Last updated 9/16/2016

Rich's lesson module checklist

- ❑ Slides and lab posted
- ❑ WB converted from PowerPoint
- ❑ Print out agenda slide and annotate page numbers

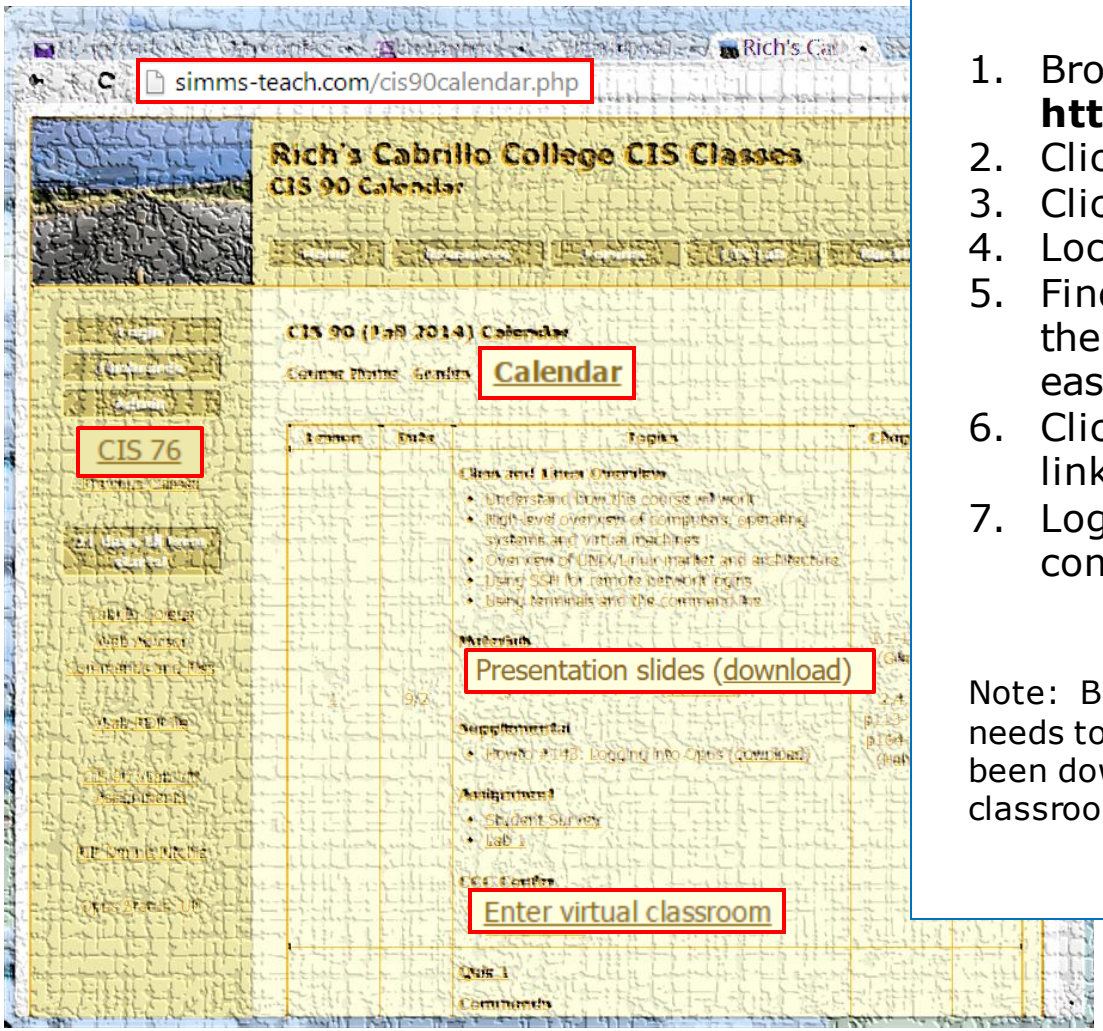
- ❑ Flash cards
- ❑ Properties
- ❑ Page numbers
- ❑ 1st minute quiz
- ❑ Web Calendar summary
- ❑ Web book pages
- ❑ Commands

- ❑ Lab 2 posted and tested
- ❑ Sample Lab 2 posted

- ❑ Rosters printed
- ❑ Add codes printed

- ❑ Backup slides, whiteboard slides, CCC info, handouts on flash drive
- ❑ Spare 9v battery for mic
- ❑ Key card for classroom door

Student checklist for attending class

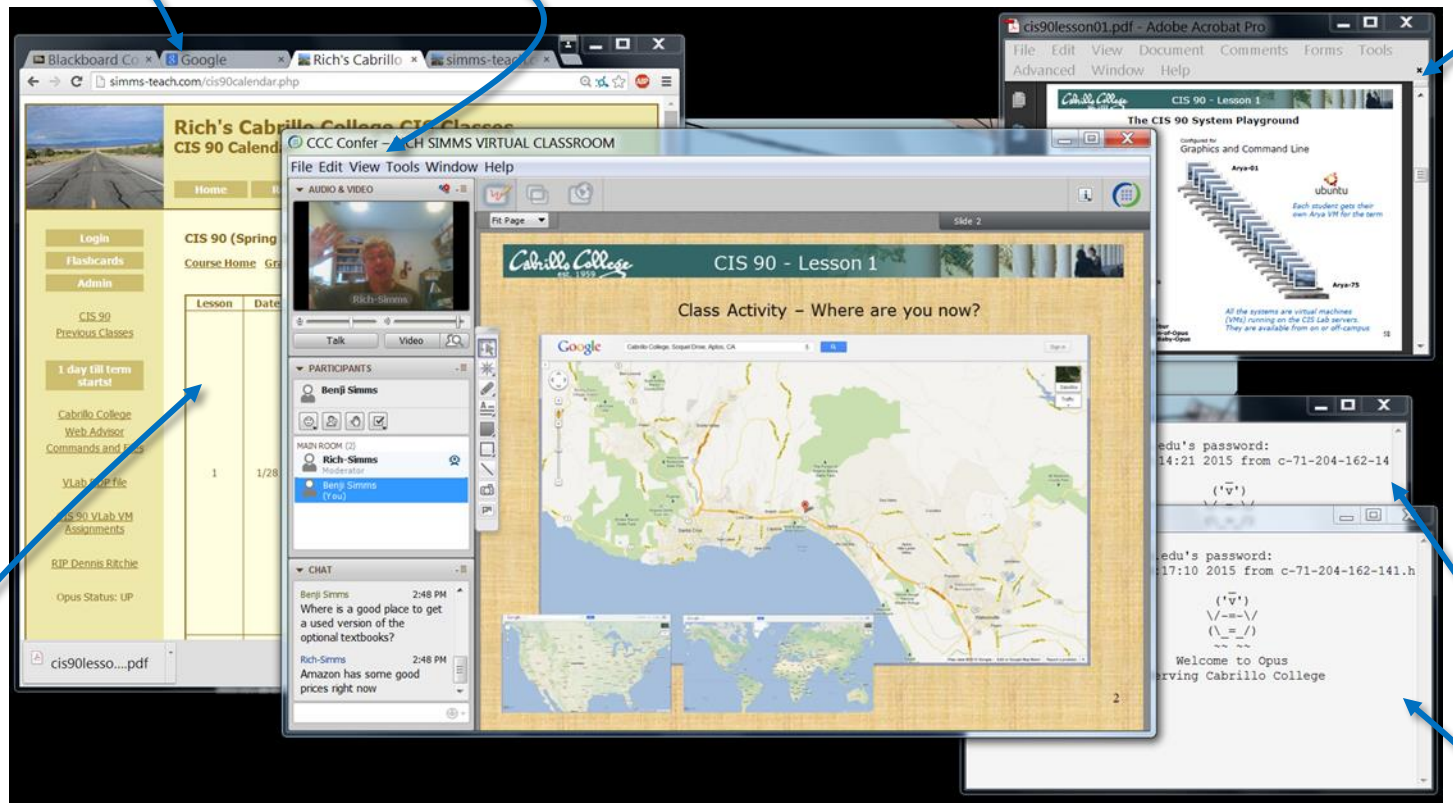


1. Browse to: **http://simms-teach.com**
2. Click the **CIS 76** link.
3. Click the **Calendar** link.
4. Locate today's lesson.
5. Find the **Presentation slides** for the lesson and **download** for easier viewing.
6. Click the **Enter virtual classroom** link to join CCC Confer.
7. Log into Opus with Putty or ssh command.

Note: Blackboard Collaborate Launcher only needs to be installed once. It has already been downloaded and installed on the classroom PC's.

Student checklist for suggested screen layout

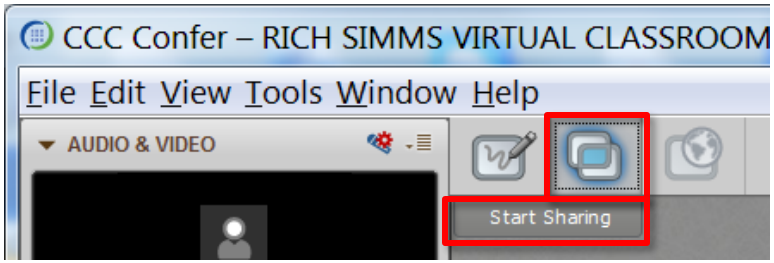
- ❑ *Google*
- ❑ *CCC Confer*
- ❑ *Downloaded PDF of Lesson Slides*



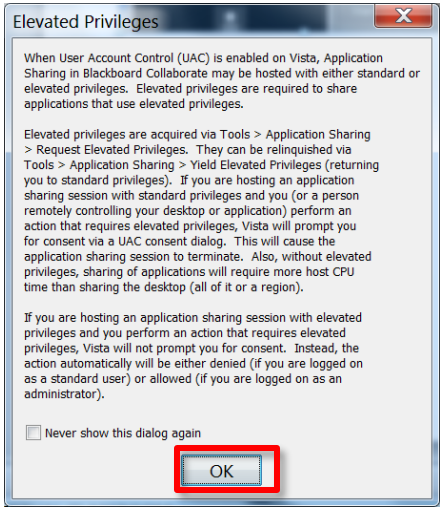
- ❑ *CIS 76 website Calendar page*
- ❑ *One or more login sessions to Opus*

Student checklist for sharing desktop with classmates

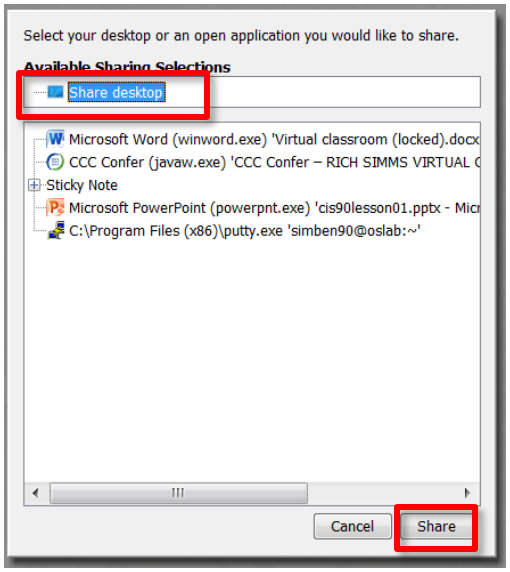
1) Instructor gives you sharing privileges.



2) Click overlapping rectangles icon. If white "Start Sharing" text is present then click it as well.



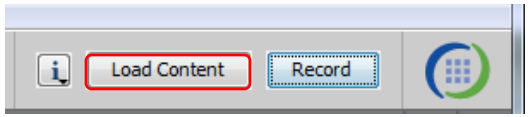
3) Click OK button.



4) Select "Share desktop" and click Share button.

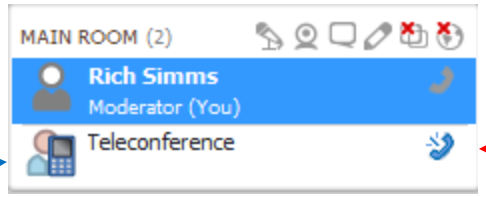
Rich's CCC Confer checklist - setup

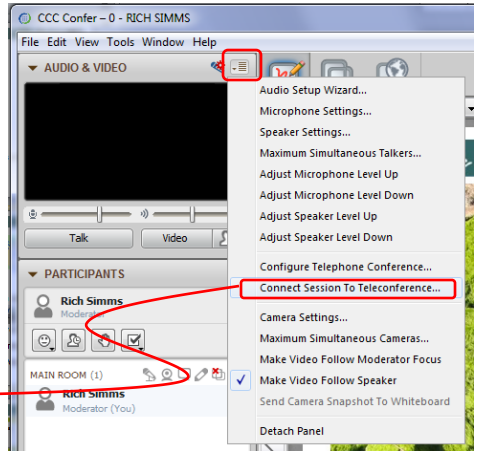
[ ] Preload White Board



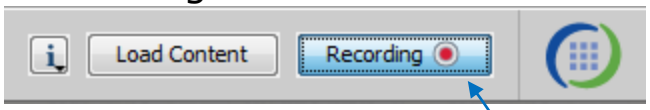
[ ] Connect session to Teleconference

Session now connected to teleconference





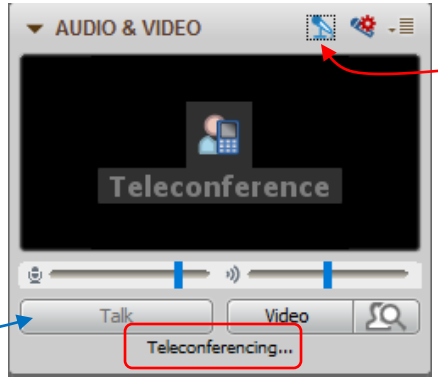
[ ] Is recording on?



Red dot means recording

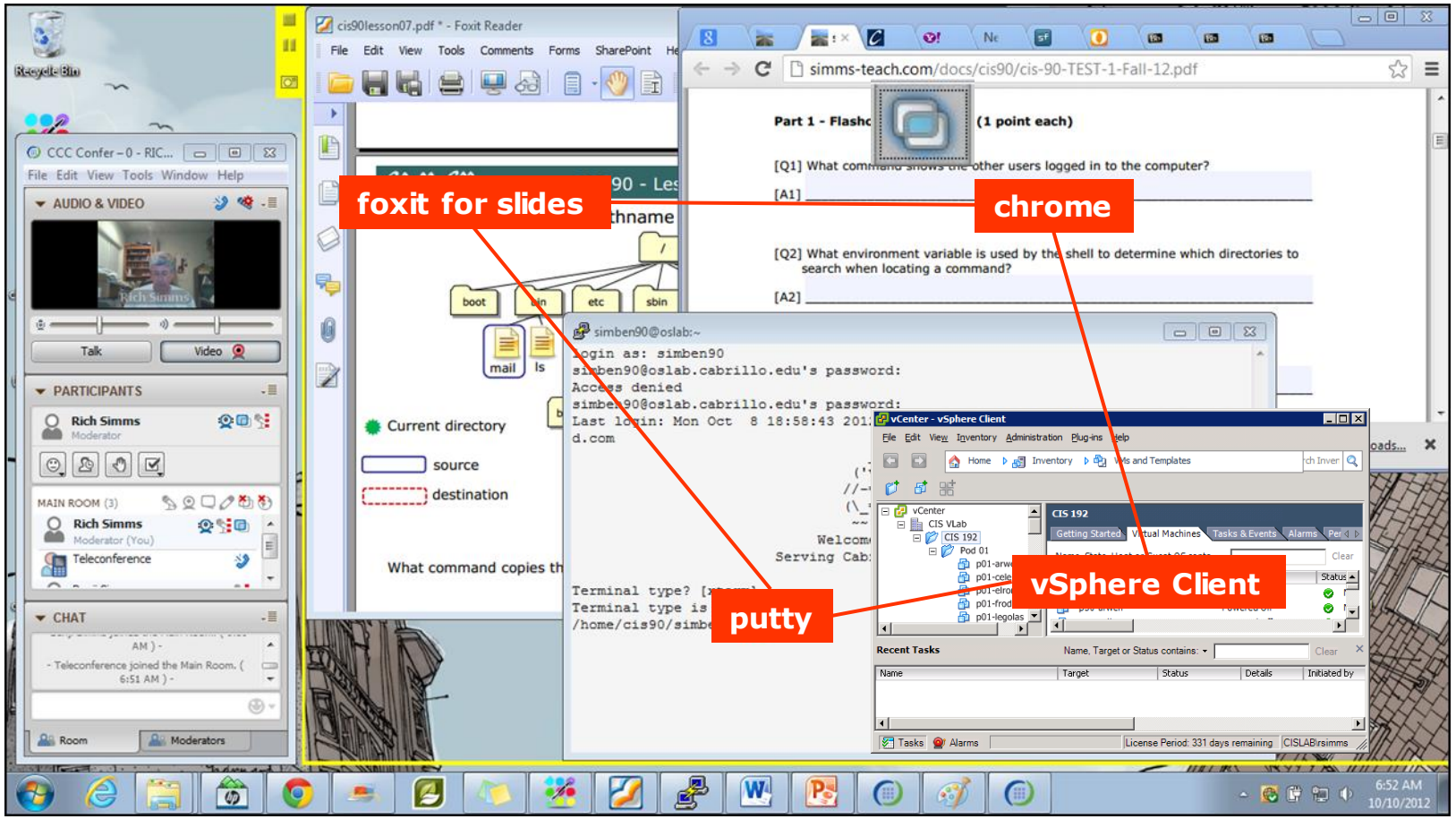
[ ] Use teleconferencing, not mic

Should be grayed out



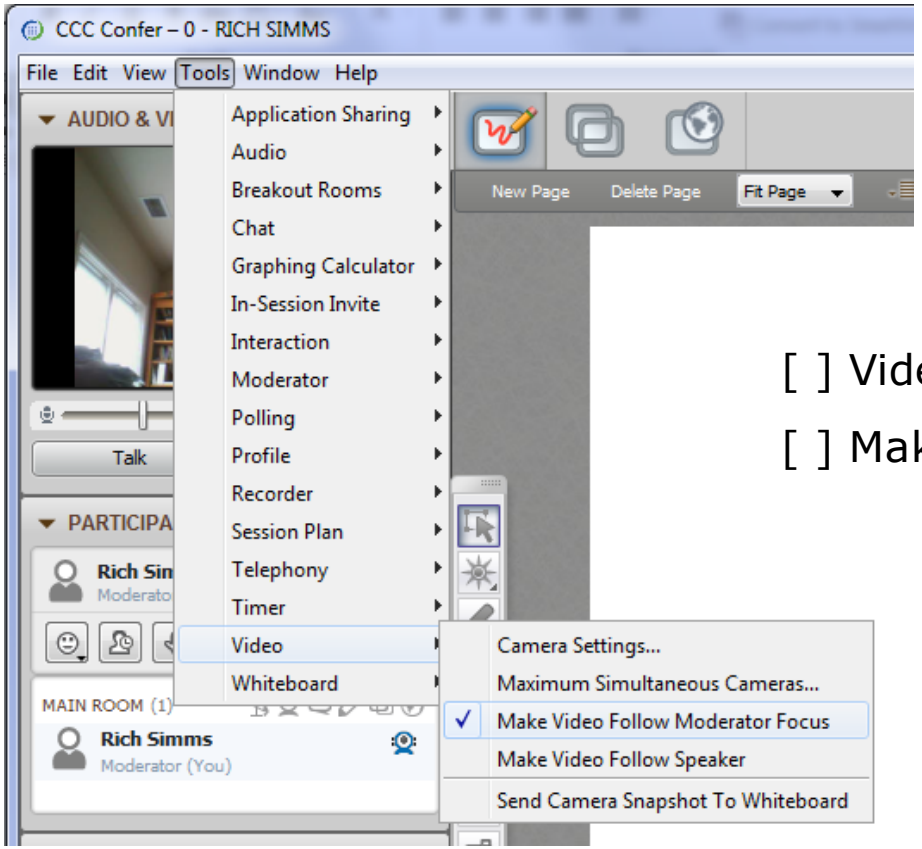
Should change from phone handset icon to little Microphone icon and the Teleconferencing … message displayed

Rich's CCC Confer checklist - screen layout



[ ] layout and share apps

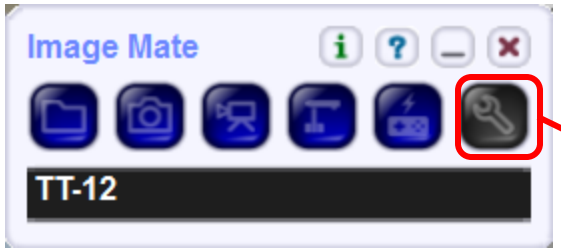
Rich's CCC Confer checklist - webcam setup

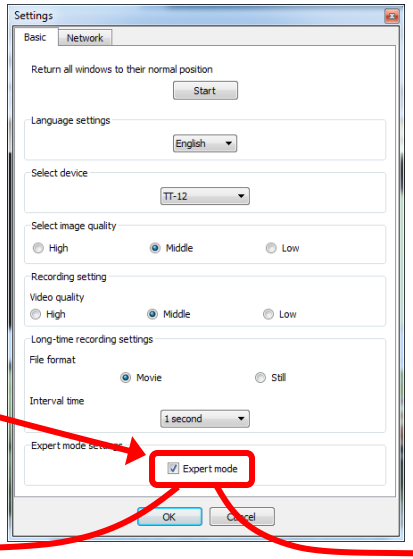


[ ] Video (webcam)

[ ] Make Video Follow Moderator Focus

Rich's CCC Confer checklist - Elmo



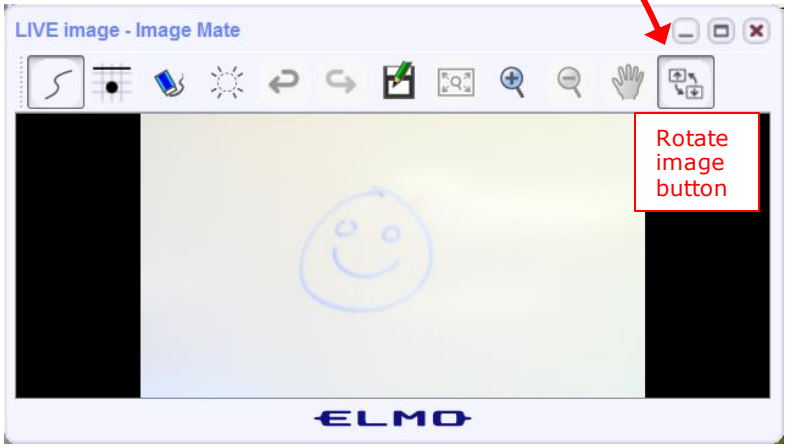


The "rotate image" button is necessary if you use both the side table and the white board.

Quite interesting that they consider you to be an "expert" in order to use this button!

Elmo rotated down to view side table

Elmo rotated up to view white board



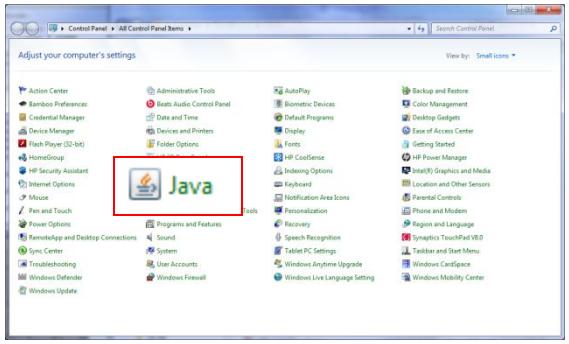
Run and share the Image Mate program just as you would any other app with CCC Confer

Rich's CCC Confer checklist - universal fixes

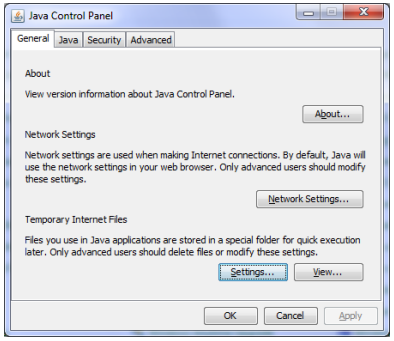
Universal Fix for CCC Confer:

1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime
3) http://www.cccconfer.org/support/technicalSupport.aspx

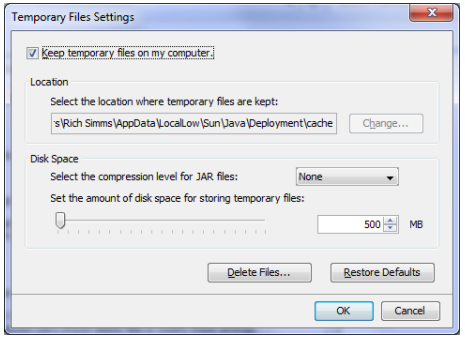
Control Panel (small icons)



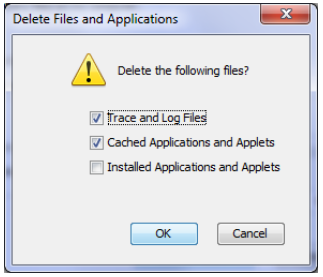
General Tab > Settings…



500MB cache size



Delete these



Google Java download

Start

Sound Check

*Students that dial-in should mute their line using *6 to prevent unintended noises distracting the web conference.*

*Instructor can use *96 to mute all student lines.*

First Minute Quiz

Please answer these questions **in the order** shown:

Use CCC Confer White Board

email answers to: `risimms@cabrillo.edu`

(answers must be emailed within the first few minutes of class for credit)

TCP/IP Review

Objectives	Agenda
• Review the TCP/IP protocol stack • Review IP addressing	• Quiz #1 • Certifications • Vocabulary • Conferences • Newsletters and Blogs • TCP/IP model • Network Access layer • Internet layer • Transport layer • Application layer • Assignment • Wrap up

Introductions and Credits

Rich Simms
- HP Alumnus.
- Started teaching in 2008 when Jim Griffin went on sabbatical.
- Rich's site: http://simms-teach.com

And thanks to:
- Steven Bolt at for his WASTC EH training.
- Kevin Vaccaro for his CSSIA EH training and Netlab+ pods.
- EC-Council for their online self-paced CEH v9 course.
- Sam Bowne for his WASTC seminars, textbook recommendation and fantastic EH website (https://samsclass.info/).
- Lisa Bock for her great lynda.com EH course.
- John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system (http://teacherjohn.com/).
- Google for everything else!

