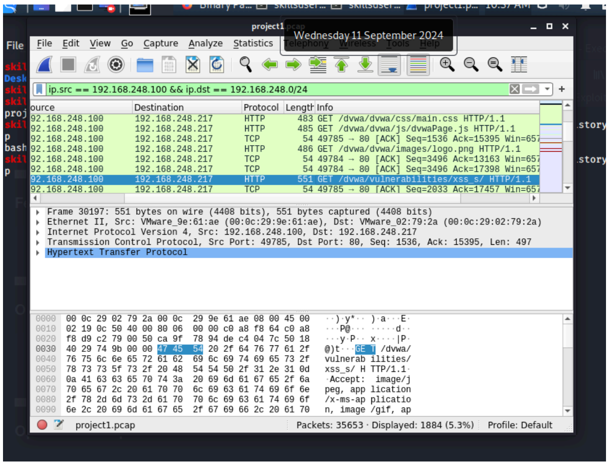
Project 1: Respond to an Incident Involving an Employee Being Compromised Using Tools to Practice Penetration Testing

Project 1: Respond to an Incident Involving an Employee Being Compromised Using Tools to Practice Penetration Testing

The customer has contacted us to find out whether or not they have been successfully breached. They sent some IT personnel to a Certified Ethical Hacker course recently and one of them downloaded a practice virtual machine from a popular Capture the Flag website. A few weeks later it came out that the site in question had been compromised months earlier and some of the VM's web applications might have been compromised by threat actors. The threat actors planted exploits via vulnerabilities like cross-site scripting (XSS) on some of the practice websites. These malicious drops or "watering hole" attacks point to IP addresses controlled by threat actors and are said to be hosting malicious code that could be used to take control of any computers that might visit the sites on the practice VM. You have just been provided with a current memory dump of the machine and a pcap file of traffic to and from the machine the suspected victim/IT employee was using. Using these artifacts, answer the following questions. You will need to use tools like Wireshark, Zeek, and Volatility to answer these questions. The suspected compromised machine's IP is 192.168.248.100


project1.pcap
Wednesday 11 September 2024
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

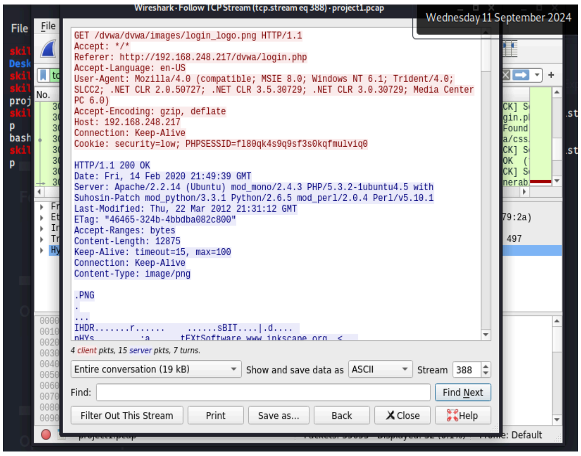
```
ip.src == 192.168.248.100 && ip.dst == 192.168.248.0/24
```

```
ource           Destination      Protocol Length Info
92.168.248.100  192.168.248.217  HTTP     483 GET /dvwa/dvwa/css/main.css HTTP/1.1
92.168.248.100  192.168.248.217  HTTP     485 GET /dvwa/dvwa/js/dvwaPage.js HTTP/1.1
92.168.248.100  192.168.248.217  TCP       54 49785 → 80 [ACK] Seq=1536 Ack=15395 Win=657
92.168.248.100  192.168.248.217  HTTP     486 GET /dvwa/dvwa/images/logo.png HTTP/1.1
92.168.248.100  192.168.248.217  TCP       54 49784 → 80 [ACK] Seq=3496 Ack=13163 Win=657
92.168.248.100  192.168.248.217  TCP       54 49784 → 80 [ACK] Seq=3496 Ack=17398 Win=657
92.168.248.100  192.168.248.217  HTTP     551 GET /dvwa/vulnerabilities/xss_s/ HTTP/1.1
92.168.248.100  192.168.248.217  TCP       54 49785 → 80 [ACK] Seq=2033 Ack=17457 Win=657
```

```
▸ Frame 30197: 551 bytes on wire (4408 bits), 551 bytes captured (4408 bits)
▸ Ethernet II, Src: VMware_9e:61:ae (00:0c:29:9e:61:ae), Dst: VMware_02:79:2a (00:0c:29:02:79:2a)
▸ Internet Protocol Version 4, Src: 192.168.248.100, Dst: 192.168.248.217
▸ Transmission Control Protocol, Src Port: 49785, Dst Port: 80, Seq: 1536, Ack: 15395, Len: 497
▸ Hypertext Transfer Protocol
```

```
0000  00 0c 29 02 79 2a 00 0c  29 9e 61 ae 08 00 45 00   ··)·y*·· )·a···E·
0010  02 19 0c 50 40 00 80 06  00 00 c0 a8 f8 64 c0 a8   ···P@··· ·····d··
0020  f8 d9 c2 79 00 50 ca 9f  78 94 de c4 04 7c 50 18   ···y·P·· x····|P·
0030  40 29 74 9b 00 00 47 45  54 20 2f 64 76 77 61 2f   @)t···GE T /dvwa/
0040  76 75 6c 6e 65 72 61 62  69 6c 69 74 69 65 73 2f   vulnerab ilities/
0050  78 73 73 5f 73 2f 20 48  54 54 50 2f 31 2e 31 0d   xss_s/ H TTP/1.1·
0060  0a 41 63 63 65 70 74 3a  20 69 6d 61 67 65 2f 6a   ·Accept:  image/j
0070  70 65 67 2c 20 61 70 70  6c 69 63 61 74 69 6f 6e   peg, app lication
0080  2f 78 2d 6d 73 2d 61 70  70 6c 69 63 61 74 69 6f   /x-ms-ap plicatio
0090  6e 2c 20 69 6d 61 67 65  2f 67 69 66 2c 20 61 70   n, image /gif, ap
```

