Case Name: WK4b

[redacted]

Examiner Name: Candace [redacted]

Table of Contents

List of Illustrative Materials

Tables

Figures

Executive Summary

Background

It is known that a credit card number recycling program was recently started by Inter0ptic. A payment processor company called MacDaddy has deployed Snort NIDS to detect anomalous inbound and outbound events. An alert was logged at 08:01:45 on 5/18/11 regarding inbound executable code sent to port 80 for inside host 192.168.1.169 from external host 172.16.16.218.

Request

The examiner was requested to determine whether the alert is true or false by examining the alert's data for context, compare the alert to the rule, retrieve the packet that triggered the alert, and compare the rule to the retrieved packet. Any malicious activities linked to the event should be investigated via timeline construction.

Findings

The alert was a true positive, as the packet flagged contained a repeated sequence of 0x90 characters, which is a custom rule in the configuration files created by local staff. The packet that triggered the alert contained a JPEG image was extracted for further analysis. A rough timeline of the events is as follows:
07:45:09 - Relevant NIDS alerts begin. These include alerts related to 192.168.1.169
08:01:45 - Reported NIDS alert occurs. It is the only alert related to 172.16.16.218.
08:04:28-08:04:38 - 192.168.1.169 sends packets to other internal hosts, triggering alerts.
08:15:08 - Relevant NIDS alerts (related to 192.168.1.169 end.

Evidence

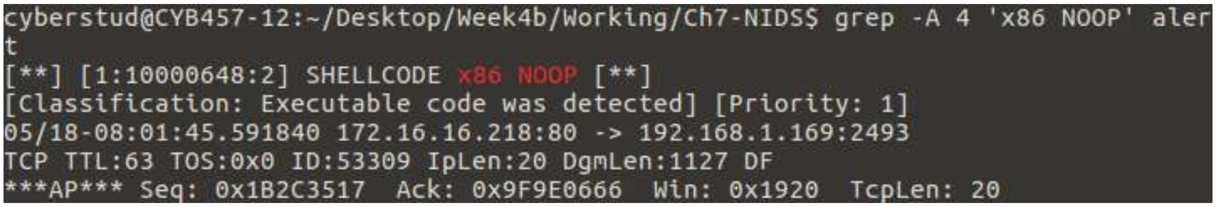
Table 1 outlines the evidence items of this case.

Description	Designation	Filename	MD5 Hash
Evidence Provided	Preservation Copy	Ch7-NIDS.zip	404DD2657EBC7077D44A86EC6877CF41
Evidence Created	Working Copy	Ch7-NIDSXX	404DD2657EBC7077D44A86EC6877CF41
Evidence Examined	Working Copy	Ch7-NIDSXX	404DD2657EBC7077D44A86EC6877CF41

