**INTRODUCTION TO GNU/LINUX SECURITY**



In the world of cyber-security there is a saying that says: "the only truly secure computer system is the one that is disconnected from the network" The feeling may be a little true, but it is not a practical solution to the problems we face today in protecting servers and desktops from external intrusions.

There are more computer systems connected to the Internet, either directly or through local area networks, than at any time in the history of technology, and numbers are growing at a rapid pace. It seems that not a month goes by without another story in the news about the internal network of a large corporation that is being compromised by an intruder.

The simple fact is that there really isn't a truly secure system if it is connected to a network. Large corporations with expensive firewalls and extremely talented tech personnel cannot always prevent cyber criminals from breaking their security.

Fortunately, not everything is lost, and we don't have to remove the network cables from the back of our computer systems. With careful planning and system configuration, it is quite possible to create a secure environment that makes the hacker's move to the next destination easier without making the entire system useless.

**The First Part of the Defense**

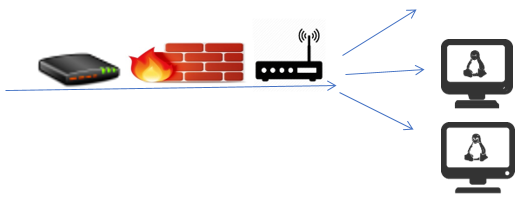
**“Firewall”**

The first phase; the most important to develop the secure environment is to avoid, whenever possible, that your Linux system is the first line of defense against an external attack. The best way to do this is to make sure you have a firewall installed between your GNU/Linux system (or the network on which it is installed) and your Internet connection. If, for example, your Linux system is connected directly to a DSL modem cable or box, you should seriously consider installing a wireless router or base station that includes a firewall feature between the modem and your Linux system. Linux comes with a firewall that can be configured to protect you and which we will detail later in this guide. However, it is better not to rely solely on this.

A firewall essentially stands between your computer, or network on which your computer resides, and protects you from the dangers that lurk on the Internet. It can be a software program that run on a computer system or can be integrated into a hardware device, or even a Raspberry Pi, also as a wireless base station or router hub. In this part we will see the firewalls as part of a wired or wireless hub. In later chapters, we will look ah how to configure firewall software on a Linux system to provide a second layer of defense against attack or intrusion from outside or within our network.

First, let’s define what a firewall is

Imagine that the Internet is a hell of a virus and cybercriminals looking for systems they can invade. Something like a real-life firewall prevents this unpleasant thing from spreading to your environment. In other words, it is a barrier that prevents intruders from entering. The firewall prevents unwanted connections from entering your internal network. Rather like the gateway on the fortress wall, the firewall only allows data and connections that meet certain criteria created or established by us to pass through the wall to the internal network.



**How does a Firewall work?**

The basic functions of a firewall are as follows:

**Invisible Mode:**

Normally, a utility called ping receives the IP address of the remote system. The ping utility sends a packet of data to the remote system represented by the IP address and waits for a response. If you receive a response, then the user knows that the system at that address is available on the network.

**Forwarding and Port Blocking**

What we know as port blocking is the most fundamental level of firewall security and will be used by the most home or small business users to protect their systems. We have mentioned this before; computer systems communicate through ports. A firewall can be used to block any port that you do not want to open to your systems within the firewall. For example, FTP operates through port 21. If you do not want anyone outside to have FTP access to your systems, you will need to configure the firewall to block port 21 of FTP.

Conversely, port forwarding is also a useful tool. Suppose you have three Linux systems on your internal network and want to be able to Telnet on one of those systems when you are outside your firewall (perhaps at the local restaurant that uses the free Wi-Fi connection while eating dinner or while you are staying on a trip). In this situation, you will configure your firewall to forward connections from port 21 to the system you want to access from the outside. When you connect to your IP address via Telnet, the firewall will see packets arriving al port 21 and knows that they have to be sent to the IP address of the designated machine. If you have more than one system on your network, it is essential that you configure port forwarding to handle this.

**Package Filtering**

Data is transmitted over networks and the Internet in so-called packets. Each packet contains information about where the data comes from and where it goes (i.e., the sender's IP address and its IP address). In fact, a packet contains a great deal of information about the nature of the data being transmitted and many advanced firewall solutions allow you to filter the packets of data that enter through your Internet connection to allow or reject packets based on what are called configured filtering rules. For example, you can allow a telnet session (which allows you to log on to your Linux system from the outside) but not allow ftp packets (which allow files to be transferred to and from your Linux system). You can also choose to block packets coming from an IP address that you know is suspicious.

