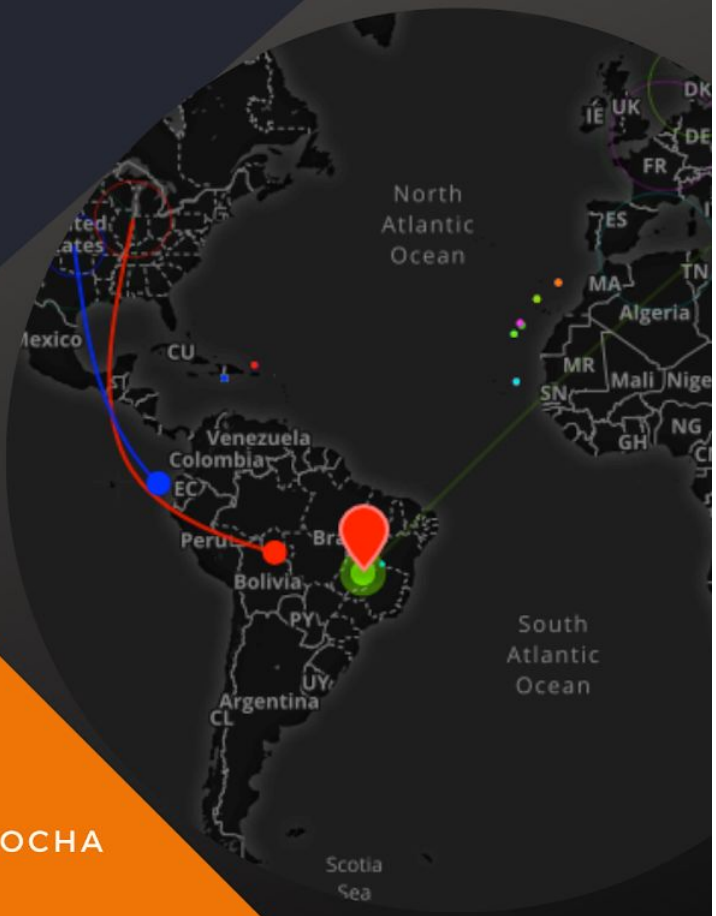
OPEN SOURCE SECURITY

YOUR NETWORK MORE SECURE WITH OPEN SOURCE TOOLS



DIEGO BRUM LIMA ROCHA

OPEN SOURCE SECURITY

YOUR NETWORK MORE SECURE WITH OPEN SOURCE TOOLS

DIEGO BRUM LIMA ROCHA

To my daughter Bianca Brum.

Summary

Preface

Usually a preface is written by renowned and prestigious authors as a way of endorsing a new writer who presents himself. I tell you, don't expect any of this from me. I'm not even an author or a writer. I am just a person who has been working directly with Diego for some years - as a coworker, now coordinating his work - who knows how determined, pioneering, knowledgeable and passionate about Information Security he is. So, believe me: if you work or think about going into this area, don't miss this book!

I met Diego in late 2015 when we started working together. Me as a Technologist in Computer Networks and he as a Technologist in Information Security. The mission was arduous, but deeply instigating. In search of solutions, he entered the universe of Open Source tools focused on Information Security. There he faced many difficulties, doubts, work and rework, but also with daily achievements. Many of the solutions that we needed at the time came from the tools that will be presented here and the way they were implemented. Author's merit.

This work also allows for reflection if, in fact, a safe environment is one that has a high investment cost attached to it, by presenting free tools that, when well configured and associated with some simple practices, ranging from changing passwords to awareness of users, are examples of procedures with little or no financial cost that have a great impact when it comes to Information Security.

All this knowledge and this learning curve are offered to you in this book, in a clear and didactic language, so that you can optimize your time, in order to use it for new discoveries and not to face issues that have already been overcome by someone else. opportunity.

Want to know how to install and configure Open Source tools such as Linux Operating System, password vault, second authentication factor, firewall, HIDS, reverse proxy, WAF, SIEM, attack map, among others? Then browse this book and find a chapter dedicated to each of the tools presented, with a tutorial that will guide you through the entire process: download, installation and configuration. All this for you to enjoy the tool in the best possible way and still optimize your time.
Computer, internet and book ready? So, get to work!

Edimaria Cerqueira Rodrigues Lamounier, specialist in Computer Networks.

Introduction

I have been working with Information Security since 2015. I started to get interested in the subject when I was still an Officer of the Brazilian Air Force. I remember that there was a defacement (attack that aims to modify a web page, like graffiti) in one of the web systems of aeronautics and this shocked me, as I believed the network was well protected. Only later, working in the area, I realized that sometimes what happens is a false sense of security, perhaps due to the huge expenses with the best solutions for Firewall, WAF (Web Application Firewall), Antivirus, etc. I realized that, even if you have resources for all these solutions, nothing replaces the performance of a team making adjustments to these tools and mainly investing in network visibility. Knowing what happens is fundamental. I believe that one of the best resources in the security area is Logs (system events). Knowing what is happening on your network, where the attacks are coming from, what the targets are and what type of exploitation is trying to be done in your environment makes all the difference. A good IDS (Intrusion Detection System) is already capable of revolutionizing security on your network.

I learned the importance of knowing what happens on the network in the worst possible way. My first days working with information security were not easy. A series of attacks coming, hacked websites, malware on the network compromising the domain, among others. In order to protect the network, there were a firewall and an antivirus. I had no idea what was going on and I had no light on it.

I realized then that it would be necessary to implement a security architecture, which would involve tools, but also processes that, although simple, were fundamental. The simple management of network passwords and the updating of all systems / software have already taken a huge leap in security. Then came the tools and, to my surprise, there were excellent Open Source options.

