# Teoria e Prática de Ataques de Segurança

**2025/2026**

**André Baptista**
andre.baptista@fc.up.pt

**Miguel Regala**
miguel.regala@fc.up.pt

https://tpas.alunos.dcc.fc.up.pt

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

# Catch-up

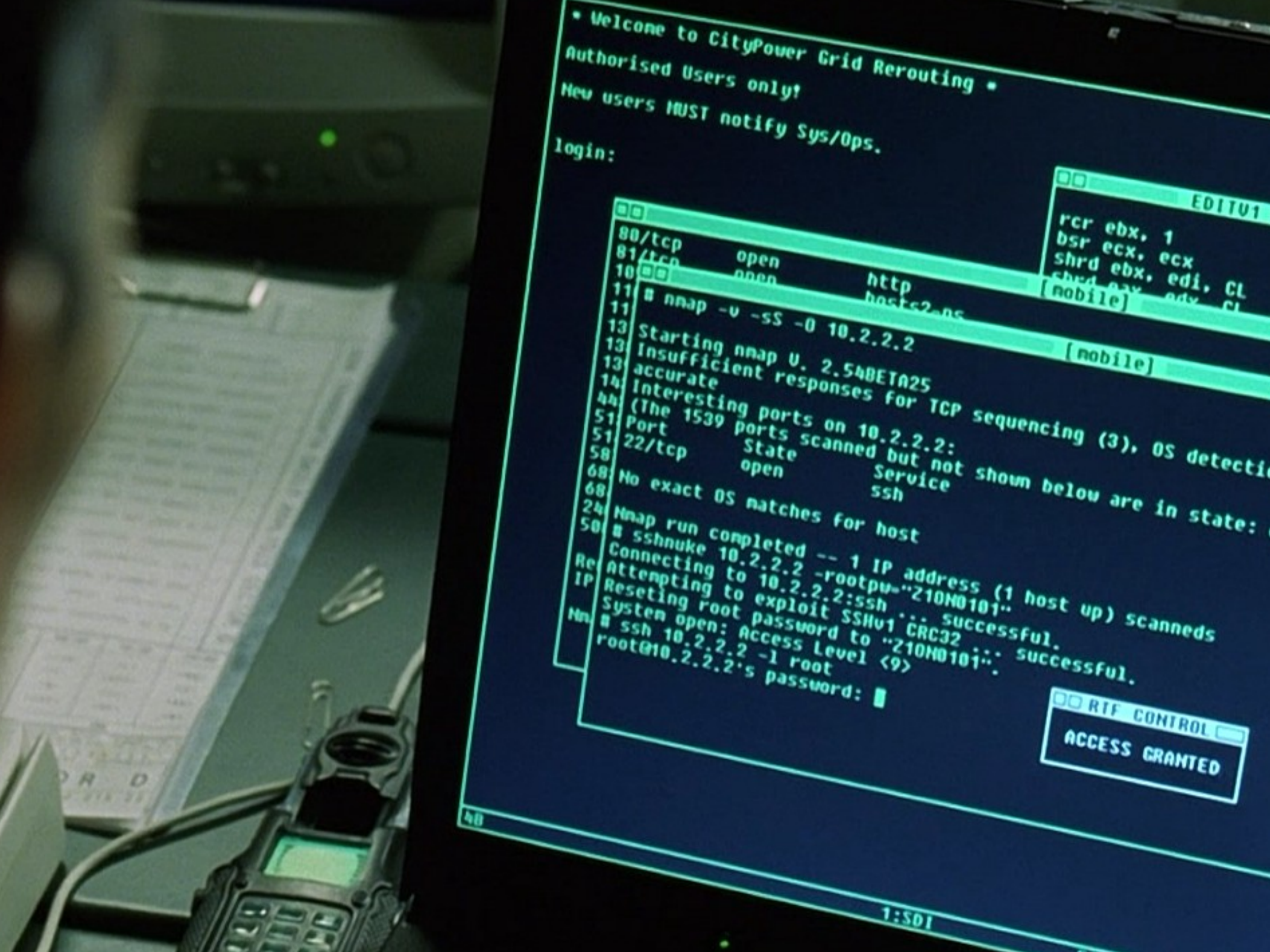
- Lab 2 deadline **today** until **23:59**
- CVEs submission deadline - **today** until **23:59**
  - https://forms.gle/DsZixXuf19hZuSgK9
    - Presentation next week**, remote**
- Pick a topic & submit 2-man group for semester project until **16/10/2025**
  - https://forms.gle/Y7D4rEfHfHzHj4fz7
  - Project = 50% final grade
  - Doubts / topic ideas or discussion -> email us

# Project: Suggested Topics

- Reverse engineering / software cracking - games, keygens, serial numbers, mobile apps
- Side channel attacks - breaking cryptographic protocols, key exfiltration attacks, covert channels
- Devices - routers, smartphones, consoles, bypassing access control mechanisms
- Tokens - *smartcards*, SIM cards, transportation tickets, RFID, QR codes, bar codes
- Malware analysis
- Bug bounties - https://hackerone.com https://bugcrowd.com https://intigriti.com https://yeswehack.com https://www.openbugbounty.org/
- Binary exploitation / 0days - Identifying and exploring vulnerabilities in software, operating systems, etc. Building exploits for CVEs without a public exploit
- Smart-contract security
- Other ideas? Suggestions are welcome. Be creative!

# Class 3

## Network scanning


* Welcome to CityPower Grid Rerouting *
Authorised Users only!
New users MUST notify Sys/Ops.
login:
EDIT01
rcr ebx, 1
bsr ecx, ecx
shrd ebx, edi, CL
[mobile]
80/tcp open http
81/tcp open
10
11
11
13
13
14
44
51
51
58
68
68
24
50
Re
IP
Nm
# nmap -v -sS -O 10.2.2.2
Starting nmap V. 2.54BETA25
Insufficient responses for TCP sequencing (3), OS detecti
accurate
Interesting ports on 10.2.2.2:
(The 1539 ports scanned but not shown below are in state:
Port State Service
22/tcp open ssh
No exact OS matches for host
Nmap run completed -- 1 IP address (1 host up) scanneds
# sshnuke 10.2.2.2 -rootpw="Z10N0101"
Connecting to 10.2.2.2:ssh ... successful.
Attempting to exploit SSHv1 CRC32 ... successful.
Reseting root password to "Z10N0101".
System open: Access Level <9>
# ssh 10.2.2.2 -l root
root@10.2.2.2's password:
RTF CONTROL
ACCESS GRANTED
48
1:SDI

# Network scanning

- Mechanisms that allow us to identify machines, ports and services within a network.

