INTRODUCTION TO THE CLASS, SECURITY, AND NETWORKING

September 26th, 2024

"The one with all the introductions."

Credits

Content: Sebastian Garcia
Ondřej Lukáš
Maria Rigaki
Martin Řepa
Lukáš Forst
Veronica Valeros
Muris Sladić
Illustration: Fermin Valeros
Design: Veronica Garcia
Veronica Valeros
Ondřej Lukáš
Music: Sebastian Garcia
Veronica Valeros
Ondřej Lukáš

CLASS DOCUMENT	https://bit.ly/BSY2024-1
WEBSITE	https://cybersecurity.bsy.fel.cvut.cz/
CLASS MATRIX	https://matrix.bsy.fel.cvut.cz/
IN-PERSON CLASS CTFD	https://ctfd.bsy.fel.cvut.cz/
FEEDBACK	https://bit.ly/BSYFEEDBACK
CLASS EMAIL	13136-bsy@fel.cvut.cz
LIVESTREAM	https://www.youtube.com/watch?v=ye8lGXTcdCE&list=PLQL6z4JeTTQkqF6KkcZZDi2KFwky9SQpq&index=1
LET'S START	https://www.youtube.com/watch?v=lzuJpFs2u4s

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

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 first time, this “Introduction to Security” course at the Czech Technical University in Prague (CTU), also coded BSY, is **both in-person and online**!

Welcome everyone!

Class Dynamics

The “Introduction to Security” class 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/

Teachers and Their Motivation (14.41, 5m)

The BSY teachers belong to the **Stratosphere Laboratory**, a cybersecurity research group at the Artificial Intelligence Center, Dept. of Computer Science, Faculty of Electrical Engineering, CTU. Website: https://www.stratosphereips.org.

Stratosphere works at the intersection of machine learning and security.


Sebastian Garcia


Maria Rigaki


Ondřej Lukáš


Veronica Valeros


Lukáš Forst


Martin Řepa


Muris Sladić

Bios of Teachers

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 @AI Center, project lead @Stratosphere, and 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. Likes 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. 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 LLMs. He enjoys traveling, playing music, writing, and puzzle games.

Two Modalities of this Class (14.46, 2m)

- **In-person** students physically here in CTU in Prague.
- **Online** students from around the world on the Internet.
- Different requirements to finish the class
    - For CTU students
        - You need to approve the class itself by doing extra assignments.
        - You need to approve the final exam or approve the bonus assignment.
    - For online students
        - No assignments or exams for you.
        - If you do the whole class, in January, you will get a *certificate of completion* from Stratosphere Lab, not from CTU. You don't get official credits from it.
        - In 2025 we will open a third option, an online paid version with exams, credits, and a CTU official certificate.

Teaching Methodology (14.48, 5m)

- The class will be held every Thursday (Subscribe to Google Calendar).
    - Starts at **14:30 Prague time.** **UTC+2** now (CEST) and after October 27th is **UTC+1** (CET). See https://time.is/Prague
        i. **At 15:25hs,** there is a break of 10mins.
        ii. **At 16:30hs,** there is a break of 10mins.
    - Ends at **17:45hs**.

Class Block I	**55 min**
Break	10 min
Class Block II	**55 min**
Break	10 min
Class Block III	**65 min**

- Classes will be live-streamed on YouTube.
- We use Google Docs because this 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.
- For **in-person** students.
    - We run Linux Docker containers in our Stratosphere Laboratory servers.
    - You can ssh from your laptop to your personal Docker in our infrastructure.
    - Each student will be assigned a personal Docker container.
    - Each student will have to defend this Docker container.

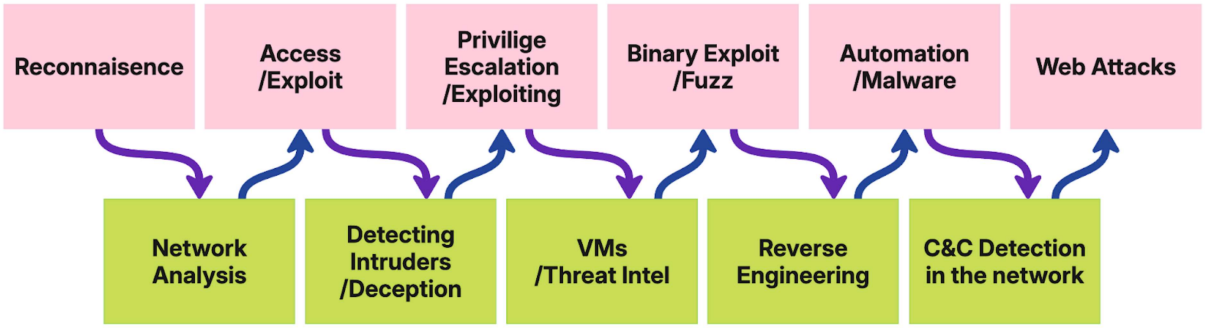
    - Each student will attack other Dockers containers to accomplish different goals.
    - Your goal as an attacker is NOT to be detected.
    - Your goal as a defender is to detect all the attackers.
- For **online** students, you will use our **StratoCyberLab** (https://github.com/stratosphereips/stratocyberlab).
    - StratoCyberlab was specifically developed for you. It uses orchestrated dockers in your computer to be completely **off-line**. 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. Like now!**

Communication (14.53, 2m)

- Questions and discussions will be 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 **#live-questions** 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 all the teachers can see it.**

Class Methodology (14.55, 3m)

1. 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.