Table 1: Case Evidence Items

Analysis

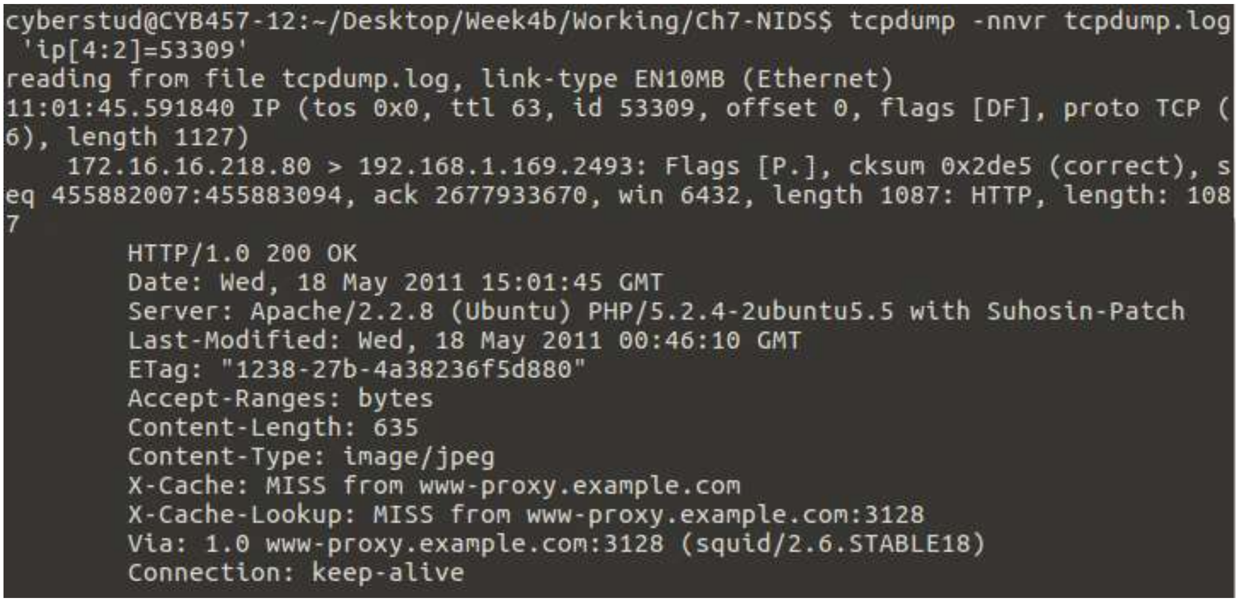
Examiner entered command “grep -A 4 ‘x86 N00P’ alert” in a terminal in the *Desktop/Week4a/Working/CH7-NIDS* directory. The result matched the information that security staff initially provided, but also confirmed just one instance of this alert occurred. The alert indicates that a remote server 172.16.16.218:80 sent traffic to local system 192.168.1.169:2493 and the NIDS flagged it because the traffic contained executable code.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -A 4 'x86 NOOP' aler
t
[**] [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
```

Figure 1: Snort alert

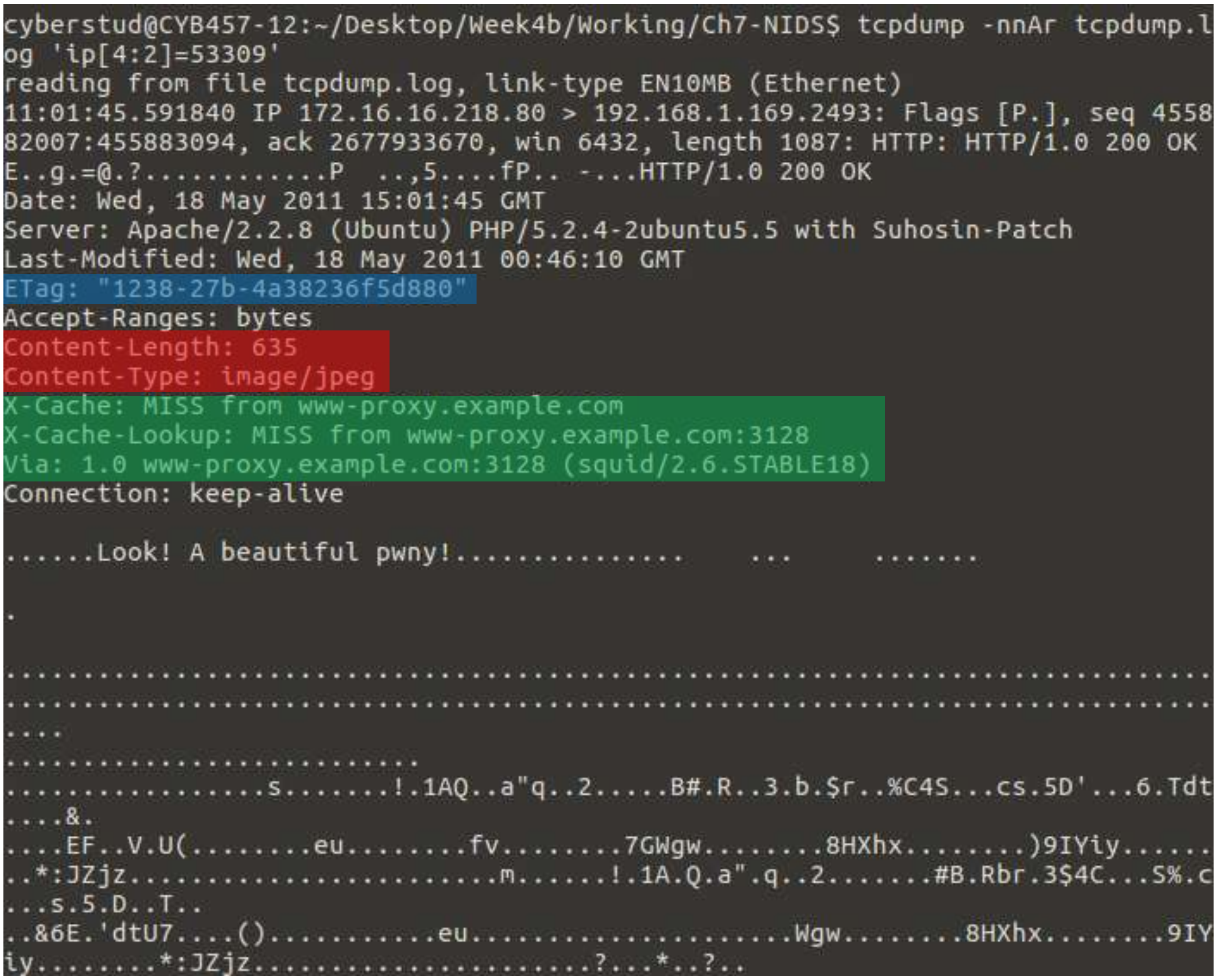
Examiner entered command “tcpdump -nnvr tcpdump.log ‘ip[4:2]=53309’” to view the packet in question. Both the TCP sequence number (455882007 or 0x1B2C3517) and acknowledgment number (2677933670 or 0x9F9E0666) seen in figure 2 match the alert, confirming this to be the correct packet.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ tcpdump -nnvr tcpdump.log
 'ip[4:2]=53309'
reading from file tcpdump.log, link-type EN10MB (Ethernet)
11:01:45.591840 IP (tos 0x0, ttl 63, id 53309, offset 0, flags [DF], proto TCP (
6), length 1127)
    172.16.16.218.80 > 192.168.1.169.2493: Flags [P.], cksum 0x2de5 (correct), s
eq 455882007:455883094, ack 2677933670, win 6432, length 1087: HTTP, length: 108
7
        HTTP/1.0 200 OK
        Date: Wed, 18 May 2011 15:01:45 GMT
        Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch
        Last-Modified: Wed, 18 May 2011 00:46:10 GMT
        ETag: "1238-27b-4a38236f5d880"
        Accept-Ranges: bytes
        Content-Length: 635
        Content-Type: image/jpeg
        X-Cache: MISS from www-proxy.example.com
        X-Cache-Lookup: MISS from www-proxy.example.com:3128
        Via: 1.0 www-proxy.example.com:3128 (squid/2.6.STABLE18)
        Connection: keep-alive
```

