Cyber Security
&
Tools, Techniques &
Threads

Presented by:

Dr. Fernaz Narin Nur

CONTENT

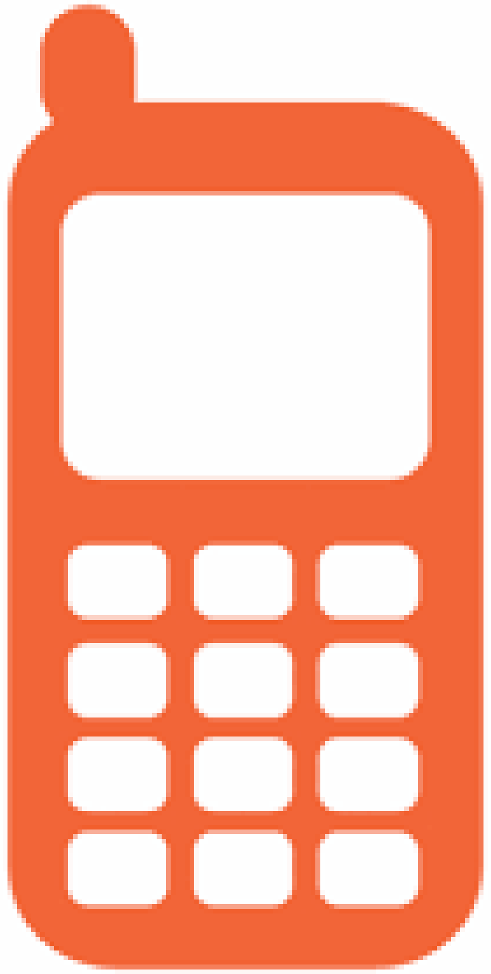
01 Enumeration

02 Vulnerability Assessment

03 Sniffing & Spoofing

04 Cryptography

05 CTF



95% of cyber attacks exploit known vulnerabilities

15,000 new vulnerabilities discover each year

CYBER INTRUSION

1,254 DATA BREACHES
EVERY PREVIOUS QUARTER FOR THE PAST 6 YEARS

VULNERABILITY WAS KNOWN

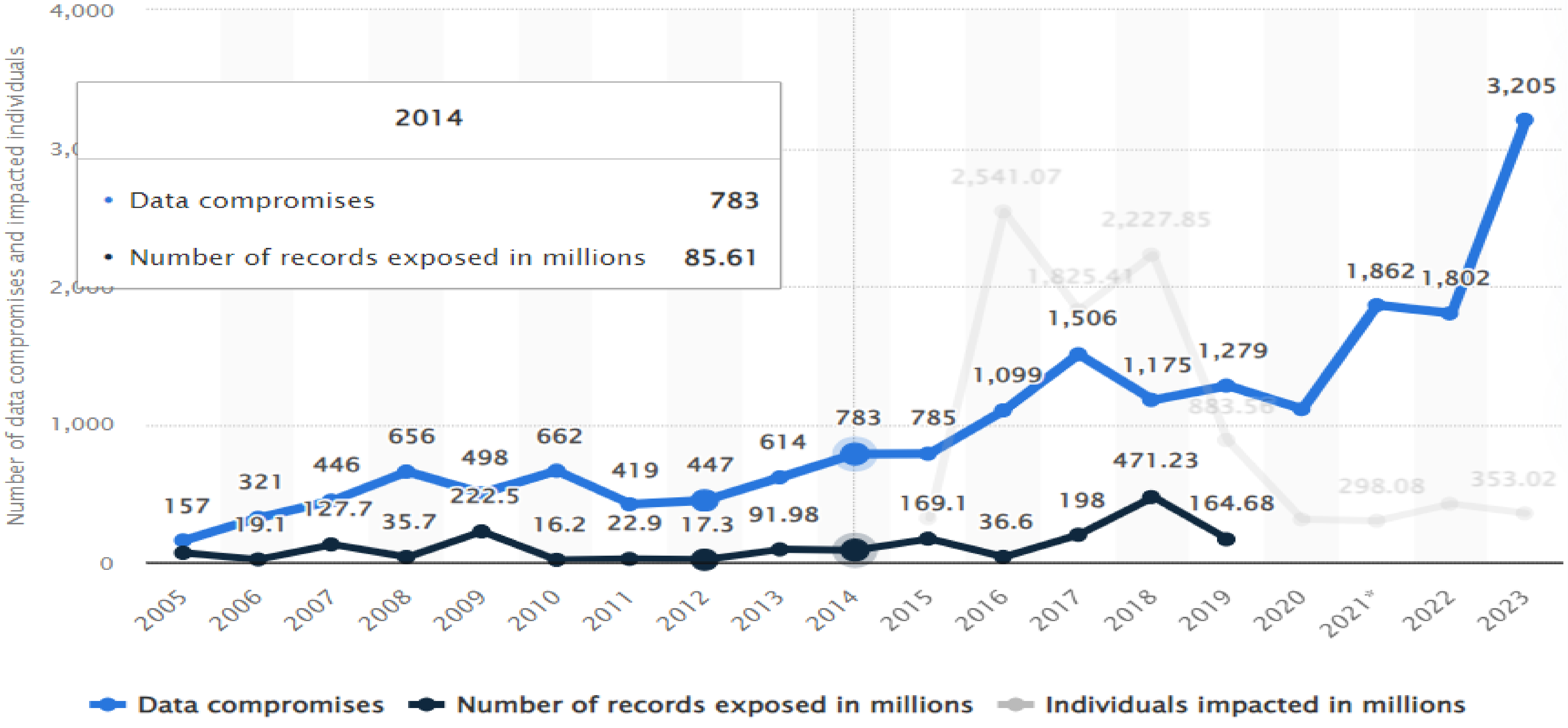
FAILURE TO IDENTIFY AND RESPOND

CYBER INTRUSION

Reputation Ruined

Bankrupt

Confidentiality lost



© Statista

Link

Enumeration is the process of gathering information about a target system or network.

Types of Enumeration

User names & Passwords

These credentials are the keys to the kingdom, allowing attackers to gain unauthorized access to systems and data.

System Information

Details about the operating system, hardware, and software versions can reveal known vulnerabilities.

Network Information

Understanding the network topology, IP addresses, and subnet masks helps attackers navigate the network and target specific systems.

Services & Applications

Identifying the services and applications running on a system can expose potential vulnerabilities in those specific programs.

Significance of Enumeration

Enumeration is a critical phase in security assessments for:

- Identifies potential attack vectors
- Informs vulnerability scanning
- Supports social engineering attacks
- Supports social engineering attacks

Research Methods for Trend Identification

Market Research

- Conduct surveys and interviews with customers, industry experts, and competitors.
- Analyze market reports and industry publications.

Social Listening

- Monitor social media platforms to understand customer sentiment and emerging trends.
- Utilize social listening tools to track industry conversations and identify influencers.

Competitive Analysis

- Closely monitor your competitors' activities, product launches, and marketing strategies.
- Identify areas where competitors are innovating and gaining traction.

Assessing the Impact of Trends and Threats & Taking Action

Once you've identified trends and threats, the next step is to assess their potential impact on your enterprise. Consider the following factors:

01 Relevance

02 Impact

03 Likelihood

04 Timeframe

Assessing the Impact of Trends and Threats & Taking Action

The insights gained from research should be used to inform strategic decision-making. Here are some ways to take action:

01 Develop new products and services

02 Invest in new technologies

03 Update security protocols

04 ....

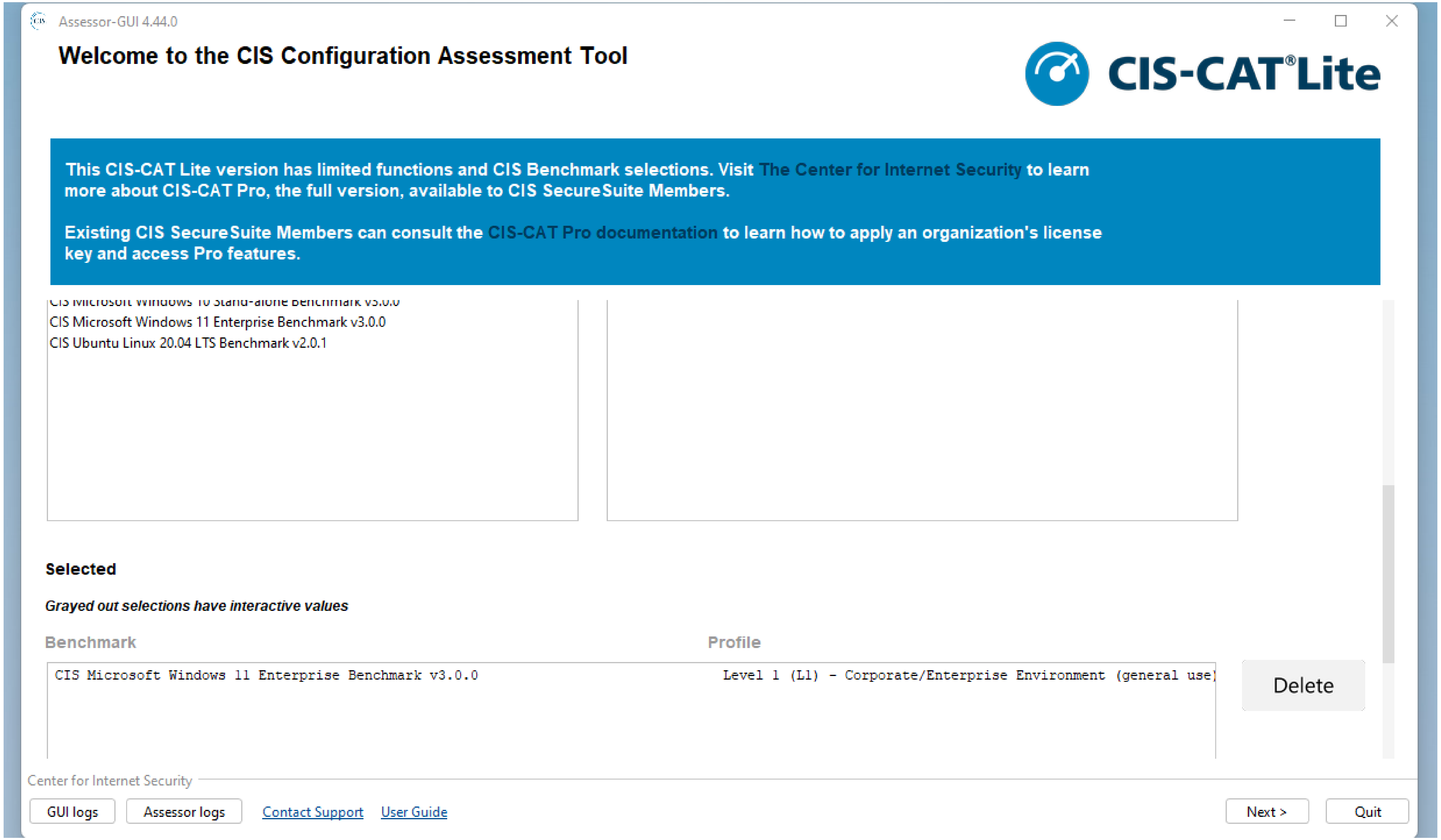
Analysis & development

Endpoint Assessment

Router Configuration & Assessment

System Hacking

Enumeration Tool



Welcome to the CIS Configuration Assessment Tool

This CIS-CAT Lite version has limited functions and CIS Benchmark selections. Visit The Center for Internet Security to learn more about CIS-CAT Pro, the full version, available to CIS SecureSuite Members.

Existing CIS SecureSuite Members can consult the CIS-CAT Pro documentation to learn how to apply an organization's license key and access Pro features.

CIS Microsoft Windows 10 Stand-alone Benchmark v3.0.0
CIS Microsoft Windows 11 Enterprise Benchmark v3.0.0
CIS Ubuntu Linux 20.04 LTS Benchmark v2.0.1

Selected

Grayed out selections have interactive values

Benchmark	Profile
CIS Microsoft Windows 11 Enterprise Benchmark v3.0.0	Level 1 (L1) - Corporate/Enterprise Environment (general use)

NIPPER

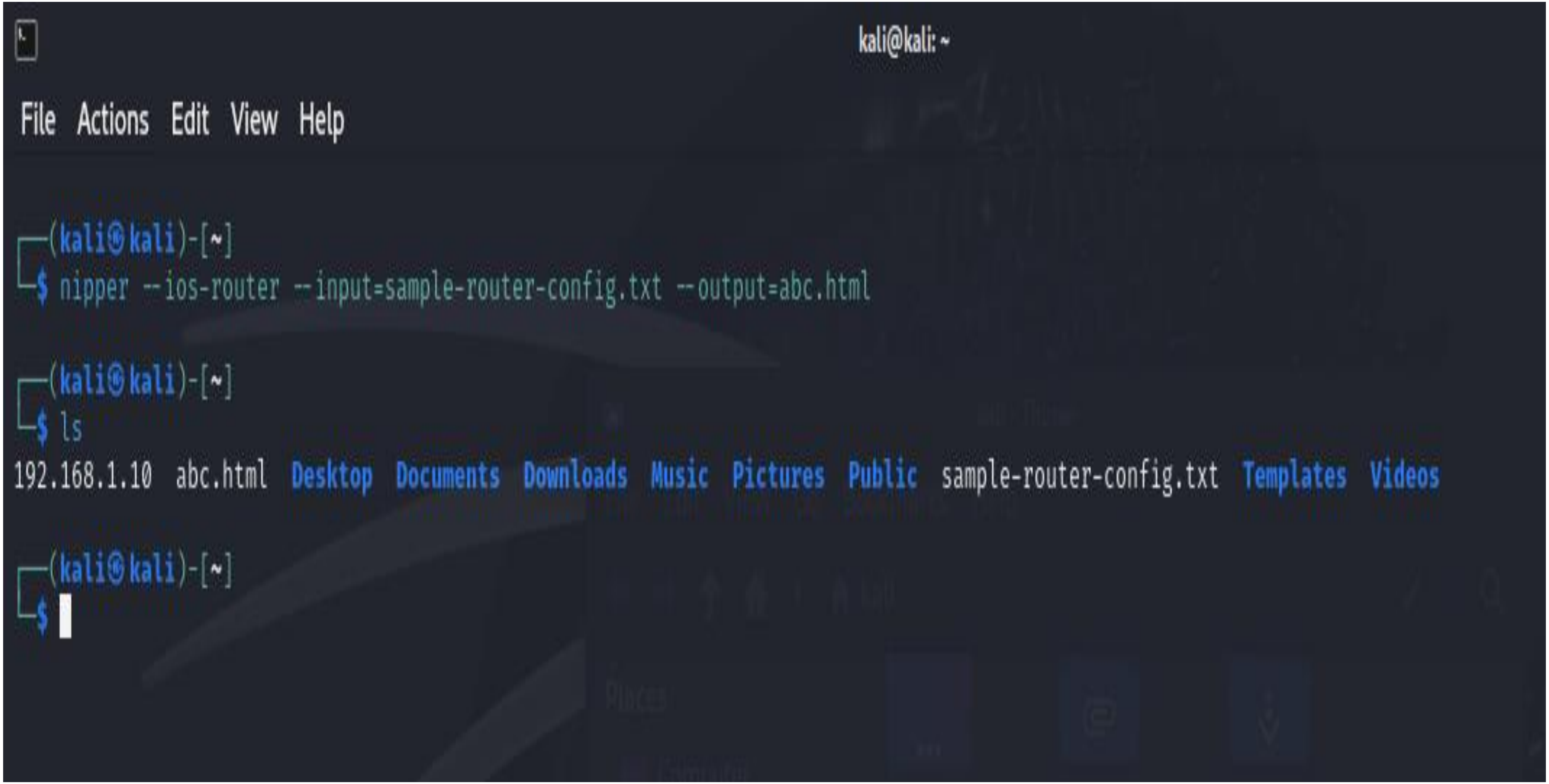
Obtain the Router Configuration File

- Log into your router (assuming it's a Cisco IOS router) via SSH or console.
- Use the following command to display the configuration:arduinoCopy code

```
show running-config
```

