Challenge: MrRobot Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Medium

Tools Used: Volatility 2, Outlook Forensics Wizard, R-Studio, Strings, Notepad++, VirusTotal

Summary: This lab involved investigating a phishing-led compromise across multiple hosts within the environment. The tools used include Volatility 2, R-Studio, and Outlook Forensics Wizard. Analysis of the front-desk machine revealed that an employee was deceived by an email from th3wh1t3r0s3@gmail.com containing a fake Cisco AnyConnect installer (AnyConnectInstaller.exe), identified as XtremeRAT. The threat actor achieved persistence via a Run-key named MrRobot, performed process hollowing in iexplore.exe, and used a mutex to prevent duplicate infections. Subsequent lateral movement via RDP to the "Gideon" machine exposed password dumping with WCE, credential reuse, and data exfiltration of multiple text files. On the POS system, the adversary injected code into iexplore.exe, communicating with 54.84.237.92 over port 80, which was confirmed as Dexter POS malware. This was an enjoyable and challenging lab, I recommend giving it a go if you enjoy memory forensics.

Scenario: 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!

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

First, let's identify the profile and KDBG address for this disk image:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" imageinfo`
- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 kdbgscan`

The most common email client used on Windows is outlook. Outlook creates .OST files to store a local copy of mailbox data, such as emails, contacts, and calendar events. We can use the dumpfiles plugin to search for all .OST files and dump them to the given directory:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 dumpfiles -n -u -r ost$ -D "\dump_files\"`

```
   \Device\HarddiskVolume2\Users\frontdesk\AppData\Local\Microsoft\Outlook\Frontdesk@allsafecybersec.com - outlook2.ost
\Device\HarddiskVolume2\Users\frontdesk\AppData\Local\Microsoft\Outlook\Frontdesk@allsafecybersec.com - outlook2.ost
```

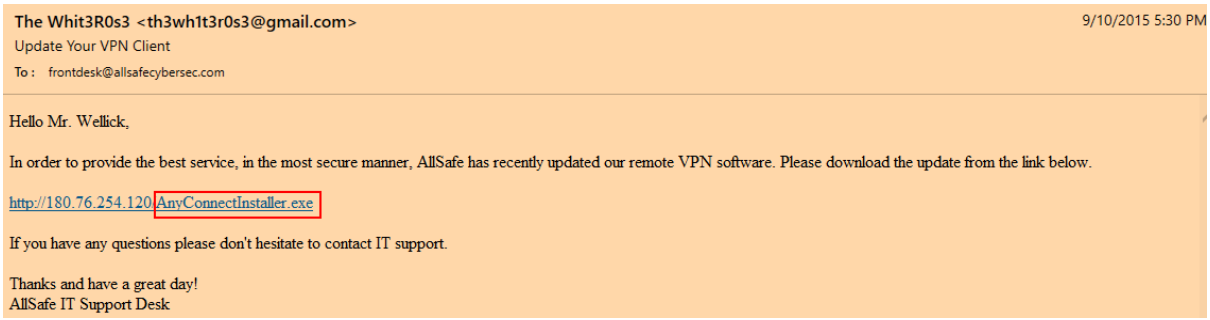
To analyse the “Frontdesk@allsafecybersec.com – outlook2.ost.dat” file, we can use an incredible tool called Outlook Forensics Wizard. Within this user’s inbox there is only one email from th3wh1t3r0s3@gmail.com which includes a link to a supposed VPN update:

Given that an employee reported suspicious activity after receiving an email for a security update, this is clearly the phishing email that compromised the victim.

Answer: th3wh1t3r0s3@gmail.com

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

In the email discovered earlier, we found a link to install a file called “AnyConnectInstaller.exe”:



This binary is clearly trying to impersonate the legitimate Cisco AnyConnect VPN client.

Answer: AnyConnectInstaller.exe

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

To start, we can use the filescan plugin to locate “AnyConnectInstaller.exe”:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 filescan | Select-String -Pattern "AnyConnectInstaller.exe"`

```
R--r-- \Device\HarddiskVolume2\Windows\Prefetch\ANYCONNECTINSTALLER.EXE-BF8040D4.pf
RW-rwd \Device\HarddiskVolume2\Users\anyconnect\AnyConnect\AnyConnectInstaller.exe
R--r-d \Device\HarddiskVolume2\Users\anyconnect\AnyConnect\AnyConnectInstaller.exe
R--r-d \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
R--rwd \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
RWD--- \Device\HarddiskVolume2\Users\anyconnect\AnyConnect\AnyConnectInstaller.exe
R--r-d \Device\HarddiskVolume2\Users\frontdesk\Downloads\AnyConnectInstaller.exe
R--r-- \Device\HarddiskVolume2\Windows\Prefetch\ANYCONNECTINSTALLER.EXE-F5AF5299.pf
```

We can then use the dumpfiles plugin and provide the physical offset for this binary:

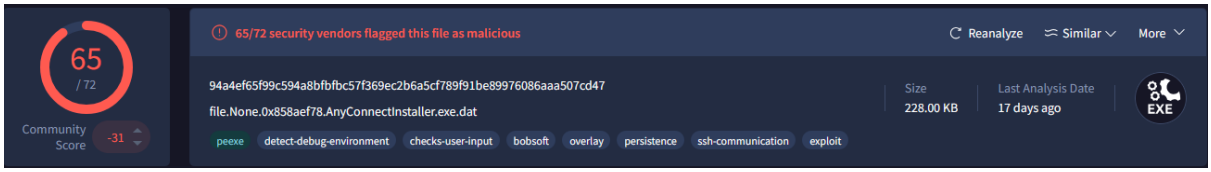
- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 dumpfiles -n -u -Q 0x000000003e0bc5e0 -D "\dump_files\"`

After navigating to the output directory, we can generate the SHA256 hash for this file:

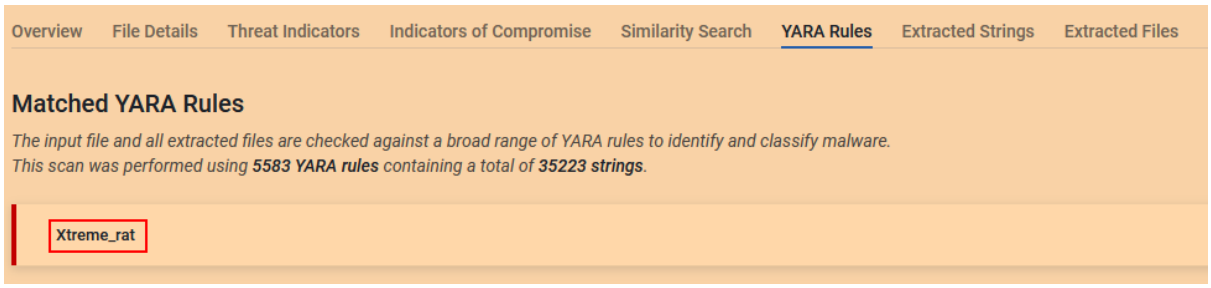
```
PS C:\Users\DFIR-USER\Desktop\88-Grrcon2015\temp_extract_dir\target1\dump_files> Get-FileHash -Path .\file.None.0x858aef78.AnyConnectInstaller.exe.dat

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          94A4EF65F99C594A8BFBFBC57F369EC2B6A5CF789F91BE89976086AAA507CD47       C:\Users\DFIR-USER\Desktop\88...
```

