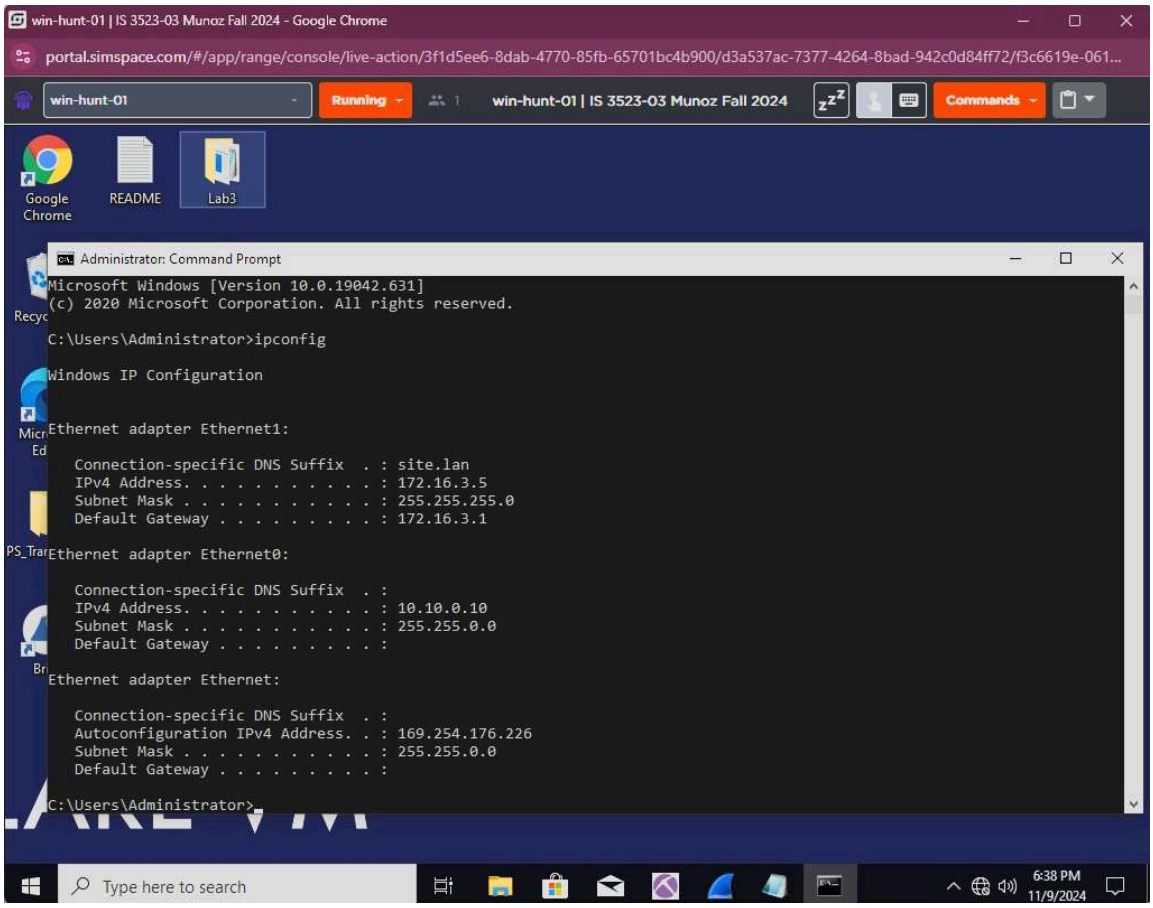
David Murillo Santiago
Professor Munoz
10 Nov 2024
IS3523

Hunting in Memory Lab

In this lab, I was tasked with investigating a memory image named KobayashiMaru.vmem to determine if the source device had been compromised. The user expressed concerns about a potential compromise but provided no additional context regarding the image source or the reasons behind their suspicion. My objective was to perform a thorough forensic analysis to uncover any evidence indicating what may have occurred on the device.

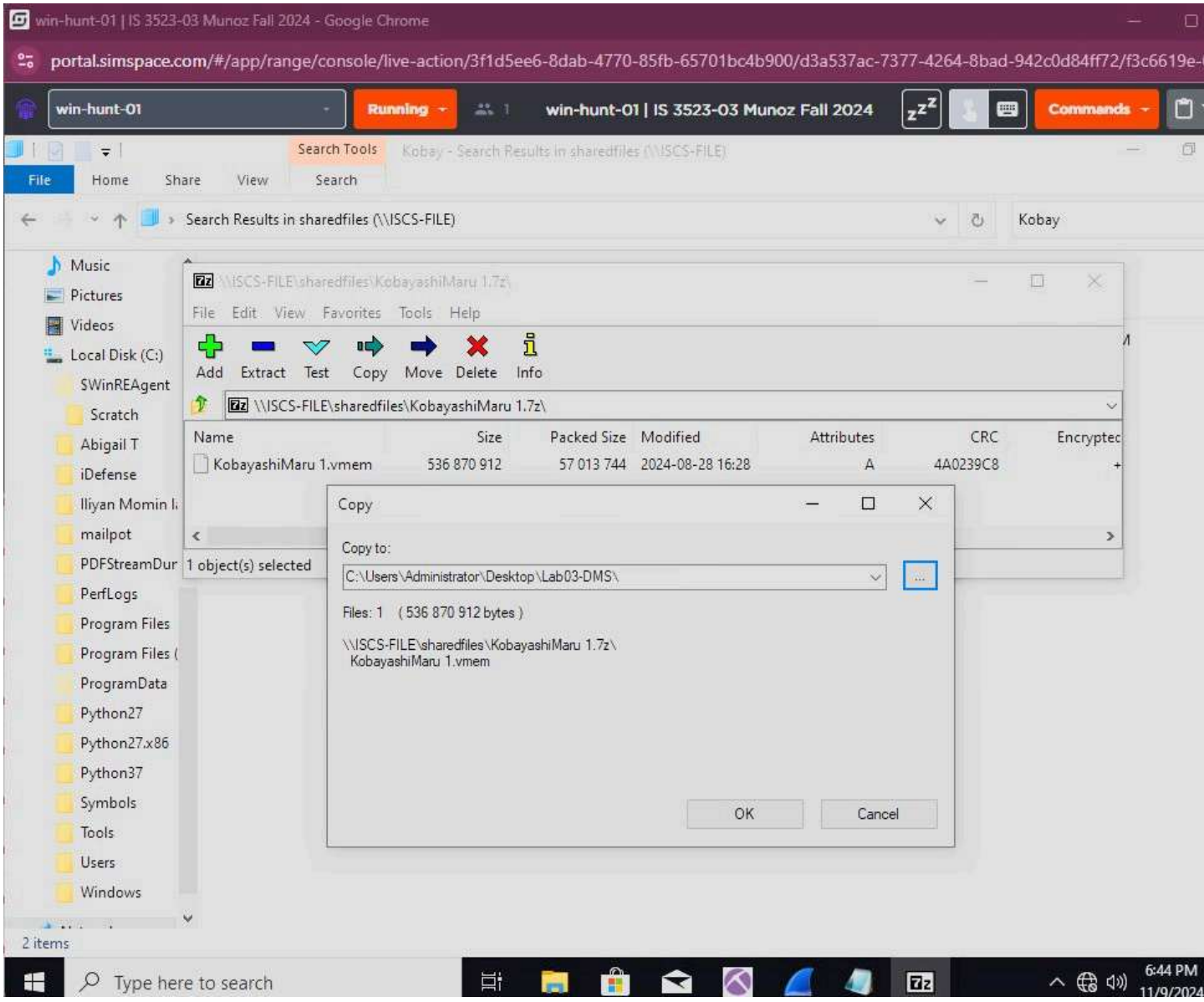
Forensic Investigation:

For this lab, I conducted the forensic investigation from the host device Win-hunt Device 1.



