

The background of the slide is a light gray gradient with several realistic water droplets of various sizes scattered across it. The droplets have highlights and shadows, giving them a three-dimensional appearance. The text 'COMPUTER SECURITY' is centered in the middle of the slide.

# COMPUTER SECURITY

# A BRIEF HISTORY OF IDS

- 1980: JAMES P. ANDERSON," COMPUTER SECURITY, THREAT MONITORING AND SURVEILLANCE," *JAMES P. ANDERSON & CO.*, 1980: A STUDY FOR USAF
- 1986: USNAVY'S SPACE AND NAVAL WARFARE SYSTEM COMMAND (SPAWARS) FUNDED RESEARCH: DOROTHY DENNING," AN INTRUSION DETECTION MODEL," *PROCEEDINGS OF THE 1986 IEEE SYMPOSIUM ON SECURITY AND PRIVACY*, MAY 1986, PP. 119-131
- 1986-92 1985: US NAVY FUNDS DEVELOPMENT OF INTRUSION DETECTION EXPERT SYSTEM (IDES) AT STANFORD RESEARCH INSTITUTE (SRI INTERNATIONAL) BASED ON DENNING'S PAPER
- 1987: FIRST ANNUAL ID WORKSHOP AT SRI

- 1989: TODD HEBERLIN, A STUDENT OF UNIV OF CALIFORNIA, DAVIS: WRITES NETWORK SECURITY MONITOR TO BE RUN ON SUN UNIX WORKSTATION
- 1992: COMMERCIAL PRODUCTS: COMPUTER MISUSE DETECTION SYSTEM (CMDS) BY SCREEN APPLICATIONS INTERNATIONAL CORP (SAIC) --BASED ON NAVY WORK
- 1992: COMMERCIAL PRODUCTS: STALKER -- BASED ON HAYSTACK LABS WORK DONE FOR USAF.
- 1994: NETWORK IDS CALLED ASIM DEVELOPED AT AIR FORCE CRYPTOLOGICAL SUPPORT CENTER -- COMMERCIAL COMPANY WHEELGROUP FORMED BY SCIENTISTS OF THE CENTER

# A BRIEF HISTORY OF IDS

- 1997: CISCO ACQUIRES WHEELGROUP AND INCORPORATES THE TECHNOLOGY OF IDS IN ITS ROUTERS
- 1997: REALSECURE FOR WINDOWS NT BY INTERNET SECURITY SYSTEMS
- 1999: PRESIDENTIAL DECISION DIRECTIVE # 63: ESTABLISHED FEDERAL ID NETWORK (FIDNET) TO DETECT ATTACKS ON GOVT INFRASTRUCTURE

# INTRUDERS

**INTRUDER: A NON-AUTHORIZED USER OF A COMPUTER SYSTEM.**

TYPES OF INTRUDERS:

- **MASQUERADER:** PENETRATES A SYSTEM'S ACCESS CONTROL LIST TO EXPLOIT A LEGITIMATE USER'S ACCOUNT; **USUALLY AN OUTSIDER**
- **MISFEASOR:** A LEGITIMATE USER, WHO ACCESSES RESOURCES, HE IS NOT AUTHORIZED TO ACCESS; **AN INSIDER**
- **CLANDESTINE USER:** SEIZES **SUPERVISORY CONTROL** AND USES IT TO ACCESS RESOURCES AND TO EVADE AUDIT; **MAY BE AN OUTSIDER/INSIDER**

- **TWO TYPES OF INTRUDERS:**

- (I) SOPHISTICATED

- (II) FOOT SOLDIERS, READY TO SPEND HOURS IN SEARCHING FOR WEAKNESSES, BY USING TOOLS DEVELOPED BY SOPHISTICATED USERS

- **ATTACKS:**

- (I) BENIGN

- (II) SERIOUS ( THREE LEVELS: UNAUTHORIZED ACCESS, UNAUTHORIZED MODIFICATION, DENIAL OF SERVICE)

- **INTRUSION DETECTION** – ACCORDING TO THE WIKIPEDIA INTRUSION DETECTION IS THE ACT OF DETECTING ACTIONS THAT ATTEMPT TO COMPROMISE THE CONFIDENTIALITY, INTEGRITY OR AVAILABILITY OF A RESOURCE

- **AN INTRUSION DETECTION SYSTEM (IDS):** DESIGNED TO DETECT INTRUDERS

# CLASSIFICATION OF IDSS

- STATISTICAL ANOMALY DETECTION:
  - THRESHOLD DETECTION: COUNT THE OCCURRENCES OF ANOMALOUS EVENTS. IF THE NUMBER CROSSES A THRESHOLD → INTRUSION ALERT.
  - PROFILE – BASED: THE REGULAR PROFILES OF USE OF THE SYSTEMS BY **USERS/ GROUPS/ APPLICATIONS** ARE CREATED. SIMILARLY A PROFILE OF THE USE OF VARIOUS **SYSTEM RESOURCES** CAN BE CREATED. IF THE USAGE IS DIFFERENT → INTRUSION ALERT.
  - LEARNING BASED SYSTEM, WHICH CONTINUOUSLY UPDATES THE PROFILE
- MAY BE ABLE TO DETECT A NEW TYPE OF ATTACK;

# CLASSIFICATION OF IDS CONTINUED

- STATISTICAL ANOMALY DETECTION: CONTINUED
  - MORE FALSE POSITIVES (FALSE ALERTS) AND FALSE NEGATIVES (ATTACKS, WHICH ARE NOT DETECTED);
  - A CAREFUL HACKER MAY BE ABLE TO AVOID DETECTION BY SLOWLY “TRAINING” THE SYSTEM TO CONSIDER THE ANOMALOUS SITUATION AS THE NORMAL STATE
- SIGNATURE-BASED (OR MISUSE-BASED) DETECTION:
  - REDUCES FALSE POSITIVES AND FALSE NEGATIVES;
  - CANNOT DETECT A NEW TYPE OF ATTACK



# INTRUSION PROCESS

- RECONNAISSANCE
- INTRUSION
- EXPLOITATION
- REINFORCEMENT
- CONSOLIDATION
- PILLAGE

## PROBLEMS CAUSED BY INTRUSION

- LOSS OF BUSINESS (THROUGH DOS ETC)
- LOSS OF (I) INTEGRITY OF DATA (II) PRIVACY (III) PERSONAL DATA (IV) FAITH IN BUSINESS PROCESS
- LEGAL LIABILITY

# INTRUSION DETECTION SYSTEMS

TRUE	FALSE	
TRUE - POSITIVE	FALSE - POSITIVE	POSITIVE
TRUE - NEGATIVE	FALSE - NEGATIVE	NEGATIVE

# WHY PACKETS?

- Packets Don't Lie
- We can't fix what we can't see
- Traffic and Protocol Analysis is the most detailed troubleshooting method



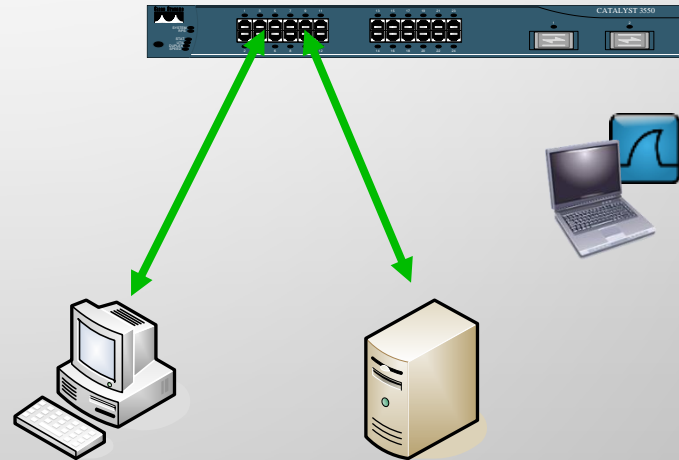
# WHY PACKETS?

- Packets Don't Lie
- We can't fix what we can't see
- Traffic and Protocol Analysis is the most detailed troubleshooting method



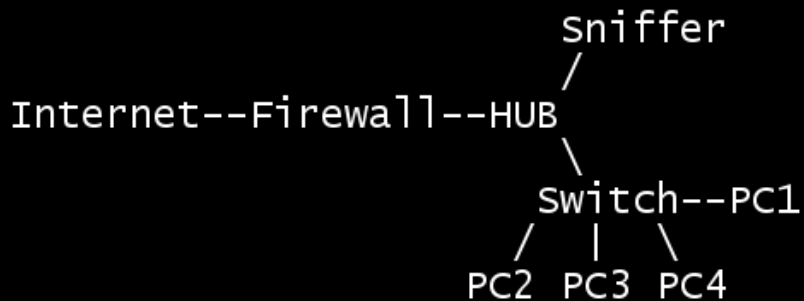
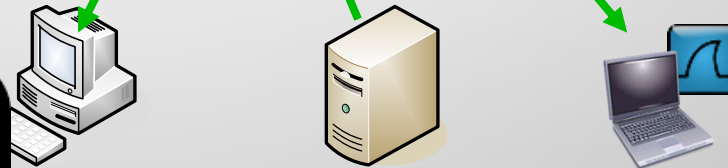
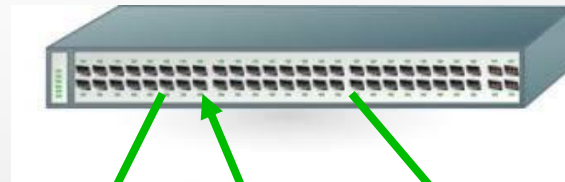
# PACKET COLLECTION

- COMMON CAPTURE METHODS:
  - SPAN/MIRROR
  - TAP / AGGREGATION TAP
  - DIRECTLY ON THE CLIENT



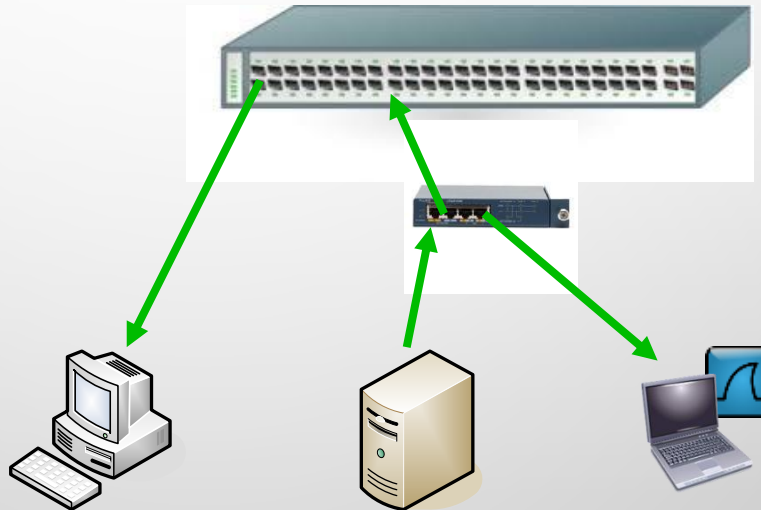
# GETTING IN THE PATH: SPAN/MIRROR

- COPIES SELECTED PORTS, HOSTS, VLANS, OR TRAFFIC PATTERNS TO A MONITOR PORT

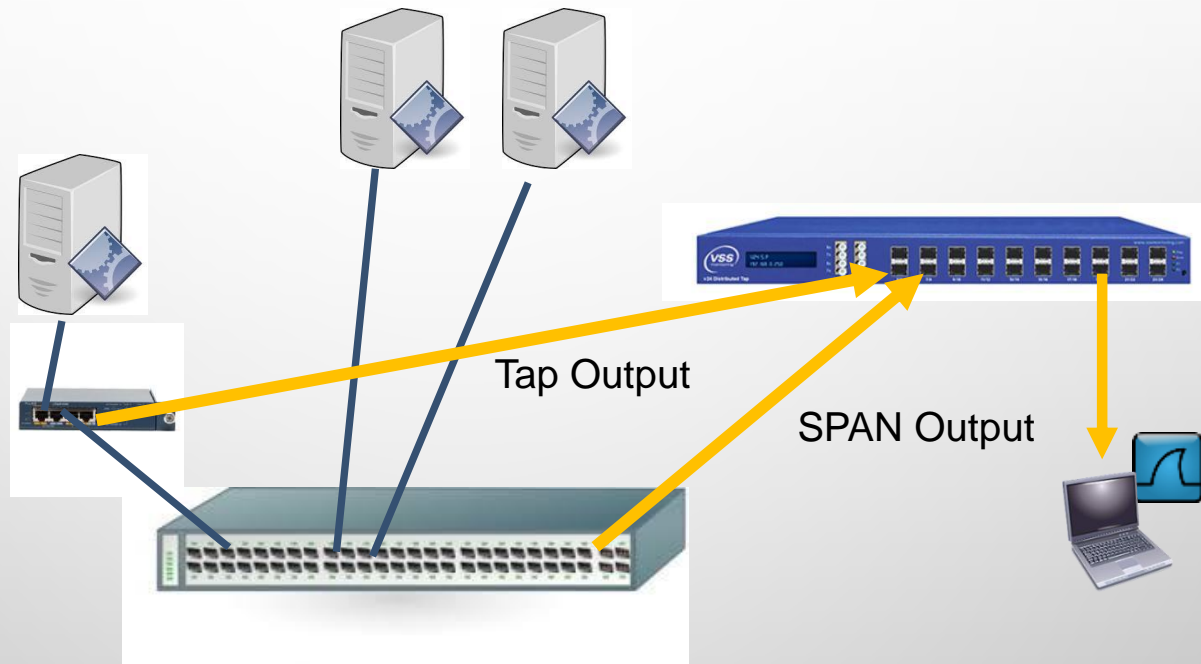


# GETTING IN THE PATH: TAPS

- A TAP IS THE BEST MEANS TO CAPTURE PACKETS
- DIRECTLY MONITORS THE CONNECTION INLINE



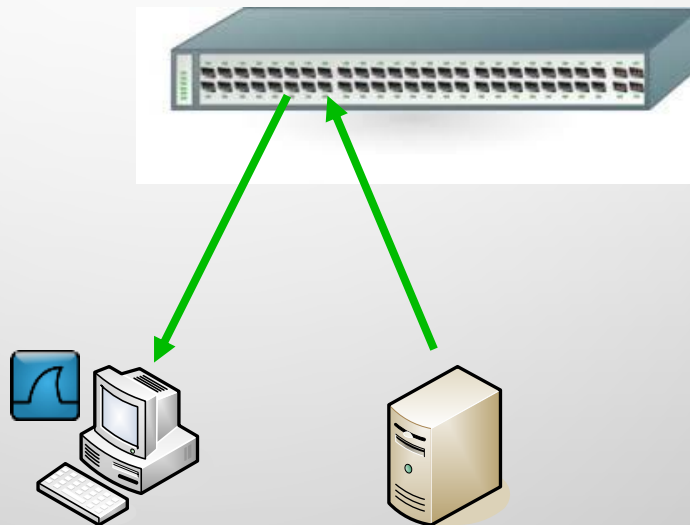
# GETTING IN THE PATH: AGGREGATION TAP





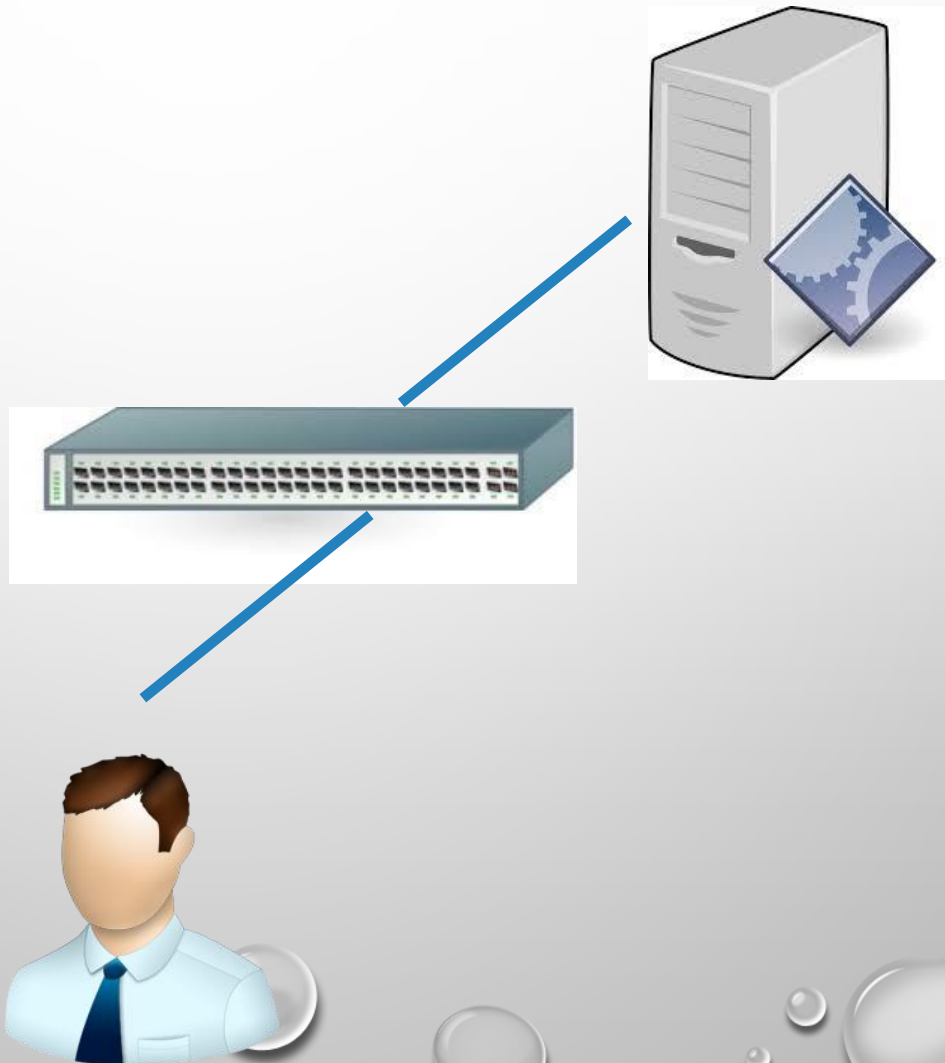
# CAPTURING DIRECTLY FROM CLIENT

- THE PROTOCOL ANALYZER CAN BE INSTALLED DIRECTLY ON THE CLIENT – BUT THERE ARE SEVERAL NEGATIVES TO THIS



# PROBLEMS AREN'T WHAT THEY USED TO BE

A wireless 802.1X client device on the wireless network, for example, may appear connected to the wireless network, but the user is not able to access network resources. After reviewing the packet trace, you may see (by observing the VLAN tagging in the appropriate packets), that the client device is connected to the guest network instead of the corporate network. This would point to a problem with the client's 802.1X supplicant.



# QUICK, FIND THE PROBLEM!

The image shows a Wireshark network traffic capture window. The title bar indicates the file is 'http-browse-ok.pcapng' and the version is 'Wireshark 1.10.7 (v1.10.7-0-g6b931a1 from master-1.10)'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. The toolbar contains various icons for file operations, navigation, and analysis. The main window is divided into three panes: a packet list, a packet details pane, and a packet bytes pane.

The packet list pane shows a table of captured packets. The columns are No., Time, Delta, Source, Destination, Length, Window size value, MSS Value, and Info. The packets are numbered 1 through 13. Packet 11 is highlighted in red, indicating a problem. The details pane shows the structure of the first packet (Frame 1), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

No.	Time	Delta	Source	Destination	Length	Window size value	MSS Value	Info
1	0.000000000	0.000000000	192.168.1.182	161.58.73.170	62	16384	1460	cp1scram
2	0.150993000	0.150993000	161.58.73.170	192.168.1.182	62	49152	1460	http >
3	0.151105000	0.000112000	192.168.1.182	161.58.73.170	60	17520		cp1scram
4	0.151487000	0.000382000	192.168.1.182	161.58.73.170	321	17520		GET / H
5	0.291530000	0.140043000	161.58.73.170	192.168.1.182	60	49152		http >
6	0.300581000	0.009051000	161.58.73.170	192.168.1.182	254	49152		HTTP/1.0
7	0.300894000	0.000313000	192.168.1.182	161.58.73.170	60	17320		cp1scram
8	0.302086000	0.001192000	161.58.73.170	192.168.1.182	60	49152		http >
9	0.302195000	0.000109000	192.168.1.182	161.58.73.170	60	17320		cp1scram
10	0.328354000	0.026159000	192.168.1.182	161.58.73.170	62	16384	1460	ff-annun
11	0.441462000	0.113108000	161.58.73.170	192.168.1.182	60	49152		[TCP Re
12	0.464725000	0.023263000	161.58.73.170	192.168.1.182	62	49152	1460	[TCP AC
13	0.464808000	0.000083000	192.168.1.182	161.58.73.170	60	0		cp1scram

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0  
Ethernet II, Src: AmbitMic\_0b:b9:44 (00:d0:59:0b:b9:44), Dst: 192.168.1.182 (08:00:27:00:00:02)  
Internet Protocol Version 4, Src: 192.168.1.182, Dst: 161.58.73.170  
Transmission Control Protocol, Src Port: cplscram, Dst Port: 80

File: "C:\Packet Pioneer\Clients\Wir... Packets: 399 · Displayed: 399 (100.0%) · Load time: 0:00.053 Profile: TCP Plain

# VERIFYING WHICH PORTS ARE LISTENING

After configuring network services, it is important to pay attention to which ports are actually listening on the system's network interfaces. Any open ports can be evidence of an intrusion.

There are two basic approaches for listing the ports that are listening on the network. The less reliable approach is to query the network stack by typing commands such as `netstat -an` or `lsof -i`.

This method is less reliable since these programs do not connect to the machine from the network, but rather check to see what is running on the system. For this reason, these applications are frequent targets for replacement by attackers. In this way, crackers attempt to cover their tracks if they open unauthorized network ports.

A more reliable way to check which ports are listening on the network is to use a port scanner such as `nmap`.

The following command issued from the console determines which ports are listening for TCP connections from the network:

```
nmap -sT -O localhost
```

```
root@kali:~# nmap localhost -sV -p9000
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-03-05 20:49 IST
```

```
Nmap scan report for localhost (127.0.0.1)
```

```
Host is up (-1100s latency).
```

```
Other addresses for localhost (not scanned): 127.0.0.1
```

```
PORT      STATE SERVICE VERSION
```

```
9000/tcp  open  http    Apache Tomcat/Coyote JSP engine 1.1
```

```
Service detection performed. Please report any incorrect results  
mit/ .
```

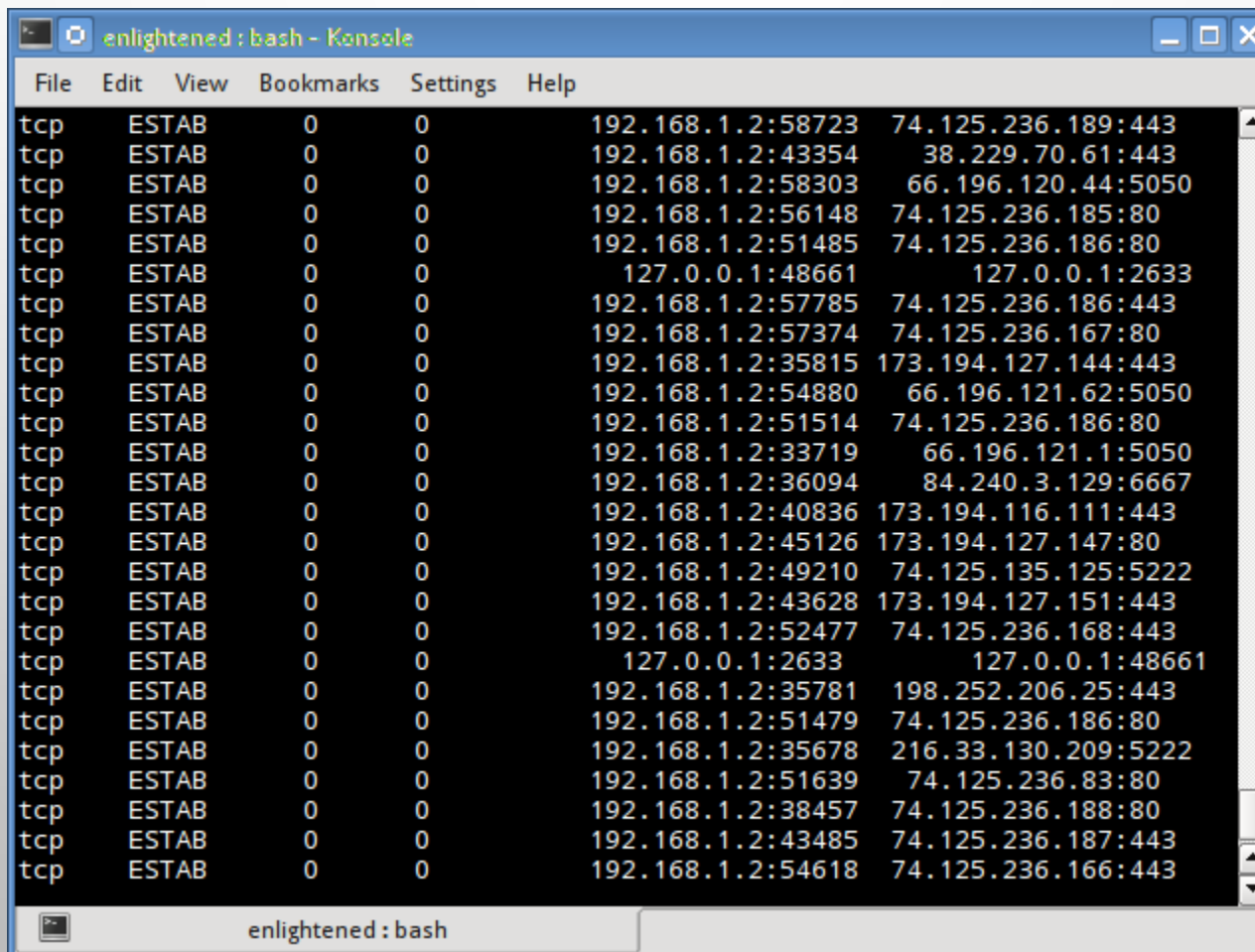
```
Nmap done: 1 IP address (1 host up) scanned in 14.21 seconds
```

# OPEN PORTS OBSERVER

- SS -TULNP

- TO CONSULT THE LIST OF ALL OPEN PORTS

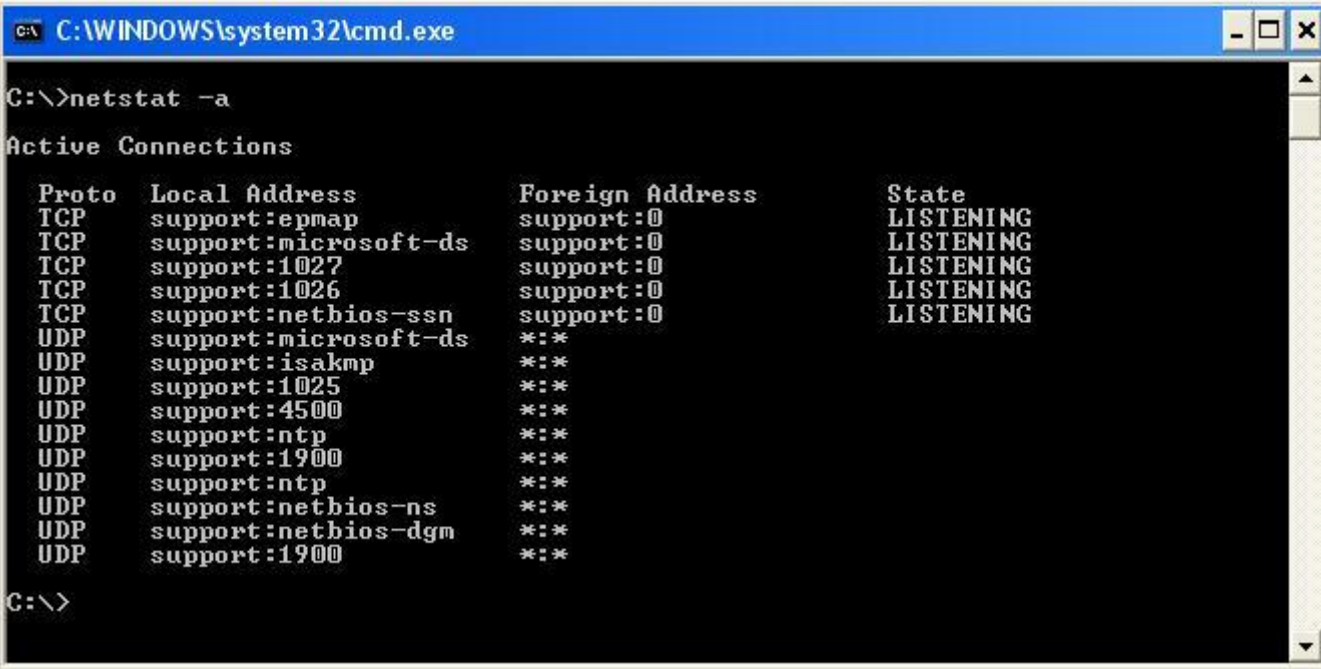
SS -AP



```
enlightened : bash - Konsole
File Edit View Bookmarks Settings Help
tcp ESTAB 0 0 192.168.1.2:58723 74.125.236.189:443
tcp ESTAB 0 0 192.168.1.2:43354 38.229.70.61:443
tcp ESTAB 0 0 192.168.1.2:58303 66.196.120.44:5050
tcp ESTAB 0 0 192.168.1.2:56148 74.125.236.185:80
tcp ESTAB 0 0 192.168.1.2:51485 74.125.236.186:80
tcp ESTAB 0 0 127.0.0.1:48661 127.0.0.1:2633
tcp ESTAB 0 0 192.168.1.2:57785 74.125.236.186:443
tcp ESTAB 0 0 192.168.1.2:57374 74.125.236.167:80
tcp ESTAB 0 0 192.168.1.2:35815 173.194.127.144:443
tcp ESTAB 0 0 192.168.1.2:54880 66.196.121.62:5050
tcp ESTAB 0 0 192.168.1.2:51514 74.125.236.186:80
tcp ESTAB 0 0 192.168.1.2:33719 66.196.121.1:5050
tcp ESTAB 0 0 192.168.1.2:36094 84.240.3.129:6667
tcp ESTAB 0 0 192.168.1.2:40836 173.194.116.111:443
tcp ESTAB 0 0 192.168.1.2:45126 173.194.127.147:80
tcp ESTAB 0 0 192.168.1.2:49210 74.125.135.125:5222
tcp ESTAB 0 0 192.168.1.2:43628 173.194.127.151:443
tcp ESTAB 0 0 192.168.1.2:52477 74.125.236.168:443
tcp ESTAB 0 0 127.0.0.1:2633 127.0.0.1:48661
tcp ESTAB 0 0 192.168.1.2:35781 198.252.206.25:443
tcp ESTAB 0 0 192.168.1.2:51479 74.125.236.186:80
tcp ESTAB 0 0 192.168.1.2:35678 216.33.130.209:5222
tcp ESTAB 0 0 192.168.1.2:51639 74.125.236.83:80
tcp ESTAB 0 0 192.168.1.2:38457 74.125.236.188:80
tcp ESTAB 0 0 192.168.1.2:43485 74.125.236.187:443
tcp ESTAB 0 0 192.168.1.2:54618 74.125.236.166:443
enlightened : bash
```

