Zentry

Enhancing Network Security with GUFW: A Usability & Configuration Guide

Introduction

In today’s cyber security landscape, firewalls are a critical aspect of protecting your assets from unauthorized access and malicious traffic. Tools like iptables and GUFW were made to accommodate those needs, but can sometimes present a steep learning curve for beginners or non-tech savvy users. This guide will go over what firewalls are and what they do, how to navigate the GUFW on your Linux system as well as a step-by-step guide on how to set it up to your own preferences.

About Us

Zentry is a specialized cybersecurity company dedicated to making network protection simple and effective. We provide user-friendly solutions that safeguard systems from unauthorized access and evolving cyber threats.

Charles Bouvay
Jamie Nguyen
Eric Liu

What are firewalls?

A firewall is a smart gate keeper on your linux system. Every time information tries to come in or out to the internet from your device, the firewall checks it against a list of rules and if it’s safe, network traffic freely flows and if it doesn't meet the criteria, the firewall blocks it. Basically, a firewall is a security tool that runs in the background quietly and monitors and controls incoming and outgoing network traffic based on predefined rules.

To explain network traffic, whenever your computer sends or receives data over the internet or network, it's called network traffic and there are two kinds of network traffic; outbound and inbound traffic.

Outbound traffic initiates traffic to the outside. For example, when you open a browser and visit a website, send an email, or run a system update. This is usually safe as you are the initiator. Inbound traffic is when another computer tries to connect to yours. For example, if someone was trying to connect to your system via ssh, a multiplayer game server sending match data to your system, or a Hacker scanning your system for open ports. This is where the danger comes from.

We don't even like it when we get unsolicited calls on our phone. We are definitely not going to like unsolicited incoming calls to our computer. This is where we use a firewall to block unsolicited incoming connections.

Why use a GUI?

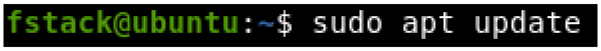
Firewalls can be complex and complicated for the uninitiated but a tool called UFW, also known as Uncomplicated FireWall, is installed on some Linux systems and if it's not, GUFW can be installed on any linux system

Why might someone might not use a GUI for linux

In an enterprise environment, it is important to keep your installed linux system very lean with as few software installed as possible. Iptables is already pre-installed and built in on most linux systems and ip tables is used by many sys admins especially for linux just because all these things like iptables are already built into linux and so UFW can be cumbersome as with any gui based product. To most sysadmins, it makes more sense to communicate with iptables which are already built into the system directly than having to understand and download a whole other different program.

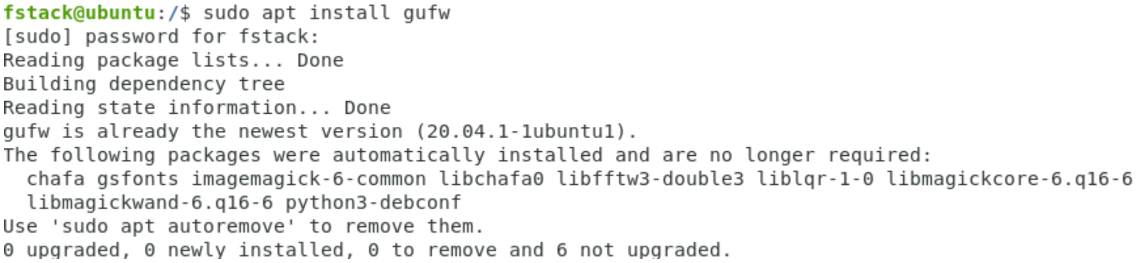
Prerequisite (Before Installing GUFW)

1. sudo apt update

   ```
   fstack@ubuntu:~$ sudo apt update
   ```

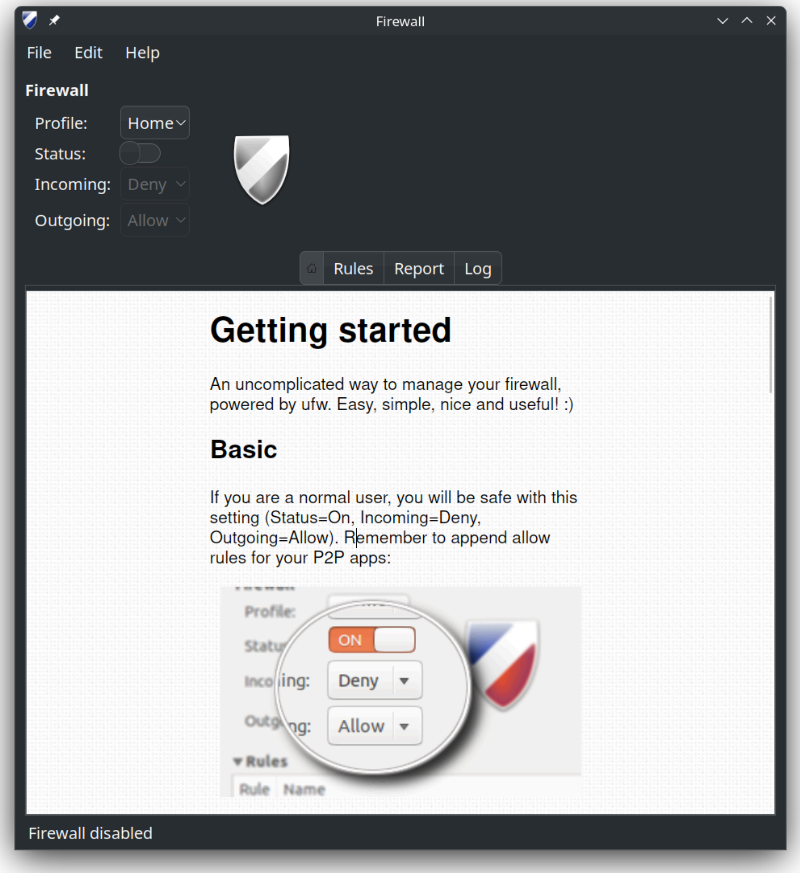
2. sudo apt install gufw

   ```
   fstack@ubuntu:/$ sudo apt install gufw
   [sudo] password for fstack:
   Reading package lists... Done
   Building dependency tree
   Reading state information... Done
   gufw is already the newest version (20.04.1-1ubuntu1).
   The following packages were automatically installed and are no longer required:
     chafa gsfonts imagemagick-6-common libchafa0 libfftw3-double3 liblqr-1-0 libmagickcore-6.q16-6
     libmagickwand-6.q16-6 python3-debconf
   Use 'sudo apt autoremove' to remove them.
   0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
   ```

3. You can open up GUFW by using the command gufw in the terminal, navigating through the gui, or searching for “firewall configurations” in your application menu.

The image below is what the GUFW should look like.