Win-hunt-01

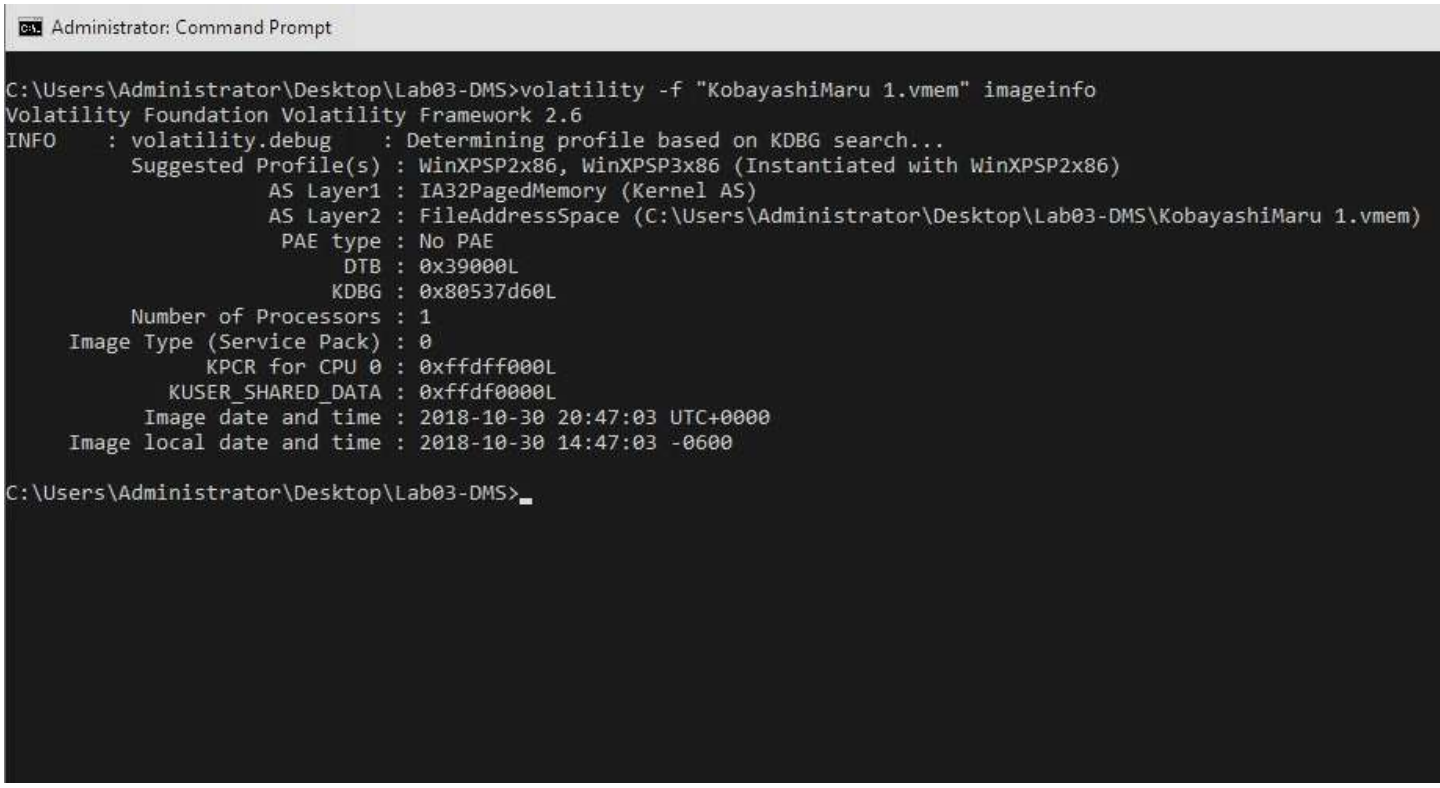
To gain access to the image, I went to the shared folder and, using 7-Zip, extracted the zipped memory image.



7Zip unzip Memory Image

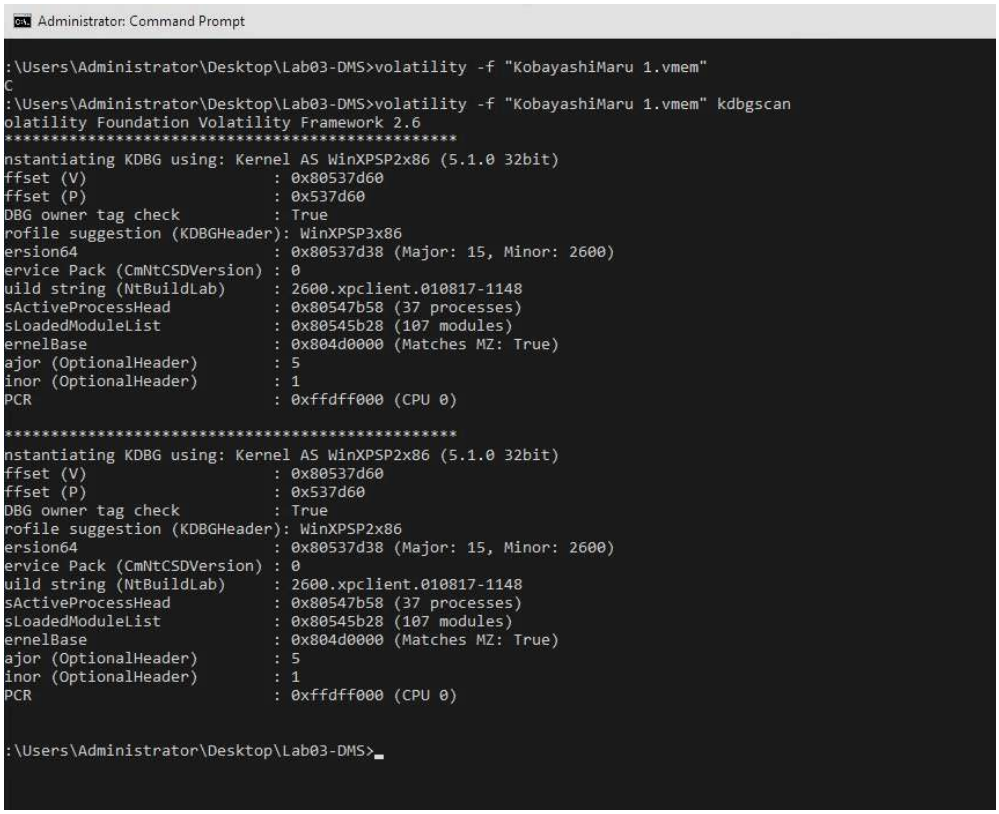
To begin the forensic investigation, I used volatility, an open-source memory forensics framework that enables extraction and analysis of digital evidence from a computer's volatile memory.

I used volatility's imageinfo command to pull critical information from the memory image. Looking into the output, I determined the following: The image originates from either WinXPSP2x86 or WinXPSP3x86. Therefore, I analyzed further to determine the version type and source OS.



Volatility Imageinfo scan

Using kdbgscan, I was able to determine the OS and version type. The image originates from a WinXPSP2x86 OS, version 5.1.0 32-bit.



Volatility kdbgscan

Additionally, I used the dir command to view the size of the image, which in turn would reveal the size of the RAM from which the image was derived. In this case, I found that the RAM was 536,870,912 bytes, meaning that the memory was approximately 500 MB in size.

