Julius Forlefac

CIS 260 XX OH

**22Project: Building a Basic Firewall**

**Introduction**

Firewalls are a critical component in network security, acting as a barrier between trusted internal networks and untrusted external networks like the internet. Building a firewall from scratch will allow me to understand the underlying concepts of packet filtering, stateful inspection, and network traffic monitoring.

In this project, I will be creating a basic firewall. The firewall will inspect incoming and outgoing network traffic based on a set of user-defined rules.

**Objectives:**

* Learn the basic concepts of firewalls.
* Build a simple packet filtering system.
* Implement rule-based access control.
* Gain practical knowledge of network security tools like iptables and Python’s socket library.

**Tools and Technologies:**

1. **Operating System**: Linux (Ubuntu/Debian preferred)
2. **Programming Language**: Python
3. **Packet Filtering**: iptables (Linux command line)
4. **Networking Libraries**: Python's socket, subprocess
5. **Firewall Logging**: System log files **(e.g., /var/log/ufw.log)**

**Prerequisites:**

* Basic understanding of networking (IP, TCP, UDP).
* Familiarity with Python.
* Basic knowledge of Linux and its terminal.
* Root or Sudo privileges for running network-related commands like iptables.

**Steps to Build the Firewall:**

**Step 1: Setting Up the Development Environment**

Ensure that your environment has the necessary tools.

1. **Install Python** (if not installed already)

sudo apt update

sudo apt install python3 python3-pip

1. **Install iptables** (usually pre-installed on most Linux distributions)

bash

sudo apt install iptables

1. **Ensure your** **firewall is not blocking essential ports** during development:

bash

sudo ufw allow ssh

sudo ufw enable

1. **Create a folder** for the project:

bash

mkdir firewall\_project

cd firewall\_project

**Step 2: Understanding iptables Basics**

Before jumping into the coding part, we must understand that iptables is the firewall tool used in Linux for packet filtering. It works by applying rules for incoming and outgoing packets. We can create a basic rule as follows:

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

This allows incoming traffic on port 22 (SSH). Similarly, you can block traffic using DROP:

sudo iptables -A INPUT -p tcp --dport 80 -j DROP

This drops incoming HTTP traffic.

**Step 3: Setting Up Basic Packet Filtering with iptables**

We can begin by manually configuring some basic firewall rules using iptables commands.

**Example Rules:**

1. **Default Policies**: Set default policies to DROP all incoming and outgoing traffic.

sudo iptables -P INPUT DROP

sudo iptables -P OUTPUT DROP

sudo iptables -P FORWARD DROP

1. **Allow Localhost Traffic**: This allows communication on the localhost interface.

sudo iptables -A INPUT -i lo -j ACCEPT

sudo iptables -A OUTPUT -o lo -j ACCEPT

1. **Allow Incoming SSH**: This allows SSH (port 22) from any source to the machine.

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

1. **Allow Outgoing HTTP**: Will allow the system to initiate outgoing HTTP requests.

sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT

1. **Save iptables Rules**: To ensure the rules persist after reboot, save them with the following command:

sudo iptables-save > /etc/iptables/rules.v4