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

 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.



 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: 00:00:00.0000000 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

1 0.00000000 Watchgua_80:9e:b9 Broadcast ARP 60 Who has 172.16.0.108? Tell 172.16.0.1		×
Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on inte Interface id: 0 Encapsulation type: Ethernet (1)	rface O	^
Arrival Time: Sep 10, 2013 18:51:07.894237000 Eastern Daylight Time [Time shift for this packet: 0.000000000 seconds] Epoch Time: 1378853467.894237000 seconds [Time delta from previous captured frame: 0.000000000 seconds] [Time delta from previous displayed frame: 0.000000000 seconds] [Time since reference or first frame: 0.000000000 seconds]		~
<	>	
0000 ff ff ff ff ff ff 00 90 7f 80 9e b9 08 06 00 01 0010 08 00 06 04 00 01 00 90 7f 80 9e b9 ac 10 00 01 0020 00 00 00 00 00 00 ac 10 00 6c 00 00 00 00 00 00 00 00 0030 00 00 00 00 00 00 00 00 00 00 00 00 00		

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.

Follow TCP Stream +	×
Stream Content	_
POST /wp-login.php HTTP/1.1 Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/pjpeg, application/x-ms-xbap, */* Referer: http://development.wse.local/wp-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) Content-Type: application/x-www-form-urlencoded Accept-Encoding: gzip, deflate Host: development.wse.local Content Lorath. 67	
Connection: Keep-Alive Cache-Control: no-cache	
<pre>log=Jsmith&pwd=wM812ugu&submit=Login+%C2%BB&redirect_to=wp-admin%2FHTTP/1.1 302 Found Date: Tue, 10 Sep 2013 22:56:30 GMT Server: Apache/2.2.14 (Ubuntu) X-Powered-By: PHP/5.3.2-1ubuntu4.20 Expires: wed, 11 Jan 1984 05:00:00 GMT Last-Modified: Tue, 10 Sep 2013 22:56:30 GMT Cache-Control: no-cache, must-revalidate Pragma: no-cache</pre>	
Set-Cookie: wordpressuser_a5577c39a5e03f6773efea4725288325=Jsmith; expires=Wed, 10- Sep-2014 22:56:30 GMT; path=/wp/blog/ Set-Cookie:	
<pre>wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; expires=wed, 10-Sep-2014 22:56:30 GMT; path=/wp/blog/ Set-Cookie: wordpressuser_a5577c39a5e03f6773efea4725288325=Jsmith; expires=wed, 10- Sep-2014 22:56:30 GMT; path=/ Cookie: Cookie: Cookie</pre>	
wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d;	-
Entire conversation (16235 bytes)	-
Eind Save As Print ASCII EBCDIC Hex Dump C Arrays Raw	
Help Filter Out This Stream Close	

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

🖌 Follow TCP Stream 🔶 🕂 🗖 💌
Stream Content
POST /wp-login.php HTTP/1.1 Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/pjpeg, application/x-ms-xbap, */* Referer: http://development.wse.local/wp-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) Content-Type: application/x-www-form-urlencoded Accept-Encoding: gzip, deflate Host: development.wse.local Content-Length: 67 Connection: Keep-Alive Cache Control: poccache
<pre>log=Jsmith&pwd=wM812ugu&submit=Login+%C2%BB&redirect_to=wp-admin%2FHTTP/1.1 302 Found Date: Tue, 10 Sep 2013 22:56:30 GMT Server: Apache/2.2.14 (Ubuntu) X-Powered-By: PHP/5.3.2-Jubuntu4.20 Expires: Wed, 11 Jan 1984 05:00:00 GMT Last-Modified: Tue, 10 Sep 2013 22:56:30 GMT Cache-Control: no-cache, must-revalidate Pragma: no-cache Set-Cookie: wordpressuser_a5577c39a5e03f6773efea4725288325=Jsmith; expires=Wed, 10- Sep-2014 22:56:30 GMT; path=/wp/blog/</pre>
Set-Cookie: wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; expires=wed, 10-sep-2014 22:56:30 GMT; path=/wp/blog/ Set-Cookie: wordpressuser_a5577c39a5e03f6773efea4725288325=Jsmith; expires=wed, 10- sep-2014 22:56:30 GMT; path=/ Set-Cookie: wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d; wordpresspass_a5577c39a5e03f6773efea4725288325=d1a75ce7d9745ad470720f0bd68ea02d;
<u>Find</u> Save <u>As</u> <u>Print</u> ASCII O EBCDIC O Hex Dump O C Arrays • Raw
Help Filter Out This Stream Close

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?