My motivation for this book was precisely to help those who have the arduous mission to protect their network. The tools and processes described here have helped me a lot and I believe they can be the solution for many network administrators and security analysts.

"If you know the enemy and you know yourself,
you need not fear the results of hundred battles,
if you know yourself, but not the enemy,
for every victory gained, you will also suffer defeat.
If you know not of the enemy, nor yourself,
you will succumb in every battle."

- Sun Tzu

1 - Password Vault

I couldn't start with a tool other than the Password Vault, as there is still little importance to password management. Information Security is in the details. When it comes to Information Security, many think only of complex, ingenious and expensive solutions. There is no shortage of password leakage scandals even in large companies and the damage is great, mainly because many people do not use the Second Authentication Factor. The best password security is one that uses at least two of the three authentication factors:

- What you know (Password);
- What you have (Token);
- What you are (Biometrics).

With these leaks, it is useless to have the best password in the world, with many characters or of high complexity. Well, it is clear that this will generate a HASH that is difficult to decipher by Rainbow Tables (pre-calculated HASH tables), but when in doubt, and aiming at a better security of your personal data, I strongly recommend that you use, if possible in all your accounts on the internet, the Second Authentication Factor.

The idea is that the network administrator, and his team, use a Vault where the network passwords are managed and share it only with his team. The Vault will assist in creating strong passwords and a host of other password information. I recommend the tool **KeePass** (https://keepass.info).

It is an Open Source tool, like all that I will show in this book, with versions for various operating system platforms and very simple to use.

So let's test the tool. First download it from the Downloads menu on the keepass website.

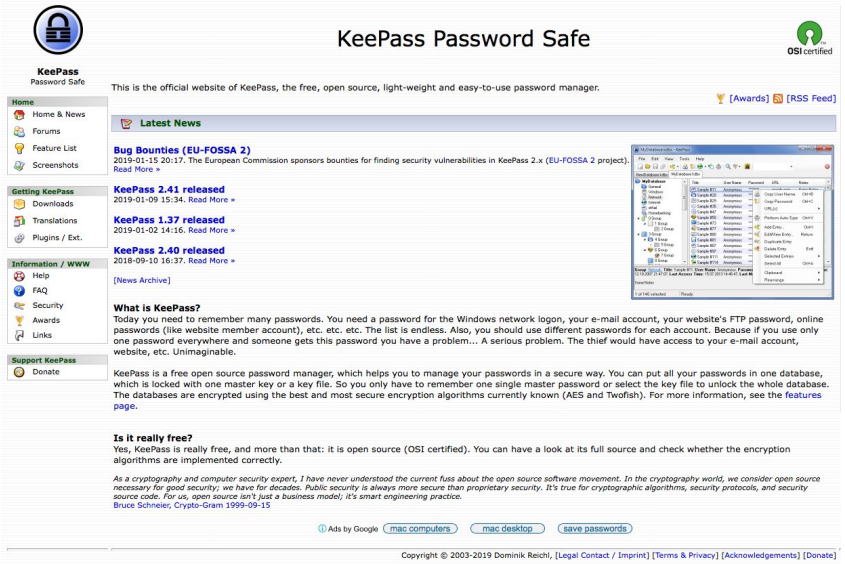


Image 1.1 - KeePass website

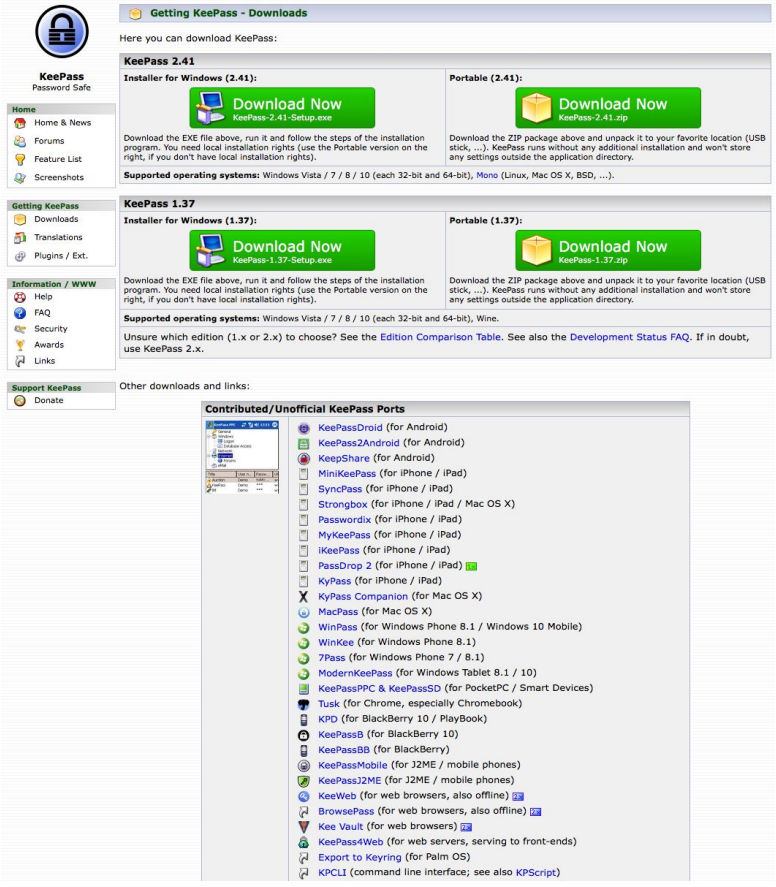


Image 1.2 - KeePass download area

I will download the version for Mac OS (MacPass), which is the platform I am using. Now, install and create a new database.

