DOTA: Laboratory #3 (Networks)

This laboratory uses aircrack to discover a WPA password[1], and a software defined radio to discover codes that could protect the world from catastophic events! The first parts of the laboratory should be done individually. Part 4 of the lab will involve sharing an SDR between two of you.

1 Lab 3, 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....

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.

[1] Note that at NUS we use a better scheme than the one explored here...

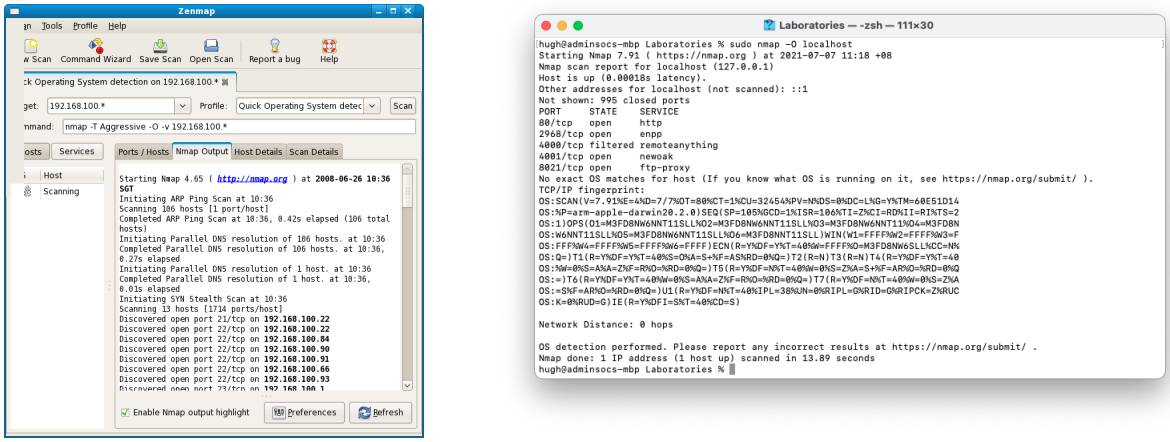


Figure 1: GUI (zenmap) and command line (nmap)

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 (what are the network IP addresses 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 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?

