CSCI369 Ethical Hacking

Lecture 2-1: TCP/IP Basics & Capturing Traffic

A/Prof Joonsang Baek

School of Computing and Information Technology

UNIVERSITY OF WOLLONGONG AUSTRALIA

Overview of TCP/IP

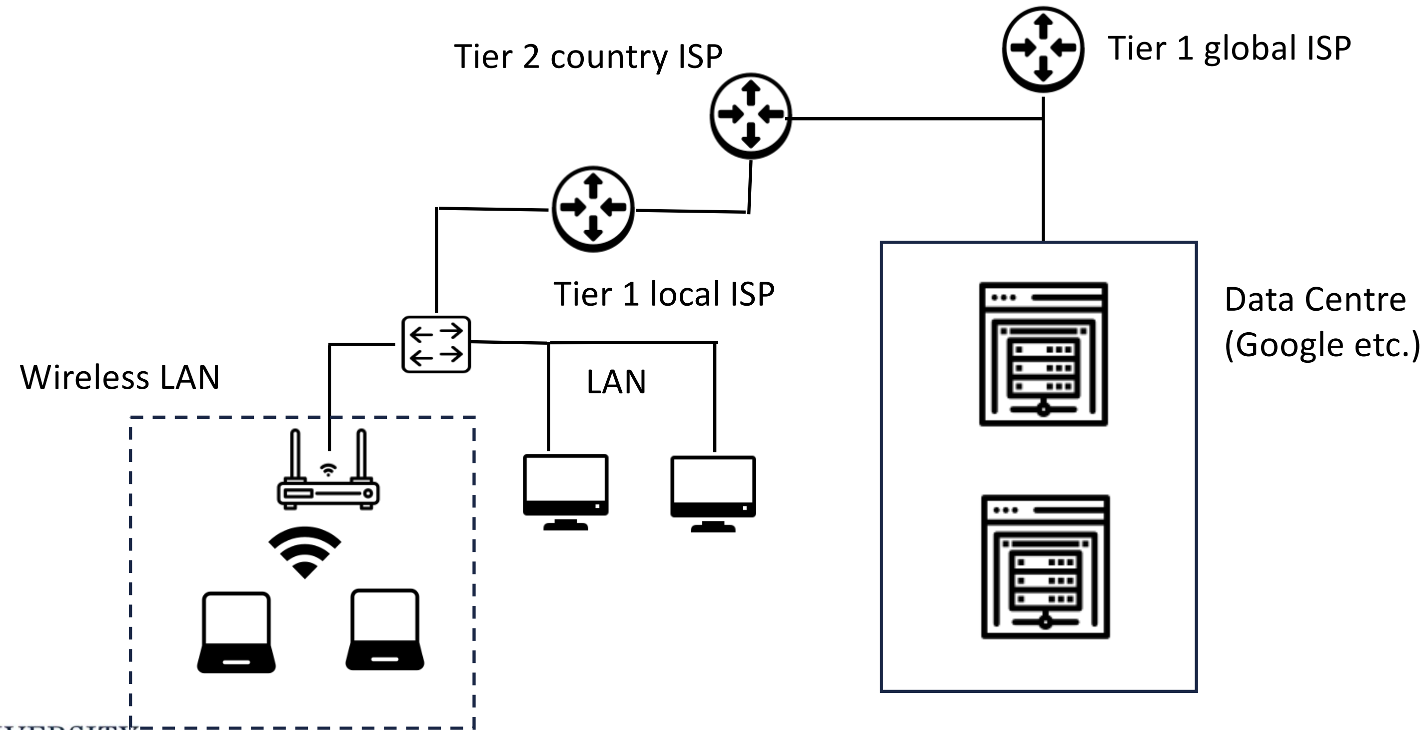
- The TCP/IP layer (TCP/IP stack)

OSI model				
Layer			Protocol data unit (PDU)	Function[5]
Host layers	7	Application	Data	High-level APIs, including resource sharing, remote file access
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1	Physical	Symbol	Transmission and reception of raw bit streams over a physical medium

(From https://en.wikipedia.org/wiki/OSI_model)

UNIVERSITY OF WOLLONGONG AUSTRALIA

Overview of Network Hierarchy



The Application Layer

- Some important applications
  - http: The primary protocol used to communicate over the web.
  - https: The secure version of http based on TLS (= SSL).
  - FTP: Allows different OS's to transfer files between one another. (Should not use anonymous username with no password.)
  - SMTP (Simple Mail Transfer Protocol) : Transmits email messages across the Internet.

The Application Layer

- Some important applications
  - SSH: Enables users to *securely* log on to a remote server and issue commands interactively.
  - Telnet: Enables users to *insecurely* log on to a remote server and issue commands interactively. (Username and password are not encrypted. Do not use!)

The Transport Layer

- Main function
  - Data is encapsulated into segments
  - Segments can be TCP or UDP
- TCP
  - Connection-oriented protocol: The source (sender) cannot send any data to the destination (receiver) until the destination node acknowledges that it's listening to the source.
    - ✓The source sends a **`SYN`** packet to the destination
    - ✓The destination sends **`SYN-ACK`** to the source
    - ✓The source sends **`ACK`** to the destination

Three-way handshake

UNIVERSITY OF WOLLONGONG AUSTRALIA

The Transport Layer

- TCP Segment Header

Bits

<table>
<tr><td colspan="7">0 15</td><td>16 31</td></tr>
<tr><td colspan="7">Source Port</td><td>Destination Port</td></tr>
<tr><td colspan="8">Sequence Number</td></tr>
<tr><td colspan="8">Acknowledgement Number</td></tr>
<tr><td>Offset Reserved</td><td>U</td><td>A</td><td>P</td><td>R</td><td>S</td><td>F</td><td>Window</td></tr>
<tr><td colspan="7">Checksum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="8">Options and Padding</td></tr>
</table>

