Project Title:

Incident Response – Detect, Analyze & Mitigate

P. Rakesh U.Shashidhar

cyber security cyber security

TAH2706251671 TRAINER

TEKS ACADEMY 96033 43287

9515924411

**Abstract:**

The project displays the process of detecting, analysing and reducing a common network attack known as ARP spoofing. Using the attacker, an afflicted virtual machine, and a controlled lab setup with black Linux as an entrance, we simulated the "Man-in-Middle" attack to understand how the dangers of the real-world network operate.

We began by identifying all the equipment on the local network using the Net Discover Tool; the attackers, the victims, and the Gateway's IP and MAC address were confirmed. The attack was launched using ARP Spoof, where the attacker imposed the victim and contrary to the gateway. This allowed the attacker to disrupt network traffic between them.

During the attack, the viresque was used to catch and analyse the packets during the attack. By applying the ARP filter, we saw the Sports ARP answers showing the Mac address of the attacker in place of the gateway. Applying HTTP filters and TCP currents revealed how uncontacted credentials and browsing activity can be captured.

Finally, a mitigation plan was developed, including ARP-A, monitoring with Arp watch, static ARP entries, HTTPS uses, VPN encryption, and infiltration detection systems such as Snort or Sucuri measures, such as prevention of prevention systems.

Practice with this hand provided practical experience in both aggressive and defensive cybersecurity techniques, not only attacks, but also showed how to identify and prevent them before they cause damage.

**Problem Statement:**

In today's connected world, cyberattacks are becoming more common and tricky, which is an issue for people, businesses, and even governments. One attack is ARP spoofing, which involves a hacker fooling devices on a network by sending fake ARP messages. This makes devices believe the hacker's system is the correct destination for their data. Then, the hacker can secretly grab, change, or stop communication, leading to stolen info, messed-up services, and loss of privacy.

Sadly, many small and medium-sized networks lack effective monitoring and plans for addressing problems, making them vulnerable to exploitation. That’s why knowing about ARP spoofing, spotting it quickly, and using ways to fight back is super important.

In this project, we're faking an ARP spoofing attack in a safe, controlled space, using programs like Wireshark to catch and check weird network stuff, and ARP spoofing to perform the attack. Lastly, we'll show some simple ways to stop and lessen these attacks, helping networks stay safe from this regular, but bad, cyber issue.

**Project Objective(s):**

* We will set up and run an ARP spoofing attack in a safe lab environment, using Kali Linux as the attacker and a chosen target system.
* During the attack, we’ll monitor the entire network traffic in real time with Wireshark, allowing us to inspect packets from start to finish.
* We’ll then look for clear signs of ARP spoofing, such as mismatched MAC addresses or unusual ARP replies, to confirm the attack is taking place.
* Next, we’ll study how the attack impacts network communication and data integrity, identifying the risks it creates.
* Once analysed, we’ll apply protective measures like setting static ARP entries or using intrusion detection tools to block similar threats in the future.
* Finally, we’ll document the whole process with clear screenshots, findings, and recommendations, creating a complete record for reporting and prevention.

**Project Scope:**

* Simulate ARP spoofing in a secure lab setup.
* Target a system using Kali Linux.
* Monitor traffic using Wireshark.
* Detect suspicious ARP packets.
* Analyze network disruption effects.
* Apply mitigation techniques.
* Document findings with visuals.
* Suggest preventive measures.

**Tools & Technologies:**

* Kali Linux – For simulating the ARP spoofing attack.
* Arp spoof – Command-line tool to perform ARP spoofing.
* Wireshark – To capture and analyse network traffic.
* Virtual Machines – To create a safe, isolated lab environment.
* VMware / VirtualBox – For running multiple OS instances.
* Linux Terminal – For executing attack and analysis commands.
* Static ARP Configuration – Used as a mitigation technique.
* Intrusion Detection Tools – For detecting ARP spoofing attempts.

**Timeline (Tentative):**

Week Task

Day 1: Set up Kali (attacker), victim machine, and Wireshark.

Day 2 Learn ARP spoofing basics & impacts.

