

# **CYBER SECURITY**

A report submitted in partial fulfillment of the  
requirement for the award of the degree of

## **Bachelor of Technology in Electronics And Communication Engineering**

by  
**YANAMALA ABHIRAM(Y21EC192)**



**Department of Electronics & Communication Engineering  
R.V.R. & J.C. COLLEGE OF ENGINEERING  
(Autonomous)**

Approved by AICTE :: Affiliated to Acharya Nagarjuna University  
Chowdavaram, Guntur - 522019, Andhra Pradesh, India

**2024**

## Department of Electronics And Communication Engineering



### CERTIFICATE

This is to certify that the report of **EC451 Internship** entitled “**CYBER SECURITY**” that is being submitted by “**YANAMALA ABHIRAM(Y21EC192)** ” in partial fulfillment of the requirement of the Degree of **Bachelor of Technology** in **Electronics And Communication Engineering** to the R.V.R. & J.C. College of Engineering is a record of bonafide work carried out by him/her under my supervision.

Date:

**Signature of Coordinator**

**Dr.X,Ascar Davix** M.E., Ph.D

**Associate Professor in ECE**

**Signature of HOD**

**Dr.T.Ranga Babu** M.Tech., Ph.D

**Professor & Head**

# Abstract

The following abstract summarizes the process of building and configuring the Un-complicated Firewall (UFW) in Kali Linux, focusing on its installation, configuration, and management. Firewalls are critical for safeguarding systems by regulating network traffic based on predefined security rules. This document presents a detailed guide on building and configuring UFW in Kali Linux, an essential tool for enhancing system security.

Users can define specific rules to allow or block traffic for various services such as SSH (port 22), HTTP (port 80), and HTTPS (port 443). The configuration process includes checking the status of UFW, adding or removing rules, and enabling logging for monitoring purposes. Additionally, users can manage IP addresses by allowing or denying specific sources.



# Table of contents

Abstract . . . . .	i
Table of Contents . . . . .	iv
<b>1 Chapter 1</b>	<b>1</b>
1 Introduction . . . . .	1
2 The Role of FireWalls in CyberSecurity . . . . .	1
3 Functions of Firewalls . . . . .	1
3.1 Traffic Monitoring: . . . . .	1
3.2 Access Control: . . . . .	1
3.3 Threat Prevention: . . . . .	1
4 Types of Firewalls . . . . .	2
4.1 Packet Filtering: . . . . .	2
4.2 Stateful Inspection Firewall: . . . . .	2
4.3 Proxy Firewalls: . . . . .	2
<b>2 Chapter 2</b>	<b>3</b>
1 Understanding UFW . . . . .	3
2 Importance of Firewall . . . . .	3
3 Key Features of UFW . . . . .	3
4 Difference between UFW and GFW . . . . .	4
<b>3 Chapter 3</b>	<b>5</b>
1 Installation of UFW . . . . .	5
2 Debian vs Ubuntu: Key Differences . . . . .	5
2.1 1. Development and Community . . . . .	5
2.2 2. Release Cycle . . . . .	6
2.3 3. Stability vs. Cutting Edge . . . . .	6
2.4 4. Package Management . . . . .	6
2.5 5. Desktop Environment . . . . .	6
2.6 6. User Experience . . . . .	6
2.7 7. Hardware Requirements . . . . .	7
2.8 8. Software Availability . . . . .	7
3 Verifying Installation . . . . .	7
4 Status Verification . . . . .	7
5 Reinstallation of Uncomplicated Firewall . . . . .	7

