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?

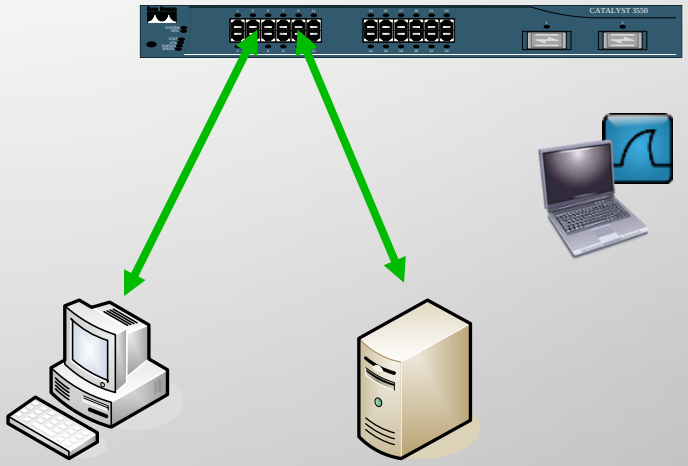
- Packets Don't Lie
- We can't fix what we can't see
- Traffic and Protocol Analysis is the most detailed troubleshooting method

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

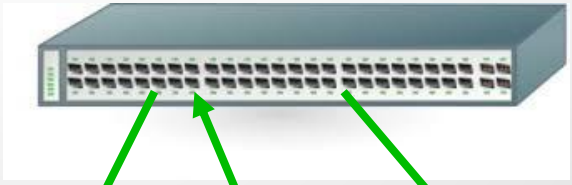
PACKET COLLECTION

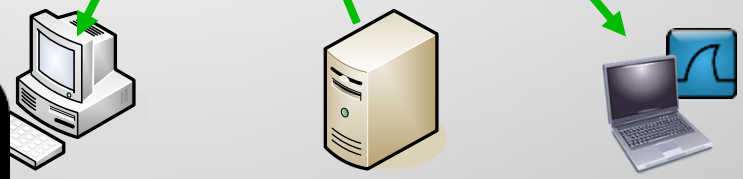
- COMMON CAPTURE METHODS:
  - SPAN/MIRROR
  - TAP / AGGREGATION TAP
  - DIRECTLY ON THE CLIENT



GETTING IN THE PATH: SPAN/MIRROR

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

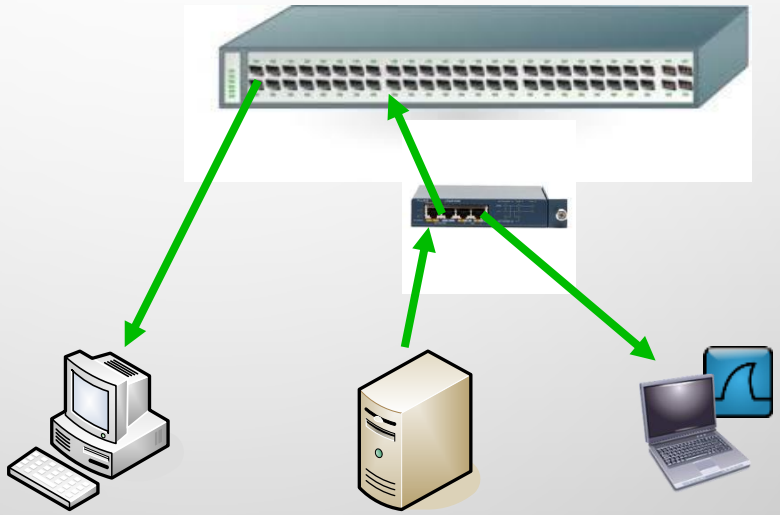




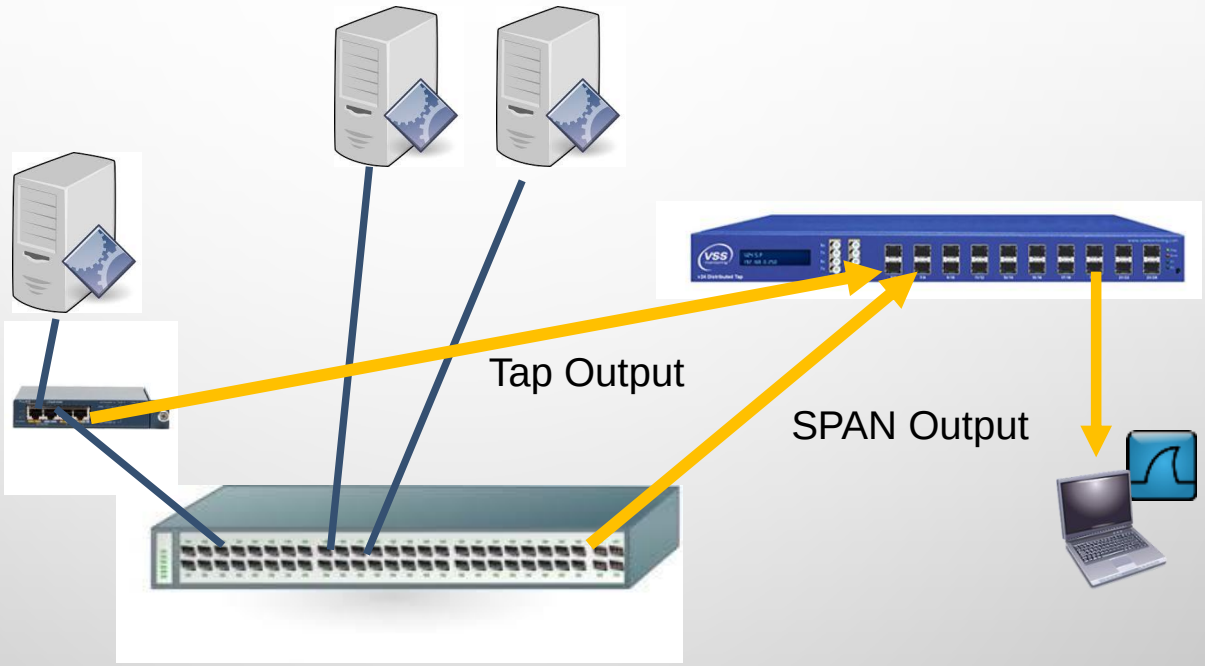
```
                    Sniffer
                   /
Internet--Firewall--HUB
                   \
                    Switch--PC1
                  /   |   \
               PC2  PC3  PC4
```

GETTING IN THE PATH: TAPS

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

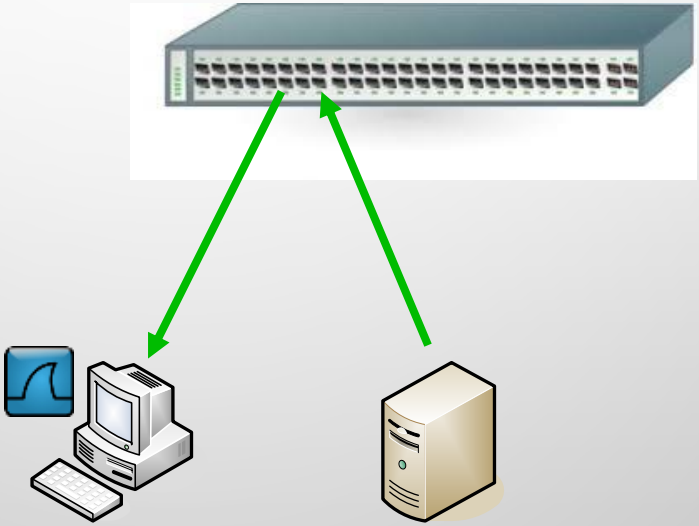


GETTING IN THE PATH: AGGREGATION TAP



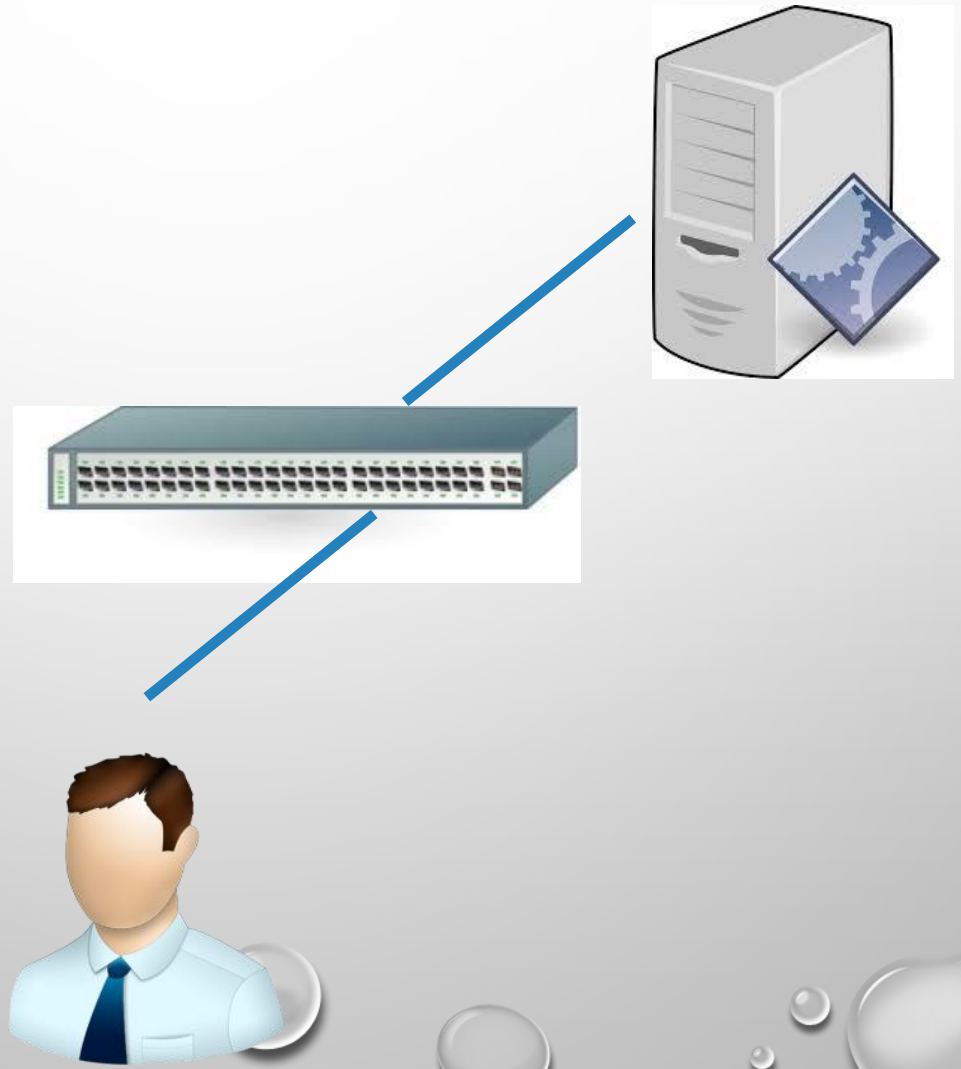
CAPTURING DIRECTLY FROM CLIENT

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

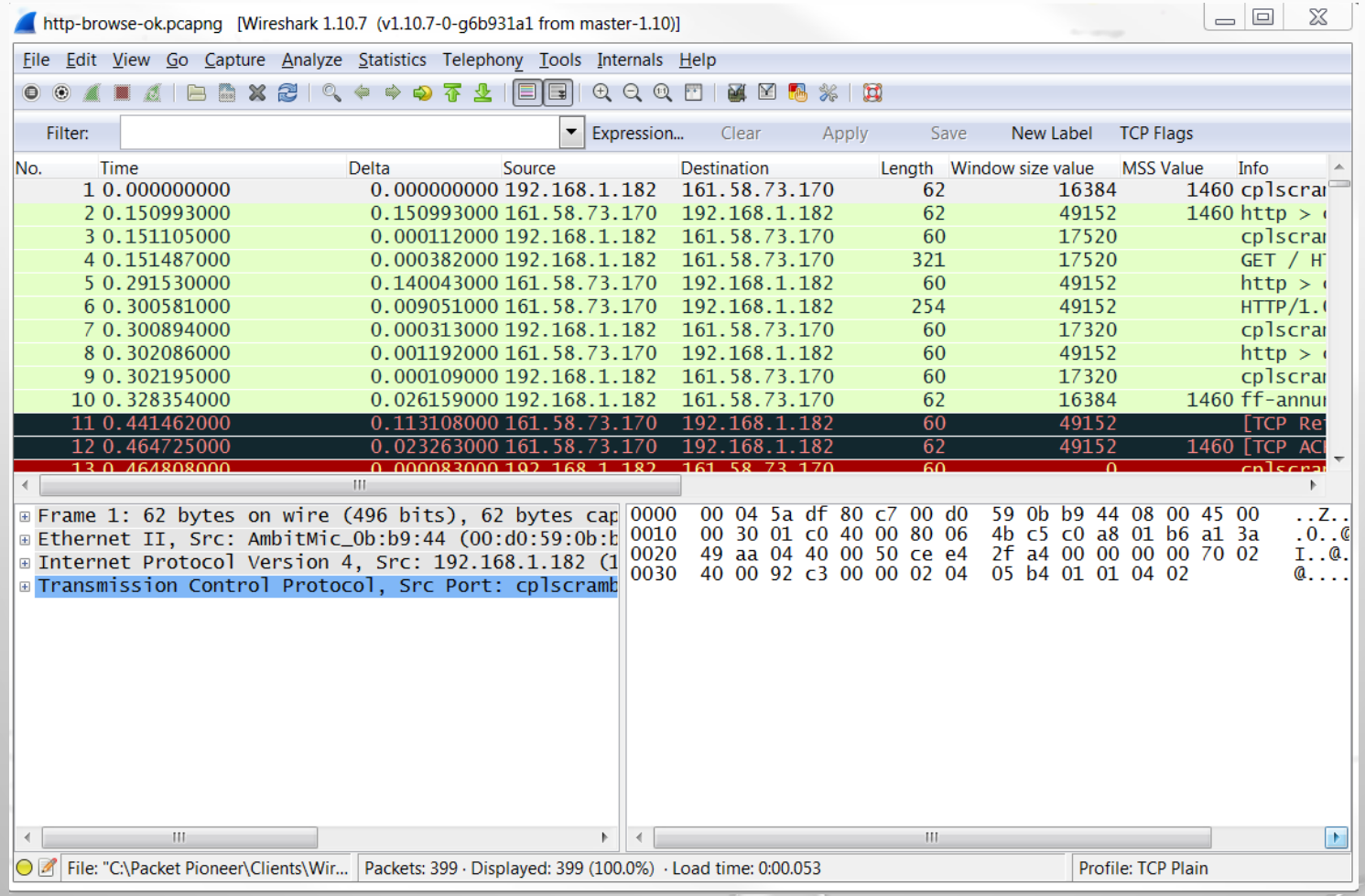


PROBLEMS AREN'T WHAT THEY USED TO BE

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!



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.

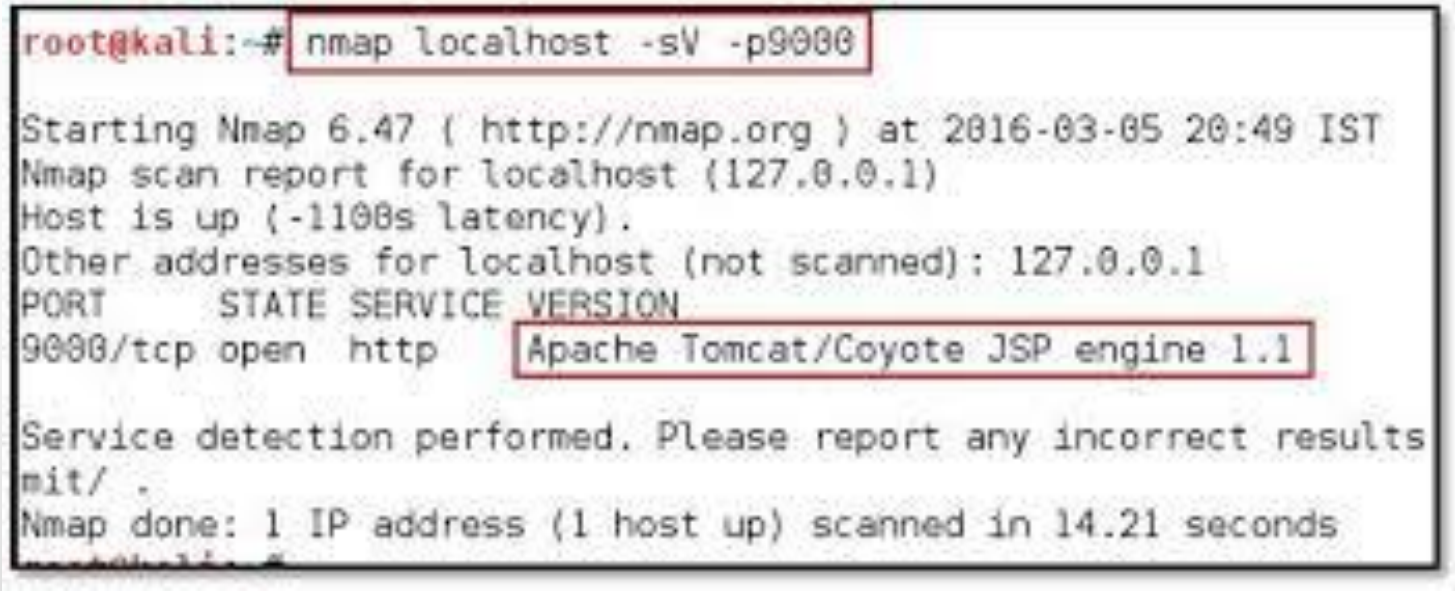
There are two basic approaches for listing the ports that are listening on the network. The less reliable approach is to query the network stack by typing commands such as `netstat -an` or `lsof -i`.

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:

