Selina Kallio

selina.kallio@gmail.com

+358 40 703 9171

# Components of IoT Application

# Week 1

1. Define following terms shortly:
    - bps vs Bps
        - bps = bits per second, unit that measures data transmission speed
        - Bps = bytes per second, unit that measures data transmission rate/data storage
    - Protocol payload
        - “Concrete” data part of communication protocol, excluding i.e. metadata and headers
    - Collision domain
        - Network segment where collisions between data packets can occur because devices share the same communication medium and only one device can transmit at a time.
    - Broadcast domain
        - Network area where broadcasted messages are sent for all devices within network domain to receive and process.
    - MAC (physical) address
        - Media Access Control address, a unique identifier assigned to a network interface card (NIC) at the hardware level, used for addressing and identification within Ethernet networks.
    - Half-duplex vs Full-duplex
        - Half-duplex = Communication where data can be transmitted in both directions, but not simultaneously.
        - Full-duplex: Communication where data can be transmitted in both directions simultaneously.
    - Ethernet auto-negotiation
        - A feature in Ethernet devices that allows them to automatically determine and configure their link speed, duplex mode, and other parameters during the initial connection.
    - Hidden node problem (wireless)
        - A situation in wireless networks where two nodes are out of each other's transmission range but can still interfere with a third node within range, causing communication issues.
    - Physical topology vs logical topology
        - Physical Topology = The physical layout (HW layout) of network devices and cables.

- Logical Topology = The way data is transmitted logically between devices in a network, regardless of the physical layout.

- "Plain text"-protocols
  - Protocols that transmit data without encryption or any form of security, making the data vulnerable to interception and unauthorized access among other cybersecurity vulnerabilities.
- Repeater
  - A network device used to extend the reach of a network by regenerating and amplifying signals, helping overcome signal degradation (weakening) over long distances.
- Hub (multiport repeater)
  - Network device operating similarly to a repeater, but it broadcasts incoming data to all connected devices, resulting in increased collisions and reduced efficiency compared to switches.
- Bridge (bridging operation)
  - A device or function that connects and filters traffic between multiple network segments, operating at the data link layer to divide collision domains.
- Router
  - A network device that connects different networks and forwards data packets between them.
- Protocol overhead (ATM is famous on this)
  - Extra data added to a transmitted packet by a protocol, including headers and control information, which can reduce the overall efficiency of data transmission but make it less susceptible to cybersecurity vulnerabilities.
- TIA/EIA-568 and ISO/IEC_11801
  - Standards for the design and installation of cabling systems within structured cabling networks, ensuring compatibility and performance.
- Packet loss and jitter
  - Packet Loss = The failure of one or more data packets to reach their destination in a network.
  - Jitter = Variability in packet arrival times, causing irregular delays and affecting real-time applications i.e. online video games.
- OSI model and TCP/IP model
  - OSI Model: Open Systems Interconnection model, a conceptual framework used to understand how network protocols interact in seven layers.

- TCP/IP Model: A practical networking model that encompasses the four-layer Transmission Control Protocol/Internet Protocol suite used for most internet communication.

2. RFC tasks
    - What are RFCs?
        - "Request for Comments" provided by the IETF, it sets a baseline for networking procedures and more.
        - Documents describing aspects of computer networking i.e. protocols, procedures and programs. RFCs serve as the standards, guidelines, and informational documents for the development and implementation of internet technologies.
    - How many PPP related RFC documents can you find from rfc-editor website?
        - 111 (RFC Search Detail (rfc-editor.org)) ranging from 1990 to 2021
    - What is the current status of RFC1597? What is the updated RFC number?
        - RFC1597 is obsoleted and replaced by RFC 1918.
    - When was RFC9000 released?
        - May 28th, 2021
    - What is RFC BCP?
        - BCP = "Best Current Practice"; documents providing guidelines and recommendations for operational practices in networking and the internet.
        - RFC BCP therefore means RFCs that include current best practices.
    - List the authors of Post Office Protocol - Version 3 (POP3) RFC
        - John G. Myers and Marshall T. Rose
    - What is the RFC number for first OSPF version?
        - OSPF = Open Shortest Path First
        - RFC1131 "OSFP Version 2" published in October 1989 but obsoleted later.

# Week 2

1. Define following terms and concepts shortly:
    - ARP
        - Networking protocol used to map an IP address to a physical MAC address on a local network.
    - IP TTL
        - A field in the IP header that specifies the maximum number of network hops a packet can take before being discarded to prevent indefinite looping.
    - Traceroute / Tracepath
        - Network diagnostic tools used to trace the route that data packets take from one host to another, showing the intermediate hops and round-trip times.
    - Default gateway (default routing)
        - The router or device to which network traffic is sent when no specific route entry exists in a routing table to reach a destination network.
    - Static routing
        - A routing method in which network administrators manually configure the routes in routers, making routing decisions based on predetermined paths.
    - Dynamic routing
        - A routing method in which routers exchange information to automatically determine optimal routes based on changing network conditions.
    - OSPF, IS-IS, BGP, RPL (ripple)
        - OSPF (Open Shortest Path First) = A link-state routing protocol used to determine the best path for routing data in IP networks.
        - IS-IS (Intermediate System to Intermediate System) = A link-state routing protocol similar to OSPF, often used in larger networks.
        - BGP (Border Gateway Protocol) = A path vector protocol used to exchange routing and reachability information between autonomous systems on the internet.
        - RPL (Routing Protocol for Low-Power and Lossy Networks) = A routing protocol designed for resource-constrained IoT networks.
2. Try traceroute (or tracert in MS Windows command prompt) to few internet web servers

```
C:\Users\skallio>tracert 10.34.39.154

Tracing route to 10.34.39.154 over a maximum of 30 hops

  1     4 ms     3 ms     3 ms  10.145.192.1
  2     3 ms     2 ms     1 ms  172.30.67.9
  3     4 ms     2 ms     2 ms  172.30.67.18
  4     3 ms     2 ms     2 ms  10.0.15.140
  5     4 ms     3 ms     2 ms  192.168.5.84
  6  2026 ms     *        *     192.168.5.67
  7     *        *        *     Request timed out.
  8     *      588 ms     *     10.34.128.129
  9     *        *        *     Request timed out.
 10   574 ms     *      594 ms  10.34.128.129
 11     *        *
```

^ My test server here at Nokia, only accessible via VPN and authentication.

```
C:\Users\skallio>tracert https://tl.oamk.fi/iot/
Unable to resolve target system name https://tl.oamk.fi/iot/.

C:\Users\skallio>ping https://moodle.oulu.fi/
Ping request could not find host https://moodle.oulu.fi/. Please check the name and try agair

C:\Users\skallio>nslookup https://moodle.oulu.fi/
Server:  fihedc001.emea.nsn-net.net
Address:  10.158.51.11

*** fihedc001.emea.nsn-net.net can't find https://moodle.oulu.fi/: Non-existent domain

C:\Users\skallio>nslookup https://www.google.com/
Server:  fihedc001.emea.nsn-net.net
Address:  10.158.51.11

*** fihedc001.emea.nsn-net.net can't find https://www.google.com/: Non-existent domain
```

^ Trying nslookup/ping since unable to reach “outside” servers from command prompt

3. List all private IPv4 networks (from RFC1918)
    - 10.0.0.0 to 10.255.255.255
    - 172.16.0.0 to 172.31.255.255
    - 192.168.0.0 to 192.168.255.255
4. What is the purpose of IPv4 private networks?
    - Private IPv4 networks are reserved for internal use within organizations and are not routable on the public internet. They enable organizations to use private addresses internally and conserve public IP addresses.
5. What are the ranges for IPv4 multicast and experimental addresses?
    - Multicast: 224.0.0.0 to 239.255.255.255
    - Experimental: 240.0.0.0 to 255.255.255.254
6. What is the 127.0.0.1 address?

- The loopback address in IPv4, used to test network connectivity on the local host (your own computer).
- Also accessible with “localhost”, 127.0.0.1 is known as “home”


7. What is the ::1 address?
   - The loopback address in IPv6, equivalent to 127.0.0.1 in IPv4.
8. What are the networks 0.0.0.0/0 and ::/0?
   - These represent default routes, indicating all possible addresses in IPv4 and IPv6. They are used when a router doesn't have a more specific route for a destination.
9. List and explain three or more purposes and features of the ICMP protocol
   - Network Testing: ICMP includes tools like "ping" and "traceroute" for diagnosing network connectivity and routing issues.
   - Error Reporting: ICMP messages inform hosts about network errors, helping troubleshoot issues.
   - Time Exceeded: Used to indicate that a packet's TTL has expired, helping to identify network loops or routing problems.
   - Echo Request/Reply (Ping): ICMP's echo messages can test if a host is reachable and measure round-trip time.
10. Try to solve these IP subnetting tasks without checking the solutions and document at least some examples/answers to the learning diary. Answers should contain (for each subnet): Network address, broadcast address and subnet mask:
    - Subnetting task 1:
      - The address space available is 172.16.64.0/23. Subnet it and create 5 (A, B, C, D and E) IPv4 subnets with following amount of hosts in each network: A = 85, B = 45, C = 95, D = 57, E = 34.
        - This address space contains 510 hosts, ranging from 172.16.64.1 to 172.16.65.254
        - For the host amounts the following masks must be used:
          A. 85: 25 is the smallest mask that fits all, hence we can use 172.16.64.1/25
          B. 45: needs bitmask of 26 to allocate 45 hosts: 172.16.64.128/26
          C. 172.16.64.255/25

D. 172.16.65.0/26

E. 172.16.65.65/26

- Subnetting task 2:
  - Same as task 1, but available address space is now 192.168.0.0/25 and networks/hosts are: A = 28, B = 10, C = 60, D = 4.
    - Contains 126 IPs ranging from 192.138.0.1 to 192 168.0.126
    - To allocate all 4 networks:
      - Bitmask 27 contains 30 hosts, making it a great choice for network A, 192.168.0.0/27
      - 192.168.0.32/28
      - 192.168.0.64/26
      - 192.168.0.128/29
- Subnetting task 3:
  - IPv6 address space available: 2001:708:510::/48. Create four /64 IPv6 networks.
    - This includes 65536 hosts in range of
      - 2001:0708:0510:0000:0000:0000:0000:0000
      - 2001:0708:0510:ffff:ffff:ffff:ffff:ffff
    - To create 4 /64 IPv6 networks inside it:
      - 2001:0708:510:::::/64
      - 2001:0708:510:1::::/64
      - 2001:0708:510:2::::/64
      - 2001:0708:510:3::::/64

# Week 3

1. Use Linux or Windows command line telnet or any other TCP socket client application (install telnet client if missing) to access a TCP service in pouta.ipt.oamk.fi listening TCP port 55555. What the server replied to your TCP connection if you send some text string endline newline?
2. Answer these questions:
    - What is TCP SYN bit?
        - The TCP SYN (Synchronize) bit is a flag in the TCP header used during the initial phase of establishing a TCP connection. It's set in the TCP packet to initiate the connection establishment process.
    - Explain shortly what are TCP acknowledgment and sequence numbers
        - Acknowledgment Number = This is the number sent by the receiver to acknowledge receipt of data. It indicates the next byte of data the sender expects to receive.
        - Sequence Number = This is the number assigned to each byte of data transmitted. It helps in ordering and reassembling data on the receiving end.
    - Explain TCP connection states: What is LISTENING? What is ESTABLISHED?
        - LISTENING: The server is ready to receive incoming connection requests.
        - ESTABLISHED: The TCP connection has been successfully established, and data can be exchanged between the client and server.
    - What is the purpose of TCP (or UDP) source port?
        - The source port helps identify the originating process or application on the sender's side.
    - What is the purpose of TCP (or UDP) destination port?
        - The destination port helps identify the intended process or application on the receiver's side.
    - What are the common services for TCP ports: 22, 23, 25, 80, 443, 445, 3306?
        - 22: SSH (Secure Shell), most common way to access host remotely (in my experience, at least)
        - 23: Telnet
        - 25: SMTP (Simple Mail Transfer Protocol)
        - 80: HTTP (Hypertext Transfer Protocol), 80 and 8080 are ways to access from internet interface, not CLI
        - 443: HTTPS (Hypertext Transfer Protocol Secure)
        - 445: Microsoft-DS (Microsoft Directory Services)

