Lab 9 – NIDS_NIPS and Web Proxy Analysis

Shrutika Joshi

University of Maryland Baltimore County

Presented To – Gina Marie

Date – 22nd July 2023

Week 9 Discussion:

IDS is a passive monitoring solution for detecting possible malicious activities/patterns, abnormal incidents, and policy violations. It is responsible for generating alerts for each suspicious event.

IPS is an active protecting solution for preventing possible malicious activities/patterns, abnormal incidents, and policy violations. It is responsible for stopping/preventing/terminating the suspicious event as soon as the detection is performed.

Difference between NIP/DS and HIP/DS

NIPS	**HIPS**
It operates at the network level	It operates at the host level

An IPS that inspects network traffic at the packet level and blocks traffic if it identifies as a malicious traffic	It monitors a single host for suspicious activities and actively blocks or quarantine traffic if a signature is identified
It protects the traffic on the entire subnet	It protects the traffic from a single endpoint device.
Cisco's FirePOWER is a network-based intrusion prevention system that offers real-time protection against advanced threats.[1][2]	McAfee Host Intrusion Prevention is a solution designed to protect individual systems from known and unknown malicious activities.[1][2]

NIDS	**HIDS**
It operates at the network level	It operates at the host level
An IDS that inspects network traffic at the packet level to identify threats but does not block it	This IDS is installed on the single system or endpoint and identifies any threats associated with only that system but does not block it
It operates by comparing network traffic patterns and signatures against known attack patterns to identify potential threats, such as intrusion attempts, malware, and denial-of-service attacks**.**	It operates by examining system logs, file integrity, and application activities to detect any suspicious or malicious behavior.
Snort is a well-known open-source network intrusion detection system. [1]	OSSEC is an open-source host-based intrusion detection system[1]

Citations –

Singh, H. (n.d.). Host-Based Intrusion Detection System (IDS): What is it? TheCyphere. https://thecyphere.com/blog/host-based-ids/

Radichel, T. (2020, April 16). IDS and IPS in the Cloud. Medium. Retrieved from https://medium.com/cloud-security/ids-and-ips-in-the-cloud-f07aac110d31

Scenario – Inter0ptic has started a credit card number recycling program. "Do you have a database filled with credit card numbers, just sitting there collecting dust? Put that data to good use!" he writes on his website. "Recycle your company's used credit card numbers! Send us your database, and we'll send YOU a check." For good measure, .Inter0ptic decides to add some bells and whistles to the site too. Meanwhile…. MacDaddy Payment Processor has deployed Snort NIDS sensors to detect an array of anomalous events, bot inbound and outbound. An alert was logged at 08:01:45 on 5/18/11 concerning an inbound chunk of executable code sent to port 80/tcp for inside host 192.168.1.169 from external host 172.16.16.218.

Introduction – Perform log analysis on given IDS/IPS snort alerts and analyze malicious traffic by analyzing these alerts. Also, we will be comparing alerts with the rule to confirm whether the alert is True positive or not. After this, we will be analyzing squid cache directories and files associated with it to understand more about the activity.

Pre-Lab – We are using Linux VM for performing this lab. Also, we will be using Wireshark, gedit, Bless Hex Editor, and command-line tool to run different commands

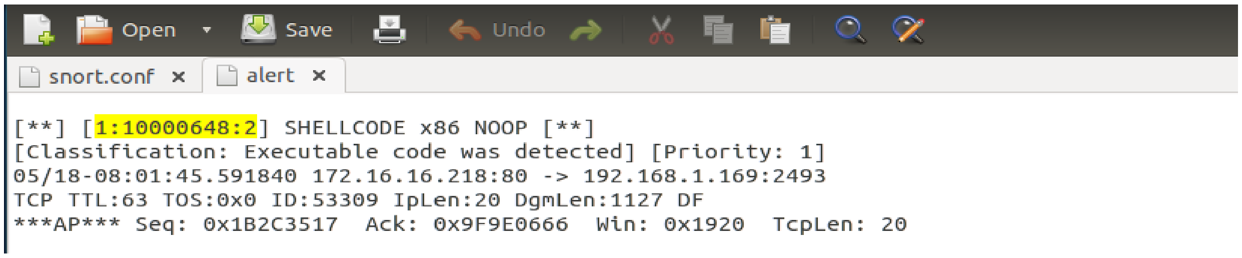
Analysis –

IDS/IPS Alert Background.

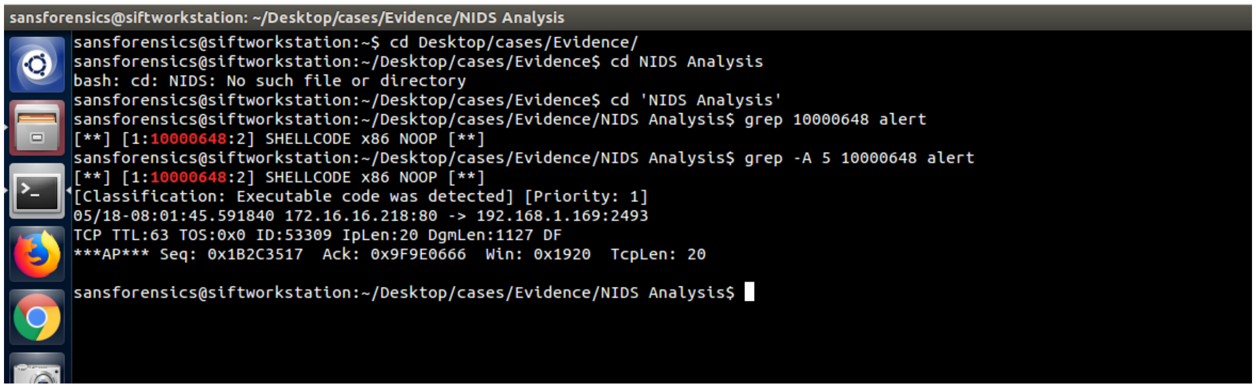
IDS/IPS Log Analysis –

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

- Below is the alert which got triggered having snort ID as '10000648'. You can open the alert.txt file using gedit and search for snort ID 10000648 or you can traverse to file location using the command line and use grep command to search the alert using snort number.
- **Snort** is an open-source network intrusion detection and prevention system (IDS/IPS) that monitors network traffic and identifies potentially malicious activities on Internet Protocol (IP) networks. [3]



```
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 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
```


sansforensics@siftworkstation: ~/Desktop/cases/Evidence/NIDS Analysis

```
sansforensics@siftworkstation:~$ cd Desktop/cases/Evidence/
sansforensics@siftworkstation:~/Desktop/cases/Evidence$ cd NIDS Analysis
bash: cd: NIDS: No such file or directory
sansforensics@siftworkstation:~/Desktop/cases/Evidence$ cd 'NIDS Analysis'
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ grep 10000648 alert
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ grep -A 5 10000648 alert
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 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

sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$
```

- **Network:**

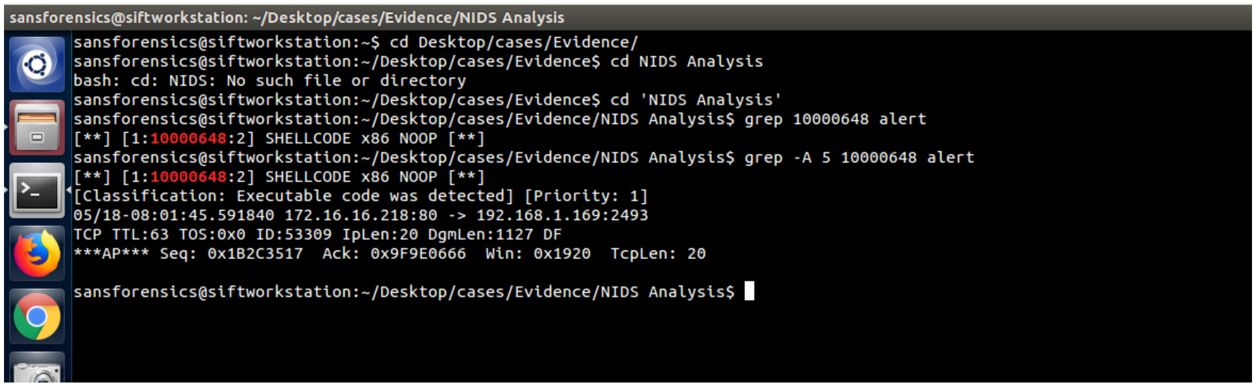
  Internal network: 192.168.1.0/24 DMZ network: 10.1.1.0/24 "Internet": 172.16.16.0/24

  Other domains and subnets of interest include: .evl – a top level domain used by Evil Systems Example.com – MacDaddy Payment Processor local domain

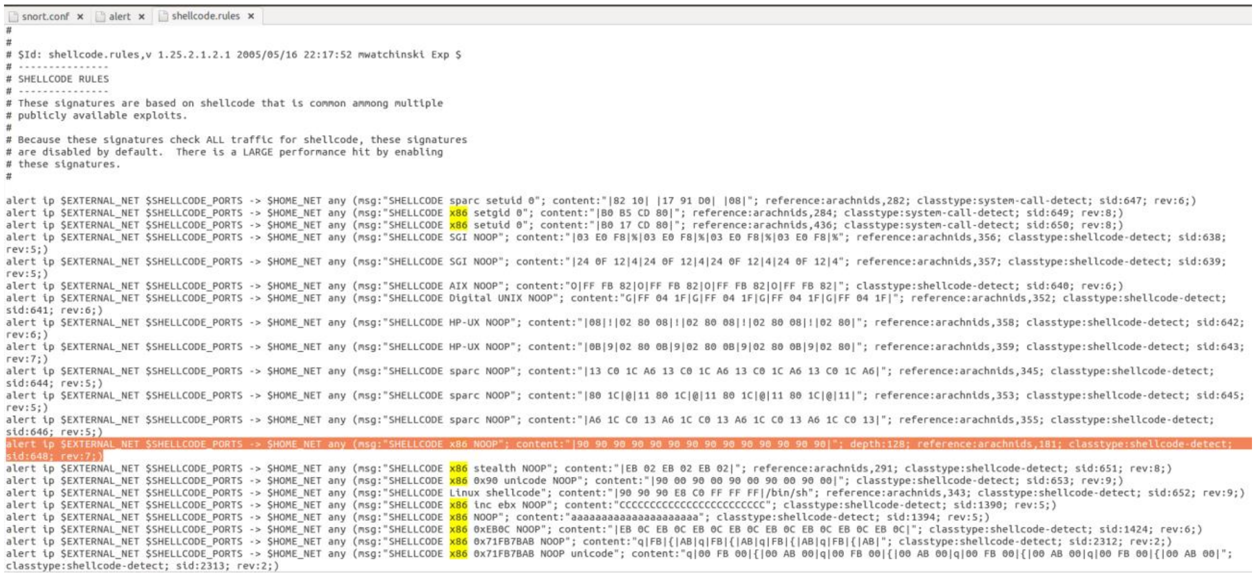
- Considering the above alert and network details provided to us, the alert indicate that Snort detected shellcode of X86 architecture with NOOP (No operation) and the timestamp is 5/18-08:01:45.591840.
- The classification of the alert is ‘Executable code was detected’. It means an alert is triggered under this category. The priority is set to ‘1’.
- The source IP address is 172.16.16.218, an internet (external) IP address communicating over port 80 (HTTP port) to an internal private IP address 192.168.1.169 over port 2493. The detected snort ID is 10000648 and the alert id is 53309.

