****

**Firewalls: Overview Setup Configuration and Management**

**Setting Up Firewalls Configuration and Management on My Kali Linux VM**

**By**

**Musyoka John Nzola**

**For**

**Future Interns CS Task 02**

**2024**

**Abstract**

This project provides an overview of firewalls, their types, features, and the essential role they play in network security. It explores the process of setting up and configuring a basic firewall on a **virtual machine (VM) using UFW (Uncomplicated Firewall)**, a front-end tool for managing iptables on Linux-based systems. The project demonstrates the step-by-step process of enabling the firewall, configuring rules to allow or block specific types of traffic, and verifying the rules using various commands. Furthermore, it covers advanced configurations such as rate limiting, logging, and managing IP-based access controls. Through this project, users gain hands-on experience in securing a virtual machine by effectively using firewall rules to control inbound and outbound traffic. The ultimate goal is to enhance system security by preventing unauthorized access while ensuring legitimate traffic flows seamlessly.

**1. Introduction to Firewalls**

A firewall is a security device or software that monitors and controls incoming and outgoing network traffic based on predetermined security rules. Its primary role is to establish a barrier between a trusted internal network and untrusted external networks, such as the internet. Firewalls are critical for preventing unauthorized access, cyberattacks, and ensuring secure communication.

**Types of Firewalls**

1. **Network-based Firewall** Protect an entire network by filtering traffic at the perimeter.
2. **Host-based Firewalls**: Installed on individual machines to filter traffic entering or leaving that particular device.

**Key Features of Firewalls**

* **Packet Filtering**: Examines packets and allows or blocks them based on predefined rules (IP addresses, ports, protocols).
* **Stateful Inspection**: Tracks the state of active connections and makes decisions based on the context of the traffic.
* **Proxying and Network Address Translation (NAT**): Acts as an intermediary for requests between a client and server, often used in secure setups.
* **Application Layer Filtering**: Monitors traffic at the application layer (e.g., web traffic) to block malicious payloads or commands.

1. **Setting Up a Basic Firewall on My Kali Linux Virtual Machine (VM)**

setting up a basic firewall on my Linux-based virtual machine (VM) using UFW (Uncomplicated Firewall), a user-friendly frontend for iptables.

**Requirements:**

A Linux-based VM (kali Linux).

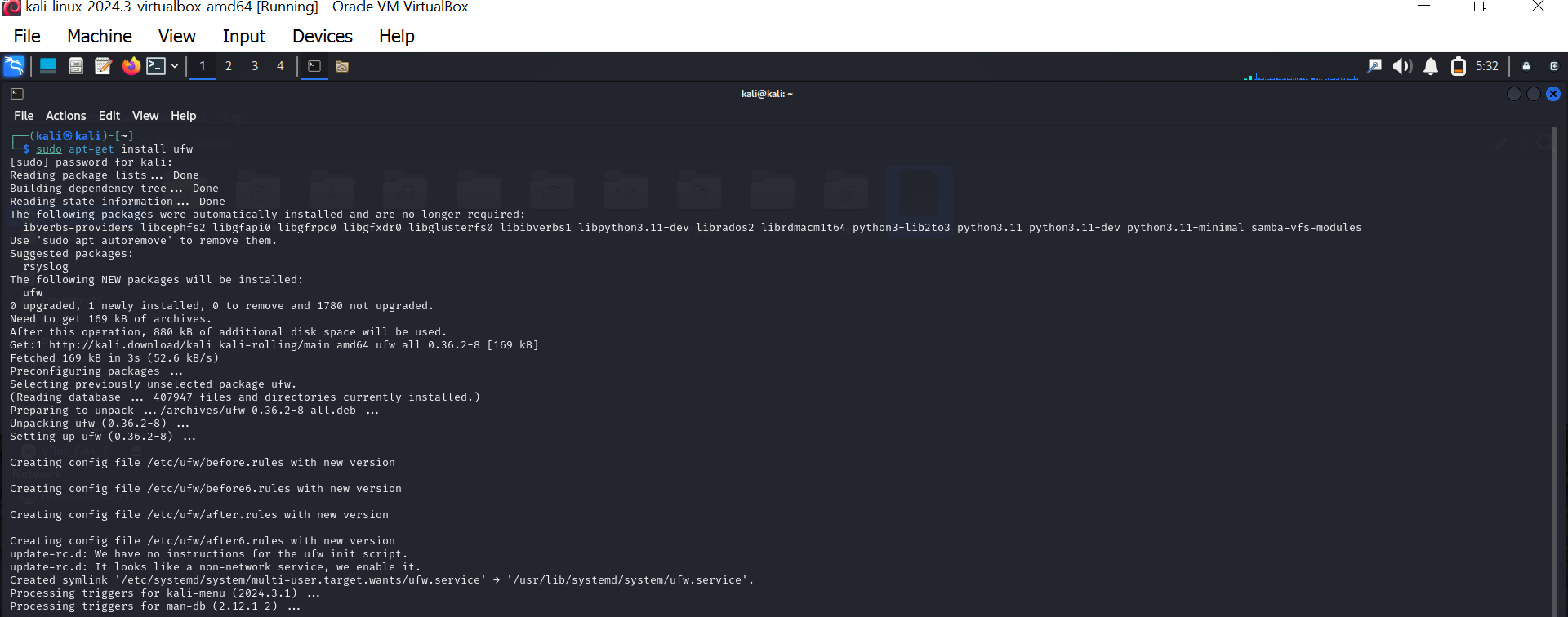
Access to the terminal with root or sudo privileges.