TCP Flags

TCP Header Diagram

UNIVERSITY OF WOLLONGONG AUSTRALIA

The Transport Layer

- TCP Flags
  - ➢Can be set 0 (off) or 1 (on)
    - ✓SYN flag – This signifies the beginning of a session
    - ✓ACK flag – This acknowledges a connection request
    - ✓PSH flag – This flag is used to deliver data directly to an application (Data is not buffered; it is sent immediately)
    - ✓URG flag – This flag is used to signify urgent data
    - ✓RST flag – This resets or drops a connection
    - ✓FIN flag – The indicates that the connection is finished

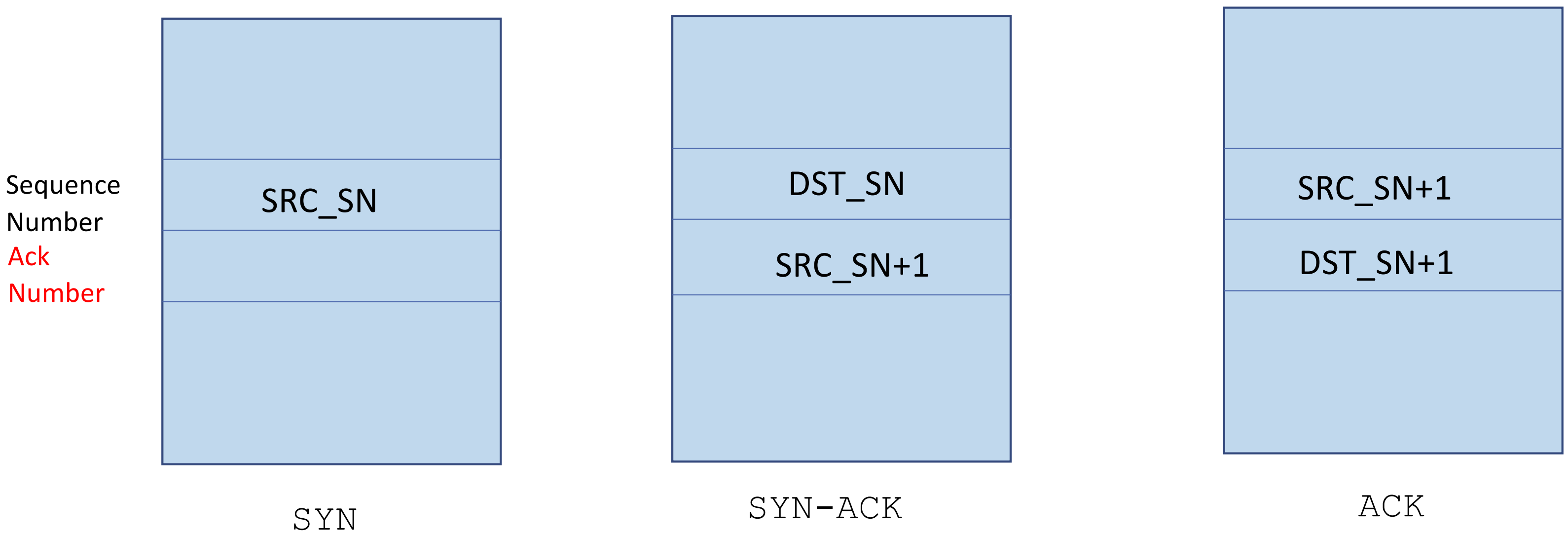
UNIVERSITY OF WOLLONGONG AUSTRALIA

The Transport Layer

- Three-way handshake
    1. A sequence number from the source node (SRC_SN) is put in the Sequence Number field (32 bits) of a `SYN` segment and is sent to the destination node.
    2. The destination node extracts the SRC_SN from the `SYN` segment and puts SRC_SN+1 in the Acknowledgement Number field; also the destination node selects a DST_SN and puts it in the Sequence Number field of a `SYN−ACK` segment, which is sent to the source node.
    3. The source node extracts the DST_SN from the `SYN−ACK` segment and puts DST_SN+1 in the Acknowledgement Number field; also the destination node puts SRC_SN+1 in the sequence number field of a `ACK` segment, which is sent to the destination node.

The Transport Layer

- Illustration - Three-way handshake with SN



UNIVERSITY OF WOLLONGONG AUSTRALIA

The Transport Layer

- TCP Ports
  - 16 bits for both source and destination ports
  - A port is a logical component of TCP connection assigned to a process requiring network connectivity.
  - A port is the way a client program specifies a specific server program.
  - **Some important ports (to remember)**
    - ✓ 21: FTP
    - ✓ 22: SSH
    - ✓ 25: SMTP
    - ✓ 53: DNS
    - ✓ 80: HTTP
    - ✓ 443: HTTPS (Secure HTTP)
    - ✓ 110: POP3 (Post Office Protocol 3)

The Transport Layer

- UDP
  - ➢UDP does not need to verify whether the destination is listening or ready to accept the packets
    - ✓It has no handshaking dialogues
    - ✓No guarantee of delivery, ordering and/or duplicate protection
  - ➢Connectionless transmission
  - ➢Unreliable but fast
    - ✓Used in applications in which queries must be fast and only consist of a single request such as DNS and DHCP

Frequently-Used Port Numbers

PROTOCOL NAME	PORT #	PROTOCOL NAME	PORT #
FTP	TCP 21	LDAP over SSL	TCP 636
SSH/SCP	TCP 22	FTP over SSL	TCP 989–990
Telnet	TCP 23	IMAP over SSL	TCP 993
SMTP	TCP 25	POP3 over SSL	TCP 995
DNS Query	UDP 53	MS-SQL	TCP 1433
DNS Zone Transfer	TCP 53	NFS	TCP 2049
DHCP	UDP 67 UDP 68	Docker Daemon	TCP 2375
TFTP	UDP 69	Oracle DB	TCP 2483–2484
HTTP	TCP 80	MySQL	TCP 3306
Kerberos	UDP 88	RDP	TCP 3389
POP3	TCP 110	VNC	TCP 5500
SNMP	UDP 161 UDP 162	PCAnywhere	TCP 5631
NetBIOS	TCP/UDP 137 TCP/UDP 138 TCP/UDP 139	IRC	TCP 6665–6669
IMAP	TCP 143	IRC SSL	TCP 6679 TCP 6697
LDAP	TCP 389	BitTorrent	TCP 6881–6999
HTTPS (TLS)	TCP 443	Printers	TCP 9100
SMTP over SSL	TCP 465	WebDAV	TCP 9800
rlogin	TCP 513	Webmin	10000

