COMP3052.SEC LAB: ATTACK AND DEFENSE

(**Thank you to Mike Pound and Joyce Addae**)

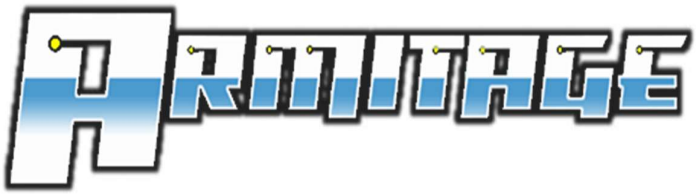
LAB DESCRIPTION

In this lab, we'll be using the hacking framework Metasploit to break into a vulnerable server. You'll then have root access to this machine, and your final task will be to secure it using the knowledge you've gained throughout the COMP3052.SEC labs and lectures.

INTRODUCTION

In an earlier lab exercise, we examined the auth.log file for a machine that had an open and accessible SSH port. Such machines are attacked on an hourly basis by computers around the world. The best solution is for government agencies to close these nets down, but it's never that easy. In the meantime, we should focus on ensuring that our machines are protected. This isn't simply a case of shutting off SSH — which is a vital tool for some users — we must protect servers while still having them remain accessible.

Metasploit is a key tool in both penetration testing and hacking. At its core is a large database of known exploits – over six hundred – targeted at a variety of operating systems and applications. These also vary from brute-force password crackers, to very specific buffer-overflow exploits. As well as exploits, Metasploit also includes hundreds of pre-attack scanners and analysis tools, mid-attack payloads and post-attack management tools. Within this software, you can scan an available host, deploy an exploit with an included payload, then take control of the machine and "loot" its contents. Metasploit is a command line tool, which, while extremely useable, is less convenient than a user interface. Luckily, as with many of these tools, a front end called Armitage has been developed. This is what we'll be using during this lab.



A good question would be, why are we telling you this? Well, knowledge of the kind of tools that hackers and security auditors use will better prepare you to secure your own machines – both at home and in the workplace.

Two Virtual Machines

Today we're again using two virtual machines, but this time it is a hacking Kali installation, and a weak Ubuntu 12 server installation. As before, we will need them to be connected to the same network: You should refer to the "Networking VMs" lab manual, where you will find instructions on setting up a network. Connect both the Kali and Ubuntu to the network (NatNetwork). Start up both machines.

Ubuntu MAC Address

This Ubuntu server does not like having its MAC address changed! If this happened when you imported the machines, you won't be able to scan or communicate with the server. This is easily fixed - we just make sure the MAC in Settings->Network->Advanced is set to 0800279E0E21. If you find the server is slow to boot, for example showing a message like "waiting for network configuration," ask for help!

Login Information

As with previous labs, a description of the Kali operating system and the general setup of the virtual machines is given in an earlier lab document, and the additional materials

Normal User	**Root**
Username: sec	Username: root
Password: security	Password: toor

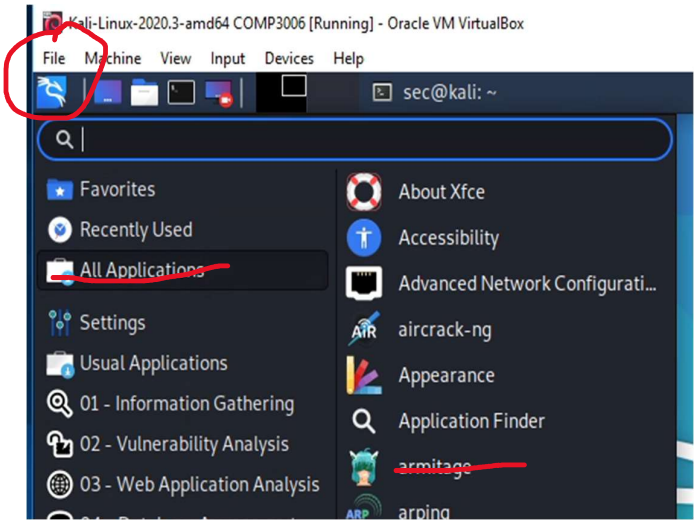
We're not going to tell you the user and password for the Ubuntu server, you'll have to find that for yourself later in the lab. I will say, though, that the server was not setup by someone competent.

