INTRODUCTION TO THE CLASS, SECURITY, AND NETWORKING

September 25, 2025

"The one with all the introductions."

Credits

Content: Sebastian Garcia, Veronica Valeros, Maria Rigaki
Martin Řepa, Lukáš Forst, Ondřej Lukáš, Muris Sladić
Illustrations: Fermin Valeros
Design: Veronica Garcia, Veronica Valeros, Ondřej Lukáš
Music: Sebastian Garcia, Veronica Valeros, Ondřej Lukáš
CTU Video Recording: Jan Sláma, Václav Svoboda, Marcela Charvatová
Audio files, 3D prints, and Stickers: Veronica Valeros

CLASS DOCUMENT	https://bit.ly/BSY2025-1
WEBSITE	https://cybersecurity.bsy.fel.cvut.cz/
MATRIX	https://matrix.bsy.fel.cvut.cz/
CTFD (CTU STUDENTS)	https://ctfd.bsy.fel.cvut.cz/
PASSCODE FORM (MOOC STUDENTS)	https://bit.ly/BSY-MOOCPasscode
FEEDBACK	https://bit.ly/BSY-Feedback
LIVESTREAM	https://bit.ly/BSY-Livestream
INTRO SOUND	https://bit.ly/BSY-IntroVideo
VIDEO RECORDINGS PLAYLIST	https://bit.ly/BSY-Recordings

🚨 We have live video from the back of the classroom, so be aware if you are in person.

Class Outline

The goal of this class: To introduce the class and organizational information. To start with, the basic concepts of security and network protocols.

Introductions (14.35, 6m)

For the second time, this “Introduction to Computer Security” course at the Czech Technical University in Prague (CTU), also coded BSY, is **both a CTU class and a MOOC**!

Welcome everyone!

About Stratosphere (14.41, 5m)

The BSY teachers belong to the **Stratosphere Laboratory**, a cybersecurity research group at the Artificial Intelligence Center, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.

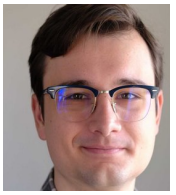
Stratosphere works at the intersection of machine learning and security. Learn more at www.stratosphereips.org.

About the Teachers









Sebastian Garcia | **Maria Rigaki** | **Ondřej Lukáš** | **Veronica Valeros** | **Lukáš Forst** | **Martin Řepa** | **Muris Sladić**

Sebastian: Loves to teach and research in machine learning and network security. Loves to hack all the things, learn, attack, defend, pentest, explore, develop, squash, run, walk, and play padel.

Veronica: Researcher and PhD student @AI Center, project lead @Stratosphere, and master student @LJMU. Loves honeypots, threat research, and working with others. Likes to watch old B&W movies.

Ondřej: PhD student @AI Center focusing on Explainable AI & Security. I like to play sports, travel, and cook. Passionate football fan.

Lukáš: Security enthusiast, alumni of the class, software/security engineer, co-founder @Recon Wave, previously Better Stack, CDN77, Wire, Blindspot.ai. Hacker by day, musician by night.

Martin: Software and security engineer, bug bounty hunter, co-founder of Recon Wave. Loves hacking and reconnaissance.

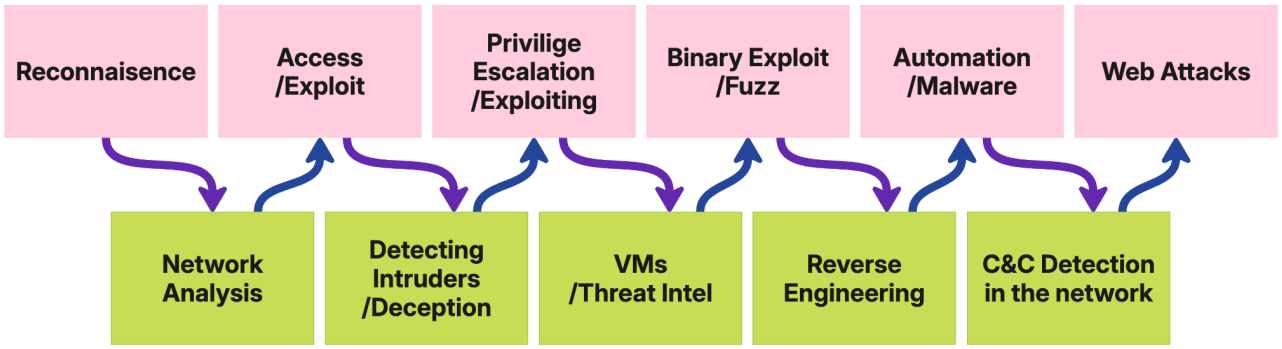
Maria: PhD student @AI Center focusing on adversarial applications of machine learning in security. I like to play CTFs, guitars, and chess.

Muris: PhD student @AI Center focusing on cyber defense and deception using Generative AI. He enjoys traveling, playing music, writing, and puzzle games.

Class Dynamics

This is a hands-on, practical, 14-week-long class on penetration testing and advanced defenses. The **goal** is to give students solid basic security knowledge by learning how to **attack** and **defend** by following a basic penetration testing methodology and a defense methodology.

We will see the following topics: https://cybersecurity.bsy.fel.cvut.cz/docs/syllabus/. Each class will alternate between **attacking** and **defending**.



At the end of the subject, you should understand the principles of security, how basic attacks are done, how to defend against them, and how to analyze attacks in the network.

Class Modalities (14.46, 2m)

	CTU Students	**Prg.AI Master Students**	**LLL Professional Tier Students**	**MOOC Free Tier Students**
Identification	**CTU Students**	**CTU Students**	**CTU Students**	**MOOC Students**
Use?	CTU Cyber Range	CTU Cyber Range	CTU Cyber Range	**S**trato**C**yber**L**ab
To complete the coursework	Weekly assignments	Weekly assignments	Weekly assignments	Submit correct passcodes every week
To pass the class	Final exam or bonus assignment	Final exam or bonus assignment	Final exam or bonus assignment	No final exam
Get what?	Credits	Credits	Official CTU/EU Approval Cert.	Certificate of Completion issued by Stratosphere

Teaching Methodology (14.48, 5m)

The class is held weekly on Thursdays (Subscribe to Google Calendar if you want) from **14:30 to 17:45** Prague time (check https://time.is/Prague).

Class Block 01	55 minutes	14:30 - 15:25 CEST
Break	10 minutes	15:25 - 15:35 CEST
Class Block 02	55 minutes	15:35 - 16:30 CEST
Break	10 minutes	16:30 - 16:40 CEST
Class Block 03	65 minutes	16:40 - 17:45 CEST

From the class of Oct 30th onward, it changes from CEST (UTC+2) to CET time (UTC+1).

Study Material

- Classes will be live-streamed on YouTube.
- Video recordings of each class will be made available within 24 hours.
- We use Google Docs because it allows you to:
    - Make a copy of the class document and add your notes.
    - Copy-paste the commands directly.
    - Have a live document that is being fixed and corrected.
    - Have a step-by-step of everything, and not just a summary.
    - You can see the whole class before taking it.

Cyber Range for CTU Students

Each student will be assigned a personal Docker container for the class duration:

- We run Linux Docker containers in our Stratosphere Laboratory servers.
- You will connect to your container via SSH from your laptop.
- Each container is personal, and it's your responsibility for the class duration.
- You will have to defend this Docker container.

- You will attack other Docker containers to accomplish different goals.
- Your goal as an attacker is NOT to be detected.
- Your goal as a defender is to detect all the attackers.

Cyber Range for MOOC Students

MOOC students will use our **StratoCyberLab,** a standalone, free software, offline cyber range (https://github.com/stratosphereips/stratocyberlab).

- StratoCyberlab (SCL) was specifically developed for you.
- The SCL uses orchestrated Docker containers on your computer to be completely **offline**.
- You can practice, submit flags, and receive **LLM assistance** without needing the Internet. But you need space, so check the constraints.
- We will update the StratoCyberLab every week with new containers and challenges.
- **You MUST have this installed for this class.**

Communication (14.53, 5m)

Questions and discussions will be conducted through the **Matrix** chat server. You should have received the credentials by email. If not, send us an email.

