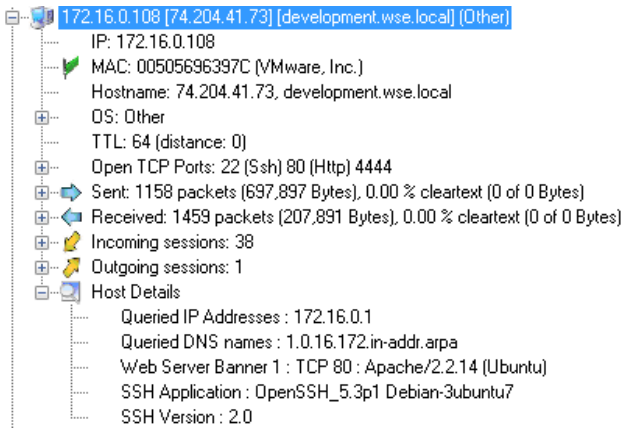
https://drive.google.com/folderview?id=0Bz3L4ZnVlUY8Q0VJbmJCV3JzR28&usp=sharing

1. Development.wse.local is a critical asset for the Wayne and Stark Enterprises, where the company stores new top secret designs on weapons. Jon Smith has access to the website and we believe it may have been compromised, according to the IDS alert we received earlier today. First determine the Public IP Address of the webserver?

Answer: 74.204.41.73

How to solve: First I opened the provided GrrCON.pcapng file in Network miner

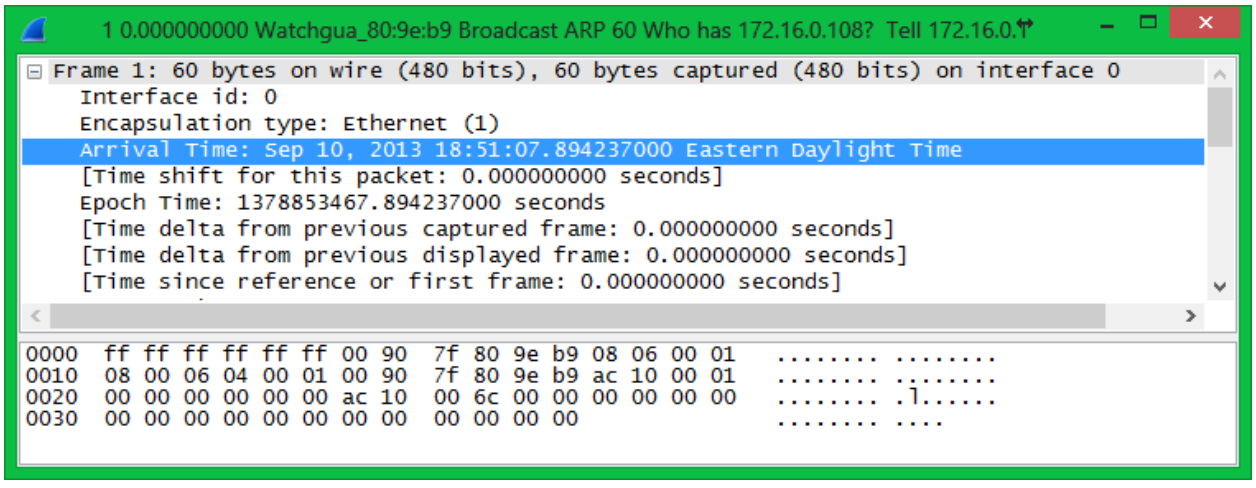
v1.5 after converting it with Wireshark to the .pcap format.



2. Alright now we need you to determine a starting point for the time line that will be useful in mapping out the incident. Please determine the arrival time of frame 1 in "GrrCON.pcapng" evidence file. Example Answer Format: 00:00:00.00000000 Do not include the date

Answer: 18:51:07.894237000

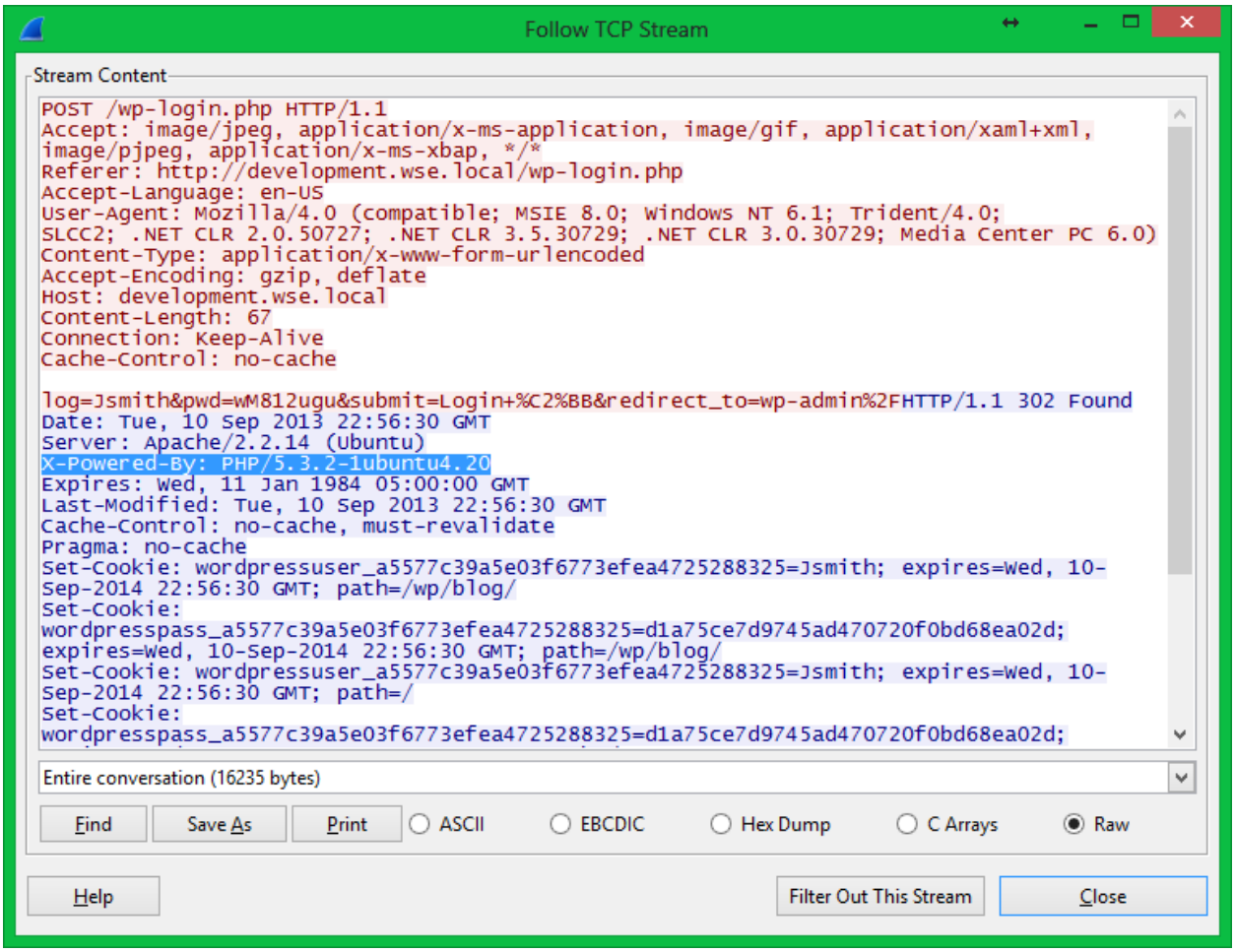
How to solve: Open the GrrCON.pcapng with Wireshark and open up the first packet (Frame 1) and expand out the first column the 3rd value down is the Arrival Time



