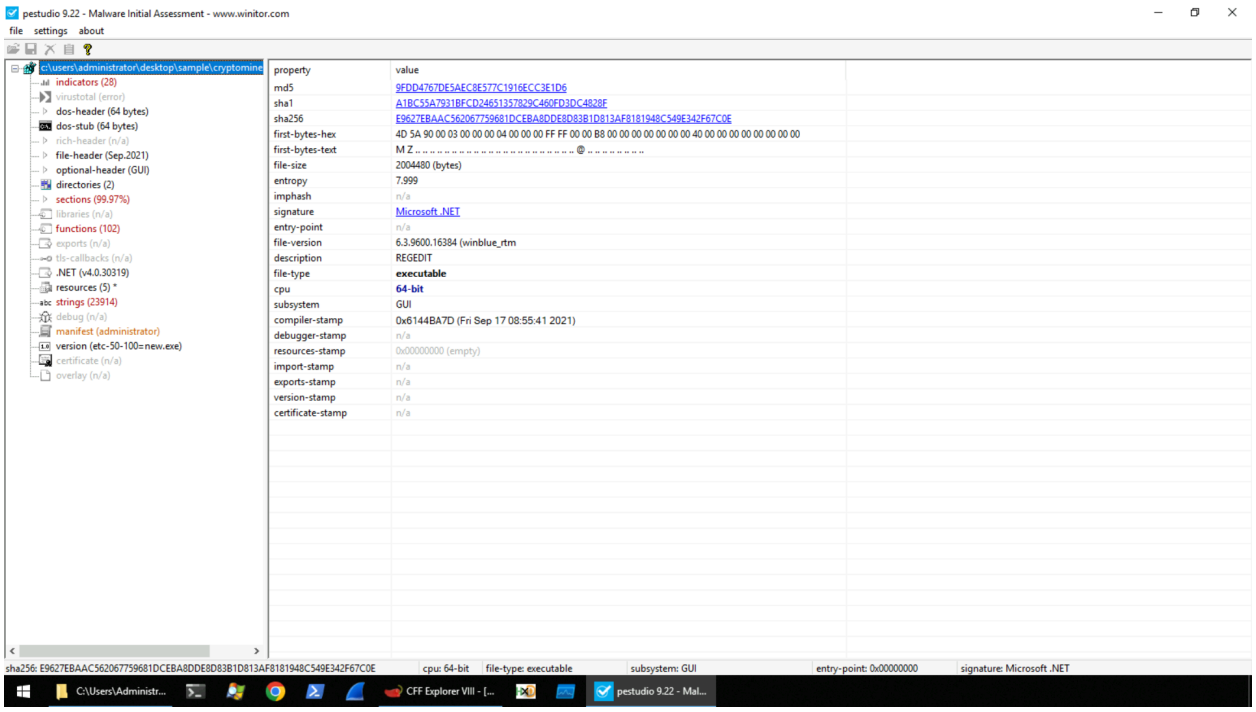
FlareVM Arsenal Tools (Malware Analysis)

Analyse using PEStudio

PEStudio is particularly useful for malware analysis, reverse engineering, and security research, as it helps analysts understand the structure and behavior of executable files without executing them.



The Main Tab provides an overview of the file being analyzed. This tab is the starting point for inspecting the properties and characteristics of a Windows executable. It is designed to give us a quick snapshot of critical information about the file, so we can start the analysis.

