CS 6500 – Network Security
Prof. Krishna Sivalingam
July-Nov. 2017
Lab 5: Snort-based Firewall for Intrusion Detection
TARGET Due Date: Sunday, **Nov. 12, 11PM**
Extended No-Penalty Due Date: Sunday, **Nov. 19, 11PM** (FIRM)
No clarifications after Nov. 10, 5PM
On-Line Submission via moodle

1 Project description

The objective of this project is to gain experience with setting up different network attacks and setting up an Intrusion Detection System (IDS) to detect the attack.

The project will be implemented using the **mininet** framework `mininet.org`.

1.1 Mininet

1. Mininet enables you to create an emulated network with switches, routers, nodes, etc. on a single Linux system. Each network node runs a real kernel from the host system.
2. It is easier to run mininet inside a Virtual Machine framework, such as Oracle VirtualBox, `https://www.virtualbox.org/`.
3. A Sample Workflow is described at: `http://mininet.org/sample-workflow/`
4. Create a topology consisting of at least 4 client nodes, one gateway node (through which the clients and servers communicate) and at least one server node. The link speeds are variable – you can choose from {10, 100, 1000} Mbps.

1.2 Legitimate traffic setup

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, ssh-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.

To obtain execution time, you can prefix '/usr/bin/time', with the appropriate "-f" or "–format" options, before the command. For example, try:

```
/usr/bin/time -f"%E seconds" scp victimIP:/tmp/some_file_there /tmp
```

1.3 Attack traffic

Decide what type of attack 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, Redirection attack, etc. At least three different attack types must be generated, such as DDoS attack, DoS attack (from one client), Spoofing attacks, etc.

Run legitimate client traffic with and without the attack traffic. Try out several attack strengths (varying 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, and then recovery after the attack has stopped. For instance, try five to eight minutes of legitimate traffic with 2 to 3 minutes of attack traffic during the legitimate traffic.

1.4 Detection using Snort

Install (if needed) and run snort (`https://www.snort.org/`) on the gateway machine.

Run ping command from the client to the victim machine, then CRTL-C snort and see whether the summary printed on the screen indicates your pings in the ICMP part of the summary.

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, so that you can use them in snort rules. Make sure that the 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`. The current snort rules can be found at `https://www.snort.org/downloads/#rule-downloads`.

1.5 Report

Prepare a report with the following components:

1. Topology file description (as a diagram).
2. Description of how you generated legitimate traffic, along with all the relevant scripts and source files
3. Description of how you generated attack traffic, along with all the relevant scripts and source files
4. 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?
5. A graph showing how long it took the client to complete all the commands as the attack traffic is varied in strength. For this you will need to run attack multiple times with different strengths. On the x-axis, show at least 5 attack strengths (including 0 for no attack); on the y-axis, show the total time needed to execute the client commands.

2 What to Submit

- ▷ Name your project directory as LAB5
- ▷ Once you are ready to submit , change directory to the directory above LAB5, create a tar-gz file and upload ROLLNO-LAB5.tgz file.

The directory should contain the following files:

- ⋆ Source Files and the Report
- ⋆ a README file containing instructions to compile, run and test your program. The README should document known error cases and weaknesses with the program.
- ⋆ a COMMENTS file which describes your experience with the project, suggestions for change, and anything else you may wish to say regarding this project. This is your opportunity for feedback, and will be very helpful.

▷ During the demo, you have to show the relevant live wireshark traces relating to the attack(s).

3 Help

1. WARNING ABOUT ACADEMIC DISHONESTY: Do not share or discuss your work with anyone else. The work YOU submit SHOULD be the result of YOUR efforts. Any violation of this policy will result in an automatic ZERO on the project, a potential F in the course, and other academic action.
2. Ask questions EARLY. Do not wait until the week before. This project is quite time-consuming.
3. Implement the solutions, step by step. Trying to write the entire program in one shot, and compiling the program will lead to frustration, more than anything else.
4. List of Useful URLS:

   ▷ `www.insecure.org/tools.html`: This lists several other network security tools, if you are interested in further exploration.

4 Grading

- ▷ Legitimate Traffic: 10
- ▷ Attack Traffic: 45 (15, per attack type)
- ▷ Snort: 25
- ▷ Report: 10
- ▷ Demo: 10

No README/COMMENTS: -5 points; Incomplete Compilation: -10 points