Figure 2: Packet 53309

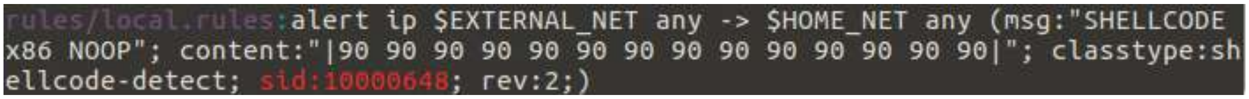
Examiner entered command “tcpdump -nnAr tcpdump.log ‘ip[4:2]=53309’” in terminal. The results, shown in figure 3, reveal the packet contains a 635-byte JPEG image (highlighted in red). The HTTP headers (highlighted in green) show the web page was provided through a web proxy where it was not already in the cache. The ETag of the packet’s content (blue) is “1238-27b-4a38236f5d880”.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ tcpdump -nnAr tcpdump.l
og 'ip[4:2]=53309'
reading from file tcpdump.log, link-type EN10MB (Ethernet)
11:01:45.591840 IP 172.16.16.218.80 > 192.168.1.169.2493: Flags [P.], seq 4558
82007:455883094, ack 2677933670, win 6432, length 1087: HTTP: HTTP/1.0 200 OK
E..g.=@.?............P  ..,5....fP.. -...HTTP/1.0 200 OK
Date: Wed, 18 May 2011 15:01:45 GMT
Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch
Last-Modified: Wed, 18 May 2011 00:46:10 GMT
ETag: "1238-27b-4a38236f5d880"
Accept-Ranges: bytes
Content-Length: 635
Content-Type: image/jpeg
X-Cache: MISS from www-proxy.example.com
X-Cache-Lookup: MISS from www-proxy.example.com:3128
Via: 1.0 www-proxy.example.com:3128 (squid/2.6.STABLE18)
Connection: keep-alive