project1.pcap Packets: 35653 · Displayed: 1884 (5.3%) Profile: Default

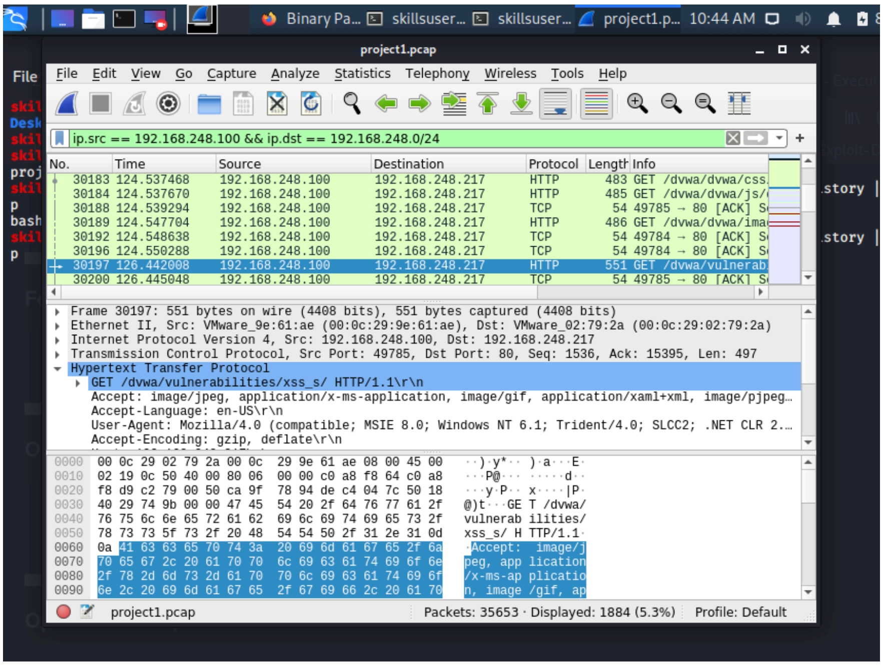

Wireshark · Follow TCP Stream (tcp.stream eq 388) · project1.pcap
Wednesday 11 September 2024

```
GET /dvwa/dvwa/images/login_logo.png HTTP/1.1
Accept: */*
Referer: http://192.168.248.217/dvwa/login.php
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center
PC 6.0)
Accept-Encoding: gzip, deflate
Host: 192.168.248.217
Connection: Keep-Alive
Cookie: security=low; PHPSESSID=f180qk4s9q9sf3s0kqfmulviq0

HTTP/1.1 200 OK
Date: Fri, 14 Feb 2020 21:49:39 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with
Suhosin-Patch mod_python/3.3.1 Python/2.6.5 mod_perl/2.0.4 Perl/v5.10.1
Last-Modified: Thu, 22 Mar 2012 21:31:12 GMT
ETag: "46465-324b-4bbdba082c800"
Accept-Ranges: bytes
Content-Length: 12875
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: image/png

.PNG
.
...
IHDR.......r......    ......sBIT....|.d....
pHYs          :a       tEXtSoftware www.inkscape.org  <
```

4 client pkts, 15 server pkts, 7 turns.