Day 3 Perform ARP spoofing and capture traffic.

Day 4 Analyze traffic, test mitigations (static ARP, Arp watch).

Day 5 Document process, prepare demo & recommendations

**Deliverables:**

Detection log or screenshot (e.g., Arp -a, Wireshark)

Short analysis summary (what happened, indicators, affected services)

Mitigation process taken (e.g., stopped services, blocked attacker, static ARP)

Log or output of mitigation steps taken

Hardening resources deployed (e.g., cron monitoring, alerts, static binds)

Screenshots or terminal outputs as evidence

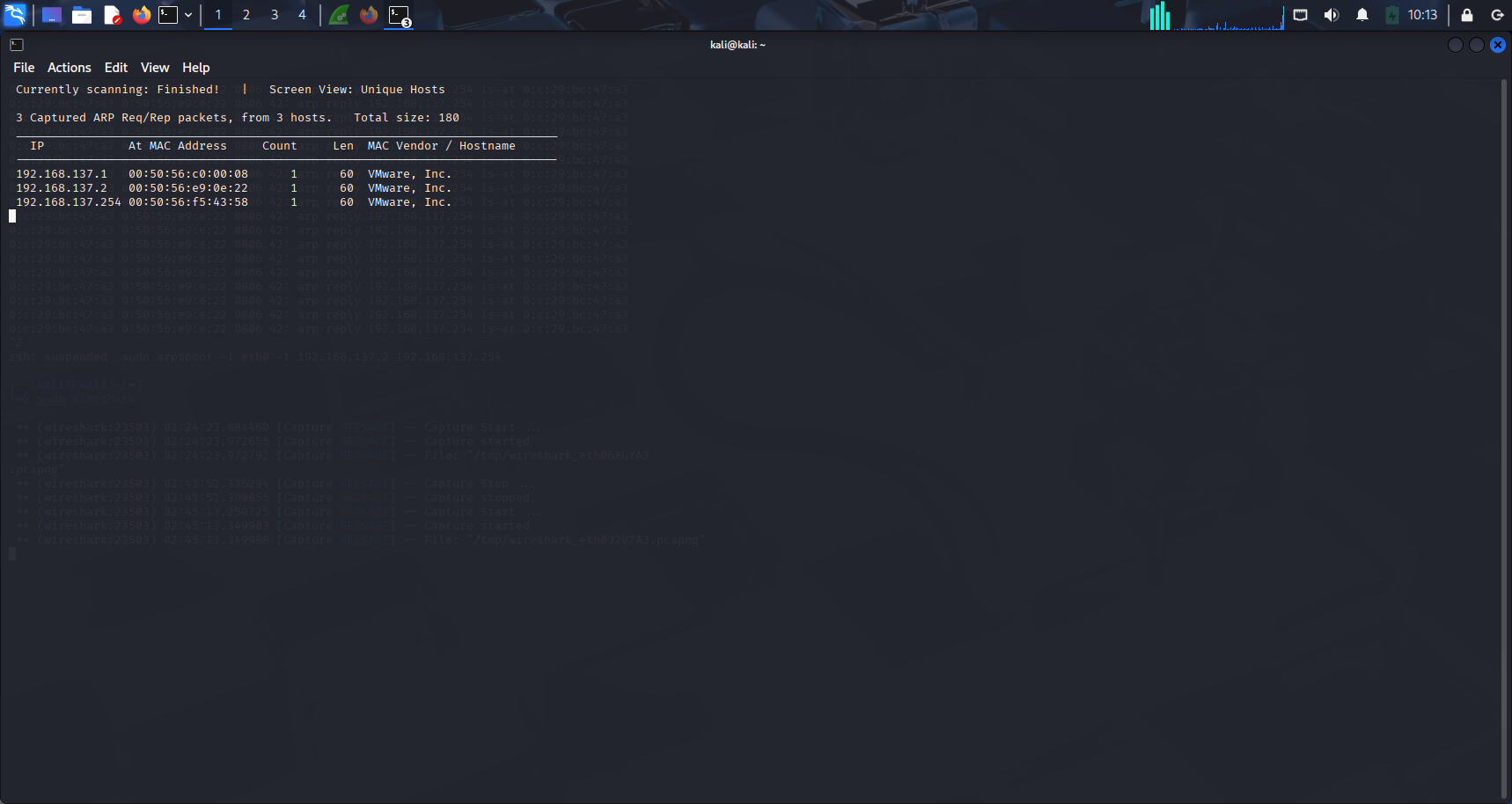
Concise mini-report (1-2 pages)

