cs3235:6: - Laboratory #6 for October 9th, 2019

> Perform the laboratory individually, or in pairs (ie - a group of 2). Get comfortable using `traceroute`, `nmapx`, `zenmap` and `wireshark` on the machines in the ITSEC/OS laboratory. Note that if you want you can install ALL these programs on your own machine at home to try them.

Every machine in the lab has an assigned network (IP) address, a single 32-bit integer, normally written as four numbers: 192.168.100.1. Each of the four numbers can have a value from 0 to 255 (and thus you only need 8 bits to represent each number: 11111111 in binary happens to be 255). All the machines in the lab lie between 192.168.100.0 and 192.168.100.255, and it is common to refer to the "network" as 192.168.100.*.

You will have to discover the specific IP information for your own machine for yourself, using the command `ifconfig`.

1 Lab 6, part 1: Using tools - nmap, wireshark

This section will familiarize you with `nmap`, a nugget in a network hacker's toolchest. While doing this laboratory, it is not absolutely necessary to wear Trinity's dark glasses. But if you want, maybe Hugh will have some spare chocolate fish for the suitably attired.

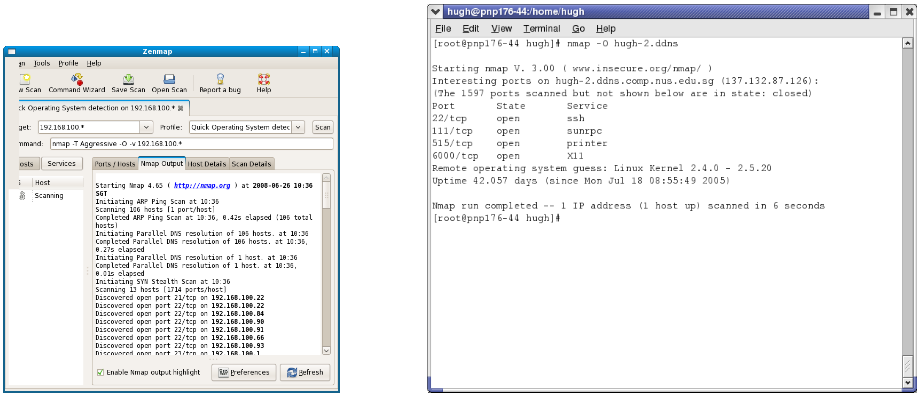
You can find `nmap` pre-installed on the lab machines, or at the home site at `http://nmap.org/`.

The program `nmap`, short for "network mapper" probes a single computer or a whole network (ie, all the computers with addresses in a specified consecutive range) for services that run on the probed machines. The program normally runs as an administrative (root) user to get all its capabilities available. On the systems you will find a version of `nmap` called `nmapx` which will run as an administrative (root) user. You can run it in a terminal window with commands like this:

```
nmapx 192.168.100.123
nmapx 192.168.100.123 > resultofscanning.123.txt
nmapx -sP 192.168.100.0/24
```

While performing its probe, `nmap` (or `nmapx`) can take care to avoid being detected. It can also make a very good guess about the architecture and operating system of the probed computer (for example, how would you determine the operating system running on `129.128.100.123`?).

Start by reading about nmap and try to understand as much as you can. You are not expected to be a TCP/IP wizard, and a rough understanding of how nmap achieves its goals is adequate. Try to understand how nmap/nmapx *stealthily* scans computers. Read nmap's man page.



get: 192.168.100.* Profile: Quick Operating System detec

mmand: nmap -T Aggressive -O -v 192.168.100.*

```
Starting Nmap 4.65 ( http://nmap.org ) at 2008-06-26 10:36
SGT
Initiating ARP Ping Scan at 10:36
Scanning 106 hosts [1 port/host]
Completed ARP Ping Scan at 10:36, 0.42s elapsed (106 total
hosts)
Initiating Parallel DNS resolution of 106 hosts. at 10:36
Completed Parallel DNS resolution of 106 hosts. at 10:36,
0.27s elapsed
Initiating Parallel DNS resolution of 1 host. at 10:36
Completed Parallel DNS resolution of 1 host. at 10:36,
0.01s elapsed
Initiating SYN Stealth Scan at 10:36
Scanning 13 hosts [1714 ports/host]
Discovered open port 21/tcp on 192.168.100.22
Discovered open port 22/tcp on 192.168.100.22
Discovered open port 22/tcp on 192.168.100.84
Discovered open port 22/tcp on 192.168.100.90
Discovered open port 22/tcp on 192.168.100.91
Discovered open port 22/tcp on 192.168.100.66
Discovered open port 22/tcp on 192.168.100.93
Discovered open port 23/tcp on 192.168.100.1
```

☑ Enable Nmap output highlight

hugh@pnp176-44:/home/hugh

```
[root@pnp176-44 hugh]# nmap -O hugh-2.ddns

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on hugh-2.ddns.comp.nus.edu.sg (137.132.87.126):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
515/tcp    open        printer
6000/tcp   open        X11
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 42.057 days (since Mon Jul 18 08:55:49 2005)

Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
[root@pnp176-44 hugh]#
```

Figure 1: The program on Linux, GUI (zenmap) and command line (nmap)

Use both the GUI and command line versions of nmap (the GUI version is zenmap). The zenmap program actually just uses the nmap program, and you can see the nmap command it will run on the screen. Try running nmapx directly using the same commands, for example:

```
nmapx -T aggressive -O -v 192.168.100.105
```

Use zenmap or nmapx to scan all the machines in the lab to determine what ports are open (try the different options from the *Profile* selector). Using the information from the scans, identify possibly vulnerable machines or services[1].