Entire conversation (19 kB) · Show and save data as: ASCII · Stream: 388 · Find:

As we can see the IP address for the virtual machine hosting the malicious website is 192.168.248.217, since the client suggested the victim was "practicing" using a local VM, got to filter it in Wireshark as we got the source address. The was a connection between the 192.168.248.2 and 192.168.248.217, the destination IP is in the same network range as the source. Packet 30197 shows the vulnerable practice server was visited. We get to see the time and date stamp of the first connection between the threat actor and the victim and we call tell any traffic from the Victicm's VM and the threat actor is bad.



project1.pcap

```
ip.src == 192.168.248.100 && ip.dst == 192.168.248.0/24
```

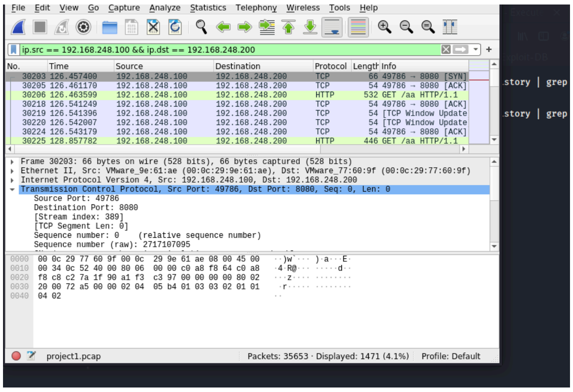
```
No.    Time        Source           Destination      Protocol Length Info
30183 124.537468  192.168.248.100  192.168.248.217  HTTP       483 GET /dvwa/dvwa/css
30184 124.537670  192.168.248.100  192.168.248.217  HTTP       485 GET /dvwa/dvwa/js/
30188 124.539294  192.168.248.100  192.168.248.217  TCP         54 49785 → 80 [ACK] S
30189 124.547704  192.168.248.100  192.168.248.217  HTTP       486 GET /dvwa/dvwa/ima
30192 124.548638  192.168.248.100  192.168.248.217  TCP         54 49784 → 80 [ACK] S
30196 124.550288  192.168.248.100  192.168.248.217  TCP         54 49784 → 80 [ACK] S
30197 126.442008  192.168.248.100  192.168.248.217  HTTP       551 GET /dvwa/vulnerab
30200 126.445048  192.168.248.100  192.168.248.217  TCP         54 49785 → 80 [ACK] S
```

```
▸ Frame 30197: 551 bytes on wire (4408 bits), 551 bytes captured (4408 bits)
▸ Ethernet II, Src: VMware_9e:61:ae (00:0c:29:9e:61:ae), Dst: VMware_02:79:2a (00:0c:29:02:79:2a)
▸ Internet Protocol Version 4, Src: 192.168.248.100, Dst: 192.168.248.217
▸ Transmission Control Protocol, Src Port: 49785, Dst Port: 80, Seq: 1536, Ack: 15395, Len: 497
▾ Hypertext Transfer Protocol
  ▸ GET /dvwa/vulnerabilities/xss_s/ HTTP/1.1\r\n
    Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/pjpeg…
    Accept-Language: en-US\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.…
    Accept-Encoding: gzip, deflate\r\n
```

```
0000  00 0c 29 02 79 2a 00 0c  29 9e 61 ae 08 00 45 00   ··)·y*·· )·a···E·
0010  02 19 0c 50 40 00 80 06  00 00 c0 a8 f8 64 c0 a8   ···P@··· ·····d··
0020  f8 d9 c2 79 00 50 ca 9f  78 94 de c4 04 7c 50 18   ···y·P·· x····|P·
0030  40 29 74 9b 00 00 47 45  54 20 2f 64 76 77 61 2f   @)t···GE T /dvwa/
0040  76 75 6c 6e 65 72 61 62  69 6c 69 74 69 65 73 2f   vulnerab ilities/
0050  78 73 73 5f 73 2f 20 48  54 54 50 2f 31 2e 31 0d   xss_s/ H TTP/1.1·
0060  0a 41 63 63 65 70 74 3a  20 69 6d 61 67 65 2f 6a   ·Accept:  image/j
0070  70 65 67 2c 20 61 70 70  6c 69 63 61 74 69 6f 6e   peg, app lication
0080  2f 78 2d 6d 73 2d 61 70  70 6c 69 63 61 74 69 6f   /x-ms-ap plicatio
0090  6e 2c 20 69 6d 61 67 65  2f 67 69 66 2c 20 61 70   n, image /gif, ap
```

