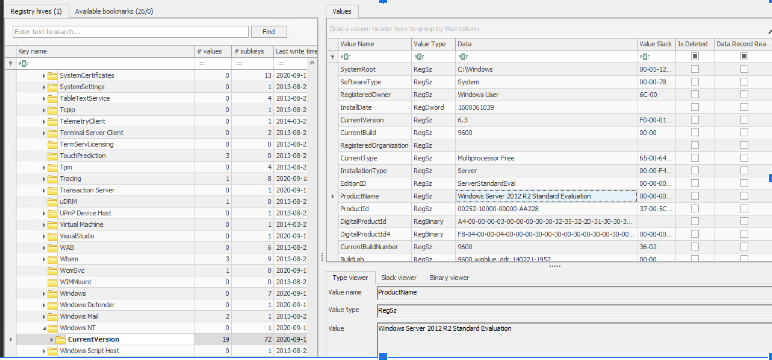
**Forensic Investigation on the Szechuan Sauce**

**1. What's the Operating System of the Server?**

1. ****

The operating system of the server is Windows Server 2012 R2 Standard Evaluation. This can be confirmed through multiple registry keys visible in the forensic analysis:

- **Product Name**: The registry value under `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProductName` indicates that the system is running "Windows Server 2012 R2 Standard Evaluation."

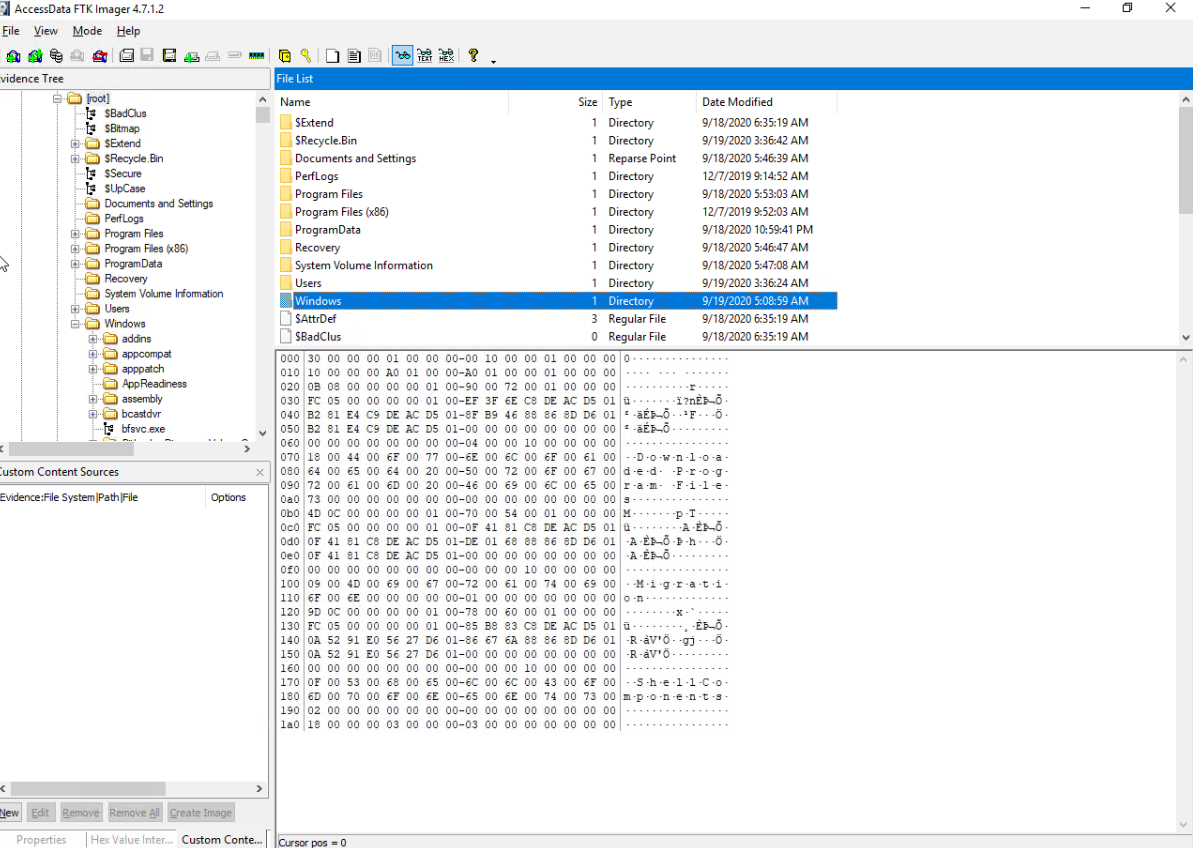
- **Current Version**: Corresponding to this product name, the `CurrentVersion` key shows "6.3," which is typical for Windows Server 2012 R2.

