MrRobot Lab (Cyber Defenders) - Walkthrough

Sunday, September 8, 2024 11:12 PM

Story:
An employee reported that his machine started to act strangely after receiving a suspicious email for a security update.
The incident response team captured a couple of memory dumps from the suspected machines for further inspection.
Analyze the dumps and help the SOC analysts team figure out what happened!

- We received three memory dump files, all with the same profile.
Volatility3 caused several issues during the analysis, so I switched to using Volatility 2 instead.

POS1 - **python2 vol.py -f ../pos01/POS-01-c4e8f786.vmss --profile=Win7SP1x86_23418**
Target1 - **python2 vol.py -f ../target1/Target1-1dd8701f.vmss --profile=Win7SP1x86_23418**
Target2 - **python2 vol.py -f ../target2/target2-6186fe9f.vmss --profile=Win7SP1x86_23418**

Q1: Machine:Target1 What email address tricked the front desk employee into installing a security update?

- At the beginning of the challenge, I extracted the 'Process Tree' for all machines and noticed that 'OUTLOOK.exe' was running on both the 'Target1' and 'POS1' machines.

  I focused my investigation on the 'Target1' machine due to significant 'Cmd.exe' activity, which had executed 'mstsc.exe'—a command-line interface used to launch the Remote Desktop client, along with 'Outlook.exe'.

  To dig deeper, I dumped the memory of the 'Outlook.exe' process using the 'memdump' plugin.
  After running 'strings' on the dump and grepping for 'From:', I successfully found the answer!

```
┌──(kali㉿kali)-[~/Desktop/volatility/Outlook.exe]
└─$ strings 3196.dmp| grep -i 'From:'
        h=mime-version:date:message-id:subject:from:to:content-type;
From: The Whit3R0s3 <th3wh1t3r0s3@gmail.com>
FROM:
From:
```

Q2: Machine:Target1 What is the filename that was delivered in the email?

- This question was quite challenging.
  I used Bulk Extractor to parse the memory file, which also included the 'Prefetch' data showing executions on the host.
  Since the prefetch file isn't sorted by timestamp, I found several reconnaissance commands and identified a suspicious process that was executed after Outlook.

```
718397440       ANYCONNECTINSTALLER.EXE <prefetch><os>Windows Vista or Windows 7</os><filename>ANYCONNECTINSTALLER.EXE</filename><header_size>240</header_size><atime>2015-10-09T11:25:06Z</atime><runs>16</runs><
filenames></filenames><volume><path></path><creation>1601-01-01T00:00:00Z</creation><serial_number>0</serial_number><dirnames></dirnames></volume></prefetch>
718430208       ANYCONNECTINSTALLER.EXE <prefetch><os>Windows Vista or Windows 7</os><filename>ANYCONNECTINSTALLER.EXE</filename><header_size>240</header_size><atime>2015-10-09T11:25:06Z</atime><runs>14</runs><
filenames></filenames><volume><path></path><creation>1601-01-01T00:00:00Z</creation><serial_number>0</serial_number><dirnames></dirnames></volume></prefetch>
```