project1.pcap   Packets: 35653 · Displayed: 1884 (5.3%)   Profile: Default

The attacker's IP address is 192.168.248.200, after connecting to the web server, we can see the immediate connection to port 8080 on that IP address as well as port 4444 this is all happening in the same IP address, we get this from filtering the IP addresses, were you see the behavior and other connections between the two. For the malicious processes running on the suspected compromised device, I clicked on files, and export objective to get the malicious file, There were several including ghost.exe, eDqYEC.exe, HgRgTVSdX.exe, tior.exe, but I couldn't run it on Virustotal as with VM provided it will always be blank when I login into the page and with Hybrid analysis still did not give me the answers I needed with the known Metasploit privilege escalation process.



```
ip.src == 192.168.248.100 && ip.dst == 192.168.248.200
```

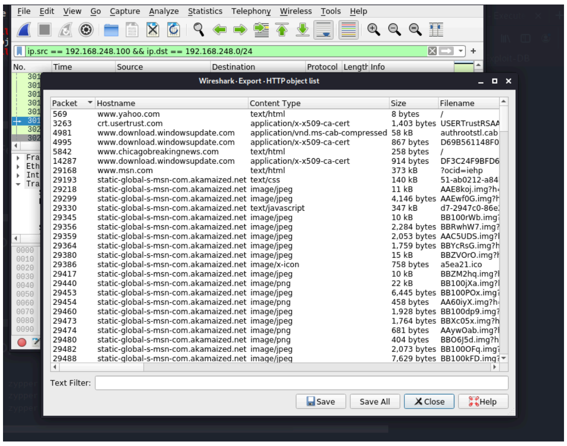
```
No.    Time        Source           Destination      Protocol Length Info
30203 126.457400  192.168.248.100  192.168.248.200  TCP         66 49786 → 8080 [SYN]
30205 126.461170  192.168.248.100  192.168.248.200  TCP         54 49786 → 8080 [ACK]
30206 126.463599  192.168.248.100  192.168.248.200  HTTP       532 GET /aa HTTP/1.1
30218 126.541249  192.168.248.100  192.168.248.200  TCP         54 49786 → 8080 [ACK]
30219 126.541396  192.168.248.100  192.168.248.200  TCP         54 [TCP Window Update
30220 126.542007  192.168.248.100  192.168.248.200  TCP         54 [TCP Window Update
30224 126.543179  192.168.248.100  192.168.248.200  TCP         54 49786 → 8080 [ACK]
30225 128.857782  192.168.248.100  192.168.248.200  HTTP       446 GET /aa HTTP/1.1
```

```
▸ Frame 30203: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: VMware_9e:61:ae (00:0c:29:9e:61:ae), Dst: VMware_77:60:9f (00:0c:29:77:60:9f)
▸ Internet Protocol Version 4, Src: 192.168.248.100, Dst: 192.168.248.200
▾ Transmission Control Protocol, Src Port: 49786, Dst Port: 8080, Seq: 0, Len: 0
    Source Port: 49786
    Destination Port: 8080
    [Stream index: 389]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 2717107095
```

```
0000  00 0c 29 77 60 9f 00 0c  29 9e 61 ae 08 00 45 00   ··)w`··· )·a···E·
0010  00 34 0c 52 40 00 80 06  00 00 c0 a8 f8 64 c0 a8   ·4·R@··· ·····d··
0020  f8 c8 c2 7a 1f 90 a1 f3  c3 97 00 00 00 00 80 02   ···z···· ········
0030  20 00 72 a5 00 00 02 04  05 b4 01 03 03 02 01 01    ·r····· ········
0040  04 02                                              ··
```

project1.pcap   Packets: 35653 · Displayed: 1471 (4.1%)   Profile: Default

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
story | grep
story | grep



```
ip.src == 192.168.248.100 && ip.dst == 192.168.248.0/24
```