1.1 Wireshark

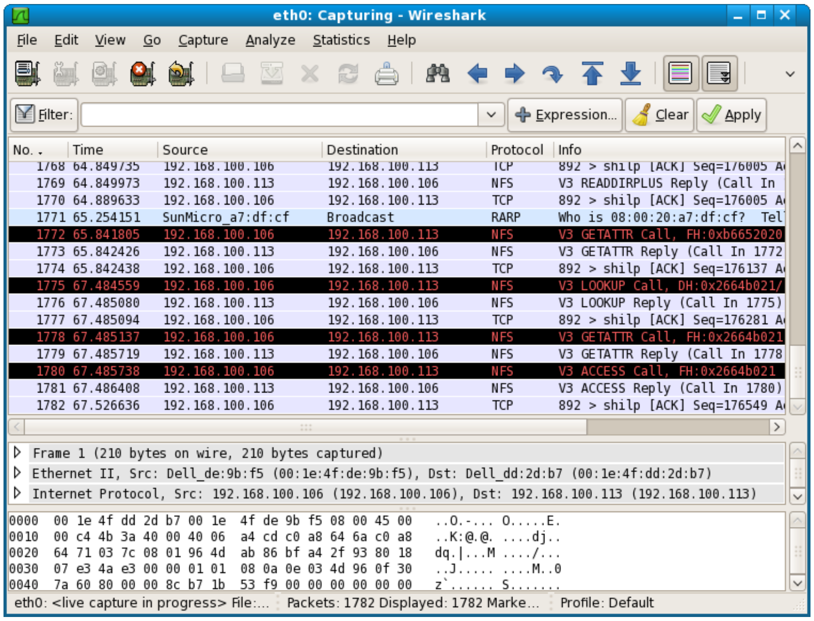
While OS fingerprinting a machine, use wireshark to capture the packet flow between the two machines. This can be done by running wireshark (unprivileged) on the machine running nmapx and setting a filter to capture IP traffic to and from the target machine. Can you identify from the wireshark trace whether the machine is being scanned by nmapx? Hint: Are there funny packets that nmapx uses to do its fingerprinting that can be manually identified in a packet trace? Are there suddenly too many connections to the target machine from a single machine?

1.2 Capturing logins, and reconnaissance

On UNIX systems, it is relatively easy to remotely login to OTHER computers (ie other than the one you are sitting in front of), if you have an account on that other computer. Try to remotely login to the computer next to you. You will first need to find out its IP address, and then use

```
ssh -X a00**@192.168.100.***
```

[1] There ARE some open machines in the lab, please do not modify them.


eth0: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression...
Clear
Apply

No.	Time	Source	Destination	Protocol	Info
1768	64.849735	192.168.100.106	192.168.100.113	TCP	892 > shilp [ACK] Seq=176005 A
1769	64.849973	192.168.100.113	192.168.100.106	NFS	V3 READDIRPLUS Reply (Call In
1770	64.889633	192.168.100.106	192.168.100.113	TCP	892 > shilp [ACK] Seq=176005 A
1771	65.254151	SunMicro_a7:df:cf	Broadcast	RARP	Who is 08:00:20:a7:df:cf? Tel
1772	65.841805	192.168.100.106	192.168.100.113	NFS	V3 GETATTR Call, FH:0xb6652020
1773	65.842426	192.168.100.113	192.168.100.106	NFS	V3 GETATTR Reply (Call In 1772
1774	65.842438	192.168.100.106	192.168.100.113	TCP	892 > shilp [ACK] Seq=176137 A
1775	67.484559	192.168.100.106	192.168.100.113	NFS	V3 LOOKUP Call, DH:0x2664b021/
1776	67.485080	192.168.100.113	192.168.100.106	NFS	V3 LOOKUP Reply (Call In 1775)
1777	67.485094	192.168.100.106	192.168.100.113	TCP	892 > shilp [ACK] Seq=176281 A
1778	67.485137	192.168.100.106	192.168.100.113	NFS	V3 GETATTR Call, FH:0x2664b021
1779	67.485719	192.168.100.113	192.168.100.106	NFS	V3 GETATTR Reply (Call In 1778
1780	67.485738	192.168.100.106	192.168.100.113	NFS	V3 ACCESS Call, FH:0x2664b021
1781	67.486408	192.168.100.113	192.168.100.106	NFS	V3 ACCESS Reply (Call In 1780)
1782	67.526636	192.168.100.106	192.168.100.113	TCP	892 > shilp [ACK] Seq=176549 A

```
▷ Frame 1 (210 bytes on wire, 210 bytes captured)
▷ Ethernet II, Src: Dell_de:9b:f5 (00:1e:4f:de:9b:f5), Dst: Dell_dd:2d:b7 (00:1e:4f:dd:2d:b7)
▷ Internet Protocol, Src: 192.168.100.106 (192.168.100.106), Dst: 192.168.100.113 (192.168.100.113)
```

```
0000  00 1e 4f dd 2d b7 00 1e  4f de 9b f5 08 00 45 00   ..O.-... O.....E.
0010  00 c4 4b 3a 40 00 40 06  a4 cd c0 a8 64 6a c0 a8   ..K:@.@. ....dj..
0020  64 71 03 7c 08 01 96 4d  ab 86 bf a4 2f 93 80 18   dq.|...M ..../...
0030  07 e3 4a e3 00 00 01 01  08 0a 0e 03 4d 96 0f 30   ..J..... ....M..0
0040  7a 60 80 00 00 8c b7 1b  53 f9 00 00 00 00 00 00   z`...... S.......
```