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

- Alert :

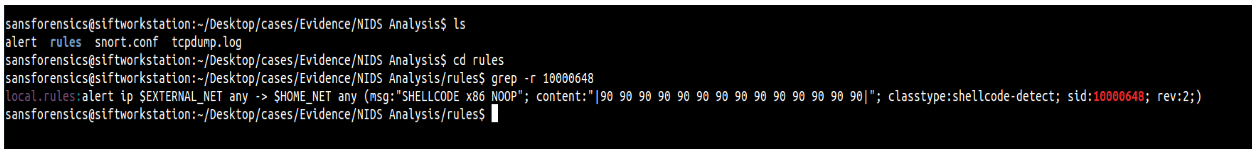


```
sansforensics@siftworkstation: ~/Desktop/cases/Evidence/NIDS Analysis
sansforensics@siftworkstation:~$ cd Desktop/cases/Evidence/
sansforensics@siftworkstation:~/Desktop/cases/Evidence$ cd NIDS Analysis
bash: cd: NIDS: No such file or directory
sansforensics@siftworkstation:~/Desktop/cases/Evidence$ cd 'NIDS Analysis'
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ grep 10000648 alert
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ grep -A 5 10000648 alert
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 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

sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$
```



```
snort.conf × | alert × | shellcode.rules ×
#
#
# $Id: shellcode.rules,v 1.25.2.1.2.1 2005/05/16 22:17:52 mwatchinski Exp $
# ---------------
# SHELLCODE RULES
# ---------------
# These signatures are based on shellcode that is common ammong multiple
# publicly available exploits.
#
# Because these signatures check ALL traffic for shellcode, these signatures
# are disabled by default.  There is a LARGE performance hit by enabling
# these signatures.
#

alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE sparc setuid 0"; content:"|82 10| |17 91 D0| |08|"; reference:arachnids,282; classtype:system-call-detect; sid:647; rev:6;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 setgid 0"; content:"|B0 B5 CD 80|"; reference:arachnids,284; classtype:system-call-detect; sid:649; rev:8;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 setuid 0"; content:"|B0 17 CD 80|"; reference:arachnids,436; classtype:system-call-detect; sid:650; rev:8;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE SGI NOOP"; content:"|03 E0 F8|%|03 E0 F8|%|03 E0 F8|%|03 E0 F8|%"; reference:arachnids,356; classtype:shellcode-detect; sid:638;
rev:5;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE SGI NOOP"; content:"|24 0F 12|4|24 0F 12|4|24 0F 12|4|24 0F 12|4"; reference:arachnids,357; classtype:shellcode-detect; sid:639;
rev:5;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE AIX NOOP"; content:"O|FF FB 82|O|FF FB 82|O|FF FB 82|O|FF FB 82|"; classtype:shellcode-detect; sid:640; rev:6;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE Digital UNIX NOOP"; content:"G|FF 04 1F|G|FF 04 1F|G|FF 04 1F|G|FF 04 1F|"; reference:arachnids,352; classtype:shellcode-detect;
sid:641; rev:6;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE HP-UX NOOP"; content:"|08|!|02 80 08|!|02 80 08|!|02 80 08|!|02 80|"; reference:arachnids,358; classtype:shellcode-detect; sid:642;
rev:6;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE HP-UX NOOP"; content:"|0B|9|02 80 0B|9|02 80 0B|9|02 80 0B|9|02 80|"; reference:arachnids,359; classtype:shellcode-detect; sid:643;
rev:7;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE sparc NOOP"; content:"|13 C0 1C A6 13 C0 1C A6 13 C0 1C A6 13 C0 1C A6|"; reference:arachnids,345; classtype:shellcode-detect;
sid:644; rev:5;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE sparc NOOP"; content:"|80 1C|@|11 80 1C|@|11 80 1C|@|11 80 1C|@|11|"; reference:arachnids,353; classtype:shellcode-detect; sid:645;
rev:5;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE sparc NOOP"; content:"|A6 1C C0 13 A6 1C C0 13 A6 1C C0 13 A6 1C C0 13|"; reference:arachnids,355; classtype:shellcode-detect;
sid:646; rev:5;)
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 stealth NOOP"; content:"|EB 02 EB 02 EB 02|"; reference:arachnids,291; classtype:shellcode-detect; sid:651; rev:8;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 0x90 unicode NOOP"; content:"|90 00 90 00 90 00 90 00 90 00|"; classtype:shellcode-detect; sid:653; rev:9;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8 C0 FF FF FF|/bin/sh"; reference:arachnids,343; classtype:shellcode-detect; sid:652; rev:9;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 inc ebx NOOP"; content:"CCCCCCCCCCCCCCCCCCCCCCCC"; classtype:shellcode-detect; sid:1390; rev:5;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"aaaaaaaaaaaaaaaaaaaaa"; classtype:shellcode-detect; sid:1394; rev:5;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 0xEB0C NOOP"; content:"|EB 0C EB 0C EB 0C EB 0C EB 0C EB 0C EB 0C EB 0C|"; classtype:shellcode-detect; sid:1424; rev:6;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 0x71FB7BAB NOOP"; content:"q|FB|{|AB|q|FB|{|AB|q|FB|{|AB|q|FB|{|AB|"; classtype:shellcode-detect; sid:2312; rev:2;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 0x71FB7BAB NOOP unicode"; content:"q|00 FB 00|{|00 AB 00|q|00 FB 00|{|00 AB 00|q|00 FB 00|{|00 AB 00|q|00 FB 00|{|00 AB 00|";
classtype:shellcode-detect; sid:2313; rev:2;)
```

- I have used grep –r command with snort ID: 10000648 check for a string in the rule in a recursive manner.



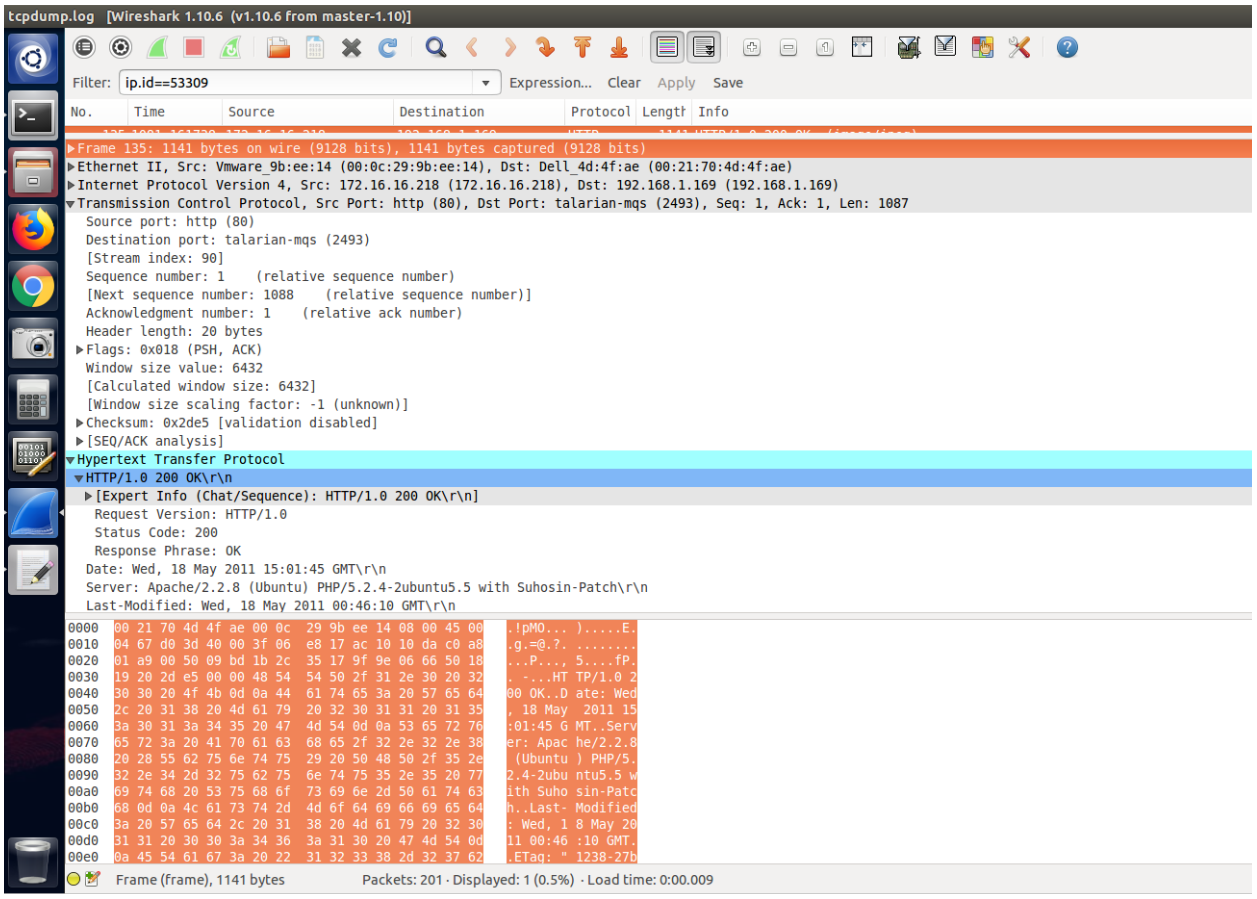
```
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ ls
alert  rules  snort.conf  tcpdump.log
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ cd rules
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis/rules$ grep -r 10000648
local.rules:alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; classtype:shellcode-detect; sid:10000648; rev:2;)
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis/rules$
```

- The triggered rule is designed to detect x86 shellcode with a NOOP instruction in the network traffic from external sources to the internal network on any protocol.
- The rule is analyzing IP packets traffic going from an external network ($EXTERNAL_NET) to an internal network ($HOME_NET) on any protocol and it is

looking for a sequence of 14 bytes each having the hexadecimal value ‘90’ which corresponds to the NOOP instruction in x86 assembly.

3. Retrieve the packet that triggered the alert

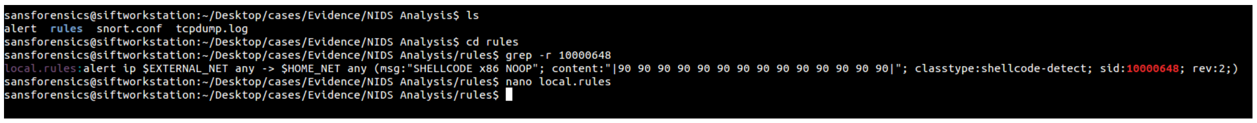
- To view the packet that triggers the alert, open the tcpdump.log file in the Wireshark application and use the command ip.id==53309, the ID number from the alert details. Expand the tabs to view packet details.