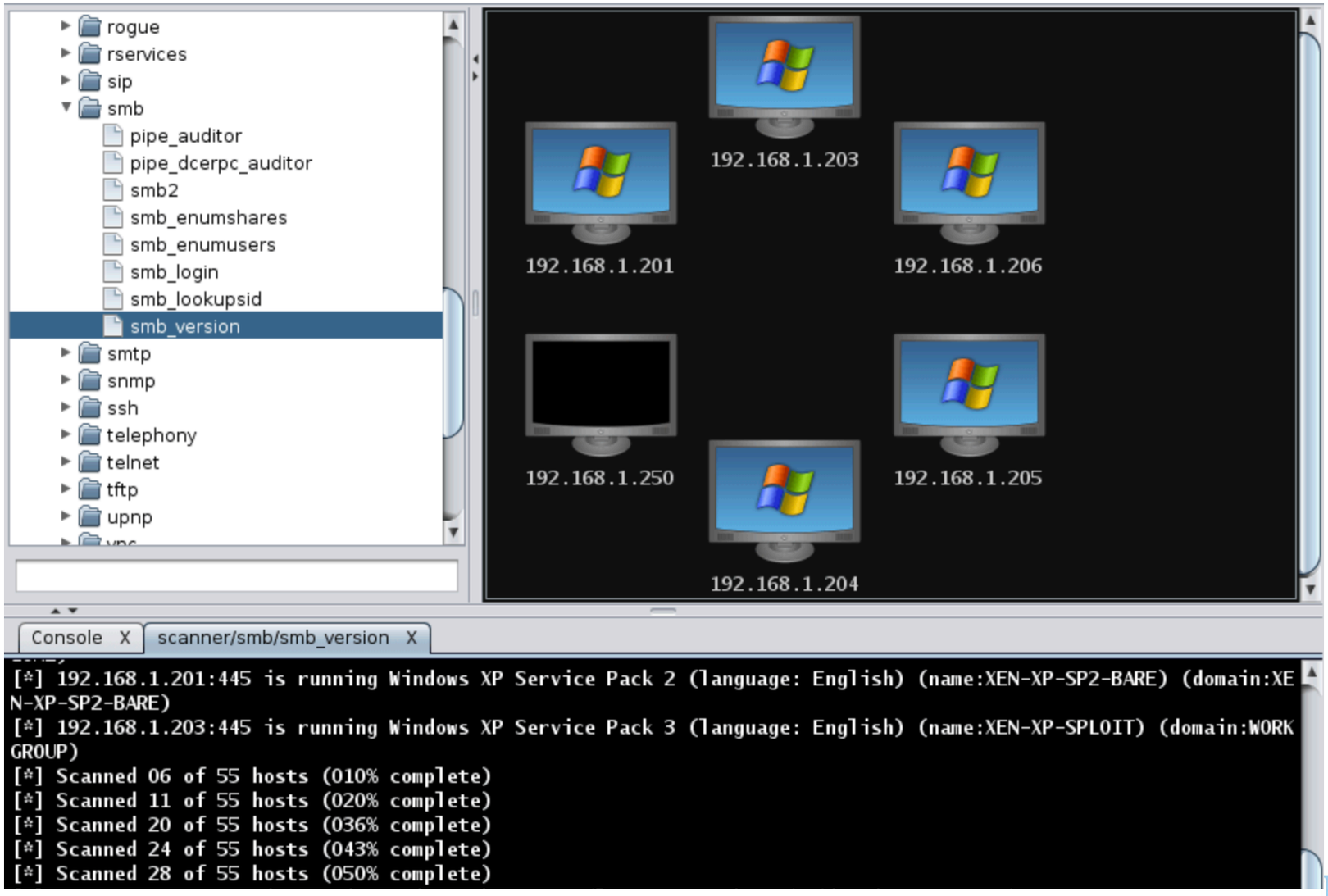
- Obtaining information from responses (or lack of responses).

- This is a very important **active recon mechanism**, available to hackers.

- Accuracy and coverage depend a lot about the quality of the information gathered in the previous recon phases.