3. What version number of PHP is the development.wse.local server running? Example 5.5

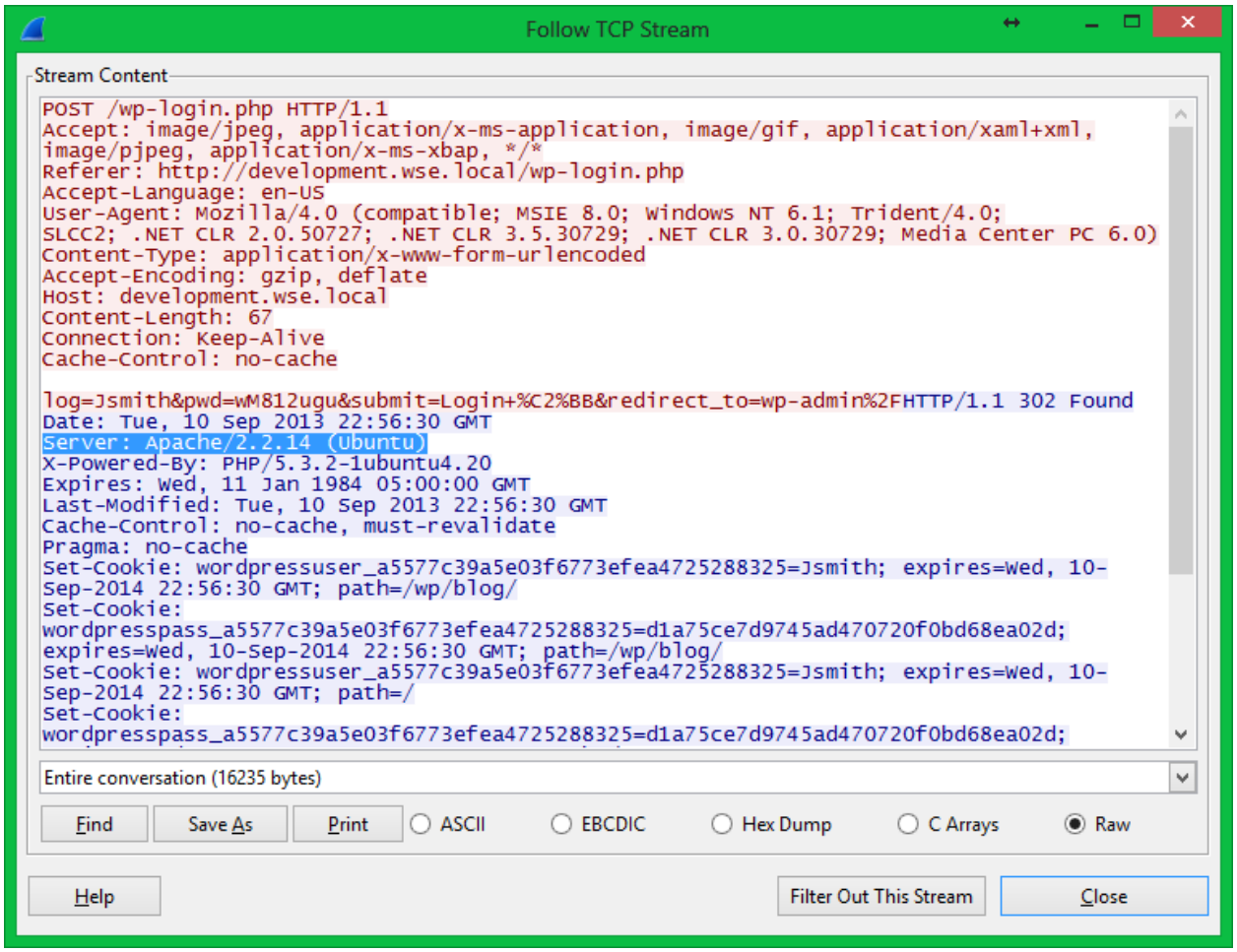
Answer: 5.3.2

How to solve: With Wireshark open I would recommend finding a potential GET request and following the TCP Stream. In the response from the webserver you should see a "X-Powered-By:" value that is followed by the full PHP version.



4. What version number of Apache is the development.wse.local web server using? (Do not include minor release information, Example 2.4

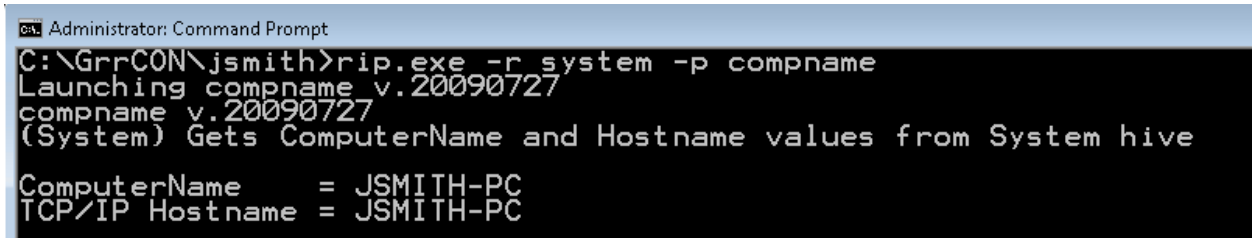
Answer: 2.2.14



5. What is the FQDN of the computer with the IP address 172.16.0.109?

Answer: JSMITH-PC.wse.local

How to solve: I just used rip.exe –r system –p compname to output the compute name, since we know the domain name is wse.local that gives us the answer. The System hive file was provided in the evidence files within the rar file provided.