Starting Metasploit

Metasploit uses a Postgres database back end, we need to start the Postgres server before we can begin. Run the following command:

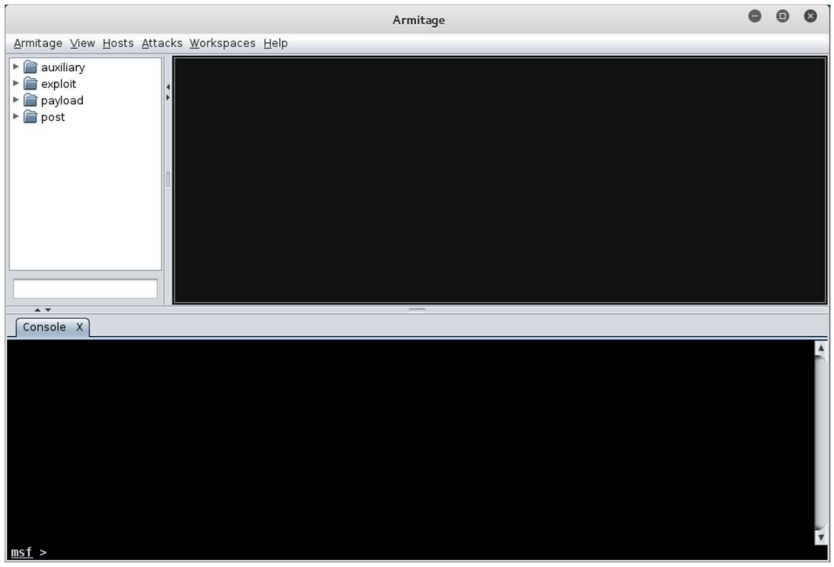
```
sec@kali:~$ sudo service postgresql start
```

The tables within postgres are already set up, we should now be able to use Metasploit. Start Armitage from “All Applications” under the main menu:



You may need to enter your password. Wait a moment, and it will prompt you to make a connection to localhost. Armitage runs Metasploit as a server, with it as a client front end. Click Connect, and you will be prompted to start the server, click Yes. **It will take a little while to connect**, don’t worry if it throws up some error messages, it’s waiting for the server to start. Once up and running, you’ll see the interface below.

The left region is the database browser, which includes all of the exploits and other tools available. It also has a text filter box, which you’ll find extremely useful. The right box is the host window, showing all of the hosts that are currently being managed by this Metasploit installation. The lower box is the command window, you can manually input Metasploit framework (msf) commands here, but it is also where tool and plugins will output their results.



Let's get started! Before we can examine the status of the Ubuntu machine, we need to know what IP address it has. We can't log in and read ifconfig, so we'll need to perform an nmap scan. You can do this on the terminal if you wish, but nmap is also built into Armitage. Click Hosts -> NMap Scan -> Ping Scan. Scan the range "10.0.2.*" or "10.0.2.1-100" and you'll see a few IP addresses appear, .1 and .2, and possibly .3 are used by VirtualBox, the Ubuntu machine will be the IP that isn't the machine you're currently on! The machines will appear in the host box as just blank screens at this point. As we learn more about the machine, this icon might change.