Assignments for In-person 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 docker network of stratosphere for in-person 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.

Special Assignments for Online Students (15:08, 1m)

For online students, the **StratoCyberLab** **may** have, from time to time, new challenges for you to play. But this year, we will **not** grade them.

We don't promise there will be, just maybe.

Grading for In-Person 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/requirements/ctustudents/#grading-scale

Certificate of Completion. Online Students (15:12, 3m)

1. We are going to check if you attended and saw the videos for the classes.
2. You can **only** miss two classes. So from the 14 classes, you must have done at least 12.
3. How do we check?
    a. We show and say a secret password during the live class. It is not going to be in this document.
    b. You must see the class live or the recording.
    c. Then you go to this place and put the password: https://bit.ly/BSY-VerifyClass
4. The password for this first 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 the Docker for tasks other than the ones 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 the other **students' Dockers.**

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.

~~~~ 💗 First Break! 💗 ~~~~ (~15:20)

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 the data without authorization.
    - More technically: "information is not made available or disclosed to unauthorized individuals, entities, or processes" [wiki]
    - E.g., encryption is one way we can force confidentiality
- **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."
    - E.g., hashing functions are a way to check the integrity of data.
- **Availability**
    - Data should be available for those authorized when needed.
    - Everything 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."
- **Non-repudiation**
    - Neither party can deny sending, receiving, or accessing the data.

Transversal pillars

- **Authentication**: the ability to confirm with a high degree of certainty that they are indeed who they say they are
- **Authorization**: Once authenticated, what you are authorized to do or not to do.

⚔️ 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?

Connecting to Dockers. In-person (15:50, 2m)

Let's connect to your dockers and test that you can access.

- You should have received an email from us with all the necessary data.
- Connect now and confirm it works! If not, chat with us in Matrix.

- Do **not** remove the authorized_keys from the SSH.
- Please do **not** delete the whole docker.

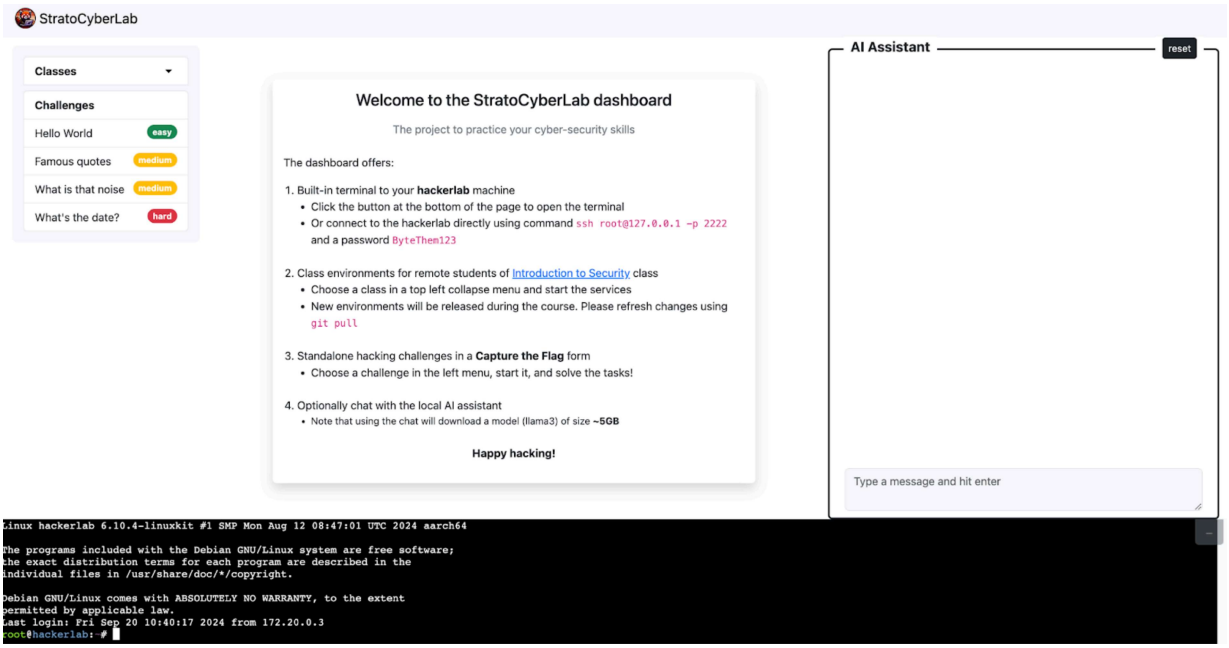
- Veronica has a message for you: “Do not try a fork bomb on the containers, it was already done by past students, thank you very much.”

Warm-up time! Find the file with the flag in the Docker and input the flag on CTFd. This is a warm-up exercise, and it will not give you points, but it will check you can use docker and CTFd correctly.

Connect to StratoCyberLab. Online (15:52, 5m)

You must have the StratoCyberLab working for this class already.

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



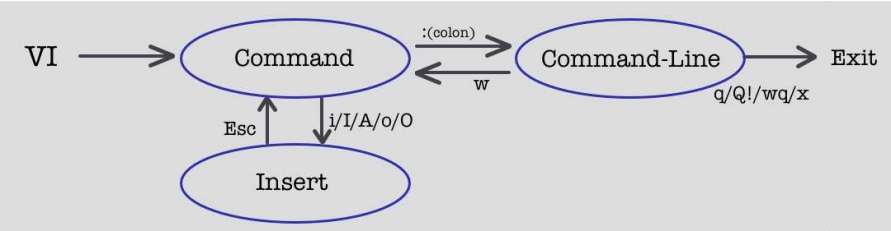
Warm-up time! Find the file with the flag in the Docker and input the flag on the “Hello World” challenge.

Basic Linux Commands (15:57, 25m)

- ls
  List files
  - `ls -alh`
    - a: all files

        - l: long listing
        - h: human sizes
- ps
  List processes
    - `ps -afx`
        - a: all users, not just you.
        - f: Display the uid, pid, parent pid, recent CPU usage, process start time, controlling tty, elapsed CPU usage, and the associated command.
        - x: Include processes that do not have a controlling terminal