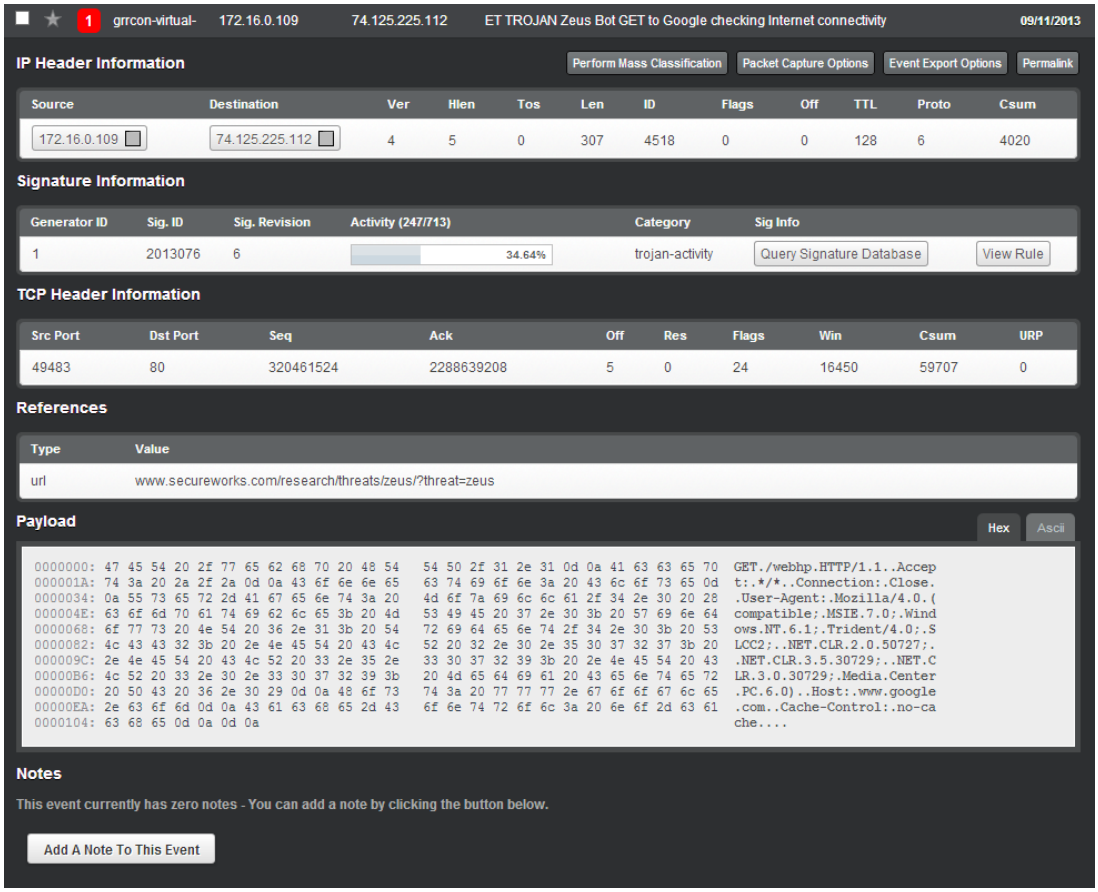
6. To verify the evidence files provide please provide the MD5 Checksum on the jsmith-disk-9-10.E01?

Answer: 62a3cd738c185a9ed987fe04d308f216

How to solve: In the provided files for JSMITH's computer they provided the jsmith-disk-9-10.E01.txt log file from the creation of the .E01 disk file which provides the MD5hash value of 62a3cd738c185a9ed987fe04d308f216. Now just open the jsmith-disk-9-10.E01 image file with FTK imager lite and run a verify disk to ensure you have the proper evidence files provided.

7. What is the common name of the malware reported by the IDS alert provided?

Answer: Zeus

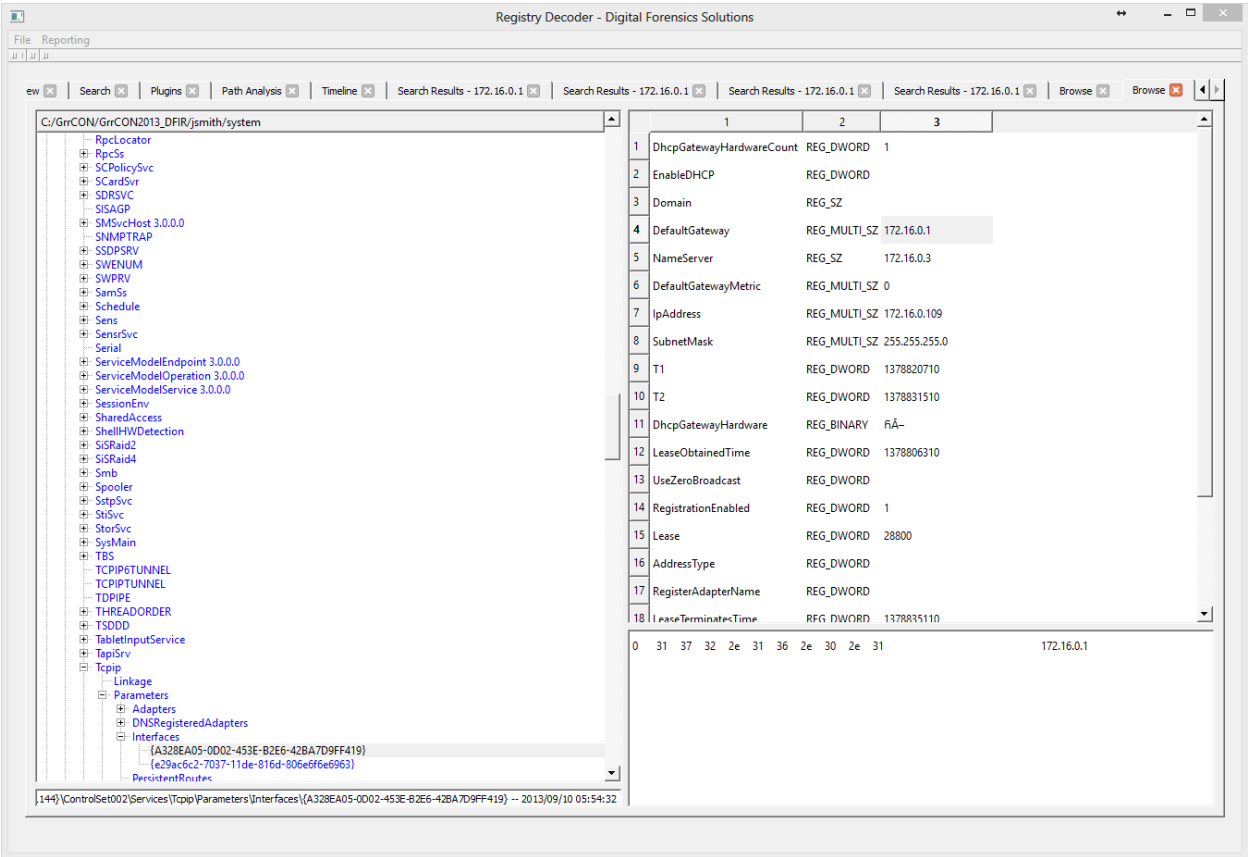


How to solve: Open the provided picture of the IDS alert that started it all. It's clearly a Zeus bot running a connectivity test with Google. Later we will verify the conclusion in memory on a different question.

8. Please identify the Gateway IP address of the LAN because the infrastructure team reported a potential problem with the IDS server that could have corrupted the PCAP?

