules/compromised-ips.txt	New fixed event	✖	✔	✔	Your organisation only	osint:source-type="block-or-filter-list"	✖

2. From here, navigate to the far right-hand corner and to the down arrow surrounded by a circle. This will fetch all of the events from that source. For example, click on the button for source number five. After syncing, an additional entry is made within the events window:



Depending on the type of threat intelligence feed that the organization uses, MISP is able to enrich its dataset by utilizing a connection from those sites. This allows analysts to keep their own dataset from various providers in one searchable location. From here, they can utilize this data as part of their proactive detection capability or to assist during an incident investigation.

Using threat intelligence

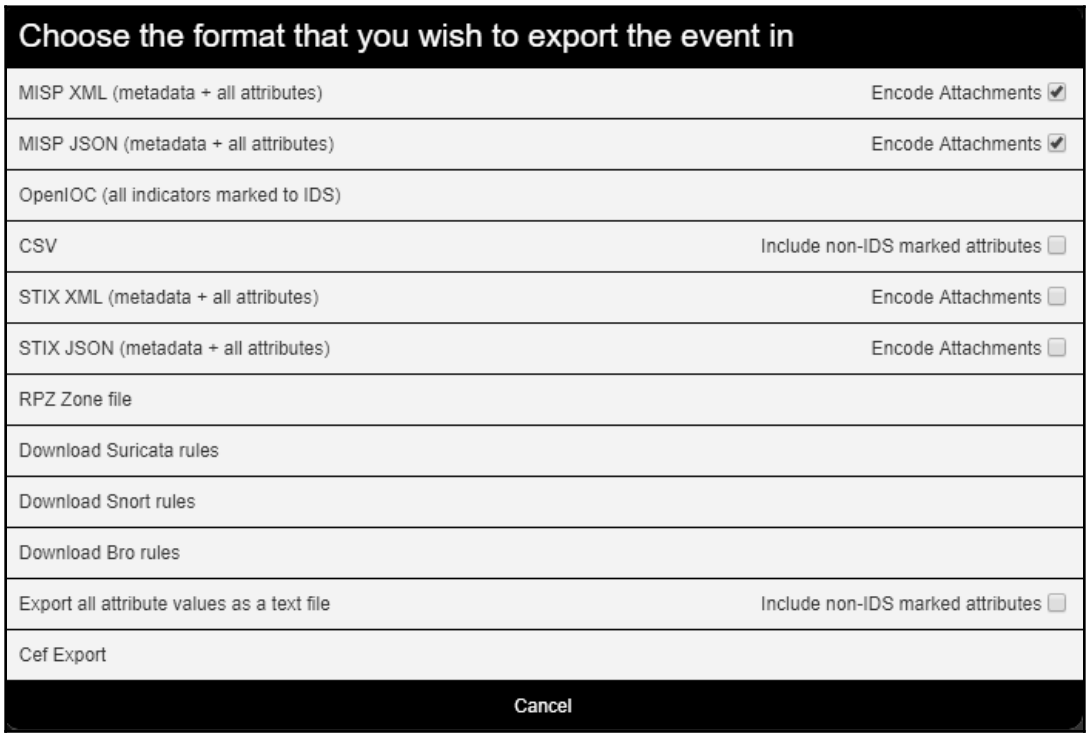
There are two distinct advantages to integrating threat intelligence into an organization's security methodology and response capability. From the proactive approach, organizations can utilize this data to increase the effectiveness of their detective and preventive controls. This allows organizations to enhance their ability to either block attacks through such mechanisms as blacklisting known malware sites or through their detective capability by alerting you to specific host behavior that is indicative of a compromise. On a reactive stance, organizations can integrate threat intelligence into their existing toolset and bring those to an investigation. This allows them to find evidentiary items that may have gone undetected with traditional information.

Proactive threat intelligence

Threat intelligence providers will often provide CSIRT and SOC teams with threat intelligence that can be easily fed into their SIEM of choice. This allows these teams to enhance their detective capability with intelligence that is timely, possibly allowing them to keep pace with the current threats and increase the probability that they will detect one or more of these threats before damage can be done.

In the MISP platform, events with specific IOCs can have those IOCs converted into several different types of detective rules. For example, an organization is concerned about ransomware impacting the organization and wants to enhance their detective capability. Event number 711 in the MISP platform is associated with the *Locky* ransomware campaign. Clicking on the event number produces the following screen:

Navigate to the left-hand column and click on **Download as...**. This produces the following window:



From a proactive/detective perspective, responders can export the IOCs as a rule for three open source network intrusion detection systems. In this case, rules can be exported for Suricata, Snort, or Bro. Each of these is an open source network intrusion detection system that examines network traffic and compares that traffic against a defined ruleset.

Each of these tools has a specific syntax for the detection rules. For example, the Snort rule will be reviewed. Download the Snort rule associated with this event by clicking on **Download Snort rules**. The file will be downloaded. Once completed, open the file with a text editor and the various rules associated with the event can be seen:

```
1  #This part might still contain bugs, use and your own risk and report any issues.
2  #
3  # MISP export of IDS rules - optimized for snort
4  #
5  # These NIDS rules contain some variables that need to exist in your configuration.
6  # Make sure you have set:
7  #
8  # $HOME_NET - Your internal network range
9  # $EXTERNAL_NET - The network considered as outside
10 # $SMTP_SERVERS - All your internal SMTP servers
11 # $HTTP_PORTS   - The ports used to contain HTTP traffic (not required with suricata export)
12 #
13 alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg: "MISP e711 [] Outgoing HTTP URL:
14 alert udp any any -> any 53 (msg: "MISP e711 [] Hostname: 1010technologies.com"; content:"|
15 alert tcp any any -> any 53 (msg: "MISP e711 [] Hostname: 1010technologies.com"; content:"|
```

There are a number of rules that can be set with this download. Examining line 14, for example, indicates that the particular rule is setting Snort to alert you if there is any attempted connection over UDP port `53` to the `101otechnologies.com` host. This host is, in some fashion, associated with this ransomware campaign. If this rule is incorporated, an organization would be alerted to this type of connection and be able to respond much quicker than finding out about ransomware activity when a user contacts the helpdesk indicating that their files have been encrypted.

The advantage that Snort rules have is that a great deal of commercial IDS/IPS vendors have the capability to ingest Snort rules into their own proprietary platform. This allows SOC and CSIRT personnel to load these rules from various sources, thereby enhancing their capabilities without having to have several different platforms to maintain.

Reactive threat intelligence

During an investigation, the CSIRT or analysts may come across a situation where an incident investigation seems to have stalled. This could be due to the fact that the analysts know something is wrong or have indicators of a compromise but no concrete evidence to point in a specific direction. Threat intelligence can be leveraged by analysts to enhance their ability to discover previously undiscovered evidence.

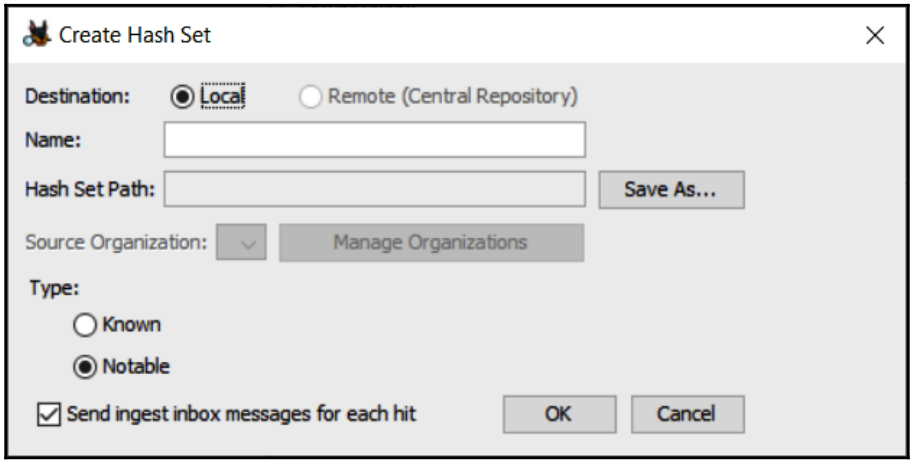
Autopsy

A large number of tools that can ingest threat intelligence are available to incident response analysts. For example, disk forensic platforms discussed in `Chapter 8`, *Analyzing System Memory*, have the ability to ingest hashes from threat intelligence feeds to search for IOCs. In addition to commercial disk forensic tools, the Autopsy platform can conduct searches against a hash set. Navigating back to the export format in MISP, there is the ability to download a `.csv` file of the event indicators. For event 711, download the CSV file. Next, filter the data and select the hash values from the **type** column. This produces the following list:

