**PCAP File Analysis**

**Task 2, Report 1**

**Performed by Julie Haynes**

A PCAP file was retrieved (provided to me) called Suspicious+Traffic.pcap

Logged on to the Linux server that was provided to me. Snort was already installed on the server.

Downloaded WinSCP and set it to connect to port 22

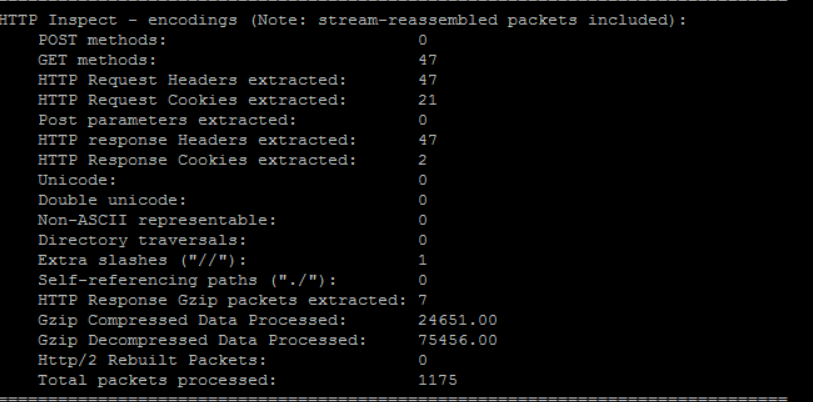
Copied the PCAP file to my D directory on my VM and then uploaded it to WinSCP (drag and drop)

Opened PuTTY and connected to my Linux serving using SSH.

Initially I created my own /etc/snort directory in my /home/phantom4814 directory (so the full path name was /home/phantom4814/etc/snort) and placed the PCAP file there but I was unable to run snort. I was using the command: “sudo snort -c snort.conf -r Suspicious+Traffic.pcap”

Then I navigated back to the root directory on my Linux server and saw that there was already a /etc/snort directory. So I deleted the directory I had created and moved a copy of the PCAP file to the /etc/snort directory. Once I did that I was able to run snort on the PCAP file immediately using the sudo command.

Some of the output I observed that I found interesting was the number of POST and GET methods, as well as the total packets processed.:



Next, still in PuTTY, I navigated to the alert file that was created by snort in /var/log/snort. I open the file using “vi alert” and viewed the contents in PuTTY. I downloaded a copy to my vm drive and opened it in notepad, but was hard to read so I copied the contents directly from PuTTY to notepad++ on my PC and was much easier to read:

**ALERT 1**

[\*\*] [1:2017552:6] ET CURRENT\_EVENTS Cushion Redirection [\*\*]

[Classification: A Network Trojan was detected] [Priority: 1]

02/15-23:33:45.849418 192.168.137.81:49201 -> 108.178.15.187:80

TCP TTL:48 TOS:0x0 ID:59923 IpLen:20 DgmLen:569 DF

\*\*\*A\*\*\*\* Seq: 0x2258C79F Ack: 0xF9E0F2D4 Win: 0x3D80 TcpLen: 20

[Xref => http://malwaremustdie.blogspot.co.uk/2013/09/302-redirector-new-cushion-attempt-to.html]

**ALERT 2**

[\*\*] [120:3:1] (http\_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE [\*\*]

[Classification: Unknown Traffic] [Priority: 3]

02/15-23:33:51.712406 108.178.15.187:80 -> 192.168.137.81:49212

TCP TTL:48 TOS:0x0 ID:57355 IpLen:20 DgmLen:1409 DF

\*\*\*A\*\*\*\* Seq: 0x95EF0ACE Ack: 0x2AD82B1F Win: 0x7B TcpLen: 20

**ALERT 3**

[\*\*] [1:2014726:86] ET POLICY Outdated Windows Flash Version IE [\*\*]

[Classification: Potential Corporate Privacy Violation] [Priority: 1]

02/15-23:33:54.010625 192.168.137.81:49212 -> 108.178.15.187:80

TCP TTL:48 TOS:0x0 ID:57368 IpLen:20 DgmLen:428 DF

\*\*\*A\*\*\*\* Seq: 0x2AD82B1F Ack: 0x95EF5261 Win: 0x4180 TcpLen: 20

[Xref => http://www.adobe.com/software/flash/about/]

**ALERT 4**

[\*\*] [1:2020311:10] ET CURRENT\_EVENTS DRIVEBY Nuclear EK SWF M2 [\*\*]

[Classification: A Network Trojan was detected] [Priority: 1]

02/15-23:33:54.669953 108.178.15.187:80 -> 192.168.137.81:49212

TCP TTL:128 TOS:0x0 ID:1099 IpLen:20 DgmLen:8831 DF

\*\*\*A\*\*\*\* Seq: 0x95EF5261 Ack: 0x2AD82CA3 Win: 0x0 TcpLen: 20

**ALERT 5**

[\*\*] [1:2020317:5] ET CURRENT\_EVENTS DRIVEBY Nuclear EK SilverLight M2 [\*\*]

[Classification: A Network Trojan was detected] [Priority: 1]

02/15-23:33:56.058965 108.178.15.187:80 -> 192.168.137.81:49213

TCP TTL:128 TOS:0x0 ID:1108 IpLen:20 DgmLen:15935 DF

\*\*\*A\*\*\*\* Seq: 0xE7E0447F Ack: 0x3D2A12C6 Win: 0xE800 TcpLen: 20

**ALERT 6**

[\*\*] [1:2019873:3] ET CURRENT\_EVENTS DRIVEBY Nuclear EK Payload [\*\*]

[Classification: A Network Trojan was detected] [Priority: 1]