- 3306: MySQL

- Study available options with command line command “netstat /?” (windows) or netstat –help (linux). What different things you can see with netstat command?
  - The netstat command provides information about network connections, routing tables, interface statistics, masquerade connections, etc. Using the command with appropriate options can display active connections, listening ports, routing information, etc.
  - In my cybersecurity-related work I use netstat commonly to listen to ports and see if unnecessary ports are being “listened” to
- Why UDP is “connectionless” protocol?
  - UDP is "connectionless" because it does not establish a formal connection before sending data. Each UDP packet is independent and can be sent without prior communication.
- Why UDP lacks flow-control features?
  - UDP lacks flow-control mechanisms because it's designed for applications where speed and low overhead are more important than reliable delivery. Applications using UDP need to implement their own error checking and handling mechanisms.
- Why most services using UDP prefer max 512 byte UDP datagrams?
  - Many networking devices have limited buffer sizes, and keeping UDP datagrams small reduces the risk of packet fragmentation and loss.
- When it is more reasonable to use UDP instead of TCP?
  - UDP when speed and low overhead are critical, and data integrity can be traded off for performance. Uses like streaming media, online gaming, and real-time communication often use UDP.
- What is the length of TCP header without extra options? What about UDP header?
  - The TCP header length without extra options is typically 20 bytes.
  - The UDP header length is 8 bytes.
- What is TCP Nagle’s algorithm? When it should be disabled for networking applications?
  - Nagle's algorithm is used in TCP to reduce network congestion by delaying the transmission of small packets. It should be disabled for applications requiring low latency, such as real-time applications (e.g. video calls).
- Why are some applications using or offer “keepalive” mechanisms over TCP or UDP to maintain established connection (for example SSH)?
  - Keepalive mechanisms are critical in ensuring that the connection is still active even when there's no data being transmitted but so that the connection may stay open for future actions. For example in my own experience keeping an SSH connection open provides the possibility to monitor scripts running and make future references to the PC easier.

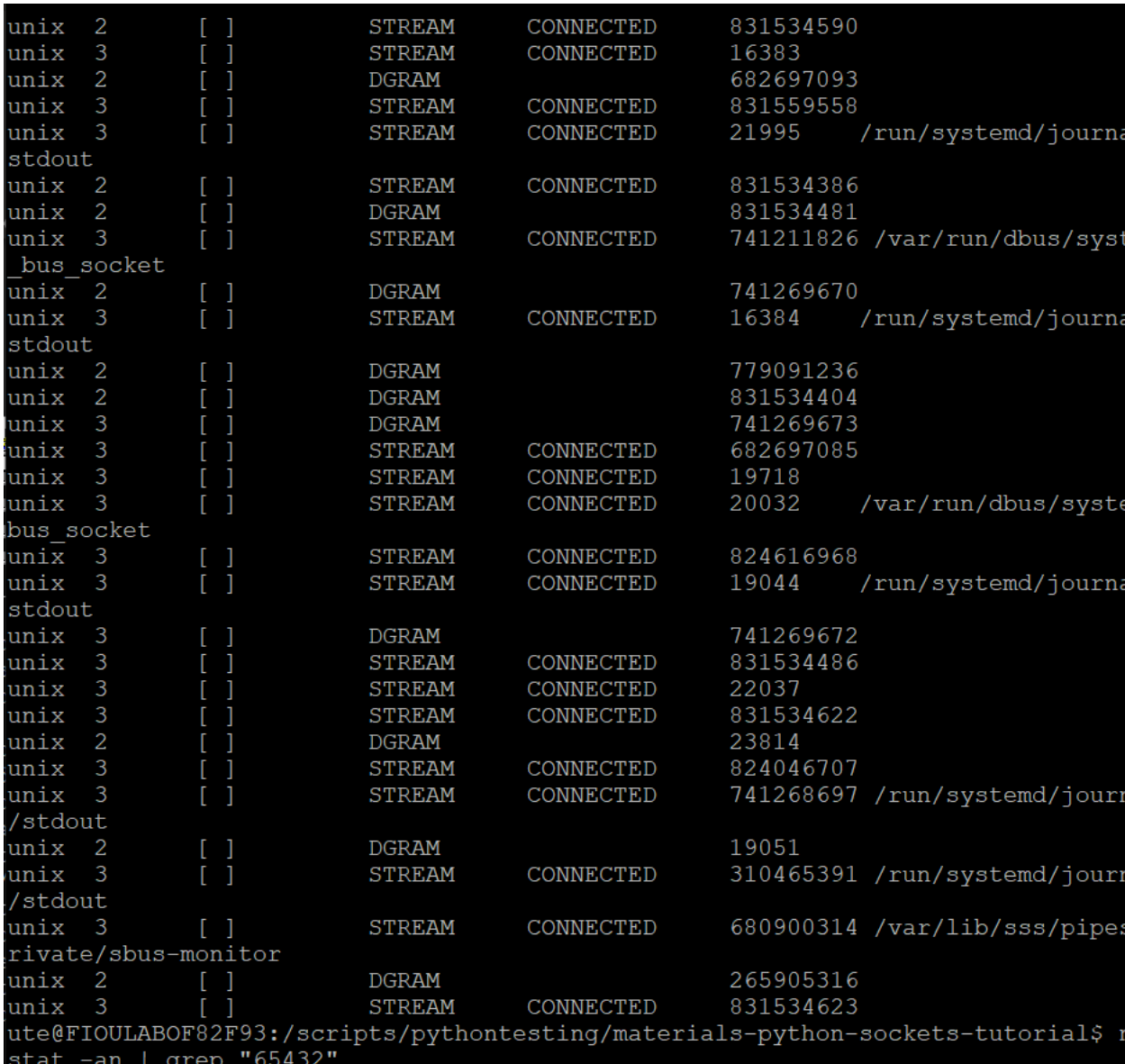
- What is a raw socket?
    - A raw socket allows direct access to network packets at the protocol level, bypassing the TCP/IP stack. It's often used for network monitoring, packet manipulation, and certain specialized applications.

# Week 4

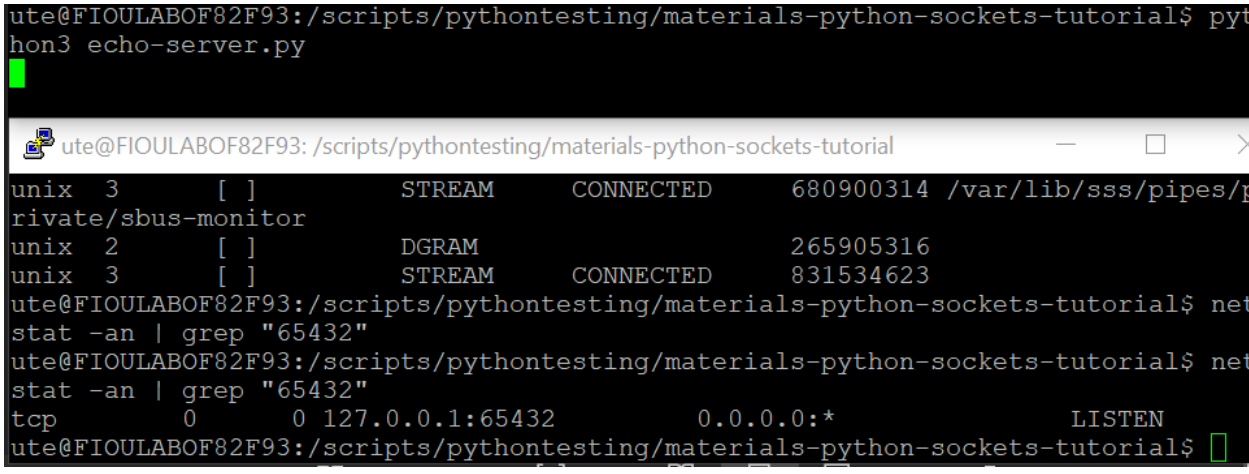
15. Do these Python programming tasks with Windows or Linux (or with MacOS if you want and know how)
    - For example, use https://realpython.com/python-sockets/ or similar site(s) for socket programming example codes and create TCP client and TCP server Python scripts
    - Establish a TCP connection between your client and server Python scripts (either as localhost traffic or between two separate hosts if you have access to two or more Python running hosts)
    - Transfer some ASCII text strings between the hosts: TCP client connects to the server, sends a text string and then disconnects. Server prints the text. Save your source codes and work. You need it again during the course week #5 (Wireshark protocol analyzer tasks)


(shown in picture above, code from realpython.com)

    - Use netstat or similar command line tools to check the TCP connection status (for example the server listening the selected TCP port)