Answer: 172.16.0.1

Well this is obvious just by looking at the PCAP but to confirm open the registry hive files provided in a tool like registry decoder and browse the SYSTEM hive file to ControlSet002\Services\Tcpip\Parameters\Interfaces\{A328EA05-0D02-453E-B2E6-42BA7D9FF419} and the 4th value is the Default Gateway of 172.16.0.1



9. According to the IDS alert, the Zeus bot attempted to ping an external website to verify connectivity. What was the IP address of the website pinged?

Answer: 74.125.225.112 – Google server pinged

How to solve: Just open the IDS alert picture provided and look at the Destination address provided (For reference look at the picture for question #7)

10. It's critical to the infrastructure team to identify the Zeus Bot CNC server IP address so they can block communication in the firewall as soon as possible. Please provide the IP address?

Answer: 88.198.6.20

How to solve: Zeus is known for injecting processes and opening connections to the CNC server with those injected processes. You can use the malfind plugin in volatility to identify the injected processes then run netscan with the |grep taskhost . Taskhost is one of the injected processes. Below is the output of those results.

```
root@bt:/volatility# python vol.py -f jsmith9-10.mem --profile=Win7SP0x86 netscan |grep taskhost
Volatile Systems Volatility Framework 2.3_beta
0x1eaa8a88 TCPv4    0.0.0.0:30639                  0.0.0.0:0              LISTENING        996      taskhost.exe
0x1eaa8a88 TCPv6    :::30639                       :::0                   LISTENING        996      taskhost.exe
0x1ed29580 TCPv4    0.0.0.0:30639                  0.0.0.0:0              LISTENING        996      taskhost.exe
0x1e735cd8 TCPv4    172.16.0.109:49497             88.198.6.20:80         CLOSE_WAIT       996      taskhost.exe
```

11. What was the process name that Zeus first started after it infected Jon Smiths computer?

Answer: eqhay.exe

How to solve: Using volatility you can import the plugin Zeusscan2 and run it against the memory image. Which the executable is the initial process that infected the machine.

```
root@bt:/volatility# python vol.py --plugins=contrib/plugins/malware/ -f jsmith9-10.mem --profile=Win7SP0x86 zeusscan2
Volatile Systems Volatility Framework 2.3_beta
**************************************************
Process                        : taskhost.exe
Pid                            : 996
Address                        : 1572864
URL 0                          : http://88.198.6.20/cf.bin
Identifier                     : JSMITH-PC_74DEB1E3432FA5CA
Mutant key                     : 763595725
XOR key                        : 3153831136
Registry                       : HKEY_CURRENT_USER\SOFTWARE\Microsoft\Toavk
 Value 1                       : Cesi
 Value 2                       : Awmugaleq
 Value 3                       : Misevoteo
Executable                     : Biqa\eqhay.exe
Data file                      : Lepiyr\ilwie.ovz
```

12. For the creation of a proper timeline please identify the CreateTime of the process Zeus started?

Answer: 2013-09-10 22:53:07

How to solve: Using volatility you can use the psscan plugin to identify when processes started and ended. I used the | grep eqhay to narrow the results.

```
root@bt:/volatility# python vol.py -f jsmith9-10.mem --profile=Win7SP0x86 psscan |grep eqhay
Volatile Systems Volatility Framework 2.3_beta
0x1e56c290 eqhay.exe          2180   2456 0x1ec9c400 2013-09-10 22:53:07 UTC+0000   2013-09-10 22:53:21 UTC+0000
```

13. The infrastructure team also requests that you identify the filename of the ".bin" configuration file that the Zeus bot downloaded right after the infection. Please provide the file name?

Answer: cf.bin

How to solve: Like question 11 you can use volatility with the plugin Zeusscan2 and run it against the memory image. Included in the URL 0 value is the http://88.198.6.20/cf.bin which provides the config file the zeusbot downloaded from the CNC server.

```
root@bt:/volatility# python vol.py --plugins=contrib/plugins/malware/ -f jsmith9-10.mem --profile=Win7SP0x86 zeusscan2
Volatile Systems Volatility Framework 2.3_beta
**************************************************
Process                        : taskhost.exe
Pid                            : 996
Address                        : 1572864
URL 0                          : http://88.198.6.20/cf.bin
Identifier                     : JSMITH-PC_74DEB1E3432FA5CA
Mutant key                     : 763595725
XOR key                        : 3153831136
Registry                       : HKEY_CURRENT_USER\SOFTWARE\Microsoft\Toavk
 Value 1                       : Cesi
 Value 2                       : Awmugaleq
 Value 3                       : Misevoteo
Executable                     : Biqa\eqhay.exe
Data file                      : Lepiyr\ilwie.ovz
```

14. So you can extract the process for reverse engineering please provide the physical offset location of the connection between the infected machine and the CnC server?

Answer: 0x1e735cd8

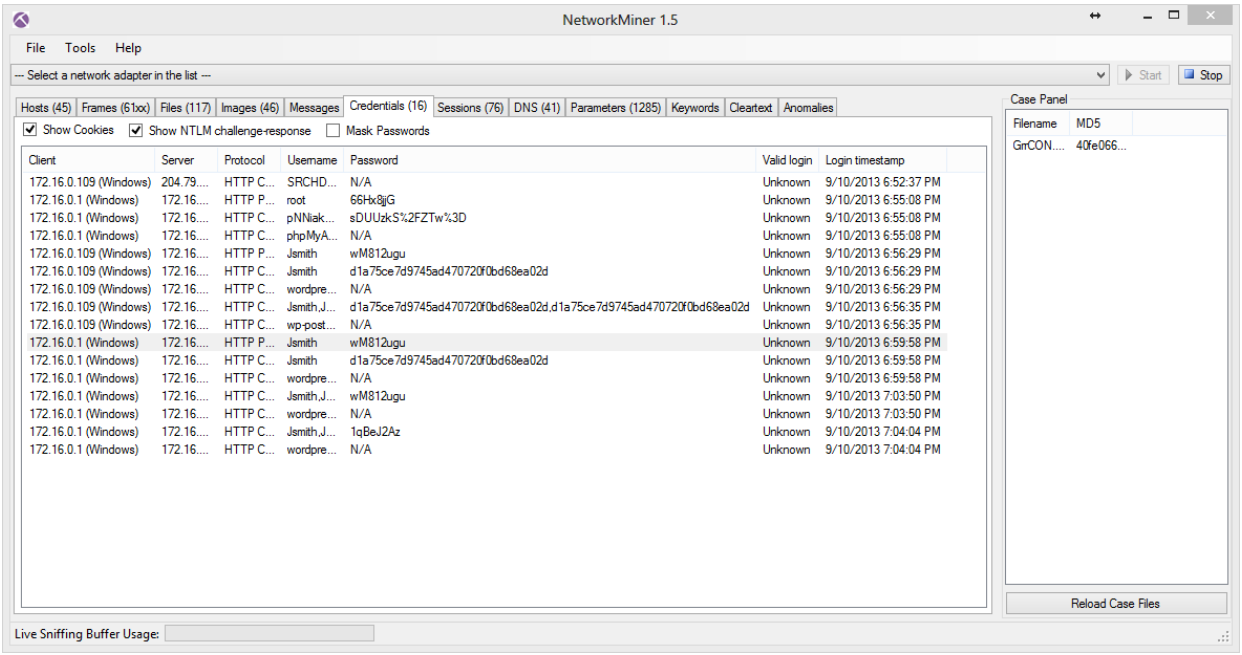
How to solve: Using the netscan plugin in volatility allows you to identify the physical offset of the connection structure that connected with the CNC server.

```
root@bt:/volatility# python vol.py -f jsmith9-10.mem --profile=Win7SP0x86 netscan |grep taskhost
Volatile Systems Volatility Framework 2.3_beta
0x1eaa8a88 TCPv4    0.0.0.0:30639                  0.0.0.0:0              LISTENING        996      taskhost.exe
0x1eaa8a88 TCPv6    :::30639                       :::0                   LISTENING        996      taskhost.exe
0x1ed29580 TCPv4    0.0.0.0:30639                  0.0.0.0:0              LISTENING        996      taskhost.exe
0x1e735cd8 TCPv4    172.16.0.109:49497             88.198.6.20:80         CLOSE_WAIT       996      taskhost.exe
```