- cat
  Printing the content of a file on the screen (standard input to standard output)
    - `cat /etc/passwd`
- ifconfig
  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`
        - Directory name
- htop
  Show the resources of the system, CPU, and Mem, etc.
- vi
  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 editor. Whatever.

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

- grep

  Searches text using patterns

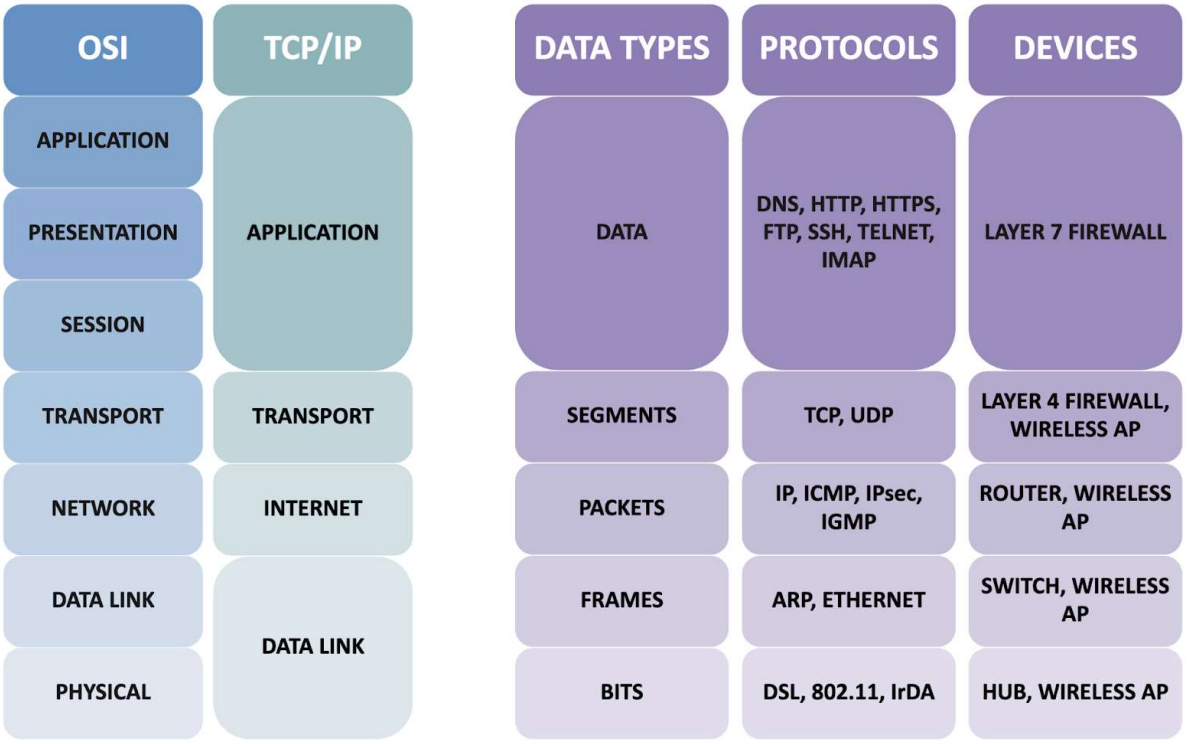
  - `grep -r assw /etc`
    - -r: Search recursively in all files in that directory
    - You can also give a file as a parameter.

📚 Read/Do More

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

Introduction to Networking (16:22, 5m)

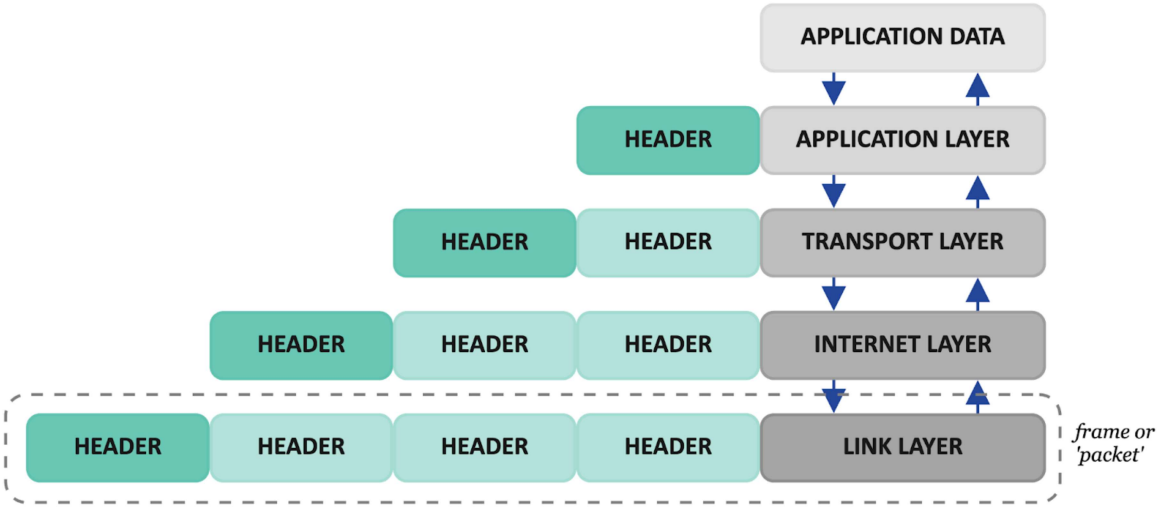
The **OSI**[1] model conceptualizes the communication of a computer system in 7 layers. The **TCP/IP**[2] model defines how it should be implemented in the TCP/IP stack of protocols in 4 layers.



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

We say 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.

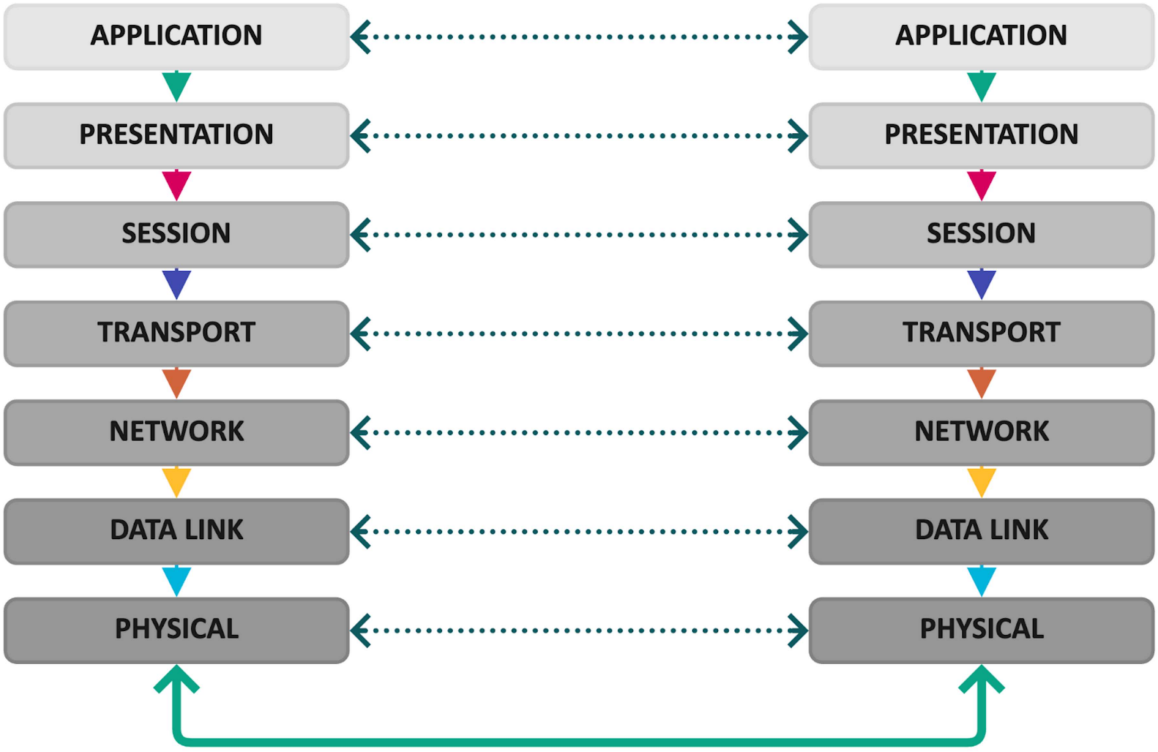
[1] Wikipedia Contributors (2024). OSI model. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/OSI_model [Accessed 24 Sep. 2024].
[2] Wikipedia Contributors (2024). Internet protocol suite. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Internet_protocol_suite [Accessed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



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 meant to be read and understood by the same corresponding layer on the other host of the communication. So, for example, the IP header is written by the IP layer on the source computer, and it is read by the IP layer on the other host of the communication. Therefore, the IP header and data are carried by others 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.[3]