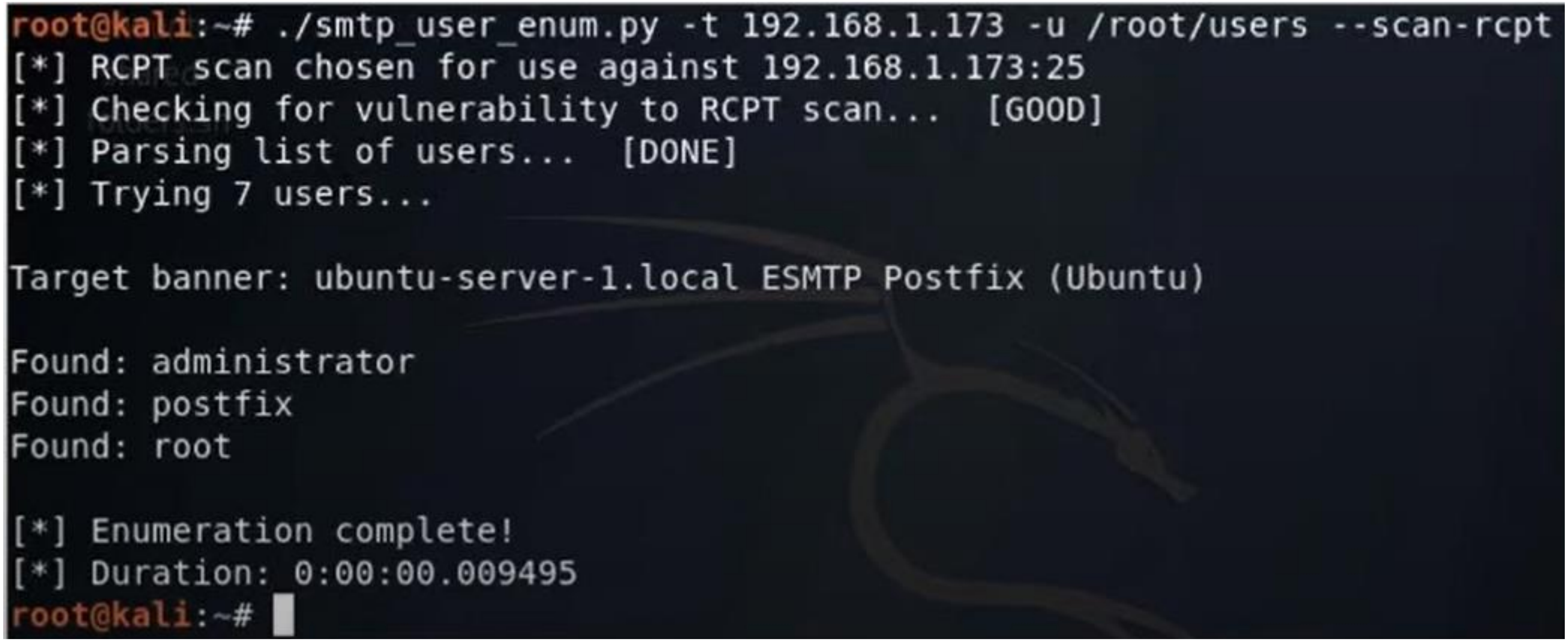
- Copy the output to a text file, or directly save the configuration to a file on the router:arduinoCopy code
- copy running-config tftp:
- Alternatively, if you have the configuration saved already, make sure it's in a readable text file,

e.g., router-config.txt.



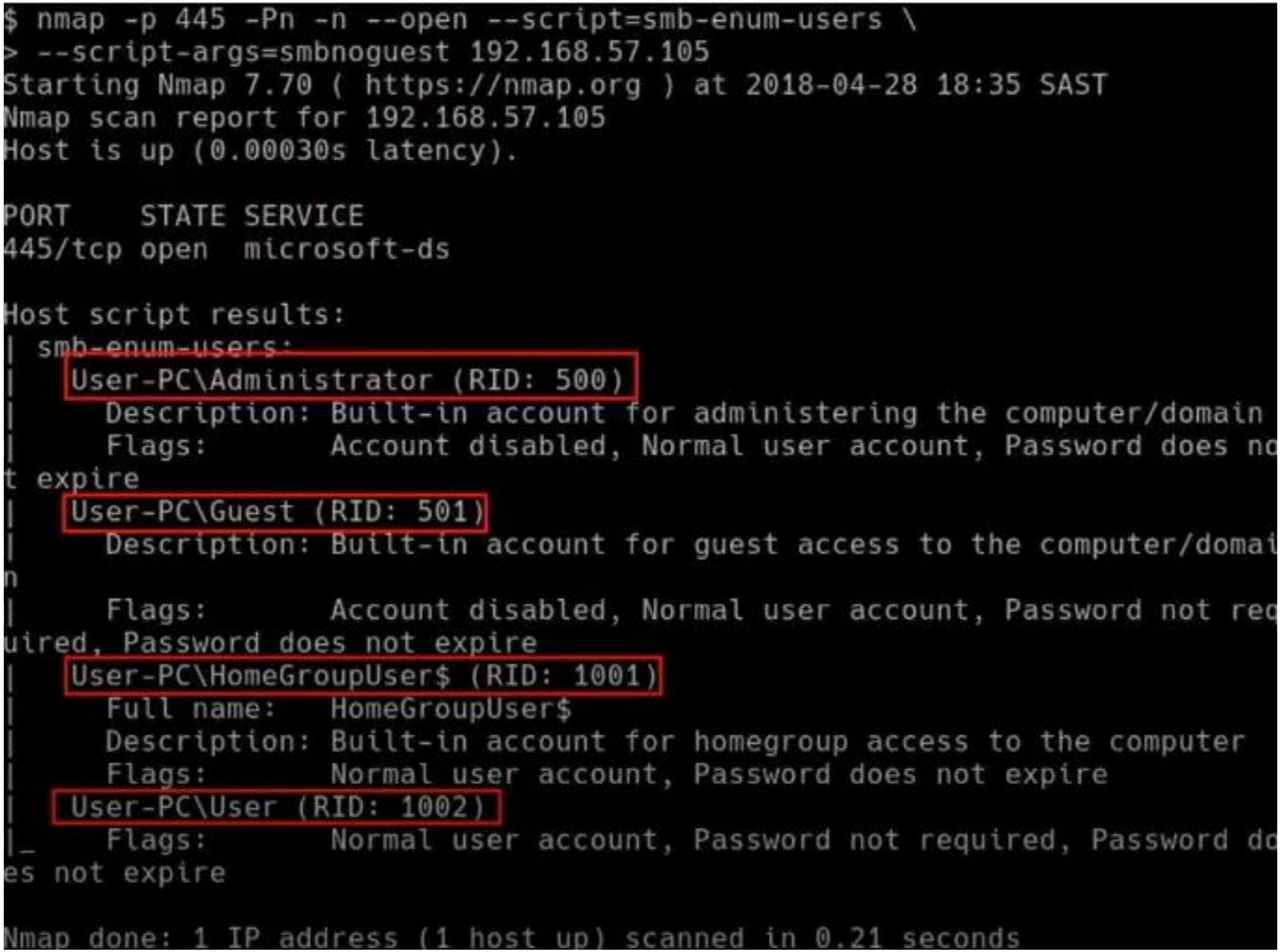
```
┌──(kali㉿kali)-[~]
└─$ nipper --ios-router --input=sample-router-config.txt --output=abc.html

┌──(kali㉿kali)-[~]
└─$ ls
192.168.1.10  abc.html  Desktop  Documents  Downloads  Music  Pictures  Public  sample-router-config.txt  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ 
```

Using Python Script – SMTP User enumeration



```
root@kali:~# ./smtp_user_enum.py -t 192.168.1.173 -u /root/users --scan-rcpt
[*] RCPT scan chosen for use against 192.168.1.173:25
[*] Checking for vulnerability to RCPT scan...   [GOOD]
[*] Parsing list of users...   [DONE]
[*] Trying 7 users...

Target banner: ubuntu-server-1.local ESMTP Postfix (Ubuntu)

Found: administrator
Found: postfix
Found: root

[*] Enumeration complete!
[*] Duration: 0:00:00.009495
root@kali:~# 
```

```
#!/usr/bin/env python3

import smtplib
import sys

def smtp_user_enum(target_ip, user_file):
# Open the file containing the list of usernames
try:
with open(user_file, 'r') as f:
users = f.read().splitlines()
except FileNotFoundError:
print(f"Error: The file {user_file} was not found.")
sys.exit(1)

# Connect to the SMTP server
try:
server = smtplib.SMTP(target_ip)
server.set_debuglevel(0)  # Set to 1 for more verbosity
except Exception as e:
print(f"Error: Unable to connect to SMTP server at {target_ip}.")
print(f"Details: {e}")
sys.exit(1)

# Attempt to enumerate users using the RCPT TO command
for user in users:
try:
# Initiate the SMTP conversation
server.ehlo_or_helo_if_needed()

# Send the RCPT TO command with the username
response = server.rcpt(f"<{user}@example.com>")

# Interpret the server's response
if response[0] == 250:  # 250 is the typical "OK" response
print(f"[+] Valid user found: {user}")
else:
print(f"[-] Invalid user: {user}")

except Exception as e:
print(f"Error: {e}")
continue

# Close the connection to the SMTP server
server.quit()

if __name__ == "__main__":
if len(sys.argv) != 3:
print(f"Usage: {sys.argv[0]} <target_ip> <user_file>")
sys.exit(1)

target_ip = sys.argv[1]
user_file = sys.argv[2]

smtp_user_enum(target_ip, user_file)
```



```
$ nmap -p 445 -Pn -n --open --script=smb-enum-users \
> --script-args=smbnoguest 192.168.57.105
Starting Nmap 7.70 ( https://nmap.org ) at 2018-04-28 18:35 SAST
Nmap scan report for 192.168.57.105
Host is up (0.00030s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-users:
|   User-PC\Administrator (RID: 500)
|     Description: Built-in account for administering the computer/domain
|     Flags:       Account disabled, Normal user account, Password does no
t expire
|   User-PC\Guest (RID: 501)
|     Description: Built-in account for guest access to the computer/domai
n
|     Flags:       Account disabled, Normal user account, Password not req
uired, Password does not expire
|   User-PC\HomeGroupUser$ (RID: 1001)
|     Full name:   HomeGroupUser$
|     Description: Built-in account for homegroup access to the computer
|     Flags:       Normal user account, Password does not expire
|   User-PC\User (RID: 1002)
|_    Flags:       Normal user account, Password not required, Password do
es not expire

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Research scope

1. Advanced Enumeration Techniques

Topic: "**Evaluating Advanced Techniques for Network Enumeration in Modern Environments**."

Objective: Analyze modern enumeration methods and their effectiveness in identifying network services and vulnerabilities.

Focus Areas:

Impact of IPv6 on enumeration techniques.

Effectiveness of tools like Nmap, Nessus, and Netcat.

2. Enumeration in IoT Devices

Topic: "**Challenges and Solutions in Enumerating IoT Devices in a Smart Home Environment**."

Objective: Explore methods for enumerating IoT devices and identifying vulnerabilities specific to interconnected systems.

Focus Areas:

Challenges posed by device heterogeneity.

Tools and techniques for IoT-specific enumeration.

VULNERABILITY ASSESSMENT

A process of defining, identifying, classifying and prioritizing security weaknesses and vulnerabilities in system, including servers, applications and network infrastructures.

TYPES

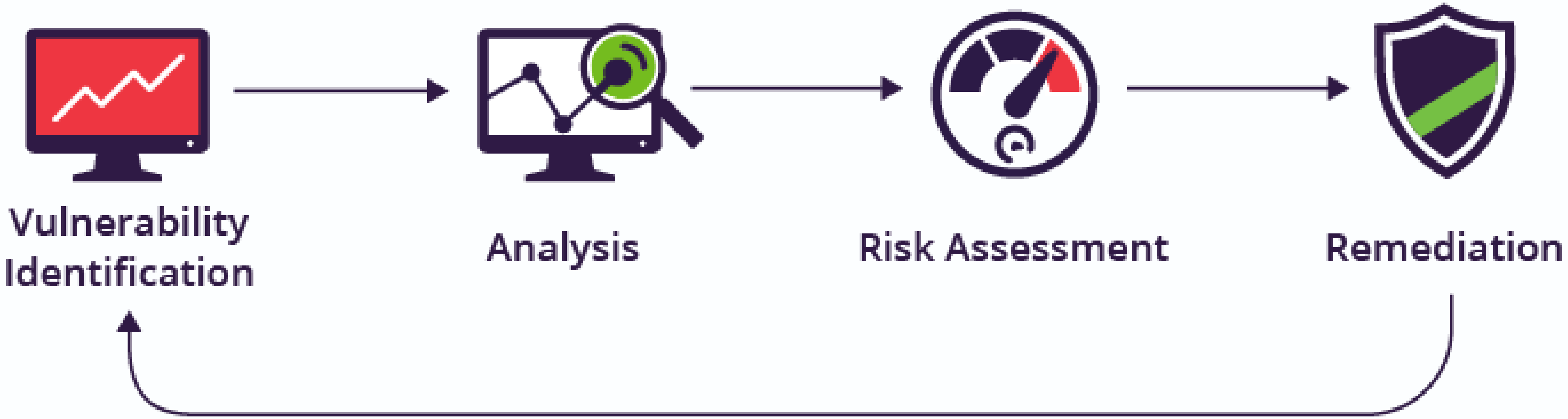
NETWORK BASED SCANS

HOST BASED SCANS

APPLICATION SCANS

DATABASE SCANS

VULNETRABILITY ASSESSMENT PROCESS



VULNERABILITY ASSESSMENT TOOLS

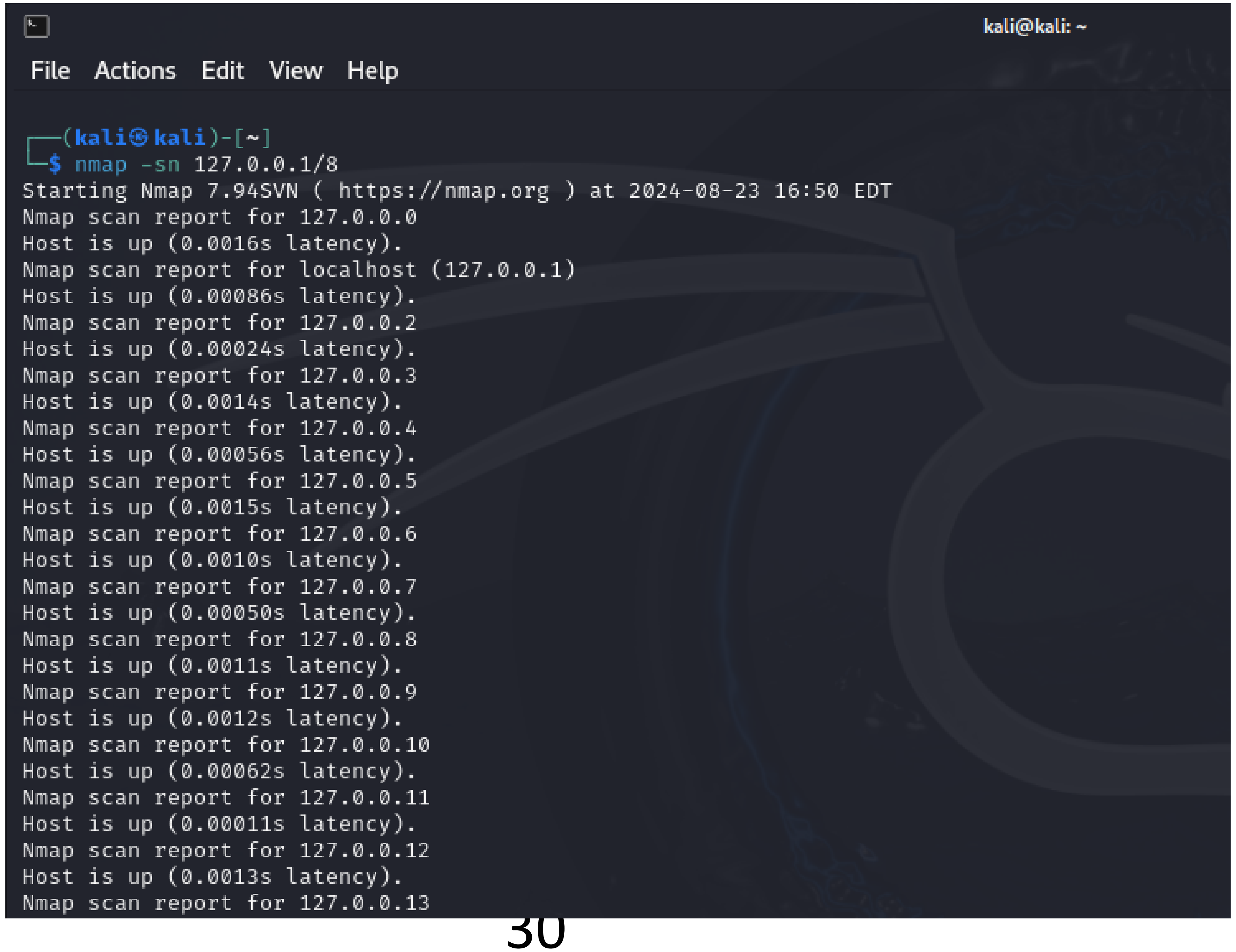
Nessus
nexpose
Nikto
NMAP
OpenVAS
Open Vulnerability Assessment System

VULNERABILITY ASSESSMENT TOOLS FOR CLOUD

Vulnerability Assessment

nmap -sn 192.168.1.0/24

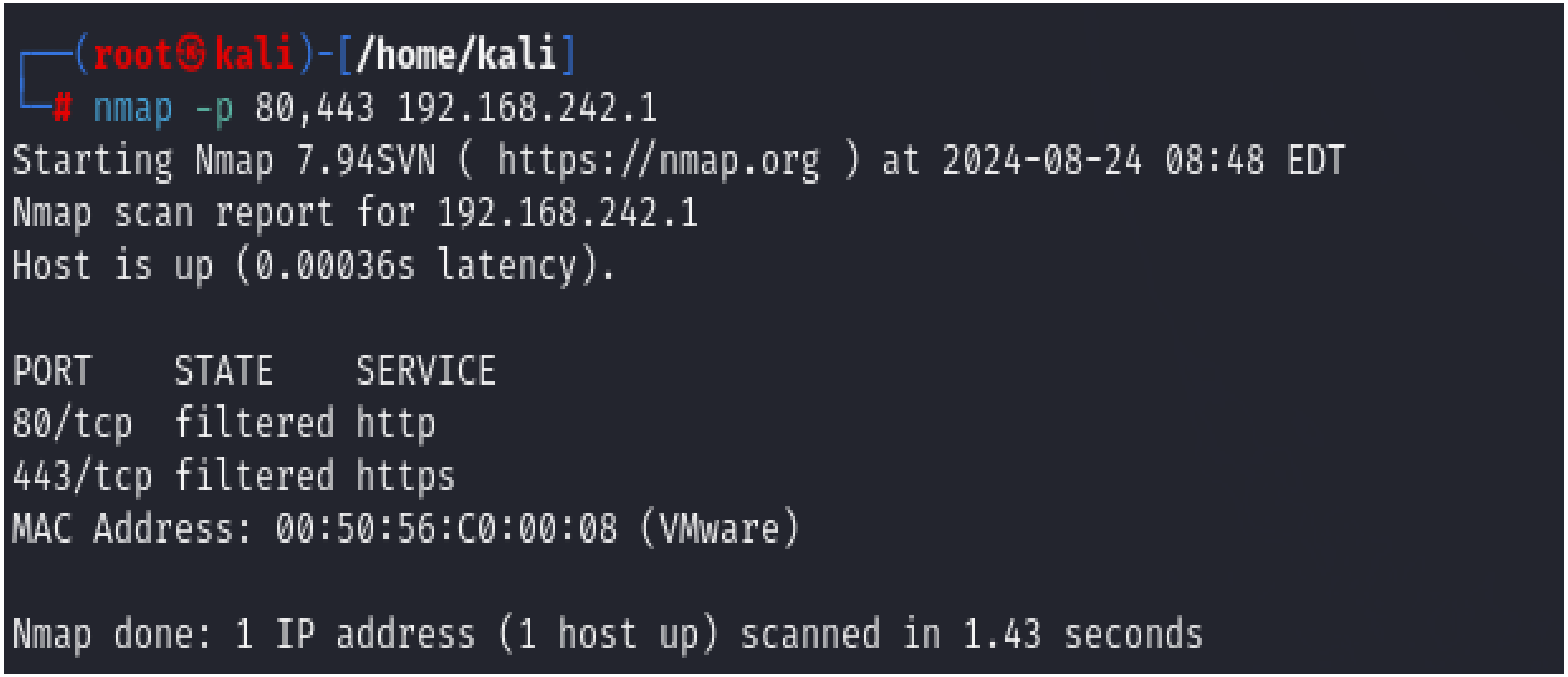
Objective: Show how to identify live hosts in a network.