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:

```
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 machine, use `nmap/zenmap` ("Quick scan plus") to discover the services, running on the machines in the lab. One of them is running a (very insecure) telnet service. What is the IP address of that machine?

- Access the DOTA grading website, and enter your username, password, and IP address of the telnet machine along with whatever you want to give yourself as a mark:

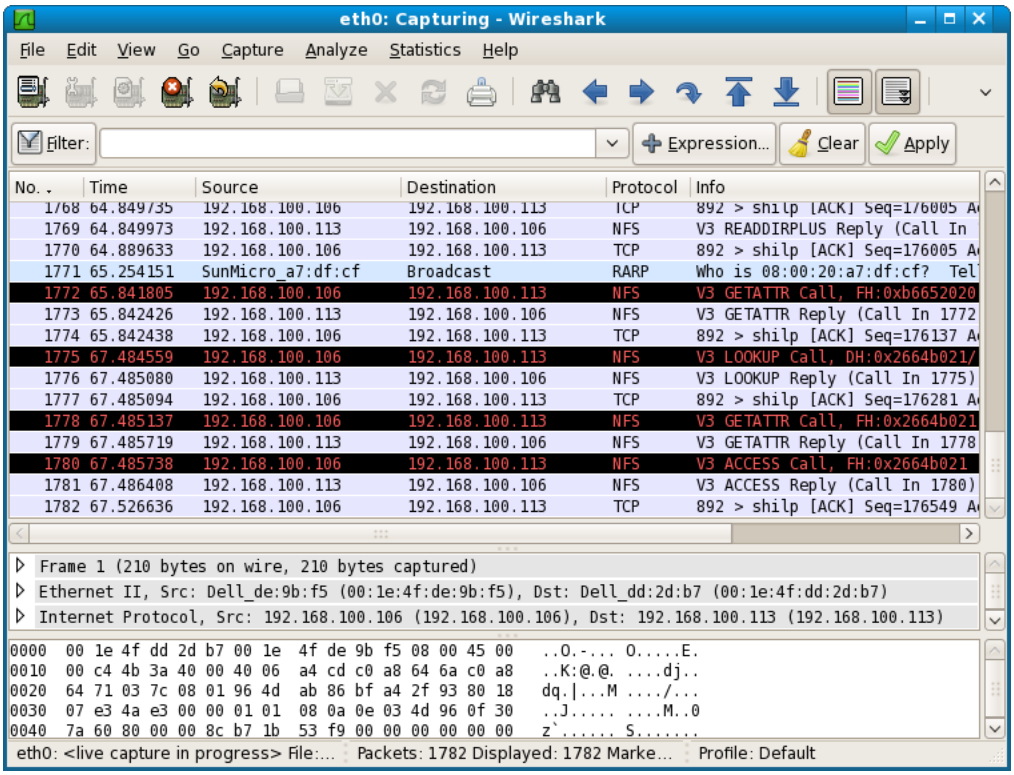


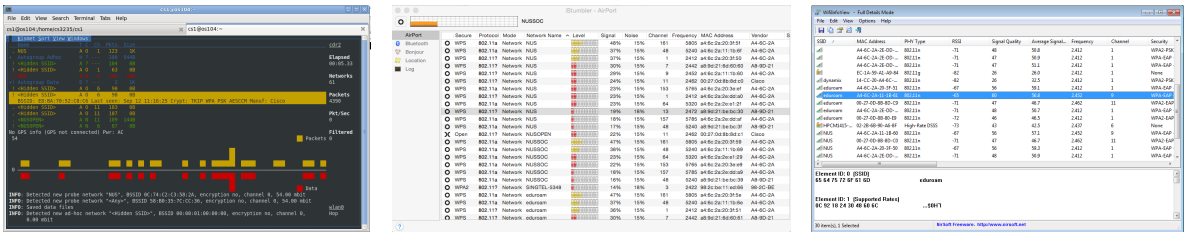
Figure 2: Wireshark on Linux

`https://hugh.comp.nus.edu.sg/DOTA/lab3/gradeslab3-1.php`
You can give yourself a mark that you would be proud to have.

2 Lab 3, part 2: Getting started with Wifi

Unfortunately, we only have a small number of Wifi network adapters, and so we will have to share them. I have attached a Wifi adapter to the server machine, and have run a program on that machine which allows you to use the `kismet` program on any of the linux boxes, and share this single Wifi adapter. It is all a bit unstable, but we will just have to make the best of it.

The program `kismet` is a useful program on Linux which allows you to monitor Wifi transmissions. Wardrivers[2] often use `kismet` to record networks as they move around, with a GPS unit to record the locations of each network. The program normally uses the wireless adapter on your PC, but since we only have a few adapters, we will use `kismet` in a different way. You could install similar programs for viewing WiFi on your own computer. For example, there is also `iStumbler` for a Mac, or WiFiInfoView for a Windows machine:



(a) Kismet on Linux (b) iStumber on Mac (c) WiFiInfoView on Windows

Figure 3: Wifi Capture Software

To run `kismet` using the remote Wifi capture device, use the command

```
kismet_client
```

When it starts, it will ask you if you want to start the kismet_server - you should answer No. After this, set the "server" to be `cdr4`:

```
Kismet
    +----> Preferences
                +-------> Servers
                     +-------------> cdr4  (instead of localhost)
```

After changing this preference, and restarting `kismet`, the remote `kismet` capture device is on the computer `cdr4.comp.nus.edu.sg`, from port 2501.

The user interface is a character-oriented one, rather than a GUI-oriented one. There are some GUI-oriented versions of `kismet` (`gkismet`, `kismet-qt`), but people seem to mostly use this character-oriented one. When using this version of `kismet`, you can use the mouse, or also character commands. For example, if you type the characters '`~Ss`', you change the sorting order to "`SSID`[3]".

The `kismet` display should fairly quickly show wireless access points and systems available in some sort of sorted order. The default sorting order is called "`autofit`", shown at the top left of the window. Change the sorting order to `SSID`. You will now be able to select individual wireless access points, by clicking on them, and for each one find out detailed information.

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

[3] If you do not know what SSID is, find out.

You will see lots of WiFi networks around the laboratory. Which company is the manufacturer of the WiFi access points? You will need the "Manuf" to access the DOTA grading website (you know the drill by now):

`https://hugh.comp.nus.edu.sg/DOTA/lab3/gradeslab3-2.php`

You can give yourself whatever mark you believe you have in the past deserved.

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). `Kismet` hops from channel to channel looking for Wifi transmissions. You can stop this channel hopping if you wish to examine one channel more closely, but note that this will stop channel hopping for ALL other users using the remote Wifi capture device (so remember to restart the hopping).

> When you finish using kismet, be careful about finishing the program. This version wants to stop the kismet capture server, which will annoy everyone else doing the laboratory. Either DISCONNECT before you QUIT, or use ctrl-C to quit, or if you exit the program normally, select "Leave" not "Kill" on the last screen. If you do manage to shut down the kismet server, ask the tutor to restart it.

3 Lab 3, 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 minutes, 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://hugh.comp.nus.edu.sg/DOTA/lab3/wpa.cap`.

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