Credits

Rick Graziani

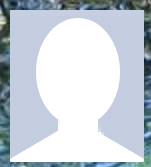
- Thanks to Rick Graziani for the use of some of his great network slides
- Rick's site: http://www.cabrillo.edu/~rgraziani/

Instructor: **Rich Simms**
Dial-in: **888-886-3951**
Passcode: **136690**


Jordan

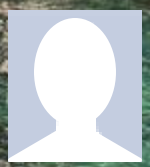
Ryan


Takashi


Karl-Heinz


Sean


Benji


Joshua


Brian


Tess


Jeremy

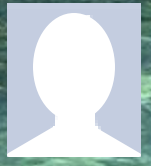

David H.


Roberto


Nelli


Mike C.


Deryck


Alex


Michael W.


Carter


Thomas


Wes


Jennifer


Marcos

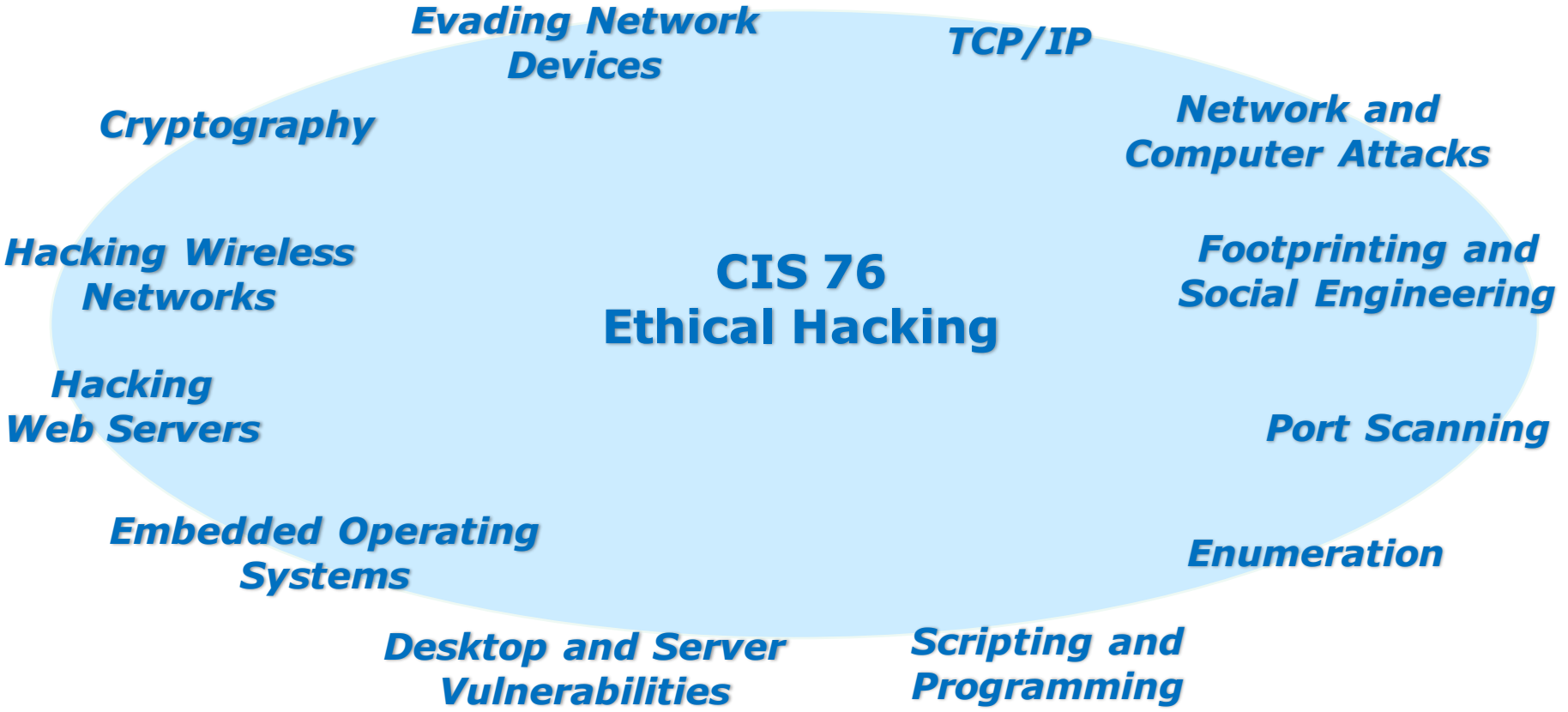

Tim


Daniel


Dave R.


Luis

Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit



Student Learner Outcomes

1. Defend a computer and a LAN against a variety of different types of security attacks using a number of hands-on techniques.

2. Defend a computer and a LAN against a variety of different types of security attacks using a number of hands-on techniques.

Admonition

Unauthorized hacking is a crime.

The hacking methods and activities learned in this course can result in prison terms, large fines and lawsuits if used in an unethical manner. They may only be used in a lawful manner on equipment you own or where you have explicit permission from the owner.

Students that engage in any unethical, unauthorized or illegal hacking may be dropped from the course and will receive no legal protection or help from the instructor or the college.

Questions

Questions

How this course works?

Past lesson material?

Previous labs?

Chinese Proverb	他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。 *He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.*

Certifications

	SB	KV	Simpson Textbook	Concise Cybersecurity
A+ (CompTIA)		1		
Linux Essentials (LPI)		3		
Linux+ (CompTIA)	x			
Network+ (CompTIA)		2	x	
Security+ (CompTIA)	1	4	x	x
CISSP (ISC^2)		6a	x	
CEH (EC-Council)	2	5	x	x
GPEN (SANS/GIAC)	3	6b	x	x
OPST (ISECOM)			x	
OSCP (Offensive Security)	x			x

Vocabulary

Some Terminology

- Hacking
- Cracking
- White hat hacker
- Grey hat hacker
- Black hat hacker
- Nation-state actors
- Cybercriminals
- Adversary
- Hacktivist
- Pen Test
- Security audit
- White box testing
- Grey box testing
- Black box testing
- Red Team
- Blue Team
- Vulnerability
- Exploit
- Threat
- Denial of Service attack
- Brute force attack
- Buffer overflow
- Spoofing
- Zero-day
- Botnet
- Ransomware (link)
- Watering hole attack (link)
- Man in the middle attack
- Fuzzing (link)
- Drive-by-download (link)
- Cross-site scripting (link)
- SQL injection (link)
- Malware
- Virus
- Trojan (link)
- Worm (link)
- Spyware
- Rootkit (link)
- Firewall
- Signatures (link)
- Polymorphism
- Exfiltrate
- Social engineering
- Phishing
- Vishing (listen)
- Spear-phishing

Acronyms

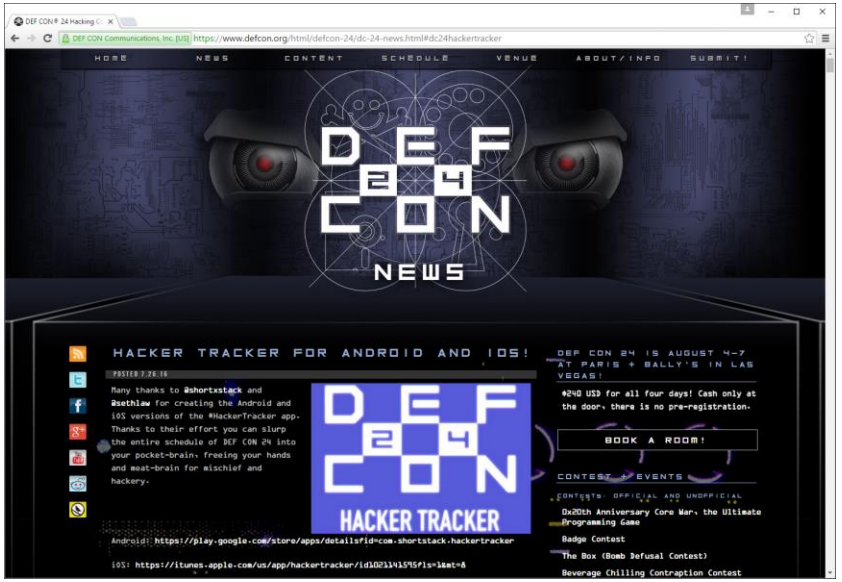
- CVE (Common Vulnerabilities and Exposures)
- DoS (Denial of Service attack)
- DDoS (Distributed Denial of Service attack)
- XSS (Cross-Site Scripting)
- IDS (Intrusion Detection System)
- IPS (Intrusion Prevention System)
- C&C (Command and Control)
- AV (Anti-Virus)
- APT (Advanced Persistent Threat)
- RAT (Remote Access Trojan)

Slang

- Owned
- Pwned
- Meat chicken ("rouji" in Chinese)
- Doxing
- Script Kiddie
- Packet Monkey

Conferences

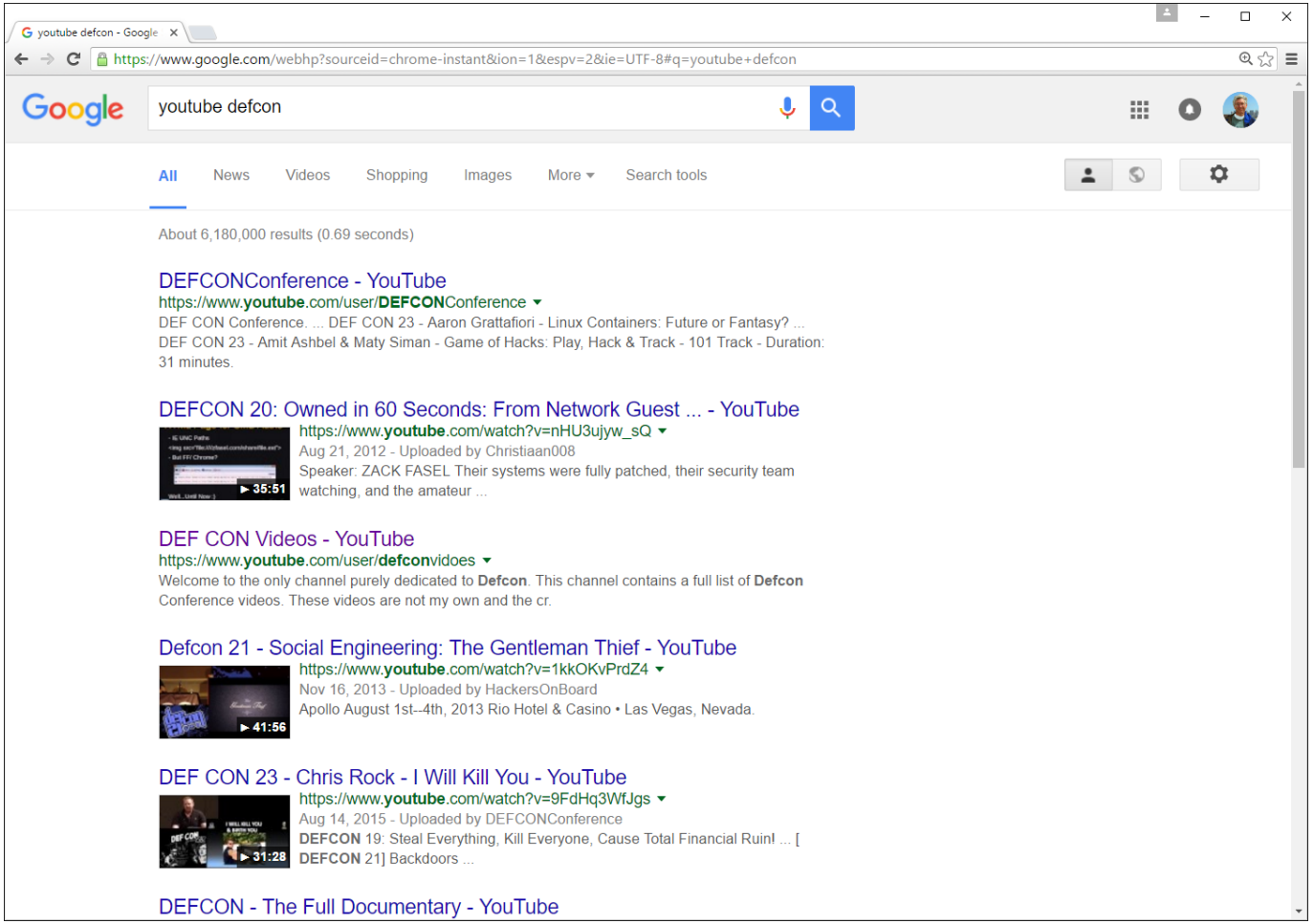
Black Hat



DEF CON

And many more: ToorCon, Hackers Halted, RSA, OWASP events, ShmooCon, DerbyCon, Thotcon, USENIX...

Google: youtube defcon



Looking ahead ...

Sept 10-19 2016, SANS Network Security 2016 Las Vegas

July 22-27 2017, Black Hat USA 2017 Las Vegas

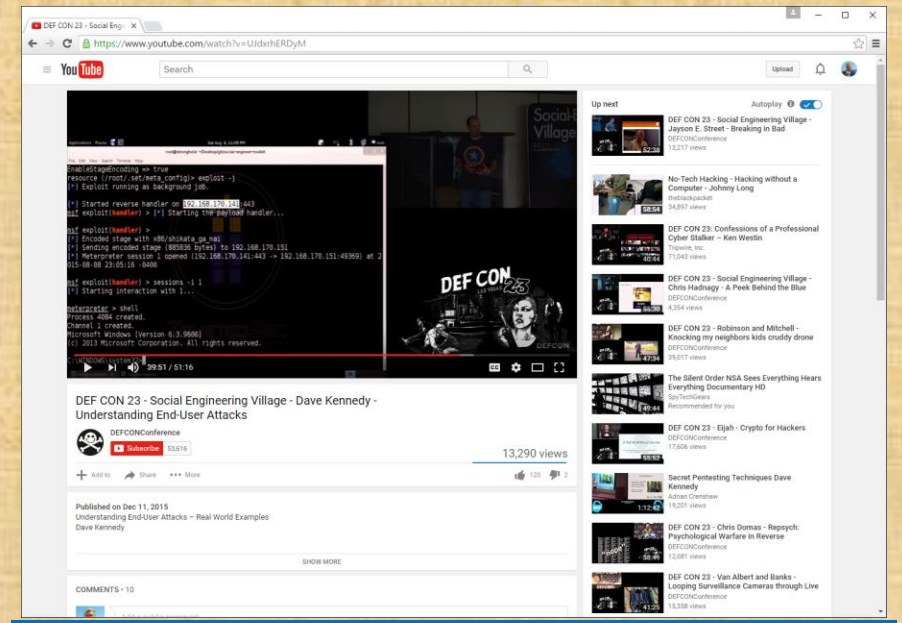
July 27-30 2017, DEF CON 25 Las Vegas

https://www.concise-courses.com/security/conferences-of-2016/

https://www.concise-courses.com/security/conferences-of-2017/

An Expert at Work Activity

David Kennedy at Def Con 23 hacking a PC with the Social Engineering Toolkit and Metasploit



https://www.youtube.com/watch?v=UJdxrhERDyM

1. Watch a portion of this video (34:00-39:45). In the HTA attack what did he mean when he said "there we go, we get our shell"?
   (put your answer in the chat window)

2. Watch a portion of this video (39:45-44:00). In the web-jacking attack what was he able to accomplish?
   (put your answer in the chat window)

Newsletters and Blogs

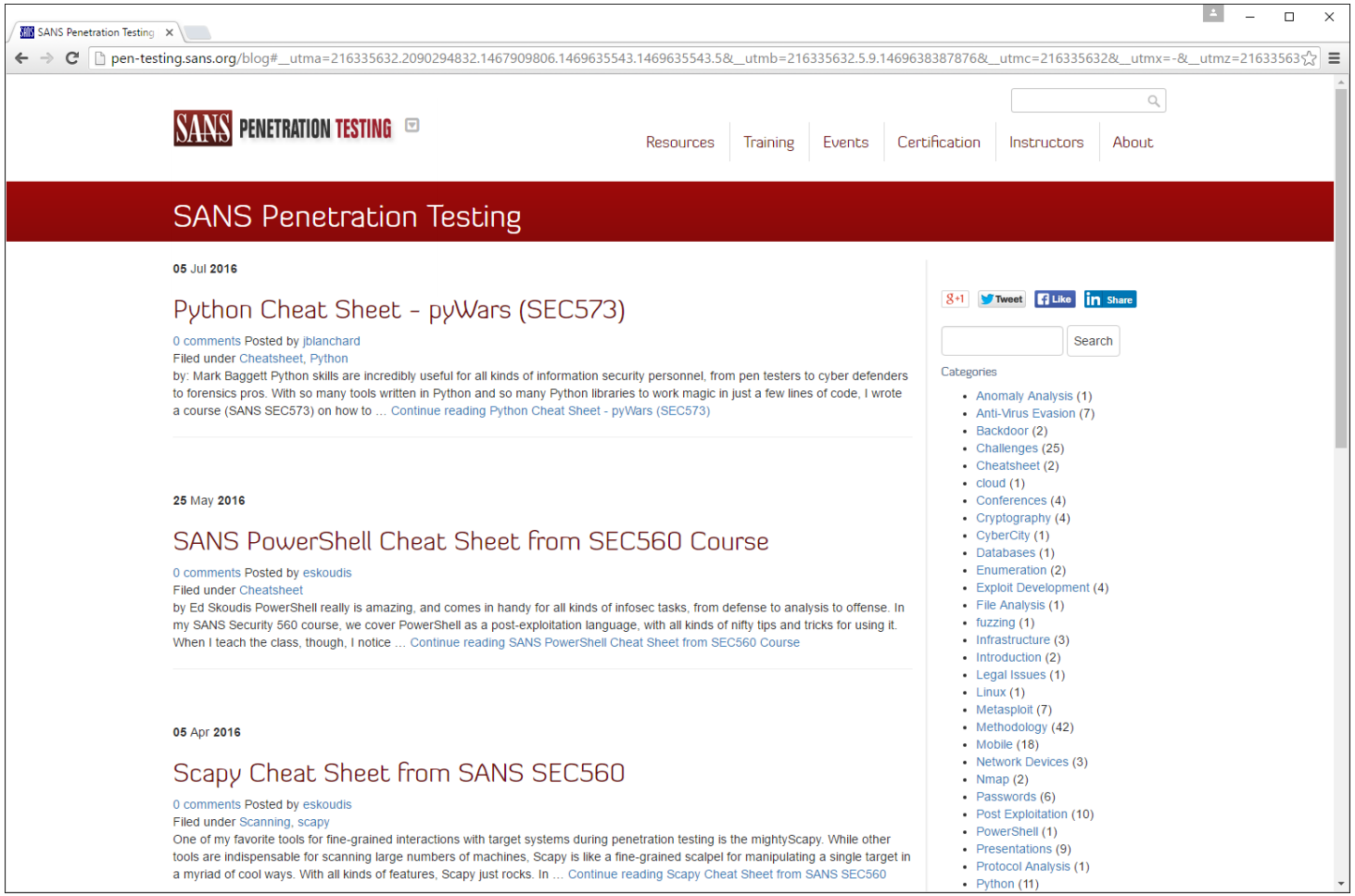
Subscribe or sign up for cyber security newsletters, alerts, blogs and feeds

- US-CERT
- SANS
- Cybrary
- FireEye
- CrowdStrike
- AlienVault
- HackerNews
- Many more ...

Department of Homeland Security - US-CERT

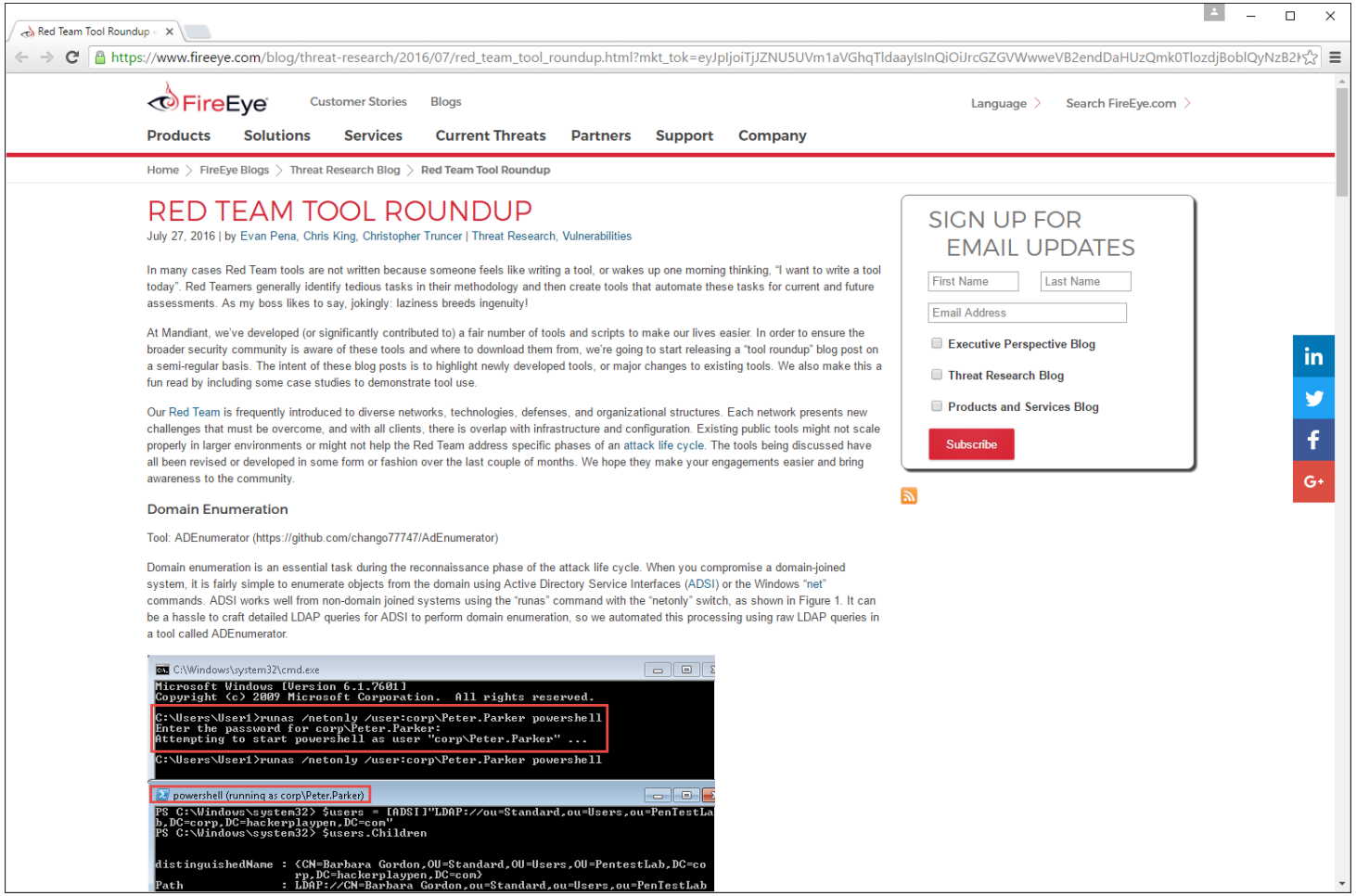
https://www.us-cert.gov/mailing-lists-and-feeds

SANS Blogs



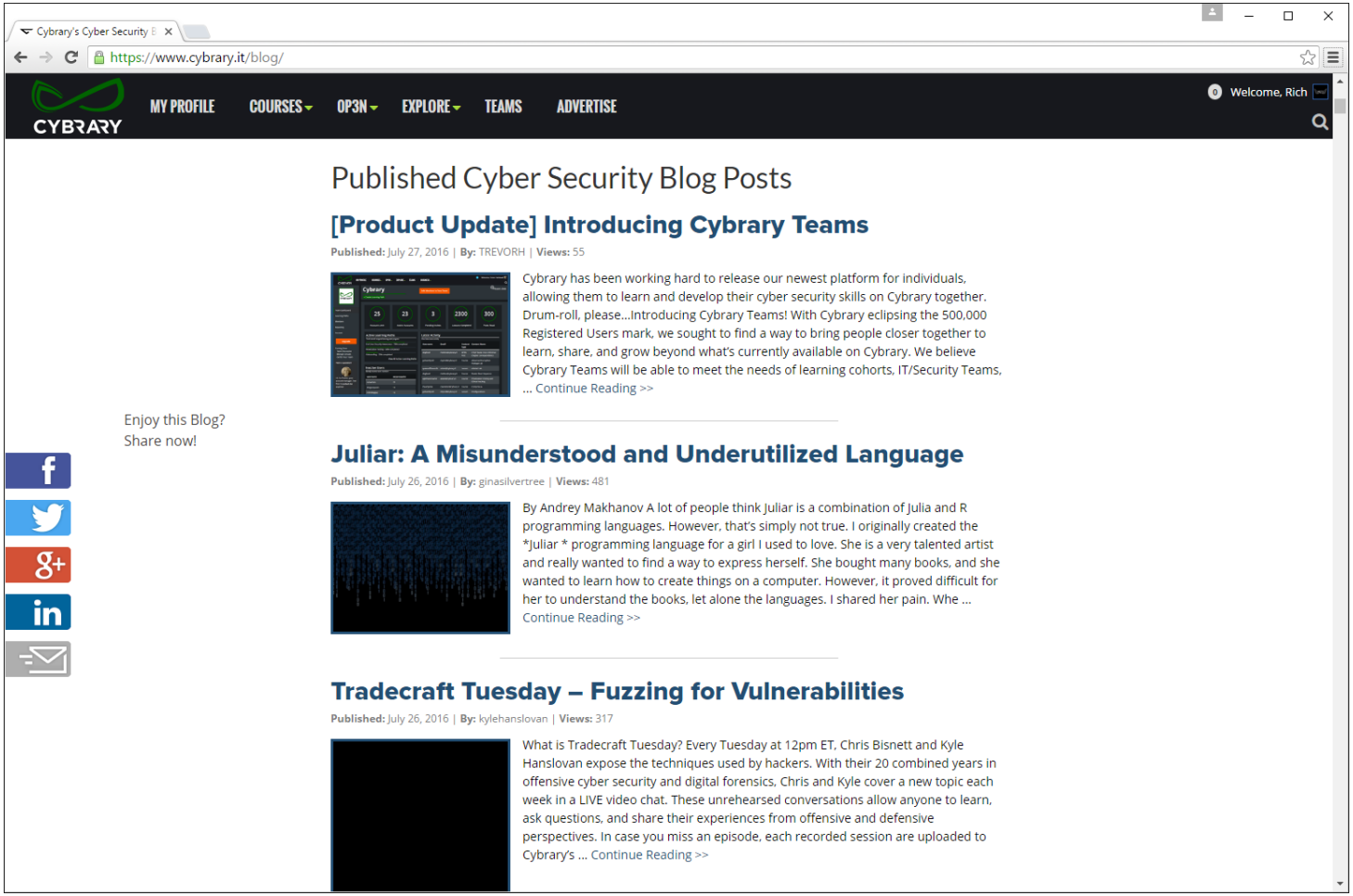
https://www.sans.org/security-resources/blogs

FireEye Blogs



https://www.fireeye.com/blog/threat-research.html

Cybrary



https://www.cybrary.it/blog/

Hacker News

http://thehackernews.com/

Housekeeping

Housekeeping

1. Send me your student survey & agreement today.
2. Lab 1 due by 11:59PM (Opus time) tonight.
3. Last day to drop/add is this Saturday.