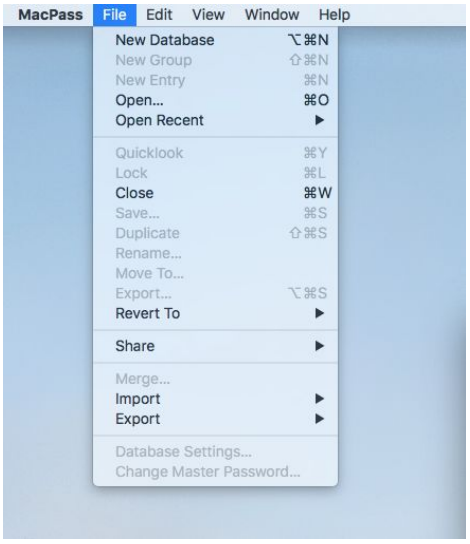


Image 1.3 - Creating a password database

Now MacPass already has password groups, such as Windows, Network, Internet, Email and Homebanking. You can create others or delete those. In the example below I will create a password in the Windows group.

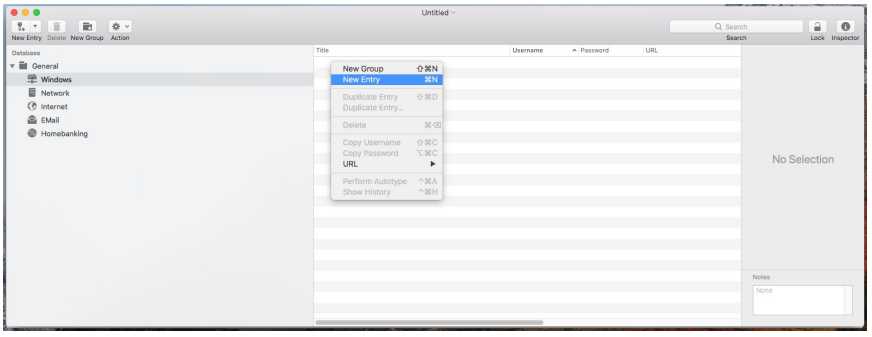


Image 1.4 - New Password

Here at MacPass, when clicking New Entry, I was asked if I would like to create the password for accessing the Password Vault. As the image below:

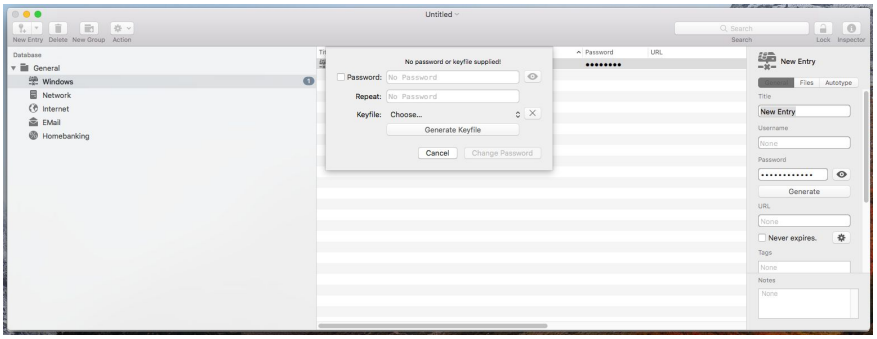


Image 1.5 - Password to access the Password Vault

You will see the fields for creating a password and creating a Password File (KeyFile). Creating a password for the password vault leads us to a problem: as we are creating a Password Vault to avoid sharing a password that will end up on some post-it on the network administrator's monitor. So I recommend using the Password File and making a control through GitLab (software repository manager based on git). GitLab is also Open Source and excellent for versioning software. It can be used to manage changes to your Password Vault and your Firewall file, as we will see in the next chapter. However, installing and using GitLab is outside the scope of this book. The idea is quite simple: generate the Password File and place it in the GitLab project together with the Password Vault database, in such a way that only project members can have access, both to the Password Vault bank and the Password File.

Continuing with password creation, KeePass (or MacPass) already creates a password for you, which can be seen by clicking on the icon  next to the password field. However, I recommend changing the password by placing it in accordance with its Information Security Policy, in terms of size and complexity.