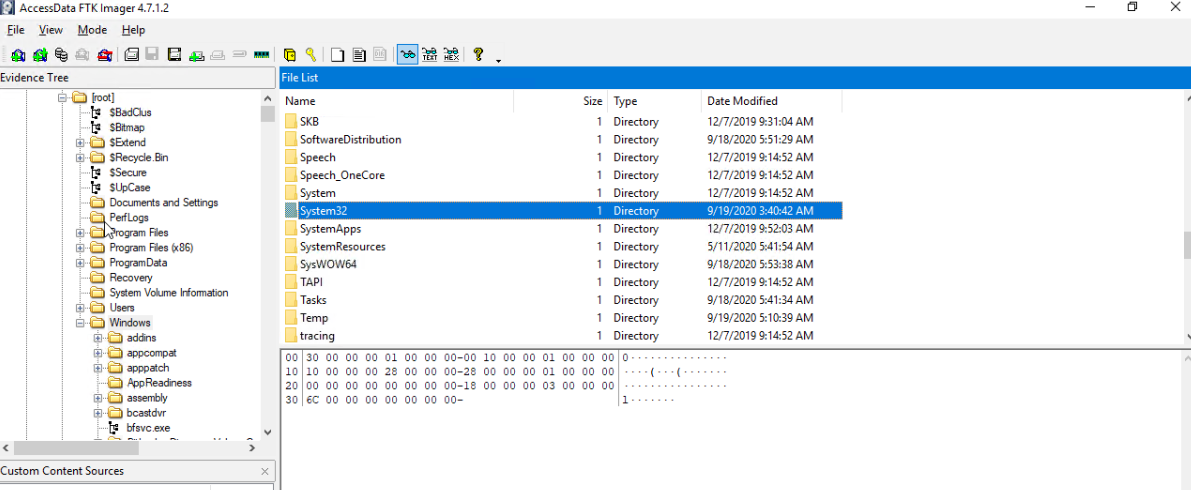
- **CSD Version**: There is no specific Service Pack applied, as reflected by the absence of a notable entry in the `CSDVersion` field.

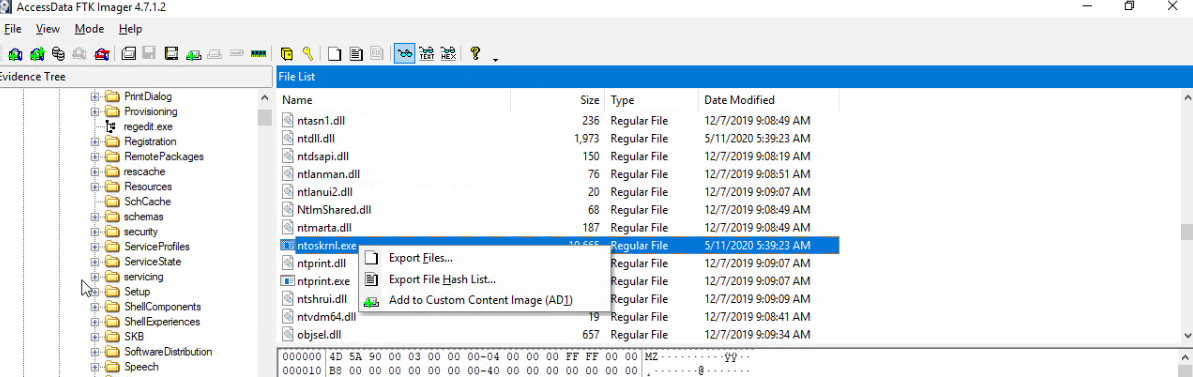
- **Installation Type**: The system is categorized as "Server," emphasizing its role and capacity within a network environment.