# WINDOWS

To view the list of open ports, enter the following command in command line prompt: **netstat -a**



```
C:\WINDOWS\system32\cmd.exe

C:\>netstat -a

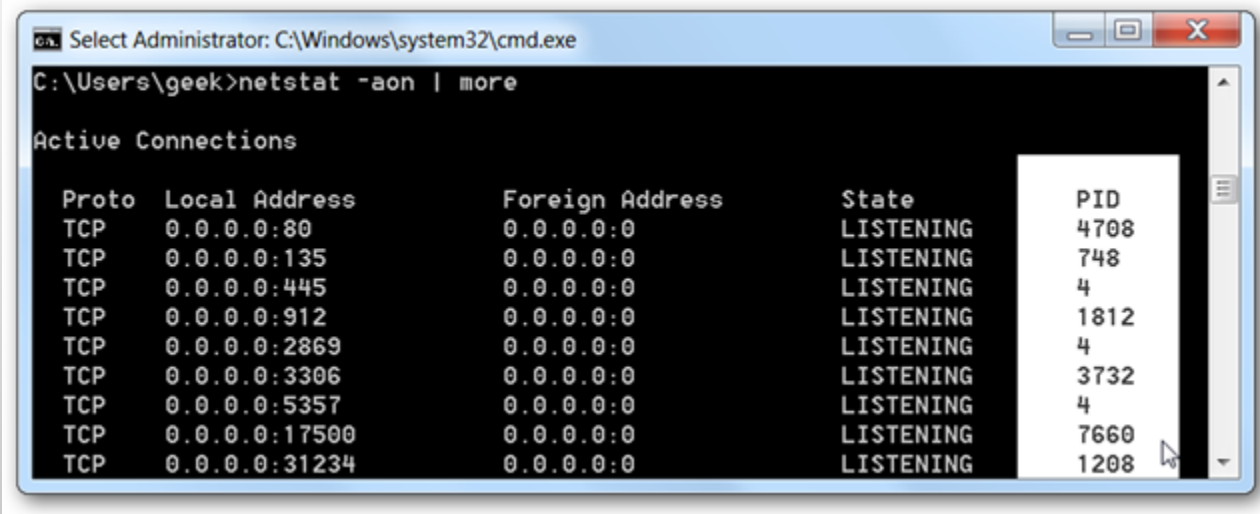
Active Connections

Proto Local Address          Foreign Address         State
TCP   support:epmap          support:0               LISTENING
TCP   support:microsoft-ds  support:0               LISTENING
TCP   support:1027           support:0               LISTENING
TCP   support:1026           support:0               LISTENING
TCP   support:nethios-ssn   support:0               LISTENING
UDP   support:microsoft-ds  *:*                    *:*
UDP   support:isakmp        *:*                    *:*
UDP   support:1025           *:*                    *:*
UDP   support:4500           *:*                    *:*
UDP   support:ntp            *:*                    *:*
UDP   support:1900           *:*                    *:*
UDP   support:ntp            *:*                    *:*
UDP   support:nethios-ns    *:*                    *:*
UDP   support:nethios-dgm   *:*                    *:*
UDP   support:1900           *:*                    *:*

C:\>
```

```
netstat -aon | more
```

If you look on the right-hand side, you'll see where I've highlighted the list of PIDs, or Process Identifiers. Find the one that's bound to the port that you're trying to troubleshoot—for this example, you'll see that 0.0.0.0:80, or port 80, is in use by PID 4708.



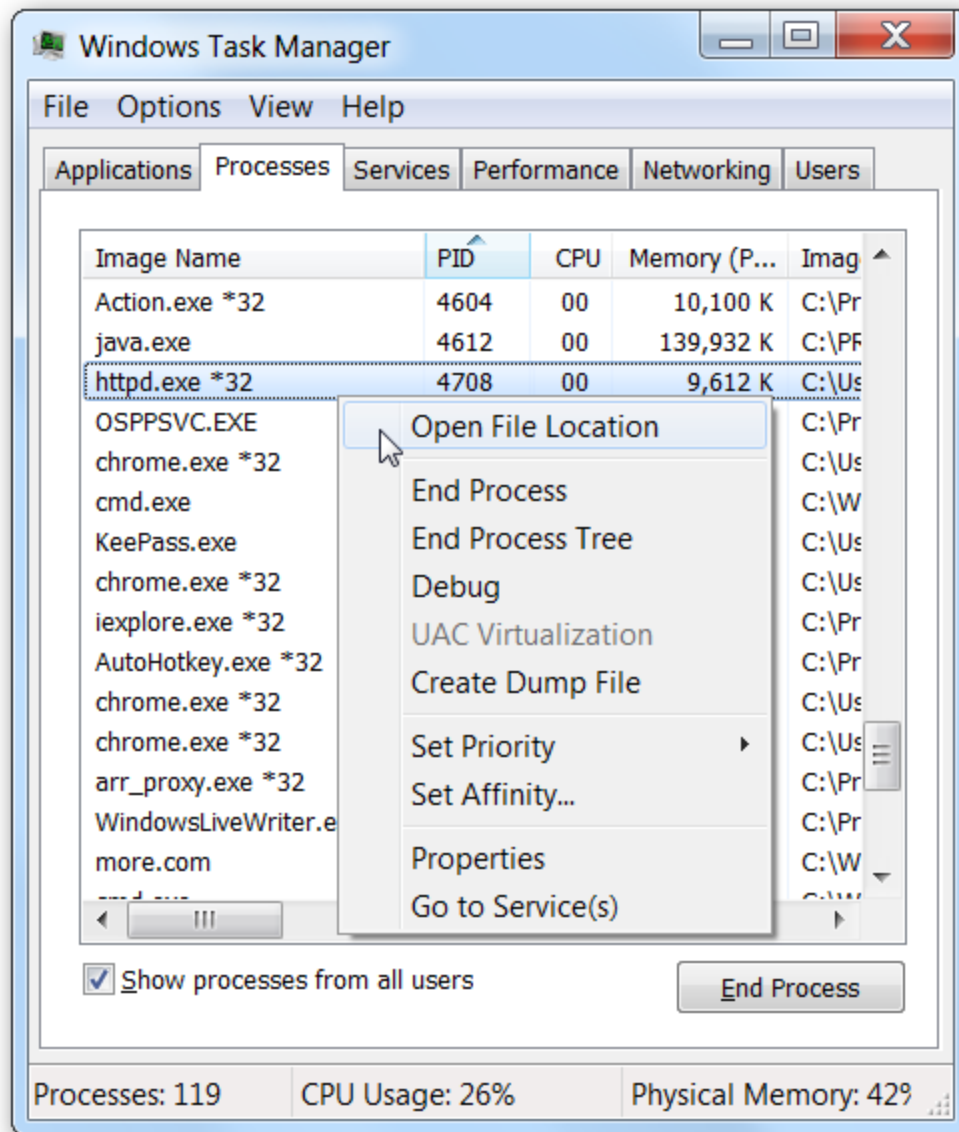
```
Select Administrator: C:\Windows\system32\cmd.exe
C:\Users\geek>netstat -aon | more

Active Connections

Proto Local Address           Foreign Address         State       PID
TCP   0.0.0.0:80              0.0.0.0:0               LISTENING  4708
TCP   0.0.0.0:135            0.0.0.0:0               LISTENING  748
TCP   0.0.0.0:445           0.0.0.0:0               LISTENING  4
TCP   0.0.0.0:912           0.0.0.0:0               LISTENING  1812
TCP   0.0.0.0:2869          0.0.0.0:0               LISTENING  4
TCP   0.0.0.0:3306          0.0.0.0:0               LISTENING  3732
TCP   0.0.0.0:5357          0.0.0.0:0               LISTENING  4
TCP   0.0.0.0:17500         0.0.0.0:0               LISTENING  7660
TCP   0.0.0.0:31234         0.0.0.0:0               LISTENING  1208
```

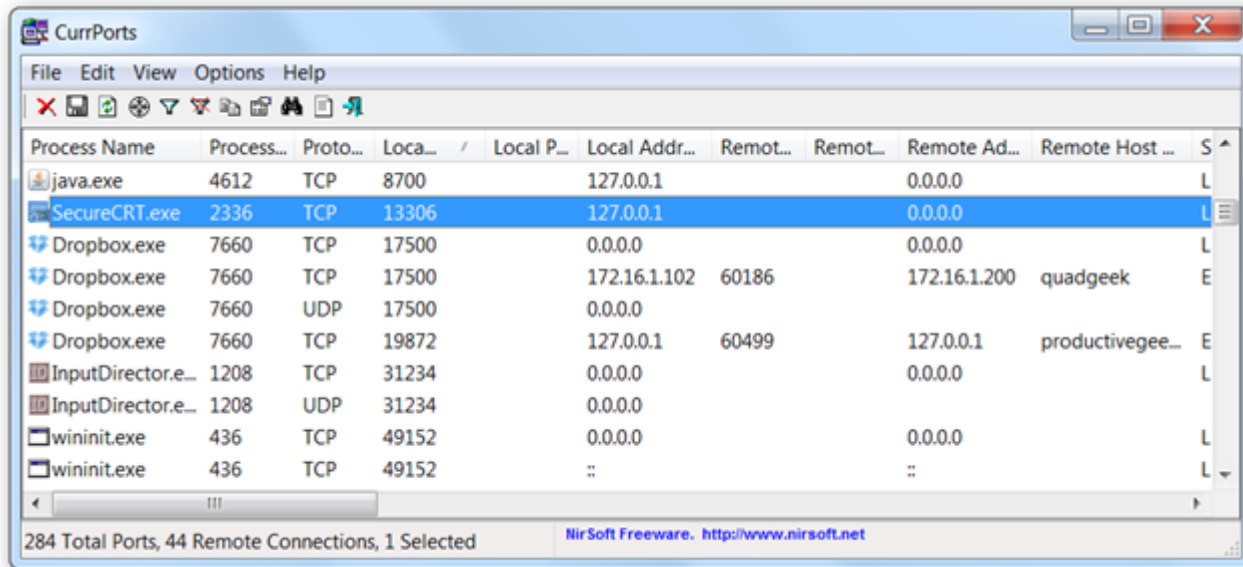


Now you can simply open up Task Manager—you might have to use the option to Show Processes for All Users, and then you'll be able to find the PID in the list. Once you're there, you can use the End Process, Open File Location, or Go to Service(s) options to control the process or stop it.



## Use CurrPorts to View What is Listening

excellent freeware CurrPorts utility by NirSoft



The screenshot shows the CurrPorts utility window. The title bar reads "CurrPorts". The menu bar includes "File", "Edit", "View", "Options", and "Help". The toolbar contains various icons for file operations and network-related functions. The main area is a table with the following columns: Process Name, Process ID, Protocol, Local Port, Local Address, Remote Port, Remote Address, Remote Host, and Status. The "SecureCRT.exe" process is selected, showing it is listening on TCP port 13306 on local address 127.0.0.1. Other processes like java.exe, Dropbox.exe, InputDirector.exe, and wininit.exe are also listed with their respective ports and protocols. The status bar at the bottom indicates "284 Total Ports, 44 Remote Connections, 1 Selected" and provides the NirSoft Freeware website URL: <http://www.nirsoft.net>.

Process Name	Process...	Proto...	Loca...	Local P...	Local Addr...	Remot...	Remot...	Remote Ad...	Remote Host ...	S
java.exe	4612	TCP	8700		127.0.0.1			0.0.0.0		L
SecureCRT.exe	2336	TCP	13306		127.0.0.1			0.0.0.0		L
Dropbox.exe	7660	TCP	17500		0.0.0.0			0.0.0.0		L
Dropbox.exe	7660	TCP	17500		172.16.1.102	60186		172.16.1.200	quadgeek	E
Dropbox.exe	7660	UDP	17500		0.0.0.0					L
Dropbox.exe	7660	TCP	19872		127.0.0.1	60499		127.0.0.1	productivegee...	E
InputDirector.e...	1208	TCP	31234		0.0.0.0			0.0.0.0		L
InputDirector.e...	1208	UDP	31234		0.0.0.0					L
wininit.exe	436	TCP	49152		0.0.0.0			0.0.0.0		L
wininit.exe	436	TCP	49152		::			::		L

# FIREWALL: A DEFINITION

- A FIREWALL IS A SET OF RELATED HARDWARE AND/OR SOFTWARE, WHICH PROTECTS THE RESOURCES OF A PRIVATE NETWORK FROM INTRUDERS.
- WATCH SINGLE POINT RATHER THAN EVERY PC
- A FIREWALL PROVIDES STRICT ACCESS CONTROL BETWEEN THE PROTECTED SYSTEMS AND THE OUTSIDE WORLD.

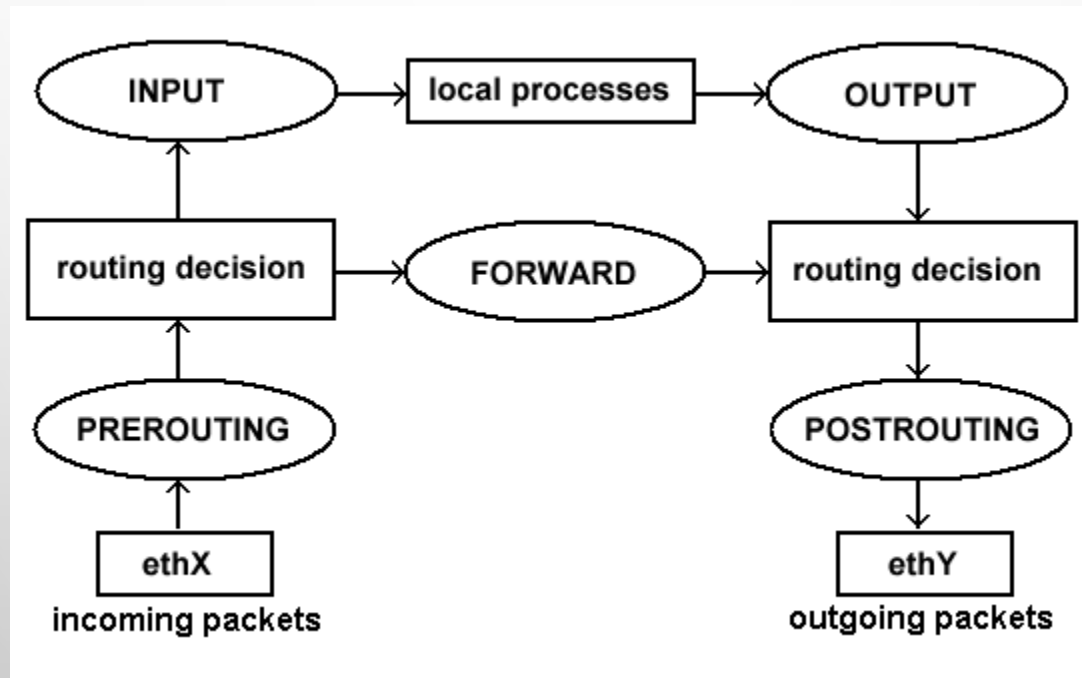
## TWO JOBS IN GENERAL:

- 1.PACKET FILTERING
- 2.APPLICATION PROXY SERVER

# NETFILTER : FIREWALL LINUX

- **NETFILTER** IS A [FRAMEWORK](#) PROVIDED BY THE [LINUX KERNEL](#) THAT ALLOWS VARIOUS [NETWORKING](#)-RELATED OPERATIONS TO BE IMPLEMENTED IN THE FORM OF CUSTOMIZED HANDLERS. NETFILTER OFFERS VARIOUS FUNCTIONS AND OPERATIONS FOR [PACKET FILTERING](#), [NETWORK ADDRESS TRANSLATION](#), AND [PORT TRANSLATION](#), WHICH PROVIDE THE FUNCTIONALITY REQUIRED FOR DIRECTING PACKETS THROUGH A NETWORK, AS WELL AS FOR PROVIDING ABILITY TO [PROHIBIT](#) PACKETS FROM REACHING SENSITIVE LOCATIONS WITHIN A COMPUTER NETWORK.
- NETFILTER REPRESENTS A SET OF [HOOKS](#) INSIDE THE LINUX KERNEL, ALLOWING SPECIFIC [KERNEL MODULES](#) TO REGISTER [CALLBACK](#) FUNCTIONS WITH THE KERNEL'S NETWORKING STACK. THOSE FUNCTIONS, USUALLY APPLIED TO THE TRAFFIC IN FORM OF FILTERING AND MODIFICATION RULES, ARE CALLED FOR EVERY PACKET THAT TRAVERSES THE RESPECTIVE HOOK WITHIN THE NETWORKING STACK

# NETFILTER : FIREWALL LINUX



# • PACKET-FILTERING ROUTER

- APPLIES A SET OF RULES TO EACH INCOMING IP

PACKET AND THEN FORWARDS OR DISCARDS THE PACKET, USUALLY FOR BOTH DIRECTIONS.

- THE RULES ARE MAINLY BASED ON THE IP AND TRANSPORT (TCP OR UDP) HEADER, INCLUDING
  - SOURCE AND DESTINATION IP ADDRESS,
  - IP PROTOCOL FIELD,
  - TCP/UDP PORT NUMBER.

# APPLICATION PROXY SERVER

ACTS AS **A RELAY** OF APPLICATION-LEVEL TRAFFIC.

USERS CONTACT THE GATEWAY USING A TCP/IP APPLICATION (SUCH AS FTP OR TELNET) WITH THE INFORMATION OF THE REMOTE HOST TO BE ACCESSED.

THE GATEWAY WILL CONTACT THE APPLICATION ON THE REMOTE HOST AND CONVEY TCP SEGMENTS CONTAINING THE APPLICATION DATA BETWEEN THE TWO ENDPOINTS.

# FIREWALL LIMITATIONS

## FIREWALL CAN NOT

- PROTECT AGAINST ATTACKS THAT BYPASS THE FIREWALL (E.G. DIAL-UP MODEM)
- PROTECT AGAINST THE TRANSFER OF VIRUS-INFECTED FILES
- PREVENT PEOPLE WALKING OUT WITH DISKS

FIREWALL **MAY** NOT PROTECT AGAINST INTERNAL THREATS, SUCH AS A BAD EMPLOYEE



# PACKET FILTERING : ADVANTAGES AND DISADVANTAGES

**ADVANTAGES:** FAST, FLEXIBLE, AND INEXPENSIVE

**DISADVANTAGES:**

- LACK THE ABILITY TO PROVIDE DETAILED AUDIT-  
INFORMATION ABOUT THE TRAFFIC THEY TRANSMIT;
- VULNERABLE TO ATTACK.

FIREWALL CAN BECOME A BOTTLENECK FOR A BIG SYSTEM. → MULTIPLE  
FIREWALLS IN PARALLEL, DIVIDED BY FUNCTION?

# FIREWALLS

## TYPES OF FILTERING POLICY

- DENY EVERYTHING, NOT SPECIFICALLY ALLOWED
- ALLOW EVERYTHING NOT SPECIFICALLY DENIED

## STRUCTURE

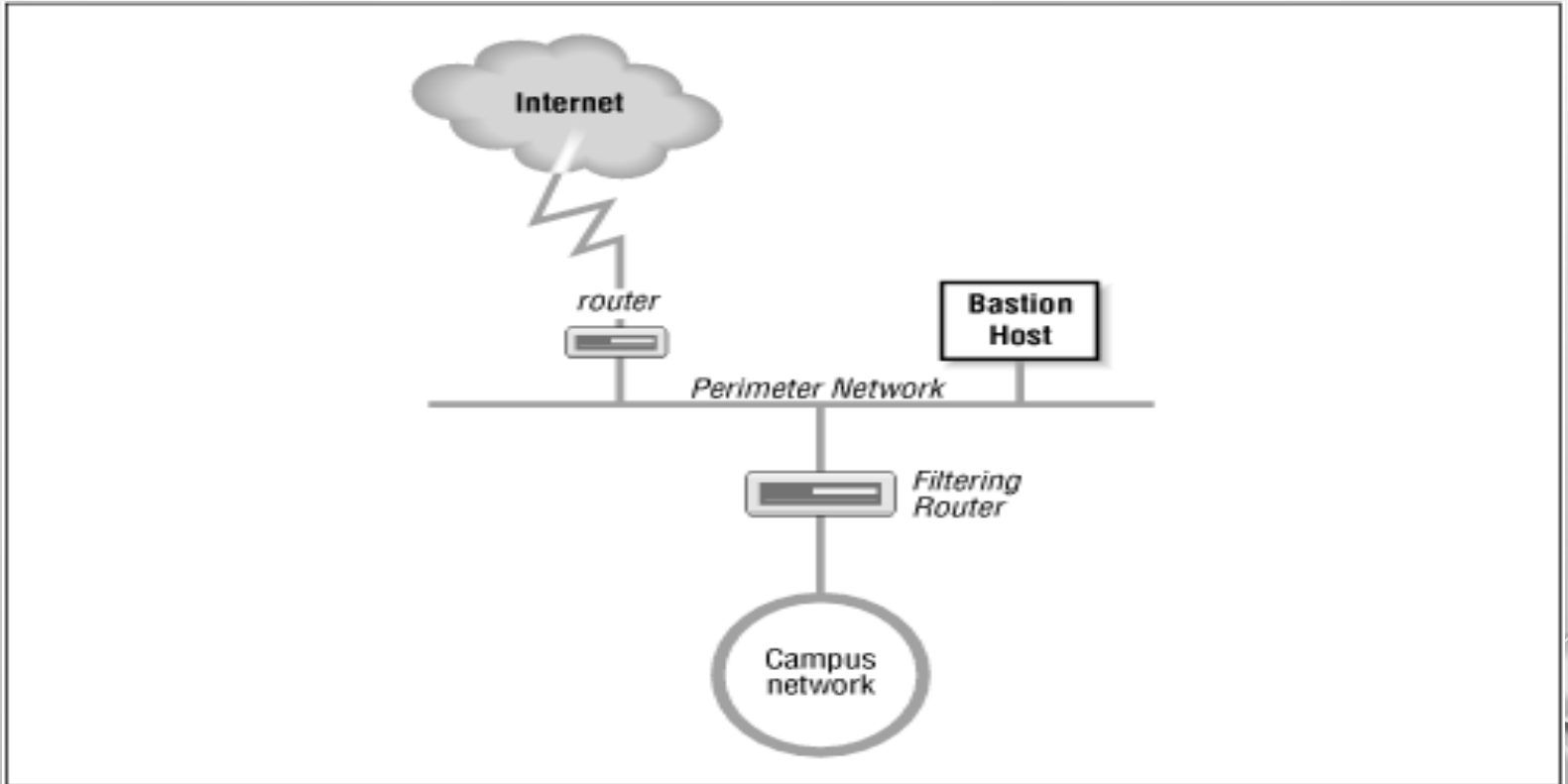
- ALL PACKETS INTO AND OUT OF THE PROTECTED NETWORK MUST PASS THROUGH THE FIREWALL
- FIREWALL CANNOT BE PENETRATED.

# FIREWALLS: THE COMMON ARCHITECTURE

- THE MOST COMMON FIREWALL ARCHITECTURE CONTAINS AT LEAST FOUR HARDWARE COMPONENTS:
  - AN (EXTERIOR) ROUTER,
  - A SECURE SERVER (CALLED A BASTION HOST),
  - AN EXPOSED NETWORK (CALLED A PERIMETER NETWORK),
  - AN (INTERIOR) FILTERING ROUTER.

# FIREWALL: AN EXAMPLE

- **SCREENED SUBNET TYPE OF FIREWALL:**



## FIREWALL: AN EXAMPLE (CONTINUED)

- EXTERIOR ROUTER: USES PACKET FILTERING TO ELIMINATE PACKETS COMING FROM THE EXTERNAL WORLD THAT HAVE A SOURCE ADDRESS THAT MATCHES THAT OF THE INTERNAL NETWORK.
- THE INTERIOR ROUTER DOES THE BULK OF THE ACCESS CONTROL WORK. IT FILTERS PACKETS ON
  - ADDRESS
  - PROTOCOL AND
  - PORT NUMBERSTO CONTROL THE SERVICES THAT ARE ACCESSIBLE TO AND FROM THE INTERIOR NETWORK.

# BASTION HOST

- A SECURE SERVER, SPECIFICALLY DESIGNED AND CONFIGURED TO WITHSTAND ATTACKS.
- GENERALLY HOSTS A SINGLE APPLICATION, FOR EXAMPLE A PROXY SERVER, AND ALL OTHER SERVICES AND END-USER-SOFTWARE ARE REMOVED OR LIMITED TO REDUCE THE THREAT TO THE COMPUTER.
- PROVIDES AN INTERCONNECTION POINT BETWEEN THE ENTERPRISE NETWORK AND THE OUTSIDE WORLD FOR SOME RESTRICTED SERVICES.
- RUNS AN IDS ON THE HOST; REGULAR SECURITY AUDIT
- USER ACCOUNTS, ESPECIALLY ROOT OR ADMINISTRATOR ACCOUNTS, ARE LOCKED DOWN; AUTHENTICATION USED FOR LOGGING; ENCRYPTED STORAGE

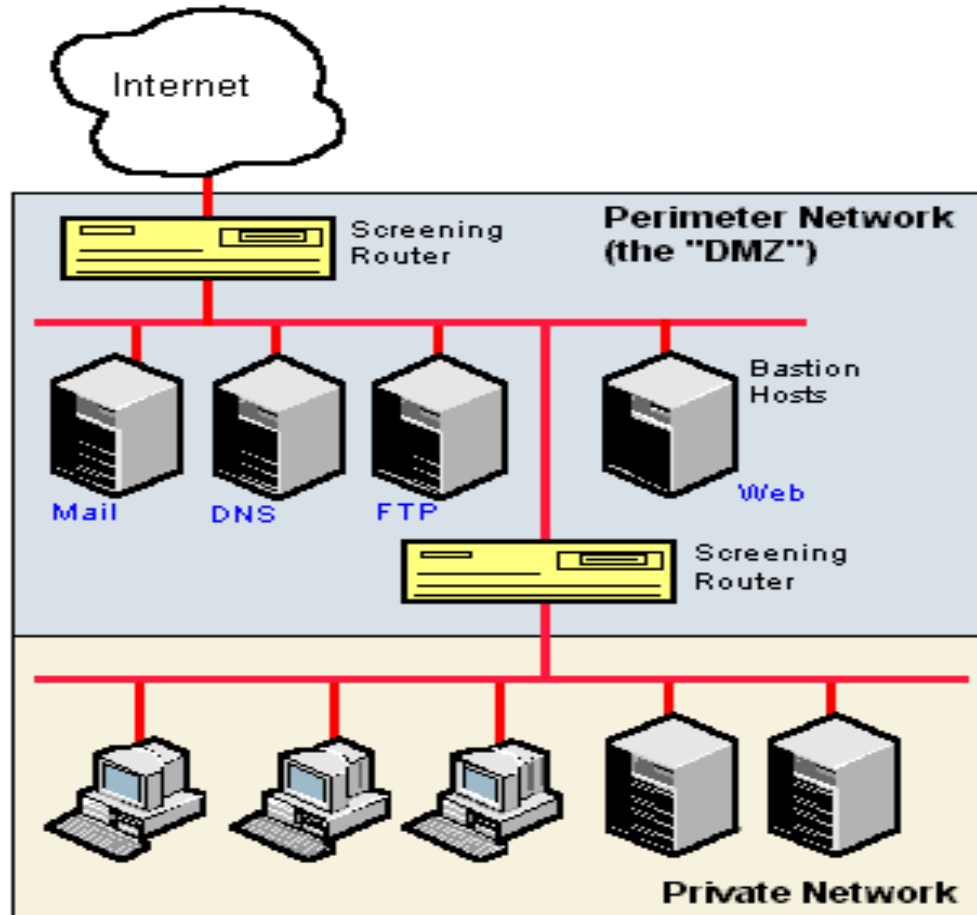
# BASTION HOST

....2

- SOME OF THE SERVICES THAT ARE RESTRICTED BY THE INTERIOR GATEWAY MAY BE ESSENTIAL FOR A USEFUL NETWORK. THOSE ESSENTIAL SERVICES ARE PROVIDED THROUGH THE BASTION HOST IN A SECURE MANNER. THE BASTION HOST PROVIDES SOME SERVICES *DIRECTLY*, SUCH AS
  - WEB SERVER
  - DOMAIN NAME SYSTEM SERVER,
  - E MAIL SERVICES,
  - ANONYMOUS FILE TRANSFER PROTOCOL
  - *PROXY SERVER*
  - *VPN SERVER*

# MULTIPLE BASTION HOSTS

.3



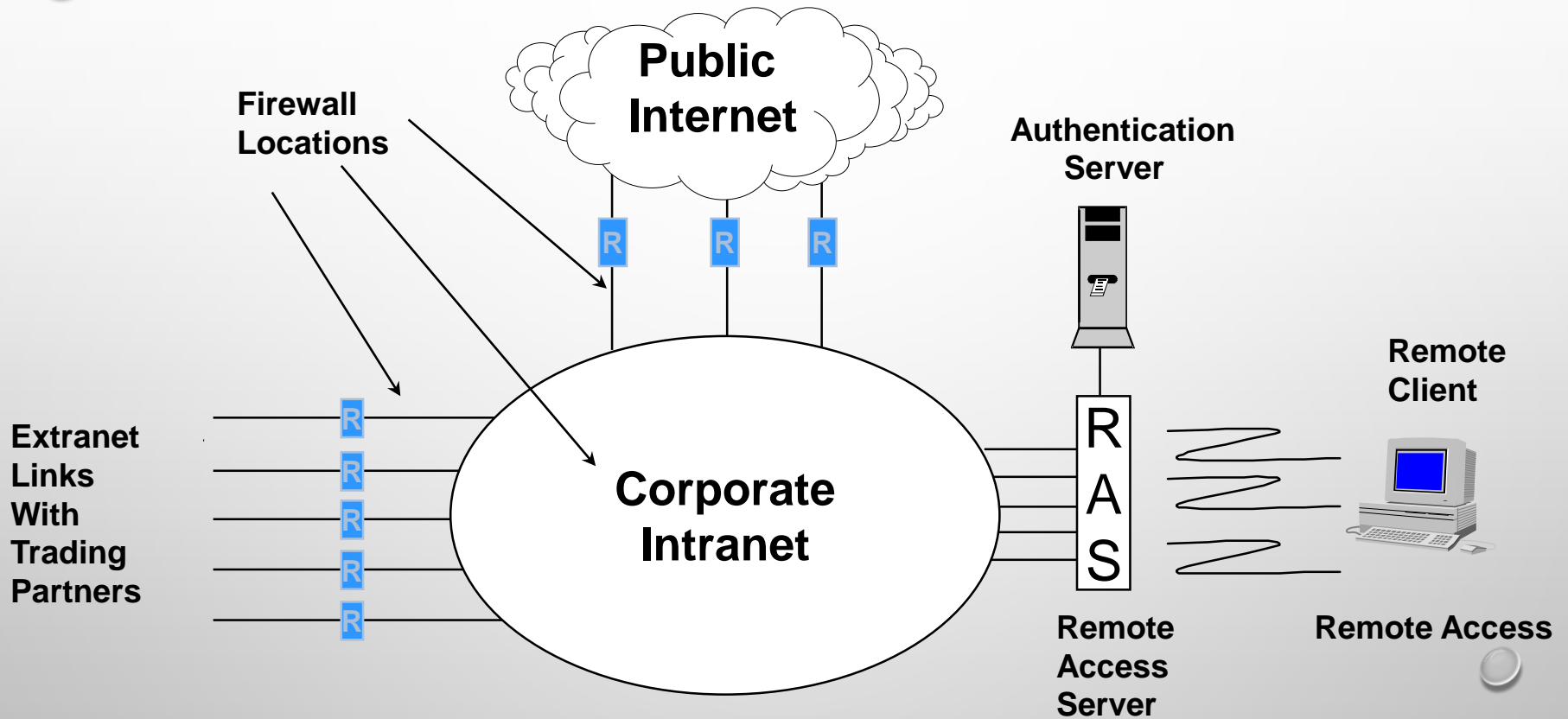


# BASTION HOST

....4

- WHEN THE BASTION HOST ACTS AS A PROXY SERVER, INTERNAL CLIENTS CONNECT TO THE OUTSIDE WORLD THROUGH THE BASTION HOSTS AND EXTERNAL SYSTEMS RESPOND BACK TO THE INTERNAL CLIENTS THROUGH THE HOST.
- AN ENTERPRISE: BASTION HOSTS ARE THE ONLY [HOST](#) COMPUTERS THAT ARE ALLOWED TO BE ADDRESSED DIRECTLY FROM THE PUBLIC NETWORK;
- DESIGNED TO SCREEN THE REST OF THE ENTERPRISE NETWORK FROM SECURITY EXPOSURE.

