

# Lab#07: Host-Based Firewall Configuration using UFW








## Objective

Learn how to configure a host-based firewall using ufw (Uncomplicated Firewall) to allow or deny specific traffic and test it from a remote machine.

## Lab Requirements:

- **Kali Linux** (Server role)
- **Parrot OS / Kali Linux** (Client role)
- ufw installed on both systems
- Network connectivity (Ping and SSH should initially work)

## Step-by-Step Instructions / Summary

-  **Part 1: Initial Setup and Connectivity**
  - Step 1: Get the IP address of Server
  - Step 2: Ping Test from Client
-  **Part 2: Install and Enable UFW on Server**
  - Step 3: Install UFW (if not present)
  - Step 4: Enable UFW
  - Step 5: Check Current Status and Rules
-  **Part 3: Block All Incoming Except SSH**
  - Step 6: Set Default Deny Policy
  - Step 7: Allow SSH
-  **Part 4: Test Firewall Blocking from Client**
  - Step 8: Try to Ping from Client (should fail)
-  **Part 5: Allow HTTP (Simulated Web Server Test)**
  - Step 10: Install Apache2 on Server
  - Step 11: Allow HTTP
  - Step 12: Access Web Server from Client
-  **Part 6: View and Delete Rules**
  - Step 13: View Rules with Numbers
  - Step 14: Delete a Rule
-  **Bonus: Enable Logging**

Steps and screenshots for this lab:

## Part 1: Initial Setup and Connectivity

Get the IP Address of Server (Kali)

ip a

```
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:6e:2f:58 brd ff:ff:ff:ff:ff:ff  
    inet 172.16.123.129/24 brd 172.16.123.255 scope global dynamic noprefixroute eth0  
        valid_lft 954sec preferred_lft 954sec  
    inet6 fe80::20c:29ff:fe6e:2f58/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Ping Test from Client (Parrot)

ping 172.16.123.129

```
[user@parrot]-[~]  
$ ping 172.16.123.129  
PING 172.16.123.129 (172.16.123.129) 56(84) bytes of data.  
64 bytes from 172.16.123.129: icmp_seq=1 ttl=64 time=220 ms  
64 bytes from 172.16.123.129: icmp_seq=2 ttl=64 time=1.14 ms  
64 bytes from 172.16.123.129: icmp_seq=3 ttl=64 time=111 ms  
64 bytes from 172.16.123.129: icmp_seq=4 ttl=64 time=1.08 ms
```

## Part 2: Install and Enable UFW on Server

Check any packages needed to be installed

sudo apt update

```
(kali㉿kali)-[~]  
$ sudo apt update  
[sudo] password for kali:  
Hit:1 http://http.kali.org/kali kali-rolling InRelease  
1118 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Installs ufw (if not present)

sudo apt install ufw -y

```
(kali㉿kali)-[~]  
$ sudo apt install ufw -y  
ufw is already the newest version (0.36.2-9).  
Summary:  
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1118
```

### Enable UFW

sudo ufw enable

```
(kali㉿kali)-[~]  
$ sudo ufw enable  
Firewall is active and enabled on system startup
```

### Check Current Status and Rules

sudo ufw status verbose

```
(kali㉿kali)-[~]  
$ sudo ufw status verbose  
Status: active  
Logging: on (low)  
Default: deny (incoming), allow (outgoing), disabled (routed)  
New profiles: skip
```

## Part 3: Block All Incoming Except SSH

### Set Default Deny Policy

sudo ufw default deny incoming

sudo ufw default allow outgoing

```
(kali㉿kali)-[~]  
$ sudo ufw default deny incoming  
Default incoming policy changed to 'deny'  
(be sure to update your rules accordingly)  
  
(kali㉿kali)-[~]  
$ sudo ufw default allow outgoing  
Default outgoing policy changed to 'allow'  
(be sure to update your rules accordingly)
```