Demo video, if possible, showing steps

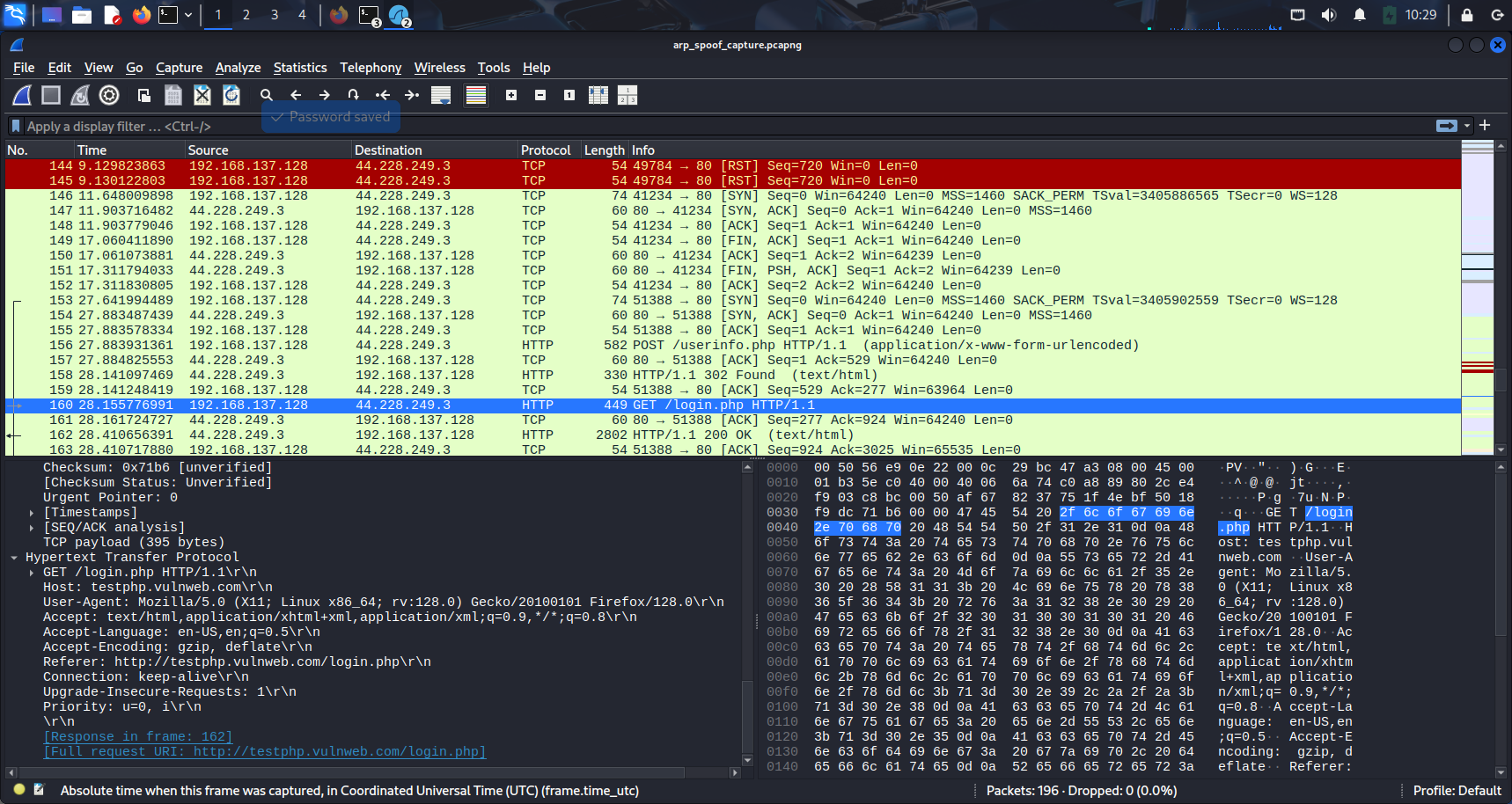
