

# **Python-Based Firewall – Project Documentation**

## Table of Contents

<b>Project Overview</b> .....	<b>3</b>
Description:.....	3
Objectives .....	3
<b>System Requirements</b> .....	<b>4</b>
<b>Design &amp; Architecture</b> .....	<b>5</b>
Workflow: .....	5
<b>Implementation Details</b> .....	<b>6</b>
Import & Configuration .....	6
Logging Function.....	6
Firewall Rules .....	6
Firewall Logic.....	6
Packets Processing and Capture .....	8
Summary & Alerts .....	8
<b>Sample Output</b> .....	<b>9</b>
<b>Security &amp; Limitations</b> .....	<b>10</b>
Security Notes: .....	10
Limitaions: .....	10
<b>Future Work</b> .....	<b>11</b>
<b>Conclusion</b> .....	<b>12</b>

# Project Overview

**Title: Python-Based Firewall**

**Developer: Aboubakr Gaber**

**Date: 6th Of November 2025**

## Description:

This project implements a basic Python-based firewall that monitors and filters live network using the Scapy library.

It allows and blocks packets based on defined IP and port rules, logs all actions, and triggers an alert if too many blocked packets are detected.

## Objectives

- Develop a lightweight, script-based firewall using python.
- Understand packet information using the Scapy library.
- Implement rule-based filtering for IPs and Ports.
- Maintain detailed logs of all network activities.
- Generate alerts when unusual blocking activity is detected.

# System Requirements

## **Hardware:**

- CPU: Dual-core processor or higher
- RAM: Min. 4GB
- Storage: >200MB

## **Software:**

- OS: Linux/Windows(Admins Prvs Required)
- Python Version: 3.8+

# Design & Architecture

Network Traffic → Scapy Packet Capture → Firewall Rules Engine →

|— Allowed → Pass

|— Blocked → Log + Alert

## Workflow:

- 1- Capture: Scapy continuously monitors live network packets.
- 2- Inspect: Each packet is analyzed for source IP, destination IP, protocol, and port.
- 3- Filter: The firewall checks if these values match the allowed or blocked rules
- 4- Log: All results are logged with timestamps.
- 5- Alert: If blocked packets exceed a threshold, a critical alert is logged.

# Implementation Details

## Import & Configuration

```
from scapy.all import sniff, IP, TCP, UDP
import datetime
- Scapy.all: Used for sniffing and analyzing network packets.
- Datetime: Generates timestamps for log entries.
```

## Logging Function

```
def log_message(message, log_type="INFO"):
    timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    log_entry = f"[{log_type}] {timestamp} - {message}"
    print(log_entry)
    with open(log_file_path, "a") as log_file:
        log_file.write(log_entry + "\n")
This function saves all the event outputs into firewall_log.txt.
```

## Firewall Rules

```
ALLOWED_IPS = ["192.168.1.1", "10.0.0.5", "192.168.100.4", "192.168.100.200"]
BLOCKED_IPS = ["192.168.1.100", "172.16.0.10"]
BLOCKED_PORTS = [22, 80]
ALLOWED_PORTS = [8080, 1234, 443]
ALERT_THRESHOLD = 5
```

Alert\_Threshold prints a warning when excessive blocks occur.

## Firewall Logic

```
def firewall(packet):
    global block_counter
    if IP in packet:
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        protocol = "TCP" if TCP in packet else "UDP" if UDP in packet else
    "Other"
        port = (
            packet[TCP].dport if TCP in packet else packet[UDP].dport if UDP in
    packet else None
        )
```

```

# Check if the source IP is blocked
if src_ip in BLOCKED_IPS:
    block_counter += 1
    log_message(f"Blocked: Malicious SRC IP {src_ip}", "WARNING")
    return f"Blocked: Malicious SRC IP {src_ip}"

# Check if the source IP is allowed
if src_ip not in ALLOWED_IPS:
    log_message(f"Blocked: SRC IP {src_ip} not in allowed list",
"WARNING")
    block_counter += 1
    return f"Blocked: SRC IP {src_ip} not allowed"

# Check if the port is blocked
if port and port in BLOCKED_PORTS:
    log_message(f"Blocked: Port {port} is restricted", "WARNING")
    block_counter += 1
    return f"Blocked: Port {port} is restricted"

# Only allow specific ports
if port and port not in ALLOWED_PORTS:
    log_message(f"Blocked: Port {port} not in allowed list", "WARNING")
    block_counter += 1
    return f"Blocked: Port {port} not allowed"

log_message(f"Allowed: Packet from {src_ip} to {dst_ip} on port {port}",
"INFO")
return f"Allowed: Packet from {src_ip} to {dst_ip} on port {port}"

log_message("Blocked: Non-IP packet", "WARNING")
block_counter += 1
return "Blocked: Non-IP packet"