**Step-by-step Setup**:

1. **Installing UFW** (if not already installed): Opening the terminal and running the following command to install UFW:

*sudo apt-get update*

*sudo apt-get install* *ufw*



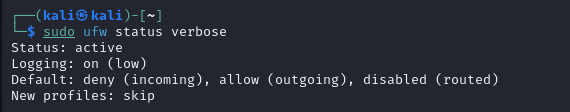
1. **Enabling the UFW Firewall**

*sudo ufw enable*



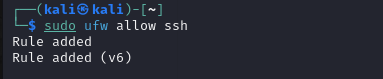
1. **Checking UFW Status**

*sudo ufw status verbose*



1. **Allowing SSH Connections**

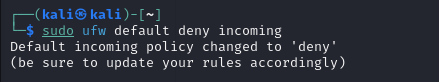
*sudo ufw allow ssh*

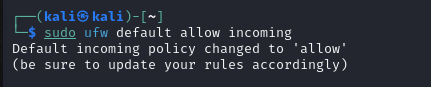


1. **Blocking Incoming Traffic by Default**

*sudo ufw default deny incoming*

*sudo ufw default allow outgoing*

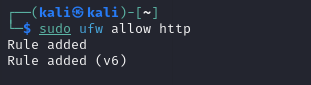


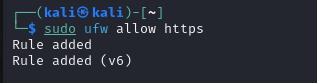


1. **Allowing HTTP and HTTPS Traffic**

*sudo ufw allow http*

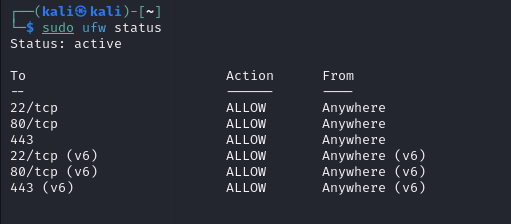
*sudo ufw allow https*





1. **Verify the Firewall Rules**

*sudo ufw status*

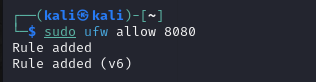


**3. Configuring Firewall Rules to Allow and Block Specific Traffic**

Firewall rules determine the type of traffic that is permitted or denied. UFW allows you to add specific rules for blocking or allowing traffic based on ports, IP addresses, and protocols.

