Lab 9 – NIDS/NIPS and Web Proxy Analysis

Sai Sasaank Srivatsa Pallerla

University of Maryland, Baltimore County

Presented to: Gina Scaldaferri

Date: 11/12/2019

About the Lab

Snort network intrusion detection system (IDS) and intrusion prevention system (IPS). It has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching and matching. It can also used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, semantic URL attacks, buffer overflows, server message block probes, and stealth port scans.

Squid is a caching and forwarding HTTP web proxy. It has a wide variety of uses, including speeding up a web server by caching repeated requests, caching web, DNS and other computer network lookups for a group of people sharing network resources, and aiding security by filtering traffic. Although primarily used for HTTP and FTP, Squid includes limited support for several other protocols including Internet Gopher, SSL, TLS and HTTPS

A hex editor or binary file editor or byte editor is a computer program that allows for manipulation of the fundamental binary data that constitutes a computer file.

Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.It is one of the most famous tools that is used to monitor network traffic and protocols used. It lets us monitor the network at a microscopic level, both wired and wireless.

Part 1. IDS/IPS Log Analysis

1. Examine the alert's data to understand the logistical context

   [1:10000648:2] SHELLCODE x86 NOOP [] [Classification: Executable code was detected] [Priority: 1] 5/18-08:01:45.591840 172.16.16.218:80 -> 192.168.1.169:2493 TCP TTL: 63 TOS:0x0 ID:53309 IpLen: 20 DgmLen: 1127 DF *AP* Seq: 0x1B2C3517 Ack: 0x9F9E0666 Win: 0x1920 TcpLen: 20

   SHELLCODE x86 NOOP: This is the message that the user gave to display when an alert pops up.

   [Classification: Executable code was detected]: This the category of the alert belongs to, generally this can also be deduced from the alert file the snort rule was written in.

   [Priority: 1]: The priority level of the alert, so that the SOC can focus and divide their attention.

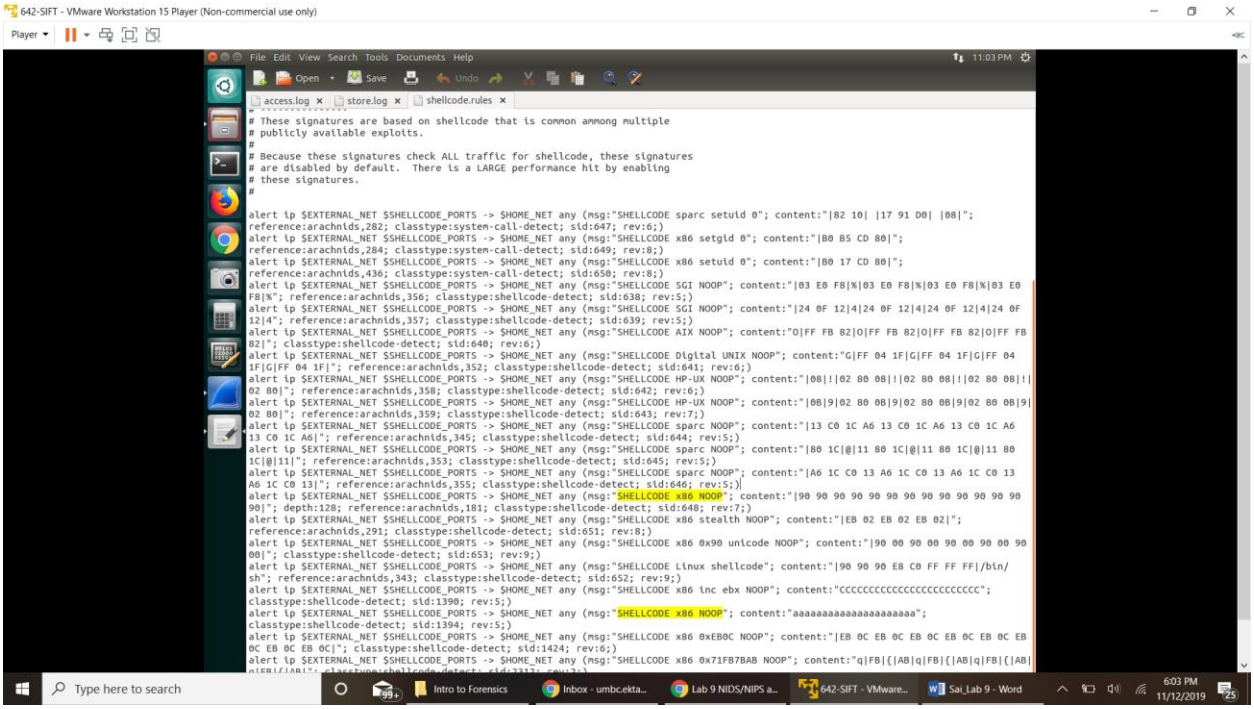
   5/18-08:01:45.591840: The time stamp when the alert/ rule was triggered.