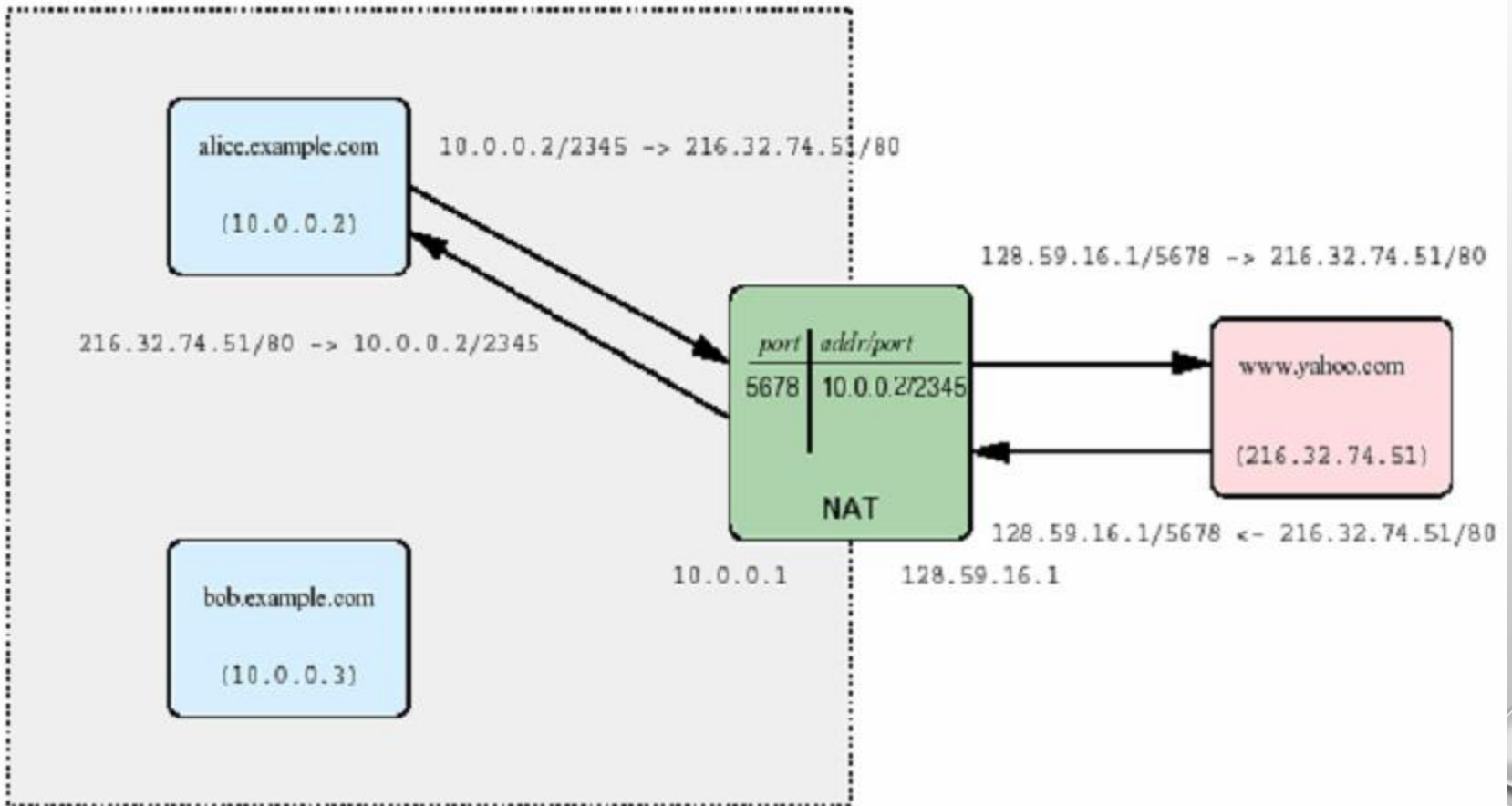
# Typical Enterprise Network Topology (without VPN)



# NETWORK ADDRESS TRANSLATOR

- **NA(P)T:** NETWORK ADDRESS (AND PORT) TRANSLATOR ARE *NOT* FIREWALLS, BUT CAN PREVENT ALL INCOMING CONNECTIONS

# NAT



# IPS VS IDS

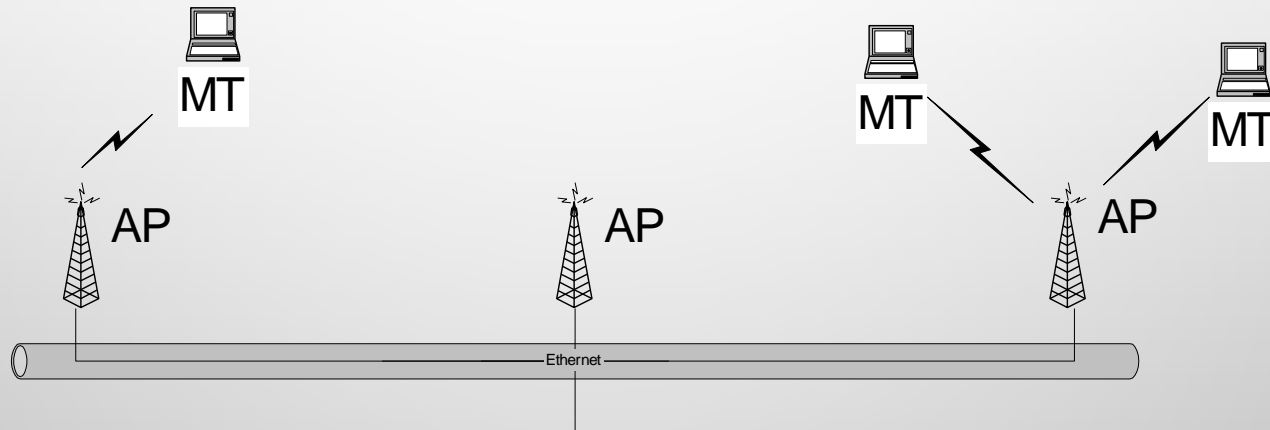
- NEW: IPS: INTRUSION PREVENTION SYSTEMS
- IDS: INTRUSION DETECTION SYSTEMS: IDS DEVICES SIT ON A MONITOR PORT AND SIMPLY REPORT PROBLEMS.
- WHILE AN IPS DEVICE TAKES ACTION, IDS PRODUCTS USUALLY JUST SEND AN ALERT TO AN IT STAFF PERSON, WHO MUST THEN EVALUATE THE ALERT AND TAKE ACTION.
- PROBLEM WITH IPS:
  - COSTLY
  - NEED TO BE PERIODICALLY TUNED SO THAT GOOD TRAFFIC IS NOT INADVERTENTLY DUMPED.

# WHAT IS IEEE 802.11?

- Standard for wireless local area networks (wireless LANs) developed in 1990 by IEEE
- Intended for home or office use (primarily indoor)
- 802.11 standard describes the MAC layer, while other substandards (802.11a, 802.11b) describe the physical layer
- Wireless version of the Ethernet (802.3) standard

# NETWORK SETUP

- Basic Network Setup is Cellular
- Mobile Terminals (MT) connect with Access Points (AP)



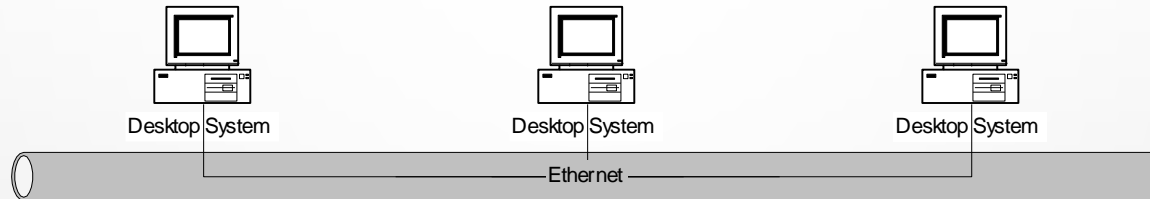
- Standard also supports ad-hoc networking where MT's talk directly to MT's

# IEEE 802.11 PHYSICAL LAYERS

	802.11b	802.11a
Modulation Scheme	DSSS	OFDM
Spectrum (GHz)	2.4 – 2.485	5.15-5.35, 5.725-5.825
Data Rate (Mbps)	1 – 11	6 - 54
Subchannels	11 overlapping	8 independent
Interference	Microwave, Cordless Phones, Bluetooth, HomeRF, Light Bulbs!	HyperLAN II
Availability	Today	Late August?
Cost	\$250 AP, \$100 PC Card	??? (same)



# MEDIA ACCESS CONTROL- ETHERNET



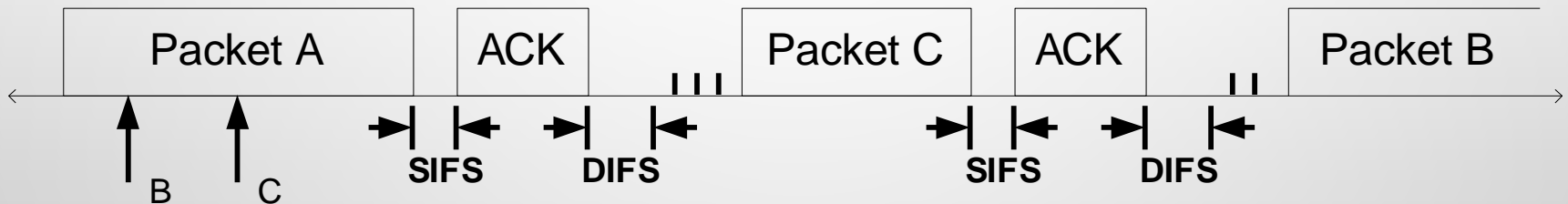
- CSMA/CD (CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION)
  - IF MEDIA IS SENSED IDLE, TRANSMIT
  - IF MEDIA IS SENSED BUSY, WAIT UNTIL IDLE AND THEN TRANSMIT IMMEDIATELY
- COLLISIONS CAN OCCUR IF MORE THAN ONE USER TRANSMITS AT THE SAME TIME
  - IF A COLLISION IS DETECTED, STOP TRANSMITTING.
  - RESCHEDULE TRANSMISSION ACCORDING TO EXPONENTIAL BACKOFF

# MEDIA ACCESS CONTROL (802.11)

- WOULD LIKE TO USE CSMA
  - NICE FOR BURSTY TRAFFIC
  - MAKE FOR SEAMLESS REPLACEMENT OF WIRED LANS WITH WIRELESS LANS
- USE CSMA, BUT CAN'T USE CD
  - $P_T/P_R$  RATIO IS TOO HIGH
  - DON'T WANT TO WASTE ENERGY ON MOBILES
- USE COLLISION AVOIDANCE INSTEAD
  - DON'T ALWAYS START TRANSMITTING IMMEDIATELY AFTER SOMEONE ELSE

# CSMA/CA DETAILS

- SIFS (SHORT INTERFRAME SPACE)
- DIFS (DISTRIBUTED INTERFRAME SPACE)



Scenario:

- B and C want to transmit, but A currently has control of medium
- B randomly selects 7 slots of backoff, C selects 4 slots
- C transmits first, then B

# FRAME FOR 802.11: FIELDS

- DURATION: PERIOD FOR WHICH THE FRAME AND ITS ACK WILL OCCUPY THE CHANNEL
- A MESSAGE MAY TRAVEL FROM SENDER NODE (AD1) → THE FIRST AP (AD2) → THE DEST AP (AD3) → THE FINAL DEST NODE (AD4).
- SEQ NO.: '12 BITS FOR FRAME' AND '4 BITS FOR FRAGMENT' IDENTIFICATION.

2 Bytes Frame Control	2 B Dura tion	6 B AD 1	6 B AD 2	6 B AD 3	2 B Seq No.	6 B AD 4	0-2312 Bytes DATA	4 B CRC
-----------------------------	---------------------	----------------	----------------	----------------	-------------------	----------------	----------------------	------------

# TYPES OF FRAMES FOR 802.11

TYPES OF FRAMES:

(I) DATA

(II) MANAGEMENT: USE ONE CELL OF A BASE STATION → NO AD 4 FIELD.

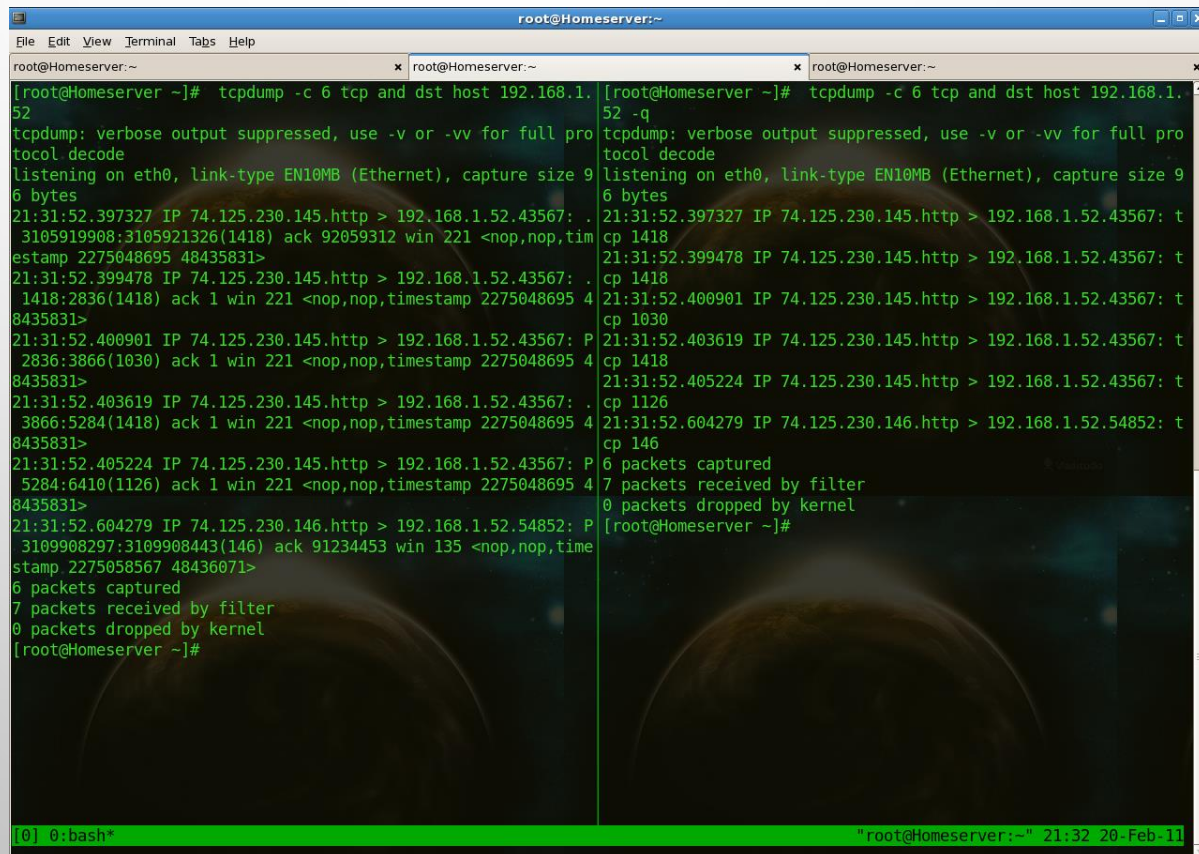
(III) CONTROL: HAVE ONLY 1 OR 2 ADDRESS FIELDS AND NO DATA AND SEQUENCE NO. FIELDS. USED FOR RTS/CTS/ACK.

The image features a light gray background with a subtle gradient. In the top-left and bottom-right corners, there are several realistic-looking water droplets of various sizes, rendered with soft shadows and highlights to give them a three-dimensional appearance. The text "NETWORK ANALYZER" is centered in the middle of the page.

# NETWORK ANALYZER

# TCPDUMP

- *TCPDUMP* PRINTS OUT A DESCRIPTION OF THE CONTENTS OF PACKETS ON A NETWORK INTERFACE THAT MATCH THE *BOOLEAN EXPRESSION*;



```
root@Homeserver:~  
[root@Homeserver ~]# tcpdump -c 6 tcp and dst host 192.168.1.52  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  
21:31:52.397327 IP 74.125.230.145.http > 192.168.1.52.43567: .  
 3105919908:3105921326(1418) ack 92059312 win 221 <nop,nop,timestamp 2275048695 48435831>  
21:31:52.399478 IP 74.125.230.145.http > 192.168.1.52.43567: .  
 1418:2836(1418) ack 1 win 221 <nop,nop,timestamp 2275048695 48435831>  
21:31:52.400901 IP 74.125.230.145.http > 192.168.1.52.43567: P  
 2836:3866(1030) ack 1 win 221 <nop,nop,timestamp 2275048695 48435831>  
21:31:52.403619 IP 74.125.230.145.http > 192.168.1.52.43567: .  
 3866:5284(1418) ack 1 win 221 <nop,nop,timestamp 2275048695 48435831>  
21:31:52.405224 IP 74.125.230.145.http > 192.168.1.52.43567: P  
 5284:6410(1126) ack 1 win 221 <nop,nop,timestamp 2275048695 48435831>  
21:31:52.604279 IP 74.125.230.146.http > 192.168.1.52.54852: P  
 3109908297:3109908443(146) ack 91234453 win 135 <nop,nop,timestamp 2275058567 48436071>  
6 packets captured  
7 packets received by filter  
0 packets dropped by kernel  
[root@Homeserver ~]#  
[0] 0:~#
```

```
root@Homeserver:~  
[root@Homeserver ~]# tcpdump -c 6 tcp and dst host 192.168.1.52 -q  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  
21:31:52.397327 IP 74.125.230.145.http > 192.168.1.52.43567: t  
cp 1418  
21:31:52.399478 IP 74.125.230.145.http > 192.168.1.52.43567: t  
cp 1418  
21:31:52.400901 IP 74.125.230.145.http > 192.168.1.52.43567: t  
cp 1030  
21:31:52.403619 IP 74.125.230.145.http > 192.168.1.52.43567: t  
cp 1418  
21:31:52.405224 IP 74.125.230.145.http > 192.168.1.52.43567: t  
cp 1126  
21:31:52.604279 IP 74.125.230.146.http > 192.168.1.52.54852: t  
cp 146  
6 packets captured  
7 packets received by filter  
0 packets dropped by kernel  
[root@Homeserver ~]#  
[0] 0:~#
```

# LOG FILES CREATED BY TCPDUMP: EXAMPLE: LINK LEVEL HEADERS

#TCPDUMP -E

THE OUTPUT IS AS FOLLOWS:

ETHERNET: SOURCE AND DEST ADDRESSES, PROTOCOL AND PACKET  
LENGTH

802.11: CONTROL, ALL THE ADDRESSES ( 2-4 USUALLY) AND PACKET  
LENGTH. (??)

OPTION -E → LINK LEVEL PACKET HEADER



# LOG FILES CREATED BY TCPDUMP:

## EXAMPLE: ARP

```
#TCPDUMP
```

```
ARP WHO-HAS HELIOS TELL SOLAR
```

```
ARP REPLY HELIOS IS-AT HELIOS
```

OPTION `-N` → NOT TO RESOLVE THE IP ADDRESS INTO NAMES

```
#TCPDUMP -N
```

```
ARP WHO-HAS 137.207.254.8 TELL 137.207.254.126
```

```
ARP REPLY 137.207.254.8 IS-AT A4:B5:C6:D7:E8:F9
```

```
#TCPDUMP -E
```

```
SOLAR BROADCAST 0806 64: ARP WHO-HAS HELIOS TELL SOLAR
```

```
HELIOS SOLAR 0806 64: ARP REPLY HELIOS IS-AT HELIOS.
```

FOR ETHERNET, TYPE = 0806, TOTAL LENGTH = 64 BYTES.

# LOG FILES CREATED BY TCPDUMP

## EXAMPLES OF TCP

### EXAMPLE OUTPUTS OF TCPDUMP:

- 15:35:23:830000 SRCHOST > 192.168.12.22: ICMP: ECHO REQUEST (TTL 251, ID 4224)
- 15:35:23:830000 ETH0 > SRCHOST.51200>  
DSTHOST.WWW:S 252 392 488: 252 392 488 (0)  
WIN 2048 <MSS 1024,NOP,NOP,TIMESTAMP  
1562755,0> (DF) (TTL 64, ID 5328)

*NOTE: MSS OPTION IS OF 4 BYTES. NOP IS ONE BYTE. TIMESTAMP TAKES 10 BYTES.*

# READING THE TCPDUMP LOG

- **15:35:23:830000**

TIME STAMP: 2 DIGIT HOURS, 2 DIGIT MINUTES, 2 DIGIT SECONDS, 6 DIGIT FRACTIONAL PART OF A SECOND

TO GIVE A UNIQUE IDENTITY TO THE EVENT, SINCE NUMEROUS EVENTS MAY HAPPEN AT ANY GIVEN SECOND

- TCPDUMP DOES NOT WRITE DATE STAMP

## ● READING THE TCPDUMP LOG (CONTINUED)

- `ETH0 >`

`ETH0` IS THE NAME OF THE INTERFACE BEING MONITORED.

(OTHER SIMILAR NAMES USED IN UNIX: `ETH0`, `HME1`, `QFE3`, `LAN0`)

`>` TELLS THE DIRECTION OF TRAFFIC

- `SCRHOST.51200`

(NAME OF THE SOURCE HOST).(PORT NUMBER)

- IF IP- ADDRESS-TO-NAME-RESOLUTION IS NOT AVAILABLE OR IF `TCPDUMP -N` OPTION IS USED, THE NAME MAY BE REPLACED BY THE IP ADDRESS.
- THE OPTION `-N` REQUESTS THAT HOST NAME RESOLUTION MAY NOT BE DONE.

## ● READING THE TCPDUMP LOG (CONTINUED)

- *WIN 2048* THE RECEIVING BUFFER SIZE OF SRCHOST, USED FOR FLOW CONTROL
- *<MSS 1024>* INFORMS THE DESTINATION HOST THAT THE PHYSICAL NETWORK OF SOURCE HOST WILL NOT RECEIVE MORE THAN 1024 BYTES OF TCP PAYLOAD.
  - IF 20 BYTES OF IP HEADER AND 24 BYTES OF TCP HEADER (INCLUDING 4 BYTES OF MSS OPTION) ARE INCLUDED, THE IP DATAGRAM MAY BE 1068 BYTES.
- *TIMESTAMP* OPTION PUTS THE TIMESTAMP OF THE SENDER. SINCE IT IS OF 10 BYTES, SO 2 BYTES OF NOP ARE USED.

# READING THE TCPDUMP LOG: IP HEADER FIELDS

## FROM IP HEADER:

- DF STANDS FOR DO NOT FRAGMENT.  
IF PACKETS ARE BEING FRAGMENTED, A FRAGMENT ID AND OFFSET APPEAR IN PLACE OF DF.
- TTL = 64
- IDENTIFICATION NUMBER: 5328
- *ICMP* APPEARS IN THE OUTPUT FOR INTERNET CONTROL MESSAGE PROTOCOL PACKETS.
- FOR MOST OF UDP RECORDS, THE WORD *UDP* APPEARS IN THE OUTPUT (EXCEPT IN TCPDUMPS OF UDP SERVICES FOR DNS AND SNMP).

<<<<<<

## ○ TCPDUMP OUTPUT: RELATIVE SEQUENCE NUMBERS

- RELATIVE SEQUENCE NUMBERS:
  - TCPDUMP OUTPUT CHANGES OVER FROM ABSOLUTE SEQUENCE NUMBERS TO RELATIVE SEQUENCE NUMBERS, AFTER THE FIRST TWO MESSAGES, GIVING ISNS, HAVE BEEN EXCHANGED.  
THUS INSTEAD OF THE SEQUENCE NUMBERS, WE MAY HAVE  $1:1025$  ( $1024$ ) WHICH INDICATES THAT RELATIVE TO ISN, THE 1<sup>ST</sup> THROUGH 1025<sup>TH</sup> (NOT INCLUDING 1025<sup>TH</sup>) BYTES HAVE BEEN SENT.
  - SIMILARLY *ACK 1* MEANS THAT ACKNOWLEDGEMENT NUMBER IS  $(ISN+1)$ .

# TCPDUMP OUTPUT: FOR A FRAGMENTED DATAGRAM CARRYING AN ICMP MESSAGE

EX: SRCHOST IS TO SEND AN ICMP ECHO REQUEST TO DESTHOST  
WITH 4200 BYTES OF ECHO DATA; TO BE SENT OVER ETHERNET  
AN IP DATAGRAM OF 4228 BYTES: AN ICMP MESSAGE OF  
4200 BYTES OF DATA AND 8 BYTES OF ICMP HEADER;  
SO THREE FRAGMENTS ARE REQUIRED.

FRAG1: 20 BYTES OF IP HEADER  
8 BYTES OF ICMP HEADER  
1472 BYTES OF ICMP DATA



# TCPDUMP OUTPUT: FOR A FRAGMENTED DATAGRAM (CONTINUED)

FRAG2: 20 BYTES OF IP HEADER

1 480 BYTES OF ICMP DATA

FRAG3: 20 BYTES OF IP HEADER

1 248 BYTES OF ICMP DATA

## **THE TCPDUMP OUTPUT FOR THE ECHO REQUEST:**

SRCHOST > DSTHOST: ICMP: ECHO REQUEST (FRAG 546768: 1480@0+)

SRCHOST > DSTHOST: (FRAG 546768: 1480@1480+)

SRCHOST > DSTHOST: (FRAG 546768: 1248@2960)

*NOTE: FOR THE FIRST PACKET: 1 480 BYTES INCLUDES 1 472 BYTES OF DATA AND 8 BYTES OF ICMP HEADER.*

## TCPDUMP OUTPUT: FOR A FRAGMENTED DATAGRAM (CONTINUED)

**FIRST FRAGMENT:** SINCE IT CONTAINS THE ICMP HEADER, TCPDUMP IS ABLE TO IDENTIFY IT AS AN ECHO REQUEST OF ICMP.

*FRAG 546768:* SPECIFIES THE IDENTIFICATION FIELD OF IP

*1480:* MEANS THAT THE FRAGMENT CONTAINS 1480 BYTES OF IP DATA

*@0:* MEANS THAT THE OFFSET IS 0 BYTES

*+:* MEANS THAT MFB FLAG IS SET

SIMILAR INTERPRETATION FOR THE TCPDUMP FOR THE **SECOND**  
AND **THIRD FRAGMENT**

# DENIAL OF SERVICE ATTACK USING FRAGMENTED PACKETS OF AN ICMP DATAGRAM

IF REPEATED FRAGMENTS WITH MFB = 1 ARE SENT TO A HOST  
AND

IF THE LAST FRAGMENT IS NOT SENT,

➔ THE HOST WOULD SLOW DOWN.

