CYBERSECURITY LAB

Alessandro Renda

Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste

AITM – LAB 2

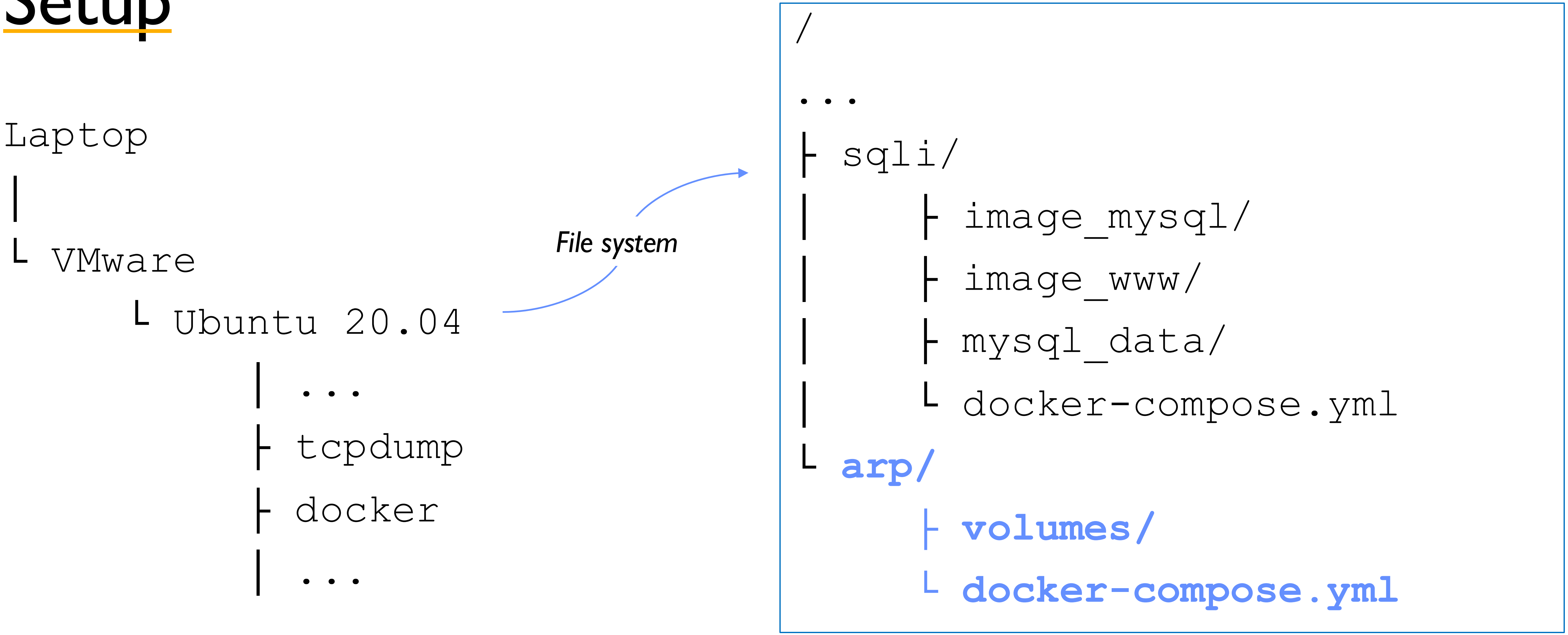
Academic Year 2025/2026

Exam
Folder name for material submission **06_AITM_2**

AitM in SeedLabs

- The SeedLabs project includes a lab on ARP Cache Spoofing
- Follow the Lab Guide and solve the tasks described in the guide

Setup



```
Laptop
|
└ VMware
     └ Ubuntu 20.04
          | ...
          ├ tcpdump
          ├ docker
          | ...
```

```
/

...
├ sqli/
|     ├ image_mysql/
|     ├ image_www/
|     ├ mysql_data/
|     └ docker-compose.yml
└ arp/
      ├ volumes/
      └ docker-compose.yml
```

- Download `Labsetup.zip` file in a dedicated folder into your VM and unzip it
- Use `docker-compose.yml` to setup the environment