~~~~ 💗 Second Break! 💗 ~~~~ (~16:27)

Basic Networking protocols

ARP (16:27, 10m)

- Address resolution protocol.
- Makes the resolution between an IPv4 address and an ethernet MAC address.
- Needs the MAC address of the sender and the target.
- Also needs the IP address of the sender and target.

[3] 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].

```
> Ethernet II, Src: 50:ed:3c:39:bc:d9, Dst: ff:ff:ff:ff:ff:ff
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 50:ed:3c:39:bc:d9
    Sender IP address: 192.168.0.65
    Target MAC address: 00:00:00:00:00:00
    Target IP address: 192.168.0.234
```

ARP Announcement Packet

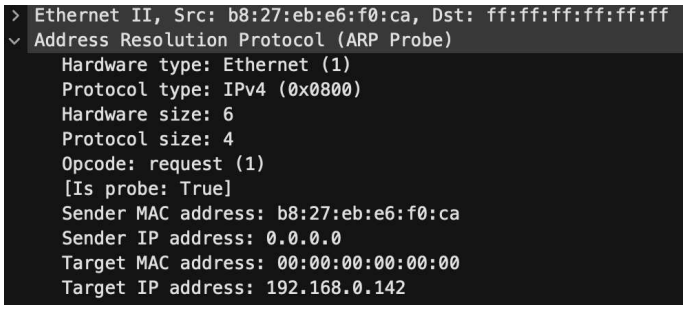
- Announce which is your MAC.
- In the ethernet header, it uses the broadcast destination
- In the ARP header
    - The Sender IP and Target IP are the **same**.
    - The target MAC is **zero**, which means all the hosts.
    - The sender MAC is what is **announced**.
- If it has an *opcode* **Request,** it is called a **Gratuitous ARP**.

```
> 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 broadcasted ARP reply.
    - Both IPs are the same, and both MACs are the same.

ARP Probe Packet

- The ARP Probe polls the network to validate that an IP address is not already in use.
- Opcode field set to 1, a Request.
- If the IP address in question is in use, we expect a Response from the owner.
- The Sender MAC address is set to the initiator's MAC address. The Sender IP address is set to 0.0.0.0.



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 commons:

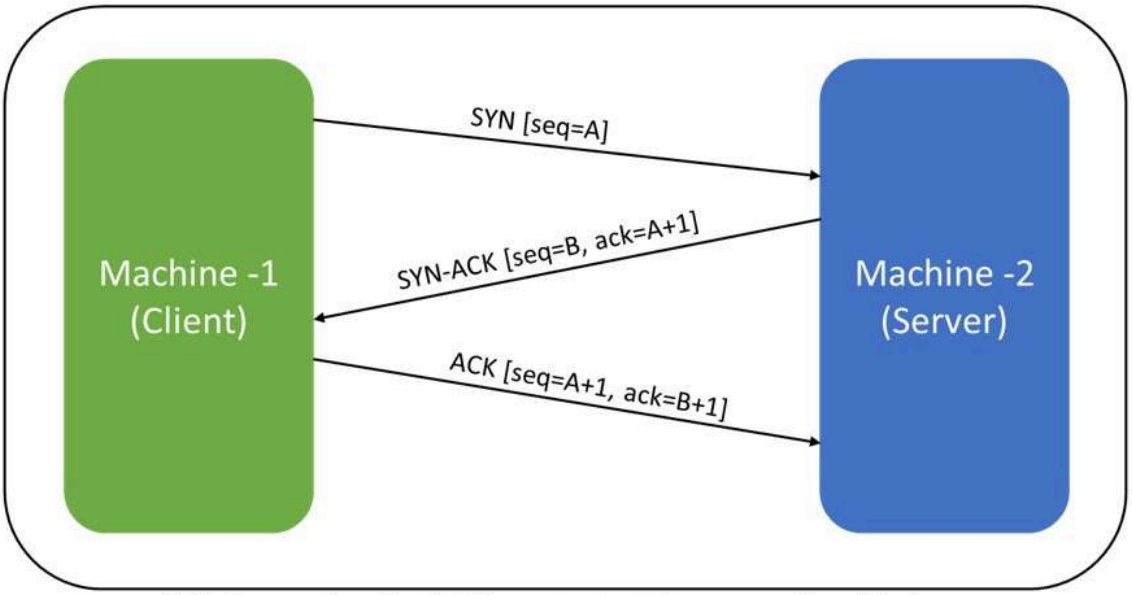
- Type 0, code 0: **Echo reply** (what ping sends)
- Type 8, code 0: **Echo request** (what ping receives)
- Type 3, various codes: **Destination unreachable**.
- Type 11, various codes: **Time exceeded in transit**.

Related Commands

- ping 123.123.123.123
  - This command shows the local cache of IP and MAC addresses in your computer.

TCP (16:40, 5m)

TCP is well-known, connection-oriented, and has error corrections, re-transmissions 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

UDP (16:45, 3m)

UDP is not connection-oriented, which means that there is no guarantee that the packets ever 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?

SCTP (16:48, 3m)

SCTP is a kind of "new" transport layer protocol (Stream Control Transmission Protocol) that combines the properties of TCP and UDP. It is message-oriented like UDP but guarantees the sequence and transmission of messages like in TCP. The key concepts are the chunks that can mix data from different applications in one message.