<b>4</b>	<b>Chapter 4</b>	<b>9</b>
1	Enabling UFW . . . . .	9
2	Setting Default Policies . . . . .	9
3	Disabling UFW . . . . .	10
<b>5</b>	<b>Chapter 5</b>	<b>11</b>
1	Allowing Specific Services . . . . .	11
2	Blocking Specific IP Address . . . . .	12
3	Allowing Traffic from Specific IP Address . . . . .	12
4	Managing Application Profiles . . . . .	12
5	Viewing Profile Information . . . . .	12
<b>6</b>	<b>Chapter 6</b>	<b>14</b>
1	Viewing current Rules . . . . .	14
2	Deleting Rules . . . . .	14
3	Rules for Ports and Address . . . . .	15
4	Important Port Numbers . . . . .	15
<b>7</b>	<b>Chapter 7</b>	<b>16</b>
1	Logging Options . . . . .	16
2	Resetting UFW . . . . .	16
3	Resetting UFW Rules . . . . .	16
<b>8</b>	<b>Chapter 8</b>	<b>18</b>
1	Testing Firewall . . . . .	18
1.1	Key Features of Nmap . . . . .	18
2	Firewall Rules . . . . .	19
3	Configuration Details . . . . .	19
<b>9</b>	<b>Chapter 9</b>	<b>21</b>
1	Conclusion . . . . .	21

# 1

## Chapter 1

---

### 1 Introduction

Cybersecurity is a critical field focused on protecting systems, networks, and data from cyber threats. As the digital landscape evolves, so do the methods employed by malicious actors, making effective cybersecurity strategies essential for individuals and organizations alike.

### 2 The Role of FireWalls in CyberSecurity

A firewall serves as a crucial line of defense in cybersecurity. It is a network security device that monitors and controls incoming and outgoing network traffic based on predetermined security rules. Firewalls can be hardware-based or software-based and are designed to establish a barrier between trusted internal networks and untrusted external networks.

### 3 Functions of Firewalls

- 3.1 Traffic Monitoring: Firewalls analyze data packets entering or leaving a network
- 3.2 Access Control: They enforce rules that determine which traffic is allowed or blocked.
- 3.3 Threat Prevention: Firewalls can prevent unauthorized access and mitigate potential attacks.

## **4 Types of Firewalls**

4.1 Packet Filtering: Inspect packets and allow or block them based on defined rules.

4.2 Stateful Inspection Firewall: Track active connections and make decisions based on the state of the connection.

4.3 Proxy Firewalls: Serve as intermediaries between users and the services they access, providing additional security.



# 2

## Chapter 2

---

### 1 Understanding UFW

Uncomplicated Firewall (UFW) is an application designed to simplify the management of firewall rules on Linux systems. It provides an easy-to-use interface for configuring iptables, allowing users to define rules without extensive networking knowledge.

### 2 Importance of Firewall

Firewalls are essential for protecting systems from unauthorized access and attacks. They act as barriers between trusted internal networks and untrusted external networks, preventing malicious activities.

### 3 Key Features of UFW

1. **User-Friendly Interface:** UFW provides a straightforward command-line interface, making it accessible for users who may not be familiar with the complexities of iptables.
2. **Default Policies:** By default, UFW denies all incoming connections and allows all outgoing connections. This setup minimizes the risk of unauthorized access while permitting outbound traffic.
3. **Rule Management:** Users can easily create, delete, and modify rules to allow or deny specific traffic based on ports, IP addresses, or protocols.

4. **Logging Capabilities:** UFW can log firewall activity, providing insights into attempts to access the system. This feature can be enabled or disabled based on user preference.
5. **IPv6 Support:** UFW supports both IPv4 and IPv6, allowing for comprehensive network security across different protocols.

## 4 Difference between UFW and GFW

- UFW is ideal for users comfortable with the command line who need a straightforward way to manage firewall rules quickly.
- GFW, on the other hand, is best suited for those who prefer a graphical interface and want to manage their firewall settings without delving into command-line complexities.

Both tools ultimately control the same underlying firewall functionality (iptables), but they cater to different user preferences and expertise levels. Users can choose one based on their comfort level with command-line interfaces versus graphical interfaces.

# 3

## Chapter 3

---

### 1 Installation of UFW

To install UFW (Uncomplicated Firewall) on a Linux system, particularly Ubuntu or Debian-based distributions, follow these detailed steps.

### 2 Debian vs Ubuntu: Key Differences

Debian and Ubuntu are two of the most popular Linux distributions, each with its own strengths and weaknesses. Below is a detailed comparison of their key differences.

