

# Project 3

## Due November 4<sup>th</sup>, 2003 at 4pm

### ***Project description***

Create an Emulab experiment containing a topology with 4 machines, perform a DDoS attack on one machine, design Snort rules to detect this attack, then suggest how to design defenses.

1. Carefully read Emulab's *Getting Started* tutorial to learn how to create experiments, how to setup topologies and routing, etc.
2. Create a topology file in your home directory (/users/yourusername) with a name YourName.top that describes a network of 4 nodes. One node will be the victim of the attack, one node will be a zombie machine, one node will be a gateway to the victim network and one node will be a legitimate client. The victim is connected to the gateway, and the zombie and the client machine are also connected to the gateway (so you end up having a star topology). Make sure you assign different addresses to different interfaces. For hints, look at /proj/CIS662/project2-topology file to see how to assign addresses to interfaces. Make the link between the victim and the gateway 10Mbps and all other links 100Mbps.
3. Create an experiment with a name YourNameExp. This experiment will be exclusively under your control so remember to always swap it out when you are not working on it, and swap it in again when you want to work on it. Also set idle swap to 0.5 hours, i.e. experiment will be swapped out after 30 minutes of idle time.
4. Decide what legitimate traffic you would like to have between a legitimate client and a victim (who will act as a server). For instance you could generate FTP transfers (using scp command), web requests, telnet-like communication, etc. You will likely need to script commands on the client to be executed as you will need multiple trials for measurements and it is tiresome to type commands always by hand.
  - a. As a first step you need to make sure you have all the software needed for the type of traffic you have chosen
    - i. For FTP transfers you need to take a few files from any place you like and place them somewhere in your home directory, say into /users/yourusername/legitimate/legitimate\_files
    - ii. For telnet-like communication you need to generate a little program that goes in the loop for the certain amount of time (or certain number of iterations) and at each moment prints out some message (possibly of variable length) on the screen. Let's call this program my\_telnet. Place it somewhere in your home directory, say into /users/yourusername/legitimate/my\_telnet

- iii. For web requests you need to install a Web server on the victim machine. You can do this by SSH-ing into the victim machine, then typing
 

```
cd /proj/CIS662/project3
./setup_apache
```

 Once you have installed it, place some HTML files that you will request somewhere in your home directory, say in
 

```
/users/yourusername/legitimate/HTML
```
  - iv. Of course, you can also do a mix of these traffic types but this is a little bit challenging. So to play it safe do the simple traffic first, then if you have plenty of time you may “upgrade” your legitimate traffic generator to do traffic mixes.
- b. Whenever you swap the experiment in, you will need to copy relevant files to the victim and client machines. I suggest you to script these steps. Simply figure out which commands you need to type (i.e. type them couple of times to make sure they do what you want them to do) then put them in a files called `/users/yourusername/setup_victim` and `/users/yourusername/setup_client`, do
- ```
chmod a+rx users/yourusername/setup_victim
chmod a+rx users/yourusername/setup_client
```
- and you are ready to go. Now what should be in these files:
- i. If you have decided to generate FTP traffic, file `setup_victim` must contain commands to copy files from `/users/yourusername/legitimate/legitimate_files` to someplace on the victim machine. For instance
 

```
cp /users/yourusername/legitimate/legitimate_files /tmp
```
  - ii. If you have decided to generate telnet-like traffic, file `setup_victim` must contain commands to copy `my_telnet` from `/users/yourusername/legitimate/my_telnet` to someplace on the victim machine. For instance
 

```
cp /users/yourusername/legitimate/my_telnet /tmp
```
  - iii. If you have decided to generate Web traffic, file `setup_victim` must contain commands to setup Apache Web server, to copy HTML files onto the victim machine, and to start Apache. Namely:
 

```
cd /proj/CIS662/project3
./setup_apache
cp /users/yourusername/legitimate/HTML/* /www/htdocs
sudo /www/bin/apachectl start
```
  - iv. Add to each `setup_victim` file
 

```
sudo chmod 777 /tmp
```
  - v. `setup_client` will copy file `run_client` that we will soon create onto the client machine. For instance:
 

```
cp /users/yourusername/legitimate/run_client /tmp
```
- c. After running `setup_victim` try whether things work:
- i. If you have decided to generate FTP traffic SSH into the client machine and type
 

```
scp victimIP: /tmp/one_of_the_files_you_put_there /tmp
```