^ Before putting echo-server.py to run. Nothing running on 127.0.0.1/65432



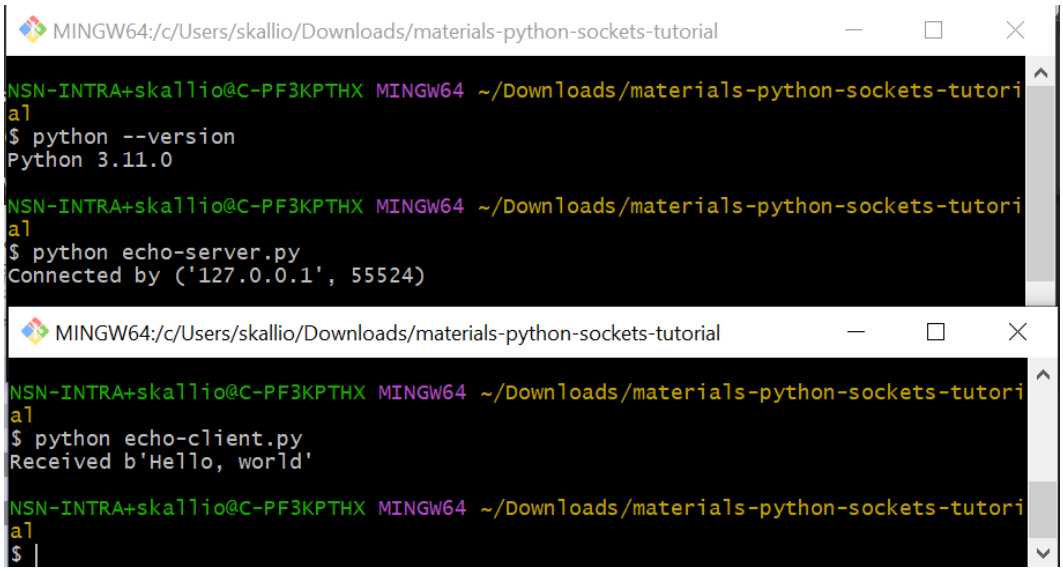

^ After putting echo-server.py to run, there is activity on localhost:65432

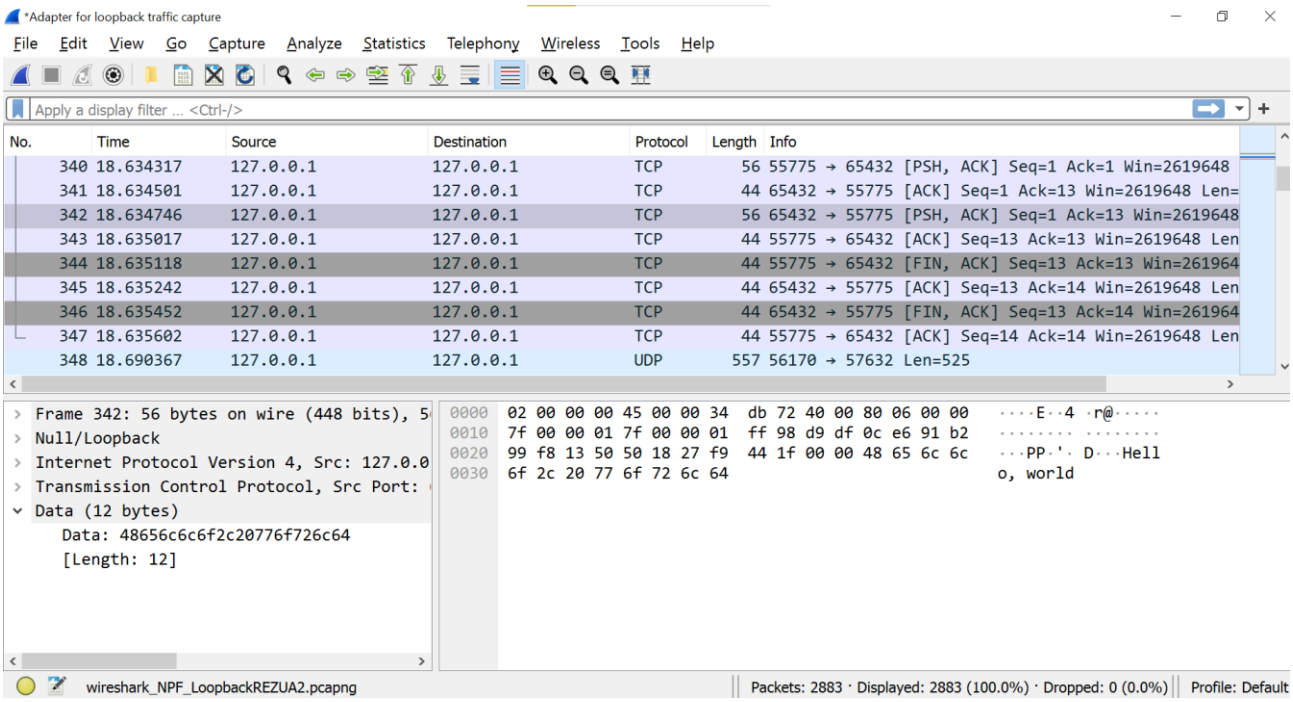
# Week 5

Wireshark tasks:

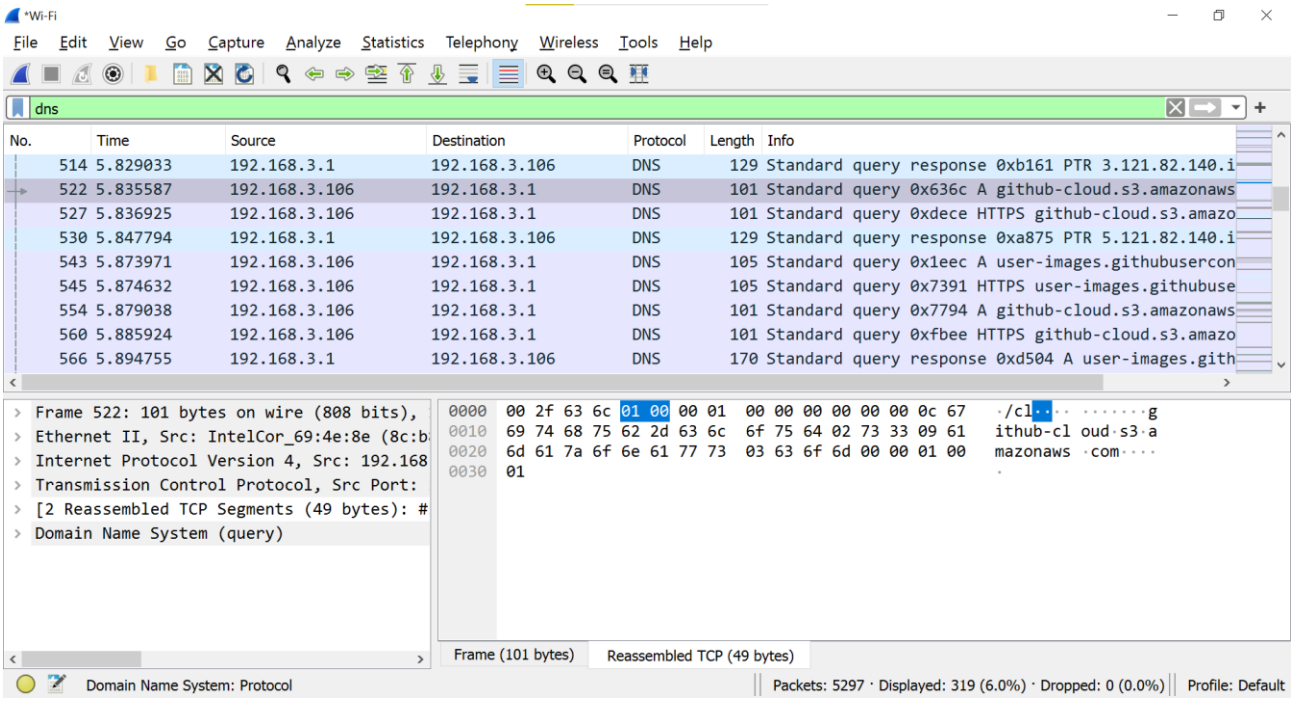
16. Install [Wireshark protocol analyzer] and inspect your IP traffic (DNS requests, web browsing and such)

    - With Wireshark: Analyze the plain text traffic between the socket applications you did during the course week #4. Note: use localhost network interface when capturing host internal traffic (localhost/127.0.0.1)



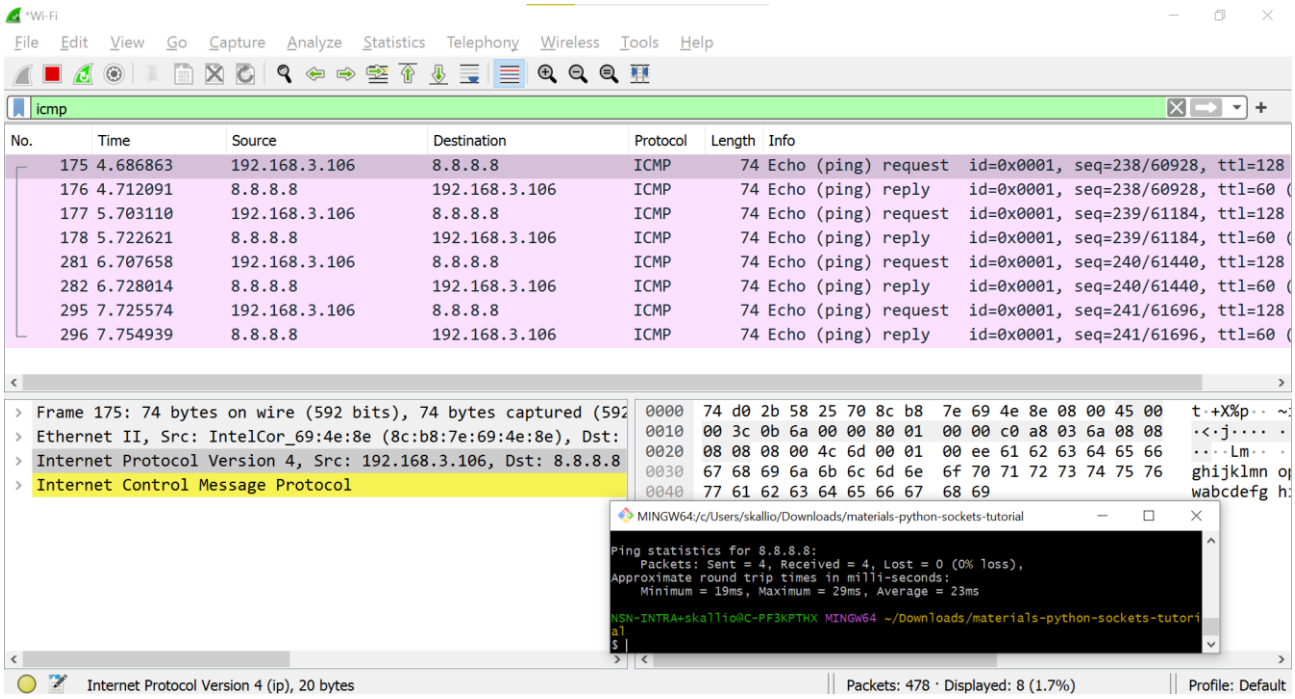



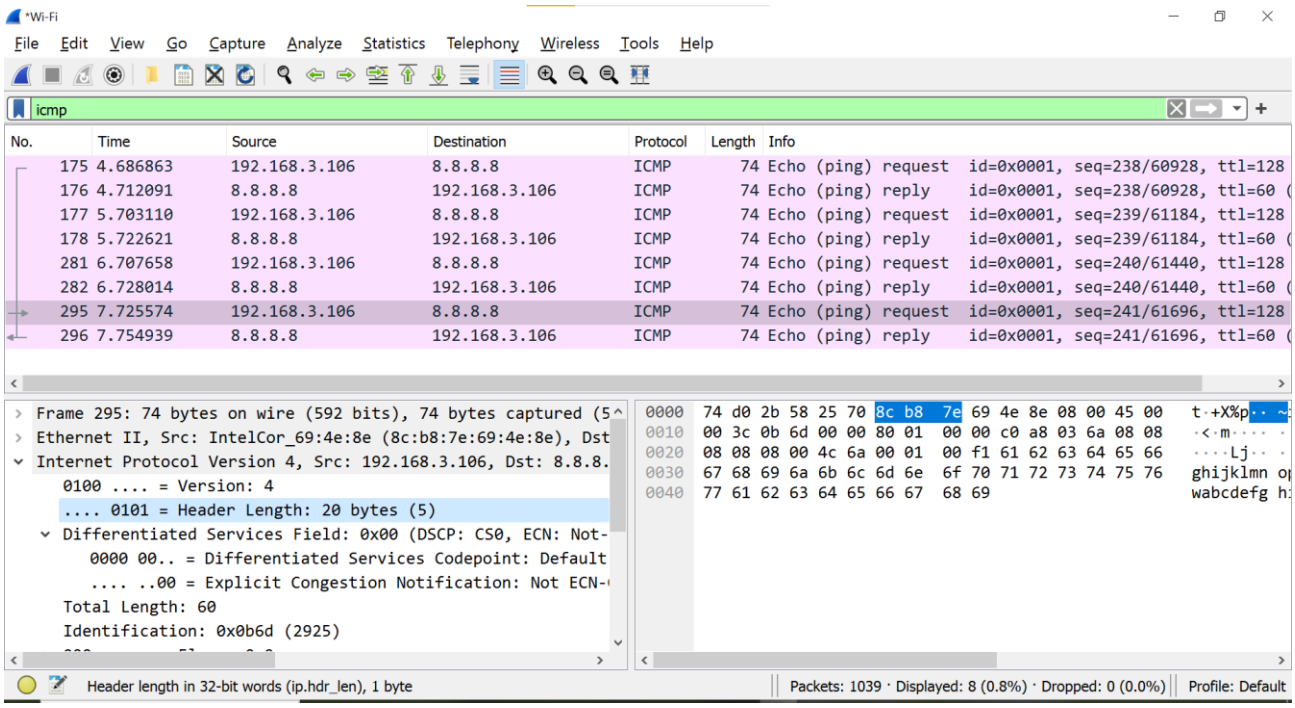

    - With Wireshark: Capture some web browsing traffic and related DNS requests. What are those A (and maybe AAAA requests)? What protocols are used for DNS requests?