- If you need to ask something, **always** use the **#general** public room in Matrix.
- Do not message us directly unless it is very private. And almost nothing is.
- If you need to ask us something *very* private (like personal problems), **always** chat with all the teachers together in Matrix.
    - Do not chat with us individually.
    - Let me repeat that. Do *not* chat with us individually.
- Please **ask questions** during the class. Do not wait until the topic is over.
- However you talk to us, **always be sure that all the teachers can see it.**

Requirements and Grading

Assignments for CTU Students (14.58, 10m)

1. Almost every week, there will be **assignments** to do at home.
2. An assignment is a challenge to solve in the CTU Cyber Range for CTU students.
3. The points for each assignment are given when you find a FLAG after solving the assignment (CTF style).
4. A flag is a unique string with the format: BSY{64chars}
    a. Example: **BSY{1a1a1a1a1a1a1a1a1...a1a1a1a1a1a1a1a1a1a}**
5. When you get these flags, you must submit them to the CTFd (capture the flag) server https://ctfd.bsy.fel.cvut.cz/
6. When you submit it, you can see if the flag is correct, and the points will appear in the CTFd scoreboard.
7. You are **not** required to submit the flags every week. But we suggest you do them as soon as possible. Deadlines for the assignments will be announced with the release of every assignment.
8. **All deadlines are final - no late submissions.**
9. However, there are **Pioneer Prizes**! For every assignment, we will award the first student who solves the assignment with some nice presents (sorted by submission time in the CTFd).
10. In each class, we will give you some clues to start solving the assignments.
11. Assignment submission will start every week on **Thursdays at 21:00 Prague time (after the class)** unless stated otherwise.

Assignments for MOOC Students (15:08, 1m)

For MOOC students, the **StratoCyberLab** **may** have, from time to time, new challenges for you to play. Note that these assignments are **not graded.** We don't promise there will be, just maybe.

Grading for CTU Students (15:09, 3m)

All the information about the grades is in one location on our web page here: https://cybersecurity.bsy.fel.cvut.cz/docs/class-variants/ctu-students/

Bonus Assignment

For CTU students, there is an option not to take the final exam if you **take** and **approve** the Bonus assignment.

The bonus assignment is a special assignment released in December that you do during the holidays, and it usually takes several days.

Grading for MOOC Students (15:12, 3m)

We check that you attended and saw the videos for the classes:

- How do we check?
    - We show and say a secret password during the live class.
    - It is not going to be in this document.
    - You must see the class live or the recording.
    - Submit the sacred password for the class through the Google Form: https://bit.ly/BSY-MOOCPasscode.
    - You will receive a copy of your submission via email.
- You can **only** miss two classes. This means that from the 14 classes, you must submit at least 12 correctly.
- Do not share the passcodes of the classes with anyone.
- Do not publish the passcodes of the classes.

Password Time

The password for the class is...

Class Ethics (15:15, 2m)

Since this class will teach real attack and defense techniques, by taking the class, you commit to the following:

1. You will **NOT**
    a. Attack others on the Internet from the Docker we give you in our infrastructure.
    b. Attack the assignment servers, CTFd servers, Matrix servers, or any other server that is used for the class.
    c. Attack other servers and services in the university network (outside of IP range given to you).
    d. Use Docker for tasks other than those related to the class. Do not use it for personal things or cryptomining.
2. You can
    a. Attack **from** the Internet **to** the **students' dockers.**
    b. Attack **from** the local Docker network to the other **students' Docker.**

Code of Conduct (15:17, 3m)

We expect all participants to engage in a respectful and positive manner. The following behaviors are strictly prohibited in class, chat, or any online interactions (including video chats):

- Aggressive behavior, bullying, or harassment of any kind.
- Disrespectful or derogatory language toward others.

All students must communicate with respect, understanding, and empathy. Any violation of this code of conduct may result in immediate removal from the course and disqualification from participating in future courses.

Be excellent to each other.

Failure to comply with these ethics rules will take you out of the class, at least.

