Forensics, Malware and Penetration Testing

Network forensics

David Oswald and Andreea Radu

University of Birmingham

a.i.radu@bham.ac.uk

Outline

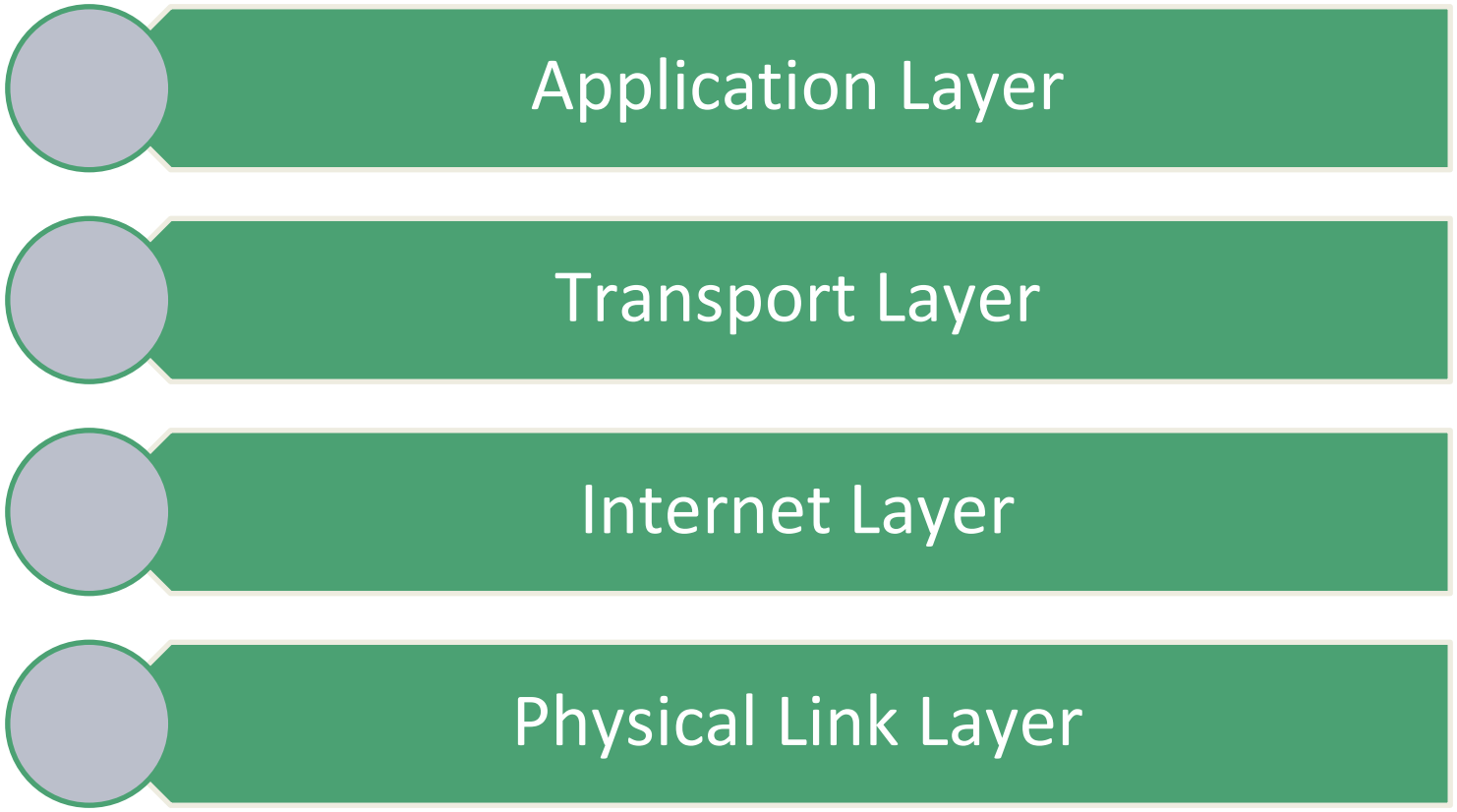
1. Disk forensics* ✓
2. Log file forensics ✓
3. Network forensics ←
4. Memory forensics
5. Mobile devices (Android)

* May need RAM forensics, e.g., in case of full-disk encryption

Why network forensics are important

- Attacks might not leave any traces on the disk (RAM-only malware)
- Attacker might wipe the disk of their target
- Some devices have mostly read-only storage (routers and other embedded devices)
- When the system is powered off, RAM forensics is usually not possible too
- One can see intermediate steps in an attack, not just the final result (on disk/RAM)

The TCP/IP model



Where to capture?