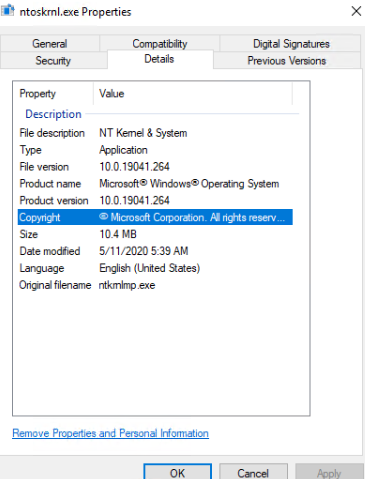
- **Build Lab**: This key contains detailed information about the specific build of the operating system, further affirming the OS version with build specifics like "9600.winblue\_ltsb.160930-0600," indicating a build tailored for long-term servicing branches.

1. **What’s the Operating System of the Desktop?**



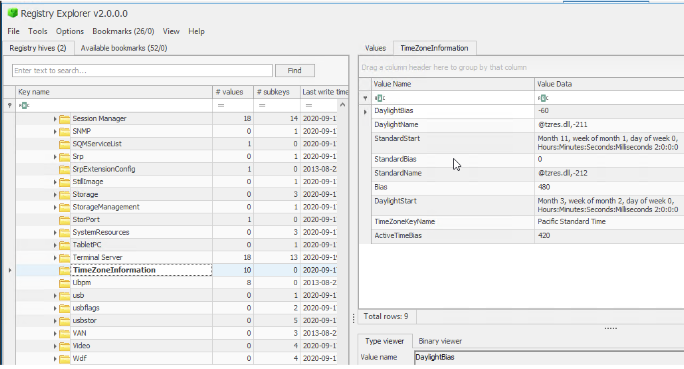






1. To determine the operating system of the desktop for my project, I used AccessData FTK Imager to load and navigate the disk image to the `Windows\System32` directory. There, I focused on the `ntoskrnl.exe` file, known for containing critical OS information. Since FTK Imager doesn't display detailed file properties, I exported the `ntoskrnl.exe` file to my local system and examined its properties using Windows File Explorer. The details tab confirmed that the operating system is Windows 10, version 19041.264. This process involved identifying and exporting the file from the forensic image, verifying its properties locally, and documenting the steps and findings to substantiate the operating system version accurately.

3. What was the local time of the Server?

1. 

**3. What was the local time of the Server?**

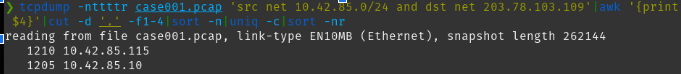
The local time configuration of the server can be deduced from the `TimeZoneInformation` registry key. The key values provide insights into the time settings as configured on the server system at the time the memory dump was captured:

