CYBERSECURITY DAILY DAIRY

Day 10: Firewall Implementation and SSH/FTP Control (June 28, 2025)

Topics Covered:

- Firewall implementation with Python.
- Managing open ports.
- SSH (Secure Shell) access management.
- FTP (File Transfer Protocol) access management.
- Identifying and closing unnecessary open ports.

What I Did:

I built a **firewall** using a Python tool to manage port traffic and define allow/deny rules. I also worked with **SSH** and **FTP** and learned to detect and secure open ports.

Firewall Implementation:

- Python script for a lightweight firewall.
- Monitored incoming/outgoing packets.
- Configured **allow/deny rules** by port/IP.
- Blocked unauthorized traffic, kept logs.
- Blocked unused ports.

SSH and FTP Management:

- **SSH:** Used for secure remote access. Tested how firewall rules affect SSH (port 22).
- **FTP:** Tested for file transfer. Verified blocking port 21 (FTP) refused connection.

Detecting and Closing Open Ports:

- Used `netstat`, `ss`, `nmap` to scan for open ports.
- Identified unnecessary services.
- Updated firewall to block vulnerable ports.
- Verified changes with scans.

Key Learnings:

- Firewalls **control access** and **block malicious traffic**.
- Python can make simple custom firewalls.
- SSH and FTP are common attack points and need tight control.
- Regular port scanning improves security.
- Logs help track intrusions.