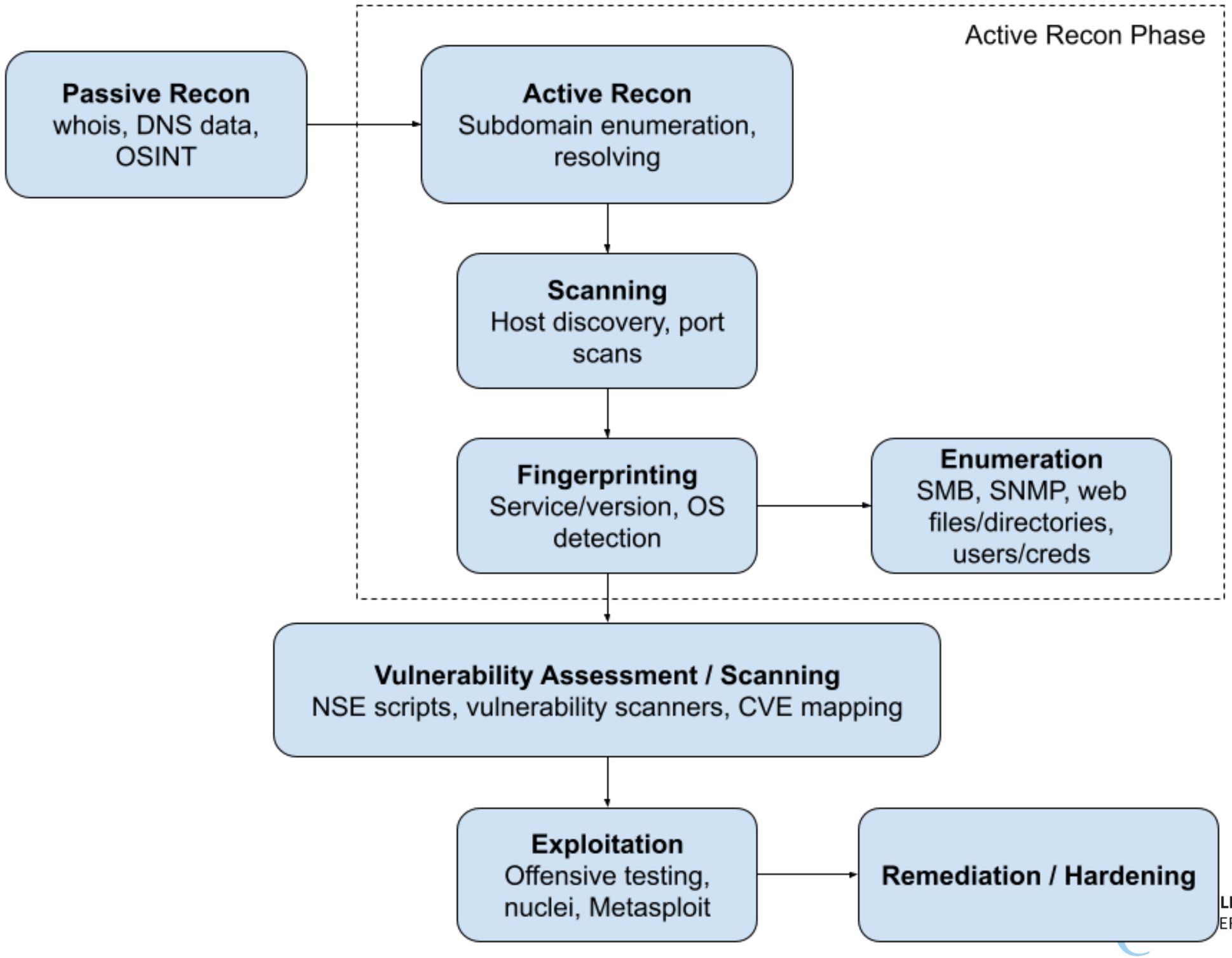
# Network scanning

**Goal:** Identifying communication channels that can be exploited to launch attacks.




From: Offensive Security

# Network scanning



# Network scanning

- Important information:
  - Machines
  - Alive hosts (IP addresses)
  - Open ports
  - Services
  - Operating systems
  - Versions and configuration errors

# Network scanning

- Scanning types:
  - Network-based
  - Port-based
  - Technology-based
  - Vulnerability-based

# Detecting alive hosts

- Internet Control Message Protocol (ICMP)
  - This is one of the protocols that can run over the IP protocol.
  - Used for management of the IP protocol itself.
  - E.g. used by routers to manage traffic, or returning error messages.

# ICMP scanning

- By using ICMP ECHO, i.e.,(implemented with the command **ping**) we can try to identify alive hosts within a network.

- Worst case scenario, we can try to understand if there's a firewall blocking ICMP requests.

- Other types: with ICMP TIMESTAMP we can receive information about the local time of the target host.

# ICMP scanning


From: https://www.squadcast.com/blog/what-is-ping-command-a-deep-dive-into-network-diagnostics

# ICMP scanning

**Table 96: ICMPv4 *Echo* and *Echo Reply* Message Format**

| Field Name | Size (bytes) | Description |
|---|---|---|
| ***Type*** | 1 | ***Type:*** Identifies the ICMP message type. For *Echo* messages the value is 8; for *Echo Reply* messages the value is 0. |
| ***Code*** | 1 | ***Code:*** Not used for *Echo* and *Echo Reply* messages; set to 0. |
| ***Checksum*** | 2 | ***Checksum:*** 16-bit checksum field for the ICMP header, as described in the topic on the ICMP common message format. |
| ***Identifier*** | 2 | ***Identifier:*** An identification field that can be used to help in matching *Echo* and *Echo Reply* messages. |
| ***Sequence Number*** | 2 | ***Sequence Number:*** A sequence number to help in matching *Echo* and *Echo Reply* messages. |
| ***Optional Data*** | Variable | ***Optional Data:*** Additional data to be sent along with the message (not specified.) |

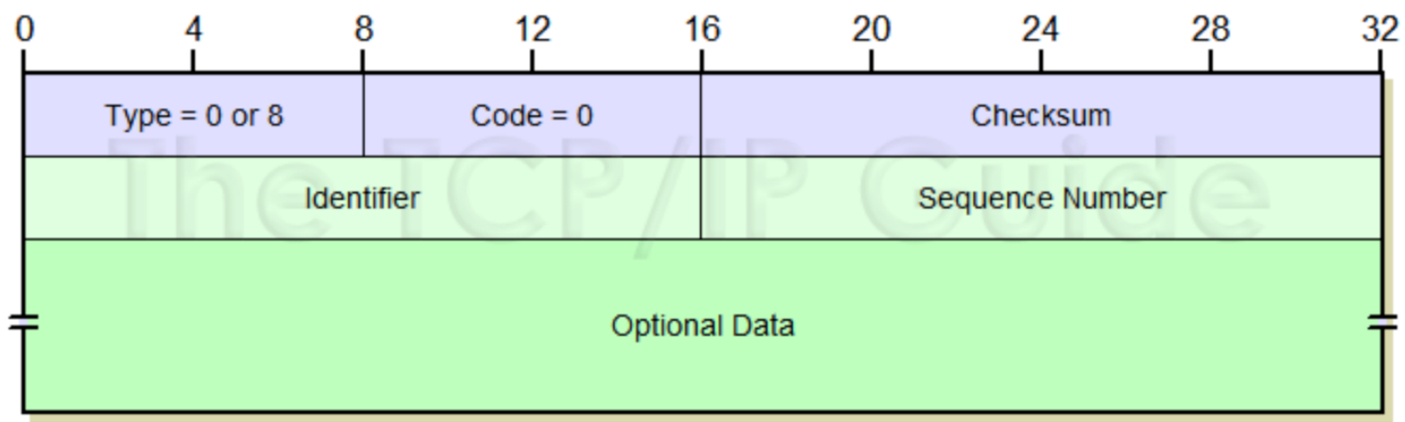



**Figure 146: ICMPv4 *Echo* and *Echo Reply* Message Format**

From: http://www.tcpipguide.com/free/t_ICMPv4EchoRequestandEchoReplyMessages-2.htm

# ICMP scanning

- With ICMP NETMASK we can try to retrieve the subnet mask used by a given network interface.