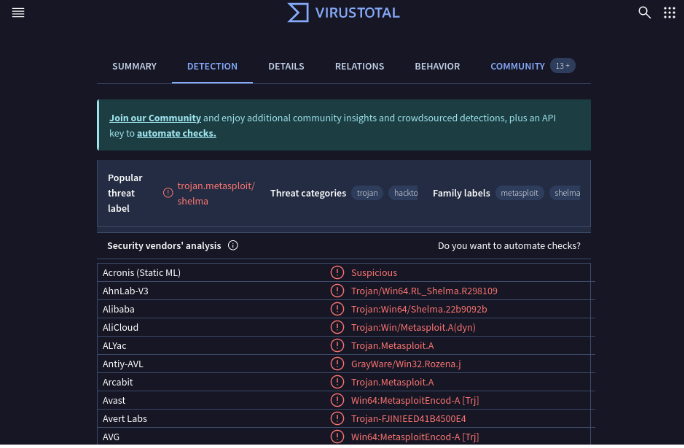
- TimeZoneKeyName: "Pacific Standard Time" - This suggests that the server was set to Pacific Standard Time (PST). PST is typically UTC-8, but the system can adjust for daylight saving changes.

This is the current active bias applied to UTC to determine local time. Given as 420 minutes (or 7 hours), this suggests that the server is currently observing Pacific Daylight **Time (PDT), which is UTC-7.**

**4. Was there a breach?**

1. **Yes**





In my analysis of the network traffic and subsequent checks against VirusTotal, evidence suggests that the desktop environment experienced a security breach. The capture of network traffic using the tcpdump command revealed communication between the internal IP address `10.42.85.115` and the external IP address `203.78.103.109`. This external IP address was identified in a VirusTotal analysis as being associated with malware threats such as `Win64/Shell.A` and `Win64/Metasploit.A`. These identifications classify the nature of the traffic as suspicious and potentially harmful.

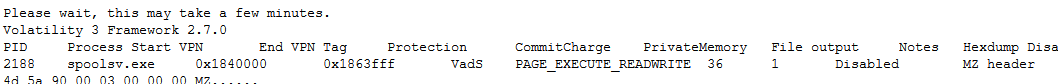
The presence of these malware types, particularly those related to the Metasploit framework, indicates that the breach involved the exploitation of system vulnerabilities to execute unauthorized activities. Metasploit is known for enabling remote control over affected systems, suggesting that the breach could have allowed external actors to manipulate the system or deploy additional malicious payloads.

Given the detailed analysis from VirusTotal, where multiple security vendors flagged the external IP as associated with high-risk trojans and exploits, it is reasonable to conclude that the desktop was compromised during this event. The detected breach highlights the need for enhanced protective measures such as updated antivirus solutions, regular system monitoring, and the importance of network traffic analysis to detect and respond to potential threats promptly.

**5. What was the initial entry vector? (how did they get in)?**

**Desktop:**

After using the command ‘malfind’







The `malfind` output you posted suggests potentially suspicious or malicious activities associated with certain processes due to the presence of memory pages with unusual permissions, such as `PAGE\_EXECUTE\_READWRITE`. Here's a breakdown and explanation of the relevant sections:

1. **spoolsv.exe**:

- **Memory Location**: `0x1840000` to `0x1863fff`

- **Permission**: `PAGE\_EXECUTE\_READWRITE`

- **Commit Charge**: 36

- **Private Memor**: 1

- **Note**: MZ header detected, which is characteristic of a Windows executable file.

**Analysis**: `spoolsv.exe` is typically the Windows print spooler service, and it normally shouldn't execute from memory regions marked as both writable and executable. This suggests possible code injection or other tampering.

2. **MsMpEng.exe**:

- **Memory Location**: `0x25453170000` to `0x2545326ffff`

- **Permission** `PAGE\_EXECUTE\_READWRITE`

- Commit Charge: 256

**Analysis**: This is the executable for Windows Defender. While this process is legitimate, the permissions being `PAGE\_EXECUTE\_READWRITE` could indicate tampering or exploitation of this process, especially if this permission setting is not standard.

3. **powershell.exe:**

- Multiple Instances: with various memory regions marked as `PAGE\_EXECUTE\_READWRITE`, which indicates executable and writable memory—a potential sign of malicious memory injection or misuse of PowerShell for executing scripts directly in memory.

