SAFE SECURITY

UFW

A Beginners Guide to Linux firewall

2021

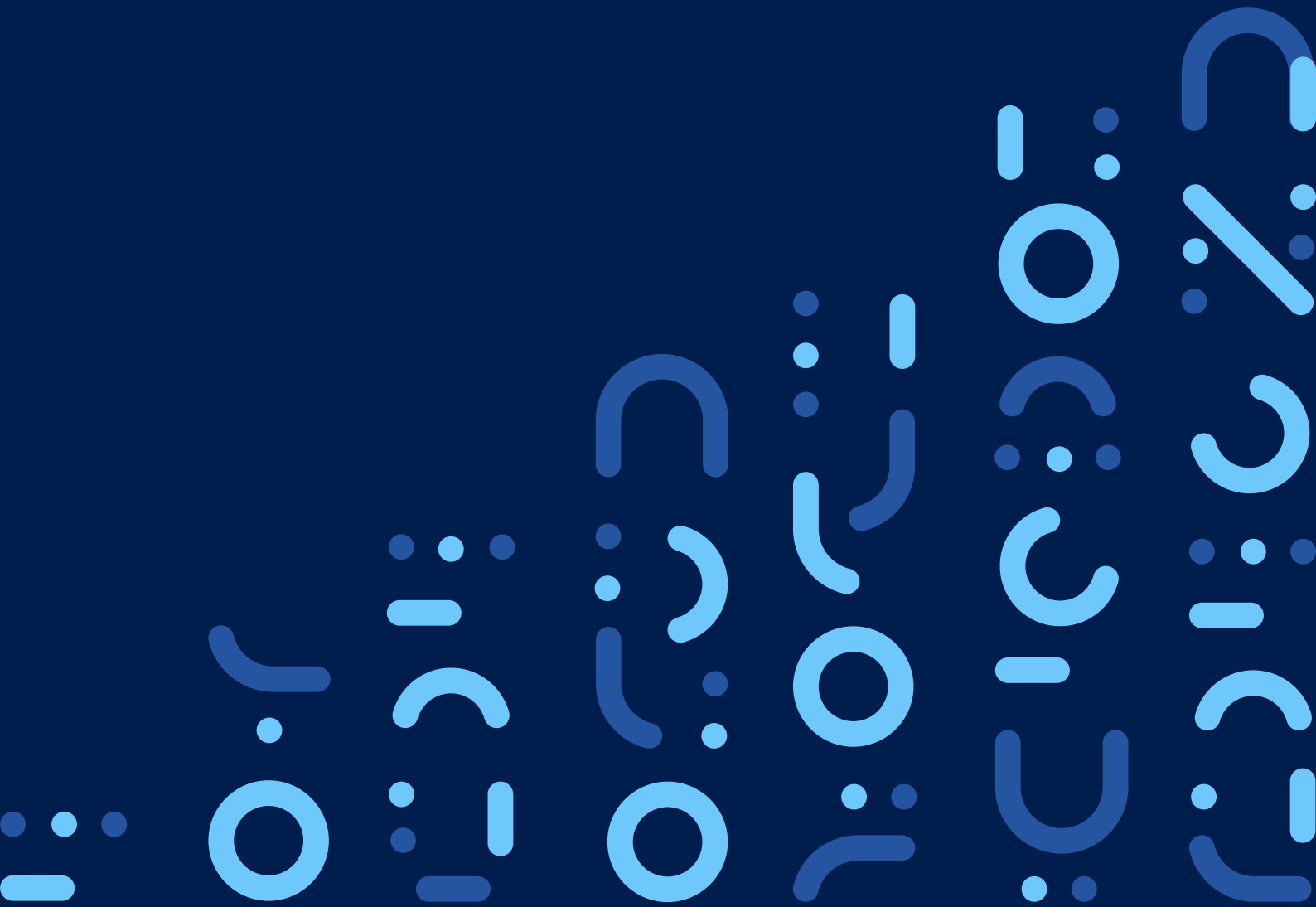


TABLE OF CONTENTS

Introduction 01

key Terms 01

Definitions 01

IP Tables & UFW 02

Getting Started 02

- Basic Commands 04
- Advanced Commands 10
- Problems with firewall 15
- Lab Exercise 16