*THE REASSEMBLY TIMER WOULD NOT TIME OUT BECAUSE THE  
FRAGMENTS GO ON ARRIVING.*

*SOME ROUTERS HAVE FILTERS THAT FILTER OUT ECHO REQUESTS.  
BUT THEY MAY BE ABLE TO FILTER OUT THE FIRST FRAGMENT  
ONLY, UNLESS THE FILTER RETAINS THE STATE MEMORY TO  
LOCATE THE LATER FRAGMENTS, WITH THE SAME  
IDENTIFICATION FROM THE SAME SOURCE.*

*TWO WELL-KNOWN ATTACKS: PING OF DEATH AND TEARDROP*

# LIMITATIONS

- **TCPDUMP: HELPS FIND THE SENDER'S ADDRESS AS AVAILABLE IN THE IP PACKET; (IT MAY BE THE SPOOFED ADDRESS.)**
- **LIMITED BY HARDWARE: ETHERNET CARDS WILL DISCARD PACKETS WITH ERRONEOUS CRC. SO SUCH PACKETS CANNOT BE EXAMINED BY USING TCPDUMP.**

# FOR INSTALLING TCPDUMP: WHY ROOT PRIVILEGE?

- EVERY LINK LAYER INTERFACE COLLECTS PACKETS,
  - WITH ITS OWN ADDRESS OR
  - WITH A BROADCAST ADDRESS.
- TCPDUMP: REQUIRES THE INTERFACE TO BE IN THE PROMISCUOUS MODE; → REQUIRES ROOT-PRIVILEGE.

# TCPDUMP MANUAL

THE MANUAL OF COMMANDS WITH OPTIONS OF  
TCPDUMP\* CAN BE SEEN BY TYPING:

**MAN TCPDUMP**

- TCPDUMP & FILTERS:

NEARLY ANY FIELD IN AN IP DATAGRAM INCLUDING  
THE ACTUAL DATA PAYLOAD CAN BE USED TO LIMIT  
THE PURVIEW OF COLLECTED RECORDS (BY A FILTER).

*\*CREATED BY THE NETWORK RESEARCH GROUP AT LAWRENCE BERKELEY  
NATIONAL LAB*

## TCPDUMP: FILTER OPTIONS

- **TCPDUMP -N**
  - ASKS TCPDUMP NOT TO RESOLVE THE IP ADDRESS
- **TCPDUMP -N**
  - DON'T PRINT DOMAIN OF HOST NAMES, FOR INSTANCE PRINT CS INSTEAD OF CS.UWINDSOR.CA
- **TCPDUMP -A**
  - ATTEMPTS TO RESOLVE THE IP ADDRESS
- **TCPDUMP -C COUNT**
  - EXIT AFTER RECEIVING 'COUNT' NUMBER OF PACKETS

# TCPDUMP: FILING THE DUMP

- **TCPDUMP -F *FILENAME***
  - INDICATES THAT THE FILTER IS LOCATED IN THE FILE '*FILENAME*'.
- **TCPDUMP -W *FILENAME***
  - WILL TRANSFER THE RAW OUTPUT TO THE FILE IN BINARY FORMAT FROM THE DEFAULT NETWORK INTERFACE.
- **TCPDUMP -R *FILENAME***
  - WILL READ THE ABOVE RAW FILE.

A FILE USING -W OPTION CAN ONLY BE READ BY USING -R OPTION.



# FOUR LEVELS OF INFORMATION

- **TCPDUMP -v**      THE LESS VERBOSE OPTION  
TIME TO LIVE, IDENTIFICATION, TOTAL LENGTH AND OPTIONS IN AN IP PACKET ARE PRINTED. ALSO ENABLES ADDITIONAL PACKET INTEGRITY CHECKS SUCH AS VERIFYING THE IP AND ICMP HEADER CHECKSUM.
- **TCPDUMP -vv**      EVEN MORE VERBOSE OPTION
- **TCPDUMP -vvv**      MAXIMUM VERBOSE OPTION
- **TCPDUMP -Q**      THE QUIET OPTION

# SNAPSHOT LENGTH (SNAPLEN)

- SNAPLEN: THE EXACT NUMBER OF BYTES COLLECTED BY TCPDUMP. THE DEFAULT VALUE, FOR MOST OF THE IMPLEMENTATIONS, IS 68 BYTES. (SOLARIS DEFAULT IS 96)

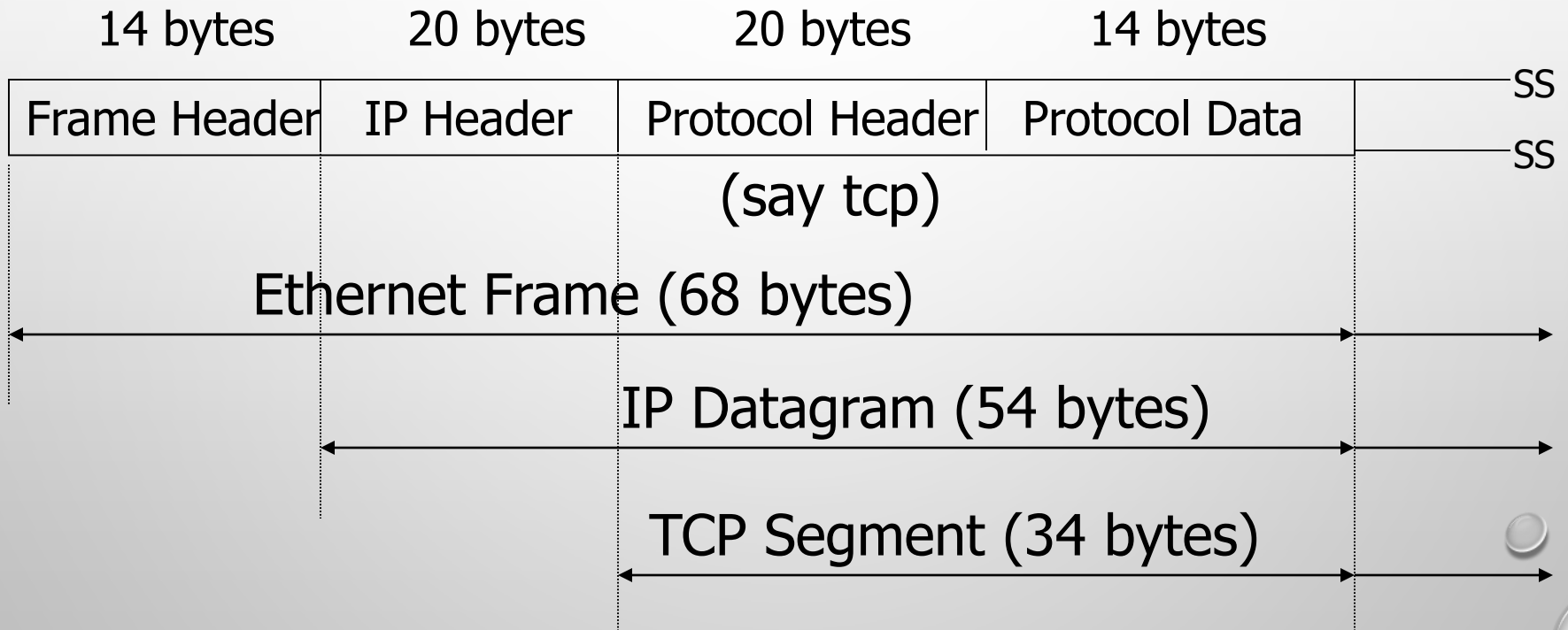
TO ALTER THE SNAPLEN (TO COLLECT NUMBER OF BYTES DIFFERENT FROM THE DEFAULT VALUE):

**TCPDUMP -S LENGTH**

- WHERE LENGTH=THE NUMBER OF BYTES TO BE COLLECTED  
IF LENGTH IS MADE 0 → THE WHOLE OF THE PACKET IS COLLECTED.

NAMESERVER REQUESTS → LEAD TO RESPONSES LARGER THAN 68 BYTES. SO -S OPTION MAY BE REQUIRED.

# EXAMPLE OF SNAPLEN



# HEXADECIMAL DUMPING

- THE OPTION **TCPDUMP -X** DUMPS THE DATAGRAM OF THE DEFAULT SIZE IN HEXADECIMAL FORMAT.
- TO CONVERT HEX FIELDS TO ASCII FOR CHARACTER, AND, DECIMAL FOR NUMERIC ONES, USE **TCPSHOW**.
- **TCPDUMP -X** FOR DUMPING IN HEX AND ASCII

# INTERFACE SELECTION

NORMALLY TCPDUMP LISTENS ON ALL THE INTERFACES OF THE SYSTEM. TO LIMIT IT TO SOME INTERFACE(S):

- **TCPDUMP -I *ETH0***

- ( 1.SOME VERSIONS OF TCPDUMP ALLOW THE IP ADDRESS TO BE WRITTEN RATHER THAN THE NAME OF THE INTERFACE.
2. WINDUMP HAS **-D**, WHICH DUMPS THE LIST OF THE INTERFACE CARDS AVAILABLE ON THE SYSTEM; RETURNS THE *NUMBER*, THE *NAME* AND THE DESCRIPTION.
3. DEFAULT VALUE IS INTERFACE NUMBER 1.)

# • ABSOLUTE SEQUENCE NUMBER OPTION

- **TCPDUMP -S**

FOR DISPLAYING ABSOLUTE TCP SEQUENCE NUMBERS

(**TCPDUMP -S LENGTH**

FOR GETTING A PARTICULAR SNAPLEN FROM THE PACKET.)

- **TCPDUMP -T**

FOR NOT PRINTING THE TIMESTAMP

NOTE: **UNDER LINUX:** YOU MUST BE ROOT OR IT MUST BE INSTALLED SETUID TO ROOT.

.....

- TO PRINT ALL PACKETS ARRIVING AT OR DEPARTING FROM A PARTICULAR HOST CALLED *SUNDOWN*:

## TCPDUMP HOST *SUNDOWN*

```
EX:# TCPDUMP HOST 192.168.2.165
```

```
TCPDUMP: LISTENING ON ETH0
```

```
19:16:04.817889 ARP WHO-HAS TSSOSS TELL PRIME
```

```
19:16:04.818025 ARP REPLY TSSOSS IS-AT 0:A0:C9:20:5B:FE
```

```
19:16:04.818182 PRIME.1219 > TSSOSS.TELNET:
```

```
S2506660519:2506660519(0) WIN 16384 <MSS
```

```
1460,NOP,NOP,SACKOK> (DF)
```

# TO OBTAIN FRAMES

WITH A SPECIFIC IP ADDRESS AND SPECIFIED PORT  
NUMBER

```
# TCPDUMP -NN HOST 192.168.2.165 AND PORT 23  
TCPDUMP: LISTENING ON ETH0
```

```
19:20:00.804501 192.168.2.10.1221 > 192.168.2.165.23:  
S2565655403:2565655403(0) WIN 16384 <MSS  
1460,NOP,NOP,SACKOK> (DF)
```

```
# TCPDUMP -NNE HOST 192.168.2.165 AND PORT 23 TCPDUMP:  
LISTENING ON ETH0
```

```
19:30:13.024247 0:5:5D:F4:9E:1F 0:A0:C9:20:5B:FE 0800 62:  
192.168.2.10.1223 > 192.168.2.165.23:  
S2718633695:2718633695(0) WIN 16384 <MSS  
1460,NOP,NOP,SACKOK> (DF)
```

**NOTE: 0800 IS FOR AN IP PACKET.**



## LOGICALLY COMPOUNDED OPTIONS: MORE EXAMPLES

- TO PRINT TRAFFIC BETWEEN *HELIOS* AND EITHER *HOT* OR *ACE*:

**TCPDUMP HOST *HELIOS* AND \ ( *HOT* OR  
*ACE* \ )**

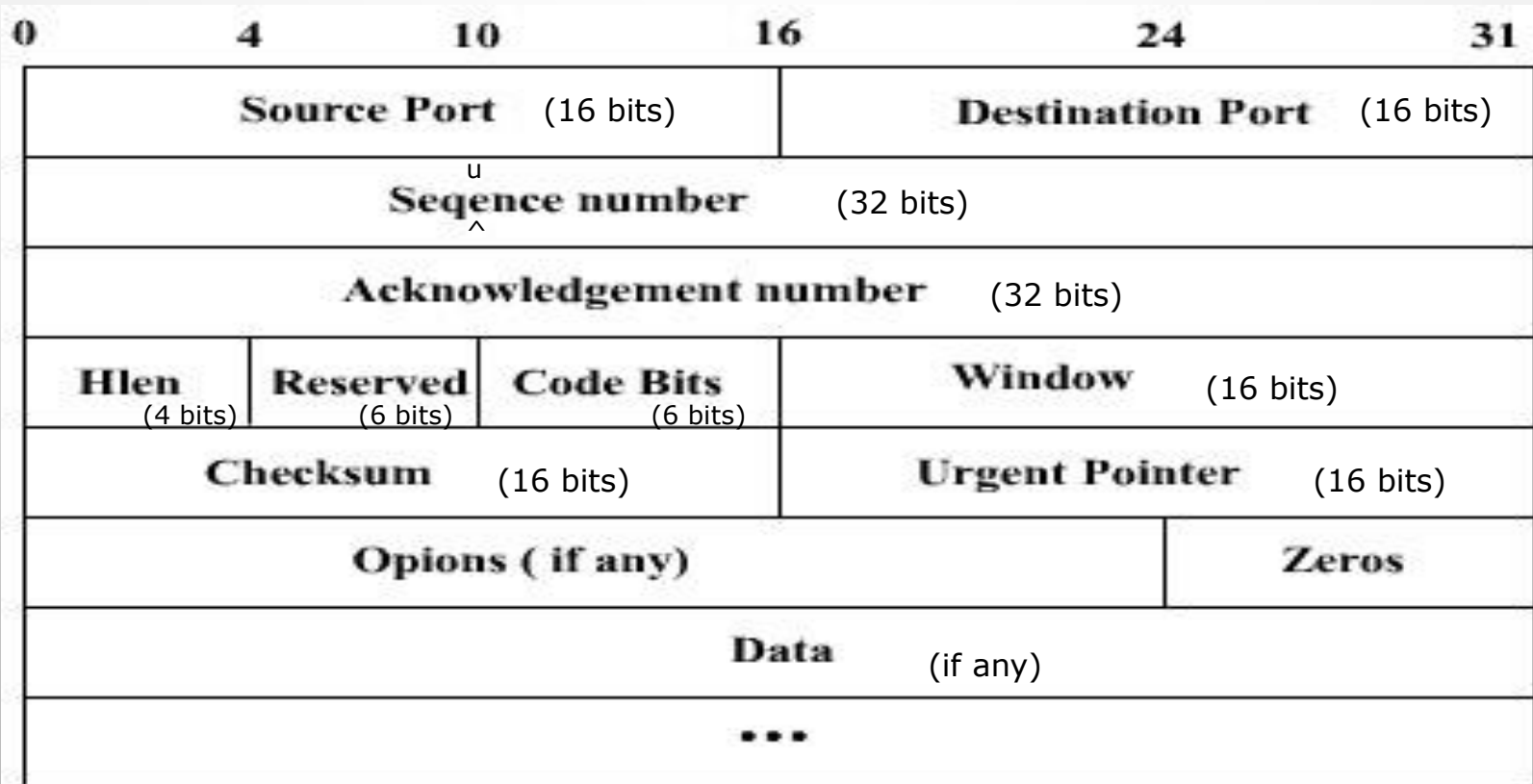
- TO PRINT ALL IP PACKETS BETWEEN *ACE* AND ANY  
HOST EXCEPT *HELIOS*:

**TCPDUMP IP HOST *ACE* AND NOT *HELIOS***

- TO PRINT ALL TRAFFIC BETWEEN LOCAL HOSTS  
AND HOSTS AT BERKELEY:

**TCPDUMP NET *UCB-ETHER***

# TCP SEGMENT: FORMAT



**The Header is of 20-60 bytes in size.**

# TCP SEGMENTS: FLAGS

CWR	Congestion Window reduced
ECE	ECN (Explicit Congestion Notification) Echo Flag ; <i>Ref: ECN: RFC 3168</i>
URG	Urgent Pointer Field is valid.
ACK	Acknowledgement Field is valid.
PSH	This segment requests a push.
RST	Reset the connection.
SYN	Synchronize Sequence Numbers. (for initiating the connection)
FIN	The Sender has reached the end of the byte stream. ( for closing the connection)

Out of the last 4 flags, normally only one is ON at a time.

# TCP FLAGS: EXAMPLE

- STARTING TO COUNT WITH 0, THE RELEVANT TCP CONTROL BITS ARE CONTAINED IN OCTET 13:

C|E|U|A|P|R|S|F ARE BITS 7 TO 0.

EX. 1: TO CAPTURE PACKETS WITH SYN BIT SET, THE 13<sup>TH</sup> BYTE WILL BE 00000010.

THEREFORE  $TCP[13] = 2$

EX. 2: TO CAPTURE PACKETS WITH SYN BIT SET, WHEN WE DON'T CARE IF ACK OR ANY OTHER TCP CONTROL BIT IS SET AT THE SAME TIME, THE 13<sup>TH</sup> BYTE WILL BE 00010010 .

THEREFORE  $TCP[13] = 18$

## EXAMPLES

## CONTINUED

- TO PRINT THE START AND END PACKETS (THE SYN AND FIN PACKETS) OF EACH TCP CONVERSATION THAT INVOLVES A NON-LOCAL HOST.

***TCPDUMP 'TCP[13] & 3 != 0 AND NOT SRC AND DST NET LOCALNET'***

*NOTE: TCP[13] MEANS 13<sup>TH</sup> OCTET OF TCP SEGMENT (WITH THE FIRST OCTET BEING THE 0<sup>TH</sup> OCTET)*

# MORE EXAMPLES

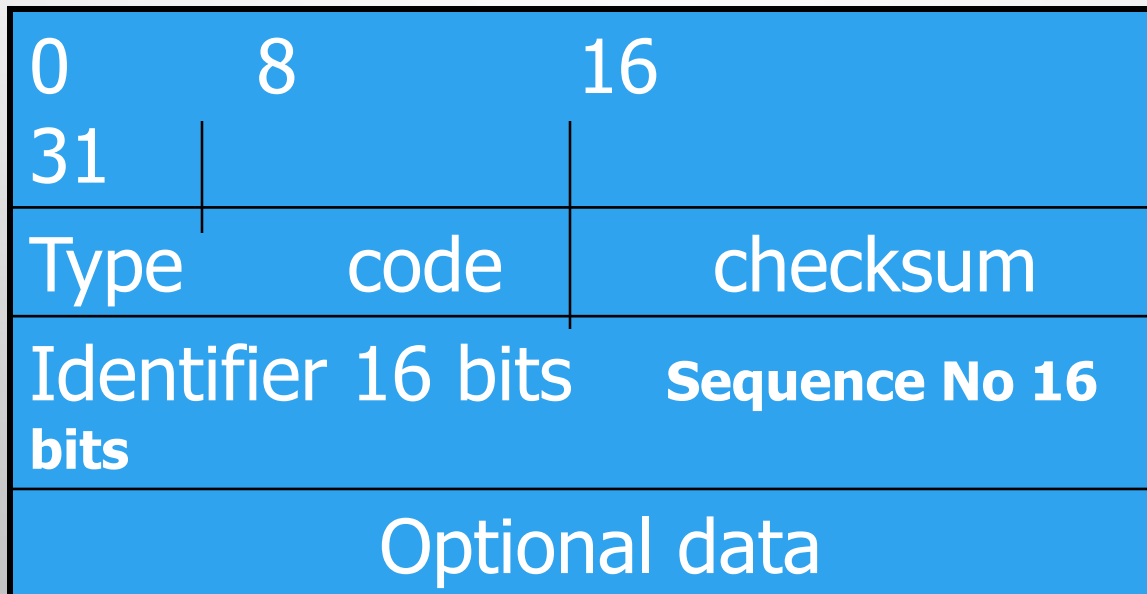
- `'IP[0] & 0XF != 5'` CATCHES ALL IP PACKETS WITH OPTIONS.
- `'IP[6:2] & 0X1FFF = 0'` CATCHES ONLY UNFRAGMENTED DATAGRAMS AND FRAG ZERO OF FRAGMENTED DATAGRAMS.

NOTE: `TCP[0]`: THE FIRST BYTE OF TCP HEADER

- **`TCPDUMP 'TCP[13] & 3 != 0 AND NOT SRC AND DST NET LOCALNET'`** → SYN AND FIN PACKETS OF A TCP CONVERASTION THAT INVOLVES A NON-LOCAL HOST.
- **`TCPDUMP 'GATEWAY SNUP AND IP[2:2] > 576'`** → GETS IP PACKETS LONGER THAN 576 BYTES AND SENT THROUGH ROUTER “SNUP”

# • PING USES ECHO REQUEST & REPLY

- **ECHO REQUEST AND REPLY**



## ANOTHER PACKET SNIFFER: WINDUMP

- **WINDUMP:** A WINDOW VERSION OF TCPDUMP, THE MOST POPULAR USED PACKET SNIFFER FOR UNIX SYSTEMS.
- WINDUMP IS RUN FROM THE COMMAND LINE; UNLESS YOU SAVED WINDUMP.EXE TO A DIRECTORY IN YOUR PATH, YOU WILL NEED TO BE IN THE SAME DIRECTORY TO RUN THE PROGRAM OR ENTER THE COMPLETE PATH.
- WHILE INSTALLING WINDOWS, INSTALL WINPCAP, TO ACCESS WINDUMP
- USE THE COMMAND: **WINDUMP -?** FOR HELP FILE.



# RUNNING WINDUMP

- IF WINDUMP GIVES AN ERROR MESSAGE ABOUT THE ADAPTER OR DEVICE, USE:

- **WINDUMP -D**

TO GET A LISTING OF THE DEVICES, WINDUMP RECOGNIZES.

- USE THE COMMAND:

- **WINDUMP -I *DEVICE\_NUM***

TO DIRECT WINDUMP TO LISTEN USING THE SELECTED DEVICE; ALSO USED TO POINT TO A SPECIFIC NETWORKING DEVICE, FOR THE CASE WHERE ONE HAS TO CHOOSE OUT OF MORE THAN ONE NICS OR MODEM.

*REFERENCE: [HTTP://WINDUMP.POLITO.IT/DOCS/MANUAL.HTM](http://windump.polito.it/docs/manual.htm)*

# WHAT IS WIRESHARK?

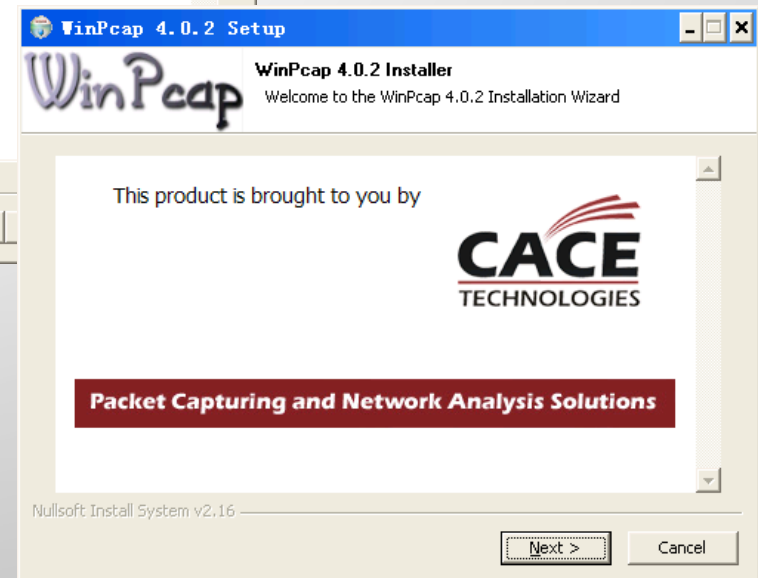
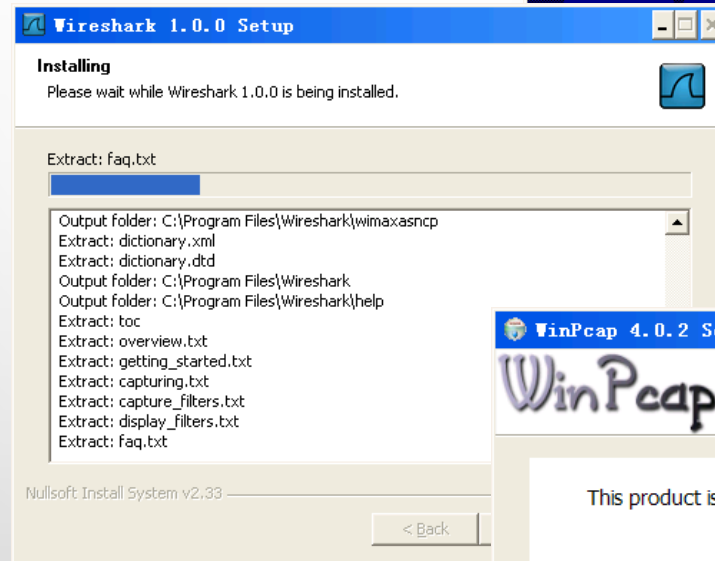
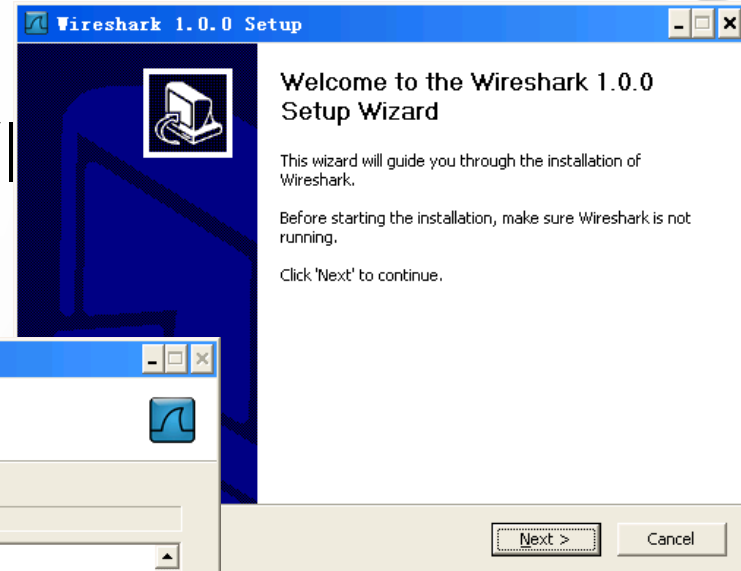
- WIRESHARK IS A NETWORK PACKET/PROTOCOL ANALYZER.
  - A NETWORK PACKET ANALYZER WILL TRY TO CAPTURE NETWORK PACKETS AND TRIES TO DISPLAY THAT PACKET DATA AS DETAILED AS POSSIBLE.
- WIRESHARK IS PERHAPS ONE OF THE BEST OPEN SOURCE PACKET ANALYZERS AVAILABLE TODAY FOR **UNIX** AND **WINDOWS**.

# SOME INTENDED PURPOSES

- NETWORK ADMINISTRATORS USE IT TO **TROUBLESHOOT NETWORK PROBLEMS**
- NETWORK SECURITY ENGINEERS USE IT TO **EXAMINE SECURITY PROBLEMS**
- DEVELOPERS USE IT TO **DEBUG PROTOCOL IMPLEMENTATIONS**
- PEOPLE USE IT TO **LEARN NETWORK PROTOCOL INTERNALS**
- WIRESHARK ISN'T AN INTRUSION DETECTION SYSTEM.
- WIRESHARK WILL NOT MANIPULATE THINGS ON THE NETWORK, IT WILL ONLY "MEASURE" THINGS FROM IT.