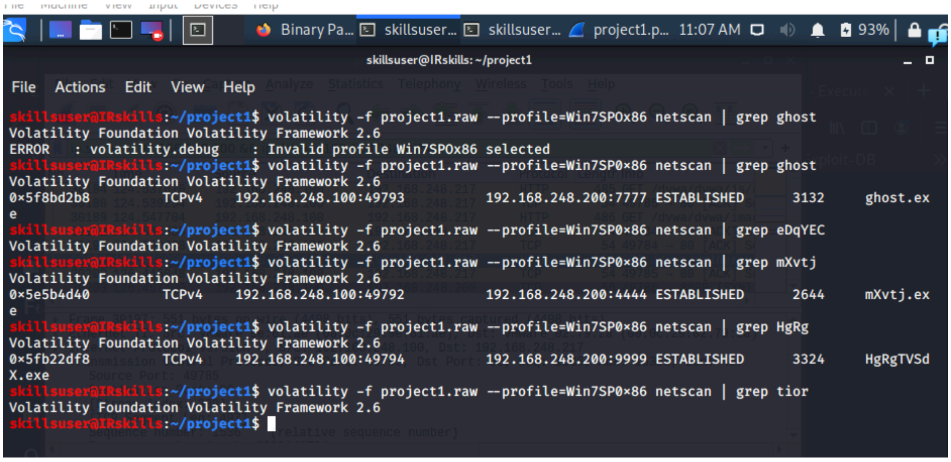
Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
569	www.yahoo.com	text/html	8 bytes	/
3263	crt.usertrust.com	application/x-x509-ca-cert	1,403 bytes	USERTrustRSAA
4981	www.download.windowsupdate.com	application/vnd.ms-cab-compressed	58 kB	authrootstl.cab
4995	www.download.windowsupdate.com	application/x-x509-ca-cert	867 bytes	D69B561148F0
5842	www.chicagobreakingnews.com	text/html	258 bytes	/
14287	www.download.windowsupdate.com	application/x-x509-ca-cert	914 bytes	DF3C24F9BFD6
29168	www.msn.com	text/html	373 kB	?ocid=iehp
29193	static-global-s-msn-com.akamaized.net	text/css	140 kB	51-ab0212-a84
29218	static-global-s-msn-com.akamaized.net	image/jpeg	11 kB	AAE8koj.img?h
29299	static-global-s-msn-com.akamaized.net	image/jpeg	4,146 bytes	AAEwf0G.img?h
29330	static-global-s-msn-com.akamaized.net	text/javascript	347 kB	d7-2947c0-86e
29345	static-global-s-msn-com.akamaized.net	image/jpeg	10 kB	BB100rWb.img?
29356	static-global-s-msn-com.akamaized.net	image/jpeg	2,284 bytes	BBRwhW7.img?
29359	static-global-s-msn-com.akamaized.net	image/jpeg	2,053 bytes	AAC5UDS.img?l
29364	static-global-s-msn-com.akamaized.net	image/jpeg	1,759 bytes	BBYcRsG.img?h
29380	static-global-s-msn-com.akamaized.net	image/jpeg	15 kB	BBZVOrO.img?h
29386	static-global-s-msn-com.akamaized.net	image/x-icon	758 bytes	a5ea21.ico
29417	static-global-s-msn-com.akamaized.net	image/jpeg	10 kB	BBZM2hq.img?l
29440	static-global-s-msn-com.akamaized.net	image/png	22 kB	BB100jXa.img?l
29453	static-global-s-msn-com.akamaized.net	image/jpeg	6,445 bytes	BB100POx.img?
29454	static-global-s-msn-com.akamaized.net	image/png	458 bytes	AA60iyX.img?h
29460	static-global-s-msn-com.akamaized.net	image/jpeg	1,928 bytes	BB100dp9.img?
29473	static-global-s-msn-com.akamaized.net	image/jpeg	1,764 bytes	BBXc05x.img?h
29474	static-global-s-msn-com.akamaized.net	image/png	681 bytes	AAywOab.img?l
29480	static-global-s-msn-com.akamaized.net	image/png	404 bytes	BBO6J5d.img?h
29482	static-global-s-msn-com.akamaized.net	image/jpeg	2,073 bytes	BB100OFq.img?
29488	static-global-s-msn-com.akamaized.net	image/jpeg	7,629 bytes	BB100kFD.img?