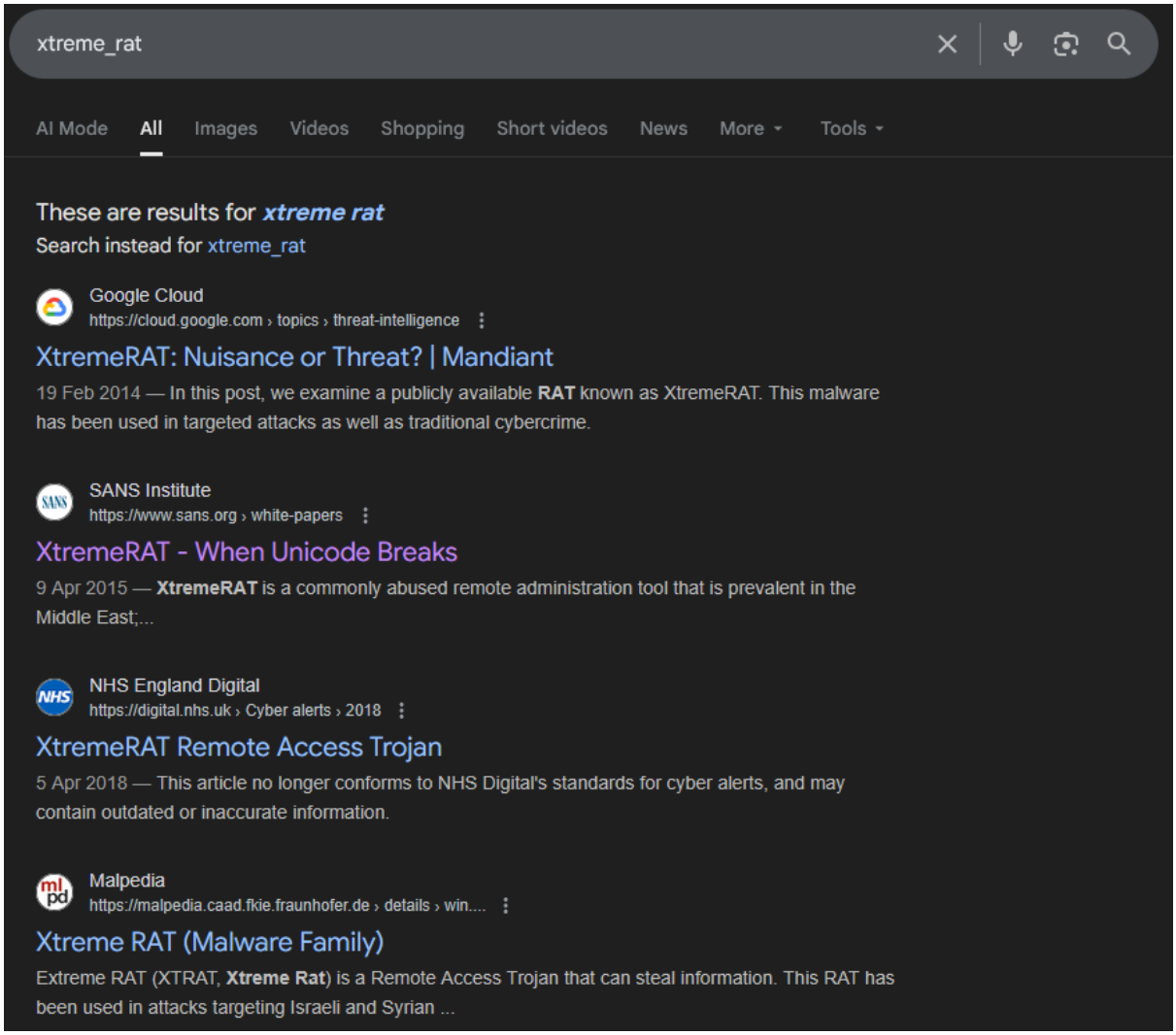
Upon submitting this to VirusTotal, we can see a significant number of detections:



Within the comments in VirusTotal, I found a link to a filescan.io report. In the YARA Rules section, we can see that it matches a YARA Rule called Xtreme_rat:



If you research this rat, we can see that it was a commonly abused remote administration tool:



Answer: xtremerat

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

Let's start by using the malfind plugin, which looks for injected code:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 malfind | Select-String -Pattern "Process:"`

This results in 5 processes we need to analyse further:

```
Process: explorer.exe Pid: 2116 Address: 0x32a0000
Process: explorer.exe Pid: 2116 Address: 0x3700000
Process: OUTLOOK.EXE Pid: 3196 Address: 0x110000
Process: OUTLOOK.EXE Pid: 3196 Address: 0x3290000
Process: OUTLOOK.EXE Pid: 3196 Address: 0x36c10000
Process: TeamViewer.exe Pid: 2680 Address: 0x2050000
Process: TeamViewer_Des Pid: 1092 Address: 0x6c0000
Process: mstsc.exe Pid: 2844 Address: 0x1410000
```

After analysing these processes further, I can't find any indication of process injection. One kind of code injection technique is called process hollowing. Process hollowing involves replacing the executable section of a legitimate process with malicious code. One method of detecting this is by looking at suspicious parent-child relationships. Using the pstree plugin:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 pstree`

We can see that iexplore.exe was used to spawn cmd.exe:

```
 0x85d0d030:iexplore.exe                                  2996   2984
. 0x83f105f0:cmd.exe                                      1856   2996
```