- A Requests = DNS (Domain Name System) requests for IPv4 addresses, convert human-readable domain names (e.g., www.example.com) into IP addresses (e.g., 192.168.3.106 in photo).
- AAAA Requests = DNS requests for IPv6 addresses. They perform the same function as A requests but for IPv6 addresses, which are represented as a series of hexadecimal numbers separated by colons (e.g., 2001:0db8:85a7:0000:0000:8b4f:7ca3:7294).
- DNS protocols: DNS operates over both UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). UDP is commonly used for standard DNS queries, while TCP is used for large responses or zone transfers.

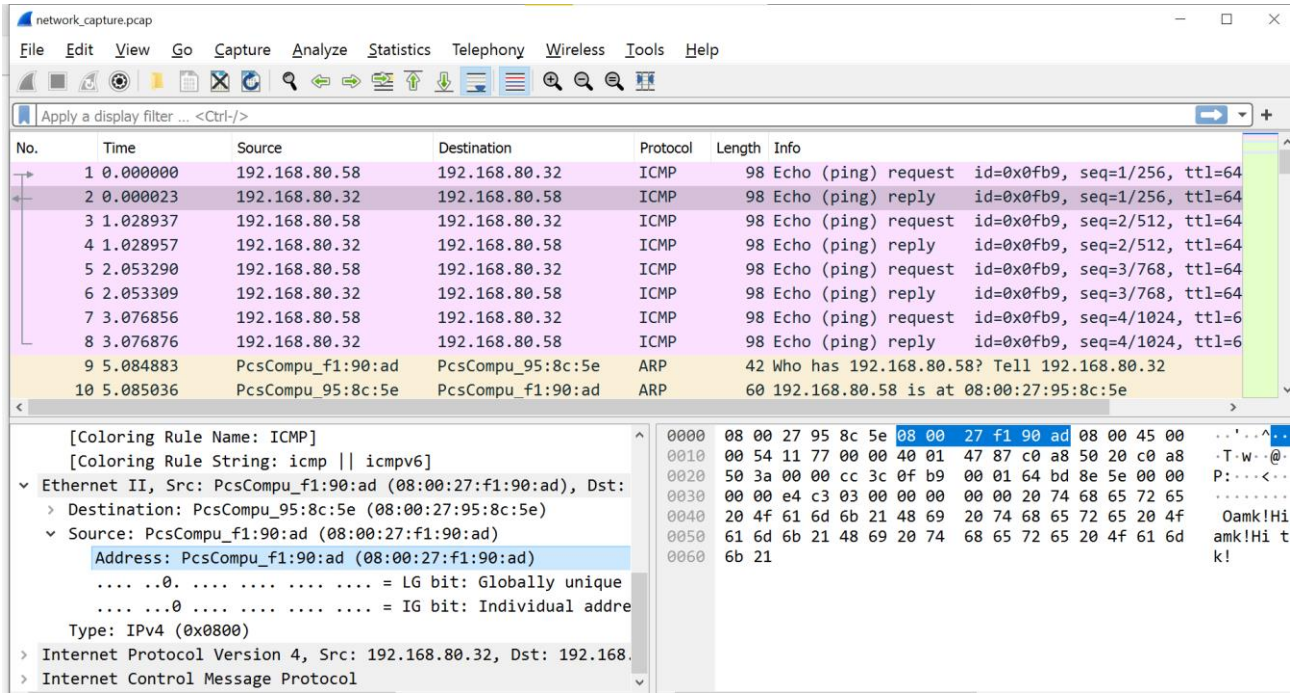
- With Wireshark: Try to ping 8.8.8.8 from command prompt and capture the traffic. What protocols ping was using? What is the total header length of your ping request (all used protocols combined)?




(header length 20 bytes)

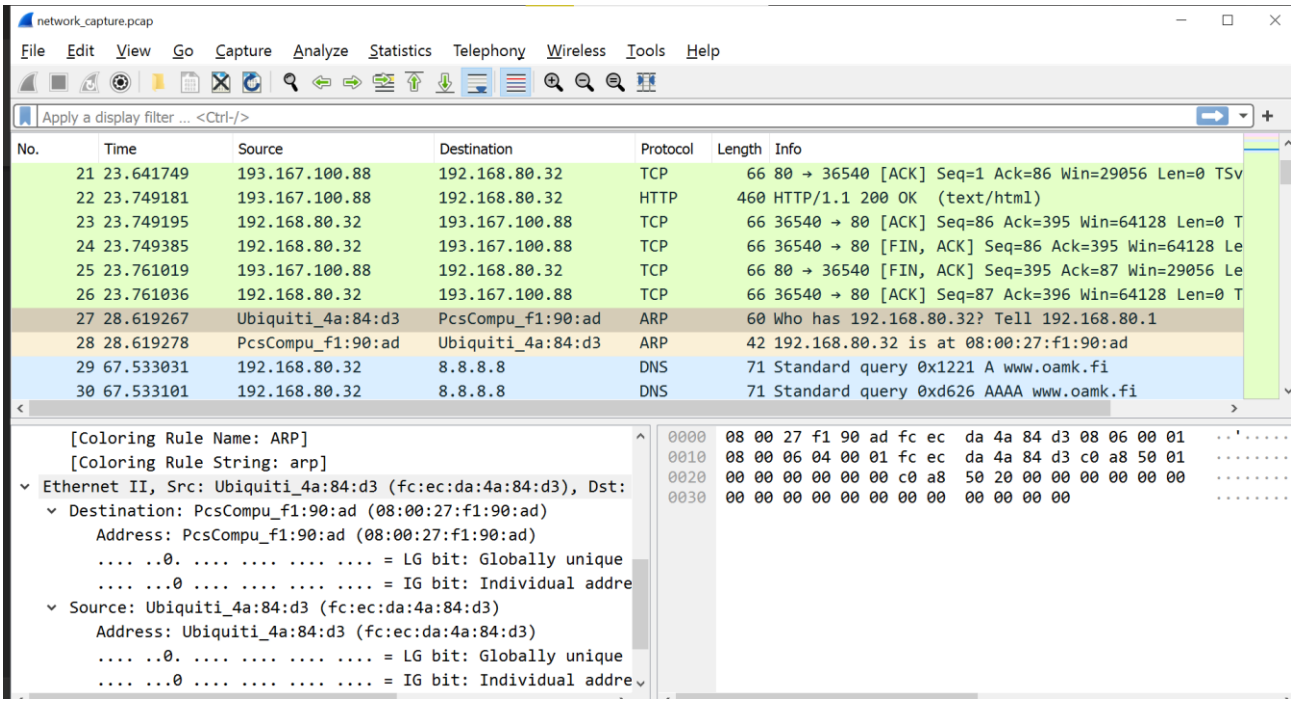
17. Download this zipped pcap traffic file and analyze it with Wireshark. Traffic has been captured from host 192.168.80.32. Answer these questions:
    - What is the MAC address of host 192.168.80.32?




MAC is 08-00-27-f1-90-ad

- What is the MAC address of host 192.168.80.1? Which vendor has built the ethernet chipset of host 192.168.80.1 (use Wireshark or IEEE OUI data)?

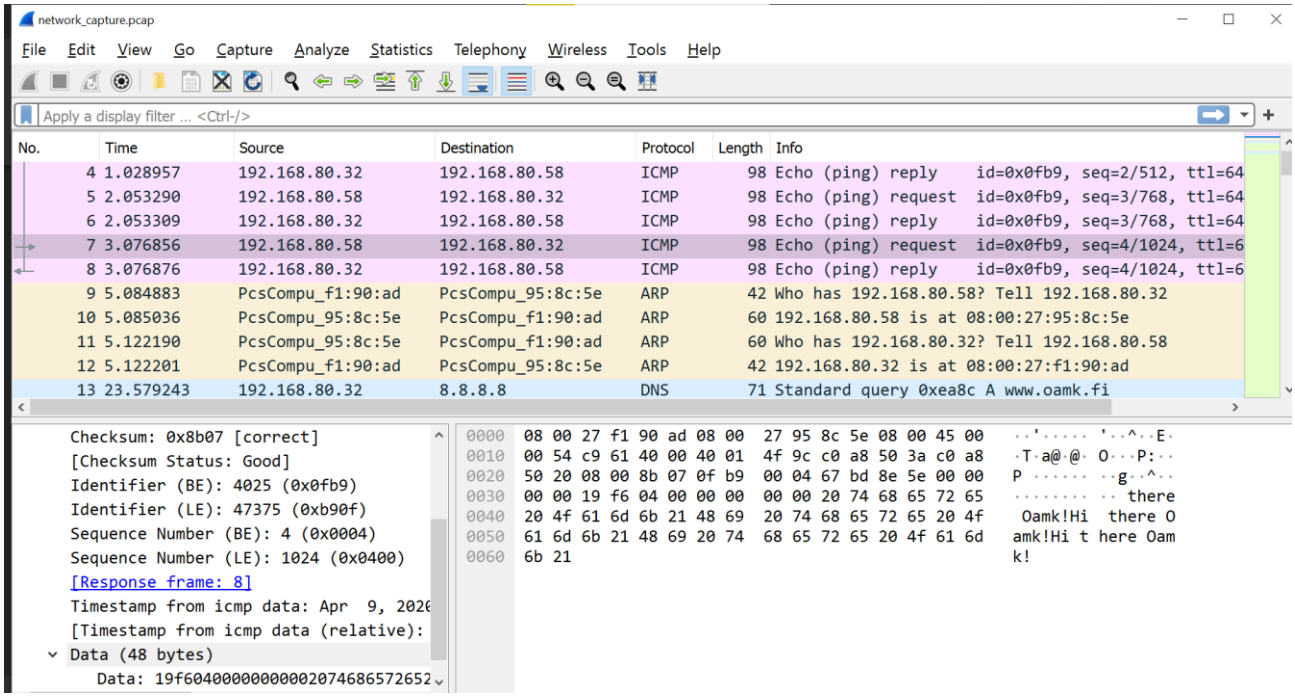
FC-EC-DA (hex) Ubiquiti Inc
FCECDA (base 16) Ubiquiti Inc
685 Third Avenue, 27th Floor
New York NY New York NY 10017
US