Change your default password on Opus

Roll Call

If you are attending class by watching the recordings in the archives email the instructor at: risimms@cabrillo.edu to provide roll call attendance.

Navigating SANS Pen Test Cheat Sheets for Fun and Profit

Thursday, September 8th, 2016 at 11:00 AM (11:00:00 EDT/US Eastern)
Ed Skodis

You can now attend the webcast using your mobile device!

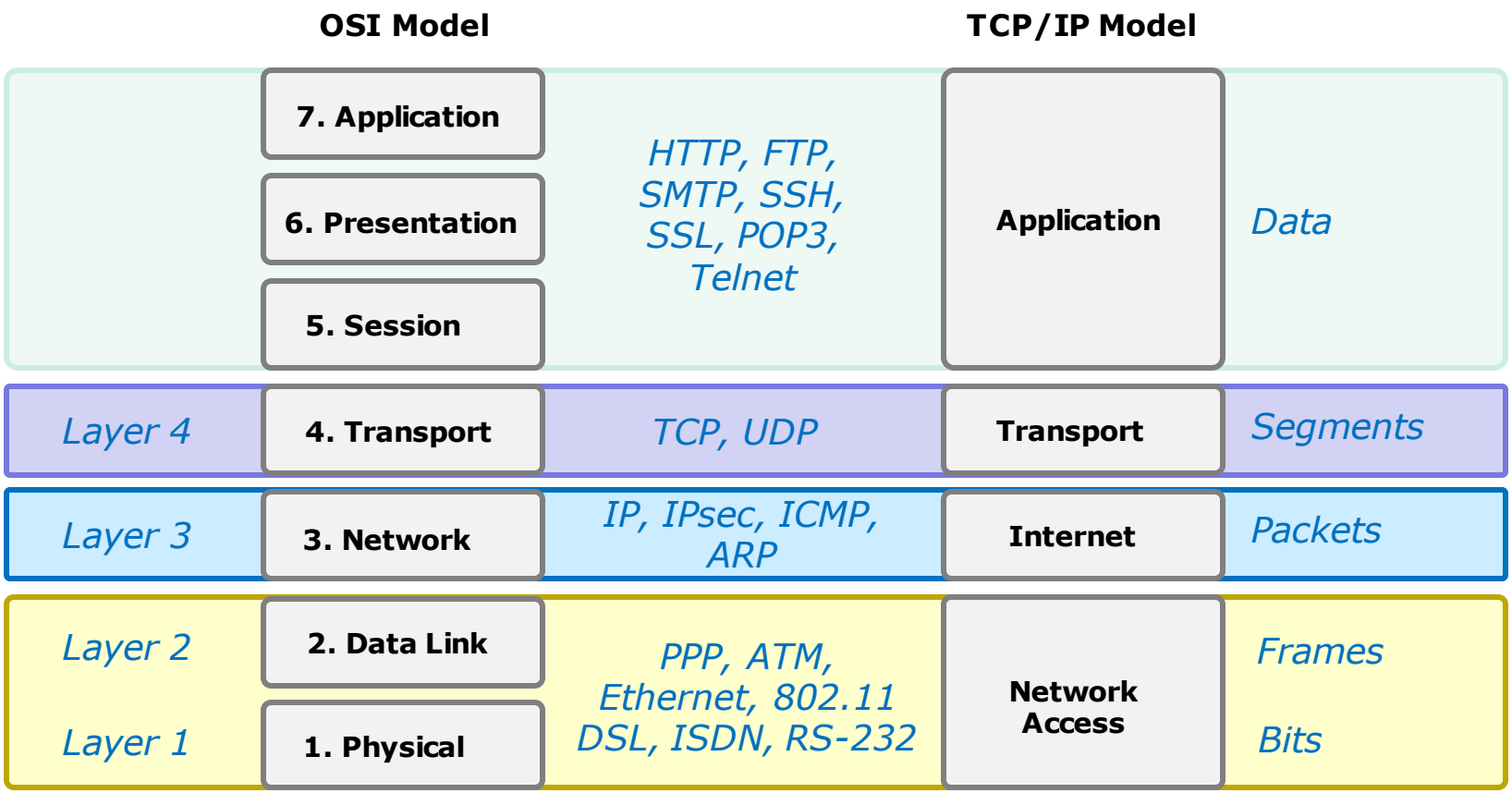
Overview

Extra! Extra! Get yer SANS Pen Test cheat sheets here! As you may have noticed, SANS Pen Test Authors and Instructors have been on a tear lately, releasing numerous cheat sheets to help people build valuable skills. Recent releases include topics such as PowerShell, Scapy, Nmap, Metasploit, and a whole bunch more. In this engaging webcast, Ed Skoudis will take you on a tour of some of the most useful tips and ideas of these cheat sheets. Five essential things to know about PowerShell? We got that. The most useful Scapy options at a glance? Got that too. The syntax for Netcat client-to-client relays? Yep. Some late-breaking Metasploit kung fu? Oh yes. Well also look at ways you can use the cheat sheets in your work and we might even go over some useful origami tricks. Ed will also solicit ideas from you on new cheat sheets youd like to see.

https://www.sans.org/webcasts/navigating-pen-test-cheat-sheets-fun-profit-102897?utm_medium=Email&utm_source=House+List&utm_content=Navigating+Pen+Test+Cheat+Sheets&utm_campaign=SANS+Webcast+Internal

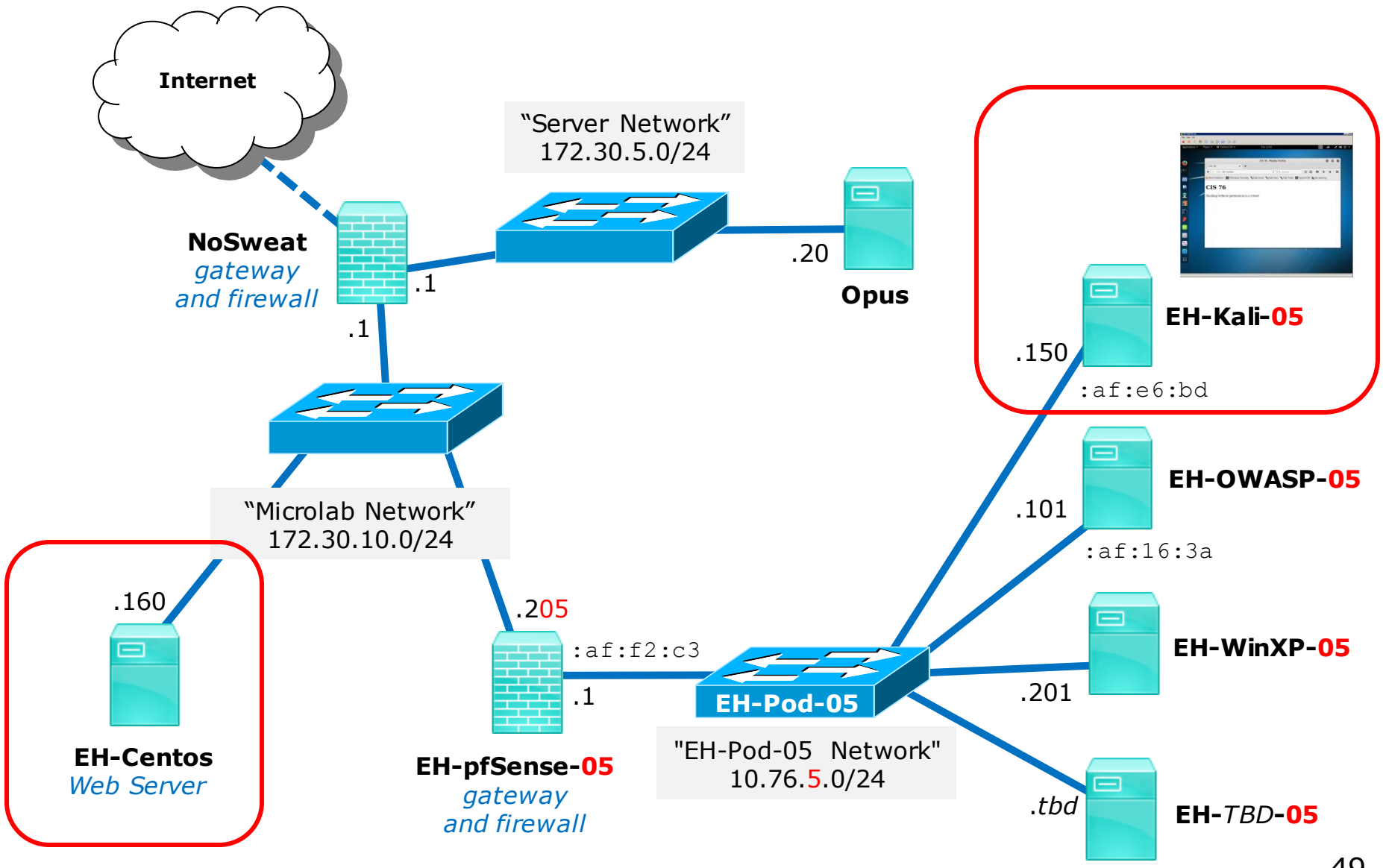
TCP/IP Review

OSI and TCP/IP Models



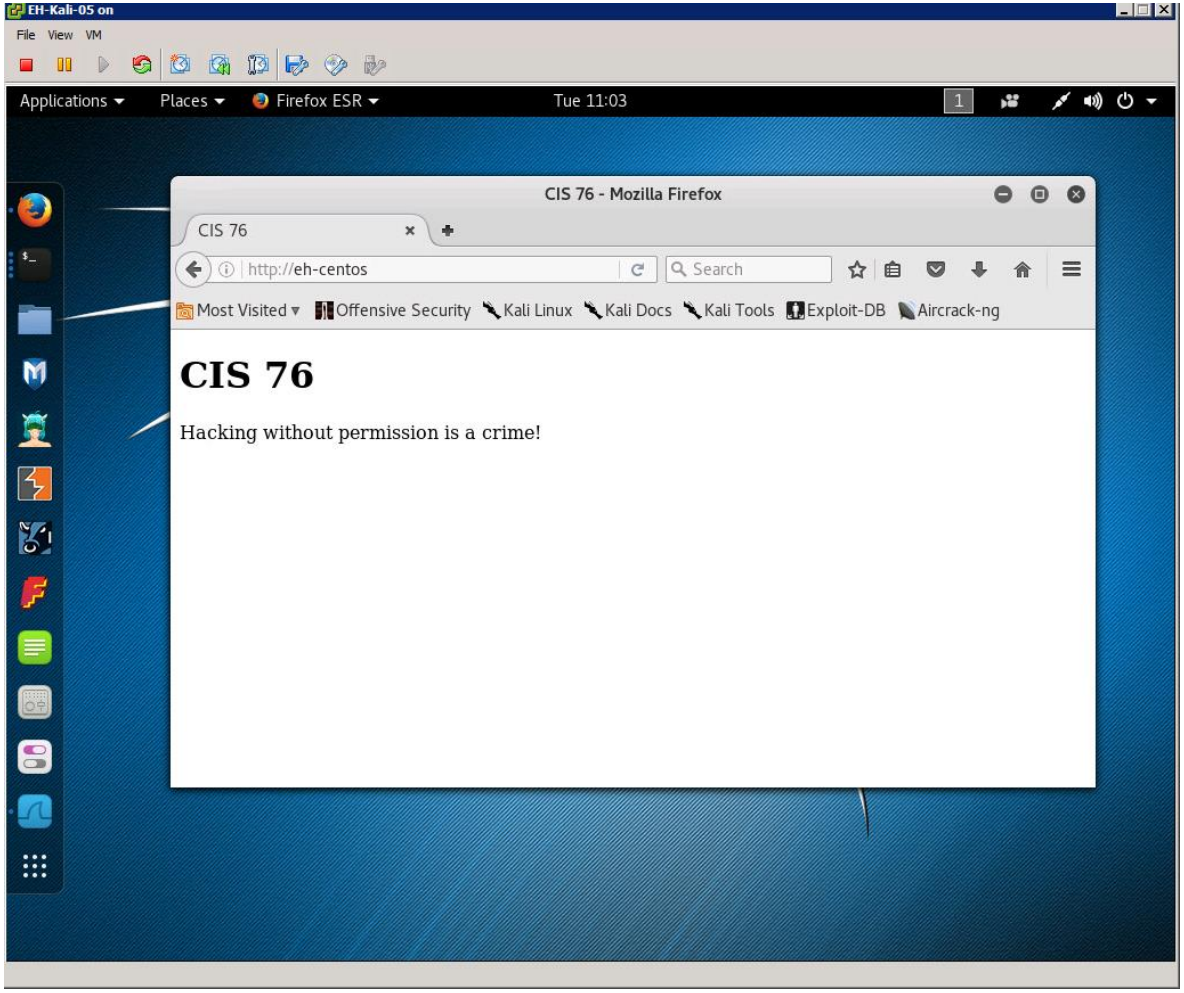
Open Systems Interconnection model

Model used to build the Internet

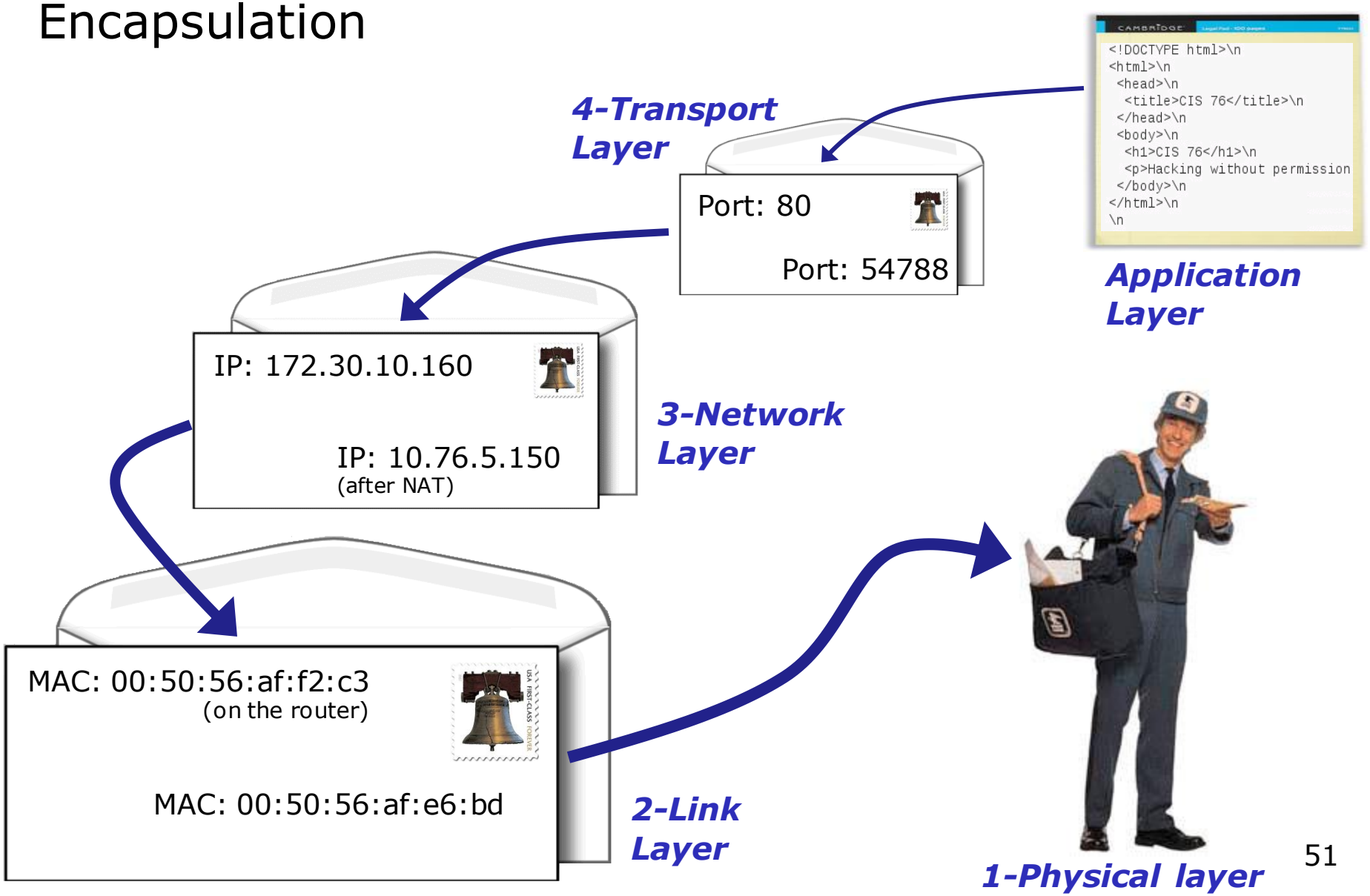

Internet
"Server Network"
172.30.5.0/24
NoSweat
gateway
and firewall
.1
.1
.20
Opus
.150
EH-Kali-05
:af:e6:bd
.101
EH-OWASP-05
:af:16:3a
.201
EH-WinXP-05
.tbd
EH-TBD-05
"Microlab Network"
172.30.10.0/24
.160
EH-Centos
Web Server
.205
:af:f2:c3
.1
EH-pfSense-05
gateway
and firewall
EH-Pod-05
"EH-Pod-05 Network"
10.76.5.0/24

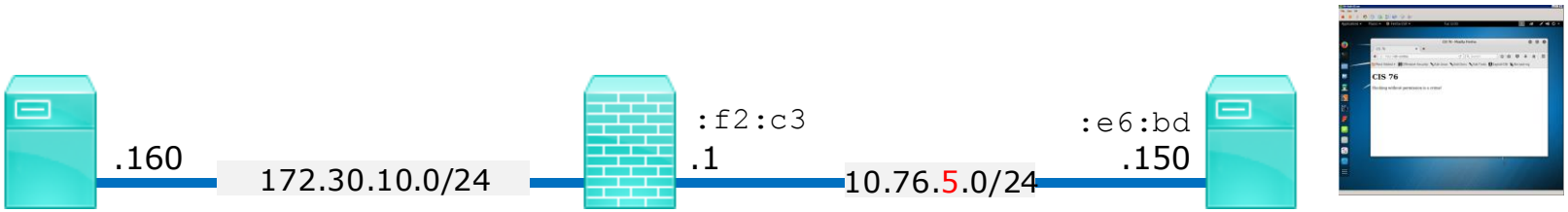
HTTP Application Example

Kali browsing a web page on EH-Centos



Encapsulation



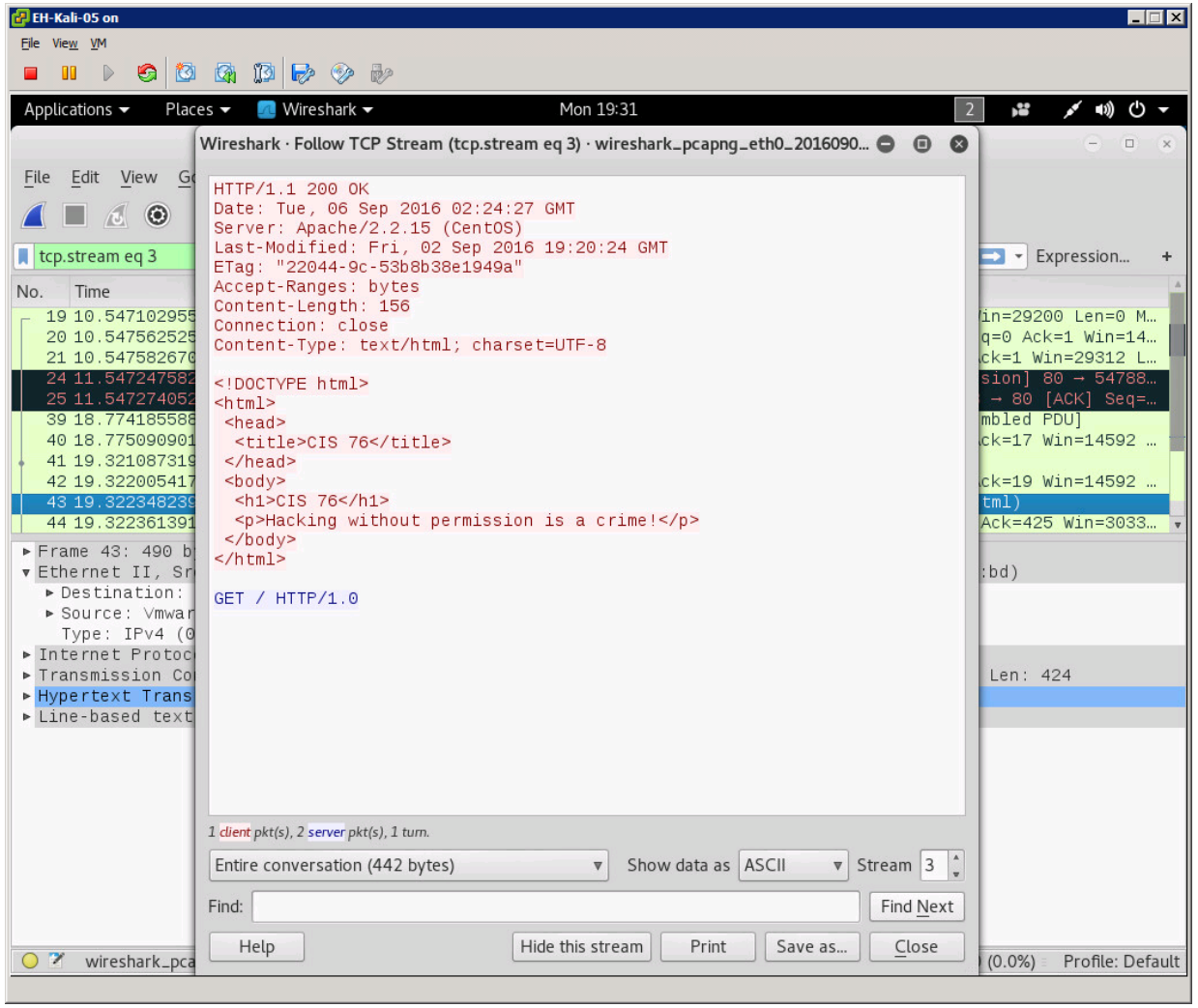


No.	Time	Source	Destination	Protocol	Leng	Info
41	19.321087319	10.76.5.150	172.30.10.160	HTTP	68	GET / HTTP/1.0
43	19.322348239	172.30.10.160	10.76.5.150	HTTP	490	HTTP/1.1 200 OK (text/html)

```
▸ Frame 43: 490 bytes on wire (3920 bits), 490 bytes captured (3920 bits) on interface 0
▾ Ethernet II, Src: Vmware_af:f2:c3 (00:50:56:af:f2:c3), Dst: Vmware_af:e6:bd (00:50:56:af:e6:bd)
  ▸ Destination: Vmware_af:e6:bd (00:50:56:af:e6:bd)
  ▸ Source: Vmware_af:f2:c3 (00:50:56:af:f2:c3)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 172.30.10.160, Dst: 10.76.5.150
▸ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 54788 (54788), Seq: 1, Ack: 19, Len: 424
▸ Hypertext Transfer Protocol
▾ Line-based text data: text/html
    <!DOCTYPE html>\n
    <html>\n
     <head>\n
      <title>CIS 76</title>\n
     </head>\n
     <body>\n
      <h1>CIS 76</h1>\n
      <p>Hacking without permission is a crime!</p>\n
     </body>\n
    </html>\n
    \n
```

