William Briguglio
104 205 372

February 3rd, 2019

Assignment 1

In order to examine the memory image, I used the following command to determine the profile:

python vol.py -f /home/caine/Desktop/Assign1/project1-c.vmem imageinfo

```
caine@caine:/usr/share/caine/pacchetti/volatility$ python vol.py -f /home/caine/
Desktop/Assign1/project1-c.vmem
Volatility Foundation Volatility Framework 2.4
ERROR   : __main__            : You must specify something to do (try -h)
caine@caine:/usr/share/caine/pacchetti/volatility$ python vol.py -f /home/caine/
Desktop/Assign1/project1-c.vmem imageinfo
Volatility Foundation Volatility Framework 2.4
Determining profile based on KDBG search...

          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/caine/Desktop/Assign1/project1-c.vmem)
                      PAE type : PAE
                           DTB : 0x2fe000L
                          KDBG : 0x80545ae0
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000
             KUSER_SHARED_DATA : 0xffdf0000
           Image date and time : 2012-07-22 02:45:08 UTC+0000
     Image local date and time : 2012-07-21 22:45:08 -0400
```

As shown, the memory image profile is WinXPSP3x86. Thus, I set the pertinent environment variables with the following commands before beginning forensics.

export VOLATILITY_PROFILE=WinXPSP3x86
export VOLATILITY_LOCATION=file:///home/caine/Desktop/Assign1/project1-c.vmem

1. To Identify running process I used the command:

python vol.py pslist

```
caine@caine:/usr/share/caine/pacchetti/volatility$ python vol.py pslist
Volatility Foundation Volatility Framework 2.4
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01 UTC+0000
caine@caine:/usr/share/caine/pacchetti/volatility$
```

The running processes at the time the memory image was create are all typical system processes that one would expect to find running on a Windows machine. Until I could gain more information, all of the listed processes were potential suspects.

William Briguglio
104 205 372

February 3rd, 2019

2. To Identify suspicious network connections, I ran the commands:

python vol.py connections
python vol.py connscan

```
caine@caine:/usr/share/caine/pacchetti/volatility$ python vol.py connections
Volatility Foundation Volatility Framework 2.4
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x81e87620 172.16.112.128:1038      41.168.5.140:8080         1484
caine@caine:/usr/share/caine/pacchetti/volatility$ python vol.py connscan
Volatility Foundation Volatility Framework 2.4
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x02087620 172.16.112.128:1038      41.168.5.140:8080         1484
0x023a8008 172.16.112.128:1037      125.19.103.198:8080       1484
caine@caine:/usr/share/caine/pacchetti/volatility$
```

Here we can see a process 1484 has made connections to the remote addresses 41.168.5.140 and 125.19.103.198. Referencing the list of running processes we obtained using **plist**, we can determine that this process is explorer.exe. This is a windows process that does not typically make connections to remote hosts, and thus is suspicious. Furthermore, we can see process 1640, reader_sl.exe, is a child process of 1484 and could also potentially be malicious.

3. Using https://iplocation.net/, we can see the country of origin of these remote addresses are South Africa and India. This is suspicious as both of these countries are not usually used by Microsoft and other North American software companies to host information. It also isn't typical for websites and webapps commonly used by North Americans to host their servers there.

IP Address	Country	Region	City
41.168.5.140	South Africa	Gauteng	Johannesburg

IP Address	Country	Region	City
125.19.103.198	India	Rajasthan	Jaipur

Additionally, the site https://www.ipvoid.com list the Indian IP address as blacklisted (however only by 1 of 108 blacklisting engines) but does not list the South African IP address as malicious.

Blacklist Status	BLACKLISTED 1/108
IP Address	125.19.103.198 Find Sites | IP Whois
Blacklist Status	POSSIBLY SAFE 0/108
IP Address	41.168.5.140 Find Sites | IP Whois

William Briguglio
104 205 372

February 3rd, 2019

4. In order to list the sockets involved I used the command:

Python vol.py sockscan

```
caine@caine:/usr/share/caine/pacchetti/volatility$ python vol.py sockscan
Volatility Foundation Volatility Framework 2.4
Offset(P)          PID   Port  Proto Protocol        Address         Create Time
---------- -------- ------ ------ -------------- --------------- ----------
0x01fd7618     1220   1900     17 UDP            172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x01fdb780      664    500     17 UDP            0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x0203f460        4    138     17 UDP            172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x02076620     1004    123     17 UDP            127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x020c23b0      908    135      6 TCP            0.0.0.0         2012-07-22 02:42:33 UTC+0000
0x02325610      788   1028      6 TCP            127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x02372808      664      0    255 Reserved       0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x02372c50      664   4500     17 UDP            0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x0239cc08        4    445      6 TCP            0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x023f0630     1004    123     17 UDP            172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x023f0d00        4    445     17 UDP            0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x02440d08     1484   1038      6 TCP            0.0.0.0         2012-07-22 02:44:45 UTC+0000
0x02476878        4    139      6 TCP            172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x02477460        4    137     17 UDP            172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x024cd2b0     1220   1900     17 UDP            127.0.0.1       2012-07-22 02:43:01 UTC+0000
caine@caine:/usr/share/caine/pacchetti/volatility$
```