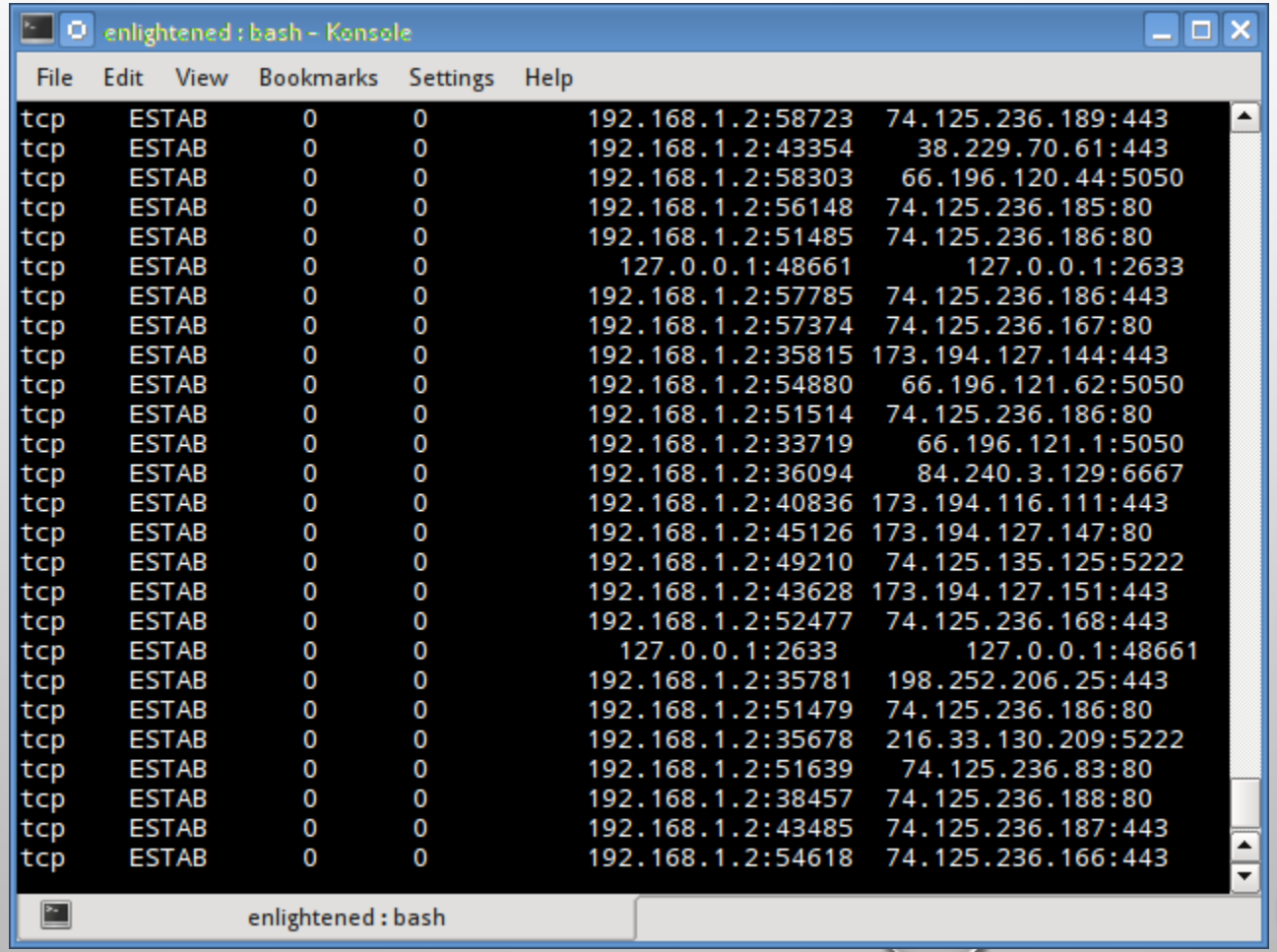
```
nmap -sT -O localhost
```


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

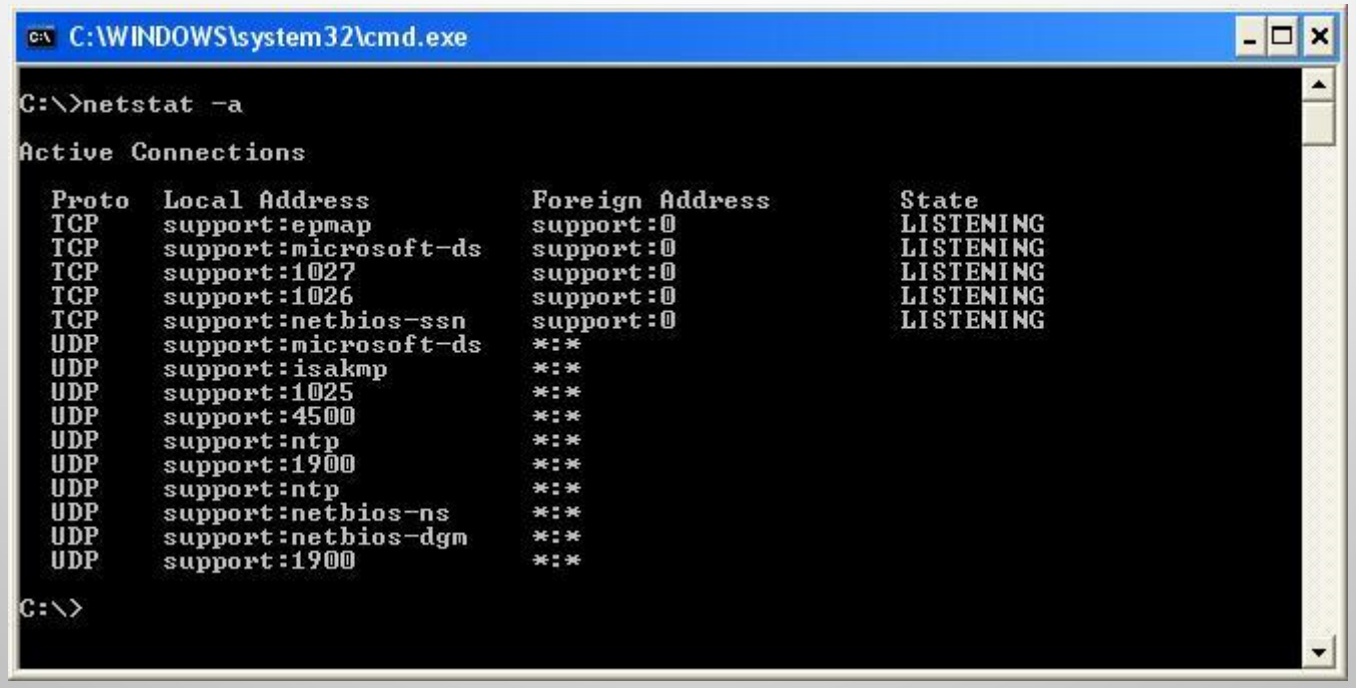
- SS –TULNP

- TO CONSULT THE LIST OF ALL OPEN PORTS SS -AP



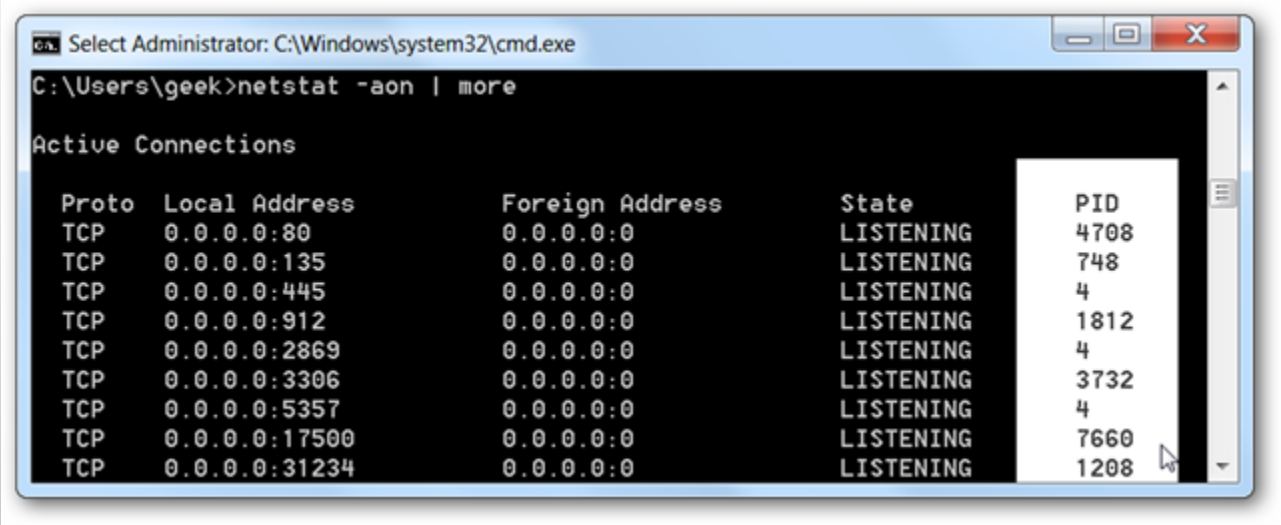
WINDOWS

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

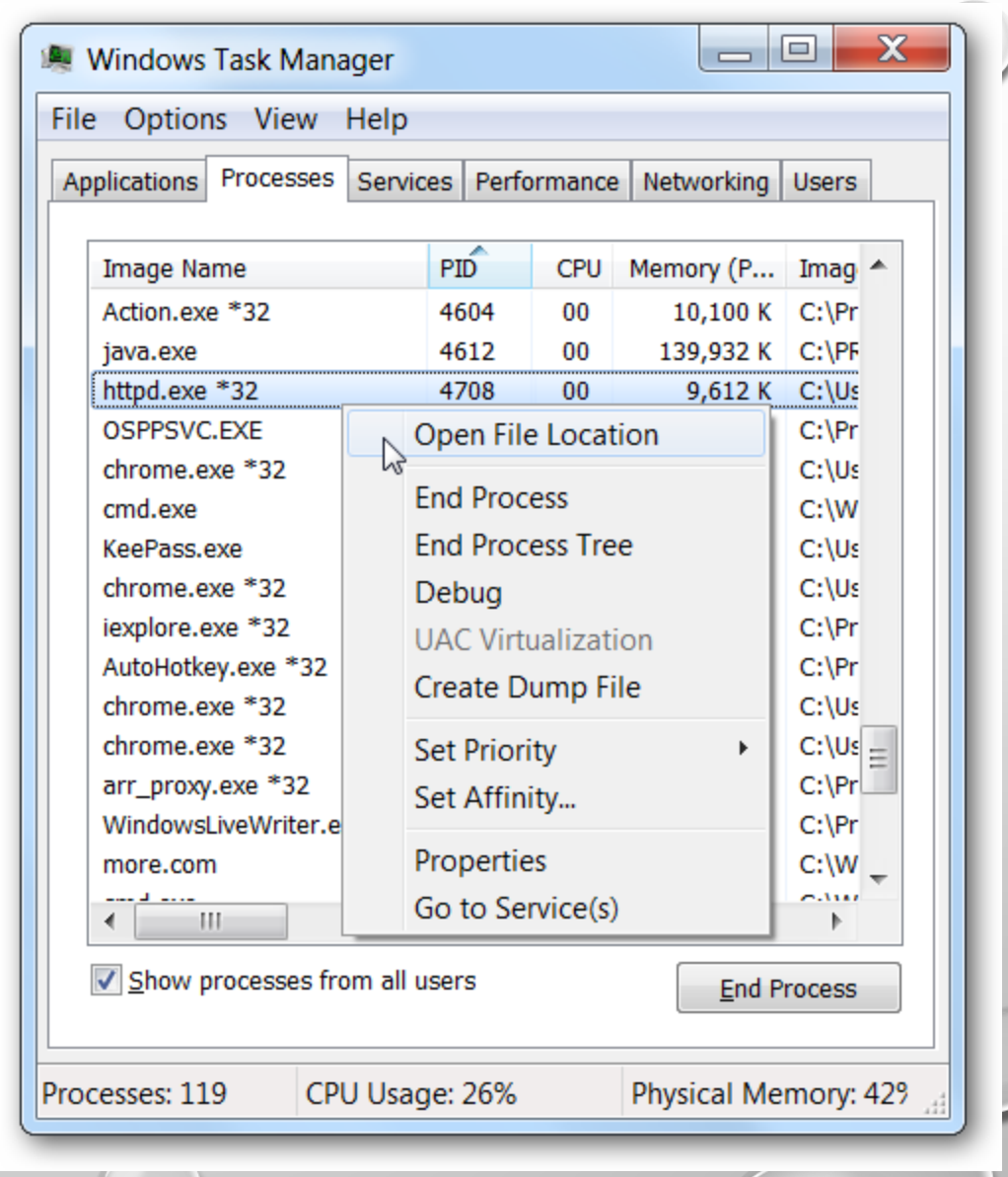


`netstat -aon | more`

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.

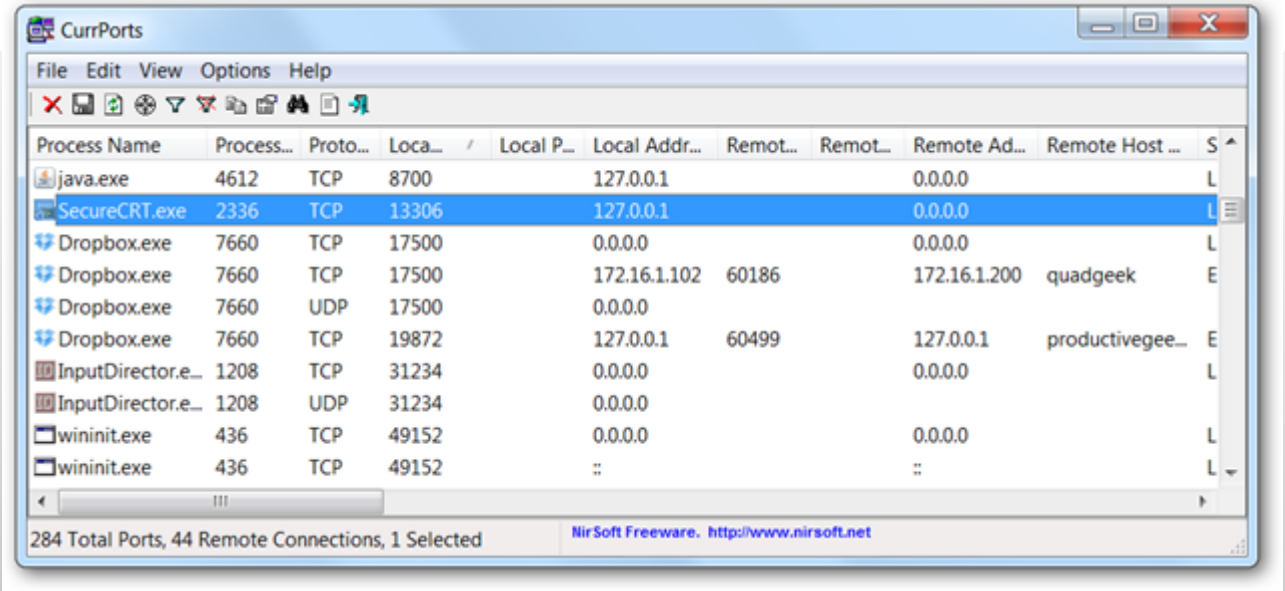


Now you can simply open up Task Manager—you might have to use the option to Show Processes for All Users, and then you'll be able to find the PID in the list. Once you're there, you can use the End Process, Open File Location, or Go to Service(s) options to control the process or stop it.



Use CurrPorts to View What is Listening

excellent freeware CurrPorts utility by NirSoft



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:

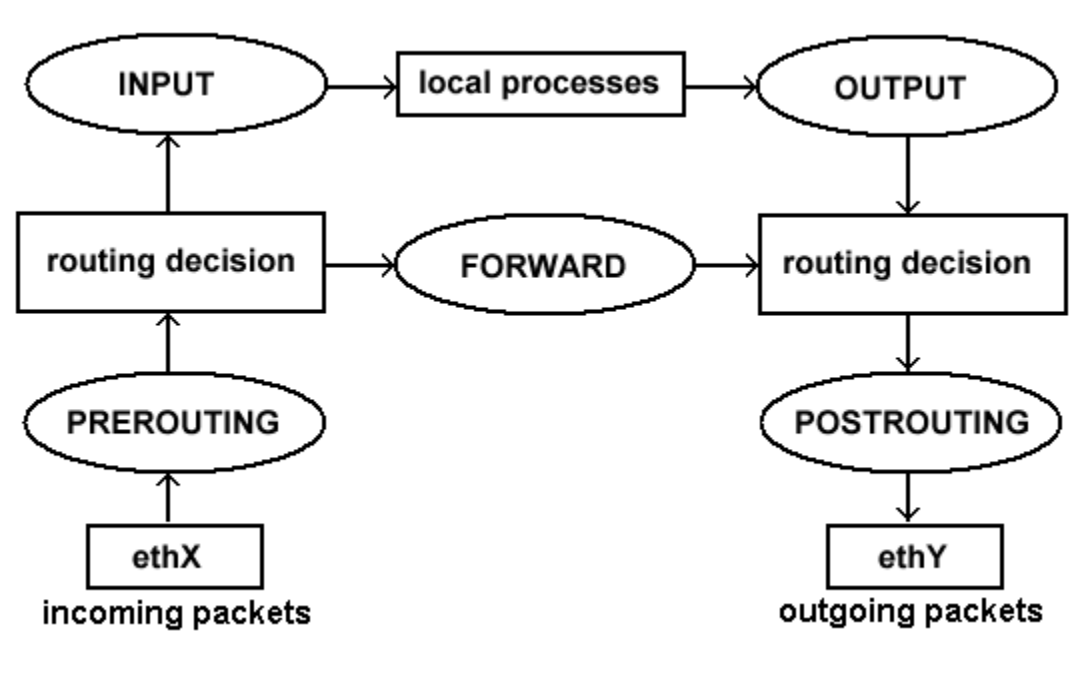
1. PACKET FILTERING
2. APPLICATION PROXY SERVER

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 TOPROHIBIT 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

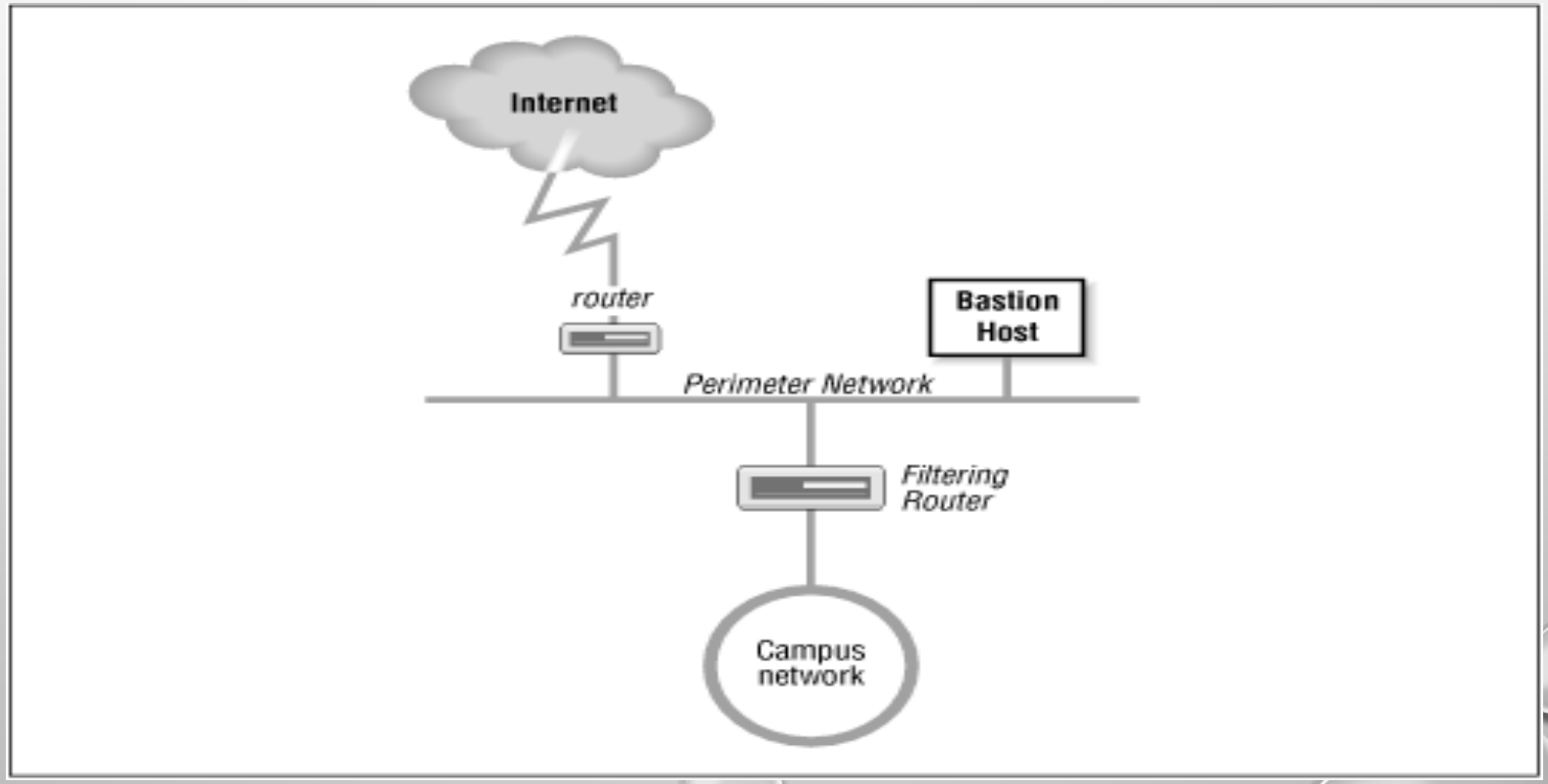
- ALL PACKETS INTO AND OUT OF THE PROTECTED NETWORK MUST PASS THROUGH THE FIREWALL
- FIREWALL CANNOT BE PENETRATED.

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 NUMBERS

  TO CONTROL THE SERVICES THAT ARE ACCESSIBLE TO AND FROM THE INTERIOR NETWORK.

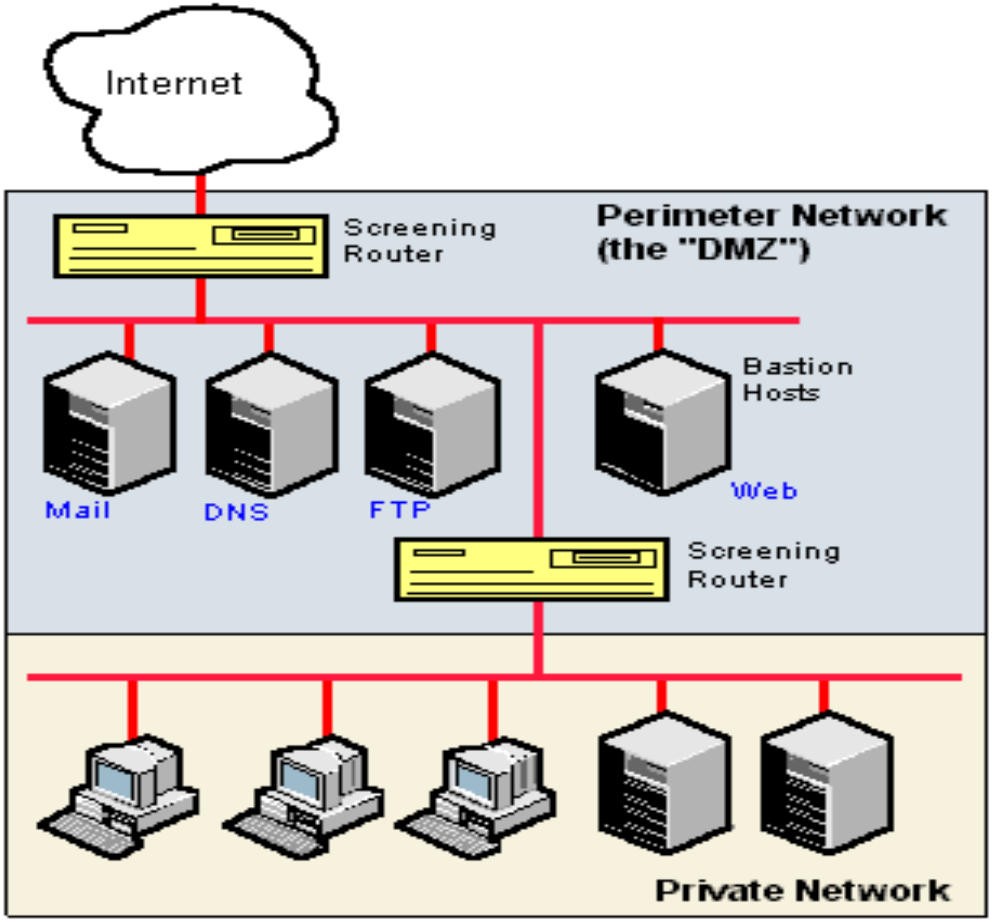
BASTION HOST

- 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*

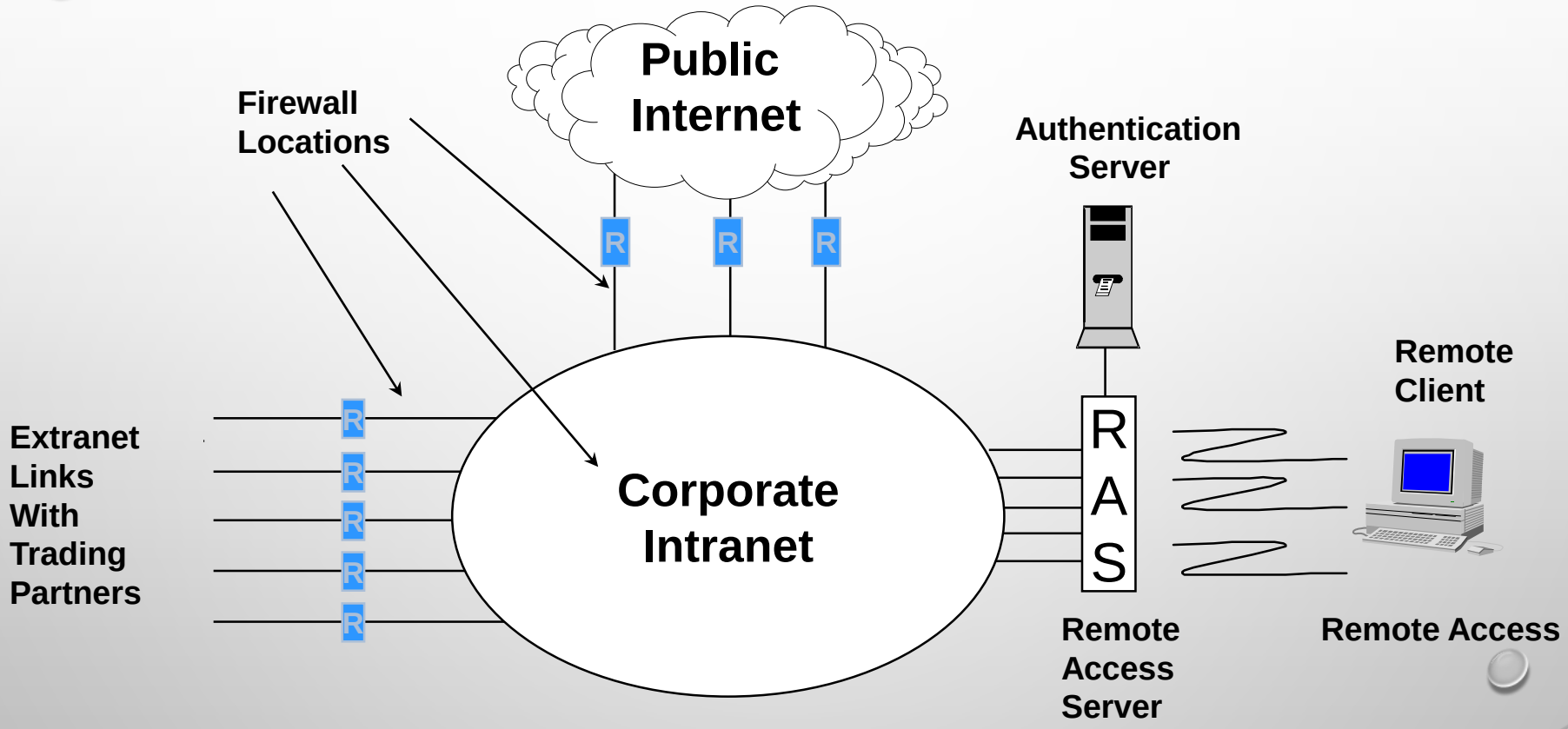
MULTIPLE BASTION HOSTS .3



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.

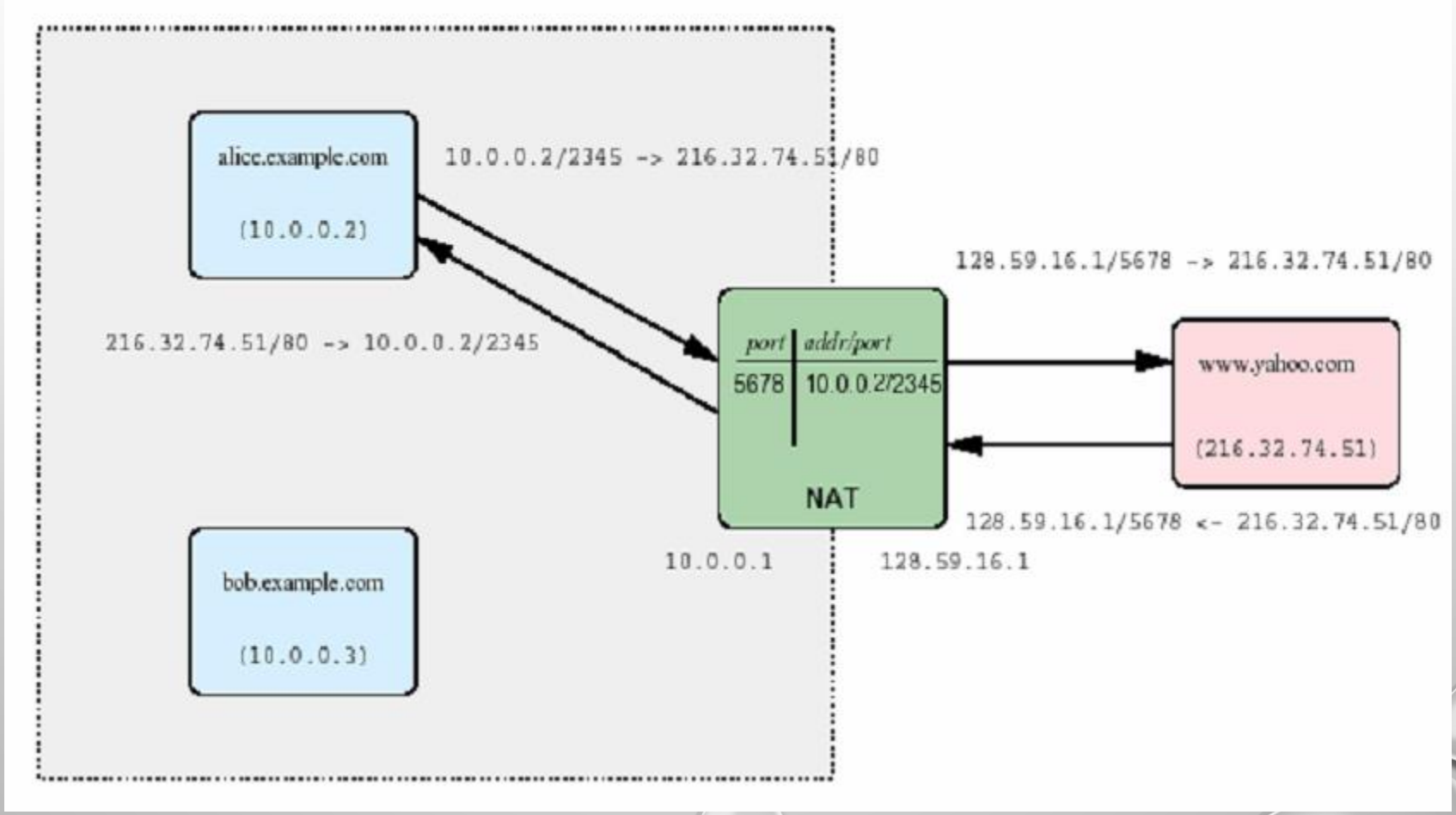
Typical Enterprise Network Topology (without VPN)



NETWORK ADDRESS TRANSLATOR

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

NAT


alice.example.com
(10.0.0.2)
10.0.0.2/2345 -> 216.32.74.51/80
128.59.16.1/5678 -> 216.32.74.51/80
216.32.74.51/80 -> 10.0.0.2/2345
port
addr/port
5678
10.0.0.2/2345
NAT
www.yahoo.com
(216.32.74.51)
128.59.16.1/5678 <- 216.32.74.51/80
bob.example.com
(10.0.0.3)
10.0.0.1
128.59.16.1