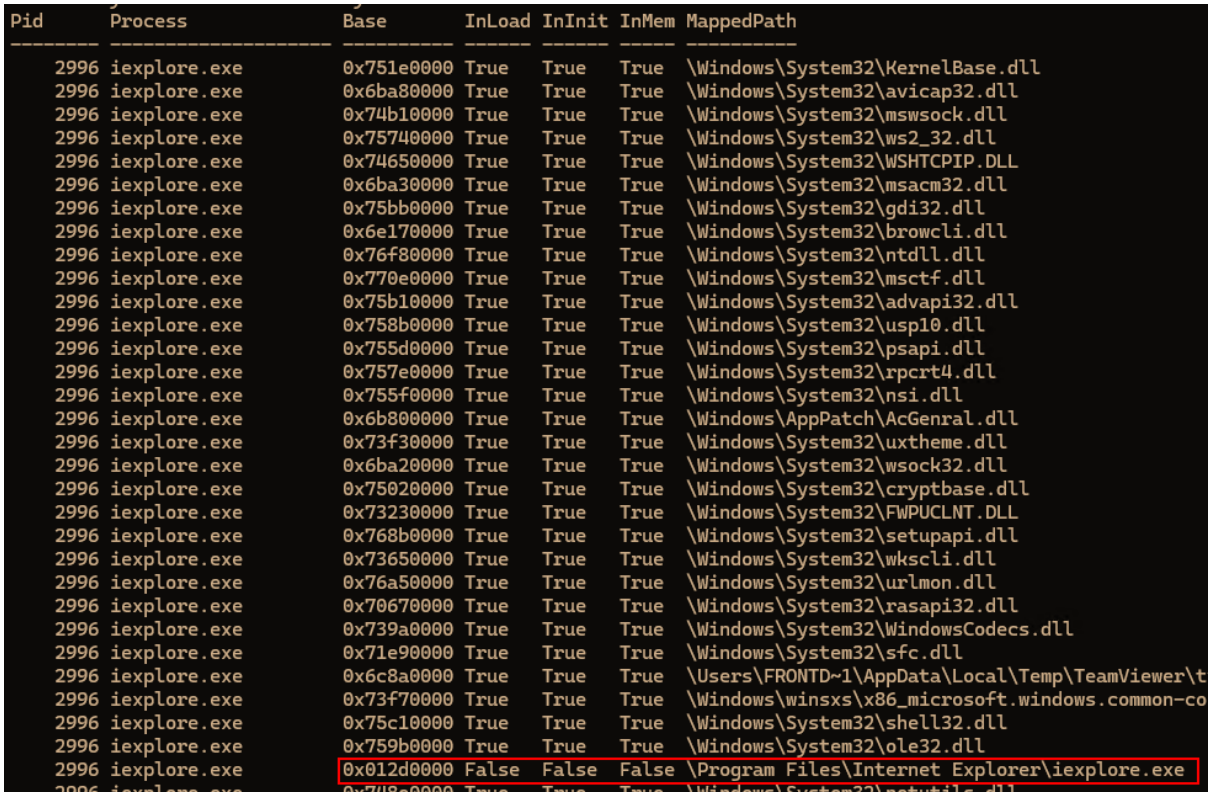
Another method of detecting process hallowing is comparing the results from the PEB (process environment block) structure and the VAD (virtual address descriptor) structure. The PEB structure resides in the process memory and tracks the full path to the executable and its base address, whereas the VAD resides in the kernel memory. Running the dlllist plugin shows the path to iexplore.exe and the base address where it is loaded (dlllist gets this information from the PEB):

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 dlllist -p 2996`

```
************************************************************************
iexplore.exe pid:   2996
Command line : "C:\Program Files\Internet Explorer\iexplore.exe"

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x13400000    0xa6000     0xffff C:\Program Files\Internet Explorer\iexplore.exe
0x76f80000   0x13c000     0xffff C:\Windows\SYSTEM32\ntdll.dll
0x76ea0000    0xd4000     0xffff C:\Windows\system32\kernel32.dll
0x751e0000    0x4a000     0xffff C:\Windows\system32\KERNELBASE.dll
0x75b10000    0xa0000     0xffff C:\Windows\system32\advapi32.dll
0x75690000    0xac000     0xffff C:\Windows\system32\msvcrt.dll
0x75890000    0x19000     0xffff C:\Windows\SYSTEM32\sechost.dll
```

The ldrmodules plugin, which relies on VAD in the kernel, shows an interesting result:



Pid	Process	Base	InLoad	InInit	InMem	MappedPath
2996	iexplore.exe	0x751e0000	True	True	True	\Windows\System32\KernelBase.dll
2996	iexplore.exe	0x6ba80000	True	True	True	\Windows\System32\avicap32.dll
2996	iexplore.exe	0x74b10000	True	True	True	\Windows\System32\mswsock.dll
2996	iexplore.exe	0x75740000	True	True	True	\Windows\System32\ws2_32.dll
2996	iexplore.exe	0x74650000	True	True	True	\Windows\System32\WSHTCPIP.DLL
2996	iexplore.exe	0x6ba30000	True	True	True	\Windows\System32\msacm32.dll
2996	iexplore.exe	0x75bb0000	True	True	True	\Windows\System32\gdi32.dll
2996	iexplore.exe	0x6e170000	True	True	True	\Windows\System32\browcli.dll
2996	iexplore.exe	0x76f80000	True	True	True	\Windows\System32\ntdll.dll
2996	iexplore.exe	0x770e0000	True	True	True	\Windows\System32\msctf.dll
2996	iexplore.exe	0x75b10000	True	True	True	\Windows\System32\advapi32.dll
2996	iexplore.exe	0x758b0000	True	True	True	\Windows\System32\usp10.dll
2996	iexplore.exe	0x755d0000	True	True	True	\Windows\System32\psapi.dll
2996	iexplore.exe	0x757e0000	True	True	True	\Windows\System32\rpcrt4.dll
2996	iexplore.exe	0x755f0000	True	True	True	\Windows\System32\nsi.dll
2996	iexplore.exe	0x6b800000	True	True	True	\Windows\AppPatch\AcGenral.dll
2996	iexplore.exe	0x73f30000	True	True	True	\Windows\System32\uxtheme.dll
2996	iexplore.exe	0x6ba20000	True	True	True	\Windows\System32\wsock32.dll
2996	iexplore.exe	0x75020000	True	True	True	\Windows\System32\cryptbase.dll
2996	iexplore.exe	0x73230000	True	True	True	\Windows\System32\FWPUCLNT.DLL
2996	iexplore.exe	0x768b0000	True	True	True	\Windows\System32\setupapi.dll
2996	iexplore.exe	0x73650000	True	True	True	\Windows\System32\wkscli.dll
2996	iexplore.exe	0x76a50000	True	True	True	\Windows\System32\urlmon.dll
2996	iexplore.exe	0x70670000	True	True	True	\Windows\System32\rasapi32.dll
2996	iexplore.exe	0x739a0000	True	True	True	\Windows\System32\WindowsCodecs.dll
2996	iexplore.exe	0x71e90000	True	True	True	\Windows\System32\sfc.dll
2996	iexplore.exe	0x6c8a0000	True	True	True	\Users\FRONTD~1\AppData\Local\Temp\TeamViewer\t
2996	iexplore.exe	0x73f70000	True	True	True	\Windows\winsxs\x86_microsoft.windows.common-co
2996	iexplore.exe	0x75c10000	True	True	True	\Windows\System32\shell32.dll
2996	iexplore.exe	0x759b0000	True	True	True	\Windows\System32\ole32.dll
2996	iexplore.exe	0x012d0000	False	False	False	\Program Files\Internet Explorer\iexplore.exe

The InLoad, InInit, and InMem flags all being False indicate that this module is not part of the module list maintained by the VAD. Legitimate modules loaded by a process should have at least InLoad = True and InMem = True if they are active. Given this, it's safe to say that iexplore.exe (PID 2996) fell victim to process hollowing.