- Note: You are able to see the executable in the Downloads folder

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../target1/Target1-1dd8701f.vmss --profile=Win7SP1x86_23418 filescan | grep 'Downloads'
Volatility Foundation Volatility Framework 2.6.1
0x000000003dec0800      7      0 R--rwd \Device\HarddiskVolume2\Users\frontdesk\Downloads\desktop.ini
0x000000003ded0910      8      0 R--rwd \Device\HarddiskVolume2\Users\frontdesk\Links\Downloads.lnk
0x000000003df2fc38      3      1 R--rw- \Device\HarddiskVolume2\Users\frontdesk\Downloads
0x000000003e0bc5e0      7      0 R--r-d \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
0x000000003e1eef60      8      0 RW---- \Device\HarddiskVolume2\Users\frontdesk\Downloads\wce.exe
0x000000003e259040      8      0 R--rwd \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
0x000000003ed57968      4      0 R--r-d \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
```

Q3: Machine:Target1 What is the name of the rat's family used by the attacker?

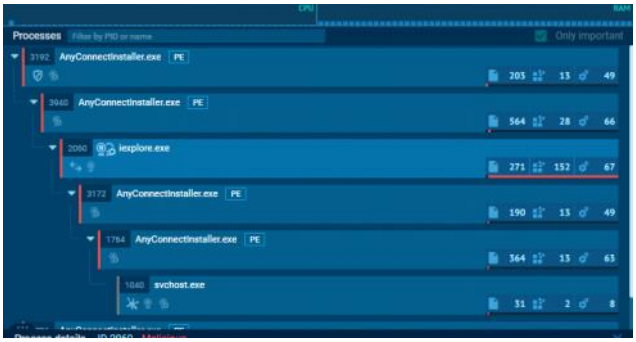
- In the previous investigation, we discovered that the attacker had delivered an executable named 'AnyConnectInstaller.exe'.

  I used the 'Dumpfiles' plugin to extract the process associated with it.
  After obtaining the process, I computed the MD5 hash and checked it on VirusTotal.
  The results indicated that the file is widely reported and associated with 'XtremeRAT'.

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../target1/Target1-1dd8701f.vmss --profile=Win7SP1x86_23418 dumpfiles -Q 0x000000003e0bc5e0 -D .
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
ImageSectionObject 0x3e0bc5e0   None    \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
DataSectionObject 0x3e0bc5e0    None    \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe

┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ md5sum file.None.0x85cd09a0.img
165a952830d0d915989c48a3275edc379  file.None.0x85cd09a0.img

┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ md5sum file.None.0x858aef78.dat
23a9329505c6eb1684b901524ca7bdc9  file.None.0x858aef78.dat
```

Q4: Machine:Target1 The malware appears to be leveraging process injection. What is the PID of the process that is injected?

- To address this question, I performed a dynamic analysis of the malware using 'AnyRun'.
  During the analysis, I identified that the malware injected into the 'Iexplore.exe' process, associated with Internet Explorer, which has a PID of 2996.




```
0×8561d030:winlogon.exe                480    412      3    115 2015-10-09 11:30:48 UTC+0000
0×85d0d030:iexplore.exe               2996   2984      6    463 2015-10-09 11:31:27 UTC+0000
. 0×83f105f0:cmd.exe                  1856   2996      1     33 2015-10-09 11:35:15 UTC+0000
 0×83fb2d40:cmd.exe                   3784   2196      1     24 2015-10-09 11:39:22 UTC+0000
```

Q5: Machine:Target1 What is the unique value the malware is using to maintain persistence after reboot?

- To address this question, I consulted the 'VirusTotal' report and navigated to the 'Registry Keys Set' section.
  There, I searched for the 'CurrentVersion/Run' key, which indicates startup entries for programs.
  I found that the value '**MrRobot**' was present.

Registry Keys Set

+ HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\MrRobot
+ HKEY_CURRENT_USER\SOFTWARE\Xtreme\InstalledServer
+ HKEY_CURRENT_USER\SOFTWARE\Xtreme\PersistInstalledServer
+ HKEY_CURRENT_USER\SOFTWARE\Xtreme\PersistSettings
+ HKEY_CURRENT_USER\SOFTWARE\Xtreme\Settings
+ HKEY_CURRENT_USER\SOFTWARE\fhack\FirstExecution
+ HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run\MrRobot

Q6: Machine:Target1 Malware often uses a unique value or name to ensure that only one copy runs on the system. What is the unique name the malware is using?

- The question requires identifying the "unique name" that the malware uses to ensure only one instance runs on the system.
  This unique identifier could be a mutex, a registry key, or a specific file name.
  As before, I reviewed the 'VirusTotal' report and discovered that the malware creates a mutex named '**fsociety0.dat**'.