- Physical signals
  - Special hardware needed - network cards usually do not expose the physical layer
- Link Layer (e.g. Ethernet)
  - Access usually possible with standard equipment
  - May lose some information for fingerprinting or attack detection

http://blog.opensecurityresearch.com/2013/03/sniffing-traffic-on-wire-with-hardware.html

Where to capture?

- Internet Layer (e.g. IP)
  - Often sufficient for secure routed networks
  - Still a lot of data to process
- Transport Layer (e.g. TCP / UDP)
  - Less information than IP layer
- Application Layer
  - Many different applications
  - e.g. HTTP(S), SMTP, SSH, FTP, Telnet, ...

What to capture?

WD 16 TB Elements Desktop E
Visit the Western Digital Store
4,971 ratings
Amazon's Choice for "16tb drive"

RRP: £406.99 Details
Price: £299.99 prime Same-Day & FREE Re
or 5 monthly payments with Amazon of £
You Save: £107.00 (26%)

From £26.50 x 12 months with 10.9% APR
Use Instalments at checkout to spread the cost. Sub
Representative example: Credit limit £1200, Annual interest rate 1
Learn more about Instalments by Barclays

May be available at a lower price from other sellers,
Promotion Message 50% off gift wrap with code GIFT
Note: This item is eligible for FREE Click and Collec

- Everything we can
  Large quantity of information (but storage cheap)
- Remove the Data Link Layer part
  Often sufficient for routed networks
- Just the prefix of packets (64 bytes for example) or first few packets of a connection - (Part of) payload missing
- Application Layer filtering, e.g. only HTTP

Which tool to capture?

- Tcpdump
  - Available almost everywhere
  - Supports filtering
  - Does not do state reconstruction
- Dumpcap
  - Supports pcap-ng file format
- Filters are given as strings
  - Compiled to bytecode
  - Transferred to the kernel -> **fast**

Tcpdump and dumpcap filtering

Examples:

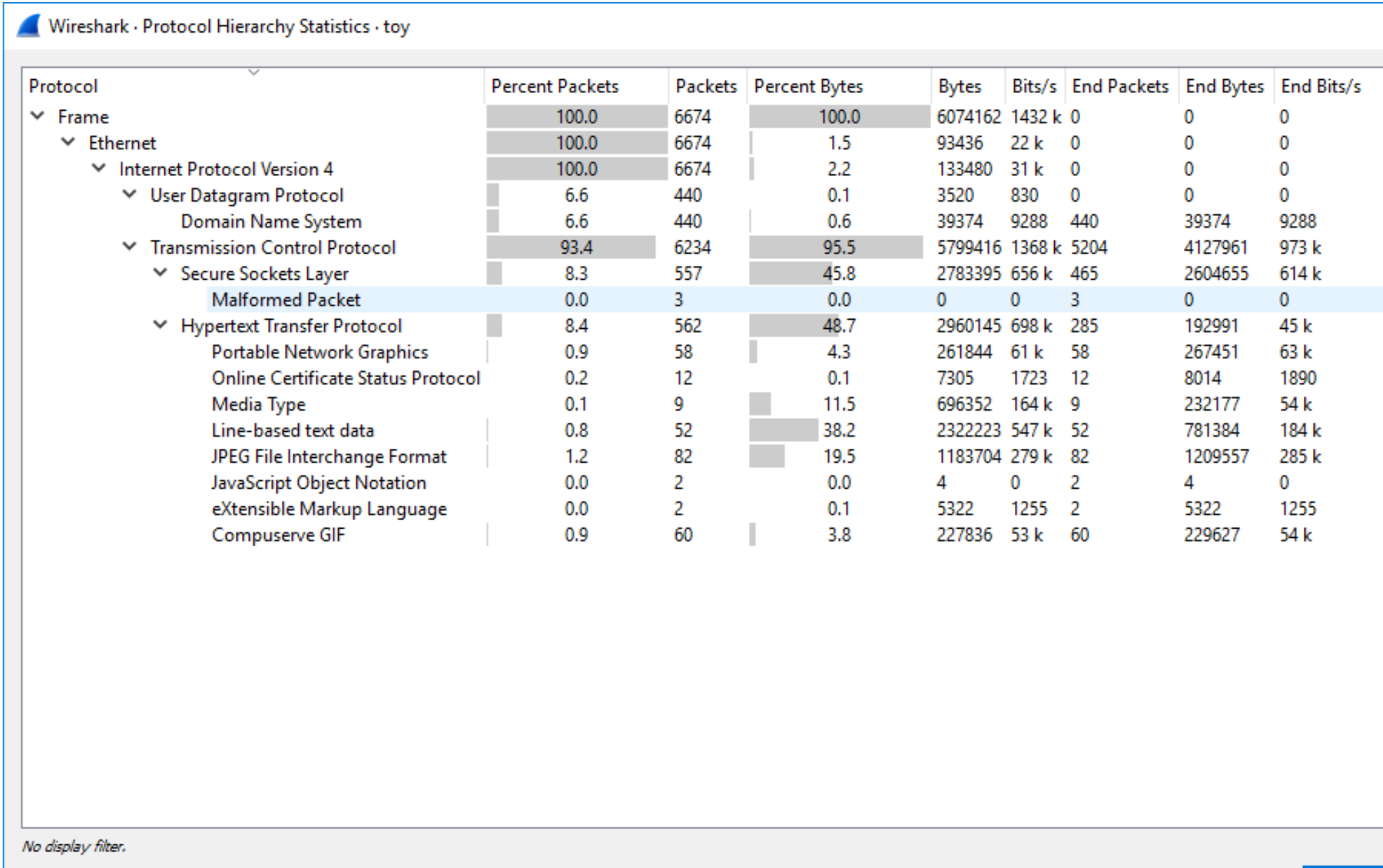
- `host 8.8.8.8`
- `src 8.8.8.8`
- `dst 8.8.8.8`
- `net 8.0.0.0/8`
- `icmp`
- `port 53`
- `udp port 53`
- `tcp port 53`
- `src 8.8.8.8 and not udp src port 53`

Analyzing capture files

Wireshark is often the tool of choice

- Supports many file formats
- Supports amazingly many protocols
- Open source, can easily be extended
- Runs on all major platforms

Protocol overview



Endpoints

Wireshark · Endpoints · toy

Ethernet · 2 | IPv4 · 55 | IPv6 | TCP · 236 | UDP · 120

Address	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Latitude	Longitude
8.26.222.254	23	13 k	11	11 k	12	1613	—	—
10.0.2.15	6,674	6074 k	3,347	381 k	3,327	5692 k	—	—
23.43.63.160	39	7283	18	4294	21	2989	—	—
23.43.75.27	62	12 k	29	9033	33	3621	—	—
23.57.10.43	55	61 k	27	59 k	28	1833	—	—
23.235.43.249	23	14 k	11	12 k	12	1424	—	—
37.157.6.251	27	7696	14	5726	13	1970	—	—
46.228.47.115	8	532	4	240	4	292	—	—
50.31.185.39	20	3629	10	1262	10	2367	—	—
50.31.185.42	40	4372	20	2166	20	2206	—	—
54.228.196.192	26	7851	12	5880	14	1971	—	—
54.228.214.19	29	3499	13	1104	16	2395	—	—
54.231.130.116	63	45 k	31	42 k	32	2887	—	—
54.235.121.3	34	5097	16	1576	18	3521	—	—
54.239.25.192	10	570	5	300	5	270	—	—
54.240.166.143	18	2856	8	1758	10	1098	—	—
62.138.116.15	143	103 k	69	97 k	74	5837	—	—
62.138.116.25	507	537 k	257	492 k	250	45 k	—	—
62.138.116.39	47	9496	23	4576	24	4920	—	—
64.233.166.95	13	1671	6	910	7	761	—	—
78.46.38.211	30	10 k	15	8182	15	1951	—	—
82.199.80.141	22	6038	11	4075	11	1963	—	—
85.114.159.76	210	129 k	105	116 k	105	12 k	—	—

