DST-3

Defensive System Training 3: Network Analysis

Instructor: Capt Jon “Mamba” Bynum

USAF Weapons School • Nellis AFB

So What?

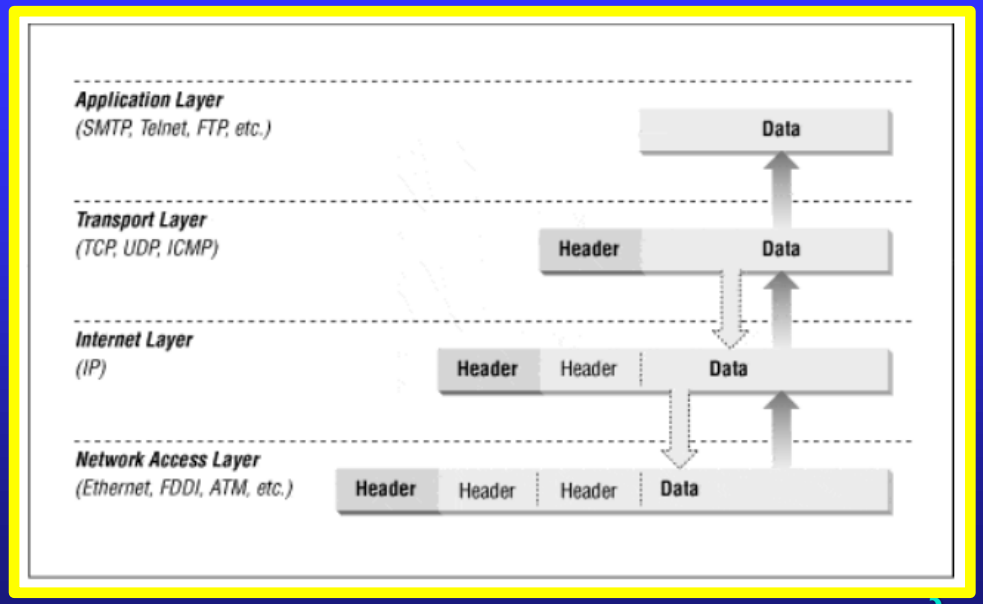
- Shake off the rust
- Become familiar with different network analysis tools
- *Do you wanna pass TA phase???*

Overview

- *Packet Headers*
- Wireshark
- TShark
- Tcpdump

Packet Headers

- The initial portion of a packet or a frame. The header contains control information such as addressing, routing, and protocol version. The format of this information depends on the protocol being used
- Headers are used to control the flow of packets through the network or over the communication link
- See excel sheet for examples



