**Uncomplicated Firewall (UFW)**

This project aims to explore the Uncomplicated Firewall (UFW) in a Debian-based Linux environment. UFW is a user-friendly front-end for managing iptables firewall rules. By properly configuring the firewall and creating rules carefully, we can help limit an organization's attack surface and significantly reduce the risk of network intrusion incidents.

Firewalls act as a first line of defense against external threats, malware, and hackers attempting to gain unauthorized access to data and systems. They monitor network traffic, providing the ability to control system entry points, prevent virus attacks, and protect against data theft.

UFW is pre-installed in Debian-based Linux distributions. It provides a simple syntax for managing firewall rules, making it easier for administrators to configure and maintain their firewall.

The UFW syntax rule is directly related to iptables. UFW simplifies the process of managing iptables rules by using a more user-friendly command-line interface. For example, to allow incoming SSH connections, you would use the following UFW command: $ **sudo ufw allow ssh.** Uncomplicated right?

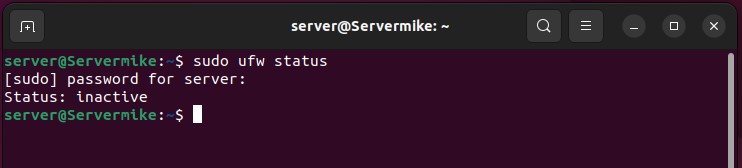
It is important to configure UFW rules before enabling the firewall to prevent the possibility of being locked out when working with a remote server. By default, the UFW firewall denies all incoming traffic and allows all outgoing traffic, which means that if you accidentally block all incoming traffic to the server you are configuring, you may lose access to the server.

To avoid this situation, it is recommended that you first set up your firewall rules to allow necessary incoming traffic, such as SSH connections, before enabling the UFW firewall. This way, you can ensure that you can still access your server after enabling the firewall.

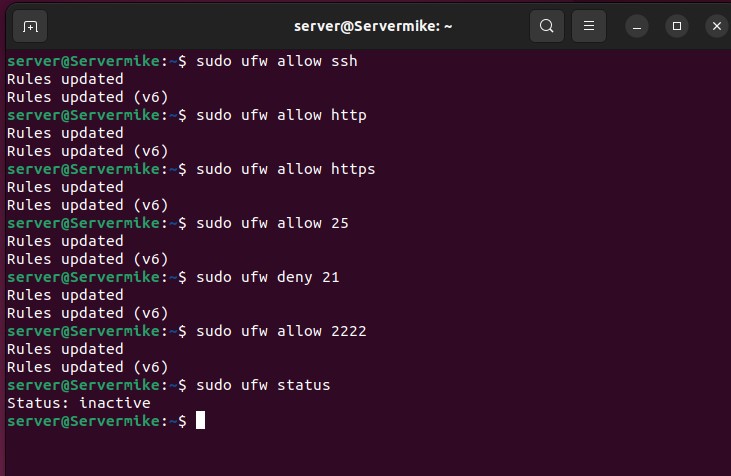
To know in more extensive detail, you can use the man page ($ man ufw ) to learn more about the Uncomplicated firewall, it options and syntax rule. For this project let’s check the UFW status to know if it’s enabled and running or if it’s not running so we can set up a few rules then enable the firewall and walk around its syntax a little.

Here are the steps to set up UFW rules before enabling the firewall:

1. Check the current UFW status:

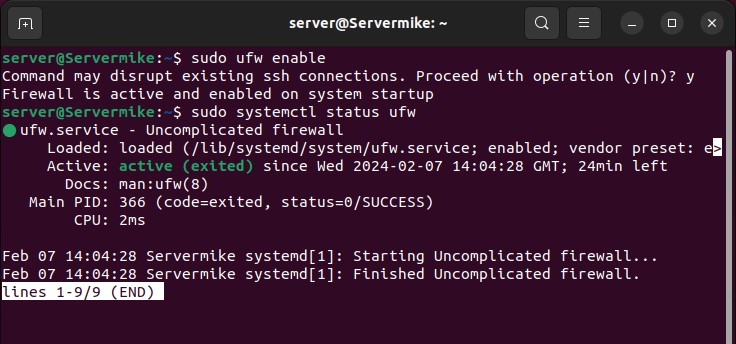


Firewall rules are customized based on the specific needs and policies of each organization. In this project, we will create a few rules for demonstration purposes only. These rules will allow traffic to common services such as SSH, HTTP, HTTPS, and SMTP, while also denying traffic to certain ports as necessary.

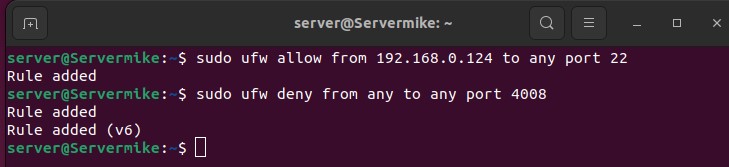


The first command enable SSH connection from any IP address and same as others while some deny access to certain ports (FTP in this example), the others allow access to port specified by either port number or protocol name. Ufw allow for long syntax, say we are trying to allow access to a particular port from a particular IP-address or say we are trying to deny access to a specific port from any IP-address; for example, we are trying to allow connection from only our IP-address to be able to make SSH connection to the server:

First let’s enable the UFW firewall:

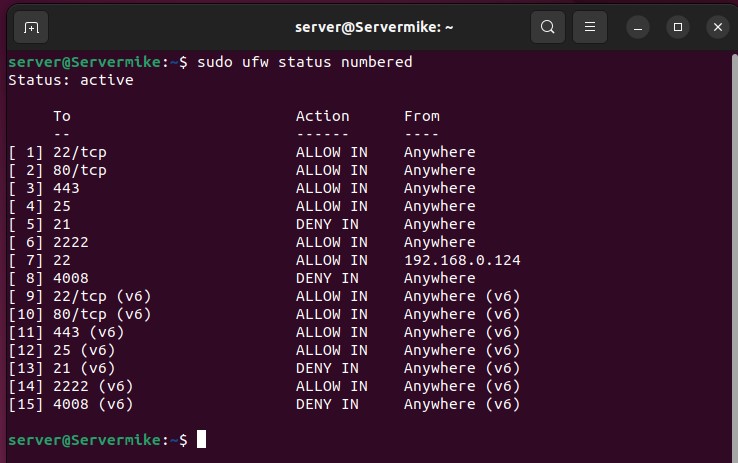


Let’s imagine we are trying to do two things allow access from a particular IP address on a specified port and also trying to deny access from any IP address to a particular port:

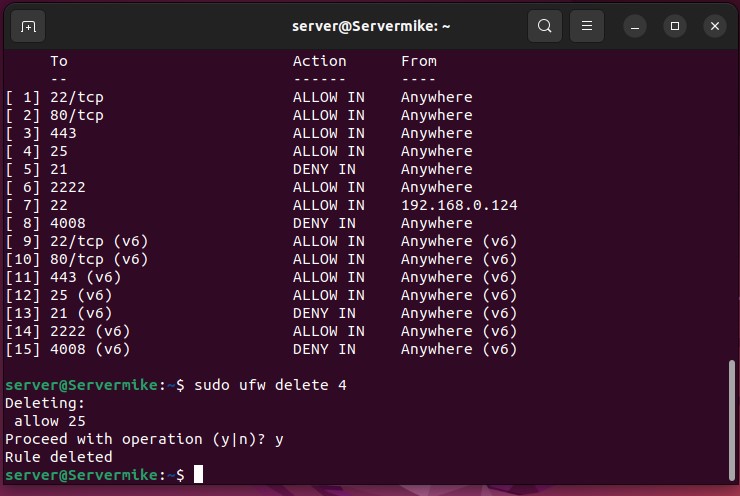


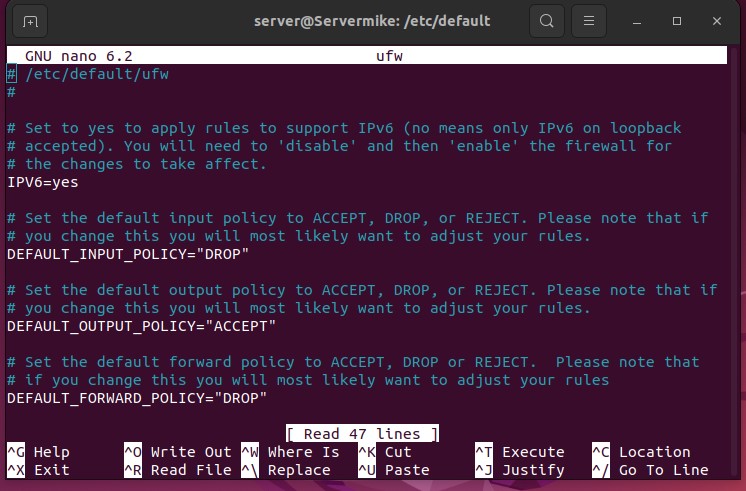
The first rule allow all incoming traffic from the specified IP-address “192.168.x.x” only to port 22 and the second rule deny any incoming traffic from any IP-address to the specified port.

To see our specified rules so far we can use the “$ sudo ufw status” command to see what and how many rules we have. Also, we can use “$ sudo ufw status numbered” to number the rules that we have, this makes it easier when we want to delete this rules.



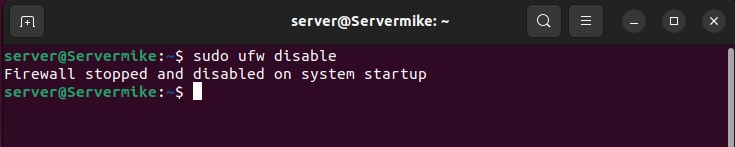
Now let’s learn to delete rules, to delete a particular rule we first number it out like above and target the specific rule we want to delete. For example;



To reset the UFW firewall, we use the command “$ sudo ufw reset”. The UFW firewall configuration file; $ sudo nano /etc/default/ufw:

In the configuration file, we can choose to configure it according to our specific organization needs and best practice. We can choose to not allow IPv6 or change our defaults options in the configuration. For example, we can change our default\_input\_policy = “ACCEPT” and many more things and also be familiar with the man page to keep learning.

To disable the ufw firewall we can use the comman, $ sudo ufw disable:



In conclusion, this project on security hardening has provided us with a deeper understanding of the importance and benefits of firewalls as a key element in strengthening the overall security of an organization. By implementing and configuring firewalls, we have learned how they can help limit the total attack surface of a system, allowing only necessary connections and ensuring that systems and servers are secure. NGFWs enable deeper application-level security, multi-layered protection, and real-time adaptability to the changing threat landscape. By integrating various security measures into one seamless product, NGFWs streamline network security and enable more effective decision-making processes for controlling data flow.

Overall, this project has helped me learn the importance of firewalls in security hardening and provided valuable insights into the benefits and capabilities of NGFWs. As cybersecurity threats continue to evolve and increase in volume and sophistication, understanding and implementing firewall solutions is essential for securing and protecting digital infrastructure.