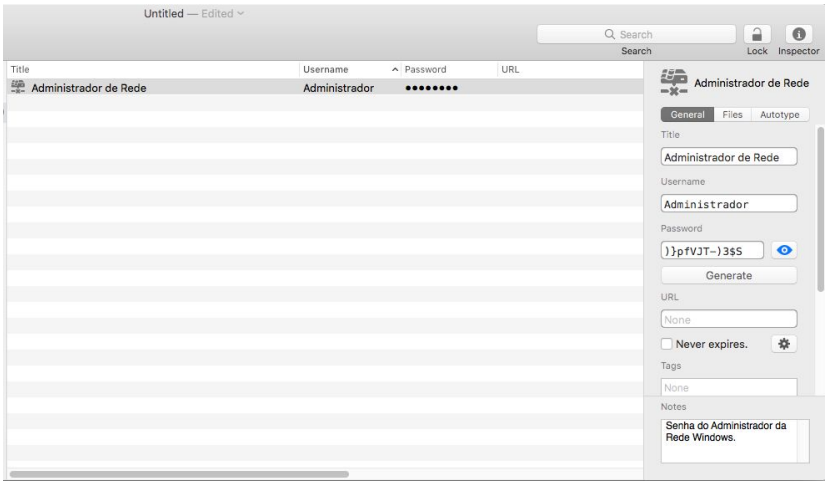


Image 1.6 - Password creation

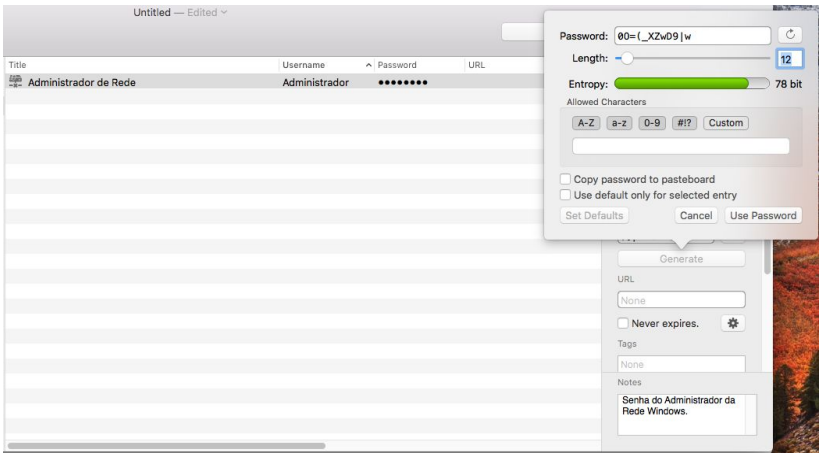


Image 1.7 - Generate button

By clicking on Generate, you can customize the password according to your needs.

1.2 Second Authenticator Factor

Understanding the importance of the second authentication factor, we will implement it on a Linux server. Later, we will install other solutions and make security configurations and all of them will occur in the Linux environment, so I recommend that you install a virtualization environment, such as **VirtualBox**, and download the **Linux Debian version 9** (the latest version available when writing this book).

Why Debian? First of all, because Debian is, among all Linux other distributions, one of the most faithful to the Open Source movement. In addition, it is extremely stable, secure (as long as it stays up to date) and easy to use. If you want to use another distribution, there will be no problem.

The Linux OS installation is available in Appendix 1.

After installing the Linux OS, we go to the commands necessary for us to implement SSH access with two-factor authentication.

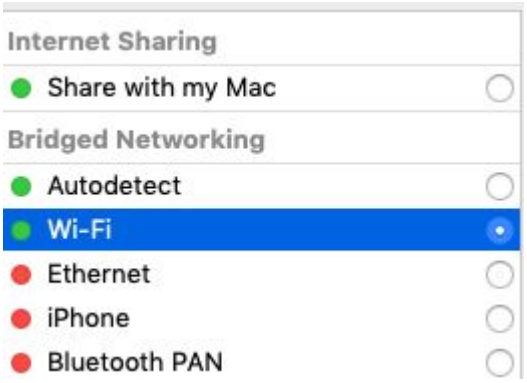


Image 1.8 - Wifi network IP

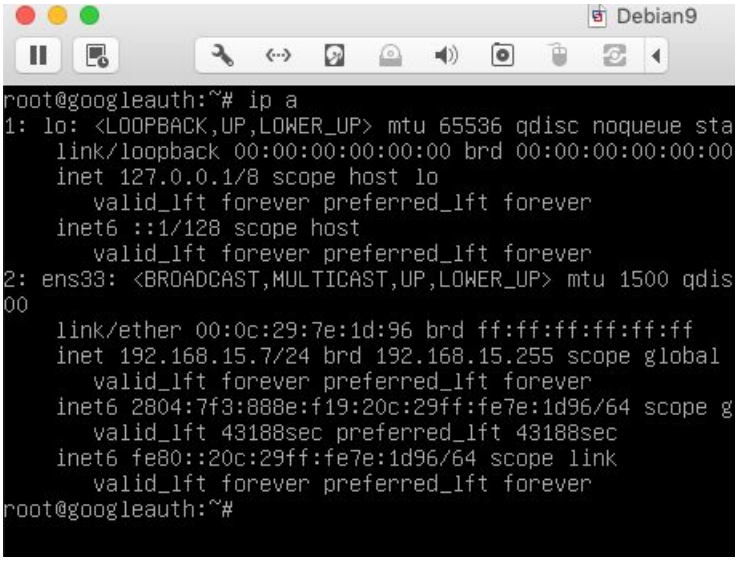


Image 1.9 - IP 192.168.15.7

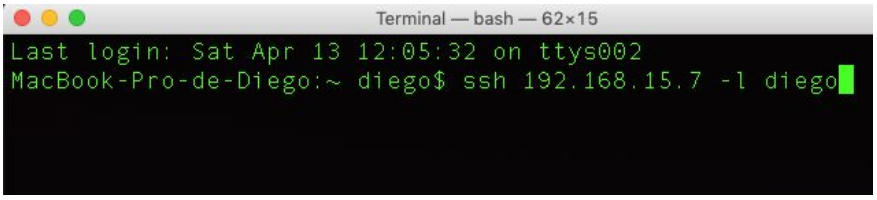


Image 1.10 - ssh on server 192.168.15.7

First, let's install the google-authenticator no Linux:

apt-get install libpam-google-authenticator -y

Then, run the google-authenticator: google-authenticator

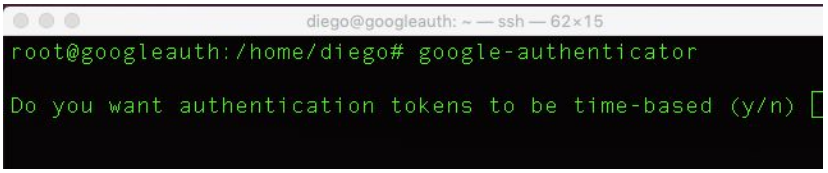


Image 1.11 - execution of google-authenticator

When answering **y** to the question, a QR Code and emergency codes will be generated, as a backup of the QR Code.