- From the subnet mask, we can identify subnets and launch a targeted scan without using broadcast addresses. Try to play with: http://www.angio.net/security/icmpquery.c

- https://securitylab.github.com/research/apple-xnu-icmp-error-CVE-2018-4407/

# nmap

- **nmap** is a standard tool for scanning with many features that we're going to play with.

- The goal is to understand these features and options, to avoid dangerous usage.

- The most basic feature is the ping scan, that allows us to use ICMP ECHO packets to understand what hosts are alive within a network range.

- Other tools

  - `masscan` (faster but sometimes not so reliable; designed for mass scanning)

  - `naabu` (implemented in go, fast & simple)

- Passive tool (shodan-based): `smap`

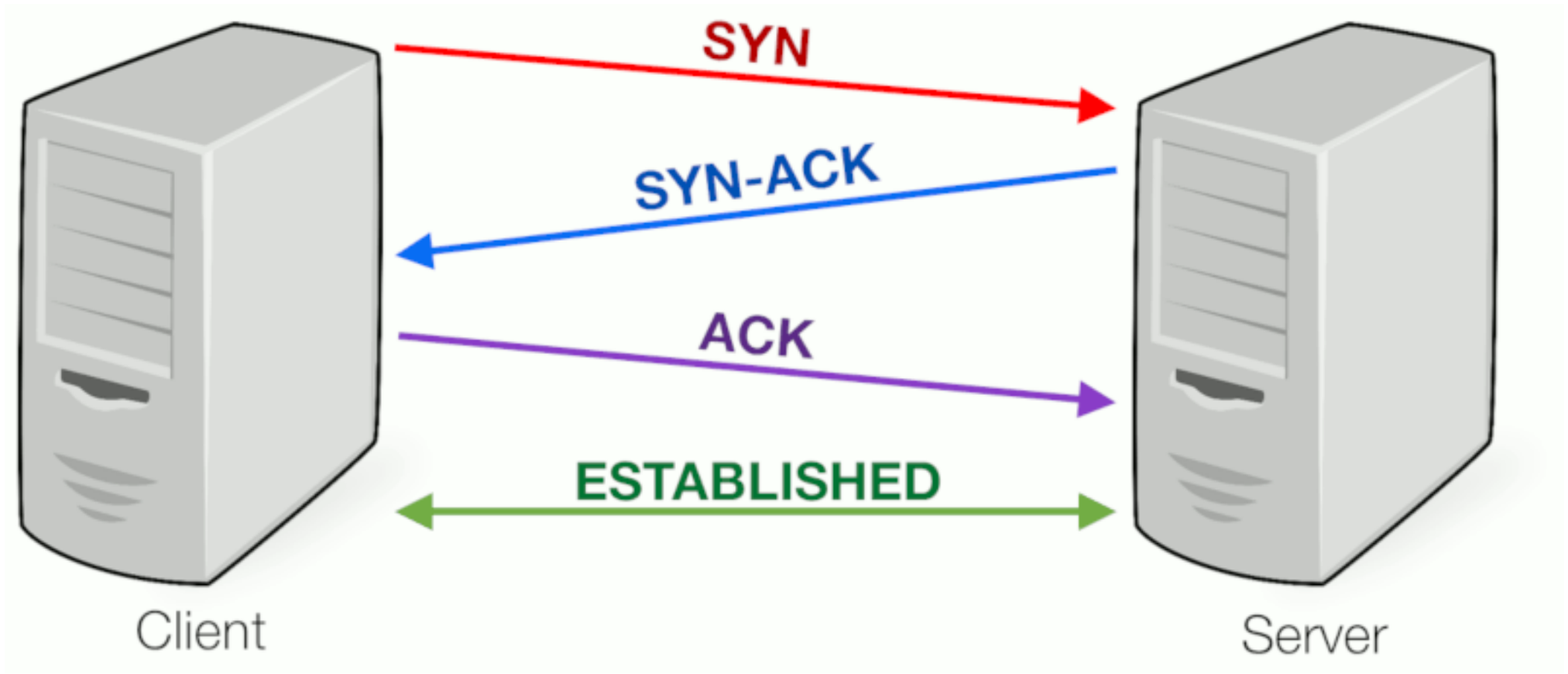
# nmap

- The best way to use it: identify subnets first, then calculate the number of hosts and network ranges (can be useful - subnet mask calculator). Perform a ping sweep after identifying the network ranges.

# Port scanning

- TCP three-step handshake:
  - Client sends **SYN, SEQ #X**
  - Server responds with **SYN+ACK, ACK #X+1, SEQ #Y**
  - Client responds with **ACK, ACK #Y+1, SEQ #X+1**

# Port scanning




From: https://www.coengoedegebure.com/tcp-3-way-handshake-port-scanning/

# Port scanning

- From there, packets use the sequence numbers used on the handshake phase…
- Until one of them sends FIN or RST.
- Port scanning is all about sending packets with multiple flag combinations and analysing responses.
- Very important to scan standard ports and discover services.