......Look! A beautiful pwny!..............    ...     .......

.

..............................................................................
..............................................................................
....
...........................
................s.......!.1AQ..a"q..2.....B#.R..3.b.$r..%C4S...cs.5D'...6.Tdt
....&.
....EF..V.U(........eu........fv........7GWgw........8HXhx........)9IYiy......
..*:JZjz......................m......!.1A.Q.a".q..2.......#B.Rbr.3$4C...S%.c
...s.5.D..T..
..&6E.'dtU7....()...........eu.....................Wgw........8HXhx........9IY
iy........*:JZjz......................?...*..?..
```

Figure 3: Packet 53309 contents

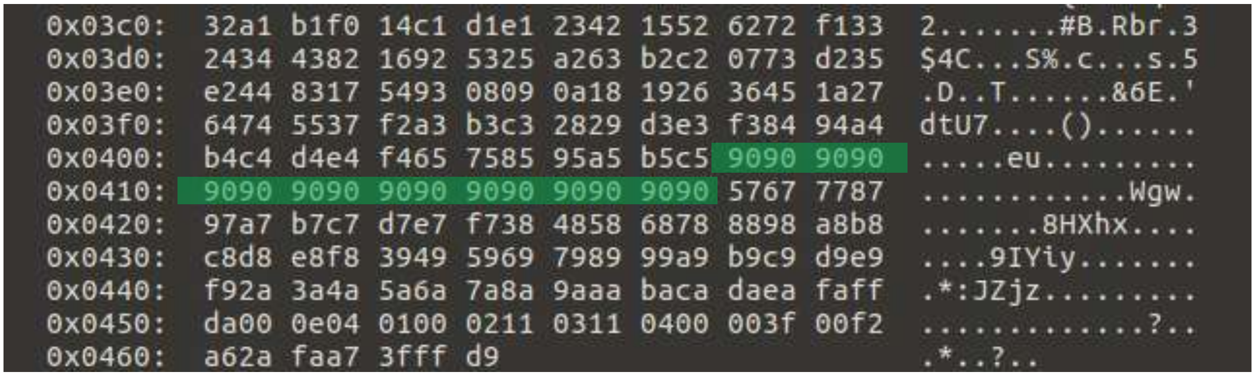
Examiner entered command “grep -r sid:10000648 rules” in terminal to view the snort rule flagged in the alert (found in figure 1). The rule, seen in figure 4 below, is in the local.rules file and has an SID greater than 1,000,000; it is a custom rule created by local staff. It is designed to flag any inbound content with 14 bytes of 0x90 or more - a common feature of a buffer overflow attack.



