TCP SYN flood - USC/ISI

Created by: Jelena Mirkovic, USC/ISI, sunshine@isi.edu.

Contents

Overview

This exercise demonstrates a well-known denial-of-service attack, called **TCP SYN flood**. Students will be able to create a real attack using DETER tools, and to observe its effect on legitimate traffic. Afterwards, they will be asked to apply a known defense against SYN flood known as **SYN cookies**, repeat the attack and observe the protection.

This exercise helps students learn the following concepts:

1. How TCP/IP works and how its design can be misused for attacks
2. How easy it is to perpetrate a DoS attack, with fully legitimate traffic and at a low rate
3. How easy it is to protect machines from this type of attacks via built-in OS mechanisms.

Additionally, extra credit questions improve a student's understanding of how networks and TCP/IP work.

Required Reading

All students should have completed an introductory networking course with grade B or better.

- Short summary of SYN flood attack on Wikipedia
- SYN flood attacks in the Internet Denial of Service book (optional reading)
- SYN cookie overview
- Tcpdump's man page

Introduction

Denial of service attacks deny service to legitimate clients by tying up resources at the server with a flood of legiitmate-looking service requests or junk traffic. Before proceeding to the assignment instructions make sure that you understand how TCP SYN flood attack works, which resource it ties up and how, and how syncookies help mitigate this attack.

Assignment Instructions

Setup

Each student should load the supplied topology file synf.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.

Especially important are the lines in the NS file:

```
   #Add SEER support to each node
   tb-set-node-startcmd $node "sudo python
/share/seer/v160/experiment-setup.py Basic"
```

that add support for traffic generation and visualization via SEER. Once an experiment is swapped in use SEER GUI by clicking at "Launch SEER now" option from here. The code should work on any computer supporting a reasonably recent version of Java. Once the code starts, choose the "Emulab" interface in the first dialog.

Then, the experiment must be "attached" to the GUI. This is done by choosing *Emulab Interface->Attach to Experiment* from the GUI and then input Project and Experiment Name. **Note: capitalization matters here**. Next, you will need to input your username on DETER (for example smith**@users.deterlab.net**) and the same password you use to log on to DETER. You may want to input your public key into the dialog (SSH Key File) if you have already set up passwordless access to DETER. Otherwise delete the information in the SSH Key File input field and you will be asked for your DETER password.

Once attached, experiment topology is visible in the Topology tab. Right clicking on nodes and selecting "Open Traffic Graph" will add a graph of that node's traffic to the Graph tab. When you first swap in the experiment, there isn't much (or any) traffic flowing between the nodes. However, SEER includes traffic generation tools which you will use to create realistic traffic. Once traffic flows are established, legitimate traffic will show as green and attack traffic will show as red on the Graphs tab. Notice that you can change the counters to show packets per second (pps) or bytes per second (bps) at the top selection option on top of a graph. If the *Src* option in the Graph tab is set to "forward" the attack traffic will show as black. Change it to the IP address option and you should be able to see incoming and outgoing traffic for any node, both legitimate and attack. The rest of this assignment assumes that you have opened the graph that shows traffic reaching and leaving the **server** node.

Tasks

Generating legitimate traffic

Create a Web traffic stream between the **client** and the **server** nodes by following these

steps from Controls tab in SEER:

1. Click on Traffic item on the left sidebar
2. Click on Web
3. Click on New Web group
4. Choose some name for the group
5. Choose **client** node as a client
6. Choose **server** node as a server.
7. Under "Think Time," Choose exponential distribution of arrival times with lambda 5, scale 1 and max 5.
8. Under "File Sizes," Choose minmax distribution of file sizes with min 1,000 and max 10,000.
9. Click Start. You should be able to see some green traffic now on your graph. Use **Print Screen** button on your keyboard to capture this graph, paste it into a Word document, and include it in your project submission.

Turning off SYN cookies

SYN cookies are often on by default in Linux and FreeBSD. If they are on, the attack will not work. To check if they are on do the following:

```
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. 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
```

Generating attack traffic

Create a SYN flood between the **attacker** and the **server** nodes.

1. Click on the Attack item on the left sidebar
2. Click on Packet Flooder
3. Click on New Packet Flooder
4. Choose some name for the group
5. Choose **attacker** node as a participating node.
6. Source field should be 1.1.2.0 and mask (the portion after the "/") should be set to 255.255.255.0 to ensure subnet spoofing.
7. The **server** node is the target node.
8. Leave the Target Mask blank.
9. The protocol type for traffic should be TCP.