uuid	event_i	category	type	value	comme	to_ids	date
5950fd86-	711	Artifacts dropped	md5	8cd9f803947badddbfafc584edfdeebb		1	20170627
5950fd87-	711	Artifacts dropped	md5	a0d81f0bffb0e20a34191385031cf17a		1	20170627
59520c6b-	711	Artifacts dropped	sha1	f5fce485a72ab82a5e5b48b98befd5e0568a83e1	#NAME?	1	20170627
59520c6b-	711	Artifacts dropped	sha256	83b366204ef60cca5468c2db1baadeb7590f97493c451fa005f9b583ce691133	- Xchecked	1	20170627
59520c6b-	711	Artifacts dropped	sha1	3e19f754ea0fef9e62d91dfd4f22e6c73240bcbc	- Xchecked	1	20170627
59520c6b-	711	Artifacts dropped	sha256	8015133c16d41fdfbeb5f86f5d82ffb124a131ed012375d3cf70babe2f440ac8	#NAME?	1	20170627

From here, the hash values can be loaded into Autopsy:

1. First, in Autopsy, click on **Tools** and then **Options**. Then, click on **Hash Sets** and then **New Hash Set**. The following window will appear:



2. Enter a name for the hash set. A suggestion is to use a title and the MISP event number 711. Click on **Save As...** and navigate to where the database will be saved. Leave the default settings in place. This will indicate a hit on any of the hash files located. Click on **OK**.
3. In the next window, click on **Add Hashes to Database**. Copy the hashes to the clipboard from the CSV file and then right-click on the blank space and select **Paste**.
4. The hashes are now loaded. Click on **Add Hashes to Database**.

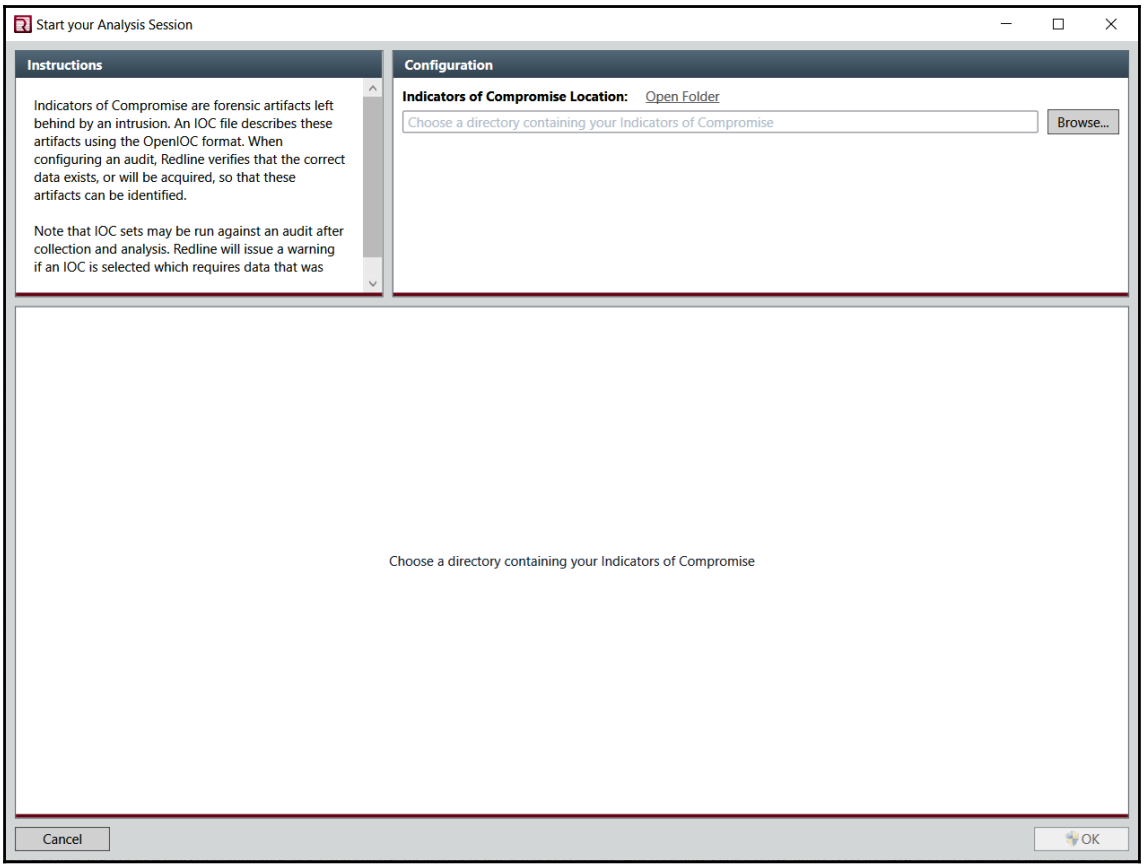
This capability allows analysts to search through disk images for matching hashes. This is a much more efficient way to search for evidence than attempting to find the files through other methods. Autopsy also allows for different databases depending on the incident. This ability to continually feed updated information allows analysts to find evidence of a new type of compromise from an event from a week or two ago that would have gone undetected if using traditional searching.

Adding IOCs to Redline

Threat intelligence can also be utilized with Redline. Redline allows for searching for IOCs through a collector, or IOCs can be loaded and searched in an existing memory capture. For example, if analysts would like to search for matching IOCs in a memory image, they would first open the memory image:

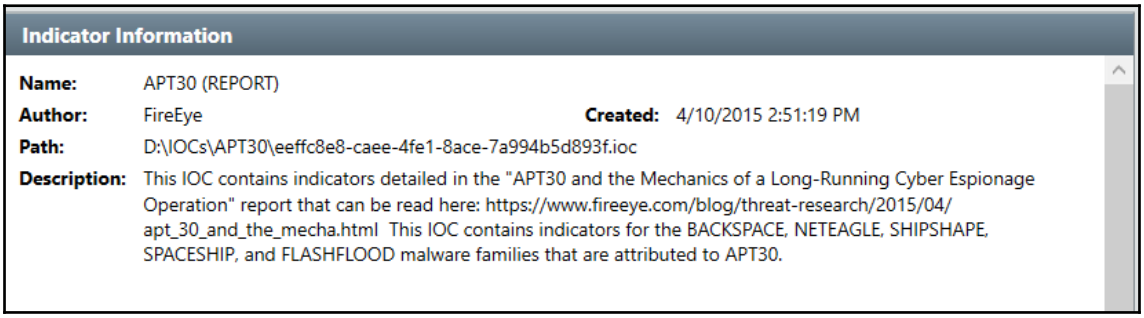
1. In the lower-left corner, click on the **IOC Reports** tab. This will create a new button titled **Create a New IOC Report**.

The following window will appear:

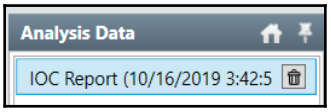


Redline has the ability to ingest IOCs within the OpenIOC format. Analysts should create a folder on their system where the IOC files can be placed, as Redline will not read a single file but all IOC files within the folder.