Mutexes Created

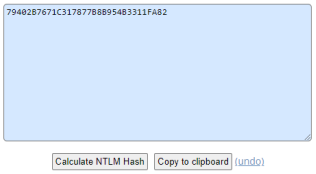
- XTREMEPASSWORDS
- XTREMESERVER
- XTREMESERVERKEYLOGGER
- XTREMESERVERPERSIST
- fsociety0.dat
- fsociety0.dat_PERSIST
- CTF.Asm.MutexDefaultS-1-5-21-1482476501-1645522239-1417001333-500
- CTF.Compart.MutexDefaultS-1-5-21-1482476501-1645522239-1417001333-500
- CTF.LBES.MutexDefaultS-1-5-21-1482476501-1645522239-1417001333-500
- CTF.Layouts.MutexDefaultS-1-5-21-1482476501-1645522239-1417001333-500

Q7: Machine:Target1 It appears that a notorious hacker compromised this box before our current attackers. Name the movie he or she is from.

- While navigating the file system using R-Studio, I identified a user named 'Zerocool', which is a character from the movie ***Hackers***.

Q8: Machine:Target1 What is the NTLM password hash for the administrator account?

- I used the 'Consoles' plugin to observe the threat actor's operations, noting that they utilized 'wsc.exe', a password dumper. I
  discovered that the threat actor extracted the 'Administrator' password in plaintext as 'flagadmin@1234' using this tool.
  I then used the website https://www.browserling.com/tools/ntlm-hash to convert this password into its NTLM hash.




```
C:\Windows\Temp>wce.exe -w
WCE v1.42beta (Windows Credentials Editor) - (c) 2010-2013 Amplia Security - by
Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.


Administrator\front-desk-PC:flagadmin@1234
frontdesk\ALLSAFECYBERSEC:THzV7mpz
FRONT-DESK-PC$\ALLSAFECYBERSEC:o0077qj:^zctL2T]ljn3<niK2Kbqi`(:LeBoO7zE>'d8<>J"P
K;\*5IS@Oxg:rC:P:z Y!%fUiIX0y_J& uNUTJ?%:Y;qJY,xq/:)%5^f&zDK.)F%H;V?.^Z

C:\Windows\Temp>wce.exe -w > w.tmp
```

Q9: Machine:Target1 The attackers appear to have moved over some tools to the compromised front desk host. How many tools did the attacker move?

- We observed that the attacker tampered with the 'Temp' directory, where we identified several tools they brought

onto the compromised system, including 'Rar.exe', 'nbtscan.exe', 'wce.exe', and 'getlsasrvaddr.exe'.
Initially, I believed the answer was '4', but it is actually '3' because some of these files may be part of the same package.

