Homework 2: Botnet Lab and Attack Trace Analysis

1 Lab Overview

The goal of this lab is to introduce you to the concept of Botnets, showcase some features of popular bots, and learn to analyze attack traces.

A Botnet is a collection of software agents, or robots, that run autonomously and automatically. The term is most commonly associated with malicious software, but it can also refer to a network of computers using distributed computing software.

Meanwhile, Honeynet has been widely used to collect botnet information. Honeynet is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated, unprotected, and monitored, and which seems to contain information or a resource of value to attackers.

In this lab, You will install bot and use them to carry out attacks, and analyze the results. You will analyze a given attack trace and analyze what is happening there. You need to evaluate the analysis result and sum up the phenomena you observed in your lab report. In the report, you need to include all the necessary screenshots, the answers to all questions, and your experience/thoughts during the lab.

2 Lab Tasks

2.1 Initial setup

Your need to first install VMware player and configure it properly. In this lab, you need to build a virutal environment to simulate the real botnet scenario. You need to install two virtual machines: One `Ubuntu` Linux (any recent version) and one `Windows XP` on VMware player. Your can download free VMware player for Windows or Linux at `http://www.vmware.com/products/player/playerpro-evaluation.html`. [You may consider VirtualBox if you are using Mac.] You can download Linux OS (or pre-built Linux image for VMware) from the Internet. For your convenience, we provide a pre-built `Ubuntu` 9 Linux image for VMware at `http://faculty.cse.tamu.edu/guofei/download/ubuntu9VM-1.1.tar.gz`. In this image, you can login using two accounts:

- User ID: root, Password: seedubuntu
- User ID: seed, Password: dees

We also provide a clean Windows image for you to do this lab at `http://faculty.cse.tamu.edu/guofei/download/xp-image.tar.gz`. **[Important]: Please keep a clean copy of the Windows image and use a clone in the following lab because running malware on it will poison the OS and there is apparently no snapshot function in VMware player. In case you need the password for this Windows image, it's "1234567".**

The lab scenario is illustrated in Figure 1:

To isolate our environment from the outside network, you need to form a virtual network (as illustrated in the lab basics slides on the class website). Basically, you are recommended to group your virutal machines into a *team* and connect each virtual machine using `VLAN`. The procedure of creating a *team* and connect through `VLAN` is illustrated in Figure 2 (Note your actually software might be slightly different from this due to the version difference).

You also need to create `Snapshot` before running your bot program. Creating *Snapshot* helps you to revert to previous clean state of virtual machine, like in Figure 3.

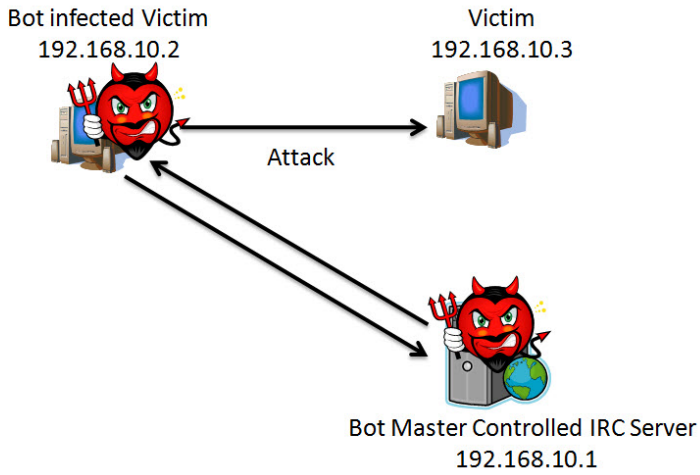

Figure 1: Botnet Scenario

In the controlled environment, the IP address of your linux and windows box can to be statically configured so they can communicate with each other. A sample basic configuration could be like:

```
  Ubuntu Box
IP: 192.168.10.1  Subnet Mask: 255.255.255.0 Gateway: 192.168.10.1
  Windows XP Box
IP: 192.168.10.2  Subnet Mask: 255.255.255.0 Gateway: 192.168.10.1
```

After that, you need to install both IRC Server and Client on your `Ubuntu` machine. IRC networks, while not as popular as many web-based chatrooms, are considered part of the "underground" Internet, and public IRC servers are home to many hacking groups and illegal software release groups, mainly because of the relative anonymity users can have while connected to IRC. Because of this, botnets are a feasible method of controlling victims without directly connecting to them. IRC servers are usually part of a network, providing multiple servers for clients to connect to (if one is closer, or less loaded), which enhances the hard-to-trace nature of IRC. We recommend using the `ircd` as the IRC Server and `XChat` as the client.