2. Click on **Browse** and navigate to the IOC folder. Then, IOCs are loaded and specific information is loaded into the Redline platform:



3. Clicking on **OK** runs the IOCs against the memory capture. Depending on the number of IOC files and the memory image, this could take several minutes. Then, once completed, the **IOC Report** will be listed under the **Analysis Data** section. Any hits on the IOCs will be listed there:

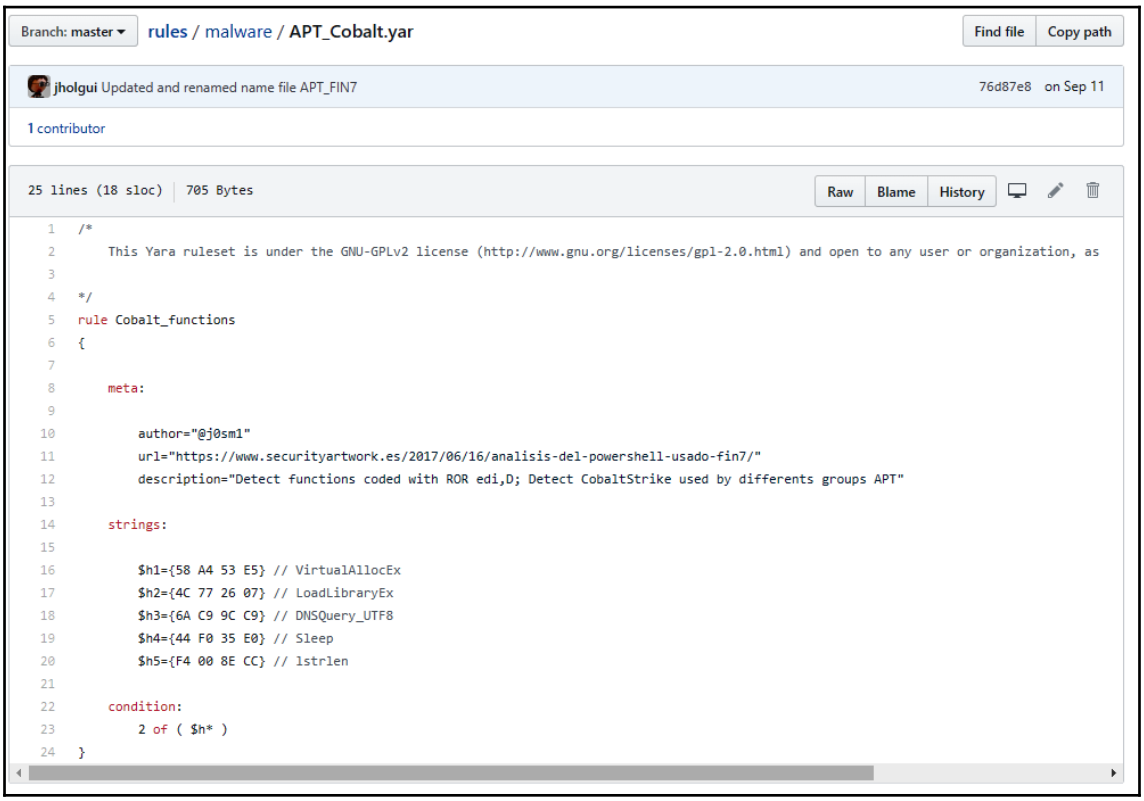


Next, we will look at two integrated tools called Yara and Loki.

Yara and Loki

Two integrated tools that allow for leveraging threat intelligence during an incident are Yara and Loki. Yara is often referred to as the *Swiss Army Knife* of pattern matching. It was created to assist malware researchers with classifying malware. Through the use of Boolean expressions and strings, a malware sample can be classified.

For example, the Yara rule (`https://github.com/Yara-Rules/rules/blob/master/malware/APT_Cobalt.yar`) for a variation of the CobaltStrike looks like this:



Branch: master ▾ rules / malware / APT_Cobalt.yar Find file Copy path

jholgui Updated and renamed name file APT_FIN7 76d87e8 on Sep 11

1 contributor

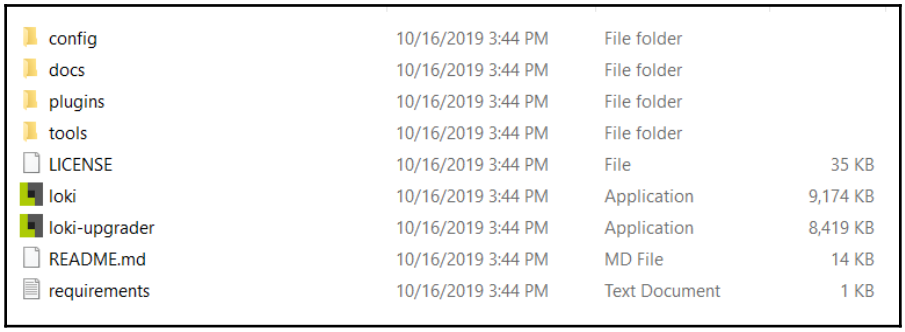
25 lines (18 sloc) | 705 Bytes Raw Blame History

```
/*
    This Yara ruleset is under the GNU-GPLv2 license (http://www.gnu.org/licenses/gpl-2.0.html) and open to any user or organization, as

*/
rule Cobalt_functions
{

    meta:

        author="@j0sm1"
        url="https://www.securityartwork.es/2017/06/16/analisis-del-powershell-usado-fin7/"
        description="Detect functions coded with ROR edi,D; Detect CobaltStrike used by differents groups APT"

    strings:

        $h1={58 A4 53 E5} // VirtualAllocEx
        $h2={4C 77 26 07} // LoadLibraryEx
        $h3={6A C9 9C C9} // DNSQuery_UTF8
        $h4={44 F0 35 E0} // Sleep
        $h5={F4 00 8E CC} // lstrlen

    condition:
        2 of ( $h* )
}
```

The preceding rule configures Yara to alert you to any strings found as `Cobalt_functions`.

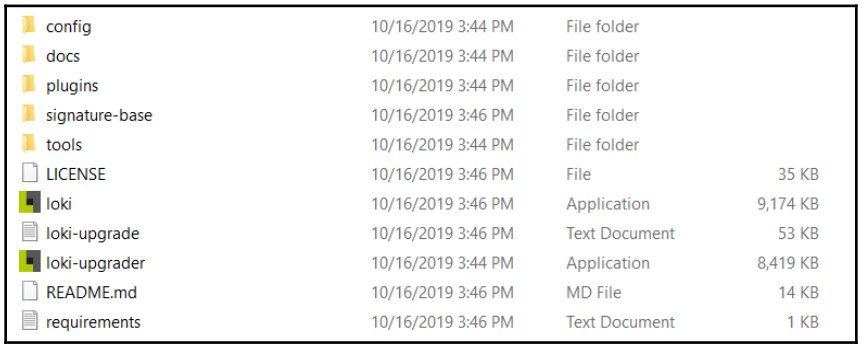
Yara is available as a tool for use in malware research, but one of the useful features is the ability to integrate the Yara functionality into other tools. One such tool is Loki—a simple IOC scanner (`https://github.com/Neo23x0/Loki`). This lightweight platform allows incident response analysts to scan folders, files, or even entire volumes for IOCs such as Yara rules, known bad file hashes, filename IOCs, and known C2 servers. Out of the box, Loki has an extensive library of IOCs that are updated regularly:

1. To check a system volume for specific IOCs, download and extract Loki to a USB device. Open the `Loki` folder and the following files are found:



config	10/16/2019 3:44 PM	File folder	
docs	10/16/2019 3:44 PM	File folder	
plugins	10/16/2019 3:44 PM	File folder	
tools	10/16/2019 3:44 PM	File folder	
LICENSE	10/16/2019 3:44 PM	File	35 KB
loki	10/16/2019 3:44 PM	Application	9,174 KB
loki-upgrader	10/16/2019 3:44 PM	Application	8,419 KB
README.md	10/16/2019 3:44 PM	MD File	14 KB
requirements	10/16/2019 3:44 PM	Text Document	1 KB