```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 127.0.0.1/8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-23 16:50 EDT
Nmap scan report for 127.0.0.0
Host is up (0.0016s latency).
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00086s latency).
Nmap scan report for 127.0.0.2
Host is up (0.00024s latency).
Nmap scan report for 127.0.0.3
Host is up (0.0014s latency).
Nmap scan report for 127.0.0.4
Host is up (0.00056s latency).
Nmap scan report for 127.0.0.5
Host is up (0.0015s latency).
Nmap scan report for 127.0.0.6
Host is up (0.0010s latency).
Nmap scan report for 127.0.0.7
Host is up (0.00050s latency).
Nmap scan report for 127.0.0.8
Host is up (0.0011s latency).
Nmap scan report for 127.0.0.9
Host is up (0.0012s latency).
Nmap scan report for 127.0.0.10
Host is up (0.00062s latency).
Nmap scan report for 127.0.0.11
Host is up (0.00011s latency).
Nmap scan report for 127.0.0.12
Host is up (0.0013s latency).
Nmap scan report for 127.0.0.13
```

nmap -p 80,443 192.168.1.10

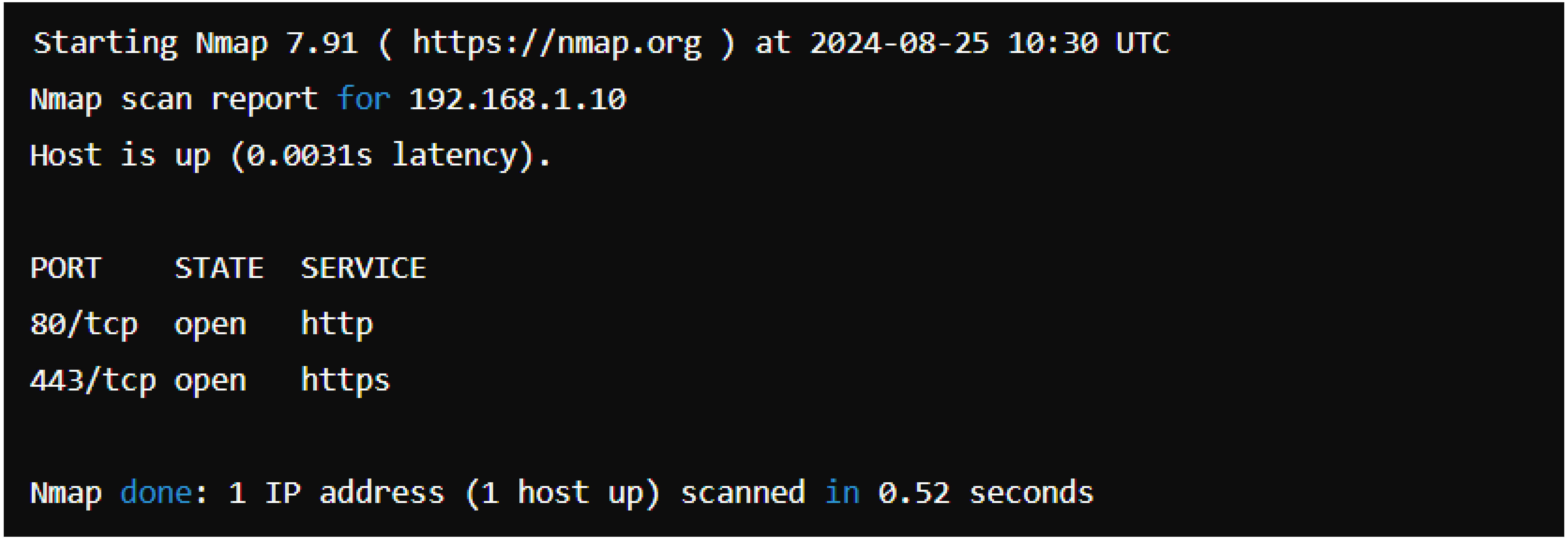
Objective: identify open ports and services specifically on ports 80 (HTTP) and 443 (HTTPS) on the target host



```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p 80,443 192.168.242.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-24 08:48 EDT
Nmap scan report for 192.168.242.1
Host is up (0.00036s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
```

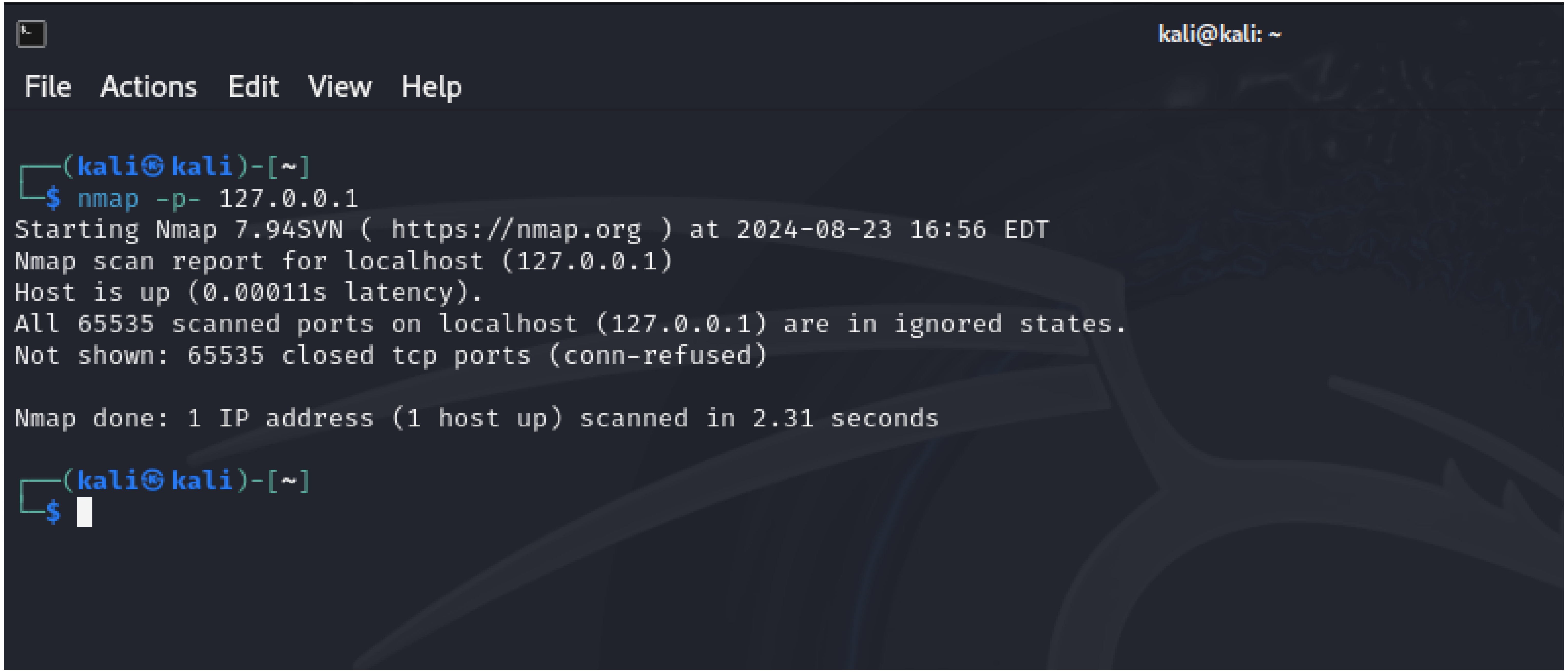
nmap -p 80,443 192.168.1.10

Objective: identify open ports and services specifically on ports 80 (HTTP) and 443 (HTTPS) on the target host



```
Starting Nmap 7.91 ( https://nmap.org ) at 2024-08-25 10:30 UTC
Nmap scan report for 192.168.1.10
Host is up (0.0031s latency).

PORT    STATE  SERVICE
80/tcp  open   http
443/tcp open   https

Nmap done: 1 IP address (1 host up) scanned in 0.52 seconds
```

nmap -p- 192.168.1.10

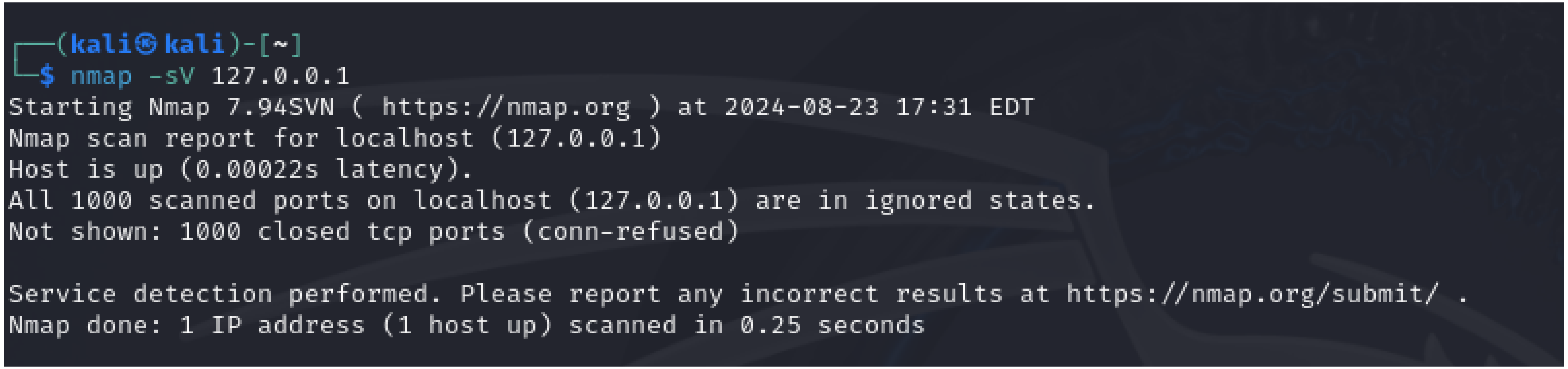


```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 127.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-23 16:56 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00011s latency).
All 65535 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 65535 closed tcp ports (conn-refused)

Nmap done: 1 IP address (1 host up) scanned in 2.31 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Service Version Detection:

nmap -sV 192.168.1.10

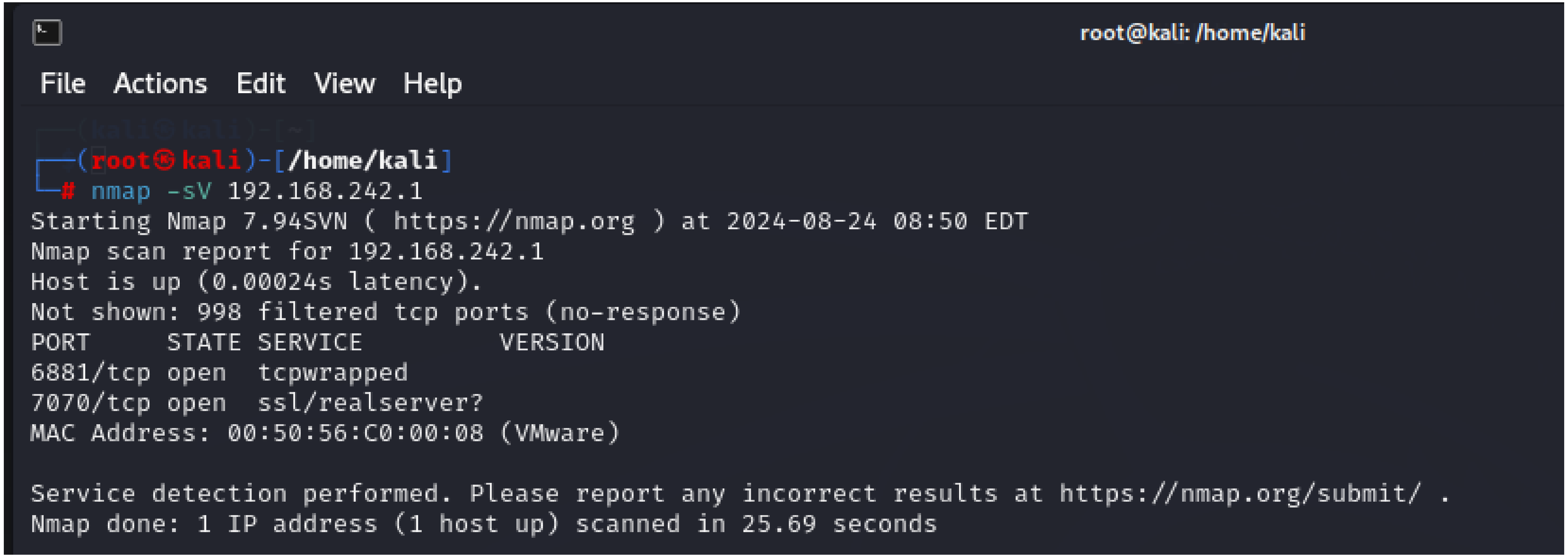
Objective: Demonstrate how to identify open ports and running services on a target.