Environment

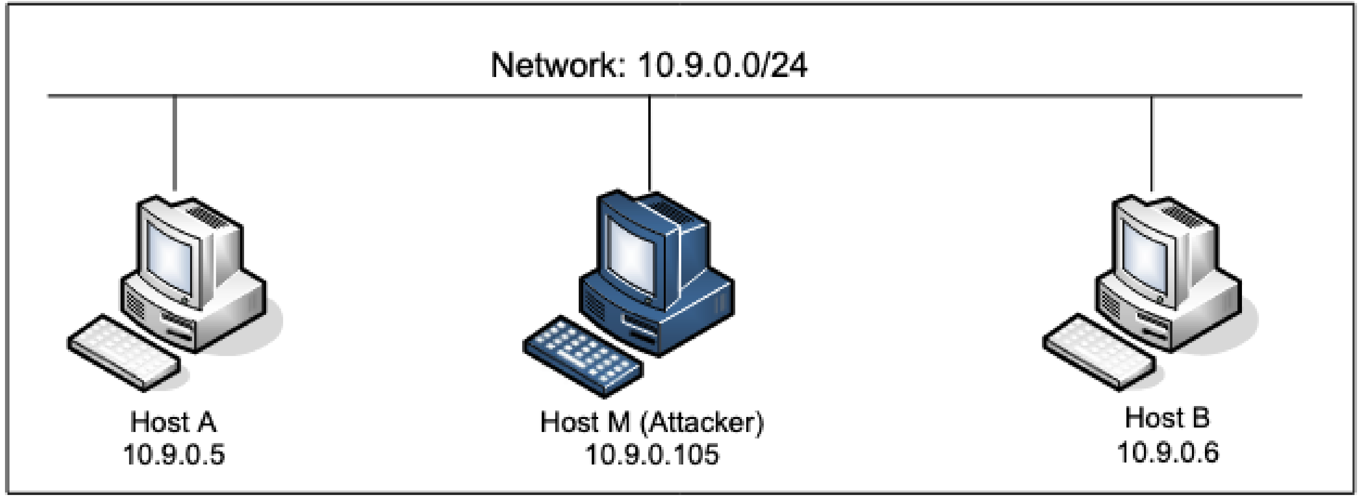


Figure 1: Lab environment setup

- Positioning as AitM through ARP cache spoofing
- Acting as AitM: `telnet` and `netcat` traffic manipulation

Telnet

- Client-server application protocol over TCP (typically server listens on port 23)
- Main, historical, use: access to a **command-line interface on a remote host**
- Security flaws:
  - Information is transmitted in plain-text
  - It has **largely been replaced by SSH**

- "*Raw character mode with server echo*" in our SeedLab
  - Each character typed by the client is sent immediately to the server
  - The server echoes the same character back
  - The echoed character appears on the client’s terminal

netcat

- Networking version of `cat` utility for reading from and writing to network connections using TCP or UDP

- On host B (10.9.0.6) listen on a given port: `nc -l -p 12345`
- On host A (10.9.0.5) connect to host B : `nc 10.9.0.6 12345`

- In our seedLab
  - Each byte typed by the client is sent immediately to the server
  - The server receives the data and display it