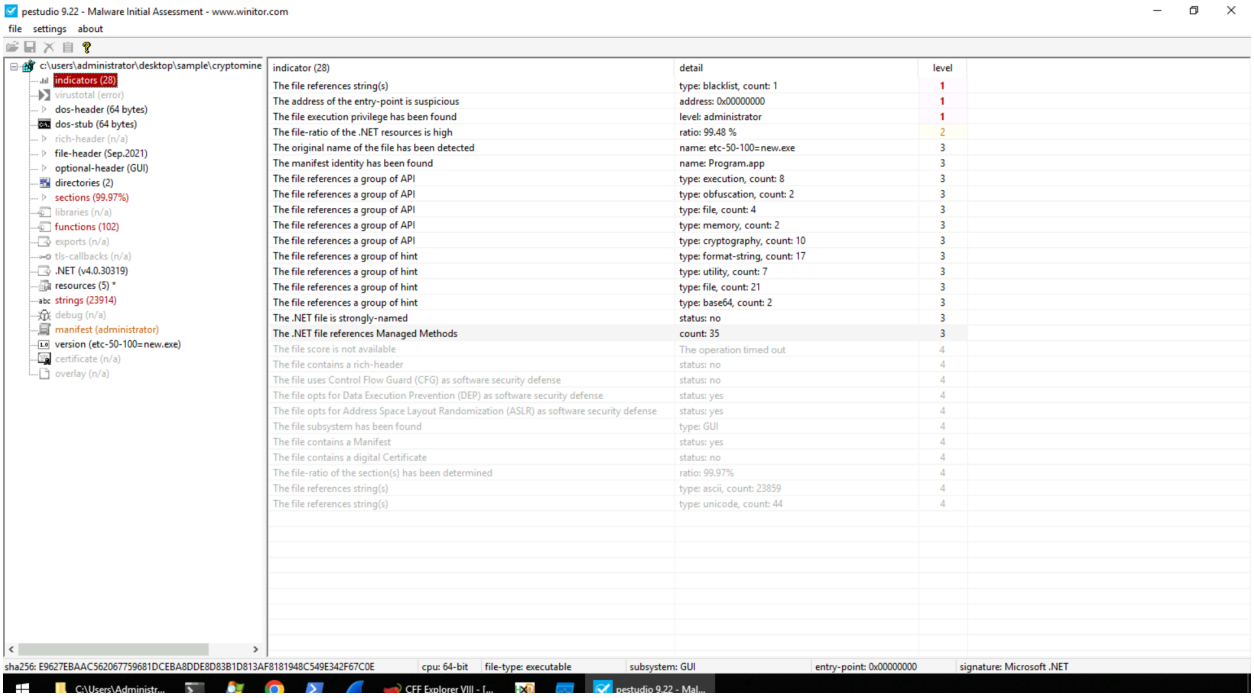
PEStudio also displays the initial bytes of the file in hexadecimal format. For instance, if the first bytes are '4D 5A', this confirms that the file is a Windows Executable. While users typically recognize files by their extensions, like '.exe', the operating system identifies file types based on the byte patterns in their headers. A Windows executable will always start with '4D 5A' in hex, which corresponds to 'MZ' in ASCII.

Additionally, PEStudio lists the file's entropy, which is useful for determining whether the malware is packed. Packed malware is often obfuscated to make it harder for analysts to quickly understand its functionality. Entropy is measured on a scale from 0 to 8, with higher values indicating a greater likelihood that the malware is packed. Values between 7 and 8 strongly suggest that the sample is packed, indicating that unpacking the malware will be necessary to extract useful indicators of compromise (IoCs).

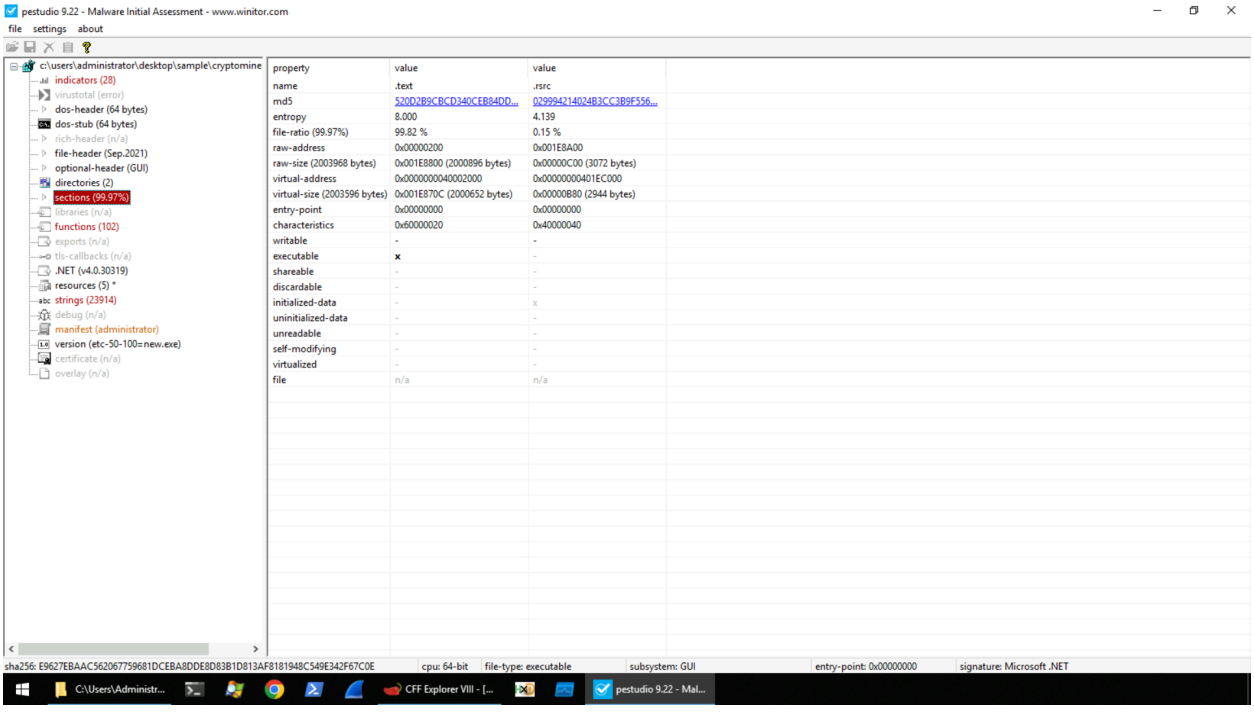
Muhammad Hisyam
Github - Cybersecurity Notes