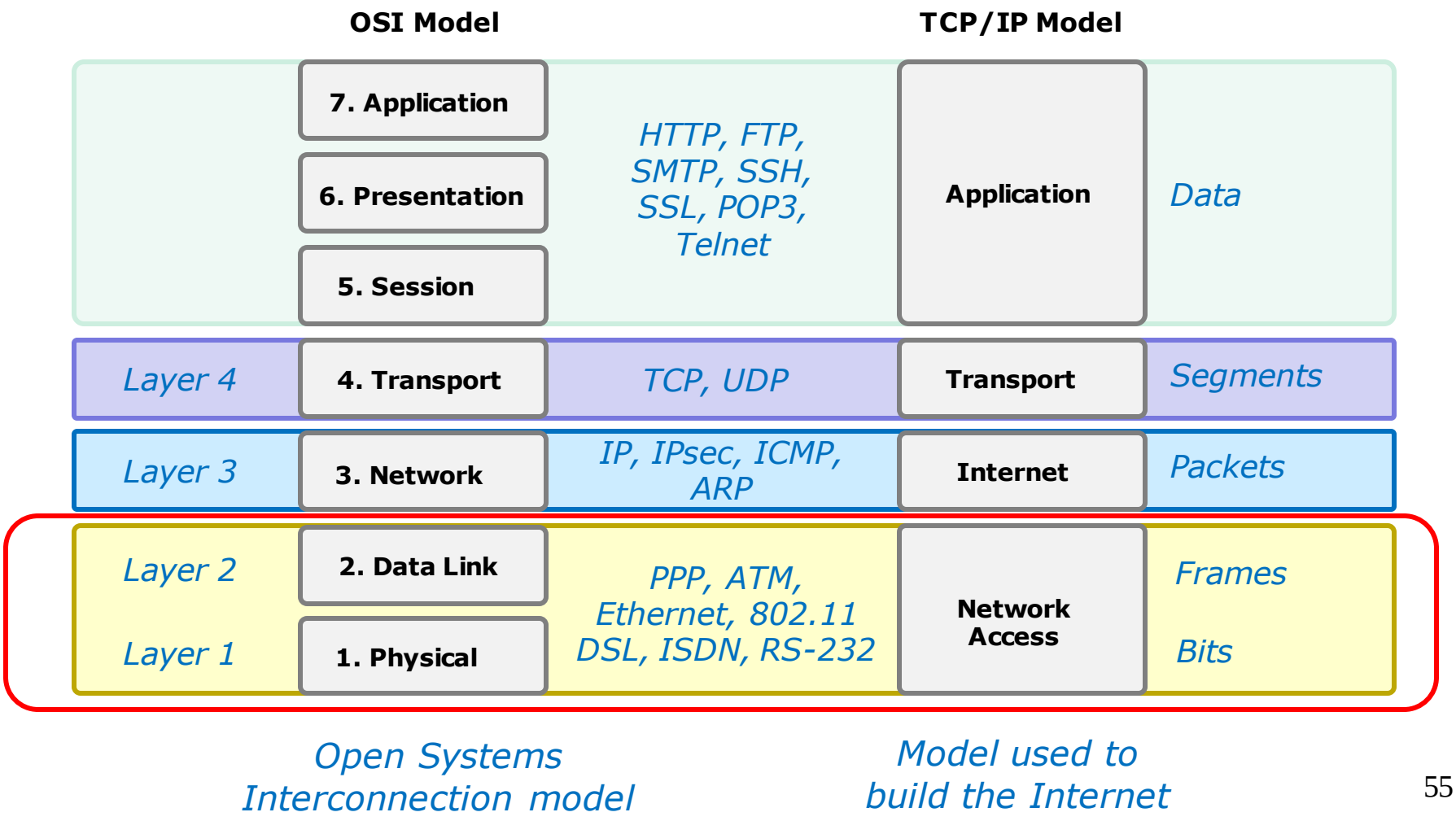
Wireshark View
On Kali

Wireshark Follow TCP Stream View On Kali

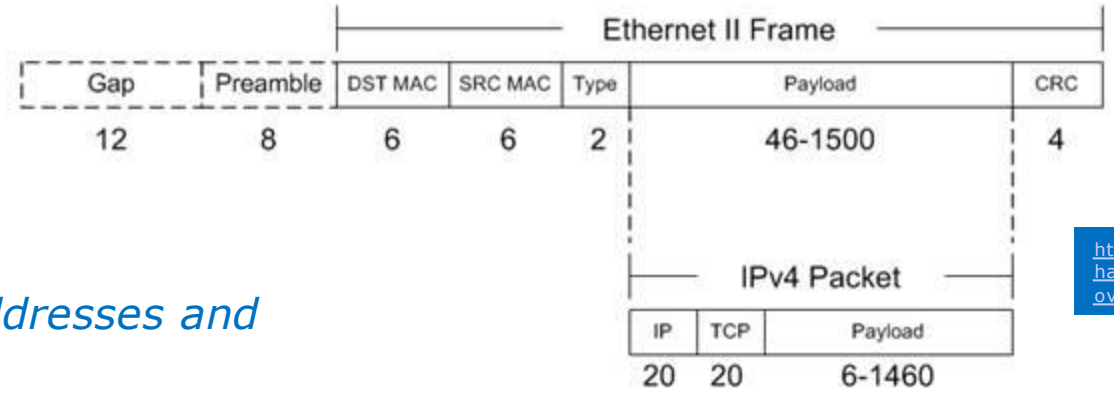


Network Access Layer

OSI and TCP/IP Models



Lets start at the bottom



http://www.tamos.net/~rhay/overhead/ip-packet-overhead.htm

Note the MAC addresses and type of payload

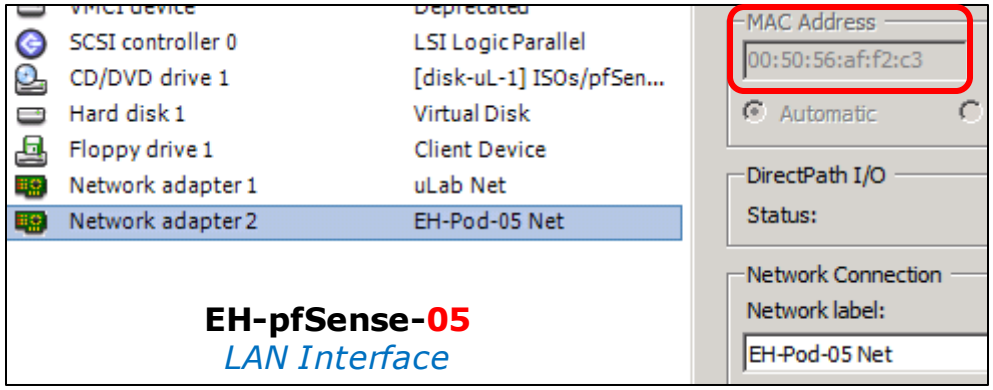
```
No.  Time          Source          Destination     Protocol Leng Info
  41 19.321087319  10.76.5.150     172.30.10.160   HTTP       68 GET / HTTP/1.0
  43 19.322348239  172.30.10.160   10.76.5.150     HTTP      490 HTTP/1.1 200 OK  (text/html)

▸ Frame 43: 490 bytes on wire (3920 bits), 490 bytes captured (3920 bits) on interface 0
▾ Ethernet II, Src: Vmware_af:f2:c3 (00:50:56:af:f2:c3), Dst: Vmware_af:e6:bd (00:50:56:af:e6:bd)
  ▸ Destination: Vmware_af:e6:bd (00:50:56:af:e6:bd)
  ▸ Source: Vmware_af:f2:c3 (00:50:56:af:f2:c3)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 172.30.10.160, Dst: 10.76.5.150
▸ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 54788 (54788), Seq: 1, Ack: 19, Len: 424
▸ Hypertext Transfer Protocol
▸ Line-based text data: text/html
```

Layer 2 - Ethernet MAC Address

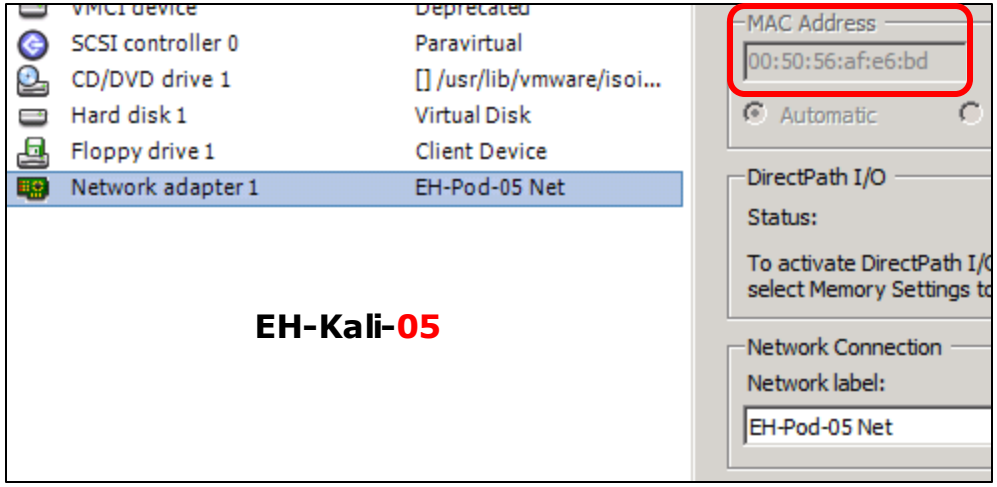
- Layer 2 defines how the streams of bits are organized into frames.
- In Ethernet each frame has a source and destination MAC address.
- MAC (Media Access Control) addresses came from the original Xerox Ethernet addressing scheme.
- A MAC address has 48 bits (6 octets).
  - e.g. 00:50:56:af:e6:bd
  - Note the use of hexadecimal digits to specify the octets.
- First three octets are the OUI (Organizationally Unique Identifier).
- Last three octets are unique to the NIC (Network Interface Controller).

Layer 2 - Ethernet MAC Addresses on VMs



EH-pfSense-05
LAN Interface

`00:50:56:af:f2:c3`

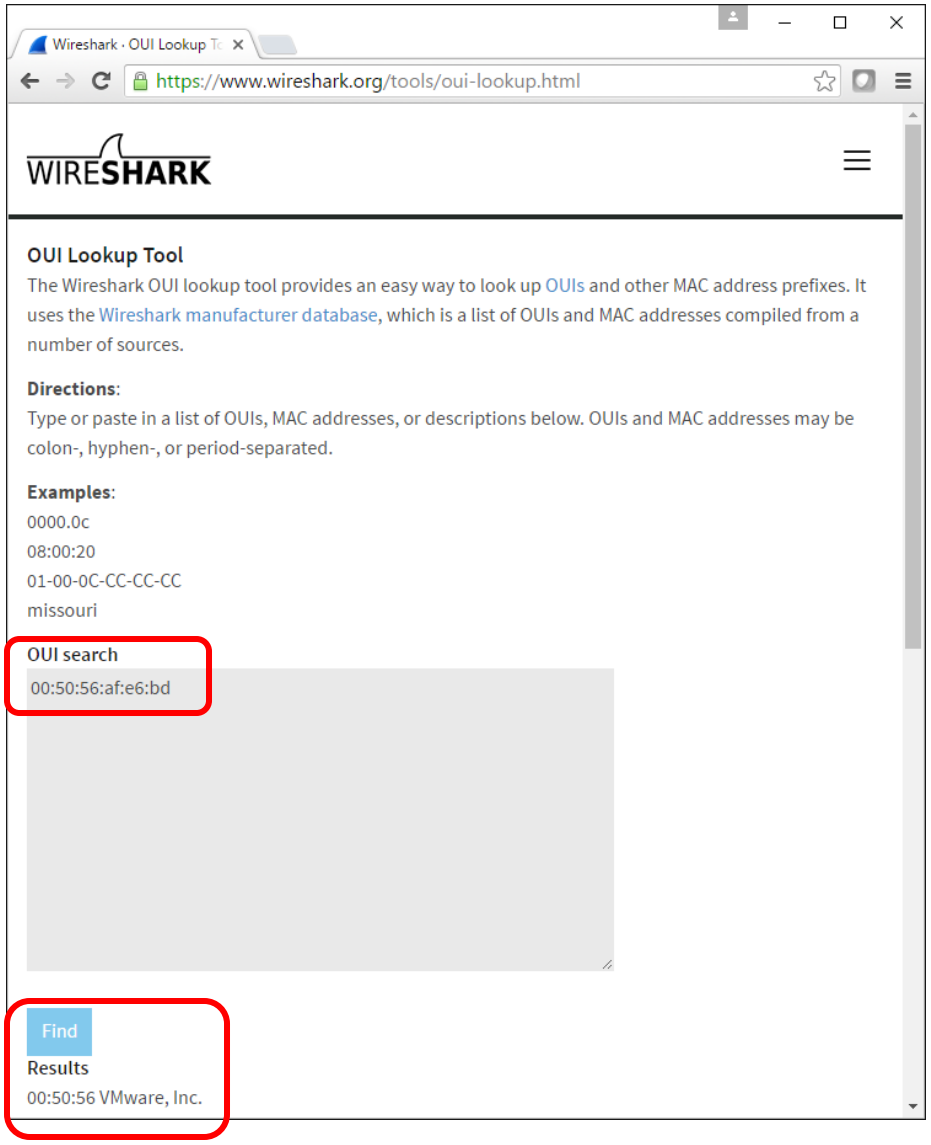


EH-Kali-05

`00:50:56:af:e6:bd`

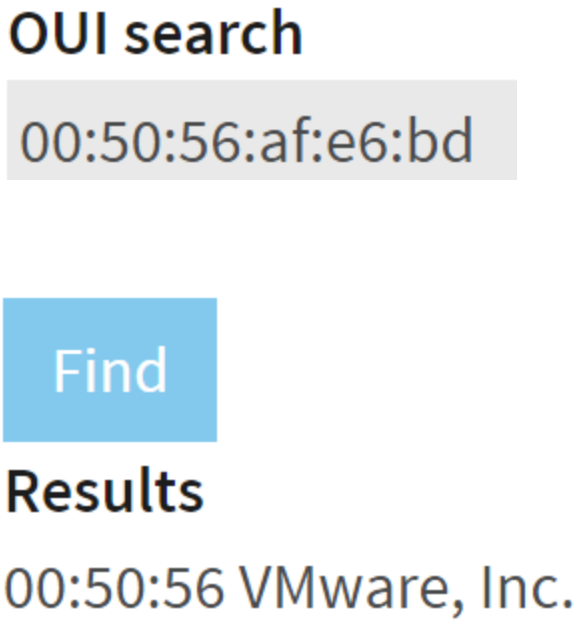
Use "Edit Settings" to view MAC addresses on the network adapters

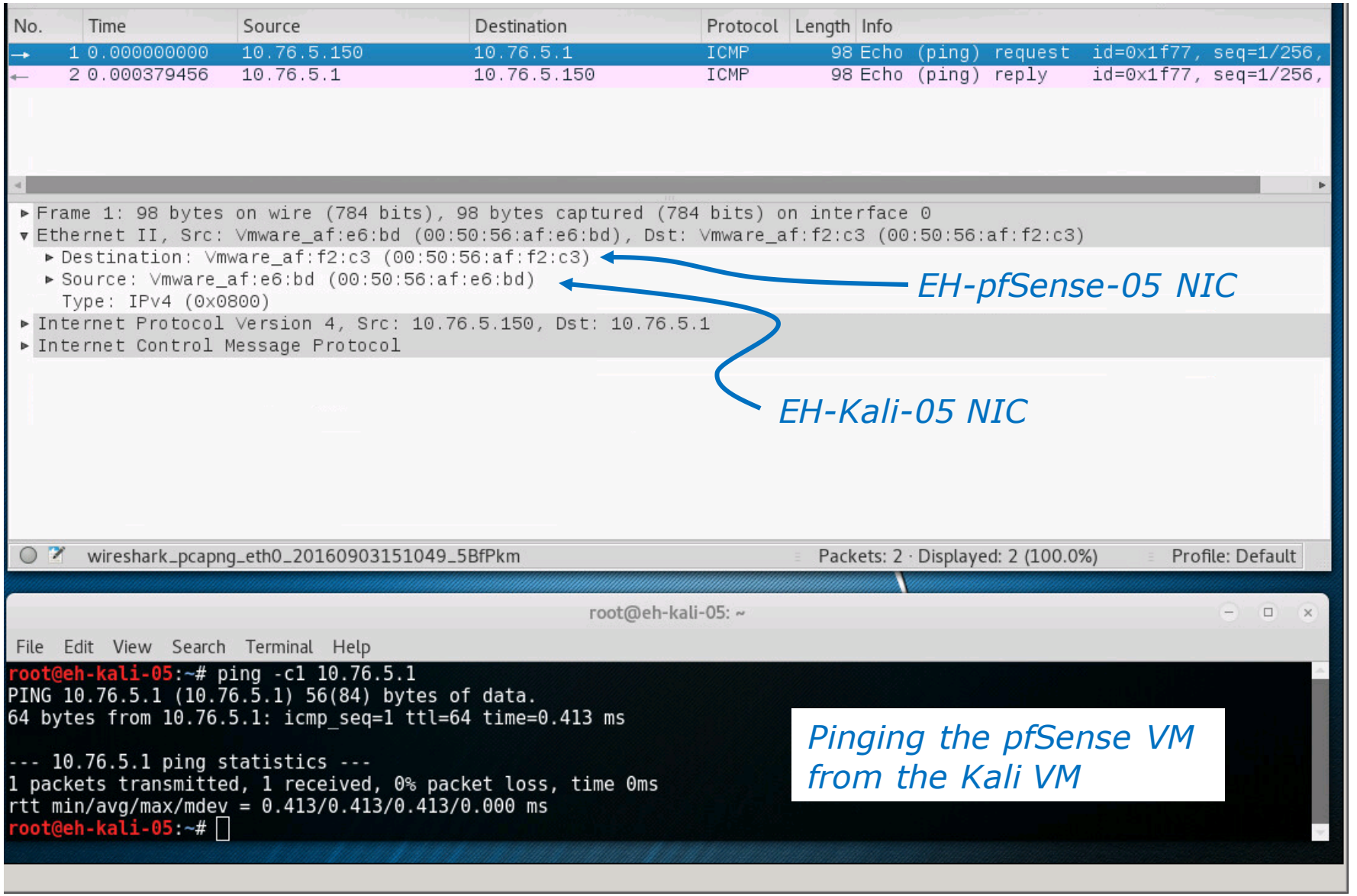
https://www.wireshark.org/tools/oui-lookup.html



There are many MAC Lookup tools available on the Internet to identify the company producing the network device

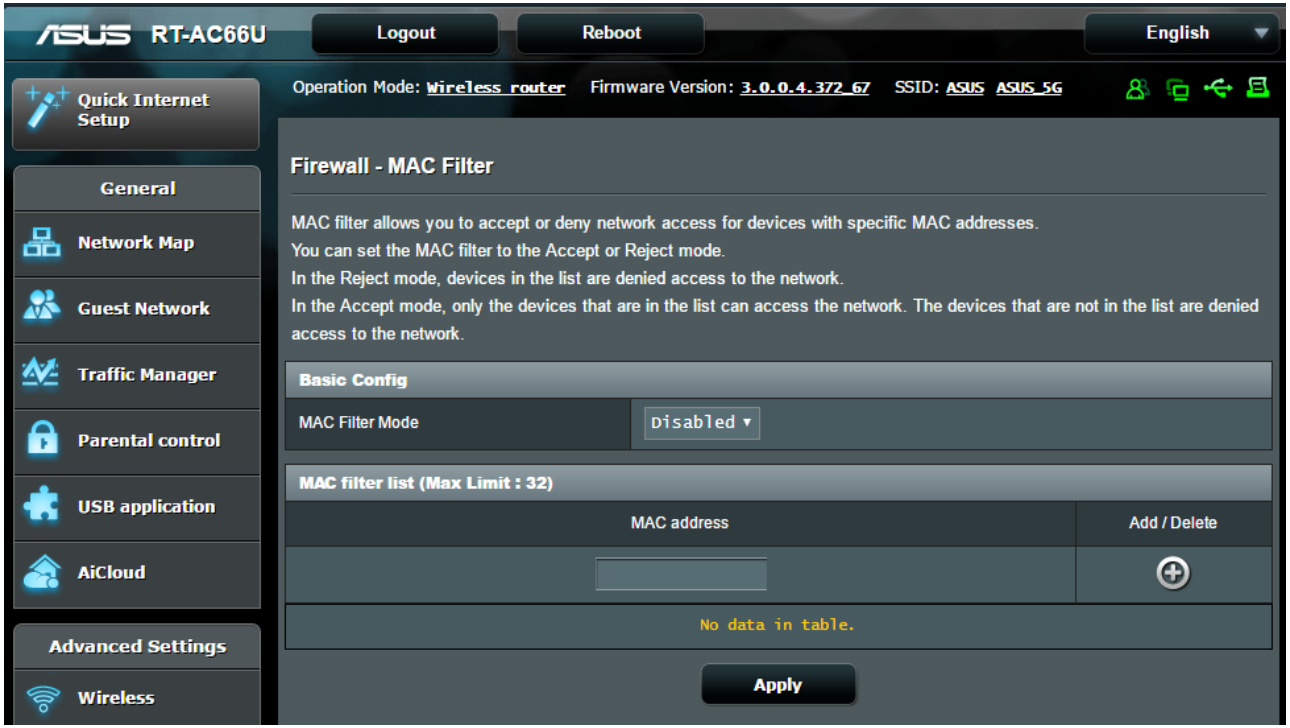
https://www.wireshark.org/tools/oui-lookup.html




No. Time Source Destination Protocol Length Info
1 0.000000000 10.76.5.150 10.76.5.1 ICMP 98 Echo (ping) request id=0x1f77, seq=1/256,
2 0.000379456 10.76.5.1 10.76.5.150 ICMP 98 Echo (ping) reply id=0x1f77, seq=1/256,
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: Vmware_af:e6:bd (00:50:56:af:e6:bd), Dst: Vmware_af:f2:c3 (00:50:56:af:f2:c3)
Destination: Vmware_af:f2:c3 (00:50:56:af:f2:c3)
Source: Vmware_af:e6:bd (00:50:56:af:e6:bd)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.76.5.150, Dst: 10.76.5.1
Internet Control Message Protocol
EH-pfSense-05 NIC
EH-Kali-05 NIC
wireshark_pcapng_eth0_20160903151049_5BfPkm
Packets: 2 · Displayed: 2 (100.0%)
Profile: Default
root@eh-kali-05: ~
File Edit View Search Terminal Help
root@eh-kali-05:~# ping -c1 10.76.5.1
PING 10.76.5.1 (10.76.5.1) 56(84) bytes of data.
64 bytes from 10.76.5.1: icmp_seq=1 ttl=64 time=0.413 ms
--- 10.76.5.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.413/0.413/0.413/0.000 ms
root@eh-kali-05:~#
Pinging the pfSense VM from the Kali VM

Example Mac Address Filtering

ASUS RT-AC66U MAC Filtering

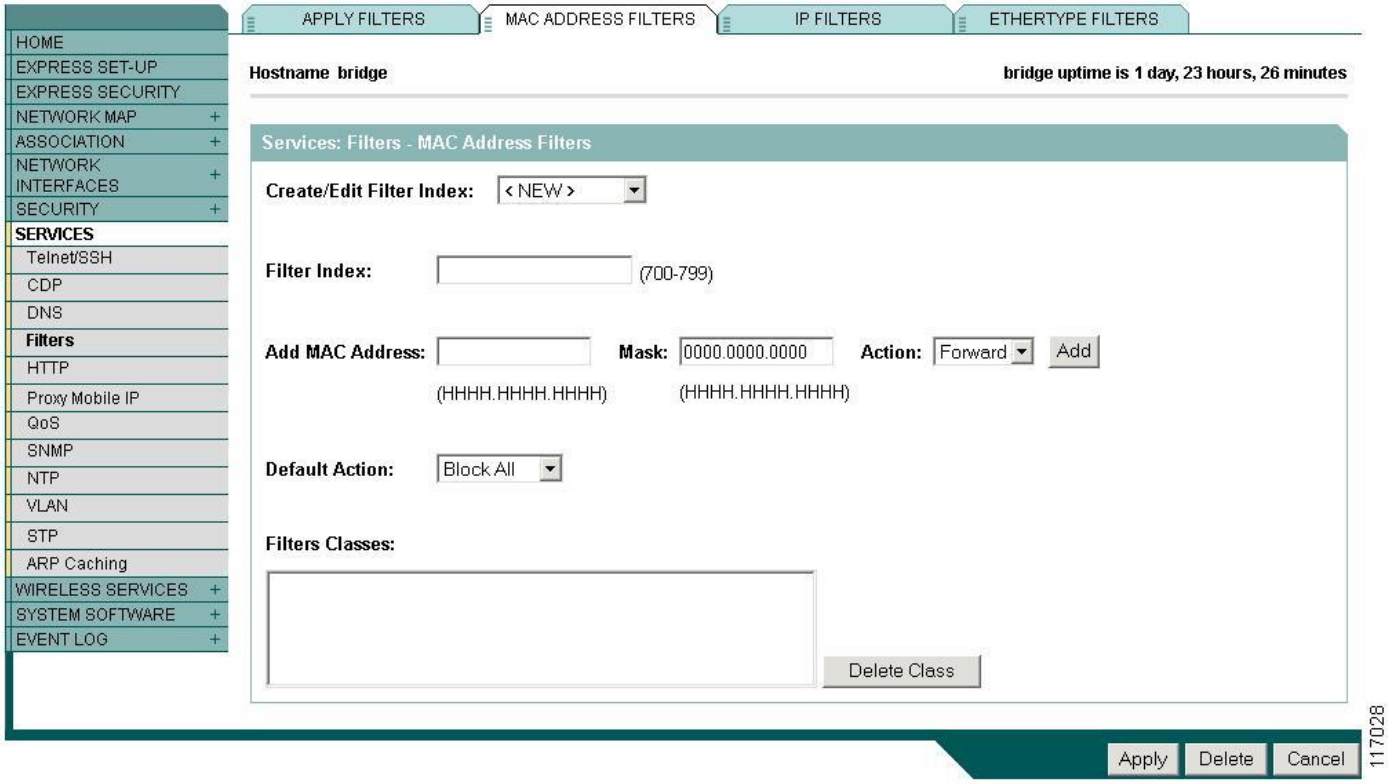


This router enables MAC address filtering to Accept or Reject MAC addresses

http://event.asus.com/2012/nw/dummy_ui/en/Advanced_MACFilter_Content.html

Example Mac Address Filtering

Cisco Aironet 1300 Series Outdoor Access Point



Configuring address filters on a Cisco Access Point

http://www.cisco.com/c/en/us/td/docs/wireless/access_point/1300/12-3_7_JA/configuration/guide/brsc1237/b37filt.html

MAC Address Spoofing

Layer 2 - MAC Address Spoofing

Why would a hacker do this?