Exam: report required

- You are expected to produce a report for this lab

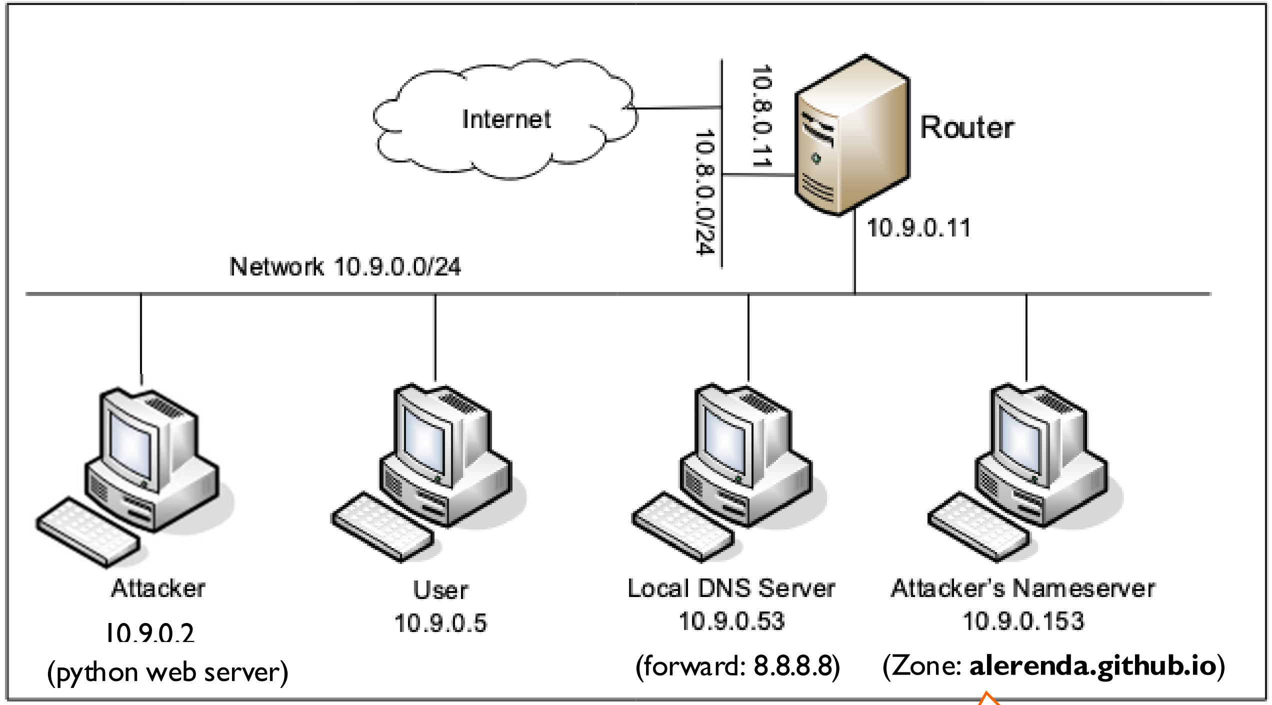
- The report should describe and discuss the resolution of the tasks and any other relevant observation you see fit

Nice looking under the hood, but …

- For practical AitM scenarios, from an **application perspective**, `netcat` and `telnet` obviously play a minor role
- Can you think to a more interesting / relevant application?