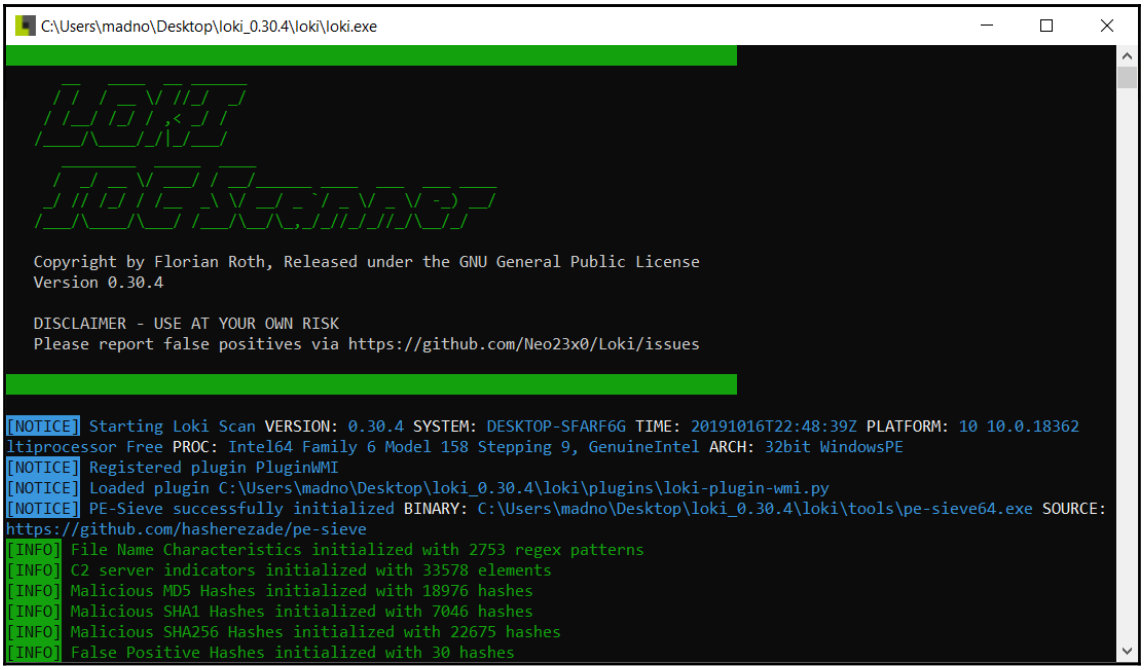
2. Loki has to be updated with the most current IOCs, so right-click on `loki-upgrader` and run as administrator. The upgrader will run, updating both the executable and the signature files. Once completed, the updater will close.
3. Navigate back to the Loki file and a new file called `signature-base` will have been added:



config	10/16/2019 3:44 PM	File folder	
docs	10/16/2019 3:44 PM	File folder	
plugins	10/16/2019 3:44 PM	File folder	
signature-base	10/16/2019 3:46 PM	File folder	
tools	10/16/2019 3:44 PM	File folder	
LICENSE	10/16/2019 3:46 PM	File	35 KB
loki	10/16/2019 3:46 PM	Application	9,174 KB
loki-upgrade	10/16/2019 3:46 PM	Text Document	53 KB
loki-upgrader	10/16/2019 3:44 PM	Application	8,419 KB
README.md	10/16/2019 3:46 PM	MD File	14 KB
requirements	10/16/2019 3:46 PM	Text Document	1 KB

This folder contains all of the IOCs that Loki can search for a volume against. This also allows analysts who create their own Yara rules to load them into the file as well, giving them the ability to customize the solution.

4. To run a scan of a system, right-click on the `loki` application and run it as an administrator. This will start the executable and open the following window:



After the ruleset is updated, Loki will then begin searching the volume for any matching patterns or IOCs:

From here, the analyst can take note of any hits and conduct an examination later. Another key feature is that Loki can be deployed on multiple systems as part of a triage of systems that have possibly been infected with a new strain of malware. For example, an incident response analyst may be able to search for the IOC of the Petya ransomware attack using Yara rules taken from a threat intelligence provider, such as Kaspersky's SecureList, which includes a download of the Yara rules.

From here, the Yara rules can then be fed into Loki or another platform and utilized to triage suspected systems.

The number of tools that an incident response analyst can bring to bear is increasing every day. These include commercial tools and freeware tools that integrate a variety of threat intelligence feeds and functionality. These tools can be used proactively to detect and alert as well as investigate an incident in progress. CSIRTs should make a concerted effort to examine these tools and integrate them into their processes. Doing so will aid them in detecting and efficiently investigating an incident.

Summary

Sun Tzu's *Art of War* includes the strategic concept of knowing your adversary and knowing yourself. Through this, you can be confident in your ability to prevail in the contest. Threat intelligence has quickly become a critical component of an organization's proactive security controls, as well as an important factor in its ability to respond to an incident. This chapter examined the emerging techniques and methodologies of cyber threat intelligence, and the sources, technology, and methods to put this data to use.

To move forward, organizations looking to leverage the advantages that threat intelligence provides must first understand the threat. From there, they can define their requirements and begin the intelligence process. Finally, by integrating their toolset to utilize threat intelligence, they can position themselves to have more effective proactive controls and the ability to respond efficiently. While threat intelligence may not remove the fear of an adversary entirely, it allows organizations a good deal more ammunition to combat today's threats. Threat intelligence also serves as an important function when looking at the proactive practice of identifying threats in an environment through threat hunting, which is the subject of the next chapter.

Questions

1. What is not a key element for intelligence?

   A) Indicator of Compromise
   B) Utility
   C) Evidence-Based
   D) Actionable

2. Which of the following is part of the cyber kill chain?

   A) Phishing
   B) Weaponization
   C) Malware
   D) IOC

3. TTPs describe actions taken by adversaries during a network attack.

   A) True
   B) False

4. Which is not a threat intelligence type?

   A) Operational
   B) Strategic
   C) Defense
   D) Tactical

Further reading

- *What Is Threat Intelligence? Definition and Examples*: `https://www.recordedfuture.com/threat-intelligence-definition/`
- *Threats/Vulnerabilities*: `https://www.sans.org/reading-room/whitepapers/threats/paper/38790`
- Yara GitHub repository: `https://github.com/VirusTotal/yara`
- Suricata: `https://suricata-ids.org/`
- *The Zeek Network Security Monitor*: `https://www.zeek.org/`
- Snort: `https://www.snort.org/`

14
Hunting for Threats

The release of Mandiant's APT1 report provided information security professionals with a deep insight into one of the most experienced and prolific threat groups operating. The insight into the Chinese PLA Unit 61398 also provided a context around these sophisticated threat actors. The term **Advanced Persistent Threat** (**APT**) became part of the information security lexicon. Information security and incident responders now had insight into threats that conducted their activities without detection, and over a significant period of time.

Continued research has also demonstrated that organizations still lag far behind in their ability to detect a breach that has occurred or that is currently ongoing. The 2018 *Cost of a Data Breach Study: Global Overview* authored by IBM and Ponemon Institute determined that of the 477 organizations that were surveyed, there was an average of 197 days that passed before the breach was detected. This average indicates that threat actors had over half a year to conduct their activities free from defenders' actions.

With the threat that APTs pose, coupled with the average time even moderately sophisticated groups can spend in a target network, organizations have started to move from passive detection and response to a more active approach, to identify potential threats in the network. This practice, called **threat hunting**, is a proactive process, whereby digital forensics techniques are used to conduct analysis on systems and network components to identify and isolate threats that have previously gone undetected. As with incident response, threat hunting is a combination of processes, technology, and people that does not rely on preconfigured alerting or automated tools, but rather, incorporates various elements of incident response, threat intelligence, and digital forensics.

This chapter will provide an overview of the practice of threat hunting by examining several key elements.

First is an understanding of the threat hunting maturity model, which provides a construct to the various aspects of threat hunting. Second, the threat hunt cycle explores the process that encompasses threat hunting, from start to finish. Third, as threat hunting is a proactive process, to ensure it is properly executed, effective planning is necessary. This chapter will provide an overview of how to plan for a threat hunt. Understanding these topics will provide you with the foundation to incorporate threat hunting into your own operations, and be better positioned to identify previously unidentified threats.