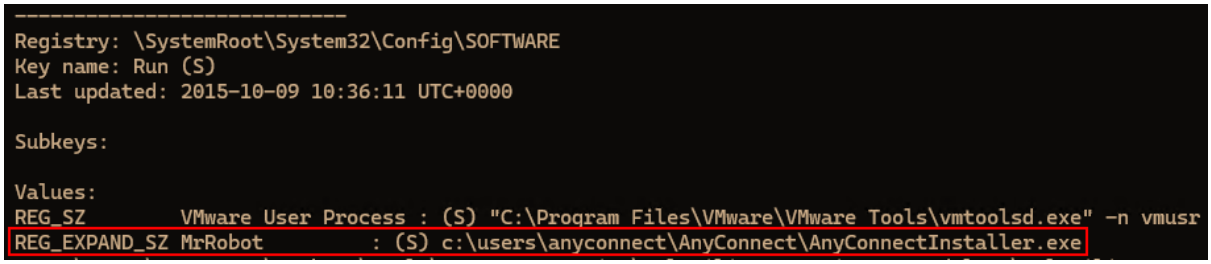
Answer: 2996

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

A common persistence mechanism is to add a registry Run key. The following command checks for one of these run key locations:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 printkey -V 0x8b79d008 -K Microsoft\Windows\CurrentVersion\Run`

In the output, we can see a Run key called MrRobot that executes "AnyConnectInstaller.exe":



```
----------------------------
Registry: \SystemRoot\System32\Config\SOFTWARE
Key name: Run (S)
Last updated: 2015-10-09 10:36:11 UTC+0000

Subkeys:

Values:
REG_SZ        VMware User Process : (S) "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
REG_EXPAND_SZ MrRobot         : (S) c:\users\anyconnect\AnyConnect\AnyConnectInstaller.exe
```

Answer: MrRobot

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?

Let's investigate the handles for iexplore.exe (PID 2996):

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 handles -p 2996`

In the output, we can see a mutant handle called "fsociety0.dat":

```
2996      0x150   0x1f0001 Mutant          fsociety0.dat
```

Malware often uses mutants (mutexes) to identify if a machine is already infected to prevent multiple copies from running. In this instance, it would check to see if "fsociety0.dat" exists.

Answer: fsociety0.dat

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

We can use the filescan plugin and grep for "users" to identify users on the system:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 filescan | grep -i "users"`

Here I found a user called zerocool:

```
\Device\HarddiskVolume2\Users\zerocool\
```

Zerocool is a character from the movie "Hackers":

Plot [edit]

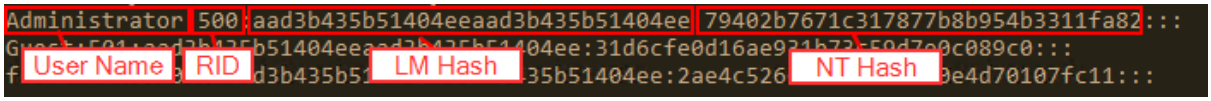
On August 10, 1988, 11-year-old Dade "Zero Cool" Murphy is barred from owning or

Answer: Hackers

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

To find the NTLM password hash for the administrator user, we can use the hashdump plugin:

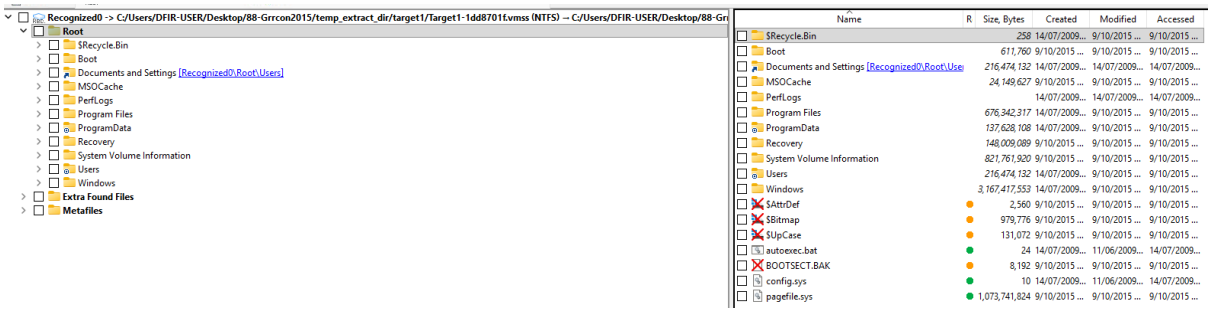
- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 hashdump`