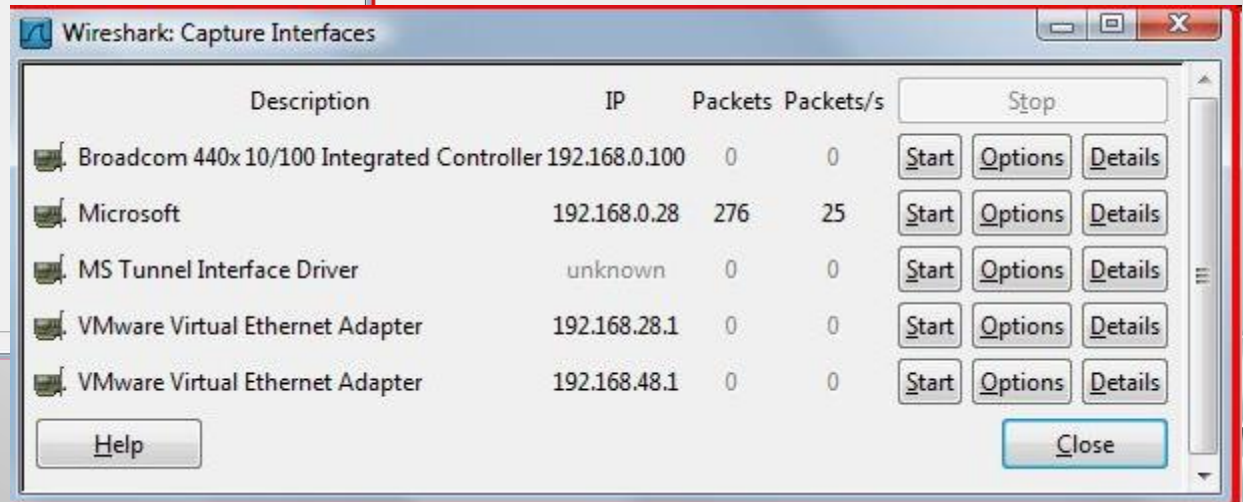
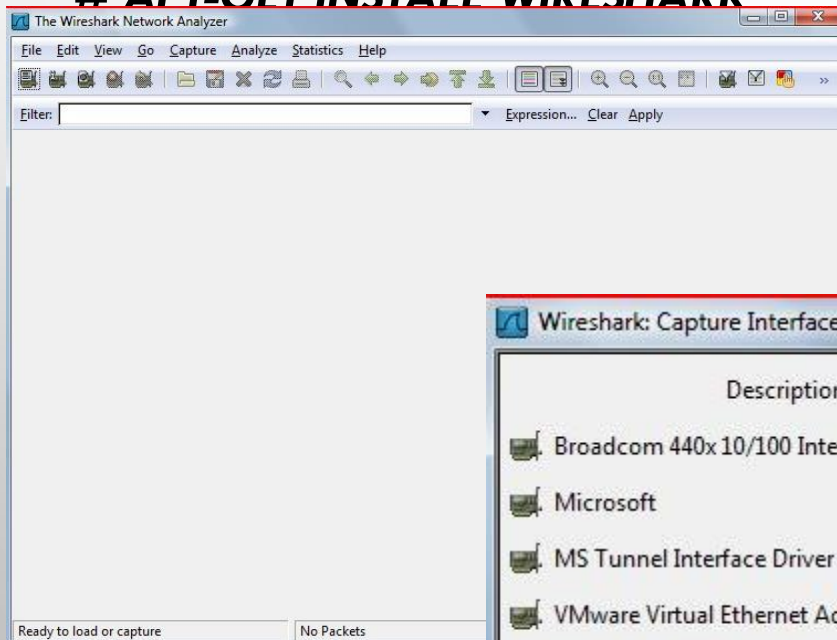
# INSTALL UNDER WIN

- DOWNLOAD
- INSTALL



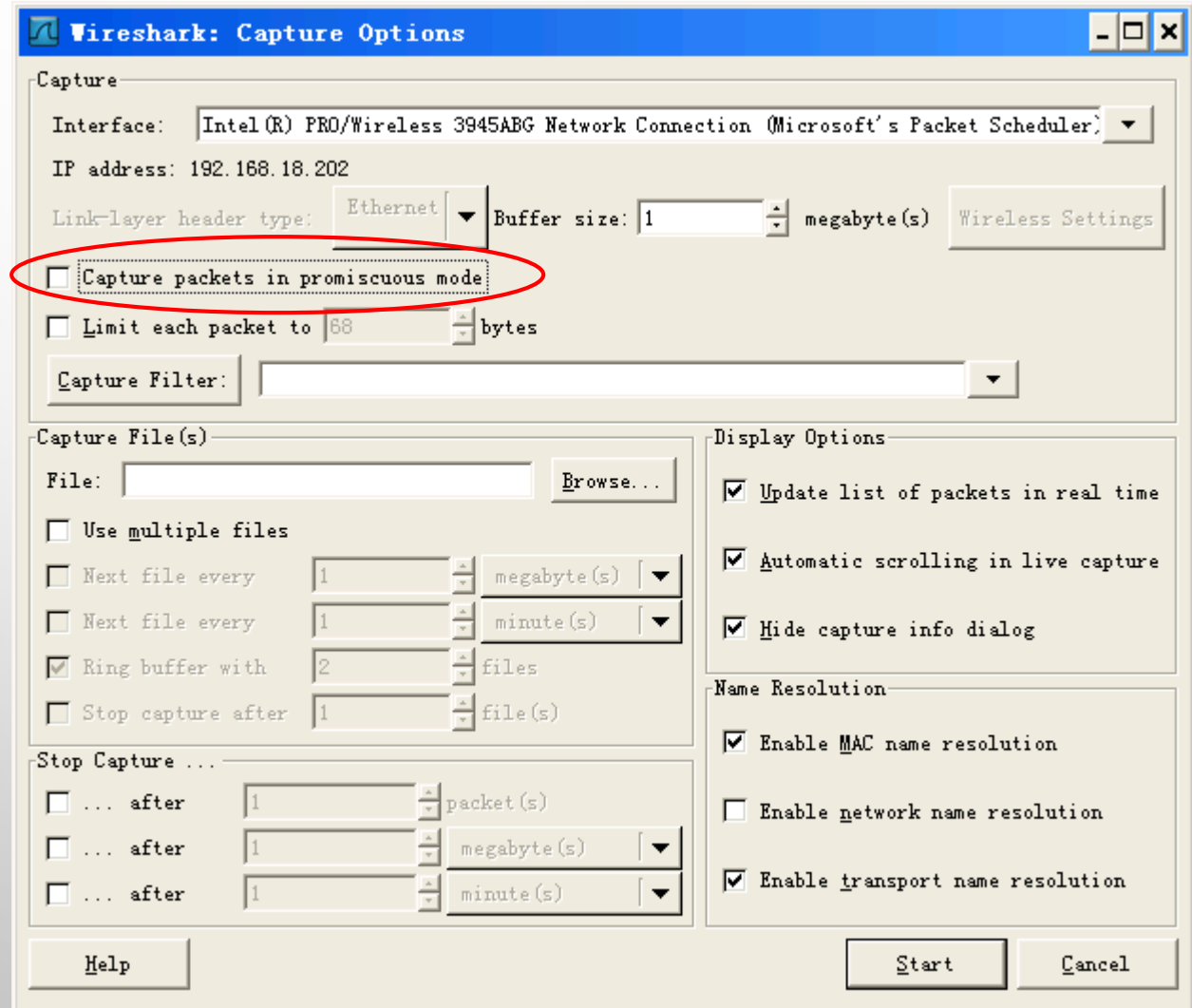
# INSTALL UNDER DEBIAN/ UBUNTU

- **# APT-GET INSTALL WIRESHARK**



# CONFIGURATION

This checkbox allows you to specify that Wireshark should put the interface in **promiscuous** mode when capturing. If you do not specify this, Wireshark will only capture the packets going to or from your computer (not all packets on your LAN segment).



The image shows the 'Wireshark: Capture Options' dialog box. The 'Capture' section is highlighted with a red oval around the 'Capture packets in promiscuous mode' checkbox. The interface is set to 'Intel (R) PRO/Wireless 3945ABG Network Connection (Microsoft's Packet Scheduler)' with an IP address of '192.168.18.202'. The link-layer header type is 'Ethernet' and the buffer size is '1 megabyte(s)'. The 'Capture File(s)' section shows a file name field, a 'Browse...' button, and options for multiple files, ring buffer, and stop capture after. The 'Display Options' section has checkboxes for 'Update list of packets in real time', 'Automatic scrolling in live capture', and 'Hide capture info dialog'. The 'Name Resolution' section has checkboxes for 'Enable MAC name resolution', 'Enable network name resolution', and 'Enable transport name resolution'. Buttons for 'Help', 'Start', and 'Cancel' are at the bottom.

**Wireshark: Capture Options**

Capture

Interface: Intel (R) PRO/Wireless 3945ABG Network Connection (Microsoft's Packet Scheduler) ▾

IP address: 192.168.18.202

Link-layer header type: Ethernet ▾ Buffer size: 1 megabyte(s) Wireless Settings

Capture packets in promiscuous mode

Limit each packet to 68 bytes

Capture Filter: ▾

Capture File(s)

File:  Browse...

Use multiple files

Next file every 1 megabyte(s) ▾

Next file every 1 minute(s) ▾

Ring buffer with 2 files

Stop capture after 1 file(s)

Stop Capture ...

... after 1 packet(s)

... after 1 megabyte(s) ▾

... after 1 minute(s) ▾

Display Options

Update list of packets in real time

Automatic scrolling in live capture

Hide capture info dialog

Name Resolution

Enable MAC name resolution

Enable network name resolution

Enable transport name resolution

Help Start Cancel

# IMPORTANT

- TURN PROMISCUOUS MODE OFF!
- IF YOU'RE AT WORK, YOUR NETWORK ADMINISTRATOR MAY SEE YOU RUNNING IN PROMISCUOUS MODE AND SOMEBODY MAY DECIDE TO FIRE YOU FOR THAT.

# MORE RESOURCE

- [HTTP://WIKI.WIRESHARK.ORG](http://wiki.wireshark.org)
- [HTTP://WIKI.WIRESHARK.ORG/SAMPLECAPTURES](http://wiki.wireshark.org/samplecaptures)
- SEARCH “WIRESHARK TUTORIAL”



# SO WHAT IS WIRESHARK?

- PACKET SNIFFER/PROTOCOL ANALYZER
- OPEN SOURCE NETWORK TOOL
- LATEST VERSION OF THE ETHEREAL TOOL



# WHAT IS TSHARK?

- THE COMMAND-LINE BASED PACKET CAPTURE TOOL
- EQUIVALENT TO WIRESHARK



# WIRESHARK INTERFACE

Tucker Ellis & West aaa.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
2	0.746308	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
3	0.751270	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
4	9.318731	silicom_01:6e:bd	Broadcast	ARP	who has 192.168.1.1? Tell 19
5	0.000664	Castlene_00:34:56	silicom_01:6e:bd	ARP	192.168.1.1 is at 00:30:54:00
6	0.000026	192.168.1.2	192.168.1.1	DNS	Standard query A sip.cybercit
7	0.995383	192.168.1.2	192.168.1.1	DNS	Standard query A sip.cybercit
8	2.003039	192.168.1.2	192.168.1.1	DNS	Standard query A sip.cybercit
9	0.169652	192.168.1.1	192.168.1.2	DNS	Standard query response A 212
10	1.006246	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
11	0.996899	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
12	2.003024	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
13	0.992343	Castlene_00:34:56	silicom_01:6e:bd	ARP	who has 192.168.1.2? Tell 19
14	0.000049	silicom_01:6e:bd	Castlene_00:34:56	ARP	192.168.1.2 is at 00:e0:ed:01
15	1.010378	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
16	4.005777	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
17	8.002019	192.168.1.2	192.168.1.1	DNS	Standard query PTR 1.0.0.127.
18	0.001489	192.168.1.1	192.168.1.2	DNS	Standard query response PTR 1
19	0.001640	192.168.1.2	212.242.33.35	SIP	Request: REGISTER sip:sip.cyb

Header length: 20 bytes

Differentiated services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)

Total Length: 78

Identification: 0x698c (27020)

Flags: 0x00

Fragment offset: 0

```
0000 ff ff ff ff ff 00 e0 ed 01 6e bd 08 00 45 00 ..... ..n...E.
0010 00 4e 69 8c 00 80 80 11 4c c1 c0 a8 01 02 c0 a8 .N[..... L.....
0020 01 ff 00 89 00 89 00 3a 5b b4 84 e7 01 10 00 01 .....: [.....
0030 00 00 00 00 00 00 20 45 46 45 41 45 4a 46 50 45 ..... E FEDEJFPE
0040 45 45 50 45 4e 45 42 45 4a 45 4f 43 41 43 41 43 FEPEDEB EEOCACAL
0050 41 43 41 43 41 42 4d 00 00 20 00 01 ACACABM. ....
```

Identification (ip.id), 2 bytes

Packets: 691 Displayed: 691 Marked: 0

Profile: Default

Highlighted packets here

Are shown here as well

command  
menus

display filter  
specification

listing of  
captured  
packets

details of  
selected  
packet  
header

packet content  
in hexadecimal  
and ASCII

The screenshot shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. Below the menu is a toolbar with various icons for file operations, navigation, and analysis. A filter field is present with the text "Expression... Clear Apply".

The main display area is divided into three sections:

- Packets List:** A table with columns for No., Time, Source, Destination, Protocol, and Info. It shows several packets, with packet 4 selected. The selected packet is a GET request for /news/ HTTP/1.1.
- Packet Details:** A tree view showing the structure of the selected packet. It includes Ethernet II, Internet Protocol, Transmission Control Protocol, and Hypertext Transfer Protocol. The HTTP section is expanded to show the request line, host, user-agent, accept, accept-language, accept-encoding, accept-charset, keep-alive, connection, referer, and cookie.
- Packet Bytes:** A hex dump of the selected packet's data, showing hexadecimal values on the left and their corresponding ASCII characters on the right.

The status bar at the bottom indicates the current file path and statistics: File: C:\DOCUME~1\FALLAW~1\LOCALS~1\Temp\{the\X\00a00324\ 453 KB 00:00:00; P: 671 B; 671 M; 0 Drops: 0

# STATUS BAR

Tucker Ellis & West aaa.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
2	0.746308	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
3	0.751270	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
4	9.318731	Silicom_01:6e:bd	Broadcast	ARP	who has 192.168.1.1? Tell 19
5	0.000664	Castlene_00:34:56	Silicom_01:6e:bd	ARP	192.168.1.1 is at 00:30:54:00
6	0.000026	192.168.1.2	192.168.1.1	DNS	Standard query A sip.cybercit
7	0.995383	192.168.1.2	192.168.1.1	DNS	Standard query A sip.cybercit
8	2.003039	192.168.1.2	192.168.1.1	DNS	Standard query A sip.cybercit
9	0.169652	192.168.1.1	192.168.1.2	DNS	Standard query response A 212
10	1.006246	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
11	0.996899	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
12	2.003024	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
13	0.992343	Castlene_00:34:56	Silicom_01:6e:bd	ARP	who has 192.168.1.2? Tell 19
14	0.000049	Silicom_01:6e:bd	Castlene_00:34:56	ARP	192.168.1.2 is at 00:e0:ed:01
15	1.010378	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
16	4.005777	192.168.1.2	192.168.1.1	DNS	Standard query SRV _sip._udp.
17	8.002019	192.168.1.2	192.168.1.1	DNS	Standard query PTR 1.0.0.127.
18	0.001489	192.168.1.1	192.168.1.2	DNS	Standard query response PTR 1
19	0.001640	192.168.1.2	212.242.33.35	SIP	Request: REGISTER sip:sip.cyt

Header length: 20 bytes  
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)  
Total Length: 78  
Identification: 0x698c (27020)  
Flags: 0x00  
Fragment offset: 0

```
0000 ff ff ff ff ff ff 00 e0 ed 01 6e bd 08 00 45 00 .....n...E.  
0010 00 4e 69 8c 00 00 80 11 4c c1 c0 a8 01 02 c0 a8 .N...L.....  
0020 01 ff 00 89 00 89 00 3a 5b b4 84 e7 01 10 00 01 .....[.....  
0030 00 00 00 00 00 00 20 45 46 45 44 45 4a 46 50 45 .....E FEDEJFPE  
0040 45 45 50 45 4e 45 42 45 4a 45 4f 43 41 43 41 43 EEPENEJEJEOCACAC  
0050 41 43 41 43 41 42 4d 00 00 20 00 01 ACACABM. ...
```

Identification (ip.id), 2 bytes      Packets: 691 Displayed: 691 Marked: 0      Profile: Default

# CAPTURE OPTIONS

**Tucker Ellis & West Wireshark: Capture Options**

**Capture**

Interface: Intel(R) PRO/1000 PL Network Connection: \Device\NPF\_{97708CAB-FF09-4180-9...}

IP address: 10.1.14.117

Link-layer headertype: Ethernet Buffer size: 1 megabyte(s) [Wireless Settings](#)

Capture packets in promiscuous mode

Limit each packet to 68 bytes

Capture Filter:

**Capture File(s)**

File:  [Browse...](#)

Use multiple files

Next file every 1 megabyte(s)

Next file every 1 minute(s)

Ring buffer with 2 files

Stop capture after 1 file(s)

**Stop Capture ...**

... after 1 packet(s)

... after 1 megabyte(s)

... after 1 minute(s)

**Display Options**

Update list of packets in real time

Automatic scrolling in live capture

Hide capture info dialog

**Name Resolution**

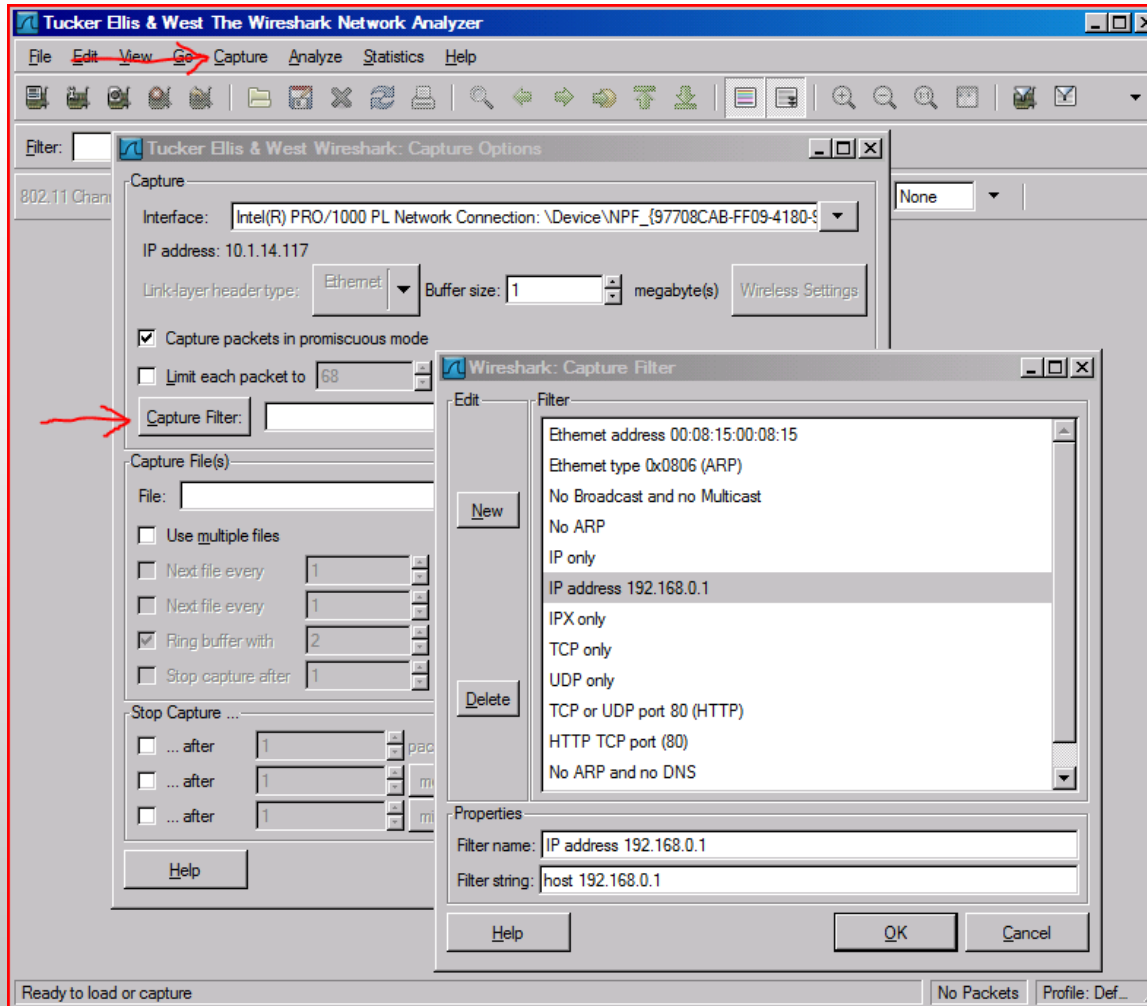
Enable MAC name resolution

Enable network name resolution

Enable transport name resolution

[Help](#) [Start](#) [Cancel](#)

# CAPTURE FILTER



# CAPTURE FILTER EXAMPLES

**HOST 10.1.11.24**

**HOST 192.168.0.1 AND HOST 10.1.11.1**

**TCP PORT HTTP**

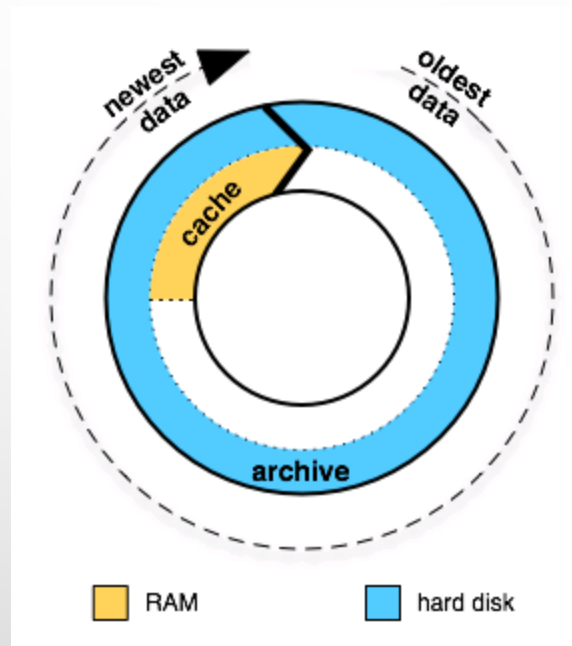
**IP**

**NOT BROADCAST NOT MULTICAST**

**ETHER HOST 00:04:13:00:09:A3**



# CAPTURE BUFFER USAGE



**Tucker Ellis & West Wireshark: Capture Options**

**Capture**

Interface: Intel(R) PRO/1000 PL Network Connection: \Device\NPF\_{97708CAB-FF09-4180-5...}

IP address: 10.1.14.117

Linklayer header type: Ethernet Buffer size: 1 megabyte(s) Wireless Settings

Capture packets in promiscuous mode

Limit each packet to 68 bytes

Capture Filter:

**Capture File(s)**

File: c:\cap1.pcap Browse...

Use multiple files

Next file every 1 megabyte(s)

Next file every 1 minute(s)

Ring buffer with 2 files

Stop capture after 1 file(s)

**Stop Capture ...**

... after 1 packet(s)

... after 1 megabyte(s)

... after 1 minute(s)

**Display Options**

Update list of packets in real time

Automatic scrolling in live capture

Hide capture info dialog

**Name Resolution**

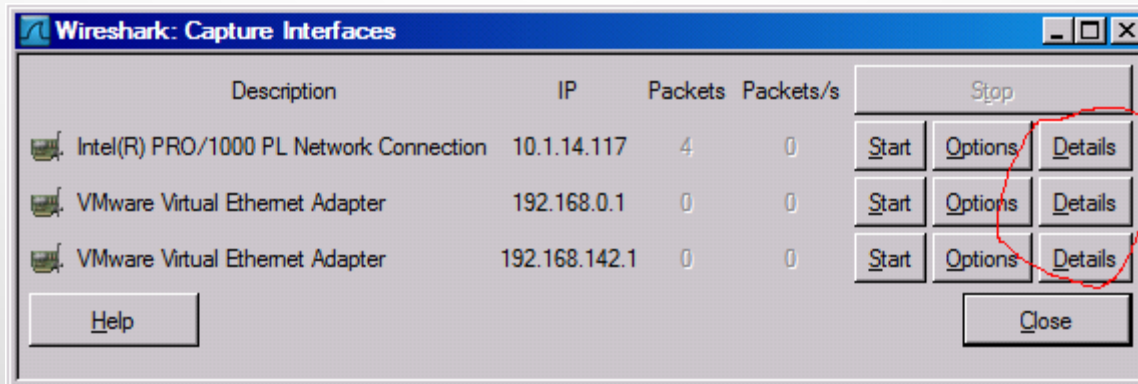
Enable MAC name resolution

Enable network name resolution

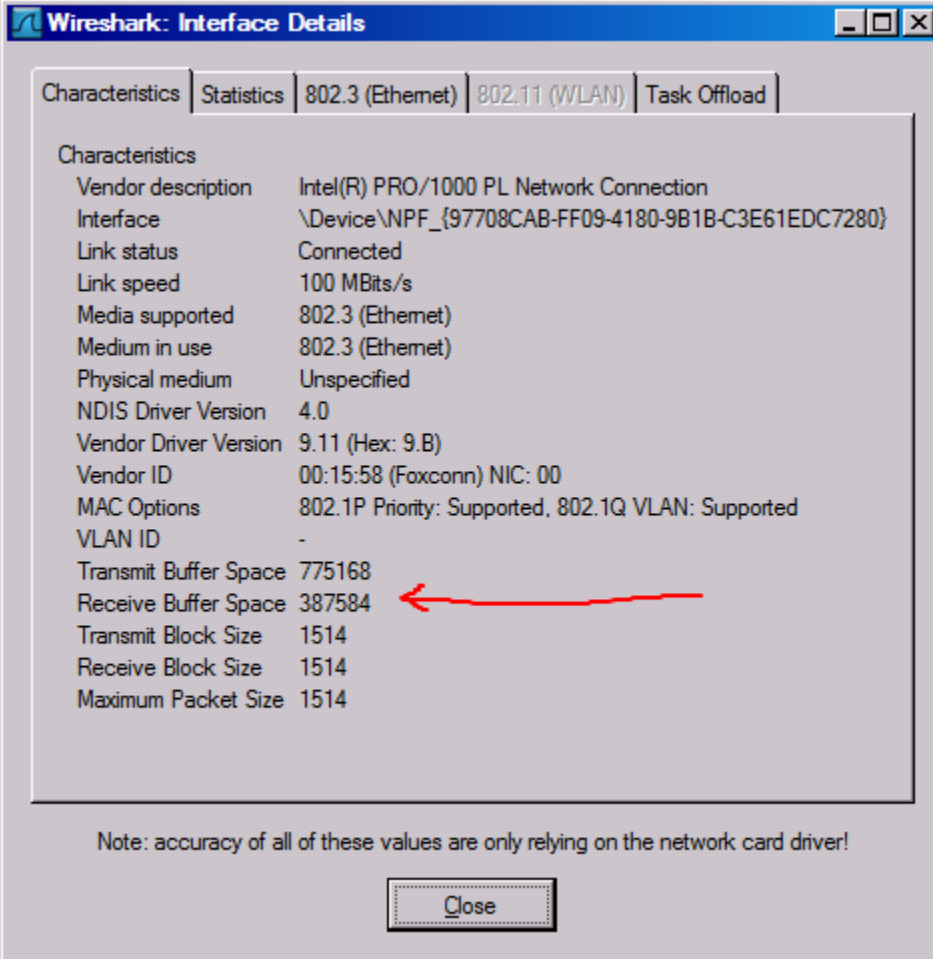
Enable transport name resolution

Help Start Cancel

# CAPTURE INTERFACES



# INTERFACE DETAILS: CHARACTERISTICS



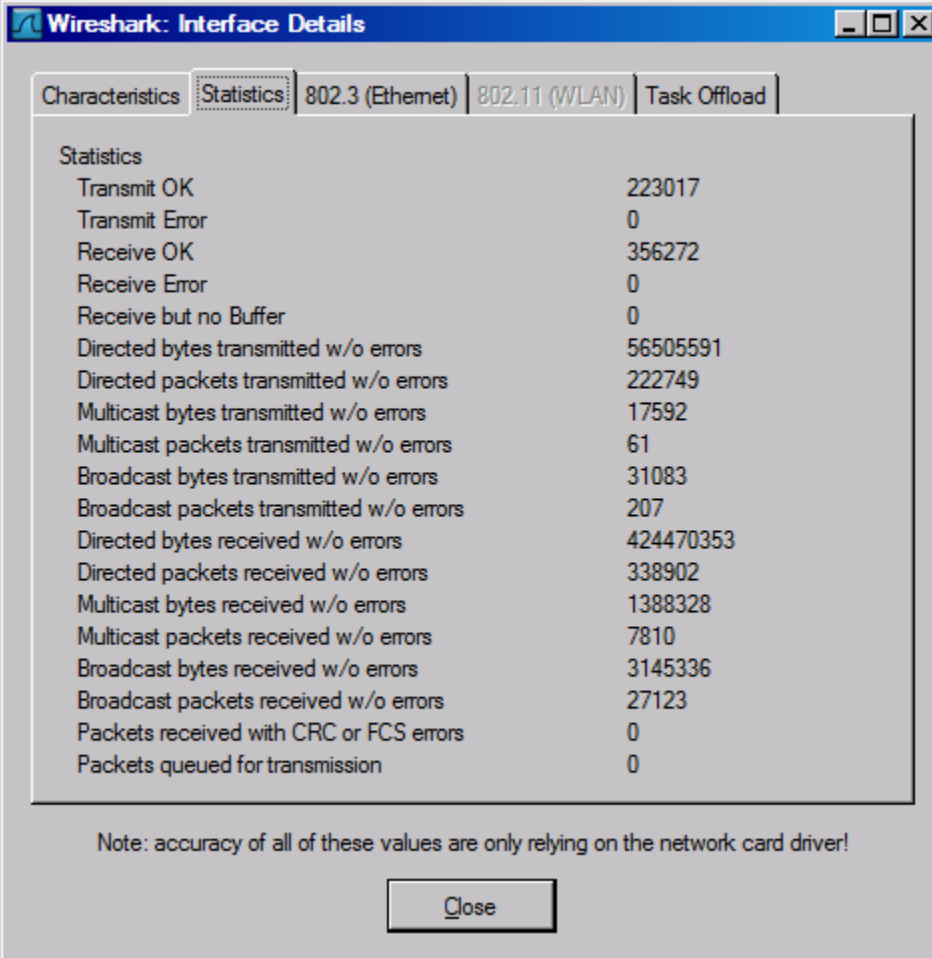
The image shows a screenshot of the 'Wireshark: Interface Details' window. The window has a blue title bar and a tabbed interface with 'Characteristics', 'Statistics', '802.3 (Ethernet)', '802.11 (WLAN)', and 'Task Offload' tabs. The 'Characteristics' tab is active, displaying a list of network interface properties. A red arrow points to the 'Receive Buffer Space' value of 387584. At the bottom of the window, there is a note and a 'Close' button.

