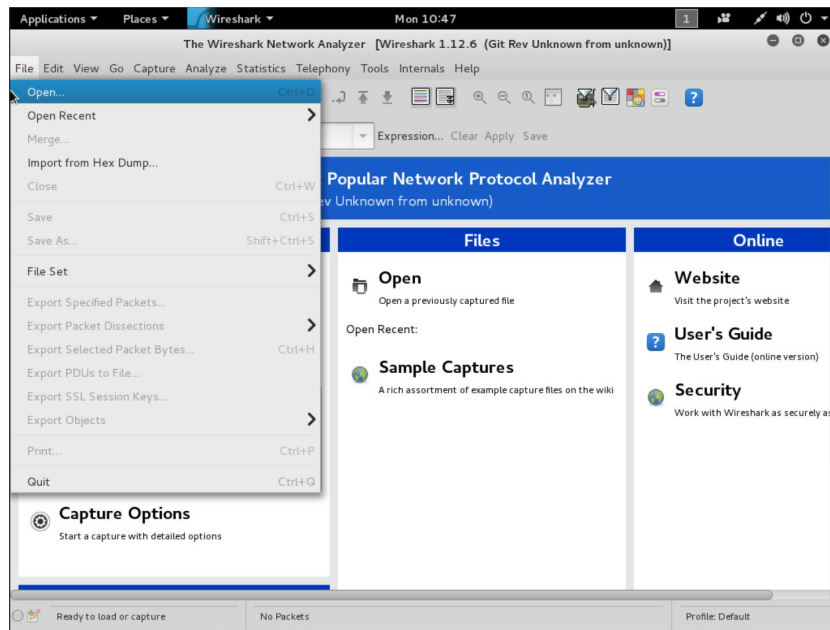
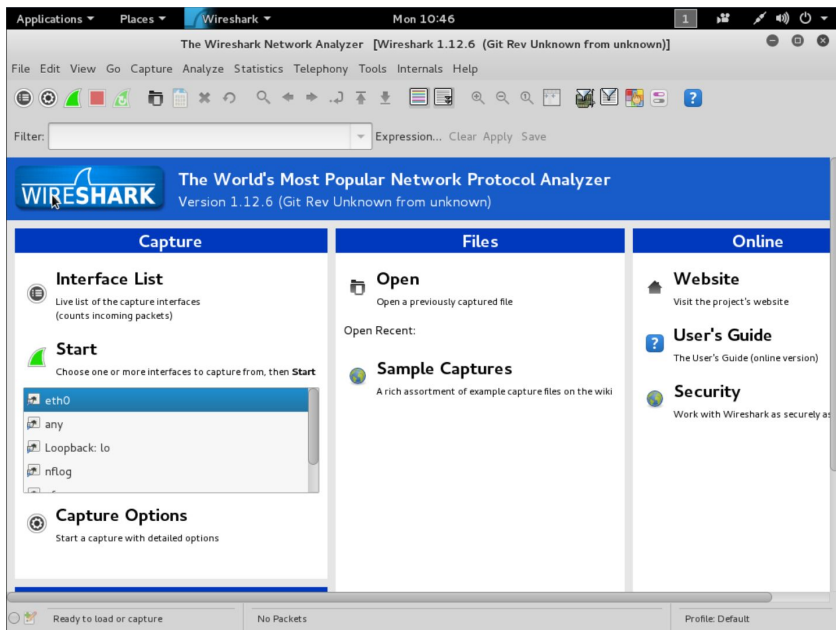


Project 3 Introduction

- Part 1: Network traces (1-2 hours)
- Part 2: Anomaly Detection (1-3 hours)
- Part 3: Penetration Testing (2-4 hours, save time for this one)

Part 1: Exploring Network Traces

You will need to use Wireshark to open and analyze the packet trace.



Part 1: Exploring Network Traces (continued)