🔲 🛨 🚹 grrcon-v	virtual- 172	.16.0.109	74.125.225.1	12 ET T	ROJAN	Zeus Bot G	ET to Google c	necking Interr	iet connectivit	у	09/11/2013
IP Header Informat	tion					Perform N	ass Classificatior	Packet Cap	oture Options	Event Export Op	tions Permalink
Source	Desti	nation	Ver	Hlen	Tos	Len	ID	Flags	Off TTL	Proto	Csum
172.16.0.109	74.1	25.225.112	4	5	0	307	4518	0	0 128	6	4020
Signature Informat	tion										
Generator ID Sig	g. ID Si	ig. Revision	Activity (247/7	'13)			Category	Sig Info			
1 20	013076 6			3	4.64%		trojan-activity	Query	Signature Dat	tabase	View Rule
TCP Header Inform	ation										
Src Port D:	st Port	Seq		Ack		Off	Res	Flags	Win	Csum	URP
49483 80	0	320461524	;	2288639208		5	0	24	16450	59707	0
References											
Type Valu	ıe										
url www	w.securework	s.com/research/thr	reats/zeus/?tl	nreat=zeus							_
Payload											Hex Ascii
0000000: 47 45 5 000011: 74 3a 2 000034: 0a 55 7 000004E: 63 6f 6 000006E: 6f 77 7 000009C: 2c 4a 4 000009C: 2c 4a 4 000009C: 4c 53 2 0000001: 2c 53 4 00000EA: 2c 53 6 0000104: 63 68 6	4 20 2f 77 0 2a 2f 2a 3 65 72 2d d 70 61 74 3 20 4e 64 3 32 3b 20 5 54 20 43 0 33 2e 30 0 33 2e 30 20 36 2e f 6d 0d 0a 5 0d 0a 0d	65 62 68 70 20 0d 0a 43 6f 6e 41 67 65 6e 74 69 62 6c 65 3b 20 36 2e 31 3b 2e 4e 45 54 20 4c 52 20 33 2e 2e 33 30 37 32 30 29 0d 0a 48 43 61 63 68 65 0a	48 54 5 6e 65 6 3a 20 4 20 4d 5 20 54 5 33 2c 3 33 3c 2e 3 35 2e 3 35 2e 3 6f 73 2 2d 43 6	4 50 2f 31 3 74 69 6f d 6f 7a 69 3 49 45 20 2 69 64 65 2 20 32 2e 3 30 37 32 0 4d 65 64 4 3a 20 77 f 6e 74 72	2e 31 6e 3a 6c 6c 37 2e 6e 74 30 2e 39 361 77 77 6f 6c	Od Oa 41 20 43 60 61 2f 34 30 3b 20 2f 34 2e 35 30 37 20 2e 4e 20 43 65 2e 67 6f 3a 20 6e	63 63 65 70 6f 73 65 0d 2e 30 20 28 57 69 6e 64 30 3b 20 53 30 3b 20 53 32 37 3b 20 45 54 20 43 6e 74 65 72 6f 67 66 361	GET./webh t:.*/*(.User-Age compatibl) ows.NT.6. LCC2;NE .NET.CLR. LR.3.0.30 .PC.6.0) .com.Cac che	<pre>up.HTTP/1.1 Connection: ant:.Mozill le;.MSIE.7. 1;.Trident ST.CLR.2.0. 3.5.30729; 0729;.Media .Host:.www. che-Control</pre>	Accep .Close. a/4.0.(0;.Wind /4.0;.S 50727;. NET.C Center google :.no-ca	
Notes This event currently ha Add A Note To This	s zero notes s Event	- You can add a no	ote by clickin	g the button I	below.						