Characteristics	
Vendor description	Intel(R) PRO/1000 PL Network Connection
Interface	\Device\NPF_{97708CAB-FF09-4180-9B1B-C3E61EDC7280}
Link status	Connected
Link speed	100 MBits/s
Media supported	802.3 (Ethernet)
Medium in use	802.3 (Ethernet)
Physical medium	Unspecified
NDIS Driver Version	4.0
Vendor Driver Version	9.11 (Hex: 9.B)
Vendor ID	00:15:58 (Foxconn) NIC: 00
MAC Options	802.1P Priority: Supported, 802.1Q VLAN: Supported
VLAN ID	-
Transmit Buffer Space	775168
Receive Buffer Space	387584
Transmit Block Size	1514
Receive Block Size	1514
Maximum Packet Size	1514

Note: accuracy of all of these values are only relying on the network card driver!

Close

# INTERFACE DETAILS: STATISTICS



Wireshark: Interface Details

Characteristics | **Statistics** | 802.3 (Ethernet) | 802.11 (WLAN) | Task Offload

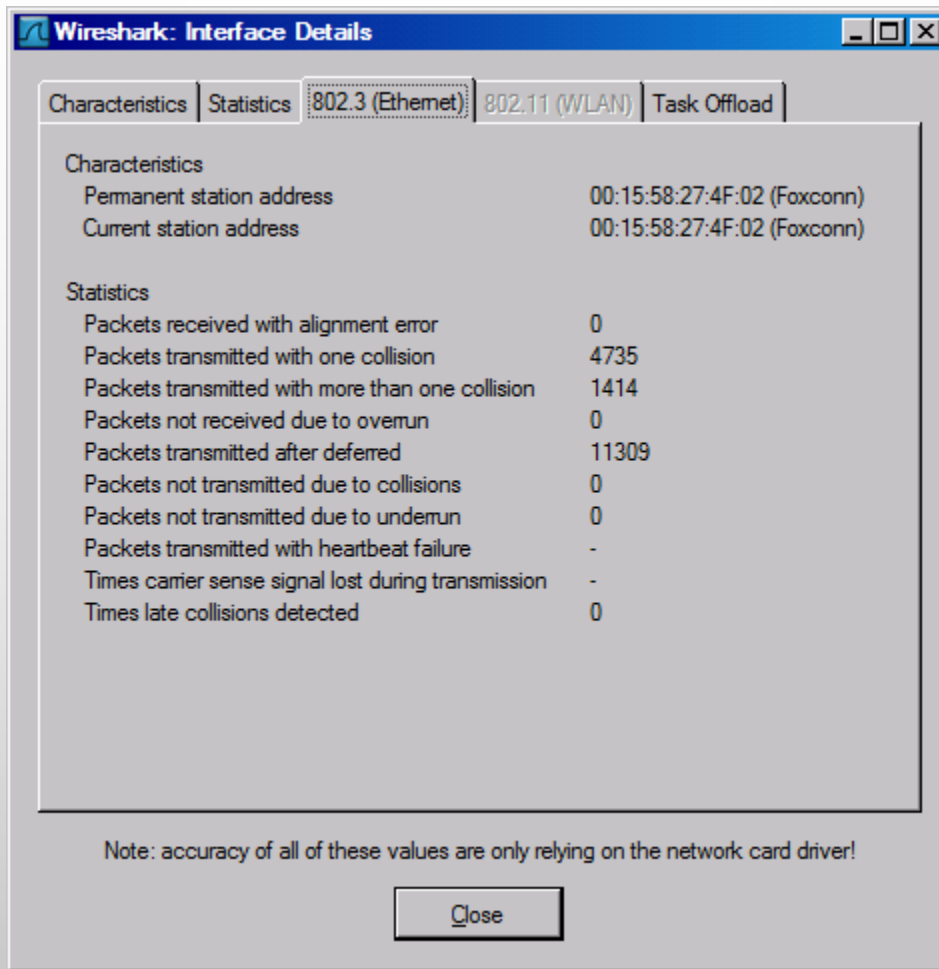
Statistics

Transmit OK	223017
Transmit Error	0
Receive OK	356272
Receive Error	0
Receive but no Buffer	0
Directed bytes transmitted w/o errors	56505591
Directed packets transmitted w/o errors	222749
Multicast bytes transmitted w/o errors	17592
Multicast packets transmitted w/o errors	61
Broadcast bytes transmitted w/o errors	31083
Broadcast packets transmitted w/o errors	207
Directed bytes received w/o errors	424470353
Directed packets received w/o errors	338902
Multicast bytes received w/o errors	1388328
Multicast packets received w/o errors	7810
Broadcast bytes received w/o errors	3145336
Broadcast packets received w/o errors	27123
Packets received with CRC or FCS errors	0
Packets queued for transmission	0

Note: accuracy of all of these values are only relying on the network card driver!

Close

# INTERFACE DETAILS: 802.3



The image shows a screenshot of the Wireshark 'Interface Details' pane. The '802.3 (Ethernet)' tab is selected, displaying two sections: 'Characteristics' and 'Statistics'. The 'Characteristics' section shows the permanent and current station addresses as 00:15:58:27:4F:02 (Foxconn). The 'Statistics' section lists various network metrics such as alignment errors, collisions, and carrier sense signal losses, with values ranging from 0 to 11309. A note at the bottom states that the accuracy of these values depends on the network card driver. A 'Close' button is located at the bottom center of the pane.

Characteristics	
Permanent station address	00:15:58:27:4F:02 (Foxconn)
Current station address	00:15:58:27:4F:02 (Foxconn)

Statistics	
Packets received with alignment error	0
Packets transmitted with one collision	4735
Packets transmitted with more than one collision	1414
Packets not received due to overrun	0
Packets transmitted after deferred	11309
Packets not transmitted due to collisions	0
Packets not transmitted due to underrun	0
Packets transmitted with heartbeat failure	-
Times carrier sense signal lost during transmission	-
Times late collisions detected	0

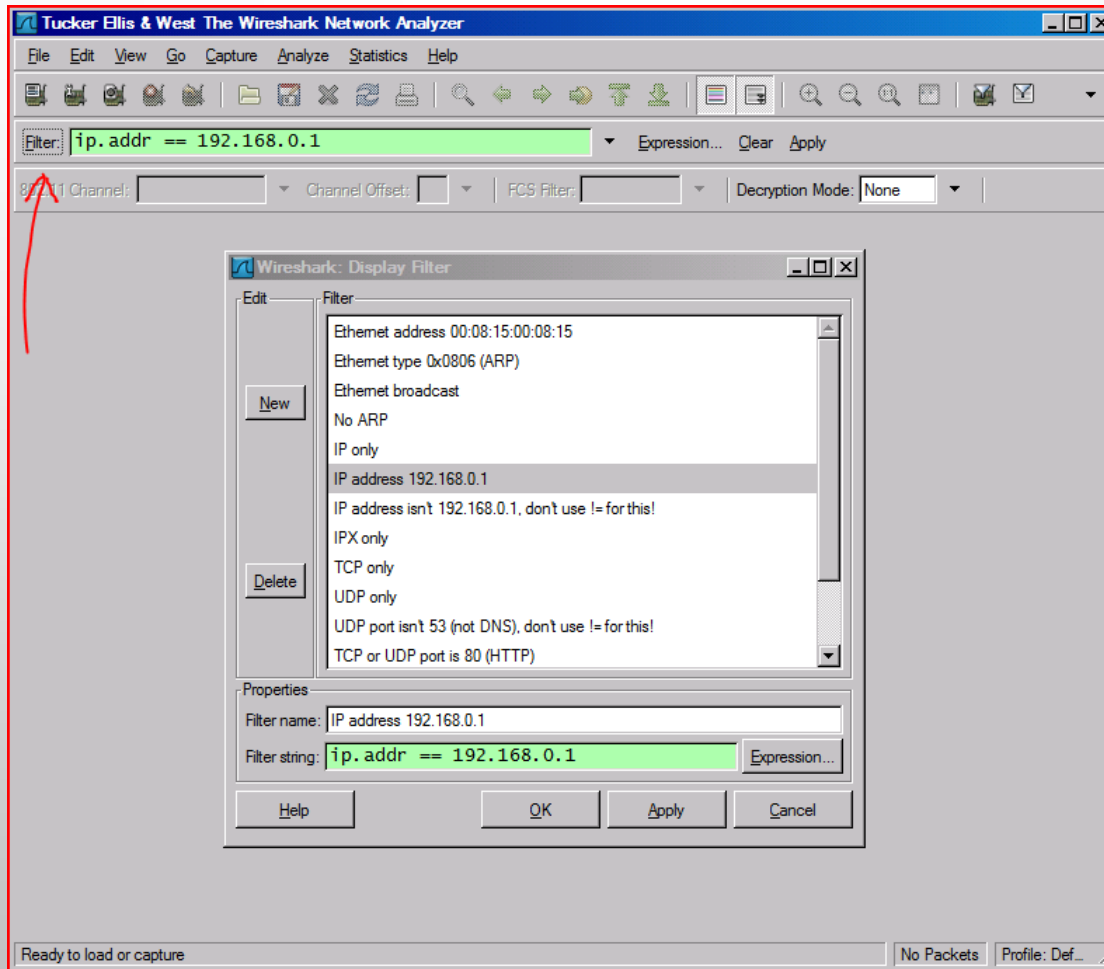
Note: accuracy of all of these values are only relying on the network card driver!

Close

# DISPLAY FILTERS (POST-FILTERS)

- DISPLAY FILTERS (ALSO CALLED POST-FILTERS) ONLY FILTER THE VIEW OF WHAT YOU ARE SEEING. ALL PACKETS IN THE CAPTURE STILL EXIST IN THE TRACE
- DISPLAY FILTERS USE THEIR OWN FORMAT AND ARE MUCH MORE POWERFUL THEN CAPTURE FILTERS

# DISPLAY FILTER





# DISPLAY FILTER EXAMPLES

**IP.SRC==10.1.11.00/24**

**IP.ADDR==192.168.1.10 && IP.ADDR==192.168.1.20**

**TCP.PORT==80 || TCP.PORT==3389**

**!(IP.ADDR==192.168.1.10 && IP.ADDR==192.168.1.20)**

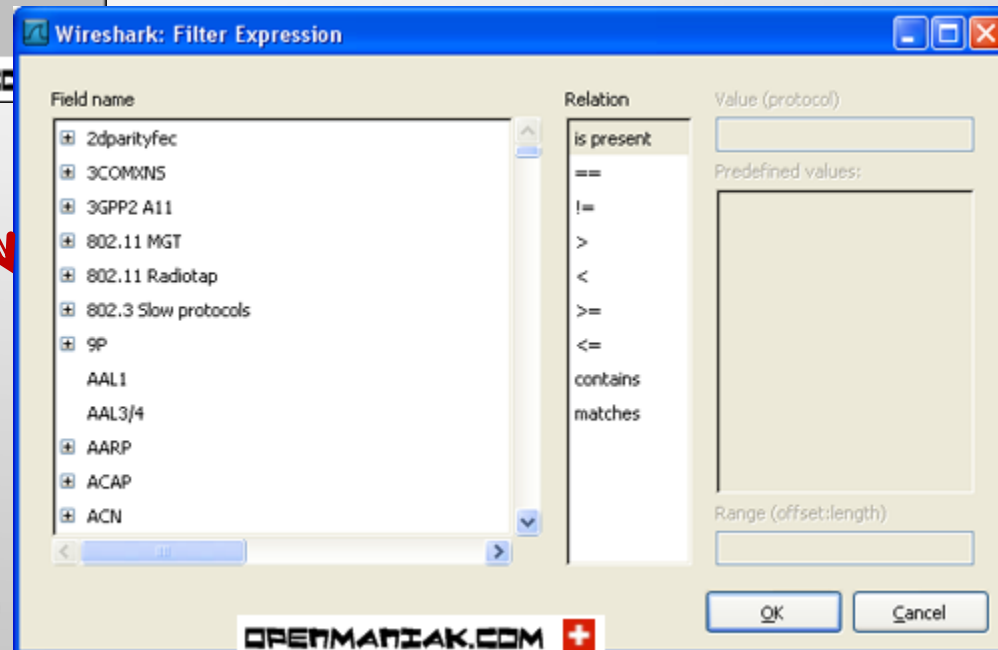
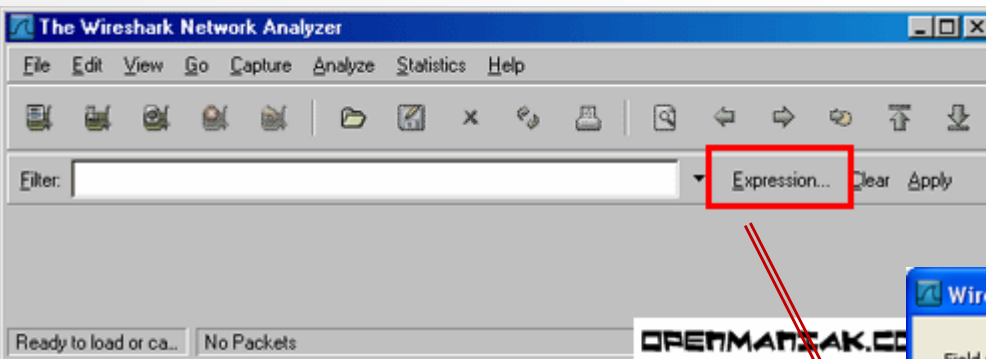
**(IP.ADDR==192.168.1.10 && IP.ADDR==192.168.1.20) && (TCP.PORT==445 ||  
TCP.PORT==139**

**(IP.ADDR==192.168.1.10 && IP.ADDR==192.168.1.20) && (UDP.PORT==67 ||  
UDP.PORT==68)**

**TCP.DSTPORT == 80**

# DICDI AV FILTERED

Syntax:	<b>Protocol</b>	<b>String 1</b>	<b>String 2</b>	<b>Comparison operator</b>	<b>Value</b>	<b>Logical Operations</b>	<b>Other expression</b>
Example:	ftp	passive	ip	==	10.2.3.4	xor	icmp.type

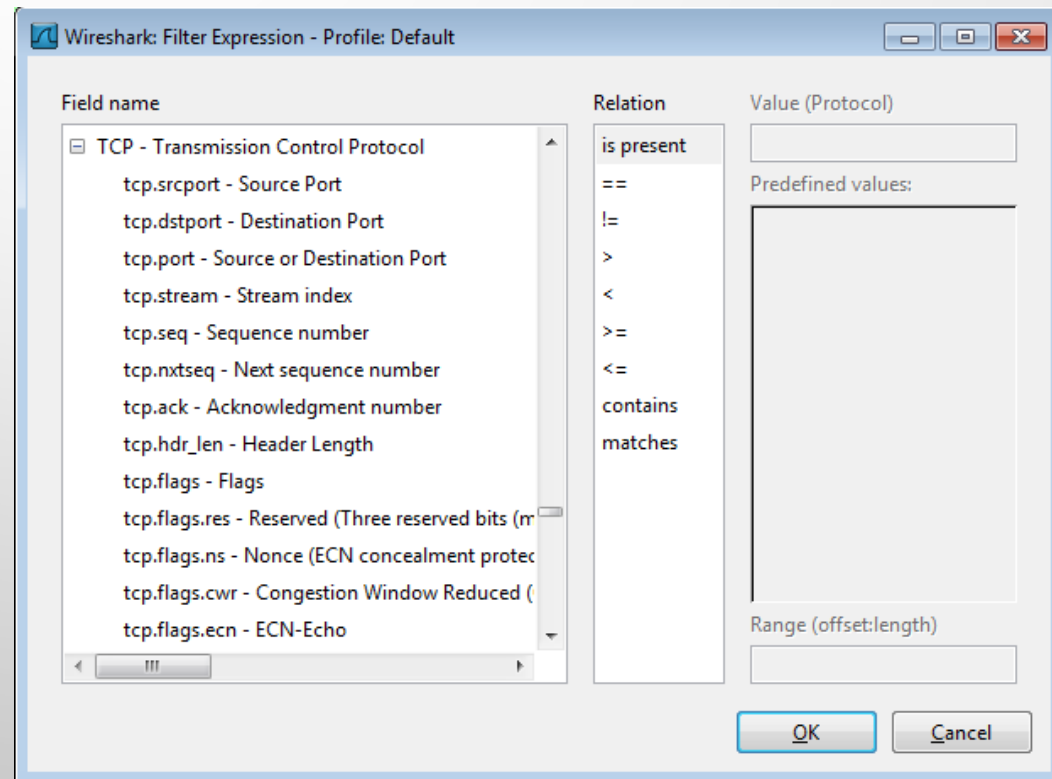


# DISPLAY FILTER

- STRING1, STRING2 (OPTIONAL SETTINGS):
  - SUB PROTOCOL CATEGORIES INSIDE THE PROTOCOL.
  - LOOK FOR A PROTOCOL AND THEN CLICK ON THE "+" CHARACTER.
  - EXAMPLE:
    - **TCP.SRCPORT == 80**
    - **TCP.FLAGS == 2**
      - SYN PACKET
      - TCP.FLAGS.SYN==1
    - **TCP.FLAGS == 18**
      - SYN/ACK

- **NOTE OF TCP FLAG FIELD:**

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N



# DISPLAY FILTER EXPRESSIONS

- SNMP || DNS || ICMP
  - DISPLAY THE SNMP OR DNS OR ICMP TRAFFICS.
- TCP.PORT == 25
  - DISPLAY PACKETS WITH TCP SOURCE OR DESTINATION PORT 25.
- TCP.FLAGS
  - DISPLAY PACKETS HAVING A TCP FLAGS
- TCP.FLAGS.SYN == 0X02
  - DISPLAY PACKETS WITH A TCP SYN FLAG.

Six comparison operators are available:

English format:	C like format:	Meaning:
eq	==	Equal
ne	!=	Not equal
gt	>	Greater than
lt	<	Less than
ge	>=	Greater or equal
le	<=	Less or equal

→ **Logical expressions:**

English format:	C like format:	Meaning:
and	&&	Logical AND
or		Logical OR
xor	^^	Logical XOR
not	!	Logical NOT

If the filter syntax is correct, it will be highlighted in green, otherwise if there is a syntax mistake it will be highlighted in red.

Filter: `tcp.port == 100`

Filter: `tcp.port = 100`

Correct syntax

Wrong syntax

# SAVE FILTERED PACKETS AFTER USING DISPLAY

## FILTER

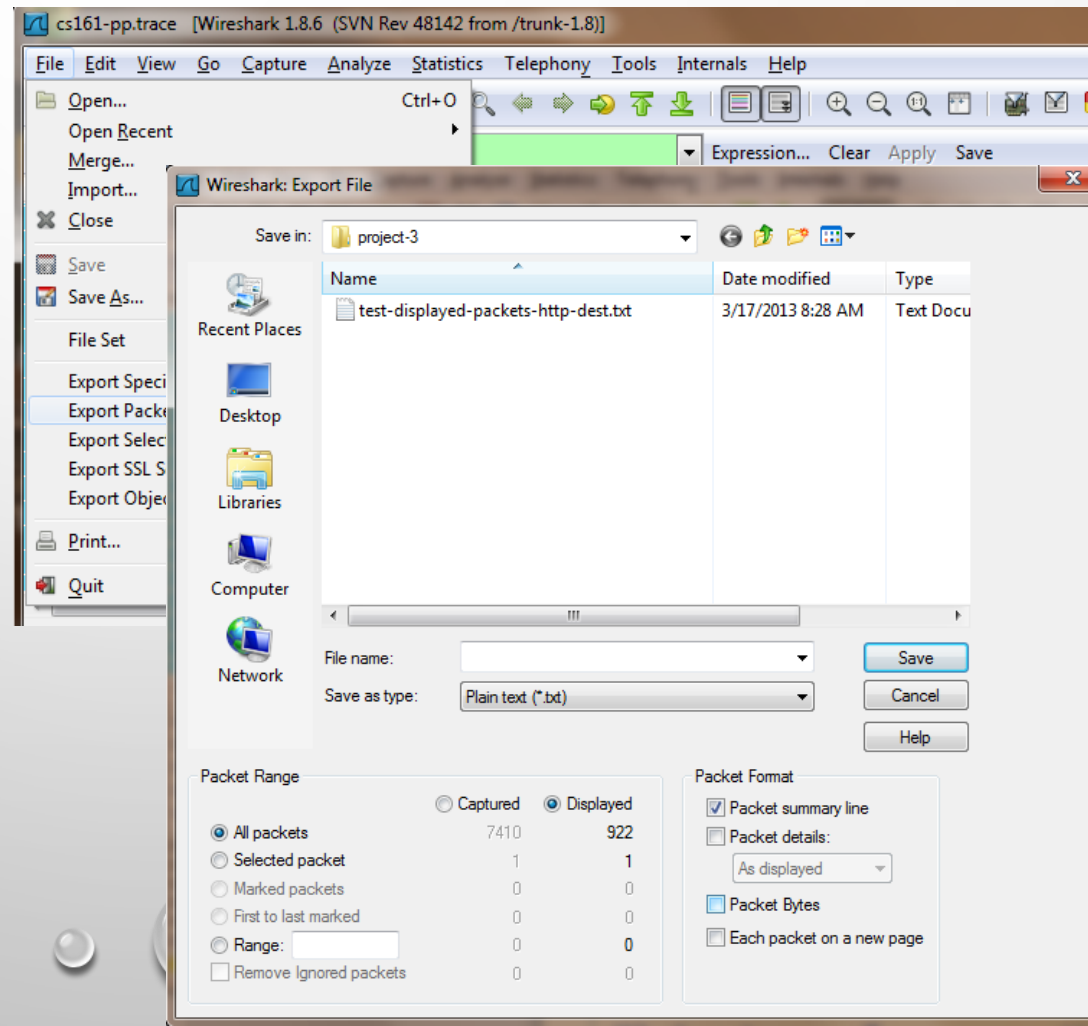
- WE CAN ALSO SAVE ALL FILTERED PACKETS IN TEXT FILE FOR FURTHER ANALYSIS

- OPERATION:

File → Export packet dissections  
→ as “plain text” file

1). In “packet range” option, select  
“Displayed”

2). In choose “summary line” or  
“detail”



Tucker Ellis & West Obsolete\_Packets.cap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Summary  
**Protocol Hierarchy**  
 Conversations  
 Endpoints  
 IO Graphs

Filter:

802.11 Channel:  Chan

No.	Time	Source
1	0.000000	::
2	0.000010	::
3	2.179063	192.168.1.1
4	2.439522	192.168.1.1
5	2.715733	192.168.1.1
6	2.821401	192.168.1.1
7	2.821546	192.168.1.1
8	2.824683	192.168.1.1
9	2.990859	192.168.1.1
10	3.266913	192.168.1.1
11	3.495707	fe80::20c
12	3.495727	fe80::20c
13	3.542893	192.168.1.1
14	3.543088	192.168.1.1

ANSI  
 Fax T38 Analysis...  
 GSM  
 H.225...  
 MTP3  
 RTP  
 SCTP  
 SIP...  
 VoIP Calls  
 WAP-WSP...

BOOTP-DHCP...  
 Destinations...  
 Flow Graph...  
 HTTP  
 IP address...  
 ISUP Messages...  
 Multicast Streams  
 ONC-RPC Programs  
 Packet Length...  
 Port Type...  
 SMPP Operations...  
 TCP Stream Graph  
 WLAN Traffic...

Expression... Clear Apply

Decryption Mode: None

No.	Time	Protocol	Info
1	0d:56e3	ICMPv6	Multicast listener rep
2	0d:56e3	ICMPv6	Multicast listener rep
3	255	NBNS	Name query NB LOCALHOST
4	255	NBNS	Name query NB LOCALHOST
5	255	NBNS	Name query NB LOCALHOST
6	254	DNS	Standard query PTR 66.1
7	254	DNS	Standard query PTR 255
8	66	DNS	Standard query response
9	03.255	NBNS	Name query NB LOCALHOST
10	03.255	NBNS	Name query NB LOCALHOST
11		ICMPv6	Router solicitation
12		ICMPv6	Router solicitation
13	254	DNS	Standard query A DoCoMo
14	03.255	NBNS	Name query NB LOCALHOST

Frame 1 (88 bytes on wire, Linux cooked capture)  
 Internet Protocol Version 4  
 Internet Control Message Protocol

```

0000  00 04 00 01 00 06 00 0c 3a 00 05 02 00 00 01 00
0010  60 00 00 00 00 20 00 01 00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030  00 00 00 01 ff 0d 56 e3 00 00 05 02 00 00 01 00
0040  83 00 d2 c2 00 00 00 00 ff 02 00 00 00 00 00 00
0050  00 00 01 ff 0d 56 e3
  
```

File: "C:\Users\vo2.TEW\Downloads\Obsolete\_Packets.cap" Packets: 10949 Displayed: 10949 Marked: 0 Profile: Default

Wireshark: Protocol Hierarchy Statistics

Display filter: none

Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
[-] Frame	100.00%	10949	1433310	0.004	0	0	0.000
[-] Linux cooked-mode capture	100.00%	10949	1433310	0.004	0	0	0.000
[-] Internet Protocol Version 6	0.16%	18	1392	0.000	0	0	0.000
Internet Control Message Protocol v6	0.16%	18	1392	0.000	18	1392	0.000
[-] Internet Protocol	82.62%	9046	1312691	0.004	0	0	0.000
[+] User Datagram Protocol	17.33%	1898	262866	0.001	0	0	0.000
[+] Transmission Control Protocol	64.69%	7083	1046121	0.003	2350	163598	0.000
Internet Group Management Protocol	0.57%	62	3440	0.000	62	3440	0.000
Internet Control Message Protocol	0.03%	3	264	0.000	3	264	0.000
DEC DNA Routing Protocol	2.60%	285	14820	0.000	285	14820	0.000
Address Resolution Protocol	7.63%	835	46928	0.000	835	46928	0.000
MS Network Load Balancing	1.26%	138	8280	0.000	138	8280	0.000
Data	2.75%	301	25143	0.000	301	25143	0.000
[-] Logical-Link Control	2.23%	244	20024	0.000	0	0	0.000
Appletalk Address Resolution Protocol	0.37%	40	2480	0.000	40	2480	0.000
[+] Internetwork Packet eXchange	1.46%	160	14328	0.000	0	0	0.000
[+] Datagram Delivery Protocol	0.40%	44	3216	0.000	0	0	0.000
[+] Internetwork Packet eXchange	0.27%	30	1680	0.000	0	0	0.000
[+] Banyan Vines IP	0.47%	52	2352	0.000	0	0	0.000

Help Close

# FOLLOW TCP STREAM

The image shows the Wireshark network protocol analyzer interface. The title bar reads "Tucker Ellis & West http-ethereal-trace-1 - Wireshark". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. The toolbar contains various icons for file operations, capture, and analysis. The filter bar shows the filter: "(ip.addr eq 192.168.1.102 and ip.addr eq 128.119.245.12)". Below the filter bar, the interface is divided into three main sections: Packet List, Packet Details, and Packet Bytes.

**Packet List:**

No.	Time	Source	Destination	Protocol	Info
7	4.675312	192.168.1.102	128.119.245.12	TCP	unikeypro > http [SYN]
8	4.694429	128.119.245.12	192.168.1.102	TCP	> unikeypro [SYN, ACK]
9	4.694458	192.168.1.102	128.119.245.12	TCP	ypro > http [ACK]
10	4.694850	192.168.1.102	128.119.245.12	TCP	ethereal-labs/lab
11	4.717289	128.119.245.12	192.168.1.102	TCP	> unikeypro [ACK]
12	4.718993	128.119.245.12	192.168.1.102	TCP	1.1 200 OK (text/css)
13	4.724332	192.168.1.102	128.119.245.12	TCP	favicon.ico HTTP/1.1
14	4.750366	128.119.245.12	192.168.1.102	TCP	1.1 404 Not Found
15	4.859777	192.168.1.102	128.119.245.12	TCP	ypro > http [ACK]

**Packet Details:**

- Frame 8 (62 bytes on wire, 62 bytes captured)
- Ethernet II, Src: LinksysG\_da:af:73 (00:06:25:da:af:73), Dst: 192.168.1.102 (08:00:27:00:00:02)
- Internet Protocol, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.102 (192.168.1.102)
- Transmission Control Protocol, Src Port: http (80), Dst Port: http (80), Seq: 0, Ack: 1, Win: 0, Len: 0

**Packet Bytes:**

```
0000 00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00  ..tO6#..%.s..E.  
0010 00 30 00 00 40 00 37 06 0c 36 80 77 f5 0c c0 a8  .O..@.7..6.w....  
0020 01 66 00 50 10 1f 6b a6 54 91 f5 32 64 b2 70 12  .f.P..k. T..2d.p.  
0030 16 d0 0a 21 00 00 02 04 05 b4 01 01 04 02  ....!.....
```

The context menu is open over packet 8, with "Follow TCP Stream" selected. Other options include Mark Packet (toggle), Set Time Reference (toggle), Apply as Filter, Prepare a Filter, Conversation Filter, Colorize Conversation, SCTP, Follow UDP Stream, Follow SSL Stream, Copy, Export Selected Packet Bytes..., Decode As..., Print..., and Show Packet in New Window.

