Analysing ASA Firewall Logs: A Beginner's Approach

Introduction

Analysing firewall logs is an essential skill for any cybersecurity professional, helping to identify and investigate potential threats and network anomalies. In this writeup, I will be analysing ASA firewall logs generated by Cisco firewalls. While I am still learning the best practices for log analysis, this writeup aims to provide insights for those new to investigating logs (like me) and serve as a learning experience.

If you want to follow along, you can clone the following GitHub repository:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ git clone https://github.com/strandjs/IntroLabs/
```

Within the IntroLabs directory, there is a file named ASA-syslogs.txt, which contains the ASA firewall logs we will be investigating. This lab is largely inspired by the Antisyphon training classes, which offer excellent and accessible cybersecurity training. I highly recommend exploring their labs for further learning.

Why Not Just Use Cat?

Most people might start by dumping the logs using the cat utility:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ head ASA-syslogs.txt
2023-01-26 14:26:10     Local7.Debug    127.0.0.1       Kiwi Syslog Server - Test message number 0001
2023-01-26 14:49:44     Local4.Notice   192.168.1.1     %ASA-5-111008: User 'cisco' executed the 'logging host inside 192.168.1.6' command.
2023-01-26 14:49:44     Local4.Notice   192.168.1.1     %ASA-5-111010: User 'cisco', running 'N/A' from IP 192.168.1.6, executed 'logging host inside 192.168.1.6'
2023-01-26 14:49:44     Local4.Info     192.168.1.1     %ASA-6-302014: Teardown TCP connection 287866 for inside_1:192.168.1.6/64788 to identity:192.168.1.1/443 duration 0:00:00 bytes 784 TCP Reset-O
2023-01-26 14:49:44     Local4.Info     192.168.1.1     %ASA-6-106015: Deny TCP (no connection) from 192.168.1.6/64788 to 192.168.1.1/443 flags FIN ACK  on interface inside_1
2023-01-26 14:49:44     Local4.Debug    192.168.1.1     %ASA-7-710005: TCP request discarded from 192.168.1.6/64788 to inside_1:192.168.1.1/443
2023-01-26 14:49:44     Local4.Info     192.168.1.1     %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7/0 gaddr 192.168.1.1/0 laddr 192.168.1.1/0 type 3 code 3
2023-01-26 14:49:44     Local4.Info     192.168.1.1     %ASA-6-725007: SSL session with client inside_1:192.168.1.6/64788 to 192.168.1.1/443 terminated
2023-01-26 14:49:44     Local4.Notice   192.168.1.1     %ASA-5-321001: Resource 'conns' limit of 20000 reached for system
2023-01-26 14:49:44     Local4.Info     192.168.1.1     %ASA-6-305011: Built dynamic TCP translation from inside_1:192.168.1.6/64789 to outside:24.230.56.6/64789
```

However, given the large volume of logs, this approach is inefficient. Instead, we can leverage command-line utilities like grep, cut, uniq, and sort to filter and analyse the data efficiently.

Filtering Logs More Efficiently

To reduce unnecessary entries, we can exclude logs related to a known local gateway (e.g., 24.230.56.6) using the grep command:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ cat ASA-syslogs.txt | wc -l
71433
```

Vs:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ cat ASA-syslogs.txt | grep -v 24.230.56.6 | wc -l
19356
```

As you can see, this dramatically reduces the number of logs we need to investigate.

Analysing Connection Closures

We can further refine our results by extracting specific fields from the logs. Here, we will use the cut command to help extract specific columns that we want to see:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ cat ASA-syslogs.txt | grep -v 24.230.56.6 | grep FIN | cut -d ' ' -f 1,3,4,5,7,8,9,10,11,12,13,14
```

- -d specifies what the delimiter is, in this case a space is what separates different columns/field values. If we were investigating a csv file via the command line, the delimiter would be a comma.
- -f specifies what fields/columns we want to extract. So, let's say we have a single log entry like as follows:
    - 2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/63708 duration 0:00:01 bytes 8519

  If we just wanted to extract 2023-01-26 and Teardown, all we need to enter is -f 1, 2.

