

UNCLASSIFIED

DST-3

---

# Defensive System Training 3: Network Analysis



Instructor: Capt Jon “Mamba” Bynum

USAF Weapons School • Nellis AFB

UNCLASSIFIED



UNCLASSIFIED

# So What?

---

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

UNCLASSIFIED



UNCLASSIFIED

# Overview

---

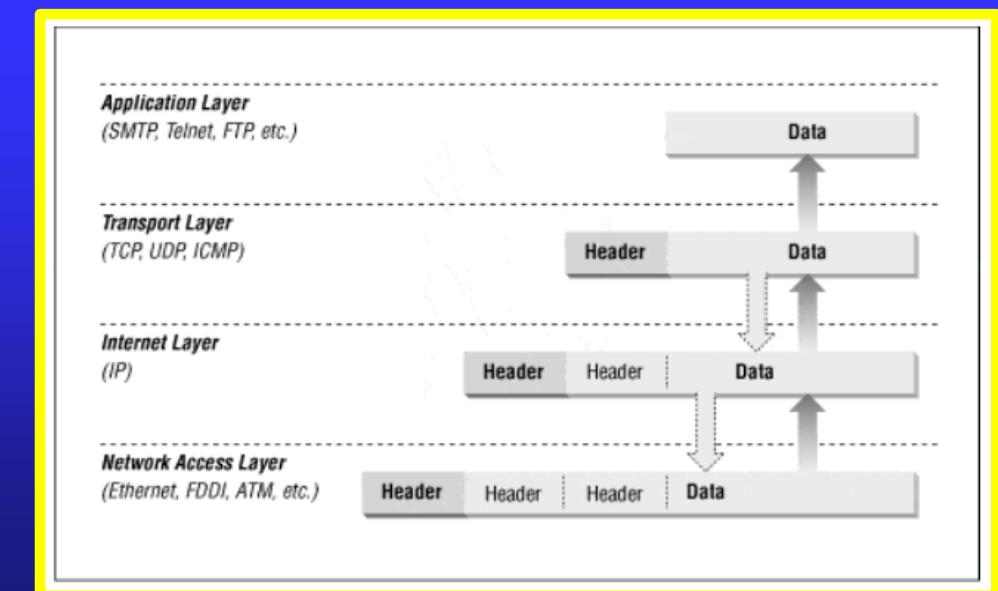
- *Packet Headers*
- Wireshark
- TShark
- Tcpdump

UNCLASSIFIED



# 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





UNCLASSIFIED

# Overview

---

- Packet Headers
- *Wireshark*
- TShark
- Tcpdump

UNCLASSIFIED



# 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



UNCLASSIFIED

# Wireshark - Demo

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

The screenshot shows the Wireshark interface with a yellow border. The main pane displays a list of network frames, and the bottom pane shows the raw hex and ASCII data for selected frame 1.

**Frame 1 details:**

- Frame 1: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface unknown, id 0
- Ethernet II, Src: 08:00:27:56:00:62, Dst: 08:00:27:e1:ec:97
- Internet Protocol Version 4, Src: 192.168.88.46, Dst: 192.168.88.78
- Transmission Control Protocol, Src Port: 38802, Dst Port: 6667, Seq: 1, Ack: 1, Len: 17

**TCP Segment Len: 17]**

Source Port: 38802  
Destination Port: 6667  
[Stream index: 0]  
[TCP Segment Len: 17]  
Sequence number: 1 (relative sequence number)  
Sequence number (raw): 2452676765  
[Next sequence number: 18 (relative sequence number)]  
Acknowledgment number: 1 (relative ack number)  
Acknowledgment number (raw): 602622695  
1000 ... = Header Length: 32 bytes (8)  
Flags: 0x018 (PSH, ACK)  
Window size value: 666  
[Calculated window size: 666]

**Raw Hex Data:**

```
0000  08 00 27 e1 ec 97 08 00 27 56 80 62 08 00 45 00  .V..b..E..  
0010  00 45 36 21 40 00 40 06 d2 c4 c0 a8 58 2e c0 a8  ..E!@..X...  
0020  58 4e 97 92 1a 0b 92 30 e0 9d 23 eb 4a e7 80 18  XN....0..#.J...  
0030  02 9a 4d 20 00 00 01 01 08 0a 00 97 24 80 00 3d  ..M.....$..=.  
0040  2e ef 57 48 4f 20 23 69 72 63 73 75 70 70 6f 72  ..WHO #! rcsuppor  
0050  74 0d 0a  t..
```