Image 1.12 - QR Code

Use your cell phone, with the second factor authentication application, and enter the secret key or scan the QR Code.

Image 1.13 - app on the cellphone

When using the application, you will now have a token like the one in the image above.

There will be five questions, the first, when answering **y**, made the QR Code and emergency codes appear. The second question about

creating a hidden file in the home of the user who is installing google authenticator. Type **y**, it is in this file that will be checked if the the code that was entered is correct at the time of login via ssh. The third wants to know if multiple users will be able to use the same token. Type y, as the idea is that only team members share the token and access to the Password Vault. The penultimate wants to know if there will be an increase in the time window in case of time synchronization problems on your server. For our book, where we have not yet configured an NTP (Network Time Protocol) service, something we will see in the Hardening chapter, type **y**. Finally, type **y** to block multiple login attempts.

We finished the installation part, now we have to configure the SSH service to use the password (something you know) and the token (something you have). First edit the file ***/etc/pam.d/sshd***.

Place below the ***@include common-auth*** the ***auth required pam_google_authenticator.so***.

```
# Standard Un*x authentication.
@include common-auth
auth required pam_google_authenticator.so
```

Image 1.14 - /etc/pam.d/sshd

Save and edit the file now ***/etc/ssh/sshd_config***. For this book only, to facilitate the process, enable ssh access with the root user: ***PermitRootLogin yes***. After, ***ChallengeResponseAuthentication yes***. Save the file and restart the ssh service: ***service ssh restart***

Let's test, open a terminal and try SSH access with the root user on IP 192.168.15.7:

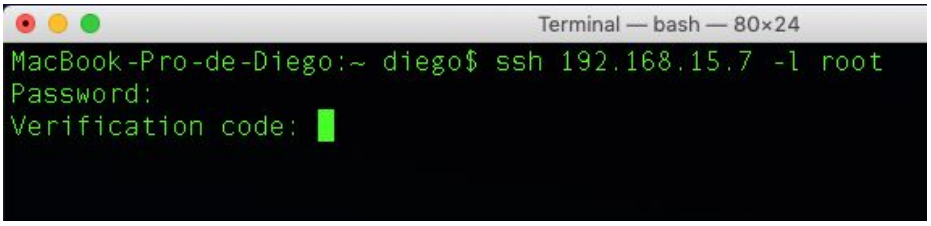


Image 1.15 - 2 authentication factors working

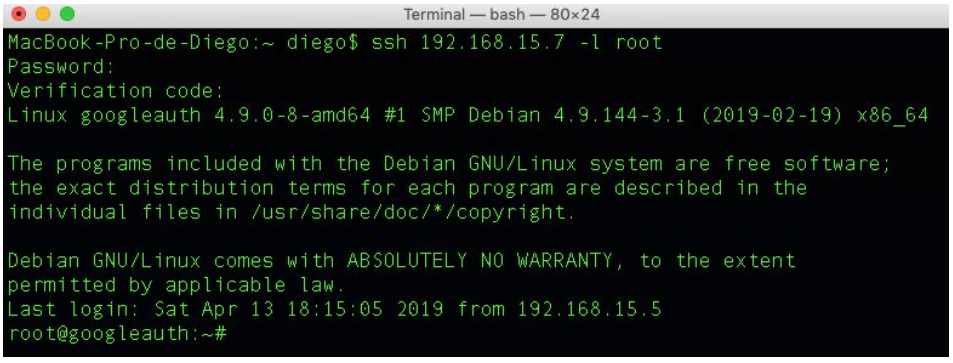


Image 1.16 - successful login

1.3 Resume

We have seen that password management is important in network administration and also in your home. Yes, I strongly recommend that you use a password safe for your home passwords as well, making you diversify your passwords and make them more complex. The Second Authentication Factor is also something to be considered for use on your network (corporate or not), because as we have seen this makes it difficult and can make it impossible for third parties who may have discovered their passwords.

2 - Firewall

Now let's talk about edge security. Edge refers to data entering or leaving your network. Also known as a network perimeter.

Currently, it is argued that the perimeter is no longer your Edge Firewall, but your cell phone. With the new wave, Internet of Things or simply IoT, anything with IP and access to external networks can be a new perimeter. Why your cell phone? Because people practically work on their cell phones: they access the intranet, email, vpn, etc ... So your network can have many perimeters and the vulnerabilities start to multiply. In any case, there will always be a perimeter, which is exactly where the Firewall enters.

When you think about Firewall, at least in my head, the first thing that comes up is Iptables. Iptables is a user interface tool that allows the creation of rules from the Netfilter module, which provides the Linux Operating System with the functions of Firewall, NAT and log. I do not know how much knowledge you have in relation to Iptables, but I already say that it is not trivial to use it without the use of tools such as Firewall Builder, which we will talk about later.

Anyway, let's go to the basics of Iptables. Iptables sees only IP and Port, not understanding layer 7 of the OSI model (Application). Nothing prevents you from installing a Squid so that you have better visibility of the layer, but the scope of the book is to set up an Iptables Firewall through the Firewall Builder and still implement a basic Shell Script for blocking Countries, something that can avoid a lot of pain Of Head. Iptables works with tables: Filter, NAT and Mangle.