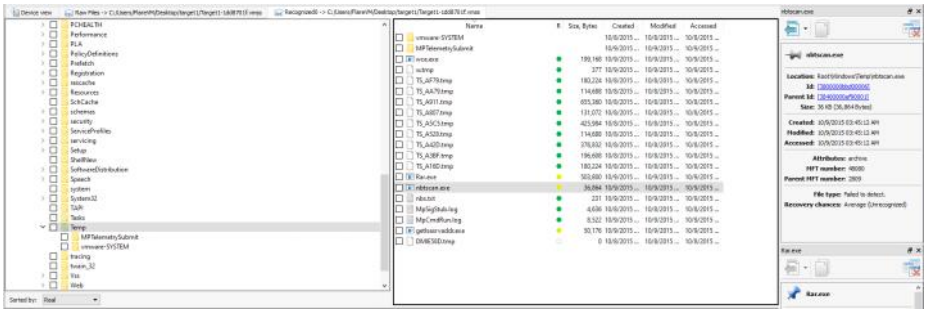
```
C:\Windows\Temp>dir
 Volume in drive C has no label.
 Volume Serial Number is FE0F-F423

 Directory of C:\Windows\Temp

10/09/2015  07:29 AM    <DIR>          .
10/09/2015  07:29 AM    <DIR>          ..
10/09/2015  01:27 AM                 0 DMIES8D.tmp
10/09/2015  06:57 AM            50,176 getlsasrvaddr.exe
10/09/2015  02:02 AM             7,572 MpCmdRun.log
10/09/2015  12:07 AM             4,630 MpSigStub.log
10/09/2015  03:37 AM    <DIR>          MPTelemetrySubmit
10/09/2015  06:45 AM            36,864 nbtscan.exe
10/09/2015  06:44 AM           503,800 Rar.exe
10/09/2015  01:28 AM           180,224 TS_A16D.tmp
10/09/2015  01:28 AM           196,608 TS_A3BF.tmp
10/09/2015  01:28 AM           376,832 TS_A42D.tmp
10/09/2015  01:28 AM           114,688 TS_A528.tmp
10/09/2015  01:28 AM           425,984 TS_A5C5.tmp
10/09/2015  01:28 AM           131,072 TS_A807.tmp
10/09/2015  01:28 AM           655,360 TS_A911.tmp
10/09/2015  01:28 AM           114,688 TS_AA79.tmp
10/09/2015  01:28 AM           180,224 TS_AF79.tmp
10/08/2015  11:43 PM    <DIR>          vmware-SYSTEM
10/09/2015  07:34 AM               333 w.tmp
10/09/2015  06:45 AM           199,168 wce.exe
              17 File(s)      3,178,229 bytes
               4 Dir(s)  22,603,194,368 bytes free
```

Q10: Machine:Target1 What is the password for the front desk local administrator account?

- We already found it in the previous question, **flagadmin@1234**

Q11: Machine:Target1 What is the std create data timestamp for the nbtscan.exe tool?

To address this question, I initially used 'R-Studio' to navigate to the 'Temp' directory, where I located the mentioned file and viewed the creation time as '2015-09-10 03:45:12'.
However, this was the wrong answer. Suspecting that the timestamp was not in UTC, I parsed the MFT using the 'mftparser' plugin in Volatility and found the correct answer.



```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../target1/Target1-1dd8701f.vmss --profile=Win7SP1x86_23418 mftparser | grep -i 'nbtscan.exe'
Volatility Foundation Volatility Framework 2.6.1
2015-10-09 10:45:12 UTC+0000 2015-10-09 10:45:12 UTC+0000   2015-10-09 10:45:12 UTC+0000   2015-10-09 10:45:12 UTC+0000   Windows\Temp\nbtscan.exe
2015-10-09 10:47:07 UTC+0000 2015-10-09 10:47:07 UTC+0000   2015-10-09 10:47:07 UTC+0000   2015-10-09 10:47:07 UTC+0000   Windows\Prefetch\NBTSCAN.EXE-44BD0B89.pf
```

Q12: Machine:Target1 The attackers appear to have stored the output from the nbtscan.exe tool in a text file on a disk called nbs.txt. What is the IP address of the first machine in that file?

- I navigate to the 'Temp' directory and opened the file via 'R-Studio' to address this question.

R-Viewer - Recognized0\Windows\Temp\nbs.txt

File Actions View Help

```
10.1.1.2     ALLSAFECYBERSEC\AD01       SHARING DC
10.1.1.3     ALLSAFECYBERSEC\EX01       SHARING
10.1.1.20    ALLSAFECYBERSEC\FRONT-DESK-PC  SHARING
10.1.1.21    ALLSAFECYBERSEC\GIDEON-PC      SHARING
```

Q13: Machine:Target1 What is the full IP address and the port was the attacker's malware using?