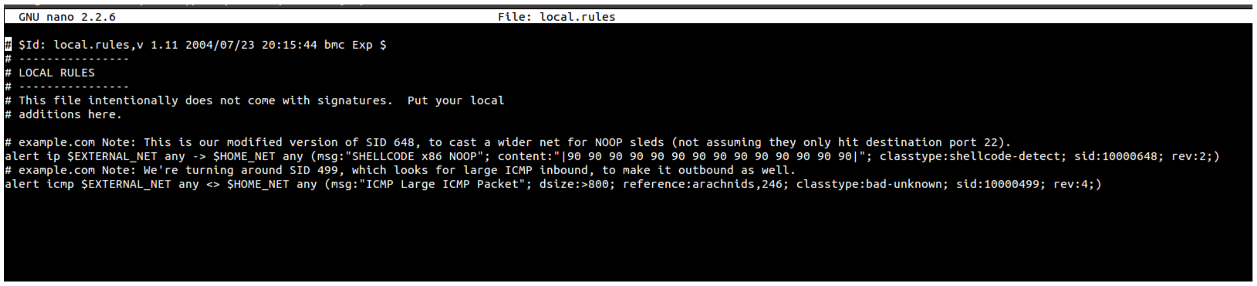
```
tcpdump.log   [Wireshark 1.10.6  (v1.10.6 from master-1.10)]
Filter: ip.id==53309                         Expression...  Clear  Apply  Save
No.   Time   Source   Destination   Protocol  Length  Info
▶Frame 135: 1141 bytes on wire (9128 bits), 1141 bytes captured (9128 bits)
▶Ethernet II, Src: Vmware_9b:ee:14 (00:0c:29:9b:ee:14), Dst: Dell_4d:4f:ae (00:21:70:4d:4f:ae)
▶Internet Protocol Version 4, Src: 172.16.16.218 (172.16.16.218), Dst: 192.168.1.169 (192.168.1.169)
▼Transmission Control Protocol, Src Port: http (80), Dst Port: talarian-mqs (2493), Seq: 1, Ack: 1, Len: 1087
   Source port: http (80)
   Destination port: talarian-mqs (2493)
   [Stream index: 90]
   Sequence number: 1    (relative sequence number)
   [Next sequence number: 1088    (relative sequence number)]
   Acknowledgment number: 1    (relative ack number)
   Header length: 20 bytes
  ▶Flags: 0x018 (PSH, ACK)
   Window size value: 6432
   [Calculated window size: 6432]
   [Window size scaling factor: -1 (unknown)]
  ▶Checksum: 0x2de5 [validation disabled]
  ▶[SEQ/ACK analysis]
▼Hypertext Transfer Protocol
 ▼HTTP/1.0 200 OK\r\n
   ▶[Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
    Request Version: HTTP/1.0
    Status Code: 200
    Response Phrase: OK
   Date: Wed, 18 May 2011 15:01:45 GMT\r\n
   Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch\r\n
   Last-Modified: Wed, 18 May 2011 00:46:10 GMT\r\n
0000  00 21 70 4d 4f ae 00 0c  29 9b ee 14 08 00 45 00   .!pMO... ).....E.
0010  04 67 d0 3d 40 00 3f 06  e8 17 ac 10 10 da c0 a8   .g.=@.?. ........
0020  01 a9 00 50 09 bd 1b 2c  35 17 9f 9e 06 66 50 18   ...P..., 5....fP.
0030  19 20 2d e5 00 00 48 54  54 50 2f 31 2e 30 20 32   . -...HT TP/1.0 2
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 57 65 64   00 OK..D ate: Wed
0050  2c 20 31 38 20 4d 61 79  20 32 30 31 31 20 31 35   , 18 May  2011 15
0060  3a 30 31 3a 34 35 20 47  4d 54 0d 0a 53 65 72 76   :01:45 G MT..Serv
0070  65 72 3a 20 41 70 61 63  68 65 2f 32 2e 32 2e 38   er: Apac he/2.2.8
0080  20 28 55 62 75 6e 74 75  29 20 50 48 50 2f 35 2e    (Ubuntu ) PHP/5.
0090  32 2e 34 2d 32 75 62 75  6e 74 75 35 2e 35 20 77   2.4-2ubu ntu5.5 w
00a0  69 74 68 20 53 75 68 6f  73 69 6e 2d 50 61 74 63   ith Suho sin-Patc
00b0  68 0d 0a 4c 61 73 74 2d  4d 6f 64 69 66 69 65 64   h..Last- Modified
00c0  3a 20 57 65 64 2c 20 31  38 20 4d 61 79 20 32 30   : Wed, 1 8 May 20
00d0  31 31 20 30 30 3a 34 36  3a 31 30 20 47 4d 54 0d   11 00:46 :10 GMT.
00e0  0a 45 54 61 67 3a 20 22  31 32 33 38 2d 32 37 62   .ETag: " 1238-27b
Frame (frame), 1141 bytes        Packets: 201 · Displayed: 1 (0.5%) · Load time: 0:00.009
```

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

To understand why the Snort rule fired for the given packet, we need to compare the content of the rule with the packet's content. The rule that triggered the alert is as follows:

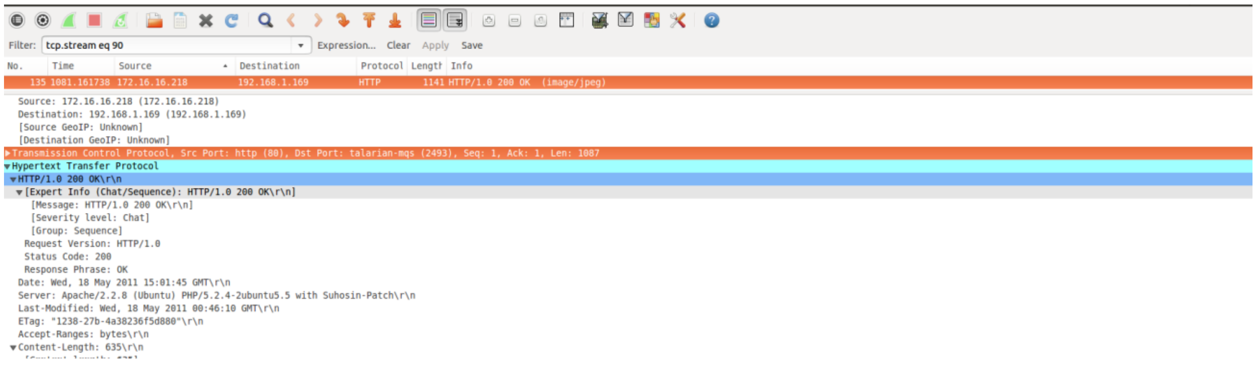


```
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ ls
alert  rules  snort.conf  tcpdump.log
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ cd rules
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis/rules$ grep -r 10000648
local.rules:alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; classtype:shellcode-detect; sid:10000648; rev:2;)
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis/rules$ nano local.rules
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis/rules$ 
```



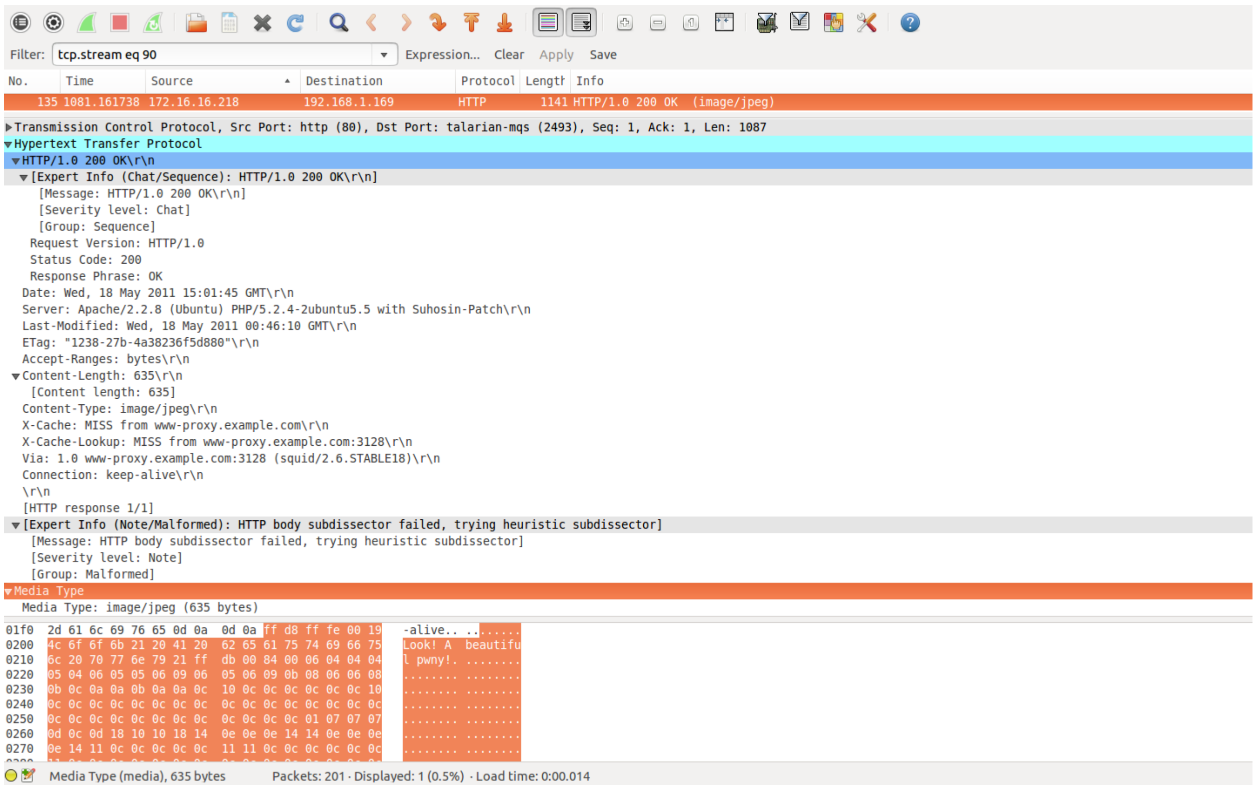
```
  GNU nano 2.2.6                                                        File: local.rules

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

# example.com Note: This is our modified version of SID 648, to cast a wider net for NOOP sleds (not assuming they only hit destination port 22).
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; classtype:shellcode-detect; sid:10000648; rev:2;)
# example.com Note: We're turning around SID 499, which looks for large ICMP inbound, to make it outbound as well.
alert icmp $EXTERNAL_NET any <> $HOME_NET any (msg:"ICMP Large ICMP Packet"; dsize:>800; reference:arachnids,246; classtype:bad-unknown; sid:10000499; rev:4;)
```

- Now, let's examine the relevant part of the packet's payload where the rule detected the shellcode:



```
Filter: tcp.stream eq 90                    Expression...  Clear  Apply  Save
No.    Time         Source          Destination     Protocol Length Info
   135 1081.161738  172.16.16.218   192.168.1.169   HTTP       1141 HTTP/1.0 200 OK  (image/jpeg)
  Source: 172.16.16.218 (172.16.16.218)
  Destination: 192.168.1.169 (192.168.1.169)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
▶Transmission Control Protocol, Src Port: http (80), Dst Port: talarian-mqs (2493), Seq: 1, Ack: 1, Len: 1087
▼Hypertext Transfer Protocol
 ▼HTTP/1.0 200 OK\r\n
  ▼[Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
     [Message: HTTP/1.0 200 OK\r\n]
     [Severity level: Chat]
     [Group: Sequence]
    Request Version: HTTP/1.0
    Status Code: 200
    Response Phrase: OK
  Date: Wed, 18 May 2011 15:01:45 GMT\r\n
  Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch\r\n
  Last-Modified: Wed, 18 May 2011 00:46:10 GMT\r\n
  ETag: "1238-27b-4a38236f5d880"\r\n
  Accept-Ranges: bytes\r\n
 ▼Content-Length: 635\r\n
```