```
rules/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:sh
ellcode-detect; sid:10000648; rev:2;)
```

Figure 4: Snort rule 10000648

Examiner entered command “tcpdump -nnXr tcpdump.log ‘ip[4:2] = 53309’ in terminal to view the packet in hexadecimal and ASCII. Figure 5 shows a truncated selection, where consecutive 0x90 characters can be found.

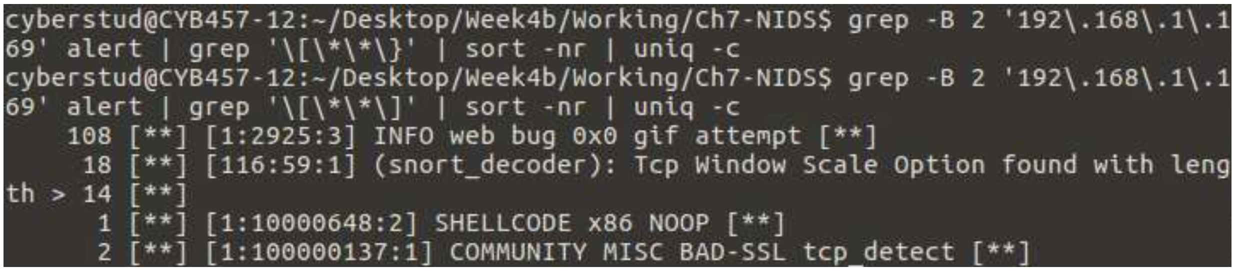


```
0x03c0:  32a1 b1f0 14c1 d1e1 2342 1552 6272 f133  2.......#B.Rbr.3
0x03d0:  2434 4382 1692 5325 a263 b2c2 0773 d235  $4C...S%.c...s.5
0x03e0:  e244 8317 5493 0809 0a18 1926 3645 1a27  .D..T......&6E.'
0x03f0:  6474 5537 f2a3 b3c3 2829 d3e3 f384 94a4  dtU7....()......
0x0400:  b4c4 d4e4 f465 7585 95a5 b5c5 9090 9090  .....eu.........
0x0410:  9090 9090 9090 9090 9090 9090 5767 7787  ............Wgw.
0x0420:  97a7 b7c7 d7e7 f738 4858 6878 8898 a8b8  .......8HXhx....
0x0430:  c8d8 e8f8 3949 5969 7989 99a9 b9c9 d9e9  ....9IYiy.......
0x0440:  f92a 3a4a 5a6a 7a8a 9aaa baca daea faff  .*:JZjz.........
0x0450:  da00 0e04 0100 0211 0311 0400 003f 00f2  .............?..
0x0460:  a62a faa7 3fff d9                        .*..?..
```

Figure 5: 0x90 occurrence in packet

Examiner opened Wireshark and applied display filter “ip.id==53309”, then exported the jpeg image from the packet as a file called *evidenceXX(.bin).* Its MD5 hash value is 13C303F746A0E8826B749FCE56A5C126.

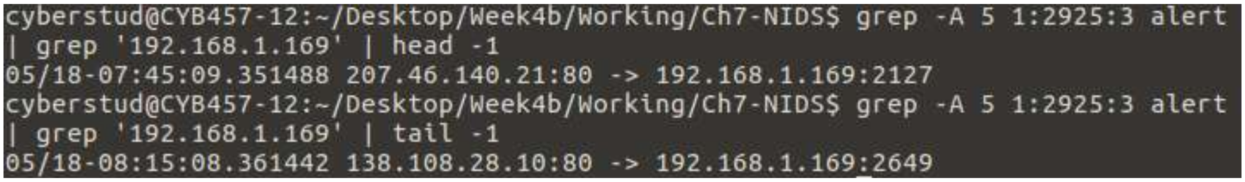
Examiner entered command “grep -B 2 ‘192\.168\.1\.169’ alert | grep ‘\[**\]’ | sort -nr | uniq -c”, which displayed alerts relating to the target IP address 192.168.1.169, as shown in figure 6. 108 instances of web bugs were identified.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -B 2 '192\.168\.1\.1
69' alert | grep '\[\*\*\}' | sort -nr | uniq -c
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -B 2 '192\.168\.1\.1
69' 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 [**]
```