   172.16.16.218:80 -> 192.168.1.169:2493: From and to IP address of the packet that was detected/ triggered the rule.

   TCP TTL: 63 TOS:0x0 ID:53309 IpLen: 20 DgmLen: 1127 DF ***AP*** Seq: 0x1B2C3517 Ack: 0x9F9E0666 Win: 0x1920 TcpLen: 20: This mentions the details of the packet, these details help the SOC the pinpoint the packet and conduct further inverstigation.

2. Compare the alert to the rule, in order to better understand WHAT it has been built to detect

   There are many rules mentioned in the rules folder of snort, to narrow down our search we can simply look at the SHELLCODE rules written the shellcode.rules file.



Two of the relevant alerts were:

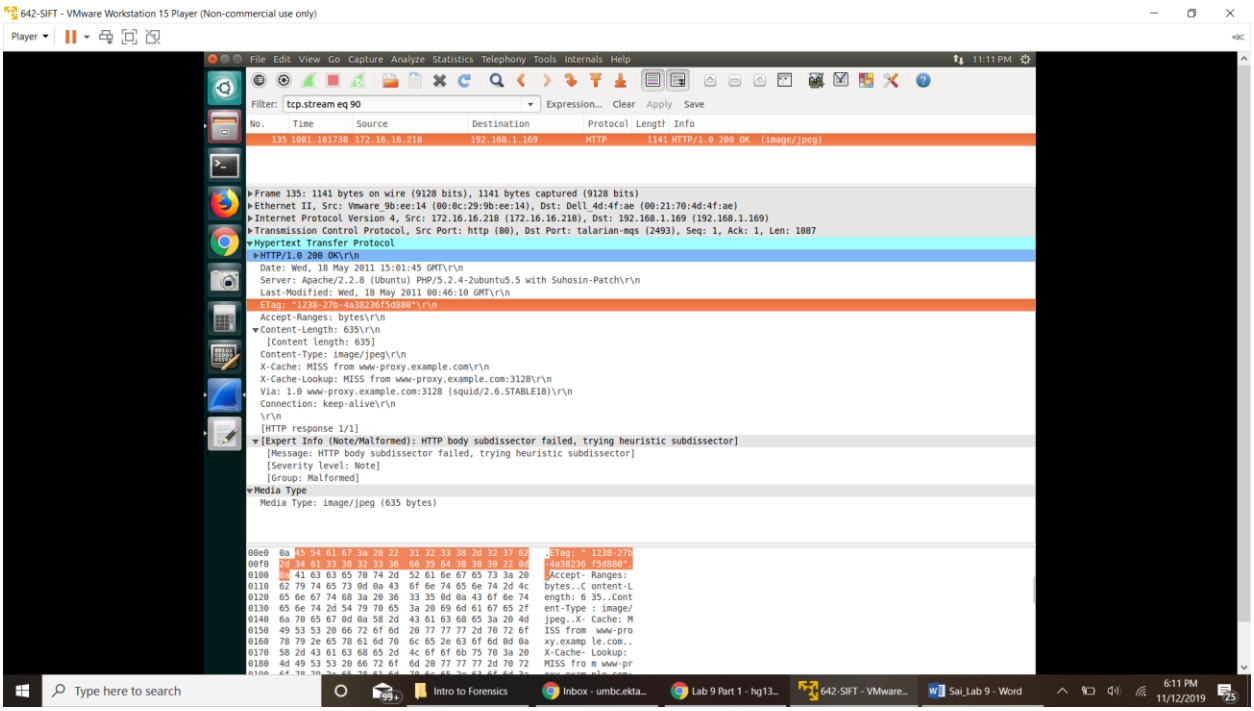
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth:128; reference:arachnids,181; classtype:shellcode-detect; sid:648; rev:7;)

alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"aaaaaaaaaaaaaaaaaaaa"; classtype:shellcode-detect; sid:1394; rev:5;)

These two rules basically serve the same purpose, When inspecting a packet these rules check of “aaaaaaaaaaaaaaaaaaaa" or its hex value "|90 90 90 90 90 90 90 90 90 90 90 90 90 90|". The intention behind this is when the attacker sends executable code, they try to mask the code by simply XORing the code and the output is usually aaaaaaaaaaaaaaaaaaaa or any repeated characters.