Firewall

Profile: Home

Status:

Incoming: Deny

Outgoing: Allow

Getting started

An uncomplicated way to manage your firewall, powered by ufw. Easy, simple, nice and useful! :)

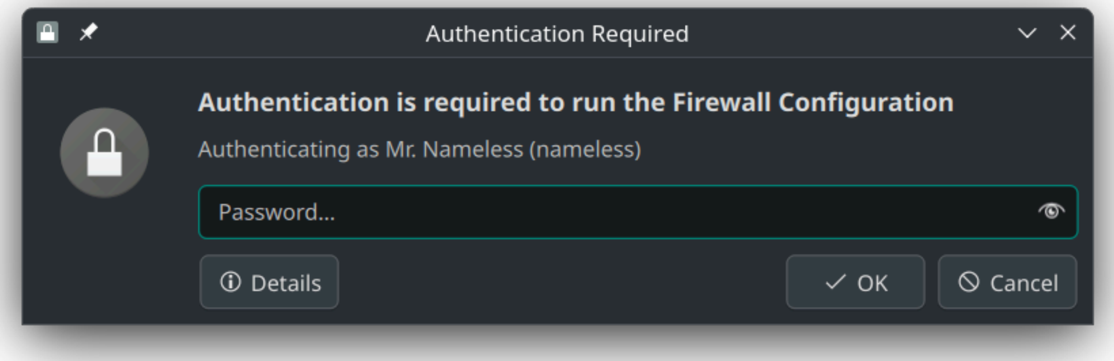
Basic

If you are a normal user, you will be safe with this setting (Status=On, Incoming=Deny, Outgoing=Allow). Remember to append allow rules for your P2P apps:

Firewall disabled

Basically it's a gui for configuring iptables which is the more complex underlying linux firewall, GUFW gives you a simple intuitive way to manage your firewall without the command line.

When launched, you will be prompted to enter your password because it requires admin privileges.



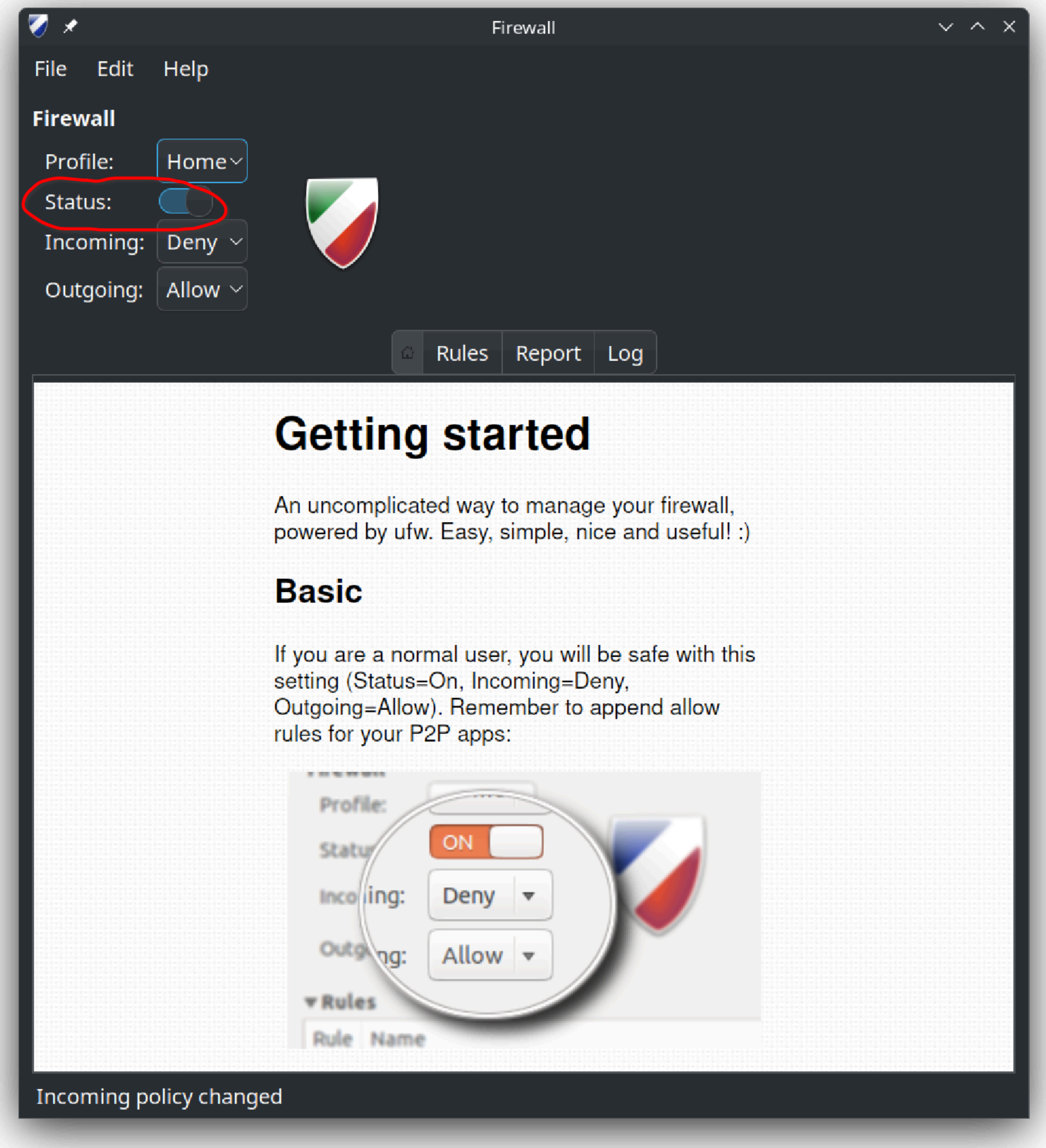
Authentication Required

Authentication is required to run the Firewall Configuration

Authenticating as Mr. Nameless (nameless)

Password...

Toggle the standard switch on and that's it. You do not need to do anything else as the firewall is now active and running on your computer and your device is now protected from incoming assaults.



Firewall

Profile: Home

Status:

Incoming: Deny

Outgoing: Allow

Getting started

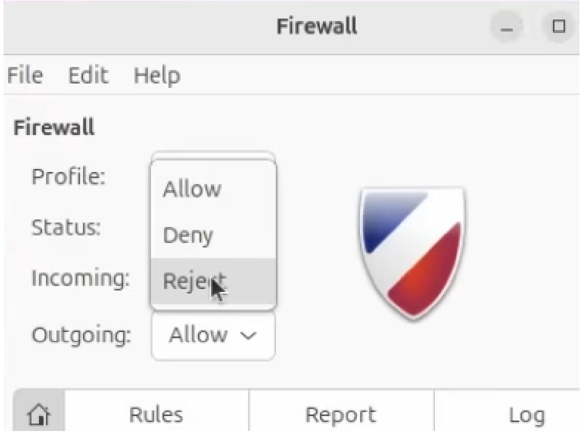
An uncomplicated way to manage your firewall, powered by ufw. Easy, simple, nice and useful! :)

Basic

If you are a normal user, you will be safe with this setting (Status=On, Incoming=Deny, Outgoing=Allow). Remember to append allow rules for your P2P apps:

Incoming policy changed

GUFW gives you 4 different actions that you can configure it to take based on predefined firewall rules.