Overview

- Packet Headers
- *Wireshark*
- TShark
- Tcpdump

Wireshark

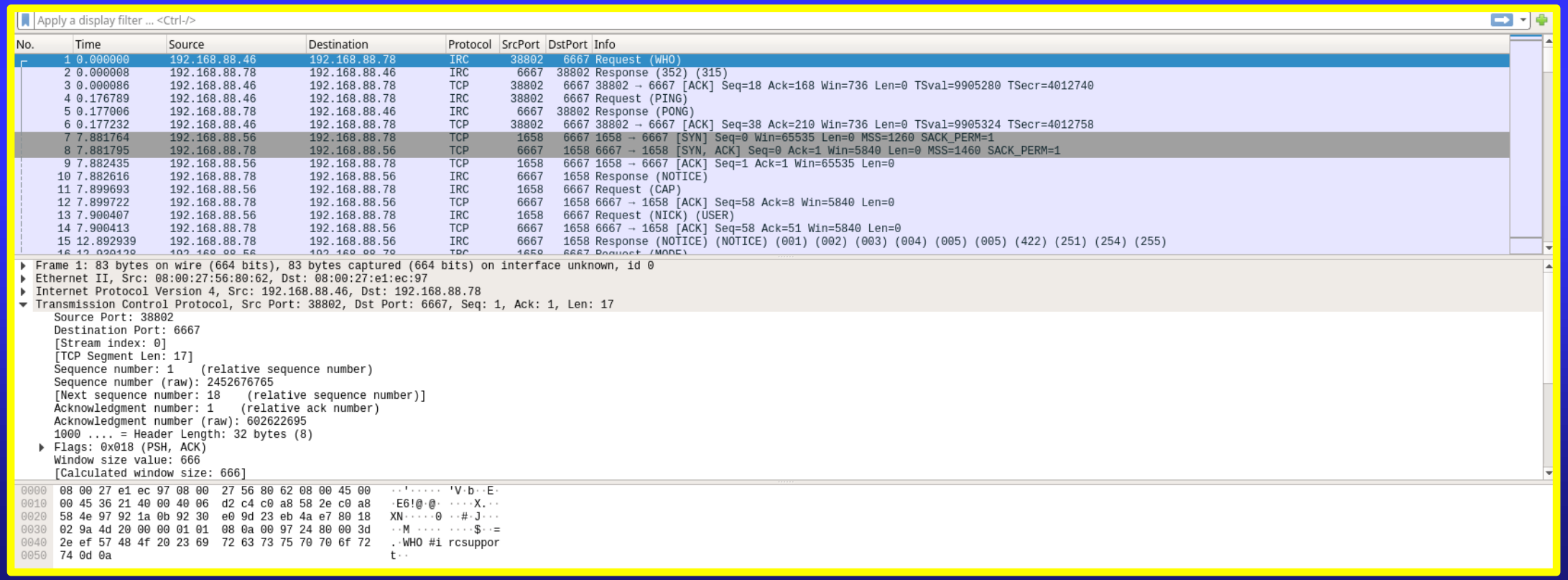
- Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.
- You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).
- Wireshark can capture traffic from many different network media types, including Ethernet, Wireless LAN, Bluetooth, USB, and more

Wireshark - Features

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics

Wireshark - Demo

- Wireshark *.pcap &
  - Load pcap via the command line
  - “&” gives you the terminal back



Wireshark - Demo

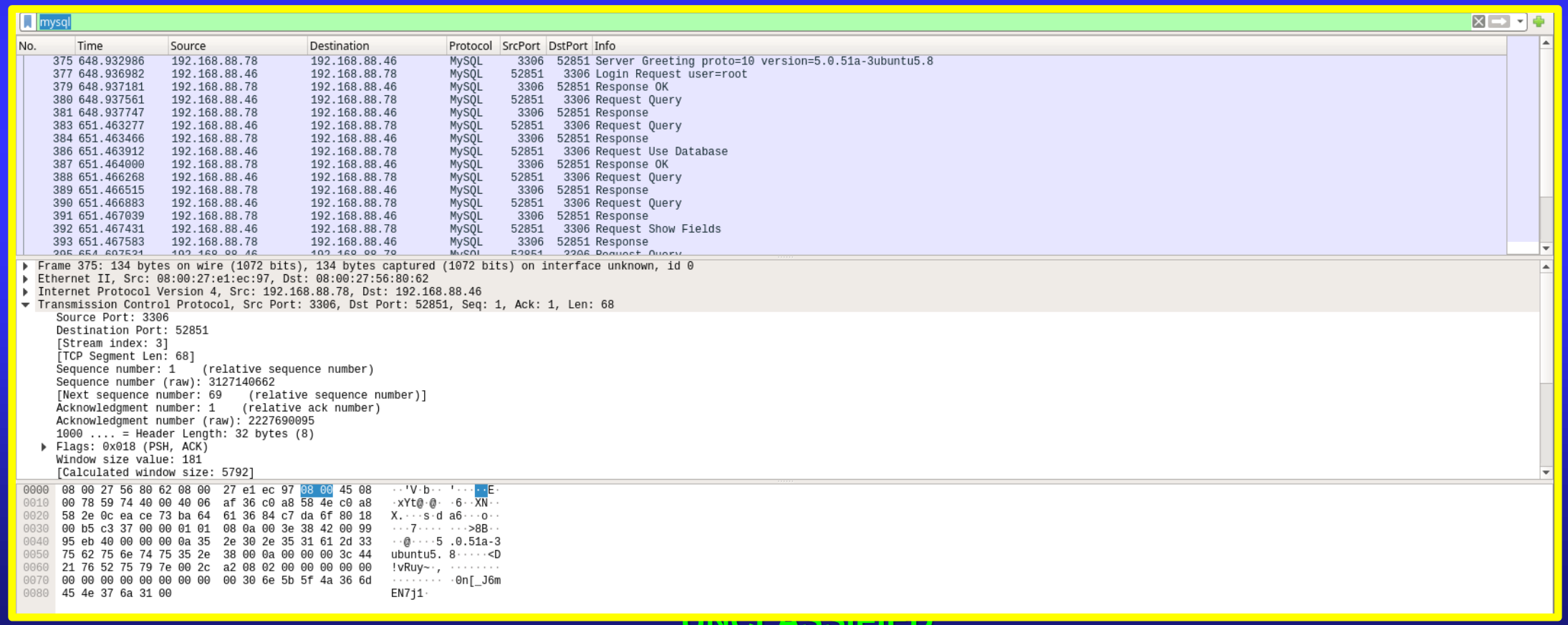
- Different IP address involved in conversations
- Bytes Exchanged
- Duration of the conversation that last the longest
  - Statistics > Conversations