- Create an anonymous identity for a network device.
- Impersonate another network device.
- Gain unauthorized access to services.
- Bypass access control lists that allow and block specific MAC addresses.

https://en.wikipedia.org/wiki/MAC_spoofing

Live demo

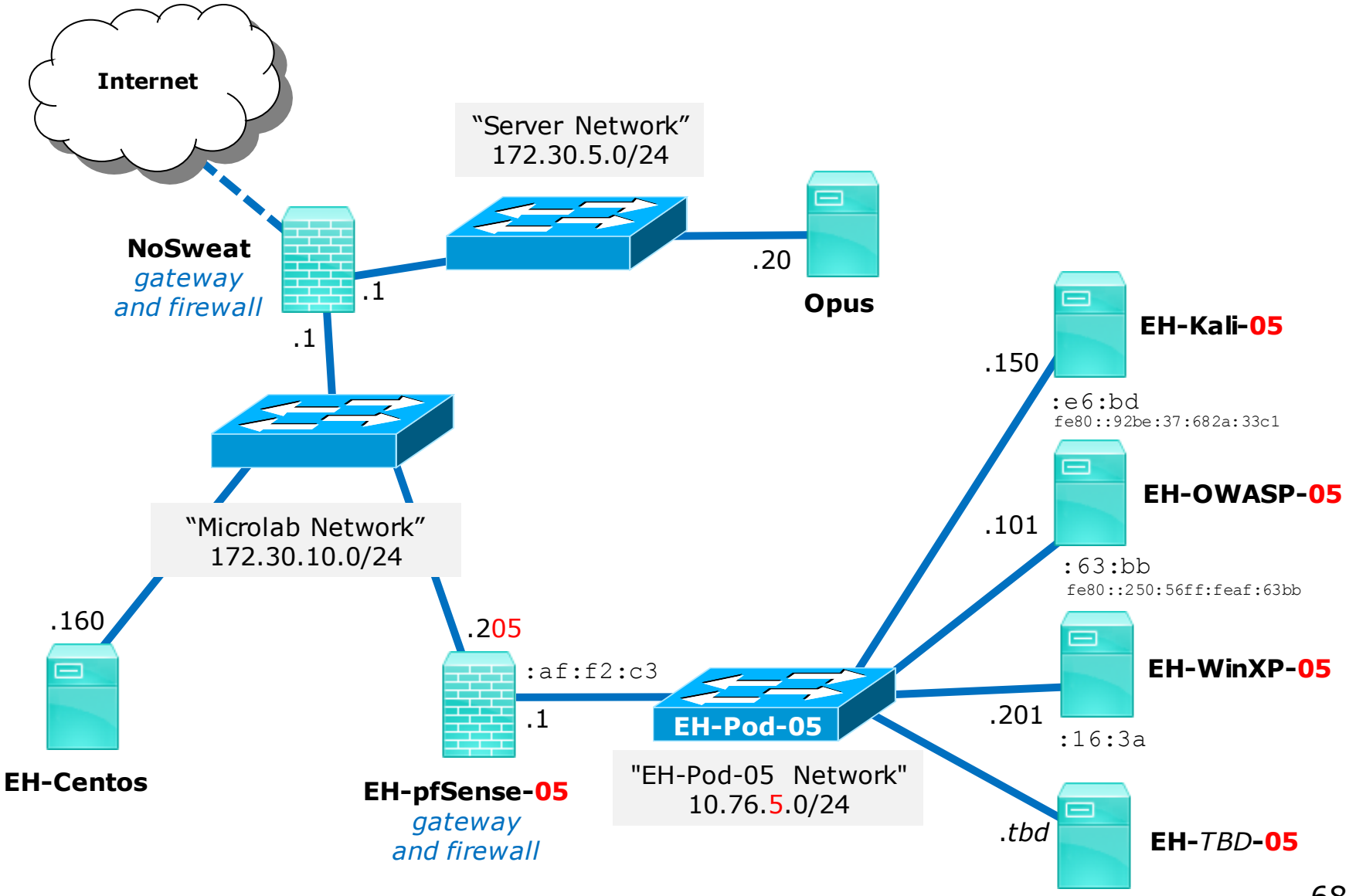
https://simms-teach.com/docs/cis76/cis76-MAC-spoofing.pdf

ARP

ARP - Address Resolution Protocol

- ARP uses layer 2 for transport but unlike IP has no headers and is not routable.
- Before an IP packet can be sent the sender needs to know the MAC address of either:
  - The destination device if it is on the same subnet.
  - The next-hop router if the destination is on a remote network.
- The sender "shouts out" (broadcasts) to the subnet "Who has such and such IP address"
- The IP address owner sends back (unicast) the MAC address.
- The sender can then encapsulate the IP packet into an Ethernet frame and send it to the appropriate MAC address.
- Devices will temporarily save IP/MAC pairs in an arp cache for reuse.
- ARP has been replaced by Neighbor Solicitation & Advertisement in IPv6.

https://keepingitclassless.net/2011/10/neighbor-solicitation-ipv6s-replacement-for-arp/


Internet
"Server Network"
172.30.5.0/24
NoSweat
gateway
and firewall
.1
.1
.20
Opus
.150
EH-Kali-05
:e6:bd
fe80::92be:37:682a:33c1
.101
EH-OWASP-05
:63:bb
fe80::250:56ff:feaf:63bb
.201
EH-WinXP-05
:16:3a
.tbd
EH-TBD-05
"Microlab Network"
172.30.10.0/24
.160
EH-Centos
.205
:af:f2:c3
.1
EH-pfSense-05
gateway
and firewall
EH-Pod-05
"EH-Pod-05 Network"
10.76.5.0/24

ARP Example - getting Kali VM MAC

WinXP VM requests the MAC address of the Kali VM before pinging

0.00000000	Vmware_af:16:3a	Broadcast	ARP	42 Who has 10.76.5.150? Tell 10.76.5.201
0.00029100	Vmware_af:e6:bd	Vmware_af:16:3a	ARP	60 10.76.5.150 is at 00:50:56:af:e6:bd
0.00030700	10.76.5.201	10.76.5.150	ICMP	74 Echo (ping) request id=0x0200, seq=3328/13, ttl=128 (r
0.00049900	10.76.5.150	10.76.5.201	ICMP	74 Echo (ping) reply id=0x0200, seq=3328/13, ttl=64 (re

WinXP Wireshark view

```
C:\WINDOWS\system32\cmd.exe
C:\>arp -a
No ARP Entries Found

C:\>ping 10.76.5.150

Pinging 10.76.5.150 with 32 bytes of data:

Reply from 10.76.5.150: bytes=32 time<1ms TTL=64
Reply from 10.76.5.150: bytes=32 time<1ms TTL=64
Reply from 10.76.5.150: bytes=32 time<1ms TTL=64
Reply from 10.76.5.150: bytes=32 time<1ms TTL=64

Ping statistics for 10.76.5.150:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>arp -a

Interface: 10.76.5.201 --- 0x2
  Internet Address      Physical Address      Type
  10.76.5.150           00-50-56-af-e6-bd     dynamic

C:\>_
```

WinXP command line

Notice the arp cache is populated after the ping operation

ARP Example - getting Kali VM MAC (drill-down)

```
0.00000000 Vmware_af:16:3a     Broadcast          ARP      42 Who has 10.76.5.150?  Tell 10.76.5.201
0.00029100 Vmware_af:e6:bd     Vmware_af:16:3a    ARP      60 10.76.5.150 is at 00:50:56:af:e6:bd
0.00030700 10.76.5.201         10.76.5.150        ICMP     74 Echo (ping) request  id=0x0200, seq=3328/13, ttl=128 (r
0.00049900 10.76.5.150         10.76.5.201        ICMP     74 Echo (ping) reply    id=0x0200, seq=3328/13, ttl=64 (re
```

```
⊞ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
⊟ Ethernet II, Src: Vmware_af:16:3a (00:50:56:af:16:3a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊞ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊞ Source: Vmware_af:16:3a (00:50:56:af:16:3a)
    Type: ARP (0x0806)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Vmware_af:16:3a (00:50:56:af:16:3a)
    Sender IP address: 10.76.5.201 (10.76.5.201)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.76.5.150 (10.76.5.150)
```

Who has 10.76.5.150, tell 10.76.5.201?

```
⊞ Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊟ Ethernet II, Src: Vmware_af:e6:bd (00:50:56:af:e6:bd), Dst: Vmware_af:16:3a (00:50:56:af:16:3a)
  ⊞ Destination: Vmware_af:16:3a (00:50:56:af:16:3a)
  ⊞ Source: Vmware_af:e6:bd (00:50:56:af:e6:bd)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Vmware_af:e6:bd (00:50:56:af:e6:bd)
    Sender IP address: 10.76.5.150 (10.76.5.150)
    Target MAC address: Vmware_af:16:3a (00:50:56:af:16:3a)
    Target IP address: 10.76.5.201 (10.76.5.201)
```

I do, it is at 10.76.5.150 is at 00:50:56:af:e6:bd

ARP Example - getting OWASP VM MAC

WinXP VM requests the MAC address of the OWASP VM before pinging

Time	Source	Destination	Protocol	Length	Info
0.00000000	vmware_af:16:3a	Broadcast	ARP	42	who has 10.76.5.101? Tell 10.76.5.201
0.00037300	vmware_af:63:bb	vmware_af:16:3a	ARP	60	10.76.5.101 is at 00:50:56:af:63:bb
0.00039000	10.76.5.201	10.76.5.101	ICMP	74	Echo (ping) request id=0x0200, seq=4352/17, ttl=128 (r
0.00052400	10.76.5.101	10.76.5.201	ICMP	74	Echo (ping) reply id=0x0200, seq=4352/17, ttl=64 (re

WinXP Wireshark view

```
C:\WINDOWS\system32\cmd.exe
C:\>arp -a
No ARP Entries Found

C:\>ping 10.76.5.101

Pinging 10.76.5.101 with 32 bytes of data:

Reply from 10.76.5.101: bytes=32 time<1ms TTL=64
Reply from 10.76.5.101: bytes=32 time<1ms TTL=64
Reply from 10.76.5.101: bytes=32 time<1ms TTL=64
Reply from 10.76.5.101: bytes=32 time<1ms TTL=64

Ping statistics for 10.76.5.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>arp -a

Interface: 10.76.5.201 --- 0x2
  Internet Address      Physical Address      Type
  10.76.5.101           00-50-56-af-63-bb     dynamic

C:\>_
```

WinXP command line

Notice the arp cache is populated after the ping operation

ARP Example - getting OWASP VM MAC (drill-down)

Time	Source	Destination	Protocol	Length	Info
0.00000000	Vmware_af:16:3a	Broadcast	ARP	42	Who has 10.76.5.101? Tell 10.76.5.201
0.00037300	Vmware_af:63:bb	Vmware_af:16:3a	ARP	60	10.76.5.101 is at 00:50:56:af:63:bb
0.00039000	10.76.5.201	10.76.5.101	ICMP	74	Echo (ping) request id=0x0200, seq=4352/17, ttl=128 (r
0.00052400	10.76.5.101	10.76.5.201	ICMP	74	Echo (ping) reply id=0x0200, seq=4352/17, ttl=64 (re

```
⊞ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
⊟ Ethernet II, Src: Vmware_af:16:3a (00:50:56:af:16:3a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊞ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊞ Source: Vmware_af:16:3a (00:50:56:af:16:3a)
    Type: ARP (0x0806)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Vmware_af:16:3a (00:50:56:af:16:3a)
    Sender IP address: 10.76.5.201 (10.76.5.201)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.76.5.101 (10.76.5.101)
```

Who has 10.76.5.101, tell 10.76.5.201?

```
⊞ Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊟ Ethernet II, Src: Vmware_af:63:bb (00:50:56:af:63:bb), Dst: Vmware_af:16:3a (00:50:56:af:16:3a)
  ⊞ Destination: Vmware_af:16:3a (00:50:56:af:16:3a)
  ⊞ Source: Vmware_af:63:bb (00:50:56:af:63:bb)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Vmware_af:63:bb (00:50:56:af:63:bb)
    Sender IP address: 10.76.5.101 (10.76.5.101)
    Target MAC address: Vmware_af:16:3a (00:50:56:af:16:3a)
    Target IP address: 10.76.5.201 (10.76.5.201)
```

I do, it is at 10.76.5.101 is at 00:50:56:af:63:bb

ICMPv6 Neighbor Solicitation Example

Kali getting OWASP VM MAC

Kali Wireshark view

No.	Time	Source	Destination	Protocol	Leng	Info
2	60.048792053	fe80::92be:37:6…	ff02::1:ffaf:63bb	ICMPv6	86	Neighbor Solicitation for fe80::250:56ff:f…
3	60.049136713	fe80::250:56ff:…	fe80::92be:37:68…	ICMPv6	86	Neighbor Advertisement fe80::250:56ff:feaf…
4	60.049155306	fe80::92be:37:6…	fe80::250:56ff:f…	ICMPv6	118	Echo (ping) request id=0x5691, seq=1, hop …
5	60.049331414	fe80::250:56ff:…	fe80::92be:37:68…	ICMPv6	118	Echo (ping) reply id=0x5691, seq=1, hop li…
6	61.049723000	fe80::92be:37:6…	fe80::250:56ff:f…	ICMPv6	118	Echo (ping) request id=0x5691, seq=2, hop …
7	61.049953479	fe80::250:56ff:…	fe80::92be:37:68…	ICMPv6	118	Echo (ping) reply id=0x5691, seq=2, hop li…

```
▸ Frame 2: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▾ Ethernet II, Src: Vmware_af:e6:bd (00:50:56:af:e6:bd), Dst: IPv6mcast_ff:af:63:bb (33:33:ff:af:63:bb)
  ▸ Destination: IPv6mcast_ff:af:63:bb (33:33:ff:af:63:bb)
  ▸ Source: Vmware_af:e6:bd (00:50:56:af:e6:bd)
    Type: IPv6 (0x86dd)
▾ Internet Protocol Version 6, Src: fe80::92be:37:682a:33c1, Dst: ff02::1:ffaf:63bb
    0110 .... = Version: 6
  ▸ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::92be:37:682a:33c1
    Destination: ff02::1:ffaf:63bb
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▾ Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0xefd9 [correct]
    Reserved: 00000000
    Target Address: fe80::250:56ff:feaf:63bb
  ▸ ICMPv6 Option (Source link-layer address : 00:50:56:af:e6:bd)
```

```
root@eh-kali-05:~# ping6 -c2  fe80::250:56ff:feaf:63bb
PING fe80::250:56ff:feaf:63bb(fe80::250:56ff:feaf:63bb) 56 data bytes
64 bytes from fe80::250:56ff:feaf:63bb%eth0: icmp_seq=1 ttl=64 time=0.233 ms
64 bytes from fe80::250:56ff:feaf:63bb%eth0: icmp_seq=2 ttl=64 time=0.402 ms

--- fe80::250:56ff:feaf:63bb ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.233/0.317/0.402/0.086 ms
root@eh-kali-05:~# ip -6 neighbor show
fe80::250:56ff:feaf:63bb dev eth0 lladdr 00:50:56:af:63:bb REACHABLE
root@eh-kali-05:~#
```

Kali command line

Notice the multicast solicitation is asking for the MAC address of the OWASP VM

ICMPv6 Neighbor Advertisement Example

Kali getting OWASP VM MAC

Kali Wireshark view

No.	Time	Source	Destination	Protocol	Leng	Info
2	60.048792053	fe80::92be:37:6…	ff02::1:ffaf:63bb	ICMPv6	86	Neighbor Solicitation for fe80::250:56ff:f…
3	60.049136713	fe80::250:56ff:…	fe80::92be:37:68…	ICMPv6	86	Neighbor Advertisement fe80::250:56ff:feaf…
4	60.049155306	fe80::92be:37:6…	fe80::250:56ff:f…	ICMPv6	118	Echo (ping) request id=0x5691, seq=1, hop …
5	60.049331414	fe80::250:56ff:…	fe80::92be:37:68…	ICMPv6	118	Echo (ping) reply id=0x5691, seq=1, hop li…
6	61.049723000	fe80::92be:37:6…	fe80::250:56ff:f…	ICMPv6	118	Echo (ping) request id=0x5691, seq=2, hop …
7	61.049953479	fe80::250:56ff:…	fe80::92be:37:68…	ICMPv6	118	Echo (ping) reply id=0x5691, seq=2, hop li…

```
▸ Frame 3: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▾ Ethernet II, Src: Vmware_af:63:bb (00:50:56:af:63:bb), Dst: Vmware_af:e6:bd (00:50:56:af:e6:bd)
  ▸ Destination: Vmware_af:e6:bd (00:50:56:af:e6:bd)
  ▸ Source: Vmware_af:63:bb (00:50:56:af:63:bb)
    Type: IPv6 (0x86dd)
▾ Internet Protocol Version 6, Src: fe80::250:56ff:feaf:63bb, Dst: fe80::92be:37:682a:33c1
    0110 .... = Version: 6
  ▸ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::250:56ff:feaf:63bb
    [Source SA MAC: Vmware_af:63:bb (00:50:5
    Destination: fe80::92be:37:682a:33c1
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▾ Internet Control Message Protocol v6
    Type: Neighbor Advertisement (136)
    Code: 0
    Checksum: 0xb90f [correct]
  ▸ Flags: 0x60000000
    Target Address: fe80::250:56ff:feaf:63bb
  ▸ ICMPv6 Option (Target link-layer address : 00:50:56:af:63:bb)
```

```
root@eh-kali-05:~# ping6 -c2  fe80::250:56ff:feaf:63bb
PING fe80::250:56ff:feaf:63bb(fe80::250:56ff:feaf:63bb) 56 data bytes
64 bytes from fe80::250:56ff:feaf:63bb%eth0: icmp_seq=1 ttl=64 time=0.233 ms
64 b                                                                    ms
--- fe80::250:56ff:feaf:63bb ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.233/0.317/0.402/0.086 ms
root@eh-kali-05:~# ip -6 neighbor show
fe80::250:56ff:feaf:63bb dev eth0 lladdr 00:50:56:af:63:bb REACHABLE
root@eh-kali-05:~#
```

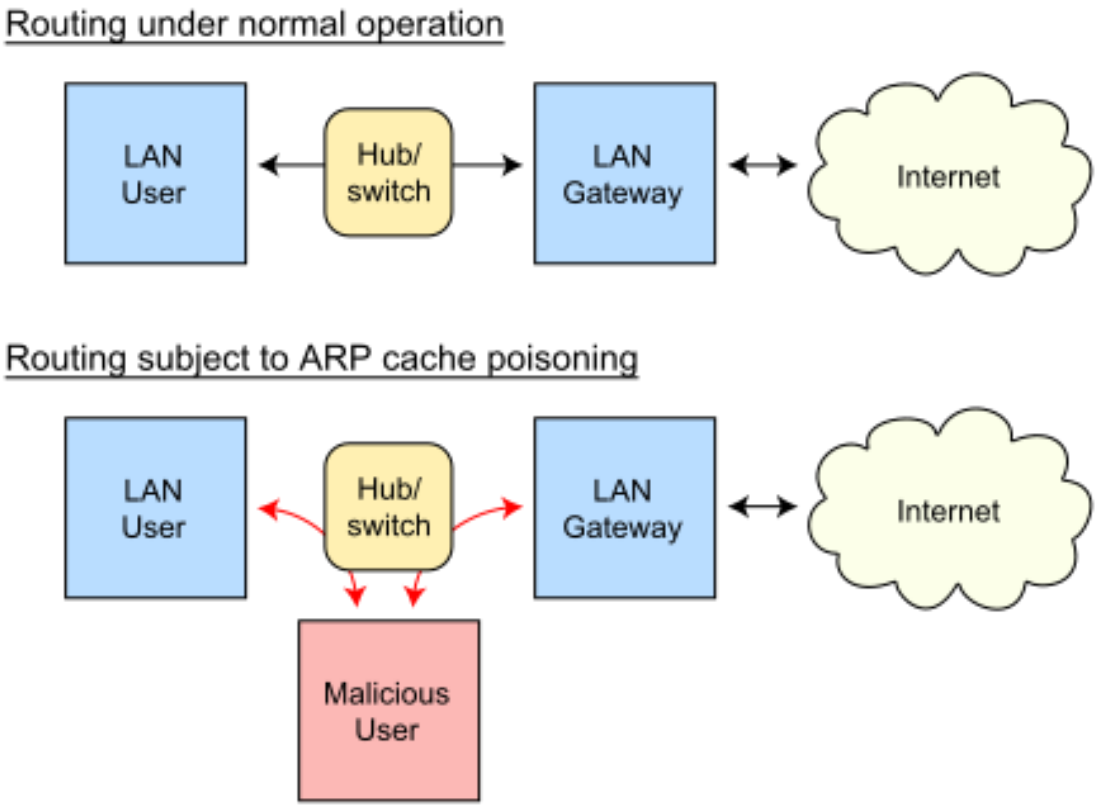
Notice the neighbor list on Kali is populated now

Kali command line

Notice the advertisement contains the OWASP MAC address

MITM attack using ARP Poisoning

Background on ARP Spoofing



Source: By 0x55534C - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=15034709

ARP Spoofing

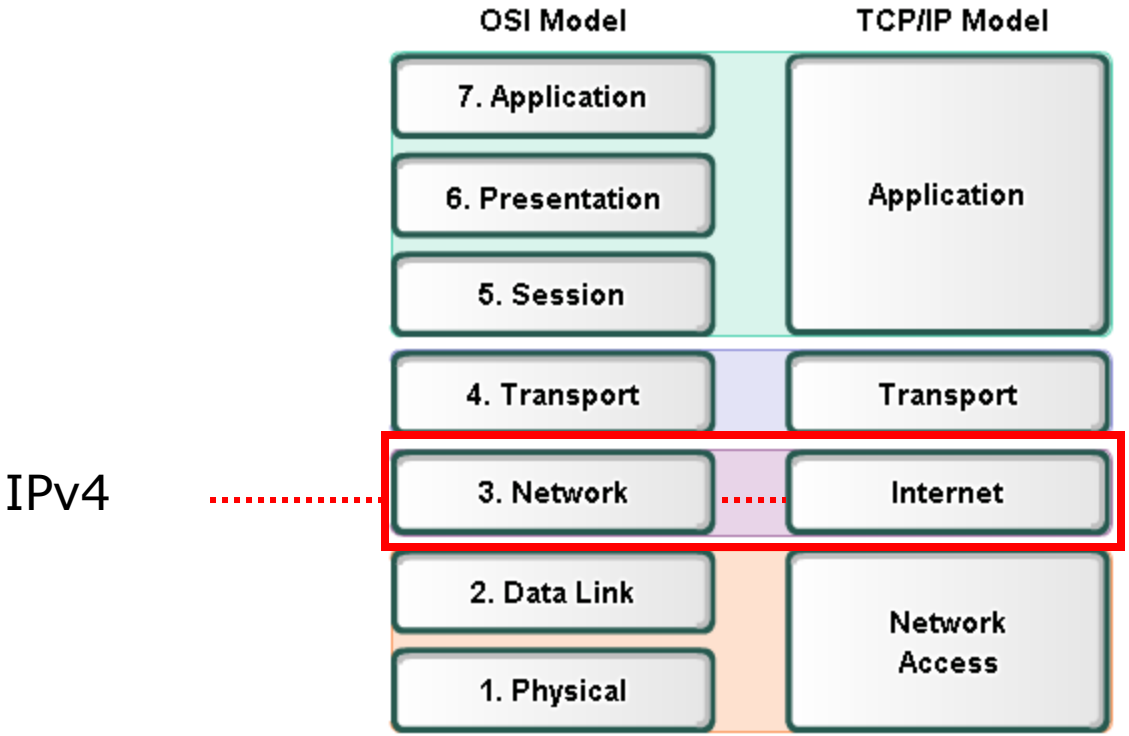
https://https://en.wikipedia.org/wiki/ARP_spoofing *Wiki article on ARP spoofing*

Live demo

https://simms-teach.com/docs/cis76/cis76-MITM-arp-poison.pdf

Network Layer

Network Layer

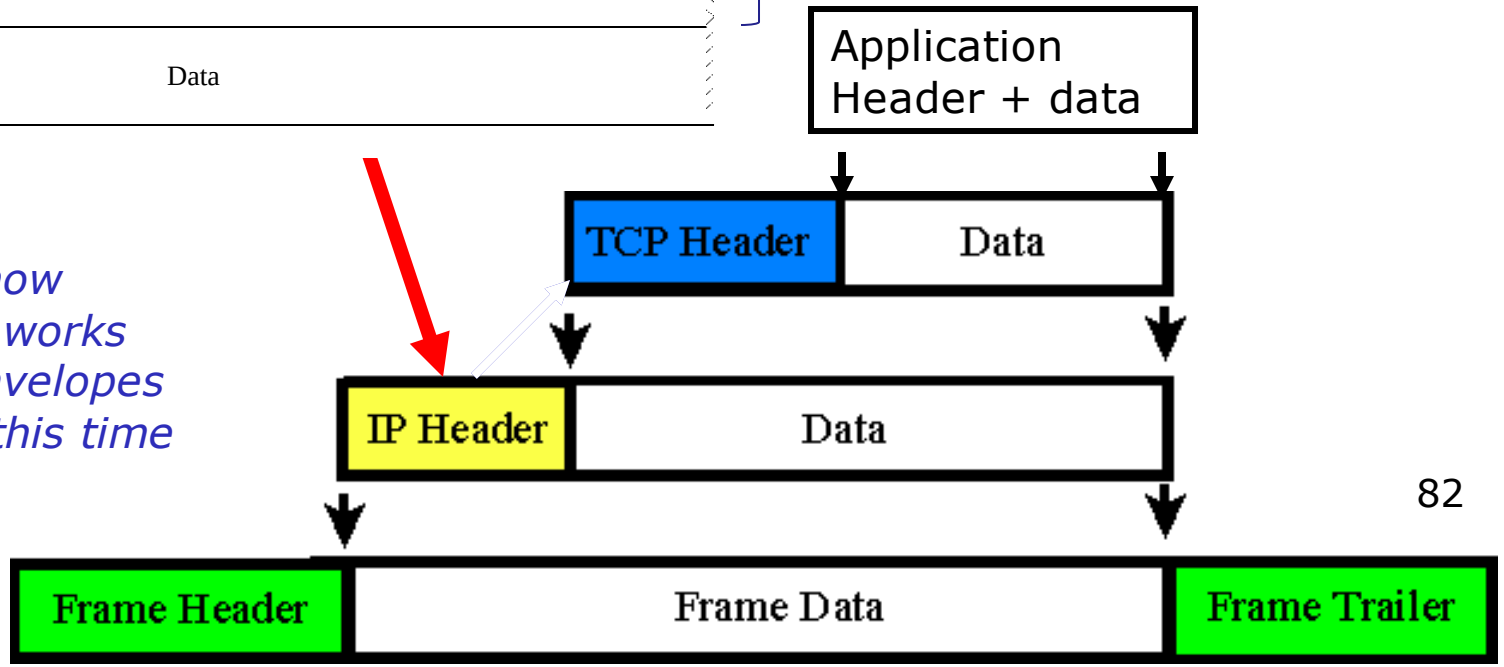


RS: More on Layer 3 tonight

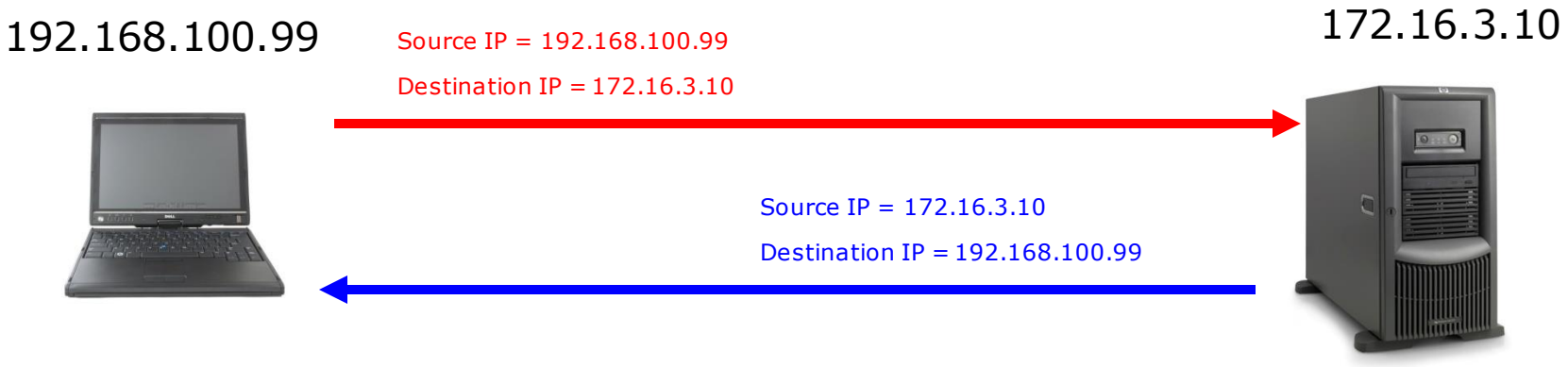
0		15	16	31
4-bit Version	4-bit Header Length	8-bit Type Of Service (TOS)	16-bit Total Length (in bytes)	
16-bit Identification			3-bit Flags	13-bit Fragment Offset
8 bit Time To Live TTL		8-bit Protocol	16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Data				

IP Header

RS: showing how encapsulation works without the envelopes and postman this time



Addressing



- Source IP Address
- Destination IP Address

- More later!