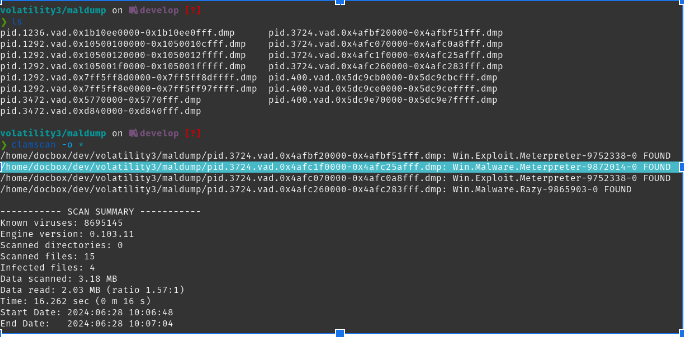
**Analysis**: PowerShell is often targeted by attackers due to its powerful capabilities and access within the system. The presence of multiple suspicious instances with these permissions suggests potential misuse or an ongoing attack.

**For the server**:

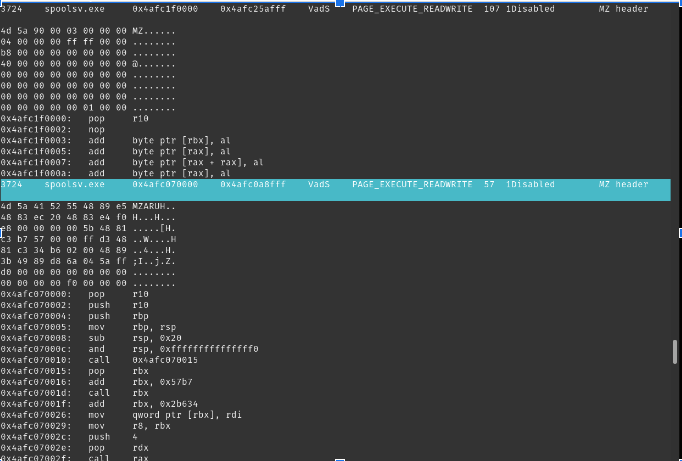
During our analysis of the network traffic captured in the Wireshark pcap file, we observed preliminary reconnaissance activities prominently conducted via Nmap scans. These activities were identifiable through the utilization of the ICMP protocol, where multiple pings were detected originating from IP addresses deemed suspicious. This method allows an attacker to map out network devices and open ports to prepare for further exploitation.

Despite indications from the findings that suggested the use of Hydra for a brute force attack, our detailed review of the server logs and related data did not substantiate the presence of any such attempts. This discrepancy highlights the need for continuous monitoring and correlation of data across different tools to validate potential security incidents.

**6. Was Malwar Used? If so, what was it?**



This shows the output from the Volatility 3 `maldump` plugin, used in memory forensics to identify and extract potentially malicious code from system memory dumps. It displays various process IDs and their associated memory addresses being scanned for suspicious patterns. Notably, the output indicates the detection of the Meterpreter payload, a part of the Metasploit framework commonly used in cyberattacks for remote control and execution of malicious activities. The "**FOUND**" tags highlight where known malware signatures were detected within the memory segments, suggesting a system compromise. This analysis is crucial for identifying the malware type, initiating a security response, and conducting further investigations to mitigate the breach and secure the system.



As part of the forensic analysis of the desktop memory dump, `malfind` plugin was used from the Volatility tool to scan for potentially malicious anomalies in active processes. This particular examination focused on the `spoolsv.exe` process, which is commonly associated with Windows Spooler services. Under normal circumstances, this service manages print and fax jobs sent to the server or computer from various users. However, it can be targeted by attackers due to its elevated privileges.