Figure 6: Alerts related to 192.168.1.169

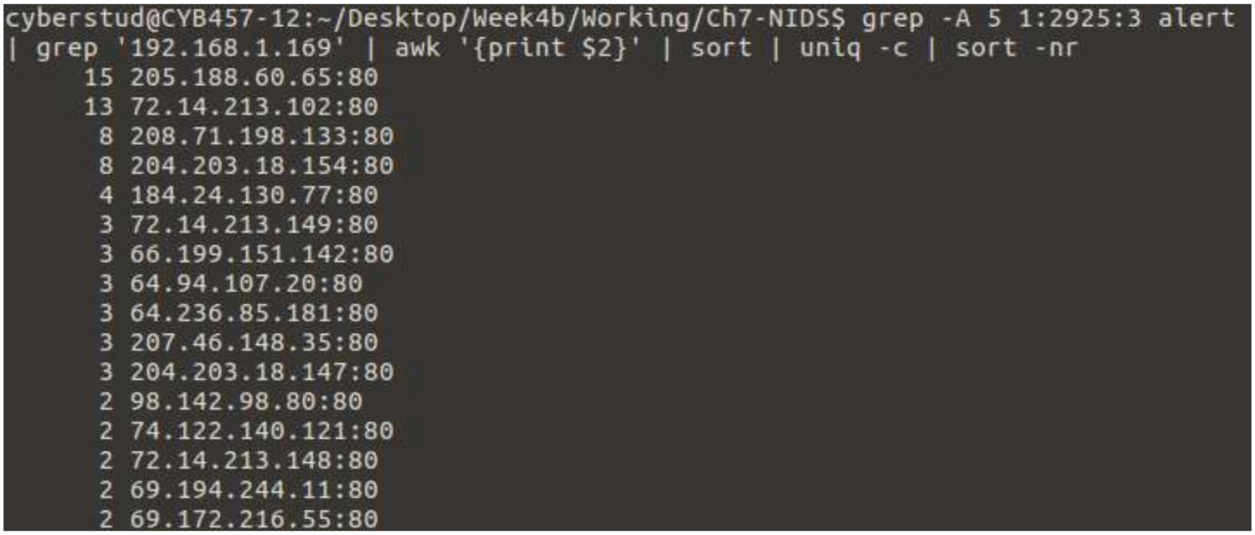
Examiner entered command “grep -A 5 1:2925:3 alert | grep ‘192.168.1.169’ | head -1” and “grep -A 5 1:2925:3 alert | grep ‘192.168.1.169’ | tail -1”. The results in figure 7 show that the web bug alerts took place over a half hour of web browsing.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -A 5 1:2925:3 alert
| grep '192.168.1.169' | head -1
05/18-07:45:09.351488 207.46.140.21:80 -> 192.168.1.169:2127
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -A 5 1:2925:3 alert
| grep '192.168.1.169' | tail -1
05/18-08:15:08.361442 138.108.28.10:80 -> 192.168.1.169:2649
```

Figure 7: Timespan of web bug alerts

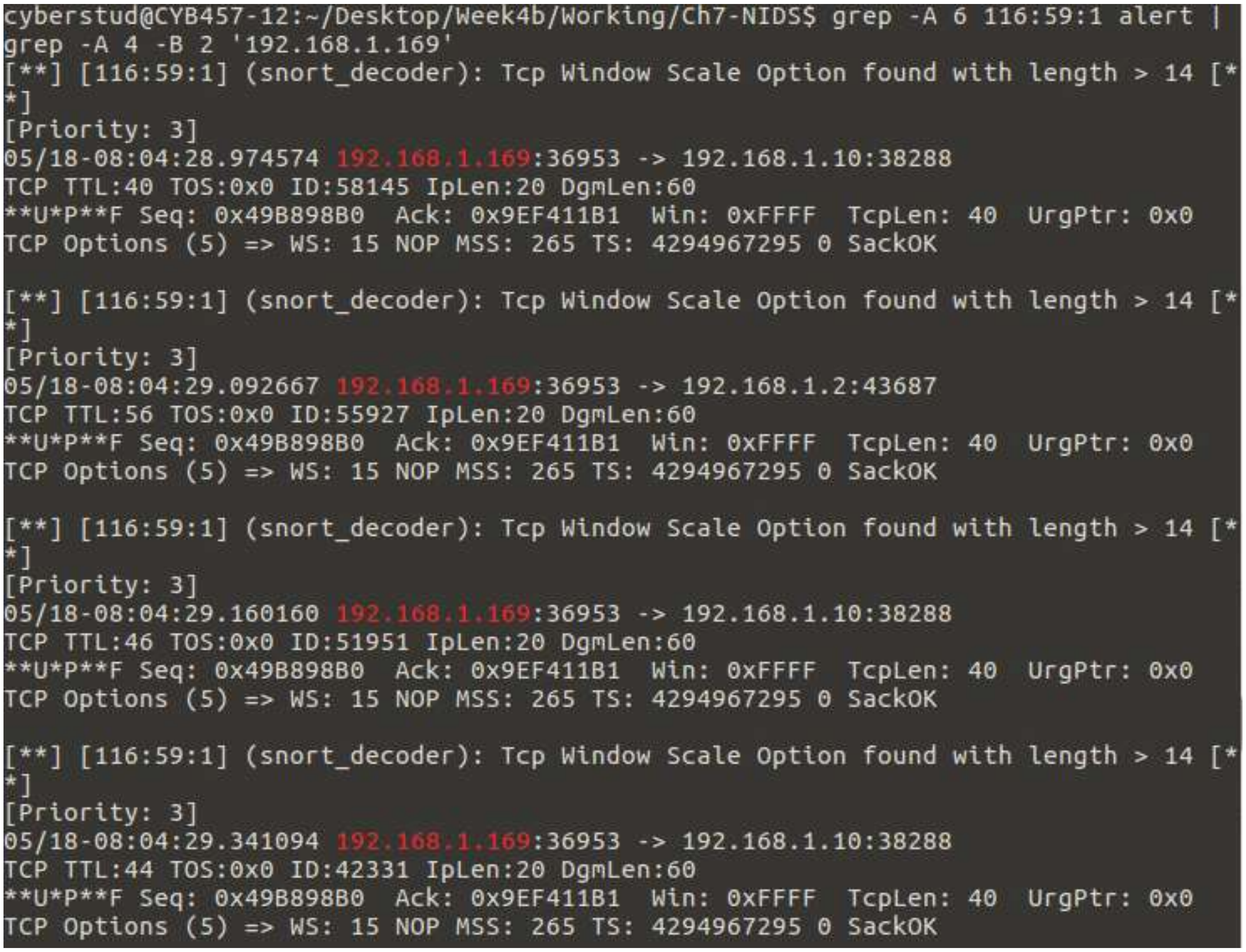
Examiner entered command “grep -A 1:2925:3 alert | grep ‘192\.168\.1\.169’ | awk ‘{print $2}’ | sort | uniq -c | sort -nr”. The results, truncated in figure 8, reveal the web bugs came from 42 different sources.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -A 5 1:2925:3 alert
| grep '192.168.1.169' | awk '{print $2}' | sort | uniq -c | sort -nr
     15 205.188.60.65:80
     13 72.14.213.102:80
      8 208.71.198.133:80
      8 204.203.18.154:80
      4 184.24.130.77:80
      3 72.14.213.149:80
      3 66.199.151.142:80
      3 64.94.107.20:80
      3 64.236.85.181:80
      3 207.46.148.35:80
      3 204.203.18.147:80
      2 98.142.98.80:80
      2 74.122.140.121:80
      2 72.14.213.148:80
      2 69.194.244.11:80
      2 69.172.216.55:80
```