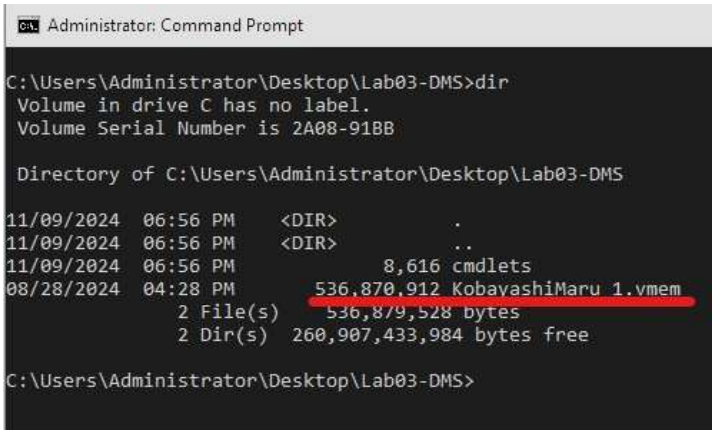
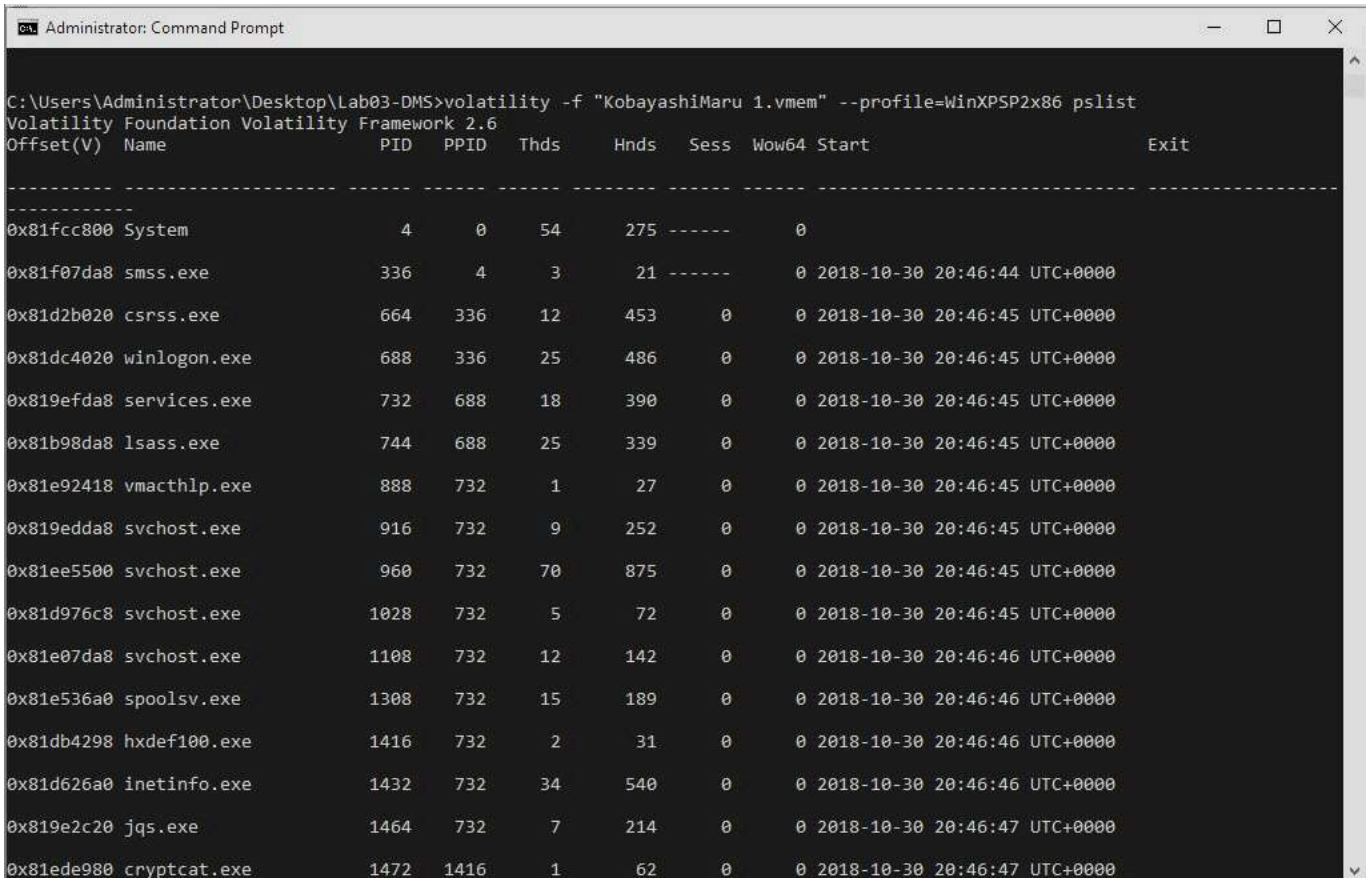


Image Size in bytes

Next, I analyzed the running processes of the image by using the pslist command. I examined the process hierarchy, verifying the process ID and parent process ID. I realized that several processes had spawned without an apparent valid parent process. This means the process appears to have spawned out of nowhere, as the listed parent process is not part of the process list. For this reason, the following processes are suspicious:



Volatility Pslist Scan

- **Threat name**: iroffer.exe
  - Process ID: 1692
  - Parent Process: 1488
- **Threat name**: soffice.exe
  - Process ID: 516
  - Parent Process ID: 496
- **Threat name**: nc.exe
  - Process ID: 532
  - Parent Process ID: 508
- **Threat name**: winvnc4.exe
  - Process ID: 548
  - Parent Process ID: 508
- **Threat name**: cmd.exe
  - Process ID: 560
  - Parent Process ID: 508
- **Threat name**: poisonivy.exe

    - Process ID: 480
    - Parent Process ID: 404
    - **Reasoning**: poisonivy.exe is a known remote access tool that can be used to backdoor Windows 2000, Windows XP, and Windows Server 2003. Unlike the previously listed threats, poisonivy.exe does have a listed parent process, as it was invoked by explorer.exe (404).
- **Threat name**: cryptcat.exe
    - Process ID: 1472
    - Parent Process: 1416
    - **Reasoning**: Cryptcat is netcat but secured by encryption. At its safest, this application is considered riskware, as it can be exploited to create backdoors and to install rootkits.

The following three processes also have a listed parent process, but they originate from a suspicious process that seemingly appeared out of nowhere. Therefore, I will log them for further analysis.

- **Threat name**: iroffer.exe
    - Process ID: 1824
    - Parent Process: 1728
- **Threat name**: soffice.bin
    - Process ID: 524
    - Parent Process: 516
- **Threat name**: iroffer.exe
    - Process ID: 1728
    - Parent Process: 1692

Name	PID	PPID	Start	Exit	
System	4	0	20:46:44		
smss.exe	336	4	20:46:45		
csrss.exe	664	336	20:46:45		
winlogon.exe	688	336	20:46:45		
services.exe	732	688	20:46:45		
lsass.exe	744	688	20:46:45		
vmacthlp.exe	888	732	20:46:45		
svchost.exe	916	732	20:46:45		
svchost.exe	960	732	20:46:45		
svchost.exe	1028	732	20:46:45		
svchost.exe	1108	732	20:46:46		
spoolsv.exe	1308	732	20:46:46		
hxdef100.exe	1416	732	20:46:46		
inetinfo.exe	1432	732	20:46:46		
jqs.exe	1464	732	20:46:47		
cryptcat.exe	1472	1416	20:46:47		
bircd.exe	1480	1416	20:46:47		
VMwareService.e	1624	732	20:46:47		
iroffer.exe	1692	1488		20:46:47	
iroffer.exe	1728	1692	20:46:47		# DID have PPID but from SUS app
iroffer.exe	1824	1728	20:46:47	20:46:36	# DID have PPID but from SUS app
wmiapsrv.exe	216	732	20:46:36		
wmiapsrv.exe	252	916	20:46:37		
userinit.exe	368	688	20:46:38		
explorer.exe	404	368	20:46:38		
VMwareTray.exe	456	404	20:46:38		
VMwareUser.exe	464	404	20:46:38		
jusched.exe	472	404	20:46:38		
poisonivy.exe	480	404	20:46:38		# rootkit
msmsgs.exe	488	404	20:46:39		
soffice.exe	516	496	20:46:39		
soffice.bin	524	516	20:46:39		# DID have PPID but from SUS app
nc.exe	532	508	20:46:39		
winvnc4.exe	548	508	20:46:39		
cmd.exe	560	508	20:46:39		
logonui.exe	636	688	20:46:40		
rundll32.exe	984	404	20:46:43		