Note: If the IP address of the server machine does not appear, please try to increase the number of CPUs to 2, by clicking Settings -> System -> Processor**.**

Let's get some more information. Select Hosts -> NMap Scan -> Intense Scan, and point it directly at the IP of the server. It'll take a little while (a message will appear when it's finished). It should determine that we are dealing with a Linux box, and what software is installed and running on different ports. You can right click on the host and select Services to see a breakdown of the services and their versions. This information is crucial in determining vulnerabilities. Some buffer overflows, for example, are known to work only on a limited number of versions. Since this is an old server, let's see if we can exploit the heartbleed bug. Search for heartbleed on the left, or browse to auxiliary/scanner/ssl/openssl_heartbleed. Double click the module, then if it isn't already there, enter the IP of the server in RHOSTS. Click launch. The green + indicates success, this machine is indeed vulnerable to heartbleed. Let's exploit this: in the window at the bottom type:

```
msf auxiliary(scanner/ssl/openssl_heartbleed) > show actions
```

You'll see the module is capable of dumping and analysing the memory of the server. Let's do a memory dump first:

```
msf auxiliary(scanner/ssl/openssl_heartbleed) > set ACTION DUMP
msf auxiliary(scanner/ssl/openssl_heartbleed) > run
```

This time it will output the memory of the server to a file, click View -> Loot to open the loot window, then double click the file to take a look. A lot of it won't be readable (this is raw server memory) but it gives you an idea. Let's go further, set the action to KEYS then run again. This will repeatedly heartbleed the server, attempting to find the RSA private key. If it does, you'll see it on the screen. This sometimes works, sometimes doesn't — Heartbleed reads unpredictable memory after all. Any obtained private key could be used to decrypt any information sent to the server,

if the server wasn't set up to use forward secrecy, this decryption could be performed on any previous communication!

Gaining Access

We've managed to run an exploit on the server, let's now try and gain root access. For well-defended servers, this may rely on a software bug that lets you execute a shell from within some code running as root, and it may not be possible. Luckily, we don't have to worry about that — this server is poorly secured. Find the ssh_login module at auxiliary/scanner/ssh/ssh_login. This module performs a **brute force attack** on the root password, a similar attack to the ones that we saw being performed on the machine in the auth.log file. We'll need to supply a list of common passwords: there is one in /home/sec/lab7/common_passwords, add this to the "PASS_FILE" variable. Add "root" as the username, then click launch. It will begin attempting to log in, be patient!

We're in! This password is clearly not secure, but there's no reason you couldn't use a much more complex password file and leave this going for days on end. You'll see the icon for the host has changed, and this means that we have a session open with root privileges. Have a look at the output of the module, it will say "Command shell session #" opened. If we were a botnet administrator, we could now install the necessary client software to take intructions from a command and control server.

Exploiting the Root Shell

There is nothing in Linux we can't do with root access, but let's not get too carried away. The post modules in Metasploit let us manage a host that is now under our control, usually those that have a session already open. `post/linux/gather/enum_system` lets us easily gather a lot of information about who and what is operating on this system. Run this, it requires a session number, use the one that was opened by `ssh_login`. Wait a while for the module to finish, it will tell you when it does, then head to the loot viewer and inspect what it found. As a root user, there are no files off limits on a Linux machine, in particular, any personal user documents could be easily obtained this way.

Securing a Server

Given the different aspects of Linux systems we've looked at over the last few labs, you should now have a good idea about how to secure a server. This is what you should do now, you have the root account and password, make all of the changes you deem necessary to prevent intrusions onto this system. There is no limit on how secure we might consider "secure enough" in this lab, so do as much as you can. Feel free to use the web for further information. Some pointers:

- Obviously the root user / password situation must be fixed.
- Have a look at what services are running, SSH and Apache2 should be preserved because they're useful, but insecure services should be stopped.
- You don't need to run updates, they take time, but note which updates you would perform.

Conclusion

In this lab we learned to use the metasploit framework to examine and infiltrate a weakly secured server. You were then able to use all of the knowledge you've gained from the previous labs to secure this server. Knowing how to best secure a machine is an important step in really understanding the kind of attacks you might frequently see, if you are in an administrative role.