- To address this question, I used the 'netscan' plugin and found that the injected process (iExplorer.exe) had an active connection with the IP address '180.76.254.120' over port 22.

```
0×3de98df8      TCPv4    10.1.1.20:49261      10.1.1.21:445        ESTABLISHED     4       System
0×3e0d0df8      TCPv4    10.1.1.20:49208      10.1.1.3:80          ESTABLISHED     3196    OUTLOOK.EXE
0×3e0eedf8      TCPv4    10.1.1.20:49205      180.76.254.120:22    ESTABLISHED     2996    iexplore.exe
0×3e1e5008      TCPv4    10.1.1.20:49330      10.1.1.2:139         CLOSED          4       System
0×3e1f0df8      TCPv4    10.1.1.20:49207      10.1.1.3:80          ESTABLISHED     3196    OUTLOOK.EXE
```

Q14: Machine:Target1 It appears the attacker also installed legit remote administration software. What is the name of the running process?

- At the beginning of the challenge, when we used 'pstree' plugin I noticed '**TeamViewer.exe**' is also running on the compromised host.

```
. 0x84013598:TeamViewer.exe                    2680   1696   28    632 2015-10-09 12:08:46 UTC+0000
. 0x858bc278:TeamViewer_Des                    1092   2680   16    405 2015-10-09 12:10:56 UTC+0000
. 0x84017d40:tv_w32.exe                        4064   2680    2     83 2015-10-09 12:08:47 UTC+0000
 0x85c1e5f8:explorer.exe                       2116   2060   23    912 2015-10-09 11:31:04 UTC+0000
. 0x83eb5d40:cmd.exe                           2496   2116    1     22 2015-10-09 11:33:42 UTC+0000
. 0x83f1ed40:mstsc.exe                         2844   2116   11    484 2015-10-09 12:12:03 UTC+0000
. 0x83fb86a8:cmd.exe                           3064   2116    1     22 2015-10-09 11:37:32 UTC+0000
. 0x859281f0:vmtoolsd.exe                      2388   2116    7    164 2015-10-09 11:31:04 UTC+0000
. 0x85cd3d40:OUTLOOK.EXE                       3196   2116   22   1678 2015-10-09 11:31:32 UTC+0000
 0x855f6d40:csrss.exe                           432    412   11    366 2015-10-09 11:30:48 UTC+0000
. 0x83f13d40:conhost.exe                       1624    432    3     81 2015-10-09 11:35:15 UTC+0000
. 0x83fa9030:conhost.exe                        676    432    3     83 2015-10-09 11:37:32 UTC+0000
. 0x83e5cd40:conhost.exe                        916    432    3     83 2015-10-09 11:33:42 UTC+0000
. 0x83fc7c08:conhost.exe                       1824    432    3     85 2015-10-09 11:39:22 UTC+0000
 0x8561d030:winlogon.exe                       480    412    3    115 2015-10-09 11:30:48 UTC+0000
 0x85d0d030:iexplore.exe                      2936   2984    6    463 2015-10-09 11:31:27 UTC+0000
. 0x83f105f0:cmd.exe                          1856   2996    1     33 2015-10-09 11:35:15 UTC+0000
 0x83fb2d40:cmd.exe                           3784   2196    1     24 2015-10-09 11:39:22 UTC+0000
```

Q15: Machine:Target1 It appears the attackers also used a built-in remote access method. What IP address did they connect to?

