tshark Packet analyzer CLI

CONCEPTS

Only capture on networks and hosts you are authorised to monitor. Packet data can contain passwords and personal data; store and share captures with care.

Capture -i, -f filter
↓ -w
pcapng file raw packets
↓ -r
Display filter -Y
↓ -T / -z
Output text, fields, JSON, stats

Two filter languages: -f is a capture filter (BPF, same as tcpdump, applied while capturing); -Y is a display filter (Wireshark fields like ip.addr, applied after dissection).

Live capture usually needs root or the dumpcap capability; a trailing filter argument is a capture filter live, but a display filter with -r.

CAPTURE

`tshark -D`
List capture interfaces

`tshark -i <iface>`
Capture live on an interface

`tshark -i <iface> -c <n>`
Stop after n packets

`tshark -i <iface> -f "<filter>"`
Capture filter (BPF syntax)

`tshark -i any`
Capture on all interfaces

`tshark -i <iface> -s <len>`
Limit bytes kept per packet

FILES AND STOPPING

`tshark -i <iface> -w <file>`
Write raw packets to a file

`tshark -r <file>`
Read a capture file

`tshark -i <iface> -a duration:<s>`
Stop after n seconds

FILES AND STOPPING (cont.)

`tshark -i <iface> -a filesize:<kB>`
Stop at file size

`tshark -i <iface> -b filesize:<kB> -b files:<n>`
Ring buffer of files

`tshark -i <iface> -b duration:<s>`
Start a new file every n seconds

`tshark -r <file> -w <out> -Y "<filter>"`
Save a filtered subset

DISPLAY FILTERS

`-Y "ip.addr == <ip>"`
Packets to or from a host

`-Y "ip.src == <ip>"`
Packets from a source

`-Y "tcp.port == <n>"`
TCP port, either side

`-Y "udp.port == <n>"`
UDP port, either side

`-Y "http.request"`
HTTP requests

`-Y "http.request.method == \"POST\""`
HTTP POST requests

`-Y "dns.qry.name contains \"<name>\""`
DNS queries by name

`-Y "tcp.flags.syn == 1 && tcp.flags.ack == 0"`
New TCP connections

`-Y "tcp.analysis.retransmission"`
Retransmitted segments

`-Y "tcp.analysis.flags"`
Any TCP problem flagged

`-Y "tls.handshake.type == 1"`
TLS Client Hello

`-Y "<a> && !(<b>)"`
Combine with and, or, not

`-Y "ip.addr == <net>/24"`
Match a whole subnet

OUTPUT

`tshark -r <file> -V`
Full packet detail

OUTPUT (cont.)

`tshark -r <file> -n`
No name resolution

`tshark -r <file> -T json`
Dump packets as JSON

`tshark -d tcp.port==<n>,<proto>`
Decode a port as a protocol

`tshark -o tls.keylog_file:<file>`
Decrypt TLS with a key log

EXTRACT FIELDS

`tshark -r <file> -T fields -e <field>`
Print chosen fields only

`-T fields -e ip.src -e ip.dst -e tcp.dstport`
Several fields, tab separated

`-T fields -E header=y`
Add a header row

`-T fields -E separator=,`
Comma separated output

STATISTICS

`tshark -r <file> -q -z io,phs`
Protocol hierarchy

`tshark -r <file> -q -z conv,tcp`
TCP conversations

`tshark -r <file> -q -z endpoints,ip`
IP endpoints

`tshark -r <file> -q -z io,stat,<s>`
Packets and bytes per interval

`tshark -r <file> -q -z expert`
Expert info summary

STREAMS AND OBJECTS

`tshark -r <file> -q -z follow,tcp,ascii,<n>`
Follow TCP stream n as text

`-Y "tcp.stream == <n>"`
All packets of one TCP stream

`tshark -r <file> --export-objects http,<dir>`
Save files from HTTP traffic