Project title: Network Attacks Detection and Prevention System Using Raspberry Pi

Project Report

Over the past few days (Late May 1404), I worked as an intern on a network security project alongside Mr. Mahdi Gorzedin (the project lead and a cybersecurity expert) and our friend Mr. Sajad Karimi. This was my very first hands-on experience in this field, and I learned many new and interesting things.

Project Topic

Our project was focused on developing a tool for detecting and preventing network attacks. The tool was written in Python and is capable of inspecting the packets exchanged within a network. If it detects an ongoing attack, it identifies the threat and prevents it from continuing.

We worked on a system related to network security that falls under the following categories:

- ❖ IDS (Intrusion Detection System): A system that detects attacks and raises alerts.
- ❖ IPS (Intrusion Prevention System): A system that not only detects attacks but also actively prevents them (e.g., by blocking IP addresses).

Hardware Used

The core of our system was deployed on a Raspberry Pi, a small and power-efficient computer capable of performing many tasks. We connected the Raspberry Pi to a network so it could act as a gateway, allowing it to monitor packets and detect potential attacks.

In addition, we used three laptops for executing scripts, testing attacks, and observing the results. These included:

- a Regular user laptop,
- an Attacker user laptop,
- and a user connected to the Raspberry Pi system.

These devices helped us generate and monitor various types of packets within the network to analyze how the system would respond under different conditions.

The types of attacks we examined included:

DoS Attack – An attack that overwhelms the system with excessive requests, causing it to crash or become unresponsive.

Port Scanning – A technique used to identify open ports on a target system, which could be potential entry points for exploitation.

SSH Brute-force – An attack where the attacker repeatedly tries different password combinations to gain unauthorized access to an SSH service.

ARP Spoofing – An attack in which the attacker sends forged ARP responses to the network, impersonating a trusted device. As a result, victim devices mistakenly send their traffic to the attacker, enabling packet interception or manipulation.

How the System Prevents Attacks

Imagine someone is sending a flood of requests from another device in the network (for example, to perform a DoS attack). Our system acts like a guard at a gate, monitoring incoming packets. If it detects that a specific IP is sending a suspiciously high number of requests, it takes the following actions:

- Generates a warning by logging the event.
- Sends an email alert to the system administrator, e.g., "A DoS attack is being attempted from IP xxx.xxx.xxx.xxx!"
- Blocks the malicious IP using a Linux tool called iptables, preventing further traffic from that source.

After 10 minute (this duration is customizable), if the IP no longer shows suspicious behavior, it is unblocked automatically.

My Role in the Project

Although I didn't contribute directly to the core coding part (as Mahdi and the team were moving very fast due to time constraints, and I lacked the required coding experience), I was actively involved in other crucial aspects:

1. Assisting in project execution and operational tasks.

2. Debugging support – When the code failed, we would clear it, debug, reload it into the terminal, and run it again.

3. Working with terminal commands, such as:

- Connecting to the Raspberry Pi via SSH: ssh pi@raspberrypi.local
- Deleting files: rm file.py
- Editing files: nano file.py

- Running the script: sudo python3 file.py
- Helping during test attacks, such as resetting the script, unblocking IPs, or re-executing commands.

What I Learned

I became familiar with several key networking and security concepts:

- Packets: Small units of data transmitted across a network.
- iptables: A powerful Linux firewall tool we used to block malicious IPs.
- Sniffing: Monitoring and analyzing network traffic.
- Ports: Communication endpoints (e.g., port 22 is used for SSH, which we used extensively).

Most importantly, I learned how Python can be used to build a system that detects attacks, sends alerts, writes logs, and even prevents threats automatically.

Challenges & Fun Moments

Besides the learning experience, we had some funny moments too. For instance, once Mahdi accidentally plugged the VGA cable into the PC case instead of the monitor. We all thought the Raspberry Pi was dead! Everyone kept saying, “What if it really burned out?” – but it turned out to be a simple mistake. We had a good laugh, and the project never felt boring or too serious. Also we borrowed the Raspberry Pi from the university with prior coordination, which gave us a greater sense of responsibility. Since only a limited number of devices were available, we worked with extra care and motivation to achieve meaningful results and make the most of the opportunity we were given.

Conclusion

This project gave me a real-world introduction to network security. I realized that even with a small device like a Raspberry Pi, it’s possible to detect and stop network attacks. It was my first experience working on a cybersecurity project, and I genuinely found it exciting. I’m now even more interested in the field and eager to participate in similar projects in the future.

So, if someone asks what our project was about, I can confidently say:

“We built a system that monitors network traffic, detects attacks, send alerts and takes action to prevent them.”