IPS VS IDS

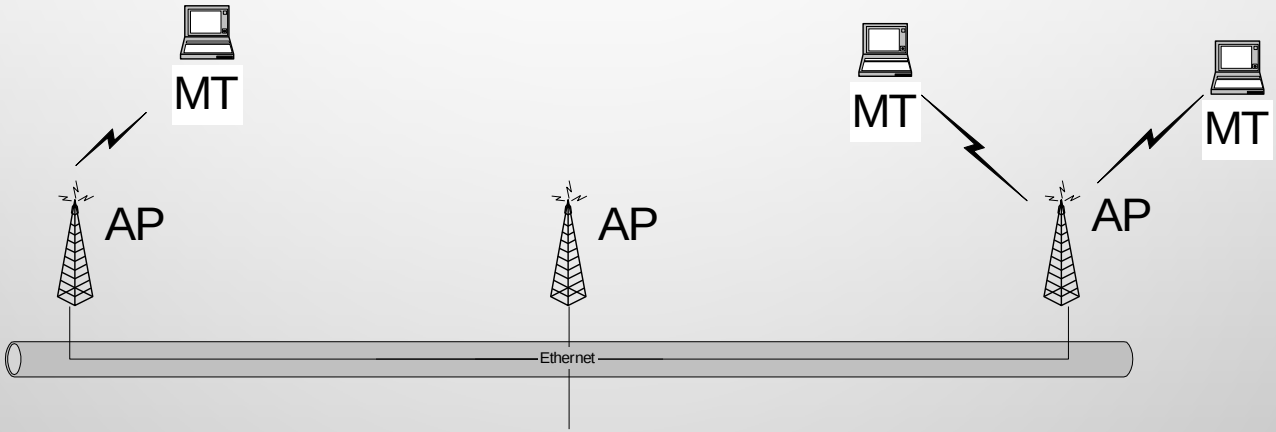
- NEW: IPS: INTRUSION PREVENTION SYSTEMS
- IDS: INTRUSION DETECTION SYSTEMS: IDS DEVICES SIT ON A MONITOR PORT AND SIMPLY REPORT PROBLEMS.
- WHILE AN IPS DEVICE TAKES ACTION, IDS PRODUCTS USUALLY JUST SEND AN ALERT TO AN IT STAFF PERSON, WHO MUST THEN EVALUATE THE ALERT AND TAKE ACTION.
- PROBLEM WITH 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)

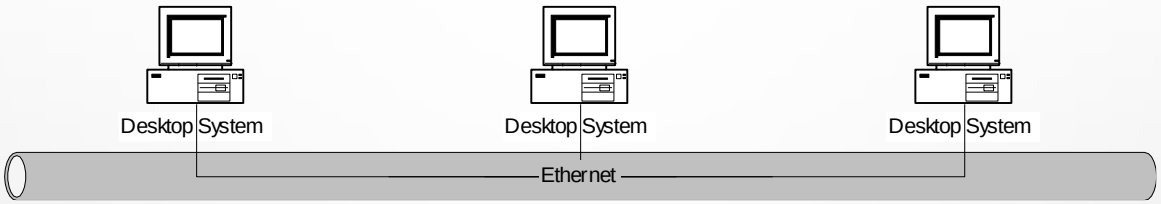


- Standard also supports ad-hoc networking where MT's talk directly to MT's

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



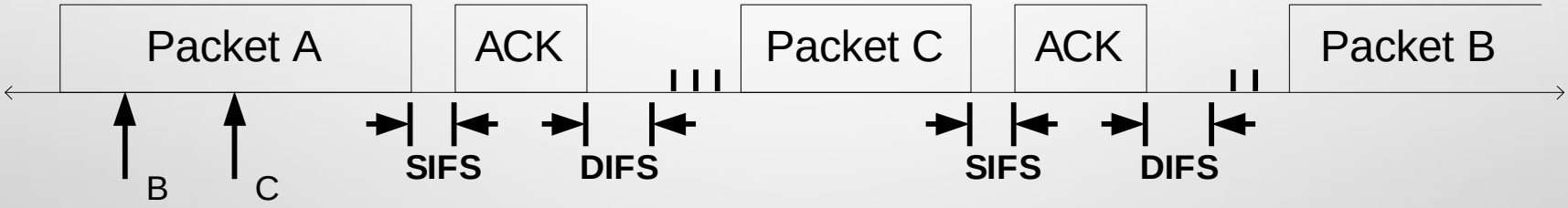
- CSMA/CD (CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION)
  - IF MEDIA IS SENSED IDLE, TRANSMIT
  - IF MEDIA IS SENSED BUSY, WAIT UNTIL IDLE AND THEN TRANSMIT IMMEDIATELY
- COLLISIONS CAN OCCUR IF MORE THAN ONE USER TRANSMITS AT THE SAME TIME
  - IF A COLLISION IS DETECTED, STOP TRANSMITTING.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

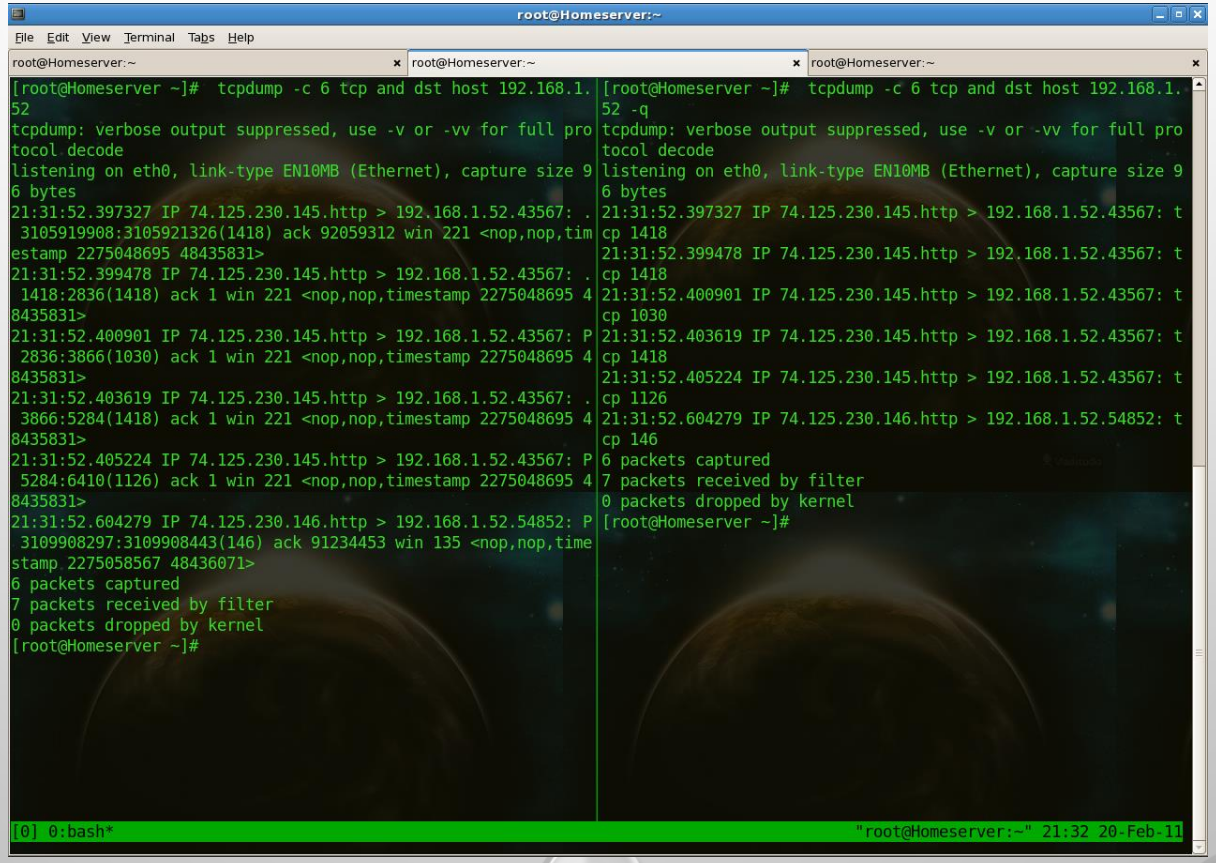
(II) MANAGEMENT: USE ONE CELL OF A BASE STATION → NO AD 4 FIELD.

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

NETWORK ANALYZER

TCPDUMP

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



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 1ST THROUGH 1025TH (NOT INCLUDING 1025TH) 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

1480 BYTES OF ICMP DATA

FRAG3: 20 BYTES OF IP HEADER

1248 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: 1480 BYTES INCLUDES 1472 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):

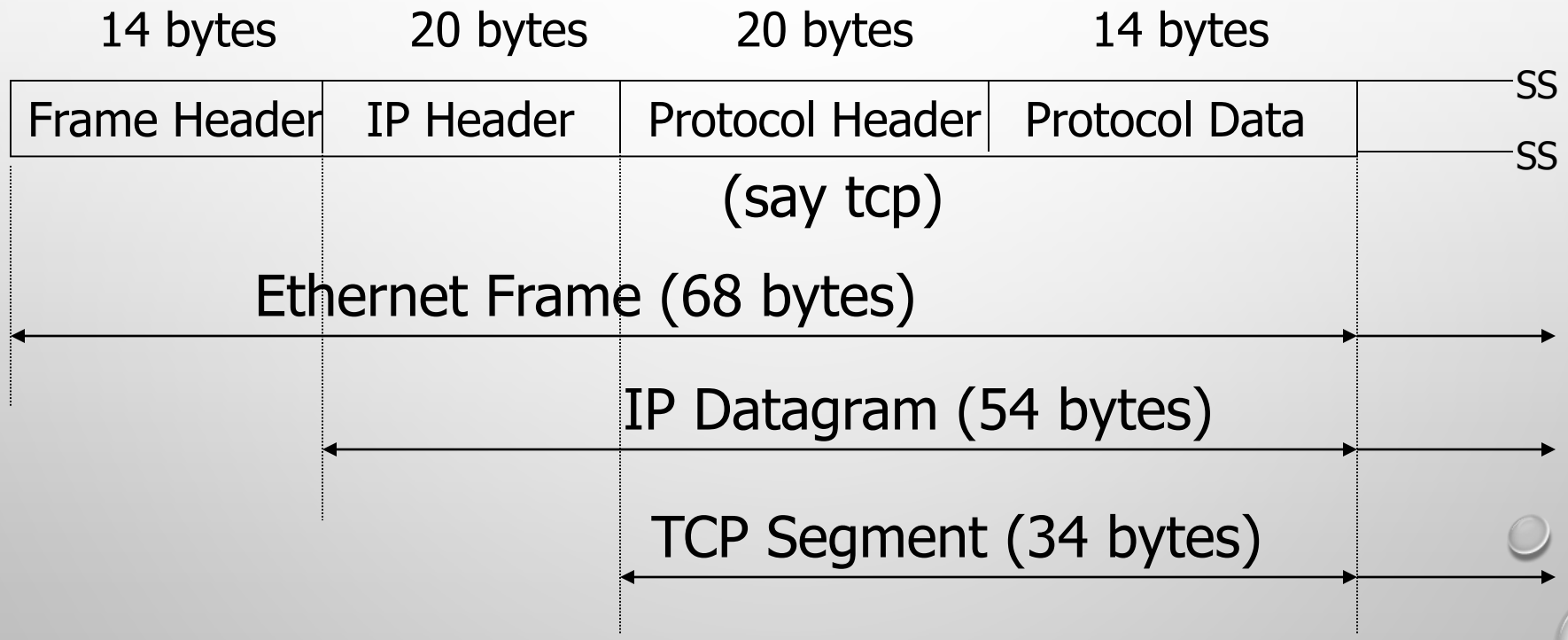
TCPDUMP –S *LENGTH*

- 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 \(HOT OR ACE \)*

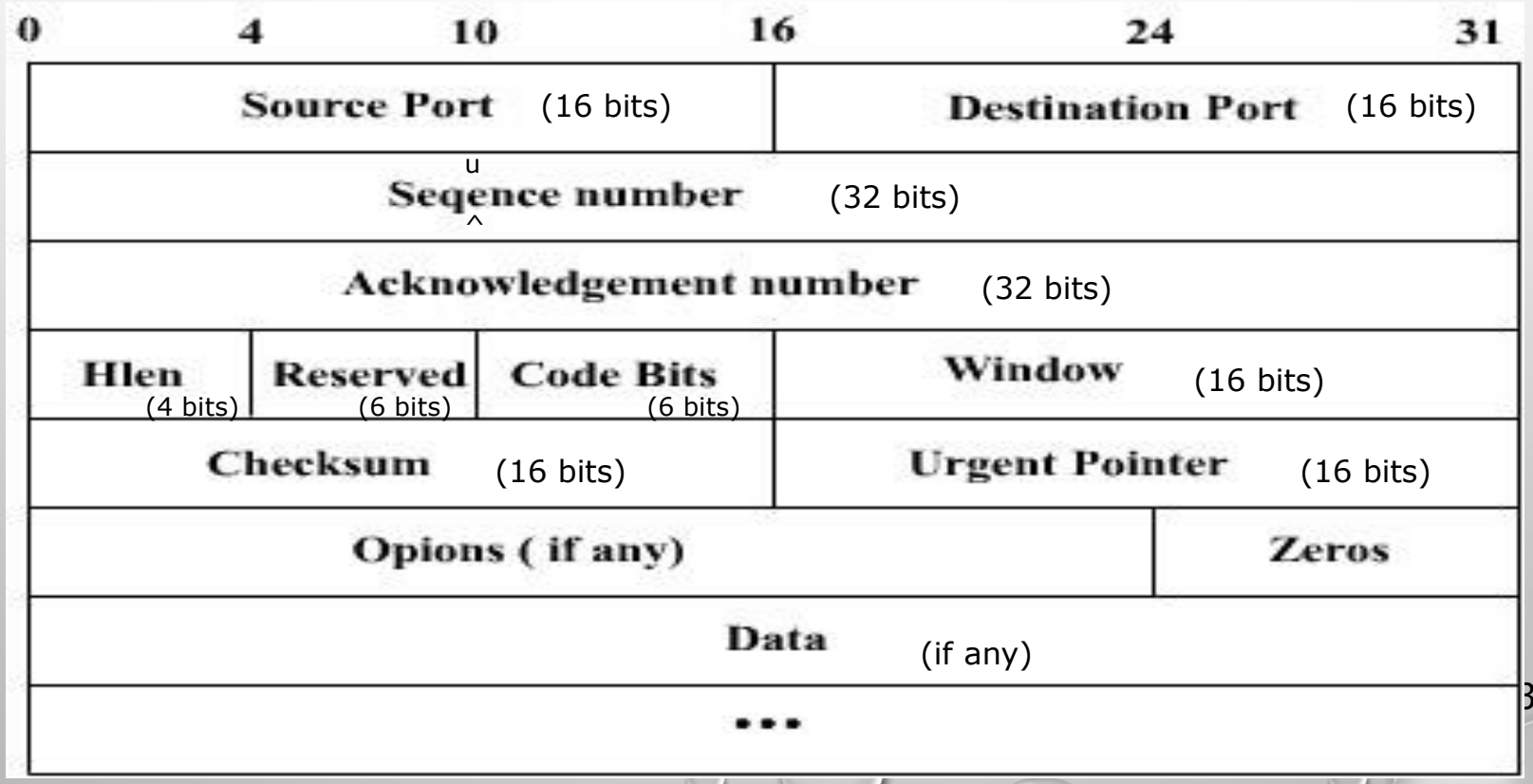
- TO PRINT ALL IP PACKETS BETWEEN *ACE* AND ANY HOST EXCEPT *HELIOS*:

  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 ; *Ref: ECN: RFC 3168*
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 13TH 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 13TH 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 13TH OCTET OF TCP SEGMENT (WITH THE FIRST OCTET BEING THE 0TH OCTET)

MORE EXAMPLES

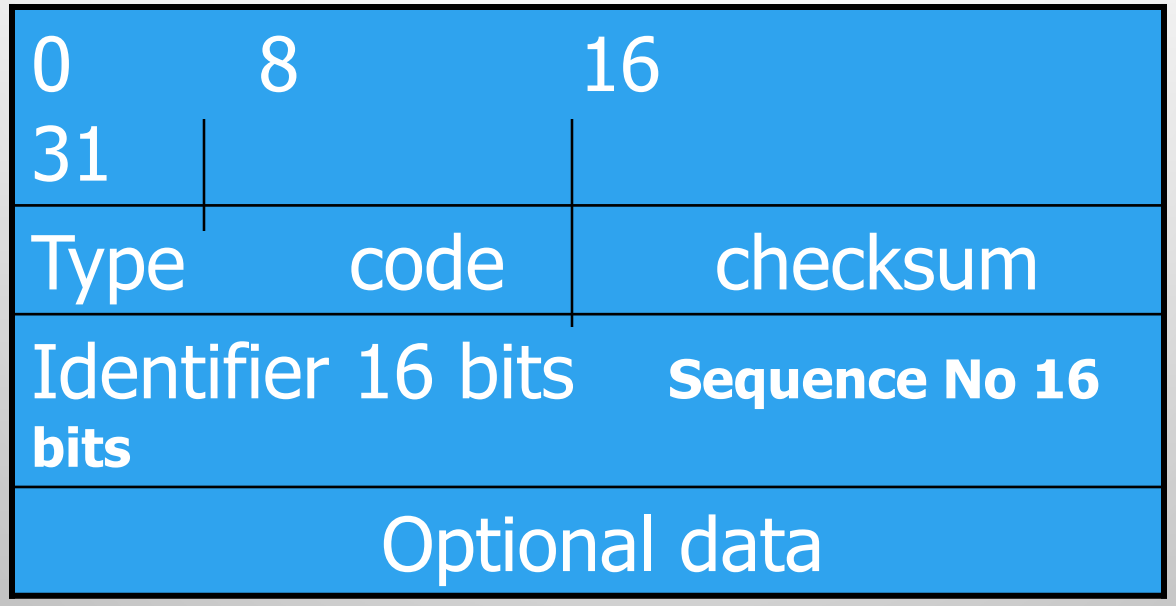
- `**IP[0] & 0XF != 5**' CATCHES ALL IP PACKETS WITH OPTIONS.
- `**IP[6:2] & 0X1FFF = 0**' CATCHES ONLY UNFRAGMENTED DATAGRAMS AND FRAG ZERO OF FRAGMENTED DATAGRAMS.

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

WHAT IS WIRESHARK?

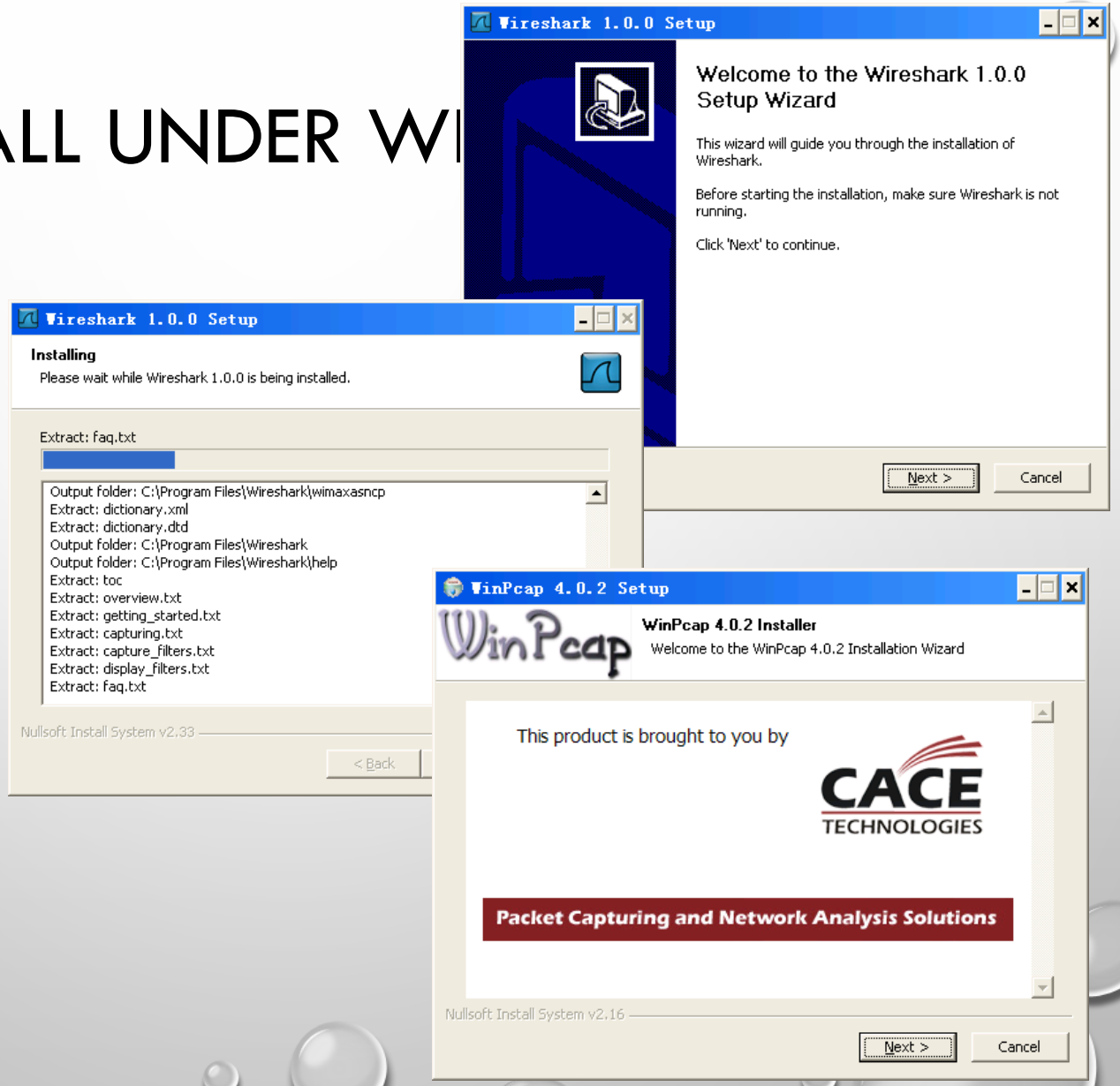
- WIRESHARK IS A NETWORK PACKET/PROTOCOL ANALYZER.
  - A NETWORK PACKET ANALYZER WILL TRY TO CAPTURE NETWORK PACKETS AND TRIES TO DISPLAY THAT PACKET DATA AS DETAILED AS POSSIBLE.
- WIRESHARK IS PERHAPS ONE OF THE BEST OPEN SOURCE PACKET ANALYZERS AVAILABLE TODAY FOR **UNIX** AND **WINDOWS**.

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.