15. No other users accessed the development.wse.local WordPress site during the timeline of the incident and the reports indicate that an account successfully logged in from the external interface. Please provide the password they used to login to the WordPress page around 6:59 PM EST?

Answer: wM812ugu

How to solve: Just use the tool Network miner v1.5 to process the GrrCON.pcap and browse to the Credentials tab. Search the login timestamp and find the JSMITH login with the password wM812ugu.



16. After reporting that the WordPress page was indeed accessed from an external connection, your boss comes to you in a rage over the potential loss of confidential top secret documents. He calms down enough to admit that the designs page has a separate access code outside to ensure the security of their information. Before storming off he provided the password to the designs page "1qBeJ2Az" and told you to find a time stamp of the access time or you will be fired. Please provide the time of the accessed Designs page? (Example 1:11:11AM)

Answer: 7:04:04PM

How to solve: Just use the tool Network miner v1.5 to process GrrCON.pcap and browse to the Credentials tab. Search the password logins for the 1qBeJ2Az password and the answer is right next with the login timestamp at 7:04:04PM. Refer to the screenshot in question 15 for reference.

17. What is the XOR cipher key for the Zeus bot?

Answer: 3153831136

How to solve: Using volatility you can import the plugin Zeusscan2 (Like question's 11 and 13) and run it against the memory image. The XOR key is one of the provided fields in the output of this plugin.

```
root@bt:/volatility# python vol.py --plugins=contrib/plugins/malware/ -f jsmith9-10.mem --profile=Win7SP0x86 zeusscan2
Volatile Systems Volatility Framework 2.3_beta
************************************************
Process                       : taskhost.exe
Pid                           : 996
Address                       : 1572864
URL 0                         : http://88.198.6.20/cf.bin
Identifier                    : JSMITH-PC_74DEB1E3432FA5CA
Mutant key                    : 763595725
XOR key                       : 3153831136
Registry                      : HKEY_CURRENT_USER\SOFTWARE\Microsoft\Toavk
 Value 1                      : Cesi
 Value 2                      : Awmugaleq
 Value 3                      : Misevoteo
Executable                    : Biqa\eqhay.exe
Data file                     : Lepiyr\ilwie.ovz
```

18. What is the full file path of the system shell spawned through the attacker's meterpreter session?

Answer: /bin/sh

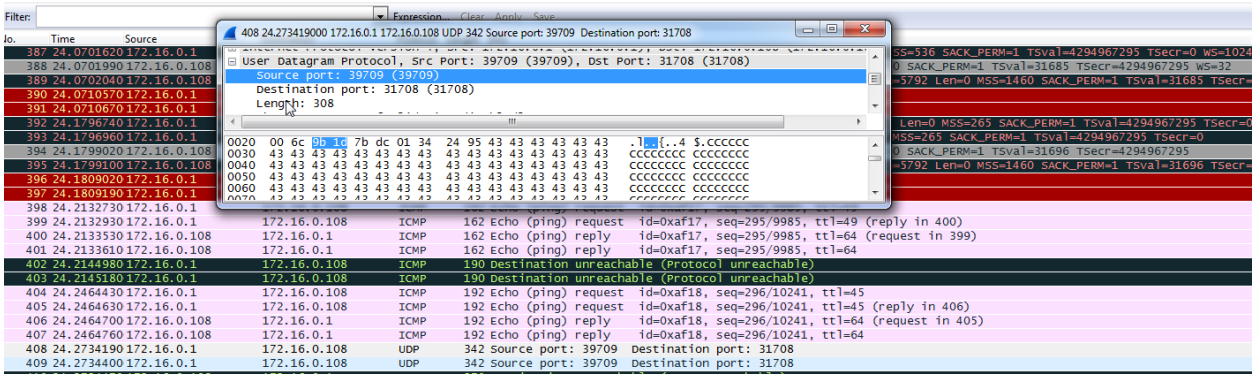
How to solve: Using the provided profile for the Linux server you can scan the memory image with the plugin linux_psaux which provides a list of the active processes like pslist but with the added functionality that mimics the ps aux command on a live system. Process 1274 you can see the command that starts the process 1275 in dir /bin/sh