UNIVERSITY OF WOLLONGONG AUSTRALIA

The Internet Layer

- Main functionality
  - ➢Responsible for routing a packet (datagram) to a destination address
  - ➢Routing is done by using an IP address
  - ➢Connectionless transmission like UDP

IP Address Basics (IPv4)

- IP address: Four integers separated by decimals; each integer represents 1 byte = 8 bits = 1 *octet*
  - ➢Ex) 128.214.18.16 = 1000000.11010110.00010010.00010000 (Binary)
- Each IP address is separated into a network address and a host address → One part of the IP address represents a network prefix and the other part represents a host
  - ✓Ex) In the classful IP addressing, 192.168.1.231 belongs to Class C address, which uses the first three numbers to identify the network and the last number to identify the host

Subnet mask

- A bitmask that yields the network prefix (network address) when applied by a bitwise AND operation with any IP address in the network
- Example
  - ➢Network prefix of an IP address 192.168.54.3 with a subnet mask 255.255.255.0 is 192.168.54
  - ➢An IP address 192.168.54.3 with a subnet mask 255.255.0.0 produces a network prefix 192.168 → This means that to be on the same network, two machines must have IP addresses starting with 192.168
  - ➢Hence,
    - ✓Subnet mask for Class A: 255.0.0.0
    - ✓Subnet mask for Class B: 255.255.0.0
    - ✓Subnet mask for Class C: 255.255.255.0

UNIVERSITY OF WOLLONGONG AUSTRALIA

CIDR IP Addressing

- **CIDR (Classless Inter-Domain Routing)** addressing is a compact method for representing an IP address and its associated network prefix.
- The CIDR notation is constructed from an IP address, a slash ('/') character, and a decimal number. The trailing number is the count of leading 1 bits in the subnet mask.
  - Form: a.b.c.d/x
- Motivation: The classful network does not fully represent more fine-grained network prefixes. CIDR resolves this issue.

CIDR IP Addressing

CIDR	Subnet mask (decimal)	Subnet mask (binary)	Available addresses	
/0	0.0.0.0	00000000.00000000.00000000.00000000	4.294.967.296	2^{32}
/1	128.0.0.0	10000000.00000000.00000000.00000000	2.147.483.648	2^{31}
/2	192.0.0.0	11000000.00000000.00000000.00000000	1.073.741.824	2^{30}
/3	224.0.0.0	11100000.00000000.00000000.00000000	536.870.912	2^{29}
/4	240.0.0.0	11110000.00000000.00000000.00000000	268.435.456	2^{28}
/5	248.0.0.0	11111000.00000000.00000000.00000000	134.217.728	2^{27}
/6	252.0.0.0	11111100.00000000.00000000.00000000	67.108.864	2^{26}
/7	254.0.0.0	11111110.00000000.00000000.00000000	33.554.432	2^{25}
/8	255.0.0.0	11111111.00000000.00000000.00000000	16.777.216	2^{24}
/9	255.128.0.0	11111111.10000000.00000000.00000000	8.388.608	2^{23}
/10	255.192.0.0	11111111.11000000.00000000.00000000	4.194.304	2^{22}
/11	255.224.0.0	11111111.11100000.00000000.00000000	2.097.152	2^{21}
/12	255.240.0.0	11111111.11110000.00000000.00000000	1.048.576	2^{20}
/13	255.248.0.0	11111111.11111000.00000000.00000000	524.288	2^{19}
/14	255.252.0.0	11111111.11111100.00000000.00000000	262.144	2^{18}
/15	255.254.0.0	11111111.11111110.00000000.00000000	131.072	2^{17}
/16	255.255.0.0	11111111.11111111.00000000.00000000	65.536	2^{16}
/17	255.255.128.0	11111111.11111111.10000000.00000000	32.768	2^{15}
/18	255.255.192.0	11111111.11111111.11000000.00000000	16.384	2^{14}
/19	255.255.224.0	11111111.11111111.11100000.00000000	8.192	2^{13}
/20	255.255.240.0	11111111.11111111.11110000.00000000	4.096	2^{12}
/21	255.255.248.0	11111111.11111111.11111000.00000000	2.048	2^{11}
/22	255.255.252.0	11111111.11111111.11111100.00000000	1.024	2^{10}
/23	255.255.254.0	11111111.11111111.11111110.00000000	512	2^{9}
/24	255.255.255.0	11111111.11111111.11111111.00000000	256	2^{8}
/25	255.255.255.128	11111111.11111111.11111111.10000000	128	2^{7}
/26	255.255.255.192	11111111.11111111.11111111.11000000	64	2^{6}
/27	255.255.255.224	11111111.11111111.11111111.11100000	32	2^{5}
/28	255.255.255.240	11111111.11111111.11111111.11110000	16	2^{4}
/29	255.255.255.248	11111111.11111111.11111111.11111000	8	2^{3}
/30	255.255.255.252	11111111.11111111.11111111.11111100	4	2^{2}
/31	255.255.255.254	11111111.11111111.11111111.11111110	2	2^{1}
/32	255.255.255.255	11111111.11111111.11111111.11111111	1	2^{0}

Note that /8, /16, /24 represent Class A, Class B and Class C, respectively.

UNIVERSITY OF WOLLONGONG AUSTRALIA

CIDR IP Addressing

- Example 1) What IP ranges does 192.168.101.3/22 represent?
  - ➢First, calculate the network prefix by applying “AND (Λ)" to bit representations of IP address (192.168.101.3) and subnet mask (/22).