# nmap

- We can perform port scanning and obtain the following results:
  - **Open** - reachable and accepting connections
  - **Closed** - reachable but not accepting connections
  - **Unfiltered** - reachable but we can’t tell if it’s open
  - **Filtered** - we can’t tell if the port is open or not (e.g. - firewall)
  - **Open I Filtered** - absence of response, we can’t tell if the port is open or filtered
  - **Closed I Filtered** - we can’t tell if the port is closed or is being filtered

# nmap

| Nmap State | Meaning | Port Reachability | Service Running |
|---|---|---|---|
| **open** | An application is actively accepting TCP/UDP connections on this port. | ✅ Yes | ✅ Yes |
| **closed** | No application is listening, but the port responds to probes. | ✅ Yes | ❌ No |
| **filtered** | Nmap cannot determine if the port is open because a firewall or filter blocks probes. | ❓ Unknown | ❓ Unknown |
| **unfiltered** | The port is reachable, but Nmap cannot determine whether it is open or closed. | ✅ Yes | ❓ Unknown |
| **open_filtered** | Nmap cannot distinguish whether the port is open or filtered; the probe got no definitive response. | ❓ Unknown | ❓ Unknown |
| **closed_filtered** | Nmap cannot distinguish whether the port is closed or filtered; it may be closed or blocked. | ❓ Unknown | ❓ Unknown |

# TCP scanning

- When we're using a machine without root permissions, we may have to use the TCP stack without modifications

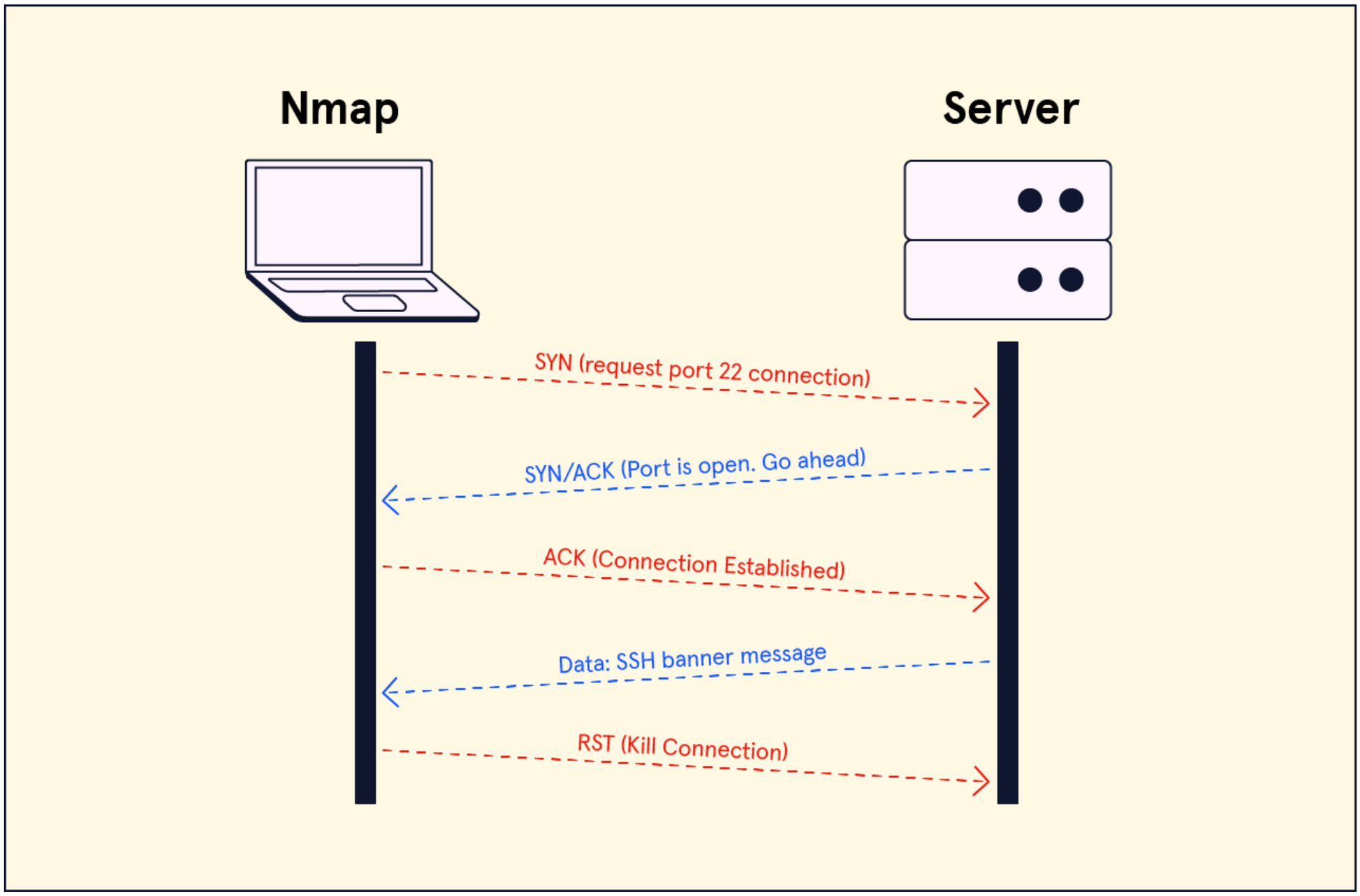
- Also known as **connect scan** (`nmap -sT`)

- A TCP scan is about trying to establish complete connections (full handshakes) with the ports/services