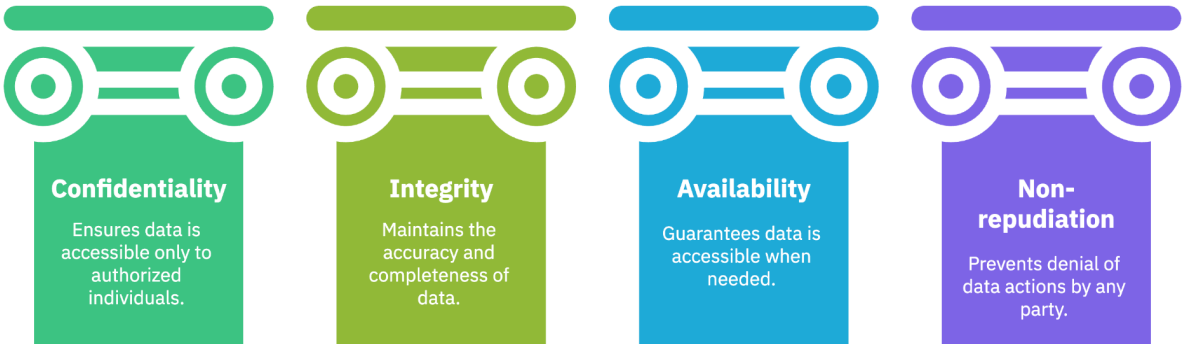
~~~~ 💗 Break! 💗~~~~

Stretch, breathe, and relax! See you all back in 10 minutes!

Basics of Security (15:20)

Why security? Computer security is a very important topic for our society and technological world. It doesn't matter what you work on in IT, it will be important for you to think about security.

🏛️ Pillars of Security (15:20, 15m)



- **Confidentiality**
  - You can not access/read/understand the data without authorization.
  - More technically: *"information is not made available or disclosed to unauthorized individuals, entities, or processes"*[1]
  - E.g., encryption is one way we can enforce confidentiality
  - 🤖AI View: So far, AI can not directly help you break encryption faster.
- **Integrity**
  - Data is not modified without authorization.
  - More technically, it means *"maintaining and assuring the accuracy and completeness of data. Data cannot be modified in an unauthorized or undetected manner."*[1]
  - E.g., hashing functions are a way to verify that the integrity of data is maintained.

[1] Information security - Wikipedia. (2024). Retrieved April 27, 2025, from Wikipedia.org website: https://en.wikipedia.org/wiki/Information_security

    - 🤖AI View: So far, AI can not help you create collisions in hash functions.
- **Availability**
    - Data should be available for those authorized when needed.
    - Every part of the system should work correctly for this to happen: *"computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly."*[1]
    - 🤖AI View: AI may, on purpose, make something not available if it has the chance.
- **Non-repudiation**
    - Neither party can deny sending, receiving, or accessing the data.
    - 🤖AI View: AI can not deny sending after signing with a private key.

Transversal pillars

- **Authentication**: the ability to confirm with a high degree of certainty that they are indeed who they say they are.
    - 🤖AI View: AI can help you brute force in a smart way, or, for example, change your face for age verification. [link]
- **Authorization**: Once authenticated, what you are authorized to do or not to do.
    - 🤖AI View: So far, AI can not directly help you change this directly.

⚔️ What is an attack? (15:35, 15m)

When dealing with network traffic in security, it's common to focus on attacks on the network. But what is an attack? Is an attack an exploit in a packet arriving at a server? Is an attack a cross-site script included in an email link? Is an attack a bunch of DDoS packets? What about one packet? Can it be an attack?

- Difference with normal traffic:
    - Which ones do you think are the main differences between normal and attack traffic?
    - Content? Behavior? Destinations?
- What is malware?

    - What is, for you, the definition of malware?
    - How can you know if something is malware?
- What is a botnet?
    - What are the characteristics of a botnet?
    - Is a botnet malware? Is malware a botnet?
- What other types of attacks are there?
    - If you want to detect attacks, what should you detect? Why is Snort/Suricata not enough?
- 🤖AI View: Can an AI attack?
    - What about open-ended goals that result in an attacking behavior? Like in the famous article here.
    - **AI cannot "intend" an attack in the human sense**, but it can *carry out actions that functionally are attacks*.
    - Responsibility then hinges on the humans behind it, unless future law or ethics frameworks decide to treat autonomous AI behavior as quasi-intentional for practical reasons.

Cyber Range for CTU Students (15:50, 5m)

CTU Students: Docker container.

CTU Students, including Prg.AI Master students and students enrolled through the Lifelong Learning program, have access to our CTU Cyber Range.

Connect

Connect to your Docker and test that you can access it:

- You should have received an email from us with all the necessary data.
- Connect now and confirm it works!
- Chat with us on Matrix if you have issues connecting

Important

- Do **not** remove the SSH `authorized_keys` from your Docker container.

- Please do **not** delete the whole Docker container.
- Veronica has a message for you: *"Do not try a fork bomb on the containers, it was already done by past students, thank you very much."*

Warm-up Challenge for CTU Students

Inside your Docker container, there is a hidden file. Try to see if you can find it (15:55, 2m)

SCL for MOOC Students

MOOC Students: SCL HackerLab

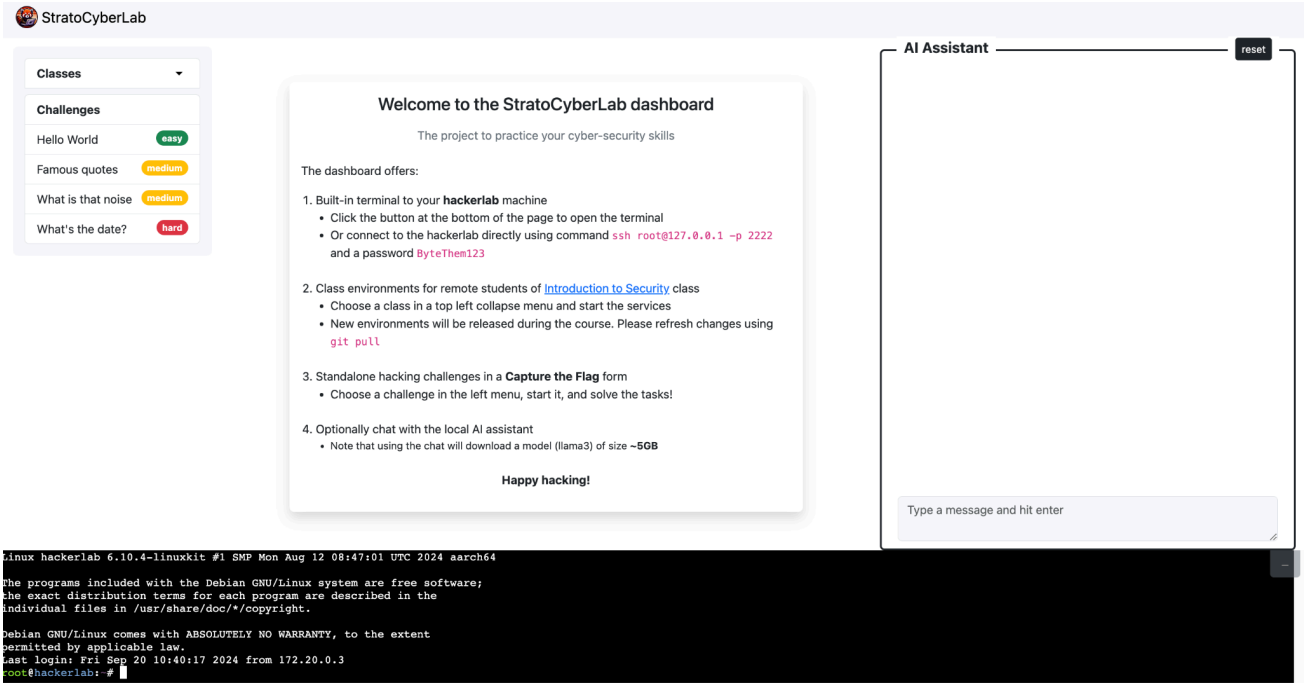
MOOC students can use the **StratoCyberLab**, an offline cyber range created just for you to be able to take the classes and practice your skills through a series of challenges.

You must have the StratoCyberLab working for this class already.

Build and Connect

Once downloaded, you can access the StratoCyberLab using your browser:

- `git clone https://github.com/stratosphereips/stratocyberlab.git`
- `cd stratocyberlab`
- `docker compose up --build --force-recreate`
- `http://localhost/`



Important

- Make sure you pull the latest changes every week with enough time.
- If something breaks or is not working, chat with us through Matrix on the SCL troubleshooting channel