Can filter packets by

- protocol (http, tls, etc.)
- ip address

Packet List

Protocol Breakdown

Bytes

The screenshot shows the Wireshark interface with a network trace loaded. The top menu bar includes 'Applications', 'Places', 'Wireshark', and 'Mon 10:51'. The title bar reads 'proj2-1.pcap [Wireshark 1.12.6 (Git Rev Unknown from unknown)]'. The main window contains a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, Help) and a toolbar with various icons. Below the toolbar is a 'Filter:' field with a dropdown arrow and the text 'Expression... Clear Apply Save'. The main area is divided into three panes: a 'Packet List' table, a 'Protocol Breakdown' pane, and a 'Bytes' pane. The 'Packet List' pane shows a table of network packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 4510) is highlighted in blue. The 'Protocol Breakdown' pane shows the hierarchical structure of the selected packet, with 'Hypertext Transfer Protocol' expanded to show the raw HTTP request. The 'Bytes' pane shows the raw hexadecimal and ASCII data of the selected packet. At the bottom, the status bar displays 'File: /root/proj2-1.pcap 15 MB 00:0...', 'Packets: 22650 · Displayed: 22650 (100.0%) · Load time: 0:00.192', and 'Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
4508	40.162955	74.125.225.212	10.0.2.3	HTTP	74	HTTP/1.1 200 OK (application/json)
4509	40.166902	10.0.2.3	74.125.225.212	TCP	54	55573->80 [ACK] Seq=13151 Ack=9859 Win=64768
4510	40.278720	10.0.2.3	74.125.225.212	HTTP	1076	GET /s?hl=en&sugexp=les%3B&gs_nf=1&gs_mss=is%2020my%20wifi%20se&pp=is%20my%20wifi%20secure&cp=12&gs_id=3p&xhr=t&
4511	40.331945	74.125.225.212	10.0.2.3	TCP	895	[TCP segment of a reassembled PDU]
4512	40.332416	74.125.225.212	10.0.2.3	HTTP	74	HTTP/1.1 200 OK (application/json)
4513	40.338180	10.0.2.3	74.125.225.212	TCP	54	55573->80 [ACK] Seq=14173 Ack=10720 Win=65536
4514	40.388241	10.0.2.3	74.125.225.212	HTTP	1077	GET /s?hl=en&sugexp=les%3B&gs_nf=1&gs_mss=is%2020my%20wifi%20se&pp=is%20my%20wifi%20secure&cp=12&gs_id=3p&xhr=t&
4515	40.448876	74.125.225.212	10.0.2.3	TCP	897	[TCP segment of a reassembled PDU]
4516	40.449848	74.125.225.212	10.0.2.3	HTTP	74	HTTP/1.1 200 OK (application/json)

▶ Frame 4510: 1076 bytes on wire (8608 bits), 1076 bytes captured (8608 bits)
▶ Ethernet II, Src: IntelCor_50:f0:a6 (8c:a9:82:50:f0:a6), Dst: Apple_e5:66:07 (00:26:08:e5:66:07)
▶ Internet Protocol Version 4, Src: 10.0.2.3 (10.0.2.3), Dst: 74.125.225.212 (74.125.225.212)
▶ Transmission Control Protocol, Src Port: 55573 (55573), Dst Port: 80 (80), Seq: 13151, Ack: 9859, Len: 1022
▶ Hypertext Transfer Protocol
▶ [truncated]GET /s?hl=en&sugexp=les%3B&gs_nf=1&gs_mss=is%2020my%20wifi%20se&pp=is%20my%20wifi%20secure&cp=12&gs_id=3p&xhr=t&
Host: www.google.com\r\n
Connection: keep-alive\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.79 Safari/537.4\r\n
Accept: */*\r\n
Referer: http://www.google.com/search?q=why+is+this+so+complicated&rlz=1C1CHKZ_enUS430US430&oq=why+is+this+so+complicated&su
Accept-Encoding: gzip,deflate\r\n

0010 04 26 18 3a 40 00 80 06 a6 43 0a 00 02 03 4a 7d .&.:@... .C....J}
0020 e1 d4 d9 15 00 50 01 a9 e4 24 f9 b3 ec 67 50 18P..\$...gP.
0030 00 fd fd 44 00 00 47 45 54 20 2f 73 3f 68 6c 3d ...D..GE T /s?hl=
0040 65 6e 26 73 75 67 65 78 70 3d 6c 65 73 25 33 42 en&sugexp=les%3B
0050 26 67 73 5f 6e 66 3d 31 26 67 73 5f 6d 73 73 3d &gs_nf=1 &gs_mss=
0060 69 73 25 32 30 6d 79 25 32 30 77 69 66 69 25 32 is%20my% 20wifi%2
0070 20 73 65 26 70 71 24 60 73 26 20 20 6d 70 25 32 On&pp=is%20my%2

File: /root/proj2-1.pcap 15 MB 00:0... Packets: 22650 · Displayed: 22650 (100.0%) · Load time: 0:00.192 Profile: Default

Part 2: Anomaly Detection

- SYN, SYN+ACK packets
 - SYN is the client-side initial handshake
 - SYN+ACK is server-side acknowledgement of the handshake
- Port scanning
 - Attackers may send SYN packets to identify active network hosts listening to a specified port
 - You will need to find sources sending more SYN packets than receiving SYN+ACK packets
- nmap (this command will be useful for Part 3)
 - Very (most?) popular network scanner (sudo apt-get install nmap)
 - Command “nmap 192.168.0.0/24” will scan top 1000 most-used ports on your local network
 - Be CAREFUL! Port scanning sends traffic out on the network that may interfere with other processes. Don't port scan on networks you don't own or have authorization for.

Part 3: Penetration Test

What's a penetration test?

- A fun project where you get paid to hack someone!
- A formal agreement between you and a company defining the scope of work to be performed, the rules of engagement to be followed, and giving you official authorization to begin.

Why is it important to have a Pen-test Pre-Agreement?

- To protect yourself!
- It gives you explicit authorization to conduct the pen test. Without it, you are subject to criminal penalties.