```
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 boxes 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
```

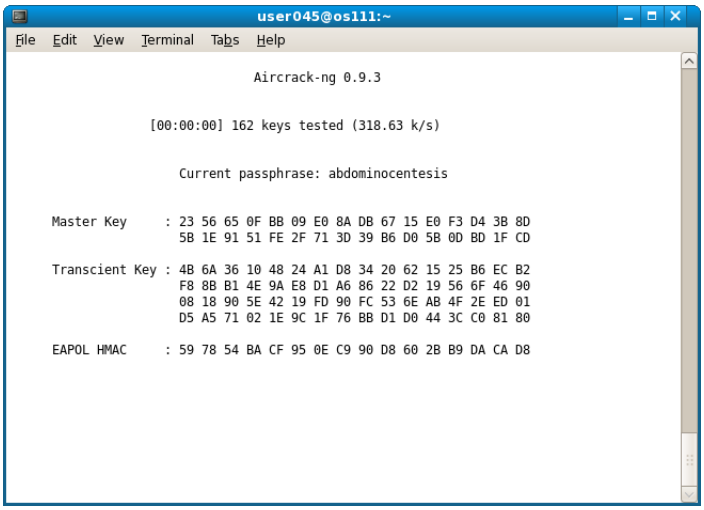


Figure 4: Aircrack-ng on Linux

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/lab3/gradeslab3-3.php`

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

4 Lab 3, part 4: Saving the world from catastrophe

Recently at NUS, during renovations at COM1, a secret hatch was discovered, directly under our laboratory, and when it was eventually opened, it was revealed to be a research station built by the Dharma Initiative, a scientific research project that involved conducting experiments in Singapore decades earlier. A man named Desmond Hume had been living in the station for three years, entering a secret code into a computer every two seconds to prevent a catastrophic event from occurring.

The CS people at NUS do not want the catastrophic event to happen, but Desmond is bat-crazy, and they cannot trust him to stay down there, typing in the four hex codes. The CS people have re-purposed an *American X10 (RF) home automation system* to enter the code automatically. The code is generated in the computer room (where all codes should be generated), and sent via the American home automation system to the computer behind the hatch, under our laboratory. It would be a good thing to discover this code, so that if a similar research station built by the Dharma Initiative is found back in your home city, you will be able to prevent catastrophic events from happening.

Your goal then, is to discover the four secret hex codes, from the transmission.

In this laboratory, you will be given an SDR. You can plug it into the USB port, and use software to monitor the signals that are in the laboratory. Run gqrx, and see what you can see. Look for a repetitive signal at a frequency between (say) 300 and 320 MHz. Do you find such a signal? What frequency is it?

Once you find the frequency of the repetitive signal, search for ways to listen and decode such signals using an SDR. I will not tell you the name of an application to listen to these signals, but one such application is already installed on the Linux machine, and is commonly used with RTL (Realtek RTL2832 dongles), particularly at 433 MHz (but they will also work on other frequencies). The command begins rtl_.... (Note that you will have to specify the frequency, and the particular protocol - X10 - in your command).

Use the program to analyze any X10-RF signals at the correct frequency.

Access the DOTA grading website:

`https://hugh.comp.nus.edu.sg/DOTA/lab3/gradeslab3-4.php`

You can give yourself whatever mark you believe you will in the future deserve.