Gabriel Naples
IST 451
3/26/19

Option 1 – Analysis

Part A - Review the supplied IDS Log. Report some basic statistics about it to include:

- When is the first alert?
  - April 14th at 8:12 (04/14-08:12:12.736634)
- When is the last alert?
  - April 15th at 7:58 (04/15-07:58:02.045009)
- How many total alerts are reported?
  - 13,086 total alerts

Part B – An attack takes place at some time during this log file

Fragmentation Overlap (spp_frag3): could be a DDoS attack

- What time does it start?
  - April 14th at 9:04 (04/14-09:04:04.984694)
- What is the source IP address of the attack?
  - 192.168.2.56
- What is the destination of the attack?
  - 192.168.1.2
- How many alerts are related to this initial attack?
  - There are 18 alerts related to this attack that involve the source IP sending and receiving requests to and from the server.
- When does it end? (do not include the attack change in part C as part of this answer)
  - The last attack packet that has to deal with this attack involving the above source address is on April 14 at 9:04 (04/14-09:04:05.082515)

Bare Byte Unicode Encoding: can mean there is an attack on the web server

- What time does it start?
  - April 14th at 9:01 (04/14-09:01:45.317053)
- What is the source IP address of the attack?
  - 192.168.2.175
- What is the destination of the attack?
  - 192.168.1.10
- How many alerts are related to this initial attack?
  - There are 9 alerts sent that use the same source IP address and target slightly different destinations. (eg: 192.168.1.10, 192.168.1.11)
- When does it end? (do not include the attack change in part C as part of this answer)
  - The last attack packet that was sent that has the source IP address above is at April 14 at 9:09 (04/14-09:09:26.727527)

<u>Part C –</u> At some point in time, the attack changes.

Fragmentation Overlap (spp_frag3)

- When does this occur?
    - April 14th at 10:59 and 6 seconds (04/14-10:59:06.710191)
- Describe the change. What is different about it?
    - The initial attack comes from the source IP of 192.168.2.56. Exactly halfway through the entire attack there is a break and when it starts back up again the source IP address changes to 192.168.2.51.
- Consider the entire attack from beginning to end, including the change. How many different alerts are related to this attack?
    - Including the alerts that come from the different IP address there are 36 total alerts relating to this attack.
- When does this new attack end?
    - The new attack using the different source IP ends on April 14th at 10:59 and 7 seconds. (04/14-10:59:07.092205)

Bare Byte Unicode Encoding

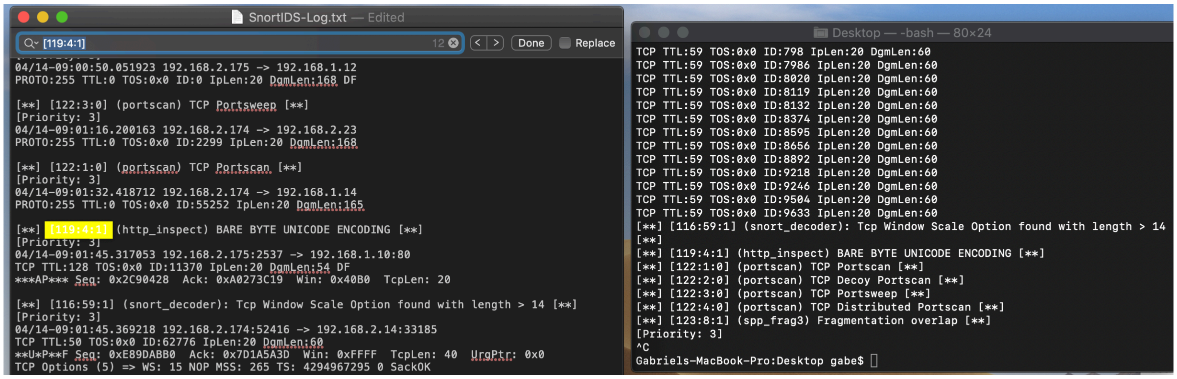
- When does this occur?
    - April 14th at 9:04 and 30 seconds (04/14-09:04:30.459313)
- Describe the change. What is different about it?
    - The initial attack comes from the source IP of 192.168.2.175. After a few alerts the source IP address changes to 192.168.2.174 which means that the attacker was trying to attack from a different device.
- Consider the entire attack from beginning to end, including the change. How many different alerts are related to this attack?
    - Including the alerts that come from the different IP address there are 12 total alerts indicating the attack.
- When does this new attack end?
    - The new attack using the different source IP ends on April 14th at 9:04 and 36 seconds. (04/14-09:04:36.647124) and then the attack switches back to the source IP address that the attack started with.