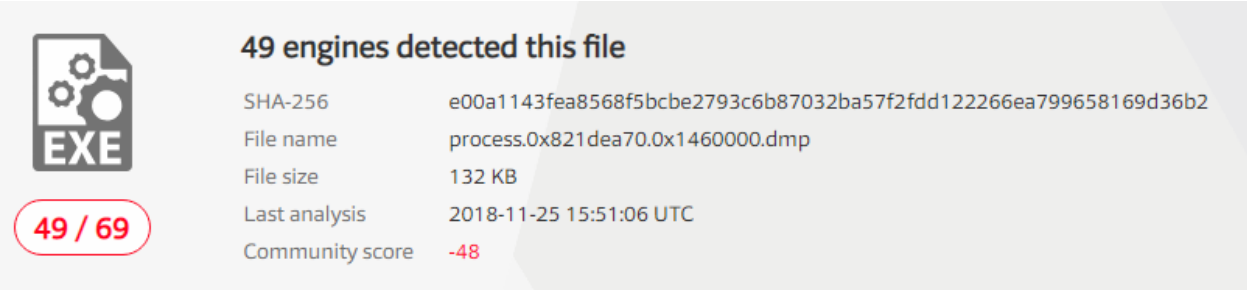
Here we can see that port 1038 is open by process 1484, the suspicious process we found earlier. This is also the most recently created socket, so I don't yet have reason to suspect any of these other sockets are related.

5. Next I extracted malicious process executables by using the following commands:

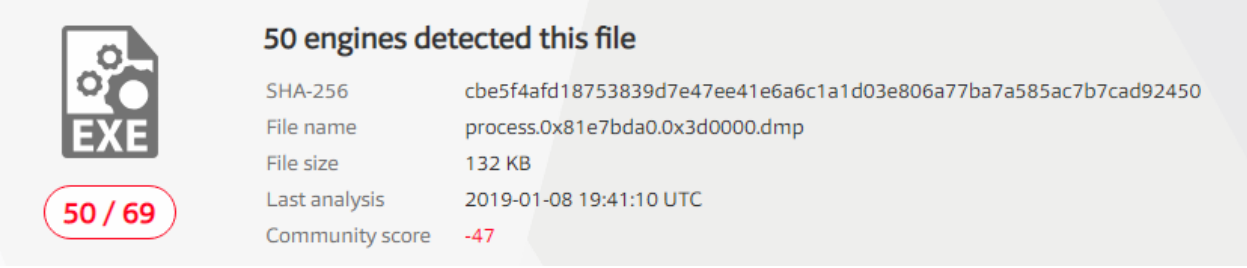
python vol.py malfind -p 1484 --dump-dir /home/caine/Desktop/Assign1/1484mf/

python vol.py malfind -p 1640 --dump-dir /home/caine/Desktop/Assign1/1640mf/

Each commanded yielded one executable. I then used https://www.virustotal.com to check if the extracted executables are malicious.