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 4^{th} 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

volucite Systems ve	fulle for the second se				
0xleaa8a88 TCPv4	0.0.0:30639	0.0.0.0:0	LISTENING	996	<pre>taskhost.exe</pre>
0xleaa8a88 TCPv6	:::30639	:::0	LISTENING	996	<pre>taskhost.exe</pre>
0xled29580 TCPv4	0.0.0:30639	0.0.0.0:0	LISTENING	996	<pre>taskhost.exe</pre>
0x1e735cd8 TCPv4	172.16.0.109:49497	88.198.6.20:80	CLOSE WAIT	996	<pre>taskhost.exe</pre>

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.

<pre>root@bt:/volatility# Volatile Systems Vola</pre>	python vol.pyplugins=contrib/plugins/malware/ -f jsmith9-10.memprofile=Win7SP0x86 zeusscan2 atility Framework 2.3 beta
*****	Act::::::::::::::::::::::::::::::::::::
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	: Awmugaleg
Value 3	: Misevoteo
Executable	: Biga\eghay.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 <u>http://88.198.6.20/cf.bin</u> which provides the config file the zeusbot downloaded from the CNC server.

Volatile Systems Vola	atility 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	: Awmugaleg
Value 3	: Misevoteo
Executable	: Biga\eghay.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	LISTENING	996	<pre>taskhost.exe</pre>
0x1eaa8a88 TCPv6	:::30639	:::0	LISTENING	996	<pre>taskhost.exe</pre>
0x1ed29580 TCPv4	0.0.0.30639	0.0.0:0	LISTENING	996	<pre>taskhost.exe</pre>
0x1e735cd8 TCPv4	172.16.0.109:49497	88.198.6.20:80	CLOSE WAIT	996	<pre>taskhost.exe</pre>

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.

8				NetworkMiner 1.5				↔	-	
File Tools Help										
Select a network adapter	in the list							~	Start	Stop
Hasta (45) Emmas (C1ar)	Eles (117)	Images (46)	Manager	Credentials (16) Sessions (76) DNS (41) Removies (1295) Knowed a Class	4	line.	Case Pa	nel		
Hosts (40) Frames (6100)	Files (117)	inages (40)	Messages	Made Deserved	text Anoma	iiies	Filename	MD5		
Show Cookies V S		challenge-res	ponse 🔄	Mask Passwords			GrrCON	40fe066.		
Client	Server	Protocol	Usemame	Password	Valid login	Login timestamp				
172.16.0.109 (Windows)	204.79	HTTP C	SRCHD	N/A	Unknown	9/10/2013 6:52:37 PM				
172.16.0.1 (Windows)	172.16	HTTP P	root	66Hx8jjG	Unknown	9/10/2013 6:55:08 PM				
172.16.0.1 (Windows)	172.16	HTTP C	pNNiak	sDUUzkS%2FZTw%3D	Unknown	9/10/2013 6:55:08 PM				
172.16.0.1 (Windows)	172.16	HTTP C	phpMyA	N/A	Unknown	9/10/2013 6:55:08 PM				
172.16.0.109 (Windows)	172.16	HTTP P	Jsmith	wM812ugu	Unknown	9/10/2013 6:56:29 PM				
172.16.0.109 (Windows)	172.16	HTTP C	Jsmith	d1a75ce7d9745ad470720f0bd68ea02d	Unknown	9/10/2013 6:56:29 PM				
172.16.0.109 (Windows)	172.16	HTTP C	wordpre	N/A	Unknown	9/10/2013 6:56:29 PM				
172.16.0.109 (Windows)	172.16	HTTP C	Jsmith,J	d1a75ce7d9745ad470720f0bd68ea02d,d1a75ce7d9745ad470720f0bd68ea02d	Unknown	9/10/2013 6:56:35 PM				
172.16.0.109 (Windows)	172.16	HTTP C	wp-post	N/A	Unknown	9/10/2013 6:56:35 PM				
172.16.0.1 (Windows)	172.16	HTTP P	Jsmith	wM812ugu	Unknown	9/10/2013 6:59:58 PM				
172.16.0.1 (Windows)	172.16	HTTP C	Jsmith	d1a75ce7d9745ad470720f0bd68ea02d	Unknown	9/10/2013 6:59:58 PM				
172.16.0.1 (Windows)	172.16	HTTP C	wordpre	N/A	Unknown	9/10/2013 6:59:58 PM				
1/2.16.0.1 (Windows)	1/2.16	HTTPC	Jsmith,J	wM812ugu	Unknown	9/10/2013 7:03:50 PM				
172.16.0.1 (Windows)	172.16	HTTPC	wordpre	N/A 1-D-124-	Unknown	9/10/2013 7:03:50 PM				
172.16.0.1 (Windows)	172.16	HTTPC	Jsmith,J	IqBeJ2Az	Unknown	9/10/2013 7:04:04 PM				
172.16.0.1 (windows)	1/2.16	HITPU	wordpre	N/A	Unknown	9/10/2013 7:04:04 PM				
								Paland C	en Elen	
L								nei0ad Ca	ise riles	
Live Sniffing Buffer Usage	8									