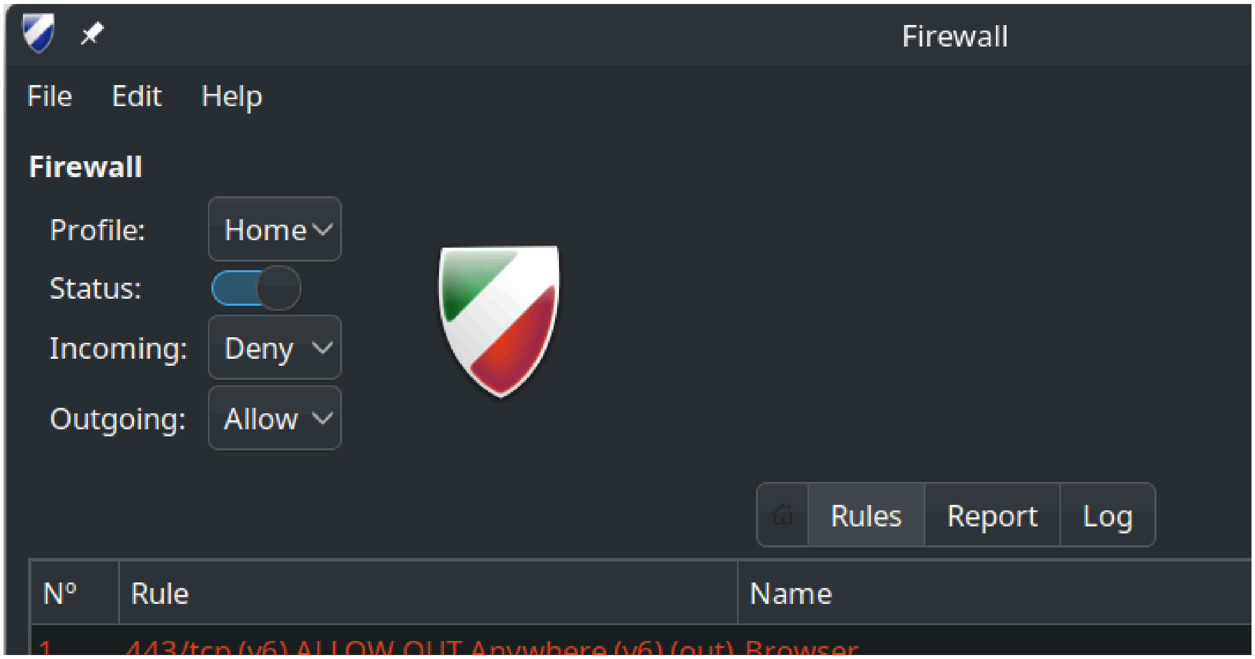
Profile:

Status:

Incoming:

Outgoing: Allow

Allow allows your connection to go through.



Profile: Home

Status:

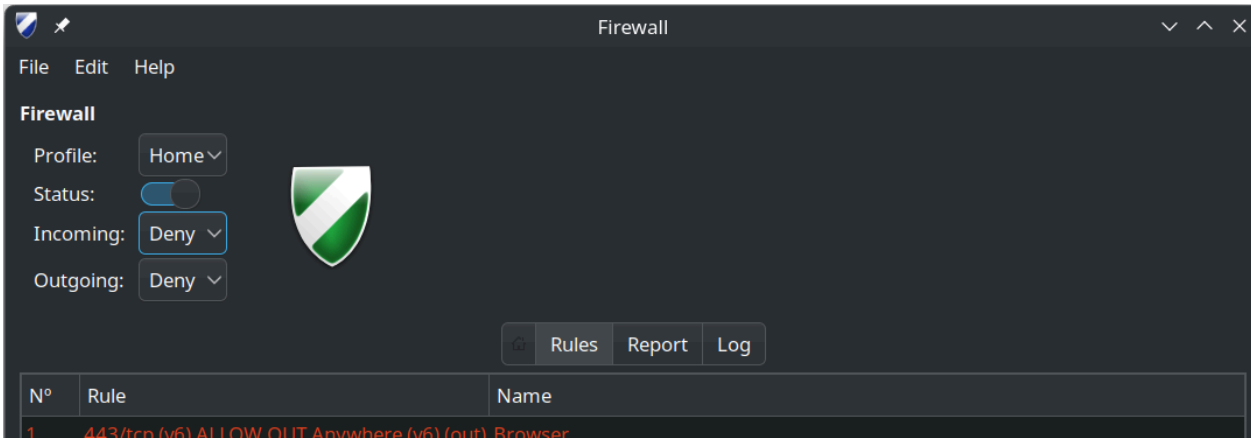
Incoming: Deny

Outgoing: Allow

Nº	Rule	Name
1	443/tcp (v6) ALLOW OUT Anywhere (v6) (out) Browser	

*Above is a screenshot of allowing traffic out while still denying incoming traffic

Outgoing connections are allowed because when you want to connect to the internet, the firewall lets your request through. Deny will deny your request.



Firewall

Profile: Home

Status:

Incoming: Deny

Outgoing: Deny

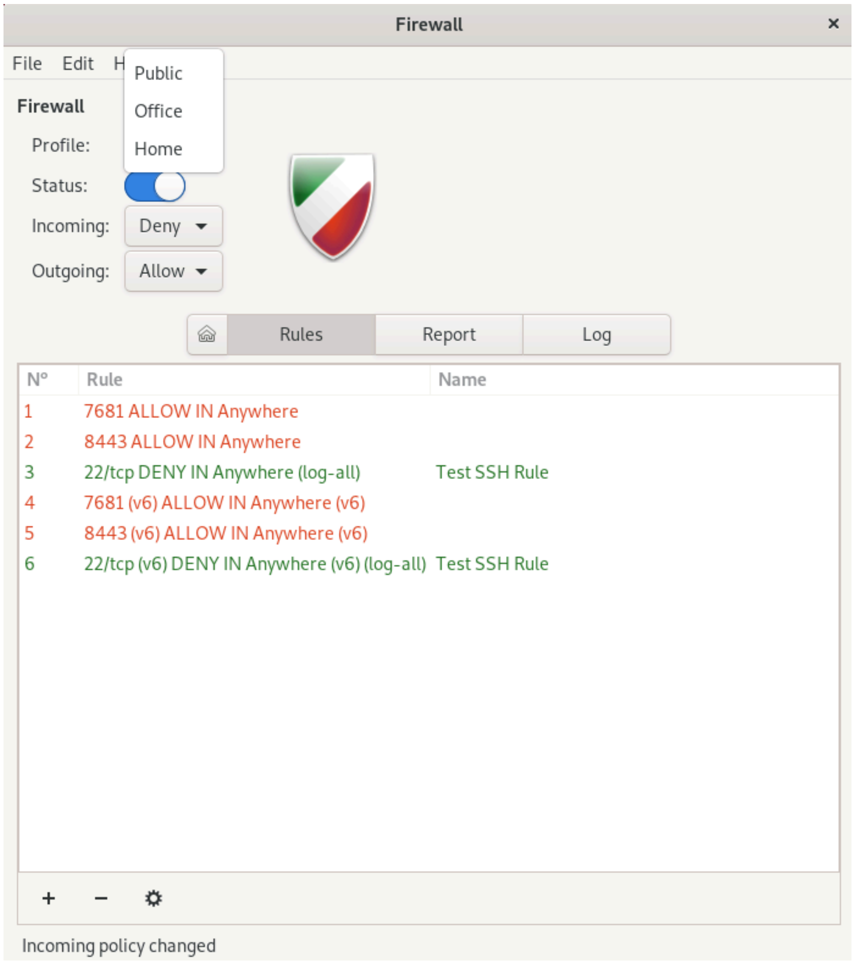
N°	Rule	Name
1	443/tcp (v6) ALLOW OUT Anywhere (v6) (out)	Browser