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 127.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-23 17:31 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00022s latency).
All 1000 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

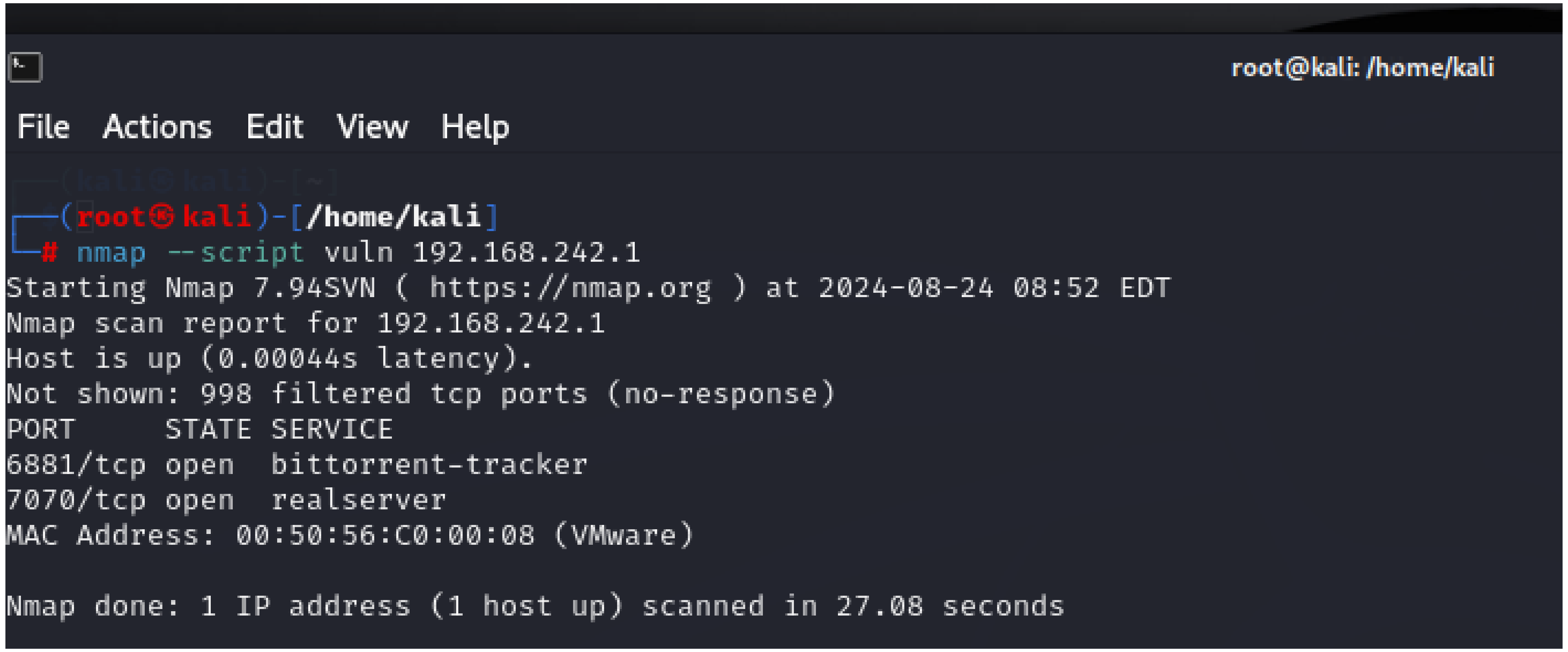
OS Detection:

nmap -O 192.168.1.10



```
┌──(kali㉿kali)-[~]
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV 192.168.242.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-24 08:50 EDT
Nmap scan report for 192.168.242.1
Host is up (0.00024s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE         VERSION
6881/tcp open  tcpwrapped
7070/tcp open  ssl/realserver?
MAC Address: 00:50:56:C0:00:08 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.69 seconds
```

Using Nmap Scripting Engine (NSE):
nmap --script vuln 192.168.1.10



```
┌──(kali㉿kali)-[~]
┌──(root㉿kali)-[/home/kali]
└─# nmap --script vuln 192.168.242.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-24 08:52 EDT
Nmap scan report for 192.168.242.1
Host is up (0.00044s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
6881/tcp open  bittorrent-tracker
7070/tcp open  realserver
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 27.08 seconds
```

Web Server Enumeration:

```
nmap -p 80,443 --script http-enum 192.168.1.100
```

Objective: Show how to identify directories, files, and potential vulnerabilities in the web service.

Research Scope

1. Vulnerability Assessment

- Identify known vulnerabilities in your organization’s systems and software.
- Cross-check the CVE list with your software inventory to find affected products.

2. Patch Management

- Use the CVE details to prioritize patches or updates for vulnerable systems.
- Apply vendor-provided security patches or temporary mitigations.

3. Risk Management

- Evaluate the severity of vulnerabilities using CVSS scores and impact metrics.
- Identify critical vulnerabilities that pose the most risk to your organization and allocate resources to address them.

4. Incident Response

- Monitor for active exploits tied to CVEs.
- Use the list to detect and mitigate vulnerabilities that attackers might exploit during an ongoing incident.

Research Scope

5. Compliance and Audit

- Ensure your organization complies with industry security standards by addressing listed CVEs.
- Provide evidence of vulnerability scanning and remediation efforts during audits.

6. Predictive Vulnerability Scoring

- Use ML models to predict the likelihood of exploitation for newly discovered vulnerabilities.
- Train the model using historical CVE data, exploit databases, and attack patterns.
- Incorporate metrics like Common Vulnerability Scoring System (CVSS) scores, patch availability, and exploit trends to prioritize vulnerabilities for remediation.

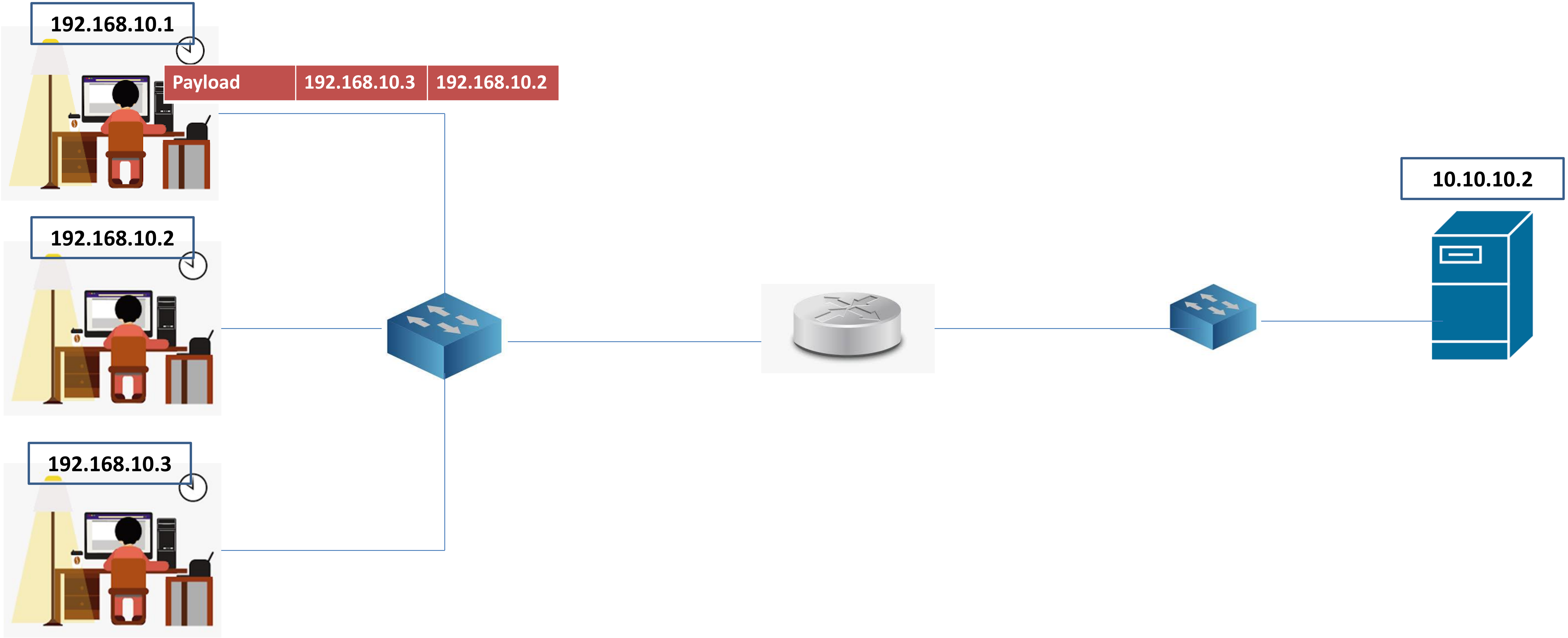
7. Automated Threat Detection

- Implement ML algorithms to analyze network traffic and system logs in real-time to identify signs of vulnerabilities being exploited.
- Use anomaly detection models (e.g., clustering or neural networks) to flag unusual patterns associated with known vulnerabilities.

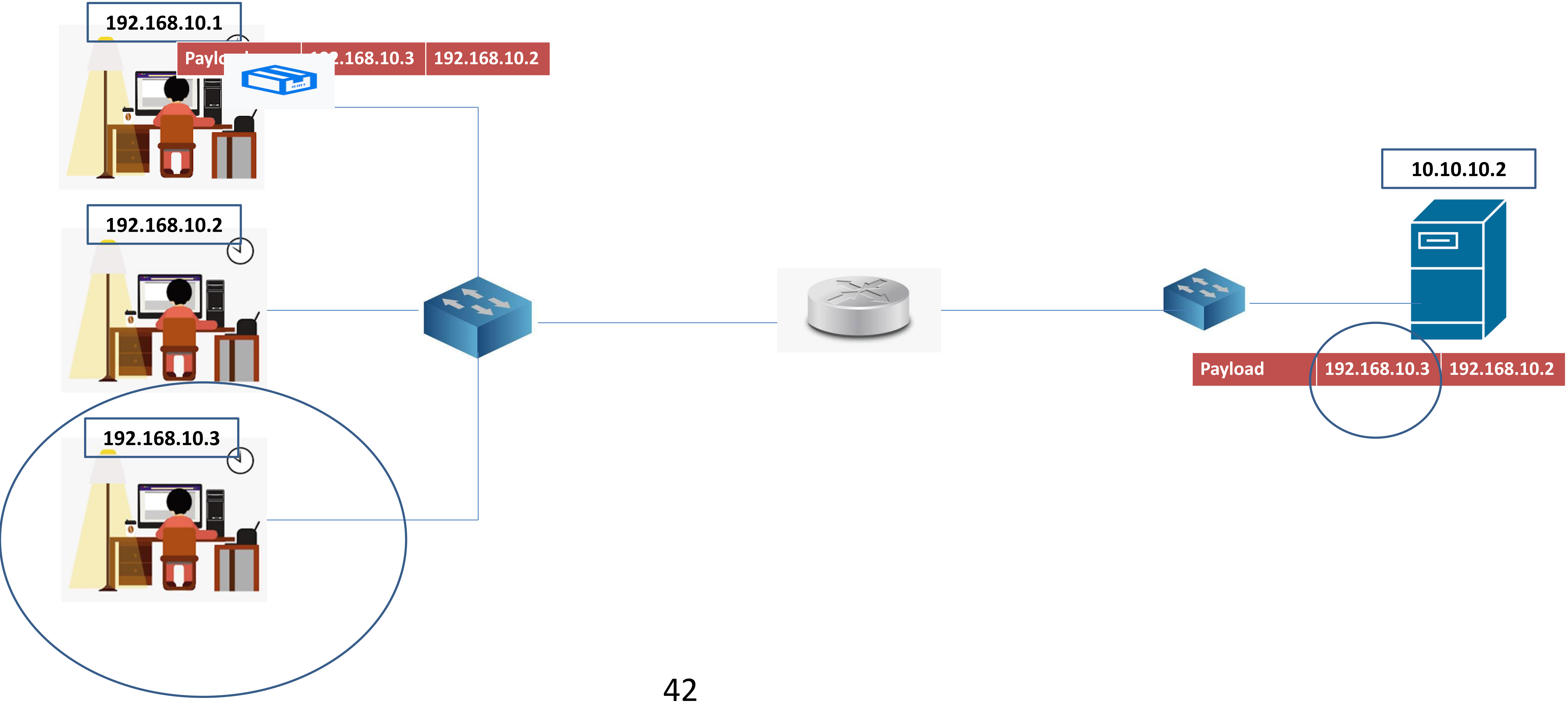
Sniffing & Spoofing

Awareness and Implement learning in work environment.