Ethernet · 3 | IPv4 · 3 | IPv6 | TCP · 4 | UDP

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.88.46	192.168.88.78	292	31 k	159	14 k	133	16 k	0.000000	776.1629	149	
192.168.88.56	192.168.88.78	148	16 k	74	6,190	74	10 k	7.881764	768.4260	64	
192.168.88.73	192.168.88.78	43	7,617	23	4,083	20	3,534	358.488881	2.6367	12 k	

Wireshark - Demo

- Protocol Filtering
  - Make sure to clear the filter prior to a new filter



Wireshark - Demo

- Additional Protocol Filtering
  - arp.opcode == 1 (Right click > Apply as Filter > Selected)
  - Make sure to clear the filter prior to a new filter

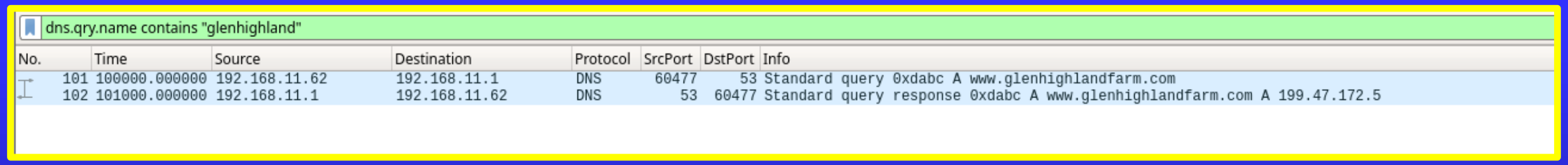
arp.opcode == 1

No.	Time	Source	Destination	Protocol	SrcPort	DstPort	Info
4	3000.000000	aa:00:04:00:0a:04	ff:ff:ff:ff:ff:ff	ARP			Who has 192.168.11.1? Tell 192.168.11.11
6	5000.000000	aa:00:04:00:0a:04	ff:ff:ff:ff:ff:ff	ARP			Who has 192.168.11.111? Tell 192.168.11.44
9	8000.000000	00:07:0d:af:f4:54	ff:ff:ff:ff:ff:ff	ARP			Who has 24.166.173.159? Tell 24.166.172.1 [ETHERNET
10	9000.000000	00:07:0d:af:f4:54	ff:ff:ff:ff:ff:ff	ARP			Who has 24.166.172.141? Tell 24.166.172.1 [ETHERNET
11	10000.000000	00:07:0d:af:f4:54	ff:ff:ff:ff:ff:ff	ARP			Who has 24.166.173.161? Tell 24.166.172.1 [ETHERNET
12	11000.000000	00:07:0d:af:f4:54	ff:ff:ff:ff:ff:ff	ARP			Who has 65.28.78.76? Tell 65.28.78.1 [ETHERNET FRAM
13	12000.000000	00:07:0d:af:f4:54	ff:ff:ff:ff:ff:ff	ARP			Who has 24.166.173.163? Tell 24.166.172.1 [ETHERNET
14	13000.000000	11:22:33:44:55:66	ff:ff:ff:ff:ff:ff	ARP			Gratuitous ARP for 192.168.11.13 (Request)
56	55000.000000	00:0c:29:2f:de:ad	ff:ff:ff:ff:ff:ff	ARP			Who has 192.168.1.104? Tell 192.168.1.109
57	56000.000000	00:0c:29:2f:7b:d0	ff:ff:ff:ff:ff:ff	ARP			Who has 192.168.1.104? Tell 192.168.1.103