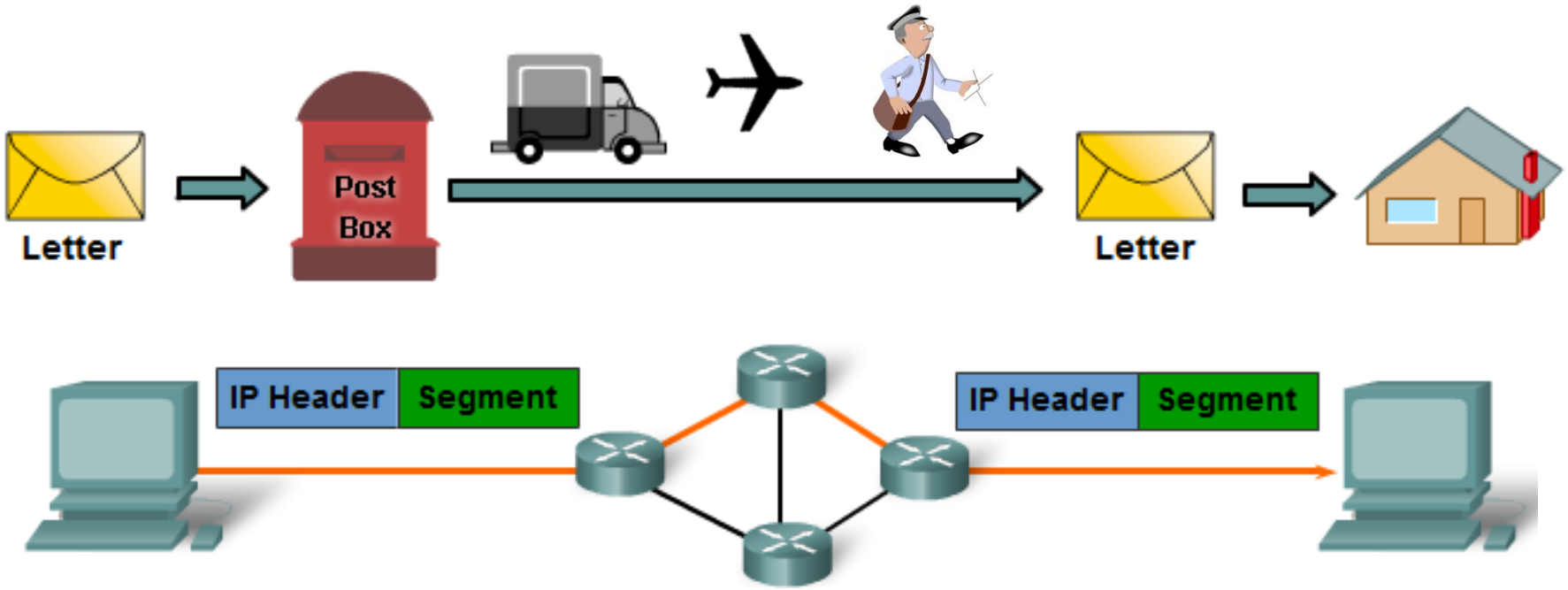
RS: Layer 3 is where IP addresses are used. They are put in the header of the layer three packets.

Network Layer Protocols

- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk
- Connectionless Network Service (CLNS/DECNet)

- The Internet Protocol (IPv4 and IPv6) is the most widely-used Layer 3 data carrying protocol and will be the focus of this course.

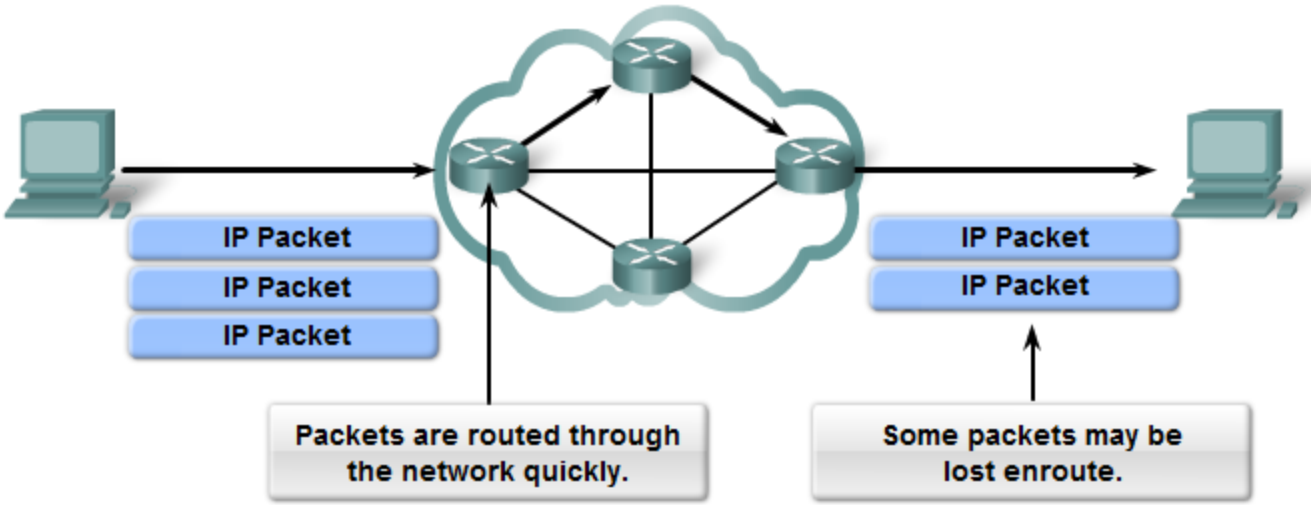
Connectionless



IP packets are sent without notifying the end host that they are coming. *(Layer 3)*

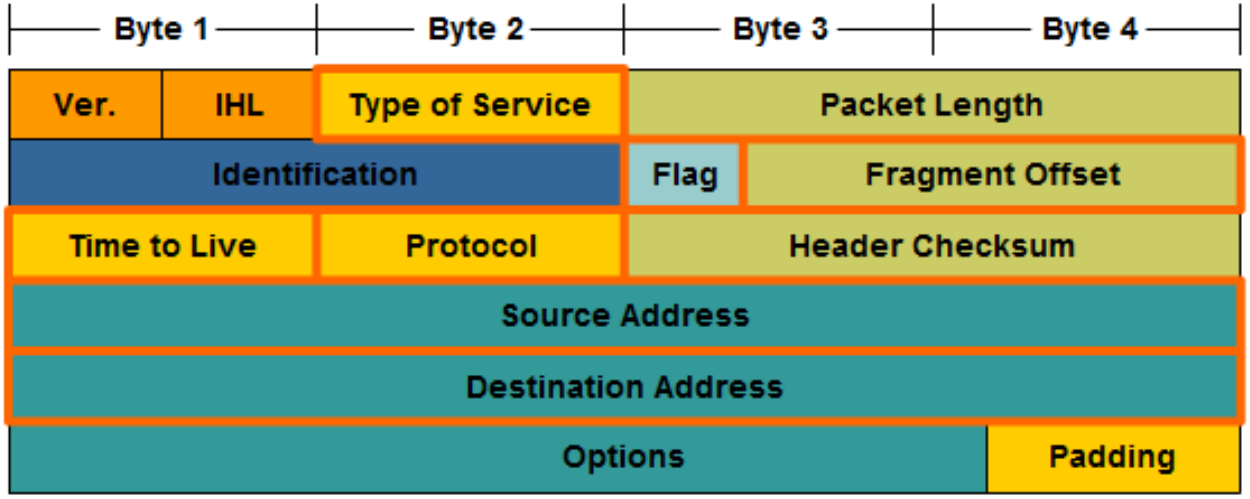
- **TCP**: A connection-oriented protocol does require a connection to be established prior to sending TCP segments. *(Layer 4)*
- **UDP**: A connectionless protocol does not require a session to be established. *(Layer 4)*

Best Effort Service (unreliable)



- The mission of Layer 3 is to transport the packets between the hosts while placing as little burden on the network as possible.
  - Speed over reliability
- Layer 3 is not concerned with or even aware of the type of data contained inside of a packet.
  - This responsibility is the role of the upper layers as required.
- **Unreliable**: IP does not have the capability or responsibility to manage or recover from, undelivered or corrupt packets.
  - TCP's responsibility at the end-to-end hosts

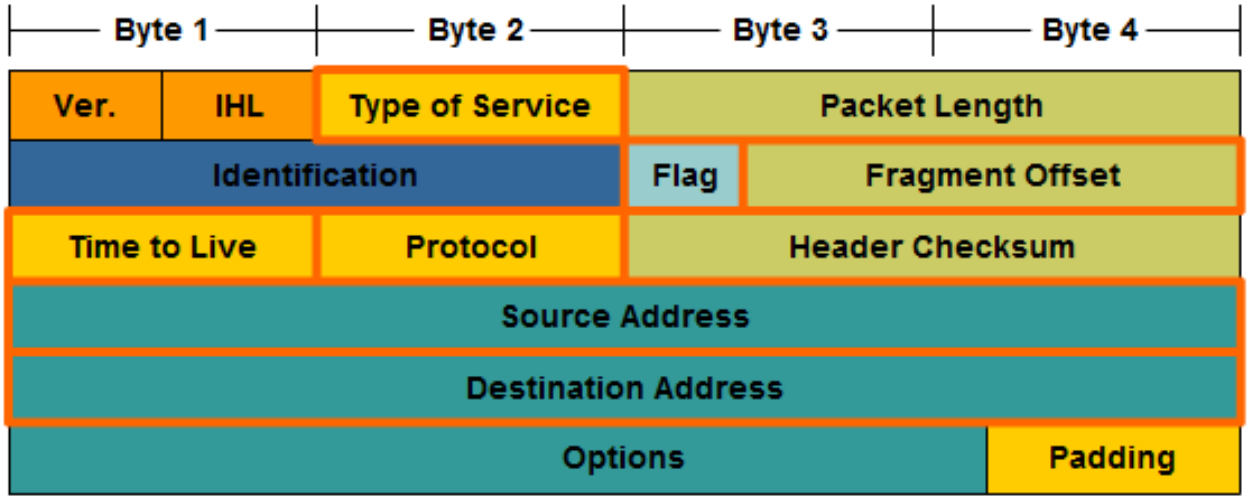
IP Header



- **IP Destination Address**
  - 32-bit binary value that represents the packet destination Network layer host address.
- **IP Source Address**
  - 32-bit binary value that represents the packet source Network layer host address.

IP's TTL - Time To Live field

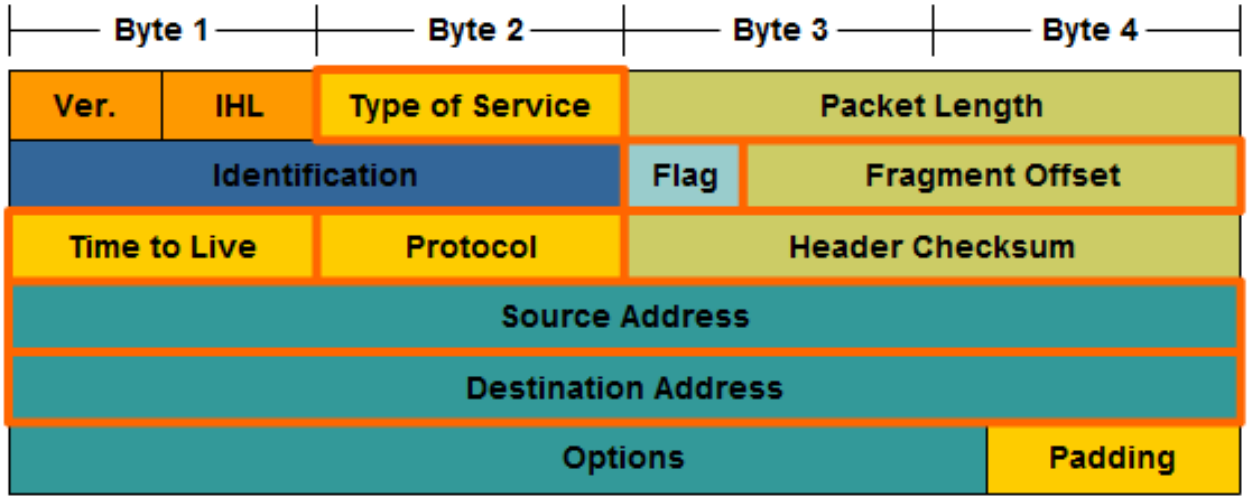
Decrement by 1, if 0 drop the packet.



- If the router decrements the TTL field to 0, it will then drop the packet (unless the packet is destined specifically for the router, i.e. ping, telnet, etc.).
- Common operating system TTL values are:
  - UNIX: **255**
  - Linux: **64 or 255** depending upon vendor and version
  - Microsoft Windows 95: **32**
  - Other Microsoft Windows operating systems: **128**

IP's TTL - Time To Live field

Decrement by 1, if 0 drop the packet.

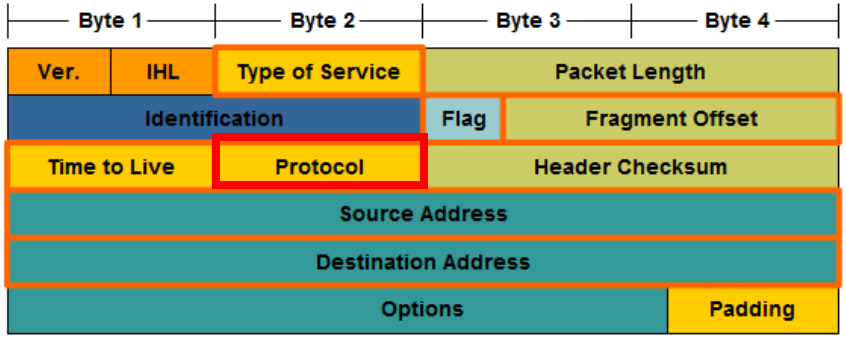


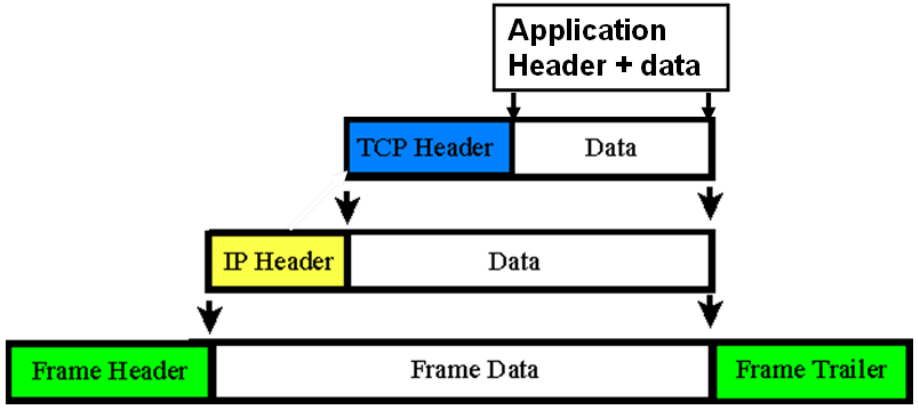
- The idea behind the TTL field is that IP packets can not travel around the Internet forever, from router to router.

- Eventually, the packet's TTL which reach 0 and be dropped by the router, even if there is a routing loop somewhere in the network.

RS: TTL errors are used by traceroute and mtr to discover the path a packet takes

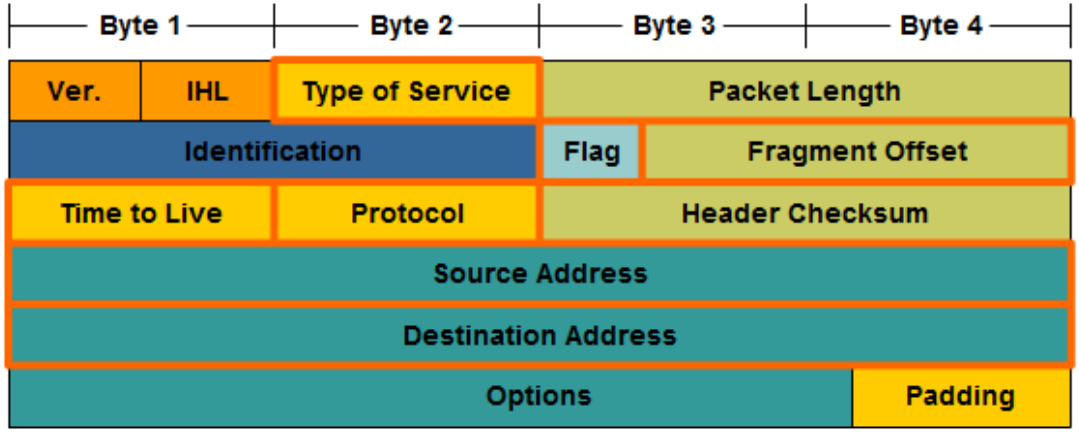
IP's Protocol Field





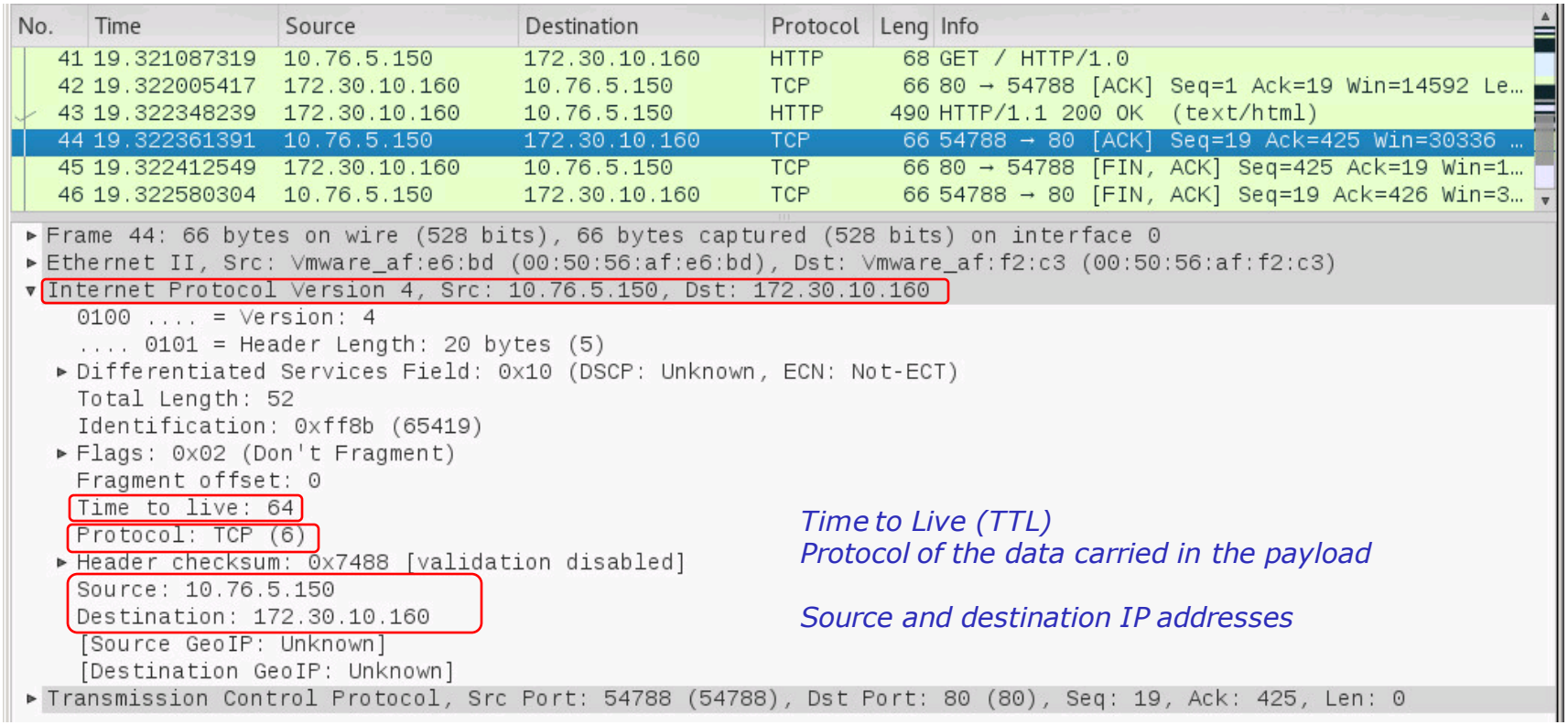
- **Protocol field** enables the Network layer to pass the data to the appropriate upper-layer protocol.
- Example values are:
  - 01 ICMP
  - 06 TCP
  - 17 UDP

Other IPv4 fields



- **Version** - Contains the IP version number (4)
- **Header Length (IHL)** - Specifies the size of the packet header.
- Packet Length - This field gives the entire packet size, including header and data, in bytes.
- **Identification** - This field is primarily used for uniquely identifying fragments of an original IP packet
- **Header Checksum** - The checksum field is used for error checking the packet header.
- **Options** - There is provision for additional fields in the IPv4 header to provide other services but these are rarely used.

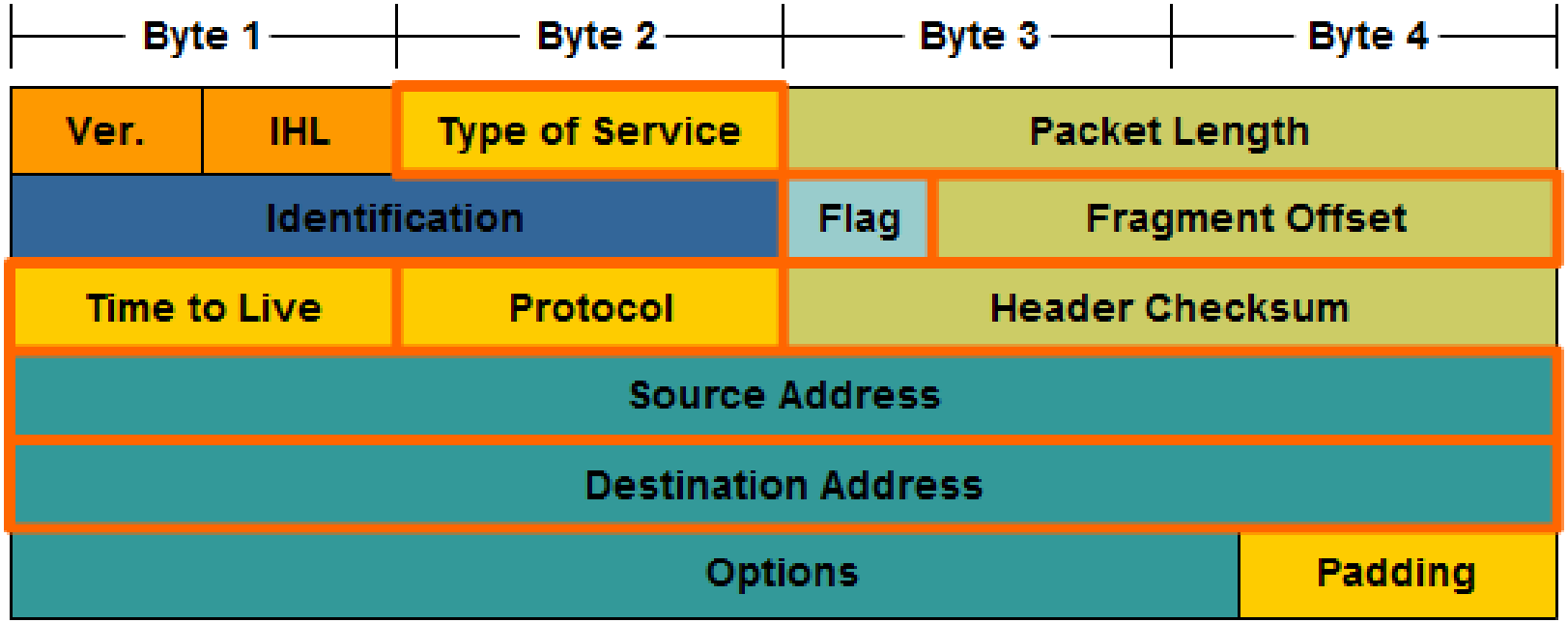
Viewing Layer 3 information with Wireshark



Traffic between EH-Centos VM and EH-Kali VM

IPv4 addressing & subnetting

IPv4 Addresses



- IPv4 addresses are 32 bit addresses

IPv4 Addresses

- IPv4 Addresses are 32 bit addresses:

10101001110001110100011011000100

10101001 11000111 01000101 10001001

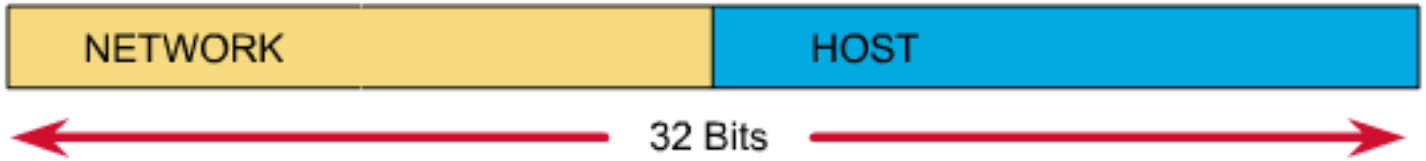
- We use dotted notation (or dotted decimal notation) to represent the value of each byte (octet) of the IP address in decimal.