file-size	2004480 (bytes)
entropy	7.999

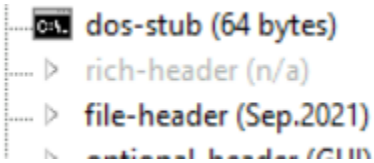
The indicators tab, this highlights data within the sample that may be malicious.

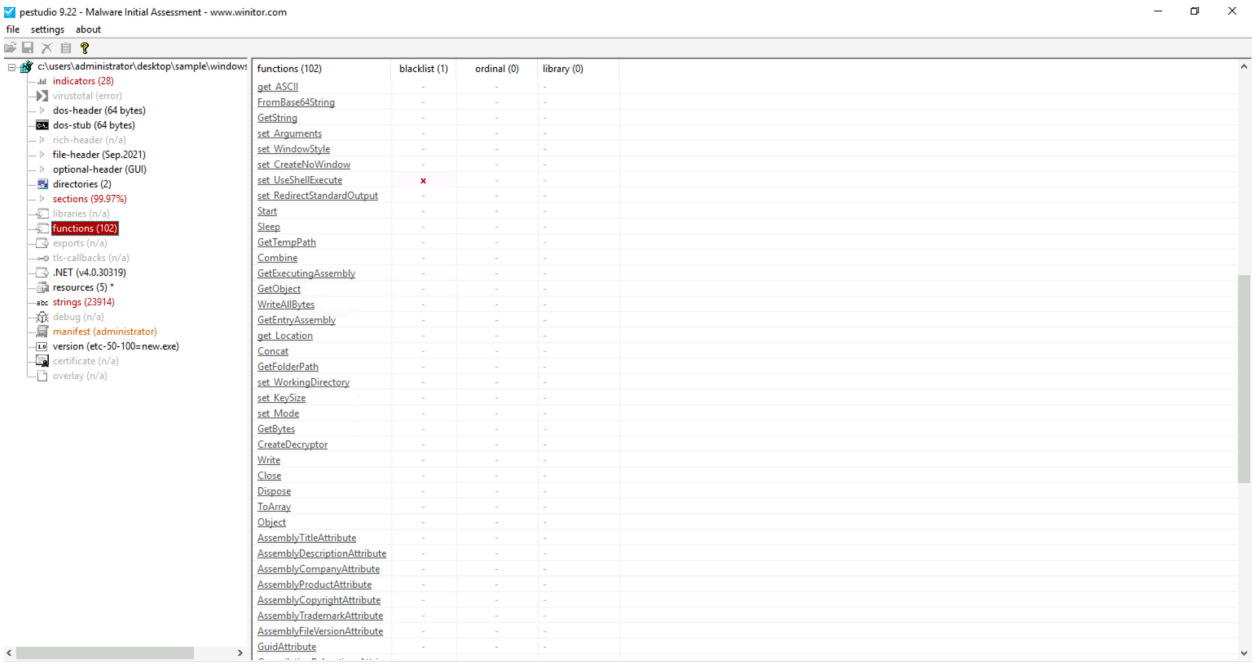


The image above demonstrates how PeStudio has detected several indicators and categorized them on a scale from 1 to 3, with 1 representing a highly likely malicious indicator. In this case, PeStudio has flagged certain suspicious strings, sections, and imports, as well as identified another file within the sample. This information is valuable because PeStudio provides tabs on the left for strings, sections, and imports, enabling further exploration of these potentially harmful indicators.