- ii. If you have decided to generate telnet-like traffic SSH into the client machine and type  
ssh -C victimIP "/tmp/my\_telnet duration\_or\_number\_of\_iterations"
    - iii. If you have decided to generate Web traffic SSH into the client machine and type  
wget http://victimIP/one\_of\_the\_HTML\_files\_you\_put\_there
  - d. Now try the same thing but put /usr/bin/time -f"%e" in the front. This will give you in the end how long it took to execute the command, in seconds
  - e. You have now all you need to script the legitimate traffic. Assemble several commands and put them into  
/users/yourusername/legitimate/run\_client
5. Decide what type of DDoS traffic you want to generate. You will be writing a tool to generate this traffic so keep it simple. You could for instance decide to generate TCP SYN flood, UDP flood, ICMP flood, etc.
  6. Write a DDoS tool that can send a customizable number of packets per second, for a given number of seconds **to a specific target address (victimIP) that is hardcoded in the program**. This is a foolproof feature so you don't accidentally let anything loose in the network. You can make all packets the same length. Look for guidelines how to write a program that sends packets out in the network at  
<http://www.ecst.csuchico.edu/~beej/guide/net/html/>  
Keep it simple. You don't have to spoof addresses or anything else, simply send packets out. Call this tool my\_doostool, place it into /users/yourusername/attack
  7. Run legitimate client traffic with and without the attack. Try out several attack strengths (number of packets per second) and note how the attack affects legitimate traffic. Make the legitimate traffic duration longer than the attack duration so that you can see the cycle: everything is fine for the legitimate client, delays due to the attack, recovery after the attack has stopped. For instance 5 minutes of legitimate traffic with 2-3 minutes of attack traffic in the middle should do the trick.
  8. Install snort on the gateway machine by SSH-ing to it and typing  
/proj/CIS662/project3/setup\_snort  
Then go to  
/etc/snort  
and put in the network range of your victim network into snort.conf file replacing the line var HOME\_NET any with var HOME\_NET your\_victim\_net\_range
  9. Run snort by typing on the gateway machine  
cd /etc/snort  
snort -A fast -b -c snort.conf  
Try to ping from client to the victim machine, then CTRL-C snort and see whether the summary printed on the screen indicates your pings in the ICMP part of the summary. If it does you did everything right.
  10. Try to come up with snort rules (located in /etc/snort/rules) that would detect the attack you are performing. You can make any assumption you want about the values/contents/types of attack traffic and code these assumptions into my\_ddostool, so that you can use them in snort rules. Make sure rules do not adversely affect legitimate client traffic (e.g., you can exclude it from observation). For more info about snort look at <http://www.snort.org/docs/FAQ.txt>

11. Provide a write-up containing seven parts:
- a. **Topology file YourName.top**
  - b. **Description of how you generated legitimate traffic, along with all the relevant scripts and source files** (e.g. my\_telnet source code, but not data files that you transferred in FTP or Web transactions)
  - c. **Source code of my\_doostool.** I won't run this program, I'll just look at the code you submit. Also provide a brief description which type of DDoS traffic it generates.
  - d. **A graph showing how long it took the client to complete all the commands in run\_client as the attack traffic is varied in strength.** For this you will need to run attack multiple times with different strengths. On x-axis show at least 10 attack strengths (including 0 for no attack), on y-axis show total time needed to execute all commands in run\_client (you can get this by typing `time -f"%e" run_client` on the client machine). If you don't notice any change, make the attack longer or make the commands in run\_client shorter. Provide a few sentences about what you have learned from this experience. How does DDoS attack affect legitimate client?
  - e. **A graph showing how long it took the client to execute each command in run\_client for 3 attack strengths, as compared to the base times (no attack).** On x-axis number each command (e.g., 1, 2, 3) and on y-axis show the relative times needed for each command in 3 cases (e.g., 134% for attack strength 1, 150% for attack strength 2, 300% for attack strength 3 for the first command ...)
  - f. **A list of snort rules you added and a summary how well they detected attacks you generated, for different attack strengths.** Provide a few sentences about what you have learned from this experience. Was it easy to come up with these rules?
  - g. **Imagine a scenario when DDoS attack traffic is very similar to legitimate traffic in your network. Provide a page of suggestions how would you defend against this.** Address both detection and response. You can reference existing DDoS defenses, combine them, or make up your own. How easy would it be to modify your solution so that new attacks are detected and contained? Give a simple example of a new attack you could detect with the modified solution.

### ***Project Submission Instructions***

Place the PDF or PS file containing your writeup in a folder named "YourName\_project3" (naturally, replace YourName with your name, e.g. I would create a folder named JelenaMirkovic\_project3). Tar and zip this folder and send it as attachment by E-mail to **sunshine@cis.udel.edu**.