Warm-up Challenge for SCL

For MOOC students with the SCL, you can try the warm-up challenge

- Go to Challenge “Hello World”
- Start it by clicking ‘Start’
- Solve the Submission Test.
- Submit the flag in the box on the webpage.
- Solve the Hackerlab test
- Submit the flag in the box on the webpage.
- For homework, solve the “Network test”

Basic Linux Commands (15:57, 15m)

CTU Students: Docker container. MOOC Students: SCL HackerLab

The class requires you to use the Linux Docker containers. The more familiar you are with Linux commands, the easier it will be to do the assignments and keep up with the content of the classes.

This section gives you a gist of the most important commands we will be using weekly.

COMMAND	DESCRIPTION	EXAMPLE
ls	List files	`ls -alh` • `a: all files` • l: long listing • h: human sizes
ps	List processes	`ps -afx` • a: all users • f: show full format • x: no controlling terminal
cat	Printing the content of a file on the screen (standard input to standard output)	`cat /etc/passwd`
ifconfig/ip	Info about the network interfaces.	`ifconfig eth0` • Name of the interface you want
ping	Send ICMP ECHO_REQUEST packets to network hosts.	`ping 1.1.1.1` • IP or domain
mkdir	Create a directory	`mkdir /tmp/test`
htop	Show the resources of the system, CPU, and memory, etc.	`htop`

grep	Searches text using patterns	`grep -r assw /etc` • -r: Search recursively in all files in that directory • You can also give a file as a parameter.
find	Search for files and directories based on name, type, size, etc.	`find /etc -name passwd` • -name: Search for a file named passwd
mktemp	Create a temporary file or directory with a unique name	`mktemp /tmp/mytempfile.XXXXXX` • Creates a temp file in /tmp with a random suffix
awk	A text-**processing** and programming language.	`cat /etc/passwd|awk -F':' '{print $1}'`

Vi file editor

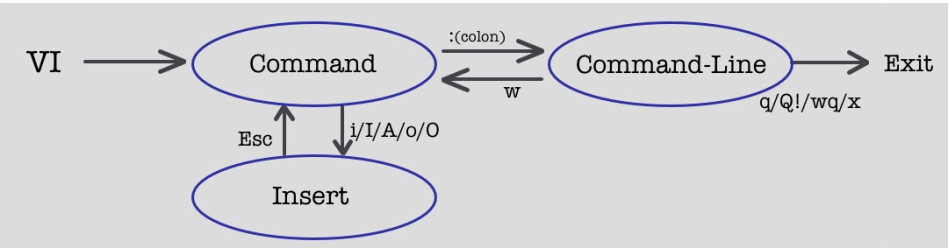
File editor. Just the best editor in the universe.

- `vi test.txt`

The vi editor operates in two modes: command mode and insert mode.

- **In command mode**, you can navigate, delete, and manipulate text
- **In insert mode**, you can type and edit text directly.

To switch modes, press i to enter insert mode and Esc to return to command mode.



How to exit vi:

- Save and exit from command mode: ZZ

- Or use `:q!` To not save and exit.

nano

~~Some~~ ~~awesome~~ weird editor. Whatever. Simple and easy 👍

- Save with CTRL+O
- Exit with CTRL+X

📚**Read More**

- https://cheatography.com/davechild/cheat-sheets/linux-command-line/
- https://overthewire.org/wargames/bandit/

Introduction to Networking (16:12, 15m)

The **OSI**[2] model conceptualizes the communication of a computer system in 7 layers. The **TCP/IP**[3] model defines how it should be implemented in the TCP/IP stack of protocols in 4 layers.

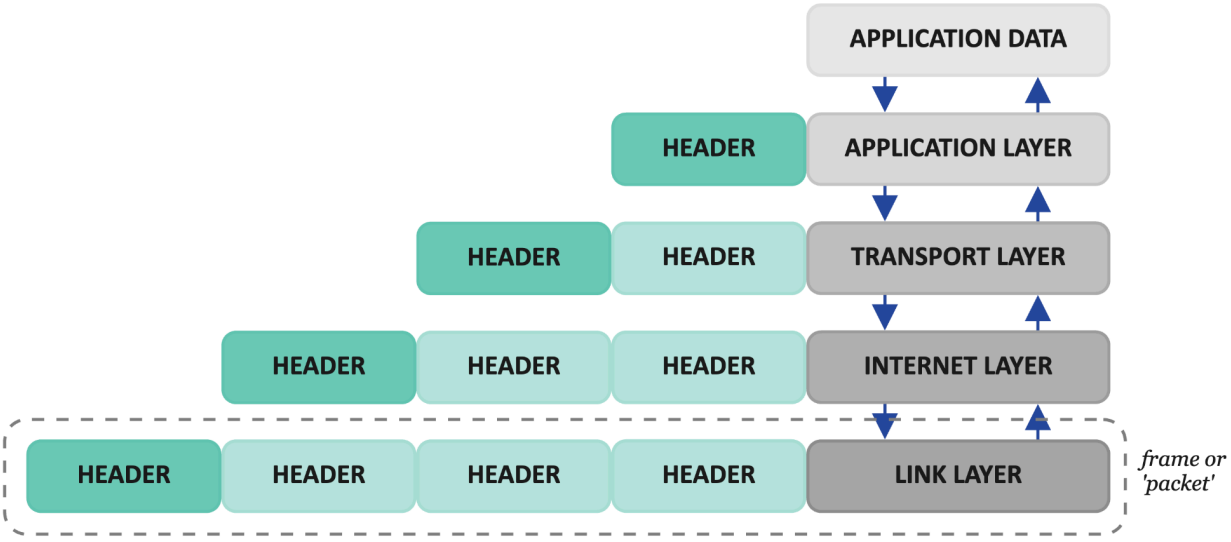
[2] Wikipedia Contributors (2024). OSI model. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/OSI_model [Accessed 24 Sep. 2024].

[3] Wikipedia Contributors (2024). Internet protocol suite. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Internet_protocol_suite [Accessed 24 Sep. 2024].

OSI	TCP/IP	DATA TYPES	PROTOCOLS	DEVICES
APPLICATION	APPLICATION	DATA	DNS, HTTP, HTTPS, FTP, SSH, TELNET, IMAP	LAYER 7 FIREWALL
PRESENTATION				
SESSION				
TRANSPORT	TRANSPORT	SEGMENTS	TCP, UDP	LAYER 4 FIREWALL, WIRELESS AP
NETWORK	INTERNET	PACKETS	IP, ICMP, IPsec, IGMP	ROUTER, WIRELESS AP
DATA LINK	DATA LINK	FRAMES	ARP, ETHERNET	SWITCH, WIRELESS AP
PHYSICAL		BITS	DSL, 802.11, IrDA	HUB, WIRELESS AP

Layers of the TCP/IP protocol suite and some of their protocols.

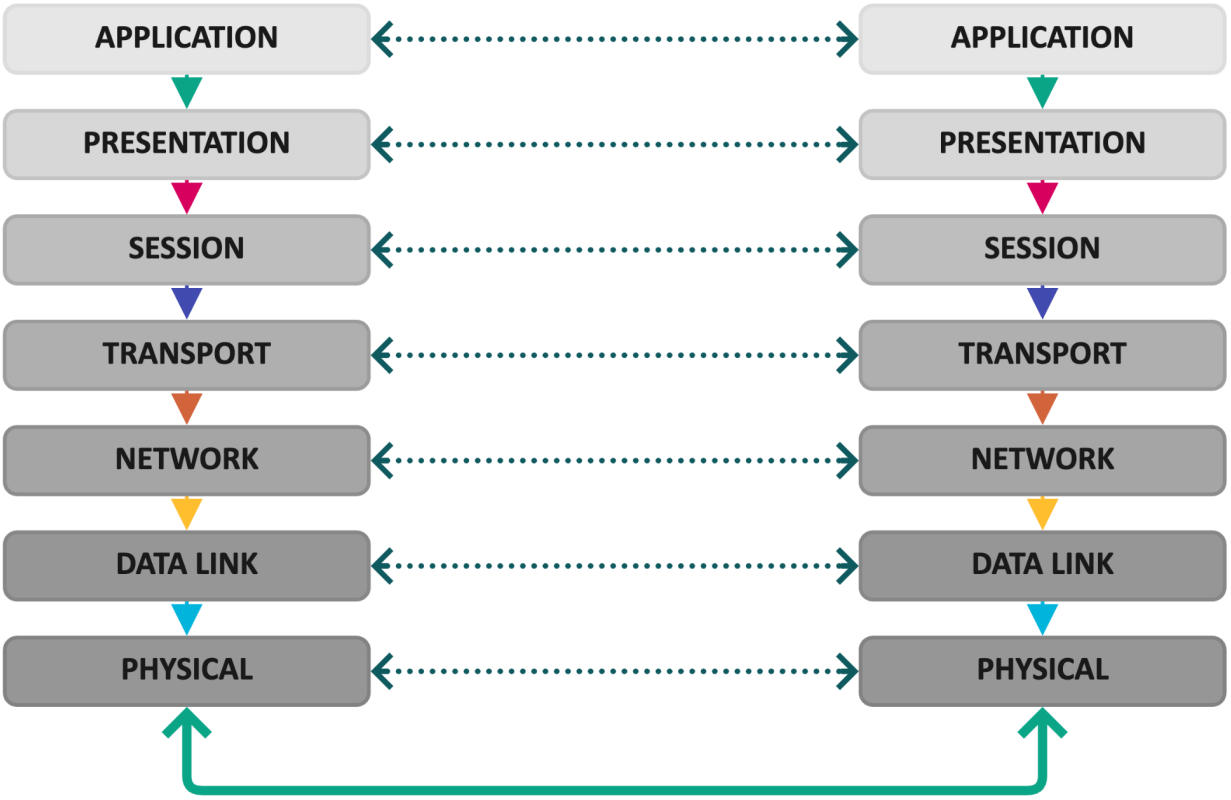
There is **vertical** communication where the information goes vertically from the top of the stack to the bottom, and each layer encapsulates the information given by the upper layer. The information given by the upper layer is the payload of the current layer, which adds a header to it.



Vertical communication in the TCP/IP protocol suite. Each layer encapsulates data coming from the upper layer and sends it to the layer below.

The successive encapsulations end up creating a final packet at the end. (If we are rigorous, the structure created in the link layer should be called a frame, but everybody calls it a packet).

There is also **horizontal** communication. This means that each header on each layer is intended to be read and understood by the same corresponding layer on the other host of the communication. For example, the IP header is written by the IP layer on the source computer, and is meant for the IP layer on the destination host. Therefore, the IP header and data are carried by other layers but read by the IP layer.



Horizontal communication in the TCP/IP protocol suite. The header of each layer is meant to be used by the same protocol in the recipient computer.[4]

[4] Tcpipguide.com. (2024). The TCP/IP Guide - Protocols: Horizontal Communication. [online] Available at: http://www.tcpipguide.com/free/t_ProtocolsHorizontalCorrespondingLayerCommunication-2.htm [Accessed 24 Sep. 2024].

