NETWORK ANALYSIS TOOLS

A study material for the students of GLS University

Prepared by : Harshita Maheshwari

Subject : Network Analysis Tools

Packet Capture using Dumpcap

Prepared by : Prof. Harshita Maheshwari

What is dumpcap?

- Command-line tool for capturing packets
- Comes with Wireshark suite
- Used for fast, low-overhead packet capture
- Does not analyze packets, only captures

Basic Syntax

- dumpcap -i <interface> -w <output_file>
- -i: Interface to capture on (e.g., eth0, 1)
- -w: Output file to save packets (e.g., capture.pcapng)

Step 1: List Interfaces

- Command: dumpcap -D
- Shows list of available interfaces with index numbers

Step 2: Basic Capture

- Command: dumpcap -i 1 -w capture.pcapng
- Captures packets on interface 1 and saves to file

Step 3: Capture Limits

- -c: Capture a specific number of packets (e.g., -c 100)
- -a duration:10 → Stop after 10 seconds

-a filesize:5000 → Stop after 5MB file size

Step 4: Ring Buffer Capture

- Use -b to create multiple rotating files
- Example: dumpcap -i 1 -b filesize:10240 -b files:5 -w capture
- Creates 5 files of 10MB each in rotation

Step 5: Apply Capture Filter

- Use -f to apply BPF (Berkeley Packet Filter)
- Example: dumpcap -i 1 -f "tcp" -w tcp_traffic.pcapng
- Only captures TCP traffic

Common Use Cases

- Basic: dumpcap -i 1 -w output.pcapng
- Timed: dumpcap -i 1 -a duration:30 -w timed.pcapng
- Limit packets: -c 500
- Ring buffer: -b filesize:5120 -b files:3
- Filter HTTP: -f "tcp port 80"

Name Resolution

What is Name Resolution?

Name resolution in Wireshark refers to converting numeric values like IP addresses, MAC addresses, and port numbers into human-readable names to make analysis easier.

Types of Name Resolutions

1. MAC Name Resolution: MAC → Manufacturer
2. Network Name Resolution: IP → Hostname
3. Transport Name Resolution: Port → Protocol Name

Enable/Disable (GUI)

- View > Name Resolution > Toggle Options
- Edit > Preferences > Name Resolution
- Capture > Options > Uncheck 'Enable network name resolution'

Best Practices

- Name resolution may slow down analysis.
- DNS queries might be sent to resolve names.
- Prefer disabling when working offline or with sensitive data.
- Use raw values (IP, MAC, Port) during filtering.
- Enable name resolution after analysis for readability.
- Avoid DNS leakage: Preferences > Name Resolution > Check 'Only use the profile's hosts file'

Extract Files/Traffic from .pcapng

1. Open the .pcapng File

- Open Wireshark.
- Go to File > Open.
- Select your .pcapng capture file.

2. Identify the Protocol

- Look at the Protocol column in packet list.
- Common file transfer protocols: HTTP, FTP, SMB, SMTP, TFTP, TLS/SSL.
- Focus on protocols used for transferring files.

3. Use 'Follow Stream'

- Right-click a packet (e.g., HTTP GET).
- Choose Follow > TCP Stream or HTTP Stream.
- View the conversation and Save As to extract content.

4. Export Objects

- Go to File > Export Objects.
- Choose from: HTTP, SMB, DICOM.
- Select desired files and click Save.

5. Filter Specific Transfers

- Use filters like:
  - http.request.uri contains ".pdf"
  - ftp-data
  - tcp.port == 445
- Helps isolate specific file types or protocols.

6. Encrypted Traffic (HTTPS, etc.)

- Encrypted traffic can't be extracted directly.
- Requires SSL key log file from client.
- Configure in Preferences > Protocols > TLS to decrypt.

Example: Extract from HTTP

- Apply filter: http
- Go to File > Export Objects > HTTP
- Select and save desired files (e.g., .pdf, .jpg)

Filtering Operators, regular Expressions, Functions and more…

Slice Operator

- Allows partial matching on byte sequences
- Syntax: field[offset:length] == value
- Example: eth.addr[0:3] == 00:11:22
- Useful to check portions of MAC/IP addresses or payloads

Membership Operator (in)

- Checks if a field belongs to a set of values
- Syntax: field in {value1 value2}
- Example: ip.src in {192.168.1.1 192.168.1.2}
- Filters traffic from multiple IP addresses

Using Regular Expressions

- Enables pattern-based filtering with matches keyword
- Syntax: field matches "regex"
- Example: http.host matches "^www\..*\.com$"
- Useful for filtering hostnames, URLs, user agents, etc.

Common Mistakes with Regex

- Improper use of escape characters (\ vs \\\)
- Using greedy expressions (e.g., .* instead of [^/]*)
- Slower performance on large capture files
- Always test regex on sample data before applying

Combining Expressions

- Use logical operators to create complex conditions:
- AND: && | OR: || | NOT: !
- Example: ip.dst == 10.0.0.5 && tcp.port == 443
- Group conditions with parentheses if needed

Functions in Filters

- Some expressions allow use of functions:
- len(field) – field length
- lower() / upper() – string case conversion
- Not all display filters support functions – use carefully

Finding Packets in Wireshark

- Use Ctrl+F or Edit > Find Packet
- Filter by: String, Hex value, Display filter expression
- Helps locate specific packets quickly

Automatic Remote Filtering

- Useful when capturing traffic from remote systems
- Apply filters remotely to reduce data load:
- Via rpcapd or SSH
- Tools: tcpdump, dumpcap, remote capture in Wireshark
- Capture only relevant packets

Thank You