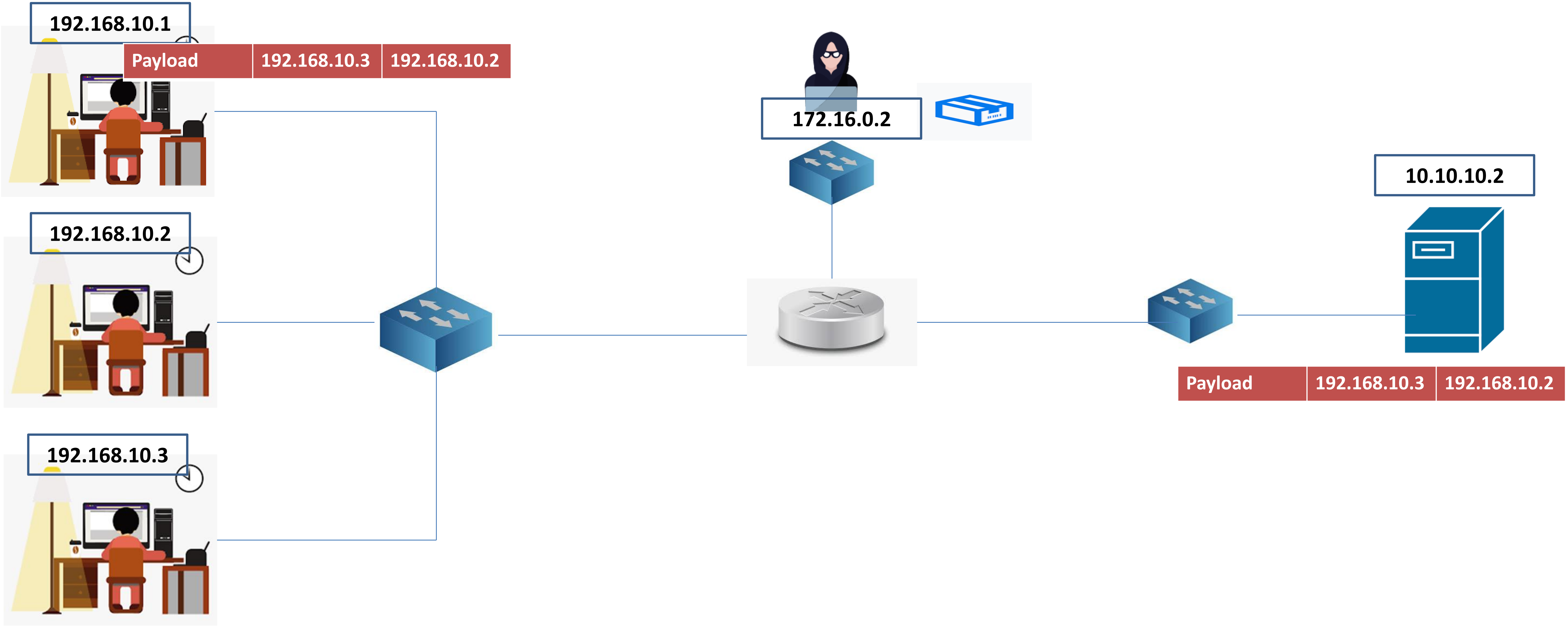
IP Spoofing Attacks



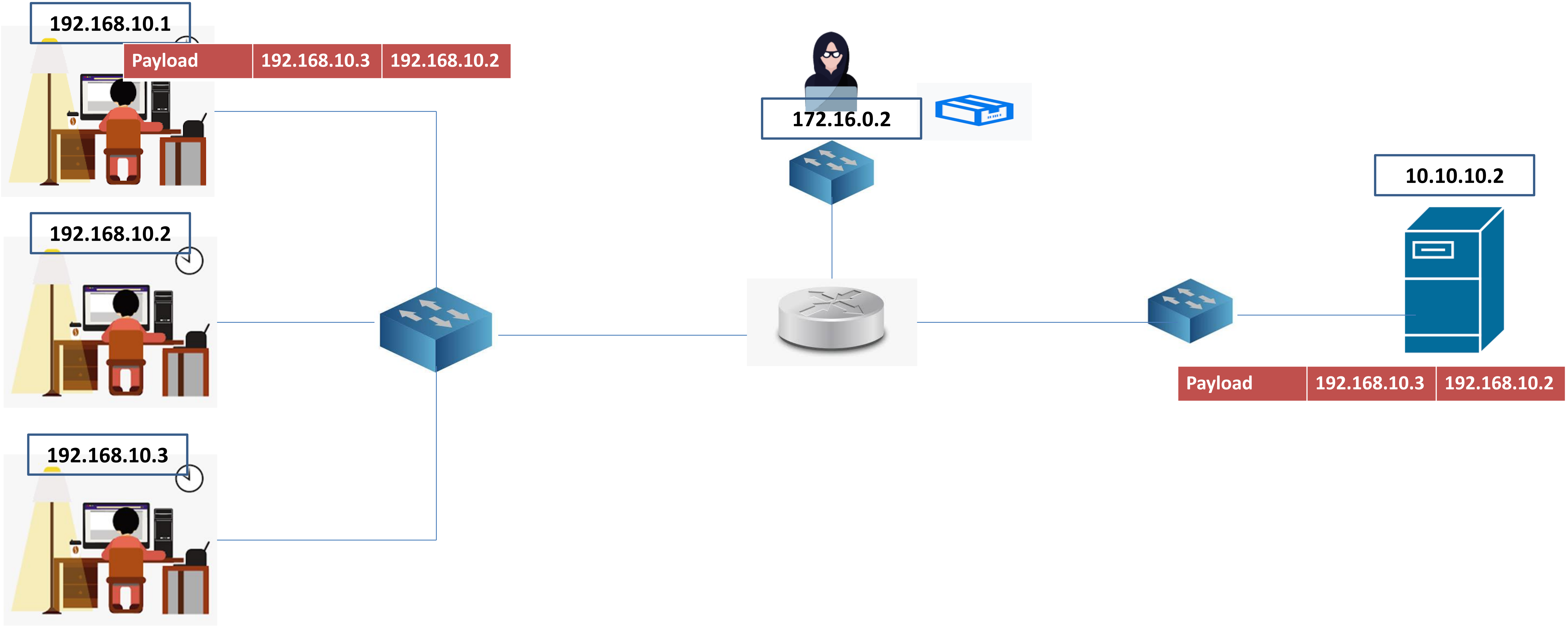
IP Spoofing Attacks



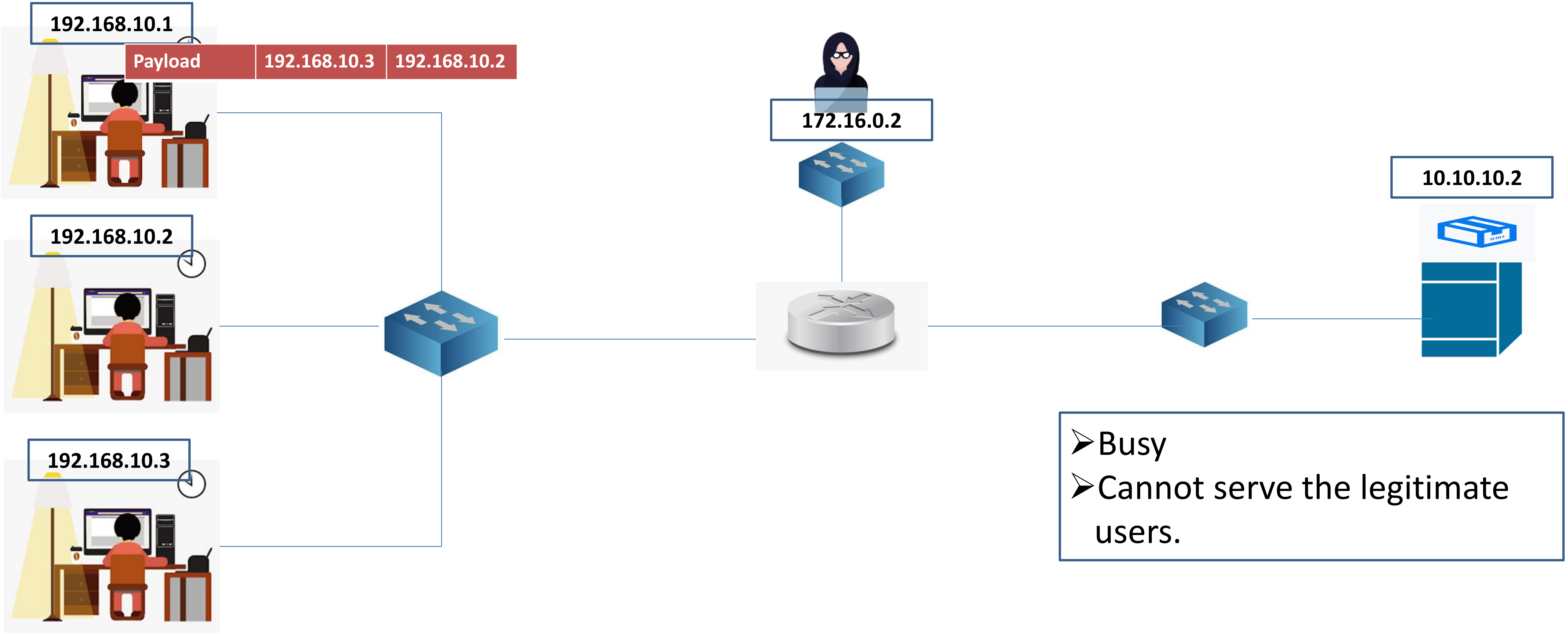
IP Spoofing Attacks



IP Spoofing Attacks



IP Spoofing Attacks



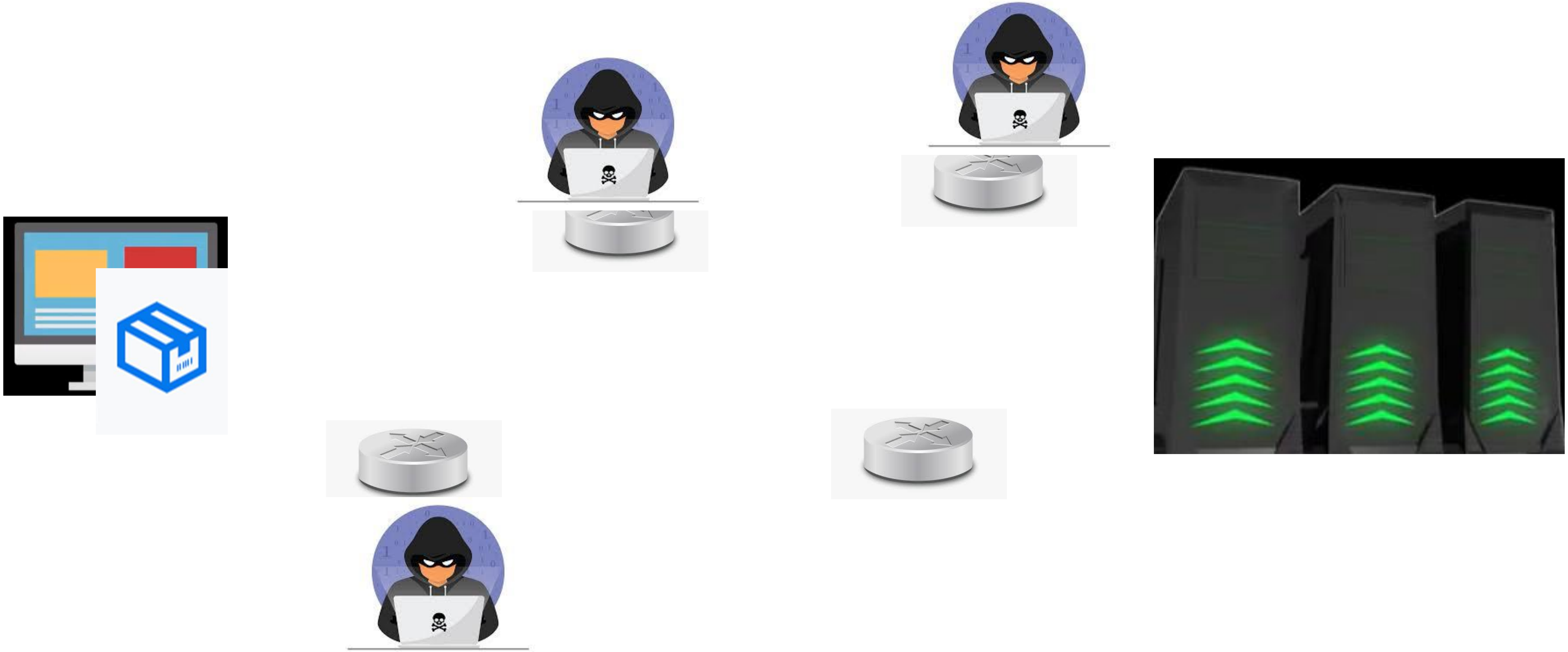
- ➢Busy
- ➢Cannot serve the legitimate users.

IP Sniffing Attacks

Payload	DA	SA

IP Sniffing Attacks

IP Sniffing Attacks



Spoofing

- Involves creating fake data packets with forged source addresses.
- By making it seem like they're coming from a trusted source, attackers can trick devices on the network into sending them data or granting access.
- Spoofing is an active attack, as it involves manipulating the network traffic.

Sniffing

- Involves eavesdropping on network traffic.
- Attackers use software called packet sniffers to capture data packets traveling across a network.
- These packets can contain sensitive information like usernames, passwords, and emails, if they are not encrypted.
- Sniffing is a passive attack, meaning the attacker doesn't alter the network traffic, they just listen in.

Analogy

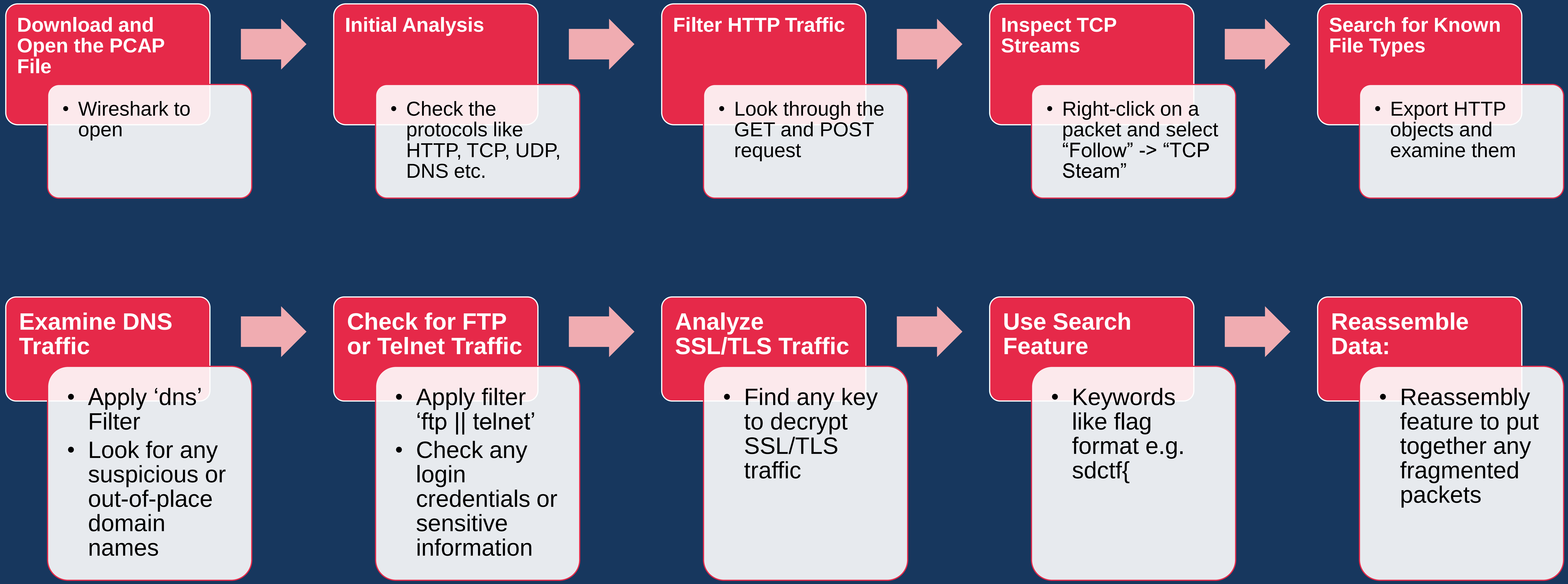
Spooof for a strglets!
Teislaw
Spoofing

Spoofing is like pretending to be one of the people in the conversation. You can trick the other person into giving you information or doing something they wouldn't normally do.

Sniffing is like eavesdropping on a conversation between two people. You can hear what they're saying, but you can't change the conversation itself.

Packet Sniffing and Analysis Tool - Wireshark

Packet Analysis



Download and Open the PCAP File

- Wireshark to open

Initial Analysis

- Check the protocols like HTTP, TCP, UDP, DNS etc.

Filter HTTP Traffic

- Look through the GET and POST request

Inspect TCP Streams

- Right-click on a packet and select “Follow” -> “TCP Steam”

Search for Known File Types

- Export HTTP objects and examine them

Examine DNS Traffic

- Apply ‘dns’ Filter
- Look for any suspicious or out-of-place domain names

Check for FTP or Telnet Traffic

- Apply filter ‘ftp || telnet’
- Check any login credentials or sensitive information

Analyze SSL/TLS Traffic

- Find any key to decrypt SSL/TLS traffic

Use Search Feature

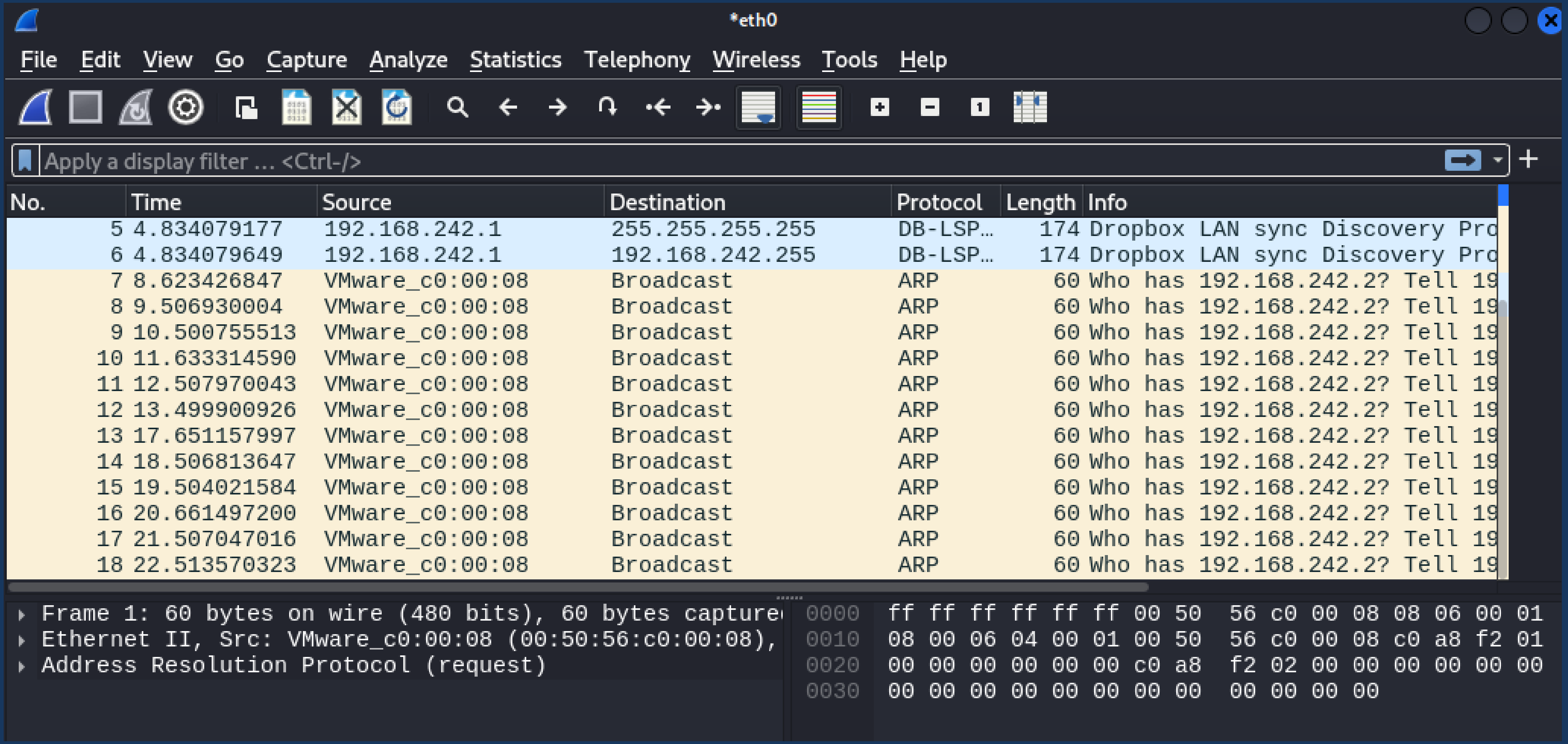
- Keywords like flag format e.g. sdctf{

Reassemble Data:

- Reassembly feature to put together any fragmented packets

Basic Packet Capture with Wireshark

Objective: Demonstrate how to capture live network traffic.