HTTP (16:51, 15m)

- 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[4]

- 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!

- Log in to your dockers
- Use ncat to connect to any server and port.
- Ncat is a simple tool from the nmap suite to connect to a port, send something, and receive something.
- ncat 1.1.1.1 80
    - GET / HTTP/1.1
    - Host: 1.1.1.1
    - (Press Enter twice)

HTTP/2[5]

- What is it?
- More efficient: Compression of headers
- SPDY[6] like: 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

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

- 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.

HTTP/3[7][8]

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

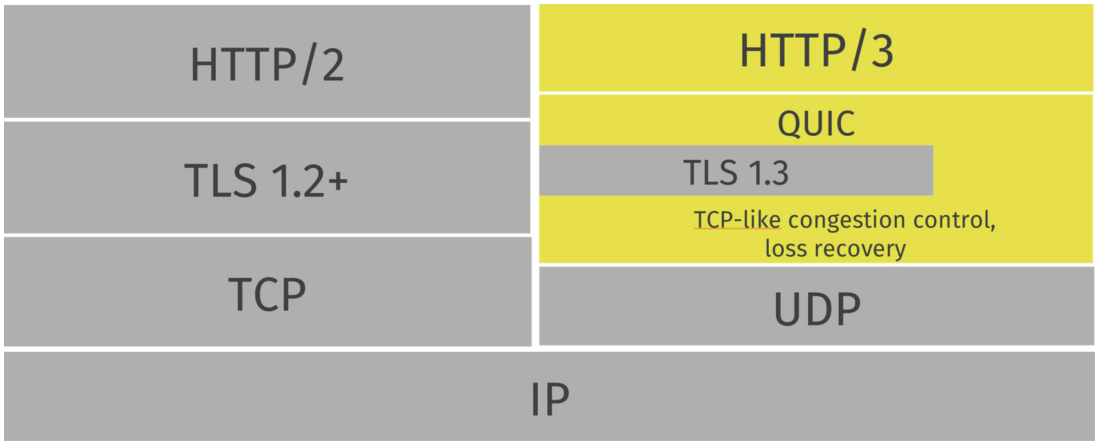


Image credit to Haxx.se[9]

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.

TLS[10][11] (17:06, 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, integrity and authentication.

- How does it work?
  - Handshake: The client and server agree on encryption methods and exchange keys.

[7] Wikipedia Contributors (2024). HTTP/3. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/HTTP/3 [Accessed 24 Sep. 2024].
[8] Haxx.se. (2023). README | HTTP/3 explained. [online] Available at: https://http3-explained.haxx.se/ [Accessed 24 Sep. 2024].
[9] Haxx.se. (2020). Protocol features | HTTP/3 explained. [online] Available at: https://http3-explained.haxx.se/en/the-protocol [Accessed 24 Sep. 2024].
[10] Xargs.org. (2017). The Illustrated TLS 1.2 Connection. [online] Available at: https://tls12.xargs.org/ [Accessed 24 Sep. 2024].
[11] Xargs.org. (2024). The Illustrated TLS 1.3 Connection. [online] Available at: https://tls13.xargs.org/ [Accessed 24 Sep. 2024].

    - 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

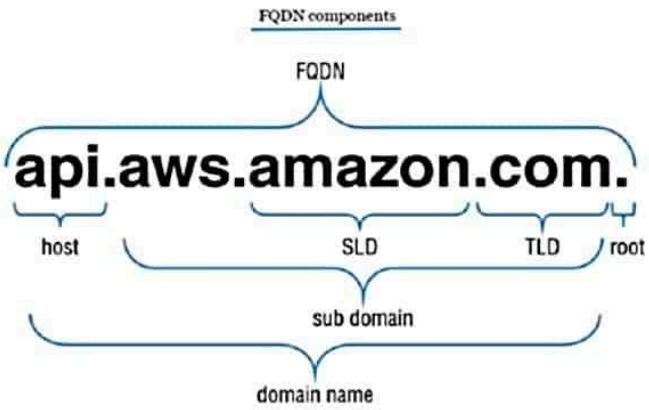
- https://tls12.xargs.org/

DNS[12] [13] (17:09, 10m)

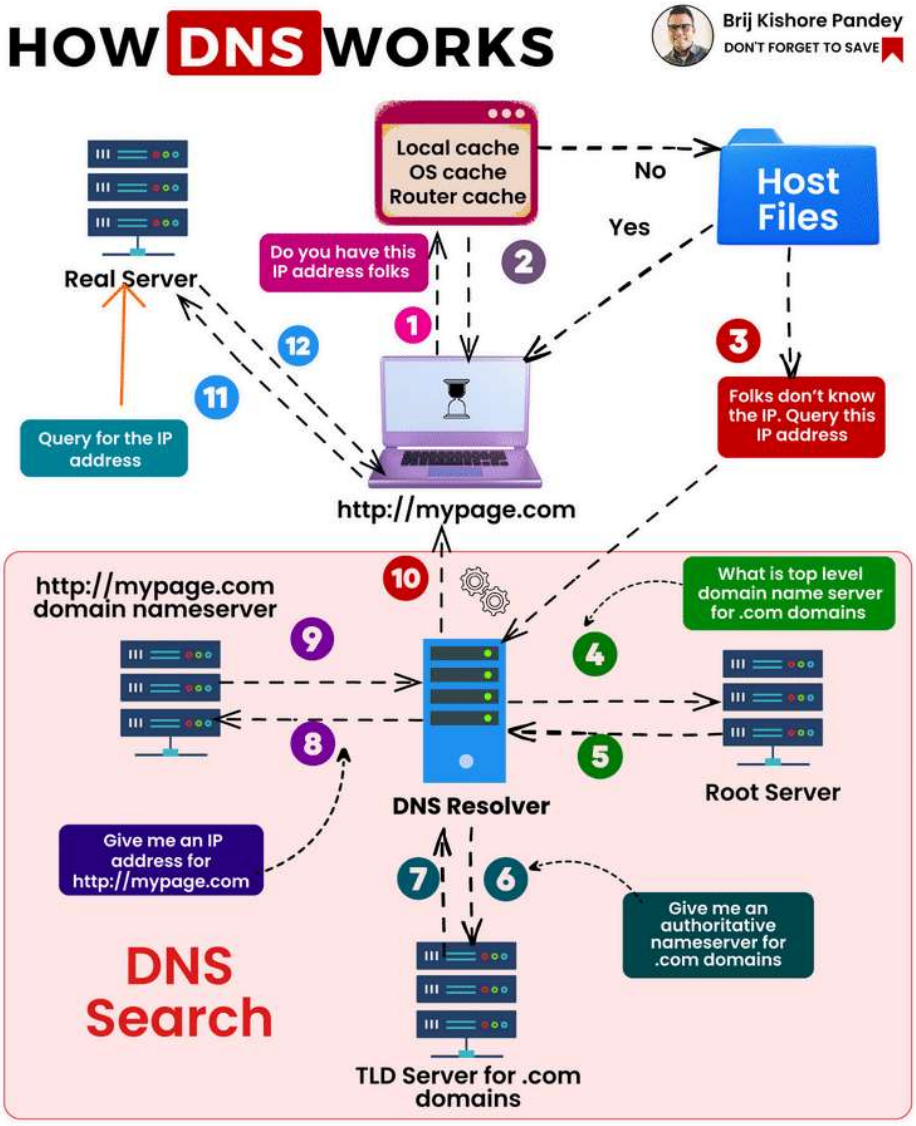
Domain Name System. Distributed hierarchical protocol.

- Domain names **and** 'host names' to IPs.
- IPs to hostnames.
- TXT, MX, A, AAAA, etc.
- What is a domain?

[12] DNSimple (2024). How DNS works. What is DNS? Learn how step by step. [online] Available at: https://howdns.works/ [Accessed 24 Sep. 2024].
[13] Wikipedia Contributors (2024). Domain Name System. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Domain_Name_System [Accessed 24 Sep. 2024].