3. Retrieve the packet that triggered the alert

   To get the packet that trigged the alert we need to inspect the elements of the alert, we can see in field “ID:53309” which is the ID of the IP. So, to get the packet we open the log file in Wireshark and put a fileter “ip.id == 53309” and we get the packet, it’s a image/jpeg file.



4. Compare the rule to the packet to understand WHY it fired

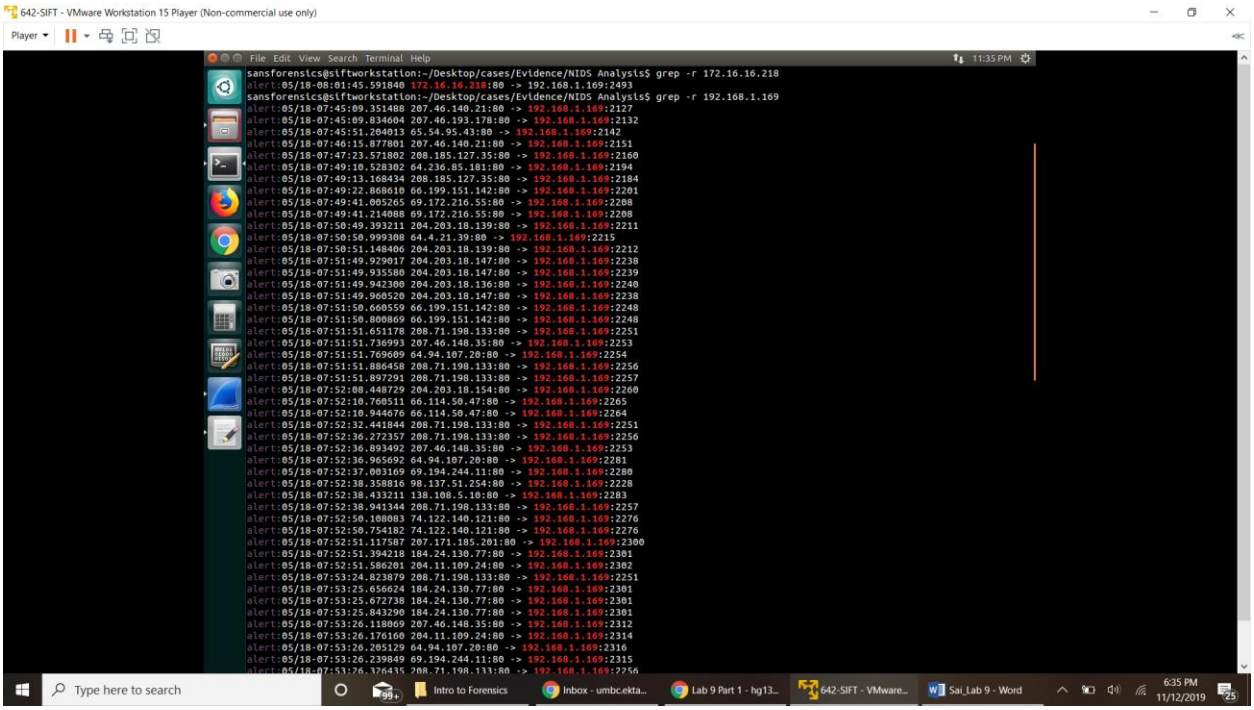
   Clearly the rule was checking for :"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|". The packet contained this int eh hex value that triggered the alert.

```
0400  55 37 f2 a3 b3 c3 28 29  d3 e3 f3 84 94 a4 b4 c4   U7....() ........
0410  d4 e4 f4 65 75 85 95 a5  b5 c5 90 90 90 90 90 90   ...eu... ........
0420  90 90 90 90 90 90 90 90  90 90 57 67 77 87 97 a7   ........ ..Wgw...
0430  b7 c7 d7 e7 f7 38 48 58  68 78 88 98 a8 b8 c8 d8   .....8HX hx......
0440  e8 f8 39 49 59 69 79 89  99 a9 b9 c9 d9 e9 f9 2a   ..9IYiy. .......*
0450  3a 4a 5a 6a 7a 8a 9a aa  ba ca da ea fa ff da 00   :JZjz
```

Subsequently, you'll want to determine if there are any other activities that are related to the original event.

