Breaking Established TCP Connections with TCP RST

CS3235 Lab 3

Abstract

This laboratory may be done in a group. The size of your group is up to you, and could comprise of 1, 2, 3 or 4 members. When you feel comfortable with the lab, ask Pradeep to assess you. He will ask you questions relating to the lab and will probably also ask you to demo breaking an active TCP connection. He'll see how efficiently you can do that and whether you understand why the attack works. There is *no* written material to be returned with the lab.

In this lab you will get a feel for how easy it is to mount active attacks in shared media networks[1]. There are powerful libraries available for the creation and injection of arbitrary packets into the network, and these can be used to significantly simplify the coding effort behind such attacks. One such library is the libnet packet creation and injection written by Mike Schiffman. While you won't be required to code using the library in this lab, you will profit considerably from understanding how to use the API (in your spare time), perhaps by writing a general purpose traffic generator with it. For this lab, you will use the program provided to you and run it to break an active TCP connection.

When a TCP connection is in the ESTABLISHED state, a RST packet received by either endpoint's TCP within the connection's receive window signals that the "other end" wants to immediately break the connection. See W. Richard Steven's TCP/IP Illustrated - I for details. If an active attacker knows or guesses details about a TCP connection then he can break it by generating a spurious RST packet and sending it to either the source or the destination.

The program that generates a TCP RST packet and sends it to the destination is provided to you. Its source code is also given in the last section and you may study it in conjunction with the libnet API to understand how it works[2]. To run the program, use the command line

```
tcp_rst -s <src_name>:<src_pt> -d <dst_name>:<dst_pt> -n <seq_num> -a <ack_num>
```

This will generate a TCP RST packet destined for the host dst_name (which may be a domain name such as ramula.ddns.comp.nus.edu.sg) and port dst_pt, with the source address of the packet being src_name (this can be a DNS name as well) with its port being src_pt. The -n argument takes a *decimal* value for the sequence number to place in the packet while the -a argument takes a *decimal* value for the acknowledgement number. On some operating systems, any value for the acknowledgement number will do, while in others, a *proper* value might be needed. The program is statically linked against libnet.a so you don't have to install the libnet library yourself. Try to understand the attack and how and why it works.

[1] Even in switched environments, a smart adversary can cause much of the same damage.

[2] This is not required for the current lab.

What you will do

Create a TCP connection between two machines, say A (a Linux box) and B (a WIndows box). A TCP connection can be made by either telnetting from one to the other or by logging in using ssh. Run ethereal on A and determine the sequence number of the next expected datagram from $A \rightarrow B$ when the connection is quiescent. Run tcp_rst as

```
tcp_rst -s A:A_pt -d B:B_pt -n S -a <ack>
```

S is the sequence number you determined above. Does it break the TCP connection? Hurray. Explain what's going on. Do you need a special value for <ack> or does any value work? If you need a special value, what is it conceptually?

Review questions

You don't have to submit answers to these in writing but you must be prepared to answer them and questions of similar ilk.

1. What is the kind of attack you are performing in this lab referred to in computer security literature?
2. TCP understands six code bits: URG, ACK, PSH, RST, SYN, FIN. Explain the purpose of each.
3. Why do we need the src_ip and src_pt numbers when constructing the attack RST packet? Why isn't it enough to just send a RST packet to the correct destination and port number?
4. Why do we need to estimate the *sequence number* of data traffic in one direction (any direction is fine) before we can apply the attack? Why doesn't an arbitrarily constructed RST packet work?

Code for the brave

The code for tcp_rst is available off the course web page. Those of you who want to use libnet for doing other things might find it useful. The libnet distribution also has lots of examples that are very instructive.

Assessment

When you are ready to be assessed, fill in this sheet with your details, give it to Pradeep and ask him to assess you.

Date: .... □

Group members: ....

Name	Matriculation ID

Leave this section for Pradeep to fill in:

- Q2 of assessment .... □ (1 mark)
- Q3 of assessment .... □ (1 mark)
- Q4 of assessment .... □ (1 mark)
- Q5 of assessment .... □ (5 marks)
- Q6 of assessment .... □ (5 marks)
- Q7 of assessment .... □ (2 marks)
- Bonus question .... □ (2 marks)

Total (out of 15) .... □