02/15-23:33:57.418697 108.178.15.187:80 -> 192.168.137.81:49212

TCP TTL:128 TOS:0x0 ID:1117 IpLen:20 DgmLen:16778 DF

\*\*\*A\*\*\*\* Seq: 0x95EF778C Ack: 0x2AD82E90 Win: 0x0 TcpLen: 20

**ALERT 7**

[\*\*] [139:1:1] (spp\_sdf) SDF Combination Alert [\*\*]

[Classification: Senstive Data] [Priority: 2]

02/15-23:33:57.938005 108.178.15.187 -> 192.168.137.81

PROTO:254 TTL:128 TOS:0x0 ID:1129 IpLen:20 DgmLen:20 DF

I added a header (Alert1 – Alert7) for ease of reference. Observations:

* **Alert 1 was a SID 1:2017552:6**. This event is described as “ET Current Events Cushion Redirection” which is a cusion that is used to evade web filtration/URL of IDS/IPS alerts for web driven attacks design by the actors behind it.
* Alert 1 shows detection of a trojan from 192.168.137.81:49201 -> 108.178.15.187:80 while Alert 4, Alert 5, and Alert 6 show detection of a trojan in the opposite direction 108.178.15.187:80 -> 192.168.137.81:49212. Having read the blog at <http://blog.malwaremustdie.org/2013/09/302-redirector-new-cushion-attempt-to.html> (which I found by googling the SID# and title before I noticed it in the alert, I guessed that Alerts 4, 5, and 6 are where the user is clicking link after link following the cushion redirection from Alert 1. I found several references to a group called Windigo who moved from a malware called Rig EK to Nuclear EK with a malware payload that was being XOR-ed. Reference a 2014 analysis at <https://www.malware-traffic-analysis.net/2014/12/10/index.html> and again in 2015 at <https://www.malware-traffic-analysis.net/2015/01/01/index.html>. I learned that Windigo is a code name for a group that may have started as early as 2011. Windigo compromises legitimae traffic to servers to redirect visotors to an exploit kit (EK). If the windows client is vulneralbe, it coulcd be infected by the EK. Windog started using the Nuclear EK to deliver its payload in December 2014. Eks are used to load malware without user consent and without tipping user off the something suspicious is happening. Nuclear EK can use a vast selection of attacks including Flash, Silverlight, PDF, and Internate Explorer. Common ways that Nuclear EK redirects users and infects systems include:

1. Compromised ad servers
2. Drive-by downloads
3. Code injection

It is common for populare ad servers to redirect traffice to affiliate networks, usually through ads that are placed on the website’s page. A common method for these malicious ads is very simple and works like this:

1. The ad lands on the page and it contains a few lines of codes, often as few as 3 lines of code.
2. The code will create a link just like you click to go to any page on the web, then it waits 7 seconds before triggering a click on the link which causes the browser to redirect you. It is believe that the 7 second wait is to avoid security tools that are actively scanning sites for ads like these.

* In 2018 WordPress experienced a large compromise caused by Nuclear EK. Many Wordpress sites were inject with rogue code that performed a silent redirection to domains that appeared to be hosting adds, but th ad was stuffed with more code that sent visitors to the Nuclear Exploit Kit.
* XOR cipher is a type of additive ciher, an encryption algorithm where a string of text can be encrypted by applying the bitwise XOR operator to every character using a given key. To decrypt the output, merely reappling the XOR function with the key will remove the cipher.
* <https://isc.sans.edu/forums/diary/A+recent+decline+in+traffic+associated+with+Operation+Windigo/20065/>
* **Alert 2 was a SID 120:3:1**. This event is described as “No Content-Length or Transfer-Encoding in HTTP response” and it is generated when the http\_inspect preprocessor detects anomalous network traffic. The impact is unkown but it is an indication that there has been some type of anomalous or non-standard behaviour between the systems/assets on the network. Using the (()) configuration options can control this type of event. There are no known false positives or false negatives and it is rated as “simple” in ease of attack.
* **Alert 3 was a SID 1:2014726:86** This event is described as “ET POLICY Outdated Windows Flash Version IE”. I found an article from Fireeye that provides a detailed explanation of how the Nuclear EK incorporated an exploit for Adobe Flash: <https://www.fireeye.com/blog/threat-research/2015/03/cve-2015-0336_nuclea.html>
* **Alert 5 was a SID 1:2020317:5** This event is described as “ET CURRENT\_EVENTS DRIVEBY Nuclear EK SilverLight M2”. In 2013 the Nuclear EK evolved to include an exploit that leverages Silverlight. At that time not many security solutions had detections for Silverlight exploitation. The Nuclear EK for Silverlight will check if the system has Silverlight installed and if so, it attempts to use the Silverlight exploit to drop malware into the system.
* **Alert 6 was a SID 1:2019873:3**. This event is described as “ET CURRENT\_EVENTS DRIVEBY Nuclear EK Payload”. I assume this indicates that the exploit has been successful and the EK has delivered its payload to infect the host. I don’t yet know what the payload is – a file downloader that retrieves other malware, the intended malware itself, an encrypted binary that will be decrypted and executed. Most common payloads are ransomeware and Nuclear EK is known to be ransomware, but other payloads include botnet malware, information stealers, and banking Trojans just to name a few.
* **Alert 7 was a SID 139:1:1**. This event is described as “(spp\_sdf) SDF Combination Alert”. Snort.org showed a “Missing documentation” page for SID # 139.1. But I found some info on “SDF Combination Alert” for Snort. This info indicated that it is a “Sensitive Data Preprocessor” which is a Snort moduel that performs detection and filtering of Personally Identifiable Informatin (PII) such as credit card numbers, U.S. social security numbers, and email addresses.