IP address	192	168	101	3
	11000000	10101000	01100101	00000011
/22	11111111	11111111	11111100	00000000
Network Prefix	**11000000**	**10101000**	**01100100**	**00000000**
	192	168	100	0

NB: Binary calculator https://www.calculator.net/binary-calculator.html

UNIVERSITY OF WOLLONGONG AUSTRALIA

CIDR IP Addressing

- Then, calculate how many addresses are available:

  It is $2^{32-22}=2^{10}=1024$.

  - ✓Note that 1024/256=4.

- Therefore, it represents IPs from 192.168.100.0 to 192.168.100.255, from 192.168.101.0 to 192.168.101.255, from 192.168.102.0 to 192.168.102.255 and from 192.168.103.0 to 192.168.103.255.

- Therefore, the range is 192.168.100.0 to 192.168.103.255.

UNIVERSITY OF WOLLONGONG AUSTRALIA

CIDR IP Addressing

- Example 2) Do CIDR notations 10.10.1.45/27 and 10.10.1.61/27 have the same network prefix? How about 10.10.1.65/27?
  - First, calculate the network prefix of 10.0.1.45/27.

IP address	10	10	1	45
	00001010	00001010	00000001	00101101
/27	11111111	11111111	11111111	11100000
Network Prefix	**00001010**	**00001010**	**00000001**	**00100000**
	10	10	1	32

CIDR IP Addressing

- Then, calculate the network prefix for 10.10.1.61.

IP address	10	10	1	61
	00001010	00001010	00000001	00111101
/27	11111111	11111111	11111111	11100000
Network Prefix	**00001010**	**00001010**	**00000001**	**00100000**
	10	10	1	32

- This implies that hosts ending with .45 and .61 with belong to the same network.

CIDR IP Addressing

- Now, find out the network prefix for 10.10.1.65.

IP address	10	10	1	65
	00001010	00001010	00000001	01000001
/27	11111111	11111111	11111111	11100000
Network Prefix	**00001010**	**00001010**	**00000001**	**01000000**
	10	10	1	64

- Hence, 10.10.1.65/27 does not belong to the same network as the two previous ones.

The Number of Host IPs

- We do not use IP address that represents network (network prefix) and the last address in the IP range for a host IP address.
  - The last address is reserved for broadcast address.
- For example, in Example 1) 192.168.100.0 is not used and 192.168.103.255 is reserved as a broadcast address.
- Therefore, the number of **actual IP addresses available for hosts** in Example 1) is $2^{10}-2=1022$.

IPv6 Addressing

- Developed to increase the space of IP address (The size of IPv4 address space = 2^{32}.)
- IPv6 uses 16 bytes address: The size of IPv6 address space = 2^{128}
- Example:
  - ✓ 1111:0cb7:75a2:0110:1234:3a2e:1113:7777

    1111 → 0001 0001 0001 0001 (2 bytes)
    0cb7 → 0000 1100 1011 0111
    75a2 → 0111 0101 1010 0010
    0110 → 0000 0001 0001 0000
    1234 → 0001 0010 0011 0100
    3a2e → 0011 1010 0010 1110
    1113 → 0001 0001 0001 0011
    7777 → 0111 0111 0111 0111

128-bit representation of IPv6 address

UNIVERSITY OF WOLLONGONG AUSTRALIA

ICMP (Internet Control Message Protocol)

- It is used by network devices, including routers, to send error messages and operational information such as “Requested service is not available” or “Destination network unreachable”
- It is used in the `ping` tool
  - A source sends ICMP ECHO_REQUEST to a destination system
  - If the system is live, it will respond by sending ICMP ECHO_REPLY
- It is also used in the `traceroute` tool
  - ✓ In UNIX-like OS, `traceroute` has an option (`-I`) to use ICMP (By default, UDP is used.)
  - ✓ In Windows OS, `tracert` uses ICMP by default. (No need to use the `–I` option.)

Introduction to Wireshark

- Wireshark
  - ➢A well-known network analysis tool previously known as Ethereal.
  - ➢It captures packets in real time and displays them in human-readable format.
  - ➢Main features include filters and color coding to analyse network traffic and inspect individual packets.

Capturing Traffic

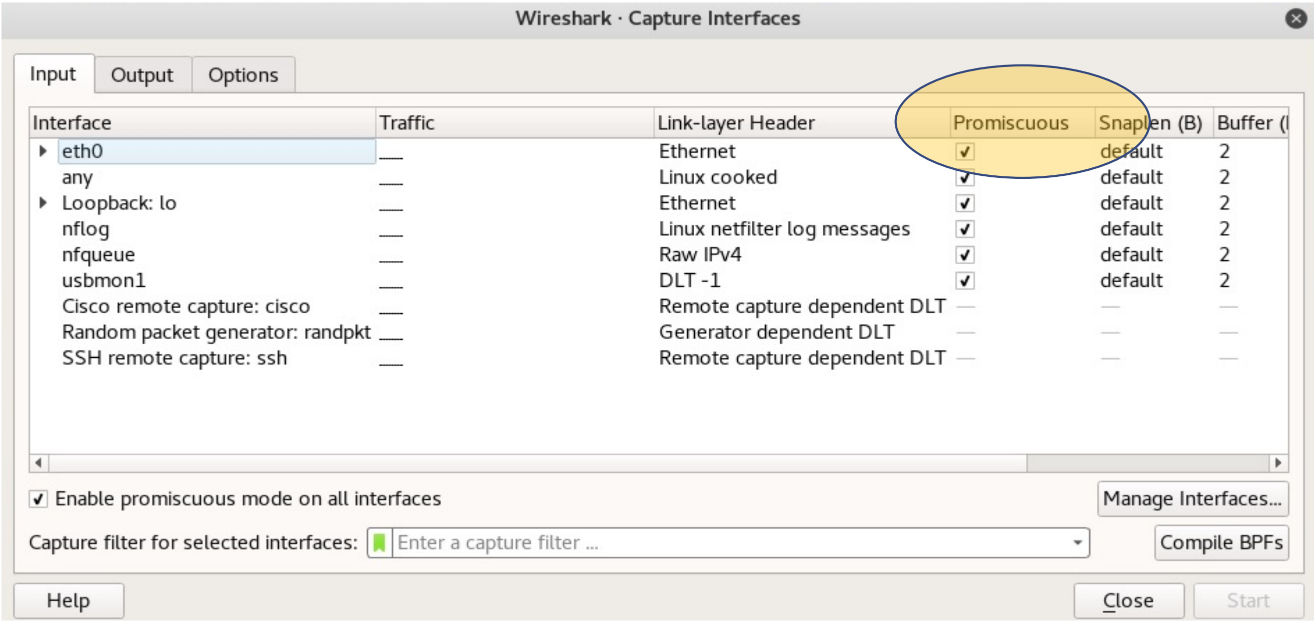
- Selecting interfaces
  - ➢After launching Wireshark, a user can select a network interface and start capturing packets on that interface.