```

- Check for IP layer: Non-IP packets are blocked.
- Block malicious IPs: Matches from `BLOCKED_IPS`.
- Allow trusted IPs: Only packets from `ALLOWED_IPS` are permitted.
- Block restricted ports: Matches from `BLOCKED_PORTS`.
- Allow selected ports: Matches from `ALLOWED_PORTS`.
- Log results: for all actions.

## Packets Processing and Capture

```
def process_packet(packet):
    result = firewall(packet)
    if "Blocked" in result:
        print(f"{result}")
    elif "Allowed" in result:
        print(f"{result}")

sniff(filter="ip", prn=process_packet, store=False)
```

- Sniff(): Captures only IP packets(filter = “ip”) in real time.
- Each packet is sent to the firewall() function for inspection.

## Summary & Alerts

```
log_message(f"Summary: Total blocked packets: {block_counter}", "INFO")

if block_counter >= ALERT_THRESHOLD:
    log_message(f"ALERT: High number of blocked packets ({block_counter})", "CRITICAL")
```

- Displays total number of blocked ports.
- Generated alert if blocked count exceeds that of the threshold set earlier.

## Sample Output

```
[INFO] 2025-11-06 14:51:21 - Starting live packet capture...
[WARNING] 2025-11-06 14:51:22 - Blocked: Port 19312 not in allowed list
Blocked: Port 19312 not allowed
[WARNING] 2025-11-06 14:51:22 - Blocked: Port 53 not in allowed list
Blocked: Port 53 not allowed
[WARNING] 2025-11-06 14:51:22 - Blocked: Port 53 not in allowed list
Blocked: Port 53 not allowed
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.8.8 not in allowed list
Blocked: SRC IP 8.8.8.8 not allowed
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.8.8 not in allowed list
Blocked: SRC IP 8.8.8.8 not allowed
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
Blocked: SRC IP 8.8.4.4 not allowed
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
Blocked: SRC IP 8.8.4.4 not allowed
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
```

### Log File(firewall\_log.txt):

```
[INFO] 2025-11-06 14:51:21 - Starting live packet capture...
[WARNING] 2025-11-06 14:51:22 - Blocked: Port 19312 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: Port 53 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: Port 53 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.8.8 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.8.8 not in allowed list
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[INFO] 2025-11-06 14:51:22 - Allowed: Packet from 192.168.100.200 to 8.8.4.4 on port 443
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
[WARNING] 2025-11-06 14:51:22 - Blocked: SRC IP 8.8.4.4 not in allowed list
```

# Security & Limitations

## Security Notes:

- The script needs administrative/root privileges to sniff packets.
- Logging file (firewall\_log.txt) should be protected from modification.
- Works at user-level; for production, kernel-level filtering (e.g., iptables or nftables) is recommended.

## Limitations:

1. Doesn't modify packet flow (read-only monitoring).
2. No support for deep packet inspection.
3. Performance may be weak with heavy traffic.

## Future Work

- Add GUI for better management
- Add Mail/SMS notifications for alerts
- Add Automatic IP blocking based on packet frequency

## Conclusion

This project demonstrates how Python and Scapy can be used to build a simple yet effective **rule-based firewall** for network packet inspection. It provides clear insights into how traffic filtering, logging, and alerting mechanisms work together to enhance network security.