So if someone tries to connect to your computer from outside the firewall, the firewall will deny that request.

The last option is to reject. Using this option will prevent the outside actor from connecting to your computer, but it will let them know that the connection has been rejected. You generally don't want to use this. You will want to use deny which will just stop and disregard the request. In most cases, this is better than using reject. There's also a limit action that is used to either throttle or reject connections after too many requests in a short period of time. This is especially useful to protect your system against brute force attacks like when someone is trying to connect to your computer and break a password, but we won't use this as we already set incoming requests to deny.

Home, Public, and Office Profiles

GUFW by default has three profiles, Home, Public, and Office, with all three having different security properties. Each profile can be toggled by navigating to the top left and selecting which profile you want. These are quick predefined rule sets that allow you to quickly switch between different sets of rules depending on your environment where you're connected to the home wifi, office work wifi or public wifi. Keep note that all 3 profiles can be customized to your own preferences.



Firewall

Profile:

Status:

Incoming: Deny

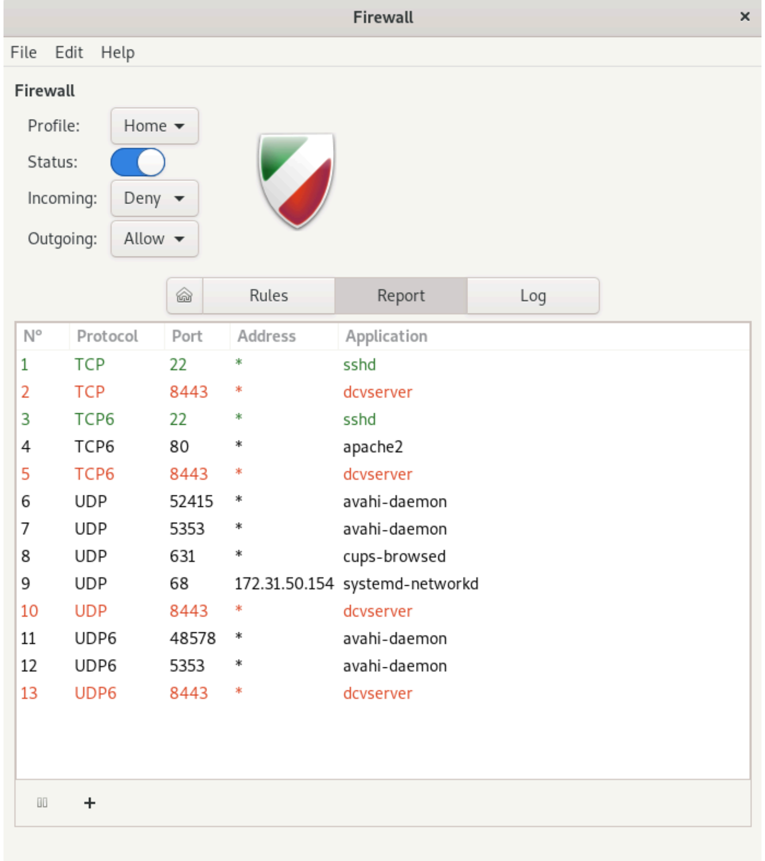
Outgoing: Allow

N°	Rule	Name
1	7681 ALLOW IN Anywhere	
2	8443 ALLOW IN Anywhere	
3	22/tcp DENY IN Anywhere (log-all)	Test SSH Rule
4	7681 (v6) ALLOW IN Anywhere (v6)	
5	8443 (v6) ALLOW IN Anywhere (v6)	
6	22/tcp (v6) DENY IN Anywhere (v6) (log-all)	Test SSH Rule

Incoming policy changed

Report and Log

The second tab in GUFW is the Report tab, and this tab gives a summary view of your firewall's recent activity, typically showing the most frequent source or destination IP addresses, ports with the most traffic, and just overall great for spotting any patterns, like multiple attempted access from specific IPs.



Firewall

Profile: Home

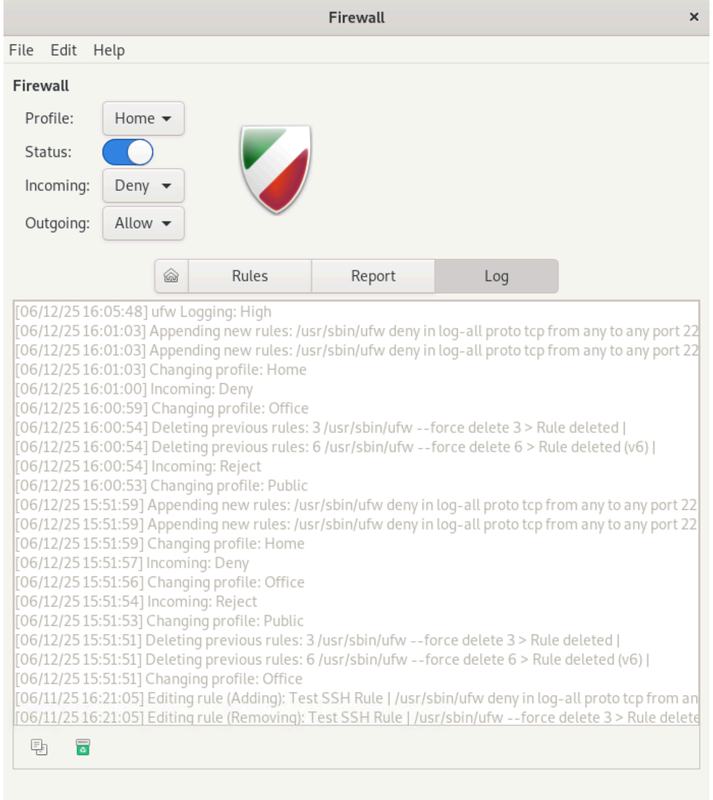
Status:

Incoming: Deny

Outgoing: Allow

N°	Protocol	Port	Address	Application
1	TCP	22	*	sshd
2	TCP	8443	*	dcvserver
3	TCP6	22	*	sshd
4	TCP6	80	*	apache2
5	TCP6	8443	*	dcvserver
6	UDP	52415	*	avahi-daemon
7	UDP	5353	*	avahi-daemon
8	UDP	631	*	cups-browsed
9	UDP	68	172.31.50.154	systemd-networkd
10	UDP	8443	*	dcvserver
11	UDP6	48578	*	avahi-daemon
12	UDP6	5353	*	avahi-daemon
13	UDP6	8443	*	dcvserver

The last tab is the Log tab, which shows real-time firewall activity, showing all packets that were blocked or allowed by your firewall. It specifically shows the timestamp, the source or destination IP, the port and protocol, the direction of traffic, and the action that was taken by the firewall. Logs are great for identifying any suspicious activity as well as testing your firewall rules if they're working.