- Since the Snort rule is designed to detect shellcode with a specific content pattern, let's focus on the TCP payload part and the Hypertext Transfer Protocol (HTTP) response details:



```
No.   Time          Source         Destination    Protocol Length Info
  135 1081.161738  172.16.16.218  192.168.1.169  HTTP     1141 HTTP/1.0 200 OK  (image/jpeg)

▶Transmission Control Protocol, Src Port: http (80), Dst Port: talarian-mqs (2493), Seq: 1, Ack: 1, Len: 1087
▼Hypertext Transfer Protocol
 ▼HTTP/1.0 200 OK\r\n
  ▼[Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
     [Message: HTTP/1.0 200 OK\r\n]
     [Severity level: Chat]
     [Group: Sequence]
    Request Version: HTTP/1.0
    Status Code: 200
    Response Phrase: OK
   Date: Wed, 18 May 2011 15:01:45 GMT\r\n
   Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch\r\n
   Last-Modified: Wed, 18 May 2011 00:46:10 GMT\r\n
   ETag: "1238-27b-4a38236f5d880"\r\n
   Accept-Ranges: bytes\r\n
 ▼Content-Length: 635\r\n
    [Content length: 635]
   Content-Type: image/jpeg\r\n
   X-Cache: MISS from www-proxy.example.com\r\n
   X-Cache-Lookup: MISS from www-proxy.example.com:3128\r\n
   Via: 1.0 www-proxy.example.com:3128 (squid/2.6.STABLE18)\r\n
   Connection: keep-alive\r\n
   \r\n
   [HTTP response 1/1]
 ▼[Expert Info (Note/Malformed): HTTP body subdissector failed, trying heuristic subdissector]
    [Message: HTTP body subdissector failed, trying heuristic subdissector]
    [Severity level: Note]
    [Group: Malformed]
▼Media Type
   Media Type: image/jpeg (635 bytes)

01f0  2d 61 6c 69 76 65 0d 0a  0d 0a ff d8 ff fe 00 19   -alive.. ........
0200  4c 6f 6f 6b 21 20 41 20  62 65 61 75 74 69 66 75   Look! A  beautifu
0210  6c 20 70 77 6e 79 21 ff  db 00 84 00 06 04 04 04   l pwny!. ........
0220  05 04 06 05 05 06 09 06  05 06 09 0b 08 06 06 08   ........ ........
0230  0b 0c 0a 0a 0b 0a 0a 0c  10 0c 0c 0c 0c 0c 0c 10   ........ ........
0240  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c 0c 0c   ........ ........
0250  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 01 07 07 07   ........ ........
0260  0d 0c 0d 18 10 10 18 14  0e 0e 0e 14 14 0e 0e 0e   ........ ........
0270  0e 14 11 0c 0c 0c 0c 0c  11 11 0c 0c 0c 0c 0c 0c   ........ ........

Media Type (media), 635 bytes    Packets: 201 · Displayed: 1 (0.5%) · Load time: 0:00.014
```

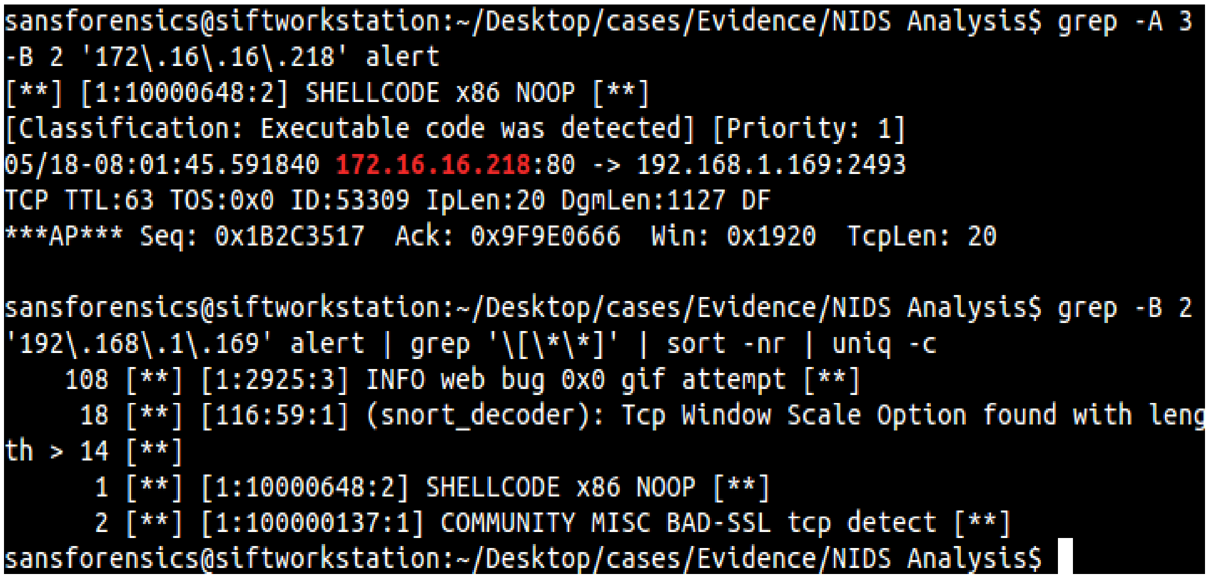
- Now, let's check the content of the Snort rule that caused the detection:

  content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|”;

- The rule is looking for a specific content pattern: 14 bytes with the hexadecimal value 90 (corresponding to the NOOP instruction) in the packet payload. However, based on the provided packet data, we do not see this specific content pattern present in the HTTP response. However, we have to analyze image content to better understand and come to any conclusion.

5. Construct a timeline of alerted activities involving the potentially malicious outside host

- To understand how many alerts detected for internal IP 192.168.1.169, I have used the command “grep -B 2 ‘192\.168\.1\.169’ alert | grep ‘\[**\]’ | sort -nr | uniq –c

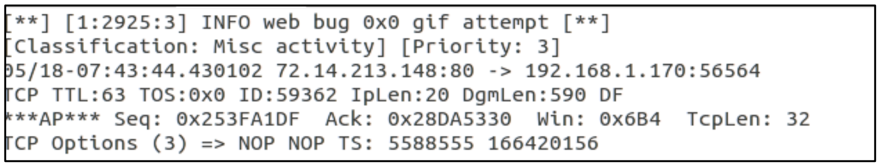


```
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ grep -A 3
-B 2 '172\.16\.16\.218' alert
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 172.16.16.218:80 -> 192.168.1.169:2493
TCP TTL:63 TOS:0x0 ID:53309 IpLen:20 DgmLen:1127 DF
***AP*** Seq: 0x1B2C3517  Ack: 0x9F9E0666  Win: 0x1920  TcpLen: 20

sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$ grep -B 2
'192\.168\.1\.169' alert | grep '\[\*\*]' | sort -nr | uniq -c
    108 [**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
     18 [**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with leng
th > 14 [**]
      1 [**] [1:10000648:2] SHELLCODE x86 NOOP [**]
      2 [**] [1:100000137:1] COMMUNITY MISC BAD-SSL tcp detect [**]
sansforensics@siftworkstation:~/Desktop/cases/Evidence/NIDS Analysis$
```

- There are four types of alerts triggered

1. SHELLCODE x86 NOOP
2. (snort_decoder): Tcp Window Scale Option found with length > 14
3. COMMUNITY MISC BAD-SSL tcp detect
4. INFO web bug 0x0 gif attempt

As per the alert detail, this alert started on 07:43:44



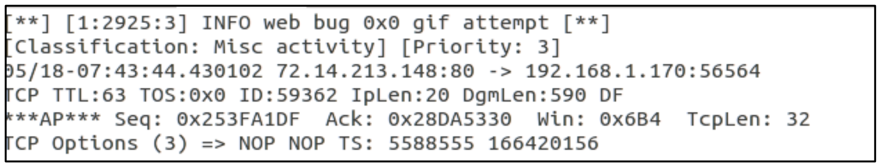
```
[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:43:44.430102 72.14.213.148:80 -> 192.168.1.170:56564
TCP TTL:63 TOS:0x0 ID:59362 IpLen:20 DgmLen:590 DF
***AP*** Seq: 0x253FA1DF  Ack: 0x28DA5330  Win: 0x6B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5588555 166420156
```

From this log entry, it seems like there was an attempted communication between the source IP 72.14.213.102 (over port 80) and the destination IP 192.168.1.169 (over port 2402) over TCP. The packet carries the ACK and PSH flags set, and the TCP window size is 11245 bytes. The log entry is categorized as "Misc activity" and has a medium-level priority. The log entry indicates a web bug attempt, suggesting some possible suspicious or unauthorized activity.

Web bugs, also known as tracking pixels or web beacons, are typically used to track users' interactions with web content and are often associated with tracking user behavior for advertising or analytics purposes.

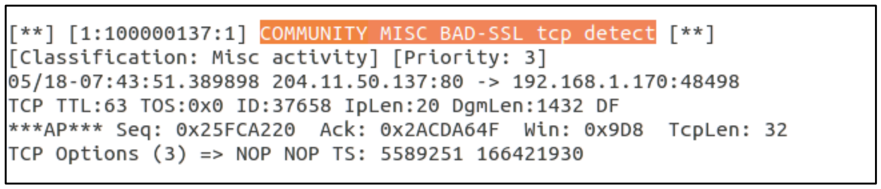
6. Construct a timeline of alerted activities involving the target

- After analyzing the logs and alerts received, we found that at 07:43:44 on 5/18/11, we received a first alert “INFO web bug 0x0 gif attempt” when the internal host IP address 192.168.1.169 was used to browse external websites which includes some web-based bugs.



```
[**] [1:2925:3] INFO web bug 0x0 gif attempt [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:43:44.430102 72.14.213.148:80 -> 192.168.1.170:56564
TCP TTL:63 TOS:0x0 ID:59362 IpLen:20 DgmLen:590 DF
***AP*** Seq: 0x253FA1DF  Ack: 0x28DA5330  Win: 0x6B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5588555 166420156
```

- At 07:43:51 first alert with name “COMMUNITY MISC BAD-SSL tcp detect” detected

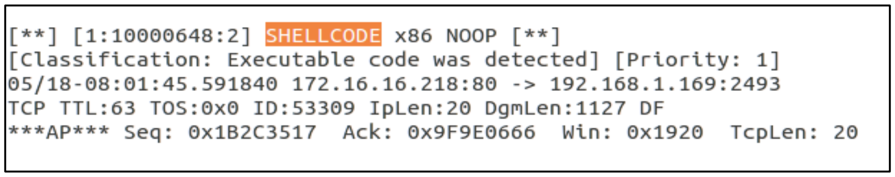


```
[**] [1:100000137:1] COMMUNITY MISC BAD-SSL tcp detect [**]
[Classification: Misc activity] [Priority: 3]
05/18-07:43:51.389898 204.11.50.137:80 -> 192.168.1.170:48498
TCP TTL:63 TOS:0x0 ID:37658 IpLen:20 DgmLen:1432 DF
***AP*** Seq: 0x25FCA220  Ack: 0x2ACDA64F  Win: 0x9D8  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5589251 166421930
```

- At 08:01:45 Shellcode was detected when external web server 172.16.16.218:80 delivered a JPEG image to 192.168.1.169 which contained an unusual binary sequence that is commonly associated with buffer overflow exploits.
- The ETag in the external webserver’s HTTP response was: 1238-27b-4a38236f5d880
- The MD5sum of the suspicious JPEG was:

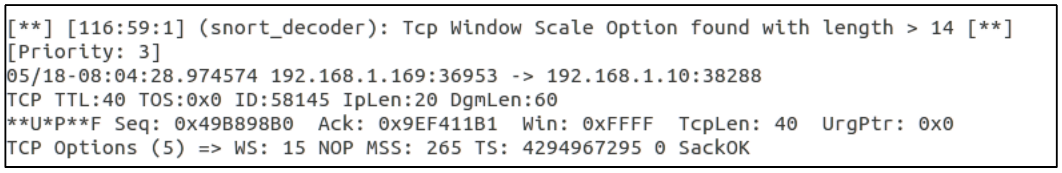
  13c303f746a0e8826b749fce56a5c126

- After that ‘INFO web bug 0x0 gif attempt’ continued



```
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 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
```

- At 08:04:28, we received first alert“Tcp Window Scale Option found with length > 14”



```
[**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14 [**]
[Priority: 3]
05/18-08:04:28.974574 192.168.1.169:36953 -> 192.168.1.10:38288
TCP TTL:40 TOS:0x0 ID:58145 IpLen:20 DgmLen:60
**U*P**F Seq: 0x49B898B0  Ack: 0x9EF411B1  Win: 0xFFFF  TcpLen: 40  UrgPtr: 0x0
TCP Options (5) => WS: 15 NOP MSS: 265 TS: 4294967295 0 SackOK
```

2. Proxy Log Analysis

Scenario: After the initial analysis we found that less than three minutes later, at 08:04:28 MST, internal host 192.168.1.169 spent roughly 10 seconds sending crafted packets to other internal hosts on the same network. Based on their nonstandard nature, the packets are consistent with those used to conduct reconnaissance via scanning and OS fingerprinting.

Network:

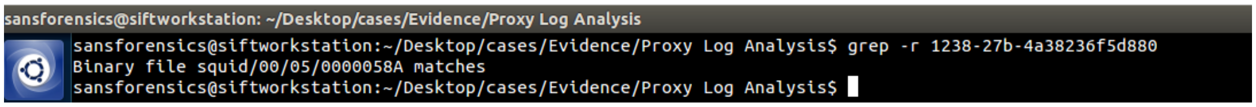
Internal network: 192.168.1.0/24 DMZ network: 10.1.1.0/24 “Internet”: 172.16.16.0/24 Other domains and subnets of interest include: .evl – a top-level domain used by Evil Systems Example.com – MacDaddy Payment Processor local domain.

Now examine the Squid cache and extract any cached pages/files associated with the Snort alert.

- Squid is a UNIX based caching proxy for the web that supports HTTP, HTTPS, FTP, and more. It runs on most available operating systems like Windows® and Linux. [2]

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

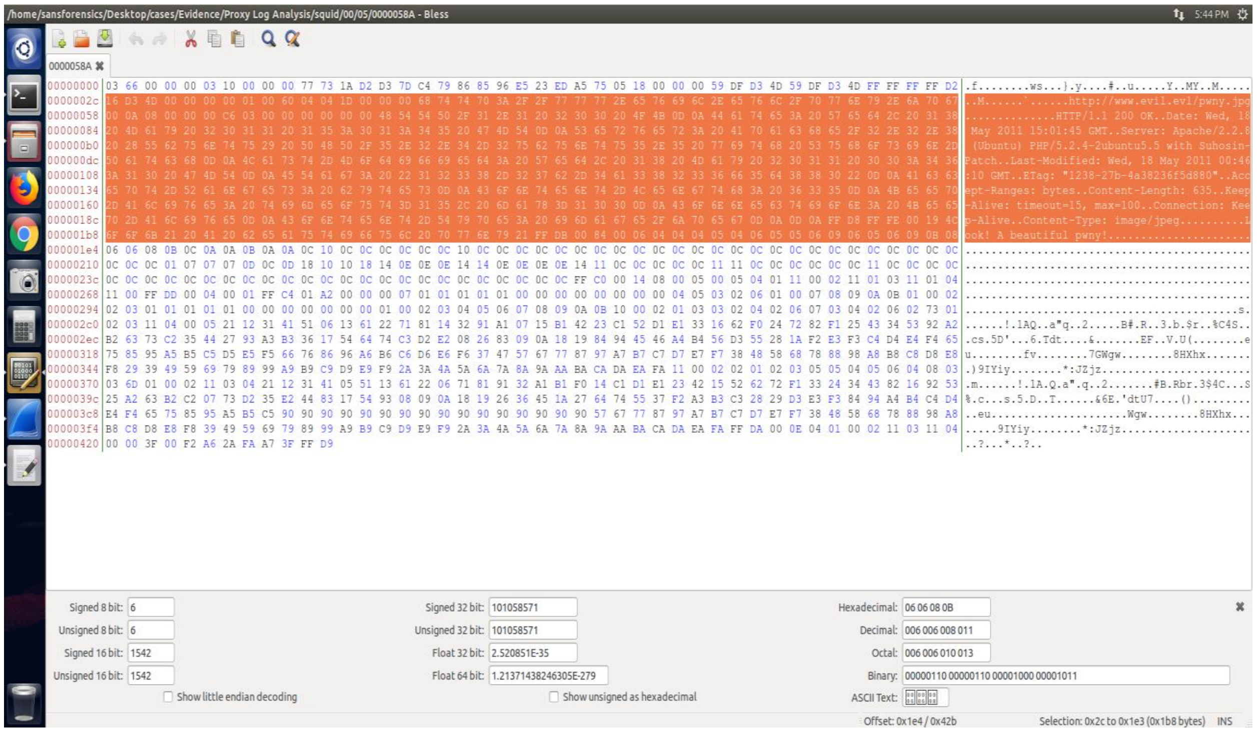
- We started with analyzing Squid proxy cache for traces of the suspicious image that we found in Snort. The Squid header we received contained a pseudo-unique ETag value of “1238-27b-4a38236f5d880.”
- Now moving on to the Proxy Log analysis → squid directory. I have used the command grep -r 1238-27b-4a38236f5d880 to check the Etag we detected of the cached pages.



```
sansforensics@siftworkstation: ~/Desktop/cases/Evidence/Proxy Log Analysis
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis$ grep -r 1238-27b-4a38236f5d880
Binary file squid/00/05/0000058A matches
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis$ 
```

- We know the cache file ‘0000058A’which contain this Etag

  To open the binary file, I am using a tool called Bless which is a binary (hexadecimal) editor. After opening the cached page in the Bless, we can find the website http://www.evil.evl/pwny.jpg and squid metadata and Etag value which we found previously. These details match with the HTTP headers within the packet we carved earlier from the snort ‘tcpdump.log’ file.



/home/sansforensics/Desktop/cases/Evidence/Proxy Log Analysis/squid/00/05/0000058A - Bless

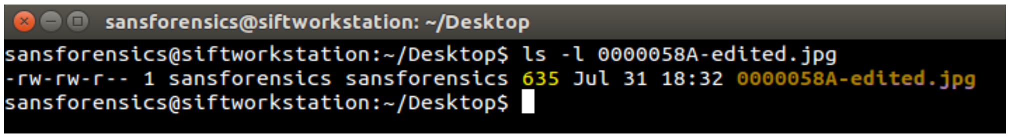
5:44 PM

0000058A

```
00000000 03 66 00 00 00 03 10 00 00 00 77 73 1A D2 D3 7D C4 79 86 85 96 E5 23 ED A5 75 05 18 00 00 00 59 DF D3 4D 59 DF D3 4D FF FF FF FF D2 .f........ws...}.y....#..u.....Y..MY..M.....
0000002c 16 D3 4D 00 00 00 00 01 00 60 04 04 1D 00 00 00 68 74 74 70 3A 2F 2F 77 77 77 2E 65 76 69 6C 2E 65 76 6C 2F 70 77 6E 79 2E 6A 70 67 ..M......`......http://www.evil.evl/pwny.jpg
00000058 00 0A 08 00 00 00 C6 03 00 00 00 00 00 00 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D 0A 44 61 74 65 3A 20 57 65 64 2C 20 31 38 ..............HTTP/1.1 200 OK..Date: Wed, 18
00000084 20 4D 61 79 20 32 30 31 31 20 31 35 3A 30 31 3A 34 35 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70 61 63 68 65 2F 32 2E 32 2E 38  May 2011 15:01:45 GMT..Server: Apache/2.2.8
000000b0 20 28 55 62 75 6E 74 75 29 20 50 48 50 2F 35 2E 32 2E 34 2D 32 75 62 75 6E 74 75 35 2E 35 20 77 69 74 68 20 53 75 68 6F 73 69 6E 2D  (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-
000000dc 50 61 74 63 68 0D 0A 4C 61 73 74 2D 4D 6F 64 69 66 69 65 64 3A 20 57 65 64 2C 20 31 38 20 4D 61 79 20 32 30 31 31 20 30 30 3A 34 36 Patch..Last-Modified: Wed, 18 May 2011 00:46
00000108 3A 31 30 20 47 4D 54 0D 0A 45 54 61 67 3A 20 22 31 32 33 38 2D 32 37 62 2D 34 61 33 38 32 33 36 66 35 64 38 38 30 22 0D 0A 41 63 63 :10 GMT..ETag: "1238-27b-4a38236f5d880"..Acc
00000134 65 70 74 2D 52 61 6E 67 65 73 3A 20 62 79 74 65 73 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 36 33 35 0D 0A 4B 65 65 70 ept-Ranges: bytes..Content-Length: 635..Keep
00000160 2D 41 6C 69 76 65 3A 20 74 69 6D 65 6F 75 74 3D 31 35 2C 20 6D 61 78 3D 31 30 30 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 -Alive: timeout=15, max=100..Connection: Kee
0000018c 70 2D 41 6C 69 76 65 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 69 6D 61 67 65 2F 6A 70 65 67 0D 0A 0D 0A FF D8 FF FE 00 19 4C p-Alive..Content-Type: image/jpeg..........L
000001b8 6F 6F 6B 21 20 41 20 62 65 61 75 74 69 66 75 6C 20 70 77 6E 79 21 FF DB 00 84 00 06 04 04 04 05 04 06 05 05 06 09 06 05 06 09 0B 08 ook! A beautiful pwny!......................
000001e4 06 06 08 0B 0C 0A 0A 0B 0A 0A 0C 10 0C 0C 0C 0C 0C 0C 10 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C ............................................
00000210 0C 0C 0C 01 07 07 07 0D 0C 0D 18 10 10 18 14 0E 0E 0E 14 14 0E 0E 0E 0E 14 11 0C 0C 0C 0C 0C 11 11 0C 0C 0C 0C 0C 0C 11 0C 0C 0C 0C ............................................
0000023c 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C FF C0 00 14 08 00 05 00 05 04 01 11 00 02 11 01 03 11 01 04 ............................................
00000268 11 00 FF DD 00 04 00 01 FF C4 01 A2 00 00 00 07 01 01 01 01 01 00 00 00 00 00 00 00 00 04 05 03 02 06 01 00 07 08 09 0A 0B 01 00 02 ............................................
00000294 02 03 01 01 01 01 01 00 00 00 00 00 00 00 01 00 02 03 04 05 06 07 08 09 0A 0B 10 00 02 01 03 03 02 04 02 06 07 03 04 02 06 02 73 01 ..........................................s.
000002c0 02 03 11 04 00 05 21 12 31 41 51 06 13 61 22 71 81 14 32 91 A1 07 15 B1 42 23 C1 52 D1 E1 33 16 62 F0 24 72 82 F1 25 43 34 53 92 A2 ......!.1AQ..a"q..2.....B#.R..3.b.$r..%C4S..
000002ec B2 63 73 C2 35 44 27 93 A3 B3 36 17 54 64 74 C3 D2 E2 08 26 83 09 0A 18 19 84 94 45 46 A4 B4 56 D3 55 28 1A F2 E3 F3 C4 D4 E4 F4 65 .cs.5D'...6.Tdt....&.......EF..V.U(........e
00000318 75 85 95 A5 B5 C5 D5 E5 F5 66 76 86 96 A6 B6 C6 D6 E6 F6 37 47 57 67 77 87 97 A7 B7 C7 D7 E7 F7 38 48 58 68 78 88 98 A8 B8 C8 D8 E8 u........fv.........7GWgw.........8HXhx.......
00000344 F8 29 39 49 59 69 79 89 99 A9 B9 C9 D9 E9 F9 2A 3A 4A 5A 6A 7A 8A 9A AA BA CA DA EA FA 11 00 02 02 01 02 03 05 05 04 05 06 04 08 03 .)9IYiy........*:JZjz.......................
00000370 03 6D 01 00 02 11 03 04 21 12 31 41 05 51 13 61 22 06 71 81 91 32 A1 B1 F0 14 C1 D1 E1 23 42 15 52 62 72 F1 33 24 34 43 82 16 92 53 .m......!.1A.Q.a".q..2.......#B.Rbr.3$4C...S
0000039c 25 A2 63 B2 C2 07 73 D2 35 E2 44 83 17 54 93 08 09 0A 18 19 26 36 45 1A 27 64 74 55 37 F2 A3 B3 C3 28 29 D3 E3 F3 84 94 A4 B4 C4 D4 %.c...s.5.D..T......&6E.'dtU7....()........
000003c8 E4 F4 65 75 85 95 A5 B5 C5 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 57 67 77 87 97 A7 B7 C7 D7 E7 F7 38 48 58 68 78 88 98 A8 ..eu.......................Wgw.........8HXhx...
000003f4 B8 C8 D8 E8 F8 39 49 59 69 79 89 99 A9 B9 C9 D9 E9 F9 2A 3A 4A 5A 6A 7A 8A 9A AA BA CA DA EA FA FF DA 00 0E 04 01 00 02 11 03 11 04 .....9IYiy........*:JZjz....................
00000420 00 00 3F 00 F2 A6 2A FA A7 3F FF D9                                                                                                 ..?...*..?..
```

```
Signed 8 bit: 6              Signed 32 bit: 101058571                Hexadecimal: 06 06 08 0B
Unsigned 8 bit: 6            Unsigned 32 bit: 101058571              Decimal: 006 006 008 011
Signed 16 bit: 1542          Float 32 bit: 2.520851E-35              Octal: 006 006 010 013
Unsigned 16 bit: 1542        Float 64 bit: 1.21371438246305E-279     Binary: 00000110 00000110 00001000 00001011
☐ Show little endian decoding  ☐ Show unsigned as hexadecimal         ASCII Text:
                                           Offset: 0x1e4 / 0x42b     Selection: 0x2c to 0x1e3 (0x1b8 bytes)  INS
```