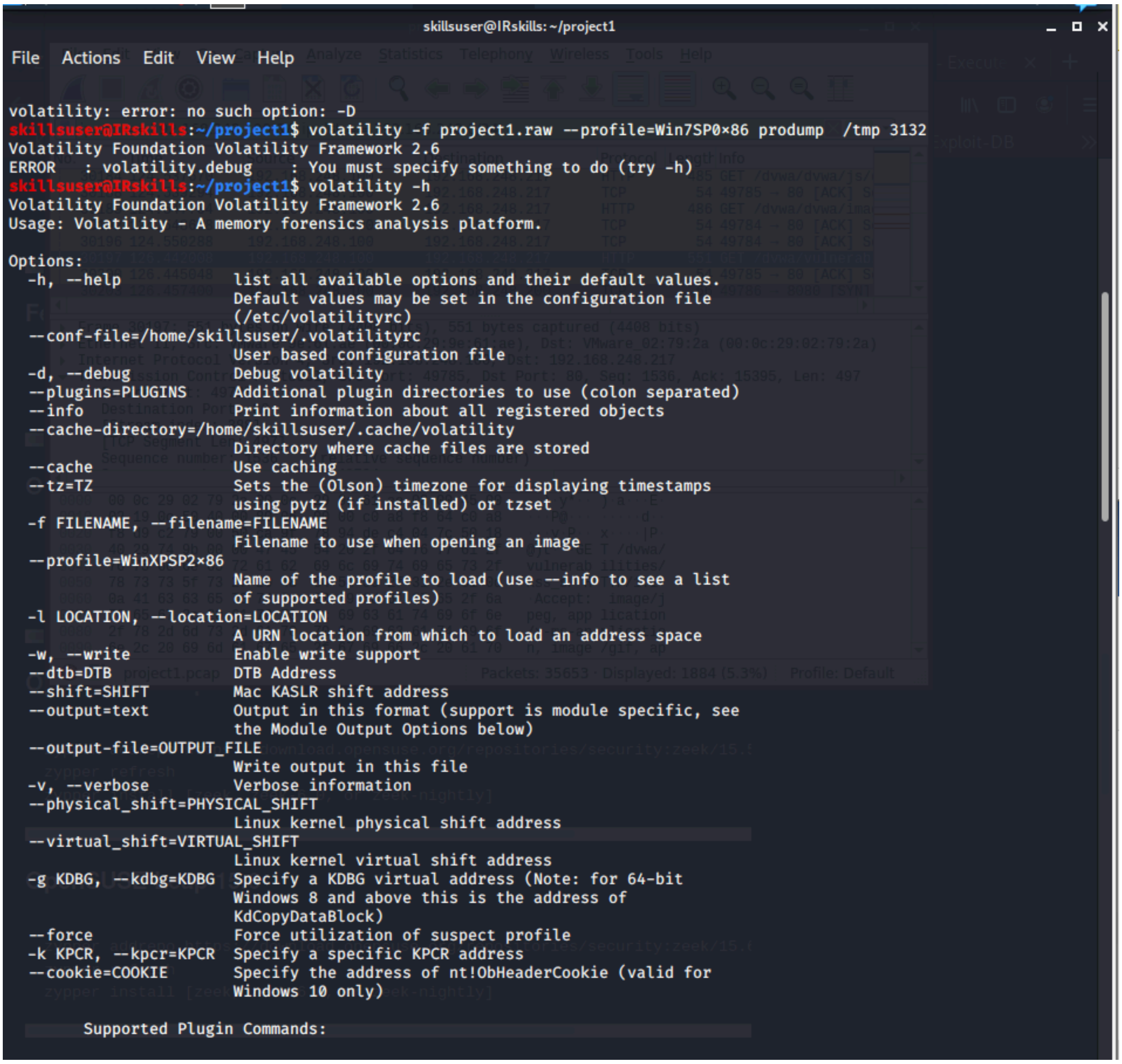
Text Filter:

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Save
Save All
Close
Help

The malicious processes running are actively connected, as we can see that ghost.exe, mXvtj.exe, and the HgTg.exe. We got this by running netscan plugin in volatility and then grepping each rogue process. With the results, we could also analyze that the connection port is connected through, for ghost.exe on port 7777, mXvtj.exe on port 4444 and HgRg.exe on port 9999 they all in the same IP address 192.168.248.200