Introduction

FIREWALL

is the first line of defence for any network, hence it is very important to learn how one can protect a network using a firewall. This document is intended to provide detailed information about how UFW can play a very important role for any Linux user in terms of securing the Linux system, this document provides all the information necessary to completely configure UFW and protect personal Linux systems from a variety of attacks.

KEY TERMS

Bash, Netfilterqueue, IPtables, UFW (Uncomplicated Firewall), Virtual Machine.

DEFINITIONS

Bash
Bash is a Unix shell written for the GNU Project as a free software replacement for the Bourne shell (sh). It is often installed as the system's default command-line interface. It provides end users an interface to issue system commands and execute scripts.

Netfilter
Netfilter is a framework provided by the Linux kernel that allows various networking-related operations to be implemented in the form of customized handlers. Netfilter offers various functions and operations for packet filtering, network address translation, and port translation.

IPtables
Iptables is a user-space utility program that allows a system administrator to configure the IP packet filter rules of the Linux kernel firewall, implemented as different Netfilter modules

UFW
Uncomplicated Firewall (UFW) is a program for managing a Netfilter firewall designed to be easy to use. It uses a command-line interface consisting of a small number of simple commands, and uses iptables for configuration.

Virtual Machine
A virtual machine is an emulation of a computer system.

IPTABLES AND UFW, WHAT'S THE DIFFERENCE?

IPtables and UFW both are Linux system firewalls, the difference between them is UFW is built upon IPtables, IPtables a very flexible tool but it's more complex as compared to UFW, other difference is that IPtables requires a deeper understanding of TCP/IP, which might not be the case with every Linux user, so UFW is the solution, UFW allows the user to configure firewall rules easily using IPtables. Hence, for an average Linux user UFW is the best way to get started with setting up different firewall rules.

UFW	• Easy to use • Less Flexible
IPtables	• Complex to use • More flexible

Before we start writing different rules for the firewall, we need to have one, so let's start by installing the firewall, since the system being used for purpose is Debian based, the command to install UFW is:

```
apt-get install ufw
```

Once installed, it's time to discover the tool itself, which can be done by writing:

```
ufw --help
```

```
Commands:
 enable                    enables the firewall
 disable                   disables the firewall
 default ARG               set default policy
 logging LEVEL             set logging to LEVEL
 allow ARGS                add allow rule
 deny ARGS                 add deny rule
 reject ARGS               add reject rule
 limit ARGS                add limit rule
 delete RULE|NUM           delete RULE
 insert NUM RULE           insert RULE at NUM
 prepend RULE              prepend RULE
 route RULE                add route RULE
 route delete RULE|NUM     delete route RULE
 route insert NUM RULE     insert route RULE at NUM
 reload                    reload firewall
 reset                     reset firewall
 status                    show firewall status
 status numbered           show firewall status as numbered list of RULES
 status verbose            show verbose firewall status
 show ARG                  show firewall report
 version                   display version information

Application profile commands:
 app list                  list application profiles
 app info PROFILE          show information on PROFILE
 app update PROFILE        update PROFILE
 app default ARG           set default application policy
```

BASIC COMMANDS

It's very important to get the basic commands right, once we know what the basic commands are we'll simply build upon them and then move to more advanced commands. To enable the UFW firewall we simply write the following command.

```
ufw enable
```

UFW firewall, it comes with some default rules, these rules might be appropriate for some situations but not necessarily for every situation , so it's best to configure the firewall from scratch.