- At the beginning of the challenge, we also noticed that the 'mstdc.exe' process was running, which is related to RDP.
I used the 'netscan' plugin again and identified an internal RDP connection.

```
0×3f505388        UDPv6    ::1:1900                *:*                                 3232    svchost.exe     2015-10-09 11:32:55 UTC+0000
0×3fa082e0        UDPv4    0.0.0.0:59560           *:*                                 2680    TeamViewer.exe  2015-10-09 12:10:57 UTC+0000
0×3fa19180        TCPv4    127.0.0.1:6039          0.0.0.0:0           LISTENING       2680    TeamViewer.exe
0×3fa4dbf8        TCPv4    10.1.1.20:49333         10.1.1.3:443        CLOSED          3196    OUTLOOK.EXE
0×3fa72a80        TCPv4    127.0.0.1:6039          127.0.0.1:49298     ESTABLISHED     2680    TeamViewer.exe
0×3fa8d1d8        TCPv4    10.1.1.20:49336         10.1.1.3:443        CLOSED          3196    OUTLOOK.EXE
0×3fa95df8        TCPv4    10.1.1.20:49297         192.96.201.138:5938 ESTABLISHED     2680    TeamViewer.exe
0×3fb7a560        TCPv4    10.1.1.20:49301         10.1.1.21:3389      ESTABLISHED     2844    mstsc.exe
0×3fc81008        UDPv4    127.0.0.1:1900          *:*                                 3232    svchost.exe     2015-10-09 11:32:55 UTC+0000
0×3fc81480        UDPv6    ::1:56812               *:*                                 3232    svchost.exe     2015-10-09 11:32:55 UTC+0000
0×3fc426a8        TCPv4    10.1.1.20:49291         107.6.97.19:5938    ESTABLISHED     2680    TeamViewer.exe
0×3fcdd8b0        TCPv4    127.0.0.1:49298         127.0.0.1:6039      ESTABLISHED     1092    TeamViewer_Des
```

Q16: Machine:Target2 It appears the attacker moved latterly from the front desk machine to the security admins (Gideon) machine and dumped the passwords. What is Gideon's password?

- Now we will analyze the 'Target2' machine after the lateral movement by the attacker.
  I used the 'consoles' plugin to examine the attacker's activity on the console and found that they used 'wce.exe' again, saving Gideon's password to a file named 'gideon/w.tmp'.
  I then loaded the memory dump into 'R-Studio', navigated to C:\Users\Gideon, and located the mentioned file.

Q17: Machine:Target2 Once the attacker gained access to "Gideon," they pivoted to the AllSafeCyberSec domain controller to steal files. It appears they were successful. What password did they use?

When I used the 'consoles' plugin, we observed the lateral movement to the DC via the 'net' utility.
The attacker executed the rar command to create an archive named crownjewlez.rar.
The *.txt specifies that the archive includes all .txt files in the current directory, and the -hp123qwe!@# option sets the password for the archive to **123qwe!@#.**

```
CommandHistory: 0×e9198 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 18 LastAdded: 17 LastDisplayed: 17
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0×60
Cmd #0 at 0×e6030: cd C:\Users
Cmd #1 at 0×e6ea8: dir
Cmd #2 at 0×ee3d0: wce.exe -w > gideon/w.tmp
Cmd #3 at 0×e0170: who ami
Cmd #4 at 0×e0188: whoami
Cmd #5 at 0×ea3c8: net use z: \\10.1.1.2\c$
Cmd #6 at 0×e01b8: cd z:
Cmd #7 at 0×e6ed8: dir
Cmd #8 at 0×e6070: cd gideon
Cmd #9 at 0×e6ef8: dir
Cmd #10 at 0×e6f08: z:
Cmd #11 at 0×e6f18: dir
Cmd #12 at 0×f2418: copy c:\users\gideon\rar.exe z:\crownjewels
Cmd #13 at 0×e0cb8: cd crownjewels
Cmd #14 at 0×e6f28: dir
Cmd #15 at 0×e6f38: rar
Cmd #16 at 0×f2478: rar crownjewlez.rar *.txt -hp123qwe!@#
Cmd #17 at 0×f24d0: rar a -hp123!@#qwe crownjewlez.rar *.txt
——
Screen 0×d0038 X:80 Y:300
Dump:
```

Q18: Machine:Target2 What was the name of the RAR file created by the attackers?

- We identified the answer to this question in the previous context: **crownjewlez.rar.**

Q19: Machine:Target2 How many files did the attacker add to the RAR archive?

I used the 'cmdscan' plugin and found commands related to 'conhost.exe' with PID 3048.
I dumped the memory of the file, executed strings, and filtered for .txt files since we knew the attacker targeted only text files.