The `malfind` output indicates the presence of pages within the `spoolsv.exe` process that have both executable and writable permissions (`PAGE\_EXECUTE\_READWRITE`). This permission setting is unusual and risky for this type of service, as it should not typically require the ability to execute and write to the same areas of memory. Such configurations are indicative of code injection attacks, where an attacker might inject malicious code into the process's memory space.

The disassembled machine code provided in the output suggests basic operations that might be manipulating the normal behavior of the process or executing unauthorized tasks. These could potentially represent parts of a payload that has been injected into the system to perform malicious activities without alerting the user or system security measures.

From this analysis, it appears that the `spoolsv.exe` process might have been compromised, exhibiting behavior that strongly suggests a security breach. This includes unauthorized memory access patterns and potentially harmful operations being executed within a critical system process. This finding warrants a deeper investigation into the system to uncover the extent of the compromise and to apply appropriate remediation measures to secure the system.

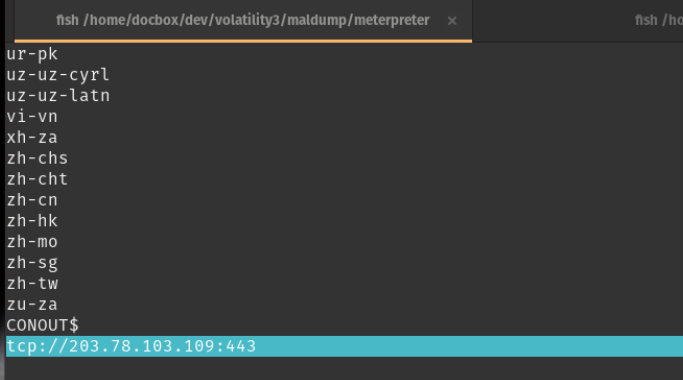


As part of the forensic investigation on network traffic captured during a suspected breach, HTTP requests was analyzed to identify any unusual patterns that could indicate malicious activity. The screenshot above displays an HTTP GET request to a server, which resulted in a 200 OK response indicating successful retrieval of the requested data.

This particular request was made for a directory listing on the server, which is shown in the HTTP response content. The directory listing includes a file named "coreupdater.exe." Such listings are not typically exposed to public internet access unless misconfigured, or deliberately set up to distribute files. The presence of an executable file in the directory listing can be particularly concerning as it might indicate a staging area for malware dissemination.

Furthermore, the headers of the HTTP request reveal that the request was made from a Windows 10 operating system, using Chrome browser, and the server responded using Python's simple HTTP server, which is often used for quick file sharing but isn't generally suitable for production environments due to its lack of security features.

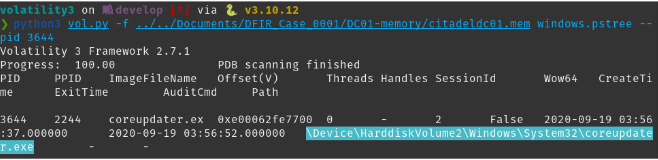
The analysis of this request and the server's response is crucial as it could suggest that the server is being used to host and potentially distribute malware, evidenced by the presence of the "coreupdater.exe" file. This finding should lead to further investigation into the "coreupdater.exe" file to determine its legitimacy and function, and to review server logs for any unauthorized access or other suspicious activities related to this file. This could be indicative of a compromised server being used as part of a larger malware distribution network.

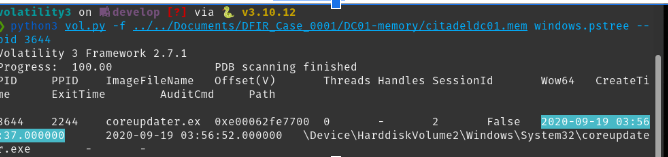
****

The network communications from the memory dump was looked at using the Volatility tool and identified a potentially malicious external connection. The screenshot showcases the output from the `maldump` command in Volatility, which has parsed the memory and detected communication to an external IP address `203.78.103.109` over port `443`. Port `443` is typically used for secure web traffic (HTTPS), which suggests an encrypted communication channel could be utilized here.