- *Default* method if SYN is not available

- A basic IDS solution will detect this scan, but more logs are saved

# TCP scanning

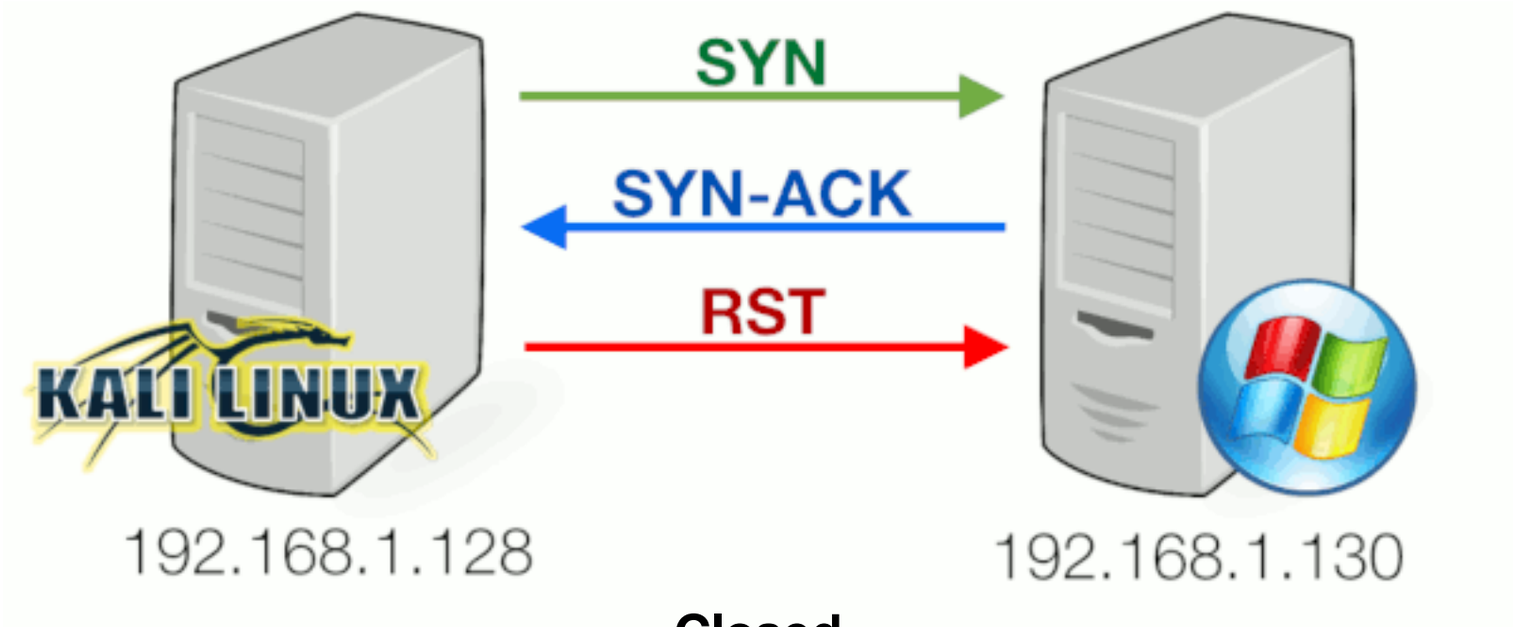



From: https://www.codecademy.com/resources/docs/cybersecurity/nmap/tcp-connect-scan

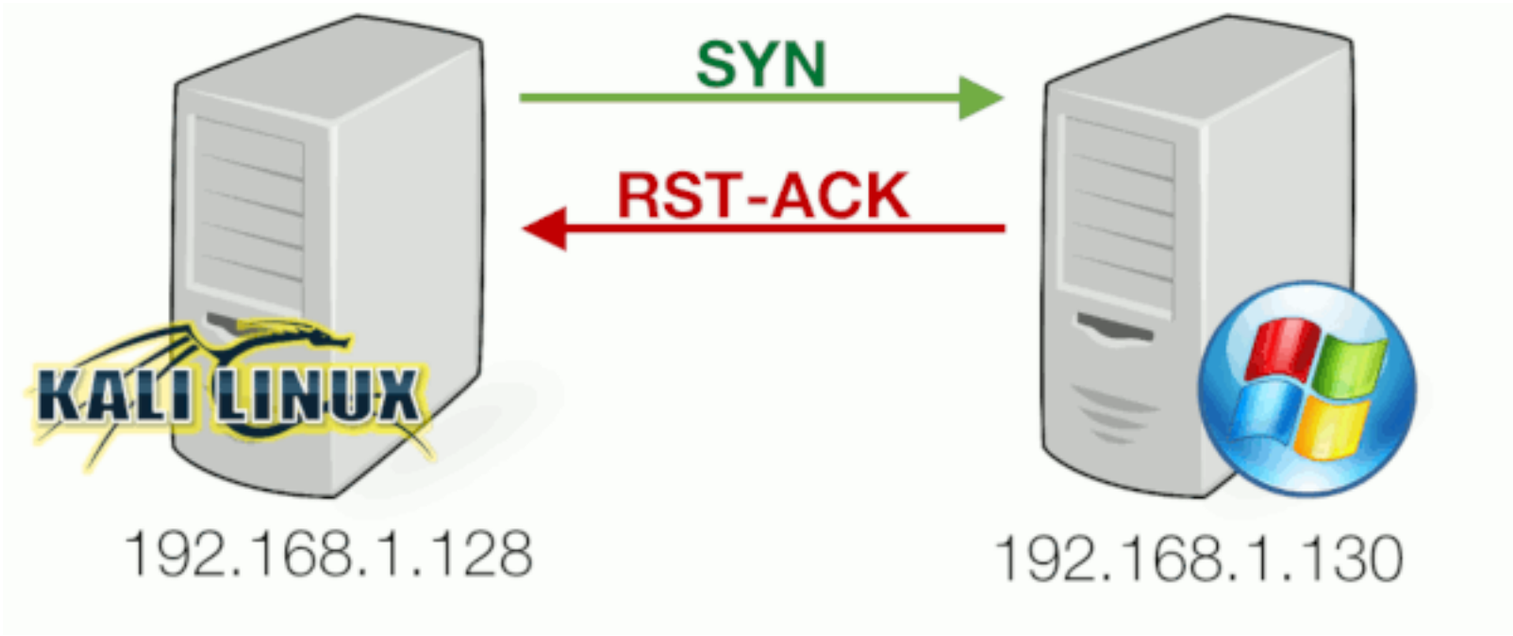
# SYN scanning (or half open)

- A simple and quick method (**`nmap -sS`**)
  - We send a SYN and analyse the response
    - If we obtain a SYN/ACK, the port is **open**
    - RST means **closed**
    - No response means that the port is probably **filtered**
    - Works for all network stacks (doesn't depend on versions)
    - The handshakes are never completed (*half-open* scanning, low-key)
    - Basic IDS solutions usually detect this activity, but we don't have so many logs