Firewall

Profile: Home

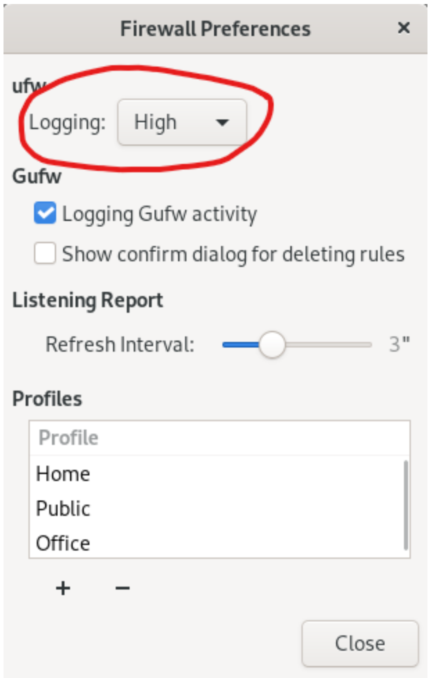
Status: On

Incoming: Deny

Outgoing: Allow

```
[06/12/25 16:05:48] ufw Logging: High
[06/12/25 16:01:03] Appending new rules: /usr/sbin/ufw deny in log-all proto tcp from any to any port 22
[06/12/25 16:01:03] Appending new rules: /usr/sbin/ufw deny in log-all proto tcp from any to any port 22
[06/12/25 16:01:03] Changing profile: Home
[06/12/25 16:01:00] Incoming: Deny
[06/12/25 16:00:59] Changing profile: Office
[06/12/25 16:00:54] Deleting previous rules: 3 /usr/sbin/ufw --force delete 3 > Rule deleted |
[06/12/25 16:00:54] Deleting previous rules: 6 /usr/sbin/ufw --force delete 6 > Rule deleted (v6) |
[06/12/25 16:00:54] Incoming: Reject
[06/12/25 16:00:53] Changing profile: Public
[06/12/25 15:51:59] Appending new rules: /usr/sbin/ufw deny in log-all proto tcp from any to any port 22
[06/12/25 15:51:59] Appending new rules: /usr/sbin/ufw deny in log-all proto tcp from any to any port 22
[06/12/25 15:51:59] Changing profile: Home
[06/12/25 15:51:57] Incoming: Deny
[06/12/25 15:51:56] Changing profile: Office
[06/12/25 15:51:54] Incoming: Reject
[06/12/25 15:51:53] Changing profile: Public
[06/12/25 15:51:51] Deleting previous rules: 3 /usr/sbin/ufw --force delete 3 > Rule deleted |
[06/12/25 15:51:51] Deleting previous rules: 6 /usr/sbin/ufw --force delete 6 > Rule deleted (v6) |
[06/12/25 15:51:51] Changing profile: Office
[06/11/25 16:21:05] Editing rule (Adding): Test SSH Rule | /usr/sbin/ufw deny in log-all proto tcp from an
[06/11/25 16:21:05] Editing rule (Removing): Test SSH Rule | /usr/sbin/ufw --force delete 3 > Rule delete
```

Note that if nothing is showing in the logs, make sure logging is enabled. You can check this by clicking Edit in the top left and then preferences. Another window will open and top will have the option Logging; make sure it is not off.



ufw

Logging: High

Gufw

☑ Logging Gufw activity

□ Show confirm dialog for deleting rules

Listening Report

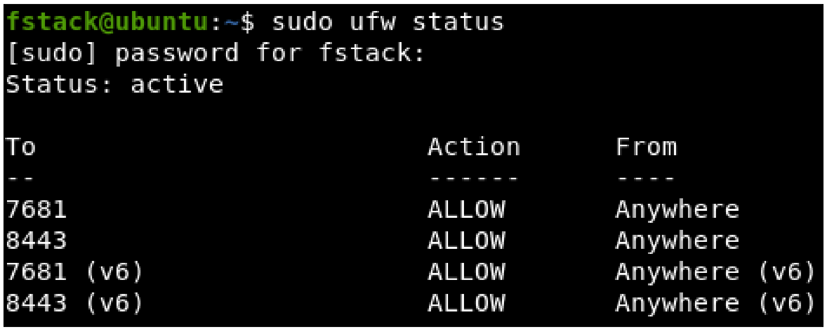
Refresh Interval: 3"

Profiles

Profile
Home
Public
Office

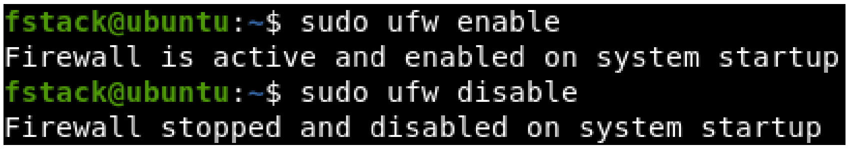
Now that the firewall is up and running, all incoming request will be automatically blocked but you don't have to necessarily want to block all the incoming traffic, If you remote into your system from another device, you want to make a rule exception for ssh or if you're running a server on your computer you want to allow incoming HTTPs. and certain multiplayer games would also require that you explicitly allow the incoming traffic, if you ssh into your computer and you set up a firewall on that computer You need to set up a rule to allow ssh here or you're going to log yourself out.

If you want to check if your firewall is running correctly, you can use the command sudo ufw status in your terminal for the firewall status.



```
fstack@ubuntu:~$ sudo ufw status
[sudo] password for fstack:
Status: active

To                         Action      From
--                         ------      ----
7681                       ALLOW       Anywhere
8443                       ALLOW       Anywhere
7681 (v6)                  ALLOW       Anywhere (v6)
8443 (v6)                  ALLOW       Anywhere (v6)
```

You can also disable the firewall by toggling off the status in gui or running sudo ufw disable, and then to enable it is sudo ufw enable.