Answer: 79402b7671c317877b8b954b3311fa82

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

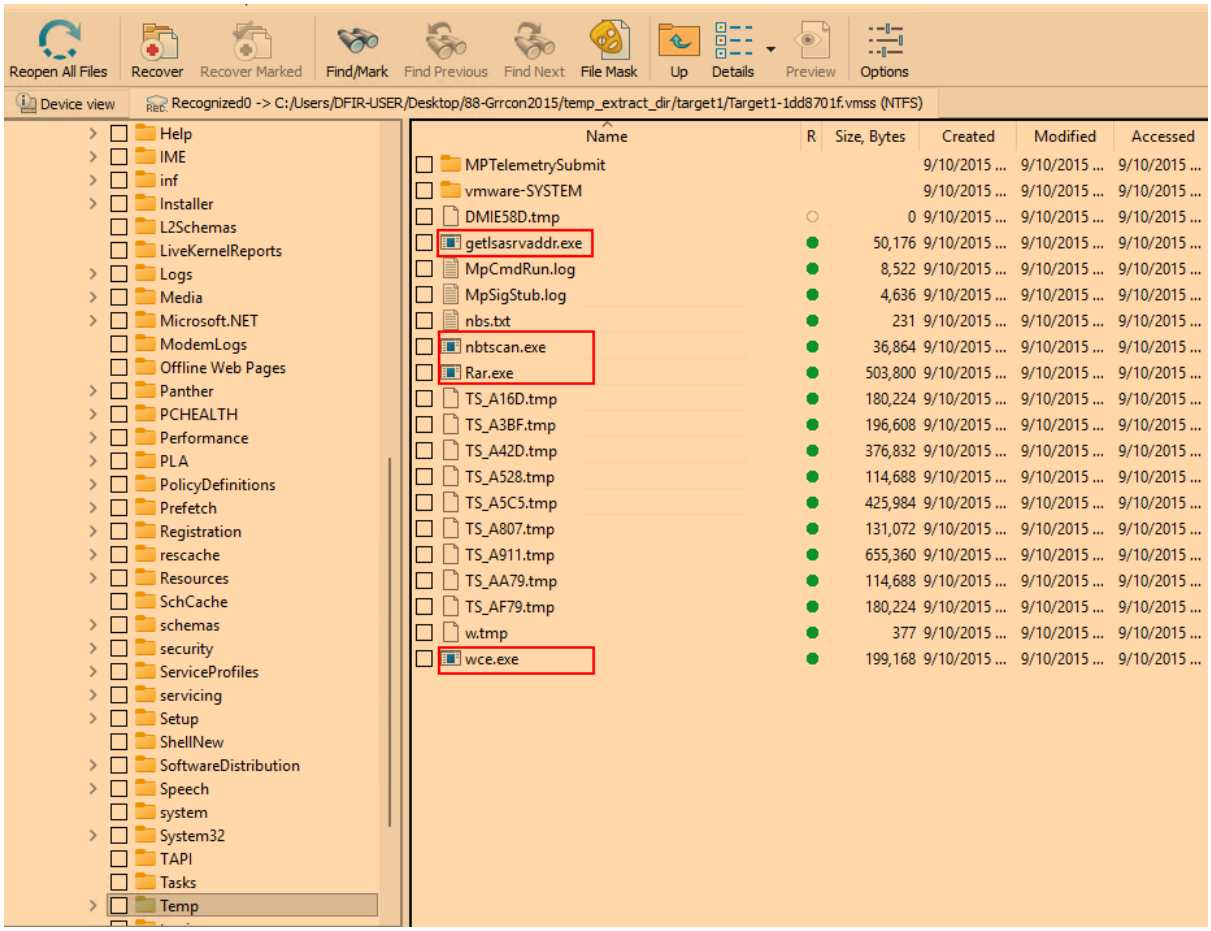
To analyse the filesystem of this host, I am going to use a tool called R-Studio, which is a data recovery tool capable of scanning memory dumps. After loading the memory dump, we can view extracted files in a file explorer-like interface:



We can now explore directories that often contain useful information (like malware):

- `C:\Windows\Temp`
- `C:\Users\<user>\Desktop`
- `C:\Users\<user>\Documents`
- `C:\Users\<user>\Downloads`
- `C:\Users\<user>\Appdata`
- `C:\Windows\System32`

Within C:\Windows\Temp, I found multiple interesting executable files:



- getlsasrvaddr.exe, a tool included with Windows Credential Editor (WCE) that can be used to read logon sessions and NTLM credentials from memory.
- nbtscan.exe, a tool that scans IP networks for NetBIOS name information.
- Rar.exe is WinRAR, potentially used for data staging.
- wce.exe, a password dumping tool.

Therefore, the threat actor moved over three tools as getlsasrvaddr.exe is part of WCE.

Answer: 3

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

Given that the threat actor transferred wce.exe to the compromised host, credential dumping likely occurred. We can use the consoles plugin in Volatility which extracts commands:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 consoles`

Here we can see wce being executed, dumping the password for the administrator account:

```
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:o0&77qj:^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
```

Following this, we can see the threat actor use the runas command to execute cmd as Administrator:

```
runas /profile /user:Administrator
runas /profile /user:Administrator cmd
```

Answer: flagadmin@1234

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

To find the standard information (SI) creation timestamp, we can use the mftparser plugin and grep for "nbtscan.exe":

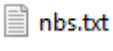
- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 mftparser | grep "nbtscan.exe"`

```
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
```

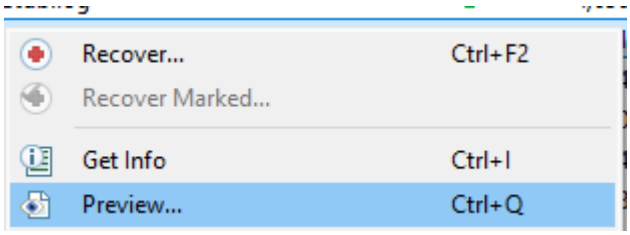
Answer: 2015-10-09 10:45:12 UTC

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?

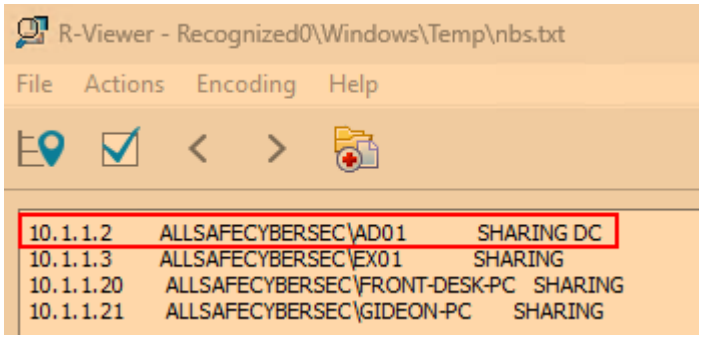
These sorts of questions are where a tool like R-Studio comes in clutch. If you navigate to the C:\Windows\Temp directory, we can see a file called nbs.txt:



If you right-click this file and select “Preview”:



We can see the contents of this file:



Here we can find the IP address of the first machine in the nbtscan.exe output. Without R-Studio, you would need to manually dump this file using a plugin like dumpfiles or another tool.

Answer: 10.1.1.2

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

Given that we know iexplore.exe (PID 2996) was injected into, let's use the netscan plugin to look for any network objects and grep for "iexplore":

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 netscan | grep "iexplore"`

```
TCPv4    10.1.1.20:49205          180.76.254.120:22      ESTABLISHED      2996      iexplore.exe
```

Here we can see a network connection from our host to 180.76.254.120 over port 22, which is the default port for SSH.