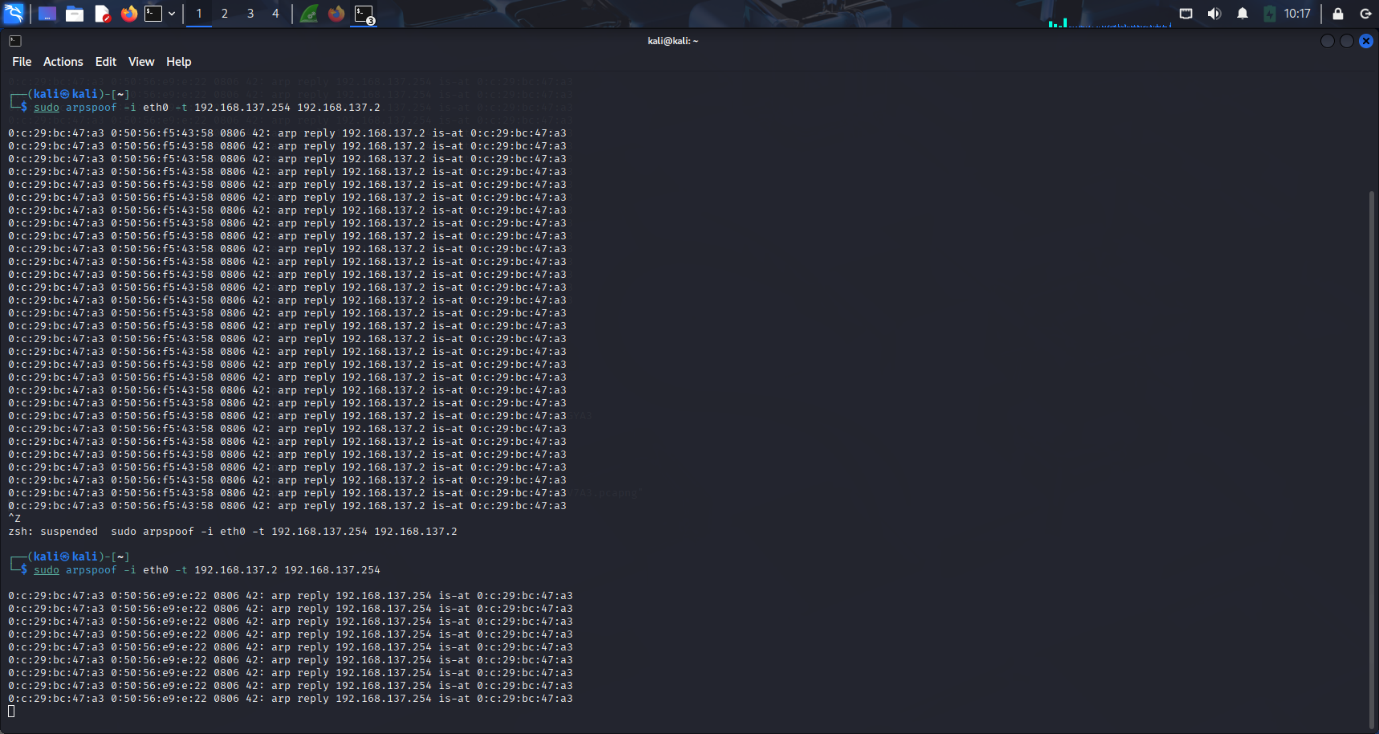
Organized project folder with all files