59	58000.000000	00:0c:29:2f:de:ad	ff:ff:ff:ff:ff:ff	ARP			Who has 192.168.1.104? Tell 192.168.1.109
62	61000.000000	00:0c:29:f0:3c:f2	00:0c:29:2f:7b:d0	ARP			Who has 192.168.1.103? Tell 192.168.1.104 [ETHERNET

```
▸ Frame 80: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface unknown, id 0
▸ Ethernet II, Src: 00:0c:29:f0:3c:f2, Dst: 00:0c:29:2f:7b:d0
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
```

Wireshark - Demo

- Additional Protocol Filtering
  - dns.qry.name contains “glenhighland”
  - Make sure to clear the filter prior to a new filter



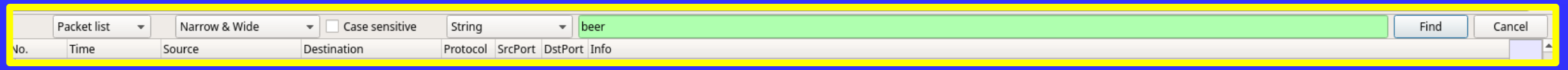
Wireshark - Demo

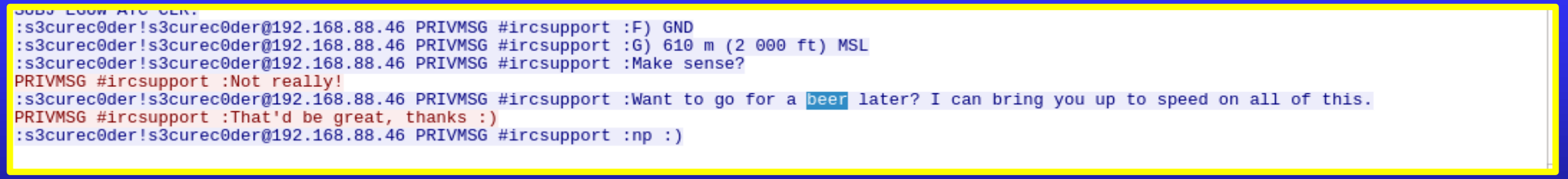
- Follow TCP Steam
  - Wireshark will set an appropriate display filter and display a dialog box with the data from the stream laid out
  - Analyze > Follow > *Stream

```
@...
5.0.51a-3ubuntu5.8.
...<D!vRuy~.,.................0n[_J6mEN7j1.:.....................................root.........e.f. j.Ha
).............!....select @@version_comment limit 1.....'....def....@@version_comment.....................   ....(Ubuntu).............SELECT
DATABASE()..... ....def...
DATABASE()...."..................................gmta..............show databases.....
1....def..SCHEMATA..Database.SCHEMA_NAME...@................"......information_schema.....gmta......."......show tables.....
9....def..TABLE_NAMES..Tables_in_gmta
TABLE_NAME...@................".....
auth_users......."......auth_users.>....def.gmta
auth_users
auth_users.user_pk.user_pk.?.......B....0?....def.gmta
auth_users
auth_users.username.username...P..........?....def.gmta
auth_users
auth_users.password.password...P..................X....insert into auth_users(username, password)values('launchmaster',
sha1('one2ThreeBOOM'))..............
```