Let's first view what the default rules are, to do that write following command:

```
ufw status verbose
```

```
Status: active
Logging: on (low)
Default: allow (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

The primary focus here is to interpret what's firewall is trying to tell us.

A. Status:
active
Means firewall is active

B. Logging:
on (low)
Means logging is turned on and set to low level, log can be found in /var/logs/ufw.log }

C. Default:
allow (incoming)
Means all the incoming connections are allowed

allow (outgoing)
Means all the outgoing connections are allowed

disabled (routed)
Means port forwarding is disabled , basically no routing

Changing Default Rules

Before changing the rules, it's important to know what all operations are available to be performed on the connection, UFW provides us four different options for the connections which are:

- o Allow : Allow the connection
- o Deny : Deny the connection
- o Reject : Reject the connection
- o Limit : To add rate limiting rules , using this rule will block an IP addresses if more than six connections are instantiated within thirty seconds

Now there might be a confusion between Deny and Reject because they seem to be similar, the difference between the two is when we use deny, it doesn't tell the client what's actually happening, but when we use reject, it actually tells the client that the connection is being rejected by the server.

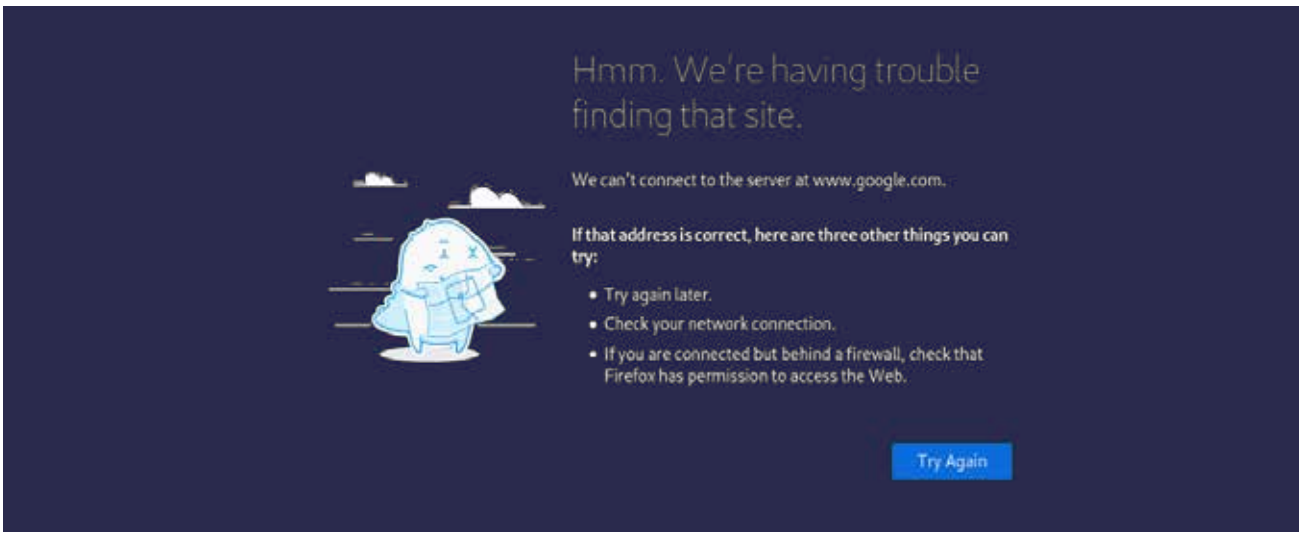
The next step is to change few default rules, using below given command

```
ufw default action flow
```

To change the default rule, specify the action (allow, deny etc.). After that, specify the flow of connection (incoming or outgoing). To deny all the outgoing connections, and to test it's working write following command:

```
ufw default reject outgoing
```

To test it, go to the browser and search for anything.



The website can't have accessed, since all the outgoing connections are rejected by the firewall, in order to allow all the outgoing connections again update with following code:

```
ufw default allow outgoing
```

Once this rule is set, one should be able to access the internet.

This was just one example; where default rules are set for outgoing connections as well similarly

Rules for ports and services

Once done with default rules, it's time to get our hands on rules for different ports and services, syntax for defining rules for port and services is

```
ufw action flow port|service
```

A. Action
allow, deny, reject, limit.

B. Flow
Here we specify the flow of traffic, these rules will be applicable for in: for incoming, out: for outgoing)

C. Port|service
Either specify port number or the service name *(for example: for ssh we can write 22 or ssh)*

UFW supports multiple applications, we can view these applications by writing:

```
ufw app list
```

Supported Applications:

- AIM
- Bonjour
- CIFS
- DNS
- Deluge
- IMAP
- IPP
- KTorrent
- Kerberos Full
- LDAP
- LPD
- MSN
- Mail Submission
- NFS
- Nginx
- OpenSSH
- POP3
- SMTP
- Samba
- Socks
- Telnet
- Transmission
- WWW
- Yahoo
- qBittorrent
- XMPP

In case the application you are trying to use a firewall for, is not in the list of supported applications you can simply mention the port number your service is running on. So, for example if you want to reject all the connections to your SSH serve, then write following command:

```
ufw allow 22
```

Or the service name if it's present in the list.

```
ufw allow ssh
```

In the above example we did not mention the flow of traffic, by default the rules will be applied for the incoming traffic, in order to apply rules for outgoing traffic, we explicitly need to mention the flow, for example:

```
ufw allow out ssh
```

Which tells the firewall to allow all the outgoing ssh connection.

Similarly, we can write a firewall for telnet, since telnet is unencrypted we want to reject all the incoming telnet connections, so to do that we write:

```
ufw reject in telnet comment "Telnet is unencrypted, rejecting connection! "
```

Once we add the rule, we can view the firewall rules by writing

```
ufw status
```

Which gives us something like this:

```
To                         Action      From
--                         ------      ----
23/tcp                     REJECT      Anywhere                   # Telnet is unencrypted , rejecting connection !
23/tcp (v6)                REJECT      Anywhere (v6)              # Telnet is unencrypted , rejecting connection !
```

Explanation:

To : The port number and the service
Action : The action to be performed
From : Means connection from any IP
V6 : Rule for IPv6 version

Command to delete rule:

```
ufw status numbered
```

Output to above command:

```
     To                         Action      From
     --                         ------      ----
[ 1] 23/tcp                     REJECT IN   Anywhere                  # Telnet is unencrypted , rejecting connection
!
[ 2] 23/tcp (v6)                REJECT IN   Anywhere (v6)             # Telnet is unencrypted , rejecting connection
!
```

The command shows us the number for each rule, which allows us to specify a rule number and delete it.

So for instance if you want to delete the telent rule for IPv6 , then write

```
ufw delete 2
```

Once our command is executed, we can check the status

```
To                         Action      From
--                         ------      ----
Anywhere                   REJECT      192.168.10.204

  #  Rejecting connections for 192.168.10.204
```

Which confirms that we have successfully deleted the rule

Note: For each rule deleted, the rule number changes, for example if a rule is deleted with number one, then the rule at number two will become the rule number one.

Rules for particular IP and Subnets

There might be cases when we want to block a particular IP, this might be applicable when some particular IP is trying to perform a brute force attack, or we might want to allow only particular IP's to access the SSH server, use cases are many, different needs different use cases, so let's learn how can we write rule for particular IP or a whole subnet.

Syntax:

```
ufw action flow IP | Subnet
```

A. Action
action to be performed (allow, deny, reject, limit)

B. Flow
incoming traffic or outgoing traffic (in, out)

C. IP | Subnet
Here we can specify either an IP address or an entire subnet (192.168.10.204 or 192.168.12.2/24)

So for example, if we want to reject connections for a particular IP (let's say 192.168.10.204) from accessing any port of our server, we can write

```
ufw reject in from 192.168.10.204 comment " Rejecting connections for 192.168.10.204 "
```

On checking the status

```
To                         Action      From
--                         ------      ----
Anywhere                   REJECT      192.168.10.204             #  Rejecting connections for 192.168.10.204
```

Now any connection from 192.168.10.204 to any port will be rejected.

In some cases, we might want to allow connections from a particular IP, for that we first need to reject all the incoming connections and then write a rule for the IP we want to allow connections from.