In summary I found an interesting article about the group that rents out the Nuclear EK and uses Locky ransomware to make money from the user of the infected host: <https://blog.checkpoint.com/2016/05/17/inside-nuclears-core-unraveling-a-ransomware-as-a-service-infrastructure/>

**Wireshark**

I customized my view by adding the following columns based on a video and some other reading:

1. Source port (unresolved)
2. Destination port (unresolved)
3. Packet Length
4. Cumulative Bytes
5. Host

Questions from Assignment:

1. Now it’s time to start building our analysis:
   1. IP address of the laptop: **192.168.137.81**
   2. Host name of the laptop: **Barto-PC** (I located by looking up the LLMNR packet for 192.168.137.81 and then checking in the queries portion of the packet, which confirmed what was posted in the Info column)
   3. MAC address of the laptop: **Dell\_6a:bd:22 (5c:f9:dd:6a:bd:22)** (I located this in the Ethernet II portion of the packet)
   4. Dateand time of the compromise: **February 15, 2015 at 17:33 (5:30 PM)**
2. If you have determined that an Exploit Kit (EK) was involved:  (Filtered on DNS Protocol for much of this)
   1. Identify the IP address(es) and host(s) that generated the EK traffic.
      * **ts.daumcdn.net at 151.249.94.142 and 151.249.94.140 (both in UK)**
      * **inhye01.itpr.co.kr at 211.111.221.233 (in Korea)**
      * **bnureb0up683ppcpb1fz9g.isbul.info at 108.178.15.187 (Florida – possible mail server)**
      * **zz1lb82z00y16gdow25csm.ilaclama.us at 108.178.15.187**
      * **f9wb0396aobdotyzddcwdtf.ilaclama.us at 108.178.15.187**
   2. What website did the user visit right before the EK traffic began (the referrer)? **Wolfgangssteakhouse.co.kr at IP address 211.49.170.155**
   3. What was the URL for the EK landing page? bnureb0up683ppcbgt1fz9g.isbul.info
   4. Which EK do you suspect? Back up your answer with evidence to support your hypothesis. Nuclear EK
   5. What vulnerability was exploited in the attack? Provide a specific CVE classification, if possible.

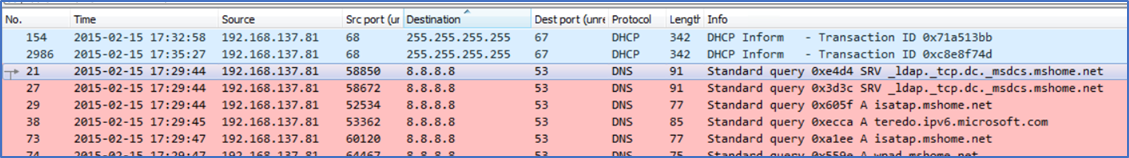
I viewed the packets in multiple different orders, from packet number, to sorting on protocol to TCP Stream – going back an forth multiple times and stopping to research frequently. Here’s a summarized list of the packet observations that led me to my conclusions, with screenshots following the list:

**LIST**

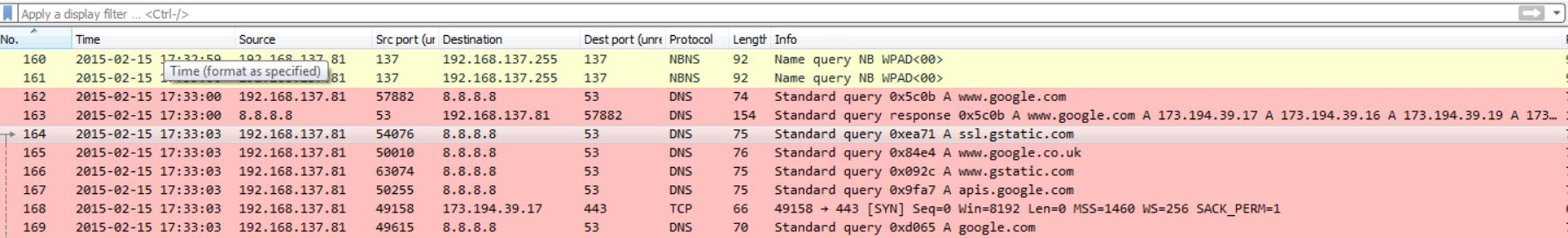
* Packet 74 - I found “wpad.mshome.net” which is Web Proxy Auto Discovery protocol. It is enabled by default in windows. It can expose computer users online accounts, web search, and other private data. It was developed to allow computers to automatically discover which web proxy they should use, the proxy is defined in a JavaScript file called a proxy auto-config (PAC) file. And the location of PAC files can be discovered through WPAD in several ways including a special DHCP option through DNS lookups or through LLMNR. Attackers can abuse these options to supply computers on a local network with a PAC file that specifies a rogue web proxy under their control. Man -in-the-middle attackers can abuse WPAD protocol to hijack peoples online accounts and steal their sensitive iformation when when they access websites over encrypted HTTPS or VPN connections. Reference this PC World article: <https://www.pcworld.com/article/3105998/disable-wpad-now-or-have-your-accounts-and-private-data-compromised.html> It can be disabled to prevent the security vulnerability.
* Packet 162 - user navigates to google (Src 192.168.137.81 sends request to 8.8.8.8 the default IPV4 IP address for google)
* Packet 164 - saw “ssl.gstatic.com” – researched and found it is a browser hijacker, wasn’t sure yet if that was involved.
* Packet 1139 and 1143 - Saw that user looked up Wolfgangs Steakhouse and received the response.
* Packet 1144 - followed a few TCP streams on this packet I discovered that Wolfgangs Steakhouse runs on Wordpress which was highly vulnerable to Nuclear EK. In 2015, the year that this pcap file was captured, there were numerous attacks on Wordpress sites by Nuclear EK, primarily through plugins. I then sorted on HTTP protocol to get a list of the Wordpress plugins and researched which ones were vulnerable:
  + Packet 1177: Contact Form 7 Integrations 1.0 – 1.3.10 (Multiple Cross-Site Scripting)
  + Packet 1180: Revslider (Local File Inclusion and Revolution Shell Upload) – not as likely because research indicated that attacks were occurring in 2016, after the date of this PCAP file.
  + Packet 1328, 1331, 1346: Kboard v. 4.0 (Cross-scripting vulnerability prior to v. 4.4) – good candidate since it’s pre 4.4
* Packet 1196 – I sorted on the DNS protocol I was able to see that there were some redirects from Wolfgangs Steakhouse as follows:
  + Packet 1196: Redirected to ts.daumcdn.net at 151.249.94.142 and 151.249.94.140
  + Packet 1434: Redirected to inhye01.itpr.co.kr at 211.111.221.233
  + Packet 1983: Redirected to bnureb0up683ppcpb1fz9g.isbul.info at 108.178.15.187
  + Packet 2484: Redirected to zz1lb82z00y16gdow25csm.ilaclama.us at 108.178.15.187
  + Packet 2555: Redirected to f9wb0396aobdotyzddcwdtf.ilaclama.us at 108.178.15.187
* Packet 2561 – I looked at the HTTP protocol packet and followed the TCP stream on it – found a lot of these characters: KbXKNgUKK..KN.UKKaXKN#UKKaXKNcU When I researched – that indicated that a payload may have been delivered.