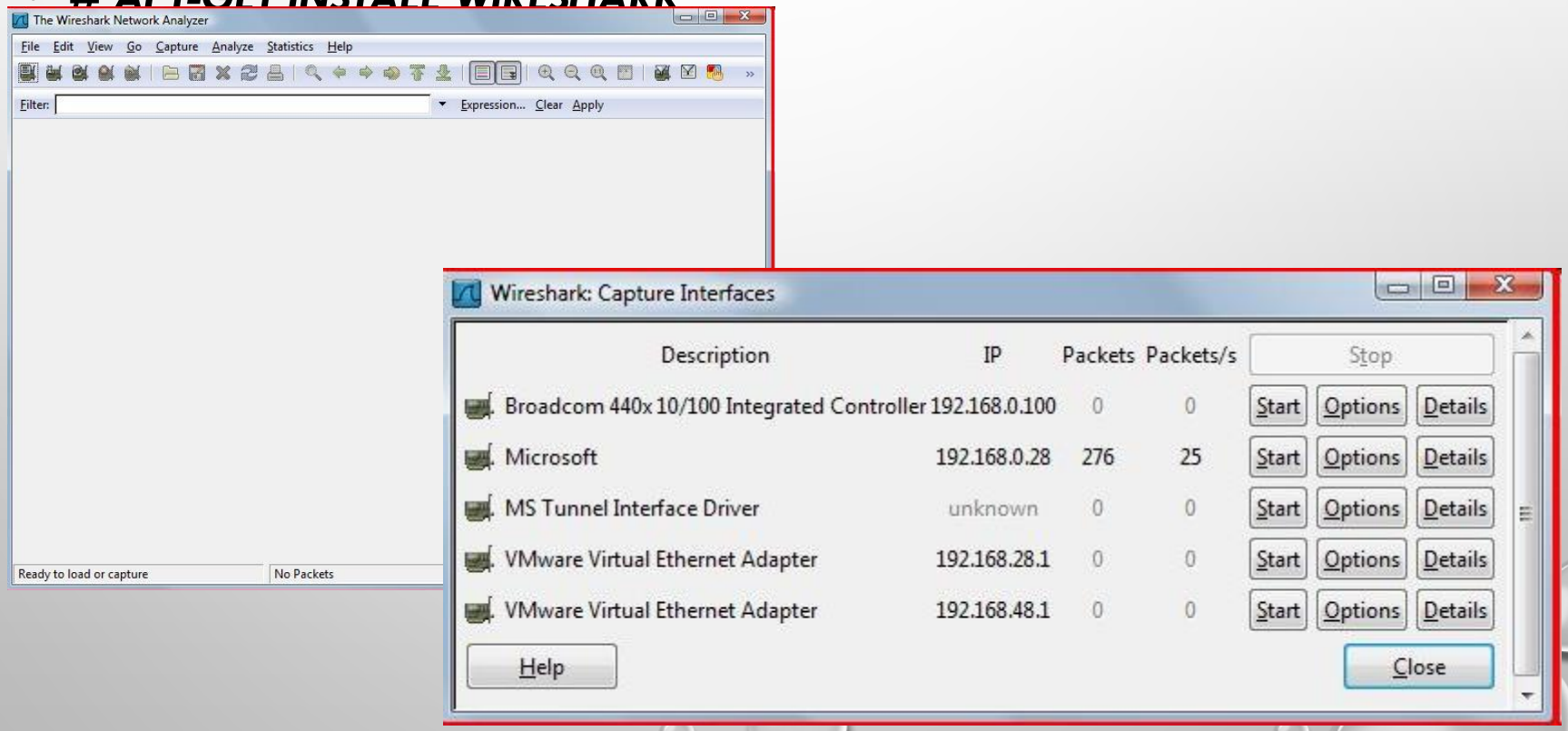
INSTALL UNDER WI

- DOWNLOAD
- INSTALL



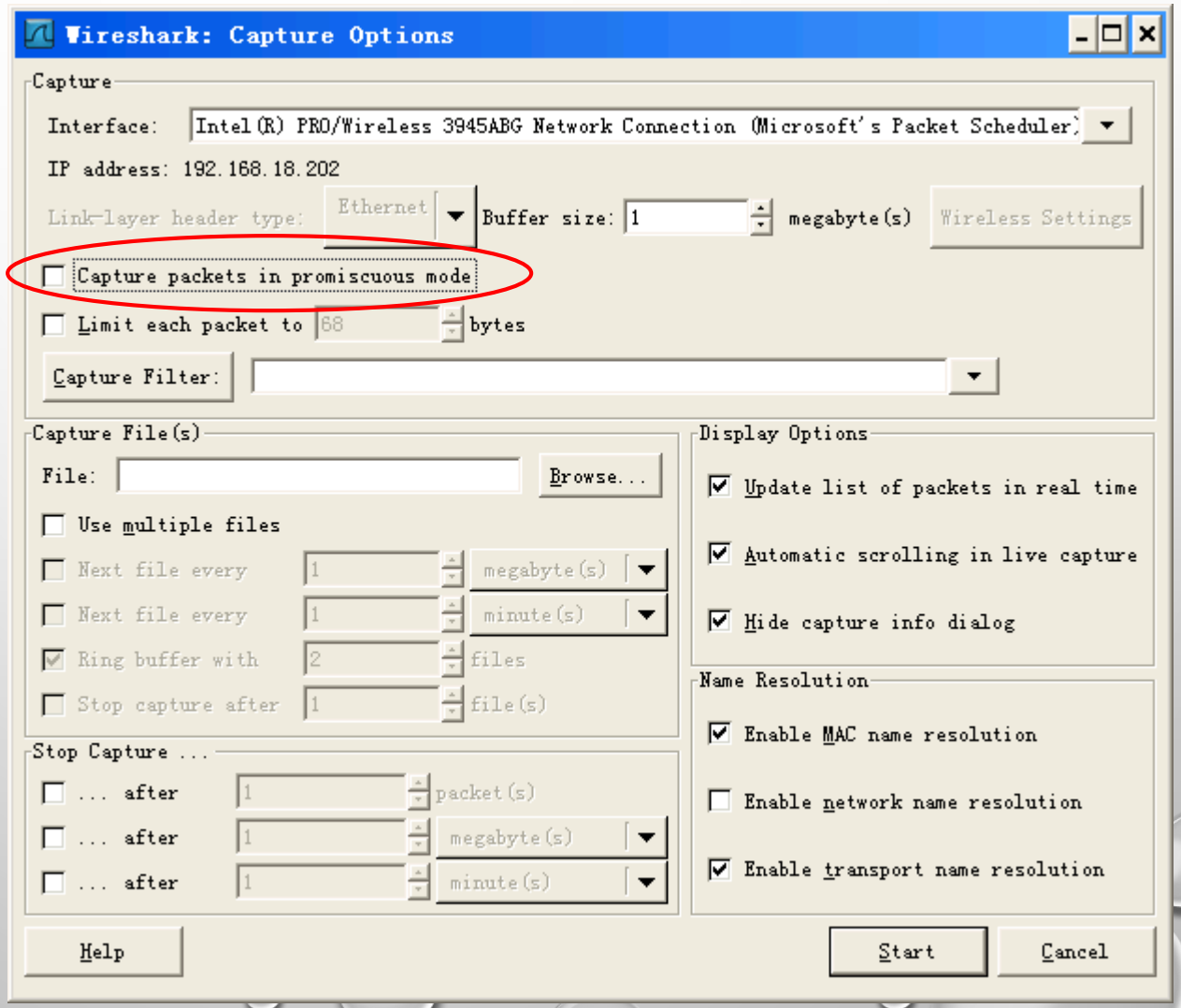
INSTALL UNDER DEBIAN/ UBUNTU

- ***# 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).



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/SAMPLECAPTURES
- SEARCH “WIRESHARK TUTORIAL”

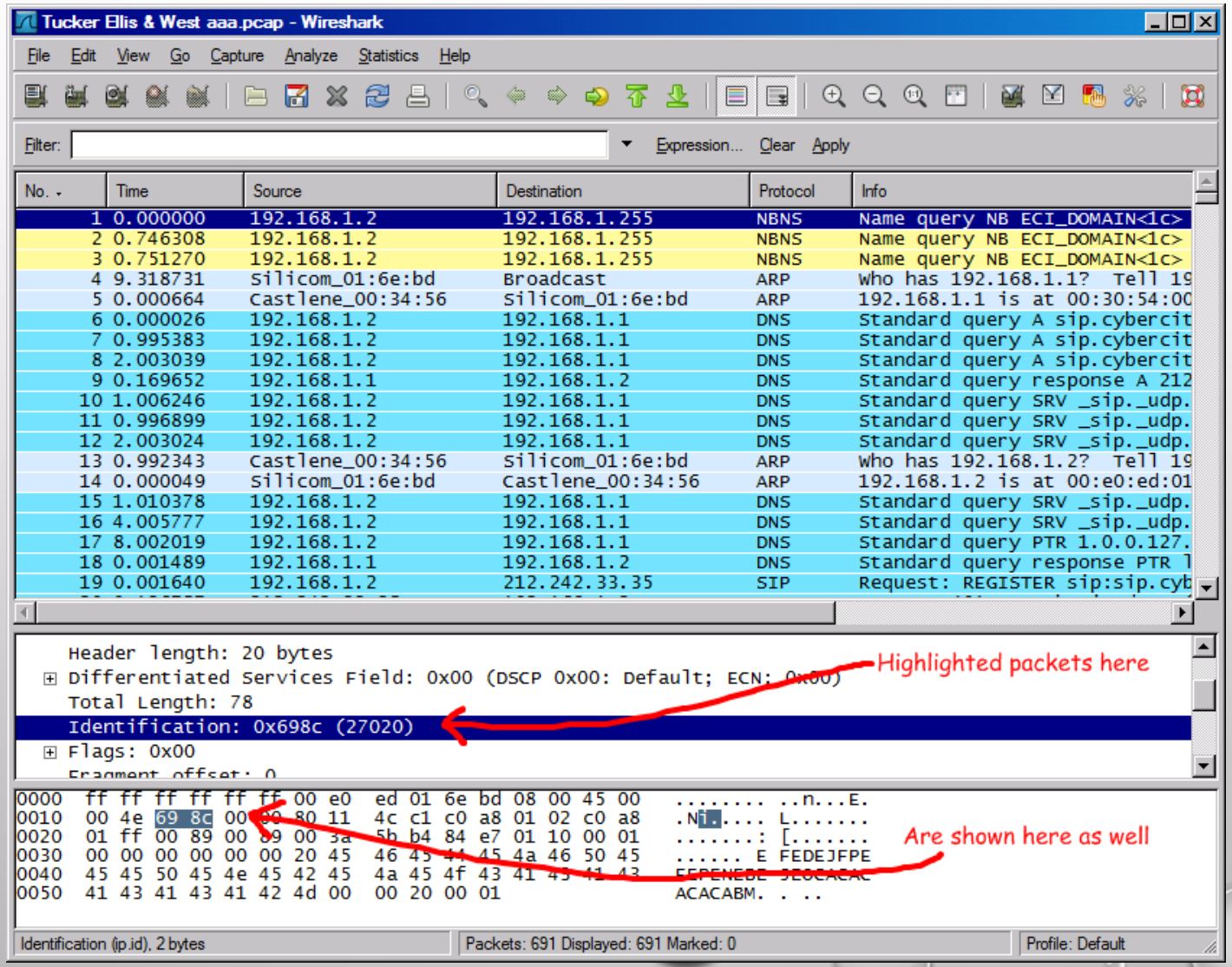
SO WHAT IS WIRESHARK?

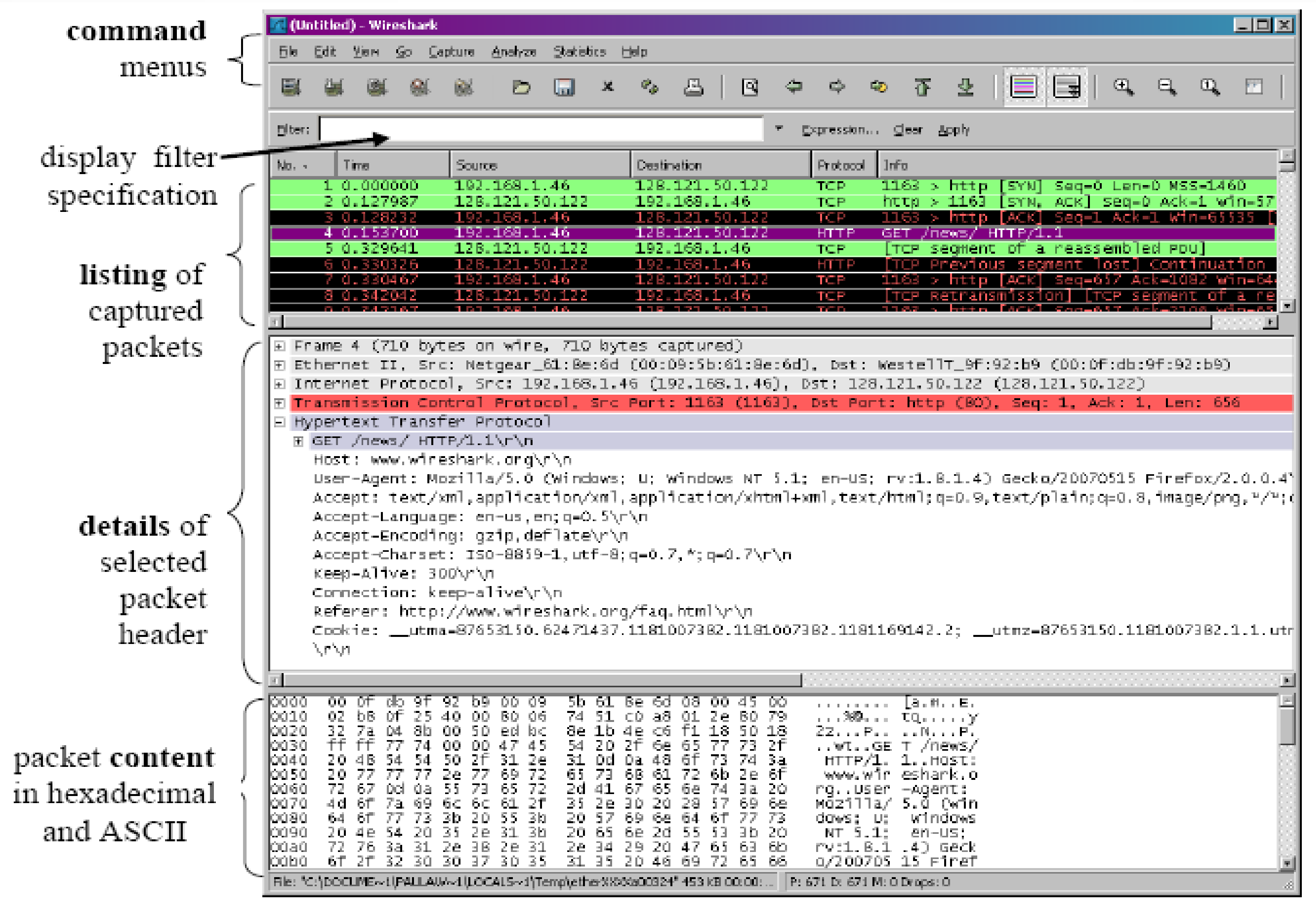
- PACKET SNIFFER/PROTOCOL ANALYZER
- OPEN SOURCE NETWORK TOOL
- LATEST VERSION OF THE ETHEREAL TOOL

WHAT IS TSHARK?

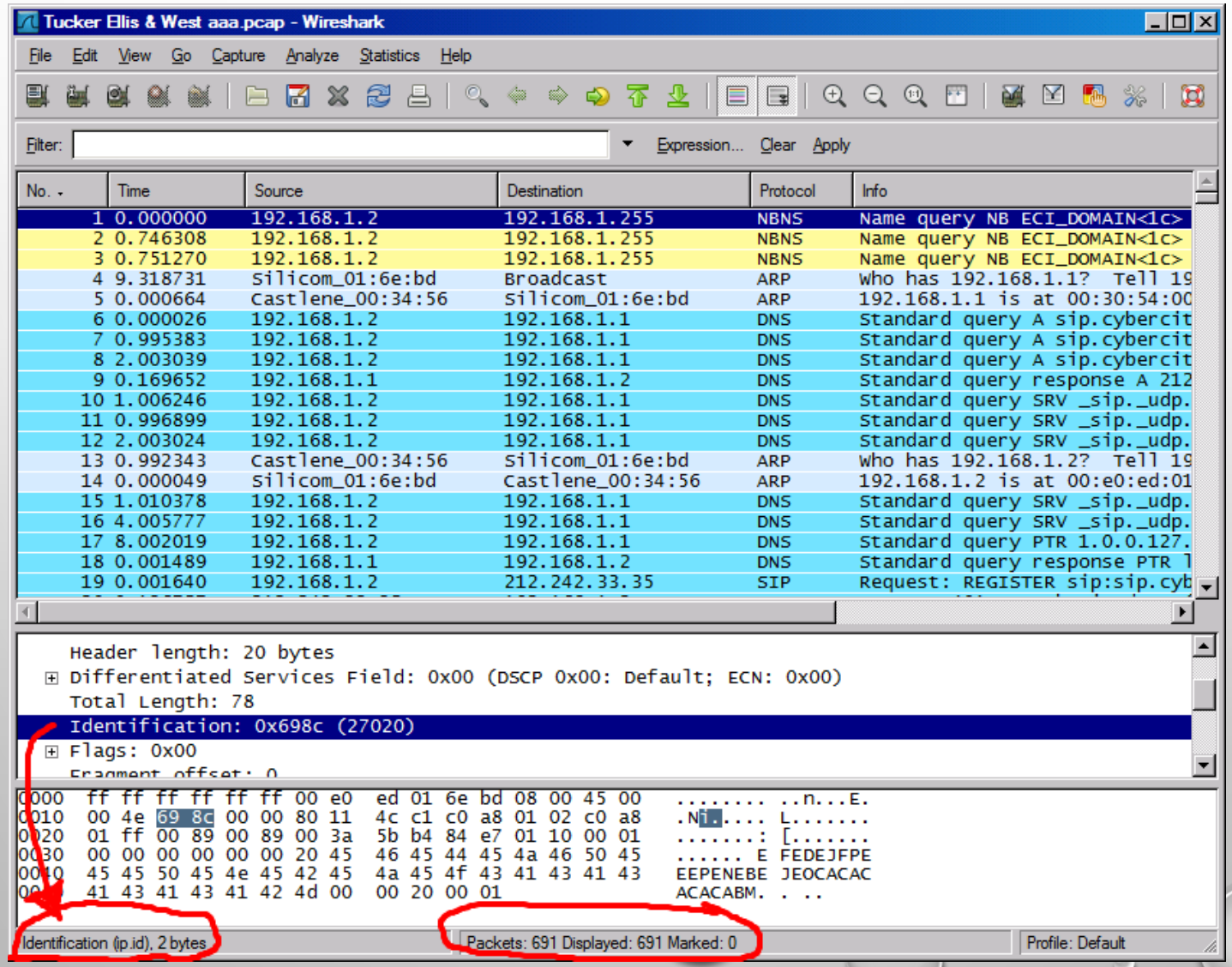
- THE COMMAND-LINE BASED PACKET CAPTURE TOOL
- EQUIVALENT TO WIRESHARK

WIRESHARK INTERFACE

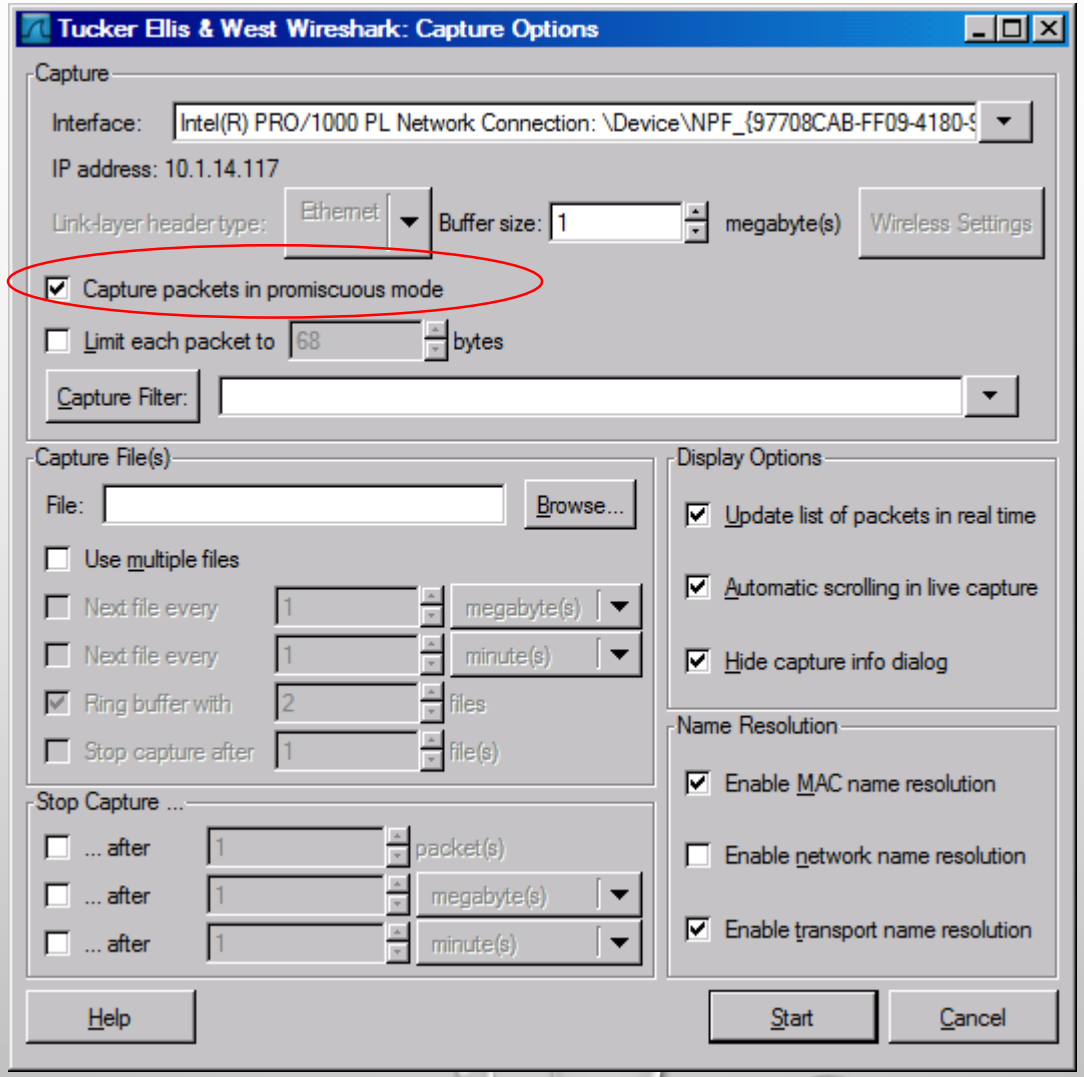



command menus
display filter specification
listing of captured packets
details of selected packet header
packet content in hexadecimal and ASCII
(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
Frame 4 (710 bytes on wire, 710 bytes captured)
Hypertext Transfer Protocol
GET /news/ HTTP/1.1\r\n
Host: www.wireshark.org\r\n

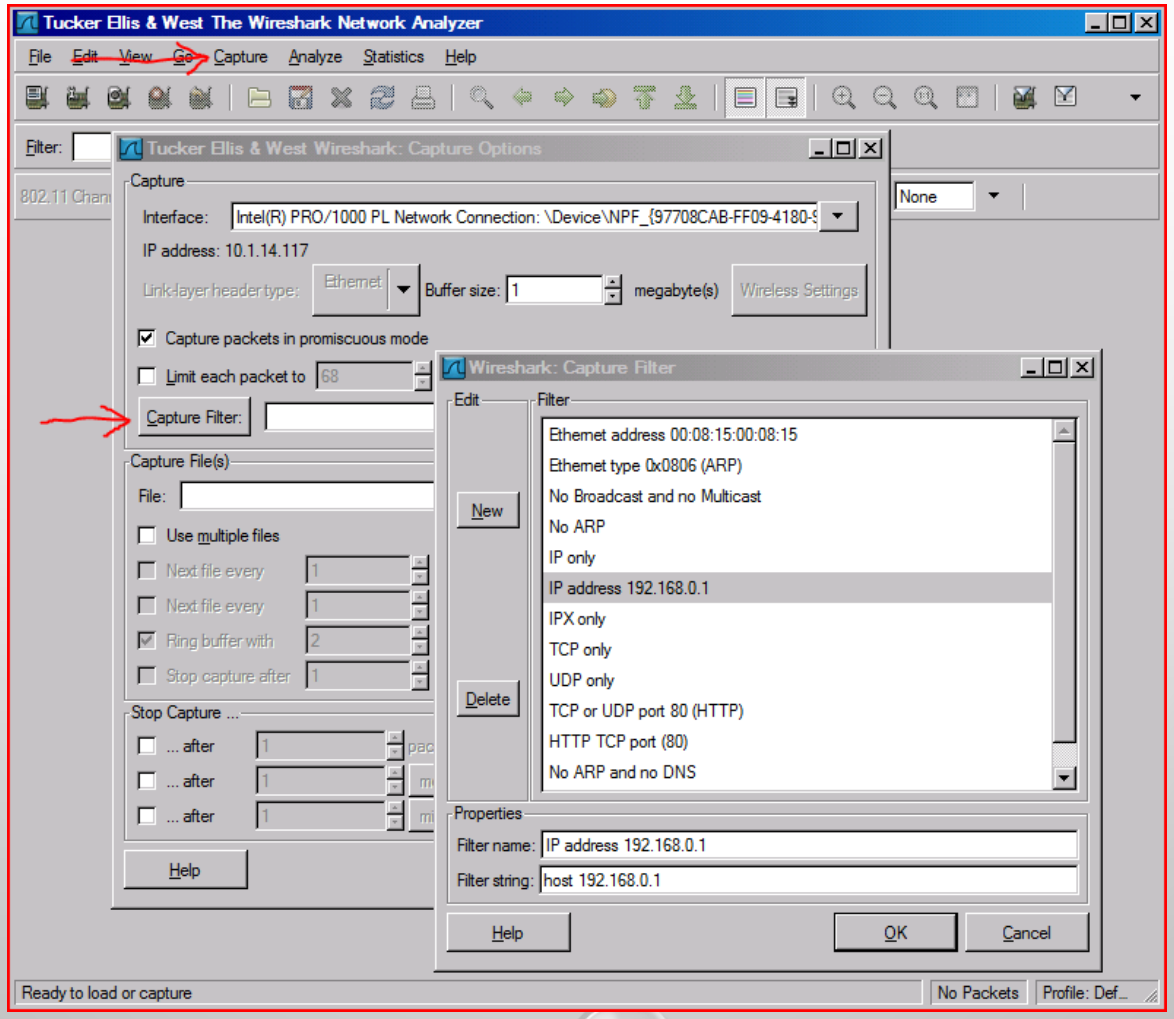
STATUS BAR


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.cyb
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 78
Identification: 0x698c (27020)
Flags: 0x00
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 .Ni..... 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 EEPENEBE JEOCACAC
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