```
ufw default reject incoming
```

```
ufw allow in from 192.168.10.208
```

```
ufw status verbose
```

Output:

```
Status: active
Logging: on (low)
Default: reject (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
Anywhere                   ALLOW IN    192.168.10.208
```

ADVANCED COMMANDS

We have covered all the basic commands till now, but none of them seems much practical, we want to do things like, restrict an IP only to a particular port, detect brute force and block the IP, restrict IP's from accessing the ftp server but not web server, all these things will be covered under this sections, we'll write more specific and meaningful rules, which we can actually implement in real life.

Rules specific to IP's and services

Blocking an IP's access to all the ports might not be something that everyone wants, people might want to restrict an IP from accessing let's say any private service but not from the web server, in such cases write rules where we specify the IP address and the port number or service name.

```
ufw action flow from IP|subnet to any port port.number|service
```

A. Action
action to be performed (allow, deny, reject, limit)

B. Flow
incoming traffic or outgoing traffic (in, out)

C. IP|subnet
specifying the IP address or subnet

D. Port.number|service
specifying the port number or service name

Example A:
We want to write a rule which will allow only host 192.168.10.207 to access the SSH server, now to solve this problem we need to do two things

1. Block SSH service for all the IP addresses.
2. Allow SSH service only for host 192.168.10.207

```
ufw reject in ssh
```

```
ufw allow in from 192.168.10.207 to any port ssh comment "SSH allowed"
```

Output:

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     REJECT      Anywhere
22/tcp                     ALLOW       192.168.10.207             # SSH allowed
22/tcp (v6)                REJECT      Anywhere (v6)
```

Note: The sequence of rule matters in UFW, for example: if there's a rule for blocking an IP address from accessing a web server at position 2, but there's another rule which allows access to web server at position 1, then the IP address will not be blocked, the firewall rules are applied in a sequence. We might need to add important rules at position 1, for that we use insert 1.

```
ufw insert 1 reject in from <IP> to any port <port.number>
```

By specifying insert, this rule will be added at position 1.

Example B:
Let's suppose we need to reject any UDP connection from an IP address (192.168.10.207)

```
ufw reject in from 192.168.10.207 to any proto udp
```

Proto is used to specify the transmission protocol (TCP or UDP).

Similarly, we can write rules for other services and protocols depending upon the usability.

Writing rules for specific interfaces

There might be some cases where, the user wants to allow access to a particular service from a particular subnet, in such cases UFW allows us to set up interface dependent rules

Example:
Allow access to ssh server to all the IP addresses on network interface eth0.

To make sure that other IP addresses are not able to access the service on port 22

```
ufw reject in ssh
```

Another rule to allow ssh access to all the IP addresses on interface eth0

```
ufw allow in on eth0 to any port 22
```

Logging Rules

When we talk about logging information about connections, it's a very important concept to understand the type of incoming connections, logging information about incoming/outgoing connections can help a system administrator, to trace back any attack to its source, logs also help system administrator to check if all the firewall rules are working properly or not.