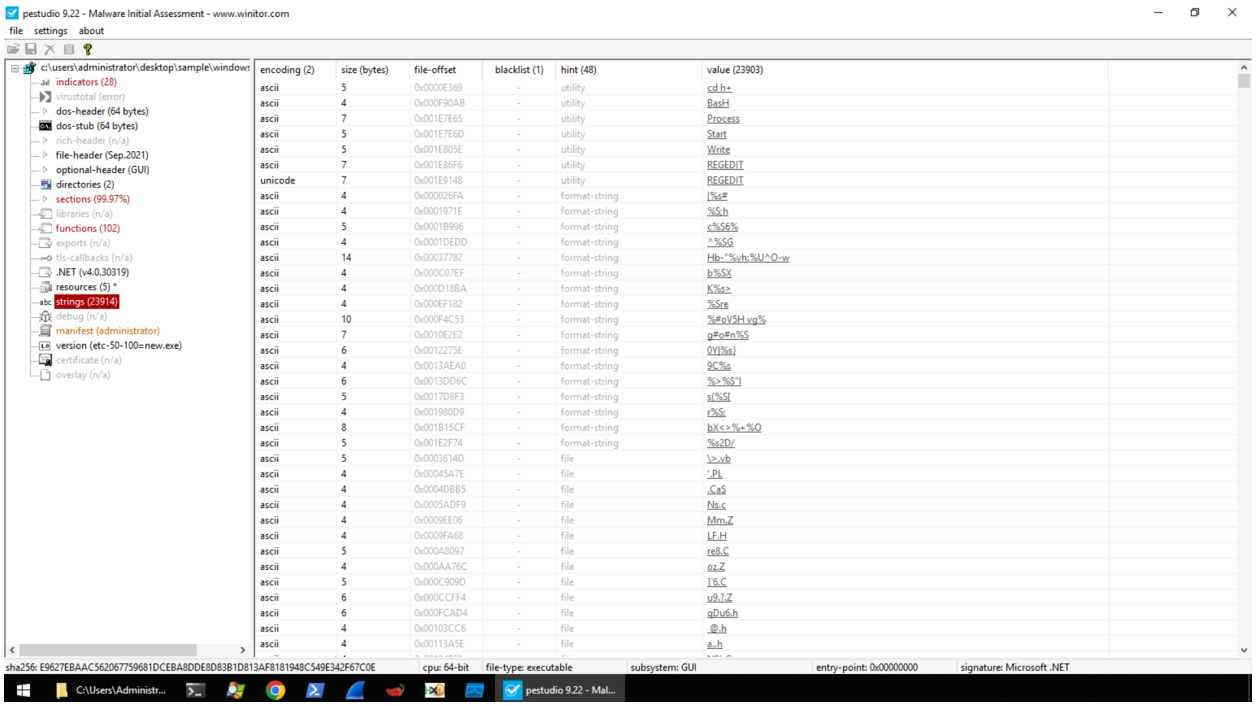


The '.text' section contains executable code, looking at the columns of the section names we can see an 'x' next to the permission pane indicating that this section has executable permissions. The '.rdata' and '.data' sections store data, and PeStudio has identified that the data section is writable. The '.idata' section stores the Import Address Table, the IAT is covered later in the article. The '.rsrc' section stores resources that can be used by the malware such as strings and additional files.





In malware analysis, set_UseShellExecute is often used to determine whether the malicious executable should be launched using the Windows shell (true), allowing it to inherit shell environment settings, or directly (false), which can be used to control and redirect output, potentially hiding its behavior.