In `Ubuntu`, you simply type command:

```
$ sudo apt-get install ircd-irc2 xchat
$ sudo /etc/init.d/ircd-irc2 start
```

to get the server and client software installed. The `ircd` default binds port 6667 to receive IRC traffic. You could modify the configure file to bind any port you prefer:

```
$ sudo vim /etc/ircd/ircd.conf
```

After correct configuration, you may login your IRC server using `XChat IRC`. You could create a channel for bot to contact using command:

```
/join #<channel_name>
```

You could also test the IRC communication between your Windows and Linux box. In Windows you could free download `mIRC` (`http://www.mirc.com/`) as the IRC Client. If the connection established, you could talk to each other in the specific channel.

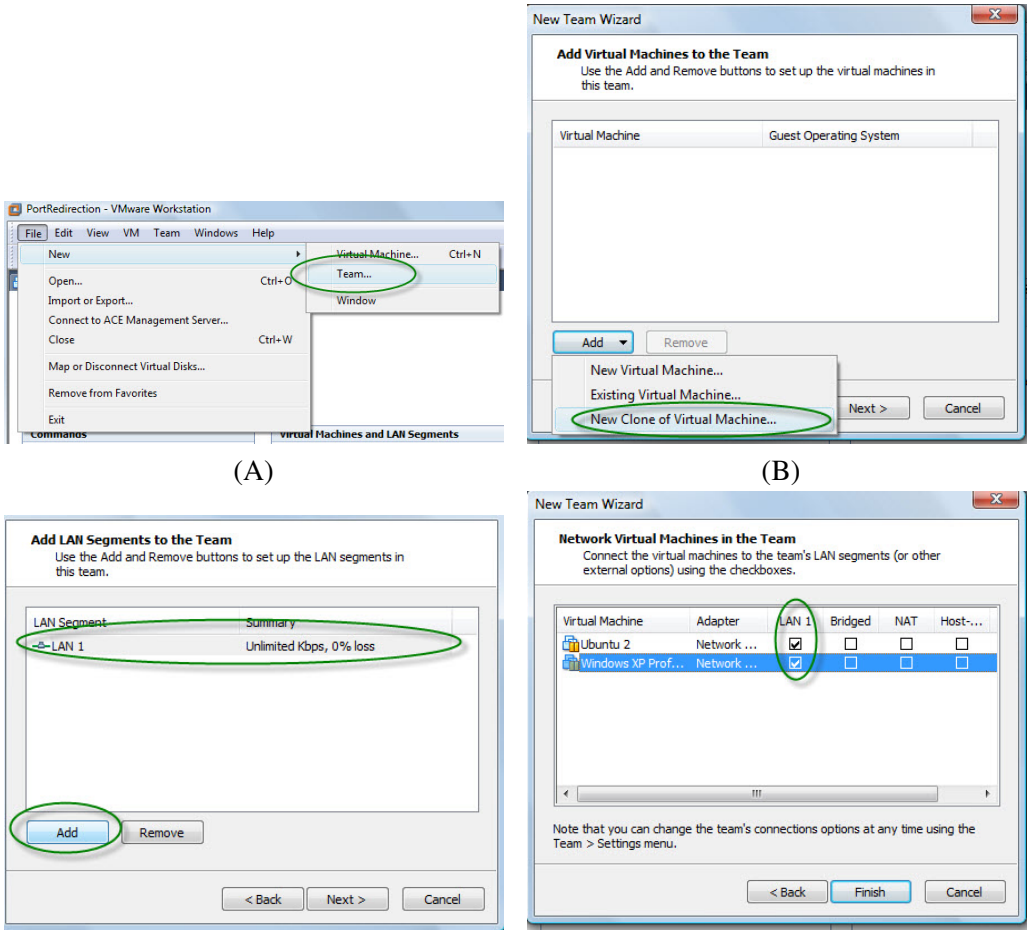

Figure 2: Setup of VMware Team and VLAN

Note: in case you have some issues with using “apt-get install” because it is a relatively old release, you can “sudo vi /etc/apt/sources.list” and replace all “http://us.archive.ubuntu.com/ubuntu” links to “http://old-releases.ubuntu.com/ubuntu/”. Then execute “sudo apt-get update”, and you should be able to install your software.