- JPEG files begin with the magic number "0xFFD8," so we can simply search the Squid and cut all the bytes before that. Now save the modified file as 0000058A-edited.jpg. To check file size, I have used the command – ls -l 0000058A-edited.jpg to check file size and type of a file.



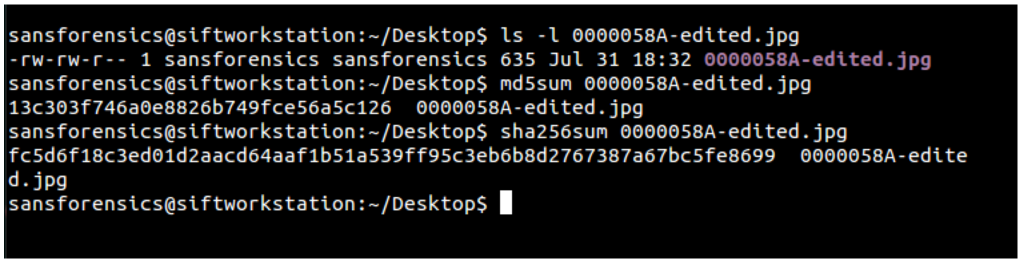
```
sansforensics@siftworkstation: ~/Desktop
sansforensics@siftworkstation:~/Desktop$ ls -l 0000058A-edited.jpg
-rw-rw-r-- 1 sansforensics sansforensics 635 Jul 31 18:32 0000058A-edited.jpg
sansforensics@siftworkstation:~/Desktop$ 
```

- To check md5 and sha256 values, I have used the command – md5sum 0000058A-edited.jpg which is the filename we saved.

  Md5 – 13c303f746a0e8826b749fce56a5c126

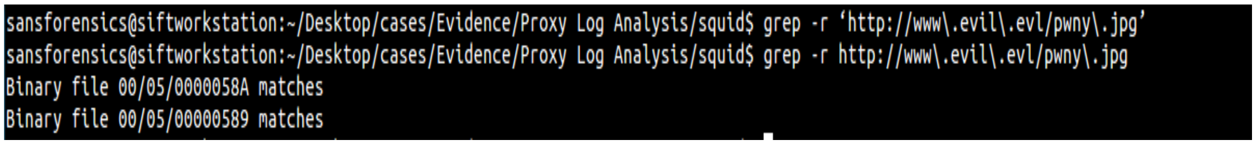
  SHA256 - fc5d6f18c3ed01d2aacd64aaf1b51a539ff95c3eb6b8d2767387a67bc5fe8699

  These checksums match with those that were previously carved from the Snort packet capture.

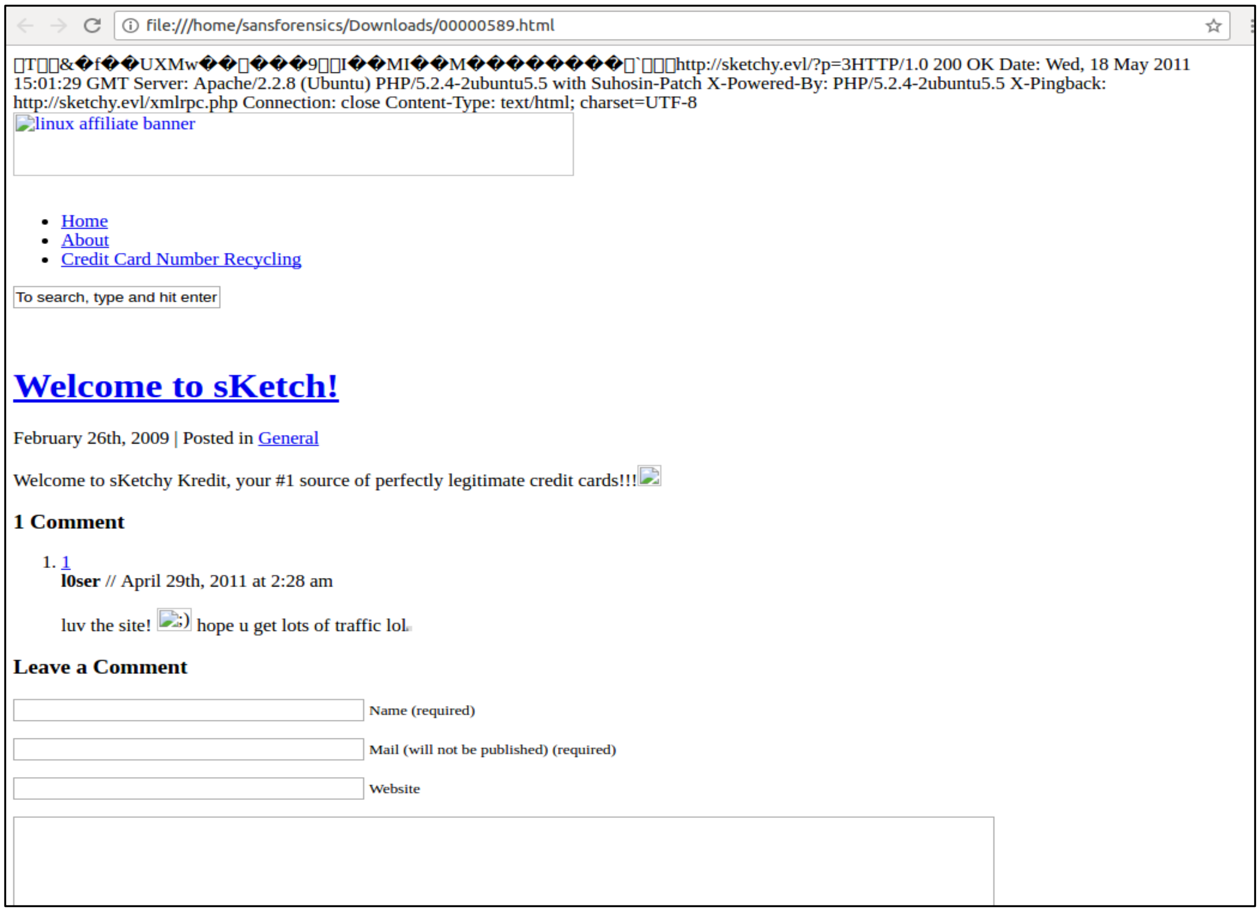

```
sansforensics@siftworkstation:~/Desktop$ ls -l 0000058A-edited.jpg
-rw-rw-r-- 1 sansforensics sansforensics 635 Jul 31 18:32 0000058A-edited.jpg
sansforensics@siftworkstation:~/Desktop$ md5sum 0000058A-edited.jpg
13c303f746a0e8826b749fce56a5c126  0000058A-edited.jpg
sansforensics@siftworkstation:~/Desktop$ sha256sum 0000058A-edited.jpg
fc5d6f18c3ed01d2aacd64aaf1b51a539ff95c3eb6b8d2767387a67bc5fe8699  0000058A-edite
d.jpg
sansforensics@siftworkstation:~/Desktop$
```

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

- To find pages that linked to the image in the proxy logs, I have used the below command Command – "grep -r 'http://www\.evil\.evl/pwny\.jpg' squid" . Only one file from this we haven't explored which is '00000589'.


```
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ grep -r 'http://www\.evil\.evl/pwny\.jpg'
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ grep -r http://www\.evil\.evl/pwny\.jpg
Binary file 00/05/0000058A matches
Binary file 00/05/00000589 matches
```

- Now open this file using the Mozilla browser and save the file using the .html extension. We found the content of file where attacker have mention comment as below in the snapshot.



file:///home/sansforensics/Downloads/00000589.html

□T□□&�f��UXMw��□���9□□I��MI��M��������□`□□□http://sketchy.evl/?p=3HTTP/1.0 200 OK Date: Wed, 18 May 2011 15:01:29 GMT Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch X-Powered-By: PHP/5.2.4-2ubuntu5.5 X-Pingback: http://sketchy.evl/xmlrpc.php Connection: close Content-Type: text/html; charset=UTF-8

linux affiliate banner

- Home
- About
- Credit Card Number Recycling

To search, type and hit enter

Welcome to sKetch!