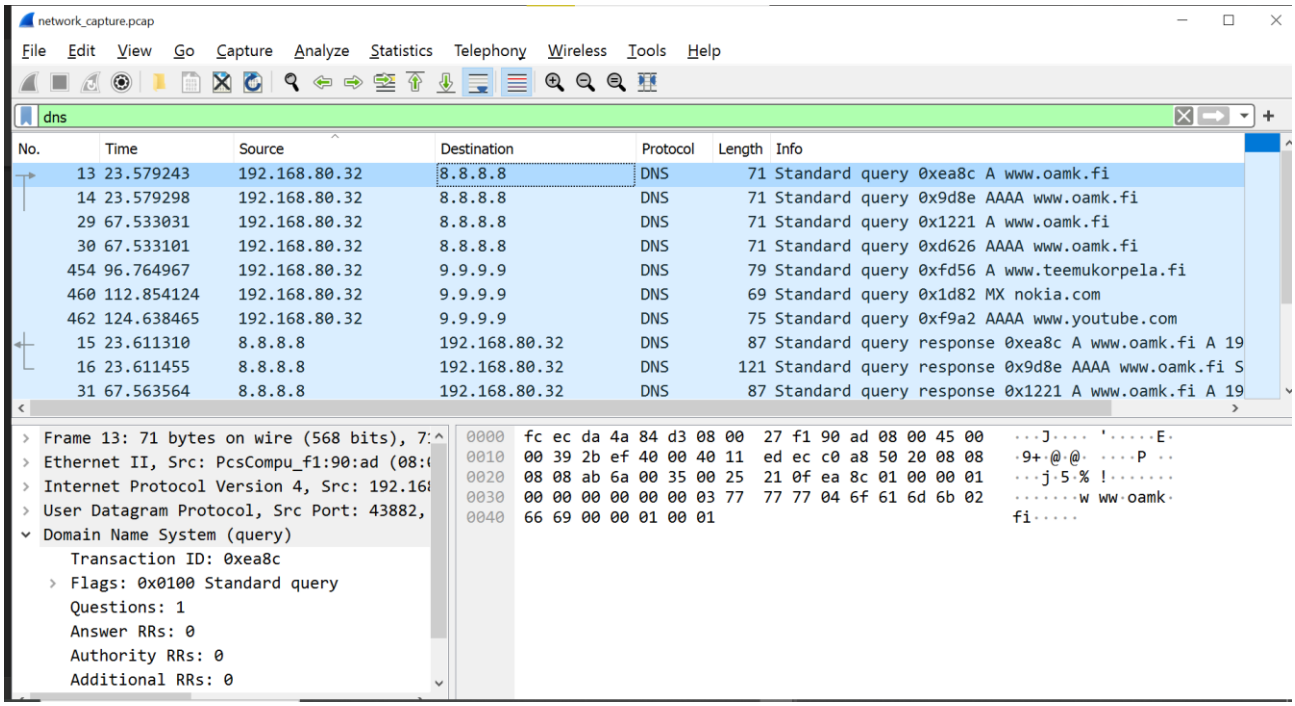
MAC address is fc-ec-da-4a-84-d3. The information ends up being told to Ubiquiti_4a:84:d3, the host at 192.168.80.1

- Which IP address sent ICMP echo requests to this (192.168.80.32) host? Also, there is a repeating short message inside ICMP datagrams the host sent as payload. What is the repeated message?

192.168.80.58 sent the echo requests. The echo contains “Hi there Oamk!”.

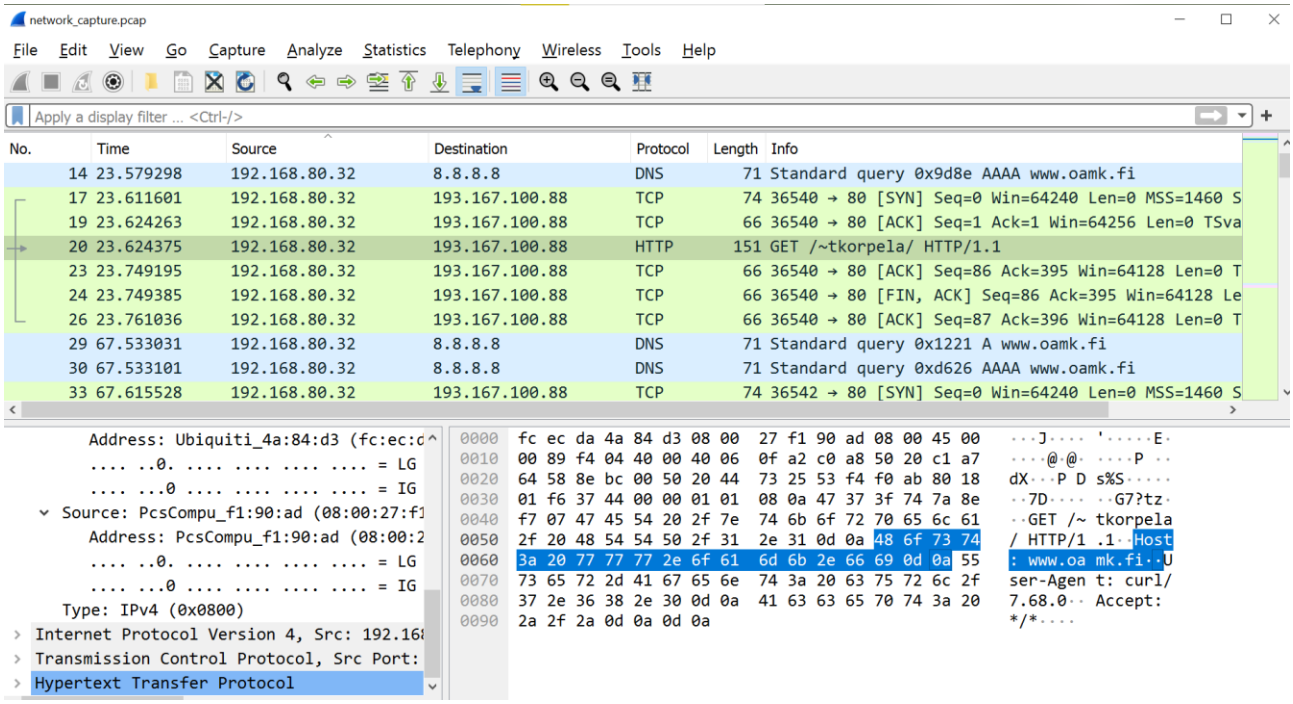



- What was the web page the host 192.168.80.32 visited first (full web address, not just the host)? What was the web browser or HTTP user agent string used to access that web server?



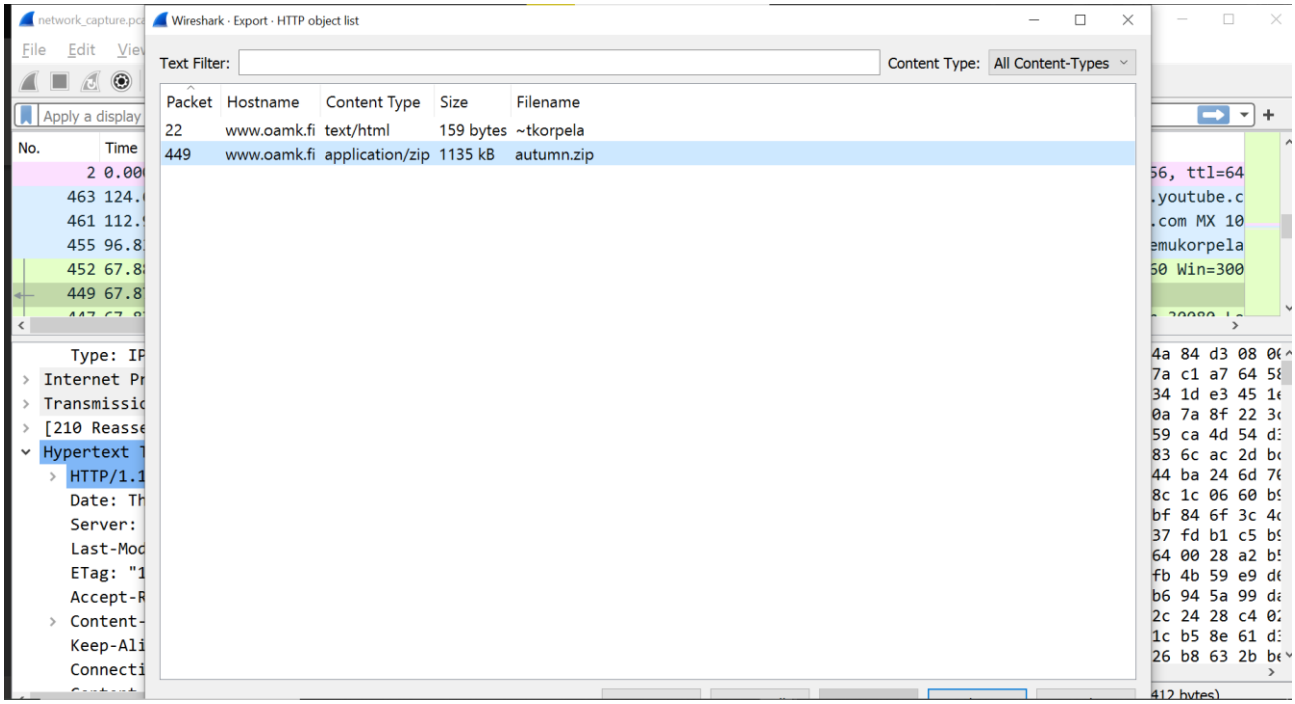
It first visited [www.oamk.fi](http://www.oamk.fi) through google.com, since 8.8.8.8 (where the request was being sent) is Google's dns service host address.

- What is the hostname in "Host:" -field of HTTP GET request sent by 192.168.80.32?




Host is www.oamk.fi

- What is most likely the default DNS server (IP address) used by the host 192.168.80.32?
  - 8.8.8.8 as it is Google's.
- Use Wireshark's file/export objects/HTTP feature to extract the ZIP file which was downloaded from the web server 193.167.100.88. What is inside the ZIP file?

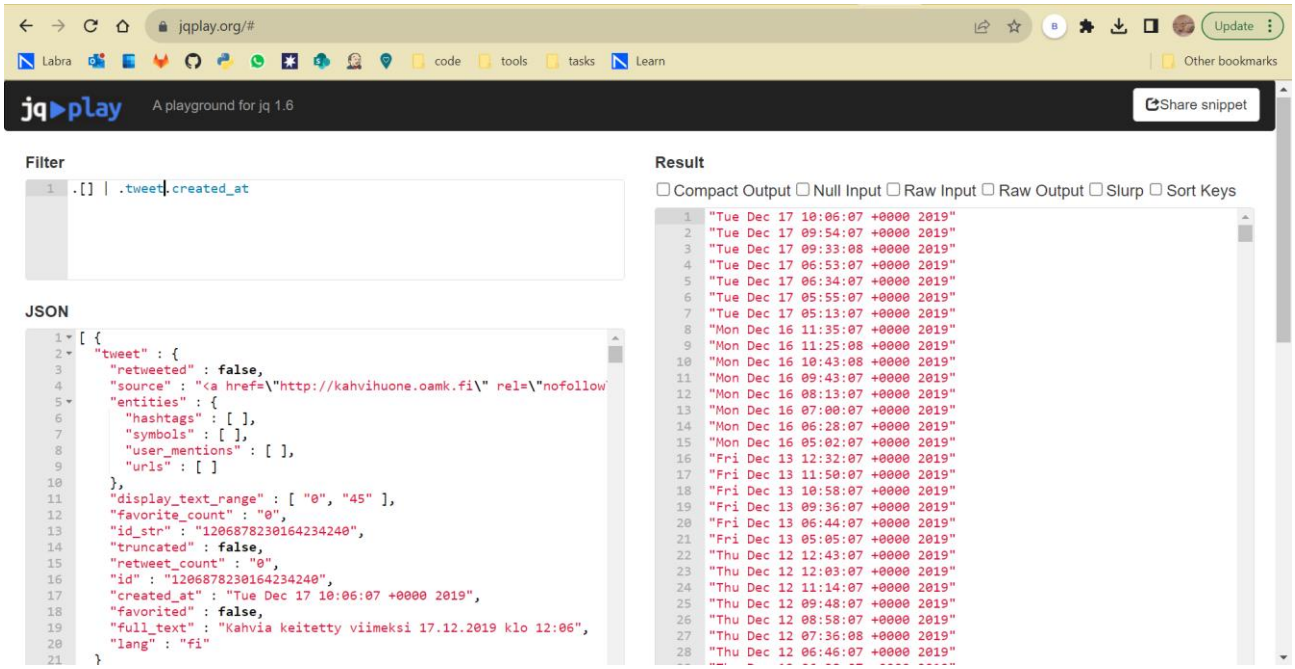


Content is this cute photo of Teemu's dogs

- Host 192.168.80.32 sent DNS requests to host 9.9.9.9. What are the requests?
  - The requests are queries to access web pages, www.teemukorpela.fi, nokia.com and then www.youtube.com

