**Tool/Utility Purpose and Benefit**

capinfos A program that provides details of a specified capture file. It is suggested to view the summary of the capture file before starting an investigation.

grep Helps search plain-text data.

cut Helps cut parts of lines from a specified data source.

uniq Filters repeated lines/values.

nl Views the number of shown lines.

sed A stream editor.

awk Scripting language that helps pattern search and processing.### TShark Command-Line Features and Queries

**1**  capinfos demo.pcapng

**Parameter Purpose**

-h

Display the help page with the most common features.

tshark -h

-v

Show version info.

tshark -v

-D

List available sniffing interfaces.

tshark -D

-i

Choose an interface to capture live traffic.

tshark -i 1

tshark -i ens55

No Parameter

Sniff the traffic like tcpdump.

tshark

**Parameter Purpose**

-r

Read/input function. Read a capture file.

tshark -r demo.pcapng

-c

Packet count. Stop after capturing a specified number of packets.

E.g. stop after capturing/filtering/reading 10 packets.

tshark -c 10

-w

Write/output function. Write the sniffed traffic to a file.

tshark -w sample-capture.pcap

-V

Verbose.

**Provide detailed information for each packet. This option will provide details similar to Wireshark's "Packet Details Pane".**

tshark -V

-q

Silent mode.

Suspress the packet outputs on the terminal.

tshark -q

-x

Display packet bytes.

Show packet details in hex and ASCII dump for each packet.

tshark -x

**tshark -r teamwork.pcap -V | grep -Eo '[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'**

**Explanation:**

tshark -r teamwork.pcap -V:

-r teamwork.pcap: Reads the specified packet capture file (teamwork.pcap).

-V: Displays the full details of each packet in verbose format, including all protocol fields.

Pipe (|):

The output of the tshark command is passed to grep for further processing.

grep -Eo '[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}':

-E: Enables extended regular expressions.

-o: Outputs only the matched parts of the input lines, not the entire line.

'[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}': This regular expression matches email addresses.

Regular Expression Breakdown:

[a-zA-Z0-9.\_%+-]+: Matches the local part of an email address (before the @ symbol). It includes letters, numbers, dots, underscores, percentage signs, plus signs, and hyphens.

@: Matches the literal @ symbol.

[a-zA-Z0-9.-]+: Matches the domain part of the email (e.g., example.com). It includes letters, numbers, dots, and hyphens.

\.[a-zA-Z]{2,}: Matches the top-level domain (e.g., .com, .org) with at least two letters.

How It Works:

TShark reads and outputs all packet details in verbose format.

The output is searched by grep for any text that matches the email address pattern.

Any email addresses found in the packet data are extracted and displayed.

**Tshark Features**

TShark is the command-line version of Wireshark, sharing many of its features such as display filters and packet analysis capabilities. Below are some key TShark queries and their purposes for reference:

tshark -r <file.pcap> --colo

tshark -z help

- Use the `-q` parameter to suppress packets and focus on statistics only

tshark -r <file.pcap> -z io,phs -q

tshark -r <file.pcap> -z io,phs,udp -q

tshark -r <file.pcap> -z plen,tree -q

**Filters**

**These filters can filter all related queries from the file**

tshark -r directory-curiosity.pcap -T fields -e dns.qry.name |awk NF|uniq -c

tshark -r demo.pcapng -T fields -e ip.src -e ip.dst -E header=y -c 5

user@ubuntu$ tshark -r demo.pcapng -Y 'http.server contains "Apache"'

user@ubuntu$ tshark -r demo.pcapng -Y 'http.request.method matches "(GET|POST)"'

user@ubuntu$ tshark -r demo.pcapng -Y 'http.server contains "Apache"' -T fields -e ip.src -e

ip.dst -e http.server -E header=y

tshark -r directory-curiosity.pcap -T fields -e http.request.full\_uri -e ip.dst |awk NF|uniq -c|grep "jx2-bavuong.com"

**Extract info**

tshark -r directory-curiosity.pcap -z expert -q