February 26th, 2009 | Posted in General

Welcome to sKetchy Kredit, your #1 source of perfectly legitimate credit cards!!!

1 Comment

1. 1
   l0ser // April 29th, 2011 at 2:28 am

   luv the site! :) hope u get lots of traffic lol

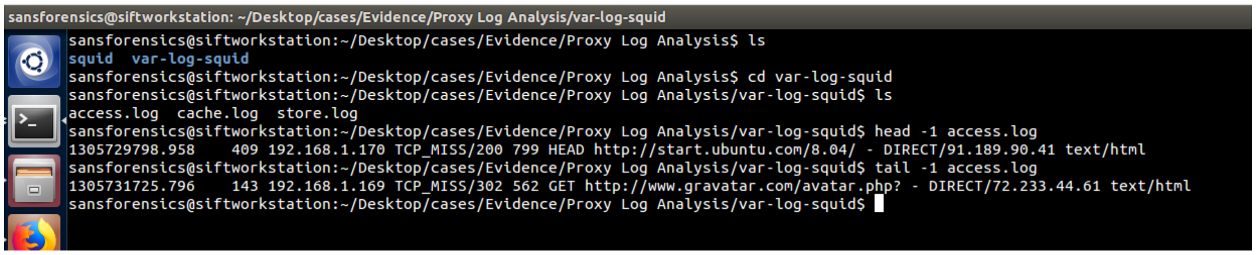
Leave a Comment

Name (required)

Mail (will not be published) (required)

Website

- We have found the timestamp of the first and last entries of the log file using the command - head -1 access.log, tail -1 access.log and have converted them into human-readable form. The first entry occurred at 14:43:18 UTC on 05/18/2011. The last entry occurred at 15:15:25 UTC on 05/18/2011.



```
sansforensics@siftworkstation: ~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis$ ls
squid  var-log-squid
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis$ cd var-log-squid
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ ls
access.log  cache.log  store.log
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ head -1 access.log
1305729798.958    409 192.168.1.170 TCP_MISS/200 799 HEAD http://start.ubuntu.com/8.04/ - DIRECT/91.189.90.41 text/html
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ tail -1 access.log
1305731725.796    143 192.168.1.169 TCP_MISS/302 562 GET http://www.gravatar.com/avatar.php? - DIRECT/72.233.44.61 text/html
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ 
```

- To isolate information about specific host 192.168.1.169, I have used the below commands -

"grep '192\.168\.1\.169' access.log > accessN.log" “wc -1 accesN.log”

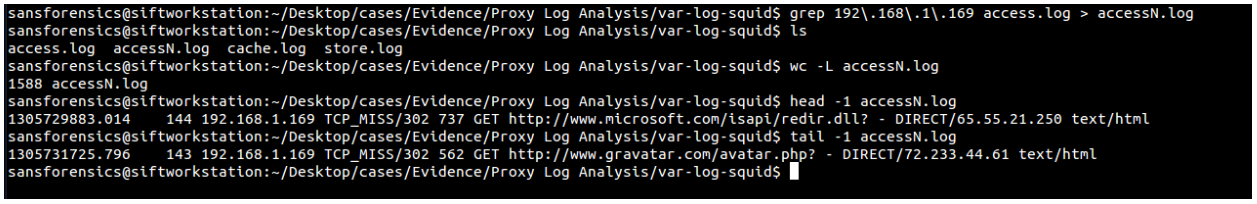
There are 1588 entries pertaining to 192.168.1.169.

- Now further examining the head and tail of a file and converting the timestamp value to human readable form below are the details-

  First entry = 14:44:43 UTC 05/18/2011.

  The destination = http://www.microsoft.com/isapi/redir.dll?, suggesting that 192.168.1.169 is configured with Microsoft software, such as Internet Explorer and Windows.

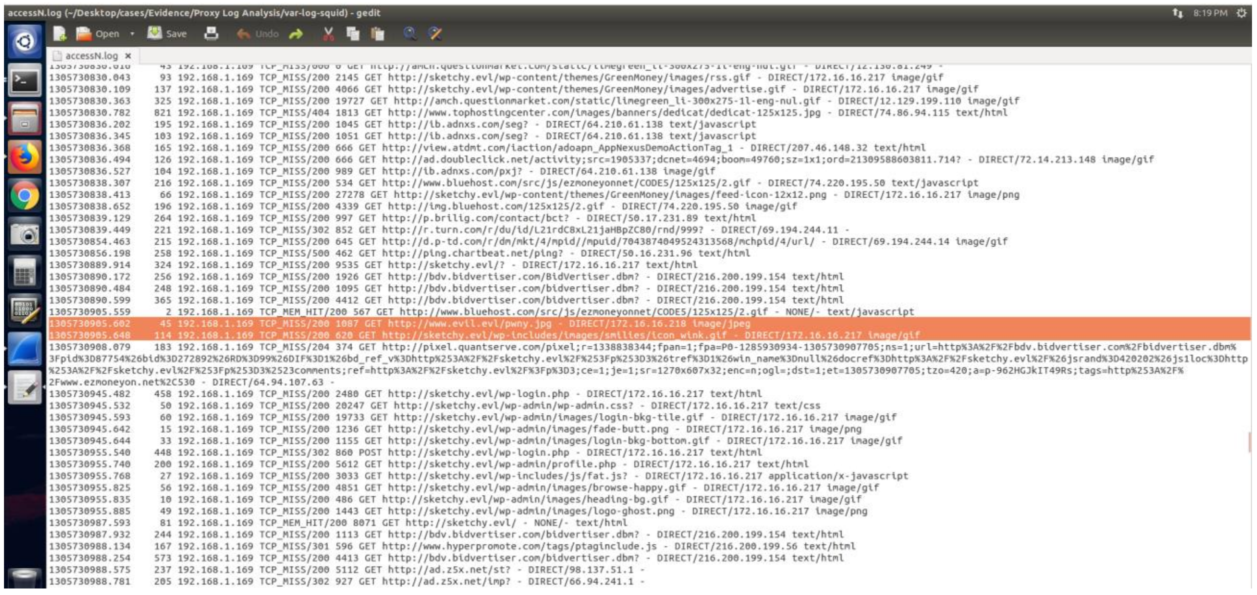
  Last entry = 15:15:25 UTC 05/11/2011.



```
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ grep 192\.168\.1\.169 access.log > accessN.log
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ ls
access.log  accessN.log  cache.log  store.log
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ wc -L accessN.log
1588 accessN.log
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ head -1 accessN.log
1305729883.014    144 192.168.1.169 TCP_MISS/302 737 GET http://www.microsoft.com/isapi/redir.dll? - DIRECT/65.55.21.250 text/html
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$ tail -1 accessN.log
1305731725.796    143 192.168.1.169 TCP_MISS/302 562 GET http://www.gravatar.com/avatar.php? - DIRECT/72.233.44.61 text/html
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid$
```

- Now to further confirm NIDS alert, I have opened access.log file we have created using gedit.

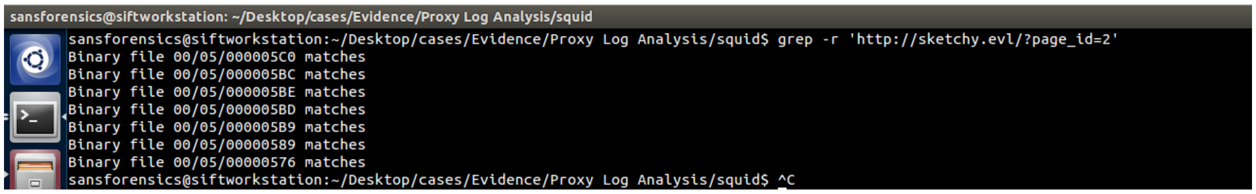
  The timestamp is 15:01:45 UTC 05/18/2011. After adjusting for the time zone, this is consistent with the previously established timeline based on Snort logs.


accessN.log (~/Desktop/cases/Evidence/Proxy Log Analysis/var-log-squid) - gedit
8:19 PM
Open
Save
Undo
accessN.log