In PEStudio, the "String" tab is where you can view all the strings embedded in a Portable Executable (PE) file. These strings may include: Text that could be visible in the program's user interface or log files. Error messages, file paths, or URLs that might indicate what the malware is doing or where it's communicating. API calls or commands used by the program. In malware analysis, the strings can provide critical clues about the file's behavior, such as suspicious domain names, file paths, or references to external resources. For instance, you might find hard coded IP addresses, commands for downloading additional payloads, or references to known malware toolkits. For example, if you're analyzing a malicious executable and the String tab shows: http://malicious-site.com C:\Windows\System32\cmd.exe dropper.exe These strings could indicate that the malware is trying to connect to a remote server, use command prompt, or drop additional malicious files. The Strings tab is an essential feature for identifying suspicious or malicious behavior based on text patterns in the binary.

Analyse Using FLOSS

FLOSS (Forensic Logical Output String Search) is a tool often used in malware analysis to extract readable strings (e.g., URLs, file paths, registry keys) from a binary executable or other types of files. It can be particularly helpful in static analysis of a binary file (such as a PE file), as it allows you to identify hidden or obfuscated text data that may indicate the behavior of the program. When you run FLOSS on a sample, it will scan the file for strings that are potentially significant. In the context of malware analysis, this can help you identify important information such as: Hardcoded URLs: Sites the malware may try to contact or download from. Command and control (C&C) servers: IP addresses or domain names the malware may communicate with. File paths: Locations where the malware may drop files or persist. Registry keys: Information about how the malware may persist on the system or alter system settings.

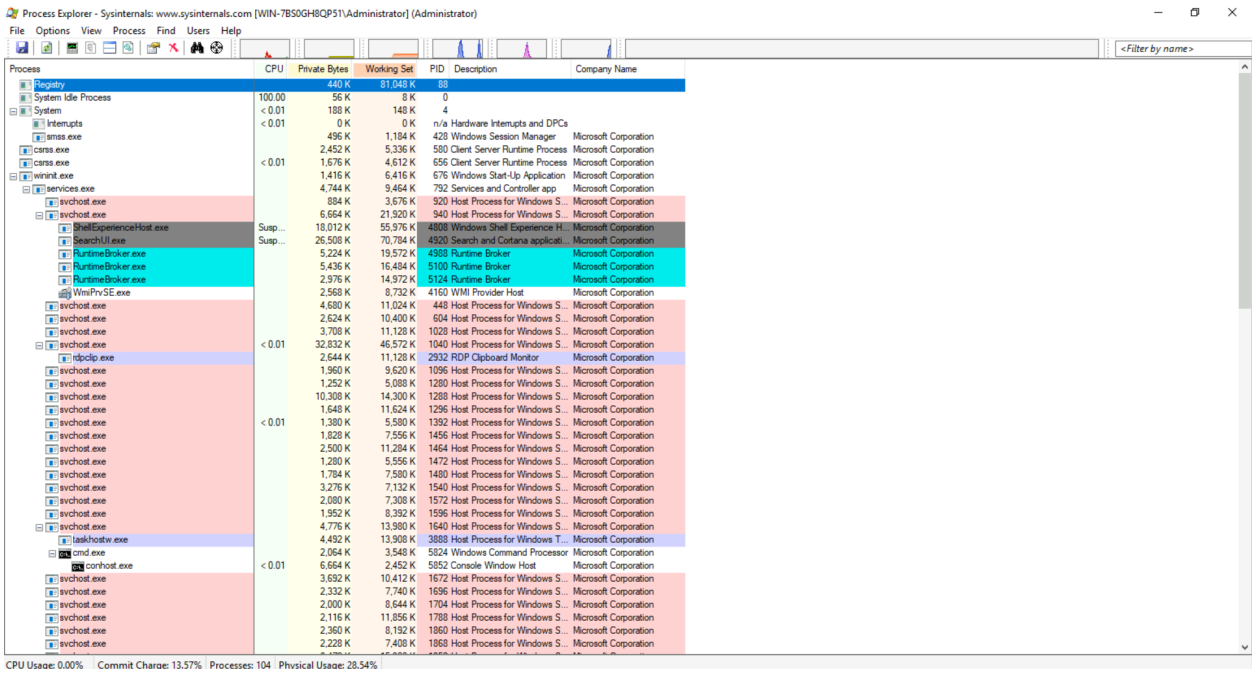
Using FLOSS for Malware Analysis: Run FLOSS on the sample: bash Copy floss sample.exe Look for suspicious strings in the output, such as: IP addresses (e.g., 192.168.1.1) Domains (e.g., malicious-site.com) File paths (e.g., C:\Windows\Temp\) Keywords related to malware (e.g., dropper, payload) Investigate: Once you have these strings, you can research whether they correspond to known malware infrastructure or patterns, giving insight into the malware's behavior.

```
PS C:\Users\Administrator\Desktop\Sample > Floss.exe windows.exe > floss.txt
WARNING: floss: .NET language-specific string extraction is not supported yet
WARNING: floss: FLOSS does NOT attempt to deobfuscate any strings from .NET binaries
INFO: floss: disabled string deobfuscation
INFO: floss: extracting static strings
INFO: floss: finished execution after 0.50 seconds
INFO: floss: rendering results
```

