Challenge: IcedID 2 Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Medium

Tools Used: Volatility 3, Notepad++, MemProcFS, VirusTotal

Summary: This challenge involved investigating a memory dump from a Windows machine infected with IcedID. The primary tools used were Volatility 3, MemProcFS, text editor, and VirusTotal. I found this room relatively enjoyable, although due to the size of the memory dump, it did take a while to process which was a tad annoying.

Scenario: You are a forensic analyst investigating a critical ransomware attack at a major financial institution. Your job is to analyze the memory image from the affected endpoint. Trace the attack from its origin, identify lateral movements, uncover persistence methods, and analyze any control commands.

You are a forensic analyst responding to a ransomware incident at a prominent financial institution. A workstation was compromised, and an in-memory artifact was captured for analysis. Your mission is to dissect this memory image to trace the ransomware's point of entry, determine how it executed, and understand its progression through the system.

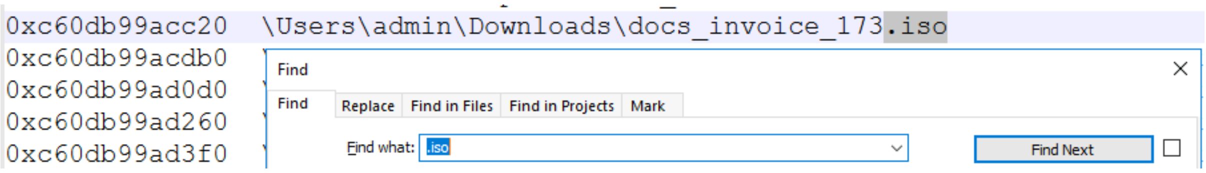
Understanding the entry point of the malware is crucial for analyzing the attack vector. Can you specify the filename of the .iso file that was used to deliver the malicious payload?

Given that we are looking for a .iso file, we can begin by using the filescan plugin on Volatility. The filescan plugin will find any files that are found in memory, we can then search the output of this command so only .iso files are displayed. Threat actors leverage .iso files as a means of delivering malware, bypassing security measures by exploiting how Windows handles mounted images. For context, ISO files are an entire optical disk stored in a single file and are similar to .rar and .zip files, however, ISOs do not use any compression.

Whenever a file is downloaded from the internet, it receives a value that gets assigned to the Zone.Identifier Alternate Data Stream (ADS). This value is referred to as the Mark-of-the-Web (MOTW) and many security tools, like Windows Defender, look for this. If you download an ISO from the internet, the file itself gets assigned a MOTW, however, the files within it do not. Therefore, once mounted, the files within the ISO image will not appear to be downloaded from the internet, which can help evade detection. To run the filescan command, execute the following command:

- `python .\vol.py -f .\memory.dmp filescan > filescan_out.txt`

We can then use any text editor to view this output, in this case, I am going to use Notepad++ and search for .iso. Given the large size of this memory dump, we are provided the output of the filescan plugin within the Artifacts directory. Upon searching for .iso, we can find a file called docs_invoice_173.iso within the admin users Downloads directory:

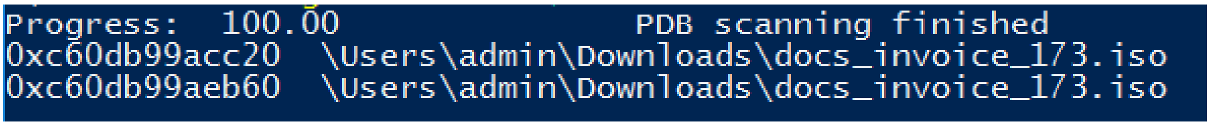


```
0xc60db99acc20   \Users\admin\Downloads\docs_invoice_173.iso
0xc60db99acdb0
0xc60db99ad0d0
0xc60db99ad260
0xc60db99ad3f0
```

Find what: .iso

Alternatively, you can use the following PowerShell command to find the .iso file:

- `python .\vol.py -f .\memory.dmp filescan | Select-String -Pattern "\.iso$"`



```
Progress:  100.00               PDB scanning finished
0xc60db99acc20  \Users\admin\Downloads\docs_invoice_173.iso
0xc60db99aeb60  \Users\admin\Downloads\docs_invoice_173.iso
```

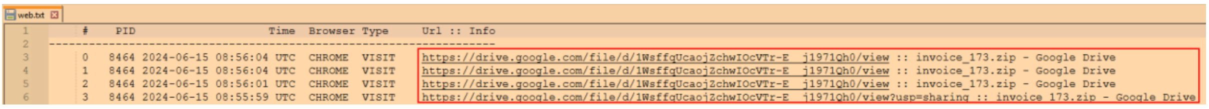
Answer: docs_invoice_173.iso

The initial delivery of the malware is crucial for understanding the attack vector. What is the link used to view the malicious malware?