1. Construct a timeline of alerted activities involving the potentially malicious outside host
2. Construct a timeline of alerted activities involving the target

   To prepare a timeline we need to inspect the IP addresses involved the suspicious packets, which are 172.16.16.218 and 192.168.1.169. The timeline involving internal and external hosts are as follows, (for this I used grep -r 172.16.16.218 and grep -r 192.168.1.169)



07:45:09 - NIDS alerts for 192.168.1.169 begin.

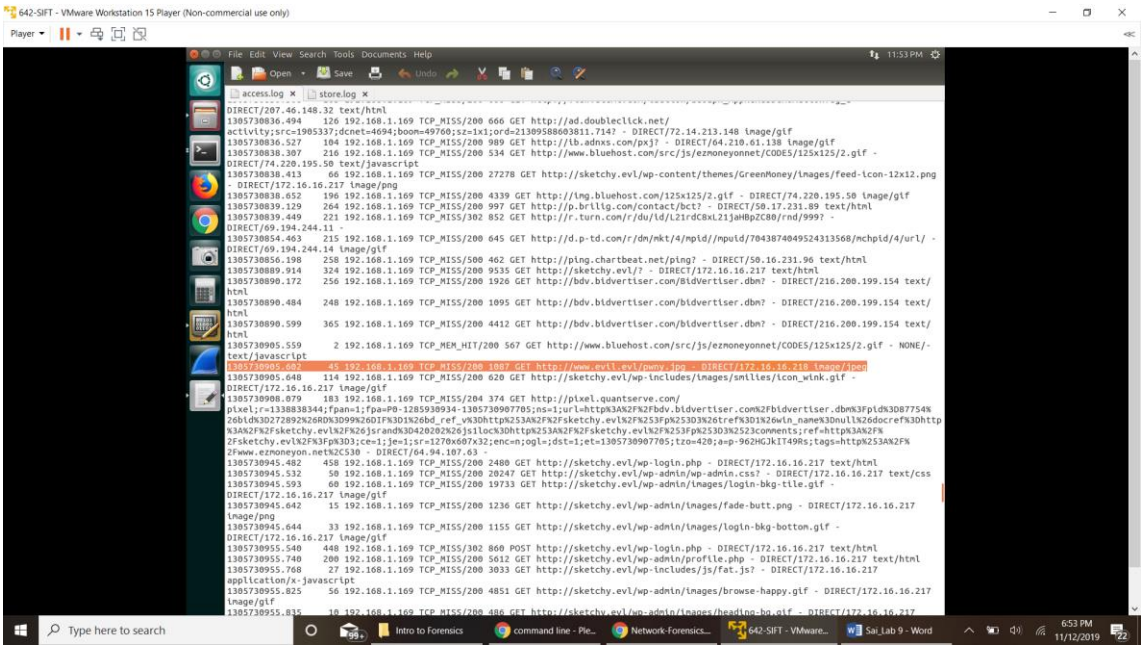
08:01:45 - NIDS alert for possible shellcode being downloaded