I didn't find any relevant .txt file entries initially, so I referred to a web walkthrough and discovered that I needed to **use the -e l option to encode the strings into UTF-16.**

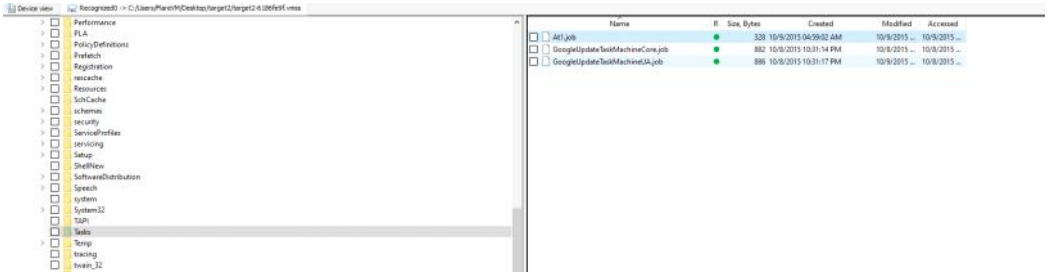
I found 3 secrets files

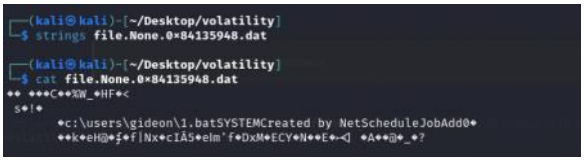
```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ strings -e l 3048.dmp | grep -E 'crownjewlez\.rar|\.txt'
```

```
crownjewlez.rar
SecretSauce1.txt
SecretSauce2.txt
SecretSauce3.txt
crownjewlez.rar
SecretSauce1.txt
SecretSauce2.txt
SecretSauce3.txt
```

Q20: Machine:Target2 The attacker appears to have created a scheduled task on Gideon's machine. What is the name of the file associated with the scheduled task?

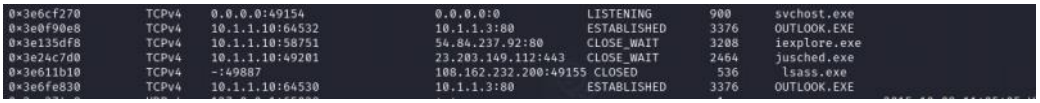
- To address this question, I used 'R-Studio' to navigate to C:\Windows\Tasks and located the suspicious task At1.job.
  I then used Volatility to dump the file and employed cat to view the task information.
  From this, I found that the related file is 1.bat.





Q21: Machine:POS What is the malware CNC's server?

- I used the 'netscan' plugin on the 'POS' machine and identified that the infected process (iexplorer.exe) had a network connection to the IP address '54.84.237.92' via port 80.



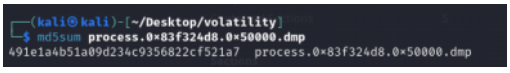
Q22: Machine:POS What is the common name of the malware used to infect the POS system?

- To be honest, I examined the C2 address '54.84.237.92' using VirusTotal and found that the communication with this malicious process is associated with the '**Dexter**' malware family.

Q23: **Machine:POS In the POS malware whitelist. What application was specific to Allsafecybersec?**

- We already know that the malicious process iexplore.exe has a PID of 3208.
  Using the malfind plugin, which helps detect hidden or injected code/DLLs in user-mode memory, I dumped the memory of the injected process from iexplore.exe with the following command:
  python2 vol.py -f ../pos01/POS-01-c4e8f786.vmss --profile=Win7SP1x86_23418 malfind --pid 3208 -D .

  After we dumped the injected process memory, I ran strings on the memory dump and found the application 'allsafe_protector.exe'



Q24: Machine:POS What is the name of the file the malware was initially launched from?

- To address this question, I used 'Bulk Extractor' to parse the entire memory dump.
  Upon analyzing the prefetch files, I identified a suspicious executable named **ALLsafe_update.exe.**