ed without

Capturing Traffic

- Promiscuous mode
  - ➢Promiscuous mode is a mode for a wired network interface controller (NIC) or wireless network interface controller (WNIC) that causes the controller to pass all traffic it receives to the CPU.
  - ➢This mode is normally used for packet sniffing which may take place on a router or on a computer connected to a hub or a part of a WLAN.
  - ➢By default, Wireshark runs in promiscuous mode but can be updated in the Capture → Option panel

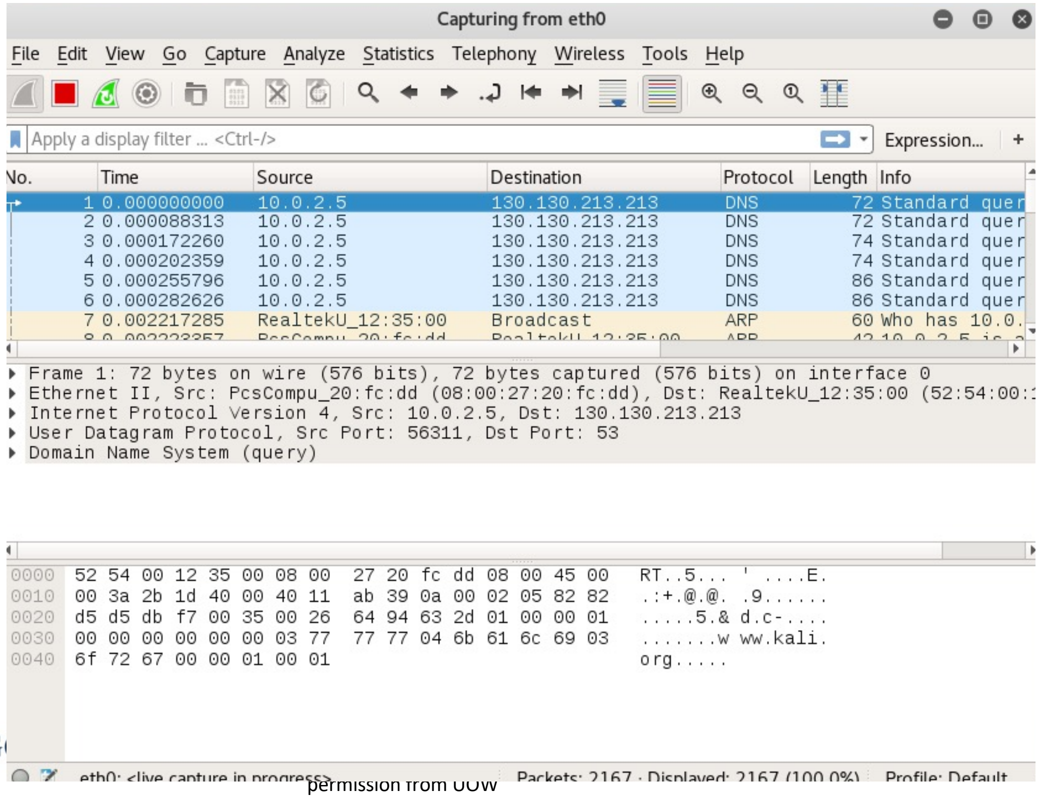
Capturing Traffic



Capturing Traffic

- Packet capturing: Upon selecting network interface, the packets start to appear in real time; Wireshark captures each packet sent to or from the system.
- In promiscuous mode, a user can see all the other packets on the network instead of only packets addressed to the user's network adapter.