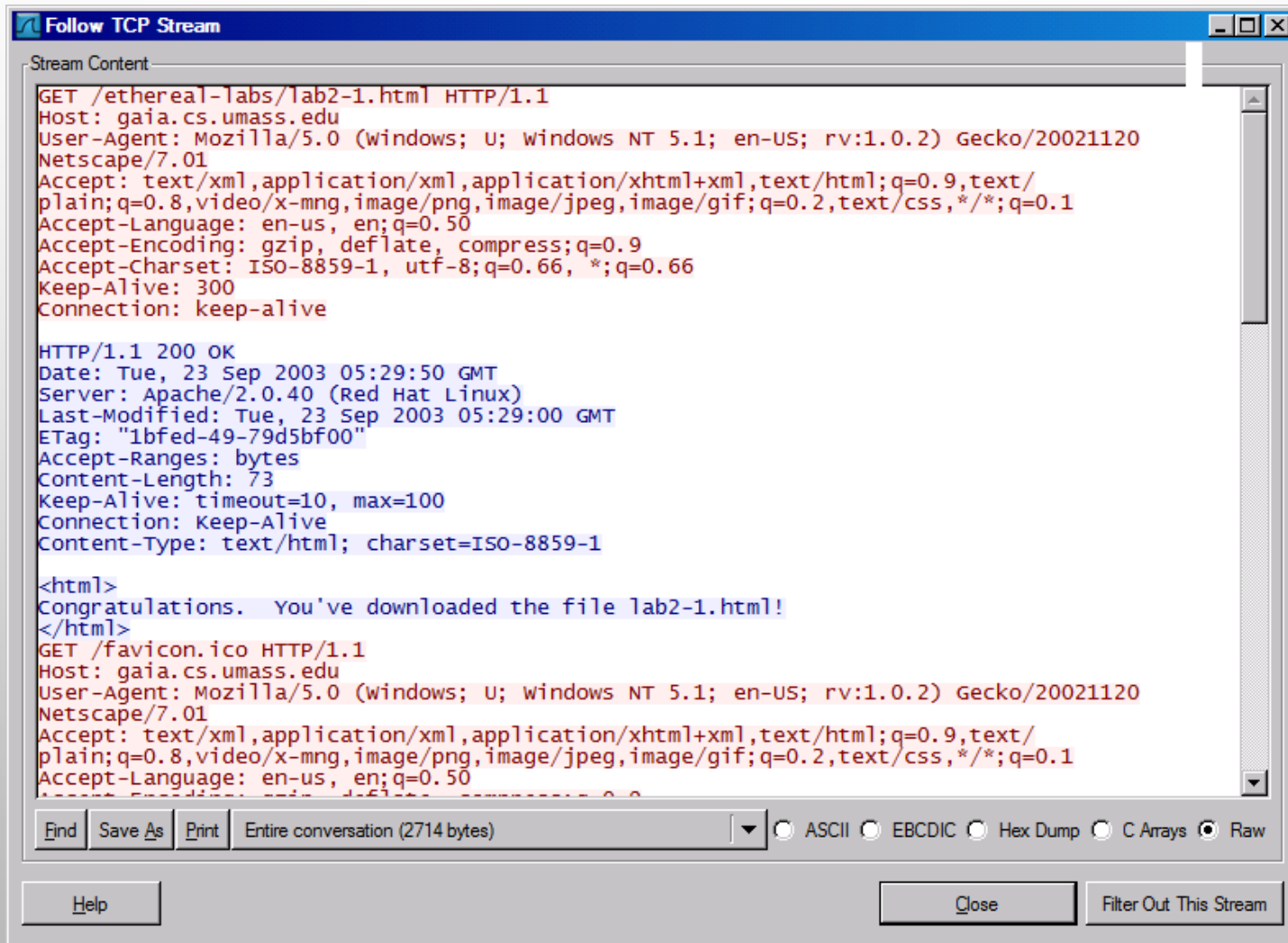
File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06      Packets: 1...      Profile: Def...



# FOLLOW TCP STREAM

red - stuff you sent

blue - stuff you get



```
Follow TCP Stream
Stream Content
GET /ethereal-labs/lab2-1.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (windows; U; windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120
Netscape/7.01
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1
Accept-Language: en-us, en;q=0.50
Accept-Encoding: gzip, deflate, compress;q=0.9
Accept-Charset: ISO-8859-1, utf-8;q=0.66, *,q=0.66
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Date: Tue, 23 Sep 2003 05:29:50 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT
ETag: "1bfed-49-79d5bf00"
Accept-Ranges: bytes
Content-Length: 73
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<html>
Congratulations. You've downloaded the file lab2-1.html!
</html>
GET /favicon.ico HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (windows; U; windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120
Netscape/7.01
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1
Accept-Language: en-us, en;q=0.50
Accept-Encoding: gzip, deflate, compress;q=0.9
Accept-Charset: ISO-8859-1, utf-8;q=0.66, *,q=0.66
Keep-Alive: 300
Connection: keep-alive
```

Find Save As Print Entire conversation (2714 bytes) [Format: ASCII EBCDIC Hex Dump C Arrays Raw]

Help Close Filter Out This Stream

# FILTER OUT/IN SINGLE TCP STREAM

- WHEN CLICK “FILTER OUT THIS TCP STREAM” IN PREVIOUS PAGE’S BOX, NEW FILTER STRING WILL CONTAIN LIKE:
  - HTTP AND !(TCP.STREAM EQ 5)
- SO, IF YOU USE “TCP.STREAM EQ 5” AS FILTER STRING, YOU KEEP THIS HTTP SESSION

The image shows a screenshot of the Wireshark network traffic analysis tool. The main window displays a list of captured packets, with a filter applied: `tcp.stream eq 5`. The packets are listed in a table with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 78 is selected, and its details are shown in the lower pane. The details pane shows the following information:

- Frame 78: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
- Null/Loopback
- Internet Protocol Version 4, Src: 172.19.99.10 (172.19.99.10), Dst: 170.31.228.231 (170.31.228.231)
- Transmission Control Protocol, Src Port: 58286 (58286), Dst Port: http (80), Seq: 0, Len: 0

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII:

```
0000 02 00 00 00 00 45 00 00 3c 00 00 40 00 40 06 9c 97  ....E..<..@..@...
0010 ac 13 63 0a aa 1f e4 e7 e3 ae 00 50 4f a1 59 e1  ....C.....PO.Y.
0020 00 00 00 00 0a 00 16 d0 05 8e 00 00 02 04 05 b4  ....
0030 04 02 08 0a 00 00 00 02 00 00 00 01 01 03 03 00  ....
```

# EXPERT INFO

The image shows the Wireshark interface with the Expert Info window open. The main packet list shows 14 packets. Packet 8 is selected, and its details are shown in the Expert Info window. The details include Ethernet II, Internet Protocol, and Transmission Control Protocol. The hex dump at the bottom shows the raw bytes of the selected packet.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.102	192.168.1.104	SNMP	get-request SNMPv2-SMI
2	0.017162	192.168.1.104	192.168.1.102	SNMP	get-response SNMPv2-SMI
3	3.017086	192.168.1.102	192.168.1.104	SNMP	get-request SNMPv2-SMI
4	3.034572	192.168.1.104	192.168.1.102	SNMP	get-response SNMPv2-SMI
5	4.626878	192.168.1.102	192.168.1.102	DNS	Standard query A qai...
6	4.663785	192.168.1.102	192.168.1.102	DNS	Standard query response
7	4.675312	192.168.1.102	192.168.1.102	TCP	unikeypro > http [SYN]
8	4.694429	192.168.1.102	192.168.1.102	TCP	http > unikeypro [SYN]
9	4.694458	192.168.1.102	192.168.1.102	TCP	unikeypro > http [ACK]
10	4.694850	192.168.1.102	192.168.1.102	HTTP	GET /ethereal-labs/lab...
11	4.717289	192.168.1.102	192.168.1.102	TCP	http > unikeypro [ACK]
12	4.718993	128.119.245.12	192.168.1.102	HTTP	HTTP/1.1 200 OK (text/...
13	4.724332	192.168.1.102	128.119.245.12	HTTP	GET /favicon.ico HTTP/...
14	4.750366	128.119.245.12	192.168.1.102	HTTP	HTTP/1.1 404 Not Found

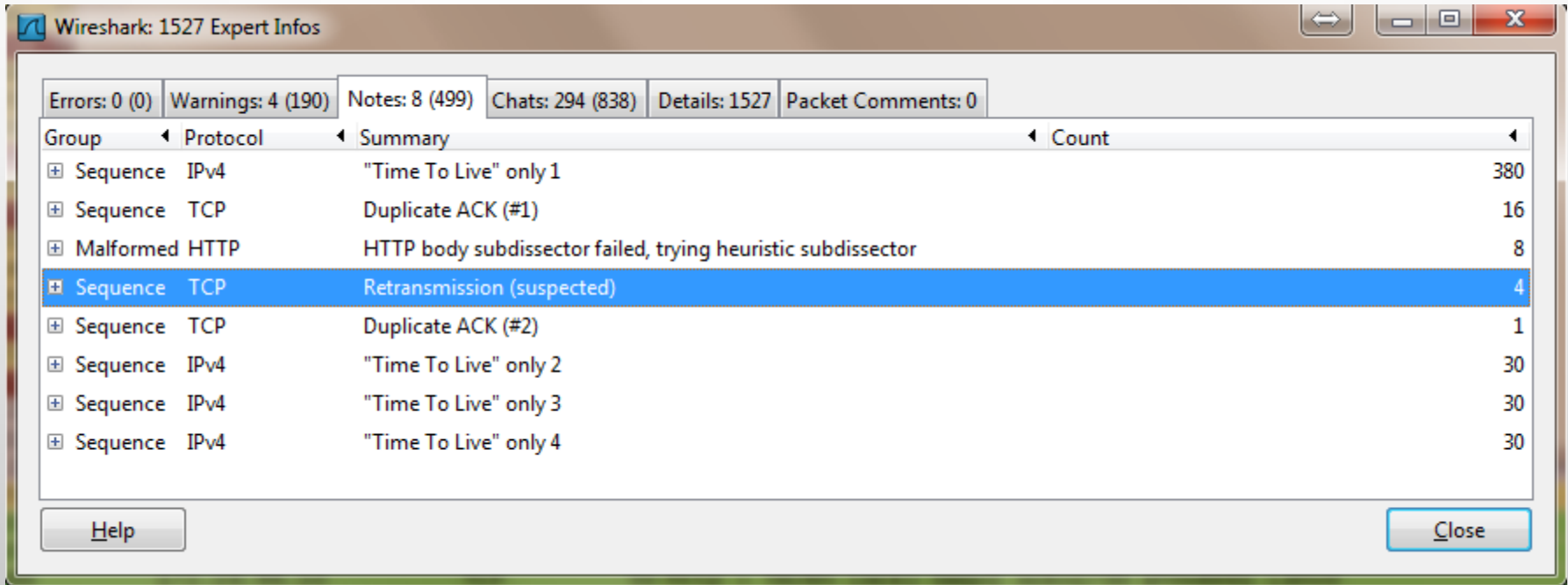
Frame 8 (62 bytes on wire, 62 bytes captured)

- Ethernet II, Src: LinksysG\_da:af:73 (00:06:25:da:af:73), Dst: DellComp\_4f:36:23 (00:08:74:4f:36:23)
- Internet Protocol, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.102 (192.168.1.102)
- Transmission Control Protocol, Src Port: http (80), Dst Port: unikeypro (4127), Seq: 0, Ack: 1,

```
0000  00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00  ..t06#..%.s..E.
0010  00 30 00 00 40 00 37 06 0c 36 80 77 f5 0c c0 a8  .0..@.7. .6.w....
0020  01 66 00 50 10 1f 6b a6 54 91 f5 32 64 b2 70 12  .f.P..k. T..2d.p.
0030  16 d0 0a 21 00 00 02 04 05 b4 01 01 04 02      ....!.....
```

File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06 Packets: 1 Profile: Def...

# EXPERT INFO



Wireshark: 1527 Expert Infos

Errors: 0 (0) Warnings: 4 (190) Notes: 8 (499) Chats: 294 (838) Details: 1527 Packet Comments: 0

Group	Protocol	Summary	Count
Sequence	IPv4	"Time To Live" only 1	380
Sequence	TCP	Duplicate ACK (#1)	16
Malformed	HTTP	HTTP body subdissector failed, trying heuristic subdissector	8
Sequence	TCP	Retransmission (suspected)	4
Sequence	TCP	Duplicate ACK (#2)	1
Sequence	IPv4	"Time To Live" only 2	30
Sequence	IPv4	"Time To Live" only 3	30
Sequence	IPv4	"Time To Live" only 4	30

Help Close

# CONVERSATIONS

The screenshot shows the Wireshark interface for a capture named "Tucker Ellis & West http-ethereal-trace-1". The "Conversations" pane is active, displaying a list of network sessions. The main packet list shows frames 1 through 14, with frame 2 selected. The packet details pane shows the structure of frame 2, including Ethernet II, Internet Protocol, User Datagram Protocol, and Simple Network Management Protocol (SNMP). The packet bytes pane shows the raw hex and ASCII data for the selected frame.

No.	Time	Source
1	0.000000	192.168.1.104
2	0.017162	192.168.1.102
3	3.017086	192.168.1.104
4	3.034572	192.168.1.102
5	4.626878	192.168.1.104
6	4.663785	63.240.76.19
7	4.675312	192.168.1.102
8	4.694429	128.119.2.102
9	4.694458	192.168.1.102
10	4.694850	192.168.1.102
11	4.717289	128.119.2.102
12	4.718993	128.119.2.102
13	4.724332	192.168.1.102
14	4.750366	128.119.2.102

No.	Protocol	Info
104	SNMP	get-request SNMPV2-SMI
102	SNMP	get-response SNMPV2-SMI
104	SNMP	get-request SNMPV2-SMI
102	SNMP	get-response SNMPV2-SMI
19	DNS	Standard query A gaia.d
102	DNS	Standard query response
15.12	TCP	unikeypro > http [SYN]
102	TCP	http > unikeypro [SYN]
15.12	TCP	unikeypro > http [ACK]
15.12	HTTP	GET /ethereal-labs/lab
102	TCP	http > unikeypro [ACK]
102	HTTP	HTTP/1.1 200 OK (text
15.12	HTTP	GET /favicon.ico HTTP/1
102	HTTP	HTTP/1.1 404 Not Found

Frame 2 (93 bytes on wire, Ethernet II, Src: Hewlett- Internet Protocol, Src: 192.168.1.102, Dst: 192.168.1.104), Dst: DellComp\_4f:36:23 (00:08:74:4f:36:23), Src: opsview-envoy (4125)

```
0000 00 08 74 4f 36 23 00 3c 00 00 00 00 00 00 00 00
0010 00 4f ec d8 00 00 3c 11 00 00 00 00 00 00 00 00
0020 01 66 00 a1 10 1d 00 3b 00 00 00 00 00 00 00 00
0030 06 70 75 62 6c 69 63 a2 24 02 02 18 31 02 01 00
0040 02 01 00 30 18 30 16 06 11 2b 06 01 04 01 0b 02
0050 02 00 04 02 01 02 02 02 01 00 04 01 10
```

# CONVERSATIONS

Conversations: cs161-pp.trace

Ethernet Fibre Channel FDDI IPv4: 173 IPv6: 1 IPX JXTA NCP RSVP SCTP **TCP: 155** Token Ring UDP: 2398 USB WLAN

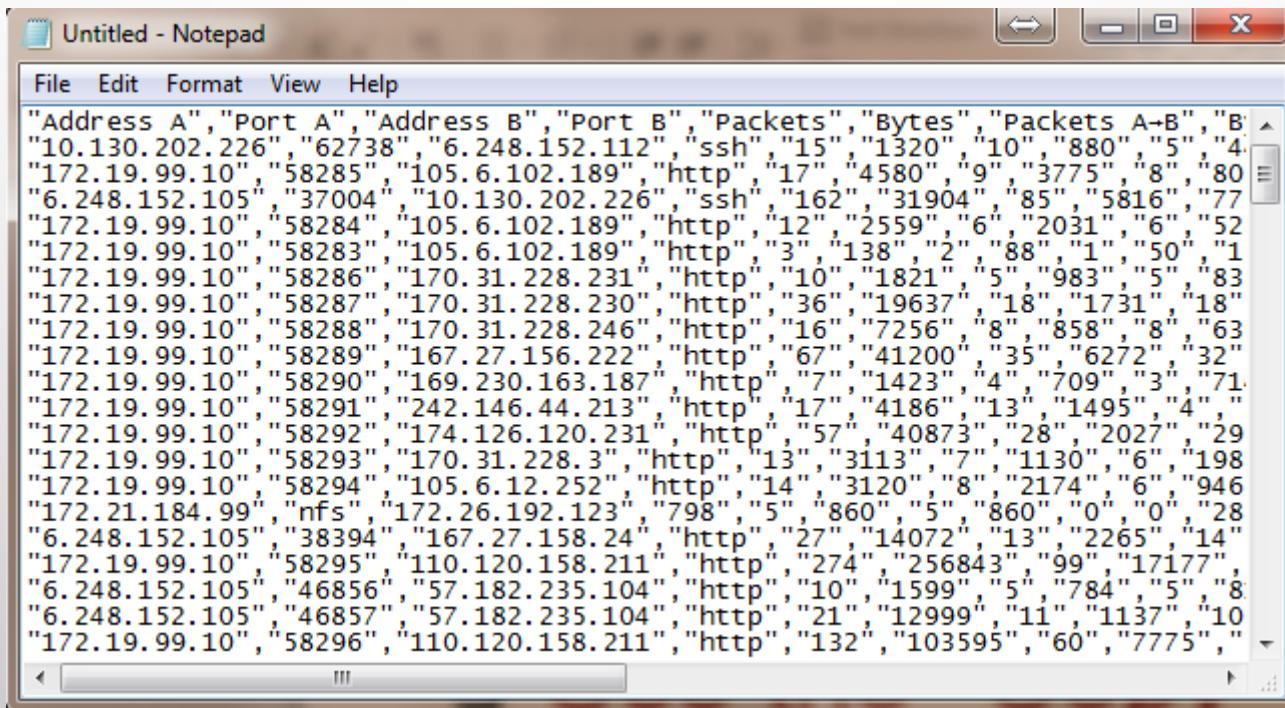
TCP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes
10.130.202.226	62738	6.248.152.112	ssh	15	1 320	10	880	5	
172.19.99.10	58285	105.6.102.189	http	17	4 580	9	3 775	8	
6.248.152.105	37004	10.130.202.226	ssh	162	31 904	85	5 816	77	
172.19.99.10	58284	105.6.102.189	http	12	2 559	6	2 031	6	
172.19.99.10	58283	105.6.102.189	http	3	138	2	88	1	
172.19.99.10	58286	170.31.228.231	http	10	1 821	5	983	5	
172.19.99.10	58287	170.31.228.230	http	36	19 637	18	1 731	18	
172.19.99.10	58288	170.31.228.246	http	16	7 256	8	858	8	
172.19.99.10	58289	167.27.156.222	http	67	41 200	35	6 272	32	
172.19.99.10	58290	169.230.163.187	http	7	1 423	4	709	3	
172.19.99.10	58291	242.146.44.213	http	17	4 186	13	1 495	4	
172.19.99.10	58292	174.126.120.231	http	57	40 873	28	2 027	29	

Name resolution  Limit to display filter

Help Copy Follow Stream Close

- USE THE “COPY” BUTTON TO COPY ALL TEXT INTO CLIPBOARD

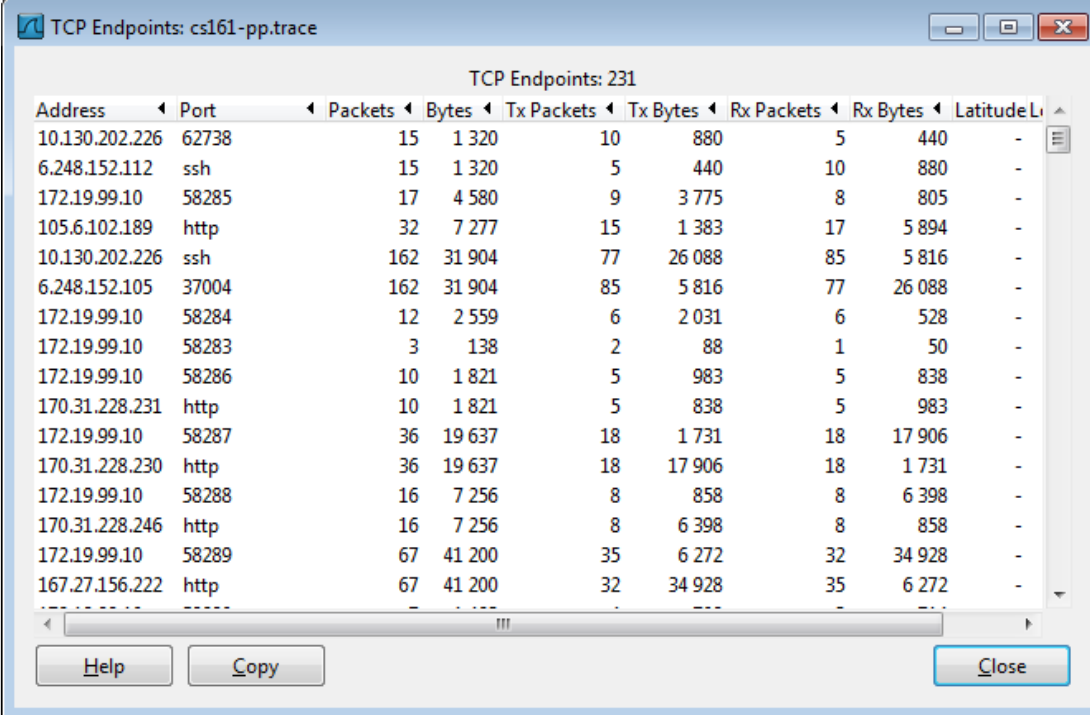


```
File Edit Format View Help
"Address A","Port A","Address B","Port B","Packets","Bytes","Packets A-B","B
"10.130.202.226","62738","6.248.152.112","ssh","15","1320","10","880","5","4
"172.19.99.10","58285","105.6.102.189","http","17","4580","9","3775","8","80
"6.248.152.105","37004","10.130.202.226","ssh","162","31904","85","5816","77
"172.19.99.10","58284","105.6.102.189","http","12","2559","6","2031","6","52
"172.19.99.10","58283","105.6.102.189","http","3","138","2","88","1","50","1
"172.19.99.10","58286","170.31.228.231","http","10","1821","5","983","5","83
"172.19.99.10","58287","170.31.228.230","http","36","19637","18","1731","18"
"172.19.99.10","58288","170.31.228.246","http","16","7256","8","858","8","63
"172.19.99.10","58289","167.27.156.222","http","67","41200","35","6272","32"
"172.19.99.10","58290","169.230.163.187","http","7","1423","4","709","3","71
"172.19.99.10","58291","242.146.44.213","http","17","4186","13","1495","4","8
"172.19.99.10","58292","174.126.120.231","http","57","40873","28","2027","29
"172.19.99.10","58293","170.31.228.3","http","13","3113","7","1130","6","198
"172.19.99.10","58294","105.6.12.252","http","14","3120","8","2174","6","946
"172.21.184.99","nfs","172.26.192.123","798","5","860","5","860","0","0","28
"6.248.152.105","38394","167.27.158.24","http","27","14072","13","2265","14"
"172.19.99.10","58295","110.120.158.211","http","274","256843","99","17177"
"6.248.152.105","46856","57.182.235.104","http","10","1599","5","784","5","8
"6.248.152.105","46857","57.182.235.104","http","21","12999","11","1137","10
"172.19.99.10","58296","110.120.158.211","http","132","103595","60","7775",""
```

- THEN, YOU CAN ANALYZE THIS TEXT FILE TO GET WHAT STATISTICS YOU WANT

# FIND ENDPOINT STATISTICS

- MENU “STATISTICS” → “ENDPOINT LIST” → “TCP”



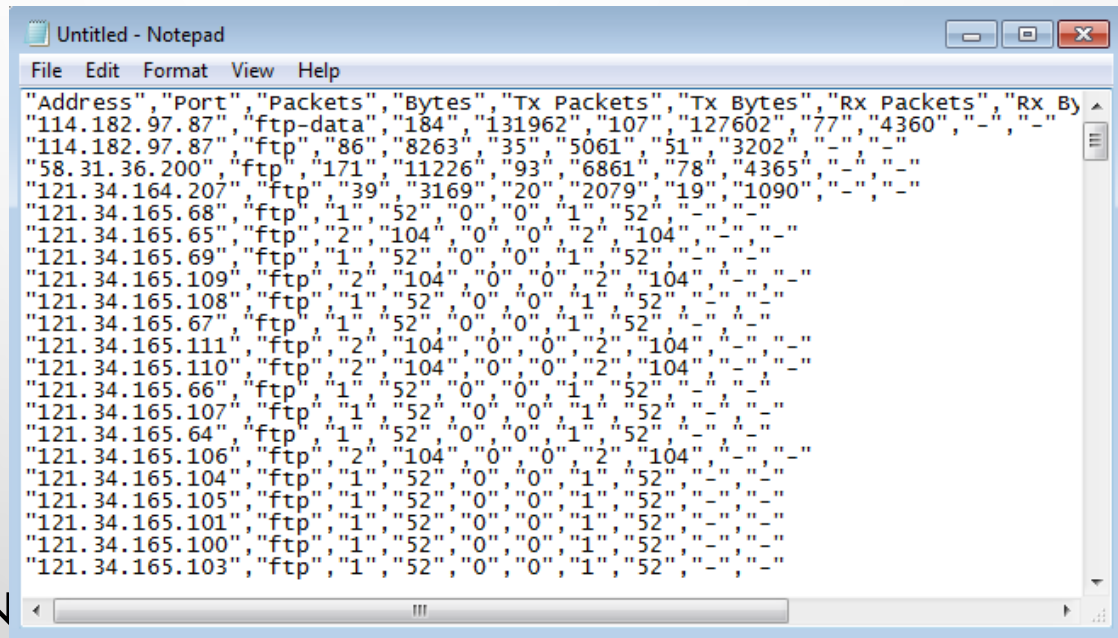
Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude L
10.130.202.226	62738	15	1 320	10	880	5	440	-
6.248.152.112	ssh	15	1 320	5	440	10	880	-
172.19.99.10	58285	17	4 580	9	3 775	8	805	-
105.6.102.189	http	32	7 277	15	1 383	17	5 894	-
10.130.202.226	ssh	162	31 904	77	26 088	85	5 816	-
6.248.152.105	37004	162	31 904	85	5 816	77	26 088	-
172.19.99.10	58284	12	2 559	6	2 031	6	528	-
172.19.99.10	58283	3	138	2	88	1	50	-
172.19.99.10	58286	10	1 821	5	983	5	838	-
170.31.228.231	http	10	1 821	5	838	5	983	-
172.19.99.10	58287	36	19 637	18	1 731	18	17 906	-
170.31.228.230	http	36	19 637	18	17 906	18	1 731	-
172.19.99.10	58288	16	7 256	8	858	8	6 398	-
170.31.228.246	http	16	7 256	8	6 398	8	858	-
172.19.99.10	58289	67	41 200	35	6 272	32	34 928	-
167.27.156.222	http	67	41 200	32	34 928	35	6 272	-

- YOU CAN SORT BY FIELD
- “TX” : TRANSMIT      “RX” : RECEIVE



# FIND ENDPOINT STATISTICS

- USE THE “COPY” BUTTON TO COPY ALL TEXT INTO CLIPBOARD



The screenshot shows a Notepad window titled "Untitled - Notepad" with a menu bar (File, Edit, Format, View, Help). The text content is a list of network statistics for various IP addresses and ports. The data is presented as a table with columns for Address, Port, Packets, Bytes, Tx Packets, Tx Bytes, Rx Packets, and Rx Bytes. The data is as follows:

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
114.182.97.87	ftp-data	184	131962	107	127602	77	4360
114.182.97.87	ftp	86	8263	35	5061	51	3202
58.31.36.200	ftp	171	11226	93	6861	78	4365
121.34.164.207	ftp	39	3169	20	2079	19	1090
121.34.165.68	ftp	1	52	0	0	1	52
121.34.165.65	ftp	2	104	0	0	2	104
121.34.165.69	ftp	1	52	0	0	1	52
121.34.165.109	ftp	2	104	0	0	2	104
121.34.165.108	ftp	1	52	0	0	1	52
121.34.165.67	ftp	1	52	0	0	1	52
121.34.165.111	ftp	2	104	0	0	2	104
121.34.165.110	ftp	2	104	0	0	2	104
121.34.165.66	ftp	1	52	0	0	1	52
121.34.165.107	ftp	1	52	0	0	1	52
121.34.165.64	ftp	1	52	0	0	1	52
121.34.165.106	ftp	2	104	0	0	2	104
121.34.165.104	ftp	1	52	0	0	1	52
121.34.165.105	ftp	1	52	0	0	1	52
121.34.165.101	ftp	1	52	0	0	1	52
121.34.165.100	ftp	1	52	0	0	1	52
121.34.165.103	ftp	1	52	0	0	1	52

- THEN, YOU CAN AN  
WANT

# FLOW GRAPHS

The image shows the Wireshark network protocol analyzer interface. The main window displays a packet capture from the interface '802.11 Channel'. The packet list pane shows 14 packets, with packet 102 selected. The packet details pane shows the structure of the selected packet: UDP (0x11), User Datagram Protocol, and Simple Network Management Protocol (SNMP). The 'Flow Graph...' option is highlighted in the context menu. The packet bytes pane shows the raw data of the selected packet.