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_psaux|grep 127
Volatile Systems Volatility Framework 2.3_beta
1270   33     33     /usr/sbin/apache2 -k start
1271   33     33     /usr/sbin/apache2 -k start
1274   33     33     sh -c /bin/sh
1275   33     33     /bin/sh
```

19. What is the source port number in the shellcode exploit? Dest Port was 31708 IDS Signature GPL SHELLCODE x86 inc ebx NOOP

Answer: 39709

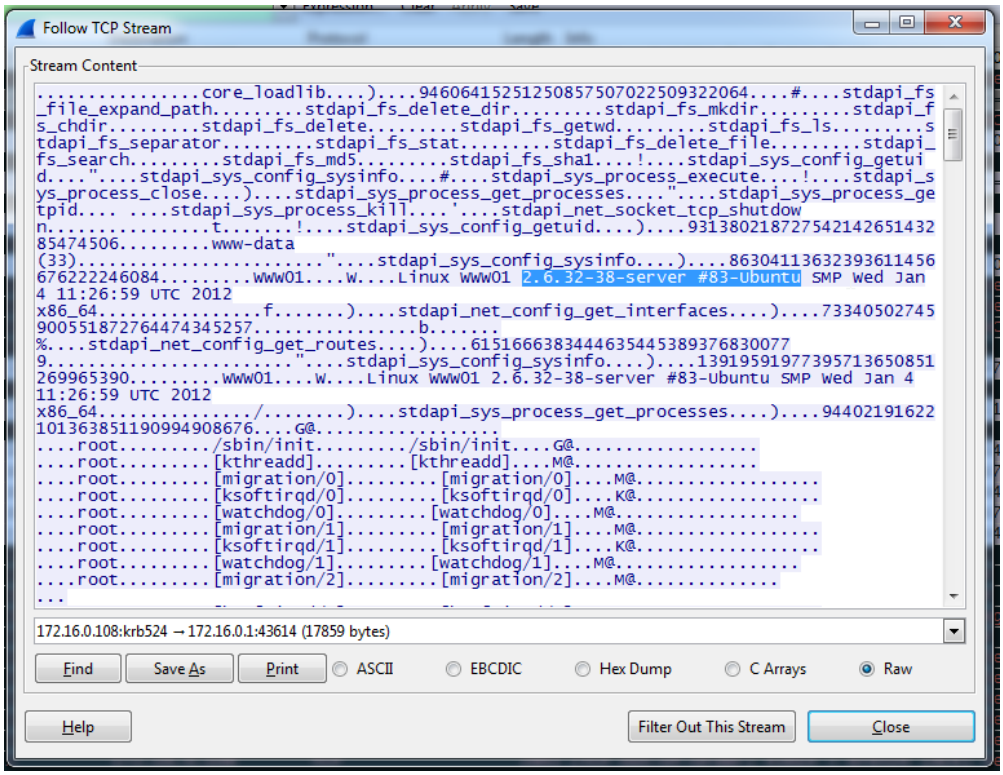
How to solve: Search through the pcap in wireshark and just after the ping echo and reply you will find the UDP traffic that starts the shellcode exploit and if you open the that frame 408 in the picture you will see the source port.



20. What was the Linux kernel version returned from the meterpreter sysinfo command run by the attacker?

Answer: 2.6.32-38-server #83-Ubuntu

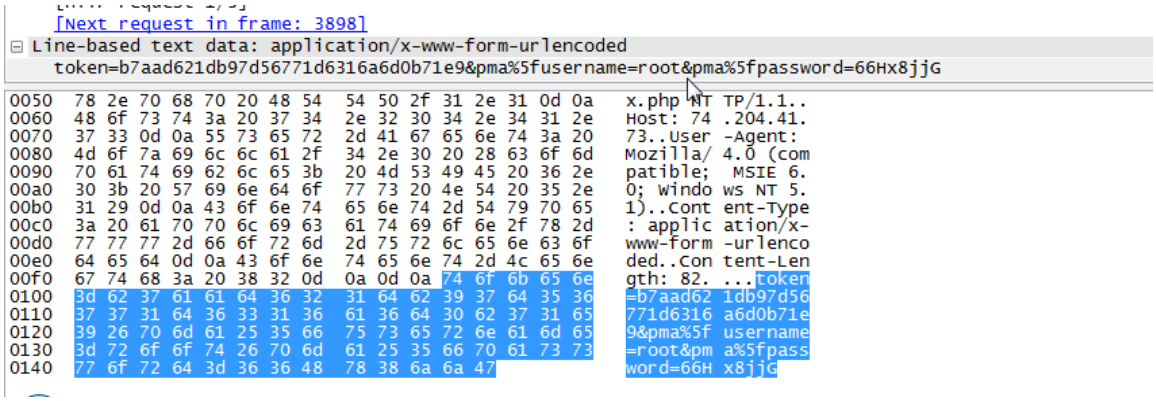
How to solve: In wireshark do a search for the string “sysinfo” and right below that command is the version outputted.



21. What is the value of the token passed in frame 3897?

Answer:
b7aad621db97d56771d6316a6d0b71e9&pma%5fusername=root&pma%5fpassword=66Hx8jjG

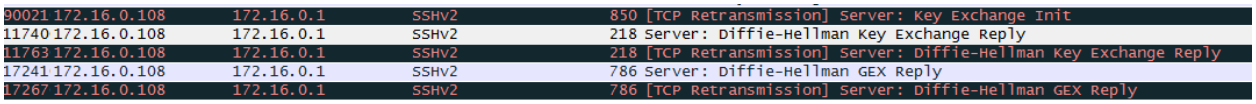
How to solve: Open the provided pcap file in wireshark and browse to the indicated frame and open it up and you will see the token passed.

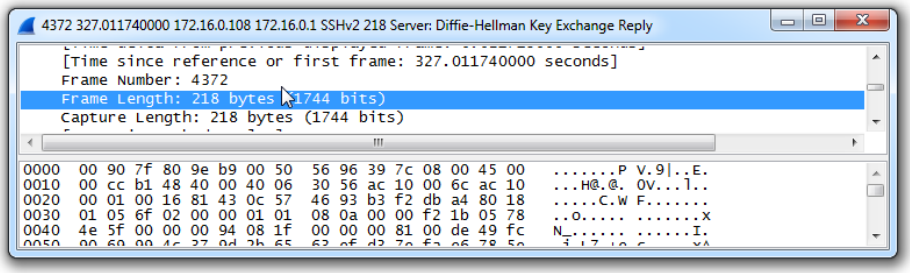


22. What was the frame length size (in bytes) of the server Diffie-Hellman key exchange reply? Example 300

Answer: 218

How to solve: Open the pcap with wireshark and do a string search to find the Diffie-Hellman key exchange reply and the frame length is provided.

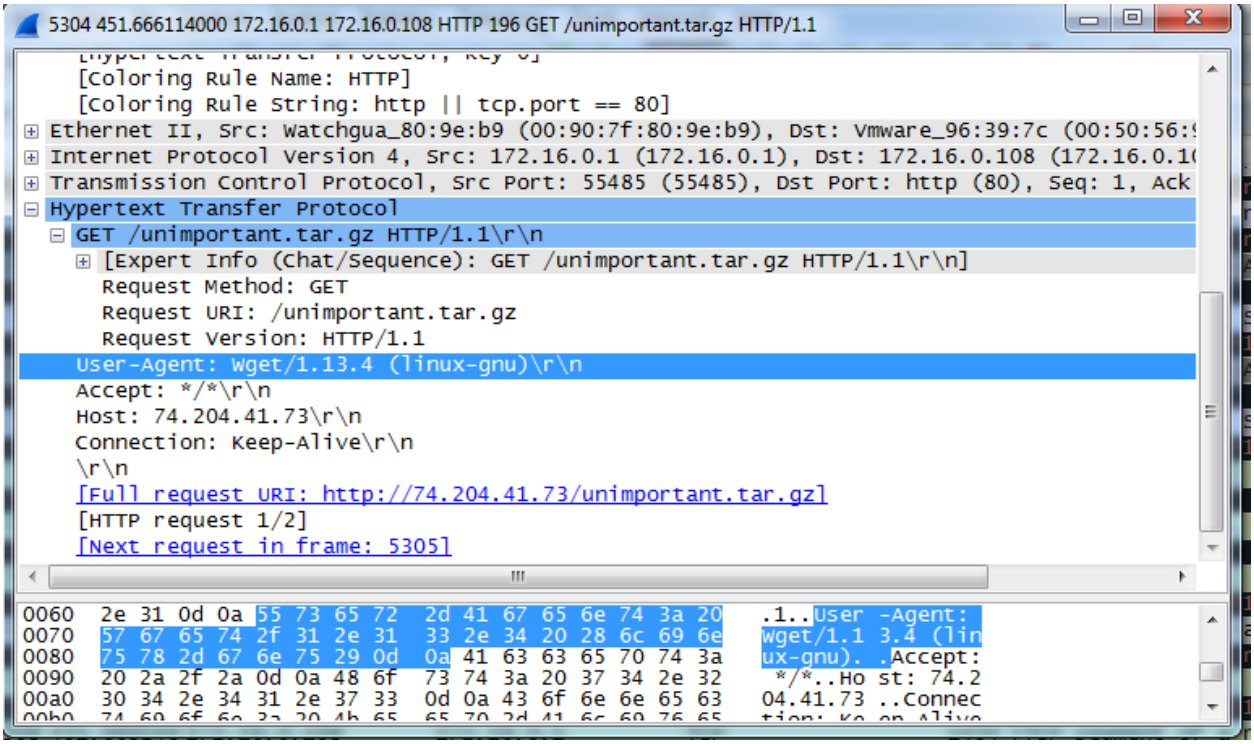




23. What was the command that was used to download a compressed file from the web server? (Only the name of the command)

Answer: wget

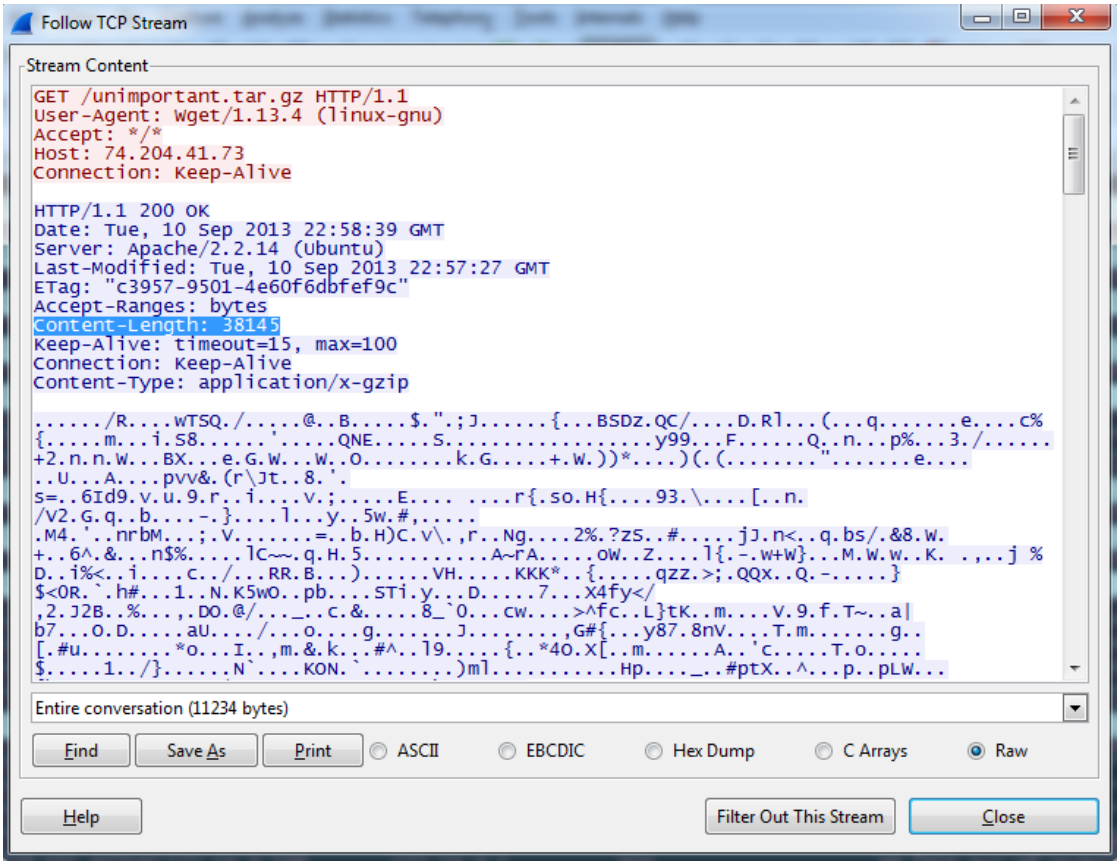
How to solve: In wireshark browse to Frame number 5304 and you will set the “GET /unimportant.tar.gz HTTP/1.1” if you open that frame and browse to Hyper Text Transfer Protocol > Get GET /unimportant.tar.gz HTTP/1.1 > User-Agent: Displays the wget command used to download the file.



24. What is the Content-Length (in bytes) of the file *.tar.gz the attacker created downloaded?

Answer: 38145

How to solve: Use follow the TCP stream on the wget request from the previous question and you will see the content-length reply in blue right after the get request.



25. What is the Mutant key used in the zeus bot?

Answer: 763595725

How to solve: Using volatility you can import the plugin Zeusscan2 (Like question's 11, 13 and 17) and run it against the memory image. The Mutant key is one of the provided fields in the output of this plugin.

26. What is the Parent Process ID of the two 'sh' sessions?

Answer: 1042

How to solve: Just use the linux_pstree plugin and use grep to filter the results for anything related to sh.

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_pstree|grep sh -B 2
Volatile Systems Volatility Framework 2.3_beta
..udevd               461             0
..udevd               462             0
.sshd                 736             0
--
..apache2             1040            33
..apache2             1042            33
...sh                 1274            33
....sh                1275            33
--
.[ext4-dio-unwrit]    287             0
.[kpsmoused]          505             0
.[flush-8:0]          1268            0
```