~~~~ 💗 Break! 💗~~~~

Stretch, breathe, and relax! See you all back in 10 minutes!

Basic Networking Protocols

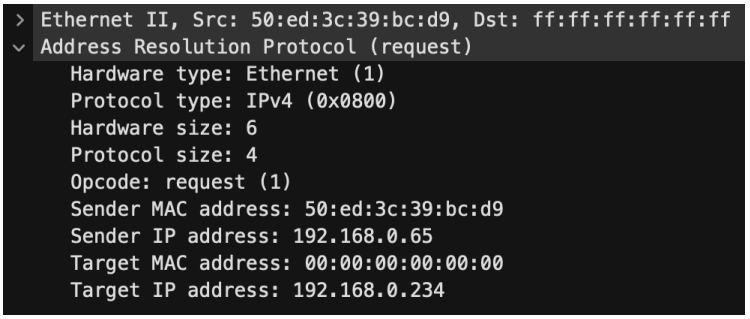
To understand the basic protocols, we will start by looking at packets.
CTU Students: Docker container. MOOC Students: SCL HackerLab

ARP (16:27, 10m)

- Address resolution protocol.
- Helps in the resolution between an **IPv4** address and an **Ethernet** MAC address.
- When going into the network, you always need
  - The **MAC** address of the sender and the destination.
  - The **IP** address of the sender and the destination.

What can you do with ARP besides resolving?

- Update the remote cache of ARP resolutions before they expire.
- Other good uses: Network discovery, monitoring, and collision detection.
- Other malicious uses: cache poisoning, MAC flooding
- Install MOOC Students: SCL HackerLab
  - `apt update`
  - `apt install -y tshark`
- Let's see the headers
  - `tshark -r /data/training-capture-001.pcap -Y "arp" -c 2 -V`
    - `-r: Read a pcap`
    - `-Y: Apply a display filter`
    - `-c: Only analyze the first 2 packers`
    - `-V: Be verbose.`


~~~~

ARP Request and ARP Reply

- The most common types of ARP packets.
- A computer requests to know the MAC address of a destination IP.
- The destination IP replies with its MAC address as the Sender MAC address.

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: RealtekU_12:35:02 (52:54:00:12:35:02)
    Sender IP address: 10.0.2.2
    Target MAC address: PcsCompu_5b:df:e1 (08:00:27:5b:df:e1)
    Target IP address: 10.0.2.16
```

ARP Announcement Packet

- The purpose is also to update the cache of other hosts quickly about your MAC.
- The **Ethernet** header uses the broadcast destination, so every host receives it.
- In the **ARP** header
    - The Sender IP and Target IP are the **same (like you are sending to yourself)**.
    - The target MAC is **zero**, which means all the hosts.
    - The sender MAC is what is **announced**.
- If it has the *opcode* "**request",** it is called a **Gratuitous ARP Request**.

```
> Ethernet II, Src: d4:d4:da:9e:a7:4c, Dst: ff:ff:ff:ff:ff:ff
v Address Resolution Protocol (ARP Announcement)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: True]
    [Is announcement: True]
    Sender MAC address: d4:d4:da:9e:a7:4c
    Sender IP address: 192.168.0.139
    Target MAC address: 00:00:00:00:00:00
    Target IP address: 192.168.0.139
```

- The *announcement* can also be done in a broadcast *ARP reply* type of packet.
    - Both IPs should be the same, and both MACs should be the same.

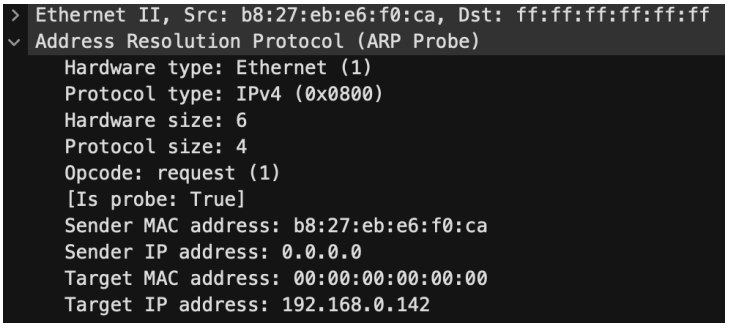
Unsolicited ARP reply

- The type of the ARP packet is *reply.*
- It is also called **Gratuitous ARP Reply**.
- The purpose is also to update the cache of other hosts quickly about your MAC.
- In the Ethernet header, the destination MAC is broadcast
- In the ARP header

    - **Sender MAC** is the host's real MAC
    - **Target MAC** is all zeros
    - Both IPs are the sender's IP

ARP Probe Packet

- The goal of the ARP Probe is to poll the network to validate that an IP address is not already in use.
- **Opcode** field set to 1, a **Request**.
- The **Sender MAC** address is set to the initiator's MAC address.
- The **Target MAC** address is set to **zeros.**
- The **Sender IP** address is set to **0.0.0.0**.
- If the IP address in question is in use, we expect a Reply.



Related Commands

- `arp -a`
    - This command shows the local cache of IP and MAC addresses of your computer.

ICMP (16:37, 3m)

Internet Control Message Protocol

In its header, it has a *type* field (255 options) and a *code* field that defines its functionality. Most common:

- Type 0, code 0: **Echo reply** (what ping receives)
- Type 8, code 0: **Echo request** (what ping sends)
- Type 3, various codes: **Destination unreachable**.
- Type 11, various codes: **Time exceeded in transit**.

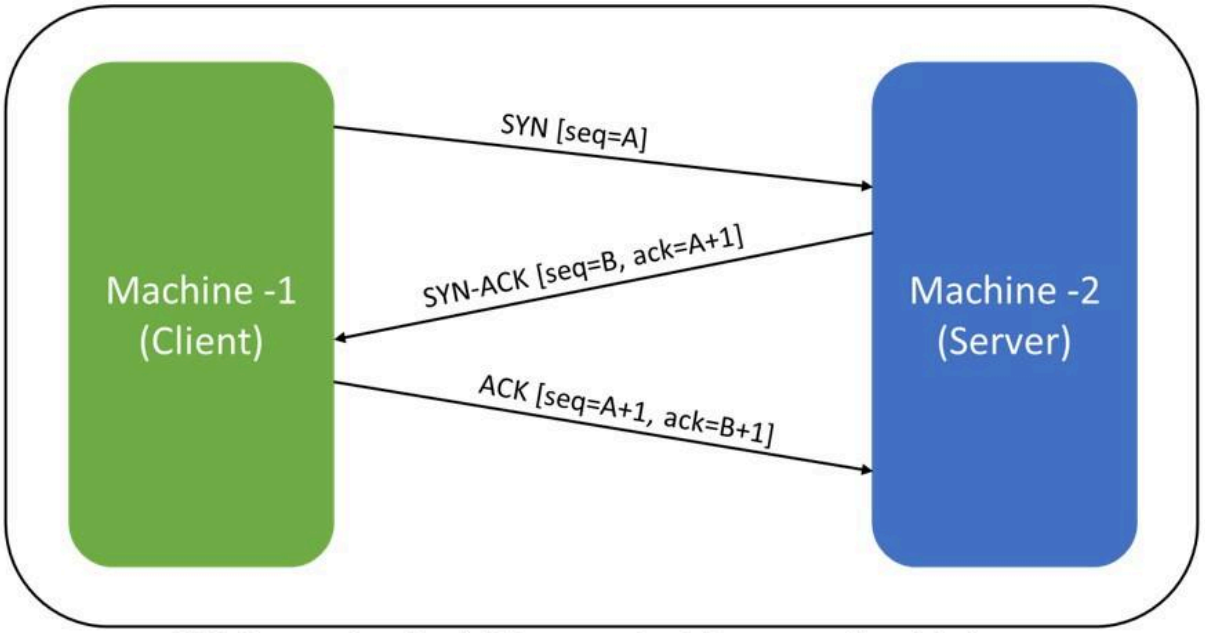
Related Commands

- `ping 123.123.123.123`

TCP (16:40, 5m)

You don't need the transport layer to move data between computers, but you do need it to deliver that data to the right application.

