Amadey Lab Walkthrough

Created	@June 30, 2025 8:26 AM

Scenario

An after-hours alert from the Endpoint Detection and Response (EDR) system flags suspicious activity on a Windows workstation. The flagged malware aligns with the Amadey Trojan Stealer. Your job is to analyze the presented memory dump and create a detailed report for actions taken by the malware.

Introduction

In this lab, you will step into the role of a cybersecurity analyst tasked with investigating a security incident involving the `Amadey Trojan Stealer`. Following an after-hours alert from the Endpoint Detection and Response (EDR) system, you are provided with a memory dump of the affected Windows workstation. Your objective is to uncover the details of the malicious activity, assess the malware's behavior, and determine the extent of the compromise.The `Amadey Trojan Stealer` is a well-known malware strain that specializes in `reconnaissance`, `data collection`, `credential harvesting`, and establishing `persistent connections` with remote servers. Its modular design often allows it to download additional payloads, enabling it to expand its capabilities based on attacker objectives. Because of its ability to operate in memory, identifying its traces requires n-depth forensic techniques.The process of analyzing the provided `memory dump` using `Volatility3`, a powerful memory forensics framework. You will investigate running processes, trace network activity, locate suspicious files, and uncover mechanisms the malware uses to establish persistence. Along the way, you'll apply various `Volatility3` plugins to extract critical information and build a comprehensive understanding of the attack.By the end of this lab, you will have practiced key skills in endpoint forensics, including memory analysis, identifying malicious processes, investigating network connections, and detecting persistence mechanisms. These skills are essential for identifying and responding to modern malware threats effectively

Step 1. Setting up lab using Volatility3

```
ubuntu@ip-172-31-30-174:~/Desktop/Tools/volatility3$ python3 vol.py -f Win7.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/Tools/volatility3/volatility3/symbols/windows/ntkrnl
```

Analysis

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?

```
0x1e94dcf0      TCPv4   192.168.195.136 49168   41.75.84.12     80      CLOSED  2748    lssass.exe      N/A
0x1e9846b0      UDPv4   192.168.195.136 68      *       0               768     svchost.exe     2023-08-09 21:50:07.0
0x1e988840      UDPv4   0.0.0.0 5355    *       0               920     svchost.exe     2023-08-09 21:50:07.000000
0x1e988840      UDPv6   ::      5355    *       0               920     svchost.exe     2023-08-09 21:50:07.000000
0x1ef43830      UDPv6   ::1     53403   *       0               2620    svchost.exe     2023-08-09 21:32:17.000000
```

Ans: lssass.exe

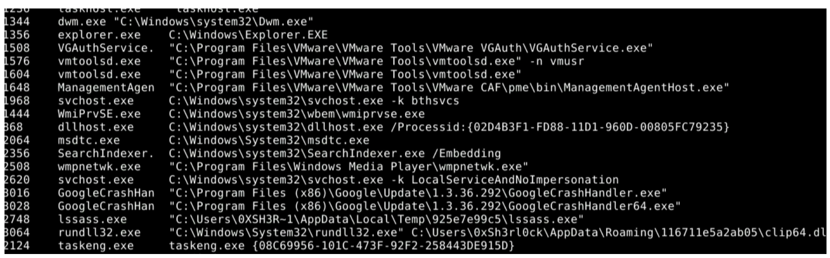
```
0x1d78f480      TCPv4   0.0.0.0 49156   0.0.0.0 0       LISTENING       508     lsass.exe
```

Another observation from the process list is the presence of a process named `lssass.exe` with PID 2524. While it closely resembles the legitimate `lsass.exe`, the extra "s" in its name suggests it is a masquerading process designed to evade detection. This technique is often used by malware to blend in with legitimate system activity. The suspicious `lssass.exe` process is also linked to a child process, `rundll32.exe` (PID 2748), which is frequently used to execute malicious code via DLL files. The creation time of `rundll32.exe` further correlates with the timeframe of the suspicious activity, reinforcing its role in the attack.The parent process responsible for triggering this behavior is identified as `lssass.exe`. Its naming convention and association with `rundll32.exe` highlight an attempt to disguise malicious operations as legitimate system activity. This finding underscores the importance of carefully examining process names, parent-child relationships, and resource usage patterns when analyzing memory dumps.

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?