Part 3: Penetration Test (continued)

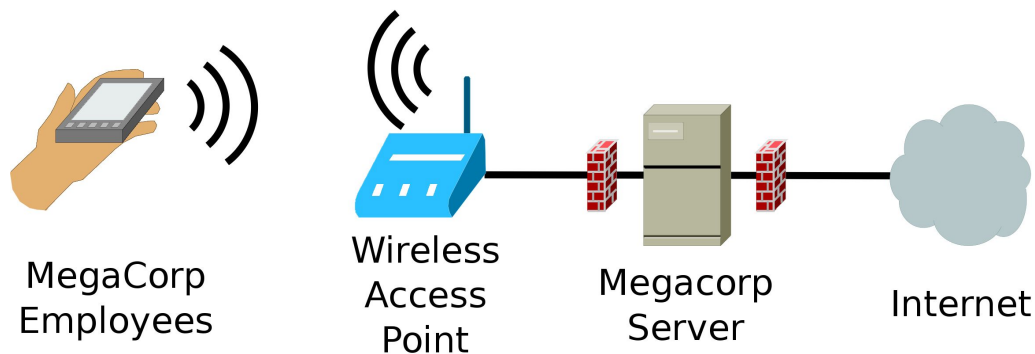
You must read and agree to
the Pen-Test Pre-Agreement!

Make sure you send us the proper email accepting the conditions of the Pen-Test **before** you begin.

(Both partners must email us.)

Pen Test: What are you trying to do?

- MegaCorp needs its systems tested to determine if they're vulnerable.
- MegaCorp recently set up a wireless network in BBB for its employees to use.
 - This would be a good starting place to see if there are any vulnerabilities.
- MegaCorp also has a server that uses Kerberos passwords to log on.
- Intentionally, this project is left somewhat open-ended.
 - You never know what you're going to find on a PenTest.



Pen Test: What are you trying to do? (continued)

- You are looking for at least three notable findings.
 - Hostnames of any machines you gain access to.
 - Encryption keys for networks you gain access to.
 - Username/Passwords you are able to obtain (not including your own).
- Rules of Engagement
 - Spelled out in detail in the project document.
 - They are *really* important. Make sure you stay within the allowed scope of the Pen Test.
 - If you have any questions, please ask us before you try something that isn't specifically allowed.

Pen Test: Getting set up

You will need a variety of hacking tools for this project. The easiest way to get set up is to use Kali Linux, widely known as the Pen-Testing Linux Distro.

Install VirtualBox: <https://www.virtualbox.org/wiki/Downloads>

On Linux, use your package manager (sudo apt-get install virtualbox)

Make sure to download and install the Extension Pack! (even Linux!)

Download the Kali Virtualbox image:

<https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/>

Import the image into VirtualBox. Start up your new virtual machine.

The default username and password is “root” / “toor”



Pen Test: Wireless Networks

- Unlike wired networks, wireless packets are *broadcast*
- Only one device can talk at a time
- Devices use different “channels” (slightly different frequencies) to talk at the same time
- Encryption
 - No security
 - Lets you right in
 - WEP
 - Can be mathematically cracked in seconds
 - WPA (WPA2 is essentially the same, just uses AES instead of TKIP)
 - Current generation of personal wireless encryption. Can't easily be mathematically broken.
 - But, you can capture the 4-way handshake and guess passwords until it decrypts.

Pen Test: Wireless attacks

(borrow one of our adapters!)

- Aircrack-ng suite
 - Set of tools for breaking into wireless networks.
 - http://www.aircrack-ng.org/doku.php?id=cracking_wpa (Start at part 2)
 - <http://lewiscomputerhowto.blogspot.com/2014/06/how-to-hack-wpawpa2-wi-fi-with-kali.html>
 - “airmon-ng check kill” -- Kill all networking processes that may interfere with wireless attacks
 - airodump-ng
 - Sets up your wireless interface to capture packets (like wireshark or tcpdump)
 - aireplay-ng
 - Inject packets into the wireless network
 - Specifically, for a WPA attack, you will want to send the client and base station a “death” packet to force the client off the network. It should reconnect automatically.
 - aircrack-ng
 - Read through captured wireless packets and crack any passwords you find.
 - For WPA attacks, it will require a wordlist. If you know the format of the password, you can create a wordlist yourself.

Pen Test: Capturing network traffic without Wireshark

Use tcpdump! (It's actually easier/faster for small stuff)

```
tcpdump -i eth1
```

- Capture network traffic on interface *eth1*

```
tcpdump -nni eth1
```

- Capture network traffic without resolving domain names or ports (faster!)

```
tcpdump -nni eth1 -w capturefile.pcap
```

- Capture network traffic and write the raw packets to file *capturefile.pcap*

```
tcpdump -Anni eth1 -w capturefile.pcap
```

- Capture traffic and write out all printable characters in packets to terminal

```
tcpdump -Anni eth1 -w capturefile.pcap tcp port 22
```

- Capture only traffic using TCP protocol and on port 22 (SSH)