08:04:28-08:04:38 - Multiple NIDS alerts

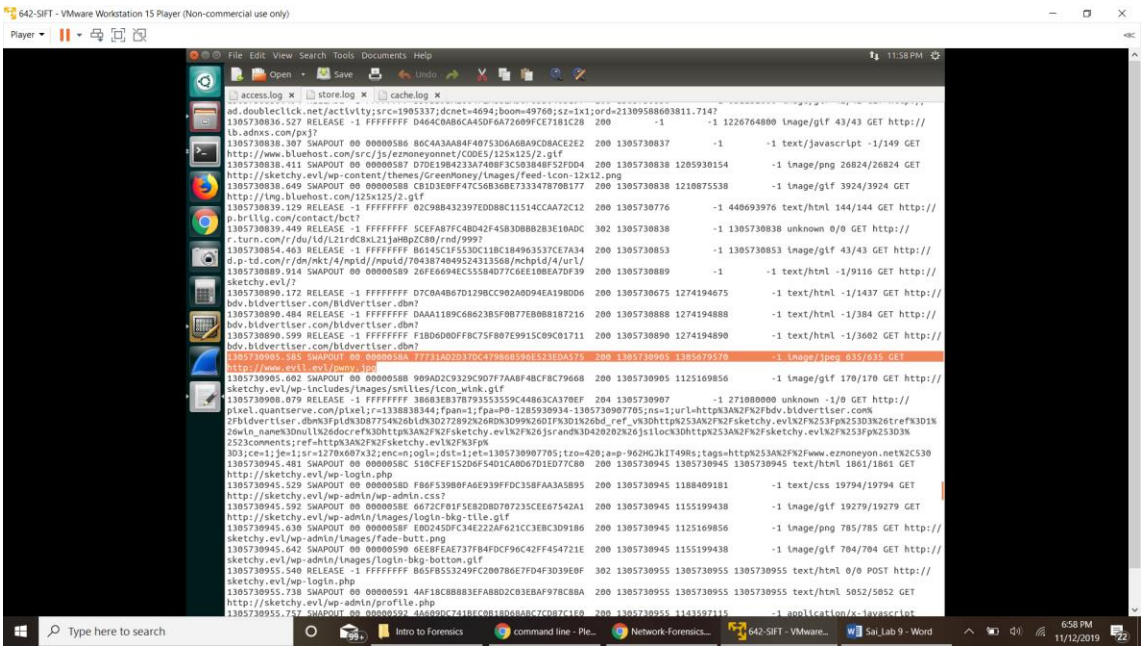
Part 2. Proxy Log Analysis

1. Determine whether the evidence extracted from the Squid cache corroborates our findings from the Snort logs.

   Yes, the evidence extracted from Squid cache corroborated our finding. First we search for the IP address 172.16.16.218 in access.log



   Then we get the website as http://www.evil.evl/pwny.jpg then we search for pwny.jpg in store.log

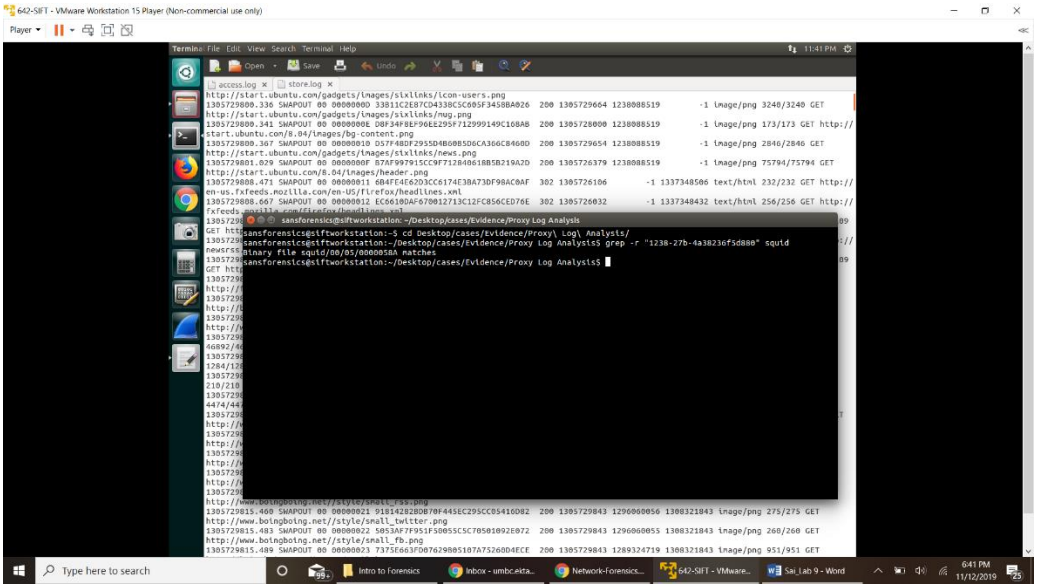


Based on web proxy access logs, gather information about the client system 192.168.1.169, including its likely operating systems and the apparent interests of any users.