Figure 8: Sources of web bugs

To view more information regarding the second alert (Tcp Window Scale Option…), examiner entered command “grep -A 6 116:59:1 alert | grep -A 4 -B 2 ‘192.168.1.169’”. A selection of the alert occurrences can be seen in figure 9. The values (such as identical sequence and acknowledgment numbers) don’t make sense, which could be evidence of a reconnaissance tool in action. The packets were sent to four hosts: 192.168.1.170, 192.168.1.30, 192.168.1.10, and 192.168.1.2.



```
cyberstud@CYB457-12:~/Desktop/Week4b/Working/Ch7-NIDS$ grep -A 6 116:59:1 alert |
grep -A 4 -B 2 '192.168.1.169'
[**] [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

[**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14 [*
*]
[Priority: 3]
05/18-08:04:29.092667 192.168.1.169:36953 -> 192.168.1.2:43687
TCP TTL:56 TOS:0x0 ID:55927 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

[**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14 [*
*]
[Priority: 3]
05/18-08:04:29.160160 192.168.1.169:36953 -> 192.168.1.10:38288
TCP TTL:46 TOS:0x0 ID:51951 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

[**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14 [*
*]
[Priority: 3]
05/18-08:04:29.341094 192.168.1.169:36953 -> 192.168.1.10:38288
TCP TTL:44 TOS:0x0 ID:42331 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
```