CAPTURE FILTER



CAPTURE FILTER EXAMPLES

HOST 10.1.11.24

HOST 192.168.0.1 AND HOST 10.1.11.1

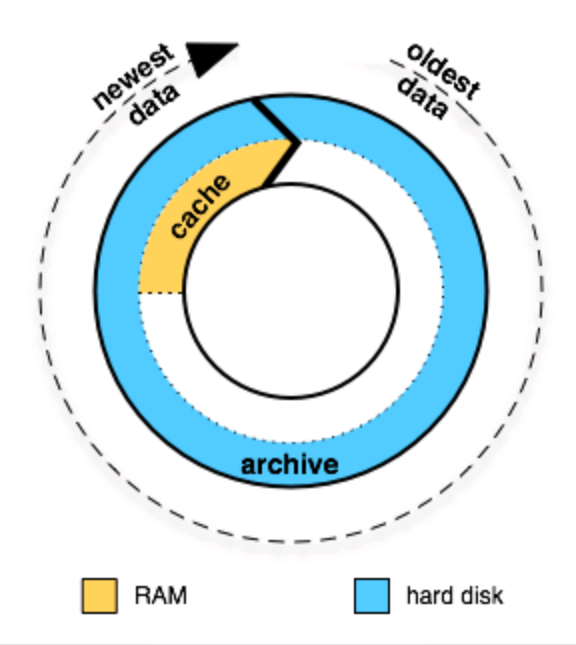
TCP PORT HTTP

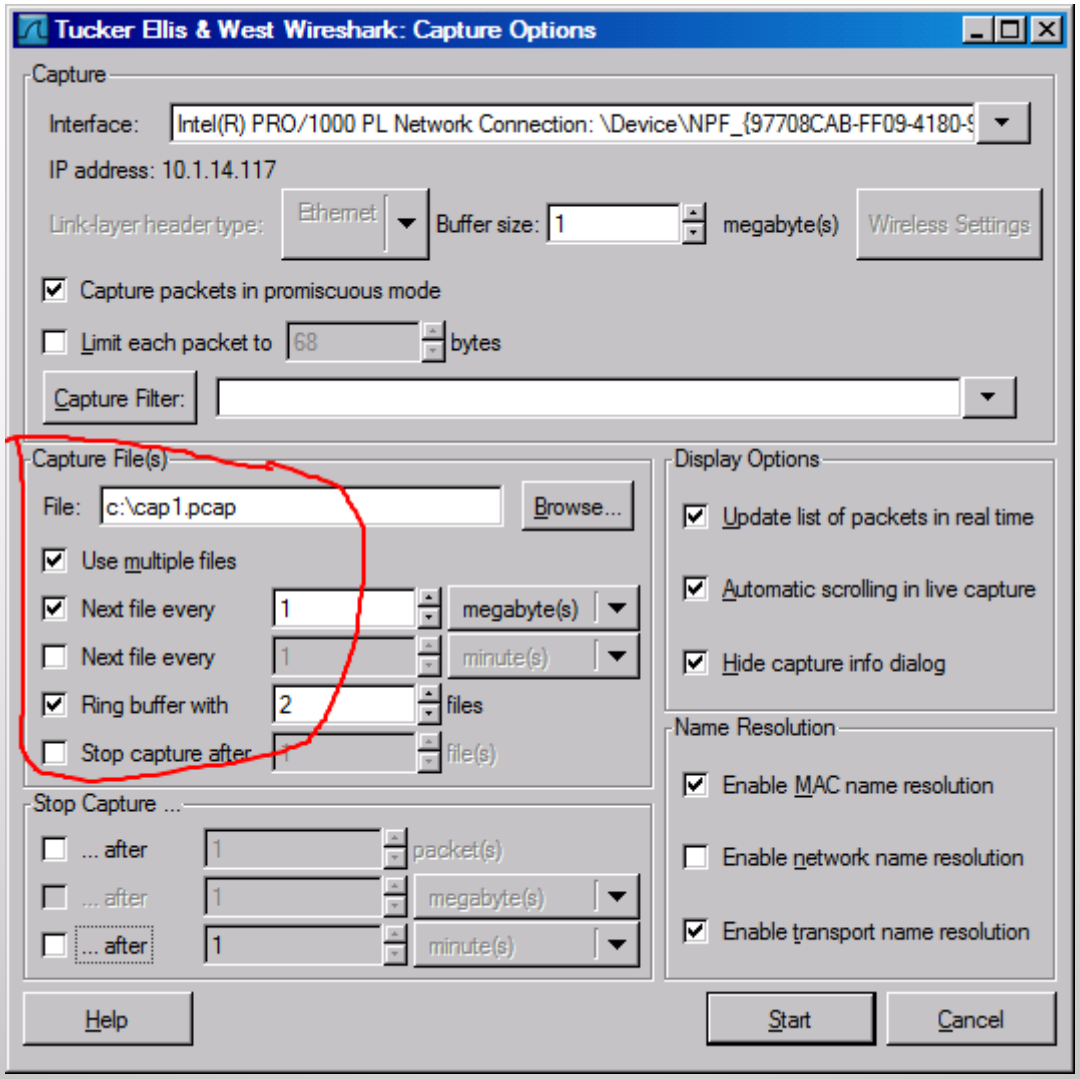
IP

NOT BROADCAST NOT MULTICAST

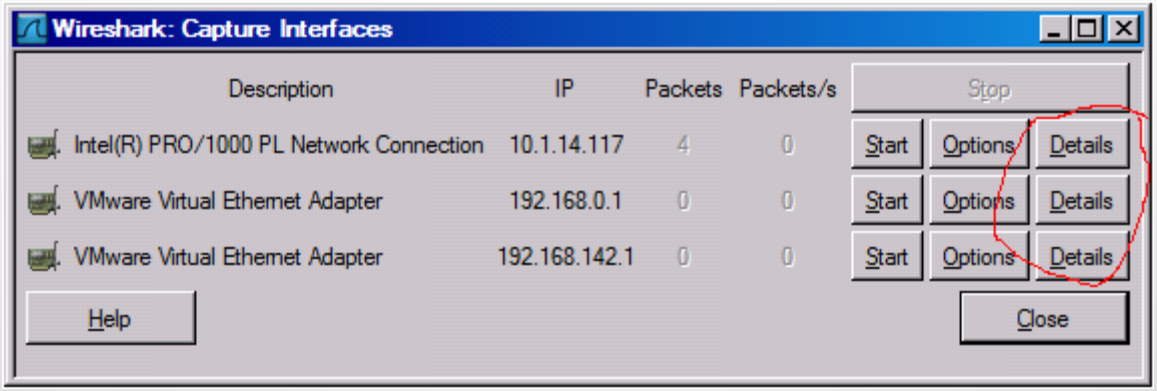
ETHER HOST 00:04:13:00:09:A3

CAPTURE BUFFER USAGE

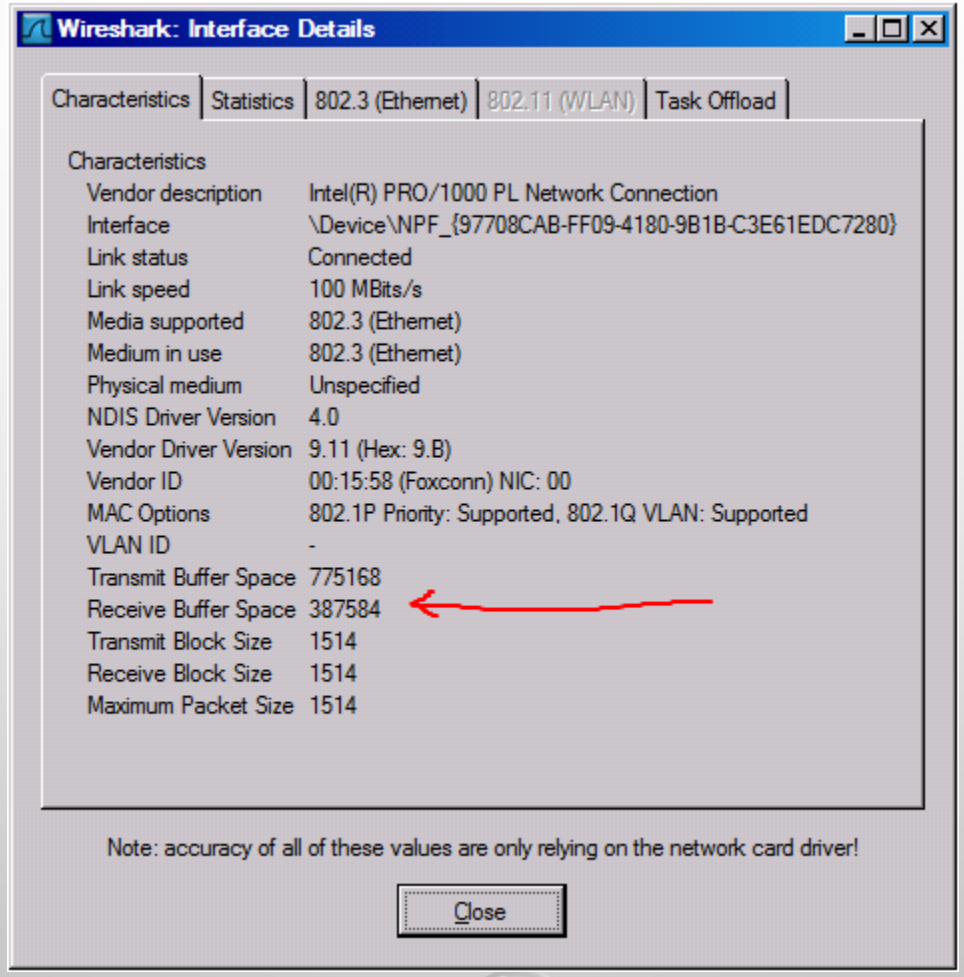



Tucker Ellis & West Wireshark: Capture Options
Capture
Interface:
Intel(R) PRO/1000 PL Network Connection: \Device\NPF_{97708CAB-FF09-4180-9
IP address: 10.1.14.117
Link-layer header type:
Ethernet
Buffer size:
1
megabyte(s)
Wireless Settings
Capture packets in promiscuous mode
Limit each packet to
68
bytes
Capture Filter:
Capture File(s)
File:
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)
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
Stop Capture ...
... after
1
packet(s)
... after
1
megabyte(s)
... after
1
minute(s)
Help
Start
Cancel

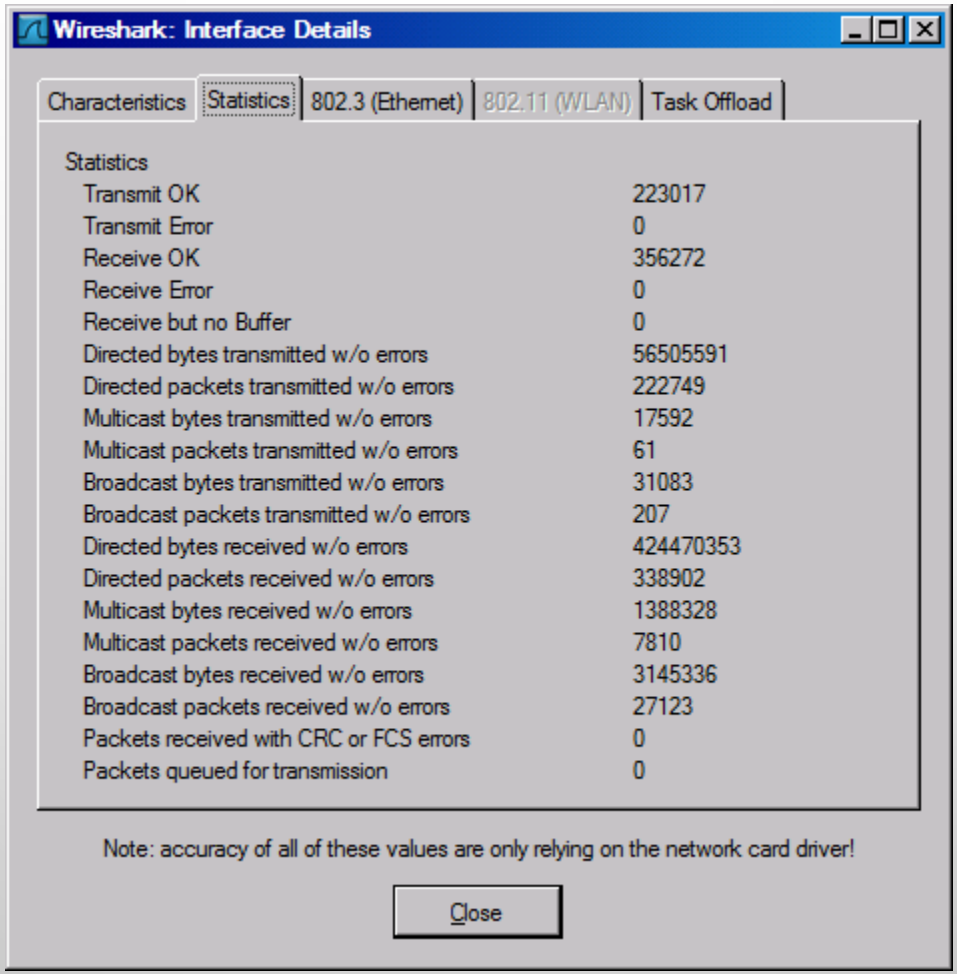
CAPTURE INTERFACES



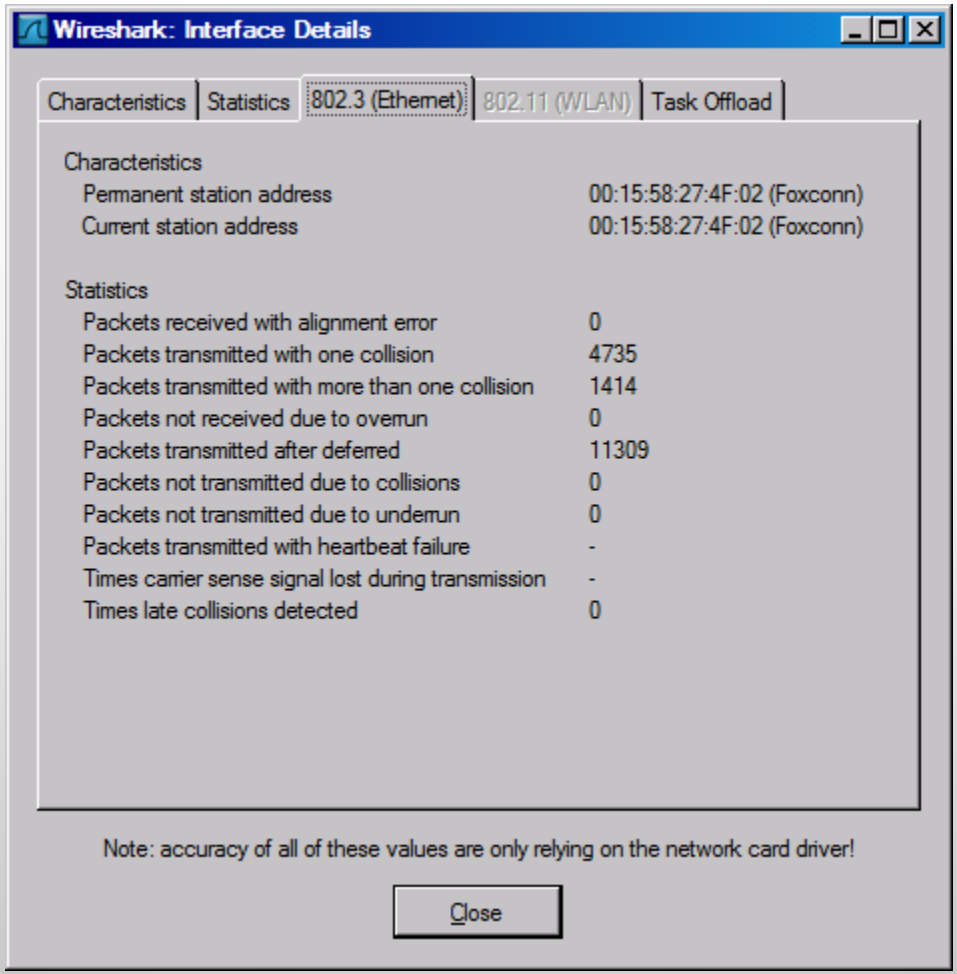
INTERFACE DETAILS: CHARACTERISTICS



INTERFACE DETAILS: STATISTICS



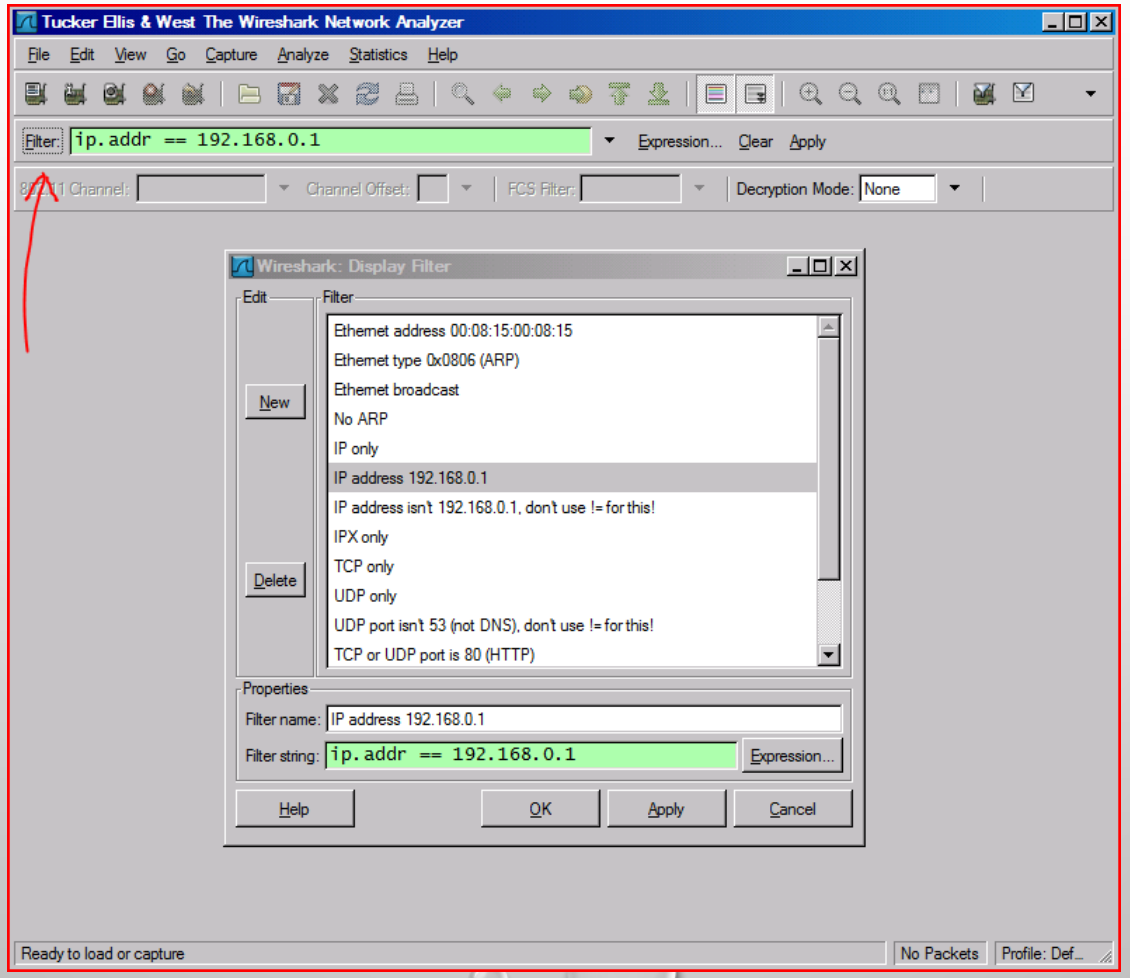
INTERFACE DETAILS: 802.3



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)

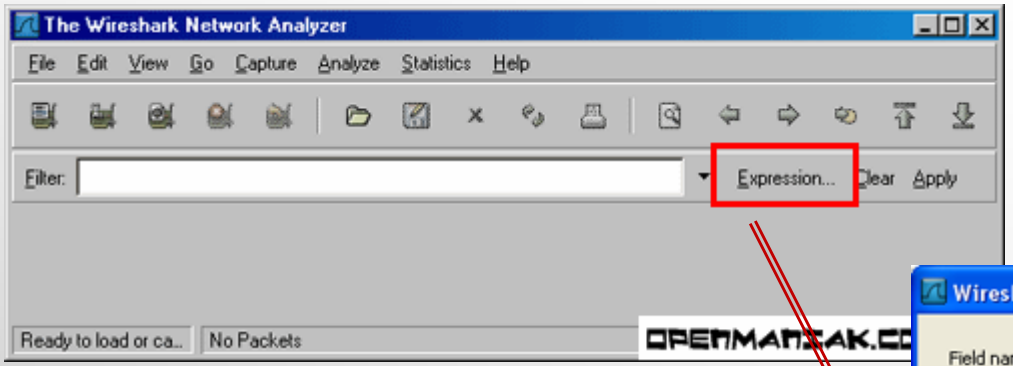
(IP.ADDR==192.168.1.10 && IP.ADDR==192.168.1.20) && (TCP.PORT==445 || TCP.PORT==139

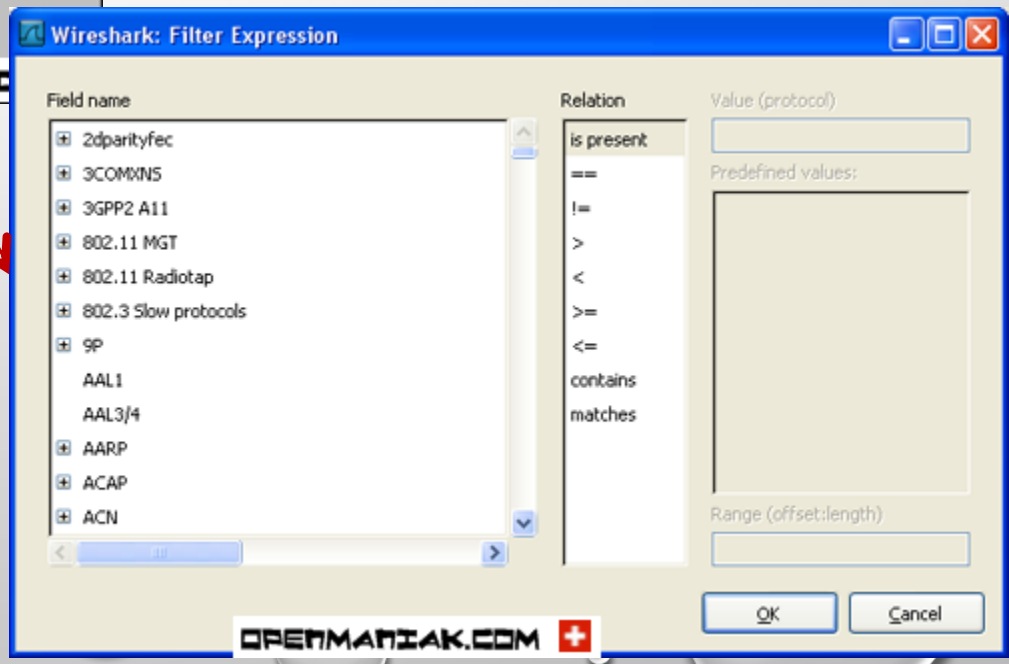
(IP.ADDR==192.168.1.10 && IP.ADDR==192.168.1.20) && (UDP.PORT==67 || UDP.PORT==68)

TCP.DSTPORT == 80

DISPLAY FILTER

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

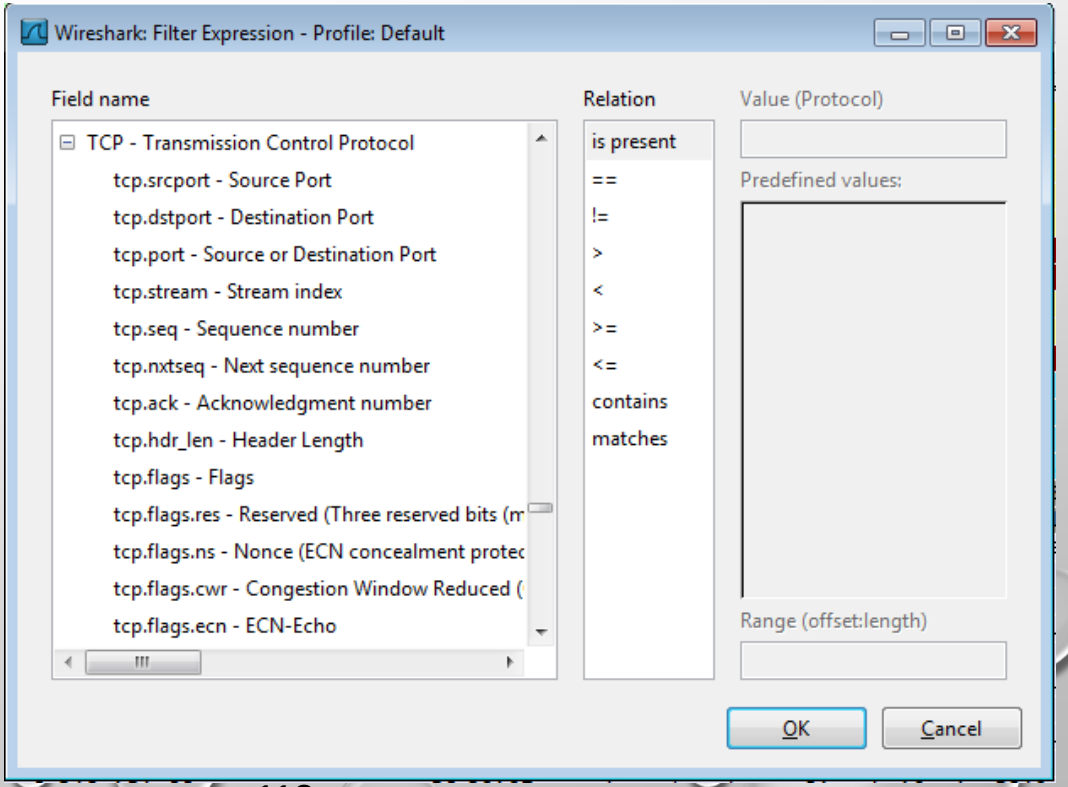




DISPLAY FILTER

- 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.

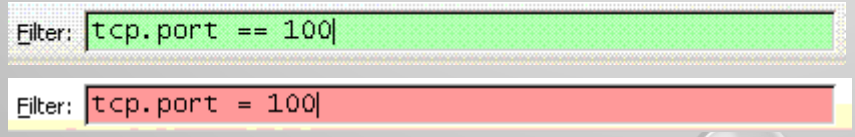
Six comparison operators are available:

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.