Answer: 180.76.254.120:22

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

Let's run the pstree plugin to see all running processes in the memory dump:

- `.\volatility_2.6_win64_standalone.exe -f "Target1-1dd8701f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 pstree`

```
Name                                                 Pid   PPid   Thds   Hnds Time
---------------------------------------------------- ------ ------ ------ ------ ----
 0x84ecbb18:csrss.exe                                 368    360      9    366 2015-10-09 11:30:47 UTC+0000
 0x84f97628:wininit.exe                               420    360      3     77 2015-10-09 11:30:48 UTC+0000
. 0x84e979f8:services.exe                             528    420      9    200 2015-10-09 11:30:48 UTC+0000
.. 0x85ae0cb0:dllhost.exe                            1888    528     13    196 2015-10-09 11:30:54 UTC+0000
.. 0x8586fd40:svchost.exe                             644    528     11    351 2015-10-09 11:30:48 UTC+0000
.. 0x85ae3030:vmtoolsd.exe                           1432    528      8    274 2015-10-09 11:30:54 UTC+0000
.. 0x85935030:svchost.exe                             796    528     19    446 2015-10-09 11:30:51 UTC+0000
.. 0x85d01510:svchost.exe                            3232    528      9    131 2015-10-09 11:31:34 UTC+0000
.. 0x858b69e8:msdtc.exe                              1980    528     12    145 2015-10-09 11:30:55 UTC+0000
.. 0x85978940:svchost.exe                             864    528     30   1036 2015-10-09 11:30:52 UTC+0000
.. 0x85969030:svchost.exe                             836    528     17    405 2015-10-09 11:30:52 UTC+0000
... 0x85c09968:dwm.exe                               2088    836      3     93 2015-10-09 11:31:04 UTC+0000
.. 0x85c39030:taskhost.exe                           2252    528      7    150 2015-10-09 11:31:04 UTC+0000
.. 0x8582c8d8:spoolsv.exe                            1228    528     12    273 2015-10-09 11:30:53 UTC+0000
.. 0x84e01448:svchost.exe                             720    528      6    276 2015-10-09 11:30:50 UTC+0000
.. 0x85a138f0:svchost.exe                            1124    528     16    484 2015-10-09 11:30:53 UTC+0000
.. 0x85a55d40:svchost.exe                            1256    528     17    304 2015-10-09 11:30:53 UTC+0000
.. 0x85b43a58:sppsvc.exe                             3900    528      4    153 2015-10-09 11:32:54 UTC+0000
.. 0x859cc2c0:svchost.exe                            1008    528     13    650 2015-10-09 11:30:52 UTC+0000
.. 0x8598c920:SearchIndexer.                         2544    528     13    670 2015-10-09 11:31:10 UTC+0000
.. 0x85976318:svchost.exe                            1784    528      5     99 2015-10-09 11:30:54 UTC+0000
. 0x8583b030:lsass.exe                                536    420      9    851 2015-10-09 11:30:48 UTC+0000
. 0x8583d960:lsm.exe                                  544    420     10    163 2015-10-09 11:30:48 UTC+0000
 0x83d334e8:System                                      4      0     94    500 2015-10-09 11:30:44 UTC+0000
. 0x84edcbf0:smss.exe                                 276      4      2     30 2015-10-09 11:30:44 UTC+0000
 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                             2996   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
```

We can see a process called TeamViewer.exe running. TeamViewer is a remote monitoring and management (RMM) tool that is often abused by threat actors.

Answer: TeamViewer.exe

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

Exploring the netscan output, we can see a connection from mstsc.exe:

```
10.1.1.20:49301          10.1.1.21:3389       ESTABLISHED     2844    mstsc.exe
127.0.0.1:1900           *:*                                  3232    svchost.exe     2015-10-09 11:32:55 UTC+0000
::1:56812                *:*                                  3232    svchost.exe     2015-10-09 11:32:55 UTC+0000
10.1.1.20:49291          107.6.97.19:5938     ESTABLISHED     2680    TeamViewer.exe
127.0.0.1:49298          127.0.0.1:6039       ESTABLISHED     1092    TeamViewer_Des
```

mstsc.exe is the executable file for the Remote Desktop (RDP) connection client on Windows. We can see that an RDP connection was established to 10.1.1.21 over port 3389, which is the default RDP port.

Answer: 10.1.1.21

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?

Using the following command:

- `.\volatility_2.6_win64_standalone.exe -f "target2-6186fe9f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 consoles`

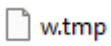
We can see wce being used to dump credentials on the machine, saving it to Gideon/w.tmp:

```
wce.exe -w > gideon/w.tmp
```

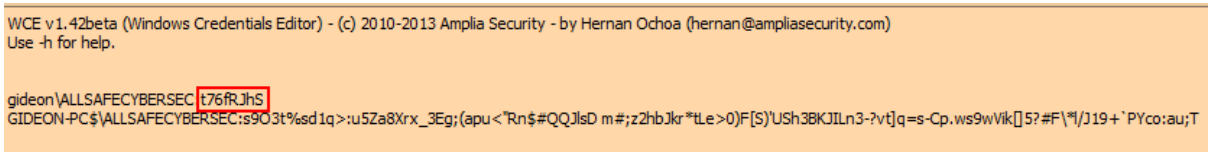
Using R-Studio, we can navigate to:

- `C:\Users\gideon`

Within this folder, we can find w.tmp:



If you preview this file, we can find Gideon's password:



Answer: t76fRJhS

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?

In the consoles output, we can see the threat actor connect to the remote computer's C: drive, assigning it to the local Z: drive:

```
net use z: \\10.1.1.2\c$
cd z:
```

Whilst in the Z: directory, we can see the threat actor encrypt the crownjewlez.rar achive using the password 123qwe!@#:

```
z:
dir
copy c:\users\gideon\rar.exe z:\crownjewels
cd crownjewels
dir
rar
rar crownjewlez.rar *.txt -hp123qwe!@#
```

Answer: 123qwe!@#

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

We discovered this in the previous question:

```
rar crownjewlez.rar
```

Answer: crownjewlez.rar

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

We need to determine how many files were archived in crownjewlez.rar. In the command history, wildcards were used "*.txt" when creating the archive, meaning the threat actor collected all .txt files in that directory. To determine how many files were added to this archive, we can examine the memory dump of conhost.exe (PID 3048) which contains relevant information as it was the console host process through which the threat actor executed their commands on the Gideon machine:

- `.\volatility_2.6_win64_standalone.exe -f "target2-6186fe9f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 memdump -p 3048 -D "\proc_dump"`

We can then run the strings command against this memory dump, searching for crownjewlez.rar and strings surrounding mentions of it:

- `strings 3048.dmp | grep "crownjewlez.rar" -A 10 -B 10`

After exploring the output, I came across these three .txt files:

```
crownjewlez.rar
Q;B
Rar.exet
SECRET~1.TXT
SecretSauce1.txt
SECRET~2.TXT
SecretSauce2.txt
SECRET~3.TXT
SecretSauce3.txt.sysx
PerfLogs
PROGRA~1
```

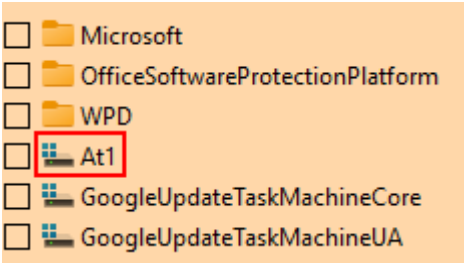
Answer: 3

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?

