**Incident handler's journal**

This journal guides incident handlers through the complexities of managing cyber threats, offering insights into strategies, tools, and case studies. It's an essential read for enhancing skills and staying updated in the cybersecurity landscape.

| **Date:** July 23, 2024 | **Entry:** #1 | | | |
| --- | --- | --- | --- | --- |
| Description | Documenting a cybersecurity incident | | | |
| Tool(s) used | None. | | | |
| The 5 W's | * **Who**: An organized group of unethical hackers * **What**: A ransomware security incident * **Where**: At a health care company * **When**: Tuesday 9:00 a.m. * **Why**: The incident happened because unethical hackers were able to access the company's systems using a phishing attack. After gaining access, the attackers launched their ransomware on the company's systems, encrypting critical files. The attackers' motivation appears to be financial because the ransom note they left demanded a large sum of money in exchange for the decryption key. | | | |
| Additional notes | 1. How could the health care company prevent an incident like this from occurring again? 2. Should the company pay the ransom to retrieve the decryption key? | | | |

| **Date:** 30/10/2023 | **Entry:** #2 | | |
| --- | --- | --- | --- |
| Description | Documenting a cybersecurity incident | | |
| Tool(s) used | For this activity, I used VirusTotal, which is an investigative tool that analyzes files and URLs for malicious content such as viruses, worms, trojans, and more. It's a very helpful tool to use if you want to quickly check if an indicator of compromise like a website or file has been reported as malicious by others in the cybersecurity community. For this activity, I used VirusTotal to analyze a file hash, which was reported as malicious.  This incident occurred in the **Detection and Analysis** phase. The scenario put me in the place of a security analyst at a SOC investigating a suspicious file hash. After the suspicious file was detected by the security systems in place, I had to perform deeper analysis and investigation to determine if the alert signified a real threat. | | |
| The 5 W's | * **Who:** An employee downloaded a suspicious file on his computer**.** * **What**: A malicious payload was executed on the employee’s computer. * **When** Monday 10:00 am * **Where**: At a financial services company. * **Why**: The employee received an email containing an attachment. The attachment was a password-protected spreadsheet file. The spreadsheet's password was provided in the email. The employee downloaded the file, then entered the password to open the file. | | |
| Additional notes | * SHA256 file hash: 54e6ea47eb04634d3e87fd7787e2136ccfbcc80ade34f246a12cf93bab527f6b * The file hash has been reported as malicious by over 56 vendors. Upon further investigation, this file hash is known as the malware Flagpro, which has been commonly used by the advanced threat actor BlackTech. | | |

| **Date:** 11/11/2023 | **Entry:** #3 | | |
| --- | --- | --- | --- |
| Description | capture and filter network traffic in a Linux environment. | | |
| Tool(s) used | tcpdump | | |
| Task | This activity consists of performing tasks associated with using tcpdump to capture network traffic. The data is captured in a packet capture (p-cap) file and then the content of the captured packet data is examined focusing on specific types of traffic. | | |
| Additional notes | 1. Identify network interfaces      1. Identify the interface options available for package capture.(if systems do not include ifconfig)      1. Inspect network traffic (eth0):   -i eth0: Capture data specifically from the eth0 interface.  -v: Display detailed packet data.  -c5: Capture 5 packets of data.     1. Capture network traffic   -i eth0: Capture data from the eth0 interface.  -nn: Do not attempt to resolve IP addresses or ports to names.This is best practice from a security perspective, as the lookup data may not be valid. It also prevents malicious actors from being alerted to an investigation.  -c9: Capture 9 packets of data and then exit.  port 80: Filter only port 80 traffic. This is the default HTTP port.  -w capture.pcap: Save the captured data to the named file.  &: This is an instruction to the Bash shell to run the command in the background.     1. Verify that data has been capture      1. Filter the captured packet data   -nn: Disable port and protocol name lookup.  -r: Read capture data from the named file.  -v: Display detailed packet dat | | |

| **Date:** 11/11/2023 | **Entry:** #3 | | |
| --- | --- | --- | --- |
| Description | capture and filter network traffic in a Linux environment. | | |
| Tool(s) used | tcpdump | | |
| Task | This activity consists of performing tasks associated with using tcpdump to capture network traffic. The data is captured in a packet capture (p-cap) file and then the content of the captured packet data is examined focusing on specific types of traffic. | | |
| Additional notes | 1. Identify network interfaces      1. Identify the interface options available for package capture.(if systems do not include ifconfig)      1. Inspect network traffic (eth0):   -i eth0: Capture data specifically from the eth0 interface.  -v: Display detailed packet data.  -c5: Capture 5 packets of data.     1. Capture network traffic   -i eth0: Capture data from the eth0 interface.  -nn: Do not attempt to resolve IP addresses or ports to names.This is best practice from a security perspective, as the lookup data may not be valid. It also prevents malicious actors from being alerted to an investigation.  -c9: Capture 9 packets of data and then exit.  port 80: Filter only port 80 traffic. This is the default HTTP port.  -w capture.pcap: Save the captured data to the named file.  &: This is an instruction to the Bash shell to run the command in the background.     1. Verify that data has been capture      1. Filter the captured packet data   -nn: Disable port and protocol name lookup.  -r: Read capture data from the named file.  -v: Display detailed packet dat | | |