10. Leave the length range unspecified.
11. Choose flat attack rate, with high rate of 1,000 packets per second (this is the unit of measure for rate fields in SEER).
12. Set the destination ports to min 80 and max 80.
13. Set the TCP flags field to SYN. Do not specify any other options - you can either delete default values or leave them as they are.
14. Click Start. You should be able to see some red traffic now on your graph. Make sure you are looking at a graph of pps not Bps (selection option on top of the graph). Use **Print Screen** button on your keyboard to capture this graph, paste it into a Word document, and include it in your project submission.

Collecting statistics and creating graphs

You will now collect `tcpdump` statistics on the **client** machine with and without syncookies. Once you have done that, you will calculate connection duration for each TCP connection. Then, using a tool such as Excel, you will graph the sessions by plotting the connection duration on the y-axis and the connection start time on the x-axis.

Perform the following steps:

1. Stop all traffic using **Stop** button for each traffic group in SEER
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, e.g. by starting SEER's Web traffic again. 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.

A TCP session's connection duration is the difference between the time of the first SYN and the final ACK following a FIN-ACK (or between the first SYN and the first RESET). Remind yourself what uniquely identifies a TCP connection; how do you determine which packets belong to the same connection when they are interspersed? If a connection did not end with an ACK following a FIN-ACK assume it had a duration of 200s. Submit two such graphs; one showing the above timeline with and without SYN cookies enabled on the server. 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.

What can go wrong

- **Experiment cannot be swapped in**. First, check the error message you will receive in the email. One possible reason for this is that the NS file was changed from the one listed above. Verify that the file looks exactly like supplied with this exercise. Another reason may be that there is a lack of available nodes in DETER and the error message will say so. This happens ocassionally and usually resources become available in a few hours. If you tried several times and could not find enough resources or could not diagnose why the experiment was not swapping in, forward the error message you get from DETER to your TA.
- **SEER GUI does not run**. The GUI usually requires the latest version of Java so if it does not run this means you need to upgrade your Java version.
- **SEER GUI cannot attach the experiment**. Very likely reason for this is that you are not typing the correct username and password. Try to SSH into **users.deterlab.net** with the same username and password you are using to attach the experiment. If this works but you still have problems with attaching email your TA.
- **There is no legitimate (attack) traffic on graphs**. First verify that the options are set exactly as shown in the pictures that go along with steps 2.4 and 4.4. If this is all fine, SSH to one of your experimental machines and try to ping another two (e.g., ping **server** and **attacker** from **client**). Use short names (e.g., **ping server** and NOT **ping server.YourExperiment.YourProject**) in ping commands. If you notice any connectivity problems click on *Modify Experiment* on your DETER Web page and then click on *Submit*. The experiment will reset to its original state - this may take several minutes and you will receive an email when it's done. If connectivity problems persist email your TA. Finally, if everything else seems fine it may be that SEER backend has died for some reason (perhaps you rebooted one of the machines without going through Modify Experiment?) To fix this click on *Modify Experiment* on your DETER Web page and then click on *Submit*. The experiment will reset to its original state - this may take several minutes and you will receive an email when it's done. If none of this helped email your TA.

  Another reason for missing traffic is if your experiment is not swapped in.
- **There is some traffic but it does not look like shown on the graphs**. If the shape of the traffic is different this is almost certainly caused by not following the project instructions closely. Verify that SYN cookies are off, that the NS file is exactly like given on this page, and that SEER's settings are exactly as shown in steps 2.4 and 4.4. If this is all fine and you are still having problems email your TA.

  If the shape looks similar but there is no red traffic on graphs this is a testbed switch problem - QoS flags get reset. Email your TA who will then contact DETER ops team.

Extra Credit

There are two extra-credit questions:

1. 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?
2. 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.

Submission Instructions

You should submit a Word document with the following items (label each section):

1. Explanation how the TCP SYN flood attack works.
2. Explanation how SYN cookies work to prevent denial-of-service effect from SYN flood attack
3. Picture of your topology in DETER. Copy this directly from the DETER Web page using high level of detail (clicking on the small picture will open a larger one with detail selection option) so IP addresses are visible.
4. Screen shot showing the legitimate traffic flowing to and from the **server** node in packets per second.
5. Screen shot showing the attack traffic flowing to and from the **server** node in packets per second. Explain why the amount of legitimate traffic increases during the attack (hint, look at `tcpdump` at the **server** node).
6. Graphs of distribution of **client's** connection duration for no-SYN-cookies and SYN-cookies cases. Each graph must be labeled and have vertical lines or arrows show the start and the stop of the attack. Explain what happens in each case. Is the attack effective? How can you tell this from the graphs?
7. Answers to extra credit questions if any.