**Open**




**Closed**




From: https://www.coengoedegebure.com/tcp-3-way-handshake-port-scanning/

# ACK scan

- To understand if there are firewalls on the network we can try to perform this scan (**`nmap -sA`**)
  - A ACK message is sent (out of order)
  - Some firewalls will let the packet go through, because it seems that a transaction or handshake is being concluded
  - This scan never identifies ports **open** or **openlfiltered**
  - A port that comes up as **unfiltered** means that we can't tell if the port is open or closed, but it's reachable for connections
  - Lack of response means the port is **filtered**

# Other variants

- TCP standards include some flags that can identify closed ports (RST is return in these scenarios)
  - XMAS scan - All TCP flags are sent as active
  - FIN scan - Only the FIN flag is sent as active
  - NULL scan - No flags are active
- Lack of response means **openlfiltered**
- More low-key, but these don't always work (basic IDS is able to detect)
- The nmap option `--scanflags` allows even more options

# UDP scanning

- Many services are UDP based, such as (SNMP, DHCP, DNS, etc).

- Scans are difficult, since we don't have an handshake.

- By sending an UDP packet to a specific port, receiving an ICMP error (unreachable) means that the port is closed.

- Lack of responses is usually interpreted as **open** or **filtered**.

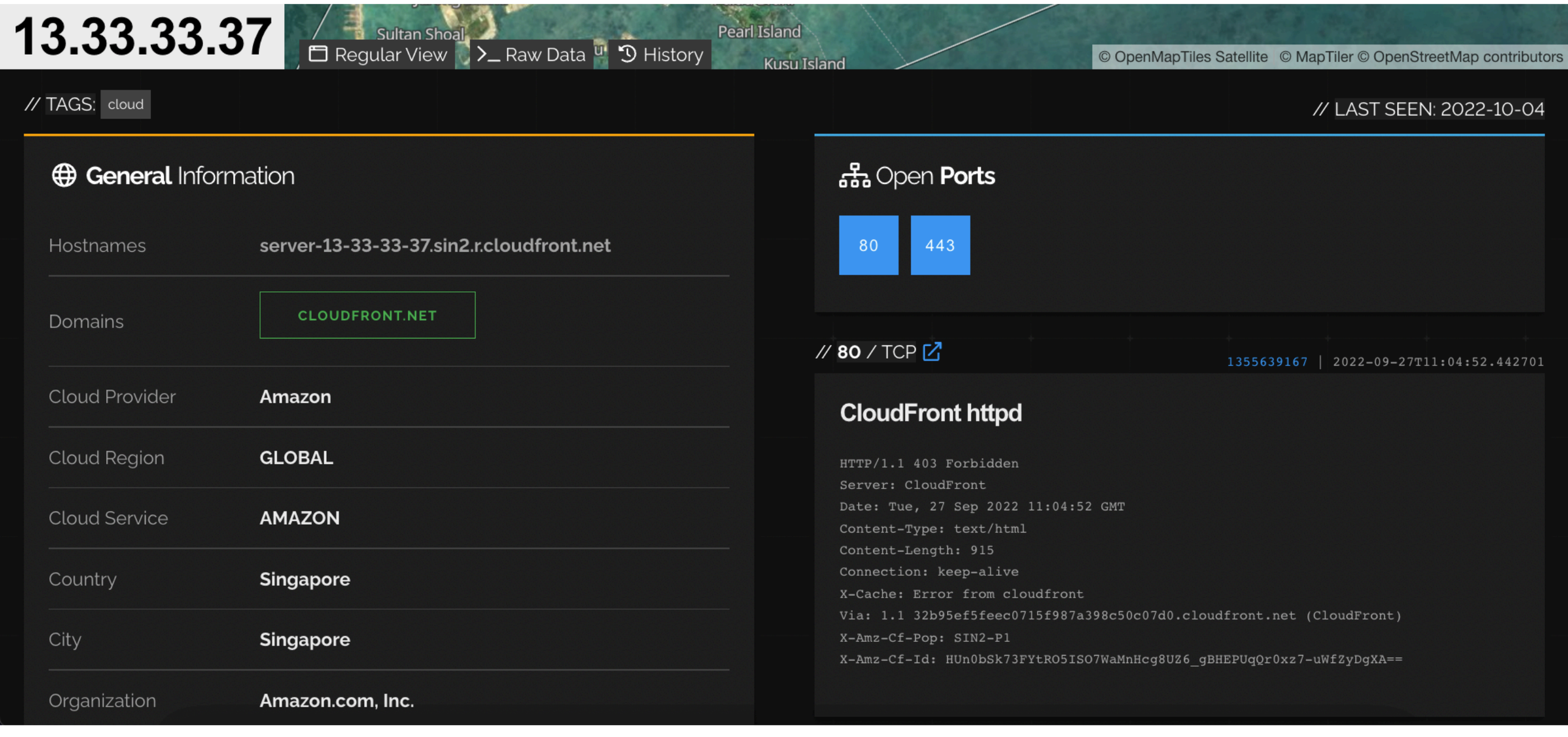
# Notice

⚠️

- Scanning is an active technique that can result in legal problems. No active scan should ever be performed against any real target, without explicit consent.

- However, you can scan your own local/virtual networks to play with scanning tools (e.g. home network).

# Shodan




https://www.shodan.io

# Countermeasures

- Properly configuring firewalls and IDS solutions to detect and block port scanning effectively
- Rate-limiting
- Use host-based firewalls, such as UFW, iptables, Windows firewall, macOS firewall
- Create custom filtering rules for your systems (e.g. block ICMP messages)
- Try all scanning types to test these protections

# Countermeasures

- ufw
  - `$ sudo ufw enable`
  - `$ sudo ufw default deny incoming`
  - `$ sudo ufw default allow outgoing`
  - `$ sudo ufw allow ssh / $ sudo ufw allow 22/tcp`

# Evasion

- An efficient technique for evasion is the usage of packet fragmentation, so that firewall and IDS solutions don't parse the content properly (**`nmap -f`**)
- Spoofing IP addresses (decoy addresses > **`nmap -D`**)
- Spoofing ports for the outgoing traffic (**`nmap -g / –source-port`**)
- If idle scanning or similar techniques still work against a given host, it's very effective for evasion purposes.