Correct syntax

Wrong syntax

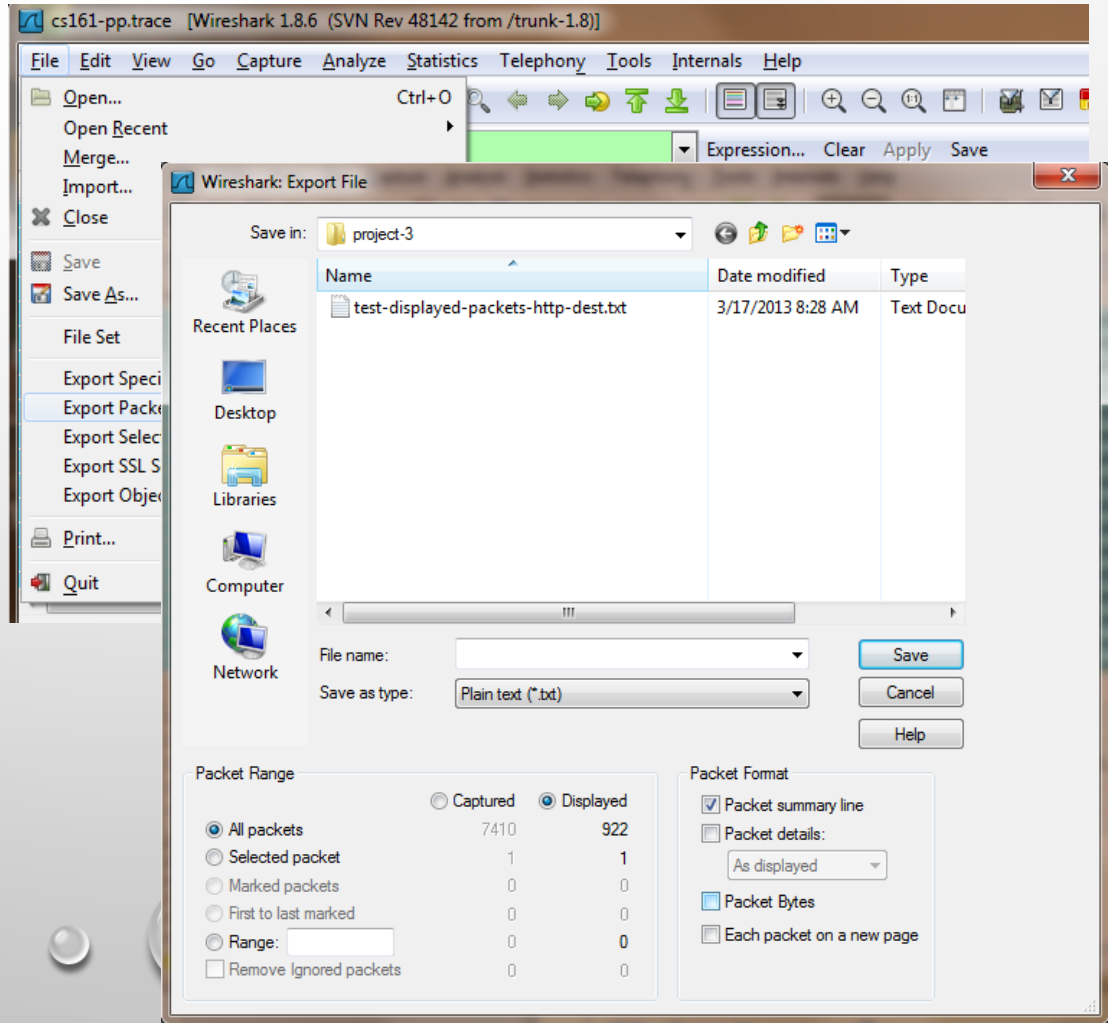
SAVE FILTERED PACKETS AFTER USING DISPLAY FILTER

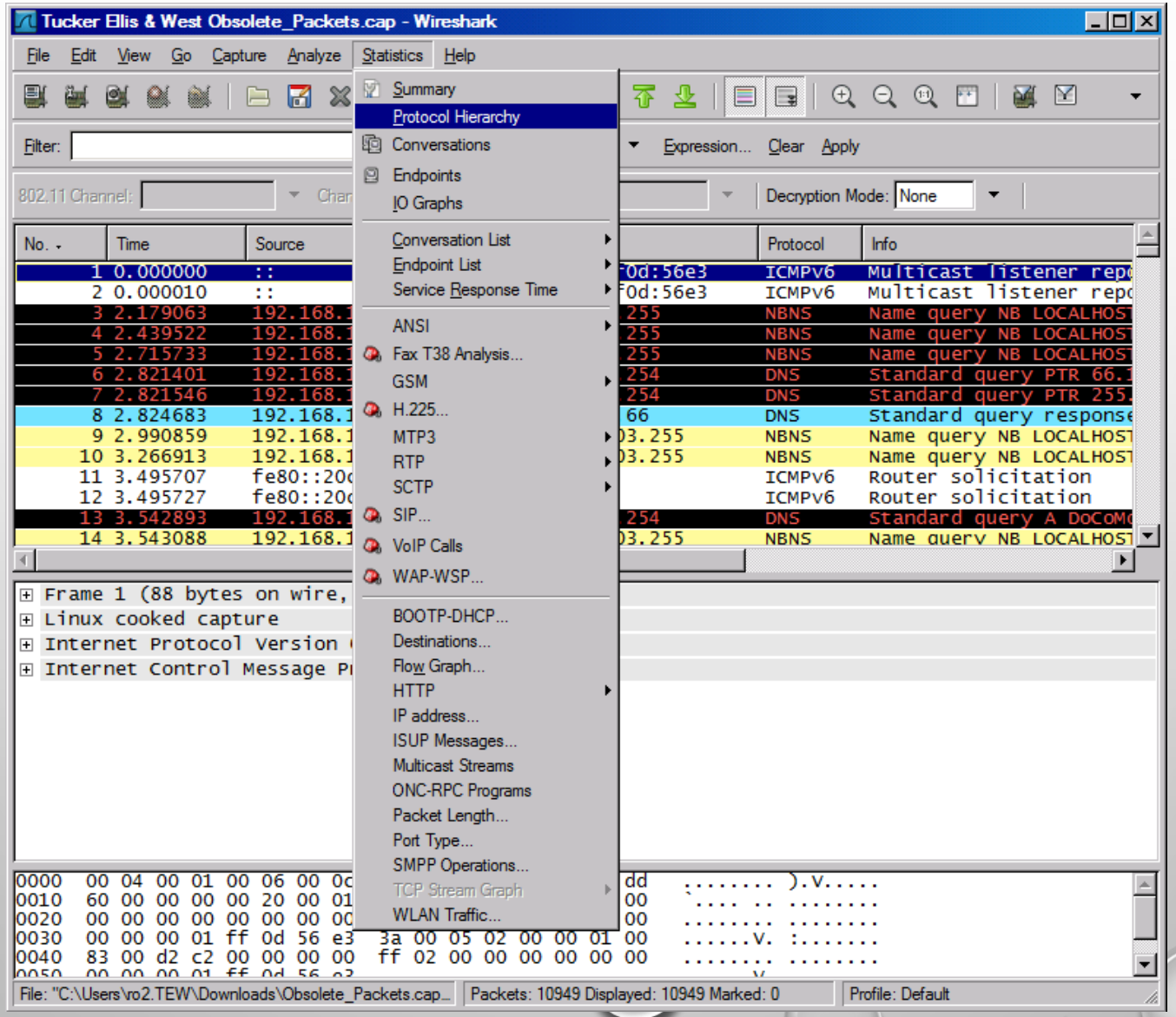
- WE CAN ALSO SAVE ALL FILTERED PACKETS IN TEXT FILE FOR FURTHER ANALYSIS
- OPERATION:

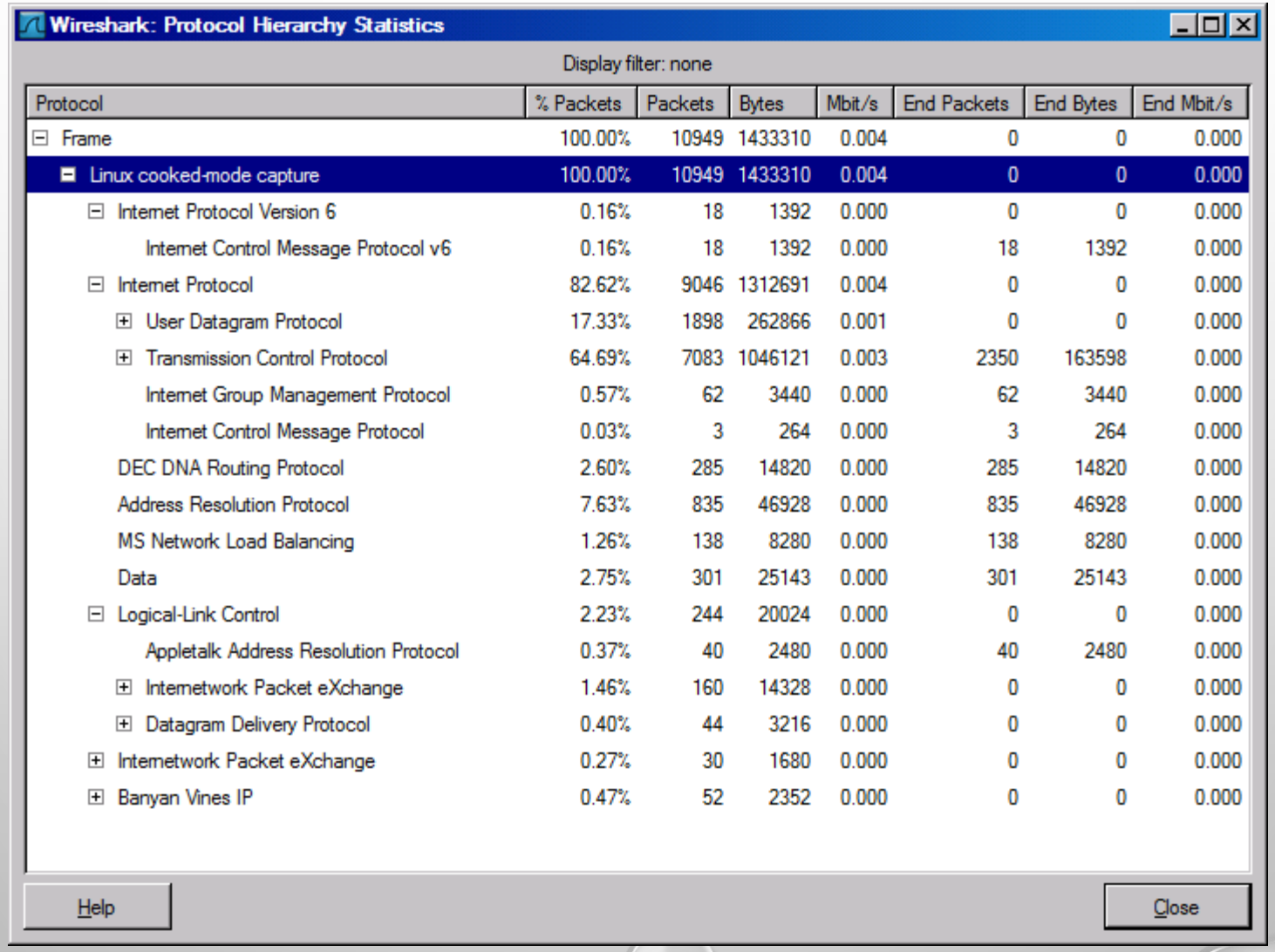
File→Export packet dissections →as “plain text” file

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

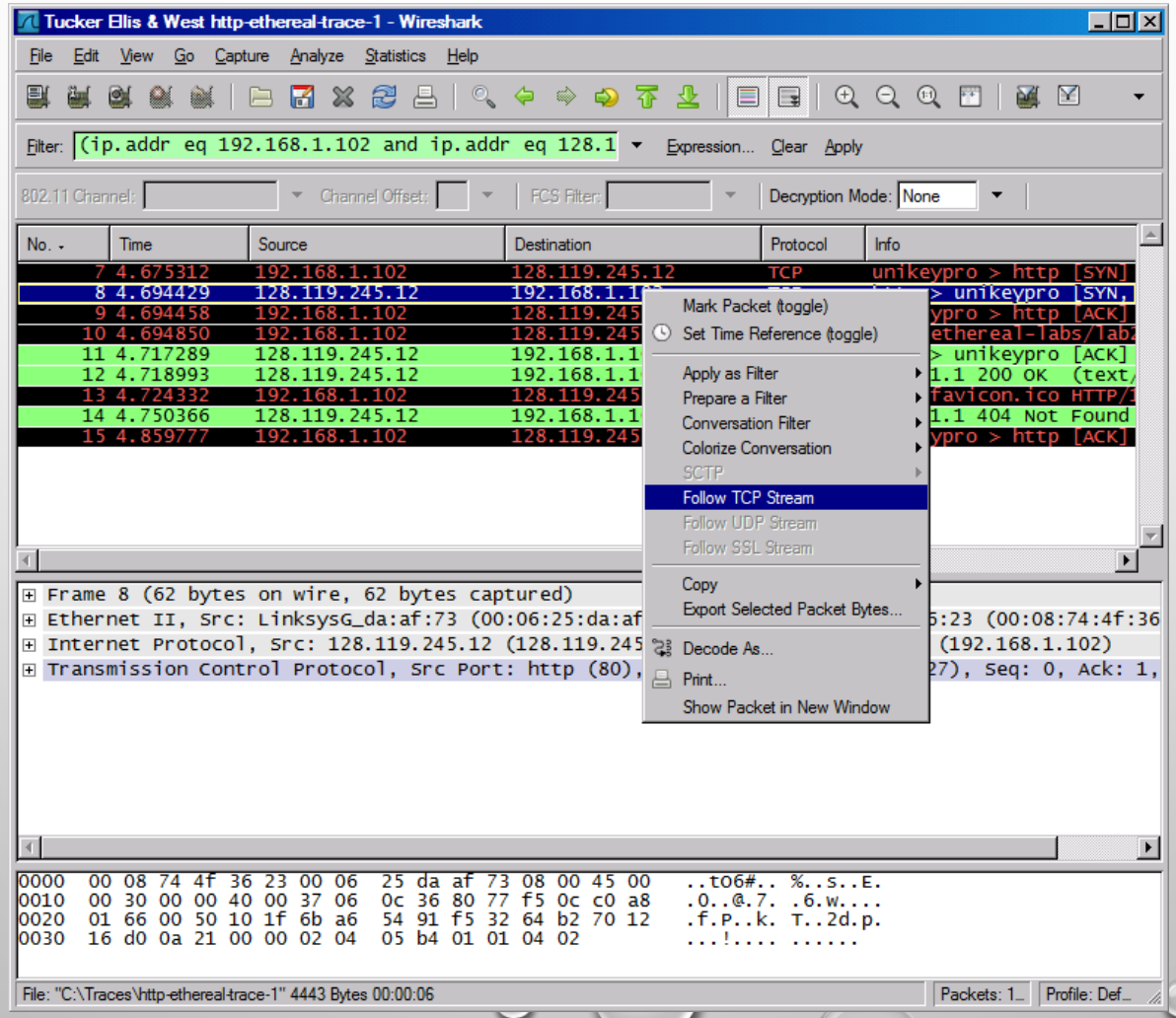
2). In choose “summary line” or “detail”




Tucker Ellis & West Obsolete_Packets.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Summary
Protocol Hierarchy
Conversations
Endpoints
IO Graphs
Conversation List
Endpoint List
Service Response Time
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...
Filter:
Expression... Clear Apply
802.11 Channel:
Decryption Mode: None
No. Time Source Protocol Info
1 0.000000 :: ICMPv6 Multicast listener rep
2 0.000010 :: ICMPv6 Multicast listener rep
3 2.179063 192.168.1 NBNS Name query NB LOCALHOST
4 2.439522 192.168.1 NBNS Name query NB LOCALHOST
5 2.715733 192.168.1 NBNS Name query NB LOCALHOST
6 2.821401 192.168.1 DNS Standard query PTR 66.1
7 2.821546 192.168.1 DNS Standard query PTR 255.
8 2.824683 192.168.1 DNS Standard query response
9 2.990859 192.168.1 NBNS Name query NB LOCALHOST
10 3.266913 192.168.1 NBNS Name query NB LOCALHOST
11 3.495707 fe80::20c ICMPv6 Router solicitation
12 3.495727 fe80::20c ICMPv6 Router solicitation
13 3.542893 192.168.1 DNS Standard query A DoCoMo
14 3.543088 192.168.1 NBNS Name query NB LOCALHOST
Frame 1 (88 bytes on wire,
Linux cooked capture
Internet Protocol Version
Internet Control Message Pr
0000 00 04 00 01 00 06 00 0c
0010 60 00 00 00 00 20 00 01
0020 00 00 00 00 00 00 00 00
0030 00 00 00 01 ff 0d 56 e3 3a 00 05 02 00 00 01 00 ......V. :.......
0040 83 00 d2 c2 00 00 00 00 ff 02 00 00 00 00 00 00 ....
File: "C:\Users\ro2.TEW\Downloads\Obsolete_Packets.cap... Packets: 10949 Displayed: 10949 Marked: 0 Profile: Default

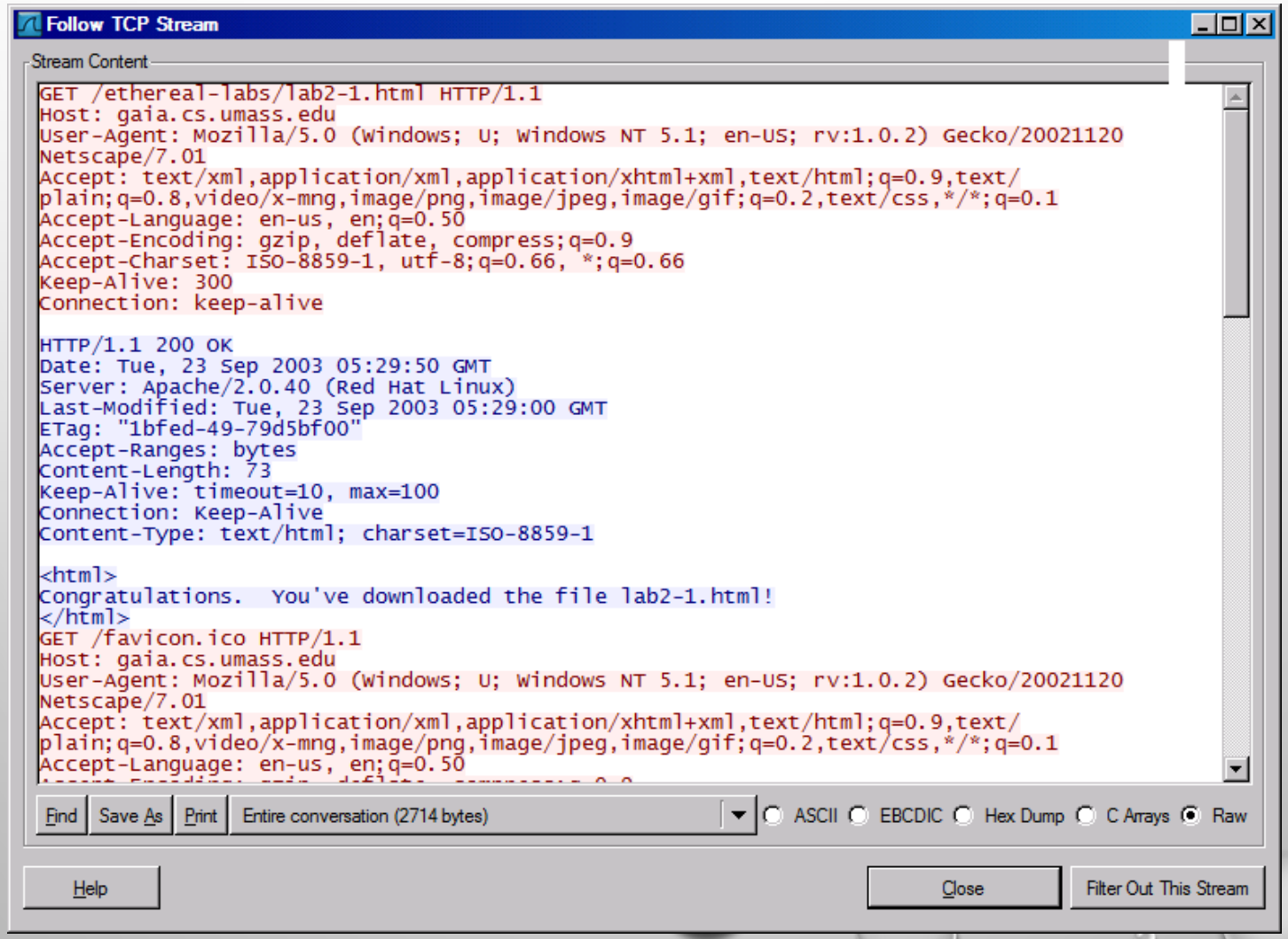

Wireshark: Protocol Hierarchy Statistics
Display filter: none
Protocol % Packets Packets Bytes Mbit/s End Packets End Bytes End Mbit/s
Frame 100.00% 10949 1433310 0.004 0 0 0.000
Linux cooked-mode capture 100.00% 10949 1433310 0.004 0 0 0.000
Internet Protocol Version 6 0.16% 18 1392 0.000 0 0 0.000
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