Proposal

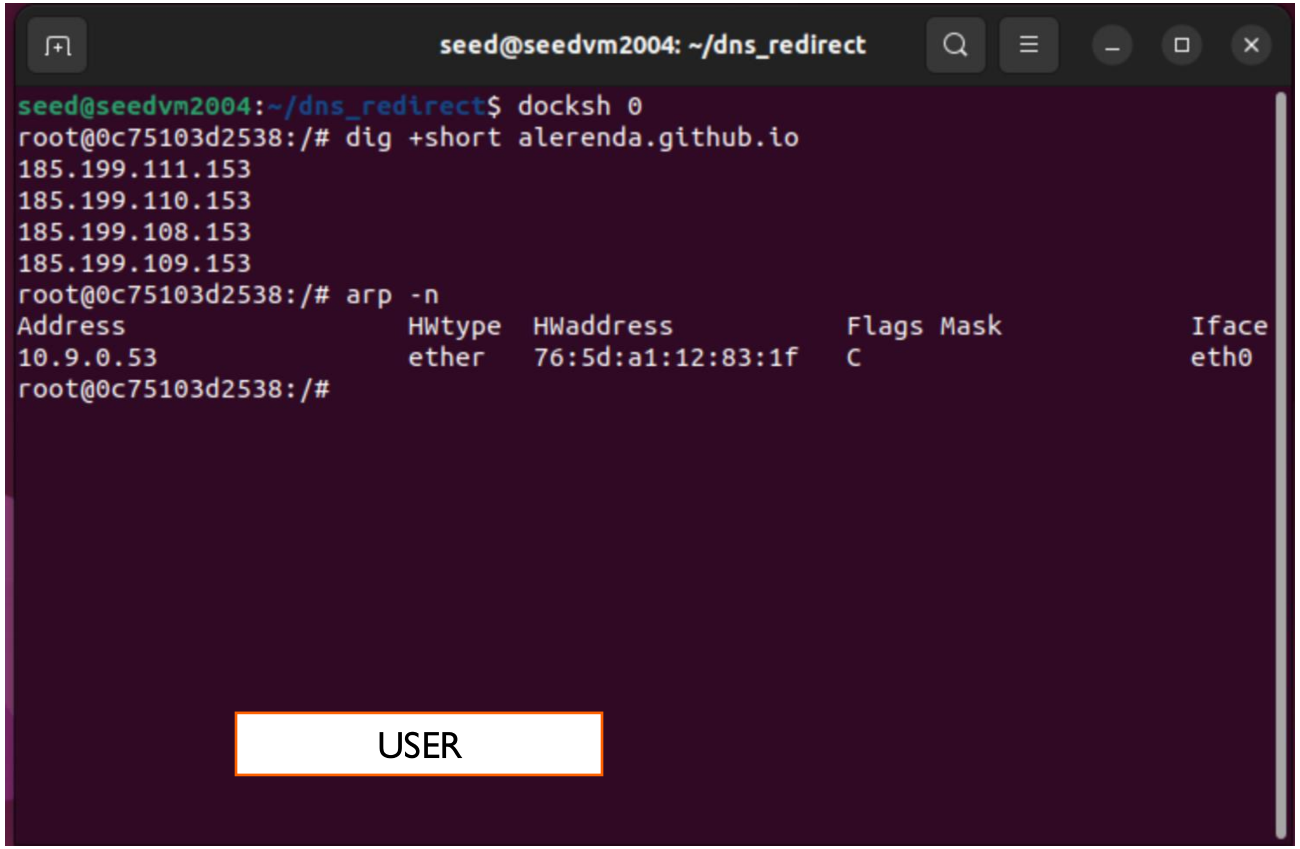
- Start from the setup discussed in this LAB
  - https://seedsecuritylabs.org/Labs_20.04/Files/DNS_Local/Labsetup-arm.zip
- *Slightly* modified as follows:



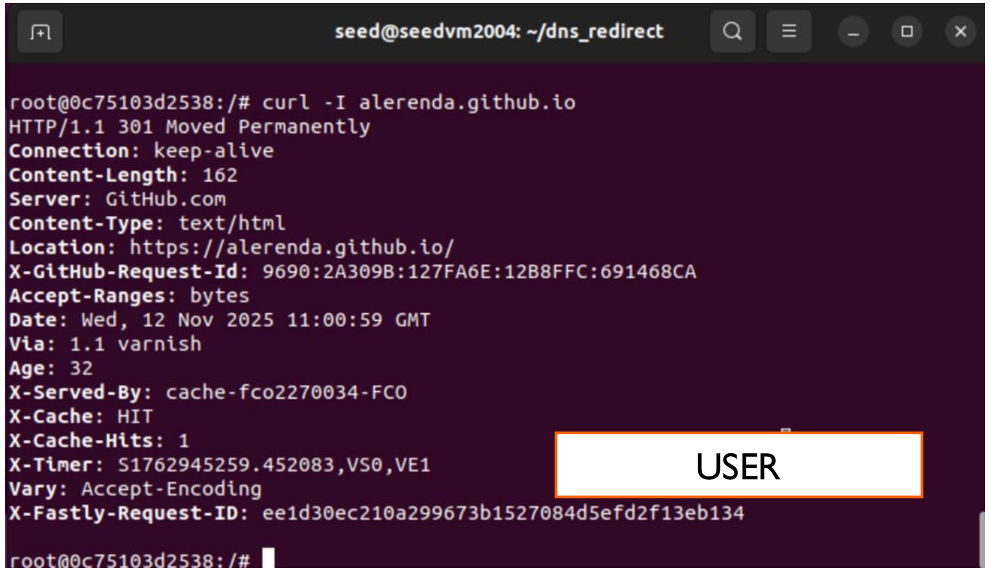
```
IN A 10.9.0.2
```

Proposal – Initial setting

- **User (10.9.0.5)**
  - DNS query for `alerenda.github.io`
  - Received reply from Local NS
  - ARP cache with MAC(Local NS)
  - cURL to `alerenda.github.io`



```
seed@seedvm2004:~/dns_redirect$ docksh 0
root@0c75103d2538:/# dig +short alerenda.github.io
185.199.111.153
185.199.110.153
185.199.108.153
185.199.109.153
root@0c75103d2538:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.53                ether   76:5d:a1:12:83:1f   C                     eth0
root@0c75103d2538:/#
```