1484 extracted executable

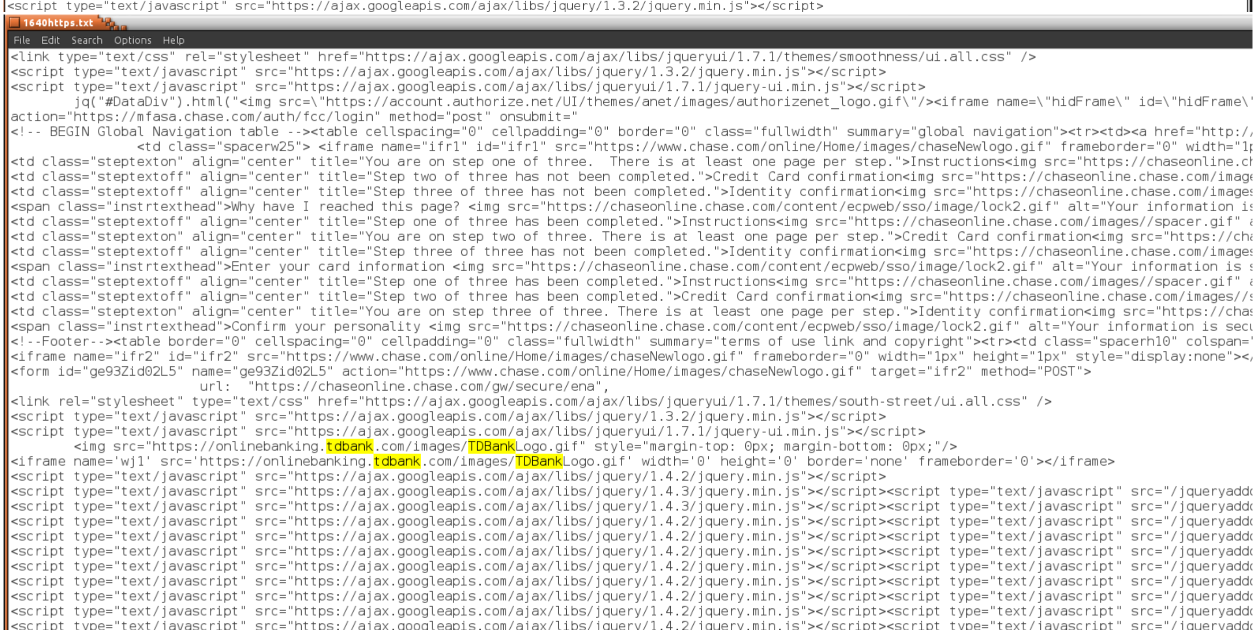


1640 extracted executable

The results shown above strongly indicate that both files are malicious.

6. Next, I used the following commands to search the process dump files for relevant URL's:

> **strings 1484.dmp I grep "http://" >> 1484http.txt**
> **strings 1484.dmp I grep "https://" >> 1484https.txt**
> **strings 1640.dmp I grep "http://" >> 1640http.txt**
> **strings 1640.dmp I grep "https://" >> 1640https.txt**



As seen above, In the process 1640 string grep results we can find mentions of TD bank as well as Chase bank, which is also mentioned in the process 1484 string grep results. The bank names appear in an HTML code, suggesting the malware authors planned to lead the users of the infected machine to a fake website to phish for banking information.

February 3rd, 2019

7. Finally, I used the following command to search the dump files for IP addresses:

strings [*PID*].dmp I grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b"

Most of the resulting strings were special addresses like 127.0.0.1, 0.0.0.0, and 255.255.255.255 or the infected machine's address and the other addresses on the network connected to the infected machine.

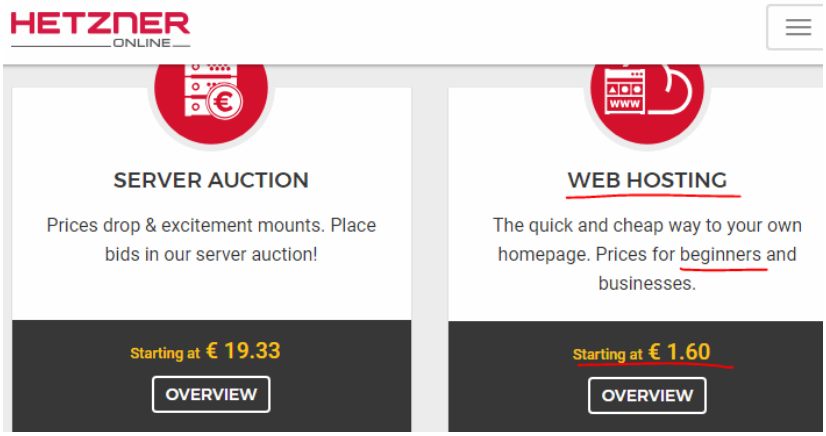
However, one address appeared in the grep results and I was able to get the full URL by searching for the address in a text file containing all the strings found in process 1640.

```
caine@caine:~/Desktop/Assign1$ strings 1640.dmp | grep -oE "\b([0-9]{1,3}\.){3
}[0-9]{1,3}\b"
188.40.0.138
5.1.0.0
6.0.0.0
1.0.0.0
```

```
File  Edit  Search  Options  Help
*/Web_Bank*
*jqueryaddonsv2.js*
http://188.40.0.138:8080/zb/v_01_a/in/cp.php
*account.authorize.net/*
<head*>
<style type="text/css">
```

Checking https://www.ipvoid.com shows the IP is not blacklisted but by checking https://www.iplocation.net we can see that the IP is located in Germany and belongs to the ISP Hetzner Online AG, a web cheap web hosting service potentially used by the attackers.

IP Address	Country	Region	City
188.40.0.138	Germany	Bayern	Gunzenhausen
ISP	**Organization**	**Latitude**	**Longitude**
Hetzner Online GmbH	Hetzner Online GmbH (hetzner.de)	49.1280	10.7704



William Briguglio
104 205 372

February 3rd, 2019

Extra. In addition to the information requirements outline in the assignment, some extra information was also retrieved.

By using the command:

strings [*pid*].dmp | grep “bank”

We can see the processes’ memory contain a list of many different banks from around the world, although, only Chase bank seems to be used. This suggest the present infection was tailored to the user of the infected machine or their location and that this attack was meant to be a world-wide customizable attack that targets users regardless of location and the bank they use.