We will cover the following topics in this chapter:

- The threat hunting maturity model
- The threat hunt cycle
- MITRE ATT&CK
- Threat hunt planning
- Threat hunt reporting

The threat hunting maturity model

The cybersecurity expert David Bianco, the developer of the Pyramid of Pain covered in the previous chapter, developed the threat hunting maturity model while working for the cybersecurity company Sqrrl. It is important to understand this maturity model in relation to *threat hunting,* as it provides threat hunters and their organization a construct in determining the roadmap to maturing the threat hunting process in their organization. The maturity model is made up of five levels, starting at **Hunt Maturity 0** (or **HM0**) to HM4. What follows is a review of the five levels of the model:

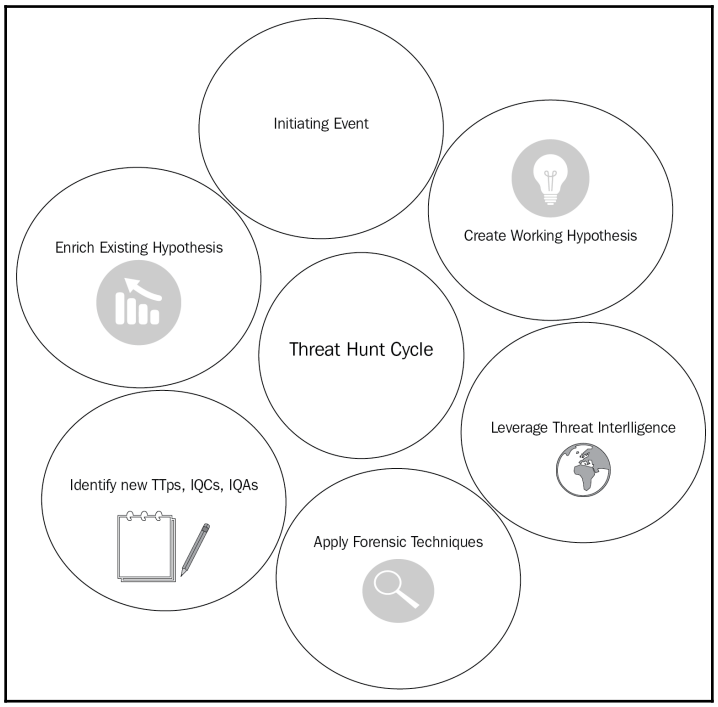
- **HM0—Initial**: During the initial stage, organizations rely exclusively on automated tools such as network- or host-based intrusion prevention/detection systems, antivirus, or **security information and event management** (**SIEM**) to provide alerts to the threat hunt team. These alerts are then manually investigated and remediated. Along with a heavy reliance on alerting, this is limited to no use of threat intelligence indicators. Finally, this maturity level is characterized by a limited ability to collect telemetry from systems. Organizations at this stage are not able to threat-hunt.

- **HM1—Minimal**: At the minimal stage, organizations are collecting more data and, in fact, may have access to a good deal of system telemetry available. In addition, these organizations manifest the intent to incorporate threat intelligence into their operations but are behind in terms of the latest data and intelligence on threat actors. Although this group will often still rely on automated alerting, the increased level of system telemetry affords this group the ability to extract threat intelligence indicators from reports and search available data for any matching indicators. This is the first level at which threat hunting can begin.
- **HM2—Procedural**: At this stage, the organization is making use of threat hunting procedures that have been developed by other organizations, which are then applied for a specific use case. For example, an organization may find a presentation or use case write-up concerning lateral movement via a Windows system's internal tools. From here, they would extract the pertinent features of this procedure and apply it to their own dataset. At this stage, the organization is not able to create its own process for threat hunting. The HM2 stage also represents the most common level of threat hunting maturity for organizations that have threat hunting programs.
- **HM3—Innovative**: At this maturity level, the threat hunters are developing their own processes. There is also increased use of various methods outside manual processes, such as machine learning, statistical, and link analysis. There is a great deal of data that is available at this level as well.
- **HM4—Leading**: Representing the *bleeding edge* of threat hunting, the Leading maturity level incorporates a good deal of the features of HM3 with one significant difference, and that is the use of automation. Processes that have produced results in the past are automated, providing an opportunity for threat hunters to craft new threat hunting systems that are better adept at keeping pace with emerging threats.

The threat hunt maturity model is a useful construct for organizations to identify their current level of maturity, as well as plan for the inclusion of future technology and processes, to keep pace with the very fluid threat landscape.

Threat hunt cycle

Threat hunting, like incident response, is a process-driven exercise. There is not a clearly defined and accepted process in place, but there is a general sequence that threat hunting takes that provides a process that can be followed. The following screenshot combines the various stages of a threat hunt into a process that guides threat hunters through the various activities to facilitate an accurate and complete hunt:

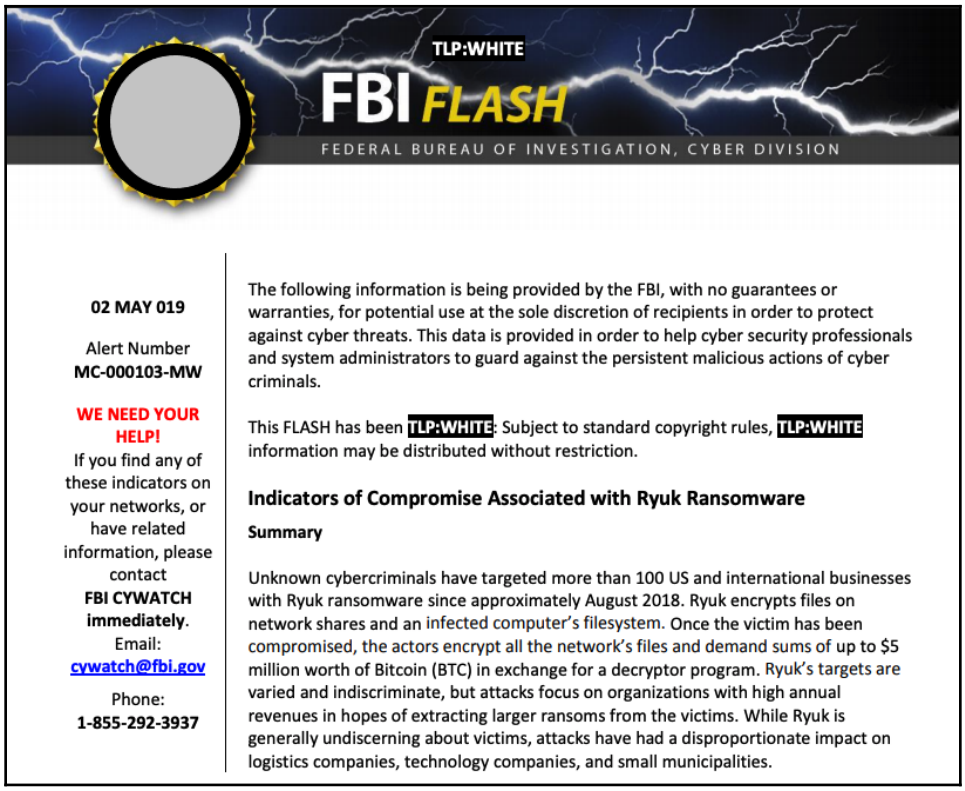


Let's begin with the first stage.

Initiating event

The threat hunt begins with an initiating event. Organizations that incorporate threat hunting into their operations may have a process or policy that threat hunting be conducted at a specific cadence or time period. For example, an organization may have a process where the security operations team conducts four or five threat hunts per month, starting on the Monday of every week. Each one of these separate hunts would be considered the *initiating event*.

A second type of initiating event is usually driven by some type of threat intelligence alert that comes from an internal or external source. For example, an organization may receive an alert such as the one shown in the following screenshot. This alert, from the United States Federal Bureau of Investigation, indicates that there are new **Indicators of Compromise** (**IoCs**) that are associated with the Ryuk family of ransomware. An organization may decide to act on this intelligence, and begin a hunt through the network for any indicators associated with the IoCs provided as part of the alert, shown here:



TLP:WHITE

FBI FLASH

FEDERAL BUREAU OF INVESTIGATION, CYBER DIVISION

02 MAY 019

Alert Number
MC-000103-MW

WE NEED YOUR HELP!
If you find any of these indicators on your networks, or have related information, please contact **FBI CYWATCH immediately**.
Email:
cywatch@fbi.gov

Phone:
1-855-292-3937

The following information is being provided by the FBI, with no guarantees or warranties, for potential use at the sole discretion of recipients in order to protect against cyber threats. This data is provided in order to help cyber security professionals and system administrators to guard against the persistent malicious actions of cyber criminals.

This FLASH has been **TLP:WHITE**: Subject to standard copyright rules, **TLP:WHITE** information may be distributed without restriction.

Indicators of Compromise Associated with Ryuk Ransomware

Summary

Unknown cybercriminals have targeted more than 100 US and international businesses with Ryuk ransomware since approximately August 2018. Ryuk encrypts files on network shares and an infected computer's filesystem. Once the victim has been compromised, the actors encrypt all the network's files and demand sums of up to $5 million worth of Bitcoin (BTC) in exchange for a decryptor program. Ryuk's targets are varied and indiscriminate, but attacks focus on organizations with high annual revenues in hopes of extracting larger ransoms from the victims. While Ryuk is generally undiscerning about victims, attacks have had a disproportionate impact on logistics companies, technology companies, and small municipalities.