For example we use windows.exe as malware files. We use > floss.txt to convert the result into a text file.

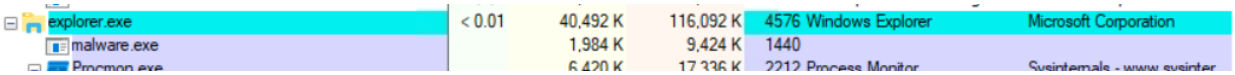
We can open the file and the result will be similar to the previous analysis using PEStudio.

Analyse using process explorer



Process Explorer is a powerful tool from Sysinternals (now owned by Microsoft) that provides detailed information about the processes running on a system, including those initiated by

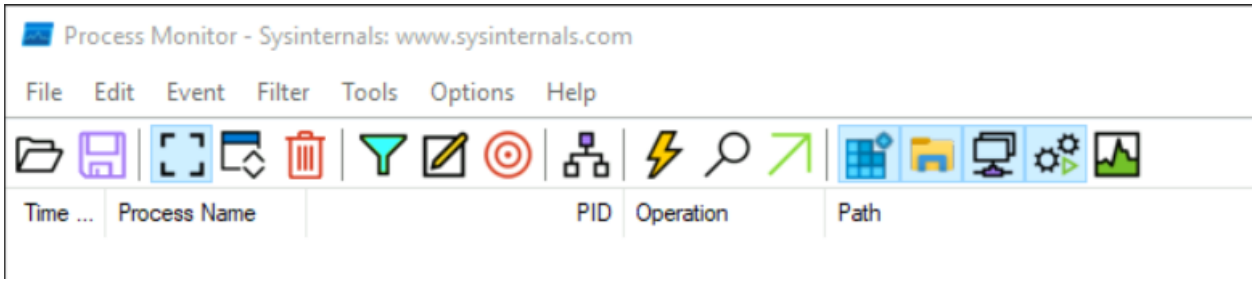
potentially malicious software. It's widely used in malware analysis because it can help analysts identify suspicious activities, understand the behavior of malware, and track down malicious processes or services running on a system.