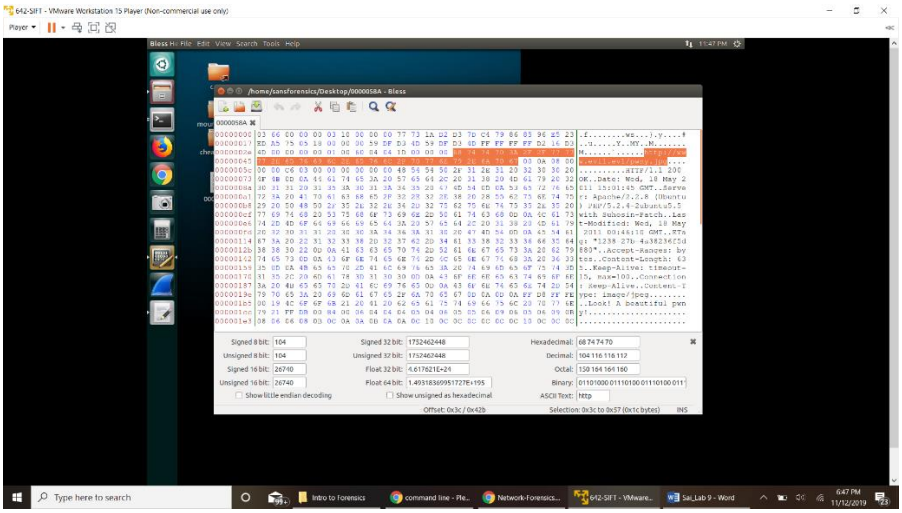
1. Present any information you can find regarding the identity of any internal users who have been engaged in suspicious activities.

   To retrieve the exact content of the packet we search it through the web proxies (which is located I the folder squid, Evidences)

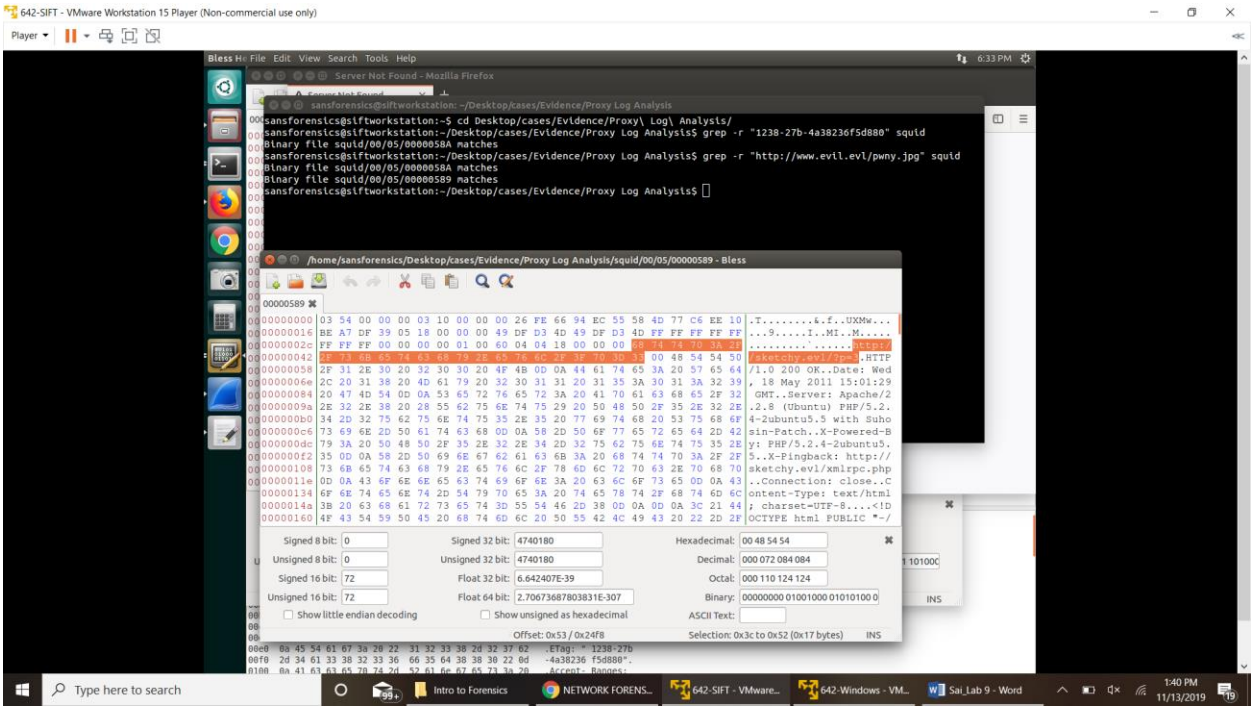
   We use grep -r Etag-value, the value we got from inspecting the packet.