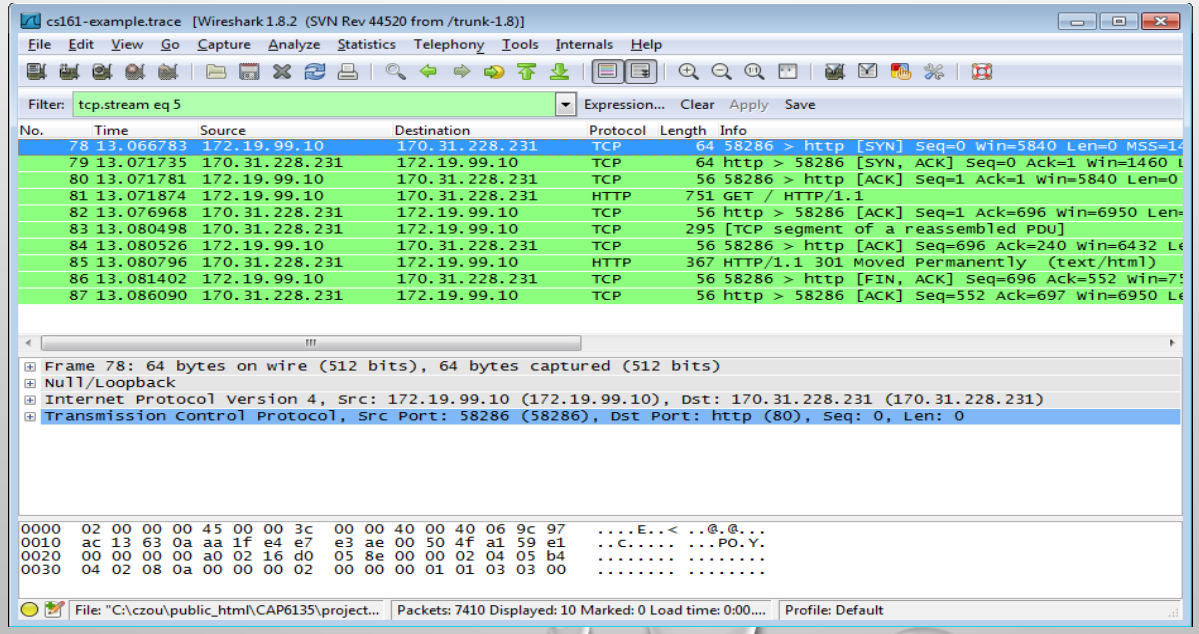
FOLLOW TCP STREAM

red - stuff you sent **blue - stuff you get**

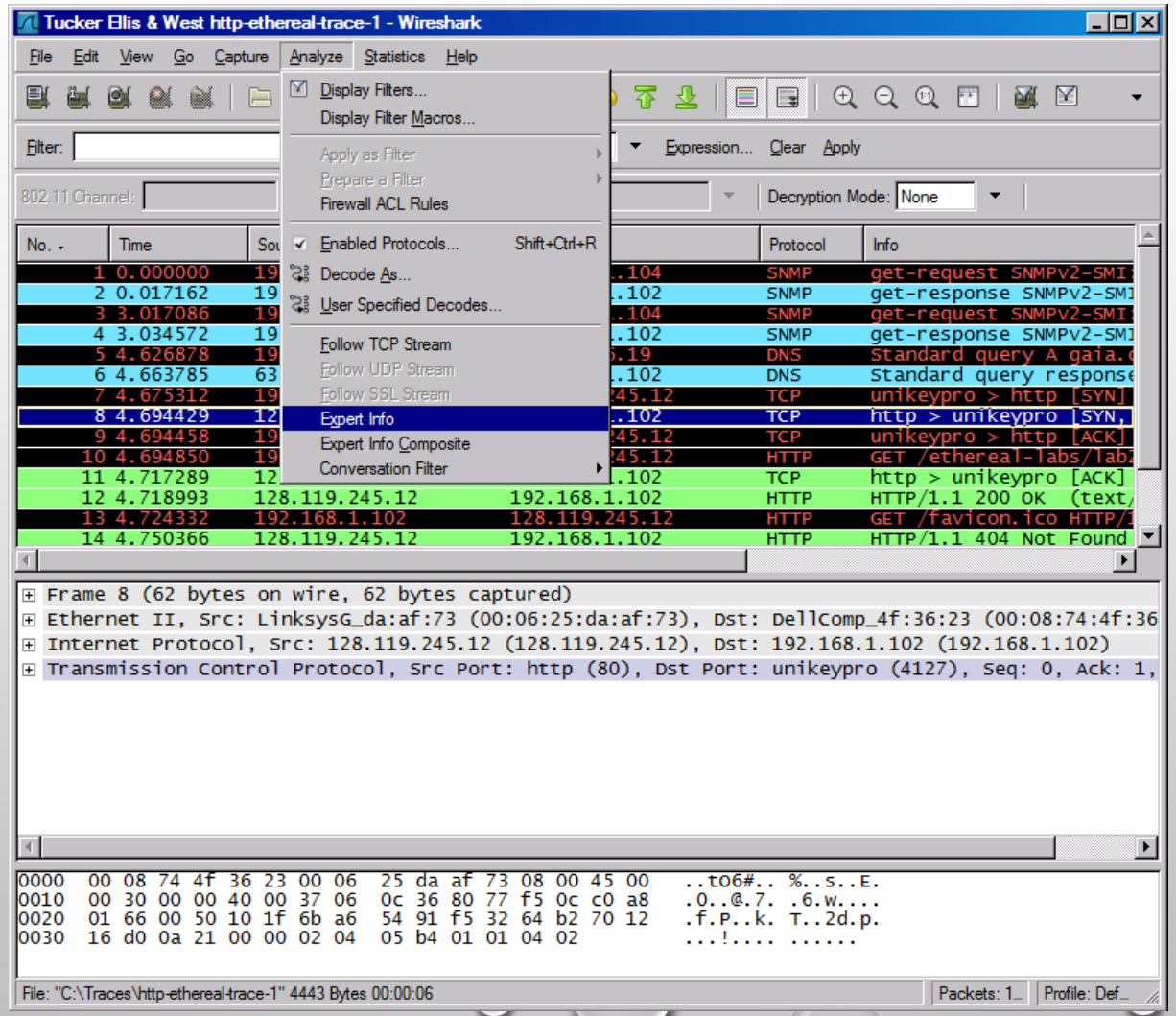


FILTER OUT/IN SINGLE TCP STREAM

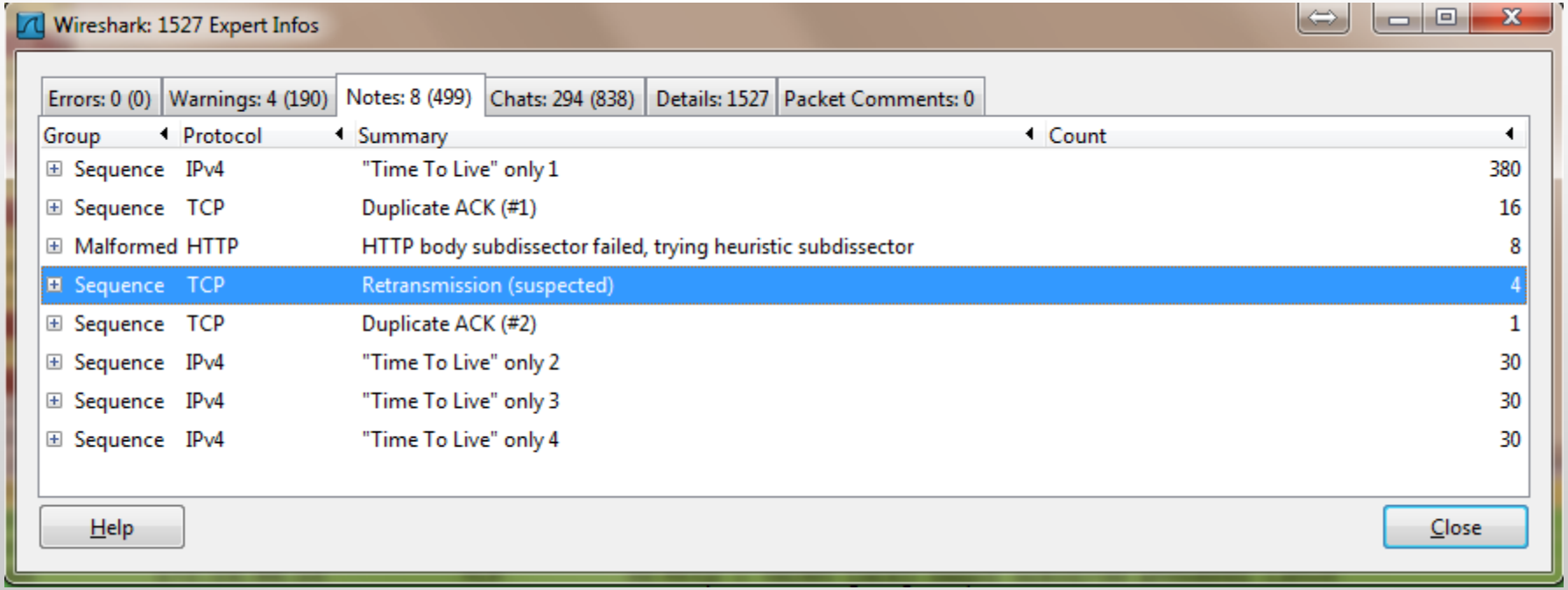
- WHEN CLICK “FILTER OUT THIS TCP STREAM” IN PREVIOUS PAGE’S BOX, NEW FILTER STRING WILL CONTAIN LIKE:
  - HTTP AND !(TCP.STREAM EQ 5)
- SO, IF YOU USE “TCP.STREAM EQ 5” AS FILTER STRING, YOU KEEP THIS HTTP SESSION



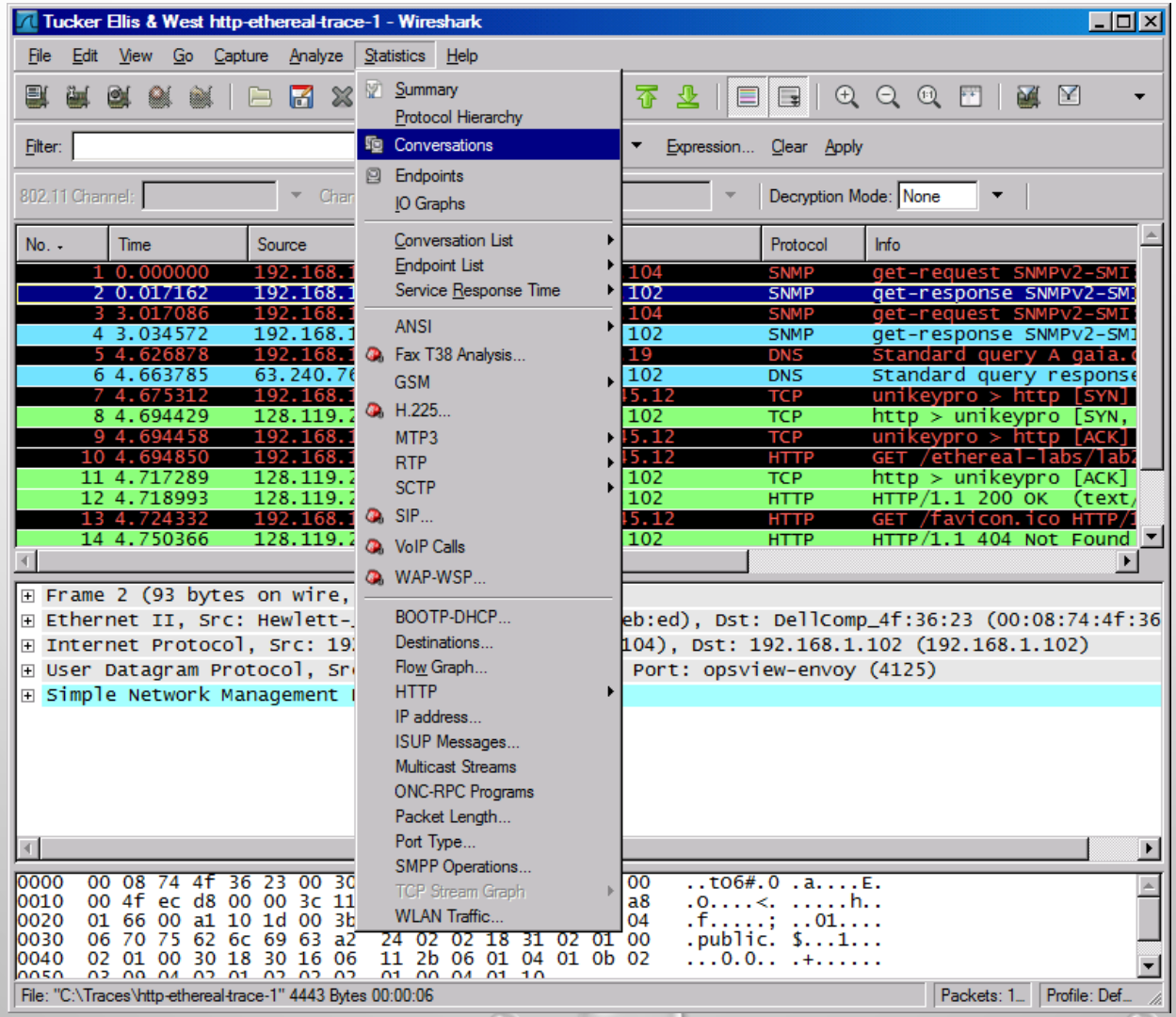
EXPERT INFO


Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Display Filters...
Display Filter Macros...
Apply as Filter
Prepare a Filter
Firewall ACL Rules
Enabled Protocols... Shift+Ctrl+R
Decode As...
User Specified Decodes...
Follow TCP Stream
Follow UDP Stream
Follow SSL Stream
Expert Info
Expert Info Composite
Conversation Filter
Filter:
Expression... Clear Apply
802.11 Channel:
Decryption Mode: None
No. Time Protocol Info
1 0.000000 SNMP get-request SNMPv2-SMI
2 0.017162 SNMP get-response SNMPv2-SMI
3 3.017086 SNMP get-request SNMPv2-SMI
4 3.034572 SNMP get-response SNMPv2-SMI
5 4.626878 DNS Standard query A gaia.
6 4.663785 DNS Standard query response
7 4.675312 TCP unikeypro > http [SYN]
8 4.694429 TCP http > unikeypro [SYN,
9 4.694458 TCP unikeypro > http [ACK]
10 4.694850 HTTP GET /ethereal-labs/lab
11 4.717289 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/1
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
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 ..tO6#.. %..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...

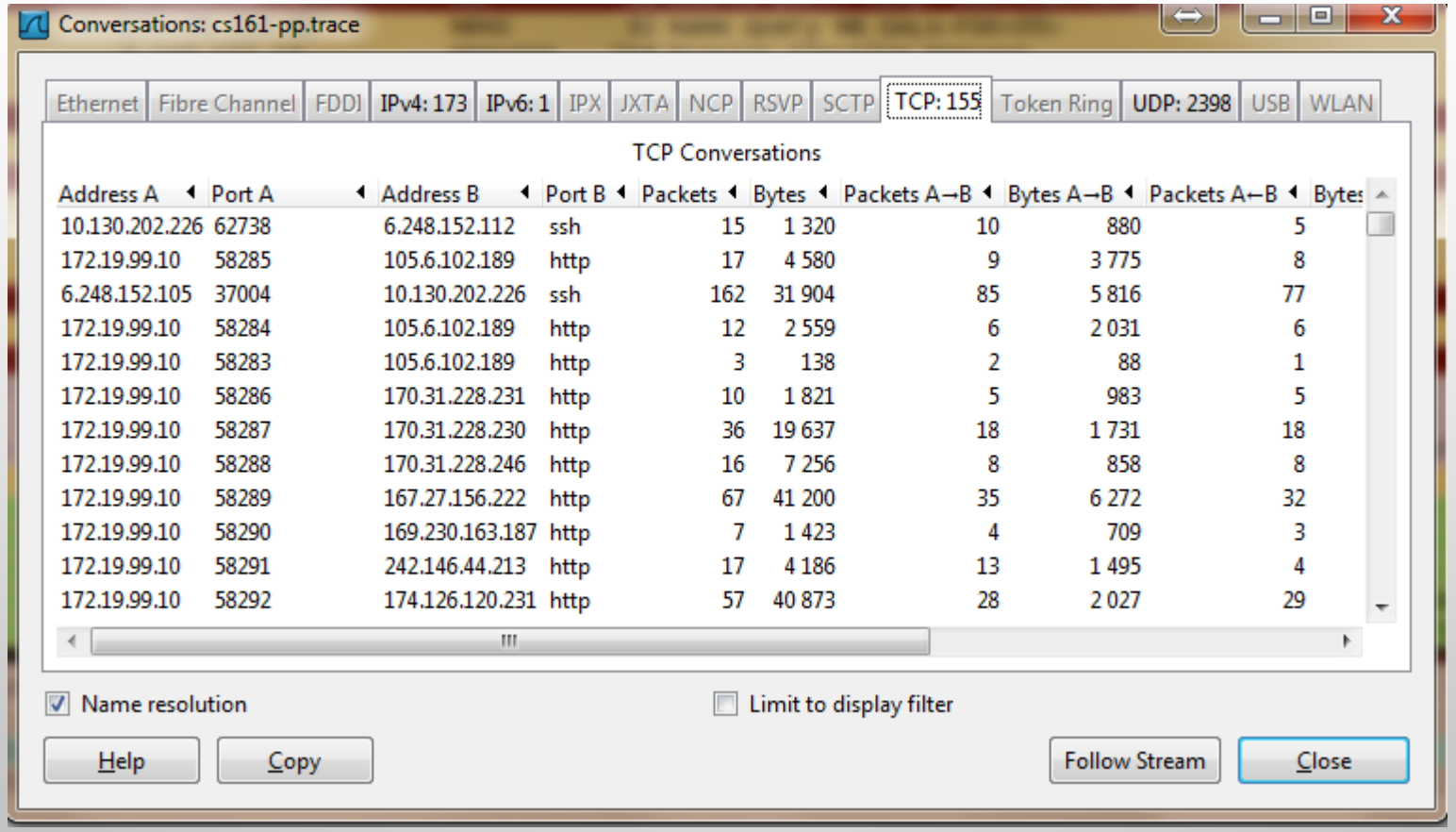
EXPERT INFO


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

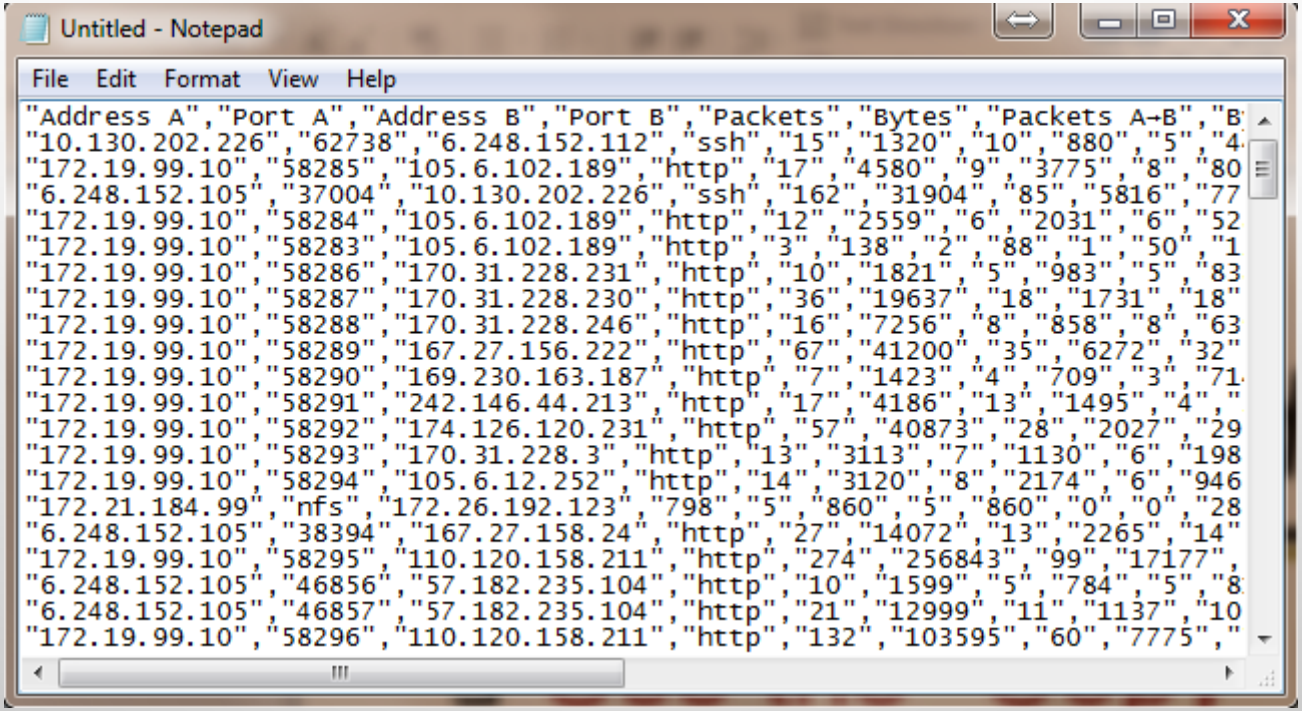
CONVERSATIONS



CONVERSATIONS



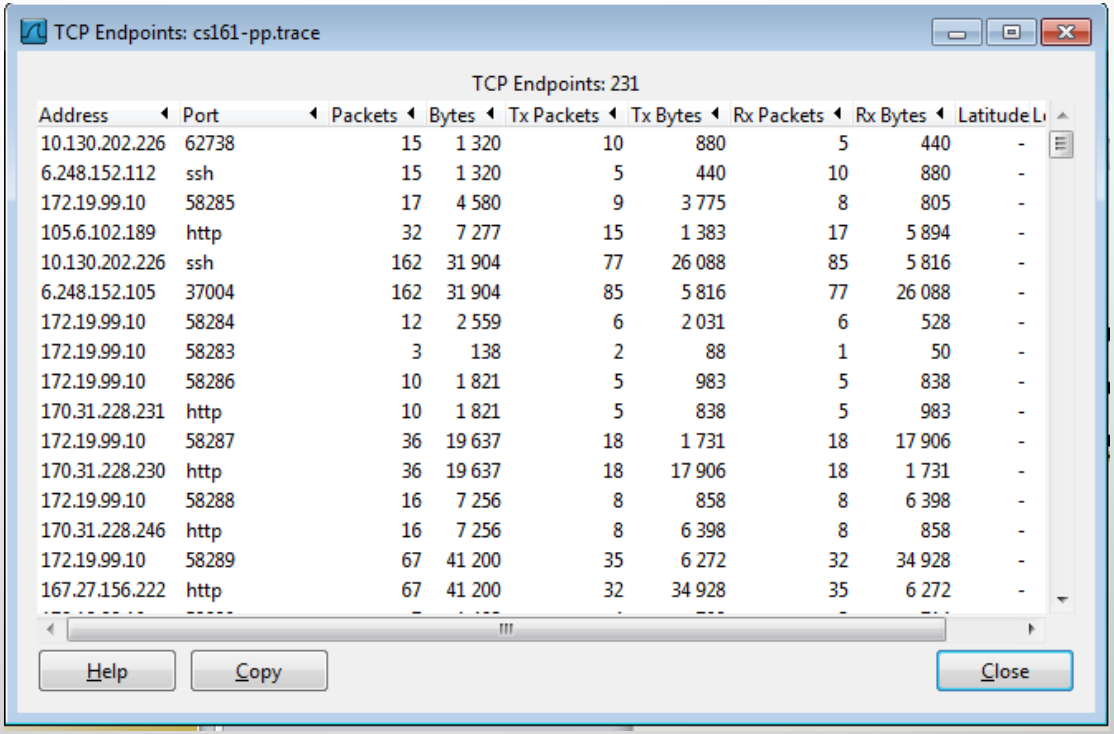
- USE THE “COPY” BUTTON TO COPY ALL TEXT INTO CLIPBOARD



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

FIND ENDPOINT STATISTICS

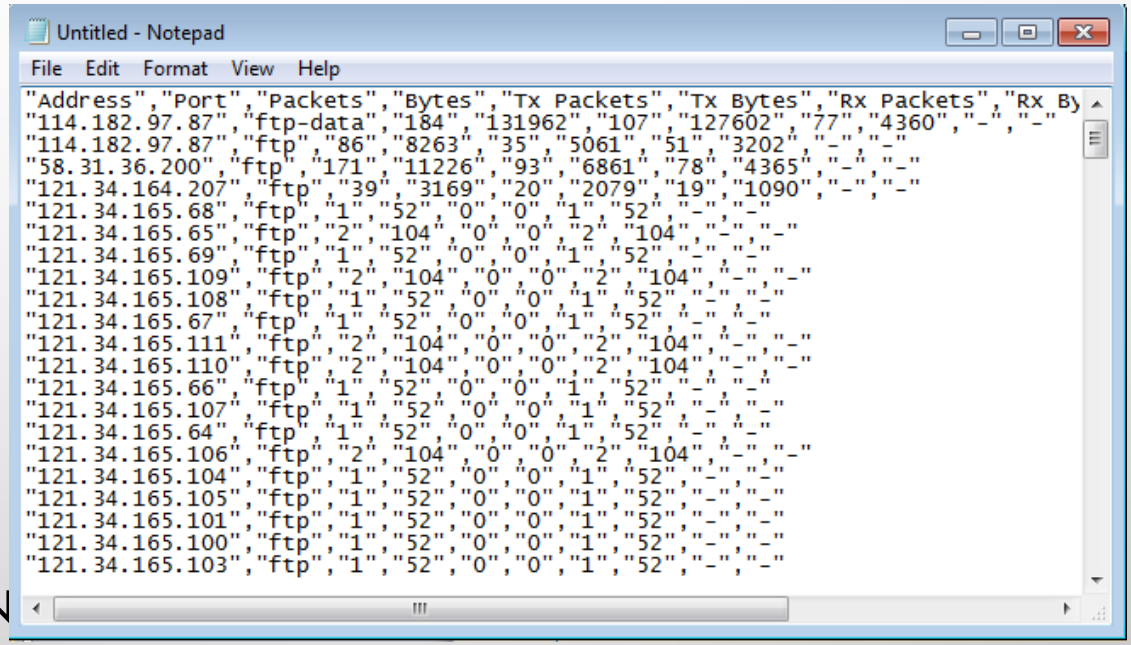
- ***MENU "STATISTICS" → "ENDPOINT LIST" → "TCP"***