**SCREENSHOTS**

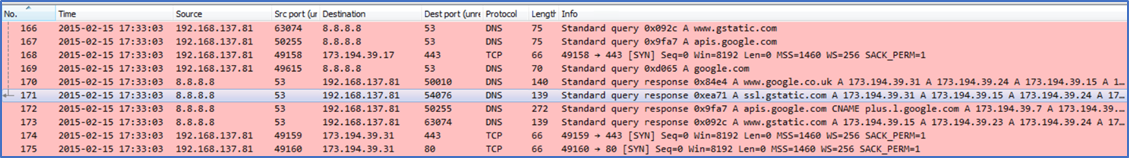
Packet 21: User goes to google based on IP 8.8.8.8 on 2/15/15 at 17:29 (3:29 PM)

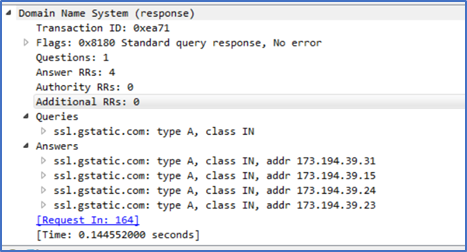


Packet 164: I found “ssl.gstatic.com” which is a browser hijacker. It can be removed but takes a lot of steps. It is usually bundled with software installers that are downloaded from various free software download sites. The symptoms that occur on a PC with “ssl.gstatic.com” include displaying various advertisements on the user’s screen and also possible search redirects. Not sure if this played a part in the redirect.



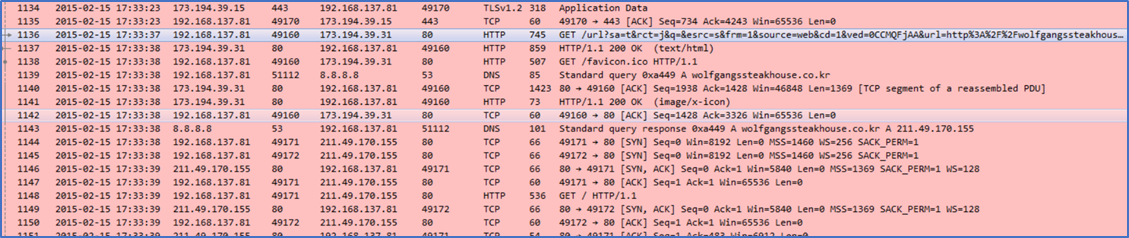
Packet 173: Appears that the “ssl.gstatic.com” may be redirecting to 173.194.39.31, 173.194.39.15, 173.194.39.24, and 173.194.39.23