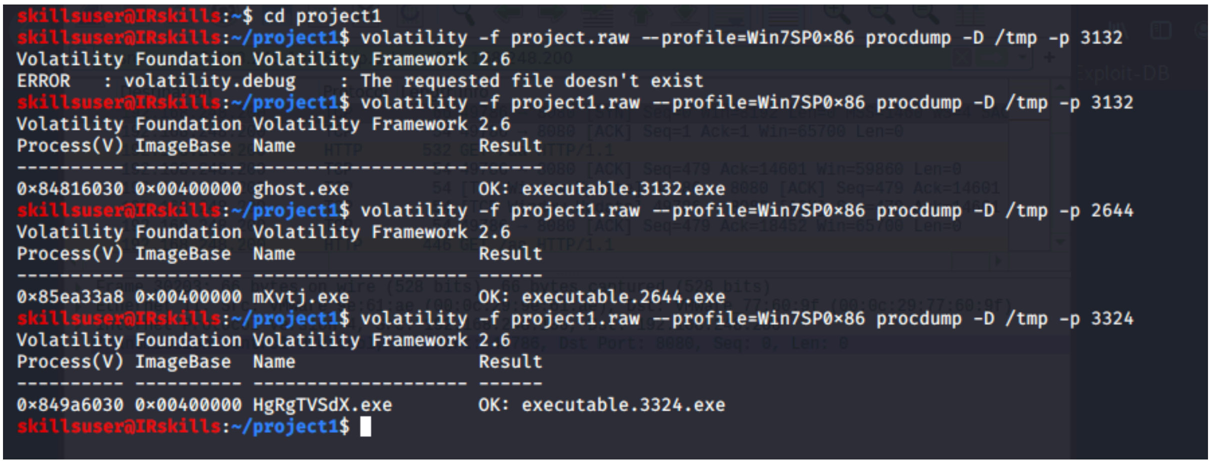
```
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SPOx86 netscan | grep ghost
Volatility Foundation Volatility Framework 2.6
ERROR   : volatility.debug    : Invalid profile Win7SPOx86 selected
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 netscan | grep ghost
Volatility Foundation Volatility Framework 2.6
0x5f8bd2b8         TCPv4    192.168.248.100:49793          192.168.248.200:7777 ESTABLISHED       3132      ghost.ex
e
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 netscan | grep eDqYEC
Volatility Foundation Volatility Framework 2.6
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 netscan | grep mXvtj
Volatility Foundation Volatility Framework 2.6
0x5e5b4d40         TCPv4    192.168.248.100:49792          192.168.248.200:4444 ESTABLISHED       2644      mXvtj.ex
e
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 netscan | grep HgRg
Volatility Foundation Volatility Framework 2.6
0x5fb22df8         TCPv4    192.168.248.100:49794          192.168.248.200:9999 ESTABLISHED       3324      HgRgTVSd
X.exe
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 netscan | grep tior
Volatility Foundation Volatility Framework 2.6
skillsuser@IRskills:~/project1$ 
```



```
volatility: error: no such option: -D
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 prodump  /tmp 3132
Volatility Foundation Volatility Framework 2.6
ERROR    : volatility.debug    : You must specify something to do (try -h)
skillsuser@IRskills:~/project1$ volatility -h
Volatility Foundation Volatility Framework 2.6
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/home/skillsuser/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/home/skillsuser/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --shift=SHIFT         Mac KASLR shift address
  --output=text         Output in this format (support is module specific, see
                        the Module Output Options below)
  --output-file=OUTPUT_FILE
                        Write output in this file
  -v, --verbose         Verbose information
  --physical_shift=PHYSICAL_SHIFT
                        Linux kernel physical shift address
  --virtual_shift=VIRTUAL_SHIFT
                        Linux kernel virtual shift address
  -g KDBG, --kdbg=KDBG  Specify a KDBG virtual address (Note: for 64-bit
                        Windows 8 and above this is the address of
                        KdCopyDataBlock)
  --force               Force utilization of suspect profile
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address
  --cookie=COOKIE       Specify the address of nt!ObHeaderCookie (valid for
                        Windows 10 only)

        Supported Plugin Commands:
```

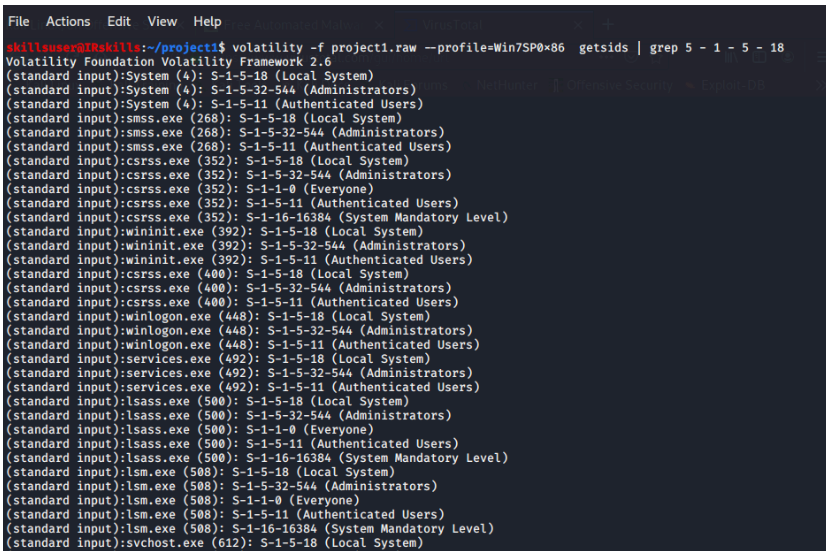
Lol, I did run into some issues, also make sure you spell everything correctly .