- YOU CAN SORT BY FIELD
- "TX" : TRANSMIT "RX" : RECEIVE

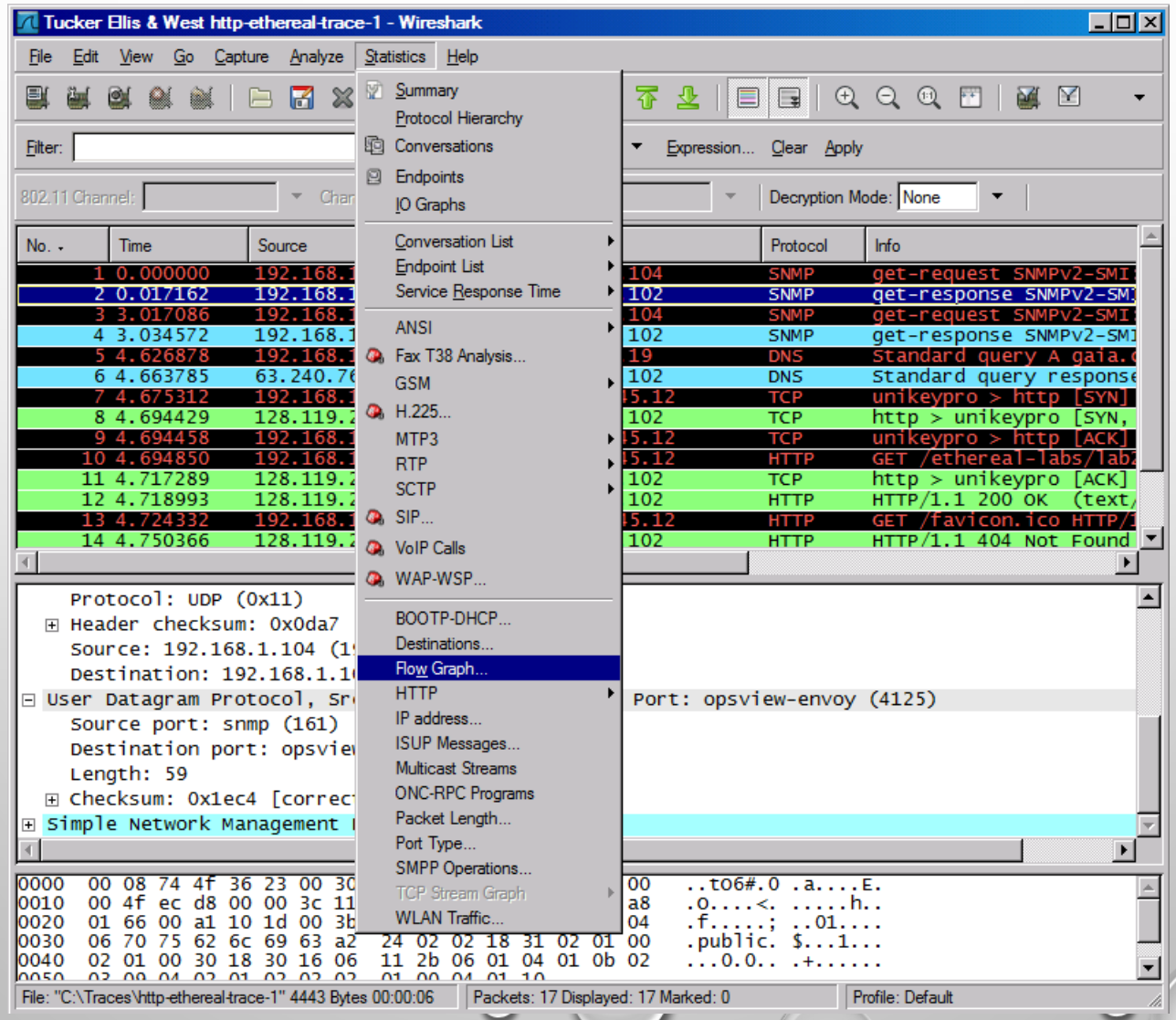
FIND ENDPOINT STATISTICS

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

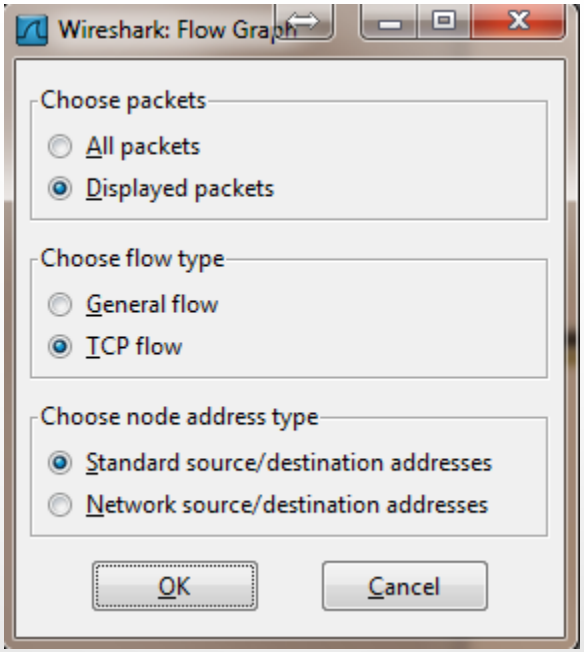


- THEN, YOU CAN AN U WANT

FLOW GRAPHS

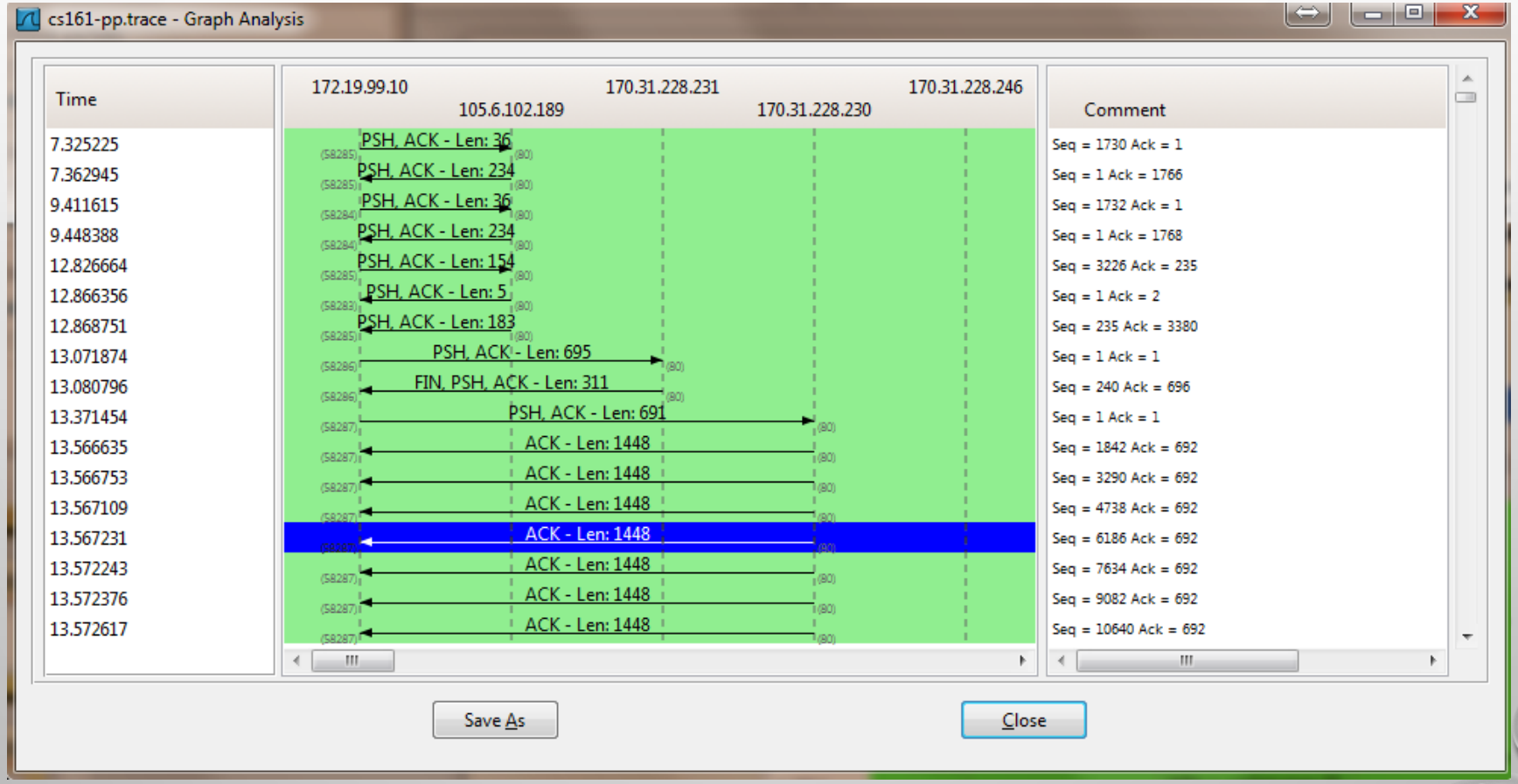

Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Summary
Protocol Hierarchy
Conversations
Endpoints
IO Graphs
Conversation List
Endpoint List
Service Response Time
ANSI
Fax T38 Analysis...
GSM
H.225...
MTP3
RTP
SCTP
SIP...
VoIP Calls
WAP-WSP...
BOOTP-DHCP...
Destinations...
Flow Graph...
HTTP
IP address...
ISUP Messages...
Multicast Streams
ONC-RPC Programs
Packet Length...
Port Type...
SMPP Operations...
TCP Stream Graph
WLAN Traffic...
Filter:
Expression... Clear Apply
Decryption Mode: None
No. Time Source Protocol Info
Protocol: UDP (0x11)
Header checksum: 0x0da7
Source: 192.168.1.104
Destination: 192.168.1.1
User Datagram Protocol, Sr
Source port: snmp (161)
Destination port: opsvie
Length: 59
Checksum: 0x1ec4 [correc
Simple Network Management
Port: opsview-envoy (4125)
File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06
Packets: 17 Displayed: 17 Marked: 0
Profile: Default

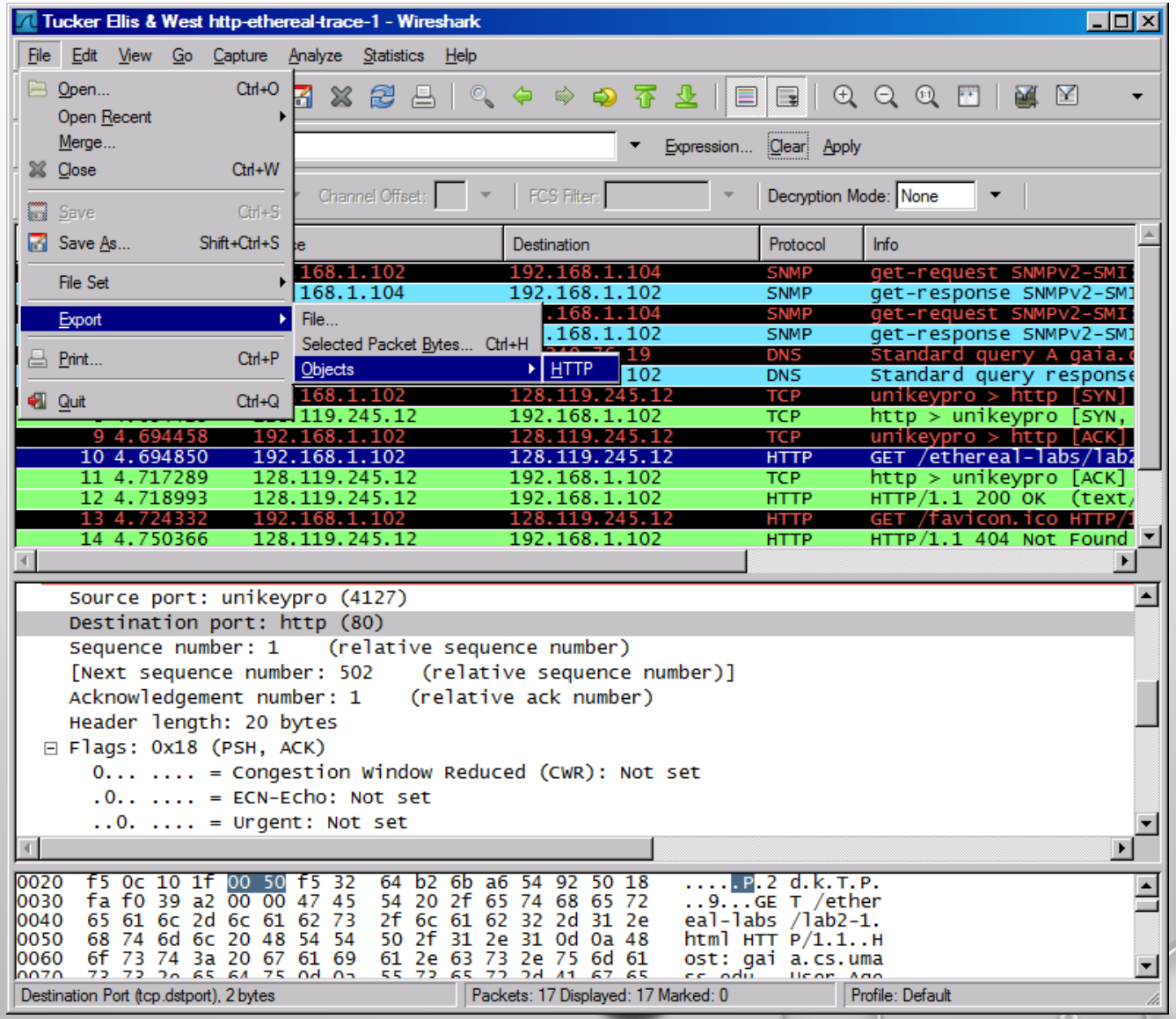
FLOW GRAPHS



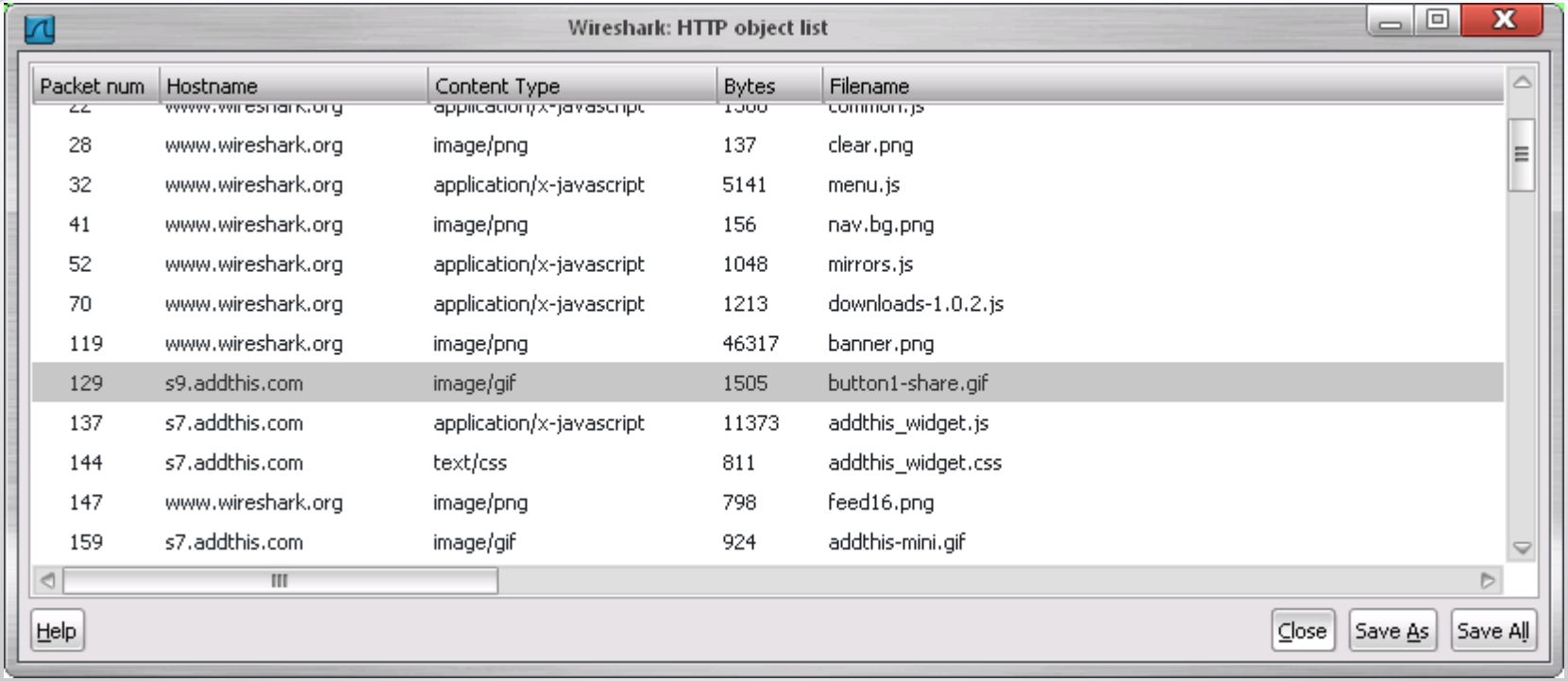
- The “displayed packet” option could let you only Show the flow of packets shown up
  for example, only display http traffic, then show The flow to analyze

FLOW GRAPHS

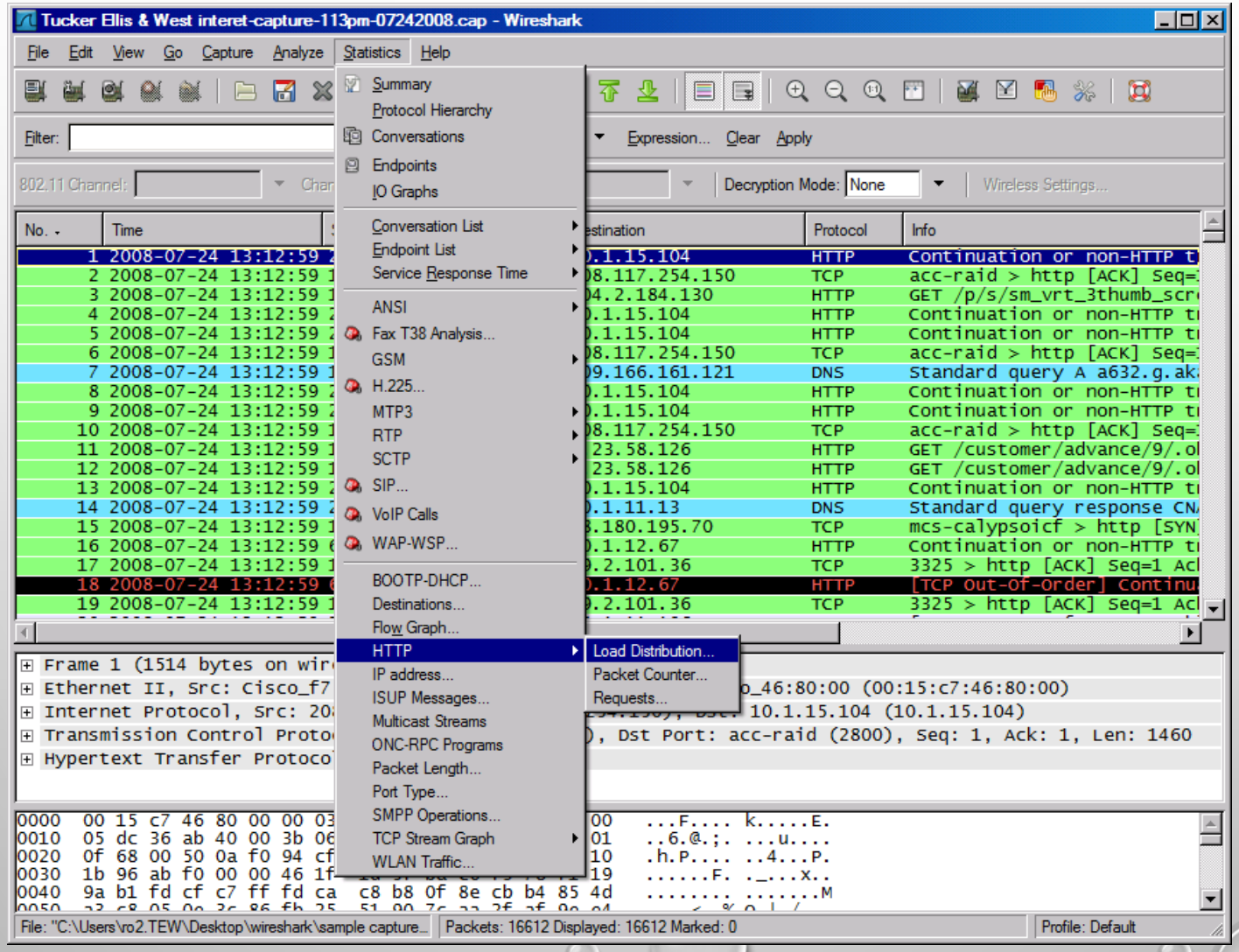

cs161-pp.trace - Graph Analysis
Time
7.325225
7.362945
9.411615
9.448388
12.826664
12.866356
12.868751
13.071874
13.080796
13.371454
13.566635
13.566753
13.567109
13.567231
13.572243
13.572376
13.572617
172.19.99.10
105.6.102.189
170.31.228.231
170.31.228.230
170.31.228.246
PSH, ACK - Len: 36
PSH, ACK - Len: 234
PSH, ACK - Len: 36
PSH, ACK - Len: 234
PSH, ACK - Len: 154
PSH, ACK - Len: 5
PSH, ACK - Len: 183
PSH, ACK - Len: 695
FIN, PSH, ACK - Len: 311
PSH, ACK - Len: 691
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
Comment
Seq = 1730 Ack = 1
Seq = 1 Ack = 1766
Seq = 1732 Ack = 1
Seq = 1 Ack = 1768
Seq = 3226 Ack = 235
Seq = 1 Ack = 2
Seq = 235 Ack = 3380
Seq = 1 Ack = 1
Seq = 240 Ack = 696
Seq = 1 Ack = 1
Seq = 1842 Ack = 692
Seq = 3290 Ack = 692
Seq = 4738 Ack = 692
Seq = 6186 Ack = 692
Seq = 7634 Ack = 692
Seq = 9082 Ack = 692
Seq = 10640 Ack = 692
Save As
Close


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
File...
Selected Packet Bytes... Ctrl+H
Objects
HTTP
Expression... Clear Apply
Channel Offset: FCS Filter: Decryption Mode: None
Destination Protocol Info
168.1.102 192.168.1.104 SNMP get-request SNMPv2-SMI
168.1.104 192.168.1.102 SNMP get-response SNMPv2-SMI
.168.1.104 SNMP get-request SNMPv2-SMI
.168.1.102 SNMP get-response SNMPv2-SMI
19 DNS Standard query A gaia.
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/lab
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/1
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
Destination Port (tcp.dstport), 2 bytes
Packets: 17 Displayed: 17 Marked: 0
Profile: Default

EXPORT HTTP OBJECTS

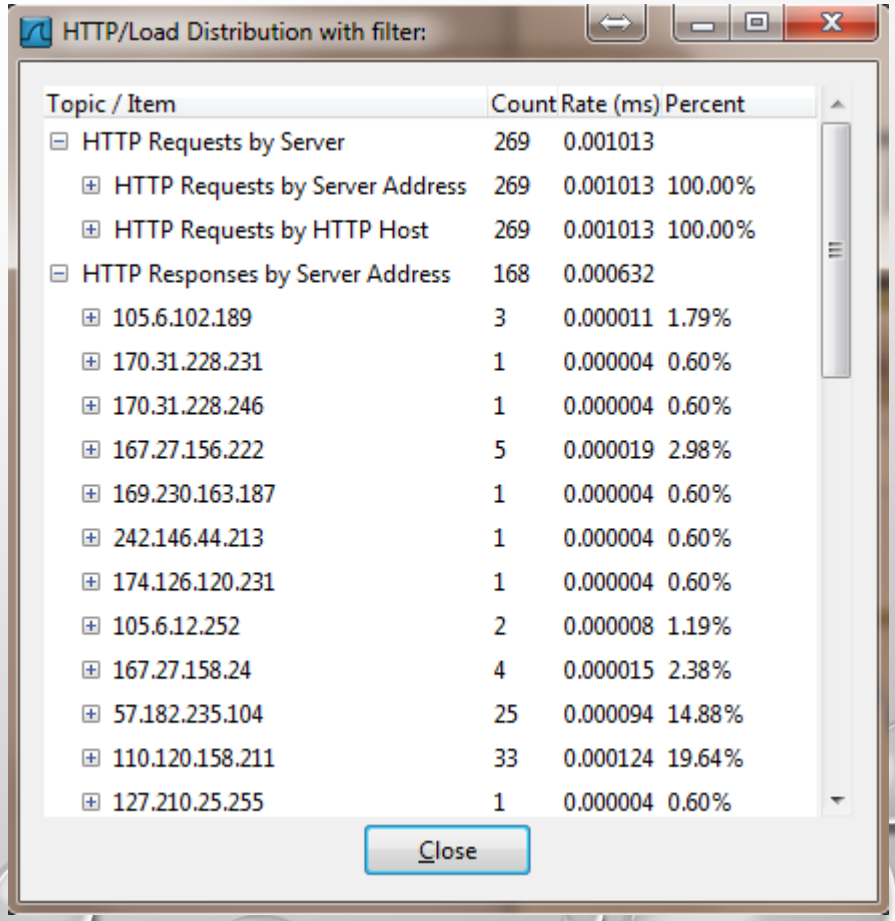


HTTP ANALYSIS

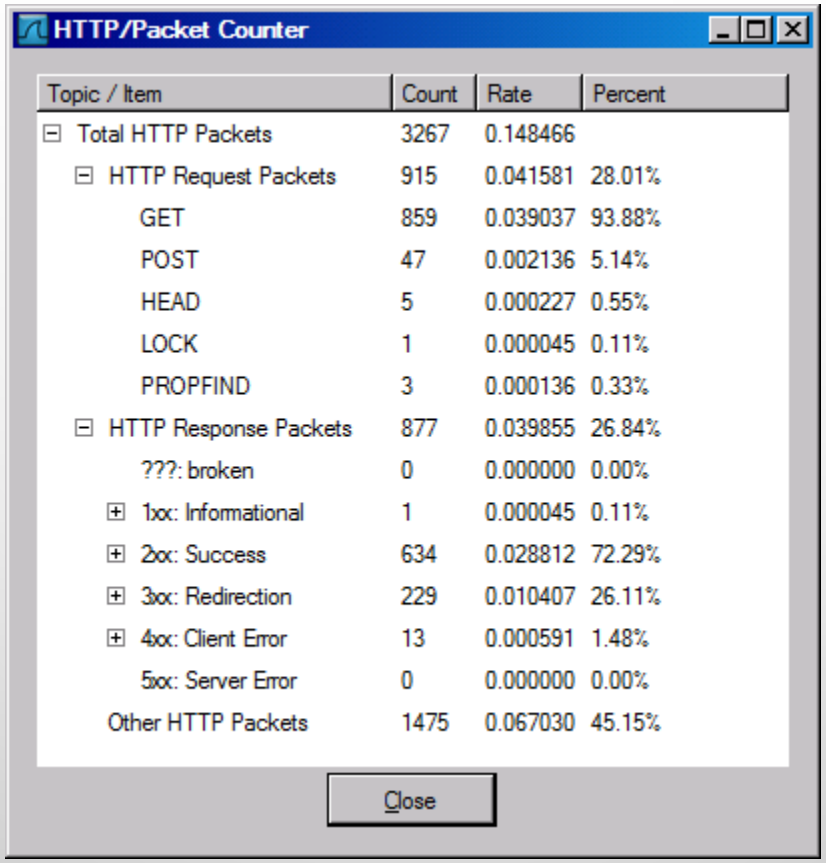


HTTP ANALYSIS – LOAD DISTRIBUTION

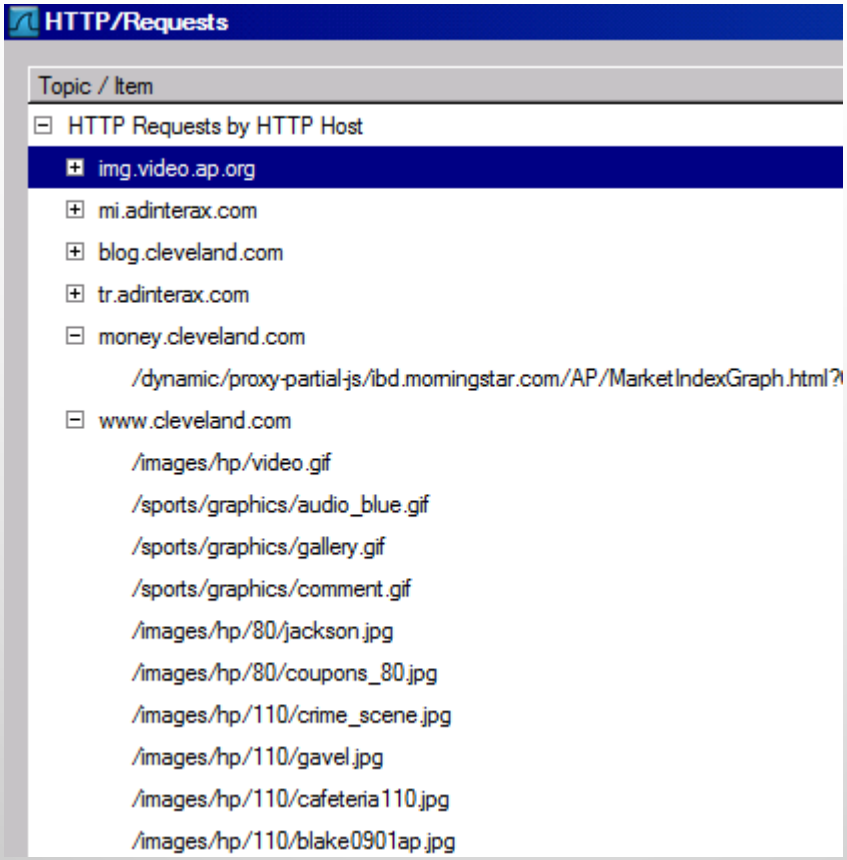
Click “Create Stat” button
You can add “filter” to only Show selected traffic



HTTP ANALYSIS – PACKET COUNTER



HTTP ANALYSIS – REQUESTS



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/
  - 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.THEGEEKSTUFF.COM/2009/03/15-PRACTICAL-UNIX-GREP-COMMAND-EXAMPLES/