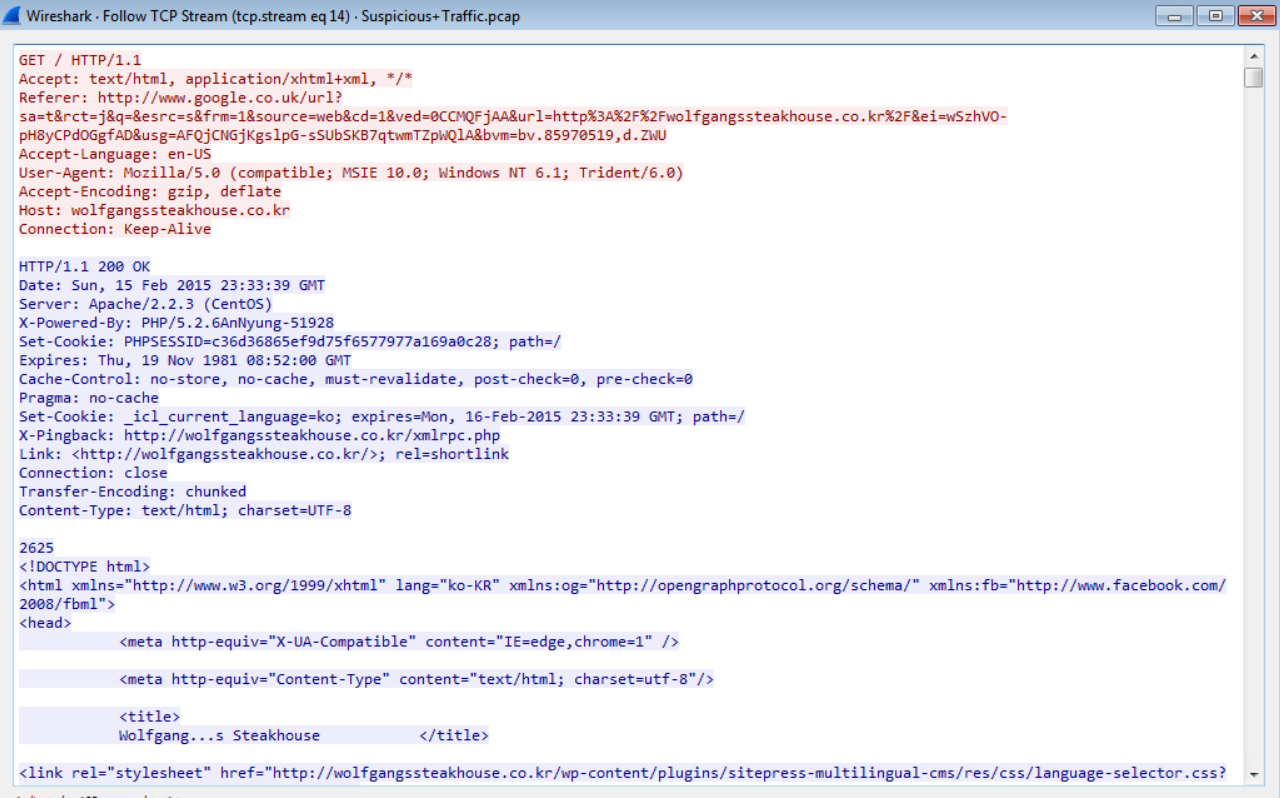


Packet 1136: User browses to wolfgangssteakhouse.com

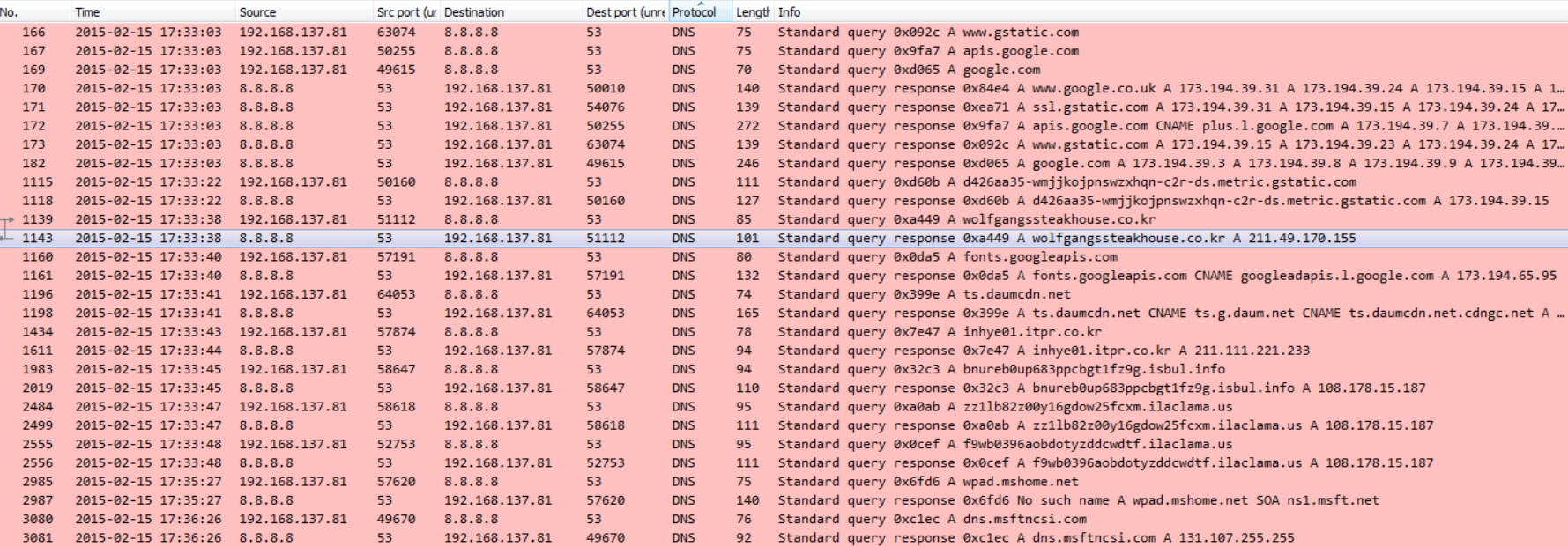
Packet 1143: Google responds with wolfgangssteakhouse.co.kr, IP 211.49.170.155



