DOTA: Laboratory #2

> Perform the laboratory individually. Get comfortable using `traceroute`, `nmapx`, `zenmap` and `wireshark` on your virtual machine. Note that if you want you can install ALL these programs on your own machine as well to try them.

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

1 Lab 2, part 1: Using tools - nmap, wireshark

This section will familiarize you with `nmap`, a nugget in a network hacker's toolchest. You can read about nmap at the home site at `https://nmap.org/`. While doing this laboratory, it is not absolutely necessary to wear  Trinity's dark glasses. But if you want....

You will have to install nmap on the SEED virtual machine. This is done in a terminal window, and will take a little while:

```
sudo apt update
sudo apt install nmap
mkdir Lab2Install
cd Lab2Install
wget http://archive.ubuntu.com/ubuntu/pool/universe/p/pygtk/python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
sudo apt install ./python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
wget http://archive.ubuntu.com/ubuntu/pool/universe/n/nmap/zenmap_7.60-1ubuntu5_all.deb
sudo apt install ./zenmap_7.60-1ubuntu5_all.deb
sudo apt install libcanberra-gtk-module
```

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. You can run it in a terminal window with commands like this:

```
nmap -sP 10.0.2.0/24    # this is the internal virtualized network
```

In the above command, 10.0.2.0 is a 32-bit IPv4 network address (in hex `0A.00.02.00`, in binary `00001010.00000000.00000010.00000000`). The `/24` indicates that the first 24 bits are the network, leaving the last 8 bits for individual address - i.e. `10.0.2.0/24` is all the machines from `10.0.2.0` to `10.0.2.255`. Note that nowdays, if you do intense scans, some sites then block you for a period. For example if you did an intense scan of www.govt.nz, it will eventually block you for a day or so.

While performing its probe, `nmap` 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 `hugh.comp.nus.edu.sg`?).

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` *stealthily* scans computers. In a terminal window, run `man nmap` to read the manual.

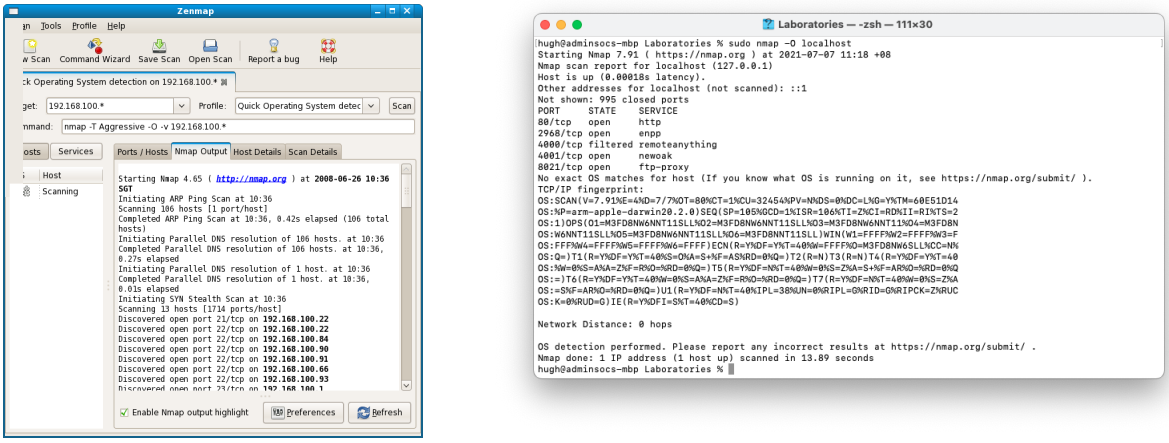


Figure 1: GUI (zenmap) and command line (nmap)

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. Use `zenmap` or `nmap` to scan all the machines on your local (i.e. real) network 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 Wireshark

While OS fingerprinting a machine, use `wireshark` to capture the packet flow between the two machines. This can be done by running `wireshark` on the machine running `nmap` 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 `nmap`? Hint: Are there funny packets that `nmap` 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?

On your SEED VM, stop all programs (Click on the X at the top right), except for a `terminal` window, and `wireshark`. Use `wireshark` to capture traffic on all the interfaces. You should see no, or very little, traffic. In the `terminal` window, `ping` the NZ government website:

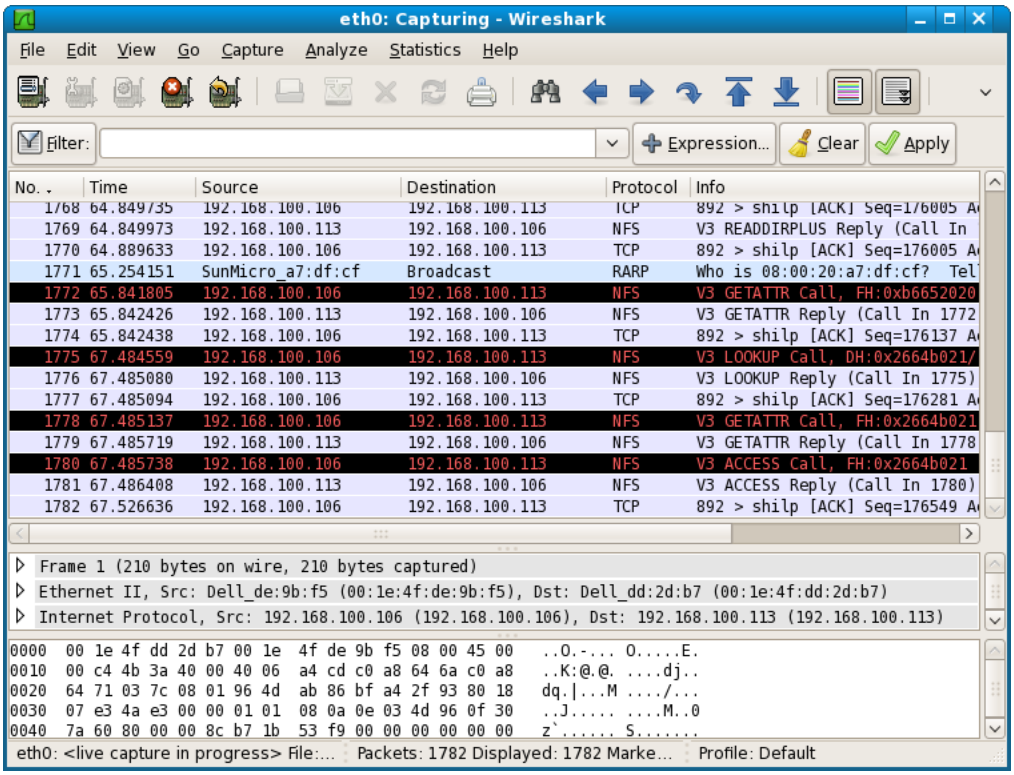


Figure 2: Wireshark on Linux

```
ping www.govt.nz        # (and then quickly ctrl-C)
```

The ping and the return value should be clearly visible (Look for ICMP). How long did the message take to get to the server and return? What are all the other packets?

1.2 Fingerprinting a machine

On your SEED VM, use `nmap/zenmap` ("Quick scan plus") to discover the services, and the OS, that is running on the machine localhost (What is this machine localhost?) Assuming you now know the OS and version number:

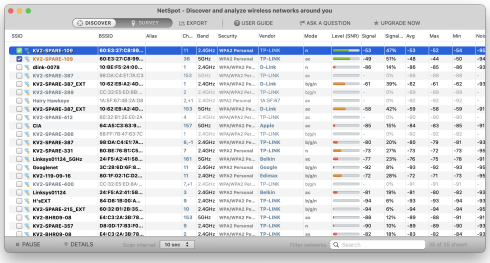
- Access the DOTA grading website, and enter your username, password, and the OS and version (12 characters) along with whatever you want to give yourself as a mark:
  `https://hugh.comp.nus.edu.sg/DOTA/lab2/gradeslab2-1.php`
  You can give yourself a mark that you would be proud to have.

2 Lab 2, part 2: Getting started with Wifi

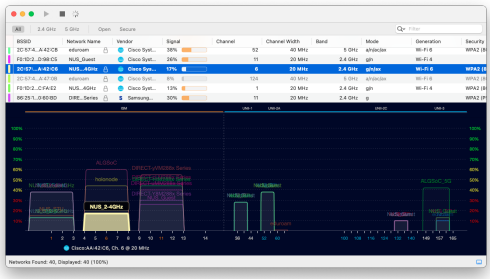
Unfortunately, some of the tools mentioned here cannot be run on all platforms.

The program `kismet` is a useful program on Linux which allows you to monitor Wifi transmissions. Wardrivers[1] often use `kismet` to record networks as they move around, with a GPS unit to record the locations of each network. If you wish to try it, you could install `kismet` on the SEED virtual machine.

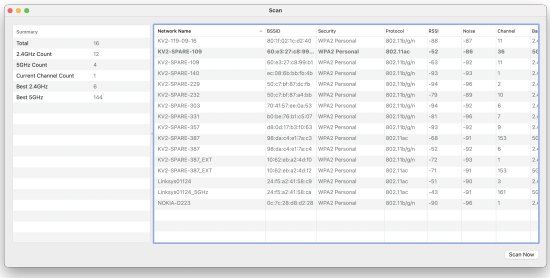
However, there are easier-to-install programs which you could install similar programs for viewing WiFi on your own computer. For example, there is also `NetSpot`, or `WifiExplorerLite`, or even the inbuilt Wireless Diagnostics for a Mac, or `WifiInfoView` for a Windows machine:



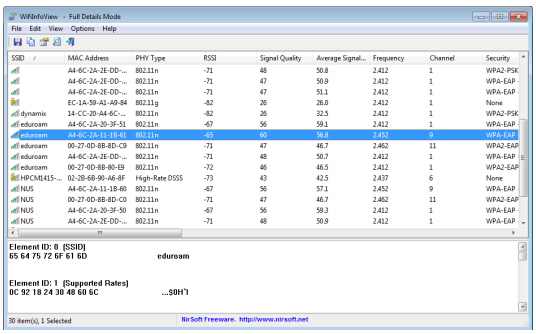
(a) NetSpot on Mac



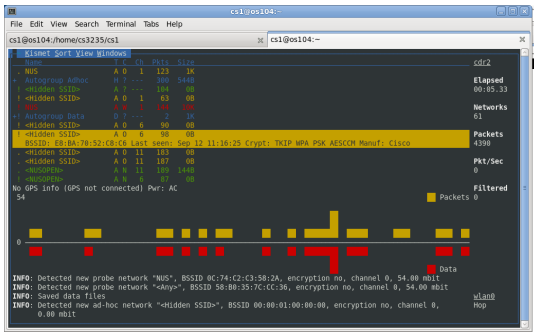
(b) WifiExplorerLite on Mac



(c) Inbuilt diagnostics on Mac



(d) WiFiInfoView on Windows



(e) Kismet on Linux

Figure 3: Wifi Capture Software

[1] If you do not recognize the term wardriver, look it up!

It is my expectation that you install and try one of the Wifi viewers, on your own machine.

The Wifi wireless spectrum is divided into a number of fixed channels (these are just different frequencies, the same way that different radio stations have different frequencies). The tools hop from channel to channel looking for Wifi transmissions.

Access the DOTA grading website, and show a snapshot (i.e. take an image of your desktop showing the application) showing your local Wifi environment. You will most likely need to upload the image to a server somewhere, and put in a link to that server in your description:

```
<img src="https://yoursite/xxx.png"/>
```

As usual the site is:

`https://hugh.comp.nus.edu.sg/DOTA/lab2/gradeslab2-2.php`

You should briefly describe your local environment - maybe something like:

```
I saw just my own network machines (A and B),
or,
I saw a lot of networks from my neighbours (X,Y and Z),
or something else...
...
And here is a screenshot of my activity:
      <img src="https://yoursite/xxx.png"/>.
```

After you submit, you will see what I will hopefully see. If it is not correct, submit again. The last submission is the one used.

3 Lab 2, part 3: WPA dictionary attack

These days, most Wifi transmissions are encrypted. The original standard was WEP (Wired-Equivalent-Privacy - a very bad name as it turned out). However, in 2001, crypto researchers discovered weaknesses in WEP, and it is now possible to *crack* a WEP key in a few seconds, if certain packets can be captured. As a result, more secure encryption techniques have been developed.

The WPA (Wifi Protected Access) and WPA2 standards are much better, but there are still attacks possible. We will look at the WPA passive dictionary attack, which relies on two elements:

1. That you are able to (passively) capture the WPA handshake, a series of messages between an access point and a host, when the host connects to the access point.
2. That you have a dictionary of strings or words, that contains the password used for the WPA.

I have captured a WPA handshake between my Mac and a Linksys router, and put the results in a capture file, found at `https://www.comp.nus.edu.sg/~hugh/DOTA2022/Labs/Lab2/wpa.cap`.

Use `wireshark` on the file to see its contents:

```
sudo apt install aircrack-ng
wireshark wpa.cap
```

Good password dictionaries can be found by hunting around on the Internet. Many of them will include quite complex words; for example `take1aspirin2day` is found in one dictionary I used to

use. Such dictionaries would also include all words formed by replacing l's with 1s, o's with 0's, S's with 5s and so on. There is a dictionary found on every unix system, used for the spell-checker. You can look at it on our linux VMs by typing

```
more /usr/share/dict/words
```

Try cracking the capture file using the program `aircrack-ng`:

```
aircrack-ng -a 2 -w /usr/share/dict/words wpa.cap
```

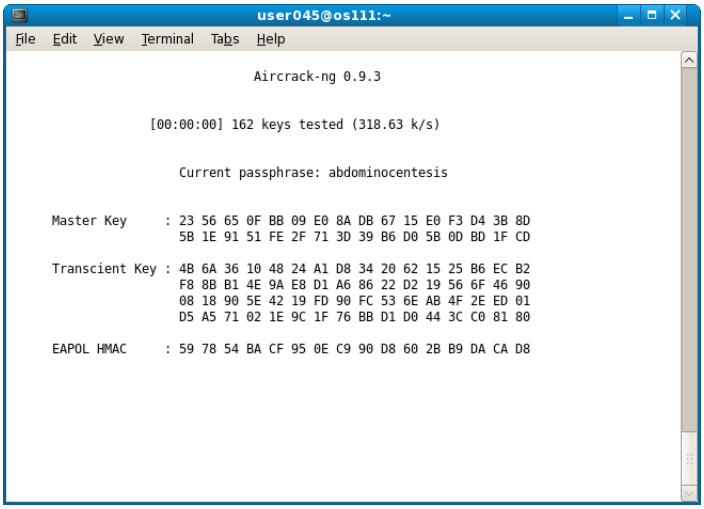


Figure 4: Aircrack-ng on Linux

What is the password for the WPA encrypted channel? How long did the crack take? What was the rate of checking words?

Access the DOTA grading website:

`https://hugh.comp.nus.edu.sg/DOTA/lab2/gradeslab2-3.php`

You can give yourself a mark that is one more than everyone elses.