Powerful filters are available in Wireshark

- Wireshark supports different filters
  - **Capture filters** are tcpdump filters
  - **Display filters** are more powerful and internally used in wireshark to process a capture
- Display filters:

  https://wiki.wireshark.org/DisplayFilters
  - http.request
  - tcp or dns
  - tcp.flags.syn == 1

Sometimes Wireshark is too slow

- Pre-filtering of a pcap can be useful
- Tcpdump filters are much faster than Wireshark display filters
- *Example:* Filter with tcpdump on the capture, then use it in Wireshark:

```
/usr/sbin/tcpdump -r all.pcap -w traffic-for-host.pcap "ether host b8:27:eb:de:20:57 or ether multicast"
```

- ... or use tshark (Wireshark on cmdline)

Automated processing with tshark

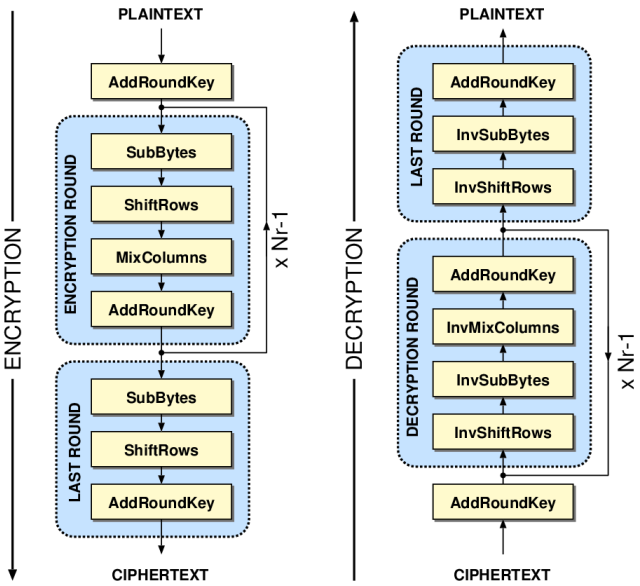
- Manpage: https://www.wireshark.org/docs/man-pages/tshark.html
- Tshark can be used to display specific frames only (as Wireshark display filters)
- *Examples:*

  `tshark -i wlan0 -Y dns` (display filter, slow)

  `tshark -i wlan0 -f "port 53"`

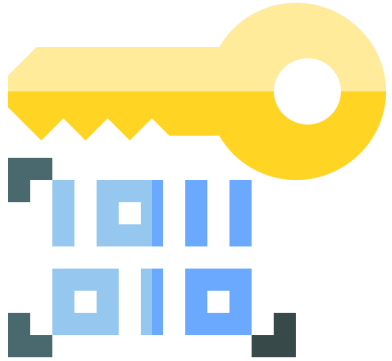
  (capture filter, fast)

What about encrypted traffic?



Encrypted traffic

- Encrypted (or obfuscated) traffic is in general hard to analyse



- But when you know the **key**, it can be in some cases decrypted

Encrypted traffic

- TLS: Not using Perfect Forward Secrecy (PFS) ciphersuites allows decryption with the server key (supported by Wireshark)
- IPSEC: In general same as TLS for no-PFS suites
- WPA: Requires capture of handshake + passphrase

Metadata of encrypted traffic

Encrypted communication still leaks metadata, for example:

- Which host was accessed from where?
  (e.g. which webserver user connected to)
- Time, date, duration
- Amount of exchanged data (approx.)
- Protocol parameters (e.g. supported cipher suites, versions, ...) that may allow to fingerprint a client / server

An alternative

- Use a TLS proxy
  - Accepts the incoming connection
  - Handles the SSL/TLS layer
  - Dispatches the unencrypted connection to an internal server
  - Might act as load balancer and static cache too
  - Logging can be enabled on that host
- For small scaled setups
  - `mitmproxy` is the tool of your choice
  - `stunnel` or `socat` as an alternative

Next part: Memory Forensics