- How is the resolution done? (Image credit to Brij Kishore Pandey[14])



- 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

[14] 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].

    - 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 (17:19, 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 BigBang.
        - 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?

NDP (17:21, 3m)

Neighbour 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.

Penetration Testing Methodology (17:24, 3m)

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

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

Guides to a Penetration test methodology (17:27, 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[15] is from 2006, and the OSSTMM[16] is from 2000.

Our simplified penetration test methodology (17:28,5m)

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 always to 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.
3. Recognize the attack surface
    - Find hosts, servers, and routers.
4. Identify the services to attack
    - Identify each port and its version
5. Find vulnerabilities
6. Plan the attack and attack
    - Consider detection measures.
7. Escalate privileges

[15] 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].
[16] Isecom.org. (2024). Open Source Security Testing Methodology Manual. [online] Available at: https://www.isecom.org/OSSTMM.3.pdf [Accessed 24 Sep. 2024].

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 (17:33,2m)

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 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 (17:35,2m)

NIST did a good job of summarizing the basics of defense[17] [18]. Including one of the best all-around guides[19].

The summary is:

- Identify your assets
  - All software. All hardware. All responsibilities.
- Protect
  - Authorization, security software, backups, encrypt, policies, trainings.
- Detect
  - Monitor all network and system logs.
  - Analyze all monitored data
  - Detection software/personnel.

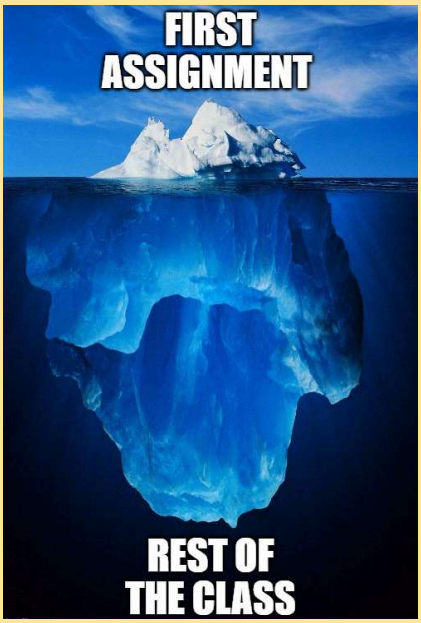
[17] NIST. (2013). Understanding the Cybersecurity Framework [online] Available at: https://www.ftc.gov/system/files/attachments/understanding-nist-cybersecurity-framework/cybersecurity_sb_nist-cyber-framework.pdf [Accessed 24 Sep. 2024].
[18] NIST. (2013). Cybersecurity Framework [online] Available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

- Respond
    - Notifications, business up, investigation, learning/updating.