**The Second Part of the Defense**

**"GNU/Linux System Firewall"**

The GNU/Linux firewall is already integrated into the kernel, currently rules are being handled in the iptables firewall, which is already integrated since the 2.4 kernel version.

**Configure Iptables Firewall on a machine with GNU/Linux**

Configure iptables firewall rules for IPv4 inbound and outbound traffic on a Debian PC (without routing).

**Show current IPv4 settings**



**Save the current IPv4 rules to a backup file**



**Empty and remove any existing chain configuration for IPv4**

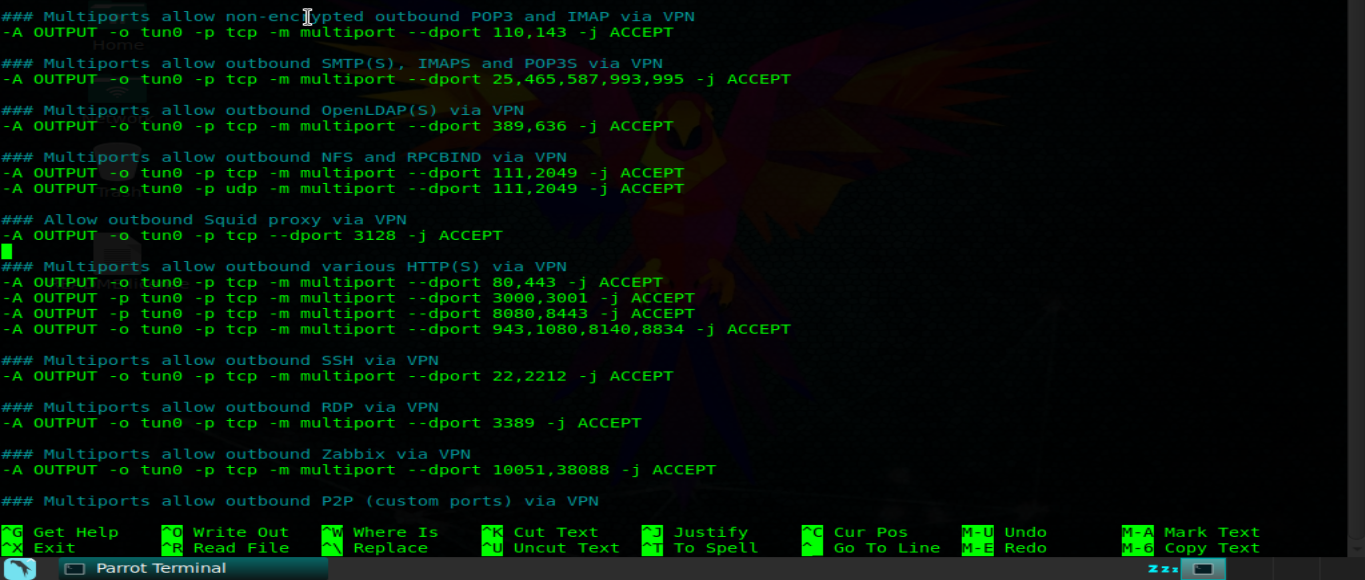
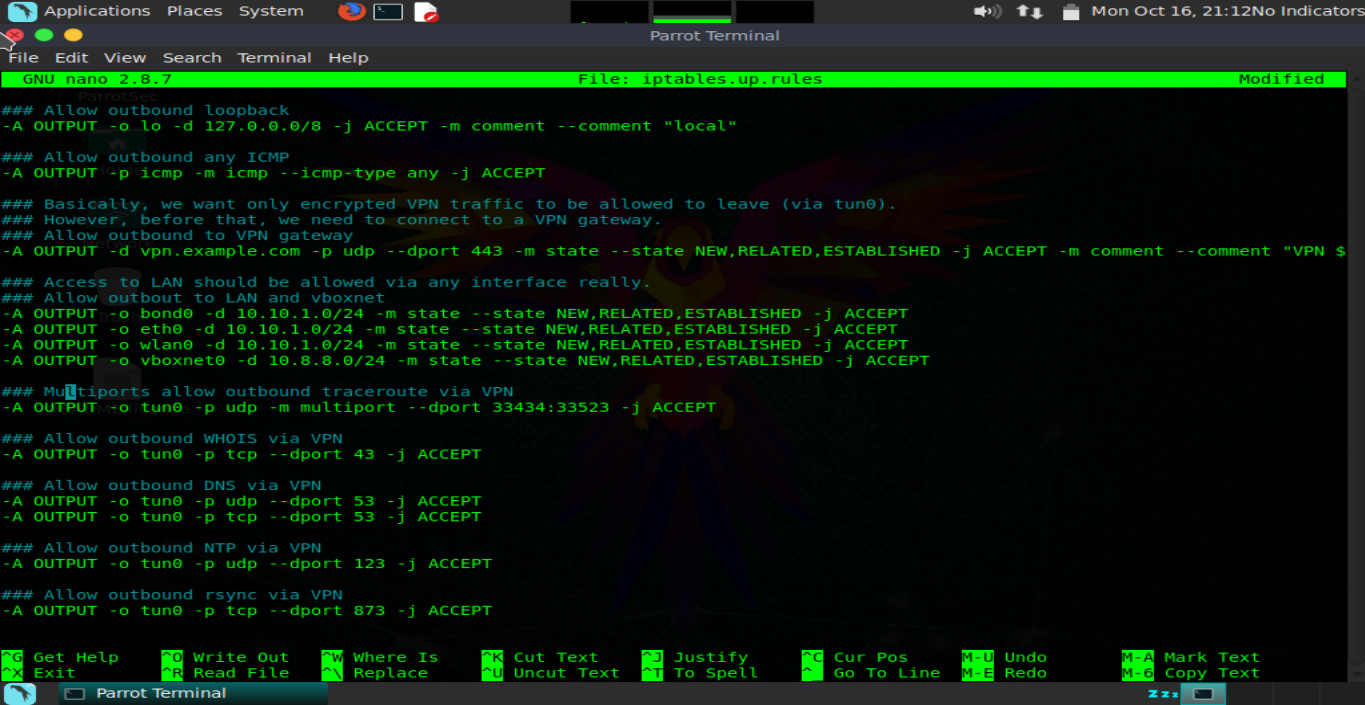
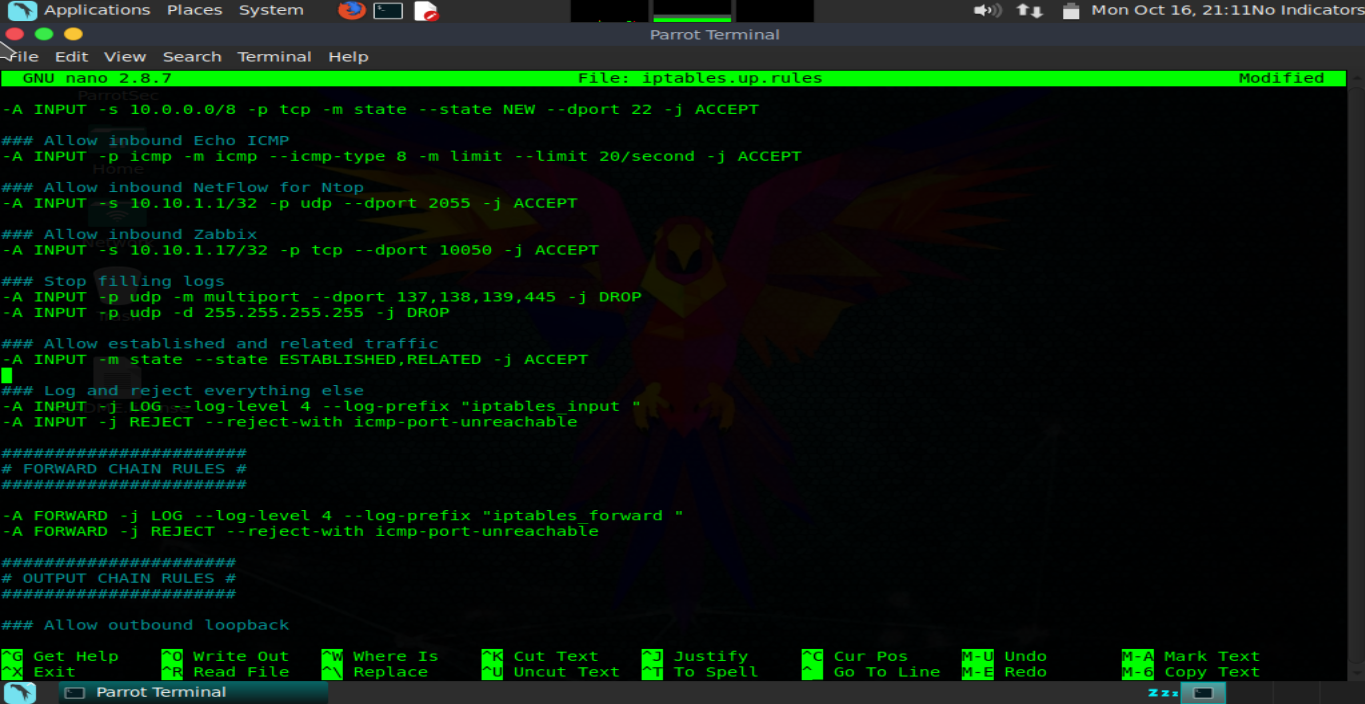
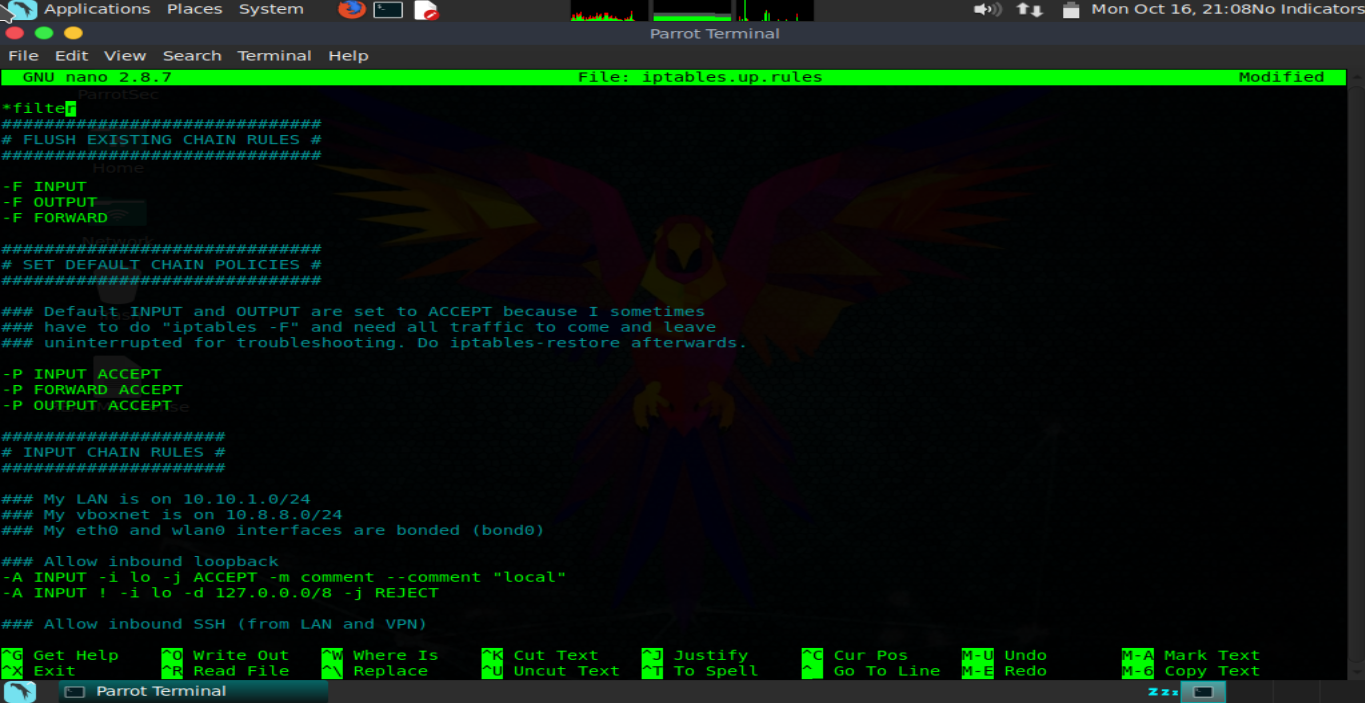


**Configure a new configuration for IPv4 and apply new rules**

**Create a file to store the configuration:** 

**Adding iptables rules to /etc/iptables. up. rules.**

**In this example are the rules:**



**Apply the changes:** 

**Make the IPv4 IPtables rules boot at system startup**

**Create a startup file:** 

**Make it executable:**



**Iptables Rules for IPv6**

**Create a file to store the configuration:** 

**Adding iptables rules to /etc/iptables. up. rules.**

**In this example are the rules:** 

**Make Iptables IPv6 rules start at startup**

**Create a startup file:** 

**Make it executable:**