# 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

mysql

No.	Time	Source	Destination	Protocol	SrcPort	DstPort	Info
375	648.932986	192.168.88.78	192.168.88.46	MySQL	3306	52851	Server Greeting proto=10 version=5.0.51a-3ubuntu5.8
377	648.936982	192.168.88.46	192.168.88.78	MySQL	52851	3306	Login Request user=root
379	648.937181	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response OK
380	648.937561	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Query
381	648.937747	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response
383	651.463277	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Query
384	651.463466	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response
386	651.463912	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Use Database
387	651.464090	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response OK
388	651.466268	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Query
389	651.466515	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response
390	651.466883	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Query
391	651.467039	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response
392	651.467431	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Show Fields
393	651.467583	192.168.88.78	192.168.88.46	MySQL	3306	52851	Response
394	651.467584	192.168.88.46	192.168.88.78	MySQL	52851	3306	Request Query

▶ Frame 375: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface unknown, id 0  
▶ Ethernet II, Src: 08:00:27:e1:ec:97, Dst: 08:00:27:56:80:62  
▶ Internet Protocol Version 4, Src: 192.168.88.78, Dst: 192.168.88.46  
▼ Transmission Control Protocol, Src Port: 3306, Dst Port: 52851, Seq: 1, Ack: 1, Len: 68  
    Source Port: 3306  
    Destination Port: 52851  
    [Stream index: 3]  
    [TCP Segment Len: 68]  
    Sequence number: 1 (relative sequence number)  
    Sequence number (raw): 3127140662  
    [Next sequence number: 69 (relative sequence number)]  
    Acknowledgment number: 1 (relative ack number)  
    Acknowledgment number (raw): 2227690095  
    1000 .... = Header Length: 32 bytes (8)  
    Flags: 0x018 (PSH, ACK)  
    Window size value: 181  
    [Calculated window size: 5792]  
0000 08 00 27 56 80 62 08 00 27 e1 ec 97 08 00 45 08 ..'V.b.. '....E.  
0010 00 78 59 74 40 00 40 06 af 36 c8 a8 58 4e c0 a8 .xYt@.0. 6.XN..  
0020 58 2e 0c ea ce 73 ba 64 61 36 84 c7 da 6f 80 18 X...s-d a6...o..  
0030 00 b5 c3 37 00 00 01 01 08 0a 00 3e 38 42 00 99 ..7.....>8B..  
0040 95 eb 49 00 00 00 0a 35 2e 30 2e 35 31 61 2d 33 ..@....5. 0.51a-3  
0050 75 62 75 6e 74 75 35 2e 38 00 0a 00 00 00 3c 44 ubuntu5. 8....<D  
0060 21 76 52 75 79 7e 00 2c a2 08 02 00 00 00 00 00 !vRuy~, ..  
0070 00 00 00 00 00 00 00 00 00 30 6e 5b 5f 4a 36 6d .....0n[\_J6m  
0080 45 4e 37 6a 31 00 EN7j1.



# 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 192.168.11.11? 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
74	72000.000000	00:0c:29:2f:de:ad	ff:ff:ff:ff:ff:ff	ARP			Who has 192.168.1.104? Tell 192.168.1.109

▶ 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)

Sender MAC address: 00:0c:29:f0:3c:f2 (00:0c:29:f0:3c:f2)



# Wireshark - Demo

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

dns.qry.name contains "glenhighland"								
No.	Time	Source	Destination	Protocol	SrcPort	DstPort	Info	
101	100000.000000	192.168.11.62	192.168.11.1	DNS	60477	53	Standard query 0xdabc A www.glenhighlandfarm.com	
102	101000.000000	192.168.11.1	192.168.11.62	DNS		53	60477 Standard query response 0xdabc A www.glenhighlandfarm.com A 199.47.172.5	



# 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~,.....On[_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')).....
```



UNCLASSIFIED

# Wireshark - Demo

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

The screenshot shows the Wireshark interface with a yellow search bar containing the word "beer". Below the search bar is a table header with columns: No., Time, Source, Destination, Protocol, SrcPort, DstPort, and Info. The main pane displays several lines of IRC traffic. A yellow box highlights the last few lines of the conversation:

```
SUBJ: LOOK AT THIS!
:s3curec0der!s3curec0der@192.168.88.46 PRIVMSG #ircsupport :F) GND
:s3curec0der!s3curec0der@192.168.88.46 PRIVMSG #ircsupport :G) 610 m (2 000 ft) MSL
:s3curec0der!s3curec0der@192.168.88.46 PRIVMSG #ircsupport :Make sense?
PRIVMSG #ircsupport :Not really!
:s3curec0der!s3curec0der@192.168.88.46 PRIVMSG #ircsupport :Want to go for a beer later? I can bring you up to speed on all of this.
PRIVMSG #ircsupport :That'd be great, thanks :)
:s3curec0der!s3curec0der@192.168.88.46 PRIVMSG #ircsupport :np :)
```

UNCLASSIFIED



UNCLASSIFIED

# Wireshark - Demo

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

The screenshot shows the Wireshark interface with a yellow border around the main window. The left pane displays network traffic, and the right pane shows the "Wireshark - Export - SMB object list". The list contains 275 entries, each detailing a file transferred via SMB. The columns are: Packet, Hostname, Content Type, Size, and Filename. The "Content Type" column shows various file types like .exe, .jpg, .pdf, and .xls. The "Size" column shows file sizes in kilobytes. The "Filename" column lists the names of the files transferred. A specific entry for file 2187 is highlighted in blue, and its details are shown in the bottom left pane, including the file size (4,096 bytes) and type (FILE (8192/155224) R [ 5.00%]).

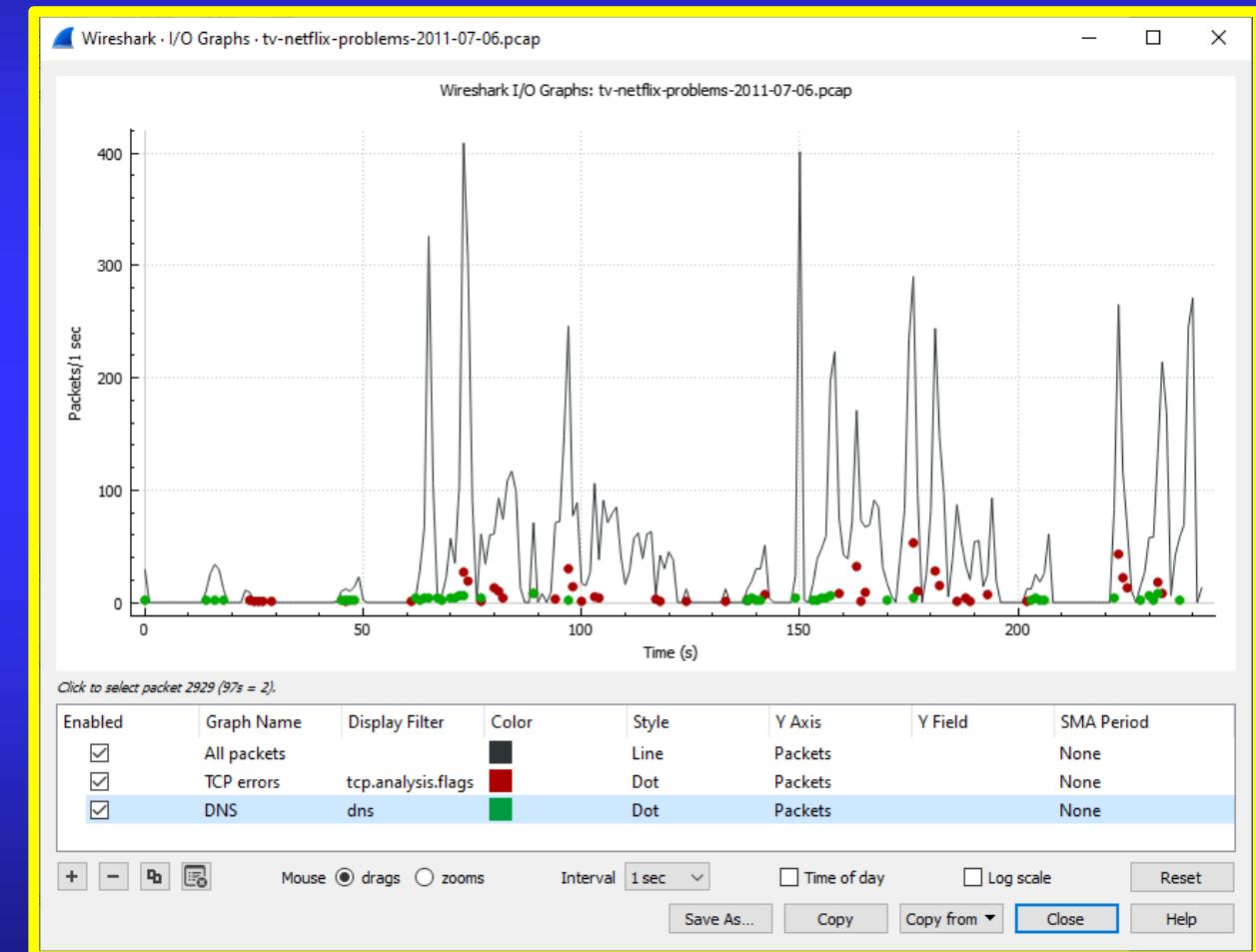
Packet	Hostname	Content Type	Size	Filename
86	\fileserver\common	FILE (24576/531368) R [ 4.00%]	524 kB	\putty.exe
97	\fileserver\common	FILE (7080/531368) R [ 1.00%]	531 kB	\putty.exe
108	\fileserver\common	FILE (16384/531368) R [ 3.00%]	524 kB	\putty.exe
144	\fileserver\common	FILE (4096/9392128) R [ 0.00%]	4,096 bytes	\HexChat 2.12.3 x64.exe
161	\fileserver\common	FILE (4096/9392128) R [ 0.00%]	9,392 kB	\HexChat 2.12.3 x64.exe
179	\fileserver\common	FILE (13824/9392128) R [ 0.00%]	245 kB	\HexChat 2.12.3 x64.exe
220	\fileserver\common	FILE (4096/7691874) R [ 0.00%]	4,096 bytes	\browser.exe
451	\fileserver\common	FILE (4096/4096) W [100.00%]	4,096 bytes	\SalThumbs.db
473	\fileserver\common	FILE (14848/92672) R [16.00%]	92 kB	\SalEvsk_badminton14-17.xls
557	\fileserver\common	FILE (4096/24410) R [16.00%]	4,096 bytes	\CruisinClubhouse.jpg