- Recover
    - Repair, update, backup.

Assignment 1 (1 Point)

Assignment 1 (1 Point). Only in-person students.

1. There is a file containing a flag somewhere in your class docker.
2. You must find it and submit the flag in CTFd in the Assignment 1 challenge field.
3. The assignment opens on Thu 26.10, 21:00 CEST
4. No pioneer prize for this assignment.
5. More details in CTFd.



Class Feedback

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

https://bit.ly/BSYFEEDBACK

Side Dish 1: The Economics of Security

Why do we still have security problems? Why can't we just solve all of them, and that's it?

New security problems appear **all the time** in old and new software, new attacks, new software, new programming languages, new hardware, new integrations of software, new networks, new humans developing badly, etc. If we create new things, we will have security vulnerabilities.

Imagine you are creating a new software. There's **always** going to be some new vulnerability that affects you.

This means that 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 fix and what you 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 by the vulnerabilities and frauds online.

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

- So 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 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

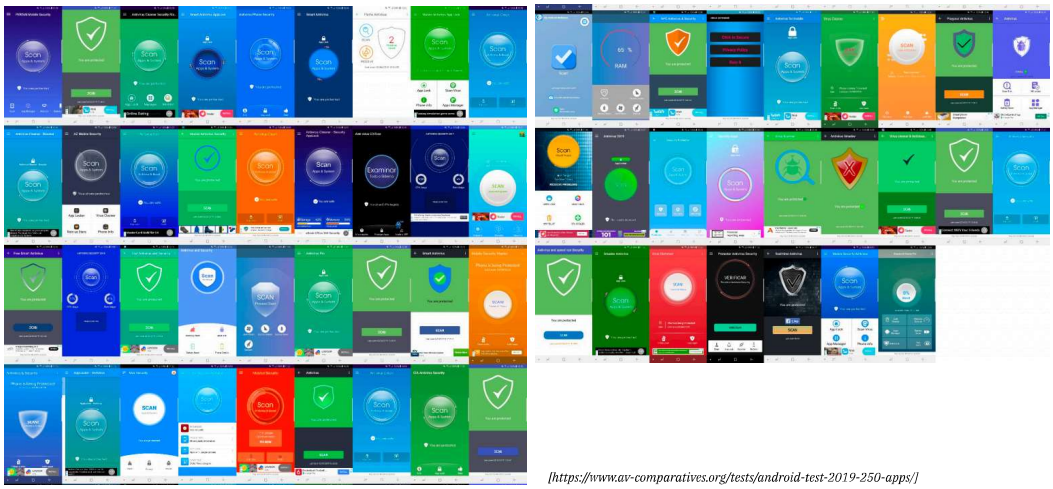
- So you want to create your next software that is super duper secure.
- But the pressure of the market will drive you to first have software that works well, that is used by people, and that has value.
- The cost of having a security vulnerability when nobody is using it is almost zero. No incentive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

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 of 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, and 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 it got a Nobel prize.



[https://www.av-comparatives.org/tests/android-test-2019-250-apps/]

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

Let's talk about your new amazing 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 that are affected that are not you[20].

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

Summary of Economics of Security

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 and understanding the defender-attacker dynamic to make more informed security decisions.

[20] 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.