TCP is well-known, connection-oriented, and has error corrections, re-transmission functionalities, and congestion controls. Designed not to lose one piece of data if possible. Here is where the typical **three-way handshake** happens:



TCP Connection Establishment using Three-way Handshake

- Let's see the headers
  - `tshark -r /data/training-capture-001.pcap -Y "frame.number == 25" -V`

UDP (16:45, 3m)

UDP is not connection-oriented, which means that there is no guarantee that the packets arrived. It does not create a connection. You send the data, and that's it. You don't care if it arrives. Errors in reception are ignored at this level and left to be handled by the upper layers.

Why do we use UDP?

- Let's see the headers
  - `tshark -r /data/training-capture-001.pcap -Y "frame.number == 21" -V`

HTTP (16:48, 20m)

- Hypertext Transfer Protocol.
- The "Web" protocol.
- The most complex and extended protocol so far. Everything runs on HTTP.
- Many methods
  - GET, POST, DELETE, PUT, HEAD, OPTIONS, CONNECT

HTTP/1.0 and HTTP/1.1[5]

- Text-based
- A valid HTTP request **must** have at least:
  - One line with the method (GET, POST, etc.), URI (/), and version (HTTP/1.1)
  - One line with the Host: www.google.com. Host: 23.23.23.23 header
  - Two newlines.

Let's try it!

- Use ncat to connect to any server and port.

[5] Wikipedia Contributors (2024). HTTP. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/HTTP [Accessed 24 Sep. 2024].

    - Ncat is a tool from the nmap suite that allows you to connect to a port, send something, and receive something.
- `ncat 1.1.1.1 80`
    - `GET /index.html HTTP/1.1`
    - `Host: 1.1.1.1`
    - (Press Enter twice)

HTTP/2[6]

- What is it?
- More efficient: Compression of headers
- SPDY[7] protocols: 443 on UDP?
- Push from servers: Data without the request!!
- Several resources in one TCP connection. Several connections per flow.
- It's binary
- Should use the same URLs! So, how to ask about the new protocol?
    - Upgrade: header
- Are you using HTTP/2?
    - Open Google Chrome and browse www.google.com
    - Right-click + Inspect (or press F12)
    - Go to 'Network section'
    - If needed, right-click on the headers and add the protocol column
    - Check which protocol you are using.
- Let's try by hand
- MOOC Students: SCL HackerLab
    - `apt install -y nghttp2`
- CTU Students: Docker container. MOOC Students: SCL HackerLab

[6] Wikipedia Contributors (2024). HTTP/2. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/HTTP/2 [Accessed 24 Sep. 2024].

[7] Wikipedia Contributors (2024). SPDY. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/SPDY [Accessed 24 Sep. 2024].

- `nghttp -vn https://www.google.com/`

HTTP/3[8][9]

- Improvement on HTTP/2
- No more TCP! 😮 Only UDP + QUICK.

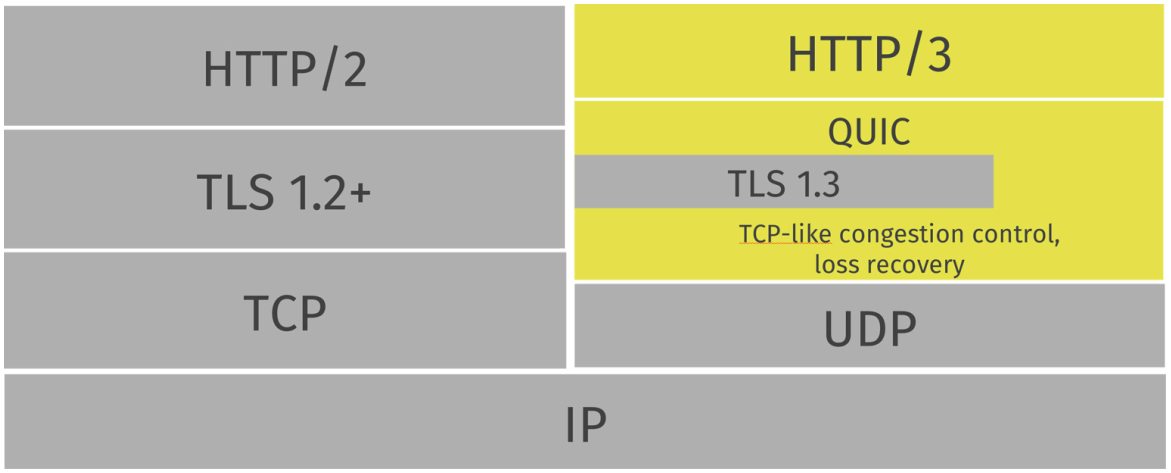


Image credit to Haxx.se[10]

QUICK is many things, but one property kind of unknown is that it can migrate connections from one interface to another. Such as from cable to WiFI.

Let's try

- CTU Students: Docker container. MOOC Students: SCL HackerLab
- `apt update`
- `apt install -y ngtcp2-client`
- `gtlsclient cloudflare-quic.com 443 https://cloudflare-quic.com/ 2>&1 |less`

[8] Wikipedia Contributors (2024). HTTP/3. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/HTTP/3 [Accessed 24 Sep. 2024].

[9] Haxx.se. (2023). README | HTTP/3 explained. [online] Available at: https://http3-explained.haxx.se/ [Accessed 24 Sep. 2024].

[10] Haxx.se. (2020). Protocol features | HTTP/3 explained. [online] Available at: https://http3-explained.haxx.se/en/the-protocol [Accessed 24 Sep. 2024].

TLS[11][12] (17:08, 3m)

TLS (Transport Layer Security) is a security protocol used to protect data transmitted over networks. It encrypts the connection between a client (e.g., a web browser) and a server, ensuring confidentiality, integrit, and authentication.

- How does it work?
    - Handshake: The client and server agree on encryption methods and exchange keys.
    - Encryption: All data sent between them is encrypted, so no one else can read it.
    - Integrity: It checks that the data isn't tampered with during transmission.
- Public/Private keys
    - Public Key: Shared openly. Used to encrypt data.
    - Private Key: Kept secret. Used to decrypt data and to sign.
    - Mathematically: What you encrypt with one can only be decrypted with the other (not even the same). But don't encrypt with the private in real life.
- Symmetric Keys:
    - One Key: The same key is used for both encryption and decryption.

📚Read More about TLS

- https://tls12.xargs.org/

DNS[13] [14] (17:11, 10m)

A **distributed and hierarchical protocol** that maps domain names to IP addresses and vice versa.

---

[11] Xargs.org. (2017). The Illustrated TLS 1.2 Connection. [online] Available at: https://tls12.xargs.org/ [Accessed 24 Sep. 2024].

[12] Xargs.org. (2024). The Illustrated TLS 1.3 Connection. [online] Available at: https://tls13.xargs.org/ [Accessed 24 Sep. 2024].

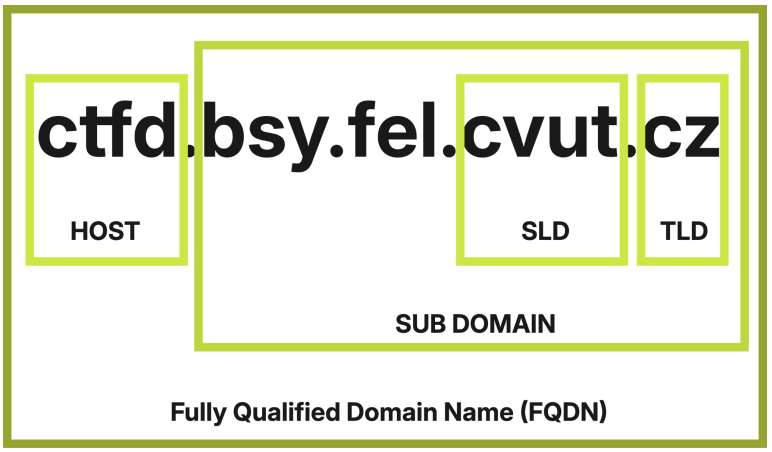
[13] DNSimple (2024). How DNS works. What is DNS? Learn how step by step. [online] Available at: https://howdns.works/ [Accessed 24 Sep. 2024].

