**York College of Pennsylvania**

CS 497

Project 2: TCP/IP Attack Lab

**Introduction**

The learning objective of this lab is for students to gain first-hand experience on vulnerabilities, as well as on attacks against these vulnerabilities. Wise people learn from mistakes. In security education, we study mistakes that lead to software vulnerabilities. Studying mistakes from the past not only help students understand why systems are vulnerable, why a seemly-benign mistake can turn into a disaster, and why many security mechanisms are needed. More importantly, it also helps students learn the common patterns of vulnerabilities, so they can avoid making similar mistakes in the future. Moreover, using vulnerabilities as case studies, students can learn the principles of secure design, secure programming, and security testing.

The vulnerabilities in the TCP/IP protocols represent a special genre of vulnerabilities in protocol designs and implementations; they provide an invaluable lesson as to why security should be designed in from the beginning, rather than being added as an afterthought. Moreover, studying these vulnerabilities help students understand the challenges of network security and why many network security measures are needed. In this lab, students need to conduct several attacks on the TCP protocol, including the SYN flood attack, the TCP reset attack, and the TCP session hijacking attack. This lab may be done as a team of 2-3 students. Recommend review of the documents under Helpful Documents at <http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Networking/TCPIP/>. These include:

* Netwox/Netwag Guides
* Slipping in the Window: TCP Reset attacks
* Strange Attractors and TCP/IP Sequence Number Analysis
* ICMP attacks against TCP

**Lab Setup**

**Network Setup.** To conduct this lab, students need to have at least 3 machines. One computer is used for attacking, the second computer is used as the victim, and the third computer is used as the observer. Students can set up 3 virtual machines on the same host computer, or they can set up 2 virtual machines, and then use the host computer as the third computer. For this lab, we put all these three machines on the same LAN, the configuration is described in Figure 1.

Ensure that Oracle VirtualBox is installed on your computer. This will house the VM used for this lab. Once done, configure the network in VirtualBox for SEED labs following the instruction at <http://www.cis.syr.edu/~wedu/seed/Documentation/VirtualBox/VirtualBox_NATNetwork.pdf>. Use the following to help setup and configure the VMs for the lab:

* User Manual of the pre-built Ubuntu 12.04 Virtual machine
* Network Configuration in VirtualBox for SEED Labs (NAT)
* How to use VirtualBox to Run Our Pre-build VM Image?
* Customization for Labs with Multiple VMs

These can be found at the above URL.

**Operating System.** This lab will be carried out using pre-built virtual machine based on Ubuntu Linux, and all the tools needed for this lab are already installed. Download and install the **SEEDUbuntu12.04 VM** from one of the servers at Syracuse University at <http://www.cis.syr.edu/~wedu/seed/lab_env.html>. A user manual can be found at this same URL.

**Netwox Tools.** We need tools to send out network packets of different types and with different contents. We will be using Netwox to accomplish this. The GUI version is called Netwag.

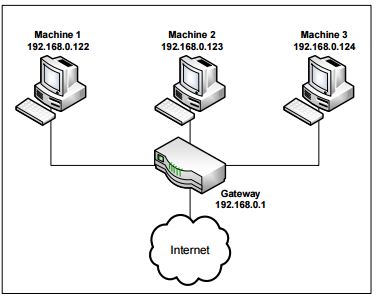


Figure 1: Environment Setup

Netwox consists of a suite of tools, each having a specific number. You can run a command like following (the parameters depend on which tool you are using). For some of the tool, you have to run it with the root privilege:

# netwox number [parameters ... ]

If you are not sure how to set the parameters, you can look at the manual by issuing "netwox number--help". You can also learn the parameter settings by running Netwag: for each command you execute from the graphic interface, Netwag actually invokes a corresponding Netwox command, and it displays the parameter settings. Therefore, you can simply copy and paste the displayed command.

**Wireshark Tool.** You also need a good network-traffic sniffer tool for this lab. Although Netwox comes with a sniffer, you will find that another tool called Wireshark is a much better sniffer tool. Both Netwox and Wireshark are included in the pre-built VM image. Since we will be using the pre-built VM image, use both and compare results. To sniff all the network traffic, both tools need to be run by the root.

