tcpdump and wireshark

Day 22/365

Summary of the command line options

Command	Explanation
`tcpdump -i INTERFACE`	Captures packets on a specific network interface
`tcpdump -w FILE`	Writes captured packets to a file
`tcpdump -r FILE`	Reads captured packets from a file
`tcpdump -c COUNT`	Captures a specific number of packets
`tcpdump -n`	Don't resolve IP addresses
`tcpdump -nn`	Don't resolve IP addresses and don't resolve protocol numbers
`tcpdump -v`	Verbose display; verbosity can be increased with `-vv` and `-vvv`

Consider the following examples:

- `tcpdump -i eth0 -c 50 -v` captures and displays 50 packets by listening on the `eth0` interface, which is a wired Ethernet, and displays them verbosely.
- `tcpdump -i wlo1 -w data.pcap` captures packets by listening on the `wlo1` interface (the WiFi interface) and writes the packets to `data.pcap`. It will continue till the user interrupts the capture by pressing CTRL-C.
- `tcpdump -i any -nn` captures packets on all interfaces and displays them on screen without domain name or protocol resolution.

Command	Explanation
`tcpdump host IP` or `tcpdump host HOSTNAME`	Filters packets by IP address or hostname
`tcpdump src host IP` or	Filters packets by a specific source host
`tcpdump dst host IP`	Filters packets by a specific destination host
`tcpdump port PORT_NUMBER`	Filters packets by port number
`tcpdump src port PORT_NUMBER`	Filters packets by the specified source port number
`tcpdump dst port PORT_NUMBER`	Filters packets by the specified destination port number
`tcpdump PROTOCOL`	Filters packets by protocol; examples include `ip`, `ip6`, and `icmp`

Consider the following examples:

- `tcpdump -i any tcp port 22` listens on all interfaces and captures `tcp` packets to or from `port 22`, i.e., SSH traffic.
- `tcpdump -i wlo1 udp port 123` listens on the WiFi network card and filters `udp` traffic to `port 123`, the Network Time Protocol (NTP).
- `tcpdump -i eth0 host example.com and tcp port 443 -w https.pcap` will listen on `eth0`, the wired Ethernet interface and filter traffic exchanged with `example.com` that uses `tcp` and `port 443`. In other words, this command is filtering HTTPS traffic related to `example.com`.

You can use `tcp[tcpflags]` to refer to the TCP flags field. The following TCP flags are available to compare with:

- `tcp-syn` TCP SYN (Synchronize)
- `tcp-ack` TCP ACK (Acknowledge)
- `tcp-fin` TCP FIN (Finish)
- `tcp-rst` TCP RST (Reset)
- `tcp-push` TCP Push

Based on the above, we can write:

- `tcpdump "tcp[tcpflags] == tcp-syn"` to capture TCP packets with **only** the SYN (Synchronize) flag set, while all the other flags are unset.
- `tcpdump "tcp[tcpflags] & tcp-syn != 0"` to capture TCP packets with **at least** the SYN (Synchronize) flag set.
- `tcpdump "tcp[tcpflags] & (tcp-syn|tcp-ack) != 0"` to capture TCP packets with **at least** the SYN (Synchronize) **or** ACK (Acknowledge) flags set.

Command	Explanation
`tcpdump -q`	Quick and quite: brief packet information
`tcpdump -e`	Include MAC addresses
`tcpdump -A`	Print packets as ASCII encoding
`tcpdump -xx`	Display packets in hexadecimal format
`tcpdump -X`	Show packets in both hexadecimal and ASCII formats

Wireshark

A traffic analyser tool
Purposes for its use:

- Detecting and troubleshooting network problems, such as network load failure points and congestion.
- Detecting security anomalies, such as rogue hosts, abnormal port usage, and suspicious traffic.
- Investigating and learning protocol details, such as response codes and payload data.

it is not a Intrusion Detection System, it only allows analysts to discover and investigate the packets in depth, it does not modify them.

Packets description follow the OSI Model