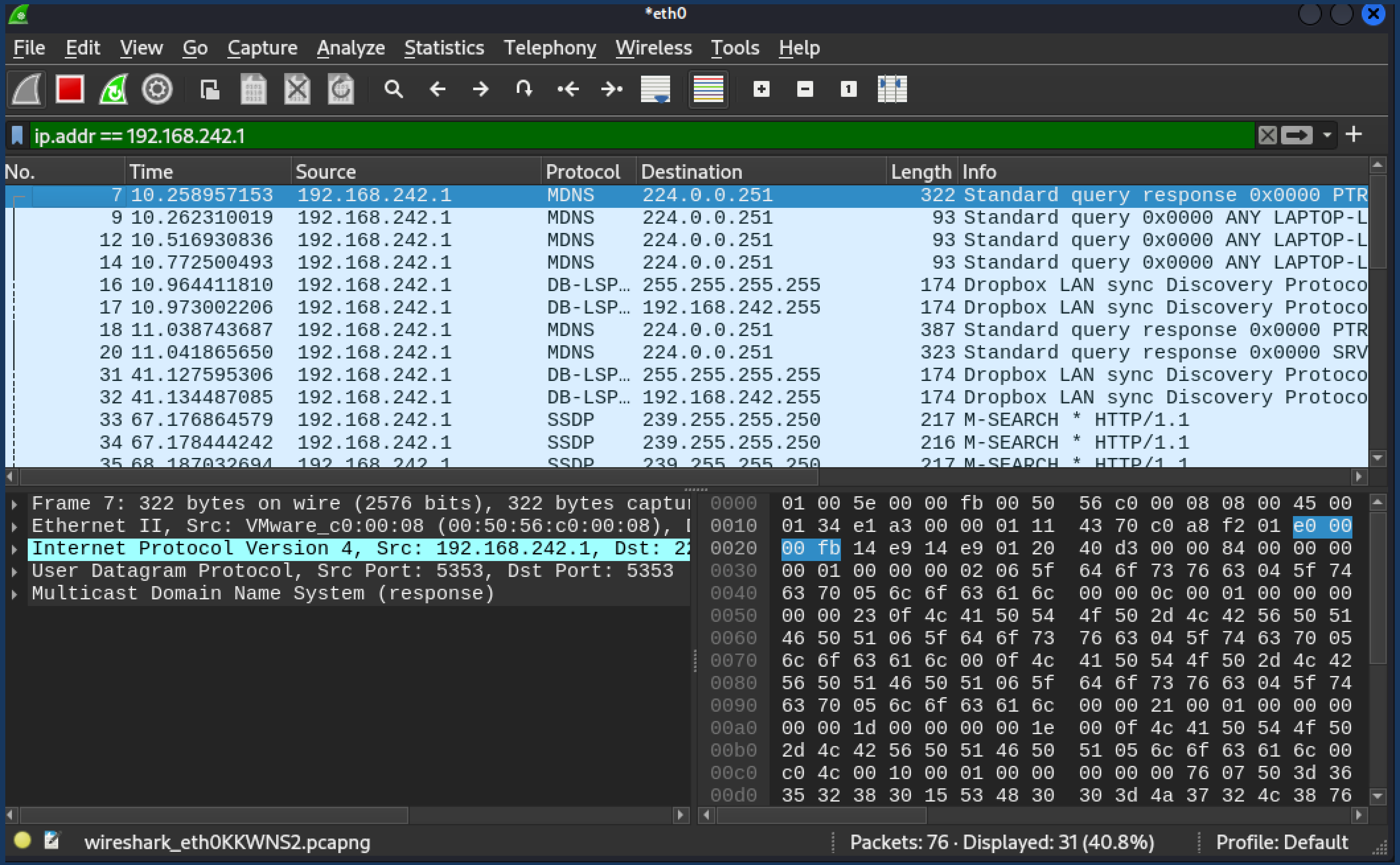
```
No.      Time          Source             Destination        Protocol  Length Info
       5 4.834079177   192.168.242.1      255.255.255.255    DB-LSP…      174 Dropbox LAN sync Discovery Pro
       6 4.834079649   192.168.242.1      192.168.242.255    DB-LSP…      174 Dropbox LAN sync Discovery Pro
       7 8.623426847   VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
       8 9.506930004   VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
       9 10.500755513  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      10 11.633314590  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      11 12.507970043  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      12 13.499900926  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      13 17.651157997  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      14 18.506813647  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      15 19.504021584  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      16 20.661497200  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      17 21.507047016  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
      18 22.513570323  VMware_c0:00:08    Broadcast          ARP           60 Who has 192.168.242.2? Tell 19
```

```
▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes capture
▸ Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08),
▸ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 00 50  56 c0 00 08 08 06 00 01
0010  08 00 06 04 00 01 00 50  56 c0 00 08 c0 a8 f2 01
0020  00 00 00 00 00 00 c0 a8  f2 02 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  00 00 00 00
```

Filtering Traffic with Wireshark

Objective: How to filter captured traffic for specific data.



```
ip.addr == 192.168.242.1
```

```
No.   Time          Source          Protocol  Destination      Length Info
    7 10.258957153  192.168.242.1   MDNS      224.0.0.251        322 Standard query response 0x0000 PTR
    9 10.262310019  192.168.242.1   MDNS      224.0.0.251         93 Standard query 0x0000 ANY LAPTOP-L
   12 10.516930836  192.168.242.1   MDNS      224.0.0.251         93 Standard query 0x0000 ANY LAPTOP-L
   14 10.772500493  192.168.242.1   MDNS      224.0.0.251         93 Standard query 0x0000 ANY LAPTOP-L
   16 10.964411810  192.168.242.1   DB-LSP…   255.255.255.255    174 Dropbox LAN sync Discovery Protoco
   17 10.973002206  192.168.242.1   DB-LSP…   192.168.242.255    174 Dropbox LAN sync Discovery Protoco
   18 11.038743687  192.168.242.1   MDNS      224.0.0.251        387 Standard query response 0x0000 PTR
   20 11.041865650  192.168.242.1   MDNS      224.0.0.251        323 Standard query response 0x0000 SRV
   31 41.127595306  192.168.242.1   DB-LSP…   255.255.255.255    174 Dropbox LAN sync Discovery Protoco
   32 41.134487085  192.168.242.1   DB-LSP…   192.168.242.255    174 Dropbox LAN sync Discovery Protoco
   33 67.176864579  192.168.242.1   SSDP      239.255.255.250    217 M-SEARCH * HTTP/1.1
   34 67.178444242  192.168.242.1   SSDP      239.255.255.250    216 M-SEARCH * HTTP/1.1
   35 68.187032694  192.168.242.1   SSDP      239.255.255.250    217 M-SEARCH * HTTP/1.1
```

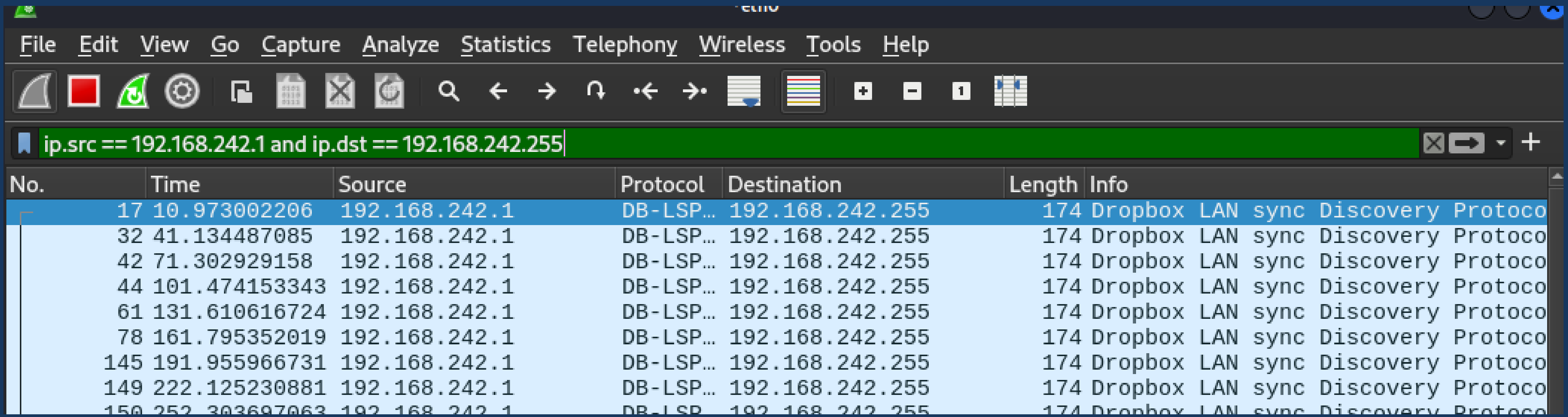
```
▸ Frame 7: 322 bytes on wire (2576 bits), 322 bytes captu
▸ Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), I
▸ Internet Protocol Version 4, Src: 192.168.242.1, Dst: 22
▸ User Datagram Protocol, Src Port: 5353, Dst Port: 5353
▸ Multicast Domain Name System (response)
```

```
0000  01 00 5e 00 00 fb 00 50  56 c0 00 08 08 00 45 00
0010  01 34 e1 a3 00 00 01 11  43 70 c0 a8 f2 01 e0 00
0020  00 fb 14 e9 14 e9 01 20  40 d3 00 00 84 00 00 00
0030  00 01 00 00 00 02 06 5f  64 6f 73 76 63 04 5f 74
0040  63 70 05 6c 6f 63 61 6c  00 00 0c 00 01 00 00 00
0050  00 00 23 0f 4c 41 50 54  4f 50 2d 4c 42 56 50 51
0060  46 50 51 06 5f 64 6f 73  76 63 04 5f 74 63 70 05
0070  6c 6f 63 61 6c 00 0f 4c  41 50 54 4f 50 2d 4c 42
0080  56 50 51 46 50 51 06 5f  64 6f 73 76 63 04 5f 74
0090  63 70 05 6c 6f 63 61 6c  00 00 21 00 01 00 00 00
00a0  00 00 1d 00 00 00 00 1e  00 0f 4c 41 50 54 4f 50
00b0  2d 4c 42 56 50 51 46 50  51 05 6c 6f 63 61 6c 00
00c0  c0 4c 00 10 00 01 00 00  00 00 00 76 07 50 3d 36
00d0  35 32 38 30 15 53 48 30  30 3d 4a 37 32 4c 38 76
```

wireshark_eth0KKWNS2.pcapng | Packets: 76 · Displayed: 31 (40.8%) | Profile: Default

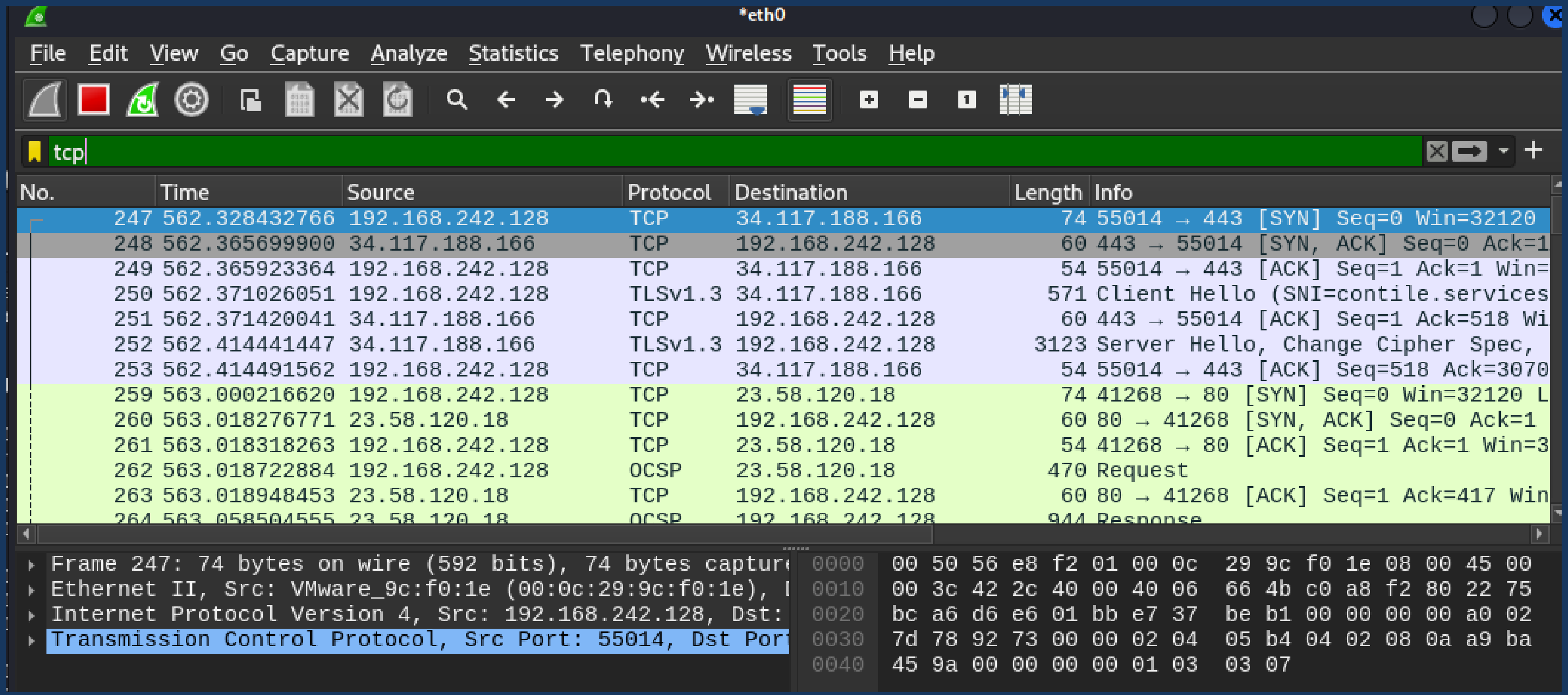
Follow TCP Stream

Objective: Demonstrate how to track a conversation between two endpoints.



ip.src == 192.168.242.1 and ip.dst == 192.168.242.255

```
No.   Time          Source          Protocol  Destination      Length Info
   17 10.973002206  192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
   32 41.134487085  192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
   42 71.302929158  192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
   44 101.474153343 192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
   61 131.610616724 192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
   78 161.795352019 192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
  145 191.955966731 192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
  149 222.125230881 192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
  150 252.303697063 192.168.242.1   DB-LSP…   192.168.242.255     174 Dropbox LAN sync Discovery Protoco
```

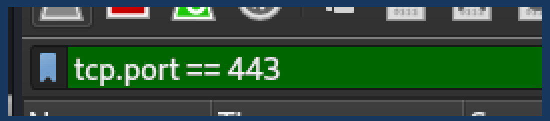


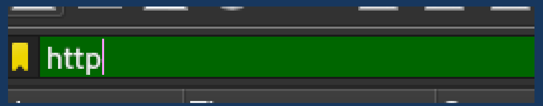
tcp

```
No.   Time          Source            Protocol Destination       Length Info
  247 562.328432766 192.168.242.128   TCP      34.117.188.166        74 55014 → 443 [SYN] Seq=0 Win=32120
  248 562.365699900 34.117.188.166    TCP      192.168.242.128       60 443 → 55014 [SYN, ACK] Seq=0 Ack=1
  249 562.365923364 192.168.242.128   TCP      34.117.188.166        54 55014 → 443 [ACK] Seq=1 Ack=1 Win=
  250 562.371026051 192.168.242.128   TLSv1.3  34.117.188.166       571 Client Hello (SNI=contile.services
  251 562.371420041 34.117.188.166    TCP      192.168.242.128       60 443 → 55014 [ACK] Seq=1 Ack=518 Wi
  252 562.414441447 34.117.188.166    TLSv1.3  192.168.242.128     3123 Server Hello, Change Cipher Spec,
  253 562.414491562 192.168.242.128   TCP      34.117.188.166        54 55014 → 443 [ACK] Seq=518 Ack=3070
  259 563.000216620 192.168.242.128   TCP      23.58.120.18          74 41268 → 80 [SYN] Seq=0 Win=32120 L
  260 563.018276771 23.58.120.18      TCP      192.168.242.128       60 80 → 41268 [SYN, ACK] Seq=0 Ack=1
  261 563.018318263 192.168.242.128   TCP      23.58.120.18          54 41268 → 80 [ACK] Seq=1 Ack=1 Win=3
  262 563.018722884 192.168.242.128   OCSP     23.58.120.18         470 Request
  263 563.018948453 23.58.120.18      TCP      192.168.242.128       60 80 → 41268 [ACK] Seq=1 Ack=417 Win
  264 563.058504555 23.58.120.18      OCSP     192.168.242.128      944 Response
```

```
▸ Frame 247: 74 bytes on wire (592 bits), 74 bytes captur
▸ Ethernet II, Src: VMware_9c:f0:1e (00:0c:29:9c:f0:1e), I
▸ Internet Protocol Version 4, Src: 192.168.242.128, Dst:
▸ Transmission Control Protocol, Src Port: 55014, Dst Por
```