[14] Wikipedia Contributors (2024). Domain Name System. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Domain_Name_System [Accessed 24 Sep. 2024].

- **Domain names** (and **hostnames**) are translated into IP addresses to locate services on the Internet.
- DNS can also resolve **IP addresses back to domain names** (reverse lookup).
- It supports different types of records, including:
    - **A** and **AAAA** records: Map domain names to IPv4 and IPv6 addresses.
    - **TXT** records: Store arbitrary text data (often used for verification).
    - **MX** records: Define mail servers for handling email.

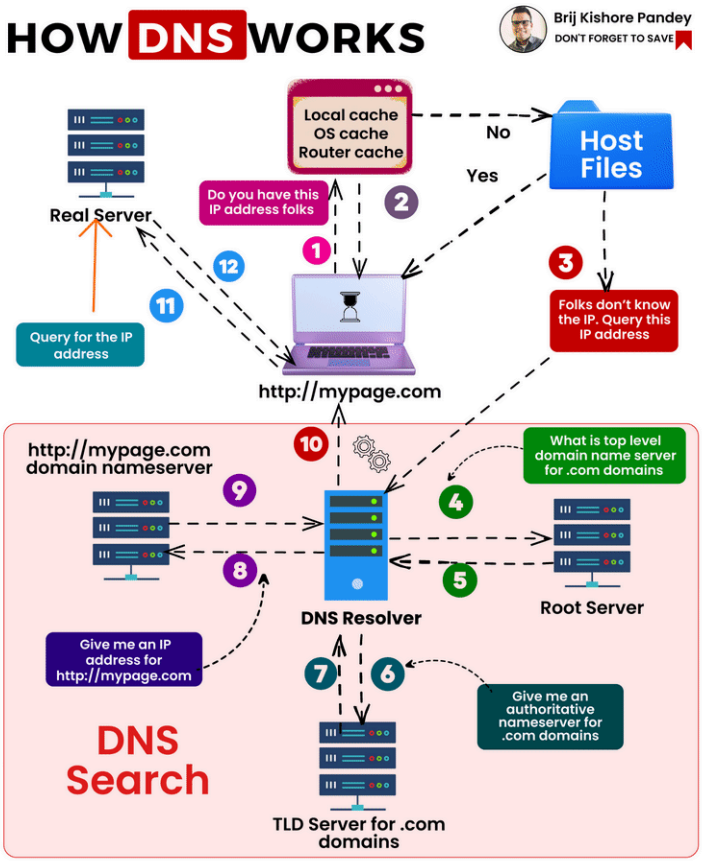
What is a domain?

A **domain** is a human-readable name (like `example.com`) that corresponds to an addressable entity on the Internet.



How is the resolution done? (Image credit to Brij Kishore Pandey[15])

[15] Brij kishore Pandey (2023). How DNS Works [online] Linkedin.com. Available at: https://www.linkedin.com/posts/brijpandeyji_softwareengineering-networkadministration-activity-7086305973648171008-Gf-i/ [Accessed 24 Sep. 2024].



Let's get some!

- Normal resolution
  - `dig test.com`
- Reverse resolution
  - `dig -x 1.1.1.1`
- Ask all public records
  - `dig -t any cvut.cz`
- Ask all public records, asking the authoritative name server (NS)
  - `dig @ns.cvut.cz  -t any cvut.cz`
- To see the whole process, simulating a non-recursive query:
  - `dig +trace test.com`

IPv6 (17:21, 2m)

Latest IP protocol. 128-bit addresses.

- How big is 128 bits? 2^128?
    - In base10: 3.4 x 10^38 IPv6 is the real address space
    - Compared
        - 4.3 x 10^26 Nanoseconds since the Big Bang.
        - 10 million trillion times the total sand grains on Earth.
        - If a new unique IPv6 address were assigned at every picosecond (one trillionth of a second) after one trillion years, there would still be lots and lots of IPv6 addresses.
        - 1.12^19 IPV4 Internets for each of the 7 billion humans.
- Are you using IPv6? Are you prepared for it?
- Try some on your own computer to see IPv6 neighbors (Docker dont have it). It does not send packets, just shows the cache.
    - `ip -6 neigh show`

NDP (17:23, 3m)

Neighbor Discovery Protocol. Only in IPv6. Designed to replace ARP from IPv4 and some ICMPv4.

It defines five ICMPv6 packet types.

Router Solicitation

Hosts inquire with Router Solicitation messages to **locate routers** on an attached link.

Router Advertisement

Routers advertise their presence together with various link and Internet parameters either periodically or in response to a Router Solicitation message.

Neighbor Solicitation

Neighbor solicitations are used by nodes to **determine the link-layer address of a neighbor** or to **verify** that a neighbor is still reachable via a cached link-layer address.

Neighbor Advertisement

Nodes use Neighbor advertisements to **respond to a Neighbor Solicitation** message or unsolicited to provide new information quickly.

Redirect

Routers may inform hosts of a better first-hop router for a destination.

🤖AI View in protocols:

- AI is already making a difference in fuzzing protocols for finding vulnerabilities [16].

Testing Methodologies (17:26, 3m)

This class follows a combination of a **penetration test methodology** and **system defense methodology**. Unfortunately, there is **no** *unique and simple* ***defense*** *methodology* to follow. You just need to defend everything all the time.

Penetration Test Methodology

A real penetration test differs from what we do in class due to time and legal constraints.

A real penetration test:

- It has a client.
- It has a **contract** with the client (not in this class).
- It can be of different types: whitebox, blackbox, on-premises, or software only.
- It has to find all the vulnerabilities.
- It has to give a comprehensive **report** at the end (not in this class).
- It has to present the results to the client (not in this class).

---

16 https://www.researchgate.net/publication/389002661_AI-Enhanced_Protocol_Fuzzing_Integrating_Machine_Learning_with_Defensics_for_Advanced_Vulnerability_Detection#:~:text=This%20integration%20of%20AI%20capabilities,detection%20across%20diverse%20deployment%20scenarios.

- It usually does not exploit the services. Not heavily.
- It usually does not try to attack other users.

Guides to a Penetration test methodology (17:29, 1m)

We would like to have a set of principles and guidelines for doing a penetration test to maintain a certain quality. Unfortunately, there are no current, complete methodologies. ISSAF[17] is from 2006, and the OSSTMM[18] is from 2000.

Our simplified penetration test methodology (17:30,5m)

This is a simplified methodology that we will follow:

1. Document all the penetration testing processes.
    - Write down what you do and when.
    - It is a very good practice for cybersecurity practitioners to always document every step. It can be an attack, a defense, a challenge in a CTF, etc. This is the base for good reporting, but also for legal defenses and (most importantly for you) **defending a grade**.
2. Plan
    - Define the IP ranges and domains you should be testing and which ones you should **not** be testing. This is called *the scope*.
    - Define your **goal**. Examples:
        - Only to find vulnerabilities.
        - To test a specific system.
        - To see if you can get access.
        - To see if you can exfiltrate data.

[17] Information Systems Security Assessment Framework (ISSAF) draft 0.2. (n.d.). Available at: https://untrustednetwork.net/files/issaf0.2.1.pdf. [Accessed 24 Sep. 2024].

[18] Isecom.org. (2024). Open Source Security Testing Methodology Manual. [online] Available at: https://www.isecom.org/OSSTMM.3.pdf [Accessed 24 Sep. 2024].

3. Recognize the attack surface
    - Find hosts, servers, applications, API,s and routers.
4. Identify the services to attack
    - Identify each port and its version.
5. Find vulnerabilities
6. Plan the attack and do the attack
    - Consider detection measures.
7. Escalate privileges if needed
8. Gain persistence
9. Lateral movement
10. Exfiltrate
11. Hide/finish/clean your tracks
12. Report the results

A Penetration Test Methodology is Not How Attackers Attack (17:35,2m)

A penetration test methodology is an organized way to test the security of a system, but it is optimized as a business. Attackers do many more different things:

1. They have much, much more time. Years.
2. They can wait until new vulnerabilities appear.
3. They can develop/research exploits just for the target.
4. They can do a DoS to hide the real attack.
5. They can destroy/delete/wipe.
6. They can blackmail, extort, etc.
7. They have no legal boundaries.