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.

<mark>root@bt:/volatility#</mark> Volatile Systems Vola	python vol.pyplugins=contrib/plugins/malware/ -f jsmith9-10.memprofile=Win7SP0x86 zeusscan2 tility 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
                   sh -c /bin/sh
             33
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.

Filter:	Fxpression Clear Apply Save	2
lo. Time Source	🚄 408 24.273419000 172.16.0.1 172.16.0.108 UDP 342 Source port: 39709 Destination port: 31708	
387 24.0701620 172.16.0.1 388 24.0701990 172.16.0.108	□ User Datagram Protocol, Src Port: 39709 (39709), Dst Port: 31708 (31708)	SS=536 SACK_PERM=1 TSva]=4294967295 TSecr=0 wS=1024 0 SACK PERM=1 TSva]=31685 TSecr=4294967295 wS=32
389 24.0702040 172.16.0.108	Source port: 39709 (39709)	-5792 Len=0 M55=1460 SACK_PERM=1 TSval=31685 TSecr-
390 24.0710570 172.16.0.1 391 24.0710670 172.16.0.1	Length: 308	
392 24.1796740 172.16.0.1	(III))) () () () () () () ()	Len=0 MS5=265 SACK_PERM=1 TSval=4294967295 TSecr=0
393 24.1796960 172.16.0.1 394 24.1799020 172.16.0.108	0020 00 6c 9b 1c 7b dc 01 34 24 95 43 43 43 43 43 43 43 . 1{	MSS=265 SACK_PERM=1 TSval=4294967295 TSecr=0 0 SACK_PERM=1 TSval=31696 TSecr=4294967295
395 24.1799100 172.16.0.108 396 24.1809020 172.16.0.1	0040 43 43 43 43 43 43 43 43 43 43 43 43 43	-5792 Len=0 MSS=1460 SACK_PERM=1 TSval=31696 TSecr=
397 24.1809190 172.16.0.1	0060 43 43 43 43 43 43 43 43 43 43 43 43 43	
398 24.2132/30 1/2.16.0.1 399 24.2132930 172.16.0.1	172.16.0.108 ICMP 162 Echo (ping) request id=0xaf17, seq=295/9985, ttl=49 (r	eply in 400)
400 24.2133530 172.16.0.108 401 24.2133610 172.16.0.108	172.16.0.1 ICMP 162 Echo (ping) reply id=0xaf17, seq=295/9985, tt]=64 (r 172.16.0.1 ICMP 162 Echo (ping) reply id=0xaf17, seg=295/9985, tt]=64	equest in 399)
402 24.2144980 172.16.0.1	172.16.0.108 ICMP 190 Destination unreachable (Protocol unreachable)	
403 24.2145180 172.16.0.1 404 24.2464430 172.16.0.1	172.16.0.108 ICMP 190 Destination unreachable (Protocol unreachable) 172.16.0.108 ICMP 192 Echo (ping) request id=0xaf18, seq=296/10241, tt]=45	
405 24.2464630 172.16.0.1	172.16.0.108 ICMP 192 Echo (ping) request id=0xaf18, seq=296/10241, tt]=45 (reply in 406)
406 24.2464700 172.16.0.108 407 24.2464760 172.16.0.108	172.16.0.1 ICMP 192 ECHO (ping) Feply 1d=0Xaf18, Seq=296/10241, tt=64 (172.16.0.1 ICMP 192 ECho (ping) reply id=0xaf18, seq=296/10241, tt]=64	request in 405)
408 24.2734190 172.16.0.1 409 24.2734400 172.16.0.1	172.16.0.108 UDP 342 Source port: 39709 Destination port: 31708 172.16.0.108 UDP 342 Source port: 39709 Destination port: 31708	