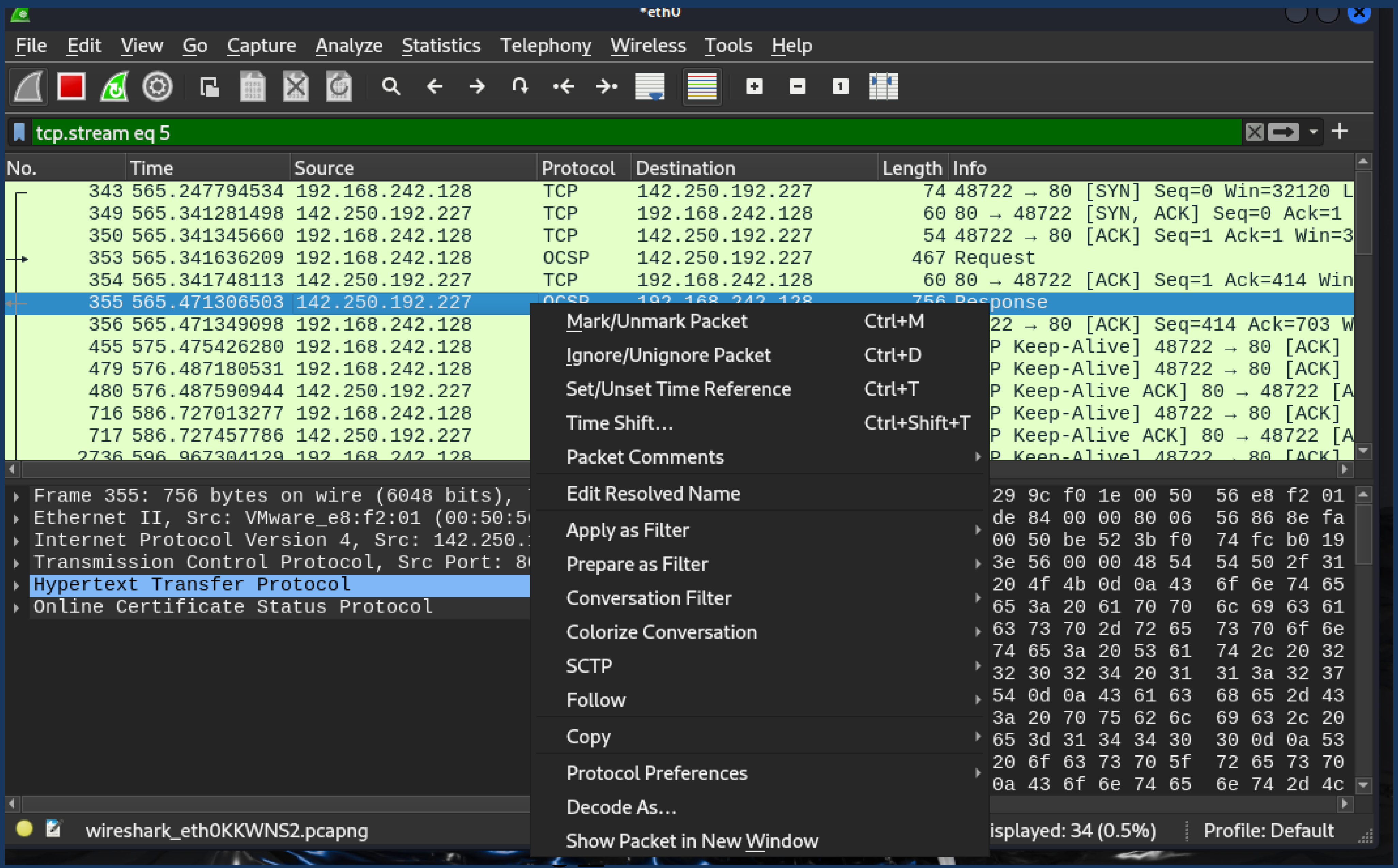
```
0000  00 50 56 e8 f2 01 00 0c  29 9c f0 1e 08 00 45 00
0010  00 3c 42 2c 40 00 40 06  66 4b c0 a8 f2 80 22 75
0020  bc a6 d6 e6 01 bb e7 37  be b1 00 00 00 00 a0 02
0030  7d 78 92 73 00 00 02 04  05 b4 04 02 08 0a a9 ba
0040  45 9a 00 00 00 00 01 03  03 07
```


tcp.port == 443


http

Follow TCP Stream

Objective: Demonstrate how to track a conversation between two endpoints.



*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
Show Packet in New Window

```
tcp.stream eq 5
```

```
No.   Time           Source           Protocol  Destination       Length Info
  343 565.247794534 192.168.242.128  TCP       142.250.192.227     74 48722 → 80 [SYN] Seq=0 Win=32120 L
  349 565.341281498 142.250.192.227  TCP       192.168.242.128     60 80 → 48722 [SYN, ACK] Seq=0 Ack=1
  350 565.341345660 192.168.242.128  TCP       142.250.192.227     54 48722 → 80 [ACK] Seq=1 Ack=1 Win=3
  353 565.341636209 192.168.242.128  OCSP      142.250.192.227    467 Request
  354 565.341748113 142.250.192.227  TCP       192.168.242.128     60 80 → 48722 [ACK] Seq=1 Ack=414 Win
  355 565.471306503 142.250.192.227  OCSP      192.168.242.128    756 Response
  356 565.471349098 192.168.242.128                                  22 → 80 [ACK] Seq=414 Ack=703 W
  455 575.475426280 192.168.242.128                                  P Keep-Alive] 48722 → 80 [ACK]
  479 576.487180531 192.168.242.128                                  P Keep-Alive] 48722 → 80 [ACK]
  480 576.487590944 142.250.192.227                                  P Keep-Alive ACK] 80 → 48722 [A
  716 586.727013277 192.168.242.128                                  P Keep-Alive] 48722 → 80 [ACK]
  717 586.727457786 142.250.192.227                                  P Keep-Alive ACK] 80 → 48722 [A
 2736 596.967304129 192.168.242.128                                  P Keep-Alive] 48722 → 80 [ACK]
```

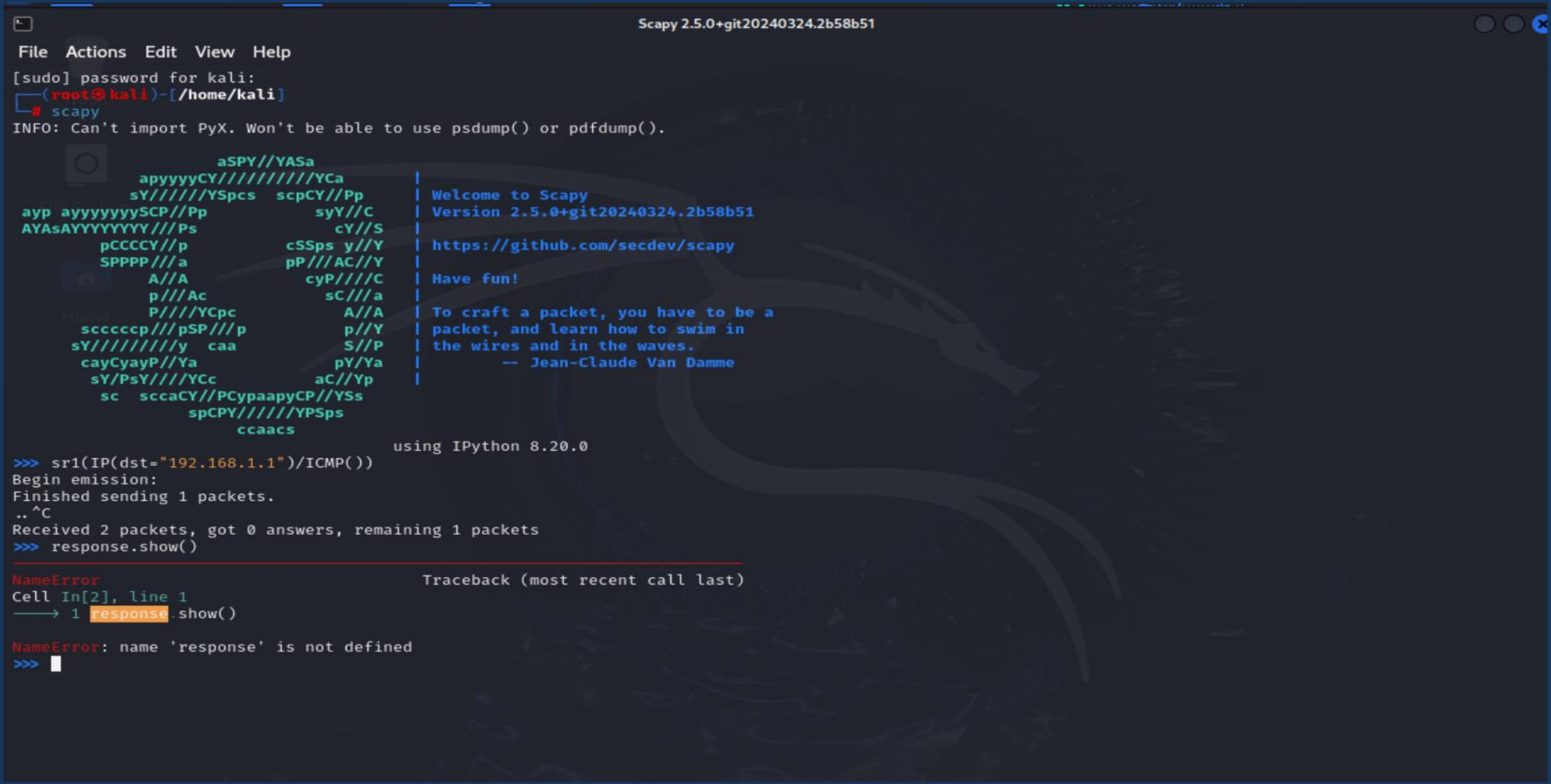
```
Frame 355: 756 bytes on wire (6048 bits),
Ethernet II, Src: VMware_e8:f2:01 (00:50:5
Internet Protocol Version 4, Src: 142.250.
Transmission Control Protocol, Src Port: 8
Hypertext Transfer Protocol
Online Certificate Status Protocol
```

```
29 9c f0 1e 00 50  56 e8 f2 01
de 84 00 00 80 06  56 86 8e fa
00 50 be 52 3b f0  74 fc b0 19
3e 56 00 00 48 54  54 50 2f 31
20 4f 4b 0d 0a 43  6f 6e 74 65
65 3a 20 61 70 70  6c 69 63 61
63 73 70 2d 72 65  73 70 6f 6e
74 65 3a 20 53 61  74 2c 20 32
32 30 32 34 20 31  31 3a 32 37
54 0d 0a 43 61 63  68 65 2d 43
3a 20 70 75 62 6c  69 63 2c 20
65 3d 31 34 34 30  30 0d 0a 53
20 6f 63 73 70 5f  72 65 73 70
0a 43 6f 6e 74 65  6e 74 2d 4c
```

wireshark_eth0KKWNS2.pcapng   isplayed: 34 (0.5%)   Profile: Default

Packet Spoofing Tool

Scapy



```
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().

                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.5.0+git20240324.2b58b51
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | To craft a packet, you have to be a
       scccccp///pSP///p          p//Y   | packet, and learn how to swim in
      sY/////////y  caa           S//P   | the wires and in the waves.
       cayCyayP//Ya              pY/Ya   |        -- Jean-Claude Van Damme
        sY/PsY////YCc          aC//Yp    |
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs
                                       using IPython 8.20.0
>>> sr1(IP(dst="192.168.1.1")/ICMP())
Begin emission:
Finished sending 1 packets.
..^C
Received 2 packets, got 0 answers, remaining 1 packets
>>> response.show()
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 response.show()

NameError: name 'response' is not defined
>>> 
```

Research scope

a. **AI-Driven Spoof Detection**

Develop machine learning models to detect spoofing attacks (e.g., IP spoofing, email spoofing) by analyzing network traffic patterns or metadata.

Explore reinforcement learning for adaptive detection of evolving spoofing techniques.

b. **Quantum Cryptography for Spoof Protection**

Investigate how quantum cryptographic methods like Quantum Key Distribution (QKD) can mitigate spoofing in critical communication systems.

c. **Blockchain-Based Anti-Spoofing**

Design decentralized systems using blockchain technology to verify the authenticity of communications, reducing the risk of DNS or IP spoofing.

d. IoT Device Spoofing Detection

Create lightweight spoofing detection algorithms for IoT devices with constrained resources.

Study spoofing attacks in IoT ecosystems (e.g., smart homes) and propose novel mitigation strategies.

Research scope

a. **Wireless Sniffing in 6G Networks**

Study the impact of sniffing attacks in emerging 6G wireless networks and propose advanced intrusion detection mechanisms.

Explore the use of secure millimeter-wave communication channels to prevent sniffing.

b. **Encrypted Traffic Sniffing**

Research methods to detect or identify malicious sniffing even in encrypted traffic using side-channel analysis or machine learning.

c. **Cyber-Physical Systems and Sniffing**

Investigate the impact of sniffing attacks on industrial control systems (ICS) and propose secure communication protocols.

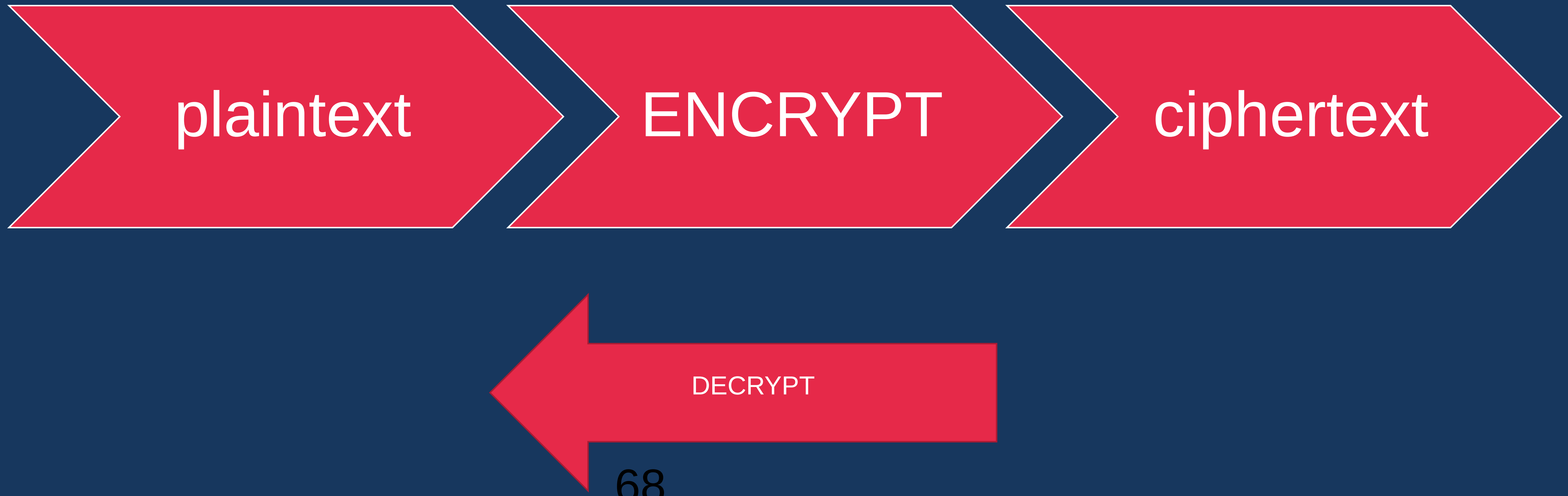
d. **Honeypots to Identify Sniffers**

Design honeypots that deliberately leak fake information to identify and neutralize sniffing tools on a network.