Capturing Traffic



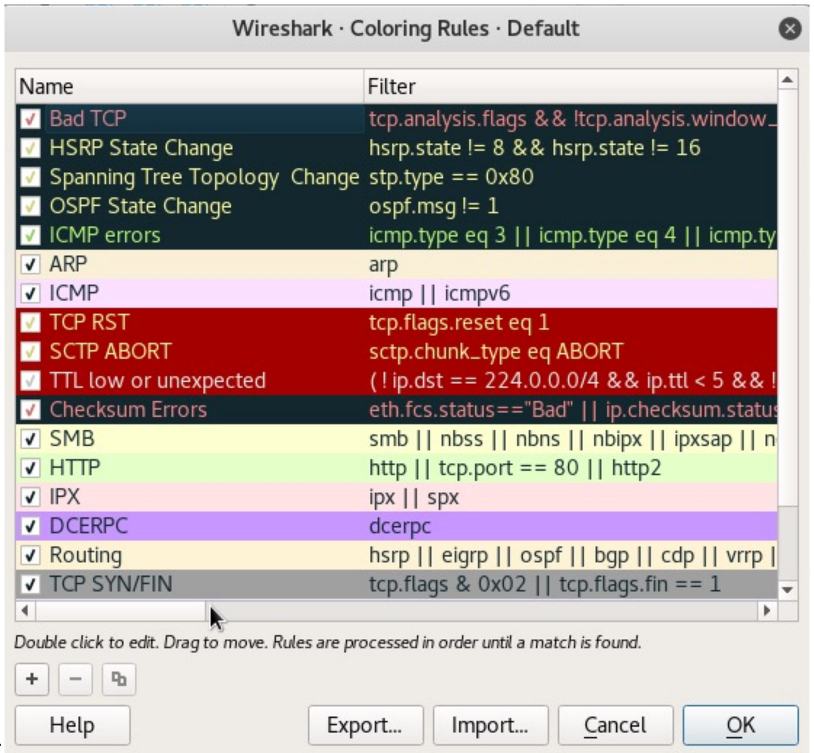
permission from UOW

Color Coding

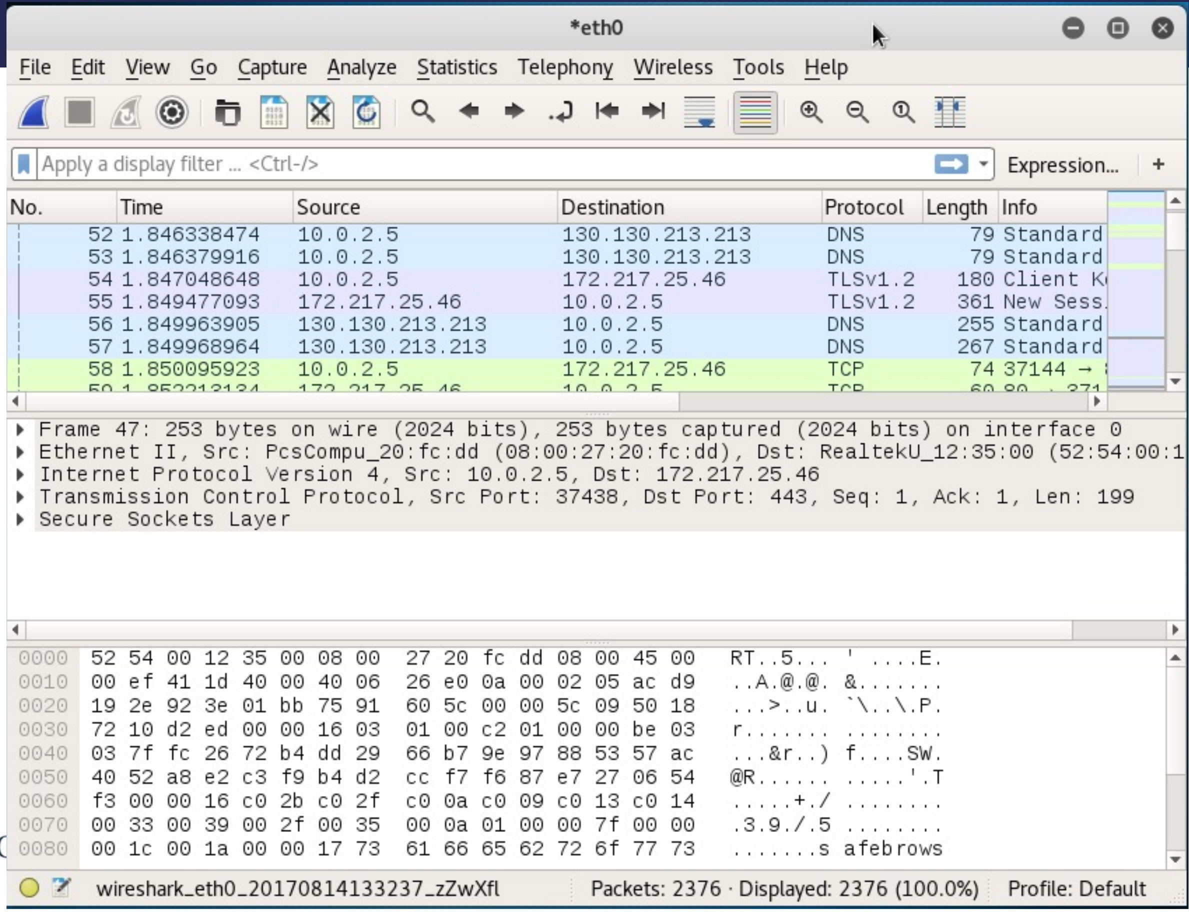
- Wireshark uses colors to identify the types of traffic at a glance.
- Examples (by default):
  - Light purple: TCP traffic
  - Light blue: UDP traffic
  - Black: Packets with errors
- Current (default) color setting can be seen on View → Coloring Rules (Modification is possible)