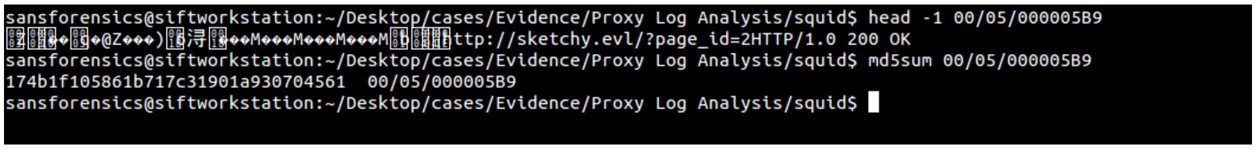
```
1305730830.043     93 192.168.1.169 TCP_MISS/200 2145 GET http://sketchy.evl/wp-content/themes/GreenMoney/images/rss.gif - DIRECT/172.16.16.217 image/gif
1305730830.109    137 192.168.1.169 TCP_MISS/200 4066 GET http://sketchy.evl/wp-content/themes/GreenMoney/images/advertise.gif - DIRECT/172.16.16.217 image/gif
1305730830.363    325 192.168.1.169 TCP_MISS/200 19727 GET http://amch.questionmarket.com/static/limegreen_li-300x275-1l-eng-nul.gif - DIRECT/12.129.199.110 image/gif
1305730830.782    821 192.168.1.169 TCP_MISS/404 1813 GET http://www.tophostingcenter.com/images/banners/dedicat/dedicat-125x125.jpg - DIRECT/74.86.94.115 text/html
1305730836.202    195 192.168.1.169 TCP_MISS/200 1045 GET http://ib.adnxs.com/seg? - DIRECT/64.210.61.138 text/javascript
1305730836.345    103 192.168.1.169 TCP_MISS/200 1051 GET http://ib.adnxs.com/seg? - DIRECT/64.210.61.138 text/javascript
1305730836.368    165 192.168.1.169 TCP_MISS/200 666 GET http://view.atdmt.com/iaction/adoapn_AppNexusDemoActionTag_1 - DIRECT/207.46.148.32 text/html
1305730836.494    126 192.168.1.169 TCP_MISS/200 666 GET http://ad.doubleclick.net/activity;src=1905337;dcnet=4694;boom=49760;sz=1x1;ord=21309588603811.714? - DIRECT/72.14.213.148 image/gif
1305730836.527    104 192.168.1.169 TCP_MISS/200 989 GET http://ib.adnxs.com/pxj? - DIRECT/64.210.61.138 image/gif
1305730838.307    216 192.168.1.169 TCP_MISS/200 534 GET http://www.bluehost.com/src/js/ezmoneyonnet/CODE5/125x125/2.gif - DIRECT/74.220.195.50 text/javascript
1305730838.413     66 192.168.1.169 TCP_MISS/200 27278 GET http://sketchy.evl/wp-content/themes/GreenMoney/images/feed-icon-12x12.png - DIRECT/172.16.16.217 image/png
1305730838.652    196 192.168.1.169 TCP_MISS/200 4339 GET http://img.bluehost.com/125x125/2.gif - DIRECT/74.220.195.50 image/gif
1305730839.129    264 192.168.1.169 TCP_MISS/200 997 GET http://p.brilig.com/contact/bct? - DIRECT/50.17.231.89 text/html
1305730839.449    221 192.168.1.169 TCP_MISS/302 852 GET http://r.turn.com/r/du/id/L21rdC8xL21jaHBpZC80/rnd/999? - DIRECT/69.194.244.11 -
1305730854.463    215 192.168.1.169 TCP_MISS/200 645 GET http://d.p-td.com/r/dm/mkt/4/mpid//mpuid/7043874049524313568/mchpid/4/url/ - DIRECT/69.194.244.14 image/gif
1305730856.198    258 192.168.1.169 TCP_MISS/500 462 GET http://ping.chartbeat.net/ping? - DIRECT/50.16.231.96 text/html
1305730889.914    324 192.168.1.169 TCP_MISS/200 9535 GET http://sketchy.evl/? - DIRECT/172.16.16.217 text/html
1305730890.172    256 192.168.1.169 TCP_MISS/200 1926 GET http://bdv.bidvertiser.com/BidVertiser.dbm? - DIRECT/216.200.199.154 text/html
1305730890.484    248 192.168.1.169 TCP_MISS/200 1095 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/216.200.199.154 text/html
1305730890.599    365 192.168.1.169 TCP_MISS/200 4412 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/216.200.199.154 text/html
1305730905.559      2 192.168.1.169 TCP_MEM_HIT/200 567 GET http://www.bluehost.com/src/js/ezmoneyonnet/CODE5/125x125/2.gif - NONE/- text/javascript
1305730905.602     45 192.168.1.169 TCP_MISS/200 1087 GET http://www.evil.evl/pwny.jpg - DIRECT/172.16.16.218 image/jpeg
1305730905.648    114 192.168.1.169 TCP_MISS/200 620 GET http://sketchy.evl/wp-includes/images/smilies/icon_wink.gif - DIRECT/172.16.16.217 image/gif
1305730908.079    183 192.168.1.169 TCP_MISS/204 374 GET http://pixel.quantserve.com/pixel;r=1338838344;fpan=1;fpa=P0-1285930934-1305730907705;ns=1;url=http%3A%2F%2Fbdv.bidvertiser.com%2Fbidvertiser.dbm%
3Fpid%3D87754%26bid%3D272892%26RD%3D99%26DIF%3D1%26bd_ref_v%3Dhttp%253A%2F%2Fsketchy.evl%2F%253Fp%253D3%26tref%3D1%26win_name%3Dnull%26docref%3Dhttp%3A%2F%2Fsketchy.evl%2F%26jsrand%3D420202%26js1loc%3Dhttp
%253A%2F%2Fsketchy.evl%2F%253Fp%253D3%2523comments;ref=http%3A%2F%2Fsketchy.evl%2F%3Fp%3D3;ce=1;je=1;sr=1270x607x32;enc=n;ogl=;dst=1;et=1305730907705;tzo=420;a=p-962HGJkIT49Rs;tags=http%253A%2F%
2Fwww.ezmoneyon.net%2C530 - DIRECT/64.94.107.63 -
1305730945.482    458 192.168.1.169 TCP_MISS/200 2480 GET http://sketchy.evl/wp-login.php - DIRECT/172.16.16.217 text/html
1305730945.532     50 192.168.1.169 TCP_MISS/200 20247 GET http://sketchy.evl/wp-admin/wp-admin.css? - DIRECT/172.16.16.217 text/css
1305730945.593     60 192.168.1.169 TCP_MISS/200 19733 GET http://sketchy.evl/wp-admin/images/login-bkg-tile.gif - DIRECT/172.16.16.217 image/gif
1305730945.642     15 192.168.1.169 TCP_MISS/200 1236 GET http://sketchy.evl/wp-admin/images/fade-butt.png - DIRECT/172.16.16.217 image/png
1305730945.644     33 192.168.1.169 TCP_MISS/200 1155 GET http://sketchy.evl/wp-admin/images/login-bkg-bottom.gif - DIRECT/172.16.16.217 image/gif
1305730955.540    448 192.168.1.169 TCP_MISS/302 860 POST http://sketchy.evl/wp-login.php - DIRECT/172.16.16.217 text/html
1305730955.740    200 192.168.1.169 TCP_MISS/200 5612 GET http://sketchy.evl/wp-admin/profile.php - DIRECT/172.16.16.217 text/html
1305730955.768     27 192.168.1.169 TCP_MISS/200 3033 GET http://sketchy.evl/wp-includes/js/fat.js? - DIRECT/172.16.16.217 application/x-javascript
1305730955.825     56 192.168.1.169 TCP_MISS/200 4851 GET http://sketchy.evl/wp-admin/images/browse-happy.gif - DIRECT/172.16.16.217 image/gif
1305730955.835     10 192.168.1.169 TCP_MISS/200 486 GET http://sketchy.evl/wp-admin/images/heading-bg.gif - DIRECT/172.16.16.217 image/gif
1305730955.885     49 192.168.1.169 TCP_MISS/200 1443 GET http://sketchy.evl/wp-admin/images/logo-ghost.png - DIRECT/172.16.16.217 image/png
1305730987.593     81 192.168.1.169 TCP_MEM_HIT/200 8071 GET http://sketchy.evl/ - NONE/- text/html
1305730987.932    244 192.168.1.169 TCP_MISS/200 1113 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/216.200.199.154 text/html
1305730988.134    167 192.168.1.169 TCP_MISS/301 596 GET http://www.hyperpromote.com/tags/ptaginclude.js - DIRECT/216.200.199.56 text/html
1305730988.254    573 192.168.1.169 TCP_MISS/200 4413 GET http://bdv.bidvertiser.com/bidvertiser.dbm? - DIRECT/216.200.199.154 text/html
1305730988.575    237 192.168.1.169 TCP_MISS/200 5112 GET http://ad.z5x.net/st? - DIRECT/98.137.51.1 -
1305730988.781    205 192.168.1.169 TCP_MISS/302 927 GET http://ad.z5x.net/imp? - DIRECT/66.94.241.1 -
```

- To find more details related to domain evil.evil and sketchy.evil, I have used grep command to check the file location of sketchy.evil file wuth different user ID's. In the below user ID, I have found some matches and after manually checking all matches I have found some content in 000005B9

  grep -r 'http://sketchy.evl/?page_id=2'

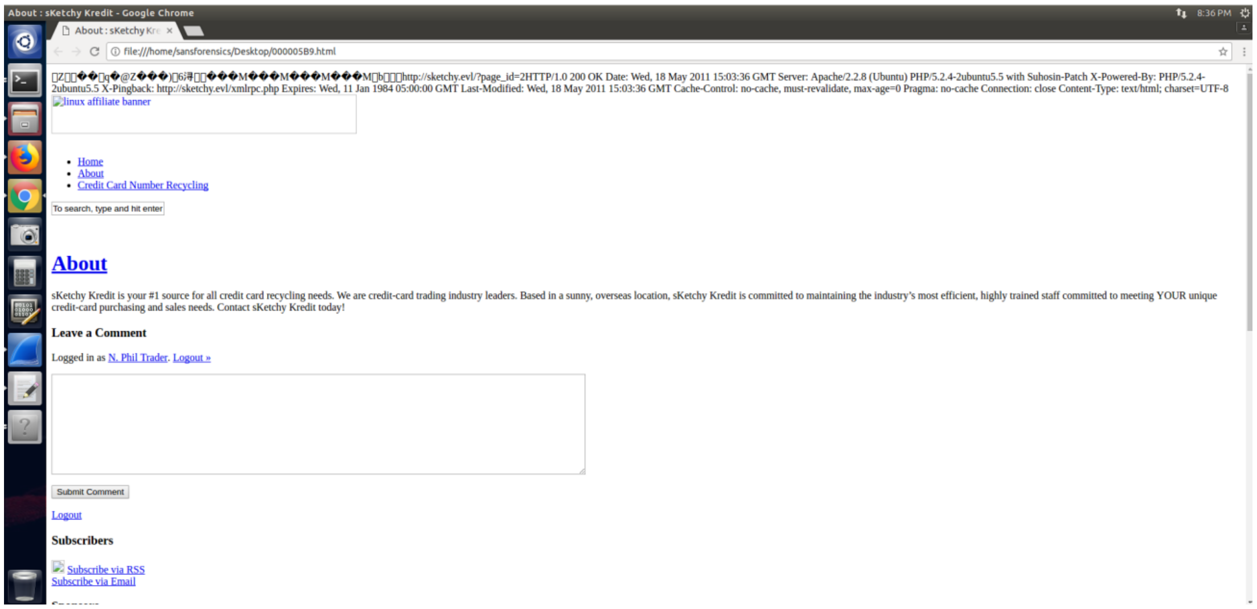


```
sansforensics@siftworkstation: ~/Desktop/cases/Evidence/Proxy Log Analysis/squid
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ grep -r 'http://sketchy.evl/?page_id=2'
Binary file 00/05/000005C0 matches
Binary file 00/05/000005BC matches
Binary file 00/05/000005BE matches
Binary file 00/05/000005BD matches
Binary file 00/05/000005B9 matches
Binary file 00/05/00000589 matches
Binary file 00/05/00000576 matches
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ ^C
```



```
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ head -1 00/05/000005B9
�Z��g�@Z���)�6浔�M���M���M���M�b��ttp://sketchy.evl/?page_id=2HTTP/1.0 200 OK
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ md5sum 00/05/000005B9
174b1f105861b717c31901a930704561  00/05/000005B9
sansforensics@siftworkstation:~/Desktop/cases/Evidence/Proxy Log Analysis/squid$ 
```

- After checking all the files we found details in one file '000005B9'. It shows that a user called "N. Phil Trader" was logged in when the page was cached.



About : sKetchy Kredit - Google Chrome
8:36 PM
About : sKetchy Kre
file:///home/sansforensics/Desktop/000005B9.html

□Z□□��□q�@Z���)□6浔□□���M���M���M���M□b□□□http://sketchy.evl/?page_id=2HTTP/1.0 200 OK Date: Wed, 18 May 2011 15:03:36 GMT Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch X-Powered-By: PHP/5.2.4-2ubuntu5.5 X-Pingback: http://sketchy.evl/xmlrpc.php Expires: Wed, 11 Jan 1984 05:00:00 GMT Last-Modified: Wed, 18 May 2011 15:03:36 GMT Cache-Control: no-cache, must-revalidate, max-age=0 Pragma: no-cache Connection: close Content-Type: text/html; charset=UTF-8

linux affiliate banner

- Home
- About
- Credit Card Number Recycling

To search, type and hit enter

About

sKetchy Kredit is your #1 source for all credit card recycling needs. We are credit-card trading industry leaders. Based in a sunny, overseas location, sKetchy Kredit is committed to maintaining the industry's most efficient, highly trained staff committed to meeting YOUR unique credit-card purchasing and sales needs. Contact sKetchy Kredit today!

Leave a Comment

Logged in as N. Phil Trader. Logout »

Submit Comment

Logout

Subscribers

Subscribe via RSS
Subscribe via Email

Citations -

1. CyberSophia. (n.d.). grep command in Linux. CyberSophia. https://cybersophia.net/linux/grep-command-in-linux/
2. SolarWinds. (n.d.). Squid Proxy Server Log Analysis. Retrieved from https://www.solarwinds.com/security-event-manager/use-cases/squid-proxy-server-log-analysis
3. Lenaerts-Bergmans, B. (2023, April 24). Snort Rules: What You Need to Know. Retrieved from https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/snort-rules/
4. Sapphire. (2023, March 21). Snort Rules: Examples. Retrieved from https://www.sapphire.net/security/snort-rules-examples/