**Enabling the ftp and telnet Servers.** For this lab, you may need to enable the ftp and telnet servers. For the sake of security, these services are usually disabled by default. To enable them in our pre-built Ubuntu virtual machine, you need to run the following commands as the root user:

Start the ftp server

# service vsftpd start

Start the telnet server

# service openbsd-inetd start

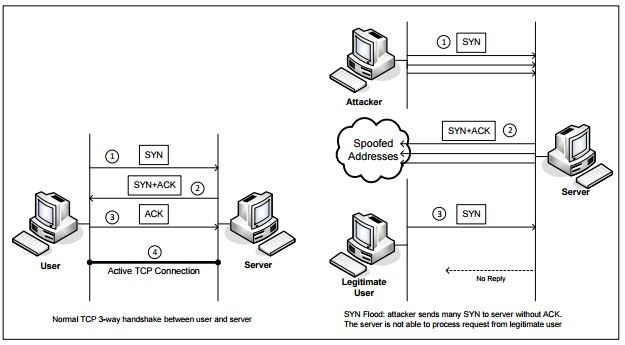
**Lab Tasks**

In this lab, you need to conduct attacks on the TCP/IP protocols. You will use the Netwox tools and/or other tools in the attacks. All the attacks are performed on Linux operating systems.

To simplify the “guess” of TCP sequence numbers and source port numbers, we assume that attackers are on the same physical network as the victims. Therefore, you can use sniffer tools to get that information. The following is the list of attacks that need to be implemented.

**Task 1: SYN Flooding Attack**

Figure 2: SYN Flooding Attack



SYN flood is a form of DoS attack in which attackers send many SYN requests to a victim’s TCP port, but the attackers have no intention to finish the 3-way handshake procedure. Attackers either use spoofed IP address or do not continue the procedure. Through this attack, attackers can flood the victim’s queue that is used for half-opened connections, i.e. the connections that has finished SYN, SYN-ACK, but has not yet gotten a final ACK back. When this queue is full, the victim cannot take any more connection. Figure 2 illustrates the attack.

The size of the queue has a system-wide setting. In Linux, we can check the setting using the following command:

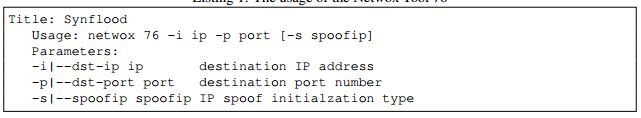
# sysctl -q net.ipv4.tcp\_max\_syn\_backlog

We can use command "netstat -na" to check the usage of the queue, i.e., the number of half opened connection associated with a listening port. The state for such connections is SYN-RECV. If the 3-way handshake is finished, the state of the connections will be ESTABLISHED.

In this task, you need to demonstrate the SYN flooding attack. You can use the Netwox tool to conduct the attack, and then use a sniffer tool to capture the attacking packets. While the attack is going on, run the "netstat -na" command on the victim machine, and compare the result with that before the attack. Please also describe how you know whether the attack is successful or not. Be sure to take screen shots of the results and include in your report.

The corresponding Netwox tool for this task is numbered 76. Here is a simple help screen for this tool. You can also type "netwox 76 --help" to get the help information.

Listing 1: The usage of the Netwox Tool 76



**SYN Cookie Countermeasure:** If your attack seems unsuccessful, one thing that you can investigate is whether the SYN cookie mechanism is turned on. SYN cookie is a defense mechanism to counter the SYN flooding attack. The mechanism will kick in if the machine detects that it is under the SYN flooding attack. You can use the sysctl command to turn on/off the SYN cookie mechanism:

# sysctl -a | grep cookie (Display the SYN cookie flag)

# sysctl -w net.ipv4.tcp\_syncookies=0 (turn off SYN cookie)

# sysctl -w net.ipv4.tcp\_syncookies=1 (turn on SYN cookie)

Run your attacks with the SYN cookie mechanism on and off, and compare the results. In your report, please describe why the SYN cookie can effectively protect the machine against the SYN flooding attack. If your instructor does not cover the mechanism in the lecture, you can find out how the SYN cookie mechanism works from the Internet.

Run Netwox command 76 on H1 to initiate a SYN flood attack. Use H2 to show the SYN and SYN-ACK messages received.

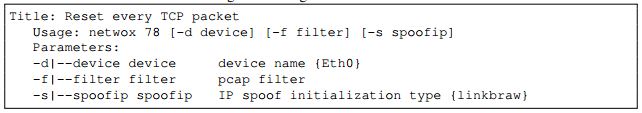
**Task 2: TCP RST Attacks on telnet and SSH Connections**

The TCP RST Attack can terminate an established TCP connection between two victims. For example, if there is an established telnet connection (TCP) between two users A and B, attackers can spoof a RST packet from A to B, breaking this existing connection. To succeed in this attack, attackers need to correctly construct the TCP RST packet.

In this task, you need to launch an TCP RST attack to break an existing telnet connection between A and B. After that, try the same attack on an SSH connection. Please describe your observations. To simplify the lab, we assume that the attacker and the victim are on the same LAN, i.e., the attacker can observe the TCP traffic between A and B.

The corresponding Netwox tool for this task is numbered 78. Here is a simple help screen for this tool. You can also type "netwox 78 --help" to get the help information. Be sure to take screen shots for your report.