- In the Filter table are the rules for blocking or releasing network packets, that is, it is the table that actually does the main functions we want in the Firewall. It treats packets that are forwarded to the Firewall as the final destination (INPUT chain), that are generated in the Firewall and leave (OUTPUT chain) and those that are forwarded and cross the Firewall (FORWARD chain).

- The NAT table has tasks such as changing the source (SNAT), destination (DNAT) IP addresses, masking (MASQUERADE) and redirecting packets (REDIRECT).

- Finally, the MANGLE table, which has special rules such as ToS (IPv4 header service type).

To see the iptables rules in linux, just type, as root or using sudo, the iptables -L command.

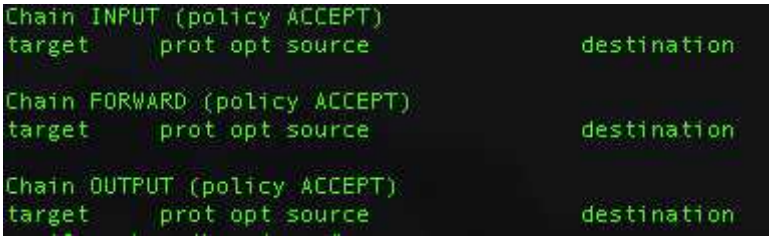


Image 2.1 - Output of the iptables -L command

See that in the example above there are no rules in our Iptables Firewall. Creating iptables rules by hand is not very productive and nothing trivial even more when Firewall rules start to grow with the demands of the network in the corporate environment, so let's learn how to use Firewall Builder to create Firewall rules just by dragging objects.

2.1 Firewall Builder

It is an excellent tool for creating Firewall rules, making the process much easier than the traditional one, through Shell Scripts with the rules in sequence and in droves. It is actually still the same, through shell script, but Firewall Builder does all the work for you. Download it on the website http://fwbuilder.sourceforge.net.

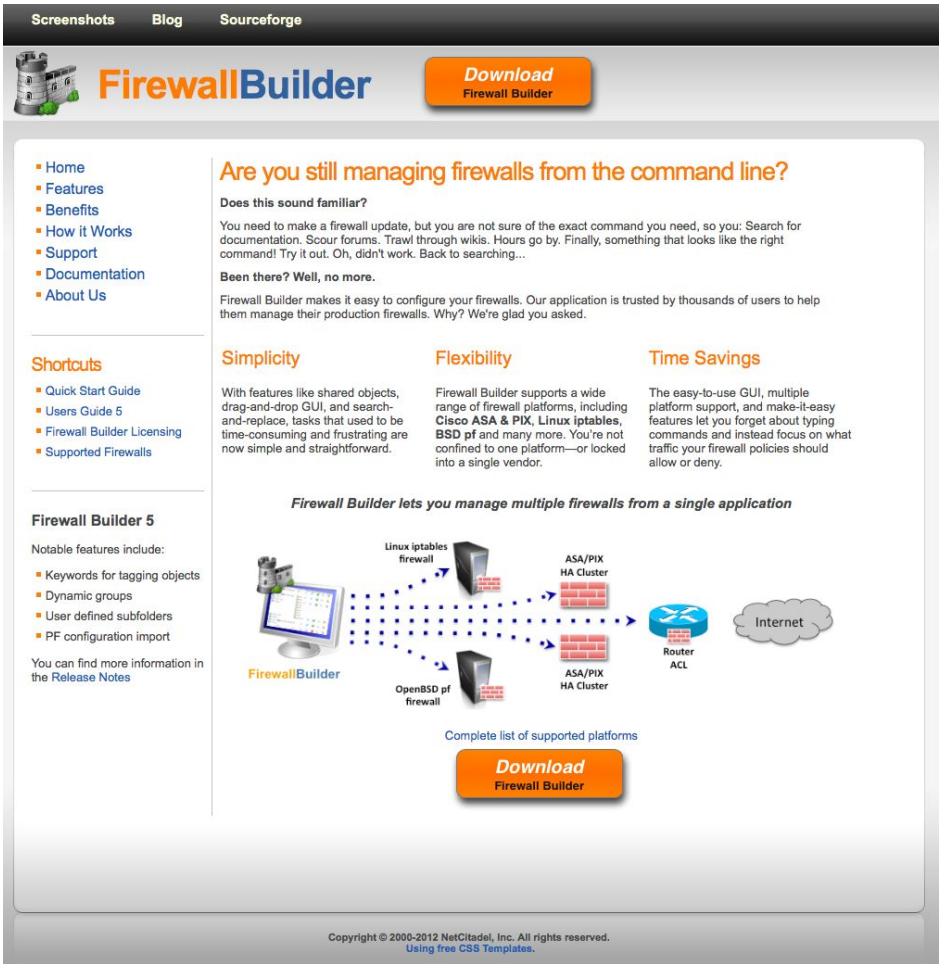


Image 2.2 - WebSite from Firewall Builder

Sad information appears at the bottom of the website: the Firewall Builder has been discontinued since 2012. However, still presenting some bugs, it meets perfectly, even in corporate environments. I've been using it for years and I have nothing to complain about.

The first step then is to create a Linux machine that will be our Firewall. The idea is that our Firewall is a gateway to two networks (corporate and DMZ). In the corporate we will have a machine that will play the role of the user, in the DMZ we will create a web server and a reverse proxy, which we will create in the respective chapter. Create the virtual machine with three network interfaces, the first in bridge mode, to get a valid IP from your wi-fi router. This way you can install Debian and its updates. The other two interfaces will have the IPs configured by the Firewall Builder.