To find the link used to view the malicious malware, we can use MemProcFS. MemProcFS is a tool that enables you to view memory images as files in a virtual file system. We can then navigate to where browsing history is stored and find the link associated with the malware. To run MemProcFS, you can execute the following command:

- `.\memprocfs.exe -f "C:\Users\Administrator\Desktop\Start Here\Artifacts\memory.dmp" -forensic 1`

This command mounts the output to a drive, in my case the drive letter assigned was M. Within this drive, you can find a bunch of important forensic information, including registry hives, processes, services, scheduled tasks, etc. In this case, we are concerned with the misc/web/web.txt file, which shows web browser history, specifically it shows three events: page visit, file download, and saved login data. Note! MemProcFS does take a while to process the memory dump, so be patient and don't worry if you can't see the web folder yet.



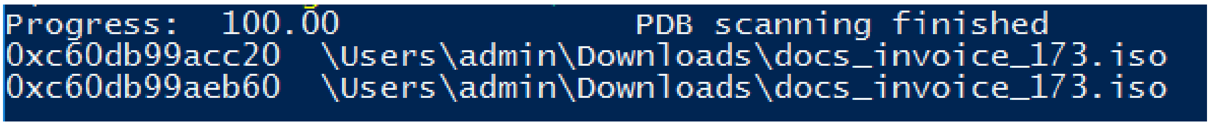
```
      #   PID                    Time  Browser Type     Url :: Info
----------------------------------------------------------------------
      0   8464 2024-06-15 08:56:04 UTC  CHROME  VISIT   https://drive.google.com/file/d/1WsffqUcaojZchwIOcVTr-E__j1971Qh0/view :: invoice_173.zip - Google Drive
      1   8464 2024-06-15 08:56:04 UTC  CHROME  VISIT   https://drive.google.com/file/d/1WsffqUcaojZchwIOcVTr-E__j1971Qh0/view :: invoice_173.zip - Google Drive
      2   8464 2024-06-15 08:56:01 UTC  CHROME  VISIT   https://drive.google.com/file/d/1WsffqUcaojZchwIOcVTr-E__j1971Qh0/view :: invoice_173.zip - Google Drive
      3   8464 2024-06-15 08:55:59 UTC  CHROME  VISIT   https://drive.google.com/file/d/1WsffqUcaojZchwIOcVTr-E__j1971Qh0/view?usp=sharing :: invoice_173.zip - Google Drive
```

As you can see in the above image, the user has downloaded a file called invoice_173.zip through google drive. Given that the iso file found previously was called docs_invoice_173.iso, it's safe to assume that this zip file contained the ISO file.

Answer: https://drive.google.com/file/d/1WsffqUcaojZchwIOcVTr-E__j1971Qh0/view

Identifying the storage location of a rogue process is critical for assessing its origin and purpose within a compromised system. What is the directory path where this process is located on the workstation?

Recall earlier when we ran the filescan plugin, we found the ISO file within the admin users Downloads directory:



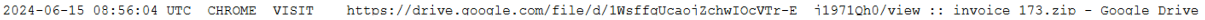
```
Progress:  100.00               PDB scanning finished
0xc60db99acc20  \Users\admin\Downloads\docs_invoice_173.iso
0xc60db99aeb60  \Users\admin\Downloads\docs_invoice_173.iso
```

Therefore, the path of this process is C:\Users\admin\Downloads.