10101001 11000111 01000101 10001001
169 . 199 . 69 . 137

IPv4 Addresses

An IP address has two parts:

- **network number**
- **host number**



Which bits refer to the network number?

Which bits refer to the host number?

IPv4 Addresses

Answer:

- Newer technology - **Classless IP Addressing**
  - The **subnet mask** determines the network portion and the host portion.
  - Value of first octet does NOT matter (older classful IP addressing)
  - Hosts and Classless Inter-Domain Routing (**CIDR**).
  - Classless IP Addressing is what is used within the Internet and in most internal networks.

- Older technology - **Classful IP Addressing**
  - **Value of first octet** determines the network portion and the host portion.
  - Used with classful routing protocols like RIPv1.
  - The Cisco IP Routing Table is structured in a classful manner (CIS 82)

Network Addresses have all 0's in the host portion.

Network Address

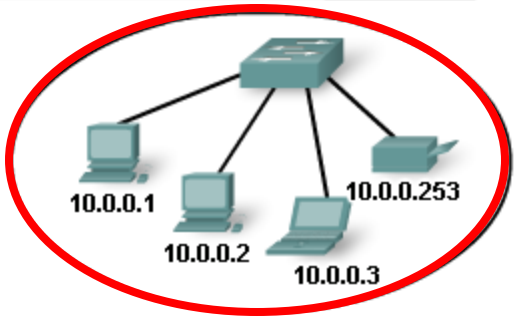
Broadcast Address

Host Address

Roll over to learn more.

	Network		Host
10	0	0	0
00001010	00000000	00000000	00000000
10	0	0	255
00001010	00000000	00000000	11111111
10	0	0	1
00001010	00000000	00000000	00000001

Subnet Mask: 255.255.255.0



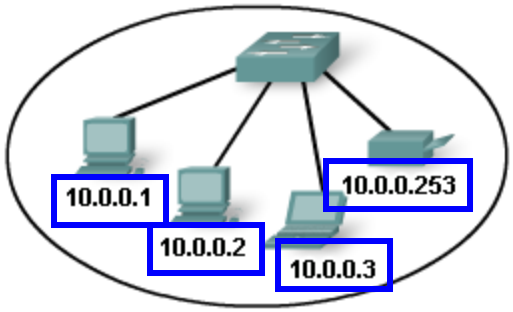
- **Network address** - The address by which we refer to the network
- **Broadcast address** - A special address used to send data to all hosts in the network
- **Host addresses** - The addresses assigned to the end devices in the network

Broadcast Addresses have all 1's in the host portion.

	Network			Host
Network Address	10	0	0	0
	00001010	00000000	00000000	00000000
Broadcast Address	10	0	0	255
	00001010	00000000	00000000	11111111
Host Address	10	0	0	1
	00001010	00000000	00000000	00000001

Roll over to learn more.

Subnet Mask: 255.255.255.0



- **Network address** - The address by which we refer to the network
- **Broadcast address** - A special address used to send data to all hosts in the network
- **Host addresses** - The addresses assigned to the end devices in the network

Types of Addresses

Network Address

Broadcast Address

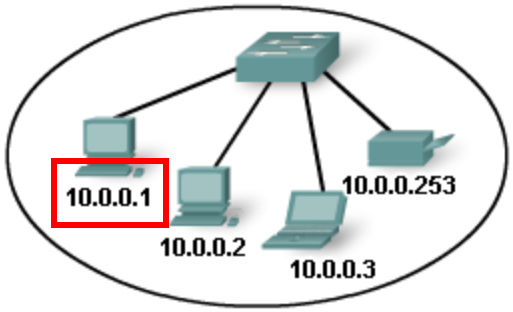
Host Address

Roll over to learn more.

	Network		Host
10	0	0	0
00001010	00000000	00000000	00000000
10	0	0	255
00001010	00000000	00000000	11111111
10	0	0	1
00001010	00000000	00000000	00000001

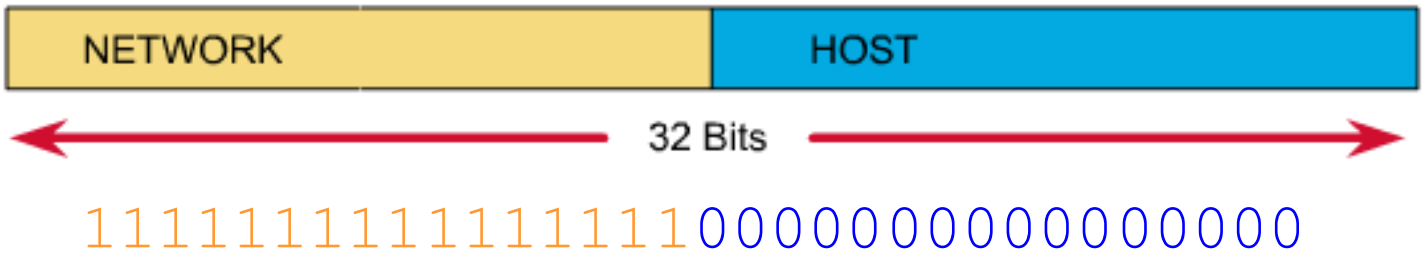
Host Addresses can not have all 0's or all 1's in the host portion.

Subnet Mask: 255.255.255.0



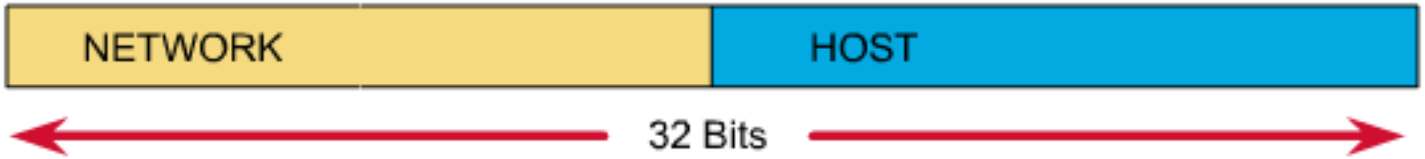
- **Network address** - The address by which we refer to the network
- **Broadcast address** - A special address used to send data to all hosts in the network
- **Host addresses** - The addresses assigned to the end devices in the network

Dividing the Network and Host Portions



- **Subnet Mask**
  - Used to define the:
    - Network portion
    - Host portion
  - 32 bits
  - Contiguous set of 1's followed by a contiguous set of 0's
    - 1's: Network portion
    - 0's: Host portion

Dividing the Network and Host Portions



11111111.11111111.00000000.00000000

Dotted decimal: 255 . 255 . 0 . 0

Slash notation: /16

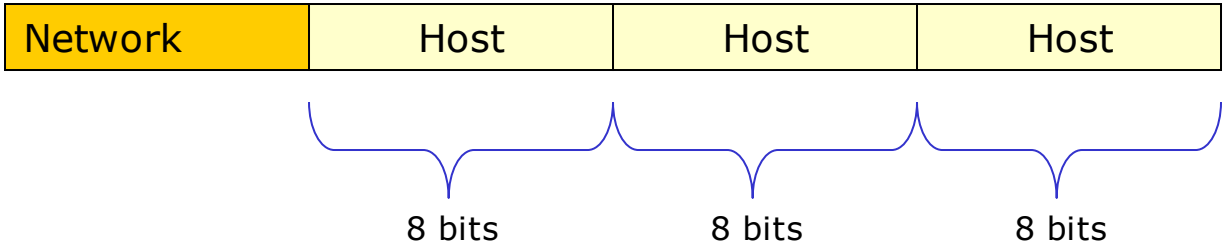
- Subnet mask expressed as:
  - Dotted decimal
    - Ex: 255.255.0.0
  - Slash notation or prefix length
    - /16 (the number of one bits)

Why the mask matters: Number of hosts!

Subnet Mask:	1st octet	2nd octet	3rd octet	4th octet
255.0.0.0 or /8	Network	Host	Host	Host
255.255.0.0 or /16	Network	Network	Host	Host
255.255.255.0 or /24	Network	Network	Network	Host

- The more host bits in the subnet mask means the more hosts in the network.
- Subnet masks do not have to end on "natural octet boundaries"

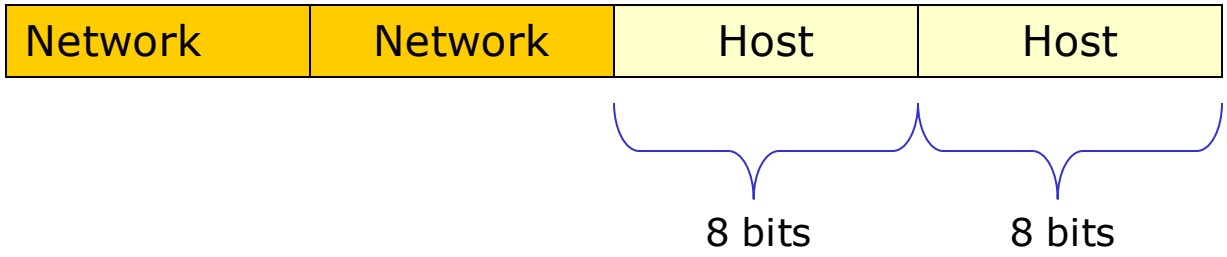
Subnet: 255.0.0.0 (/8)



With 24 bits available for hosts, there a 2^{24} possible addresses. That's 16,777,216 nodes!

- Only large organizations such as the military, government agencies, universities, and large corporations have networks with these many addresses.
- Example: A certain cable modem ISP has 24.0.0.0 and a DSL ISP has 63.0.0.0

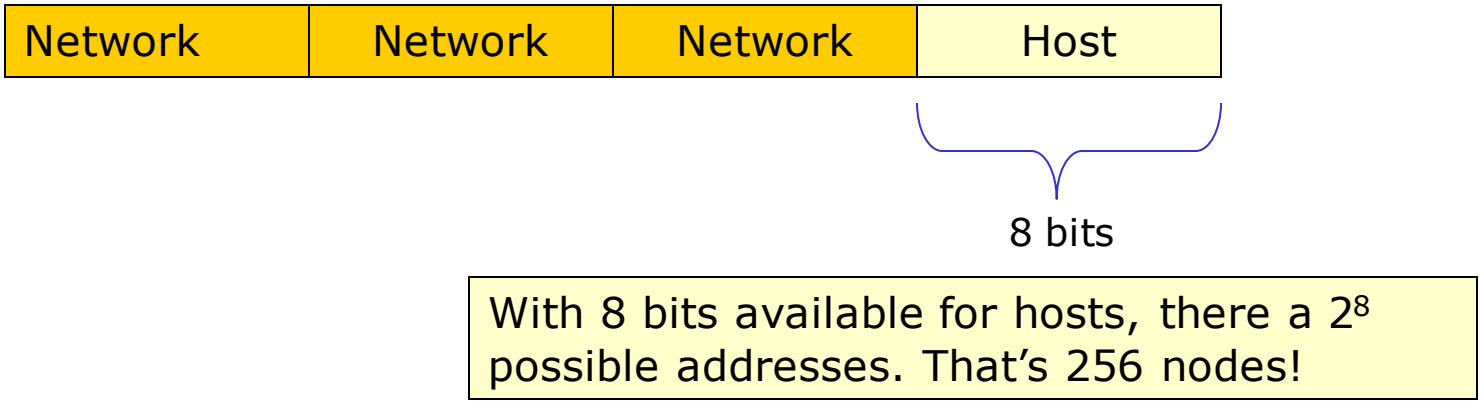
Subnet: 255.255.0.0 (/16)



With 16 bits available for hosts, there a 2^{16} possible addresses. That's 65,536 nodes!

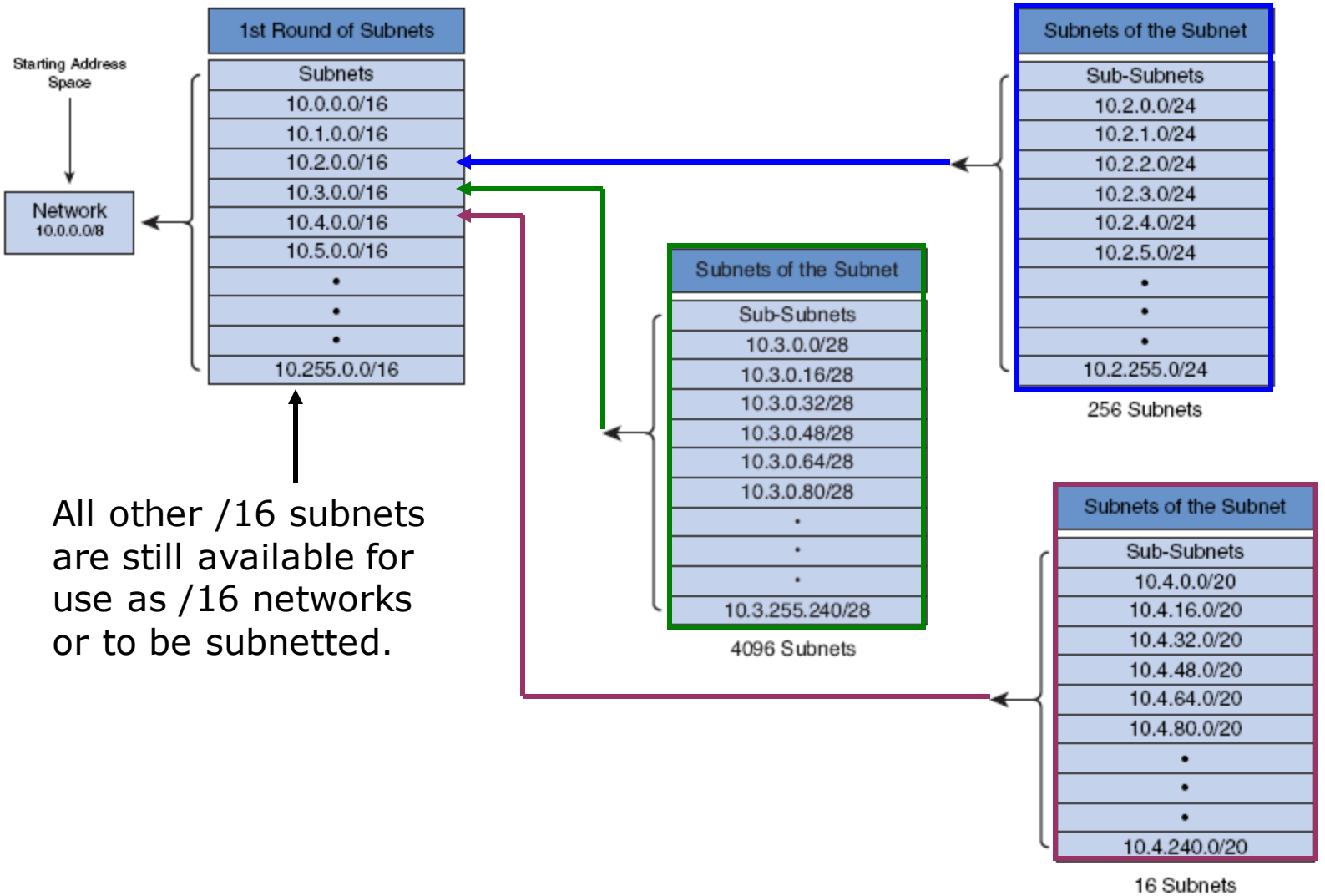
- 65,534 host addresses, one for network address and one for broadcast address.

Subnet: 255.255.255.0 (/24)



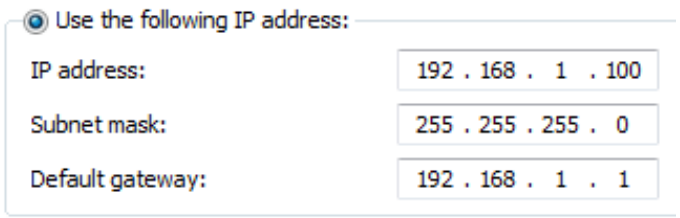
- 254 host addresses, one for network address and one for broadcast address.

VLSM - Variable Length Subnet Masks
Subnet a subnet



All other /16 subnets are still available for use as /16 networks or to be subnetted.

Special Unicast IPv4 Addresses



- **Default Route**

- **Loopback Address**
  - Special address that hosts use to direct traffic to themselves.
  - 127.0.0.0 to 127.255.255.255

- **Link-Local Addresses (APIPA)**
  - 169.254.0.0 to 169.254.255.255 (169.254.0.0 /16)
  - Can be automatically assigned to the local host by the operating system in environments where no IP configuration is available.
  - Microsoft calls this APIPA (Automatic Private IP Addressing)

- **TEST-NET Addresses**
  - 192.0.2.0 to 192.0.2.255 (192.0.2.0 /24)
  - Set aside for teaching and learning purposes.
  - These addresses can be used in documentation and network examples.

1.2 Introducing IPv6

Introducing IPv6

- Not a "new" protocol.
- Developed mid to late 1990s.
- Much learned from IPv4.
- 128-bit address space, written in hexadecimal.
- This gives us 340 undecillion addresses!

2001:DB8:CAFE:0001::100

340 undecillion
= 340,282,366,920,938,463,463,374,607,431,768,211,456

IPv6

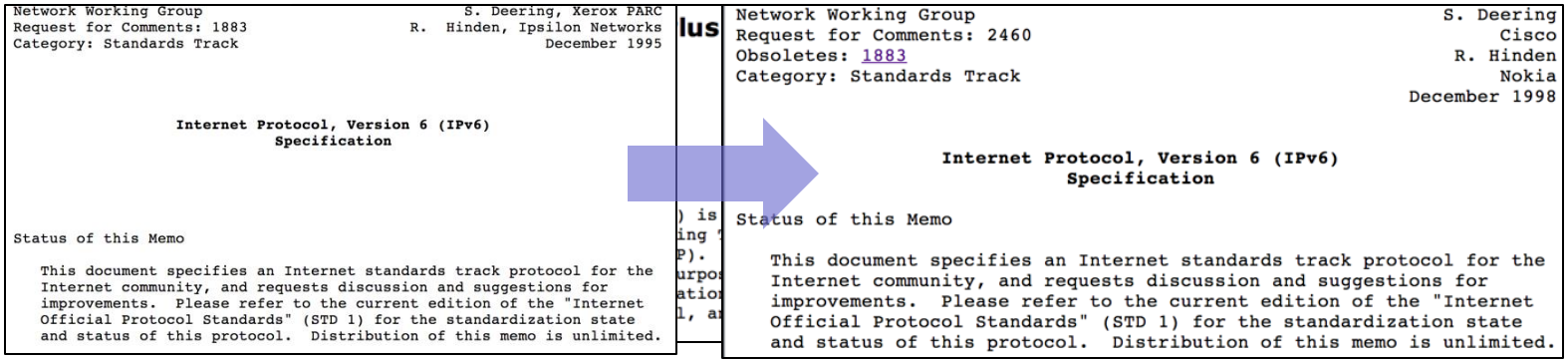
- How many is 340 undecillion?
- 340 undecillion addresses is 10 nonillion addresses per person!
- Internet is a much different place and will continue to evolve:
  - Mobile devices
  - Video on demand
  - Internet of Everything
  - A critical part in how we "live, work, play, and learn".

10 nonillion
= 10,000,000,000,000,000,000,000,000,000

IPv6

- IPv6 is not just about more addresses:
  - Stateless autoconfiguration
  - End-to-end reachability without private addresses and NAT
  - Better support for mobility
  - Peer-to-peer networking easier to create and maintain, and services such as VoIP and Quality of Service (QoS) become more robust.

IPv6: A Brief History



Network Working Group
Request for Comments: 1883
Category: Standards Track

S. Deering, Xerox PARC
R. Hinden, Ipsilon Networks
December 1995

Internet Protocol, Version 6 (IPv6) Specification

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Network Working Group
Request for Comments: 2460
Obsoletes: 1883
Category: Standards Track

S. Deering
Cisco
R. Hinden
Nokia
December 1998

Internet Protocol, Version 6 (IPv6) Specification

Status of this Memo

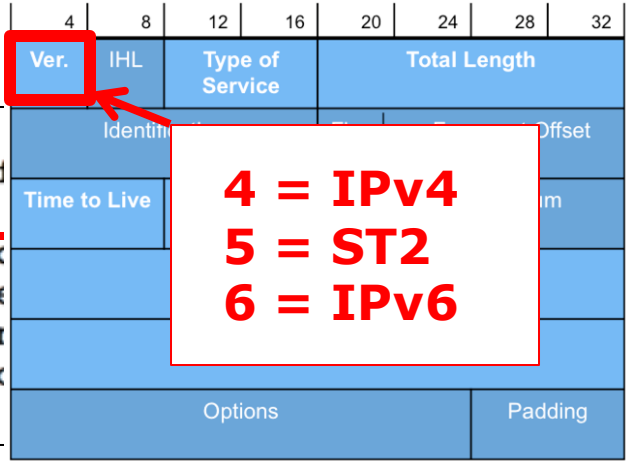
This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

- 1993, IETF announced a call for white papers with RFC 1550 *IP: Next Generation (IPng) White Paper Solicitation*.
- IETF chose Simple Internet Protocol Plus (SIPP) written by Steve Deering, Paul Francis, and Bob Hinden but changed the address size from 64 bits to 128 bits.
- 1995, IETF published RFC 1883 Internet Protocol, Version 6 (IPv6) Specification - later obsoleted by RFC 2460 in 1998.

RFC 1190

What About IPv5?

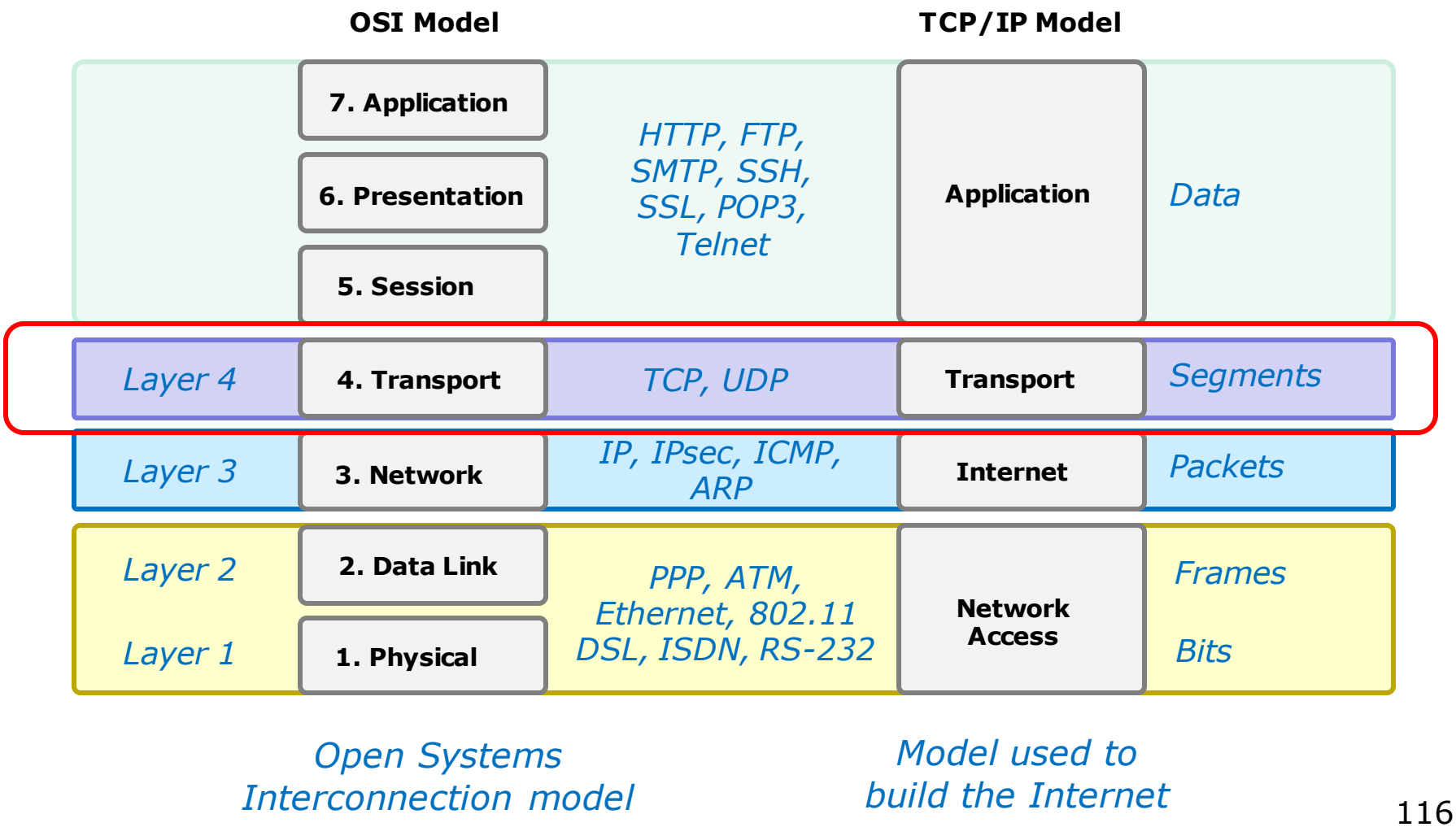
The ST packet header is not constrained to be compatible
IP packet header, except for the IP Version Number (the
bits) that is used to distinguish ST packets (IP Version
IP packets (IP Version 4). The ST packets, or protocol
(PDUs), can be encapsulated in IP either to provide conne
(possibly with degraded service) across portions of an i
that do not provide support for ST, or to allow access t
such as security that are not provided directly by ST.



- In the late 1970s, a family of experimental protocols was developed intended to provide quality of service (QoS) for real-time multimedia applications such video and voice.
- Known as Internet Stream Protocol (ST) and later ST2 – (RFC 1190 and RFC 1819).
- Although it was never known as IPv5, when encapsulated in IP, ST uses IP Protocol version 5.