After the initiating event is fully understood, the next phase is to start crafting what to look for during the threat hunt.

Creating a working hypothesis

Moving from the initiating event, the threat hunting team then creates a working hypothesis. Threat hunting is a focused endeavor, meaning that hunt teams do not just start poking through event logs or memory images, looking for whatever they can find. A working hypothesis—such as *an APT group has gained control of several systems on the network*—is general, and provides no specific threat hunting target. Threat hunters need to focus their attention on the key indicators, whether those indicators come from a persistent threat group or some existing threat intelligence.

A working hypothesis provides the focus. A better hypothesis would be *An APT-style adversary has taken control of the DMZ web* servers *and is using them as a Command and Control infrastructure*. This provides a specific target that the hunt team can then apply digital forensic techniques, to determine if this hypothesis is true.

Those threat hunts that are initiated via alerts can often find key areas of focus that can be leveraged, to craft a hypothesis. For example, the previous section contained an alert from the FBI. In addition to the IoCs associated with Ryuk, the following language was also in the alert:

The exact infection vector remains unknown, as Ryuk deletes all files related to the dropper used to deploy the malware. In some cases, Ryuk has been deployed secondary to TrickBot and/or Emotet banking Trojans, which use ***Server Message Block*** *(****SMB****) protocols to propagate through the network and can be used to steal credentials.*

From this data, the hunt team can craft a hypothesis that directly addresses these **tactics, techniques, and procedures** (**TTP**). The hypothesis may be: *An adversary has infected several internal systems and is using with a Microsoft SMB protocol to move laterally within the internal network, with the intent of infecting other systems.* Again, this sample hypothesis is specific and provides the threat hunters with a concrete area of focus to either prove or disprove the hypothesis.

Leveraging threat intelligence

In the previous chapter, there was an extensive overview of how cyber threat intelligence can be leveraged during an incident. As we are applying a variety of incident response and digital forensic techniques in threat hunting, cyber threat intelligence also plays an important role. Like the working hypothesis, threat intelligence allows the hunt team to further focus attention on specific indicators or TTPs that have been identified through a review of the pertinent threat intelligence available.

An example of this marriage between the hypothesis and threat intelligence can be provided by examining the relationship between the banking Trojan Emotet and the infrastructure that supports it.

First, the hypothesis that the hunt team has crafted is: *Systems within the internal network have been in communication with the Emotet delivery or Command and Control infrastructure*. With that hypothesis in mind, the hunt team can leverage **open source intelligence** (**OSINT**) or commercial feeds to augment their focus. For example, the following sites have been identified as delivering the Emotet binary:

- `https://www.cityvisualization.com/wp-includes/88586`
- `https://87creationsmedia.com/wp-includes/zz90f27`
- `http://karencupp.com/vur1qw/s0li7q9`
- `http://www.magnumbd.com/wp-includes/w2vn93`
- `http://minmi96.xyz/wp-includes/l5vaemt6`

From here, the hunt can focus on those systems that would have indications of traffic to those URLs.

Applying forensic techniques

The next stage in the threat hunt cycle is applying forensic techniques to test the hypothesis. The bulk of this volume has been devoted to using forensic techniques to find indicators in a variety of locations. In threat hunting, the hunt team will apply those same techniques to various evidence sources to determine if any indicators are present.

For example, in the previous section, five URLs were identified as indicators associated with the malware Emotet. Threat hunters could leverage several sources of evidence, to determine if those indicators were present. For example, an examination of proxy logs would reveal if any internal systems connected to any of those URLs. DNS logs would also be useful, as they would indicate if any system on the internal network attempted to resolve one or more of the URLs to establish connections. Finally, firewall logs may be useful in determining if any connections were made to those URLs or associated IP addresses.

Identifying new indicators

During the course of a threat hunt, new indicators may be discovered. A search of a memory image for a specific family of malware reveals a previously unknown and undetected IP address. These are the top 10 indicators that may be identified in a threat hunt:

- Unusual outbound network traffic
- Anomalies in privileged user accounts
- Geographical anomalies
- Excessive login failures
- Excessive database read volume
- HTML response sizes
- Excessive file requests
- Port-application mismatch
- Suspicious registry or system file changes
- DNS request anomalies

Enriching the existing hypothesis

New indicators that are identified during the threat hunt may force the modification of the existing threat hunt hypothesis. For example, in the course of a threat hunt for indicators of an Emotet infection, threat hunters uncover the use of the Windows system internal tool PsExec, to move laterally in the internal network. From here, the original hypothesis should be changed to reflect this new technique, and any indicators should be incorporated into the continued threat hunt.

Another option available to threat hunters regarding new indicators that are discovered is to begin a new threat hunt, utilizing the new indicators as the initiating event. This action is often leveraged when the indicator or TTP identified is well outside the original threat hunting hypothesis. This is also an option where there may be multiple teams that can be leveraged. Finally, indicators may also necessitate moving from a threat hunt into incident response. This is often a necessity in cases where data loss, credential compromise, or the infection of multiple systems have occurred. It is up to the hunt team to determine at which point the existing hypothesis is modified, or a new hypothesis created, or, in the worst-case scenario, an incident is declared.

MITRE ATT&CK

In `Chapter 13`, *Leveraging Threat Intelligence*, there was a brief exploration of the MITRE ATT&CK framework, as it pertains to the incorporation of threat intelligence into incident response. The MITRE ATT&CK framework is also extremely useful in the initial planning and execution of a threat hunt. The MITRE ATT&CK framework is useful in a variety of areas in threat hunting, but for the purposes of this chapter, the focus will be on two specific use cases. First will be the use of the framework to craft a specific hypothesis. Second, the framework can be utilized to determine likely evidence sources that would produce the best indicators.

The first use case, crafting the hypothesis, can be achieved through an examination of the various tactics and techniques of the MITRE ATT&CK framework. An examination of the various enterprise tactics located at `attack.mitre.org/tactics/enterprise`, reveals 12 separate tactics, as shown in the following screenshot:

ID	Name	Description
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.
TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

Although descriptive, the tactics are not specific enough to be useful in threat hunt hypothesis creation. What threat hunters should be focusing attention on are the various techniques that make up a tactic—for example, examining the initial access tactic, which describes the various techniques that adversaries utilize to gain an initial foothold. The MITRE ATT&CK framework describes in detail 11 such tactics.

Where the MITRE ATT&CK framework can be leveraged for a hypothesis is through the incorporation of one or more of these techniques across various tactics. For example, if a threat hunt team is concerned about Command and Control traffic, they can look under TA0011 in the MITRE ATT&CK enterprise tactics. From here, there are 22 specific techniques that fall under that tactic. From here, the threat hunt team can select a technique, such as T1132—Data Encoding. They can then craft a hypothesis that states: *An adversary has compromised a system on the internal network and is using encoding or compression to obfuscate Command and Control traffic.*

In this instance, the MITRE ATT&CK framework provided a solid foundation for crafting a hypothesis. What the MITRE ATT&CK framework also provides is an insight into the various threat actor groups and tools that have been identified as using this type of technique. For example, examining the technique T1132—Data Encoding, located at `https://attack.mitre.org/techniques/T1132/`, revealed that threat actor groups such as APT19 and APT33 both use this technique to obfuscate their Command and Control traffic. In terms of tools, MITRE indicates that a variety of malware families such as Linux Rabbit or njRAT use obfuscation techniques, such as Base64 encoding or encoded URL parameters. This can further focus a threat hunt on specific threat groups or malware families if the hunt team wishes.

The second way the MITRE ATT&CK framework can be leveraged for threat hunting is by providing guidance on evidence sources. Going back to the T1132 Data Encoding technique, MITRE indicates that the best data sources for indicators associated with this technique are packet captures, network protocol analysis, process monitoring, and identifying processes that are using network connections. From here, the threat hunter could leverage packet capture analysis with Moloch or Wireshark, to identify any malicious indicators. These can be further augmented with an examination of key systems' memory for network connections and their associated processes.