Scheduled tasks can be found on disk at:

- `%SYSTEMROOT%\System32\Tasks`

Here we can see a weird scheduled task called “At1”:



Let’s dump this file using the dumpfiles plugin:

- `.\volatility_2.6_win64_standalone.exe -f “target2-6186fe9f.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 dumpfiles -n -u -Q 0x000000003fc399b8 -D “\dump_files”`

If you open this file using a text editor, we can see that this scheduled task executes a file called 1.bat:

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.0" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo />
  <Triggers>
    <TimeTrigger>
      <StartBoundary>2015-10-09T08:00:00</StartBoundary>
    </TimeTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>@AtServiceAccount</UserId>
      <LogonType>InteractiveTokenOrPassword</LogonType>
      <RunLevel>HighestAvailable</RunLevel>
    </Principal>
  </Principals>
  <Actions Context="Author">
    <Exec>
      <Command>c:\users\gideon\1.bat</Command>
    </Exec>
  </Actions>
```

Answer: 1.bat

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

If you run the pstree plugin against the memory dump, we can see iexplore.exe spawn iexplore.exe which seems unusual:

```
Name                                                 Pid   PPid   Thds   Hnds Time
---------------------------------------------------- ------ ------ ------ ------ ----
 0x83e92b50:explorer.exe                            1836   3348     24    995 2015-10-09 05:25:15 UTC+0000
. 0x83d9b368:regsvr32.exe                           2928   1836      0 ------ 2015-10-09 05:25:18 UTC+0000
. 0x83f11958:OUTLOOK.EXE                            3376   1836     29   2185 2015-10-09 06:21:35 UTC+0000
. 0x84ae9668:notepad.exe                            2700   1836      4    261 2015-10-09 05:30:12 UTC+0000
. 0x8462c610:vmtoolsd.exe                           1200   1836      7    156 2015-10-09 05:25:35 UTC+0000
. 0x84627d40:jusched.exe                            2832   1836      2    119 2015-10-09 05:25:35 UTC+0000
. 0x83e55030:EXCEL.EXE                              2092   1836     11    386 2015-10-09 09:47:28 UTC+0000
 0x83d38bb0:System                                     4      0     93    534 2015-10-09 03:37:36 UTC+0000
. 0x84edc020:smss.exe                                280      4      2     33 2015-10-09 03:37:38 UTC+0000
 0x85ad1608:explorer.exe                             2200   2084     20    849 2015-10-09 03:39:33 UTC+0000
. 0x846fd030:vmtoolsd.exe                           2444   2200      7    145 2015-10-09 03:39:38 UTC+0000
. 0x859afd10:WINWORD.EXE                            3740   2200     10    419 2015-10-09 05:21:27 UTC+0000
. 0x846fd920:jusched.exe                            2464   2200      5    361 2015-10-09 03:39:38 UTC+0000
. 0x85989030:chrome.exe                             1960   2200      0 ------ 2015-10-09 05:05:58 UTC+0000
 0x83f324d8:iexplore.exe                            3208   3324     11    214 2015-10-09 12:35:57 UTC+0000
. 0x855d86d0:iexplore.exe                           3136   3208      2     32 2015-10-09 12:35:57 UTC+0000
 0x85409030:csrss.exe                                368    360      9    463 2015-10-09 03:37:42 UTC+0000
 0x83dd6458:wininit.exe                              432    360      3     79 2015-10-09 03:37:58 UTC+0000
. 0x858dcb20:services.exe                            528    432      6    204 2015-10-09 03:38:06 UTC+0000
.. 0x85a586a8:svchost.exe                            660    528     10    369 2015-10-09 03:38:54 UTC+0000
.. 0x9384fd40:SearchIndexer.                        2592    528     14    766 2015-10-09 03:39:44 UTC+0000
.. 0x85b61690:msdtc.exe                              388    528     12    147 2015-10-09 03:39:10 UTC+0000
.. 0x85a121d8:svchost.exe                            900    528     29   1308 2015-10-09 03:38:54 UTC+0000
... 0x83e83030:wuauclt.exe                           836    900      3     93 2015-10-09 07:00:27 UTC+0000
.. 0x85a8fb30:svchost.exe                           1828    528     12    357 2015-10-09 03:39:05 UTC+0000
.. 0x85a94330:svchost.exe                            860    528     17    461 2015-10-09 03:38:54 UTC+0000
... 0x83ee2030:dwm.exe                              4068    860      3     82 2015-10-09 05:25:15 UTC+0000
... 0x85809588:dwm.exe                              2156    860      3     70 2015-10-09 03:39:33 UTC+0000
.. 0x8586f420:vmtoolsd.exe                          1452    528      7    274 2015-10-09 03:38:57 UTC+0000
.. 0x85a9a560:svchost.exe                            824    528     18    476 2015-10-09 03:38:54 UTC+0000
.. 0x85a71030:svchost.exe                            736    528      7    323 2015-10-09 03:38:54 UTC+0000
.. 0x85b5cb80:taskhost.exe                          3528    528      8    175 2015-10-09 05:25:15 UTC+0000
.. 0x857ea030:spoolsv.exe                           1228    528     12    289 2015-10-09 03:38:55 UTC+0000
.. 0x85b1c4e0:taskhost.exe                           596    528      8    167 2015-10-09 03:39:32 UTC+0000
.. 0x83f2dd40:svchost.exe                           3544    528      9    145 2015-10-09 04:02:02 UTC+0000
.. 0x84e29308:svchost.exe                           1116    528     18    555 2015-10-09 03:38:54 UTC+0000
.. 0x85ae5258:svchost.exe                           1008    528     15    449 2015-10-09 03:38:54 UTC+0000
.. 0x85801608:svchost.exe                           1268    528     18    326 2015-10-09 03:38:55 UTC+0000
.. 0x85aca730:sppsvc.exe                            3068    528      4    152 2015-10-09 03:41:00 UTC+0000
. 0x858f6030:lsass.exe                               536    432      8    681 2015-10-09 03:38:06 UTC+0000
. 0x85906108:lsm.exe                                 552    432      9    184 2015-10-09 03:38:07 UTC+0000
 0x84dd5c10:winlogon.exe                             308   3292      3    114 2015-10-09 05:23:49 UTC+0000
 0x846f14c0:csrss.exe                               3064   3292     10    360 2015-10-09 05:23:48 UTC+0000
 0x84ac0738:javaw.exe                               3640   2052     21    631 2015-10-09 05:30:17 UTC+0000
 0x858b6d40:winlogon.exe                             480    412      3    109 2015-10-09 03:37:59 UTC+0000
 0x83dd4708:csrss.exe                                424    412     10    226 2015-10-09 03:37:58 UTC+0000
 0x84e78630:javaw.exe                               4092   2352     21    610 2015-10-09 05:01:32 UTC+0000
```

If you run the malfind plugin, we can see results for both iexplore.exe processes:

- `.\volatility_2.6_win64_standalone.exe -f "POS-01-c4e8f786.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 malfind | grep "Process:"`

Both contain an MZ file header, which suggests that a PE file was injected into the processes:

```
Process: iexplore.exe Pid: 3208 Address: 0x50000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 11, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00050000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00050010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00050020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00050030  00 00 00 00 00 00 00 00 00 00 00 00 d8 00 00 00   ................

0x00050000 4d               DEC EBP
0x00050001 5a               POP EDX
0x00050002 90               NOP
0x00050003 0003             ADD [EBX], AL
0x00050005 0000             ADD [EAX], AL
0x00050007 000400           ADD [EAX+EAX], AL
0x0005000a 0000             ADD [EAX], AL
0x0005000c ff               DB 0xff
0x0005000d ff00             INC DWORD [EAX]
0x0005000f 00b800000000     ADD [EAX+0x0], BH
0x00050015 0000             ADD [EAX], AL
0x00050017 004000           ADD [EAX+0x0], AL
0x0005001a 0000             ADD [EAX], AL
0x0005001c 0000             ADD [EAX], AL
0x0005001e 0000             ADD [EAX], AL
0x00050020 0000             ADD [EAX], AL
0x00050022 0000             ADD [EAX], AL
0x00050024 0000             ADD [EAX], AL
0x00050026 0000             ADD [EAX], AL
0x00050028 0000             ADD [EAX], AL
0x0005002a 0000             ADD [EAX], AL
0x0005002c 0000             ADD [EAX], AL
0x0005002e 0000             ADD [EAX], AL
0x00050030 0000             ADD [EAX], AL
0x00050032 0000             ADD [EAX], AL
0x00050034 0000             ADD [EAX], AL
0x00050036 0000             ADD [EAX], AL
0x00050038 0000             ADD [EAX], AL
0x0005003a 0000             ADD [EAX], AL
0x0005003c d800             FADD DWORD [EAX]
0x0005003e 0000             ADD [EAX], AL

Process: iexplore.exe Pid: 3136 Address: 0x50000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 11, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00050000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00050010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00050020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00050030  00 00 00 00 00 00 00 00 00 00 00 00 d8 00 00 00   ................
```

Using the netscan plugin, we can see PID 3208 make a connection to 54.84.237.92 over port 80 (default HTTP port):

- `.\volatility_2.6_win64_standalone.exe -f "POS-01-c4e8f786.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 netscan | grep "iexplore.exe"`

```
TCPv4    10.1.1.10:58751          54.84.237.92:80      CLOSE_WAIT      3208    iexplore.exe
```

Answer: 54.84.237.92

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

If you dump the injected PE file from iexplore.exe (PID 3208):

- `.\volatility_2.6_win64_standalone.exe -f "POS-01-c4e8f786.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 malfind -p 3208 -D .`

And generate its SHA256 hash, we can see that it gets labeled as Dexter by multiple vendors, including Microsoft:

Dexter is a point-of-sale malware which infects computers running Windows and steals sensitive information like credit and debit card information.

Answer: Dexter

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

If you run strings against the malfind dump for 3208:

- `strings "process.0x83f324d8.0x50000.dmp" | grep ".exe"`

We can see a process called “allsafe_protector.exe” mentioned:

```
allsafe_protector.exe
svchost.exe
iexplore.exe
explorer.exe
smss.exe
csrss.exe
winlogon.exe
lsass.exe
spoolsv.exe
alg.exe
wuauclt.exe
%s\%s\%s.exe
\Internet Explorer\iexplore.exe
.exe;.bat;.reg;.vbs;
.exe
tor.exe
iexplore.exe
lHost.exe
C:\Program Files\Internet Explorer\iexplore.exe
C:\Users\pos\AppData\Roaming\kdcpr\kdcpr.exe
C:\Users\pos\Downloads\allsafe_update.exe
```

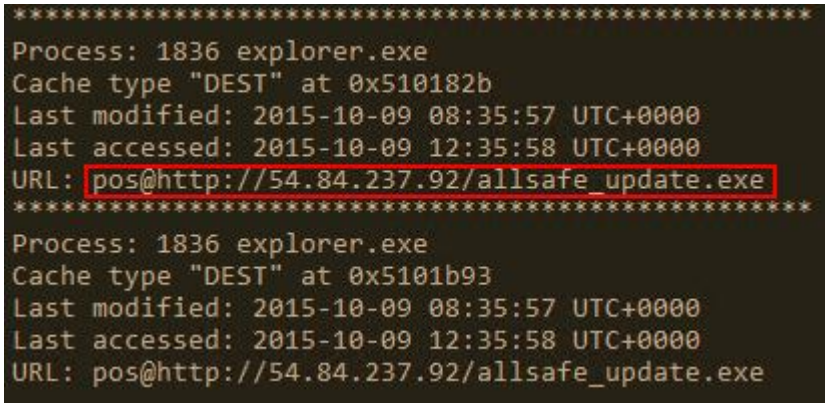
Answer: allsafe_protector.exe

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

Using the iehistory plugin:

- `.\volatility_2.6_win64_standalone.exe -f "POS-01-c4e8f786.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 iehistory`

We can see that a file called “allsafe_update.exe” was downloaded from the C2 server identified previously:



If you dump this file:

- `.\volatility_2.6_win64_standalone.exe -f "POS-01-c4e8f786.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 filescan | grep "allsafe_update"`
- `.\volatility_2.6_win64_standalone.exe -f "POS-01-c4e8f786.vmss" --profile=Win7SP1x86_23418 -g 0x82765be8 dumpfiles -n -u -Q 0x000000003e7ab038 -D .`

And run strings against it, we can see that it mentions allsafe_protector.exe. This suggests that allsafe_update.exe launches allsafe_protector.exe.

Answer: allsafe_update.exe