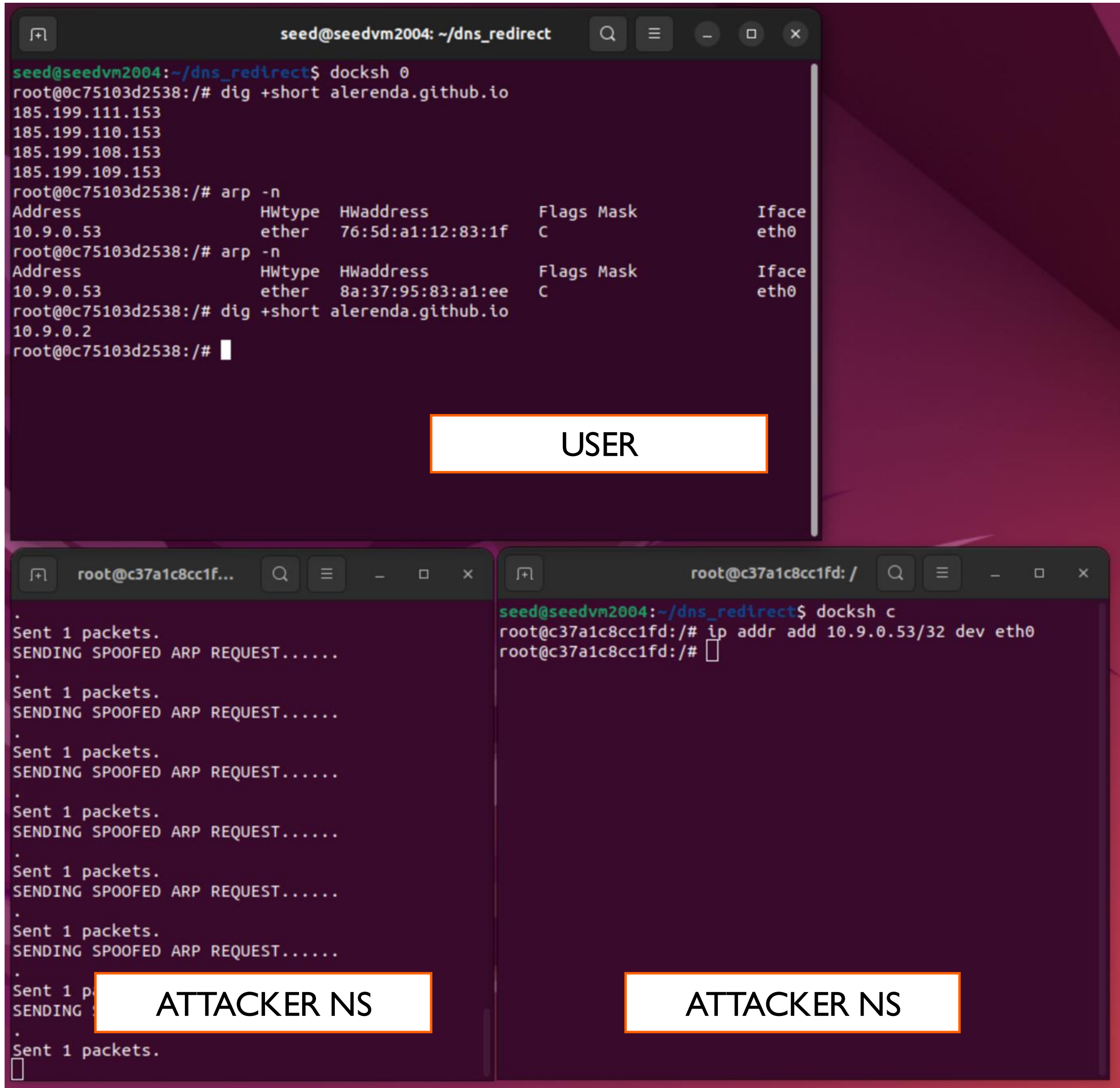
```
root@0c75103d2538:/# curl -I alerenda.github.io
HTTP/1.1 301 Moved Permanently
Connection: keep-alive
Content-Length: 162
Server: GitHub.com
Content-Type: text/html
Location: https://alerenda.github.io/
X-GitHub-Request-Id: 9690:2A309B:127FA6E:12B8FFC:691468CA
Accept-Ranges: bytes
Date: Wed, 12 Nov 2025 11:00:59 GMT
Via: 1.1 varnish
Age: 32
X-Served-By: cache-fco2270034-FCO
X-Cache: HIT
X-Cache-Hits: 1
X-Timer: S1762945259.452083,VS0,VE1
Vary: Accept-Encoding
X-Fastly-Request-ID: ee1d30ec210a299673b1527084d5efd2f13eb134

root@0c75103d2538:/#
```