Listing 2: The usage of the Netwox Tool 78

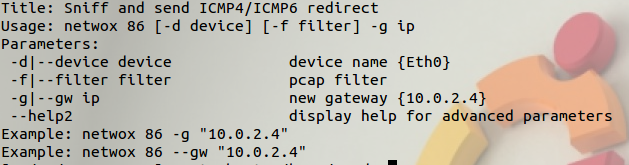


**Task 3: ICMP Redirect Attack**

The ICMP redirect message is used by routers to provide the up-to-date routing information to hosts, which initially have minimal routing information. When a host receives an ICMP redirect message, it will modify its routing table according to the message. Because of the lack of validation, if attackers want the victim to set its routing information in a particular way, they can send spoofed ICMP redirect messages to the victim, and trick the victim to modify its routing table.

In this task, you should demonstrate how the ICMP redirect attack works, and describe the observed consequence. To check the routing information in Linux, you can use the command route. You will have H1 use Netwox command 86 to redirect traffic away from the gateway. Also Wireshark will be used to show incoming ICMP redirect messages. Use H2 to telnet to the redirected host. Be sure to provide screenshots with your report.

Listing 3: The usage of the Netwox Tool 86



**Task 4: TCP Session Hijacking**

The objective of the TCP Session Hijacking attack is to hijack an existing TCP connection (session) between two victims by injecting malicious contents into this session. If this connection is a telnet session, attackers can inject malicious commands (e.g. deleting an important file) into this session, causing the victims to execute the malicious commands. Figure 3 depicts how the attack works. In this task, you need to demonstrate how you can hijack a telnet session between two computers. Your goal is to get the telnet server to run a malicious command from you. For the simplicity of the task, we assume that the attacker and the victim are on the same LAN.

Note: If you use Wireshark to observe the network traffic, you should be aware that when Wireshark displays the TCP sequence number, by default, it displays the relative sequence number, which equals to the actual sequence number minus the initial sequence number. If you want to see the actual sequence number in a packet, you need to right click the TCP section of the Wireshark output, and select "Protocol Preference". In the popup window, uncheck the "Relative Sequence Number and Window Scaling" option.

The corresponding Netwox tool for this task is numbered 40. Here is part of the help screen for this tool. You can also type "netwox 40 --help" to get the full help information. You may also need to use Wireshark to find out the correct parameters for building the spoofed TCP packet. Be sure to provide screenshots with your report.

Listing 4exit

: Part usage of netwox tool 40

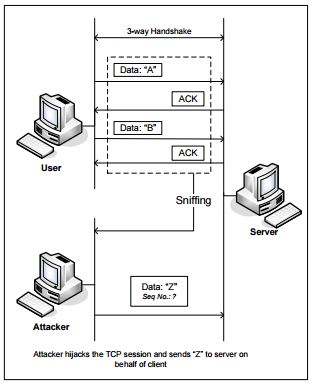
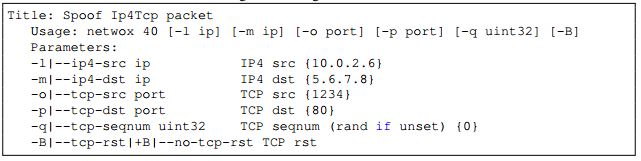


Figure 3: TCP Session Hijacking Attack

**Investigation**

The level of difficulty in TCP attacks depends on a number of factors. Please investigate the following and write down your discoveries and observations in your lab reports.

* Study the pattern of the Initial Sequence Numbers (ISN), and answer whether the patterns are predictable.
* Study the TCP window size, and describe your observations.
* Study the pattern of the source port numbers, and answer whether the patterns are predictable.

**Note**

It should be noted that because some vulnerabilities have already been fixed in Linux, some of the above attacks will fail in Linux, but they might still be successful against other operating systems.

**Report**

You should submit a lab report. The report should cover the following sections:

* **Design:** The design of your attacks, including the attacking strategies, the packets that you use in

your attacks, the tools that you used, etc.

* **Observation:** Is your attack successful? How do you know whether it has succeeded or not? What do you expect to see? What have you observed? Is the observation a surprise to you?
* Some of the attacks might fail. If so, you need to find out what makes them fail. You can find the explanations from your own experiments (preferred) or from the Internet. If you get the explanation from the Internet, you still need to find ways to verify those explanations through your own experiments. You need to convince us that the explanations you get from the Internet can indeed explain your observations.

**Grading**

Post your report in Moodle by the scheduled due date in the syllabus. Make sure to put your name and class information is on your paper. Your grade for this lab will be composed of:

* 33% - Design
* 33% - Observations
* 34% - Explanation

© 2006-2016 Wenliang Du, Syracuse University