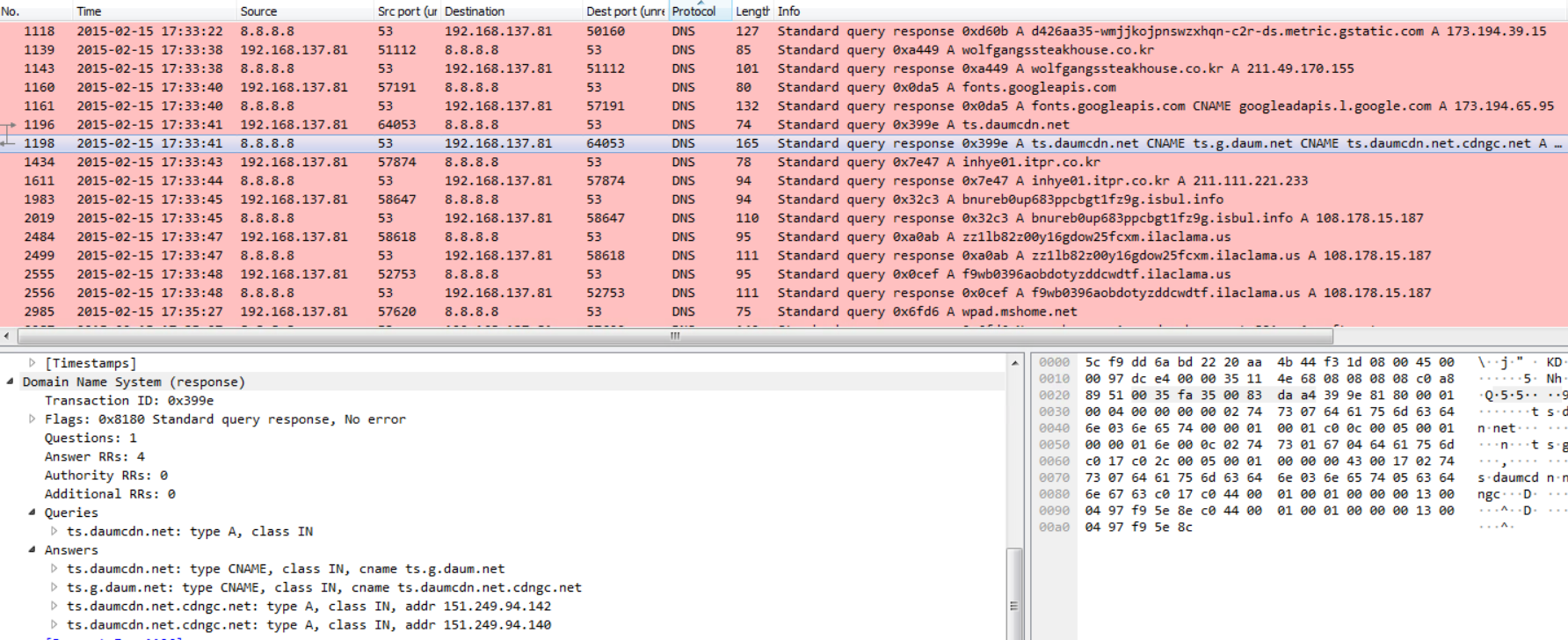
TCP Stream on Packet 1144 reveals that Wolfgangs Steakhouse is run on a Wordpress site (last line of code below). Wordpress was vulnerable to Nuclear EK in 2015 and there were many attacks recorded.



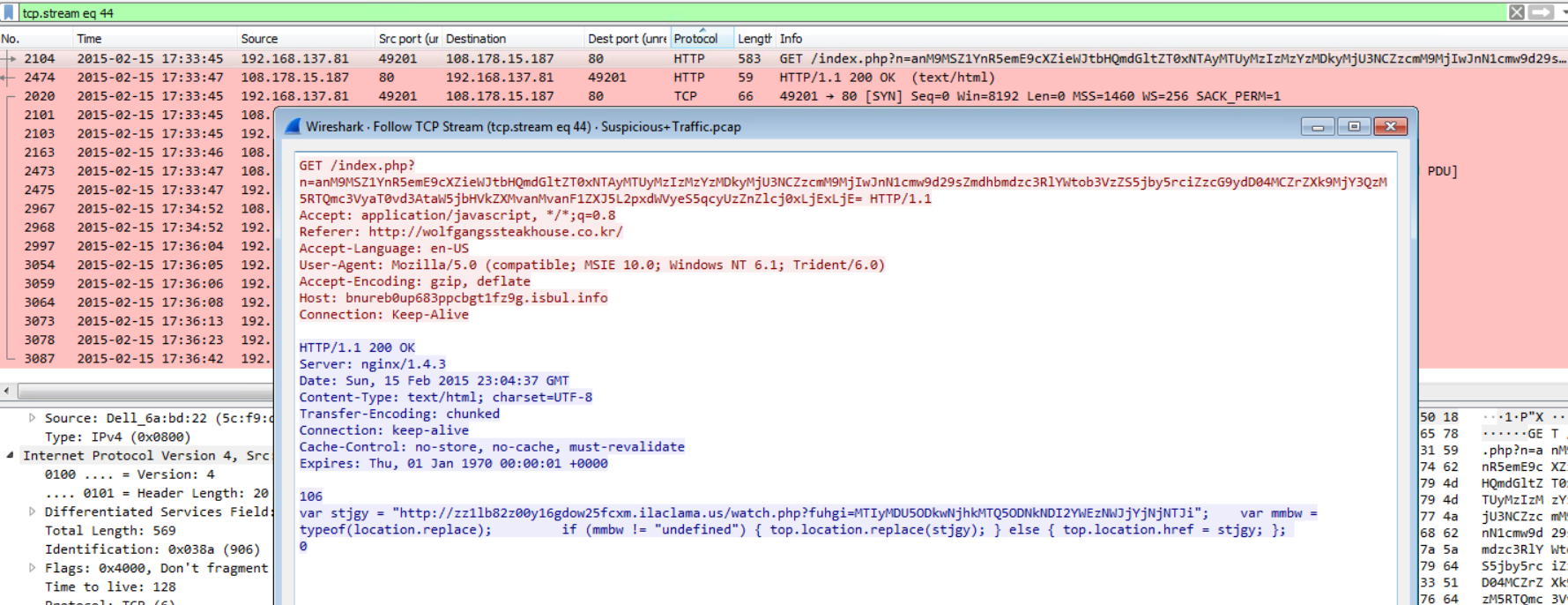
Packet 1196: DNS list of redirects with IP addresses (this is where the redirects occur)