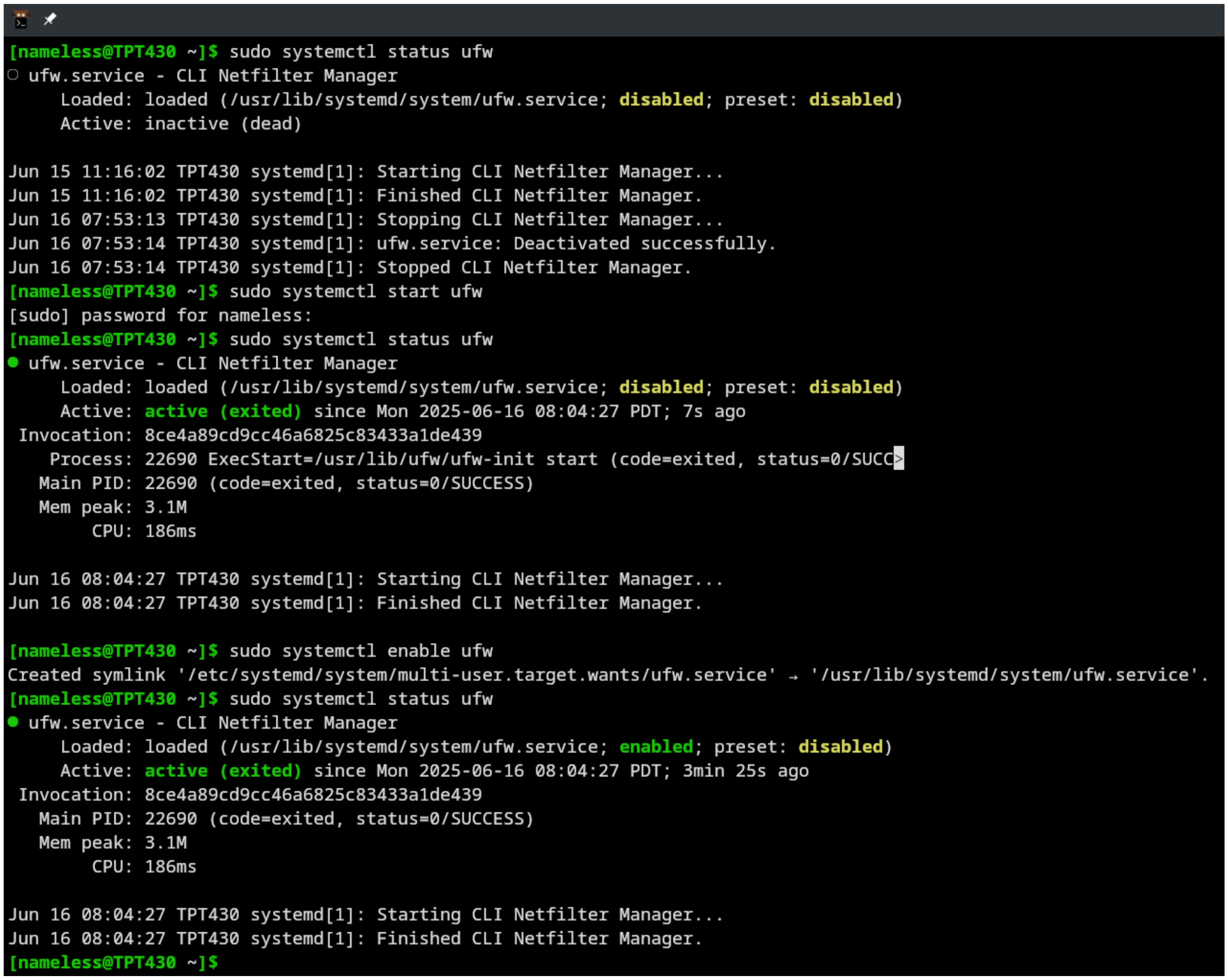


```
fstack@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
fstack@ubuntu:~$ sudo ufw disable
Firewall stopped and disabled on system startup
```

That's it you've now successfully set up a firewall to protect your linux system from unwanted incoming connections. This is a lightweight yet powerful technique that provides robust security with minimal hazard. But I do want to finish off by saying that a firewall is not a magic shield, just because you've enabled a firewall doesn't mean that your system is invincible. It does protect you from unsolicited incoming attacks, but it won't protect you from malicious websites, phishing emails, and unsafe downloads, as the firewall does not inspect what you willingly accept and let in. If you click on shady links, download unknown software, run unknown scripts you might bypass the firewall yourself, so be careful and use sound judgement.

Start UFW Automatically

To start UFW automatically on startup linux uses daemon services in the form of systemctl.



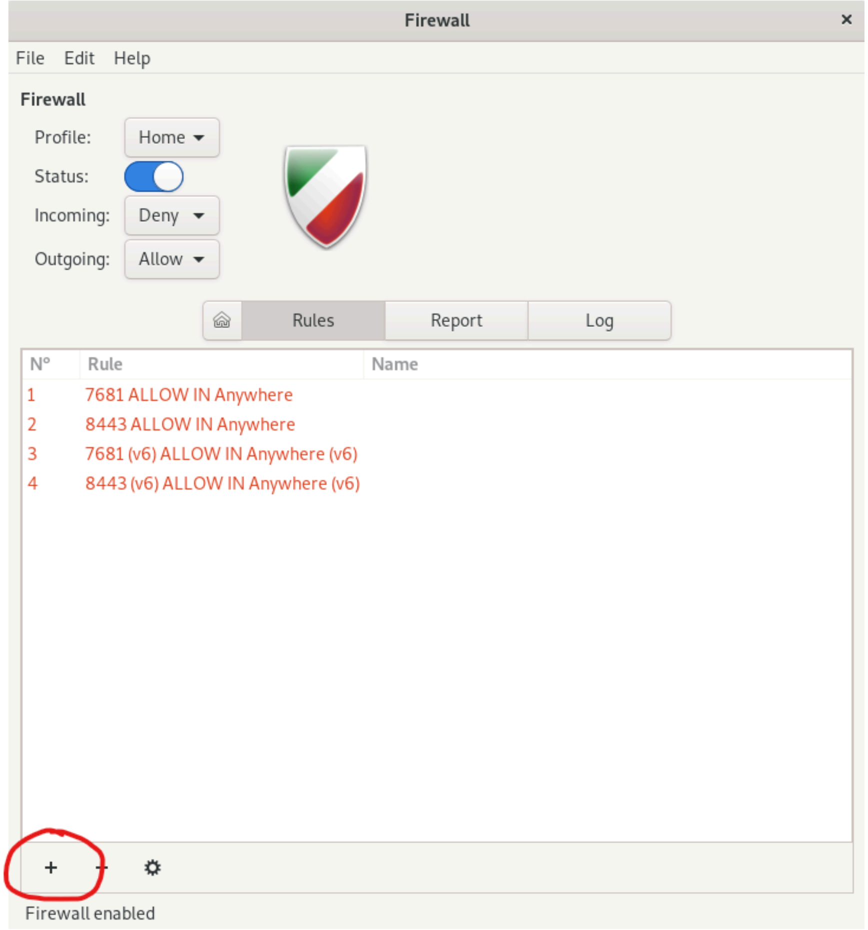
```
[nameless@TPT430 ~]$ sudo systemctl status ufw
○ ufw.service - CLI Netfilter Manager
     Loaded: loaded (/usr/lib/systemd/system/ufw.service; disabled; preset: disabled)
     Active: inactive (dead)

Jun 15 11:16:02 TPT430 systemd[1]: Starting CLI Netfilter Manager...
Jun 15 11:16:02 TPT430 systemd[1]: Finished CLI Netfilter Manager.
Jun 16 07:53:13 TPT430 systemd[1]: Stopping CLI Netfilter Manager...
Jun 16 07:53:14 TPT430 systemd[1]: ufw.service: Deactivated successfully.
Jun 16 07:53:14 TPT430 systemd[1]: Stopped CLI Netfilter Manager.
[nameless@TPT430 ~]$ sudo systemctl start ufw
[sudo] password for nameless:
[nameless@TPT430 ~]$ sudo systemctl status ufw
● ufw.service - CLI Netfilter Manager
     Loaded: loaded (/usr/lib/systemd/system/ufw.service; disabled; preset: disabled)
     Active: active (exited) since Mon 2025-06-16 08:04:27 PDT; 7s ago
 Invocation: 8ce4a89cd9cc46a6825c83433a1de439
    Process: 22690 ExecStart=/usr/lib/ufw/ufw-init start (code=exited, status=0/SUCC>
   Main PID: 22690 (code=exited, status=0/SUCCESS)
   Mem peak: 3.1M
        CPU: 186ms

Jun 16 08:04:27 TPT430 systemd[1]: Starting CLI Netfilter Manager...
Jun 16 08:04:27 TPT430 systemd[1]: Finished CLI Netfilter Manager.

[nameless@TPT430 ~]$ sudo systemctl enable ufw
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' → '/usr/lib/systemd/system/ufw.service'.
[nameless@TPT430 ~]$ sudo systemctl status ufw
● ufw.service - CLI Netfilter Manager
     Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset: disabled)
     Active: active (exited) since Mon 2025-06-16 08:04:27 PDT; 3min 25s ago
 Invocation: 8ce4a89cd9cc46a6825c83433a1de439
   Main PID: 22690 (code=exited, status=0/SUCCESS)
   Mem peak: 3.1M
        CPU: 186ms

Jun 16 08:04:27 TPT430 systemd[1]: Starting CLI Netfilter Manager...
Jun 16 08:04:27 TPT430 systemd[1]: Finished CLI Netfilter Manager.
[nameless@TPT430 ~]$
```