Answer: C:\Users\admin\Downloads

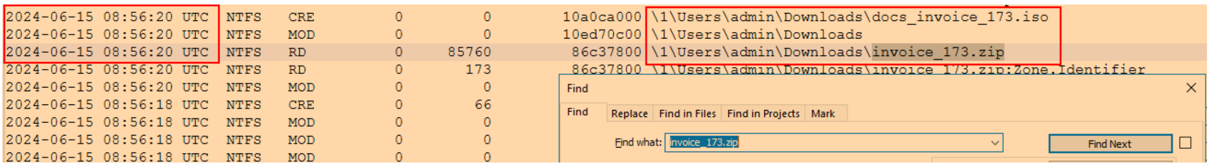
To track the timeline of the attack, it is essential to know when the malware was dropped on the system. What is the download date and time of the malicious file on the affected device?

If you look at the most recent visit time for the invouice_173.zip file in the MemProcFS web.txt file, we can see roughly when this file was downloaded:



```
2024-06-15 08:56:04 UTC  CHROME  VISIT   https://drive.google.com/file/d/1WsffgUcaojZchwIOcVTr-E_j1971Qh0/view :: invoice_173.zip - Google Drive
```

However, to be precise, we should look at NFTS artifacts to find the exact time this file was placed on disk. Fortunately for us, MemProcFS creates a timeline called timeline_ntfs.txt within the forensic folder. If you search for invoice_173.zip, we can find the exact time it was downloaded:



```
2024-06-15 08:56:20 UTC  NTFS   CRE      0        0       10a0ca000 \1\Users\admin\Downloads\docs_invoice_173.iso
2024-06-15 08:56:20 UTC  NTFS   MOD      0        0       10ed70c00 \1\Users\admin\Downloads
2024-06-15 08:56:20 UTC  NTFS   RD       0    85760        86c37800 \1\Users\admin\Downloads\invoice_173.zip
2024-06-15 08:56:20 UTC  NTFS   RD       0      173        86c37800 \1\Users\admin\Downloads\invoice_173.zip:Zone.Identifier
2024-06-15 08:56:20 UTC  NTFS   MOD      0        0
2024-06-15 08:56:18 UTC  NTFS   CRE      0       66
2024-06-15 08:56:18 UTC  NTFS   MOD      0        0
2024-06-15 08:56:18 UTC  NTFS   MOD      0        0
2024-06-15 08:56:18 UTC  NTFS   MOD      0        0
```

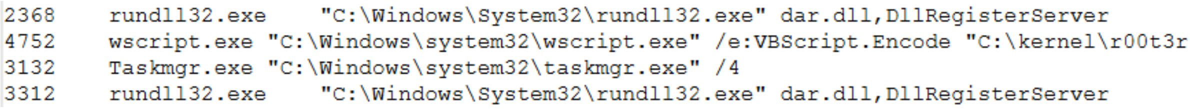
Find what: invoice_173.zip

Answer: 2024-06-15 08:56

Determining the root of the malicious activity is essential for comprehending the extent of the intrusion. What is the malicious command that triggered this malicious behaviour?

To find the malicious command that triggered this malicious behaviour, we can use the windows.cmdline Volatility plugin:

- `python .\vol.py -f .\memory.dmp windows.cmdline > cmdline_out.txt`



```
2368    rundll32.exe    "C:\Windows\System32\rundll32.exe" dar.dll,DllRegisterServer
4752    wscript.exe "C:\Windows\system32\wscript.exe" /e:VBScript.Encode "C:\kernel\r00t3r
3132    Taskmgr.exe "C:\Windows\system32\taskmgr.exe" /4
3312    rundll32.exe    "C:\Windows\System32\rundll32.exe" dar.dll,DllRegisterServer
```