Cryptographic Services

Cryptography

Cryptography involves encrypting or decrypting a piece of data.



plaintext

ENCRYPT

ciphertext

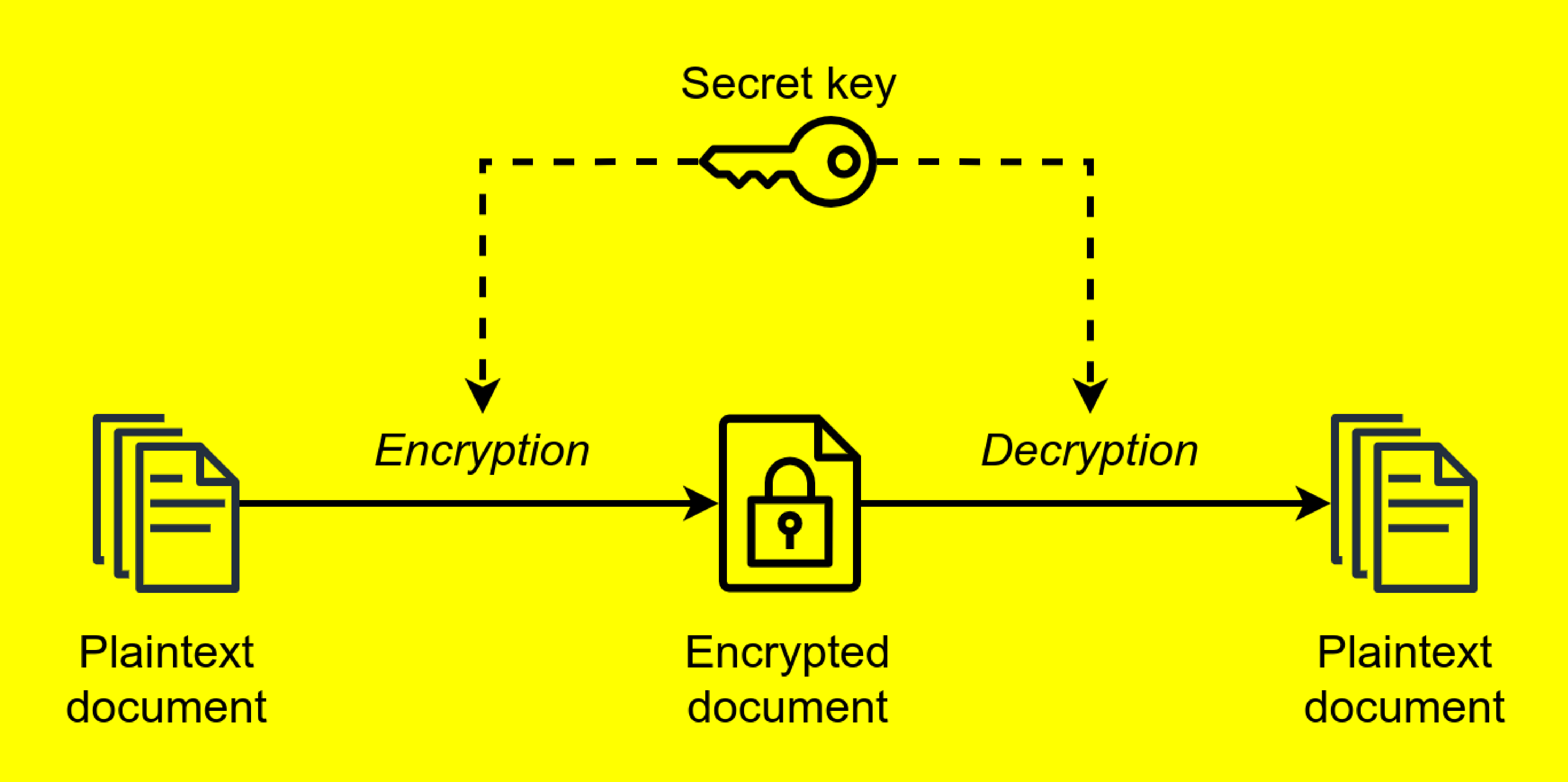
DECRYPT

Cryptography Types

- ✓ Symmetric Key Cryptography
- ✓ Asymmetric Key Cryptography
- ✓ Hash Functions
- ✓ Digital Signatures
- ✓ Cryptographic Hash Functions
- ✓ Block Ciphers
- ✓ Stream Ciphers
- ✓ Homomorphic Encryption

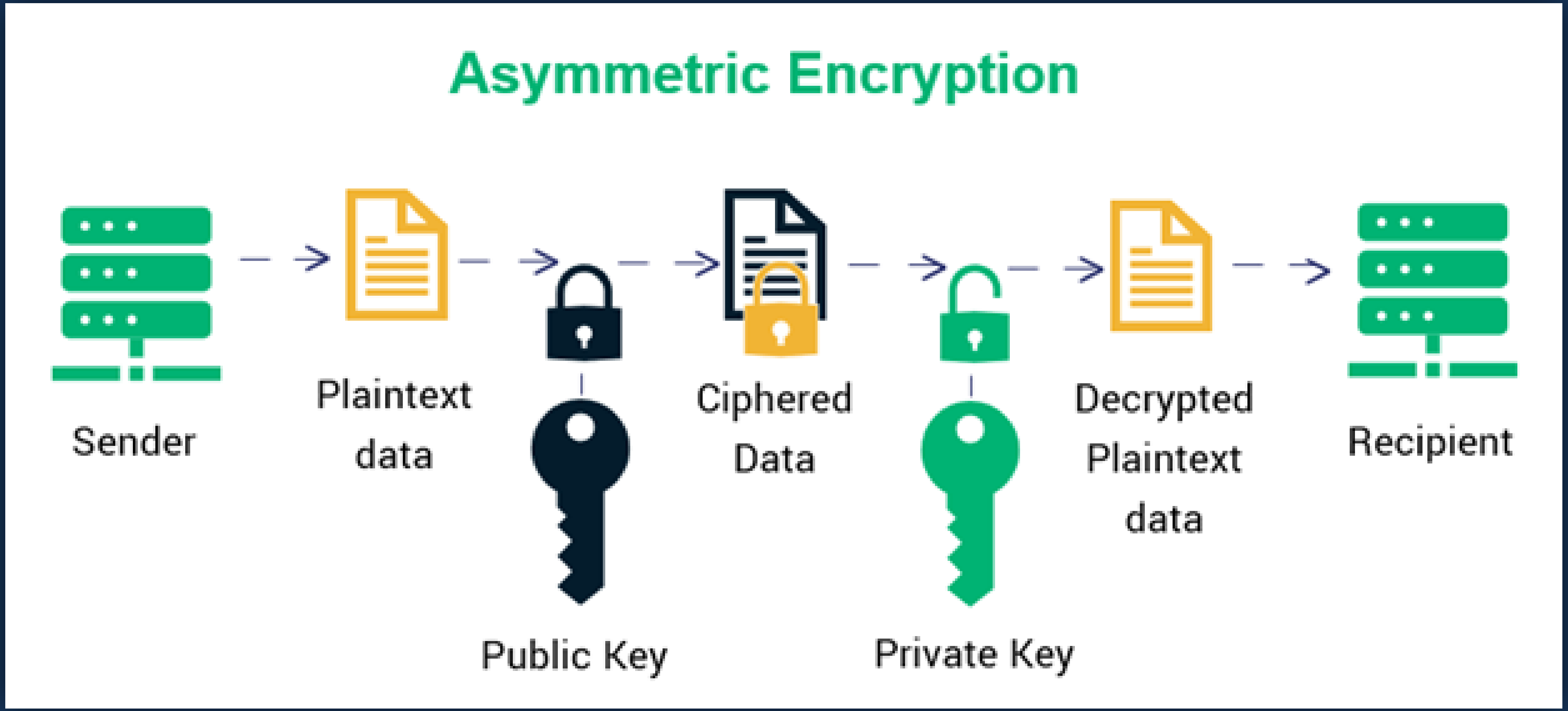
Symmetric Key Cryptography

- ✓ DES (Data Encryption Standard)
- ✓ AES (Advanced Encryption Standard)
- ✓ Blowfish



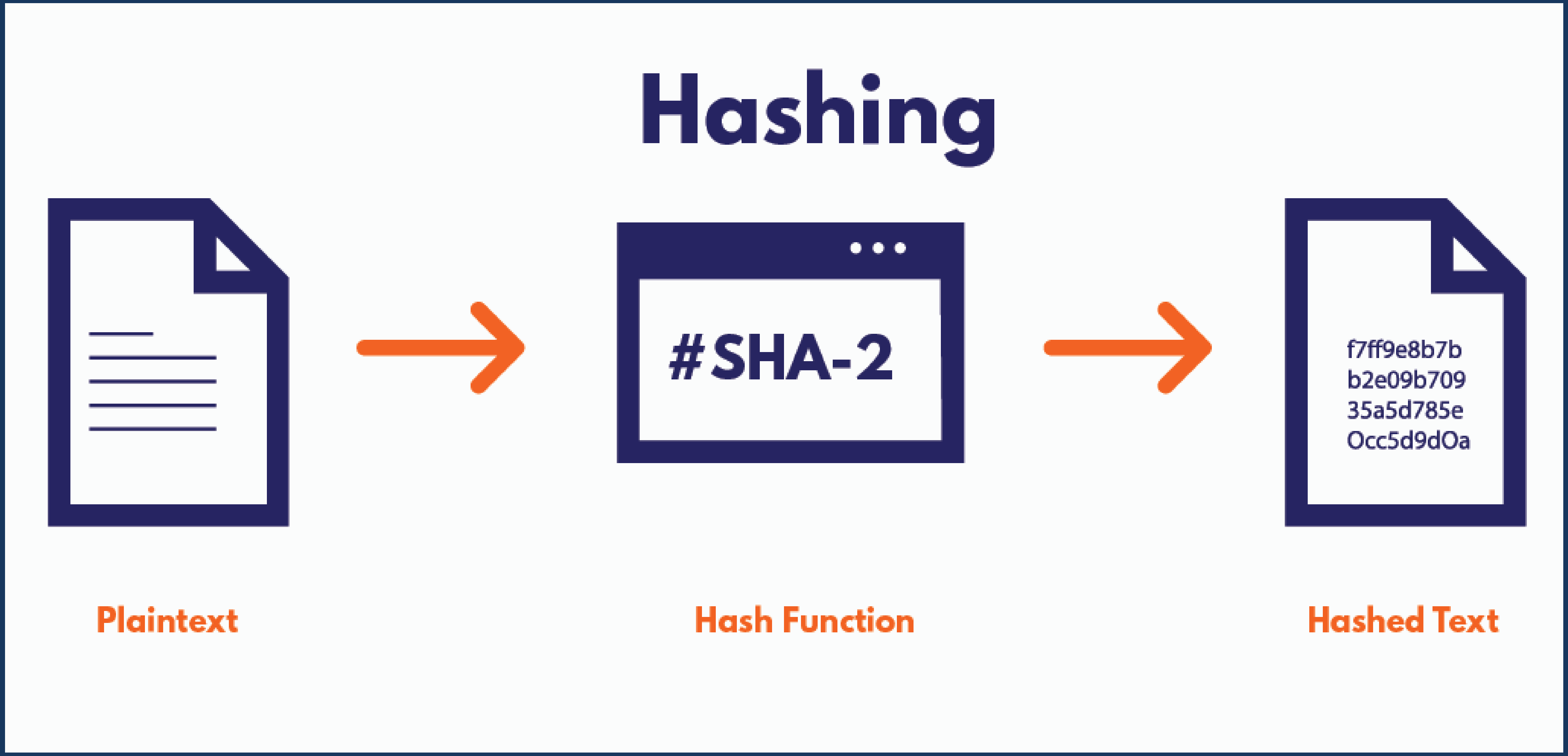
Asymmetric Key Cryptography

- ✓ RSA (Rivest-Shamir-Adleman)
- ✓ ECC (Elliptic Curve Cryptography)



Hash Functions

- ✓ SHA-256 (Secure Hash Algorithm 256-bit)
- ✓ MD5 (Message Digest Algorithm 5)



Cryptography

Encryption/decryption tools and libraries such as openssl.

Password cracking tools like John the Ripper and hashcat.

Encoding/decoding and analysis tool like CyberChef, dcode, cryptii etc

Cryptography Tools

CyberChef A web application that provides a suite of tools for data analysis and manipulation. It can be used for encryption, decryption, and many other purposes.

FeatherDuster A tool that can identify and exploit weaknesses in cryptographic implementations.

Hash Extender A tool for extending hash length attacks.

Cryptography Tools

padding-oracle-attacker A tool for attacking padding oracle vulnerabilities in web applications.

PkCrack A tool for breaking PkZip encryption.

RSACTFTool A tool for attacking RSA encryption.

RSATool A tool for recovering the RSA private key from a given public key.

Cryptography Tools

XORTool A tool for performing XOR encryption and decryption.

Cryptii A web application that provides a suite of tools for encryption, decryption, and encoding.

Keyboard Shift A tool for performing keyboard shift ciphers.

Cryptography Links

https://github.com/alinboby/CTF-Learn-HxN0n3/blob/main/Cryptography.md

Steganography Types

- ✓ **Image Steganography**
- ✓ **Audio Steganography**
- ✓ **Video Steganography**
- ✓ **Text Steganography**
- ✓ **Network Steganography**

Steganography Types

- ✓ OpenStego
- ✓ Steghide
- ✓ OutGuess
- ✓ SilentEye
- ✓ QuickStego

Learn CTF

CTF

Capture The Flag

Test participants' skills in **various aspects of cybersecurity.**

CTF Platforms

CTFtime

picoCTF

HackTheBox (HTB)

TryHackMe (THM)

CTFLearn

Crackmes

CyberTalents

Cybher

CyberEdu

CTF Types

Jeopardy-style CTF: challenges are divided into different categories

Attack-Defense CTF: teams compete against each other in a simulated network environment

King of the Hill (KOTH) CTF: teams compete to maintain control of a specific server or service (the "hill")

Mixed or Hybrid CTF

CTF Challenges Categories

Cryptography

Forensic

Web

Binary Exploitation

Reverse Engineering

PWN

OSINT

Networking

Steganography

Misc

CTF Challenges Categories (misc)

Mobile / Android

Programming

Blockchain

Boot2Root

ICS

Game-based

CTF youtube channel

JohnHammond

LiveOverflow

SloppyJoePirates

HxN0n3

Geekingjadi

carlislemc

Networking Challenges

Networking challenges can be quite varied, involving different types of tasks that test participants' understanding of network protocols, packet analysis, and security.

Packet Analysis

Examining and interpreting data packets (from packet captures PCAP files) transmitted over a network

- Identifying patterns
- Extracting hidden information
- Reconstructing sessions

Packet Analysis

Tools

Wireshark: A powerful network protocol analyzer.

tcpdump: A command-line packet analyzer.

NetworkMiner: A network forensics analysis tool.

Scapy: A Python program that enables packet manipulation.

Packet Analysis

Examples

https://www.youtube.com/watch?v=H9gzRyEEbzE
https://www.youtube.com/watch?v=cScoRiGlSUo&t=75s
https://www.youtube.com/watch?v=11SmaJ7oXvs
https://www.youtube.com/watch?v=2hM7ImYX_Bs
https://www.youtube.com/watch?v=NwyjAT4TPPg&list=PLxYdTW0sJWBDZ29Jrgh7CjldegKc0bVao

Network Traffic Analysis

Studying the flow of packets across a network

- Understand communication patterns
- Detect anomalies
- Uncover hidden data

Network Traffic Analysis

Tools

Bro/Zeek: A powerful network analysis framework.

Splunk: A tool for searching, monitoring, and analyzing machine-generated data.

ELK Stack (Elasticsearch, Logstash, Kibana): For centralized logging and analysis.

Network Scanning

Discovering devices, open ports, and services on a network

Tools

Nmap: A network mapping and vulnerability scanning tool.

Masscan: A fast port scanner.

Netcat (nc): A versatile networking tool.

Exploitation

Finding and exploiting vulnerabilities in network services.

Tools

Metasploit: A penetration testing framework.

ExploitDB: A repository of exploits and proof-of-concepts.

Burp Suite: A web vulnerability scanner and proxy tool.

Protocol Analysis

Examining and understanding specific network protocols, often to identify misconfigurations or vulnerabilities

Tools

Wireshark: (mentioned above)

Ettercap: A comprehensive suite for man-in-the-middle

attacks on LAN.

Steganography in Network Traffic

Hiding data within network traffic, such as in image files, protocols, or other forms of communication

Tools

Stegsolve: A tool for analyzing images for hidden information.

OpenStego: A steganography tool.

Firewall and IDS/IPS Evasion

Bypassing security mechanisms like firewalls and intrusion detection/prevention systems

Tools

hping: A network tool for packet crafting.

Nmap: (mentioned above)

DNS Analysis

Investigating domain name system queries and responses, which can reveal interesting information about a network's structure and activity

Tools

dnsenum: A tool for enumerating DNS information.

dnsrecon: Another tool for DNS enumeration.

Networking tools and links

1. Wireshark - Network protocol analyzer useful for network forensics and traffic analysis (https://www.wireshark.org)

2. NetworkMiner - Open source network forensic analyzer useful for investigating traffic

3. Snort - Open source intrusion detection and network monitoring system (https://www.snort.org)

4. Tcpdump - Capture and analyze network traffic on Unix-like systems (https://www.tcpdump.org)

Networking tools and links

5. Ngrep - Search within network traffic payloads like grep for text streams (http://ngrep.sourceforge.net/)

6. Hunchback - High speed packet capture and transmission tool (https://hunchback.sourceforge.net/)

7. AIL - Network and host monitoring system for identification of intrusions (https://www.cert.org/incident-management/products-services/ail.cfm)

THANK YOU