Wireshark - Demo

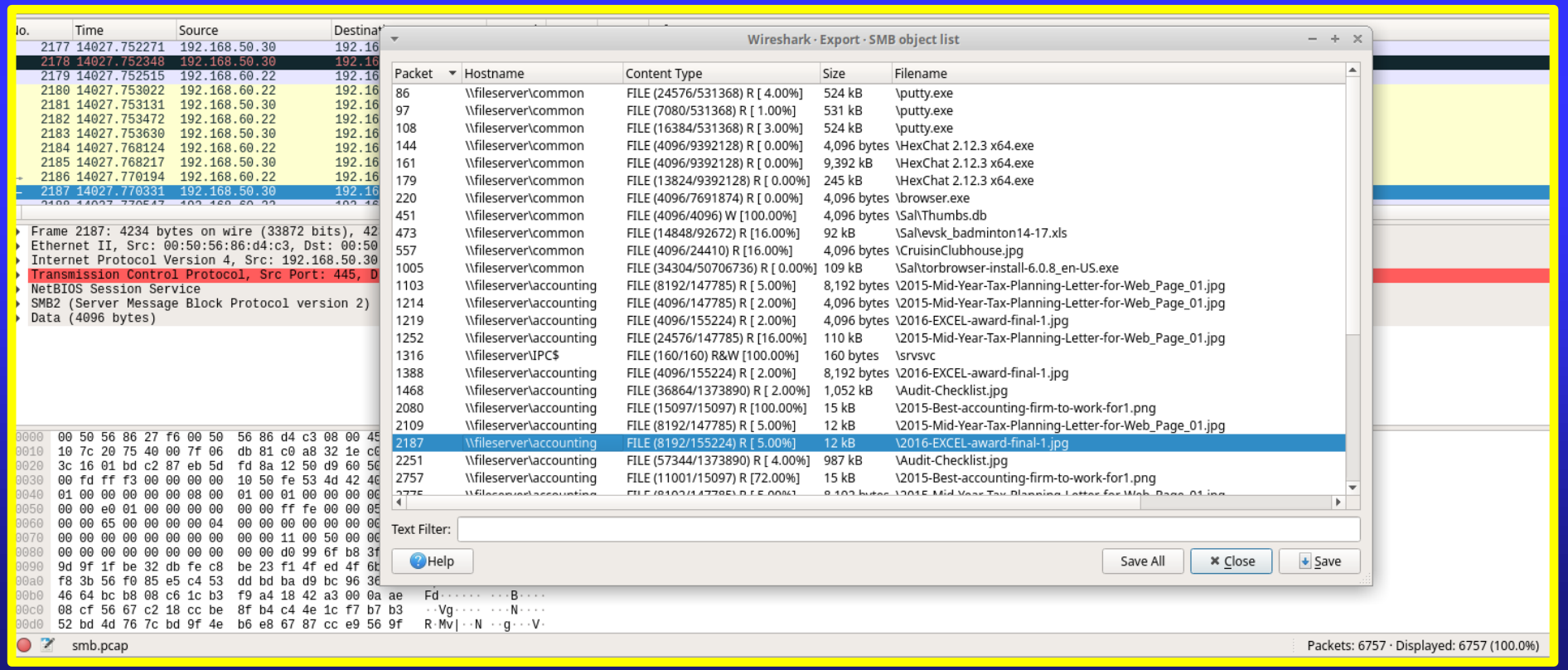
- Find Strings
  - Edit > Find Packet
  - Follow TCP Stream (*)
    - *Remember CLEAR PREVIOUS FILTER