Process Hierarchy and Highlighted Suspicious Processes

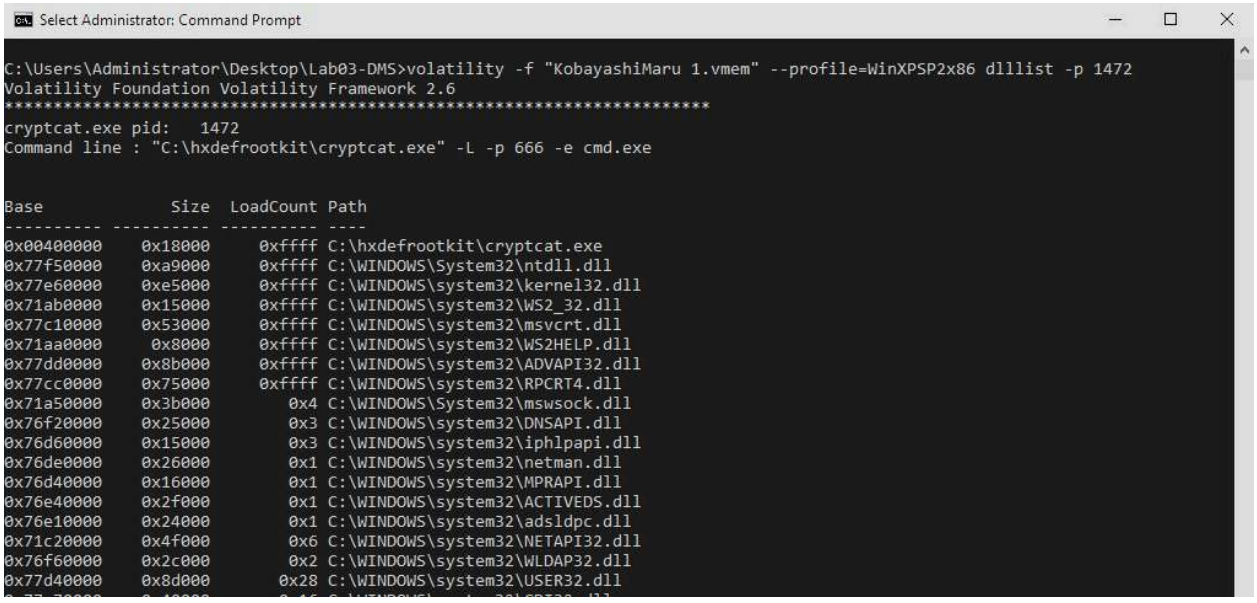
Next, I made a list of all the processes I will analyze further to organize the investigation. I will not only analyze the suspicious applications but also their parent processes in case I find more information to form a clearer picture.

App	PID	PPID
cryptcat.exe	1472	1416
hxdef100.exe	1416	732
iroffer.exe	1692	1488
UNKOWN	1488	
poisonivy.exe	480	404
explorer.exe	404	
soffice.exe	516	496
UNKOWN	496	
nc.exe	532	508
winvnc4.exe	548	508
cmd.exe	560	508
UNKOWN	508	
iroffer.exe	1728	1692
iroffer.exe	1824	1728
soffice.bin	524	516

Processes to Further Analyze

Next, I began using dlllist on every process on the list to extrapolate the command entered to start the process and to view additional information about the dynamic link libraries associated with the process.

Analyzing the cryptcat.exe process ID with dlllist, I found that the application was run by entering: C:\hxdefrootkit\cryptcat.exe -L -p 666 -e cmd.exe. This is alarming as it reveals that the cryptcat command is part of a rootkit and is establishing a reverse shell with a remote device on port 666.



Volatility Dlllist Cryptcat PID 1472

Analyzing the nc.exe process (532), I found the initiation of another reverse shell. This time from netcat and on port 6666. The following command was entered:
C:\inetpub\ftproot\nc.exe -L -p 6666 -e cmd.exe

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 dlllist -p 532
Volatility Foundation Volatility Framework 2.6
************************************************************************
nc.exe pid:    532
Command line : C:\inetpub\ftproot\nc.exe  -L -p 6666 -e cmd.exe


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00400000    0x10000     0xffff C:\inetpub\ftproot\nc.exe
0x77f50000    0xa9000     0xffff C:\WINDOWS\System32\ntdll.dll
0x77e60000    0xe5000     0xffff C:\WINDOWS\system32\kernel32.dll
0x71ab0000    0x15000     0xffff C:\WINDOWS\System32\WS2_32.dll
0x77c10000    0x53000     0xffff C:\WINDOWS\system32\msvcrt.dll
0x71aa0000     0x8000     0xffff C:\WINDOWS\System32\WS2HELP.dll
0x77dd0000    0x8b000     0xffff C:\WINDOWS\system32\ADVAPI32.dll
0x77cc0000    0x75000     0xffff C:\WINDOWS\system32\RPCRT4.dll
0x71a50000    0x3b000        0x4 C:\WINDOWS\System32\mswsock.dll
0x76f20000    0x25000        0x3 C:\WINDOWS\System32\DNSAPI.dll
0x76d60000    0x15000        0x3 C:\WINDOWS\System32\iphlpapi.dll
0x76de0000    0x26000        0x1 C:\WINDOWS\System32\netman.dll
0x76d40000    0x16000        0x1 C:\WINDOWS\System32\MPRAPI.dll
0x76e40000    0x2f000        0x1 C:\WINDOWS\System32\ACTIVEDS.dll
0x76e10000    0x24000        0x1 C:\WINDOWS\System32\adsldpc.dll
0x71c20000    0x4f000        0x6 C:\WINDOWS\System32\NETAPI32.dll
```

Volatility Dlllist Netcat PID 532

I also analyzed the winvnc4.exe process (548). This executable stood out for two reasons: the first being that the process appeared seemingly out of nowhere, and the second being the command entered to initiate the process: C:\inetpub\ftproot\VNC4\winvnc4.exe. From the command, I deduced that the winvnc4 executable originates from a shared folder with the netcat executable. This suggests that the application could be part of the attacker's toolkit. After further research, I found that winvnc4.exe is legitimate remote access software, but it is considered riskware as it can be exploited to gain remote access to a system. Another point to note is that winvnc4.exe's default location is C:\Program Files(x86)\RealVNC\VNC4. Therefore, it is abnormal for the program to be stored in the ftproot folder in this way.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 dlllist -p 548
Volatility Foundation Volatility Framework 2.6
************************************************************************
winvnc4.exe pid:    548
Command line : C:\inetpub\ftproot\VNC4\winvnc4.exe


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00400000    0x6c000     0xffff C:\inetpub\ftproot\VNC4\winvnc4.exe
0x77f50000    0xa9000     0xffff C:\WINDOWS\System32\ntdll.dll
0x77e60000    0xe5000     0xffff C:\WINDOWS\system32\kernel32.dll
0x77d40000    0x8d000     0xffff C:\WINDOWS\system32\USER32.dll
0x77c70000    0x40000     0xffff C:\WINDOWS\system32\GDI32.dll
0x77dd0000    0x8b000     0xffff C:\WINDOWS\system32\ADVAPI32.dll
0x77cc0000    0x75000     0xffff C:\WINDOWS\system32\RPCRT4.dll
0x773d0000   0x7f4000     0xffff C:\WINDOWS\system32\SHELL32.dll
0x77c10000    0x53000     0xffff C:\WINDOWS\system32\msvcrt.dll
0x772d0000    0x63000     0xffff C:\WINDOWS\system32\SHLWAPI.dll
0x71ab0000    0x15000     0xffff C:\WINDOWS\System32\WS2_32.dll
0x71aa0000     0x8000     0xffff C:\WINDOWS\System32\WS2HELP.dll
0x77c00000     0x7000     0xffff C:\WINDOWS\system32\VERSION.dll
```

Volatility Dlllist Winvnc4 PID 548

Next, I analyzed the cmd.exe process (560). The following command was entered: C:\WINDOWS\system32\cmd.exe /K C:\Inetpub\ftproot\lock.bat. This command is notable as it initiates a command prompt with the /k flag, meaning the command prompt will remain displayed after the command is entered to view the output. The command also pulls an application from the ftproot folder, this time lock.bat, a script used to hide files by changing their attributes. This may have been used by the attacker to cover their tracks, explaining the missing processes.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 dlllist -p 560
Volatility Foundation Volatility Framework 2.6
************************************************************************
cmd.exe pid:    560
Command line : C:\WINDOWS\system32\cmd.exe  /K C:\Inetpub\ftproot\lock.bat