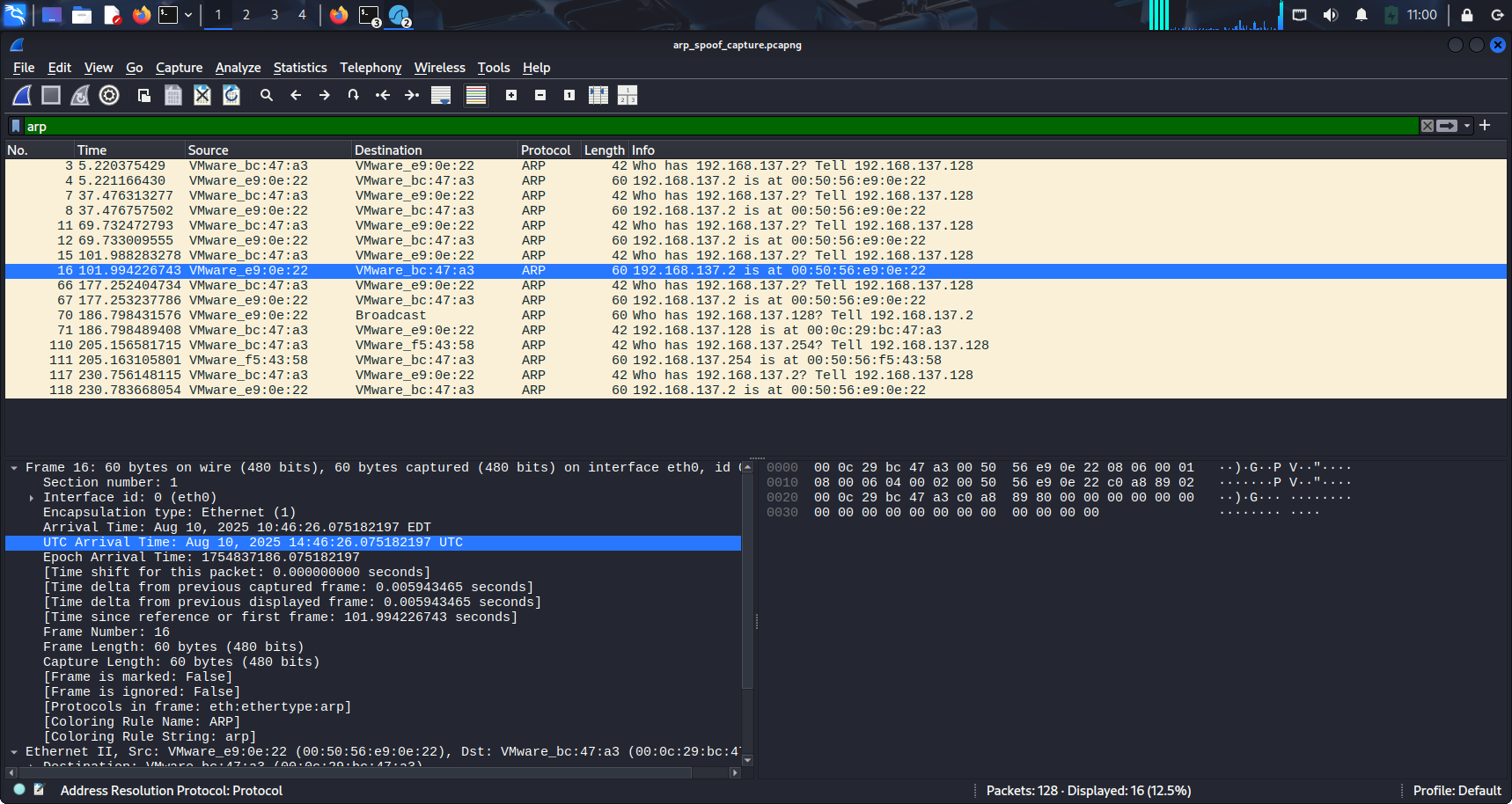
Project implementation:

Step 1: Discover Devices



Step 2: Start ARP Spoofing

Step:3  Capture Network Traffic with Wireshark

Step 4: Analyze the Capture 

Step 5: Analyze HTTP Traffic