1005	\fileserver\common	FILE (34304/50706736) R [ 0.00%]	109 kB	\SalTorbrowser-install-6.0.8_en-US.exe
1103	\fileserver\accounting	FILE (8192/147785) R [ 5.00%]	8,192 bytes	\2015-Mid-Year-Tax-Planning-Letter-for-Web_Page_01.jpg
1214	\fileserver\accounting	FILE (4096/147785) R [ 2.00%]	4,096 bytes	\2015-Mid-Year-Tax-Planning-Letter-for-Web_Page_01.jpg
1219	\fileserver\accounting	FILE (4096/155224) R [ 2.00%]	4,096 bytes	\2016-EXCEL-award-final-1.jpg
1252	\fileserver\accounting	FILE (24576/147785) R [16.00%]	110 kB	\2015-Mid-Year-Tax-Planning-Letter-for-Web_Page_01.jpg
1316	\fileserver\IPC\$	FILE (160/160) R&W [100.00%]	160 bytes	\srvsvc
1388	\fileserver\accounting	FILE (4096/155224) R [ 2.00%]	8,192 bytes	\2016-EXCEL-award-final-1.jpg
1468	\fileserver\accounting	FILE (36864/1373890) R [ 2.00%]	1,052 kB	\Audit-Checklist.jpg
2080	\fileserver\accounting	FILE (15097/15097) R [100.00%]	15 kB	\2015-Best-accounting-firm-to-work-for1.png
2109	\fileserver\accounting	FILE (8192/147785) R [ 5.00%]	12 kB	\2015-Mid-Year-Tax-Planning-Letter-for-Web_Page_01.jpg
2187	\fileserver\accounting	FILE (8192/155224) R [ 5.00%]	12 kB	\2016-EXCEL-award-final-1.jpg
2251	\fileserver\accounting	FILE (57344/1373890) R [ 4.00%]	987 kB	\Audit-Checklist.jpg
2757	\fileserver\accounting	FILE (11001/15097) R [72.00%]	15 kB	\2015-Best-accounting-firm-to-work-for1.png
2775	\fileserver\accounting	FILE (8192/147785) R [ 5.00%]	8,192 bytes	\2015-Mid-Year-Tax-Planning-Letter-for-Web_Page_01.jpg



UNCLASSIFIED

# 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



UNCLASSIFIED



UNCLASSIFIED

# Overview

---

- Packet Headers
- Wireshark
- *TShark*
- Tcpdump

UNCLASSIFIED



# 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)`



UNCLASSIFIED

# Overview

---

- Packet Headers
- Wireshark
- TShark
- *Tcpdump*

UNCLASSIFIED



# 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`



UNCLASSIFIED

# Summary

---

- Packet Headers
- Wireshark
- TShark
- Tcpdump

UNCLASSIFIED



UNCLASSIFIED

# 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](mailto:jon.bynum@us.af.mil)

UNCLASSIFIED

UNCLASSIFIED

DST-3

---

# Defensive System Training 3: Network Analysis



Instructor: Capt Jon “Mamba” Bynum

USAF Weapons School • Nellis AFB

UNCLASSIFIED