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x4ad00000    0x5e000     0xffff C:\WINDOWS\system32\cmd.exe
0x77f50000    0xa9000     0xffff C:\WINDOWS\System32\ntdll.dll
0x77e60000    0xe5000     0xffff C:\WINDOWS\system32\kernel32.dll
0x77c10000    0x53000     0xffff C:\WINDOWS\system32\msvcrt.dll
0x77d40000    0x8d000     0xffff C:\WINDOWS\system32\USER32.dll
0x77c70000    0x40000     0xffff C:\WINDOWS\system32\GDI32.dll
0x77dd0000    0x8b000     0xffff C:\WINDOWS\system32\ADVAPI32.dll
0x77cc0000    0x75000     0xffff C:\WINDOWS\system32\RPCRT4.dll
```

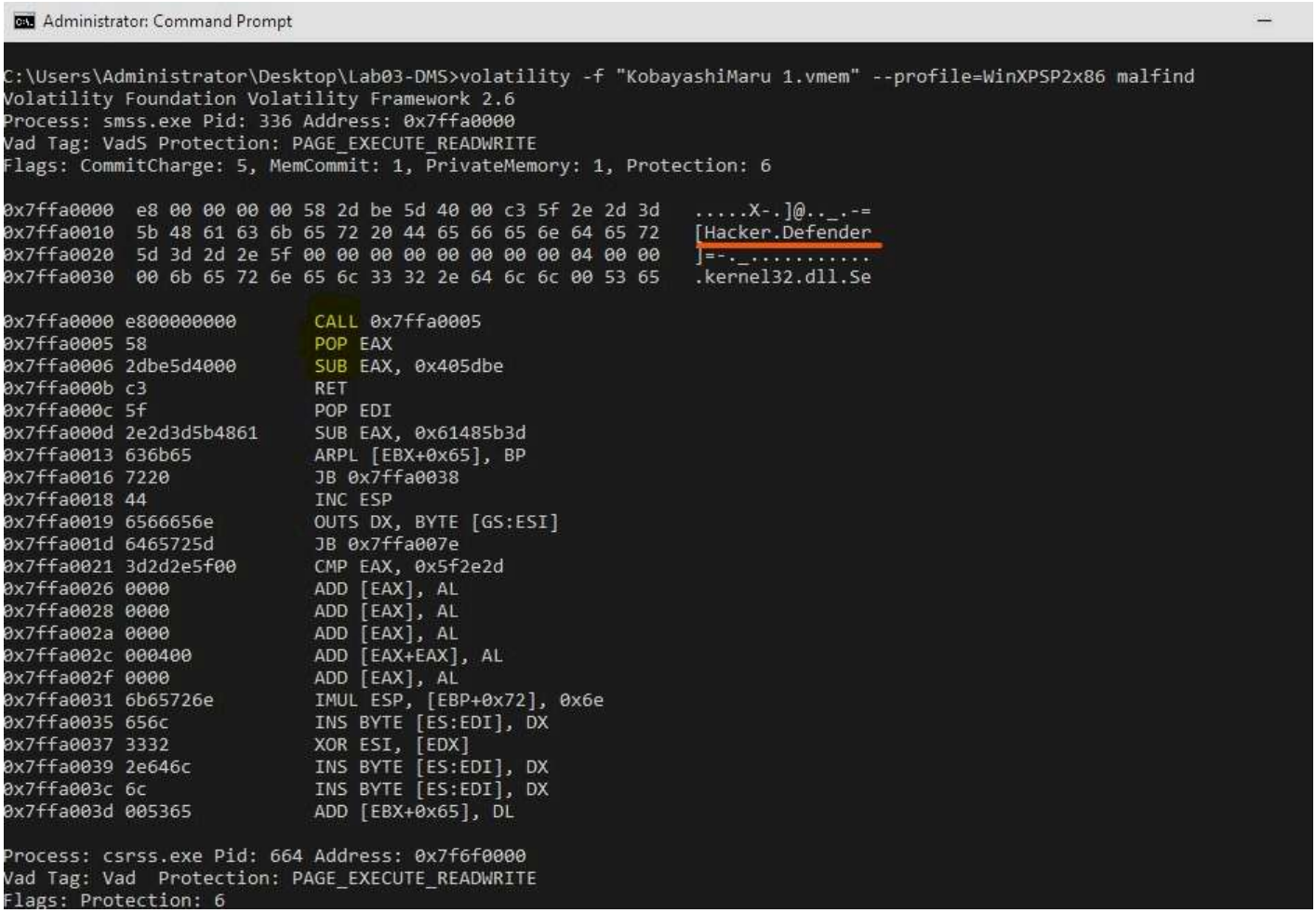
Volatility Dlllist Cmd PID 560

Next, I analyzed the iroffer.exe process (1728). This process is alarming for four reasons. First, its parent process (1692) was unlocatable by either dlllist or psscan, so no information could be pulled from its parent process. Second, iroffer.exe is a known backdoor vulnerability. Third, it was located inside a hidden folder, implying that its location was chosen to keep a low profile. Fourth, it uses two DLLs (cygcrypt-0.dll and cygwin1.dll) originating from the same hidden folder. These DLLs are used for cygwin, a tool to create a Linux-like environment on Windows. However, these DLLs are illegitimate as they originate from the obscure "hidden" folder.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 dlllist -p 1728
Volatility Foundation Volatility Framework 2.6
************************************************************************
iroffer.exe pid:   1728
Command line : C:\hidden\ir\iroffer.exe


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00400000    0x39000     0xffff C:\hidden\ir\iroffer.exe
0x77f50000    0xa9000     0xffff C:\WINDOWS\System32\ntdll.dll
0x77e60000    0xe5000     0xffff C:\WINDOWS\system32\kernel32.dll
0x10000000     0x7000     0xffff C:\hidden\ir\cygcrypt-0.dll
0x61000000   0x259000     0xffff C:\hidden\ir\cygwin1.dll
0x77dd0000    0x8b000     0xffff C:\WINDOWS\system32\ADVAPI32.DLL
0x77cc0000    0x75000     0xffff C:\WINDOWS\system32\RPCRT4.dll
0x71ad0000     0x8000        0x1 C:\WINDOWS\system32\wsock32.dll
0x71ab0000    0x15000       0x12 C:\WINDOWS\system32\WS2_32.dll
0x77c10000    0x53000       0x15 C:\WINDOWS\system32\msvcrt.dll
0x71aa0000     0x8000       0x15 C:\WINDOWS\system32\WS2HELP.dll
0x71a50000    0x3b000        0x3 C:\WINDOWS\system32\mswsock.dll
0x71a90000     0x8000        0x1 C:\WINDOWS\System32\wshtcpip.dll
0x76b40000    0x2c000        0x1 C:\WINDOWS\system32\winmm.dll
0x77d40000    0x8d000        0x2 C:\WINDOWS\system32\USER32.dll
0x77c70000    0x40000        0x2 C:\WINDOWS\system32\GDI32.dll
```

Volatility Dlllist Iroffer PID 1728

Next, I ran a malfind search on the memory to find assembly code hidden within processes. As a result, I found that 30 processes contained call, pop, and sub instructions, which could be used by threat actors to transfer control between memory addresses and manipulate the return address to hide their actions. Additionally, each of these processes contained the string **“Hacker.defender”**, a known rootkit that modifies Windows and native API functions to hide information from other applications.



Volatility Malfind Scan

Therefore, 30 processes have been affected by the Hacker Defender rootkit.

The affected processes include:

Application	Process ID
smss.exe	336
csrss.exe (PID 664)	664
winlogon.exe	688

services.exe	732
lsass.exe	744
vmacthlp.exe	888
svchost.exe	916
svchost.exe	960
svchost.exe	1028
svchost.exe	1108
spoolsv.exe	1308
inetinfo.exe	1432
jqs.exe	1464
VMwareService.exe	1624
wmiapsrv.exe	216
wmiapsrv.exe	252
userinit.exe	368
explorer.exe	404
VMwareTray.exe	456
VMwareUser.exe	464
jusched.exe	472
poisonivy.exe	480
msmsgs.exe	488
soffice.exe	516
soffice.bin	524
nc.exe	532
winvnc4.exe	548
cmd.exe	560
logonui.exe	636
rundll32.exe	984

Additionally, I ran a volatility cmdline scan on both the known infected processes, as well as the previously mentioned suspicious processes. This would provide me with key information such as file path and what command was used to execute the program.