27. What domain(s) did the attacker configure zeus bot to spy on?

Answer:*.wse.local

28. What is the last two digits of the Mac Address of interface eth0 on the webserver?

Answer: 68

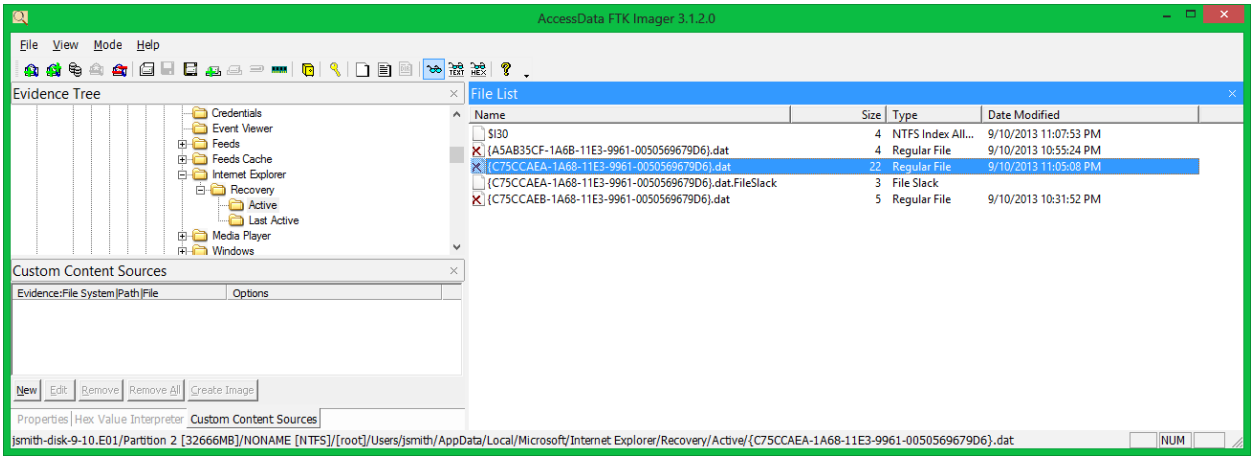
How to solve: Use the linux_ifconfig plugin in volatility on the webserver.vmss and the mac address for eth0 is displayed in the output.