#### 2.1 1. Development and Community

- **Debian:** Developed by a community of volunteers, Debian is known for its commitment to free and open-source software. It follows a democratic governance model, allowing contributors to have a say in its development. This community-driven approach ensures that all software included in Debian adheres strictly to open-source principles.
- **Ubuntu:** Developed by Canonical Ltd., Ubuntu has a more corporate structure. While it also has a strong community, Canonical leads the development efforts and makes decisions that can prioritize mainstream compatibility and user-friendliness. This can sometimes lead to decisions that favor commercial interests.

## 2.2 2. Release Cycle

- **Debian:** Debian does not follow a fixed release schedule. Instead, it releases new stable versions when the development team feels it is ready, which can take several years. This results in a stable environment with fewer updates, making it ideal for server use where stability is crucial.
- **Ubuntu:** Ubuntu has a regular release cycle, with new versions released every six months and Long Term Support (LTS) versions every two years. LTS versions receive five years of support, making them suitable for production environments where stability and security updates are essential.

## 2.3 3. Stability vs. Cutting Edge

- **Debian:** Known for its stability, Debian often uses older but thoroughly tested software packages. This approach minimizes bugs and unexpected behavior, making it a preferred choice for servers and critical applications.
- **Ubuntu:** While Ubuntu aims for stability, it tends to include newer software versions and features to enhance user experience. This means users may encounter more frequent updates and newer features but at the potential cost of some stability compared to Debian.

2.4 4. Package Management Both Debian and Ubuntu use the Advanced Package Tool (APT) for package management and share the same underlying package format (.deb). However:

- **Debian:** Focuses on stability with fewer package updates.
- **Ubuntu:** Includes additional package management tools like Snap, which allows for easier installation of applications but can introduce complexity.

## 2.5 5. Desktop Environment

- **Debian:** Does not come with a default desktop environment (DE) installed; instead, it allows users to choose from various DEs during installation (e.g., GNOME, KDE, Xfce). This flexibility appeals to users who want a customized experience.
- **Ubuntu:** Comes with a customized version of GNOME as its default DE, designed for ease of use and accessibility. Ubuntu also offers official flavors like Kubuntu (KDE), Xubuntu (Xfce), and Lubuntu (LXQt) for users who prefer different environments.

## 2.6 6. User Experience

- **Debian:** Generally considered less user-friendly due to its focus on customization and flexibility. Users may need more technical knowledge to configure their systems effectively.
- **Ubuntu:** Aims to be user-friendly with an intuitive interface and pre-installed software that caters to everyday users. Its Software Center simplifies application installation, making it accessible for newcomers.

## 2.7 7. Hardware Requirements

- **Debian:** Typically has lower hardware requirements compared to Ubuntu, making it suitable for older hardware or minimal installations.
- **Ubuntu:** Generally requires more resources due to its additional features and graphical interface enhancements.

## 2.8 8. Software Availability

- **Debian:** Prioritizes free software in its repositories and may not include proprietary drivers or applications by default. This can limit access to some popular applications but ensures adherence to open-source principles.
- **Ubuntu:** Supports both free and proprietary software, providing access to a broader range of applications out of the box. This makes it easier for users who rely on specific commercial software or drivers.

## Conclusion

In summary, both Debian and Ubuntu have their unique strengths that cater to different types of users:

- Choose Debian if you prioritize stability, customization, and open-source principles.
- Choose Ubuntu if you prefer ease of use, regular updates, and access to a wide range of software with commercial support options.

## 3 Verifying Installation

After installation, verify that UFW is installed correctly by checking its version

## 4 Status Verification

This verifies the current status of the UFW (active or inactive)

## 5 Reinstallation of Uncomplicated Firewall

**If you encounter errors during installation, check the output for specific error messages that may indicate missing dependencies or issues with the package manager. If problems persist, you can try removing and reinstalling UFW.**

Building and configuring a firewall is crucial for protecting networks from unauthorized access and potential threats. This tutorial will guide you through setting up and configuring a firewall on an Ubuntu system using UFW (Uncomplicated Firewall).

