CSCI400 Security Lab
Topic: Denial of Service

Prof. Dietrich

October 18, 2017

Group size: 4 — Setup needed: Windows/MacOS X, Unix hosts, DETER.

1 General description

In this lab, we are performing denial of service (DoS) on hosts. This exercise has been created for DETER, and is based on Jelena Mirkovic's USC/ISI DoS DETER setups.

2 Setting up for the attack

In this part of the lab, we will be setting up the experiment.

You should load the topology file `/share/education/TCPSYNFlood_USC_ISI/synflood.ns` into the DETER testbed to create a new experiment. Do not modify the topology file but read it through and identify what each directive does. Once an experiment is swapped in, install Apache on the server node by typing on server node:

```
% /share/education/TCPSYNFlood_USC_ISI/install-server
```

Similarly, install an attack tool called "flooder" on the attacker node by typing on attacker node:

```
% /share/education/TCPSYNFlood_USC_ISI/install-flooder
```

2.1 Requirement

- You use the instructions provided with this lab. You should **ONLY** use DETER, as this experiment can do damage.
- Your output should demonstrate that you have done the experiments, such as screenshots, Unix typescripts, and the traffic logs showing that packets were sent to the server.

3 Tasks

3.1 Generating legitimate traffic

Create a Web traffic stream between the client and the server nodes by writing a script at the client that each second gets index.html from the server. You can for example write this script using bash and curl.

3.2 Turning off SYN cookies

SYN cookies are often on by default in Linux and FreeBSD. To check if they are on do the following on Ubuntu Linux:

```
ssh yourusername@users.deterlab.net
ssh server.YourExperiment.YourProject
sudo sysctl net.ipv4.tcp_syncookies
```

If you see 1 as the result, SYN cookies must be set to zero for the demo to work. On FreeBSD, the command would be slightly different: `sudo sysctl net.ipv4.tcp.syncookies`. Type the following on the server machine:

```
sudo sysctl -w net.ipv4.tcp_syncookies=0
```

Verify that SYN cookies are now off by typing on the server machine:

```
sudo sysctl net.ipv4.tcp_syncookies
```

3.3 Generating attack traffic

Create a SYN flood between the attacker and the server nodes, using the Flooder tool. You can type "flooder" on the attacker node's command line to get a man page for the tool. Examples at this page show how to write a command to send a flood of SYN packets. Make sure to spoof within `1.1.2.0` range (use `mask 255.255.255.0`).

3.4 Collecting traffic statistics

You will now collect `tcpdump` statistics on client machine with and without syncookies, calculate connection duration and draw graphs of connection duration on y-axis and connection start time on x-axis. Perform the following steps:

1. Stop all traffic by stopping your legitimate client's script and flooder.

2. Start tcpdump on the client

    ```
    ssh yourusername@users.deterlab.net
    ssh client.YourExperiment.YourProject
    ip route get 5.6.7.8
    ```

You should see something like this as a result:

```
5.6.7.8 via 1.1.2.2 dev eth2  src 1.1.2.3
   cache  mtu 1500 advmss 1460 metric 10 64
```

Thus the interface name leading to 5.6.7.8 is eth2. To see the traffic flowing type:

```
sudo tcpdump -nn -i eth2
```

then generate some traffic, restart your client. You will need to discover proper `tcpdump` options to see only IP traffic and to save recorded traffic into a file. Start `tcpdump` with these options.

3. Using a stopwatch perform the following scenario:
   (a) Start legitimate traffic
   (b) After 30 seconds start the attack
   (c) After 120 seconds stop the attack
   (d) After 30 seconds stop the legitimate traffic
   (e) Stop the tcpdump on the client and save the file
4. Turn the SYN cookies on and repeat the above steps.
5. Using the recorded traffic files and tcpdump to read them, process the output and calculate connection duration for each TCP connection seen in the files. Connection duration is the difference between the time of the first SYN and of the ACK following a FIN-ACK (or between the first SYN and the first RESET) on a connection. Remind yourself what uniquely identifies a TCP connection, i.e. how to detect packets that belong to the same connection? If a connection did not end with an ACK following a FIN-ACK assign to it the duration of 200 s. Include two graphs in your submission, showing connection duration vs connection start time for the case without and with SYN cookies. Label the graphs so they can be distinguished and indicate on each graph using vertical lines or arrows the start and the end of the attack.

3.5 No spoofing

Remove spoofing from the attack. Repeat the exercise without SYN cookies and observe and explain the effect. What happens? Can you explain why this happens? For hints run a `tcpdump` on the server node and look for traffic patterns. Can you modify the attack so that it is effective without spoofing and how would you do this?

Bonus:

- Modify the NS file to introduce point-to-point routes, using the Modify Experiment option. Hint, you need to remove the server's route to `lan1` and to add routes from the server to the attacker, and from the server to the client. Then click on Submit. It will take several minutes for the experiment to be restarted and you will receive an email notification once this is done. Now repeat the exercise without SYN cookies and observe and explain the effect. What happens? Can you explain why this happens? For hints run a `tcpdump` on the server node and look for traffic patterns.

- Implement all the floods in the Shaft attack tool in perl/python, and repeat attack scenarios.
- Implement a XMAS packet flooder, and repeat attack scenarios.

4 Word Problems

1. Explain how the TCP SYN flood attack works.
2. Explain how SYN cookies work to prevent denial-of-service effect from SYN flood attacks.
3. Would changing the network buffers in the OS (e.g. as available FreeBSD) have any impact on attack effectiveness?
4. How would you defend against these attacks other than with SYN cookies?

5 Deliverables

1. A report describing all your findings above.
2. A zip file containing:
   - Any perl/python tools and text versions of traffic logs.
   - Answers to all word problems in Part 4.
3. Submit by class on October 30, 2017.

6 Grading

Points will be subtracted if any of the pieces of the deliverables are missing or incomplete.