MITRE will often break down additional details that will assist threat hunt teams in their search for indicators. Technique 1132 contains additional details concerning this specific technique, as shown here:

> *"Analyze network data for uncommon data flows (e.g. a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used."*

The details regarding the technique, the data sources, and the potential course of action are a great aid to threat hunters, as it affords them the ability to put a laser focus to the threat hunt, the hypothesis, and—finally—a course of action. These elements go a long way to crafting a plan for the threat hunt.

Threat hunt planning

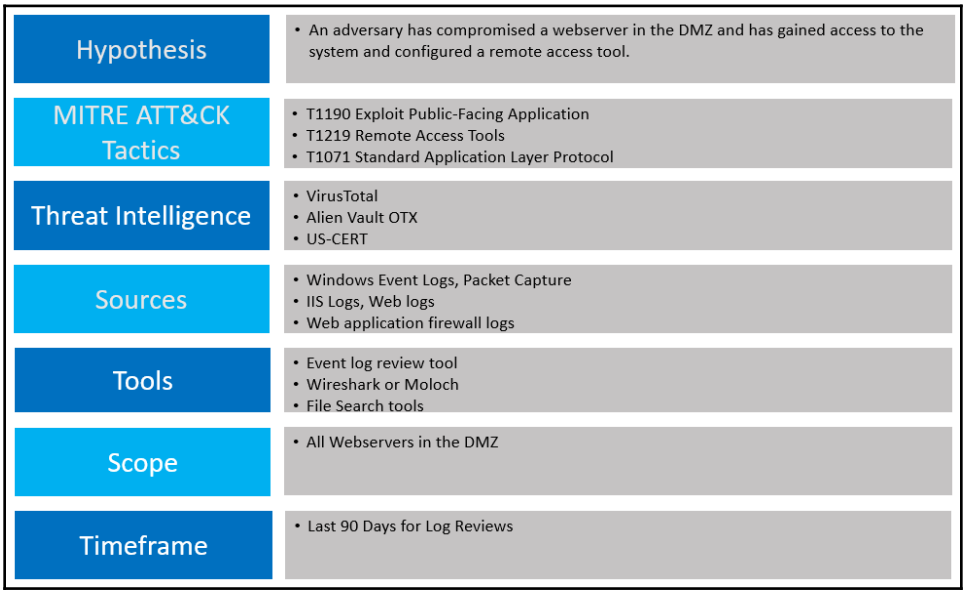
Beginning a threat hunt does not require a good deal of planning, but there should be some structure as to how the threat hunt will be conducted, the sources of data, and the time period on which the threat hunt will focus. A brief written plan will address all of the key points necessary, and place all of the hunt team on the same focus area so that extraneous data that does not pertain to the threat hunt is minimized. The following are seven key elements that should be addressed in any plan:

- **Hypothesis**: A one- or two-sentence hypothesis that was discussed earlier. This hypothesis should be clearly understood by all the hunt team members.
- **MITRE ATT&CK tactic(s)**: In the previous chapter, there was a discussion about the MITRE ATT&CK framework and its application to threat intelligence and incident response. In this case, the threat hunt should include specific tactics that have been in use by threat actors. Select the tactics that are most applicable to the hypothesis.
- **Threat intelligence**: The hunt team should leverage as much internally developed and externally sourced threat intelligence as possible. External sources can either be commercial providers or OSINT. The threat intelligence should be IoCs, indicators of attack (**IoA**s), and TTPs that are directly related to the hypothesis and the MITRE ATT&CK tactics that were previously identified. These are the data points that the hunt team will leverage during the hunt.
- **Evidence sources**: This should be a list of the various evidence sources that should be leveraged during the threat hunt. For example, if the hunt team is looking for indicators of lateral movement via SMB, they may want to leverage NetFlow or selected packet captures. Other indicators of lateral movement using Remote Desktop can be found within the Windows event logs.
- **Tools**: This section of the plan outlines the specific tools that are necessary to review evidence. For example, `Chapter 10`, *Analyzing Log Files* addressed log file analysis with the open source tool Skadi. If the threat hunt will make use of this tool, it should be included in the plan.

A group of tools that greatly aid in threat hunting is **Endpoint Detection and Response** (**EDR**) tools. These tools build on the existing methodology of antivirus platforms. Many of these platforms also have the ability to search across the enterprise for specific IoCs and other data points, allowing threat hunt teams to search an extensive number of systems for any matching IoCs. These tools should be leveraged extensively during a threat hunt.

- **Scope**: This refers to the systems that will be included in the threat hunt. The plan should indicate either a single system or systems, subnet, or network segment on which to focus. In the beginning, threat hunters should focus on a limited number of systems, and add more as they become more familiar with the toolset and how much evidence can be examined in the time given.
- **Timeframe**: As threat hunting often involves a retrospective examination of evidence, it is necessary to set a timeframe upon which the threat hunt team should focus. For example, if an originating event is relatively new (say, 48 hours), the timeframe indicated in the plan may be limited to the past 72 hours, to address any previously undetected adversarial action. Other timeframes may widen the threat hunt to 14—or even 30—days, based on the hypothesis and threat intelligence available.

Here is an example threat hunt plan that incorporates these elements into an easy-to-view framework:



In this sample plan, the hypothesis is that an adversary has taken control of one or more of the DMZ web servers. The associated MITRE ATT&CK tactics involve either exploiting the web application or establishing a Command and Control channel. In this plan, the threat hunt team will utilize OSINT. The sources and tools involve logs and packet captures and will be reviewed for the last 90 days. This is a simple plan, but it provides each member of the threat hunt team with all of the directions necessary to conduct the hunt.

Threat hunt reporting

`Chapter 11`, *Writing the Incident Report*, provided the details necessary for incident responders to properly report on their activities and their findings. Reporting a threat hunt is just as critical, as it affords managers and policymakers insight into the tools, techniques, and processes utilized by the hunt team, as well as providing potential justification of additional tools or modifying the existing processes. The following are some of the key elements of a threat hunt report:

- **Executive summary**: This high-level overview of the actions taken, indicators discovered, and if the hunt proved or disproved the hypothesis provides the decision-makers a short narrative that can be acted upon.
- **Threat hunt plan**: The plan, including the threat hunt hypothesis, should be included as part of the threat hunt report. This provides the reader with the various details that the threat hunt team utilized during their work.
- **Forensic report**: As `Chapter 11`, *Writing the Incident Report* explored, there is a great deal of data that is generated by forensic tools as well as by the incident responders themselves. This section of the threat hunt report is the lengthiest, as the detailed examination of each system or evidence source should be documented. Further, there should be a comprehensive list of all evidence items that were examined as part of the hunt.
- **Findings**: This section will indicate if the hunt team was able to either prove or disprove the hypothesis that had been set at the beginning of the hunt. In the event that the hypothesis was proved, there should be documentation as to what the follow-on actions were, such as a modification to the hypothesis, a new hypothesis, or if the incident response capability was engaged. Finally, any IoCs, IoAs, or TTPs that were found as part of the threat hunt should also be documented.

Another key area of the Findings section should be an indication of how the existing process and technology were able to facilitate a detailed threat hunt. For example, if the threat hunt indicated that Windows event logs were insufficient in terms of time or quantity, this should be indicated in the report. This type of insight provides the ability to justify additional time and resources spent on creating an environment where sufficient network and system visibility is obtained to facilitate a detailed threat hunt.

One final section of the threat hunt report is a section devoted to non-security or incident-related findings. Threat hunts may often find vulnerable systems, existing configuration errors, or non-incident-related data points. These should be reported as part of the threat hunt so that they can be remediated.

- **Recommendations**: As there will often be findings, even on threat hunts that disprove the hypothesis and include no security findings, recommendations to improve future threat hunts, the security posture of the organization, or improvements to system configuration should be included. It would also be advisable to break these recommendations out into groups. For example, strategic recommendations may include long-term configuration or security posture improvements that may take an increased amount of resources and time to implement. Tactical recommendations may include short-term or simple improvements to the threat hunt process or systems settings that would improve the fidelity of alerting. To further classify recommendations, there may be a criticality placed on the recommendations, with those recommendations needed to improve the security posture or to prevent a high-risk attack given higher priority than those recommendations that are simply focused on process improvement or configuration changes.