2.2 Task One: Run SDBot

The bot you will work with is `SDBot`, which is written in C and uses IRC to communicate with the bot master. It is neither the most powerful bot nor the most popular, but the setup is straightforward, and the version of the code we have has the self-replicating routines removed, so it is easier to control.

Copy the SDBot folder your Windows XP virtual machine. Because SDBot is a C program, we have to install a windows C compiler. In the SDBot folder run the file `lccwin32.exe` to install the compiler. Click through the install process, leaving all of the default options in place.Once LCC is installed, open the `sdbot05b.c` file in Wordpad and scroll down to the section labeled `"bot configuration."` Make the following changes to the listed variables:

```
1. botid[] = "f00f00" / botid[] = "<Your Bot ID>"
2. password[] = "bar" / password[] = "<Password to Login Your Bot>"
```

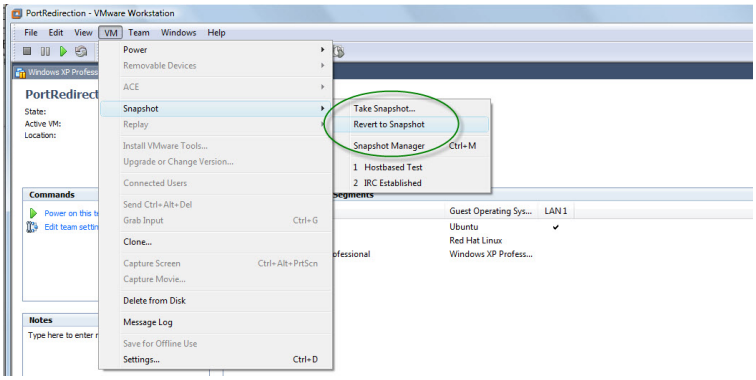

Figure 3: Manage Virtual Machine Snapshot

```
3. server[] = "irc.dal.net" / server[] = "ircserver"
4. port = 6667 / port = 6667
5. channel[] = "#foobar" / channel[] = "#<Your Channel Name>"
6. filename[] = "syscfg32-bot.exe" / filename[] = "665SDbot.exe"
```

This sets up the bot to connect to the IRC server we set up on the linux machine. Save the file as `665bot.c` and exit Wordpad. Now run the `make-lcc-665.bat` file to create a `665bot.exe` executable. This is the executable that you would need to get onto a victim machine and launch to make it part of your botnet. How to get the `.exe` onto a victim machine is beyond the scope of this lab, but recall techniques learned in Lab One. Once the `SDbot` is installed, all firewall software will need to be disabled so that it won't interfere with our experiments.

SDBot will automatically connect the `ircserver` configured. So you need to write the host DNS file. It is located at:

```
C:\Windows\System32\drivers\etc\host
```

Open it in `notepad` and append a new line at the end of file:

```
192.168.10.1 ircserver
```

Run the `665bot.exe` executable on the `Windows XP` virtual machine. Go back onto your host machine and watch the XChat window. Within a few minutes a host with random letters for a username should log into your channel. This is your bot. Log into your bot by typing:

```
.login <password> (bot responds: password accepted)
```

Screenshot One:
Take a screenshot of the XChat window showing successful login and system information printout.
Now type:

```
.repeat 6 .delay 1 .execute 1 winmine.exe
```

Q1.1. What is the result of this command?

The file `sdbot_commandref.html` is a list of commands that you can execute using SDBot. We'll take a look at a few of them now.

2.3 Task Two: Attack using SDBot

Now you need to add a new virtual machine into your environment to act as victim machine. You could simply make a another clone copy of the original linux image and connect it to the `VLAN`. The IP address could be configured as:

```
Ubuntu Victim Box
IP: 192.168.10.3 Subnet Mask: 255.255.255.0 Gateway: 192.168.10.1
```

In this machine, you need to install the `wireshark` to capture the receiving packets.

```
$ sudo apt-get install wireshark
$ sudo wireshark &
```

2.3.1 UDP Flood

We will now use our bot to execute a UDP flood attack against your victim machine.

1. Open up `Wireshark` on the host machine and filter the packets with these expressions:

```
((ip.src==<XP ip>) && (ip.dst==< Victim ip>) && udp)
```

2. Click on the `Capture`.

3. Use the command reference page to find the command for a UDP flood. Use the command to send 1000 4096-byte packets to port 23 of victim machine. Use a 1 ms delay.