### Allow SSH

sudo ufw allow ssh

```
(kali㉿kali)-[~]  
$ sudo ufw allow ssh  
Rule added  
Rule added (v6)
```

#### Check the rules that were added

sudo ufw status verbose

```
(kali㉿kali)-[~]  
$ sudo ufw status verbose  
Status: active  
Logging: on (low)  
Default: deny (incoming), allow (outgoing), disabled (routed)  
New profiles: skip  
  
To Action From  
--  
22/tcp ALLOW IN Anywhere  
22/tcp (v6) ALLOW IN Anywhere (v6)
```

#### 🔄 Part 4: Test Firewall Blocking from Client

##### Client (Parrot) pings to Server (Kali)

ping 172.16.123.129

After many moments of waiting, it still appears the request to be blocked

```
[user@parrot]-[~]  
$ ping 172.16.123.129  
PING 172.16.123.129 (172.16.123.129) 56(84) bytes of data.  
[redacted]
```

##### Try to SSH into Server (should work)

This should succeed because port 22 was allowed

ssh kali@172.16.123.129

```
[user@parrot]-[~]
$ ssh kali@172.16.123.129
The authenticity of host '172.16.123.129 (172.16.123.129)' can't be established.
ED25519 key fingerprint is SHA256:DAZoZL/NurKT+1qjNEc0uAnCPLGq42jvwMvWruRB+QQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.123.129' (ED25519) to the list of known hosts
.
kali@172.16.123.129's password:
Linux kali 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11)
x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jun 30 21:54:48 2025 from 172.16.123.129
(kali@kali)-[~]
$
```

## 🎯 Part 5: Allow HTTP (Simulated Web Server Test)

### Install Apache2 on Server

sudo apt install apache2 -y

```
(kali@kali)-[~]
$ sudo apt install apache2 -y
apache2 is already the newest version (2.4.64-1).
Upgrading:
  e2fsprogs      keyutils      libcurl3t64-gnutls  libhogweed6t64  libsasl2-modules  libxml2-utils
  gnutls-bin     krb5-locales  libext2fs2t64      libldap-common  libss2            logsave

Summary:
Upgrading: 12, Installing: 0, Removing: 0, Not Upgrading: 1106
Download size: 0 B / 2,627 kB
Space needed: 53.2 kB / 718 MB available
```

sudo systemctl start apache2

```
(kali@kali)-[~]
$ sudo systemctl start apache2
[sudo] password for kali:
```

### Check if service is running

sudo systemctl status apache2

```
(kali㉿kali)-[~]
$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled)
   Active: active (running) since Tue 2025-07-29 23:30:09 EDT; 2h 4min ago
     Invocation: 18cd895c842642539ce3868942974f94
       Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 334353 (apache2)
      Tasks: 6 (limit: 4502)
     Memory: 14.2M (peak: 25.9M)
        CPU: 1.289s
     CGroup: /system.slice/apache2.service
            └─334353 /usr/sbin/apache2 -k start
              └─376595 /usr/sbin/apache2 -k start
                └─376596 /usr/sbin/apache2 -k start
                  └─376597 /usr/sbin/apache2 -k start
                    └─376598 /usr/sbin/apache2 -k start
                      └─376599 /usr/sbin/apache2 -k start

Jul 29 23:30:09 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Jul 29 23:30:09 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
Jul 30 00:15:21 kali systemd[1]: Reloading apache2.service - The Apache HTTP Server ...
Jul 30 00:15:21 kali systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
```