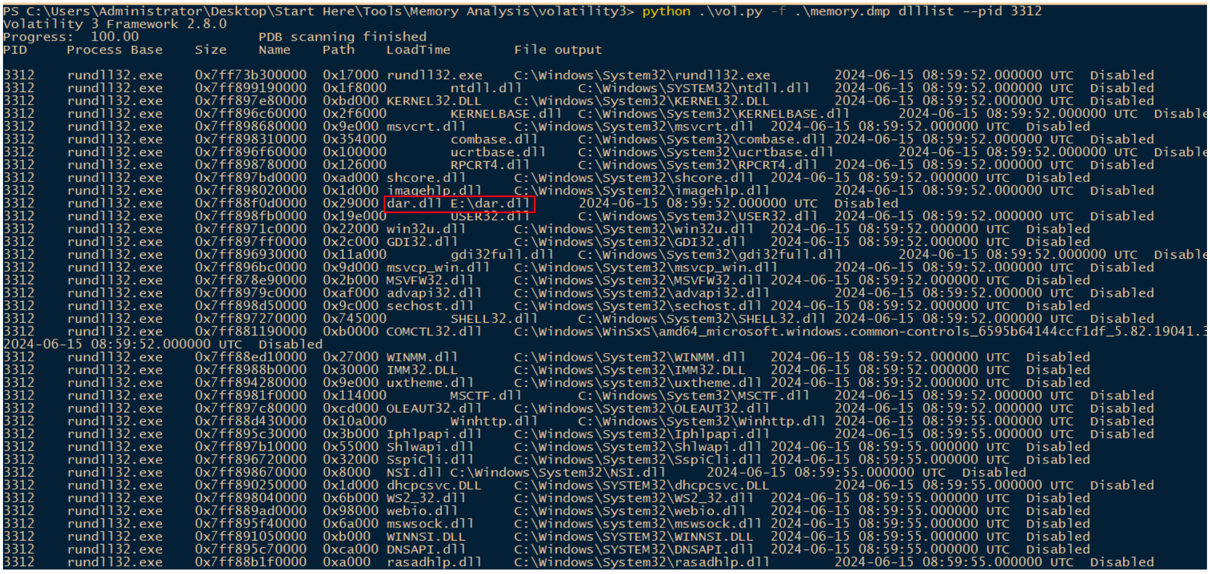
In the above image, we can see that Rundll32.exe, which is a legitimate Windows utility used to execute DLL files, was used to execute DLLRegisterServer exported by a DLL called dar.dll. Rundll32.exe is a LOLBAS that is used to execute DLL files and is often leveraged by threat actors. After googling around, I came across a post by RedCanary, that talks about how some threat actors leverage the DLLRegisterServer DLL, including Qbot, Ursnif, and Zloader.

Answer: rundll32.exe dar.dll,DllRegisterServer

Identifying file indicators is crucial for a comprehensive forensic analysis. What is the SHA256 hash of the DLL associated with the last execution of the malware?

In order to get the SHA256 hash of the dar.dll file we discovered earlier, we can use the dlllist plugin and provide the process ID (PID) of the rundll32.exe process associated with the malicious command:

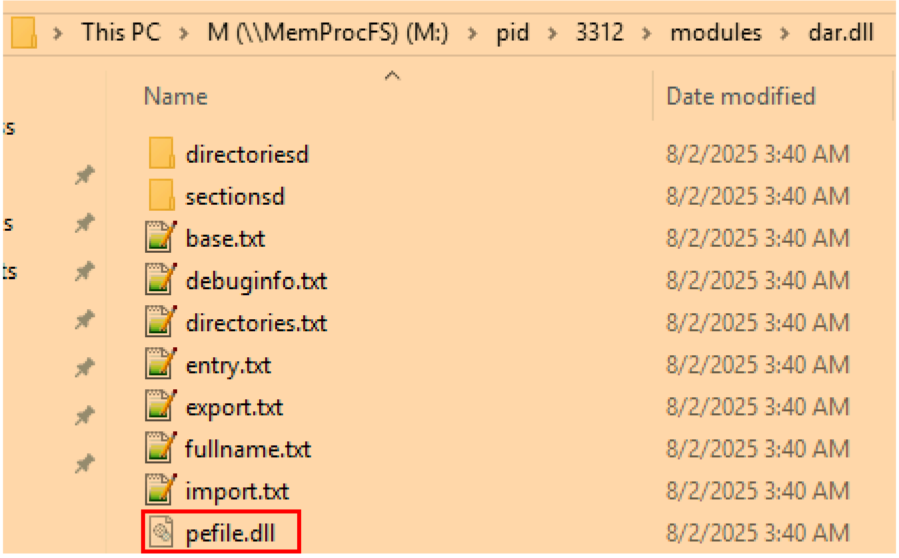
- `python .\vol.py -f .\memory.dmp dlllist --pid 3312`



```
PS C:\Users\Administrator\Desktop\Start Here\Tools\Memory Analysis\volatility3> python .\vol.py -f .\memory.dmp dlllist --pid 3312
Volatility 3 Framework 2.8.0
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output

3312    rundll32.exe    0x7ff73b300000  0x17000 rundll32.exe    C:\Windows\System32\rundll32.exe        2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff899190000  0x1f8000        ntdll.dll       C:\Windows\SYSTEM32\ntdll.dll   2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff897e80000  0xbd000 KERNEL32.DLL    C:\Windows\System32\KERNEL32.DLL        2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff896c60000  0x2f6000        KERNELBASE.dll  C:\Windows\System32\KERNELBASE.dll      2024-06-15 08:59:52.000000 UTC  Disable
3312    rundll32.exe    0x7ff898680000  0x9e000 msvcrt.dll      C:\Windows\System32\msvcrt.dll  2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff898310000  0x354000        combase.dll     C:\Windows\System32\combase.dll 2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff896f60000  0x100000        ucrtbase.dll    C:\Windows\System32\ucrtbase.dll        2024-06-15 08:59:52.000000 UTC  Disable
3312    rundll32.exe    0x7ff898780000  0x126000        RPCRT4.dll      C:\Windows\System32\RPCRT4.dll  2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff897bd0000  0xad000 shcore.dll      C:\Windows\System32\shcore.dll  2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff898020000  0x1d000 imagehlp.dll    C:\Windows\System32\imagehlp.dll        2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff88f0d0000  0x29000 dar.dll E:\dar.dll      2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff898fb0000  0x19e000        USER32.dll      C:\Windows\System32\USER32.dll  2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff8971c0000  0x22000 win32u.dll      C:\Windows\System32\win32u.dll  2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff897ff0000  0x2c000 GDI32.dll       C:\Windows\System32\GDI32.dll   2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff896930000  0x11a000        gdi32full.dll   C:\Windows\System32\gdi32full.dll       2024-06-15 08:59:52.000000 UTC  Disable
3312    rundll32.exe    0x7ff896bc0000  0x9d000 msvcp_win.dll   C:\Windows\System32\msvcp_win.dll       2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff878e90000  0x2b000 MSVFW32.dll     C:\Windows\System32\MSVFW32.dll 2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff8979c0000  0xaf000 advapi32.dll    C:\Windows\System32\advapi32.dll        2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff898d50000  0x9c000 sechost.dll     C:\Windows\System32\sechost.dll 2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff897270000  0x745000        SHELL32.dll     C:\Windows\System32\SHELL32.dll 2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff881190000  0xb0000 COMCTL32.dll    C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_5.82.19041.3
2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff88ed10000  0x27000 WINMM.dll       C:\Windows\System32\WINMM.dll   2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff8988b0000  0x30000 IMM32.DLL       C:\Windows\System32\IMM32.DLL   2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff894280000  0x9e000 uxtheme.dll     C:\Windows\system32\uxtheme.dll 2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff8981f0000  0x114000        MSCTF.dll       C:\Windows\System32\MSCTF.dll   2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff897c80000  0xcd000 OLEAUT32.dll    C:\Windows\System32\OLEAUT32.dll        2024-06-15 08:59:52.000000 UTC  Disabled
3312    rundll32.exe    0x7ff88d430000  0x10a000        Winhttp.dll     C:\Windows\System32\Winhttp.dll 2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff895c30000  0x3b000 Iphlpapi.dll    C:\Windows\System32\Iphlpapi.dll        2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff897b10000  0x55000 Shlwapi.dll     C:\Windows\System32\Shlwapi.dll 2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff896720000  0x32000 SspiCli.dll     C:\Windows\System32\SspiCli.dll 2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff898670000  0x8000  NSI.dll C:\Windows\System32\NSI.dll     2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff890250000  0x1d000 dhcpcsvc.DLL    C:\Windows\SYSTEM32\dhcpcsvc.DLL        2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff898040000  0x6b000 WS2_32.dll      C:\Windows\System32\WS2_32.dll  2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff889ad0000  0x98000 webio.dll       C:\Windows\System32\webio.dll   2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff895f40000  0x6a000 mswsock.dll     C:\Windows\system32\mswsock.dll 2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff891050000  0xb000  WINNSI.DLL      C:\Windows\SYSTEM32\WINNSI.DLL  2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff895c70000  0xca000 DNSAPI.dll      C:\Windows\SYSTEM32\DNSAPI.dll  2024-06-15 08:59:55.000000 UTC  Disabled
3312    rundll32.exe    0x7ff88b1f0000  0xa000  rasadhlp.dll    C:\Windows\System32\rasadhlp.dll        2024-06-15 08:59:55.000000 UTC  Disabled
```