When using the **grep** and **strings** commands to find the string “http://” in the malicious processes’ memory, I also found various URL’s related to web certificates authorities.

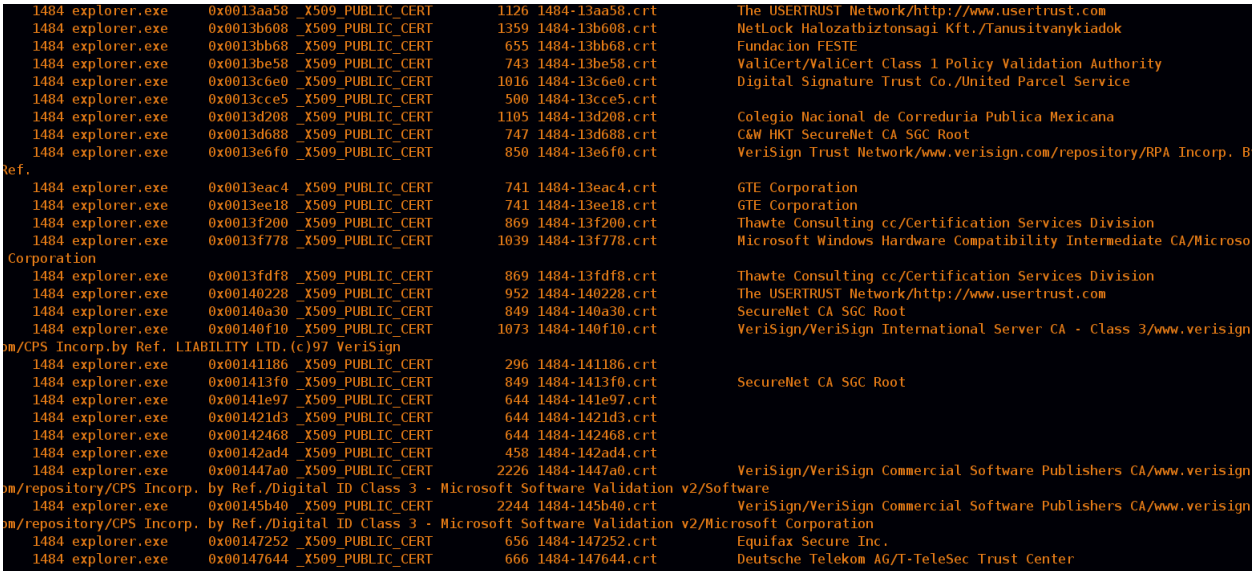
```
$http://www.trustcenter.de/guidelines0
'http://www.certplus.com/CRL/class3P.crl0
$http://crl.verisign.com/pca1.1.1.crl0G
http://www.usertrust.com1
http://www.usertrust.com1
3http://crl.usertrust.com/UTN-USERFirst-Hardware.crl01
http://www.valicert.com/1 0
http://www.valicert.com/1 0
(http://www.certplus.com/CRL/class3TS.crl0
*http://ca.sia.it/seccli/repository/CRL.der0J
http://www.usertrust.com1
http://www.usertrust.com1
,http://crl.usertrust.com/UTN-DATACorpSGC.crl0*
http://www.usertrust.com1+0)
http://www.usertrust.com1+0)
>http://crl.usertrust.com/UTN-USERFirst-NetworkApplications.crl0
*http://ca.sia.it/secsrv/repository/CRL.der0J
$http://crl.verisign.com/pca2.1.1.crl0G
http://www.valicert.com/1 0
http://www.valicert.com/1 0
&http://www.certplus.com/CRL/class1.crl0
$http://www.trustcenter.de/guidelines0
&http://www.certplus.com/CRL/class2.crl0
$http://www.trustcenter.de/guidelines0
$http://www.trustcenter.de/guidelines0
#http://www.entrust.net/CRL/net1.crl0+
&http://www.certplus.com/CRL/class3.crl0
5http://www.digsigtrust.com/DST_TRUST_CPS_v990701.html0
http://www.usertrust.com1604
http://www.usertrust.com1604
Ghttp://crl.usertrust.com/UTN-USERFirst-ClientAuthenticationandEmail.crl0
```

William Briguglio
104 205 372

February 3rd, 2019

I was able to extract the keys and certs in use by using the command:

python vol.py dumpcerts –dump-dir [*directory*] -ssl



As shown above, many of the keys belong to our malicious process, 1484. Some of these keys' descriptions show they are related to the URL's found in the memory of the malicious processes. This could suggest that the attackers created fake banking pages to phish for emails passwords and other information while using legitimate certificates provided from companies like VeriSign to stop browser warnings from alerting the user that they're accessing insecure websites.