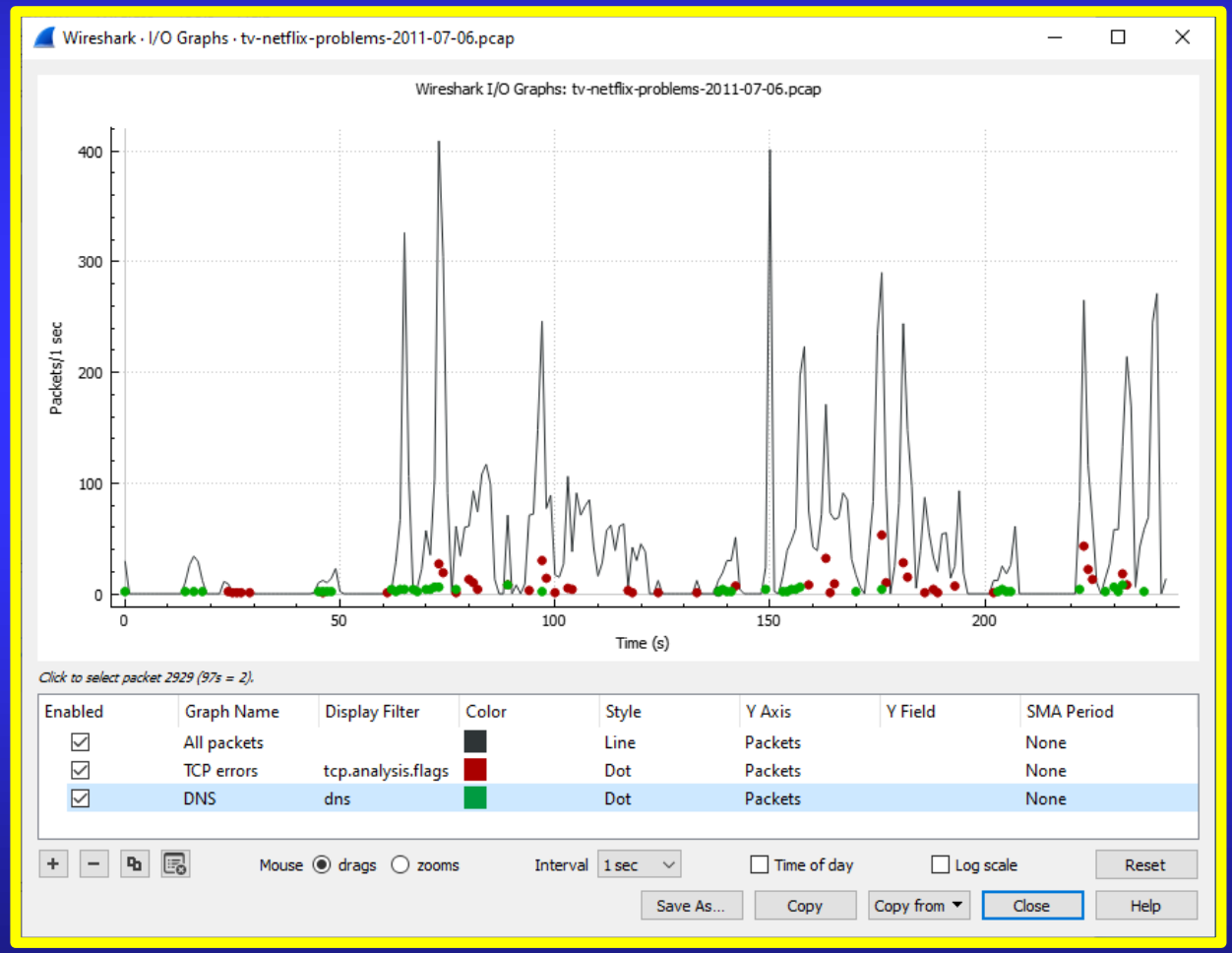
Wireshark - Demo

- Export Objects
  - File > Export Objects > [Protocol] *Example == SMB
  - Clicking on the Exported Object Pane WILL jump to respective packet