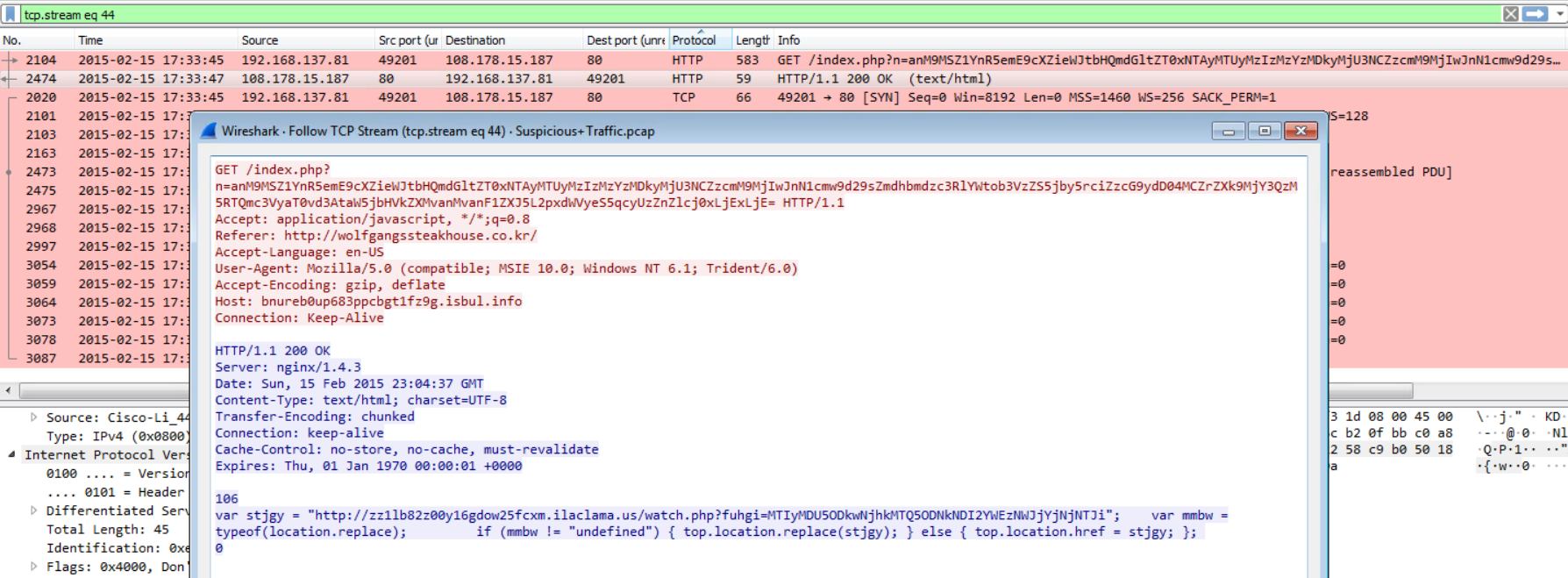
Packet 1198: IP for first redirect to ts.daumcdn.net (other redirect IP addresses are visible in DNS view above)