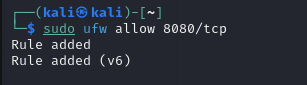
**Basic Commands for Managing Rules**

1.**Allowing a Specific Port**: To allow a specific port (e.g., port 8080): *sudo ufw allow 8080*



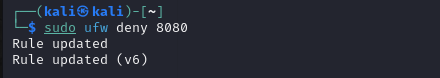
To specify the protocol (TCP or UDP), use:

*sudo ufw allow 8080/tcp*



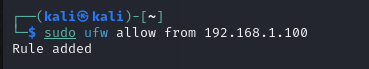
**Blocking a Specific Port**: To block a specific port (e.g., port 8080):

*sudo ufw deny 8080*



**Allowing or Blocking Specific IP Addresses**: To allow traffic from a specific IP address (e.g., 192.168.1.100):

*sudo ufw allow from 192.168.1.100*



To block traffic from a specific IP address:

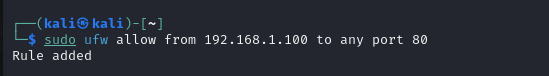
*sudo ufw deny from 192.168.1.100*



**Allowing or Blocking a Specific IP Address on a Specific Port**: For example, to allow IP 192.168.1.100 to access port 80:

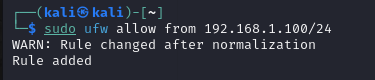
*sudo ufw allow from 192.168.1.100 to any port 80*

To block the same IP from accessing port 80:



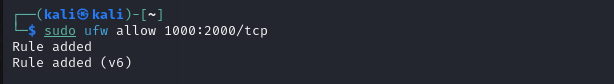
**Allowing Specific IP Ranges**: You can also allow or block IP address ranges:

*sudo ufw allow from 192.168.1.0/24*

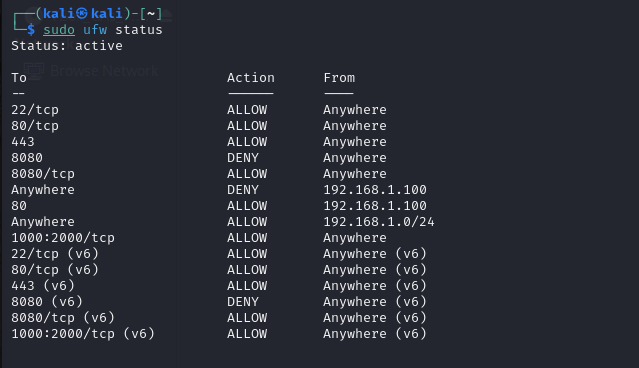


**Allowing or Blocking a Range of Ports**: For example, to allow ports 1000 to 2000:

sudo ufw allow 1000:2000/tcp



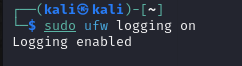
My firewall status after management



**4. Advanced Firewall Configuration with UFW**

**Logging**: UFW supports logging, which helps in tracking firewall activity. You can enable logging with:

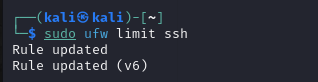
*sudo ufw logging on*



By default, logs are saved to /var/log/ufw.log.

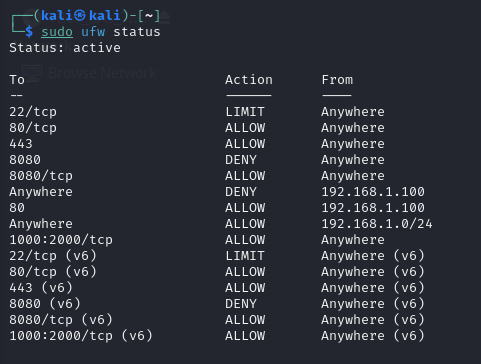
**Rate Limiting**: Protect services from brute-force attacks by limiting the rate of incoming connections. For example, to limit SSH connections to 3 attempts per minute:

*sudo ufw limit ssh*



**6. Verifying Firewall Rules**

After configuring your firewall, you can verify your rules with the ufw status command to see the allowed and denied traffic. You can also monitor firewall logs to ensure that traffic is being blocked or allowed as expected.



**Conclusion**

A firewall is an essential component of network security, helping to control traffic and prevent unauthorized access. By setting up a basic firewall on a virtual machine (using tools like UFW), you can block or allow specific types of traffic based on your requirements. Regular monitoring and configuration of firewall rules will ensure your system remains secure.