Figure 9: Occurrences of TCP Window Scale Option alert

Conclusion

The examiner was requested to determine whether the alert is true or false by examining the alert's data for context, compare the alert to the rule, retrieve the packet that triggered the alert, and compare the rule to the retrieved packet. The alert was a true positive, as the packet flagged contained a repeated sequence of 0x90 characters, which is a custom rule in the configuration files created by local staff. The packet that triggered the alert contained a JPEG image was extracted for further analysis. The events were put into a timeline:

07:45:09 - Relevant NIDS alerts begin. These include alerts related to 192.168.1.169
08:01:45 - Reported NIDS alert occurs. It is the only alert related to 172.16.16.218.
08:04:28-08:04:38 - 192.168.1.169 sends packets to other internal hosts, triggering alerts.
08:15:08 - Relevant NIDS alerts (related to 192.168.1.169 end.

The timeline suggests that a "drive-by" exploit occurred, which initiated a pattern of reconnaissance by the infected system. Further investigation is necessary to determine whether the timing of the scan is purely coincidental.

Appendix

Appendix A: Examiner Workstation Specifications

- [redacted]
- System date/time is consistent with the time zone listed above, as verified by: http://nist.time.gov/.

Appendix B: Virtual Machine (VM) Specifications

- Virtual Machine Name: CYB457-12
- Operating System Name: Ubuntu
- System Make/Model: VMware, Inc. VMware Virtual Platform
- Virtual Machine Serial Number: VMware-42 1b 8a 7d d6 56 cd 16-8c 9a 80 0e 79 fb c8 f9
- VM’s Time Zone: Eastern Daylight Time
- System date and time are consistent with the time zone listed above, as verified by: http://nist.time.gov/.

Appendix C: Tools

- Wireshark v3.0.6
- Tcpdump v4.9.3

Appendix D: Evidence Copies

On 4/12/2020, the location of the week 4 lab files was made known to the examiner. The examiner created a new *Week4b* folder on the *Desktop*, then retrieved archive file *Ch7-NIDS.zip* via Mozilla Firefox from the CYB457 course shell to the *Week4b* folder.

Examiner used the online utility located at onlinemd5.com to obtain the md5 hash value of the downloaded archive file. No checksum for comparison was provided, so the examiner was unable to determine the purity of the preservation copy.

A new folder in *Week4b* was created called *Working*. Examiner created a working copy of the *Ch7-NIDS.zip* archive file in the *Working* folder called *Ch7-NIDSXX*. Onlinemd5.com was used to determine the hash value of the evidence created, which matched the preservation copy, confirming the integrity of the working copy.

Examiner unzipped the working copy archive, revealing file folder *Ch7-NIDS.*

Appendix E: Evidence Verification

Table 2 outlines the hashes obtained throughout the evidence verification process. Onlinemd5.com was used to calculate MD5 hashes.

Designation	Filename	MD5 Hash	Description
PRE-ANALYSIS			
Preservation Copy	Ch7-NIDS.zip	404DD2657EBC7077D44A86EC6877CF41	Archive file downloaded from Engage
Working Copy	Ch7-NIDSXX	404DD2657EBC7077D44A86EC6877CF41	Working Copy created from preservation copy. This copy was analyzed.
POST-ANALYSIS			
Preservation Copy	Ch7-NIDS.zip	404DD2657EBC7077D44A86EC6877CF41	Archive file downloaded from Engage
Working Copy	Ch7-NIDSXX	404DD2657EBC7077D44A86EC6877CF41	Working Copy created from preservation copy. This copy was analyzed.

Table 2: Evidence Verification Table