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.

Follow TCP Stream
- Stream Content
<pre>core_loadlib)94606415251250857507022509322064#stdapi_fs _file_expand_pathstdapi_fs_delete_dirstdapi_fs_mkdirstdapi_fs_lsstdapi_fs_lsstdapi_fs_lsstdapi_fs_delete_filestdapi_fs_bracketstdapi_fs_delete_filestdapi_fs_lsstdapi_fs_delete_filestdapi_sys_config_getuit dstdapi_sys_config_sysinfo#stdapi_fs_bracketstdapi_sys_process_execute!stdapi_sys_process_get process_close]stdapi_sys_process_get_processes"stdapi_sys_process_get process_close]stdapi_sys_config_getuid]931380218727542142651432 85474506www-data (33)</pre>
9
x86_64
root
root[wsbrthrdq/0][wsbrthrdq/0]
root[ksort1rqd/1][ksort1rqd/1]k@ root[watchdog/1][watchdog/1]M@ root[migration/2][migration/2]M@
172.16.0.108:krb524 → 172.16.0.1:43614 (17859 bytes)
Eind Save As Print ASCII EBCDIC Hex Dump C Arrays 🖲 Raw
Help Filter Out This Stream

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.

1	L	• •	-94		+1	- L -											
	[Nex	tr	equ	est	in	fr	ame	: 38	98]								
🗆 Li	ne-b	ase	d t	ext	dat	ta:	ap	plic	ati	on/:	x-w	ww-1	For	m-u	rle	nco	oded
	toke	n=b	7aa	62	1db	97d	567	71d6	316	a6d	0b7:	Le9	±	a%5'	fuse	ern	name=root&pma%5fpassword=66Hx8jjG
													· ·				· · ·
0050	- 78	2e	70	68	70	20	48	54	54	50	2f	31	2e	31	0d	0a	ι x.php אТ тP/1.1
0060	48	6f	73	74	Зa	20	37	34	2e	32	30	34	2e	34	31	2e	Host: 74 .204.41.
0070	37	33	0d	0a	55	73	65	72	2d	41	67	65	6e	74	3a	20) 73User -Agent:
0080	4d	6f	7a	69	6c	6c	61	2f	34	2e	30	20	28	63	6f	6d	l Mozilla/ 4.Ō (com
0090	70	61	74	69	62	6c	65	3b	20	4d	53	49	45	20	36	2e	patible; MSIE 6.
00a0	30	3b	20	57	69	6e	64	6f	77	73	20	4e	54	20	35	2e	e Ö; Windo ws NT 5.
00b0	31	29	0d	0a	43	6f	6e	74	65	6e	74	2d	54	79	70	65	5 1)Cont ent-Type
00c0	3a	20	61	70	70	6c	69	63	61	74	69	6f	6e	2f	78	2d	l : applic ation/x-
00d0	77	77	77	2d	66	6f	72	6d	2d	75	72	6c	65	6e	63	6f	www-form -urlenco
00e0	64	65	64	0d	0a	43	6f	6e	74	65	6e	74	2d	4c	65	6e	e dedCon tent-Len
00f0	67	74	68	3a	20	38	32	0d	0a	0d	0a	74	6f	6b	65	6e	gth: 82 <mark>token</mark>
0100	3d	62	37	61	61	64	36	32	31	64	62	39	37	64	35	36	=b7aad62 1db97d56
0110	37	37	31	64	36	33	31	36	61	36	64	30	62	37	31	65	771d6316 a6d0b71e
0120	39	26	70	6d	61	25	35	66	75	73	65	72	бe	61	6d	65	9&pma%5f username
0130	3d	72	6f	6f		26	70	6d	61	25	35	66	70	61	73	73	=root±_a%5fpass
0140	77	6f	72	64	3d	36	36	48	78	38	бa	6a	47				word=66H x8jjG
-																	