The threat hunt report contains a good deal of data that can be used to continually improve the overall threat hunting process. Another aspect to consider is which metrics can be reported to senior managers about threat hunts. Some key data points they may be interested in are the hours utilized, previously unknown indicators identified, infected systems identified, threats identified, and the number of systems contained. Having data that provides indicators of the threat hunt's ability to identify previously unidentified threats will go a long way to ensuring that this is a continuing practice that becomes a part of the routine security operations of an organization.

Summary

Eric O'Neill, former FBI intelligence professional and cybersecurity expert, has said: *When you don't hunt the threat, the threat hunts you.* This is exactly the sentiment behind threat hunting. As was explored, the average time from compromise to detection leaves adversaries with plenty of time to do significant damage. This can be done by understanding the level of maturity in an organization in terms of proactive threat hunting, applying the threat hunt cycle, adequately planning, and—finally— recording the findings. Taking a proactive stance may reduce the time an adversary has to cause damage, and help to possibly keep ahead of the constantly shifting threat landscape.

Questions

1. At what level of the threat hunting maturity model would technologies such as machine learning be found?

   A) HM0
   B) HM1
   C) HM2
   D) HM3

2. Which of the following is a top 10 IoC?

   A) IP address
   B) Malware signature
   C) Excessive file request
   D) URL

3. A threat hunt initiating event can be a threat intelligence report.

   A) True
   B) False

4. A working hypothesis is a generalized statement regarding the intent of the threat hunt.

   A) True
   B) False

Further reading

- Your Practical Guide to Threat Hunting: `https://www.threathunting.net/files/hunt-evil-practical-guide-threat-hunting.pdf`
- MITRE ATT&CK: `https://attack.mitre.org/`

Appendix

There is a significant number of Windows Event Log types available to IT and security professionals. This Appendix includes the most critical events that pertain to security and incident investigations and have been provided as a reference.

Event ID	Event type	Primary use	Event log
21	Remote desktop services: session logon succeeded.	Event correlation, lateral movement, scoping	`TerminalServices-LocalSessionManager/Operational`
25	Remote desktop services: session reconnection succeeded.	Event correlation, lateral movement, scoping	`TerminalServices-LocalSessionManager/Operational`
102	This event is logged when the terminal services gateway service requires a valid **Secure Sockets Layer** (**SSL**) certificate to accept connections.	Event correlation, lateral movement, scoping	`Microsoft-Windows-TerminalServices-Gateway`
106	A user registered a scheduled task.	Execution, persistence	Windows task scheduler
107	Task scheduler launched a task due to a time trigger.	Execution, persistence	Windows task scheduler
131	The RDP server accepted a new TCP connection.	Event correlation, lateral movement, scoping	Remote desktop services RdpCoreTs
140	A user updated a scheduled task.	Execution, persistence	Windows task scheduler

141	A user deleted a scheduled task.	Execution, persistence	Windows task scheduler
200	Task scheduler launched the action in the instance of the task.	Execution, persistence	Windows task scheduler
201	Task scheduler successfully completed a task.	Execution, persistence	Windows task scheduler
800	Pipeline execution details.	Event correlation, lateral movement, execution	PowerShell
4103	Executing pipeline.	Event correlation, lateral movement, execution	PowerShell
1024	RDP ClientActiveX is trying to connect to a server.	Event correlation, lateral movement, scoping	`Microsoft-Windows-TerminalServices-RDPClient/Operational`
4624	An account was successfully logged on.	Event correlation (event to user), scoping, user location identification	Security
4625	An account failed to log on.	Event correlation (event to user), scoping, user location identification	Security
4634	An account was logged off.	Event correlation (event to user), scoping, user location identification	Security

4647	User initiated log off.	Event correlation (event to user), scoping, user location identification	Security
4648	A login was attempted using explicit credentials.	Event correlation, lateral movement, scoping	Security
4672	Special privileges assigned to new login.	Escalation of privilege	Security
4698	A scheduled task was created.	Persistence	Security
4727	A security-enabled global group was created.	Escalation of privilege, lateral movement, persistence	Security
4728	A member was added to a security-enabled global group.	Escalation of privilege, lateral movement	Security
4737	A security-enabled global group was changed.	Escalation of privilege, lateral movement, persistence	Security
4706	A new domain trust was created.	Validation of controls	Security
4720	A user account was created.	Escalation of privilege, lateral movement, persistence	Security
4729	A member was removed from a security-enabled global group.	Validation of controls	Security

4754	A security-enabled universal group was created.	Escalation of privilege, lateral movement, persistence	Security
4755	A security-enabled universal group was changed.	Escalation of privilege, lateral movement, persistence	Security
4776	A user account was unlocked.	Escalation of privilege, persistence	Security
5140	A network share object was accessed.	Lateral movement	Security
5145	A network share object was checked to see whether client can be granted desired access.	Lateral movement	Security
7045	A new service was installed by a user.	Execution, lateral movement	Security

Assessment

Chapter 1: Understanding Incident Response

1. A
2. C
3. B
4. A

Chapter 2: Managing Cyber Incidents

1. A
2. D
3. B

Chapter 3: Fundamentals of Digital Forensics

1. B
2. A
3. D
4. B

Chapter 4: Collecting Network Evidence

1. D
2. A
3. D
4. B

Chapter 5: Acquiring Host-Based Evidence

1. C
2. A
3. C
4. A

Chapter 6: Forensic Imaging

1. A, C
2. A
3. A
4. D

Chapter 7: Analyzing Network Evidence

1. A
2. B
3. D
4. B

Chapter 8: Analyzing System Memory

1. D
2. C
3. B
4. A

Chapter 9: Analyzing System Storage

1. D
2. C
3. A
4. C

Chapter 10: Analyzing Log Files

1. A
2. B
3. C
4. A

Chapter 11: Writing the Incident Report

1. A
2. D
3. B
4. C

Chapter 12: Malware Analysis for Incident Response

1. D
2. A
3. B
4. B

Chapter 13: Leveraging Threat Intelligence

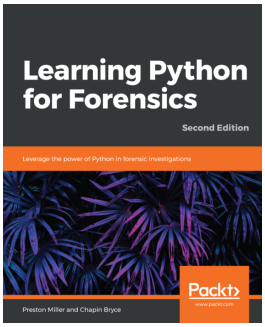
1. A
2. B
3. A
4. C

Chapter 14: Hunting for Threats

1. D
2. C
3. A
4. B

Other Books You May Enjoy

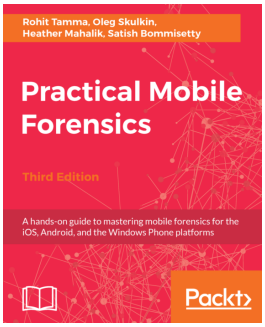
If you enjoyed this book, you may be interested in these other books by Packt:



Learning Python for Forensics - Second Edition
Preston Miller, Chapin Bryce

ISBN: 978-1-78934-169-0

- Learn how to develop Python scripts to solve complex forensic problems
- Build scripts using an iterative design
- Design code to accommodate present and future hurdles
- Leverage built-in and community-sourced libraries
- Understand the best practices in forensic programming
- Learn how to transform raw data into customized reports and visualizations
- Create forensic frameworks to automate analysis of multiple forensic artifacts
- Conduct effective and efficient investigations through programmatic processing



Practical Mobile Forensics - Third Edition
Rohit Tamma, Oleg Skulkin, Et al

ISBN: 978-1-78883-919-8

- Discover the new techniques in practical mobile forensics
- Understand the architecture and security mechanisms present in iOS and Android platforms
- Identify sensitive files on the iOS and Android platforms
- Set up a forensic environment
- Extract data from the iOS and Android platforms
- Recover data on the iOS and Android platforms
- Understand the forensics of Windows devices
- Explore various third-party application techniques and data recovery techniques

Leave a review - let other readers know what you think

Please share your thoughts on this book with others by leaving a review on the site that you bought it from. If you purchased the book from Amazon, please leave us an honest review on this book's Amazon page. This is vital so that other potential readers can see and use your unbiased opinion to make purchasing decisions, we can understand what our customers think about our products, and our authors can see your feedback on the title that they have worked with Packt to create. It will only take a few minutes of your time, but is valuable to other potential customers, our authors, and Packt. Thank you!

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

V

W