```
2023-01-26 Deny TCP (no from 192.168.1.6/64788 to 192.168.1.1/443 flags FIN ACK
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64798 duration 0:00:01 bytes 7981
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64803 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64804 duration 0:00:01 bytes 8249
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64806 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64807 duration 0:00:01 bytes 8519
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64808 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:35.186.224.25/443 to inside_3:192.168.1.7/54075 duration 0:00:00 bytes 0
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64809 duration 0:00:01 bytes 7964
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64810 duration 0:00:01 bytes 1816
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64811 duration 0:00:01 bytes 8252
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64812 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64813 duration 0:00:01 bytes 8516
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64814 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64815 duration 0:00:01 bytes 8240
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64816 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64817 duration 0:00:01 bytes 7949
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64818 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64819 duration 0:00:01 bytes 8516
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64820 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:13.107.237.38/443 to inside_1:192.168.1.6/64821 duration 0:00:01 bytes 8243
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64822 duration 0:00:01 bytes 1832
```

Identifying Suspicious IPs

The output of the previous command still results in numerous logs, just in a neater format. Why don't we count the number of closed connections per IP address to see if there's anything interesting:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ cat ASA-syslogs.txt | grep -v 24.230.56.6 | grep FIN | cut -d ' ' -f 8 | sort | uniq -c | sort -nr
```

This command:

1. Excludes logs related to the local gateway
2. Filters for finished connections (FIN flag set)
3. Extract the 8th field/column
4. Counts the number of uniq occurrences in the 8th column, and
5. Sorts numerically and in reverse so we can see the highest count first

```
7458 outside:18.160.185.174/443
7439 outside:13.107.237.38/443
  24 outside:34.196.68.227/443
  19 outside:72.21.91.29/80
  17 outside:74.125.70.147/443
  12 outside:184.87.146.116/443
  12 outside:18.160.185.229/443
  12 outside:151.101.1.171/443
  12 outside:108.177.120.95/443
  10 outside:34.120.237.76/443
  10 outside:34.117.237.239/443
  10 outside:184.31.194.139/443
```

Interpreting Findings

As we can see from the output of the previous command, there are thousands of connections from 18.160.185.174 and 13.107.237.38. This merits a deeper investigation, and a simple first step would be to look for logs generated by these IP addresses and inspect them further, for example:

```
┌──(kali㉿kali)-[~/logAnalysis/IntroLabs]
└─$ cat ASA-syslogs.txt | grep -v 24.230.56.6 | grep FIN | grep 18.160.185.174 | cut -d ' ' -f 1,3,4,5,7,8,9,10,11,12,13,14
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64803 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64806 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64808 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64810 duration 0:00:01 bytes 1816
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64812 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64814 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64816 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64818 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64820 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64822 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64824 duration 0:00:01 bytes 1848
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64826 duration 0:00:01 bytes 1816
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64828 duration 0:00:01 bytes 1816
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64830 duration 0:00:01 bytes 1832
2023-01-26 Teardown TCP connection for outside:18.160.185.174/443 to inside_1:192.168.1.6/64832 duration 0:00:01 bytes 1832
```

We can see consistent and short connection durations and the number of bytes is also very consistent. Based on the ports used, we can likely conclude that the internal host is making a series of requests to a web server at 18.160.185.174 over HTTPS, possibly indicating data exfiltration or C2 activity.

Conclusion

This writeup provided an introduction to analysing ASA firewall logs using command-line tools. By filtering unnecessary data, extracting relevant fields, and identifying high-frequency connections, we can begin to identify potential incidents. To learn more about log analysis, I recommend exploring Antisyphon's training resources.

Would love to hear your thoughts and feedback! Feel free to reach out.