### Prerequisites

- Basic knowledge of Linux commands
- An Ubuntu system (physical or virtual machine)
- Root or sudo access

### Step-by-Step Guide

#### Step 1: Update Your System

Ensure your system is up to date.

```
bash Copy code  
  
sudo apt update  
sudo apt upgrade -y
```

#### Step 2: Install UFW

UFW is included in most Ubuntu installations by default, but you can install it if it's not present.

```
bash Copy code  
  
sudo apt install ufw
```

#### Step 3: Enable UFW

By default, UFW is disabled after installation. Enable it with the following command:

```
bash  
  
sudo ufw version
```

```
`sudo ufw status`
```

```
bash  
  
sudo apt remove --purge ufw  
sudo apt install ufw
```

# 4

## Chapter 4

---

### 1 Enabling UFW

After installation, UFW is disabled by default. To enable it

```
bash
sudo ufw enable
```

### 2 Setting Default Policies

Set default policies to deny all incoming traffic and allow all outgoing traffic

```
bash
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

### 3 Disabling UFW

This disables the Uncomplicated Firewall rules being enforced.

```
`sudo ufw disable`
```



# 5

## Chapter 5

---

### 1 Allowing Specific Services

To allow specific services through the firewall:

- **Allow SSH:** To prevent yourself from being locked out of the system, allow SSH connections. You can do this with the following command:

```
bash
sudo ufw allow ssh
```

**SSH, or Secure Shell, is a cryptographic network protocol that enables secure communication between computers over an unsecured network. Its primary applications include remote login and command execution, making it a vital tool for system administrators and developers.**

## 2 Blocking Specific IP Address

To block an IP address from accessing your system

```
bash
sudo ufw deny from [IP_ADDRESS]

For example:

bash
sudo ufw deny from 192.168.1.100
```

## 3 Allowing Traffic from Specific IP Address

To allow traffic from a specific IP address

```
bash
sudo ufw allow from [IP_ADDRESS]

For example:

bash
sudo ufw allow from 192.168.1.101
```

## 4 Managing Application Profiles

UFW can manage application profiles that automatically open the necessary ports for specific applications.

## 5 Viewing Profile Information

To get more details about a specific application profile, including the ports it uses, run

```
bash
sudo ufw app list
```

To allow an application (e.g., Nginx):

```
bash
sudo ufw allow "Nginx Full"
```

```
bash
sudo ufw app info 'profile_name'
```

**Replace 'profile name' with the actual name of the profile you want to inspect (e.g., Apache, Nginx).]**

- **Application profiles in UFW provide a way to group related firewall rules for specific applications. This allows users to enable or disable access for an application with a single command, rather than having to specify multiple port numbers individually.**

# 6

## Chapter 6

---

### 1 Viewing current Rules

To check the status and view current rules

```
bash
sudo ufw status verbose
```

### 2 Deleting Rules

To delete a specific rule Or delete by rule number.

```
bash
sudo ufw delete allow [SERVICE]
```

```
bash
sudo ufw status numbered    # Get rule numbers first.
sudo ufw delete [number]
```

### 3 Rules for Ports and Address

<code>`sudo ufw allow 2222/tcp`</code>	Allows incoming TCP connections on port 2222 (useful if SSH runs on a non-standard port).
<code>`sudo ufw allow http`</code>	Allows incoming HTTP traffic (port 80).
<code>`sudo ufw allow https`</code>	Allows incoming HTTPS traffic (port 443).
<code>`sudo ufw allow from 192.168.1.100`</code>	Allows traffic from a specific IP address (192.168.1.100).
<code>`sudo ufw deny from 192.168.1.200`</code>	Denies traffic from a specific IP address (192.168.1.200).

### 4 Important Port Numbers

Port Number	Service	Description
20	FTP Data Transfer	Used for transferring files via FTP.
21	FTP Command Control	Used for controlling FTP sessions.
22	SSH (Secure Shell)	Used for secure remote login and command execution.
23	Telnet	Unencrypted text communication for remote login.
25	SMTP (Simple Mail Transfer Protocol)	Used for sending emails.
53	DNS (Domain Name System)	Used for resolving domain names to IP addresses.

# 7

## Chapter 7

---

### 1 Logging Options