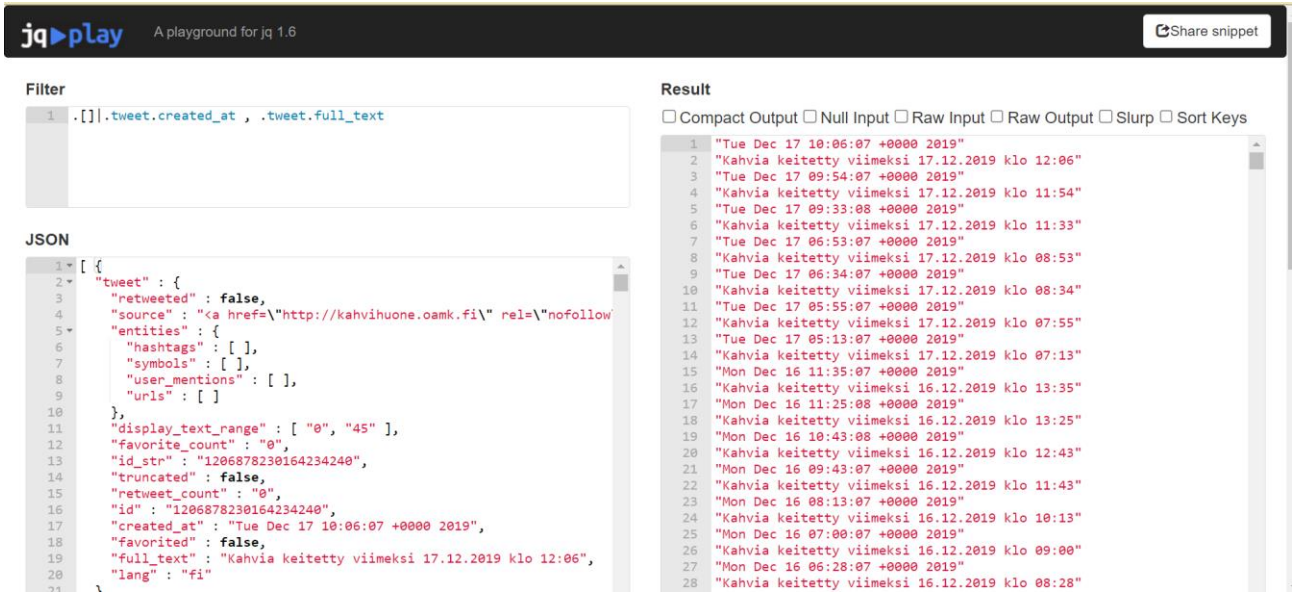
JSON tasks:

18. Install jQ JSON processor (Linux system is preferred, but tasks can be done with Windows and maybe with MacOS too)
    - Download old Twitter bot account data zipped JSON file and parse it with jQ:
    - Use jQ to list only "created_at" timestamp lines




(doing online since my laptop is company-managed and didn't allow me to install jqplay)

- Use jQ to list only "created_at" timestamp lines AND "full_text" lines




19. Create a new JSON file with any text editor. JSON file should contain data for at least two houses and related IoT sensor data. Each house must have few sensors with following information and some random data for each sensor. Something like this:

```
House:
  - IoT sensor:
    - sensor ID number
    - location description
    - notes about sensor
    - unix epoch timestamp
```

```
                - sensor values:
                - value nnn
                - value nnn
                - value nnn
```

- Validate your JSON file with validator: jsonlint.com or jsonformatter.curiousconcept.com



```
[
  {
    ”house”: 1,
    ”sensors”: [
      {
        ”sensor ID”: 1001,
        ”location”: ”in  front of house”,
        ”notes”: ”sensor located near entrance, measures humidity, temp and light in lumens”,
        ”time”: 1693747581,
        ”val”: {
          ”humidity”: 27,
          ”temperature”: 15,
          ”lum”: 97
        }
      },
      {
        ”sensor ID”: 1002,
        ”location”: ”upstairs of house, near balcony”,
        ”notes”: ”sensor located near balcony, measures humidity, temp and light in lumens”,
        ”time”: 1693747581,
        ”val”: {
          ”humidity”: 19,
          ”temperature”: 20,
          ”lum”: 137
        }
      },
```

```
      {
        "sensor ID": 1003,
        "location": "kitchen",
        "notes": "sensor located in kitchen, measures temp, CO-levels and decibels",
        "time": 1693747581,
        "val": {
          "CO": null,
          "temp": 22,
          "dec": 13
        }
      }
    ]
  },
  {
    "house": 2,
    "sensors": [
      {
        "sensor ID": 2001,
        "location": "in  front of house",
        "notes": "sensor located near entrance, measures humidity, temp and light in lumens",
        "time": 1693747581,
        "val": {
          "humidity": 39,
          "temperature": 12,
          "lum": 130
        }
      },
      {
        "sensor ID": 2002,
        "location": "inside house",
        "notes": "sensor located inside entrance/living room, measures humidity, temp and light in
lumens",
        "time": 1693747581,
        "val": {
          "humidity": 12,
          "temperature": 21,
          "lum": 200
        }
      },
      {
        "sensor ID": 2003,
        "location": "kitchen",
        "notes": "sensor located in kitchen, measures temp, CO-levels and decibels",
```

```
        "time": 1693747581,
        "val": {
          "CO": null,
          "temp": 25,
          "dec": 20
        }
      }
    ]
  }
]
```

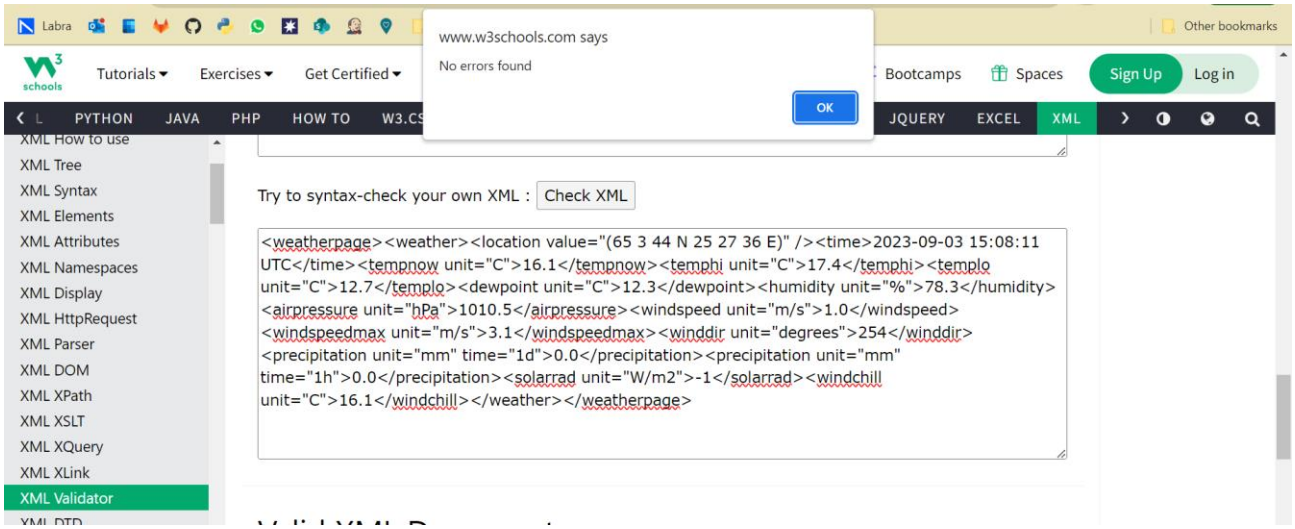

- What is GraphQL? Also, check this traffic and parking API documentation from Oulu (extra task uses this API)
    - Alternative to Rest APIs, it's a query language for APIs (Application Programming Interfaces) and a server-side runtime environment for executing those queries

XML tasks:

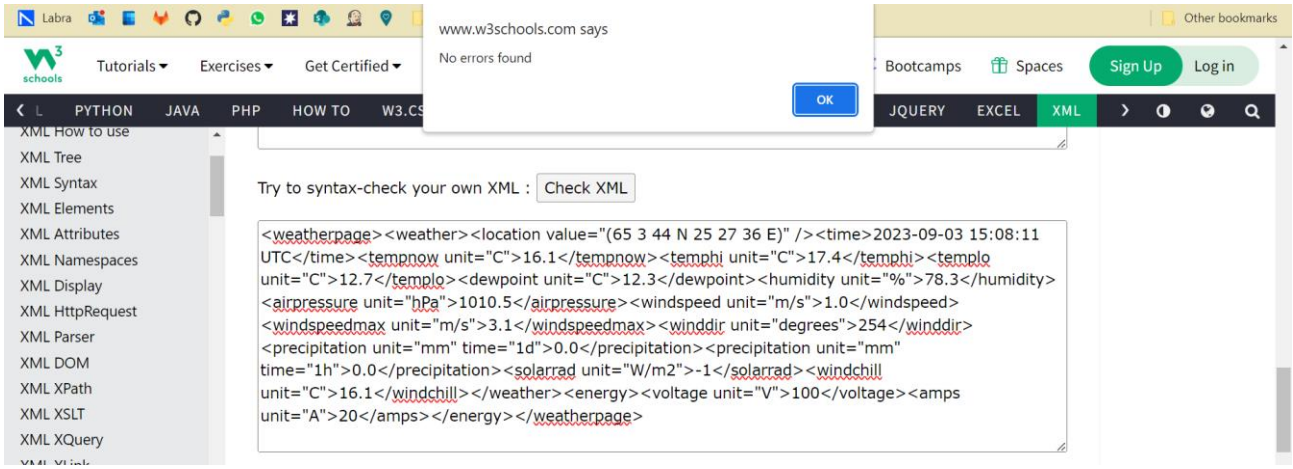
20. Install Cmder (or some other toolset where you have Curl or similar tool to make HTTP requests from command line or application. Curl is also used later.)
21. Use Curl to fetch XML formatted weather data from VTT:

```
curl -L http://weather.willab.fi/weather.xml
```

- Validate the XML file with www.w3schools.com/xml/xml_validator.asp




- Modify the XML data with text editor and add new "energy" element under the "weatherpage" root element
- "energy" element should have two new sub-elements "voltage" and "amps". Voltage units are "V" and amps units are "A". Use some random values for the data
- Validate your modified XML file with www.w3schools.com/xml/xml_validator.asp




Base64:

22. Decode this base64 message with any tool(s) you prefer:

    `SGVsbG8gdGhlcmUgT2FtayBzdHVkZW50ISBBcmUgeW91IGhhdmluZyBmdW4gbm93Pz8/`

    = Hello there Oamk student! Are you having fun now???