With the virustotal.com or other malware detector, you will be able to get more information from the feedback they give you when you generate the file also make sure it is not the organization’s data cause now you exposing them. To get feedback from Virustotal.com, you will have to extract each file using procdump , upload the extracted file on the virustotal.com



```
skillsuser@IRskills:~$ cd project1
skillsuser@IRskills:~/project1$ volatility -f project.raw --profile=Win7SP0x86 procdump -D /tmp -p 3132
Volatility Foundation Volatility Framework 2.6
ERROR   : volatility.debug    : The requested file doesn't exist
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 procdump -D /tmp -p 3132
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x84816030 0x00400000 ghost.exe            OK: executable.3132.exe
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 procdump -D /tmp -p 2644
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x85ea33a8 0x00400000 mXvtj.exe            OK: executable.2644.exe
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86 procdump -D /tmp -p 3324
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x849a6030 0x00400000 HgRgTVSdX.exe        OK: executable.3324.exe
skillsuser@IRskills:~/project1$ 
```

The attacker had system-level privileges as we got to figure out this through the running the volatility plugin getsids and grep 5 - 1 -5 -18



```
skillsuser@IRskills:~/project1$ volatility -f project1.raw --profile=Win7SP0x86  getsids | grep 5 - 1 - 5 - 18
Volatility Foundation Volatility Framework 2.6
(standard input):System (4): S-1-5-18 (Local System)
(standard input):System (4): S-1-5-32-544 (Administrators)
(standard input):System (4): S-1-5-11 (Authenticated Users)
(standard input):smss.exe (268): S-1-5-18 (Local System)
(standard input):smss.exe (268): S-1-5-32-544 (Administrators)
(standard input):smss.exe (268): S-1-5-11 (Authenticated Users)
(standard input):csrss.exe (352): S-1-5-18 (Local System)
(standard input):csrss.exe (352): S-1-5-32-544 (Administrators)
(standard input):csrss.exe (352): S-1-1-0 (Everyone)
(standard input):csrss.exe (352): S-1-5-11 (Authenticated Users)
(standard input):csrss.exe (352): S-1-16-16384 (System Mandatory Level)
(standard input):wininit.exe (392): S-1-5-18 (Local System)
(standard input):wininit.exe (392): S-1-5-32-544 (Administrators)
(standard input):wininit.exe (392): S-1-5-11 (Authenticated Users)
(standard input):csrss.exe (400): S-1-5-18 (Local System)
(standard input):csrss.exe (400): S-1-5-32-544 (Administrators)
(standard input):csrss.exe (400): S-1-5-11 (Authenticated Users)
(standard input):winlogon.exe (448): S-1-5-18 (Local System)
(standard input):winlogon.exe (448): S-1-5-32-544 (Administrators)
(standard input):winlogon.exe (448): S-1-5-11 (Authenticated Users)
(standard input):services.exe (492): S-1-5-18 (Local System)
(standard input):services.exe (492): S-1-5-32-544 (Administrators)
(standard input):services.exe (492): S-1-5-11 (Authenticated Users)
(standard input):lsass.exe (500): S-1-5-18 (Local System)
(standard input):lsass.exe (500): S-1-5-32-544 (Administrators)
(standard input):lsass.exe (500): S-1-1-0 (Everyone)
(standard input):lsass.exe (500): S-1-5-11 (Authenticated Users)
(standard input):lsass.exe (500): S-1-16-16384 (System Mandatory Level)
(standard input):lsm.exe (508): S-1-5-18 (Local System)
(standard input):lsm.exe (508): S-1-5-32-544 (Administrators)
(standard input):lsm.exe (508): S-1-1-0 (Everyone)
(standard input):lsm.exe (508): S-1-5-11 (Authenticated Users)
(standard input):lsm.exe (508): S-1-16-16384 (System Mandatory Level)
(standard input):svchost.exe (612): S-1-5-18 (Local System)
```