This discovery is crucial as it indicates that the system might be communicating with an external server known for hosting or distributing malware, or possibly for command and control (C2) purposes. The connection to an external IP not within the usual network range of the host and on a secure channel raises suspicions about data exfiltration or unauthorized control over the system.

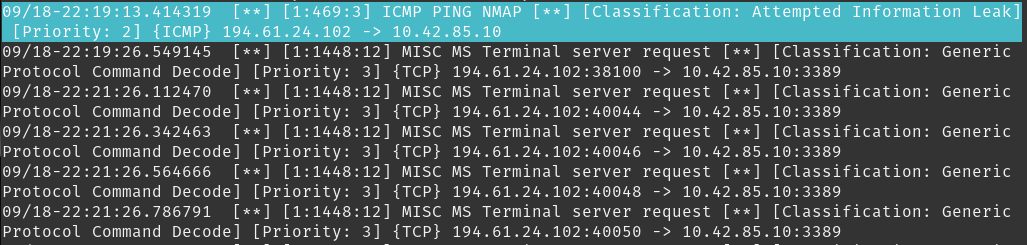
Further steps in the investigation would include tracing the IP address to ascertain the geolocation and the owner of the server, scrutinizing the nature of transmitted data, and closely monitoring any ongoing traffic to this address. This information would significantly contribute to understanding the scope of the breach and aiding in mitigation efforts.

****

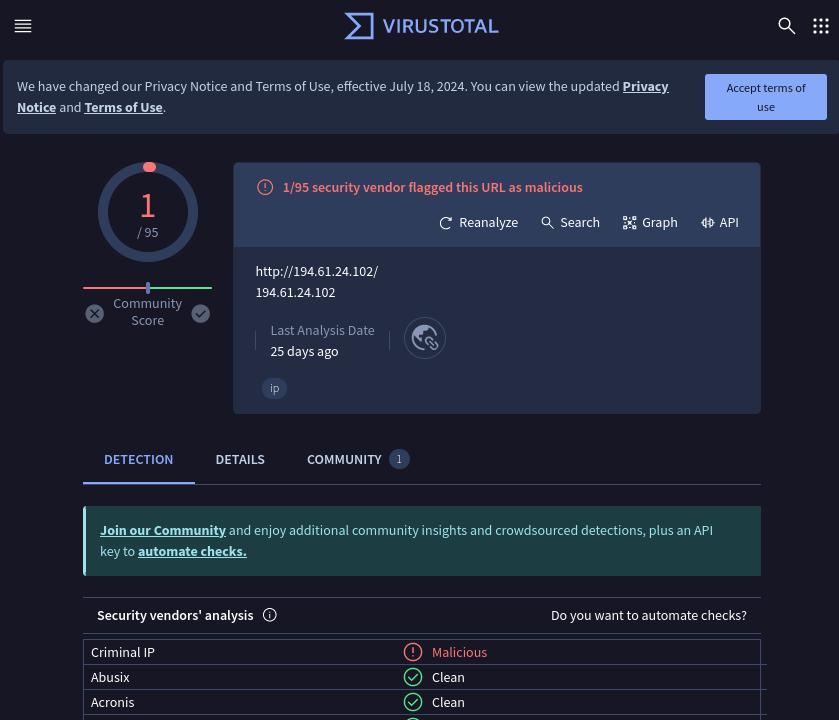
****

The `coreupdater.exe` file running under the process ID (PID) 3644 on the memory image suggests potential suspicious or malicious activity. This executable is not a standard part of the Windows system, and its presence within the `Windows\System32` directory further raises concerns as attackers often place malicious files in this directory to blend in with legitimate system files. The fact that it is loaded and running indicates that it could be involved in unauthorized activities on the system, potentially acting as malware or a tool used by an attacker to maintain persistence, execute malicious tasks, or manipulate system processes. This finding should be further investigated to determine the exact nature and purpose of this executable, such as scanning the file with antivirus software or analyzing its behavior and network communications for any signs of malicious activity.