Color Coding

- Default color coding rules



Color Coding

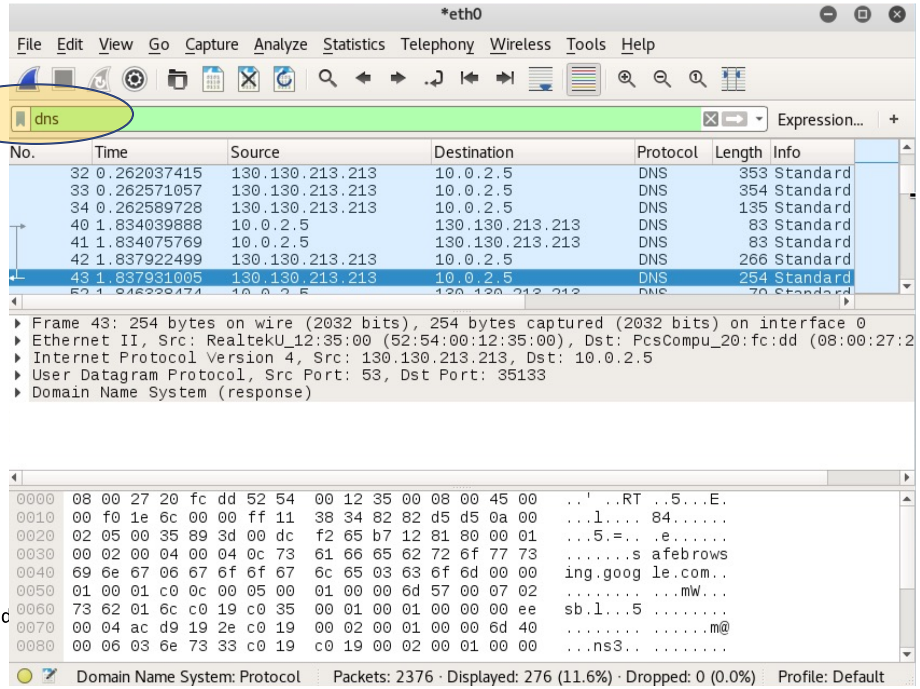


Packets captures by Wireshark

UNIVERSITY OF WOLLONGONG AUSTRALIA

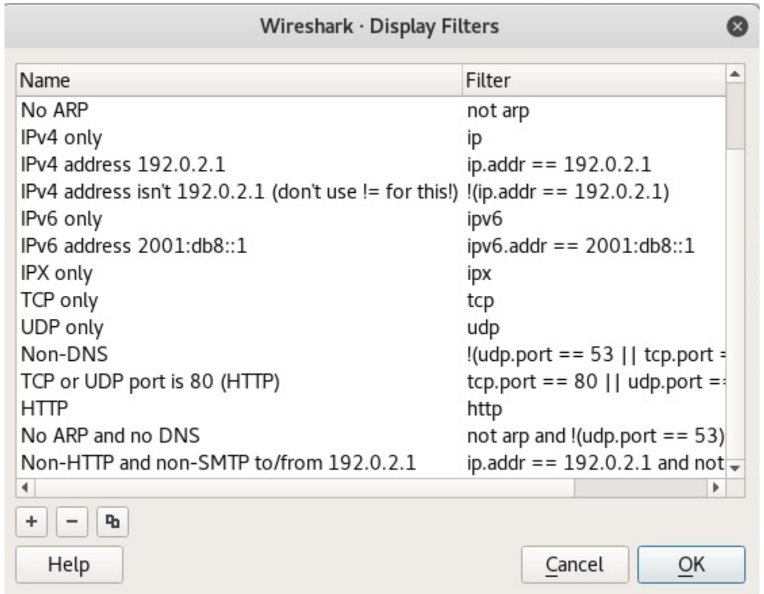
Filtering Packets

- Filter Box
  - ➢The most basic way to filter packets in Wireshark is to enter keywords in the Filter Box.



Filtering Packets

- Default filters
  - ➢Analyze → Display Filters will list default filters included in Wireshark.

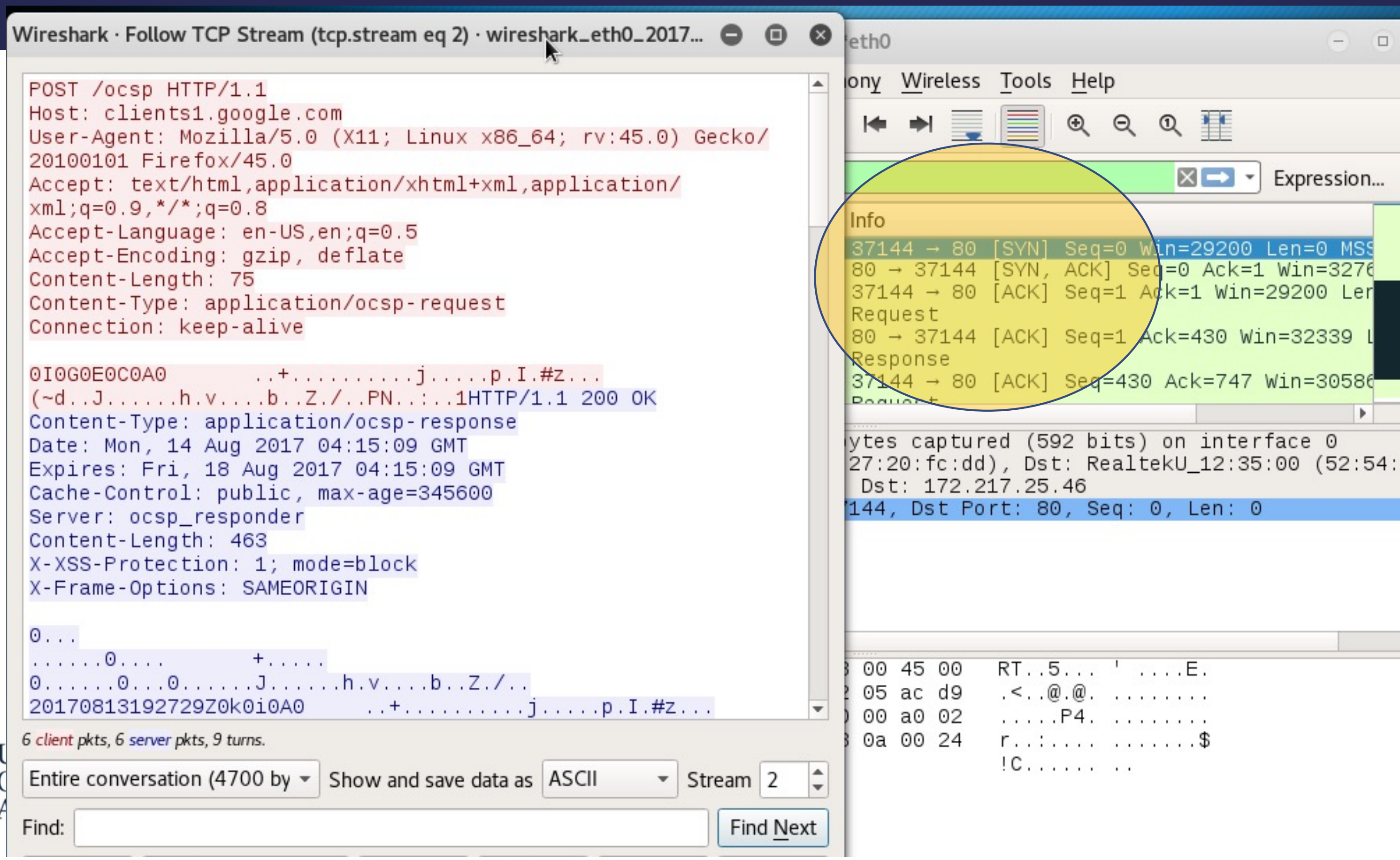

Wireshark · Display Filters

Name	Filter
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 address isn't 192.0.2.1 (don't use != for this!)	!(ip.addr == 192.0.2.1)
IPv6 only	ipv6
IPv6 address 2001:db8::1	ipv6.addr == 2001:db8::1
IPX only	ipx
TCP only	tcp
UDP only	udp
Non-DNS	!(udp.port == 53 || tcp.port =
TCP or UDP port is 80 (HTTP)	tcp.port == 80 || udp.port ==
HTTP	http
No ARP and no DNS	not arp and !(udp.port == 53)
Non-HTTP and non-SMTP to/from 192.0.2.1	ip.addr == 192.0.2.1 and not