I created the virtual machine that will be our Firewall, as shown in the image below.

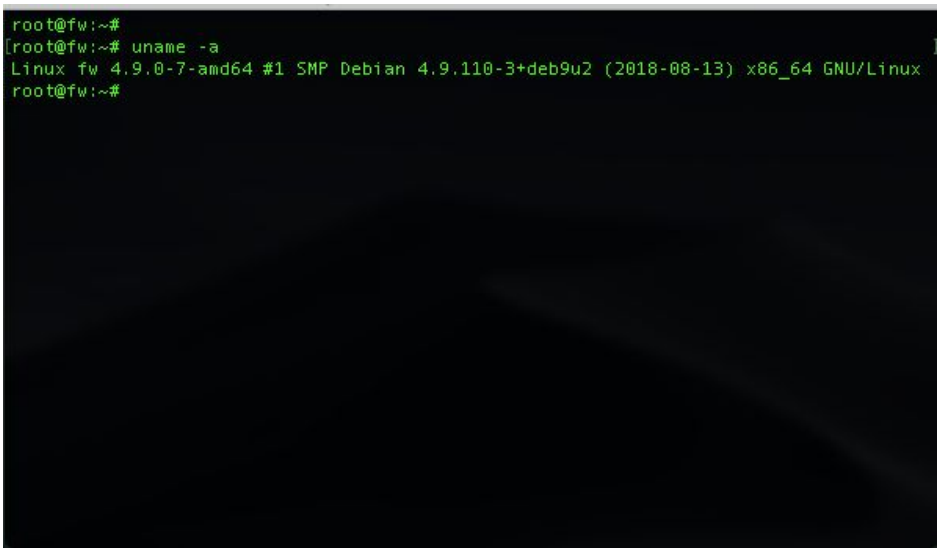


Image 2.3 - Firewall Debian 9

There are three network cards: the first is configured as Bridged Networking, so it will take an IP from my internet router, which will be the gateway of the FW (Firewall). The other two will remain as a Private Network, therefore only visible in the virtual environment. To make things easier, we will work with / 24 mask. So our networks will look like this:

- Corporate (172.16.1.0/24)

- DMZ (192.168.1.0/24)
- Firewall IP on the wifi router's network via DHCP (192.168.15.30). Probably yours will be different.
- Firewall GW (192.168.15.1). This is the IP of my Wi-Fi router.

With the networks defined, we go to the Firewall Builder.

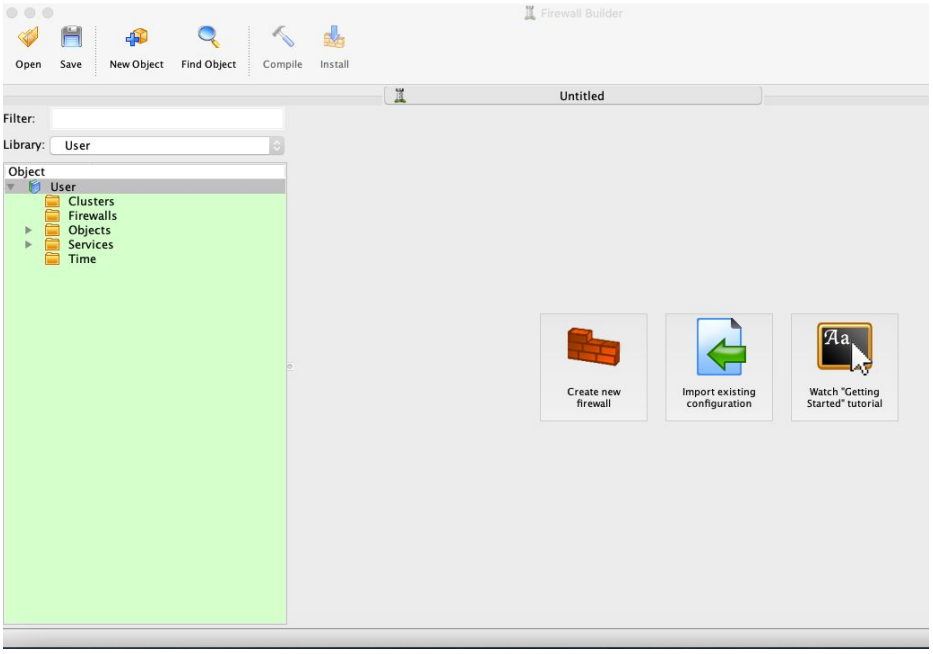


Image 2.4 - Firewall Builder

At that moment we have to create our firewall object, clicking Create new firewall. 