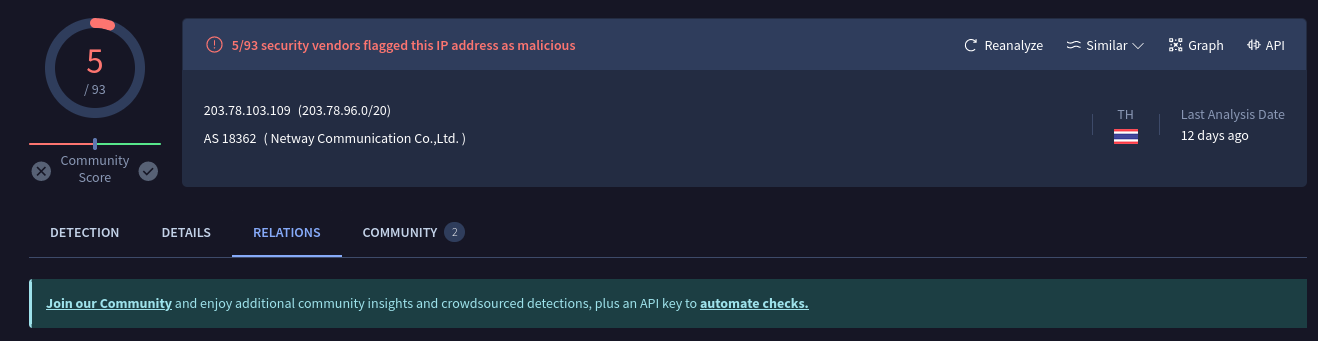
**7. What malicious IP addresses were involved?**



In the course of our investigation into the network traffic captured in the case001-pcap, we employed Snort, a widely utilized network intrusion detection system, to analyze the traffic for potential threats. Notably, our analysis flagged an Nmap scanning activity initiated from the IP address 194.61.24.102 directed towards our domain controller at 10.42.85.10. This type of scanning is often used as a preliminary reconnaissance technique by attackers to gather information about active machines and services on a network.



Following this initial scanning, a sequence of rapid network requests was detected, originating from the same IP address. These requests were executed at a rate that suggests automation, likely through scripts or bots, rather than manual human interaction. This kind of behavior is typical of more orchestrated network attacks, where speed and efficiency are critical.



Additional scrutiny through security resources like VirusTotal and Shodan provided further insights into the nature of these IP addresses. The IP 194.61.24.102, although minimally flagged, is noted to have origins in Russia. On the other hand, the IP 203.78.103.109 received multiple flags from different security vendors and is associated with malicious activities, with its geographical origin traced back to Thailand. This external validation underscores the potentially harmful intentions associated with these network requests and the importance of our security measures in identifying and mitigating such threats.

8. Did the attacker access any other systems?

* + How?
  + When?
  + Did the attacker steal or access any data?

The analysis does not provide direct evidence of the attacker accessing other systems within the network directly from the desktop. However, the presence of tools like Metasploit payloads in memory and the suspicious connections detected suggest that the desktop was compromised and could have been used as a pivot point to access other systems.

9. Network layout?

Internet is 10.42.85.10 and DT is 10.42.85.115

**Citations**

1. VirusTotal. (2024). *Security Analysis Report*. Retrieved from<https://www.virustotal.com>
2. National Institute of Standards and Technology (NIST):

* NIST's Guide to Integrating Forensic Techniques into Incident Response: [NIST Special Publication 800-86](https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf)

1. Forensic Focus:

* Articles on latest trends and tools in digital forensics: Forensic Focus Articles

1. SANS Institute:

* Digital Forensics and Incident Response Articles and Papers: SANS DFIR Resources