<u>Part D –</u> Reporting of tools you used

- Report which software tools you used.
    - For this lab I used the ‘grep’ tool in the terminal to get each unique code that appeared within the snort log. I then cross referenced these codes with Snort.org where they show what each code means. This is how I found that code 119:4 was evidence that someone was trying to infiltrate either the IDS system or the web server. This is also what I used to find the ssp_frag3 attack (123-8) that alerted me of the DDoS attack. Lastly, I used the ‘command f’ search tool on TextEdit on the snort log to search for the codes that were important. This allowed me to quickly find the start and end points of the alerts as well as jump to the different alerts.

- Identify the methods you used to find the information

    - First, I looked at the alerts and saw which ones were important. I used grep -c to count the total amount of alerts and also double checked the number of alerts that appeared in the file using the find tool to get another count. To get the time of the first alert I opened the doc in TextEdit and saw that they were all in ordered by time, so I looked at the first one and the very last one in the file. I used the command find tool to find the attack alerts after this and was able to get the information by clicking through the different twelve alerts that relate to the attack. I also noticed upon examination of these alerts that the source IP address changes in both of the attacks. With this information I was able to quickly switch between alerts and find the information about each one to answer the above questions.
- Report and functions, scripts or semi-automated methods you applied in the tools
    - 'grep --only-matching '[**].*' | sort --unique SnortIDS-Log.txt' is the grep call I used that sorted the different unique codes so I knew all of the codes that appeared in the log. Then I typed '[119:4:1]' into the command f search which allowed me to tab through the document directly to the different alerts.



```
04/14-09:00:50.051923 192.168.2.175 -> 192.168.1.12
PROTO:255 TTL:0 TOS:0x0 ID:0 IpLen:20 DgmLen:168 DF

[**] [122:3:0] (portscan) TCP Portsweep [**]
[Priority: 3]
04/14-09:01:16.200163 192.168.2.174 -> 192.168.2.23
PROTO:255 TTL:0 TOS:0x0 ID:2299 IpLen:20 DgmLen:168

[**] [122:1:0] (portscan) TCP Portscan [**]
[Priority: 3]
04/14-09:01:32.418712 192.168.2.174 -> 192.168.1.14
PROTO:255 TTL:0 TOS:0x0 ID:55252 IpLen:20 DgmLen:165

[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
[Priority: 3]
04/14-09:01:45.317053 192.168.2.175:2537 -> 192.168.1.10:80
TCP TTL:128 TOS:0x0 ID:11370 IpLen:20 DgmLen:54 DF
***AP*** Seq: 0x2C90428  Ack: 0xA0273C19  Win: 0x40B0  TcpLen: 20

[**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14 [**]
[Priority: 3]
04/14-09:01:45.369218 192.168.2.174:52416 -> 192.168.2.14:33185
TCP TTL:50 TOS:0x0 ID:62776 IpLen:20 DgmLen:60
**U*P**F Seq: 0xE89DABB0  Ack: 0x7D1A5A3D  Win: 0xFFFF  TcpLen: 40  UrgPtr: 0x0
TCP Options (5) => WS: 15 NOP MSS: 265 TS: 4294967295 0 SackOK
```

```
TCP TTL:59 TOS:0x0 ID:798 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:7986 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8020 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8119 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8132 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8374 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8595 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8656 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:8892 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:9218 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:9246 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:9504 IpLen:20 DgmLen:60
TCP TTL:59 TOS:0x0 ID:9633 IpLen:20 DgmLen:60
[**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14
[**]
[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
[**] [122:1:0] (portscan) TCP Portscan [**]
[**] [122:2:0] (portscan) TCP Decoy Portscan [**]
[**] [122:3:0] (portscan) TCP Portsweep [**]
[**] [122:4:0] (portscan) TCP Distributed Portscan [**]
[**] [123:8:1] (spp_frag3) Fragmentation overlap [**]
[Priority: 3]
^C
Gabriels-MacBook-Pro:Desktop gabe$
```

Part E – Recommendation

- Describe what you would recommend to the network administrator of this organization.

    Bare Byte Unicode

    - The attack that took place takes advantage of using non-ASCII characters as values when encoding UTF-8 values. This is a non-standard behavior and, according to Snort.org, is not recommended by RFC recommendations. It also states that all non-ASCII characters be encoded with a %. I would recommend to the network administrator to set rules on the Microsoft web server to throw away any requests that don't properly encode the non-ASCII characters. Considering that no web clients encode in a non-standard way, none of the non-standard traffic should be accepted. I would also advise the network administrator to check for any available patches to make sure the software is up to date.

    Fragmentation Overlap

    - I would recommend to the network administrator to blacklist the source IP addresses that are sending the DDoS packets so they don't accept traffic from those known malicious actors. I would also tell them to set a rule that when the snort log receives a frag alert to stop incoming traffic from that address to prevent the attack from moving forward until the administrator can examine what’s happening.