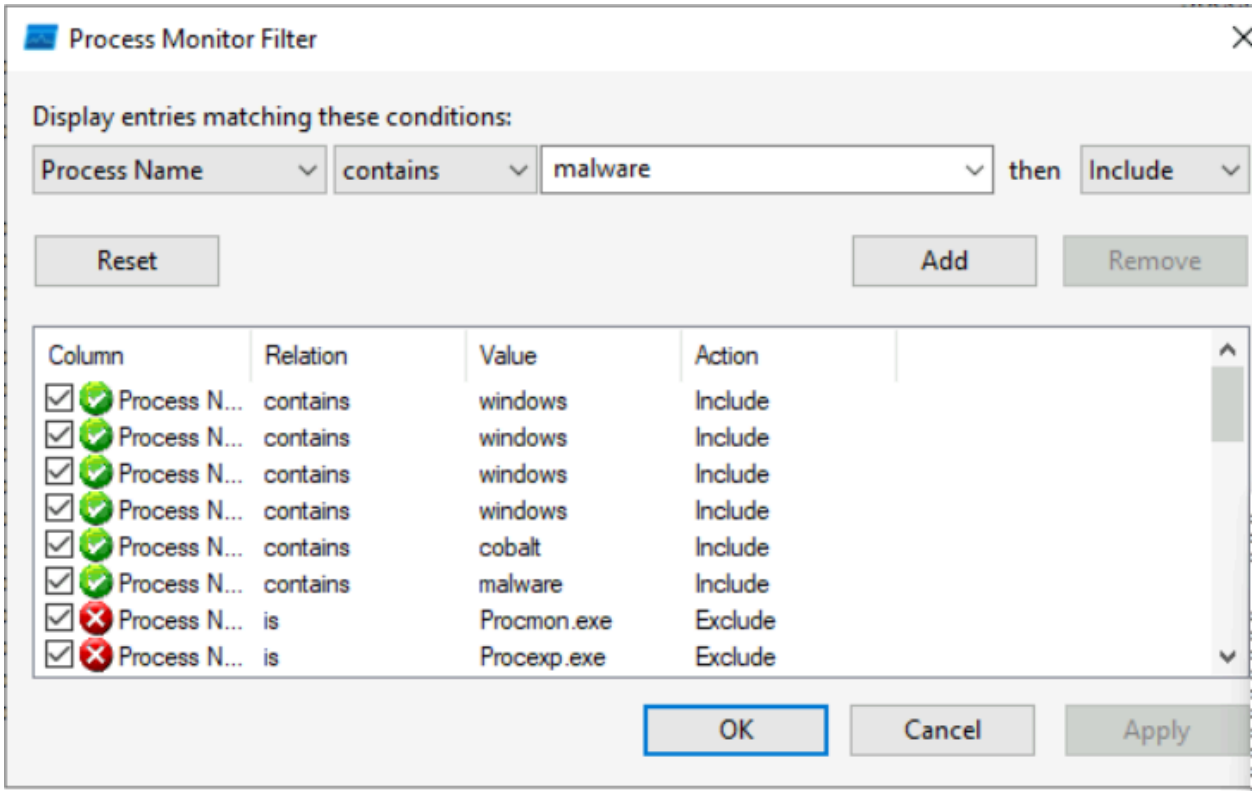
1. View Running Processes and Their Details: Process Explorer shows a detailed list of all processes, including those that might not show up in the regular Task Manager. This includes information such as:
   - Process name
   - PID (Process ID)
   - CPU, memory, and disk usage
   - Parent-child relationships between processes
   - Path to executable file
2. Malware often hides or disguises its presence by injecting into legitimate processes, so Process Explorer can help you spot unusual or unexpected processes.
3. Check the Digital Signature of Processes: If a process is signed with a valid digital certificate, Process Explorer will display the signing information. In malware analysis, if you see a process without a signature or with a suspicious or forged signature, that's often a red flag.
4. Analyze Process Tree: Process Explorer allows you to view the hierarchy of processes (parent-child relationships). This is useful when malware spawns multiple processes. For instance, a malware dropper might spawn a malicious process as a child of a legitimate one. Examining the process tree can help track the origin of the malware.
5. Check for Suspicious Process Names: Malware often uses deceptive names to blend in with legitimate processes. By carefully reviewing the process list, you can spot processes with unusual or random names, or processes running from strange locations (like temporary directories or the user's AppData folder).
6. Inspect Loaded DLLs: Malware often injects malicious code into other processes by loading DLLs (Dynamic Link Libraries). You can right-click on any process in Process Explorer and select Properties to see a list of DLLs loaded by that process. This is essential in identifying injected malicious code or trojanized DLLs.
7. Check Network Connections: You can view network activity by selecting a process and examining its TCP/IP or UDP connections. If a process is communicating with suspicious external IP addresses or domains (often associated with command-and-control servers), this could indicate malicious behavior.
   - Look for unusual ports or unfamiliar external IP addresses.
   - If malware is attempting to exfiltrate data or communicate with a C2 server, Process Explorer can help identify those connections.
8. Process Threads and Handles: Malware may spawn additional threads or create handles to resources (files, registry keys, etc.). Analyzing these threads and handles can provide insight into what the malware is doing (e.g., writing to disk, modifying the registry, etc.).

9. Search for Process Injection: Many types of malware use process injection techniques to hide in plain sight. You can look for suspicious behavior, such as when one process is loading code into another. Process Explorer allows you to inspect processes for signs of injected code or hidden threads.

Analyse using process monitor



To use the process monitor we can choose existing tools. We will use filter tools to select the applications that will be used in the analysis stage.



After we specified the filter, the result malware will be shown below.