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.

90021 172.16.0.108	172.16.0.1	SSHv2	850 [TCP Retransmission] Server: Key Exchange Init
11740 172.16.0.108	172.16.0.1	SSHv2	218 Server: Diffie-Hellman Key Exchange Reply
11763 172.16.0.108	172.16.0.1	SSHV2	218 [TCP Retransmission] Server: Diffie-Hellman Key Exchange Reply
17241 172.16.0.108	172.16.0.1	SSHv2	786 Server: Diffie-Hellman GEX Reply
17267 172.16.0.108	172.16.0.1	SSHv2	786 [TCP Retransmission] Server: Diffie-Hellman GEX Reply
C	-		
	4372 327.011740000 172.16.0.1	08 172.16.0.1 SSHv2 218	18 Server: Diffie-Hellman Key Exchange Reply
	[Time since referen Frame Number: 4372	nce or first fram	ume: 327.011740000 seconds]
	Frame Length: 218	oytes 1744 bits)	
	Capture Length: 210	bytes (1/44 bit	
	< [1 I I I I I I I I I I I I I I I I I I I
	0000 00 90 7f 80 9e b9 0010 00 cc b1 48 40 00	00 50 56 96 39 40 06 30 56 ac	9 7c 08 00 45 00P V.9 E. c 10 00 6c ac 10H@.@. 0V]
0	020 00 01 00 16 81 43 030 01 05 6f 02 00 00	0c 57 46 93 b3 01 01 08 0a 00	3 f2 db a4 80 18C.W F 0 00 f2 1b 05 78x
	0040 4e 5T 00 00 00 92	08 1T 00 00 00	0 81 00 0e 49 TC N NI.

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.

🚄 5304 451.666114000 172.16.0.1 172.16.0.108 HTTP 196 GET /unimportant.tar.gz HTTP/1.1										
[Coloring Rule Name: HTTP]										
[Coloring Rule String: http tcp.port == 80]										
⊞ Ethernet II, Src: Watchgua_80:9e:b9 (00:90:7f:80:9e:b9), Dst: Vmware_96:39:7c (00:50:56:										
Internet Protocol Version 4, Src: 172.16.0.1 (172.16.0.1), Dst: 172.16.0.108 (172.16.0.1)										
⊕ Transmission Control Protocol, Src Port: 55485 (55485), Dst Port: http (80), Seq: 1, Ack										
Hypertext Transfer Protocol										
GET /Unimportant.tar.gz HITP/1.1\r\n										
Request Method: GET										
Request URI: /unimportant.tar.gz										
Request Version: HTTP/1.1										
User-Agent: Wget/1.13.4 (linux-gnu)\r\n										
Accept: */*\r\n										
Host: 74.204.41.73\r\n	-									
Connection: Keep-Alive\r\n										
[Full request URT: http://74 204 41 73/unimportant tar gz]										
[HTTP request 1/2]										
[Next request in frame: 5305]	-									
4										
0060 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 .1User -Agent:										
0070 57 67 65 74 2f 31 2e 31 33 2e 34 20 28 6c 69 6e Wget/1.1 3.4 (lin										
0080 75 78 2d 67 6e 75 29 0d 0a 41 63 63 65 70 74 3a ux-gnu). Accept:										
0030 20 2a 2r 2a 00 0a 48 6r /3 /4 3a 20 3/ 34 2e 32 */*Ho st: /4.2 00a0 30 34 2e 34 31 2e 37 33 0d 0a 43 6f 6e 6e 65 63 04.41.73Connec										
10060 74 60 6f 60 25 20 4h 65 65 70 2d 41 6c 60 76 65 tion. Ko on Alivo										

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.