```
bash
sudo ufw logging on
```

### 2 Resetting UFW

```
bash
sudo ufw reset
```

### 3 Resetting UFW Rules

```
bash  
sudo ufw reload
```

# 8

## Chapter 8

---

### 1 Testing Firewall


Nmap (Network Mapper) is an open-source tool used for network discovery and security auditing. It is widely utilized by network administrators and security professionals to scan networks, identify active devices, and assess the security posture of systems.

#### 1.1 Key Features of Nmap

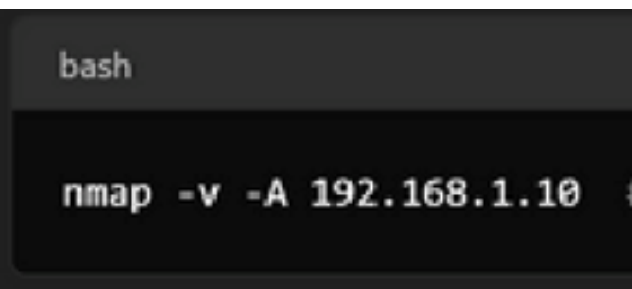
1. **Host Discovery:** Nmap can determine which hosts are up and available on a network, helping users identify active devices.
2. **Port Scanning:** It scans for open ports on target systems, providing insights into which services are running. This is crucial for identifying potential vulnerabilities.
3. **Service Version Detection:** Nmap can determine the version of services running on open ports, allowing for better vulnerability assessment.



4. **Operating System Detection:** By analyzing responses from devices, Nmap can often identify the operating system in use, aiding in targeted security assessments.
5. **Scripting Engine:** Nmap includes a powerful scripting engine (NSE) that allows users to write scripts for automated tasks, such as vulnerability detection or service enumeration.
6. **Flexible Output Options:** Nmap supports various output formats, including plain text, XML, and HTML, making it easy to integrate with other tools or for reporting purposes.



```
bash
nmap [target]
```



```
bash
nmap -v -A 192.168.1.10
```

**Use ‘Nmap’ from another machine to scan the open ports on our firewall protected system** Replace the IP address with the actual IP address of the device.

## 2 Firewall Rules

Document all the rules you have added to UFW. This can be simple text in file listing each rule.

## 3 Configuration Details

Document the configuration details of your firewall, including default policies and any logging or application profiles used.

plaintext

```
sudo ufw allow ssh
sudo ufw allow http
sudo ufw allow https
sudo ufw allow from 192.168.1.0/24
sudo ufw deny 23/tcp
```

# 9

## Chapter 9

---

### 1 Conclusion

In conclusion, configuring the Uncomplicated Firewall (UFW) is a vital step in securing your Linux system, particularly for Ubuntu and Debian-based distributions. UFW simplifies the management of firewall rules, providing a user-friendly interface on top of the more complex iptables. By default, UFW denies all incoming connections while allowing all outgoing traffic, which is an effective baseline for security. This default setting minimizes the risk of unauthorized access while permitting necessary outbound communications.

To enhance security further, it is crucial to define and allow specific services that are necessary for your operations, such as SSH for remote access or HTTP/HTTPS for web services. This selective allowance helps maintain a secure environment while ensuring that legitimate traffic can flow freely.

**Additionally, UFW's flexibility allows users to set custom rules based on ports and IP addresses, enabling tailored security configurations that meet specific needs. The ability to easily enable or disable UFW and modify rules provides administrators with the control necessary to adapt to changing security requirements.**

**Regular monitoring and adjustments of firewall rules are recommended to ensure ongoing protection against potential threats. By following best practices and leveraging UFW's capabilities, users can significantly improve their system's security posture and protect against unauthorized access and cyber threats.**