**File:** "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06    Packets: 17 Displayed: 17 Marked: 0    Profile: Default

No.	Time	Source
1	0.000000	192.168.1.104
2	0.017162	192.168.1.102
3	3.017086	192.168.1.104
4	3.034572	192.168.1.102
5	4.626878	192.168.1.104
6	4.663785	63.240.76.102
7	4.675312	192.168.1.102
8	4.694429	128.119.2.102
9	4.694458	192.168.1.102
10	4.694850	192.168.1.102
11	4.717289	128.119.2.102
12	4.718993	128.119.2.102
13	4.724332	192.168.1.102
14	4.750366	128.119.2.102

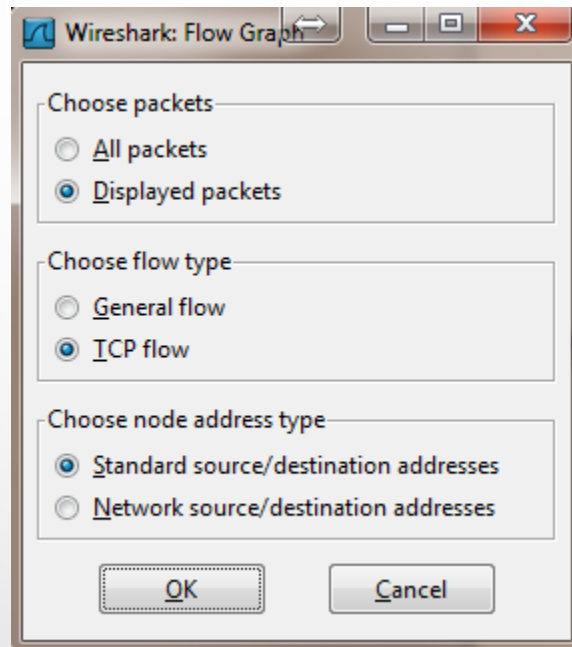
**Protocol:** UDP (0x11)  
Header checksum: 0x0da7  
Source: 192.168.1.104 (192.168.1.104)  
Destination: 192.168.1.102 (192.168.1.102)  
**User Datagram Protocol, Src Port: 161, Destination Port: 4125**  
Source port: snmp (161)  
Destination port: opsview-envoy (4125)  
Length: 59  
Checksum: 0x1ec4 [corrected]  
**Simple Network Management Protocol (SNMPv2-SMI)**

**Flow Graph...**

**Port:** opsview-envoy (4125)

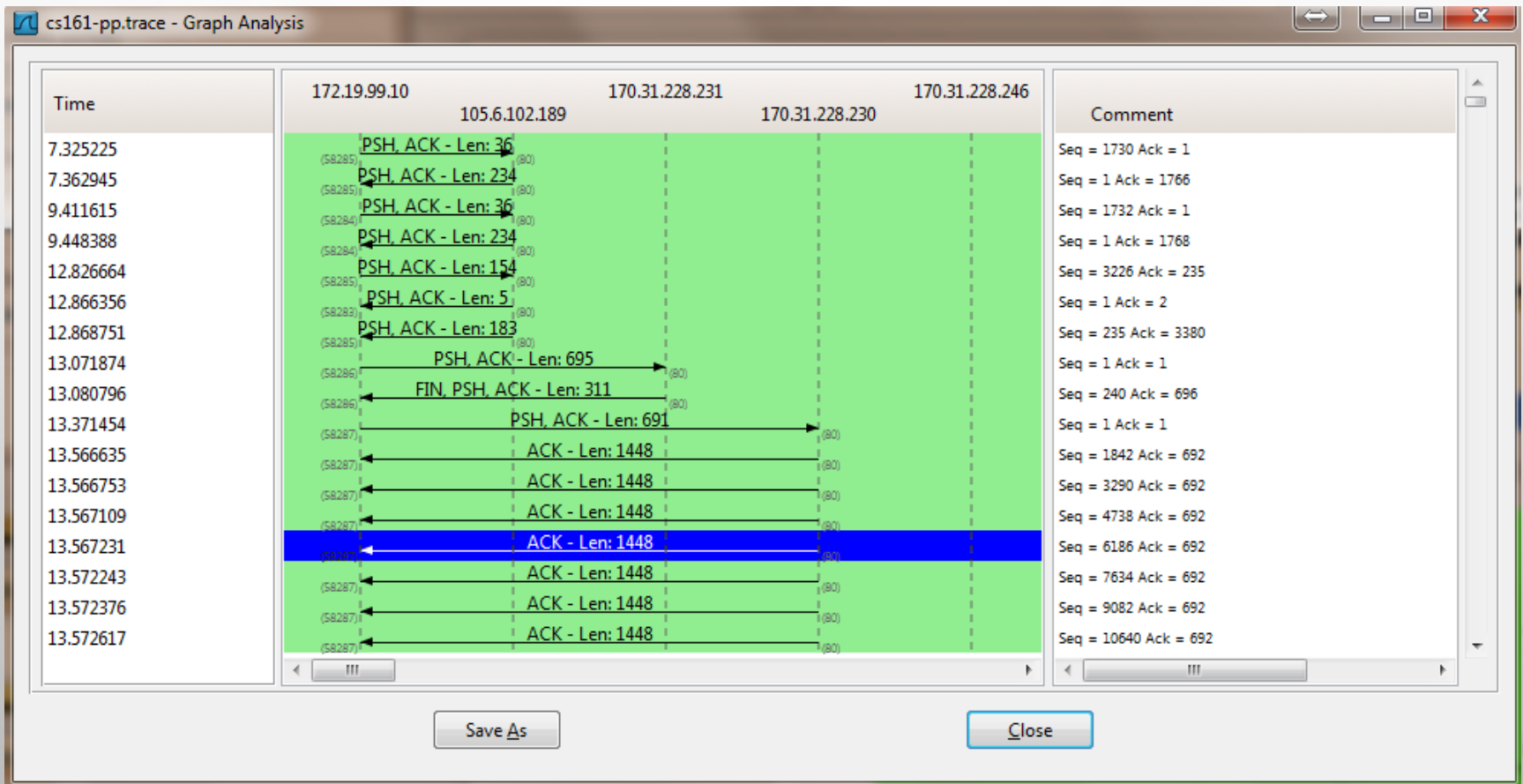
```
0000 00 08 74 4f 36 23 00 30 24 02 02 18 31 02 01 00
0010 00 4f ec d8 00 00 3c 11 11 2b 06 01 04 01 0b 02
0020 01 66 00 a1 10 1d 00 3b 01 00 04 01 10
0030 06 70 75 62 6c 69 63 a2 24 02 02 18 31 02 01 00
0040 02 01 00 30 18 30 16 06 11 2b 06 01 04 01 0b 02
0050 02 00 04 02 01 02 02 02 01 00 04 01 10
```

# FLOW GRAPHS



- The “displayed packet” option could let you only Show the flow of packets shown up  
for example, only display http traffic, then show The flow to analyze

# FLOW GRAPHS



Tucker Ellis & West http-ethereal-trace-1 - Wireshark

File Edit View Go Capture Analyze Statistics Help

Open... Ctrl+O  
 Open Recent  
 Merge...  
 Close Ctrl+W  
 Save Ctrl+S  
 Save As... Shift+Ctrl+S  
 File Set  
 Export  
 Print... Ctrl+P  
 Quit Ctrl+Q

Channel Offset: FCS Filter: Decryption Mode: None

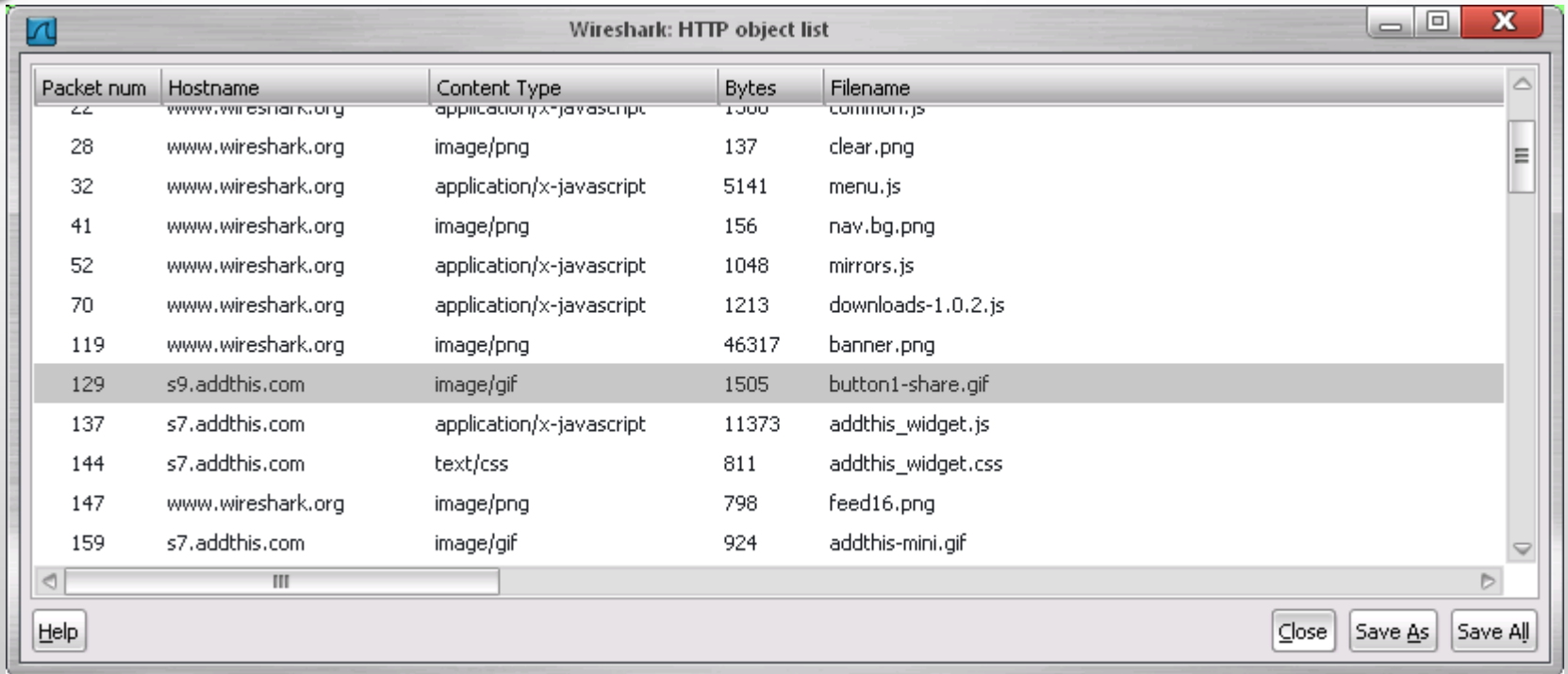
No.	Time	Source	Destination	Protocol	Info	
9	4.694458	192.168.1.102	192.168.1.104	SNMP	get-request SNMPV2-SMI	
10	4.694850	192.168.1.104	192.168.1.102	SNMP	get-response SNMPV2-SMI	
11	4.717289	128.119.245.12	192.168.1.102	SNMP	get-request SNMPV2-SMI	
12	4.718993	128.119.245.12	192.168.1.102	SNMP	get-response SNMPV2-SMI	
13	4.724332	192.168.1.102	128.119.245.12	DNS	Standard query A gai...	
14	4.750366	128.119.245.12	192.168.1.102	DNS	Standard query response	
		168.1.102	128.119.245.12	TCP	unikeypro > http [SYN]	
		119.245.12	192.168.1.102	TCP	http > unikeypro [SYN,	
		9 4.694458	192.168.1.102	128.119.245.12	TCP	unikeypro > http [ACK]
		10 4.694850	192.168.1.102	128.119.245.12	HTTP	GET /ethereal-labs/lab2
		11 4.717289	128.119.245.12	192.168.1.102	TCP	http > unikeypro [ACK]
		12 4.718993	128.119.245.12	192.168.1.102	HTTP	HTTP/1.1 200 OK (text/
		13 4.724332	192.168.1.102	128.119.245.12	HTTP	GET /favicon.ico HTTP/3
		14 4.750366	128.119.245.12	192.168.1.102	HTTP	HTTP/1.1 404 Not Found

Source port: unikeypro (4127)  
 Destination port: http (80)  
 Sequence number: 1 (relative sequence number)  
 [Next sequence number: 502 (relative sequence number)]  
 Acknowledgement number: 1 (relative ack number)  
 Header length: 20 bytes  
 Flags: 0x18 (PSH, ACK)  
 0... .. = Congestion window Reduced (CWR): Not set  
 .0.. .... = ECN-Echo: Not set  
 ..0. .... = Urgent: Not set

0020 f5 0c 10 1f 00 50 f5 32 64 b2 6b a6 54 92 50 18 ... .P.2 d.k.T.P.  
 0030 fa f0 39 a2 00 00 47 45 54 20 2f 65 74 68 65 72 ... .9...GE T /ether  
 0040 65 61 6c 2d 6c 61 62 73 2f 6c 61 62 32 2d 31 2e eal-labs /lab2-1.  
 0050 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 html HTT P/1.1..H  
 0060 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75 6d 61 ost: gai a.cs.uma  
 0070 72 72 2a 65 64 75 0d 02 55 72 65 72 2d 41 67 65 ss.edu User Age

Destination Port (tcp.dstport), 2 bytes | Packets: 17 Displayed: 17 Marked: 0 | Profile: Default

# EXPORT HTTP OBJECTS



The image shows a screenshot of the 'Wireshark: HTTP object list' window. The window contains a table with the following columns: Packet num, Hostname, Content Type, Bytes, and Filename. The table lists various objects from different hosts, including www.wireshark.org and s7.addthis.com. The object at packet 129 is highlighted.

Packet num	Hostname	Content Type	Bytes	Filename
22	www.wireshark.org	application/x-javascript	1300	common.js
28	www.wireshark.org	image/png	137	clear.png
32	www.wireshark.org	application/x-javascript	5141	menu.js
41	www.wireshark.org	image/png	156	nav.bg.png
52	www.wireshark.org	application/x-javascript	1048	mirrors.js
70	www.wireshark.org	application/x-javascript	1213	downloads-1.0.2.js
119	www.wireshark.org	image/png	46317	banner.png
129	s9.addthis.com	image/gif	1505	button1-share.gif
137	s7.addthis.com	application/x-javascript	11373	addthis_widget.js
144	s7.addthis.com	text/css	811	addthis_widget.css
147	www.wireshark.org	image/png	798	feed16.png
159	s7.addthis.com	image/gif	924	addthis-mini.gif

At the bottom of the window, there are buttons for 'Help', 'Close', 'Save As', and 'Save All'.

# HTTP ANALYSIS

The image shows the Wireshark network protocol analyzer interface. The main window displays a list of captured packets, with packet 18 selected. The packet list pane shows the following details:

No.	Time	Destination	Protocol	Info
1	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
2	2008-07-24 13:12:59.1	10.8.117.254.150	TCP	acc-raid > http [ACK] Seq=
3	2008-07-24 13:12:59.1	10.4.2.184.130	HTTP	GET /p/s/sm_vrt_3thumb_scr
4	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
5	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
6	2008-07-24 13:12:59.1	10.8.117.254.150	TCP	acc-raid > http [ACK] Seq=
7	2008-07-24 13:12:59.1	10.9.166.161.121	DNS	Standard query A a632.g.ak
8	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
9	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
10	2008-07-24 13:12:59.1	10.8.117.254.150	TCP	acc-raid > http [ACK] Seq=
11	2008-07-24 13:12:59.1	10.23.58.126	HTTP	GET /customer/advance/9/.o
12	2008-07-24 13:12:59.1	10.23.58.126	HTTP	GET /customer/advance/9/.o
13	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
14	2008-07-24 13:12:59.2	10.1.11.13	DNS	Standard query response CN
15	2008-07-24 13:12:59.1	10.8.180.195.70	TCP	mcs-calyptoicf > http [SYN
16	2008-07-24 13:12:59.1	10.1.12.67	HTTP	Continuation or non-HTTP t
17	2008-07-24 13:12:59.1	10.2.201.36	TCP	3325 > http [ACK] Seq=1 Ac
18	2008-07-24 13:12:59.1	10.1.12.67	HTTP	[TCP out-of-order] Continu
19	2008-07-24 13:12:59.1	10.2.201.36	TCP	3325 > http [ACK] Seq=1 Ac

The packet details pane for the selected packet (18) shows the following structure:

- Frame 1 (1514 bytes on wire)
- Ethernet II, Src: Cisco\_f7
- Internet Protocol, Src: 20
- Transmission Control Proto
- Hypertext Transfer Protoco

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 00 15 c7 46 80 00 00 03
0010 05 dc 36 ab 40 00 3b 06
0020 0f 68 00 50 0a f0 94 cf
0030 1b 96 ab f0 00 00 46 1f
0040 9a b1 fd cf c7 ff fd ca
0050 23 c8 05 00 2c 86 fb 25
c8 b8 7f 8e cb b4 85 4d
51 00 7c 2f 2f 0e 0d
```

The ASCII pane shows the following text:

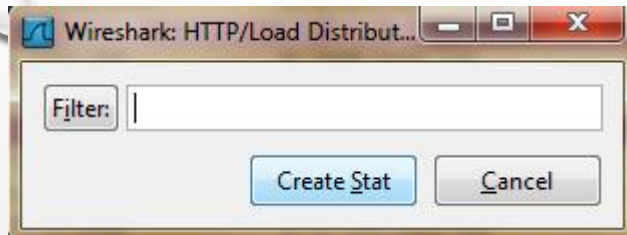
```
...F... k...E.
01 ..6.@.;. .u...
10 .h.P... .4...P.
19 .....F...X..
.....M
< %01/
```

The Statistics pane is open to the HTTP section, showing a Load Distribution... dialog box with the following data:

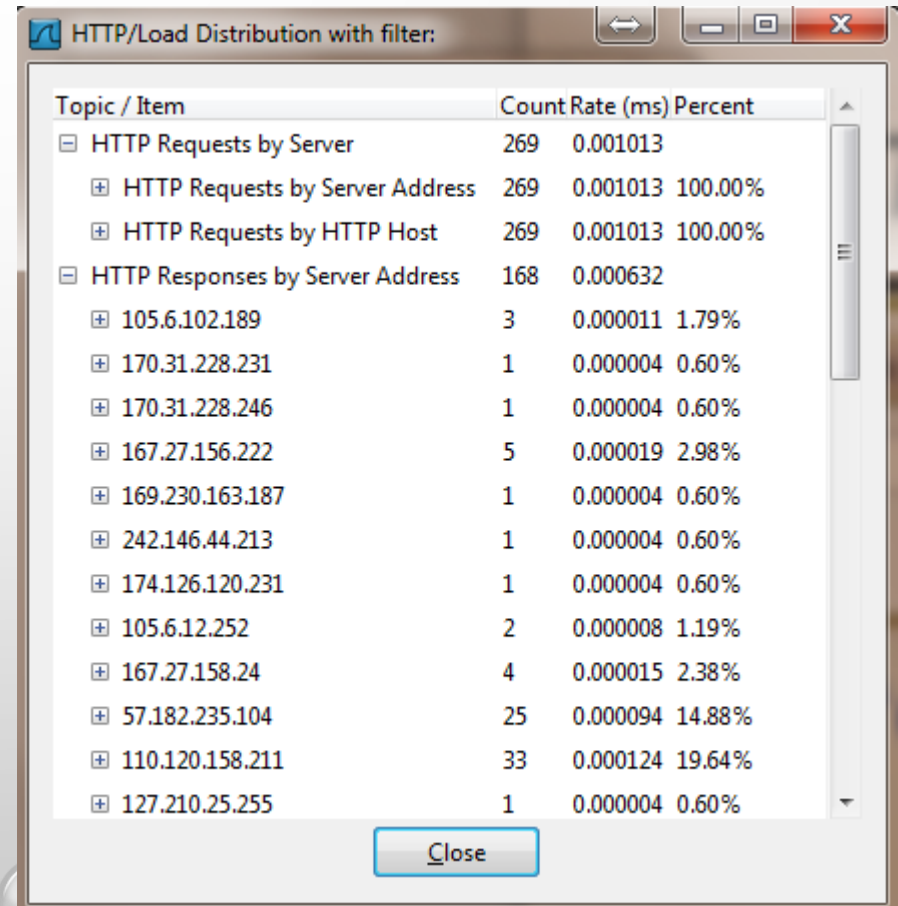
Requests...	o_46:80:00 (00:15:c7:46:80:00)
10.1.15.104 (10.1.15.104)	

The status bar at the bottom indicates: File: "C:\Users\vo2.TEW\Desktop\wireshark\sample capture..." Packets: 16612 Displayed: 16612 Marked: 0 Profile: Default

# HTTP ANALYSIS – LOAD DISTRIBUTION



Click “Create Stat” button  
You can add “filter” to only  
Show selected traffic

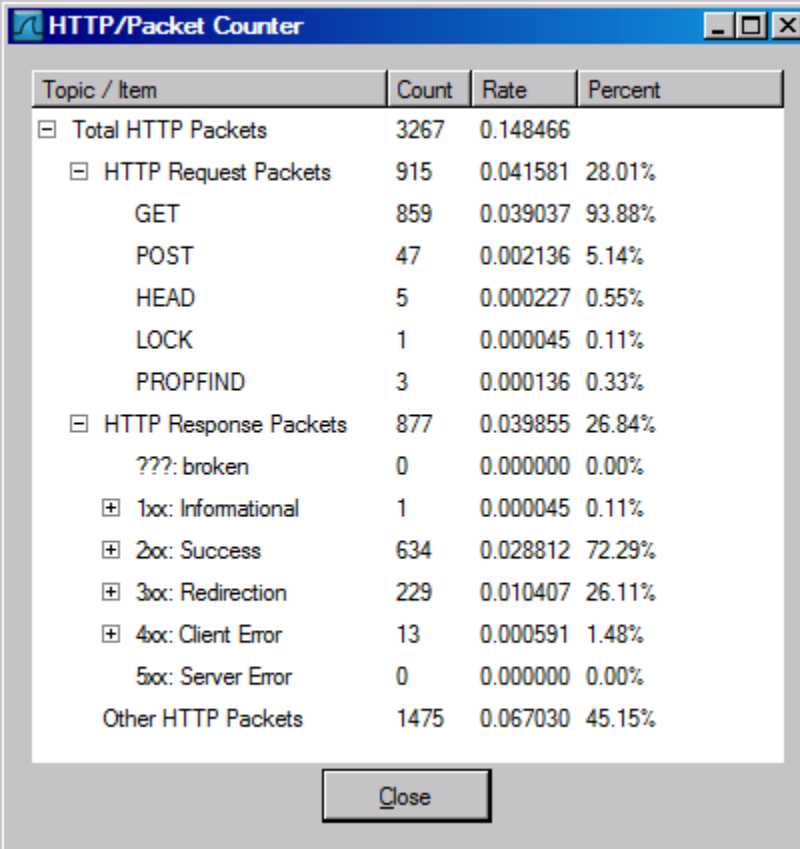


A window titled "HTTP/Load Distribution with filter:". It displays a table with three columns: "Topic / Item", "Count", "Rate (ms)", and "Percent". The table is expanded to show a list of server addresses under the "HTTP Responses by Server Address" category.

Topic / Item	Count	Rate (ms)	Percent
[-] HTTP Requests by Server	269	0.001013	
[+] HTTP Requests by Server Address	269	0.001013	100.00%
[+] HTTP Requests by HTTP Host	269	0.001013	100.00%
[-] HTTP Responses by Server Address	168	0.000632	
[+] 105.6.102.189	3	0.000011	1.79%
[+] 170.31.228.231	1	0.000004	0.60%
[+] 170.31.228.246	1	0.000004	0.60%
[+] 167.27.156.222	5	0.000019	2.98%
[+] 169.230.163.187	1	0.000004	0.60%
[+] 242.146.44.213	1	0.000004	0.60%
[+] 174.126.120.231	1	0.000004	0.60%
[+] 105.6.12.252	2	0.000008	1.19%
[+] 167.27.158.24	4	0.000015	2.38%
[+] 57.182.235.104	25	0.000094	14.88%
[+] 110.120.158.211	33	0.000124	19.64%
[+] 127.210.25.255	1	0.000004	0.60%



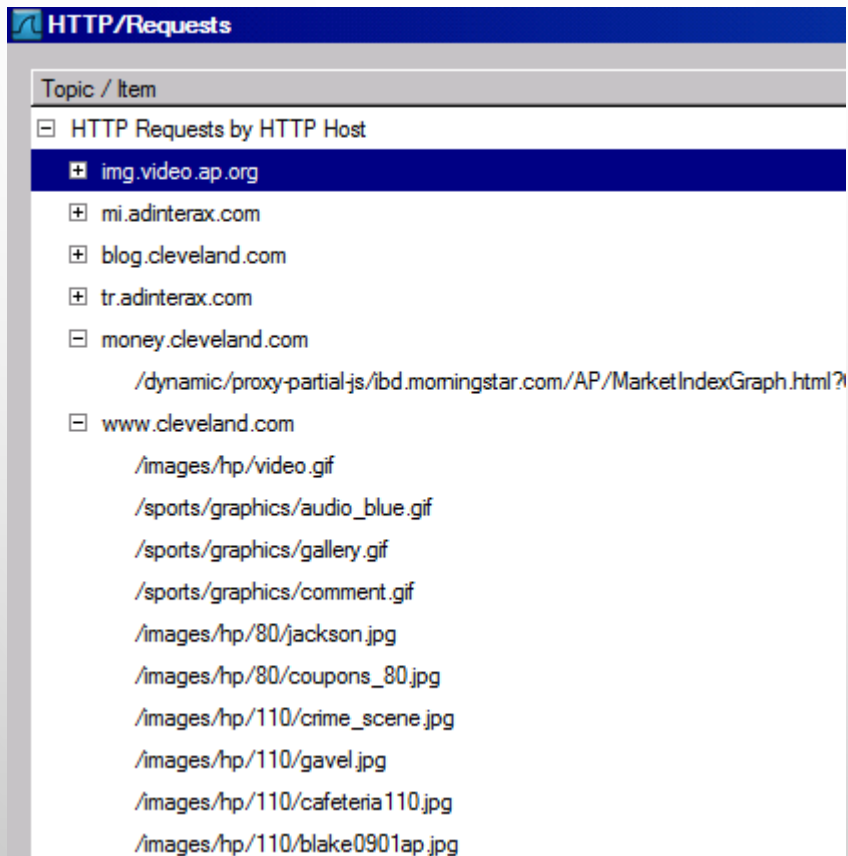
# HTTP ANALYSIS – PACKET COUNTER



The screenshot shows a window titled "HTTP/Packet Counter" with a table of statistics. The table has four columns: "Topic / Item", "Count", "Rate", and "Percent". The data is organized into a tree structure with expandable/collapsible icons. A "Close" button is located at the bottom center of the window.

Topic / Item	Count	Rate	Percent
[-] Total HTTP Packets	3267	0.148466	
[-] HTTP Request Packets	915	0.041581	28.01%
GET	859	0.039037	93.88%
POST	47	0.002136	5.14%
HEAD	5	0.000227	0.55%
LOCK	1	0.000045	0.11%
PROPFIND	3	0.000136	0.33%
[-] HTTP Response Packets	877	0.039855	26.84%
???: broken	0	0.000000	0.00%
[+] 1xx: Informational	1	0.000045	0.11%
[+] 2xx: Success	634	0.028812	72.29%
[+] 3xx: Redirection	229	0.010407	26.11%
[+] 4xx: Client Error	13	0.000591	1.48%
5xx: Server Error	0	0.000000	0.00%
Other HTTP Packets	1475	0.067030	45.15%

# HTTP ANALYSIS – REQUESTS



The screenshot displays a web application window titled "HTTP/Requests". Below the title bar is a search field labeled "Topic / Item". The main content area shows a tree view of HTTP requests, organized by host. The "img.video.ap.org" host is currently selected and highlighted in blue. Other hosts listed include mi.adinterax.com, blog.cleveland.com, tr.adinterax.com, money.cleveland.com, and www.cleveland.com. Under the www.cleveland.com host, several image and video files are listed, such as /images/hp/video.gif, /sports/graphics/audio\_blue.gif, and various .jpg and .gif files.

Topic / Item
[-] HTTP Requests by HTTP Host
+ img.video.ap.org
+ mi.adinterax.com
+ blog.cleveland.com
+ tr.adinterax.com
[-] money.cleveland.com
/dynamic/proxy-partial-js/ibd.momingstar.com/AP/MarketIndexGraph.html?
[-] www.cleveland.com
/images/hp/video.gif
/sports/graphics/audio_blue.gif
/sports/graphics/gallery.gif
/sports/graphics/comment.gif
/images/hp/80/jackson.jpg
/images/hp/80/coupons_80.jpg
/images/hp/110/crime_scene.jpg
/images/hp/110/gavel.jpg
/images/hp/110/cafeteria110.jpg
/images/hp/110/blake0901ap.jpg

# IMPROVING WIRESHARK PERFORMANCE

- DON'T USE CAPTURE FILTERS
- INCREASE YOUR READ BUFFER SIZE
- DON'T UPDATE THE SCREEN DYNAMICALLY
- GET A FASTER COMPUTER
- USE A TAP
- DON'T RESOLVE NAMES

# POST-PROCESSING TEXT FILE

- FOR SAVED TEXT-FORMAT PACKET FILES, FURTHER ANALYSIS NEEDS CODING OR SPECIAL TOOLS
- ONE USEFUL TOOL ON UNIX: GREP
  - ON WINDOWS: POWERGREP [HTTP://WWW.POWERGREP.COM/](http://www.powergrep.com/)
  - COMMAND-LINE BASED UTILITY FOR SEARCHING PLAIN-TEXT DATA SETS FOR LINES MATCHING A REGULAR EXPRESSION.

# BASIC USAGE OF GREP

## • COMMAND-LINE **TEXT-SEARCH** PROGRAM IN LINUX

### • SOME USEFUL USAGE:

- GREP 'WORD' FILENAME # FIND LINES WITH 'WORD'
- GREP -V 'WORD' FILENAME # FIND LINES WITHOUT 'WORD'
- GREP '^WORD' FILENAME # FIND LINES BEGINNING WITH 'WORD'
- GREP 'WORD' FILENAME > FILE2 # OUTPUT LINES WITH 'WORD' TO FILE2
- LS -L | GREP RWXRWXRWX # LIST FILES THAT HAVE 'RWXRWXRWX' FEATURE
- GREP '[0-4]' FILENAME # FIND LINES BEGINNING WITH ANY OF THE NUMBERS FROM 0-4
- GREP -C 'WORD' FILENAME # FIND LINES WITH 'WORD' AND PRINT OUT THE NUMBER OF THESE LINES
- GREP -I 'WORD' FILENAME # FIND LINES WITH 'WORD' REGARDLESS OF CASE

### • MANY TUTORIALS ON GREP ONLINE

- [HTTP://WWW.CYBERCITI.BIZ/FAQ/HOWTO-USE-GREP-COMMAND-IN-LINUX-UNIX/](http://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/)
- [HTTP://WWW.THEGEEKSTUFF.COM/2009/03/15-PRACTICAL-UNIX-GREP-COMMAND-EXAMPLES/](http://www.thegEEKstuff.com/2009/03/15-practical-unix-grep-command-examples/)