The Volatility3 `windows.cmdline` plugin extracts command line arguments used by processes. This command analyzes process PID 2748:

```
python3 vol.py -f "../../Artifacts/Windows 7 ×64-Snapshot4.vmem" windows.cmdline
```



```
1344    dwm.exe "C:\Windows\system32\Dwm.exe"
1356    explorer.exe    C:\Windows\Explorer.EXE
1508    VGAuthService.  "C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe"
1576    vmtoolsd.exe    "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
1604    vmtoolsd.exe    "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe"
1648    ManagementAgen  "C:\Program Files\VMware\VMware Tools\VMware CAF\pme\bin\ManagementAgentHost.exe"
1968    svchost.exe     C:\Windows\system32\svchost.exe -k bthsvcs
1444    WmiPrvSE.exe    C:\Windows\system32\wbem\wmiprvse.exe
868     dllhost.exe     C:\Windows\system32\dllhost.exe /Processid:{02D4B3F1-FD88-11D1-960D-00805FC79235}
2064    msdtc.exe       C:\Windows\System32\msdtc.exe
2356    SearchIndexer.  C:\Windows\system32\SearchIndexer.exe /Embedding
2508    wmpnetwk.exe    "C:\Program Files\Windows Media Player\wmpnetwk.exe"
2620    svchost.exe     C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation
3016    GoogleCrashHan  "C:\Program Files (x86)\Google\Update\1.3.36.292\GoogleCrashHandler.exe"
3028    GoogleCrashHan  "C:\Program Files (x86)\Google\Update\1.3.36.292\GoogleCrashHandler64.exe"
2748    lssass.exe      "C:\Users\0XSH3R~1\AppData\Local\Temp\925e7e99c5\lssass.exe"
3064    rundll32.exe    "C:\Windows\System32\rundll32.exe" C:\Users\0xSh3rl0ck\AppData\Roaming\116711e5a2ab05\clip64.dl
2124    taskeng.exe     taskeng.exe {08C69956-101C-473F-92F2-258443DE915D}
```

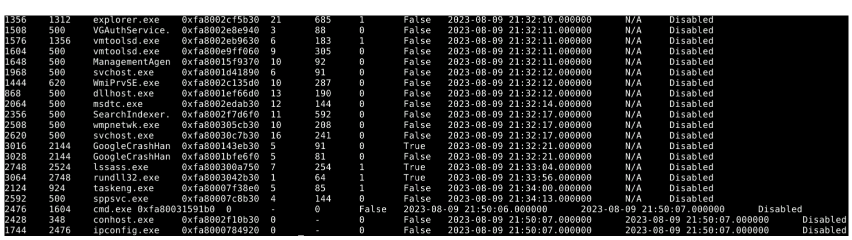
In this case, the output reveals that the executable associated with PID 2748 is named `lssass.exe` and is located in the path:

```
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?

`Command and Control` (C2) servers are central components used by attackers to remotely control infected systems and exfiltrate stolen data. These servers allow attackers to send commands to compromised machines, retrieve sensitive information, or deploy additional payloads, making them a critical part of modern malware operations. Detecting communications with C2 servers is vital for incident response as it helps identify ongoing data exfiltration or remote control activities.

`python3 vol.py -f "../../Artifacts/Windows 7 ×64-Snapshot4.vmem" windows.netscan | grep 2748`



```
1356  1312  explorer.exe    0xfa8002cf5b30  21  685  1  False  2023-08-09 21:32:10.000000  N/A  Disabled
1508  500   VGAuthService.  0xfa8002e8e940  3   88   0  False  2023-08-09 21:32:11.000000  N/A  Disabled
1576  1356  vmtoolsd.exe    0xfa8002eb9630  6   183  1  False  2023-08-09 21:32:11.000000  N/A  Disabled
1604  500   vmtoolsd.exe    0xfa800e9ff060  9   305  0  False  2023-08-09 21:32:11.000000  N/A  Disabled
1648  500   ManagementAgen  0xfa80015f9370  10  92   0  False  2023-08-09 21:32:11.000000  N/A  Disabled
1968  500   svchost.exe     0xfa8001d41890  6   91   0  False  2023-08-09 21:32:12.000000  N/A  Disabled
1444  620   WmiPrvSE.exe    0xfa8002c135d0  10  287  0  False  2023-08-09 21:32:12.000000  N/A  Disabled
868   500   dllhost.exe     0xfa8001ef66d0  13  190  0  False  2023-08-09 21:32:12.000000  N/A  Disabled
2064  500   msdtc.exe       0xfa8002edab30  12  144  0  False  2023-08-09 21:32:14.000000  N/A  Disabled
2356  500   SearchIndexer.  0xfa8002f7d6f0  11  592  0  False  2023-08-09 21:32:17.000000  N/A  Disabled
2508  500   wmpnetwk.exe    0xfa800305cb30  10  208  0  False  2023-08-09 21:32:17.000000  N/A  Disabled
2620  500   svchost.exe     0xfa80030c7b30  16  241  0  False  2023-08-09 21:32:17.000000  N/A  Disabled
3016  2144  GoogleCrashHan  0xfa800143eb30  5   91   0  True   2023-08-09 21:32:21.000000  N/A  Disabled
3028  2144  GoogleCrashHan  0xfa8001bfe6f0  5   81   0  False  2023-08-09 21:32:21.000000  N/A  Disabled
2748  2524  lssass.exe      0xfa800300a750  7   254  1  True   2023-08-09 21:33:04.000000  N/A  Disabled
3064  2748  rundll32.exe    0xfa8003042b30  1   64   1  True   2023-08-09 21:33:56.000000  N/A  Disabled
2124  924   taskeng.exe     0xfa80007f38e0  5   85   1  False  2023-08-09 21:34:00.000000  N/A  Disabled
2592  500   sppsvc.exe      0xfa80007c8b30  4   144  0  False  2023-08-09 21:34:13.000000  N/A  Disabled
2476  1604  cmd.exe 0xfa80031591b0  0  -  0  False  2023-08-09 21:50:06.000000  2023-08-09 21:50:07.000000  Disabled
2428  348   conhost.exe     0xfa8002f10b30  0  -  0  False  2023-08-09 21:50:07.000000  2023-08-09 21:50:07.000000  Disabled
1744  2476  ipconfig.exe    0xfa8000784920  0  -  0  False  2023-08-09 21:50:07.000000  2023-08-09 21:50:07.000000  Disabled
```

The output highlights two closed connections associated with this process. Both connections are directed to the external IP address `41.75.84.12` over port 80, which is commonly used for HTTP traffic.The IPaddress `41.75.84.12` is likely the Command and Control (C2) server utilized by the malware.

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?
Analyzing suspicious processes often involves inspecting memory dumps to identify any associated network activity or URLs that might point to malicious behavior. Since the process in question was shown to be using HTTP traffic and established connections with external IP addresses, memory analysis is extended to investigate further indicators such as URLs and embedded resources. HTTP traffic is frequently exploited by malware to communicate with Command and Control (C2) servers or download additional payloads. In this case, the process memory dump is examined for traces of URLs and HTTP requests that may reveal the malware’s behavior.

`python3 vol.py -f "../../Artifacts/Windows 7 ×64-Snapshot4.vmem" windows.memmap`

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

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?

To locate the malicious files downloaded by the malware, the `windows.filescan` *plugin in Volatility3 is employed.*

The output reveals the full path to one of these files:

```
0x1d791940  100.0 \Users\0xSh3rl0ck\AppData\Roaming\116711e5a2ab05\clip64.dll
```

Q6 Once retrieved, the malware aims to activate its additional components.Which child process is initiated by the malware to execute these files?Based on the analysis conducted earlier, the malware employs a modular design, downloading and utilizing additional components stored as DLL files. These files, such as `cred64.dll` and `clip64.dll`,were retrieved via HTTP requests and stored in the AppData directory, as observed in the process memory and file scan analysis. Given that the second-stage payload is a DLL, it is likely executed using a Windows utility designed to load and run DLL files.The investigation into the suspicious process `lssass.exe` with PID 2748 revealed its use of `rundll32.exe` as a child process. This utility is a legitimate Windows tool often exploited by attackers to execute malicious DLLs. The connection between `lssass.exe` and `rundll32.exe` was established earlier during process enumeration, where `rundll32.exe` appeared as a child process. This relationship is highly suspicious, as it aligns with the behavior of the Amadey Trojan, which frequently leverages `rundll32.exe` to execute downloaded payloads, enabling further actions such as credential theft, reconnaissance, or persistence.In this case, `rundll32.exe` likely served as the mechanism to load and activate the retrieved DLL files (`cred64.dll` and `clip64.dll`). This execution technique allows the malware to expand its capabilities without drawing attention, as `rundll32.exe` is a trusted Windows binary, and its activity may bypass basic security monitoring tools.

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?

To investigate additional persistence mechanisms used by the malware, the `windows.filescan` plugin in Volatility3 is executed to scan for file artifacts associated with the process.

```
ubuntu@ip-172-31-30-174:~/Desktop/Tools/volatility3$ python3 vol.py -f Win7.vmem windows.filescan | grep lssass
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-172-31-30-174:~/Desktop/Tools/volatility3$
```

The results reveal two key file locations associated with the suspicious `lssass.exe` process:

1. `\Users\0XSH3R~1\AppData\Local\Temp\925e7e99C5\lssass.exe`
2. `\Windows\System32\Tasks\lssass.exe`

The first path, located in the Temp directory, suggests a temporary drop point where the malware was initially staged or executed. However, the second path under `System32\Tasks` indicates a more deliberate persistence mechanism. Files stored in the `Tasks` directory are typically associated with Windows Task Scheduler, a legitimate tool used to automate tasks. Malware often abuses scheduled tasks to establish persistence by creating tasks that execute malicious files at predefined intervals or system events.The presence of `lssass.exe` in the `Tasks`directory implies that the malware likely registered itself as a scheduled task. This approach allows it to automatically restart upon reboot or at scheduled times, ensuring continuous execution. Task Scheduler persistence is particularly stealthy because it leverages a trusted Windows component, making it less likely to trigger alarms in security monitoring tools.