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_ifconfig
Volatile Systems Volatility Framework 2.3_beta
Interface        IP Address           MAC Address        Promiscous Mode
---------------- -------------------- ------------------ ---------------
lo               127.0.0.1            00:00:00:00:00:00  False
eth0             0.0.0.0              00:50:56:96:43:68  False
eth1             172.16.0.108         00:50:56:96:39:7c  False
```

29. Recover the {C75CCAEA-1A68-11E3-9961-0050569679D6}.dat file from the disk image from a deleted value inside Internet Explorer and provide the file size in KB

Answer: 22

How to solve: Use FTK Imager to open the provided .E01 file and browse to C:\Users\jsmith\AppData\Local\Microsoft\Internet Explorer\Recovery\Active\ and the file size is provided in the File list.



30. What is the latency_record_count for PID 1274

Answer: 0

How to solve: Using volatility you must first obtain the offset of the process 1274 linux_pslist then using this offset you use the volshell plugin to run dt (“task_struct”, 0xffff880006dd8000) which will then provide the latency_record_count for the process 1274.

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_pslist |grep 1274
Volatile Systems Volatility Framework 2.3_beta
0xffff880006dd8000 sh                   1274          33             33     0x0000000006d94000 2013-09-10 22:55:40 UTC+0000
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_volshell
Volatile Systems Volatility Framework 2.3_beta
Current context: process init, pid=1 DTB=0x176ba000
Welcome to volshell! Current memory image is:
file:///volatility/webserver.vmss
To get help, type 'hh()'
>>> dt("task_struct",0xffff880006dd8000)
[task_struct task_struct] @ 0xFFFF880006DD8000
0x0   : state                          1
0x8   : stack                          18446612132560019456
0x10  : usage                          18446612132429398032
0x14  : flags                          4202496
0x790 : delays                         0
0x798 : dirties                        18446612132429399960
0x7b0 : latency_record_count           0
0x7b8 : latency_record                 -
0x16b8: timer_slack_ns                 50000
```

31. What the download file name the user launched the eqhay.exe Zeus bot?

Answer: bt.exe

How to solve: Use the iehistory plugin on the windows 7 memory image and grep the ip address of the server the bot communicated with.

```
root@bt:/volatility# python vol.py -f jsmith9-10.mem --profile=Win7SP0x86 iehistory |grep 88.198.6.20 -A 2
Volatile Systems Volatility Framework 2.3_beta
Location: http://88.198.6.20/bt.exe
Last modified: 2013-09-06 01:57:51 UTC+0000
Last accessed: 2013-09-10 22:52:37 UTC+0000
```

32. For the PID 1274 list the first mapped file path

Answer: /bin/dash

How to solve: Use the linux_proc_maps plugin in volatility on the linux memory image on process 1274 and it will show the mapped file paths to that process.

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_proc_maps -p 1274
Volatile Systems Volatility Framework 2.3_beta
Pid      Start              End                Flags               Pgoff Major  Minor  Inode      File Path

-------- ------------------ ------------------ ------ ------------------ ------ ------ ---------- ------------------
-------------------------------------------
    1274 0x0000000000400000 0x0000000000418000 r-x                   0x0      8      1     651536 /bin/dash

    1274 0x0000000000617000 0x0000000000618000 r--               0x17000      8      1     651536 /bin/dash

    1274 0x0000000000618000 0x0000000000619000 rw-               0x18000      8      1     651536 /bin/dash
```

33. What is the registry key name that the running malware wrote in the software\microsoft

Answer: Toavk

How to solve: If you look at the registry key created by the zeusbot that the zeusscan2 plugin outputs you will see the registry values created.

```
root@bt:/volatility# python vol.py -f jsmith9-10.mem --profile=Win7SP0x86 printkey -K "Software\Microsoft\Toavk"
Volatile Systems Volatility Framework 2.3_beta
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Users\jsmith\ntuser.dat
Key name: Toavk (S)
Last updated: 2013-09-10 22:53:21 UTC+0000
```

34. What time was registry key name above last updated?Example 2013-09-01 01:01:01 UTC+0000

Answer: 2013-09-10 22:53:21 UTC+0000

How to solve: Following question 33 you must use the print key plugin on the path "Software\Microsoft\Toavk" that was created by the zeusbot to display the last updated timestamp.

```
root@bt:/volatility# python vol.py -f jsmith9-10.mem --profile=Win7SP0x86 printkey -K "Software\Microsoft\Toavk"
Volatile Systems Volatility Framework 2.3_beta
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Users\jsmith\ntuser.dat
Key name: Toavk (S)
Last updated: 2013-09-10 22:53:21 UTC+0000
```

35. What is the md5hash of the receive.1105.3 file out of the per-process packet queue? flag format in lower case numbers

Answer: 184c8748cfcfe8c0e24d7d80cac6e9bd

How to solve: Use the linux_pkt_queues plugin in volatility on the webserver.vmss image and dump them out to a dir. After that run md5 on file "receive.1105.3" to obtain the md5 hash value.

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 linux_pkt_queues -D /volatility
Volatile Systems Volatility Framework 2.3_beta
Wrote 32 bytes to receive.930.10
Wrote 32 bytes to receive.1105.3
root@bt:/volatility# md5deep -Z receive.1105.3
32      184c8748cfcfe8c0e24d7d80cac6e9bd        184c8748cfcfe8c0e24d7d80cac6e9bd        /volatility/receive.1105.3
```

36. What is the disk Signature at physical offset 0x87cb?

Answer: 14-e8-33-fa

How to solve: Use the mbrparser plugin in volatility on the webserver.vmss image and grep for the requested offset 0x87cb and the Disk Signature is listed right below the potential offset.

```
root@bt:/volatility# python vol.py -f webserver.vmss --profile=LinuxDFIRwebsvrx64 mbrparser |grep 0x87cb -A 1
Volatile Systems Volatility Framework 2.3_beta
Potential MBR at physical offset: 0x87cb
Disk Signature: 14-e8-33-fa
```