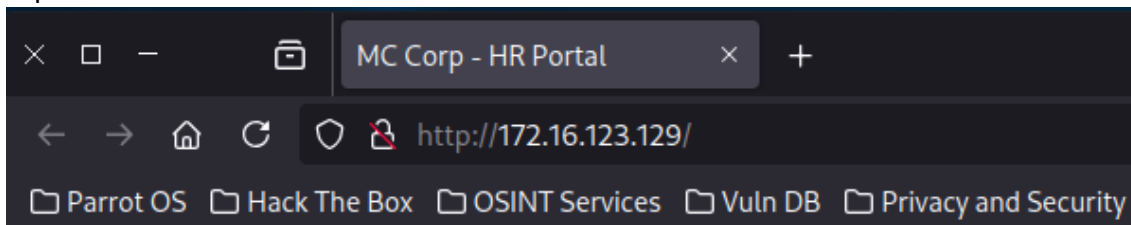
### Allow HTTP

sudo ufw allow http

```
(kali㉿kali)-[~]
$ sudo ufw allow http
Rule added
Rule added (v6)
```

### Access Web Server from Client

http://172.16.123.129/



# Welcome to MC Corp HR Portal

This is a secure internal HR site.

## Part 6: View and Delete Rules

### View Rules with Numbers

sudo ufw status numbered

```
(kali㉿kali)-[~]
$ sudo ufw status numbered
[sudo] password for kali:
Status: active

      To Action From
      --
[ 1] 22/tcp ALLOW IN Anywhere
[ 2] 80/tcp ALLOW IN Anywhere
[ 3] 22/tcp (v6) ALLOW IN Anywhere (v6)
[ 4] 80/tcp (v6) ALLOW IN Anywhere (v6)
```

### Delete a Rule

removes http access

sudo ufw delete allow http

```
(kali㉿kali)-[~]
$ sudo ufw delete allow http
Rule deleted
Rule deleted (v6)

(kali㉿kali)-[~]
$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
```

## Bonus: Enable Logging

sudo ufw logging on

```
(kali㉿kali)-[~]
$ sudo ufw logging on
Logging enabled
```

```
sudo tail -f /var/log/ufw.log
```

```
(kali@kali)-[~]
$ sudo tail -f /var/log/ufw.log
2025-07-30T02:09:37.995719-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:38.015334-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:39.168215-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:39.169944-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:39.955604-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:40.035183-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:41.008180-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:41.048839-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:42.027363-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
2025-07-30T02:09:42.063389-04:00 kali kernel: [UFW BLOCK] IN=eth0 OUT= MAC=00:0c:29:6e:2f:58:00:0c:29:89:8c:c0:08:00
=64240 RES=0x00 SYN URGP=0
```



## Tools & Skills Used

- **Operating Systems:** Kali Linux (Server), Parrot OS (Client)
- **Firewall Utility:** ufw (Uncomplicated Firewall)
- **Networking Tools:** ping (ICMP), ssh (TCP/22), ip a
- **System & Service Management:** sudo, apt, systemctl, tail
- **Core Skills:** Host-Based Firewall Configuration, Network Security, Rule Management, Port-Based Filtering (TCP), Protocol-Based Filtering (ICMP), Service Installation (Apache2), Log Monitoring



## Reflection & Takeaways

This lab was a practical and effective demonstration of the "deny by default" security principle, which is a cornerstone of network security. By configuring the firewall to block all incoming traffic first and then explicitly allowing only necessary services, I was able to create a secure and controlled environment.

My key takeaways from this lab are:

1. **The Importance of a Default Deny Policy:** The most critical step was setting ufw default deny incoming. This immediately hardens the server by ensuring that no unintended or malicious traffic can get through. It forces a deliberate and conscious decision for every service that needs to be exposed.
2. **Firewalls are Protocol-Specific:** A key lesson was seeing the ping (ICMP protocol) fail while the ssh (TCP port 22) and web server (TCP port 80) connections succeeded. This clearly illustrates that firewall rules are not a blanket "on or off" switch; they provide



granular control over specific ports and protocols, allowing an administrator to create precise security policies.

3. **Reduced Attack Surface:** By only opening ports 22 and 80, I significantly reduced the server's attack surface. An attacker running a port scan against this machine would only see these two services, leaving other potential vulnerabilities on different ports hidden and inaccessible. This is a fundamental step in preventing unauthorized access and reconnaissance.