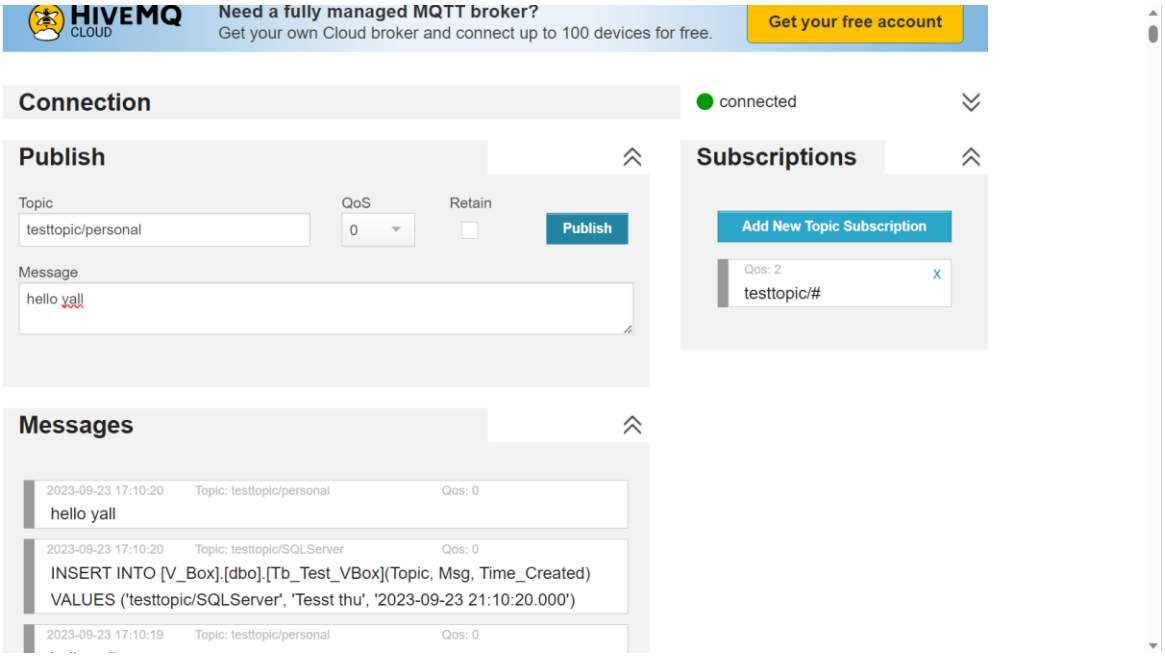
- Encode string: “I love data processing challenges!” with base64 encoding

  `SSBsb3ZlIGRhdGEgcHJvY2Vzc2luZyBjaGFsbGVuZ2VzIQ==`

# Week 6

IoT:

23. Try MQTT websocket demo application to subscribe to some existing topic(s) in HiveMQ. Try to publish some messages.¨




24. Explain what are MQTT retained messages

    - MQTT (Message Queuing Telemetry Transport) retained messages allow the latest message published on a specific topic to be stored and remembered by the broker. This feature ensures that subscribers always receive the most recent value or status for a specific topic, even if they weren't subscribed at the time the message was originally published.

25. Describe the difference between request-response and publish-subscribe communication models

    - Request-response: In this model, a client sends a request to a specific server or endpoint, and the server processes the request and sends back a response. It is a synchronous communication model where the client waits for a reply from the server before proceeding.

    - Publish-subscribe: In this model, there are multiple publishers and subscribers. Publishers send messages to specific topics or channels, and people can subscribe to specific topics. Subscribers receive messages published to the topics they are interested in. The communication is asynchronous.

26. List shortly some reasons why MQTT may be better than HTTP for IP-based IoT communication? (For example: HTTP vs. MQTT: A tale of two IoT protocols and MQTT Vs. HTTP: Understanding the Differences)

    - Lightweight protocol: MQTT is designed to be a lightweight and efficient protocol, making it suitable for IoT devices with limited resources.

- Asynchronous communication: MQTT supports asynchronous communication, making it well-suited for scenarios where devices need to send updates or events without waiting for a response.
- Quality of Service (QoS) levels: MQTT offers QoS levels that allow you to specify the reliability of message delivery, which can be crucial in IoT applications.

27. What is CoAP?

    - CoAP (Constrained Application Protocol) is a lightweight application layer protocol designed for resource-constrained IoT devices and networks. Similar to HTTP in its request-response model, but optimized for low-power, low-bandwidth, and constrained environments. CoAP is often used for IoT applications where HTTP may be too heavy.

28. What is 6LoWPAN?

    - 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) is a technology that enables the use of IPv6 over low-power and low-rate wireless networks. It is designed to be used in IoT and wireless sensor network applications, allowing devices to communicate using standard IPv6 addressing and protocols.

29. What is IETF ROLL and IETF ROLL RPL protocol?

    - IETF ROLL (Routing Over Low power and Lossy networks) is a group within the Internet Engineering Task Force (IETF) focusing on developing routing protocols for low-power and lossy networks (LLNs), which are common in IoT and wireless sensor networks. RPL (Routing Protocol for Low-Power and Lossy Networks) is one of the key protocols developed by the ROLL working group. RPL is designed to provide efficient and reliable routing for LLNs.

30. Why classic computer network protocols like TCP/IP, data formats like JSON and XML, and security systems like (PKI/HTTPS) won't usually work at all or are not optimal to be used in wireless sensor network (typically a low power and lossy network)?

    - High overhead: TCP/IP and HTTP have relatively high protocol overhead, which can be inefficient for small, low-power devices with limited resources.
    - Complexity and power consumption: JSON and XML data formats can be verbose and complex, leading to increased energy consumption during data encoding and decoding. In addition, security mechanisms like PKI/HTTPS can require very high-tech devices and systems in addition to significant power, making them unsuitable suitable for battery-powered IoT devices.

31. What is the MTU challenge for IPv4 and IPv6 over common wireless low power and lossy wireless connections (Hint: Research Zigbee/IEEE 802.15.4 and Bluetooth MTU vs IPv4 or IPv6)?

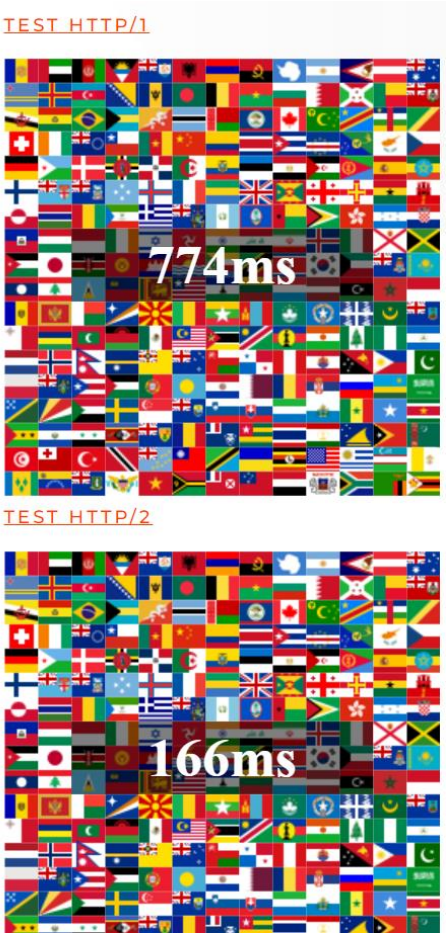
    - The MTU (Maximum Transmission Unit) challenge is that the standard MTU values for IPv4 and IPv6 (typically 1500 bytes for Ethernet) are too large for these networks. These networks often have smaller MTU values, which means that IPv4 and IPv6 packets need to be fragmented and reassembled, increasing overhead and potentially causing issues in these constrained environments. IPv6 and IPv4 need to adapt to the smaller MTU sizes of these networks to function optimally.

HTTP/1.1, HTTP/2 and HTTP/3:

23. What is Head-of-Line blocking problem with HTTP/1.1 and HTTP/2? How HTTP/3 mitigates this?

    - The Head-of-Line (HOL) blocking problem is a challenge in HTTP/1.1 and HTTP/2 because if a single resource request/response is delayed, it can block subsequent requests that are queued behind it. This can lead to poor performance and delays in loading web pages.
    - HTTP/3 mitigates the HOL blocking problem by using QUIC (Quick UDP Internet Connections) as the underlying transport protocol. QUIC allows for combining multiple streams of data over a single connection, and it doesn't suffer from HOL blocking because if one stream is delayed, it doesn't block other streams. Each stream is independently scheduled and delivered, improving the overall performance of web applications in scenarios with packet loss or latency.

24. Try this HTTP/1 vs HTTP/2 speedtest




25. Use this tool to check few websites whether the server supports HTTP/2: tools.keycdn.com/http2-test. Two examples: www.kaleva.fi and www.oamk.fi

www.kaleva.fi

HTTP/2 protocol is supported.

ALPN extension is supported.

www.oamk.fi

HTTP/2 protocol is not supported.

ALPN extension is not supported.

26. Study Google Firebase. Think and list examples how to use Firebase ecosystem with Android application(s) or with some IoT system?

    - Authentication: Firebase Authentication can be used to implement secure user authentication in Android apps. Users can sign in with email, social media accounts, or phone numbers, providing a streamlined and secure login experience. Also, For IoT systems, Firebase Authentication can secure device-to-cloud communication, ensuring that only authorized devices can send data to the cloud.
    - Realtime Database: Firebase Realtime Database can be used to store and synchronize data in real-time across Android applications and IoT devices. For example, you can build a real-time chat application or IoT dashboard in which data updates are instantly reflected.
    - Cloud Functions: Firebase Cloud Functions allow you to run serverless functions in response to events triggered by Firebase services or external sources. In IoT, you can use Cloud Functions to process incoming data, trigger alerts, or perform other actions.
    - Cloud Storage: Firebase Cloud Storage allows you to store and serve user-generated content, such as images or files, in Android apps or IoT systems. It can be used to store sensor data, images from IoT cameras, or user-generated content.

27. Compare and list few HTTP/1.1, HTTP/2 and HTTP/3 differences and features

    **HTTP/1.1:**

    - Header inefficiency: Headers are sent with each request and response, leading to redundancy.
    - No multiplexing: Requests are processed sequentially, leading to potential head-of-line blocking.
    - No built-in encryption: Encryption is optional and requires an additional layer.
    - Text-based: Messages are in normal text, increasing overhead parsing.

**HTTP/2:**

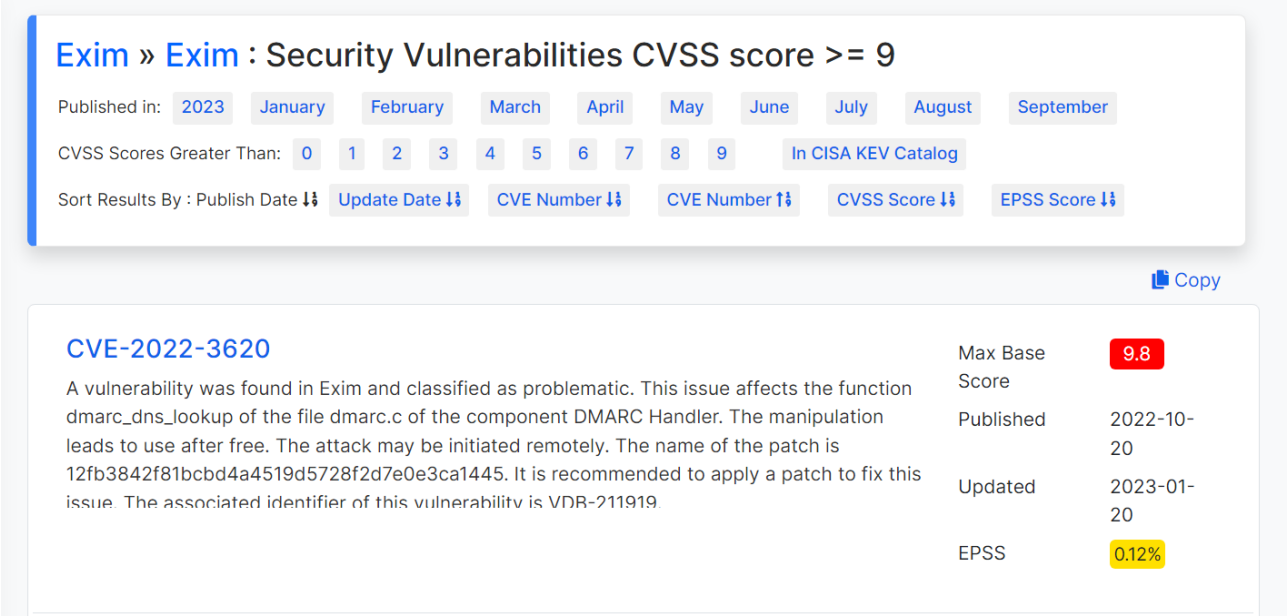
- Multiplexing: Multiple requests and responses can be multiplexed over a single connection, reducing latency.
- Header compression: Header fields are compressed, reducing overhead.
- Enhanced security: Encourages the use of HTTPS by design.
- Binary framing: Messages are sent in binary frames for efficiency.