A Defense Methodology (17:37,2m)

NIST did a good job of summarizing the basics of defense[19] [20]. Including one of the best all-around guides[21]. But there are many more guides, like MITRE D3FEND.

1. Preparation & Asset Visibility (Identify)
    a. Maintain an asset inventory (devices, software, users, cloud resources).
    b. Classify assets by criticality and risk.
    c. Build a threat model to understand likely attacker paths.
2. Hardening & Exposure Reduction (Protect)
    a. Apply secure configuration baselines.
    b. Manage vulnerabilities with patching and compensating controls.
    c. Limit access with least privilege & multi-factor authentication.
3. Monitoring & Detection (Detect)
    a. Deploy logging, SIEM, IDS/IPS, EDR/XDR.
    b. Use anomaly detection and AI/ML for behavior baselining.
4. Incident Response (Respond)
    a. Establish an incident response plan.
    b. Contain attacks quickly (isolate endpoints, block malicious domains).
    c. Communicate with stakeholders and legal/compliance teams.
5. Recovery & Improvement (Recover)
    a. Restore affected systems and data securely.
    b. Conduct post-incident reviews to identify gaps.
    c. Feed lessons learned back into hardening & detection improvements.

A Comparison of Penetration Test and Defense Methodologies (17:39,2m)

Phase	Penetration Testing	Defensive Security

[19] NIST. (2013). Understanding the Cybersecurity Framework [online] Available at: https://www.ftc.gov/system/files/attachments/understanding-nist-cybersecurity-framework/cybersecurity_sb_nist-cyber-framework.pdf [Accessed 24 Sep. 2024].

[20] NIST. (2013). Cybersecurity Framework [online] Available at: https://www.nist.gov/cyberframework [Accessed 24 Sep. 2024].

[21] NIST (n.d.). Cybersecurity for Small Business. [online] Available at: https://www.ftc.gov/system/files/attachments/cybersecurity-small-business/cybersecuirty_sb_factsheets_all.pdf. [Accessed 24 Sep. 2024].

1. Preparation	Pre-engagement interactions: scope, rules of engagement, legalities	Identify: assets, business context, risk assessment
2. Reconnaissance	Intelligence gathering: open source, scanning, fingerprinting	Protect: asset inventory, vulnerability management, access control
3. Threat Modeling	Threat modeling: choose likely attack paths based on gathered info	Identify/Protect: threat modeling and prioritization
4. Exploitation	Exploitation: gaining access, privilege escalation, pivoting	Detect: monitoring, anomaly detection, IDS/IPS
5. Post-Exploitation	Post-exploitation: data exfiltration, persistence, lateral movement	Respond: containment, eradication, communication
6. Reporting	Reporting: vulnerabilities found, impact, recommendations	Recover: restore systems, lessons learned, improve resilience

CTU Cyber Lab Students: Assignment 1 (1 Point) (17:41,2m)

Assignment 1 (1 Point)

1. There is a file containing a flag somewhere in your class Docker container.
2. You must find it and submit the flag in CTFd in the Assignment 1 challenge field.
3. The assignment opens on Thu 25.09, 21:00 CEST
4. More details in CTFd.
5. **DEADLINE: 30.10.2025 23:59 Prague time**

Class Feedback (17:43,2m)

By giving us feedback after each class, we can make the next class even better!

https://bit.ly/BSY-Feedback

Side Dish 1: The Economics of Security

Why do we still have security problems? Why can't we just solve all of them?

Security problems are inevitable due to the continuous emergence of new vulnerabilities in software, hardware, networks, and through human error.

- Absolute security is unattainable, forcing developers to prioritize which issues to address based on costs, risks, and impact.
- Economic incentives often drive companies to release products quickly with inadequate security, while attackers face little deterrence.
- Enhancing security requires aligning incentives through better regulations, standards, and education, as well as understanding the defender-attacker dynamic to make more informed security decisions.

New security problems always appear in old and new software, new attacks, new software, new programming languages, new hardware, new software integrations, new networks, new humans developing badly, etc.

Imagine you are creating a new software. There's always going to be some new vulnerability that affects you. This means you will never fix all the security problems perfectly and forever. It does not matter how much money you put in.

Then, as creators, you need to choose what you will and will not fix.

Let's Talk About Online Banking

Remember all the security problems with online banking. We can stop all those problems by not using online banking anymore. But most of humanity lives and works thanks to online banking, so the economic cost of stopping it is more than the money lost to vulnerabilities and fraud online.

This means that there is an optimal level of insecurity that is acceptable.

- How do you know which security vulnerabilities to fix first?
    - How much money and effort do you put into fixing?
    - Should you focus on what attackers attack more, that is cheap to secure?
    - Should you focus on the most valuable asset that is expensive to secure?
- And more importantly, as an attacker, how do you know which security vulnerabilities you attack first?
    - How much money and effort do you put into attacking?
    - Do you attack the lowest-hanging fruit that is cheap to access?
    - Do you attack the hard service that is expensive to access?

This balance between attackers and defenders defines the behavioral dynamics of the economics of security. Knowing the incentives of every player is very important to make decisions.

Let's Talk About Data Leaks From Customers

Why do very few companies put effort into protecting the security of the data of their customers? Why, if they can, hide the fact that they lost data?

- Because they are not directly affected by the problem. It is not their data.
- There are no incentives to protect it.
    - No lawsuits.
    - No regulation breaks.
    - No stock is lost if hidden.
    - No intellectual property lost.
- People responsible for protecting do not suffer the consequences of their errors.

And why can some people steal money, sell your data, use your identity, and ruin your life?

- Because they are not affected by these actions. It is not their data.
- There are no incentives not to do it.

Let's Talk About Your Next New Product

- So you want to create your next software that is super duper secure.
- But the pressure of the market will drive you first to have software that works well, people use that, and that has value.
- The cost of having a security vulnerability when nobody uses it is almost zero. No incentive to fix all of them.
- It is economically optimal to deploy early versions that are working but may be insecure—the 'eternal beta' paradigm.
- Only when the cost of having a vulnerability is high to the users does it make economic sense to fix it.

This economic security principle is called the misalignment of incentives and is why most security problems exist.

It may be fixed. Maybe.

- For defenders, with regulations, standards, laws, and obligations to share information about attacks.
- For attackers, with law enforcement and sanctions. And, of course, with education and awareness.

Let's Talk About Your New Product Again

- You want to make a super duper secure soft, and you did it!! It is crazy amazing.
- In security, customers do not have a way to verify that a product is more secure than others. There are no good ways to compare, test, or objectively know.
- You have to trust what the vendor tells you. But sometimes, not even the vendor knows if the product is better.
- This creates an economic market where nobody will pay more for a product they don't know is better.
- Then the market is filled with cheap, bad-quality products. Customers can not differentiate.
- This is called a 'lemon market'. The person who discovered the principle got a Nobel Prize.
- Example on Android applications[22]

This economic security principle is called the asymmetry of information between buyers and sellers.

[22] Android Test 2019 - 250 Apps. (2022, April 14). Retrieved April 27, 2025, from AV-Comparatives website: https://www.av-comparatives.org/tests/android-test-2019-250-apps/

Let's Talk About Your New Intrusion Detection System

So you get a new IDS, detect every infected computer in your network, and stop it.

- This is good! But the effects of this good security are not **all** seen by you.
- The Internet is a better place, too, and people are attacked less.

This economic security principle is called the externalities of your decisions. All those who are affected who are not you[23].

Analyzing cybersecurity as a defender-attacker dynamic is more promising, emphasizing the **incentives** rather than only focusing on the vulnerabilities.

[23] Ok, ok. Externalities are all those third-parties impacted by the production or consumption of a good that were not directly related to the production or consumption of that good.