Malicious Processes		
Process Name	PID	cmdline
smss.exe	336	\SystemRoot\System32\smss.exe
csrss.exe	664	n=1024,3072,512 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=
winlogon.exe	688	winlogon.exe
services.exe	732	C:\WINDOWS\system32\services.exe
lsass.exe	744	C:\WINDOWS\system32\lsass.exe
vmacthlp.exe	888	"C:\Program Files\Vmware\Vmware Tools\vmacthlp.exe"
svchost.exe	916	C:\WINDOWS\system32\svchost -k rpcss
svchost.exe	960	C:\WINDOWS\System32\svchost.exe -k netsvcs
svchost.exe	1028	C:\WINDOWS\System32\svchost.exe -k NetworkService
svchost.exe	1108	C:\WINDOWS\System32\svchost.exe -k LocalService
spoolsv.exe	1308	C:\WINDOWS\system32\spoolsv.exe
inetinfo.exe	1432	C:\WINDOWS\Sytem32\inetsrv\inetifo.exe
jqs.exe	1464	"C:\Program Files\Java\jre6\bin\jqs.exe" -service -config "C:\Program Files\Java\jre6\lib\deploy\jqs\jqs.conf"
VMwareService.exe	1624	C:\Program Files\VMware\VMware Tools\VMwareService.exe
wmiapsrv.exe	216	C:\WINDOWS\System32\wbem\wmiapsrv.exe
wmiapsrv.exe	252	C:\WINDOWS\System32\wbem\wmiapsrv.exe
userinit.exe	368	C:\WINDOWS\system32\userinit.exe
explorer.exe	404	C:\WINDOWS\Explorer.exe
VMwareTray.exe	456	"C:\Program Files\Vmware\Vmware Tools\VMwareTray.exe"
VMwareUser.exe	464	"C:\Program Files\Vmware\Vmware Tools\VMwareUser.exe"
jusched.exe	472	"C:\Program Files\Common Files\Java\Java Update\jusched.exe"
poisonivy.exe	480	C:\WINDOWS\System32\poisonivy.exe
msmsgs.exe	488	"C:\Program Files\Messenger\msmsgs.exe" /background
soffice.exe	516	"C:\Program Files\OpenOffice.org 3\program\soffice.exe" "-quickstart"
soffice.bin	524	"C:\Program Files\OpenOffice.org 3\program\soffice.exe" "-quickstart" "-env:000_CWD=2C:\\Program\\OpenOffice.org 3\\program"
nc.exe	532	C:\inetpub\ftproot\nc.exe -L -p 6666 -e cmd.exe
winvnc4.exe	548	C:\inetpub\ftproot\VNC4\winvnc4.exe
cmd.exe	560	C:\WINDOWS\system32\cmd.exe
logonui.exe	636	logonui.exe /status
rundll32.exe	984	C:\WINDOWS\System32\rundll32.exe fldrclnr.dll,Wizard_RunDLL
iroffer.exe	1728	C:\hidden\ir\iroffer.exe
bircd.exe	1480	C:\hidden\bewareircd-win32\bircd.exe
cryptcat.exe	1472	C:\hxdefrootkit\cryptcat.exe -L -p 666 -e cmd.exe
hxdef100.exe	1416	C:\hxdefrootkit\hxdef100.exe hxdef100.ini

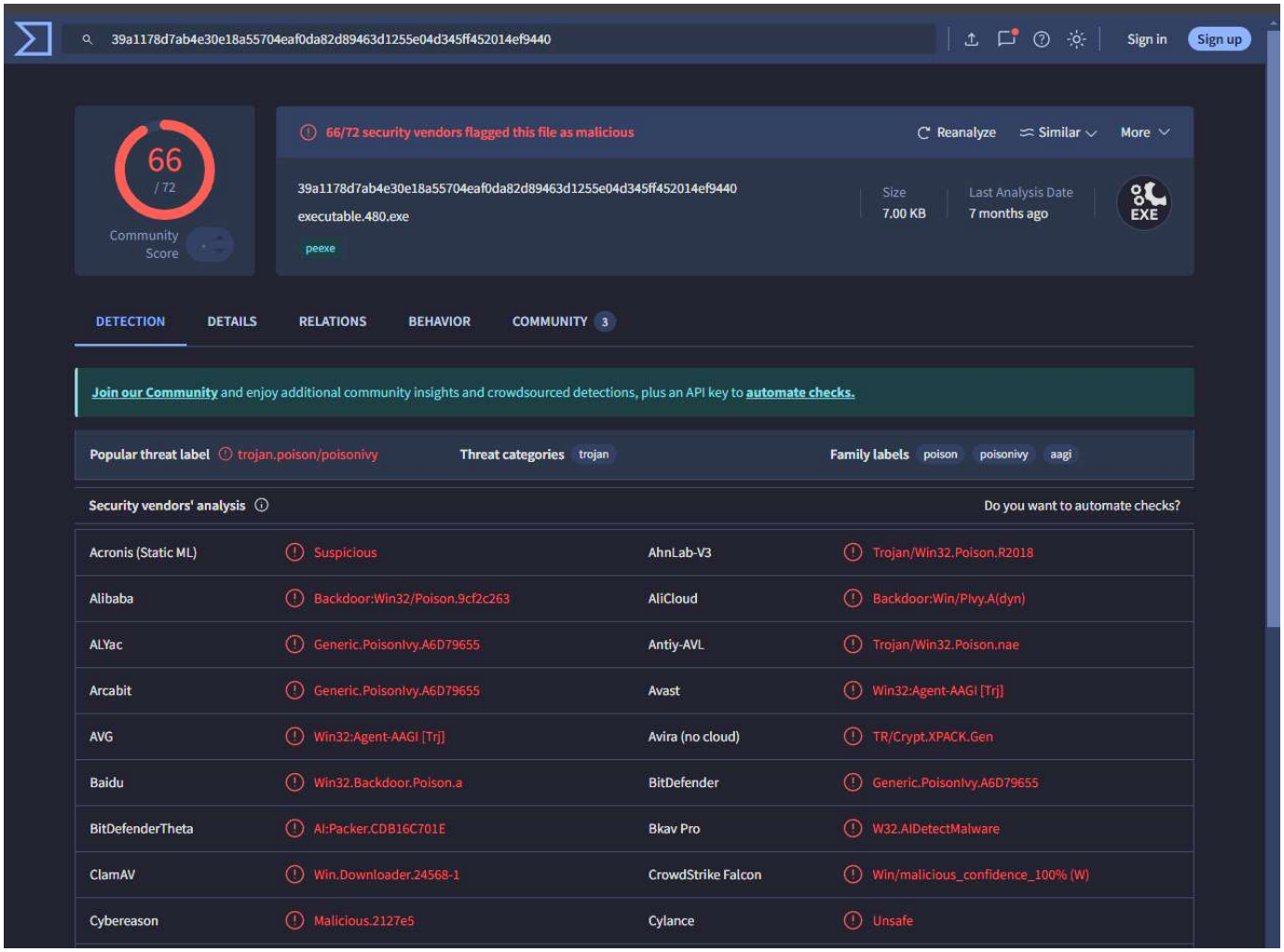
Malicious & Suspicious Processes with Cmdline scan

Next, I ran the procdump command on all the suspicious processes identified as malicious (PID 480, 404, 516, 532, 548, 560, and 524) to pull the files into my host and find more information. I then ran a command prompt script to create SHA1 hashes for each pulled executable.

```
C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.404.exe" SHA1
SHA1 hash of executable.404.exe:
df9e01d4ed83d58cc33a0ed74da43bf2f21c650f
CertUtil: -hashfile command completed successfully.

C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.480.exe" SHA1
SHA1 hash of executable.480.exe:
159f734027c96a9d913de6daec25c37bb7a5fc4a
CertUtil: -hashfile command completed successfully.

C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.516.exe" SHA1
SHA1 hash of executable.516.exe:
b0f5f6c88ca0b7159403369a878128140678338a
CertUtil: -hashfile command completed successfully.

C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.524.exe" SHA1
SHA1 hash of executable.524.exe:
fad162c0abfc5ca8d3095f00c5eac4fcf838a4a8
CertUtil: -hashfile command completed successfully.

C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.532.exe" SHA1
SHA1 hash of executable.532.exe:
4f9aa6e6470e3ba2a95b847f6f9f715c9f668fda
CertUtil: -hashfile command completed successfully.

C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.548.exe" SHA1
SHA1 hash of executable.548.exe:
c90b93ac159113985dd18c25e87ec75ddd772a81
CertUtil: -hashfile command completed successfully.

C:\Users\Administrator\Desktop\Lab03-DMS>certutil -hashfile "executable.560.exe" SHA1
SHA1 hash of executable.560.exe:
85a1dce7338a8d819ccdde1cc2339c4db9c5c756
CertUtil: -hashfile command completed successfully.
```

SHA1 Script on Pulled Procdump Executables

One by one, I searched the hashes on VirusTotal and found more information on the malware. For example, VirusTotal scored the explorer.exe hash as 28/72, with vendors such as Avast, Google, Fortinet, and Microsoft labeling the file as malicious.