Inspecting Packets Example (1)

- Another interesting thing you can do is right-click a packet and select Follow → TCP Stream.
  - ➢ You'll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.
  - ➢ Next slide (Screen shot)

Inspecting Packets Examples (1)

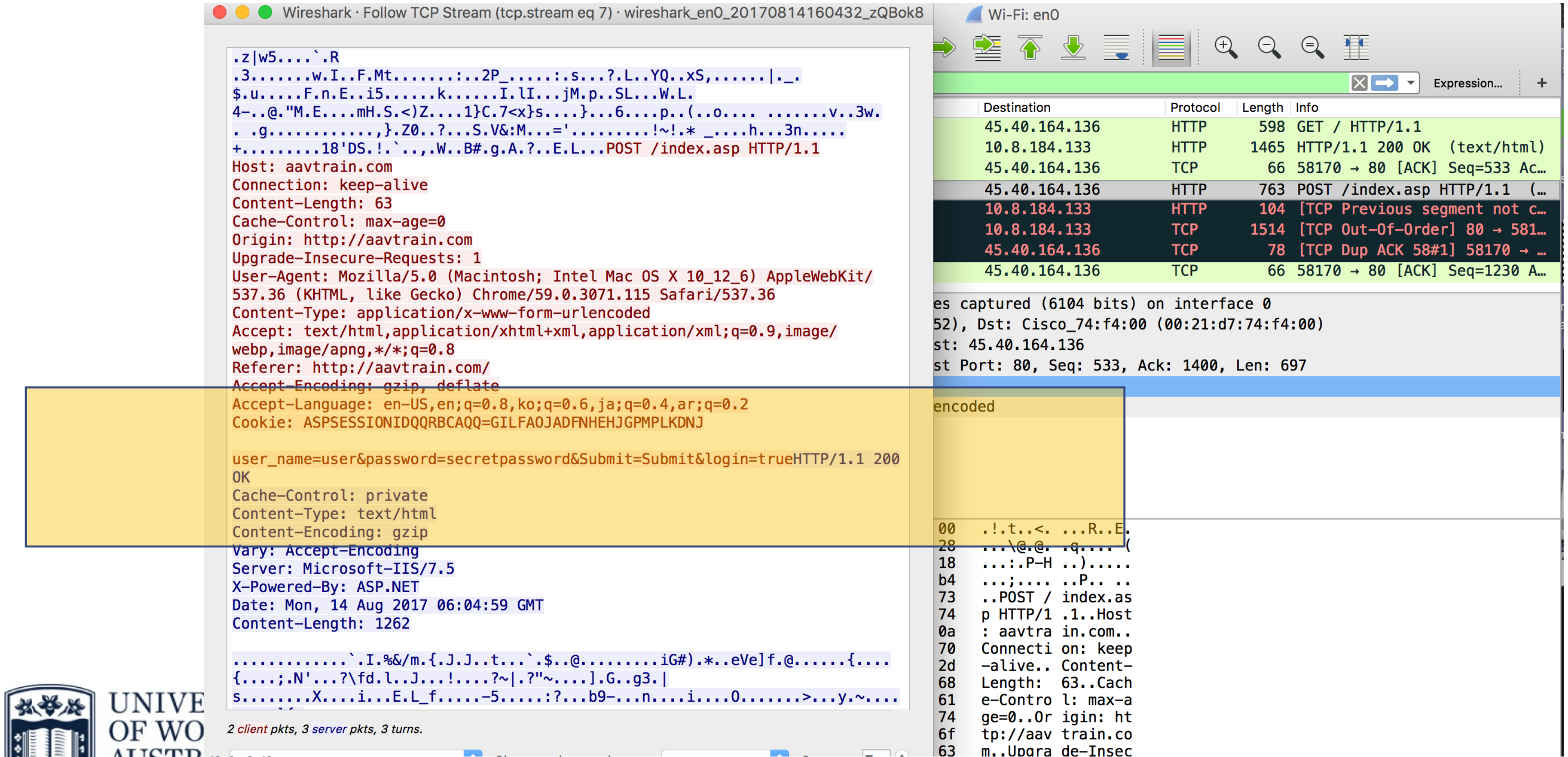


Inspecting Packets Example (2)

- In fact, Wireshark will enable us to capture a username and a password entered to *insecure* website
- A filter you need in this regard is
  - `http.request.method == "POST"`
  - Once you locate a packet right-click the packet and select Follow → TCP Stream
  - Example will follow in the next slide

UNIVERSITY OF WOLLONGONG AUSTRALIA

Inspecting Packets Example (2)



permission from UOW

Inspecting Packets Example (3)

- By using Wireshark, connections based on the secure application protocol like SSL can be analysed.

3141	36.398756	10.9.26.59	162.125.34.129	TLSv1.2	237 Client Hello
3186	36.556681	162.125.34.129	10.9.26.59	TLSv1.2	1514 Server Hello
3187	36.556682	162.125.34.129	10.9.26.59	TLSv1.2	1514 Certificate [TCP segment of a reassembled PDU]
3188	36.556684	162.125.34.129	10.9.26.59	TLSv1.2	156 Server Key Exchange, Server Hello Done
3190	36.561857	10.9.26.59	162.125.34.129	TLSv1.2	180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
3238	36.716422	162.125.34.129	10.9.26.59	TLSv1.2	296 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
3239	36.719132	10.9.26.59	162.125.34.129	TLSv1.2	473 Application Data

An example of connection based on SSL