Follow TCP Stream	
Stream Content	
GET /unimportant.tar.gz HTTP/1.1 User-Agent: wget/1.13.4 (linux-gnu) Accept: */* Host: 74.204.41.73 Connection: Keep-Alive	
HTTP/1.1 200 OK Date: Tue, 10 Sep 2013 22:58:39 GMT Server: Apache/2.2.14 (Ubuntu) Last-Modified: Tue, 10 Sep 2013 22:57:27 GMT ETag: "C3957-9501-4e60f6dbfef9c" Accept-Ranges: bytes Content-Length: 38145 Keep-Alive: timeout=15, max=100 Connection: Keep-Alive Content-Type: application/x-gzip	
<pre>/RwT5Q./@.B\$.";J{.BSDZ.QC/D.Rl(qeC% {m.i.S8QNES</pre>	
Entire conversation (11234 bytes)	
Help Filter Out This Stream Close	

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.

<pre>root@bt:/volatility#</pre>	python vol.py	-f webserver.vmss	profile=LinuxDFIRwebsvrx64	linux pstree grep	sh-B2
Volatile Systems Vola	atility Framewo	ork 2.3 beta			
udevd	461	Θ			
udevd	462	Θ			
.s <mark>sh</mark> d	736	Θ			
apache2	1040	33			
apache2	1042	33			
sh	1274	33			
sh	1275	33			
.[ext4-dio-unwrit]	287	Θ			
.[kpsmoused]	505	Θ			
.[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
                                     00:50:56:96:43:68
                0.0.0.0
                                                       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.

<pre>root@bt:/volatility# python vol.py -f v Volatile Systems Volatility Framework ?</pre>	webserver.vmssprofile=LinuxDFI 2.3 beta	Rwebsvrx64 linux_pslist g	rep 1274
0xffff880006dd8000 sh	1274 33	33 0x00000006d94000	2013-09-10 22:55:40 UTC+0000
<pre>root@bt:/volatility# python vol.py -f </pre>	webserver.vmssprofile=LinuxDFI	Rwebsvrx64 linux_volshell	
Volatile Systems Volatility Framework	2.3_beta		
Welcome to velshell! Current memory im			
file:///volatility/webserver.vmss	age is.		
To get help, type 'hh()'			
>>> dt("task_struct",0xffff880006dd800	0)		
[task_struct task_struct] @ 0xFFFF8800	06DD8000		
0x0 : state	1		
	18446612132300019430		
0x14 : flags	4202496		
· - · ·			
0x790 : delays		0	
0x798 : dirties		18446612	132429399960
0x7b0 : latency re	cord count	Θ	
0.71.0 1.1			
UX/DX : Latency re	cora	-	
Av16b8, timor clac	k nc	50000	
ovropo. crillel_stac	K_115	20000	

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.

<mark>root@bt</mark> : Volatile	/volatility# python Systems Volatility	vol.py -f webserver Framework 2.3 beta	r.vmss	profile=LinuxDFIR	websvrx	64 linu	x_proc_maps	-p 1274
Pid	Start	End	Flags	Pgoff	Major	Minor	Inode	File Path
1274	0×000000000400000	0x000000000418000	r-x	0×0	8	1	651536	/bin/dash
1274	0×000000000617000	0×000000000618000	r	0×17000	8	1	651536	/bin/dash
1274	0×000000000618000	0×000000000619000	rw-	0×18000	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.

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 /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
```