eth0: <live capture in progress> File:... Packets: 1782 Displayed: 1782 Marke... Profile: Default

Figure 2: Wireshark on Linux

Once you have logged in to that computer, your programs will run on that computer, even though the display will be on your computer. Use the process status command to verify this: `ps`.

Use `/usr/sbin/wireshark` to capture the network traffic for the `ssh` login. Can you "see" your password?

The mysterious user cs5 suspects that someone has discovered his/her login password and is interested in who is logging into his/her account. User cs5 has set up an automated system which records timestamped logins to a file called sshlogins, and retrieves these remotely using FTP (A file transfer protocol) via an automated script which runs about once every minute.

Use `/usr/sbin/wireshark` to capture the network traffic for the `ftp` login. Can you "see" cs5's password now?

Reconnaissance: assuming you have captured cs5's password...

- Log in as user cs5, and look around his/her home directory. Please do not change anything (files or passwords) in cs5's account. Perhaps you can discover other Really Secret Passwords by looking in places that no-one would ever think of looking in.
- Access the cs3235 grading website, and enter your username, password, and the admin password along with whatever you want to give yourself as a mark:
  `https://hugh.comp.nus.edu.sg/cs3235/lab6/gradeslab6-1.php`
  You can give yourself a mark that you would be proud to have.

2 Lab 6, part 2: SEED TCP/IP attack lab

Have a look at the SEED TCP/IP attack lab, found at

https://www.cis.syr.edu/~wedu/seed/Labs_16.04/Networking/TCP_Attacks/

There is a lot in this laboratory, but I would like you to just try task 4, TCP session hijacking. As you work through it, come up with a series of clear instructions that would let others repeat your hijack.

Access the cs3235 grading website, and enter in a detailed description of how you did the TCP hijack:
https://hugh.comp.nus.edu.sg/cs3235/lab6/gradeslab6-2.php

3 Lab 6, part 3: Services, issues

In this part of the laboratory, you will be investigating DoS attacks. Your final goal will be to discover how difficult it is to deny people access to a web server.

The first step is to observe that there is a web server running on each of the lab machines. The web server is commonly called `httpd` (for the HTTP Daemon) or apache (for the apache daemon). Verify that your web server is running by running a browser against the URL `http://localhost/`, or `http://192.168.100.XXX/` (from another machine). If everything is working fine you should see an initial web page. How fast does your web server serve up the page?

You might also want to look at the processes and threads in the computer, and count the webserver threads:

```
   a00XXXX@host$ ps -eLf | grep apache
```

Try out the slow loris attack

Visit the web site `https://github.com/llaera/slowloris.pl`, and download the ZIPfile. Unzip it and run the attack against your web server:

```
   a00XXXX@host$ chmod +x slowloris.pl
   a00XXXX@host$ ./slowloris.pl -dns localhost
or (from another host)...
   a00XXXX@host$ ./slowloris.pl -dns 192.168.100.XXX
```

While the attack is going on, try to browse/refresh the web page. Look at the processes/threads in the computer running the web server, and count the processes/threads. Run wireshark while you are doing a `slowloris` attack.

Assessment

When you are finished, consider the following questions:

1. Why does `nmap` normally run as the root user? What security issues are raised when a normal user can run a version of `nmap` (like `nmapx`) which can run as a root user?

2. User cs5 should be using another program (not ftp) to retrieve the file. What would be a suitable program, and why is it better?

3. What was the web server you attacked. Was it a recent version of the web server? Which web server has the most market share of web servers in the world?

4. The webserver was set up and started by the root user on the Ubuntu machines in the lab. Why are you (a normal unix user) not able to start a web server on port 80? What is the "security" issue?

5. Why is the web server running as www-data, and not as the user root? What is the "security" issue?

When you are finished with the laboratory, access the cs3235 grading website, and enter your username, password, and the answers to the previous questions (1-5), in order:
`https://hugh.comp.nus.edu.sg/cs3235/lab6/gradeslab6-3.php`