1. sudo systemctl status ufw to check status
2. sudo systemctl start ufw to start UFW as a daemon service
3. sudo systemctl enable ufw to enable UFW as a permanent service on startup
4. sudo systemctl status ufw to check status of daemon firewall service

And now UFW will always start on boot.

How to set up rules in GUFW

Setting up rules in GUFW is really simple; start by navigating to the Rules tab and clicking the + sign on the bottom left.



Firewall

Profile: Home

Status: ●

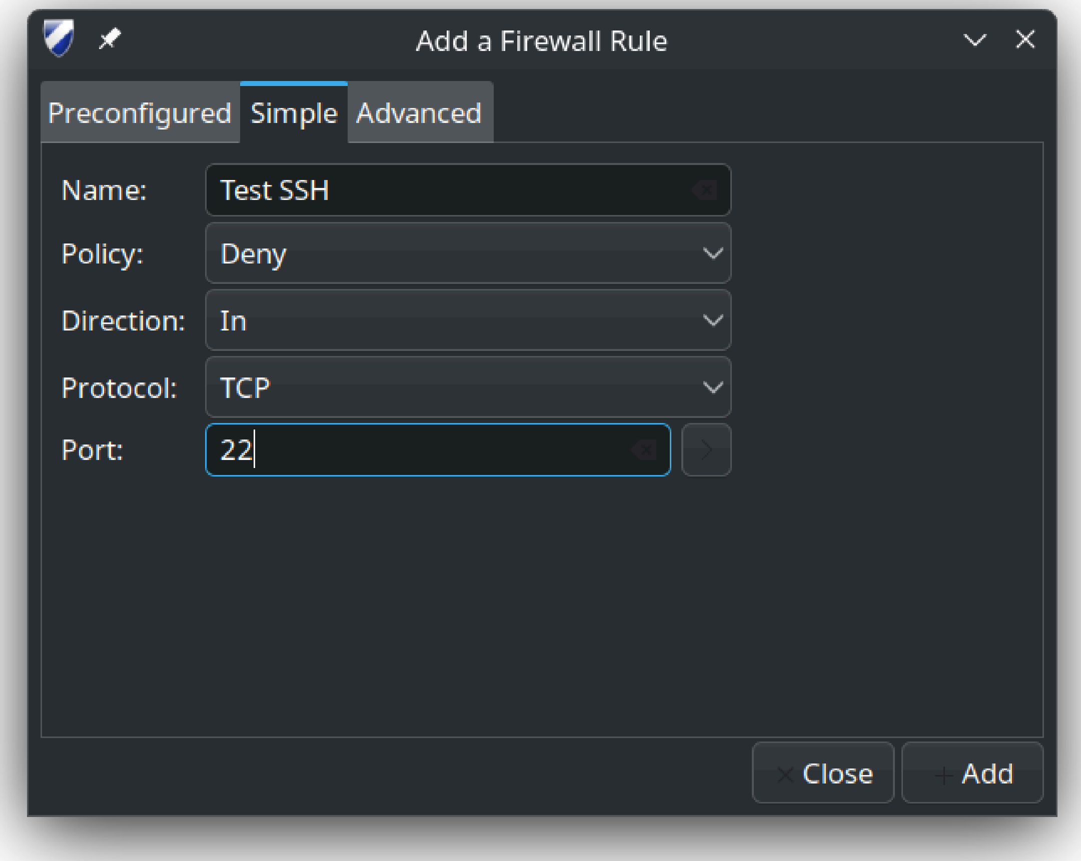
Incoming: Deny

Outgoing: Allow

N°	Rule	Name
1	7681 ALLOW IN Anywhere	
2	8443 ALLOW IN Anywhere	
3	7681 (v6) ALLOW IN Anywhere (v6)	
4	8443 (v6) ALLOW IN Anywhere (v6)	

Firewall enabled

Add a name for the rule, In this example we will be setting up a rule to deny SSH so the label can be "Test SSH" adding other rules can be done in the same fashion.
Change the policy to Deny, Direction to In, Log to Log All, Protocol to TCP, and the To Port to 22, which is the default SSH port. When you've changed the following, click Add to confirm and add the rule. What this rule does is deny any SSH connection within the same network as your computer.