Transport Layer

OSI and TCP/IP Models



Transport Layer

The Protocols

There are two primary protocols operating at the Transport layer:

User Datagram Protocol (UDP)
- Connectionless *(snmp traps are "fire and forget")*
- Stateless
- *Unreliable*
- The UDP packet is called a ***packet***

Transmission Control Protocol (TCP)
- Connection-oriented
- Stateful *(like new or established states in firewalls)*
- *Reliable* The TCP packet is called a ***segment***

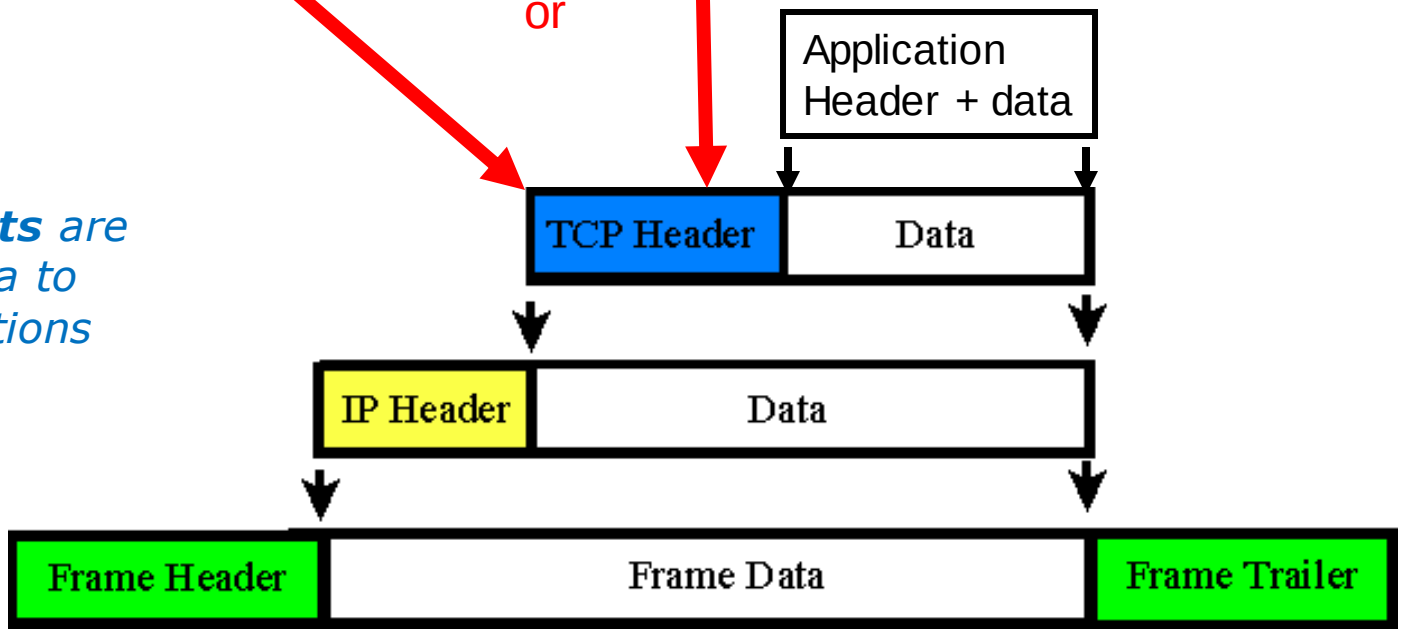
TCP Header

Source Port (16 bits)								Destination Port (16 bits)
Sequence Number (32 bits)								
Acknowledgement Number (32 bits)								
Data Offset (4 bits)	Reserved (6 bits)	URG	ACK	PSH	RST	SYN	FIN	Window (16 bits)
Checksum (16 bits)								Urgent Pointer (16 bits)
Options and Padding								

UDP Header

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data....	

or

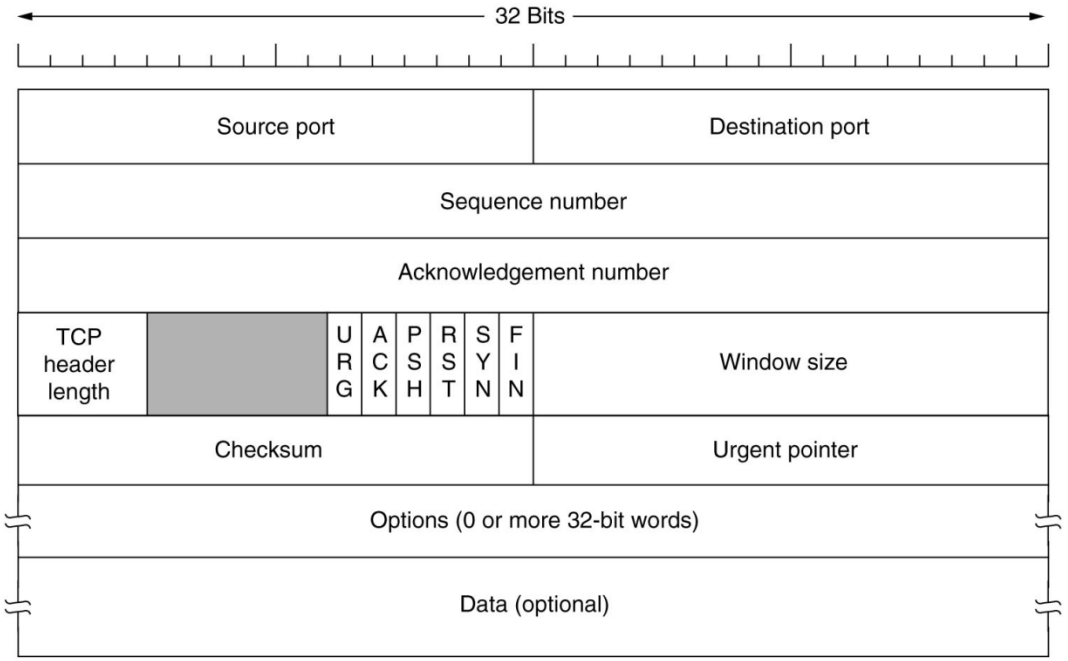


*The source and destination **ports** are used to get data to specific applications*

Transport Layer

The Transmission Control Protocol

TCP Header



*The source and destination addresses at this level are **ports***

Sequence and acknowledgement numbers are used for flow control.

ACK, SYN and FIN flags are used for initiating connections, acknowledging data received and terminating connections

Window size is used to communicate buffer size of recipient.

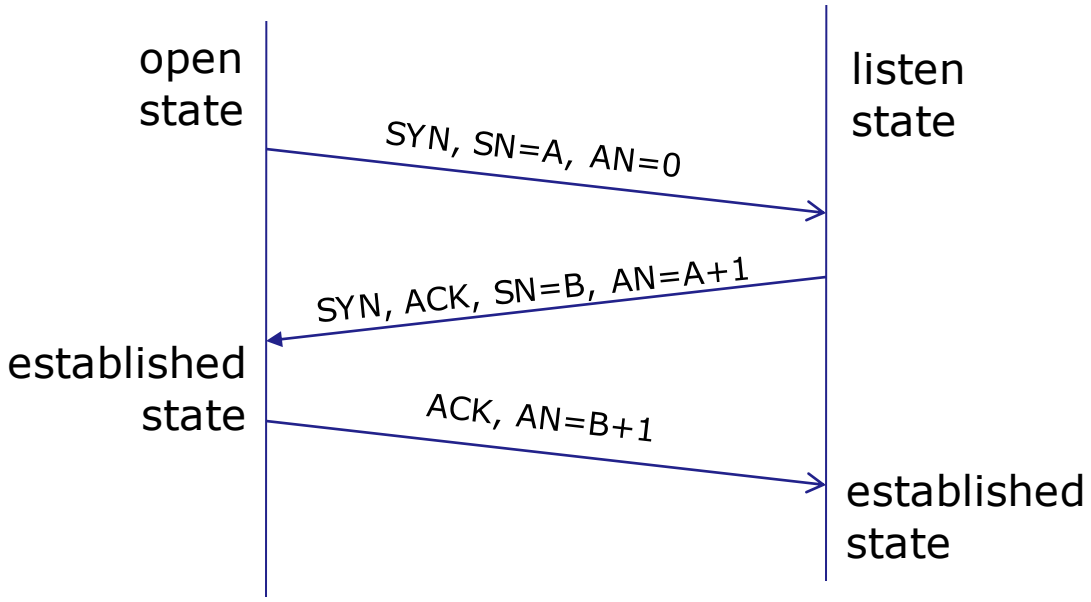
Options like SACK permit selective acknowledgement

Transport Layer

3-Way Handshake

Initiating a new TCP Connection

1. SYN
2. SYN-ACK
3. ACK



AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set
SYN=SYN flag set

Transport Layer

Sockets

Sockets are communication endpoints which define a network connection between two computers (RFC 793).

- Source IP address
- Source port number
- Destination IP address
- Destination port number

The socket is associated with a port number so that the TCP layer can identify the application to send data to.

Application programs can read and write to a socket just like they do with files.

Transport Layer

The Transmission Control Protocol (TCP)

Continuing communications on an established connection

- The Sliding Window
  Used for flow control - allows sending additional segments before an acknowledgement is received based on recipients buffer size

- Flow Control (cumulative acknowledgment)
  Recipient tells sender the size of its input buffer and sends acknowledgements (ACKs) when data has been received. Sequence numbers are used to detect missing segments.

- The SACK option
  Selective acknowledgement so only the dropped segments need to be retransmitted.

- The RST Flag
  Used to terminate a connection when an abnormal situation happens

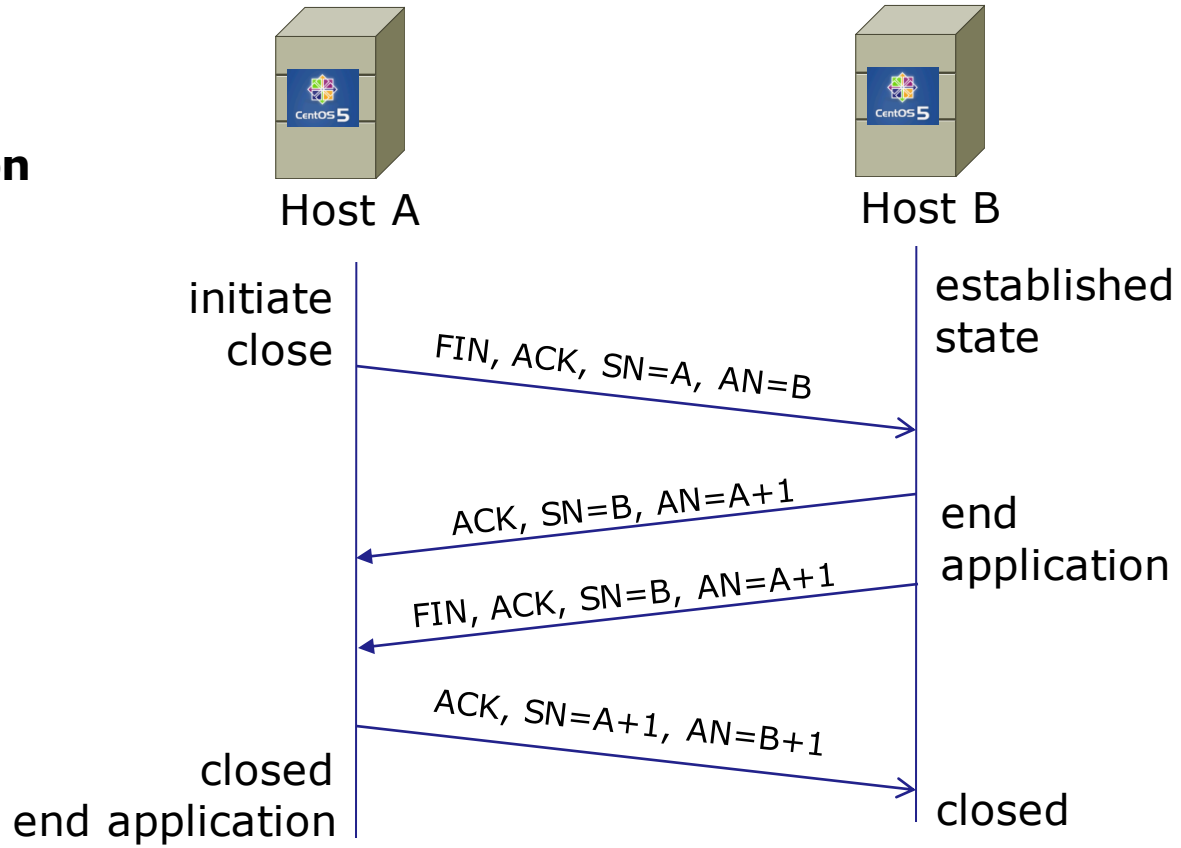
Transport Layer

Closing a TCP Connection

Four-Way Handshake

1. FIN, ACK
2. ACK
3. FIN, ACK
4. ACK

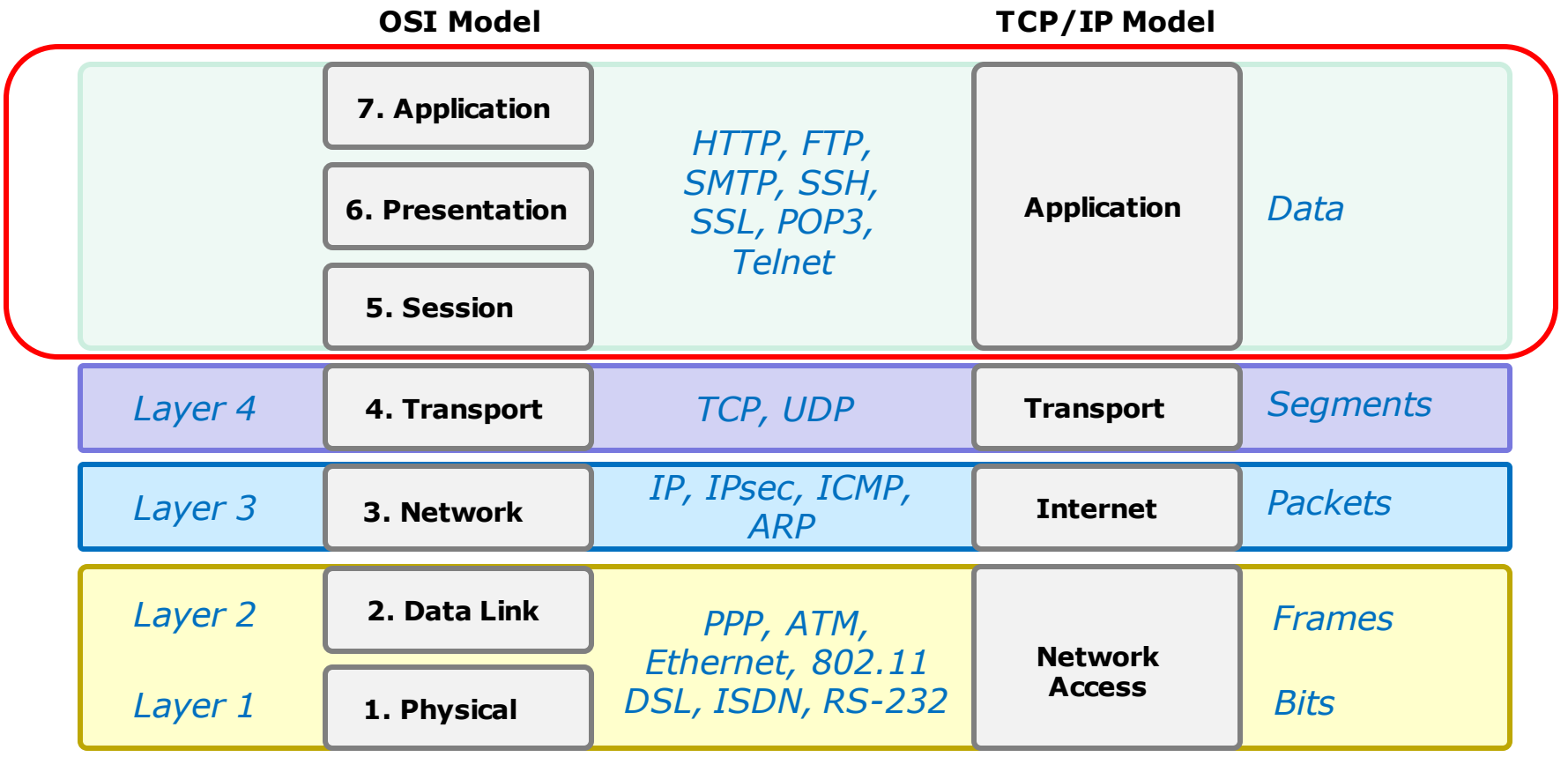
Closing with a shorter three-way handshake is also possible, where the Host A sends a FIN and Host B replies with a FIN & ACK (combining two steps into one) and Host A replies with an ACK.



AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set
FIN=FIN flag set

Application Layer

OSI and TCP/IP Models



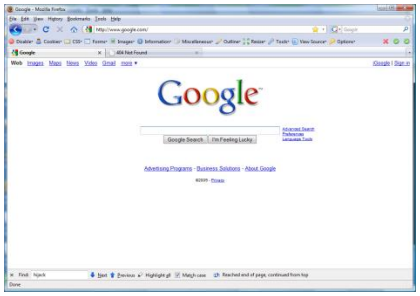
Open Systems Interconnection model

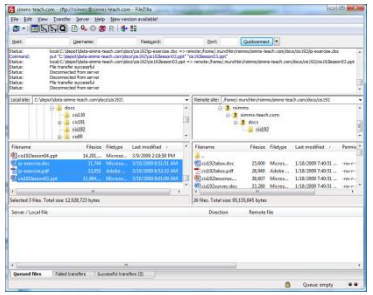
Model used to build the Internet

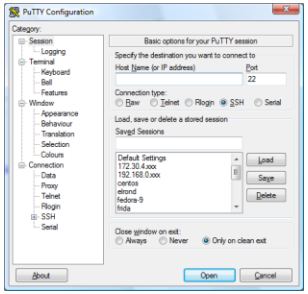
Application Layer

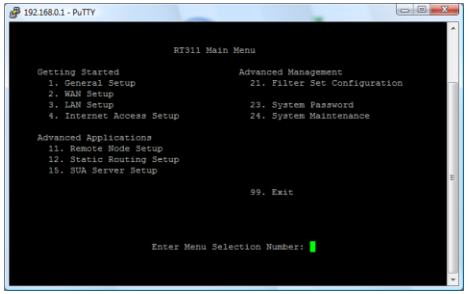
Applications

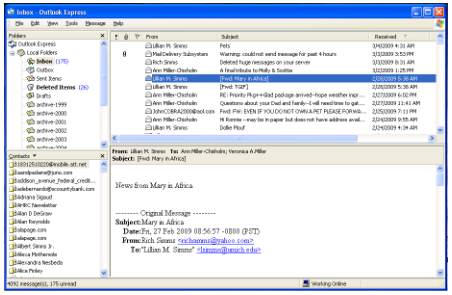
Examples:
- Web servers
- FTP servers
- SSH daemon
- Telnet server
- Mail servers











Application Layer

Responsibilities of Applications
Network connections, routing, and transfer of data are all taken care of by the lower layers of the protocol stack. What must applications do?

- Authenticate users
- Control access
- Log important information
- Format data (compress/encrypt)
- Provide whatever functionality is desired.

Application Layer

The Client-Server Model

Clients

Programs that are generally run on demand, and initiate the network connection to the server.
Examples: telnet, ftp, ssh, browsers, email clients.

Servers

Programs (services/daemons) that are constantly running in the background waiting for client connections.

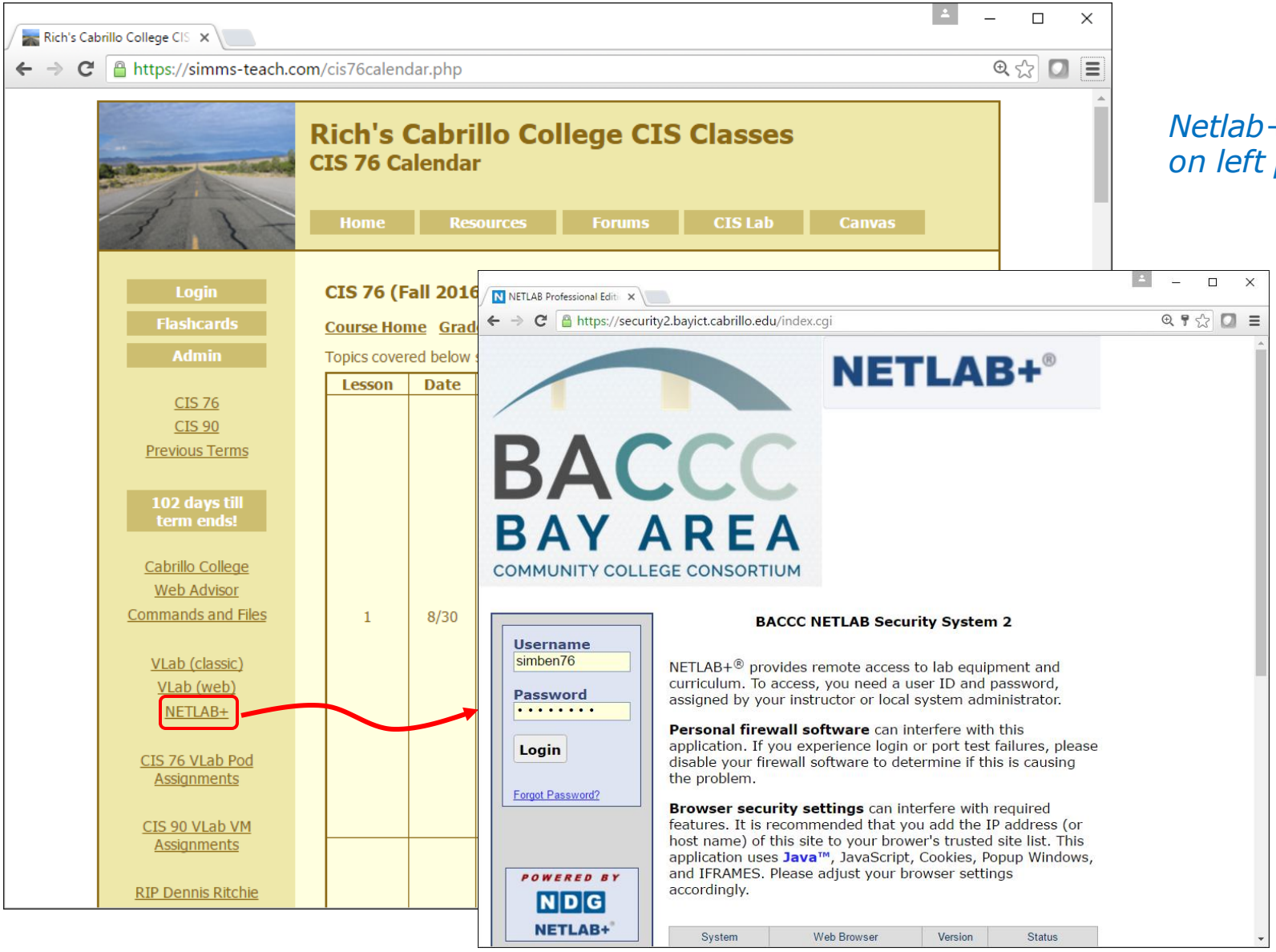
- Services and Ports: */etc/services*
- Architecture:
  - Direct or iterative servers – listen to a particular port and directly responds to requests
  - Indirect or concurrent servers (e.g. super daemons) – listen to a particular port and then starts up another server program to process the request

Service Ports

Last week we talked about Layer 4 ports. Ports are used to direct requests to the appropriate service/application

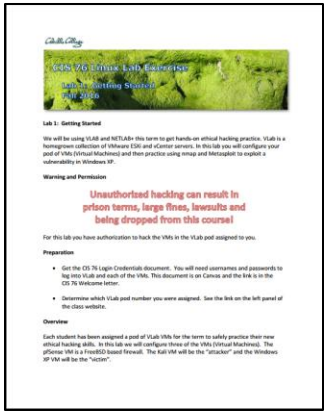
```
< snipped >
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
lmtp            24/tcp                          # LMTP Mail Delivery
lmtp            24/udp                          # LMTP Mail Delivery
smtp            25/tcp          mail
smtp            25/udp          mail
< snipped >
domain          53/tcp                          # name-domain server
domain          53/udp
whois++         63/tcp
whois++         63/udp
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp          dhcpc           # BOOTP client
bootpc          68/udp          dhcpc
tftp            69/tcp
tftp            69/udp
finger          79/tcp
finger          79/udp
http            80/tcp          www www-http    # WorldWideWeb HTTP
http            80/udp          www www-http    # HyperText Transfer Protocol
kerberos        88/tcp          kerberos5 krb5  # Kerberos v5
< snipped >
```


Assignment



Netlab+ link on left panel

Lab Assignments


Lab 1: Getting Started

We will be using VLAB and NETLAB+ this term to get hands-on ethical hacking practice. VLab is a homegrown collection of VMware ESXi and vCenter servers. In this lab you will configure your pod of VMs (Virtual Machines) and then practice using nmap and Metasploit to exploit a vulnerability in Windows XP.

Warning and Permission

Unauthorized hacking can result in prison terms, large fines, lawsuits and being dropped from this course!

For this lab you have authorization to hack the VMs in the VLab pod assigned to you.

Preparation

- Get the CIS 76 Login Credentials document. You will need usernames and passwords to log into VLab and each of the VMs. This document is on Canvas and the link is in the CIS 76 Welcome letter.
- Determine which VLab pod number you were assigned. See the link on the left panel of the class website.

Overview

Each student has been assigned a pod of VLab VMs for the term to safely practice their new ethical hacking skills. In this lab we will configure three of the VMs (Virtual Machines). The pfSense VM is a FreeBSD based firewall. The Kali VM will be the "attacker" and the Windows XP VM will be the "victim".

Pearls of Wisdom:

- Don't wait till the last minute to start.
- The *slower* you go the *sooner* you will be finished.
- A few minutes reading the forum can save you hour(s).
- Line up materials, references, equipment, and software ahead of time.
- It's best if you fully understand each step as you do it. Refer back to lesson slides to understand the commands you are using.
- Use Google for trouble-shooting and looking up supplemental info.
- Keep a growing cheat sheet of commands and examples.
- Study groups are very productive and beneficial.
- Use the forum to collaborate, ask questions, get clarifications, and share tips you learned while doing a lab.
- Plan for things to go wrong and give yourself time to ask questions and get answers.
- Late work is not accepted so submit what you have for partial credit.


Wrap up

Next Class

Assignment: Check the Calendar Page on the web site to see what is due next week.

Lab 2

Quiz questions for next class:

- What standard port is used for HTTP?
- How many bits make up an IPv6 address?
- True or false: UDP is a connectionless protocol?

Backup