Wireshark - Demo

- I/O Graph
  - Statistics > I/O Graph
  - Lets you plot packet and protocol data
  - Identify spikes in traffic
- Display Filter
  - Query specific content
- Time of Day
  - Useful pivoting information



Overview

- Packet Headers
- Wireshark
- *TShark*
- Tcpdump

TShark

- TShark is a network protocol analyzer. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file

- Without any options set, TShark will work much like tcpdump. It will use the cap library to capture traffic from the first available network interface and displays a summary line on stdout for each received packet

TShark - Examples

- Advantage over Wireshark: (Light weight) it can be used in scripts and on a remote computer through an SSH connection
- Read a pcap file
  - Syntax: tshark -r http.pcap
- Reading a pcap, don't resolve names (Layers 3 or 4)
  - Syntax: tshark -nr http.pcap | grep «bad ip»
- Reading packets with a specific host IP address
  - Syntax: tshark -r http.pcap ip.host==”192.168.1.1"

TShark – Examples Cont.

- Redirecting Tshark Output to a New File
  - Syntax: tshark -r http.pcap -w /home/DUTCH/Desktop/Tshark/bad.pcap ip.dst==”192.168.1.4"
- Read a pcap, use the display filter “http.request.method==GET”
  - Syntax: tshark -r http.pcap -R “http.request.method==GET” -2
- Read a pcap, show TCP SYN packets not sent to port 80, don’t resolve names:
  - Syntax: tshark -r http.pcap -n -R “not tcp.port==80 and tcp.flags == 0x0002” -2
- -R == Read Filter
- -2 == Perform a two-pass analysis (Buffer Output)

Overview

- Packet Headers
- Wireshark
- TShark
- *Tcpdump*

Tcpdump

- Tcpdump is a command line utility that allows you to capture and analyze network traffic going through your system. It is often used to help troubleshoot network issues, as well as a security tool

- A powerful and versatile tool that includes many options and filters, tcpdump can be used in a variety of cases. Since it's a command line tool, it is ideal to run in remote servers or devices for which a GUI is not available, to collect data that can be analyzed later. It can also be launched in the background or as a scheduled job using tools like cron.

Tcpdump - Examples

- Capture HTTP traffic over port 80
  - sudo tcpdump -i eth0 -nn -s0 -v port 80
    - -i : Select interface that the capture is to take place on
    - -nn : A single (n) will not resolve hostnames. A double (nn) will not resolve hostnames or ports.
    - -s0 : Snap length, is the size of the packet to capture. -s0 will set the size to unlimited
    - -v : Verbose, using (-v) or (-vv) increases the amount of detail shown in the output, often showing more protocol specific information.
    - port 80 : this is a common port filter to capture only traffic on port 80

Tcpdump - Examples

- Write a capture file
  - sudo tcpdump -i eth0 -s0 -w test.pcap
- Read a capture file
  - sudo tcpdump –nr test.pcap
- Capture Hosts based on IP address
  - Syntax: sudo tcpdump -i eth0 host 10.10.1.1
- Capture all ICMP packets
  - sudo tcpdump -n icmp

Summary

- Packet Headers
- Wireshark
- TShark
- Tcpdump

Questions?

- Instructor's name: Capt Jon "Mamba" Bynum
- Instructor's address: USAF Weapons School
  4269 Tyndall Avenue
  Nellis AFB NV 89191-6062
- Instructor's phone: (702) 679-2207
- Instructor's e-mail: jon.bynum@us.af.mil

DST-3

Defensive System Training 3: Network Analysis

Instructor: Capt Jon “Mamba” Bynum

USAF Weapons School • Nellis AFB