UFW provides us the ability to log, in addition to that it also provides different logging levels depending on the need, logging in UFW is enabled by default at low level. To see the logging level, we can write

```
ufw status verbose
```

```
Status: active
Logging: on (low)
Default: allow (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

We can see that logging is on and is set to low level.
UFW provides four logging levels:

A. Low:
   a. Logs blocked packets (with rate limiting)
   b. Logs packet matching logged rules

B. Medium:
   a. Logs Low level logs
   b. Logs all allowed packets not matching with policy
   c. Logs all invalid packets
   d. Logs all new connections (all the logging is done with rate limiting)

C. High:
   a. Logs medium level without rate limiting
   b. Logs all the other packets with rate limiting

D. Full:
   a. Logs high level without rate limiting

To change the log levels or to turn it off or on, we write:

```
ufw logging option
```

Option can have any of the following values

1. On
2. Off
3. Low
4. Medium
5. High
6. Full

So for example: UFW logging if off by default in my system and you need to turn it on at medium level, so command is as follow:

```
ufw logging medium
```

```
Status: active
Logging: on (medium)
Default: allow (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

Reading logs

Reading logs is very important as we have discussed earlier, they help us to detect any threat, test our firewall and other important tasks.
Logs for UFW are stored in file /var/logs/ufw.log

```
Oct 31 16:29:53 FRIEND kernel: [ 3253.426605] [UFW BLOCK] IN= OUT=eth0 SRC=SIP DST=DIP
LEN=73 TOS=0x00 PREC=0x00 TTL=64 ID=61481 DF PROTO=UDP SPT=45316 DPT=53 LEN=53
```

Here we have one of the many logs from ufw.log file, so let's break it down and understand what exactly does it mean

1. IN : If this contains a value means the event was incoming
2. OUT : If this contains a value means the event was outgoing (interface specified)
3. SRC : Source IP address
4. DST : Destination IP address
5. LEN : Packet length
6. TTL : Time to lives
7. PROTO : Packet's protocol
8. SPT : Source port
9. DPT : Destination port
10. WINDOW : The packet size , that sender can receive (not present in the above example , since it's a low level log)

PROBLEMS WITH FIREWALL

Even though firewalls are first line of defence, but they can't protect us against every type of attack, there are certain places where firewall lack, things like

1. A firewall cannot protect against internal network attacks.
2. They do not protect us against backdoor attack.
3. Firewalls cannot protect a network or pc from viruses, Trojans, worms and spyware which spread through flash drives, potable hard disk and floppy etc.
4. Firewalls cannot protect against IP spoofing

As we already know that firewall do have their own weaknesses, but that doesn't mean they are of no importance, a firewall should definitely be a part of defence but it shouldn't be the only part, we should never rely on just firewalls, other software's like IDS (intrusion detection system) and IPS (intrusion prevention system), should be used in combination with firewall.

LAB EXERCISE

Aim:

To implement whatever, learnt so far almost everything.

Prerequisites

1. A host machine with VMware or Virtual Box installed.
2. Kali Linux as virtual machine.
3. Internet Connection.

Recommendations

1. Configure and run Apache2 server on Kali Linux machine.
2. Configure and run SSH on Kali Linux machine.

Procedure

1. So the Kali machine is the machine, run UFW on and our host machine is our client machine.
2. The task is to write different rules based upon the tasks which are given.
3. Make sure that both apache2 and ssh services are running.

Tasks

1. Write a rule to change the ufw default settings for outgoing connections to reject.
2. Write a rule to block all the incoming connections and see if you are able to connect or not.
3. Write a rule to block the host machine from accessing ssh service but not web server.
4. Write a rule which will protect against brute force attacks.
5. Write a rule to block all the UDP connections from host IP.
6. Write a rule to log in high mode and read the logs generated.
7. Write a rule to allow access to web server only from a particular host.
8. What are some disadvantages of firewall?
9. Write a rule with action as deny and another rule with action as reject, analyse the difference between both.

REFERENCES

https://help.ubuntu.com/community/UFW

https://wiki.ubuntu.com/UncomplicatedFirewall

https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-18-04

https://www.linode.com/docs/guides/configure-firewall-with-ufw/

https://youtu.be/-CzvPjZ9hp8

www.safe.security | info@safe.security

Standford Research Park,
3260 Hillview Avenue,
Palo Alto, CA - 94304

SAFE
SECURITY