Name: Test SSH

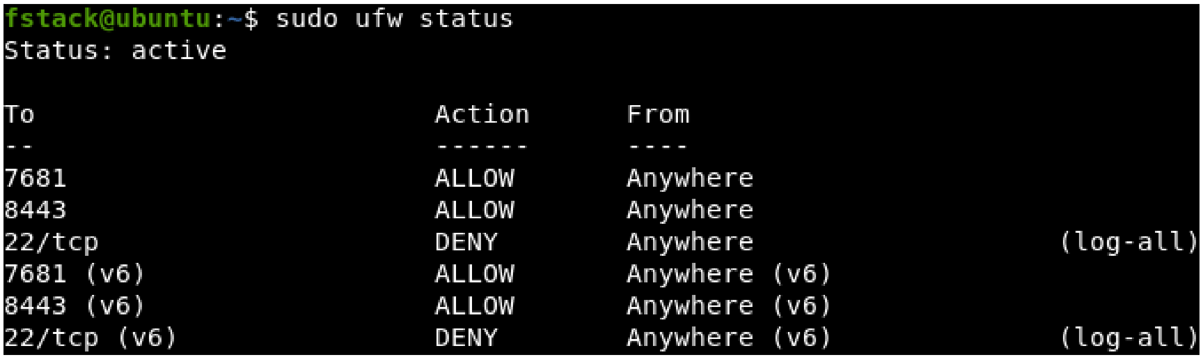
Policy: Deny

Direction: In

Protocol: TCP

Port: 22

If you want to confirm the rule was made, you can go back to your terminal and use command sudo ufw status to see the status of the firewall as well as the rule you added.



```
fstack@ubuntu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
7681                       ALLOW       Anywhere
8443                       ALLOW       Anywhere
22/tcp                     DENY        Anywhere                   (log-all)
7681 (v6)                  ALLOW       Anywhere (v6)
8443 (v6)                  ALLOW       Anywhere (v6)
22/tcp (v6)                DENY        Anywhere (v6)              (log-all)
```

Iptables, UFW, and GUFW, and which is best for you

Iptables is the core and most powerful firewall tool in Linux, directly interacting with the kernel-level netfilter framework. It allows for precise packet filtering, NAT configuration, and traffic control. However, its complexity is high—it requires deep knowledge of networking protocols like TCP, UDP, and ICMP, as well as command-line syntax, since there is no abstraction layer. This makes it challenging for beginners but ideal for advanced use cases such as custom traffic shaping, security hardening on production servers, and enterprise firewall deployment. The main strengths of iptables lie in its flexibility, fine-grained rule control, and robust rule chaining, but it comes with a steep learning curve and lacks a graphical interface, which can be a barrier for many users.

To address this complexity, UFW (Uncomplicated Firewall) was introduced as a more user-friendly command-line wrapper around iptables. It simplifies rule management with intuitive commands, making it accessible to general Linux users and sysadmins. UFW is well-suited for managing firewalls on desktops and servers, especially for common tasks like allowing SSH or HTTP traffic. It automatically handles iptables rules in the background, supports port forwarding, rate limiting, and logging, and is significantly easier to use than raw iptables. However, UFW sacrifices some advanced flexibility, making it less appropriate for highly customized or low-level firewall configurations.

For users who prefer graphical interfaces, GUFW provides a GUI front-end to UFW, offering the same functionality through a point-and-click environment. It is ideal for desktop users, visual learners, and beginners who want a simple way to manage their firewall. GUFW includes predefined rule templates, dropdowns for ports and services, and visual status indicators, making it especially useful for quick tasks like enabling the firewall or blocking peer-to-peer traffic. While GUFW excels in ease of use, it is limited to UFW's capabilities and not suitable for advanced configurations or headless/server environments.

Misconfigured Firewall Rules

Misconfigured firewall rules can have serious consequences for network security. Just one misconfigured rule can compromise the whole system.

- Unintended Open Ports: An incorrectly set rule can leave open ports for attackers to exploit, leading to direct access to sensitive information.
- Data Breaches: Misconfigurations can lead to security breaches, exposing confidential information to attackers.
- Regulatory Penalties: Companies are subject to strict data protection regulations, and firewall misconfigurations can lead to non-compliance, resulting in fine and legal consequences.
- Operational Disruptions: Firewall misconfigurations can disrupt operations by blocking legitimate traffic or allowing malicious traffic, leading to any downtime or loss of productivity and revenue.

Real World Examples:

- BlueSky Ransomware, where a public facing Microsoft SQL Server, port 1433, was exposed to the internet.
- 2019 Capital One security breach, which was due to a misconfigured web application firewall.
- 2017 Equifax Data Breach, where they suffered a massive data breach of 147 million people due to a misconfigured Apache Struts framework.

Ways to prevent and mitigate misconfigurations:

- Conduct regular firewall audits to identify and fix any misconfigurations.
- Follow the Principle of Least Privilege and implement strict access control lists to minimize risks of misconfigurations or unauthorized access.
- Proper documentation of any and all firewall rules and configurations.
- Training and education for firewall management.
- Regular updates to stay to date on security patches.

Conclusion

Firewalls are a foundational part of network security, and understanding how to properly configure and maintain them is crucial for any Linux user. Tools like iptables, UFW, and GUFW offer varying levels of complexity and control to suit users of all experience levels. While iptables is powerful and customizable, UFW and GUFW make firewall management accessible to everyday users and small businesses without compromising core functionality.

By enabling and configuring GUFW, you've taken a key step in hardening your system against unauthorized access. While a firewall protects you from many external threats, it's only one layer of defense in the vast array of tools used in cybersecurity. Practicing good opsec by avoiding suspicious downloads, keeping your system updated, and monitoring your firewall logs—ensures a more secure and resilient computing environment.

With a little effort and the right tools, effective network security doesn’t have to be complicated with UFW. The uncomplicated firewall.

Resources:

- Capital One. (2019). *Digital facts 2019*. https://www.capitalone.com/digital/facts2019/
- Comparitech. (n.d.). *Beginner's guide to iptables: How to set up, configure, and use iptables*. https://www.comparitech.com/net-admin/beginners-guide-ip-tables/?utm_source
- Cybersecurity and Infrastructure Security Agency. (n.d.). *Understanding firewalls for home and small office use*. U.S. Department of Homeland Security. https://www.cisa.gov/news-events/news/understanding-firewalls-home-and-small-office-use
- CIODive. (2019, August 1). *From Equifax to Capital One: The problem with web application security*. https://www.ciodive.com/news/from-equifax-to-capital-one-the-problem-with-web-application-security/560210/?utm_source
- Opinnate. (n.d.). *Guarding the gate: Security breaches caused by firewall misconfigurations*. https://opinnate.com/guarding-the-gate-security-breaches-caused-by-firewall-misconfigurations/
- SentinelOne. (n.d.). *BlueSky ransomware analysis*. https://www.sentinelone.com/anthology/bluesky/
- Server Fault. (2010, March 3). *Can you recommend a good intro to iptables?* https://serverfault.com/questions/158772/can-you-recommend-a-good-intro-to-iptables?utm_source
- Ubuntu Community Help Wiki. (n.d.). *Gufw*. https://help.ubuntu.com/community/Gufw
- User "Zodman". (2024, February 1). *ufw.service enabled and active* [Online forum post]. Arch Linux Forums. https://bbs.archlinux.org/viewtopic.php?id=294105