4. Wait until the bot displays "Finished sending packets to Victim IP".

5. Stop WireShark.

6. Click on the `Statistics/Summary` on the Wireshark menu bar and check the Avg MBit/s traffic Displayed

Q2.1. What command did you use?

Q2.2. What happens if you don't specify the port number to use for the UDP flood?

Q2.3. How many bots would be needed to flood a 1 Gbit link with UDP packets?

Q2.4: How might this attack be prevented from the perspective of the flood target? From the perspective of the infected victim?

2.3.2 Ping Flood

Now we'll use the bot to execute a `PING` flood attack against the same target.

1. Open up `Wireshark` and filter the packets with these expressions:

```
((ip.src==<XP ip>) && (ip.dst==< Victim IP >) && icmp)
```

2. Click on the `Capture`.

3. Use the command reference to find the command for a PING flood. Use 1000 packets of size 4096, sent to the Victim machine. Use a 1 ms delay.

4. Wait until the bot displayed "finished sending packets to Victim IP".

5. Stop `Wireshark`.

6. Click on the `Statistics/Summary` and Check the Avg MBit/s traffic Displayed

Q2.5. What command did you use?

Q2.6. How many bots would be needed to flood a 1 Gbit link with ICMP packets?

Q2.7. From the result of the two floods, which one is more efficient: UDP or ICMP flood?

Q2.8. Based on your answer to question 2.7, when would you not use the more efficient one?

2.3.3 Fraudulent Pay-per-click Count

Before this task, you need visit `http://<Victim IP>` in your `Windows XP` box to verify the HTTP web service in your victim machine is working properly.

Another use that botnets have been put to is to generate a fraudulent number of webpage referrals in pay-per-click advertising schemes. This is how it works: An advertising agency puts up a ”banner” on an individual's webpage, and pays the individual a nominal amount every time a visitor to the webpage clicks on the banner (which is a link to the sponsor's website). Botnets can be used to generate large numbers of false ”clicks” on these banners, thus fraudulently earning the individual a lot of money. This is how this is accomplished:

1. Open up `Wireshark` and filter the packets with these expressions:

```
(((ip.src==<WinXP IP>) && (ip.dst==< Victim IP >) && tcp)
||(ip.src==< Victim IP > && (ip.dst==< WinXP IP >) && tcp))
```

2. Click on the `Capture`.

3. SDbot command for fraudulent pay-per-click:

```
.visit http://< Victim IP >/index.html http://bot.com
```

4. Wait until the bot displayed “url visited”.
5. Stop `Wireshark`.
6. Now examine any Tcp packet received.

Screenshot 2: Take a screenshot of the tcp stream showing the source and referrer web page.

2.4 Task Three: Bot Removal

Open up the Task Manager `(Ctrl+Alt+Del)` and you should see the bot running under the conspicuous process name `665SDBot.exe`; if you were trying to hide the bot, you would, of course, pick a much less obvious name. Use the Task Manager to kill the process and restart your virtual machine. Once it has rebooted open up Task Manager again. Your bot should still be running. This is one of the most powerful things about bots; once you infect a computer, it stays infected (unless the user gets smart and fully deletes it).

1. Use Task Manager to kill the process again.
2. Open the file ”665bot.c”.
3. Search for the function ”void uninstall (void)” and examine its code. From this, you should be able to tell what the names of SDBot's registry entries are.

Q.3.1. Where are the registry entries? Why are the entries placed in these two locations?

4. Open the registry editor by clicking `Start->Run` and typing in ”regedit”.
5. Delete the registry entries as described by the source code and restart the virtual machine.
6. Verify that `665SDbot.exe` and `665bot.exe` no longer show up as processes in Windows Task Manager.

Q.3.2. How would a user know where in their registry the bot is located if the source code were not available for inspection?

3 Task Four: Attack Trace Analysis

Please analyze the trace attack-trace.pcap. You may use tools such as wireshark/tshark, p0f, whois, geoiplookup.

Q4.1. What IP addresses (and their roles) are involved?

Q4.2. Where is the attacker located?

Q4.3. How many TCP sessions are contained in the pcap file?

Q4.4. How long did the attack last?

Q4.5. Which operating system was targeted by the attack? And which service? Which vulnerability?

Q4.6. Can you sketch an overview of the general actions performed by the attacker?

Q4.7. What specific vulnerability was attacked?

Q4.8 (Bonus question). What actions does the shellcode perform? Pls list the shellcode. (You may use tools like Ollydbg, IDA, or Libemu.)

Q4.9. Was there malware involved? Can you find out the name of the malware?