As you can see in the above image, dar.dll was loaded from an external drive which is super suspicious, likely suggesting it was loaded from the mounted ISO image discovered earlier. If you navigate to the \pid\3312\modules\dar.dll folder in the output of MemProcFS, we can find a dll file called pefile.dll.

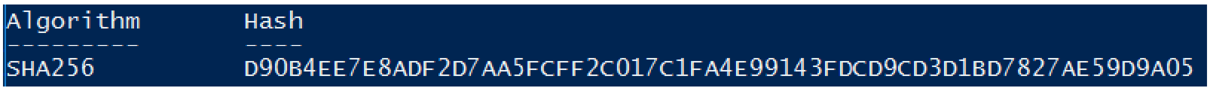


This PC › M (\\MemProcFS) (M:) › pid › 3312 › modules › dar.dll

Name	Date modified
directoriesd	8/2/2025 3:40 AM
sectionsd	8/2/2025 3:40 AM
base.txt	8/2/2025 3:40 AM
debuginfo.txt	8/2/2025 3:40 AM
directories.txt	8/2/2025 3:40 AM
entry.txt	8/2/2025 3:40 AM
export.txt	8/2/2025 3:40 AM
fullname.txt	8/2/2025 3:40 AM
import.txt	8/2/2025 3:40 AM
pefile.dll	8/2/2025 3:40 AM

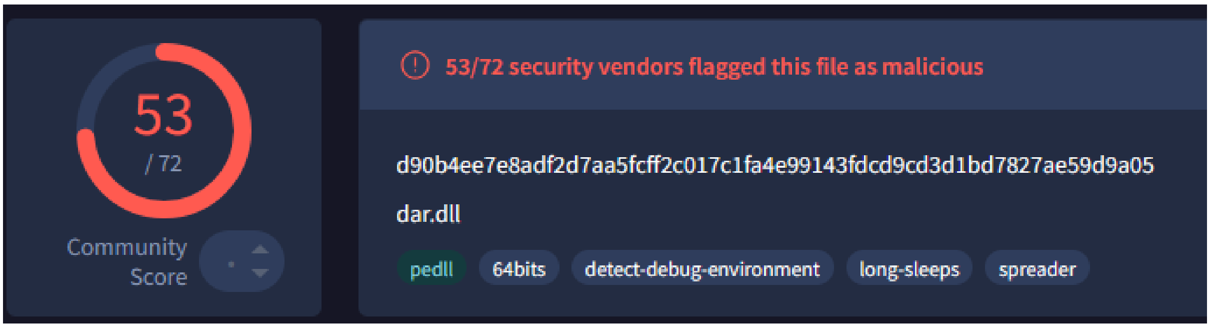
We can then use the Get-FileHash cmdlet to generate the SHA256 hash of this DLL:

- `Get-FileHash -Algorithm SHA256 .\pefile.dll`



```
Algorithm       Hash
---------       ----
SHA256          D90B4EE7E8ADF2D7AA5FCFF2C017C1FA4E99143FDCD9CD3D1BD7827AE59D9A05
```

If you submit this hash to VirusTotal, it receives 53/72 detections:



53

/ 72

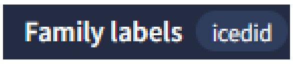
Community Score

53/72 security vendors flagged this file as malicious

d90b4ee7e8adf2d7aa5fcff2c017c1fa4e99143fdcd9cd3d1bd7827ae59d9a05

dar.dll

It is also given the family label icedid:



Family labels icedid

Answer: D90B4EE7E8ADF2D7AA5FCFF2C017C1FA4E99143FDCD9CD3D1BD7827AE59D9A05