VT Score for Poisonivy.exe

VirusTotal rating for each hash:

True file name	Extracted exe name	SHA-1 Hash	VT Score	VT Findings
explorer.exe	executable.404.exe	df9e01d4ed83d58cc33a0ed74da43bf2f21c650f	28/72	RiskWare:Win32/ATRAPS.16bbe37d
poisonivy.exe	executable.480.exe	159f734027c96a9d913de6daec25c37bb7a5fc4a	66/72	Backdoor:Win32/Poison.9cf2c263
soffice.exe	executable.516.exe	b0f5f6c88ca0b7159403369a878128140678338a	0/72	SOFFICE.EXE
soffice.bin	executable.524.exe	fad162c0abfc5ca8d3095f00c5eac4fcf838a4a8	0/71	Undetected
nc.exe	executable.532.exe	4f9aa6e6470e3ba2a95b847f6f9f715c9f668fda	39/72	RiskWare[RemoteAdmin]/Win32.NetCat.a
winvnc4.exe	executable.548.exe	c90b93ac159113985dd18c25e87ec75ddd772a81	16/71	RiskWare:Win32/WinVNC.2a6dc477
cmd.exe	executable.560.exe	85a1dce7338a8d819ccdde1cc2339c4db9c5c756	2/71	W32.AIDetectMalware

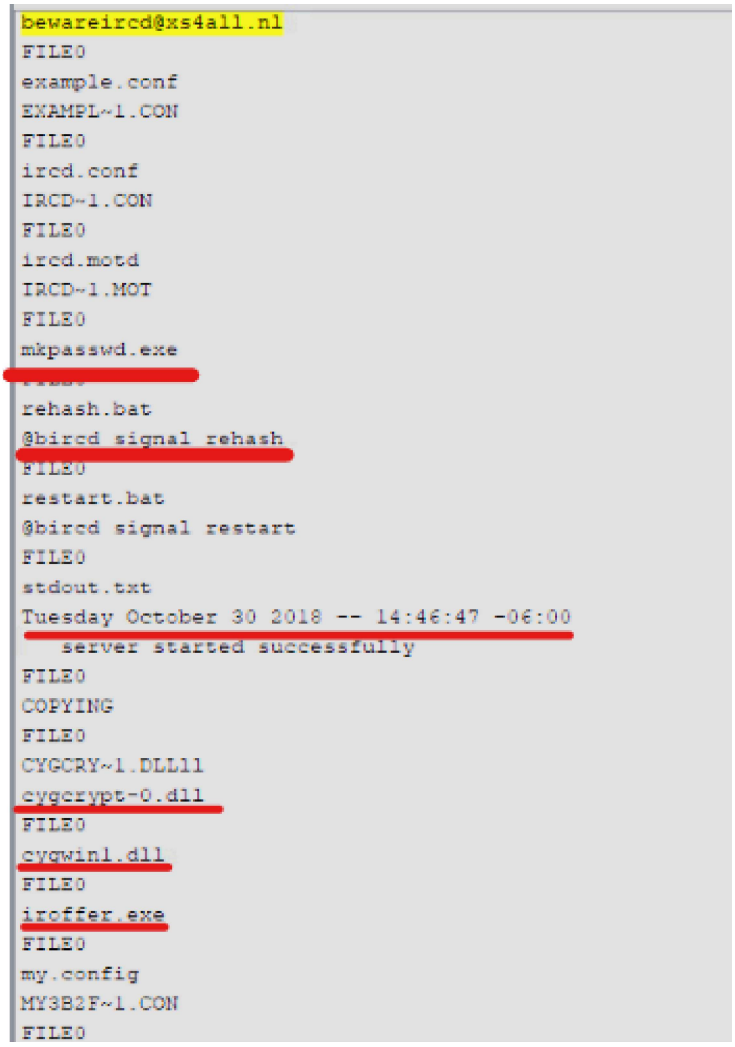
Program, SHA1, and VT Rating

Next, I attempted to look for cached users and/or their passwords within the image using sessions, userassist, truecryptpassphrase, and hashdump commands, but none provided the necessary information. Therefore, I pivoted to other applications to find the system users. One of the applications I tried was Autopsy.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 hashdump
Volatility Foundation Volatility Framework 2.6
ERROR   : volatility.debug    : Unable to read hashes from registry
```

Volatility Hashdump error

Using Autopsy, I also examined emails found within the memory image. A few of the emails stood out as they were named after processes and files identified by Volatility. For example, bewareircd@xs4all.nl and jqs@sun.com, which match folder bewareircd-win32 and process 1464, jqs.exe. I analyzed the contents of these emails. The bewareircd email contained several files identified in my analysis, such as iroffer.exe, cygwin1.dll, cygcrypt-0.dll, and bircd. It also mentioned makepasswd.exe, a cygwin program used to create password entries similar to Linux's /etc/passwd.



Autopsy Email Search: Bewareircd Email

I analyzed the passwd file in Volatility and found that it was located within the same “hidden” folder as other malicious tools used by the attacker, such as bircd and other .bat files located in the same “C:\hidden” folder.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 mftparser --dump-dir=. | more | findstr /i "bewareircd-
win32"
Volatility Foundation Volatility Framework 2.6
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\BIRCD-~1.
INI
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\bircd-qne
t.ini
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\bircd.exe

2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\bircd.ini

2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\bircd.txt

2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\example.c
onf
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\EXAMPL~1.
CON
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\ircd.conf

2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\IRCD~1.CO
N
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\ircd.motd

2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\IRCD~1.MO
T
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\mkpasswd.
exe
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\rehash.ba
t
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\restart.b
at
2010-05-25 23:55:46 UTC+0000 2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   2010-05-25 23:55:46 UTC+0000   hidden\bewareircd-win32\stdout.tx
t
^C^C
```

Volatility mftparser: hidden\bewareircd-win32\mkpasswd

Next, I returned to Autopsy to review jqs email. Analyzing the email, I found that it was detailing the process of initiating Java. This behavior is explained by jqs.exe’s functionality, as it is a process associated with the Java Quick Starter Component of the Java platform.

Next, I used the evtlogs command to view the system’s event logs.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 evtlogs --dump-dir=.
Volatility Foundation Volatility Framework 2.6
Parsed data sent to appevent.txt
Parsed data sent to sysevent.txt
Parsed data sent to secevent.txt
```

Volatility evtlogs dump

One thing I found through the event logs was the primary user of the device. The device is owned by Daniel Faraday, which I determined by viewing the event logs. The device’s name is “Faraday,” and the user logged into the device is also named “FARADAY.”

```
2010-04-30 01:26:09 UTC+0000|sysevent.evt|FARADAY|N/A|EventLog|6011|Info|MACHINENAME;FARADAY
```

Evtlogs: User and Machine Name

I conducted a “rootkit” string search on Autopsy and found information on the HackerDefender rootkit affecting the system. Underlined in purple, you can see that the rootkit utilizes the hidden folder. Highlighted in green, you can view the hidden processes used by the rootkit: hxdef, cryptcat.exe,

bircd.exe, and iroffer.exe. Highlighted in yellow, you can see that the rootkit's service name is "HackerDefender100."