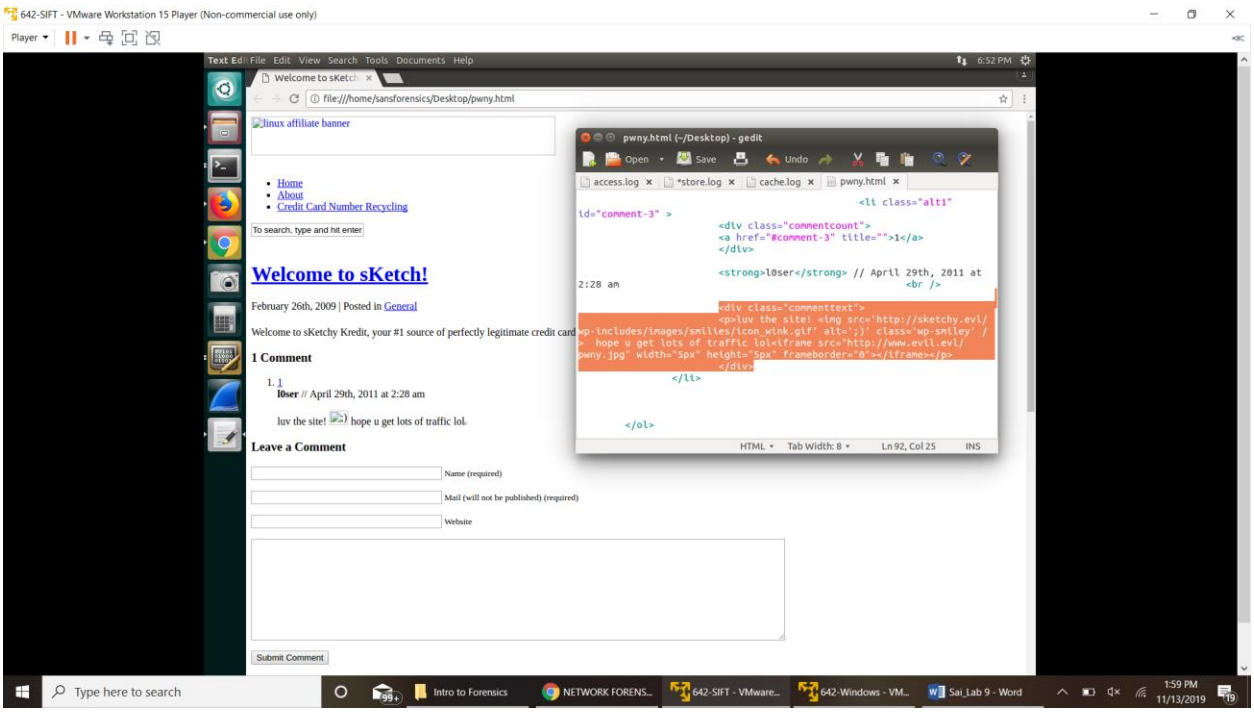


   This gives us the exact location of the file we are looking for inside the web proxy cache, because going through all of them is a huge and time taking task. We get the exact location of the file as squid/00/05/0000058A. We can open this hex file in Bless Hex Editor. The\n we get the web address as http://www.evil.evl/pwny.jpg



Now we know for sure that http://www.evil.evl/pwny.jpg was the culprit. To find more about where this originated from, we can grep search this URL address in the squid cache which leads us to another URL http://www.sketchy.evl/?p=3. When we go about examining the content, we find out that the attacker used cross site scripting vulnerability, it uses iframe and connects to a different website, in the website and this from done from the user account named **l0ser**. The content of the page is as follows:





Conclusion

From this lab we learn how to read through snort alerts and go through web proxy to find relevant data from the hex dumps. In a way we also learn how to write snort rules. We learn how view the logs to make inferences, match timelines to proves hypothesis. In short, this lab tells us that logs are important to look back at how the incident happened.

References

[1] https://forum.netgate.com/topic/138615/snort-alert-log-format/7

[2] https://github.com/Graylog2/graylog-guide-snort

[3] https://resources.infosecinstitute.com/snort-rules-workshop-part-one/#gref