Proposal – Attack (1)

- **Attacker NS (10.9.0.153)**
  - Launch ARP spoofing periodically
  - Convince OS to treat 10.9.0.53 as own IP

- **User (10.9.0.5)**
  - ARP cache with **spoofed** MAC
  - DNS query for `alerenda.github.io`
  - Received reply **from Attacker NS**



```
seed@seedvm2004:~/dns_redirect$ docksh 0
root@0c75103d2538:/# dig +short alerenda.github.io
185.199.111.153
185.199.110.153
185.199.108.153
185.199.109.153
root@0c75103d2538:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.53                ether   76:5d:a1:12:83:1f   C                     eth0
root@0c75103d2538:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.53                ether   8a:37:95:83:a1:ee   C                     eth0
root@0c75103d2538:/# dig +short alerenda.github.io
10.9.0.2
root@0c75103d2538:/# 
```

USER

```
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 p
SENDING 
.
Sent 1 packets.
```

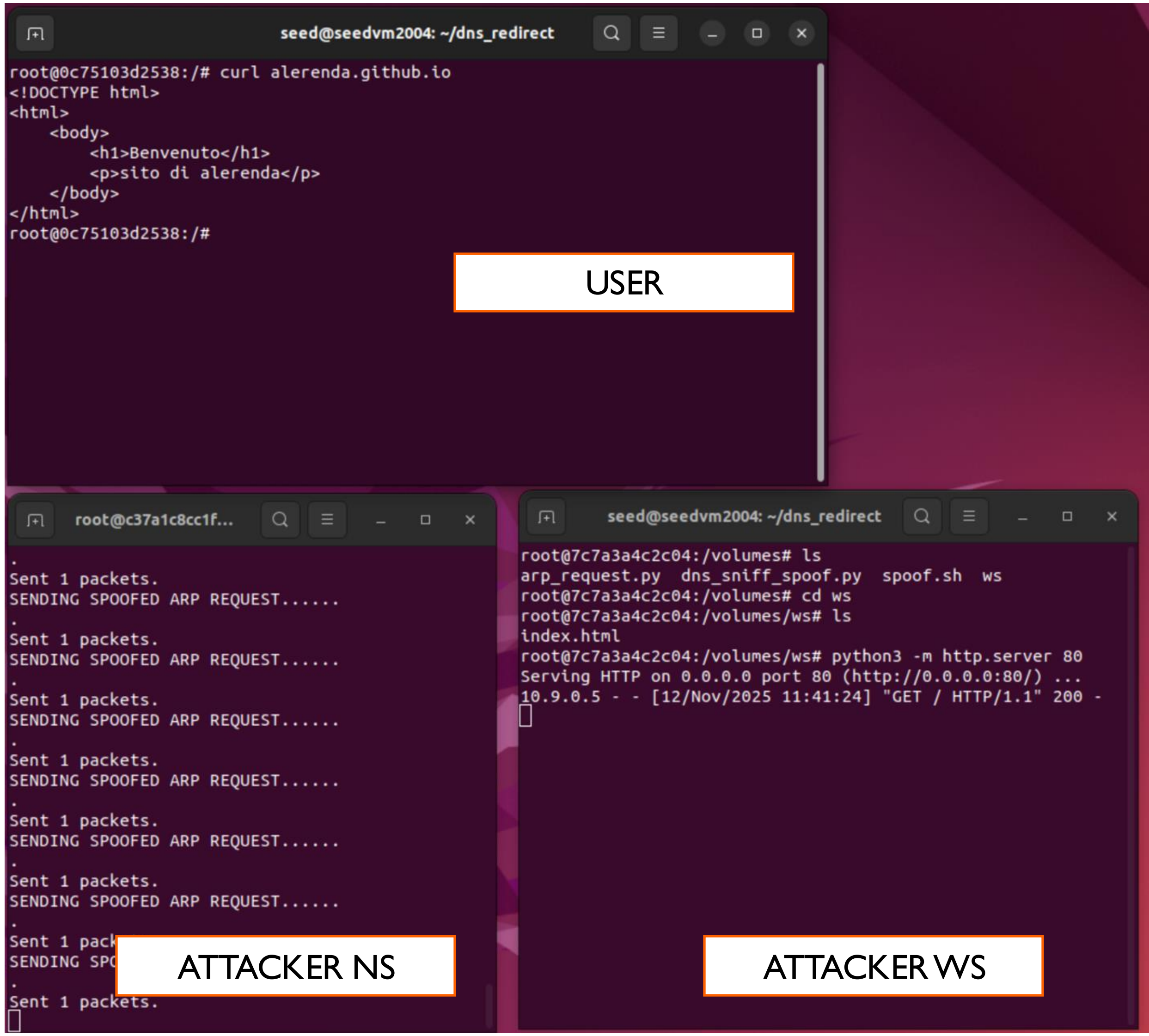
ATTACKER NS

```
seed@seedvm2004:~/dns_redirect$ docksh c
root@c37a1c8cc1fd:/# ip addr add 10.9.0.53/32 dev eth0
root@c37a1c8cc1fd:/# 
```

ATTACKER NS

Proposal – Attack (2)

- **Attacker NS (10.9.0.53)**
  - Launch ARP spoofing periodically
  - Convince OS to treat 10.9.0.53 as own IP
- **Attacker WS (10.9.0.2)**
  - Start HTTP server
- **User (10.9.0.5)**
  - cURL to `alerenda.github.io`


seed@seedvm2004: ~/dns_redirect
root@c37a1c8cc1f...
seed@seedvm2004: ~/dns_redirect

```
root@0c75103d2538:/# curl alerenda.github.io
<!DOCTYPE html>
<html>
    <body>
        <h1>Benvenuto</h1>
        <p>sito di alerenda</p>
    </body>
</html>
root@0c75103d2538:/#
```

USER

```
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 packets.
SENDING SPOOFED ARP REQUEST......
.
Sent 1 pack
SENDING SPO
.
Sent 1 packets.
```

ATTACKER NS

```
root@7c7a3a4c2c04:/volumes# ls
arp_request.py  dns_sniff_spoof.py  spoof.sh  ws
root@7c7a3a4c2c04:/volumes# cd ws
root@7c7a3a4c2c04:/volumes/ws# ls
index.html
root@7c7a3a4c2c04:/volumes/ws# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.9.0.5 - - [12/Nov/2025 11:41:24] "GET / HTTP/1.1" 200 -
```

ATTACKER WS

Exam: extra report (not mandatory)

- You may want to produce **at most one** additional report for the AitM part
- The additional report is not mandatory
- The additional report will be considered in the evaluation

- Suggested topics
  - Replicate the *proposal*
  - Explore existing AitM tools, e.g.
    - evilginx
    - mitmproxy
    - ettercap

Exam
Folder name for material submission **06_AITM_3**