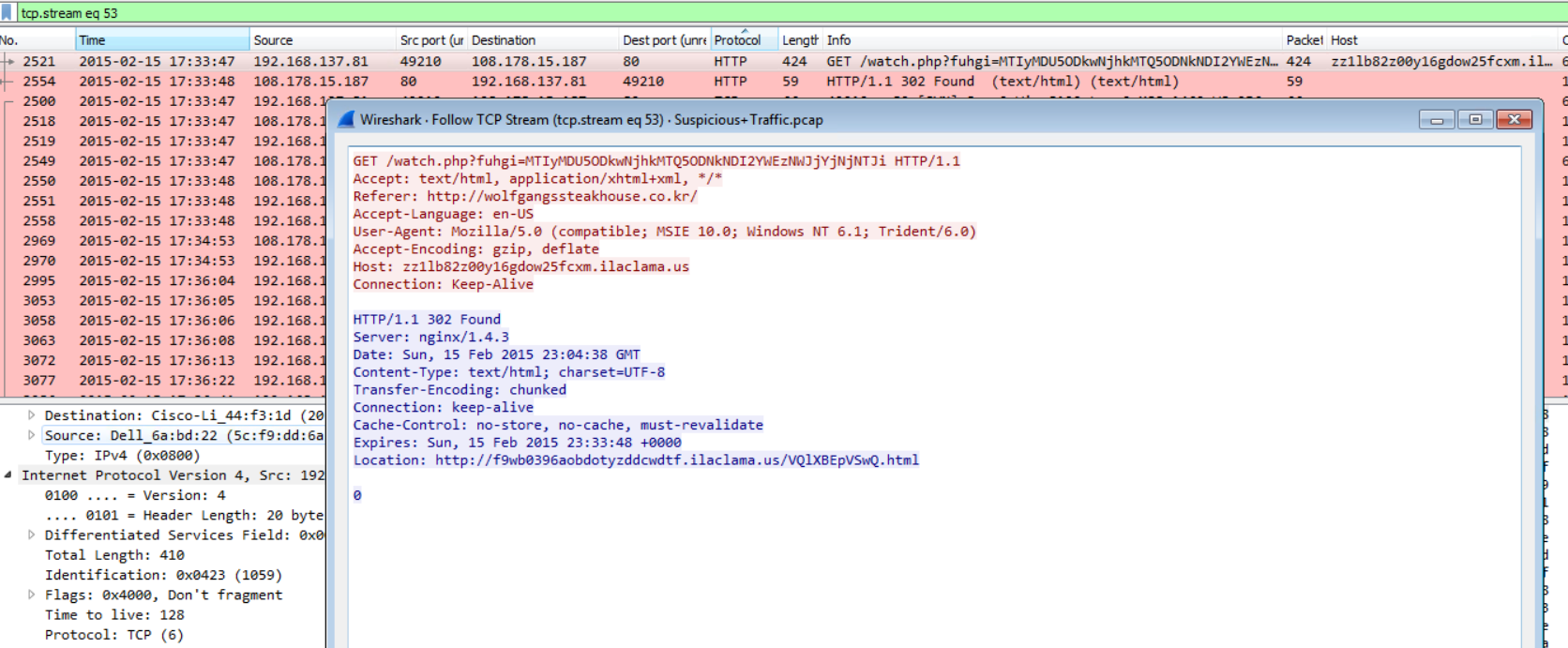


Packet 2104 HTTP GET and the TCP stream reveals a really long URL and a redirect

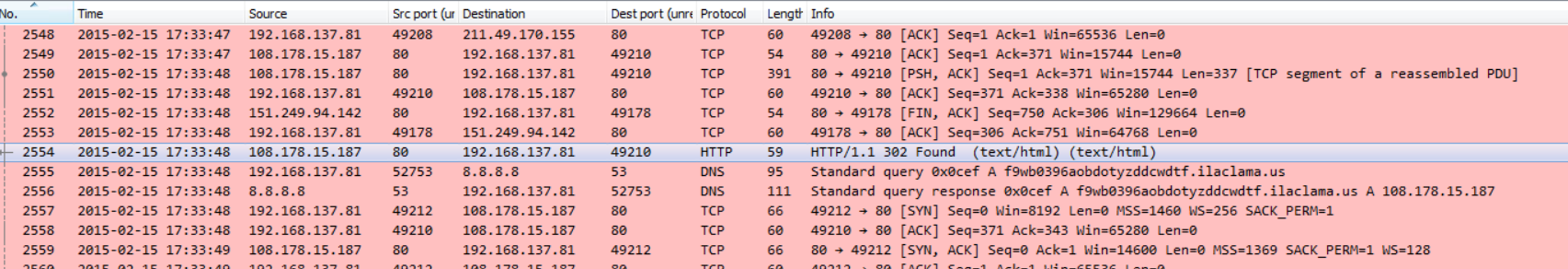


Packet 2474 and TCP Stream

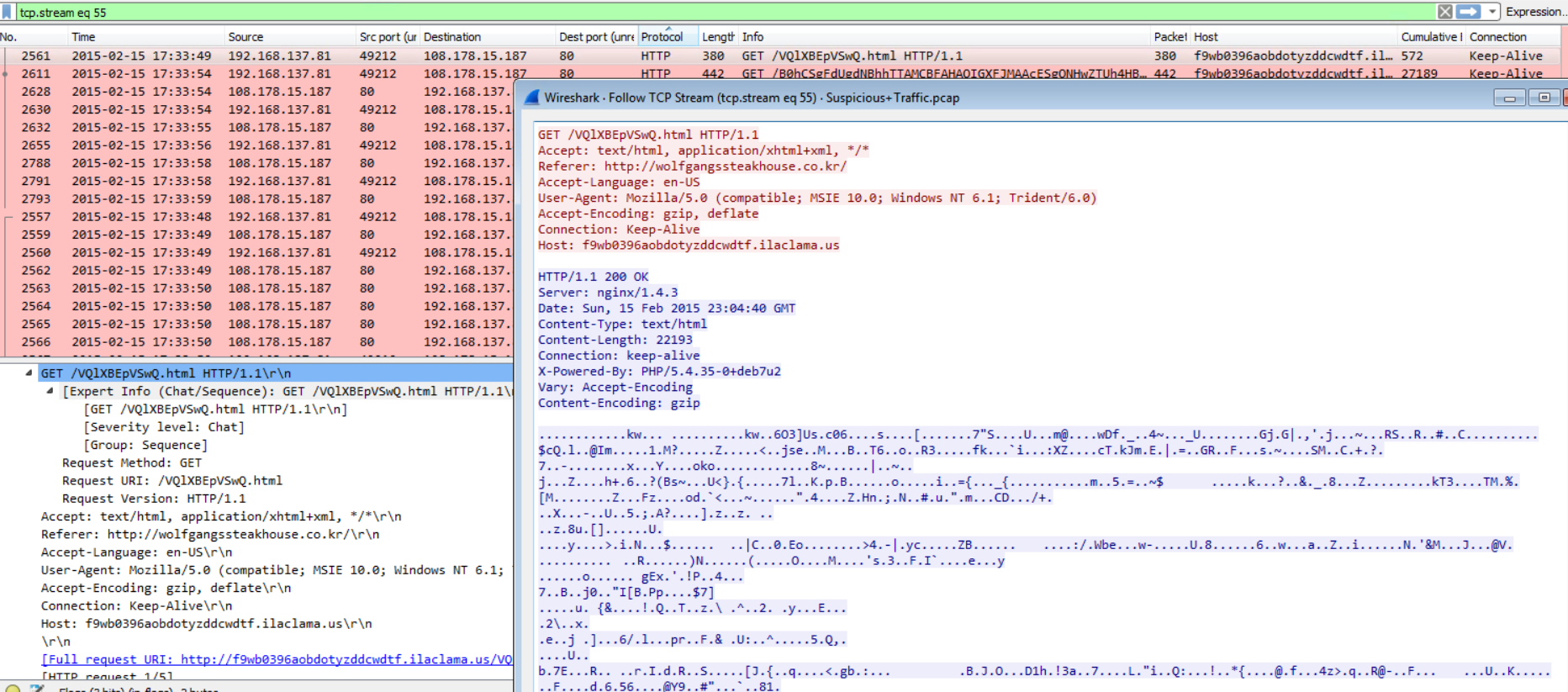


Packet 2521 

Packet 2554: the HTTP/1/1 302 Found occurs, which is a browser redirect (normally it would be “HTTP/1.1 200 Found”)



Packet 2561: TCP stream reveals redirect to different host



Packet 2586 – Followed TCP Stream, potential payload delivery (appears several times throughout the TCP stream)

Same thing appears again on Packet 2586

Referer: http://wolfgangssteakhouse.co.kr/

Host: f9wb0396aobdotyzddcwdtf.ilaclama.us