**HTTP/3:**

- Built on QUIC: Uses the QUIC transport protocol, designed for low-latency and reliability.
- Eliminates head-of-line blocking: Each stream is independent, reducing latency.
- Designed for multiplexing: Supports multiple streams of information within a single connection.
- Encrypted by default: All communication is encrypted.
- Reduced latency: Loads website material faster.

# Week 7

37. Explain Microsoft's STRIDE threat model shortly (see the old software vulnerability slides)
    - STRIDE is used to sort assorted threats ranging from Spoofing to Elevation of priviledge.
38. Explain Microsoft's DREAD risk model shortly (see the old software vulnerability slides)
    - Used for assessing the risk level in a standardized way, DREAD can help gain mutual understanding in a risk scenario by providing analyzable areas such as reproducibility and damage potential.
39. Check some CVEs of widely used applications from https://www.cvedetails.com/ and answer:
    - Describe what is the CVE scoring system
        - Common Vulnerability Scoring System (CVSS) is a framework for assessing the severity of security vulnerabilities in computer systems and software. It provides a standardized way to evaluate and communicate the impact of vulnerabilities, making it easier for organizations and security professionals to prioritize and address security issues. CVSS takes into account many factors and helps organizations make informed decisions and decide how to address them.
    - When was the last time when Exim (MTA, mail transfer agent, more modern version of the application, not the Cambridge version) had a critical vulnerability? What is the CVE number?




CVE-2022-3620

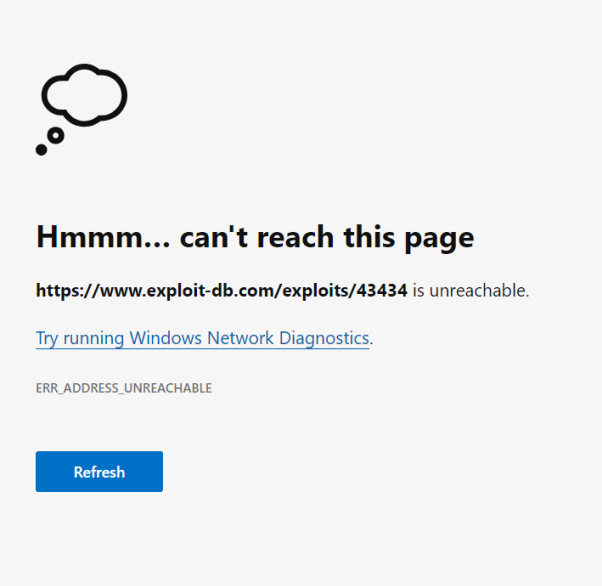
    - Describe CVE-2016-6210 vulnerability shortly
        - If corrupted auth package is sent, some versions of OpenSSH enumerate users.
    - Describe CVE-2019-15846 vulnerability shortly

- Using a trailing backslash, hackers are able to execute code with root permissions.

- Describe CWE-208 from https://cwe.mitre.org/data/archive.html (download most recent PDF)

  - Time completing an action can be observed and this can give attackers hints on details of the operation; success or private key details being examples.

40. Study D-Link DNS-320 ShareCenter write-up in the ExploitDB. What kind of software exploit is that? Try to explain shortly how the attacker can elevate his/her access to the device?




41. Read this short article about cracking SIM cards and answer these questions:

    - What is “side-channel attack”?

      - A cybersecurity threat in which attackers “measure things like power consumption, electromagnetic emissions and heat generation to work out what is going on in a chip” (Rambus)

    - How side-channel attack was used to crack SIM cards?

      - The attackers were able to correlate the results from the measurements to get clues on the cards and eventually crack them.

42. Browse this public penetration test report and news article and answer these questions:

    - Penetration test report has header *security by obscurity* (next to the item 171 and onwards). What does it mean?

      - If the information on a component is explicitly described it is much more vulnerable than an “obscure” item: having specific details on a chip (or basically any HW) makes it easier to reverse engineer compared to not having all the information readily available.

    - Penetration test report items 114 - 141 describe remote attack and vulnerability. What kind of problem is it?

- Cardiac devices (In-cardio device, ICD) are able to receive requests that drain their power at an alarming rate. The code queries a list of devices nearby and attempts to interrogate them, after which it disconnects. This is performed as a loop a few times in a minute, resulting to 3-4% battery consumption in a 24h period.

43. Describe shortly following security tools/terms/concepts:

- Disassembler
  - A tool used to convert machine code (binary) back into assembly language or a high-level programming language.
- Overflow vulnerability
  - A vulnerability that occurs when input data exceeds the expected bounds, potentially causing unintended behavior or code execution.
- Race condition vulnerability
  - A vulnerability arising when multiple threads or processes access shared resources in an unpredictable manner, potentially leading to unexpected behavior.
- RCE vulnerability
  - Remote Code Execution, a vulnerability that allows an attacker to execute arbitrary code on a target system.
- Local privilege escalation
  - Exploiting a vulnerability to gain higher levels of system access than initially intended.
- Zero-day vulnerability
  - A zero-day vulnerability is a previously unknown security weakness in software or technology that hackers discover and exploit before developers even find it or can provide a fix or patch.
- Code deobfuscation / obfuscation
  - The process of removing or adding obfuscation (confusing or complex code structures) to reveal or hide the actual code logic.
- SAST vs DAST
  - SAST (Static Application Security Testing) analyzes source code for vulnerabilities before the code is executed. DAST (Dynamic Application Security Testing) tests the application while it's running to find vulnerabilities that might not be apparent from source code analysis.
- Fuzzing
  - A testing technique that involves sending a large volume of random, unexpected, or invalid inputs to a software application to identify vulnerabilities or crashes.

# Sources

Week 1


- Difference between OSI and TCP IP Reference Model (tutorialspoint.com)
- TCP/IP vs OSI Model – Difference Between Them (guru99.com)
- OSI vs TCP/IP: What's the Difference? - javatpoint
- Difference between Simplex, Half duplex and Full Duplex Transmission Modes - GeeksforGeeks
- Messages: Payload, Header, and Overhead | Baeldung on Computer Science
- Payload Packet - an overview | ScienceDirect Topics
- Payload Definition - What is a computer payload? (techterms.com)
- What are Bits Per Second (bps)? - Definition from Techopedia
- Half & Full Duplex | Comms InfoZone (comms-express.com)
- Difference Between Half-Duplex vs Full-Duplex - Total Phase
- stanview.pdf (icsgroup.ru)
- Network Cable Standards: TIA 568 vs ISO 11801 vs EN 50173 - Structured Cabling News
- Network Cable Standards: TIA 568 vs ISO 11801 vs EN 50173 | FS Community
- 817-7526-10.fm (netcraftsmen.com)
- What is Ethernet Auto-Negotiation? – Fosco Connect (fiberoptics4sale.com)
- (and more for cross-referencing and small details)


Week 2


- How Address Resolution Protocol (ARP) works? - GeeksforGeeks
- What Is Address Resolution Protocol (ARP)? | Fortinet
- Static Vs. Dynamic Routing: What is the Difference? (techtarget.com)
- What is Dynamic Routing? - Definition from Techopedia
- ICMP: Definition & How it Works | Protocol Support Library (extrahop.com)
- What is ICMP? | Internet Control Message Protocol | Cloudflare
- What is ICMP (Internet Control Message Protocol)? | Fortinet
- Routing-Basics-webinar-2.pdf (apnic.net)
- Comparing Dynamic Routing Protocols | Network Computing

- Dynamic Routing Protocols: OSPF, EIGRP, RIPv2, IS-IS, BGP - Cisco Community
- Nokia internal sources and knowledge, colleagues

Week 3

- TCP 3-Way Handshake (SYN, SYN-ACK,ACK) (guru99.com)
- TCP Header Anaylsis - Section 2: TCP Sequence & Acknowledgement Numbers (firewall.cx)
- Understanding TCP Sequence and Acknowledgment Numbers - PacketLife.net
- TCP Sequence and Acknowledgement Numbers Explained – MadPackets
- TCP connection states – confirm blog
- TCPIP_State_Transition_Diagram.pdf (northwestern.edu)
- TCP connection status - IBM Documentation
- TCP states - explained - Google Cloud Community
- Difference Between Source Port and Destination Port - GeeksforGeeks
- TCP vs UDP: When to Use Which Protocol | Twingate
- TCP vs UDP: Differences Between TCP & UDP Protocols | Avast
- TCP Networking: Understanding the Nagle Algorithm (lifewire.com)
- What is Nagle's Algorithm? (with picture) (easytechjunkie.com)
- Understanding Idle Timeout and Keep Alive Interval settings in the TCP profile (f5.com)
- TCP keepalive overview (tldp.org)
- linux - what is RAW socket in socket programming - Stack Overflow
- A Guide to Using Raw Sockets - open source for you (opensourceforu.com)

Week 4

- (none)

Week 5

- https://support.dnsimple.com/articles/aaaa-record/
- DNS AAAA record | Cloudflare
- DNS Protocol (ns1.com)
- DNS query types and how to use DNS in performance troubleshooting (accedian.com)

Week 6

- What are Retained Messages in MQTT? – MQTT Essentials: Part 8 (hivemq.com)
- MQTT Retained Messages Explained (steves-internet-guide.com)
- The Difference Between Client-Server and Publisher-Subscriber | Nordic APIs |
- Difference Between Publish-Subscribe and Request-Response Model | by pandaquests | Level Up Coding (gitconnected.com)
- HTTP vs. MQTT: Comparison for IoT (telit.com)
- HTTP vs MQTT: Choose the best one for an IoT project | Cedalo
- MQTT vs HTTP for IoT (hivemq.com)
- CoAP - Constrained Application Protocol - Radiocrafts
- CoAP protocol – Nordic Developer Academy (nordicsemi.com)
- IPv6 LoWPAN Neighbor Discovery and Addressing Choices (ietf.org)
- What is 6LoWPAN? - GeeksforGeeks
- Analysis of routing protocol for Low Power and Lossy Networks (RPL) using Cooja simulator | IEEE Conference Publication | IEEE Xplore
- Routing Protocol for Low Power and Lossy Network | Encyclopedia MDPI
- Routing Over Low power and Lossy networks (roll) (ietf.org)
- TCP head of line blocking - HTTP/3 explained (haxx.se)
- Head-of-line (HOL) blocking in HTTP/1 and HTTP/2 | by Abhishek Varshney | CRED Engineering
- Head-of-Line Blocking - an overview | ScienceDirect Topics

Week 7

- What are Disassemblers - CTF 101
- Buffer Overflow | OWASP Foundation
- What Is a Race Condition Vulnerability? | Indusface
- What is Remote Code Execution (RCE) Vulnerability (wallarm.com)
- Remote Code Execution (RCE) | Types, Examples & Mitigation | Imperva
- Understanding Privilege Escalation and 5 Common Attack Techniques (cynet.com)
- Reverse engineering : Code Deobfuscation in the age of AI | by Alessio Trivisonno | FAUN — Developer Community
- Deobfuscation for Beginners. A Tutorial About Dealing With an… | by Roei Kriger | InfoSec Write-ups (infosecwriteups.com)

- SAST vs DAST: What they are and when to use them | CircleCI
- What Is Fuzz Testing and How Does It Work? | Synopsys
- Fuzzing | OWASP Foundation