```
C:\hidden\bewareircd-win32\bircd.exe
C:\hidden\ir\iroffer.exe?-b C:\hidden\ir\my.config
[Hidden Table]
hxdef*
hidden*
[Hidden Processes]
hxdef*
cryptcat.exe
bircd.exe
iroffer.exe
[Root Processes]
hxdef*
[Hidden Services]
HackerDefender*

[Hidden RegKeys]
HackerDefender100
LEGACY_HACKERDEFENDER100
HackerDefenderDrv100
LEGACY_HACKERDEFENDERDRV100

[Hidden RegValues]

[Free Space]
[Hidden Ports]
TCPI:23,31337,6667
TCPO:23,31337,6667
UDP:69
[Settings]
Password=
BackdoorShell=hxdef
?.exe
FileMappingName=_.-=[Hacker Defender]=-._
ServiceName=HackerDefender100
ServiceDisplayName=HXD Service 100
ServiceDescription=powerful NT rootkit
DriverName=HackerDefenderDrv100
DriverFileName=hxdefdrv.sys

[Comments]
================================[ ENGLISH INI HELP ]=======================
this is nt rootkit inifile
it must contains three file lists: [Hidden Table], [Hidden Processes]
and [Root Processes],
```

Autopsy "rootkit" search: HackerDefender100

The HackerDefender rootkit is consistent with the rootkit observed in the previously mentioned 30 infected processes. Additionally, the rootkit programs are located within the previously flagged "hidden" folder, where riskware and obfuscation programs like lock.bat are located.

Analyzing the system event logs, I found that the rootkit service "HXD Service 100" was first logged on the system on May 25, 2010, at 23:59:22.

```
2010-05-25 23:27:48 UTC+0000|sysevent.evt|FARADAY|N/A|Service Control Manager|7036|Info|FTP Publishing;running
2010-05-25 23:59:22 UTC+0000|sysevent.evt|FARADAY|S-1-5-18 (Local System)|Service Control Manager|7035|Info|HXD Service 100;start
2010-05-25 23:59:22 UTC+0000|sysevent.evt|FARADAY|N/A|Service Control Manager|7036|Info|HXD Service 100;running
2010-05-26 00:04:11 UTC+0000|sysevent.evt|FARADAY|S-1-5-18 (Local System)|USER32|1074|Info|winlogon.exe;FARADAY;No title for this reason could be
```

Evtlogs: Rootkit "HXD Service 100" Running

Next, I ran a connscan search to determine what addresses the device was connected to at the time of the memory capture. Through this, I determined that Faraday's device had established a remote connection with internal IP 192.168.5.98 on port 3460.

```
C:\Users\Administrator\Desktop\Lab03-DMS>volatility -f "KobayashiMaru 1.vmem" --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x01e76368 127.0.0.1:1031           127.0.0.1:6667           1728
0x021935e8 127.0.0.1:6667           127.0.0.1:1031           1480
0x021fd550 0.0.0.0:1037             192.168.5.98:3460        480
```

Volatility Connscan

I then ran a string search on Autopsy for the remote IP and found the following:

```
uSoftware\Microsoft\Windows\CurrentVersion\HomeNetworking\PersonalFirewall
ShowDisableFirewallWarning
Software\Microsoft\EAPOL\Parameters\General
InterfaceList
W2CSVC
ms_tcpip
SYSTEM\Setup\AnswerFileMap
u
.?AVexception@@<
.?AVbad_alloc@std@@<
'4Hh
€I*h<8
^-m-m<|<|<|M
o/o/
advapi32
ntdll
user32
ql+KY
yx{w
}#%li
}>*K
QQVP
Pjjjjjj
advpack
StubPath
(SOFTWARE\Classes\http\shell\open\commandV
5Software\Microsoft\Active Setup\Installed Components\
Poison Ivy
192.168.5.98
admin
Poison Ivy
          )!VoqA.I4-
poisonivy.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\RunW
u          jV
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Foldersh
PWjj
AppData
```

Autopsy "192.168.5.98" Search

The remote connection was established through the following steps:

First, the device's firewall warning was disabled to prevent the user from seeing firewall notifications. Next, an HTTP shell was created to invoke poisonivy.exe. Finally, poisonivy.exe established a remote connection to 192.168.5.98, granting the remote user admin privileges to Faraday's device. Therefore, the attacker on Faraday's device originates from internal IP 192.168.5.98 and used the poisonivy backdoor to gain access.

InvestigationConclusion:

Daniel Faraday's device was compromised by a skilled threat actor originating from internal IP address 192.168.5.98, who gained unauthorized access through the deployment of the remote access tool Poison Ivy. This tool allowed the attacker to establish a backdoor into Faraday's system, granting administrative privileges and remote control over the device. The attacker exploited Poison Ivy to bypass the device's defenses and maintain a covert, persistent connection with Faraday's system.

Once inside, the attacker installed a sophisticated rootkit, "Hacker Defender," known for its ability to hide itself and malicious files from detection. This rootkit included tools such as Cryptcat—a secured, encrypted version of Netcat—which allowed for stealthy communication between the attacker and Faraday's device. The attacker embedded the rootkit and various malicious applications in a folder named "hidden", aiming to keep a low profile while performing malicious activities.

To maintain persistence, the attacker spread the backdoor functionality across multiple system processes, infecting 30 of them. Each of these processes showed signs of infection when analyzed with the malfind command, specifically indicating the presence of the "Hacker Defender" rootkit. The rootkit modified system API functions and disguised itself to prevent detection by typical security tools. The infected processes included common Windows system files such as smss.exe, csrss.exe, winlogon.exe, and svchost.exe, effectively embedding the rootkit in essential system services and making its removal or detection more challenging.

"HXD Service 100," the system service running Hacker Defender, was first observed on May 25, 2010, at 23:59:22, marking the beginning of the rootkit's influence on Faraday's device. This service enabled the rootkit to operate with system-level privileges, granting the attacker extensive control over the device. By embedding itself in critical system processes and operating as a legitimate-looking service, the rootkit enabled the attacker to establish ongoing, undetected access to the compromised device. Through this setup, the attacker had full control over Faraday's computer, allowing them to manipulate files, execute commands, and monitor system activities remotely without raising suspicions.

To further obfuscate their activities and cover their tracks, the attacker utilized a script named lock.bat, located in the hidden folder alongside the other malicious tools. This batch file was designed to alter file attributes, effectively concealing specific files and directories from regular user view. By leveraging lock.bat, the attacker was able to hide components of their malware, including backdoors, command-and-control tools, and other utilities essential for maintaining access and controlling Faraday's device.

The “hidden” folder served as a centralized location for these malicious tools, which included not only the rootkit and lock.bat but also files like cryptcat.exe and iroffer.exe, all chosen for their ability to facilitate remote control, covert communication, and persistence. The attacker specifically stored these files in a location outside of typical system paths to avoid detection. This placement within a hidden directory, combined with lock.bat's file-hiding functionality, allowed the attacker to further obscure the presence of their malware on the system.

Citations

Hashnode. "An Introduction to Volatility 3." Hashnode, https://cpuu.hashnode.dev/an-introduction-to-volatility-3. Accessed 11 Nov. 2024.

ProcessLibrary. "iroffer." ProcessLibrary, https://www.processlibrary.com/en/directory/files/iroffer/23813/. Accessed 11 Nov. 2024.

MITRE ATT&CK. "Iroffer - S0012." MITRE ATT&CK Framework, https://attack.mitre.org/software/S0012/. Accessed 11 Nov. 2024.

ScienceDirect. "Cryptcat." ScienceDirect, https://www.sciencedirect.com/topics/computer-science/cryptcat. Accessed 11 Nov. 2024.

F-Secure. "Rootkit: W32/Hacdef." F-Secure, https://www.f-secure.com/v-descs/rootkit-w32-hacdef.shtml. Accessed 11 Nov. 2024.

Tenable. "Nessus Plugin ID 15517." Tenable, https://www.tenable.com/plugins/nessus/15517. Accessed 11 Nov. 2024.

Neuber. "jqs.exe." Neuber Task Manager, https://www.neuber.com/taskmanager/process/jqs.exe.html. Accessed 11 Nov. 2024.

Radware. "IRC - Internet Relay Chat." DDoSPedia by Radware, https://www.radware.com/security/ddos-knowledge-center/ddospedia/irc-internet-relay-chat /. Accessed 11 Nov. 2024.

Instructables. "Lockbat: Hide Your Files." Instructables, https://www.instructables.com/Lockbat-Hide-your-files/. Accessed 11 Nov. 2024.