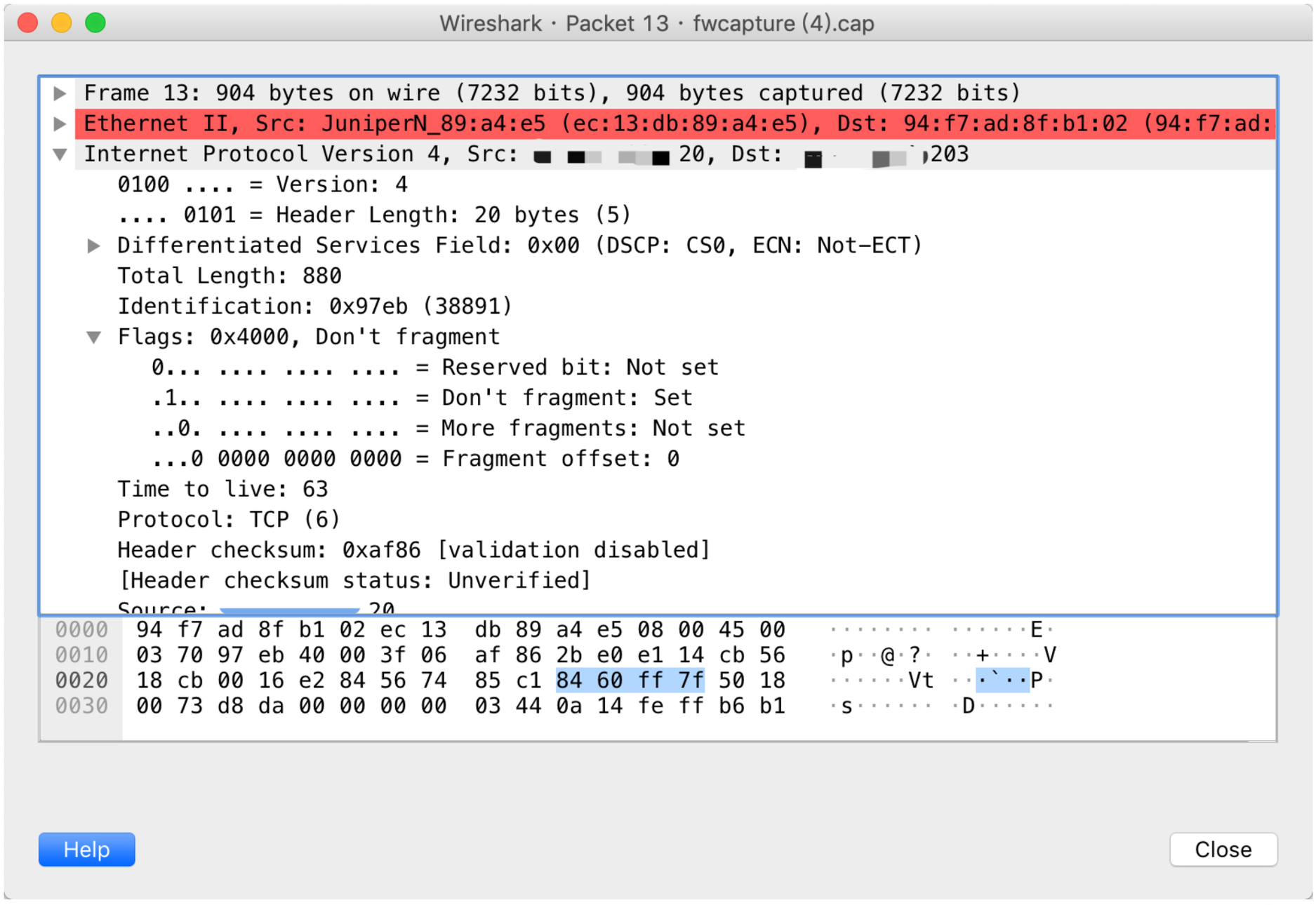
# Banner grabbing

- Technique to understand the OS and service versions of the target host:
  - **Passive**: sniffing packets in the network, parsing error messages
  - **Active**: sending specific packets
- Information about the platform (OS) is very important to search for potential exploits.

# Passive banner grabbing

- With passive techniques, we observe the properties and behaviour of packets coming from/to a target in the network:
  - TTL values
  - window size values
  - don’t fragment bit (DF) is active?
  - type of service (ToS) field is active?

# Passive banner grabbing



# Example

- **TTL:** 45
- **Window Size:** 0x7D78 (32120 in decimal)
- **DF:** Active
- **TOS:** 0x0

**Signature:** Linux kernel 2.2.x

**Tool:** https://lcamtuf.coredump.cx/p0f3/

# Active banner grabbing

- With active techniques, we send packets to target ports with many options/flags we already discussed.

- Specific payloads can be used, depending on the service and port to disclose versions, OS, etc.

- We can also try to identify patterns on sequence numbers and other heuristics.

# Example

**netcat**, **telnet** and other tools (TCP/UDP socket based) allow us to retrieve banners with service information:

```
➜  ~ nc -v ssh.alunos.dcc.fc.up.pt 22
Connection to ssh.alunos.dcc.fc.up.pt port 22 [tcp/ssh] succeeded!
SSH-2.0-OpenSSH_8.3 ←

Invalid SSH identification string.
```

Other systems may disclose the OS:

**`SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2`**

# Countermeasures

- Deactivate standard messages
- Put up fake information (for more information see: honeypots)
- Deploy firewalls
- In web-servers, try to change default configurations (apache, nginx) and try to change endpoint extensions that identify technologies
- This only makes attacker's life harder. A persistent hacker can still manage to breach systems.