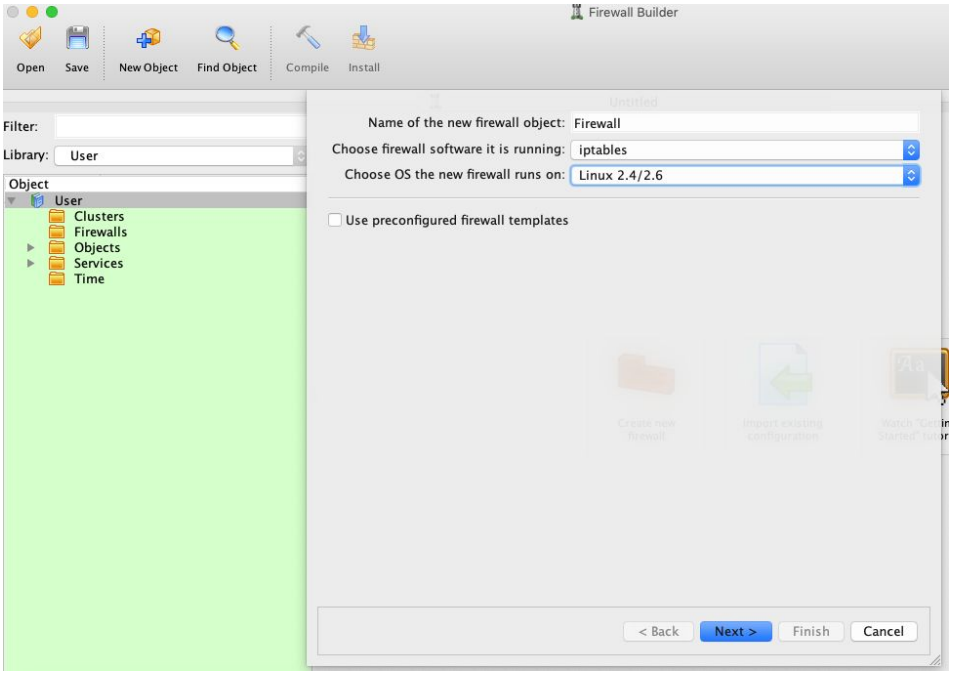


Image 2.5 - Create FW

Choose a name, FW type and kernel. Having stopped in 2012, Firewall Builder only offers the 2.4 / 2.6 kernel option, but that is irrelevant.

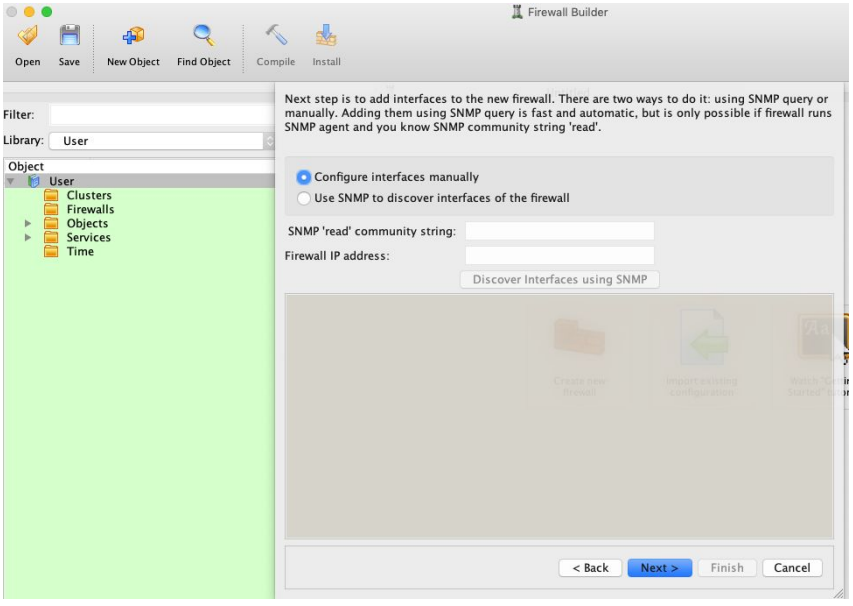


Image 2.6 - Configuration of network interfaces

Choose manual configuration. Then click on the icon  that appears in the upper left corner to appear the screen below.

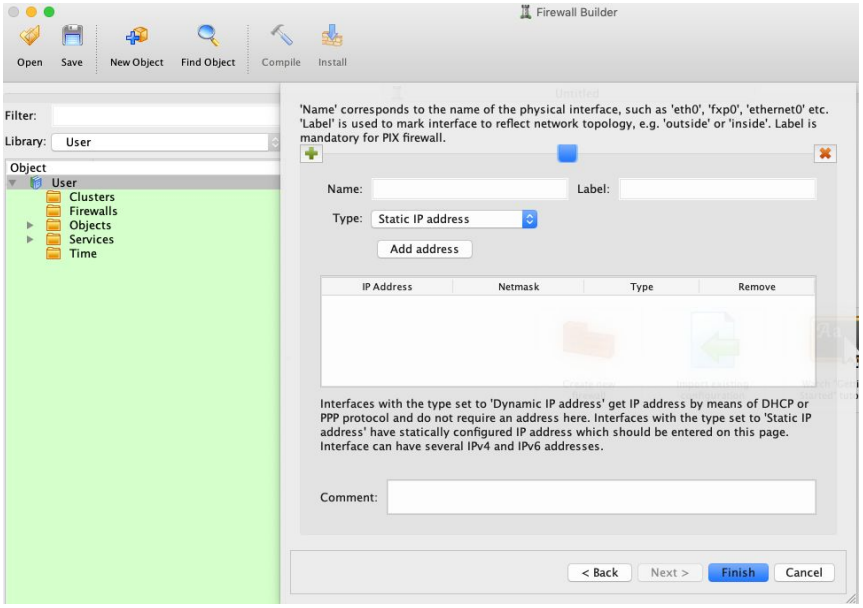


Image 2.7 - Creation of network interfaces

In Debian 9 the naming of the network interfaces is different. Instead of being eth0, eth1, etc ... it's ens, enp, etc, give the command ***dmesg |grep eth*** and see the names of your interfaces. Enjoy being in the virtual machine and, as root, create a folder in **/etc** called **fw**, because it is the default Firewall Builder directory to copy, via ssh, the shell script with the iptables rules. (***mkdir /etc/fw***)

I set ens33 or eth0 to 192.168.15.30, which is the IP that DHCP on my wi-fi network gave to my FW on the bridge interface. It will be the GW for the internet.

The other interfaces will follow the scheme I mentioned above.

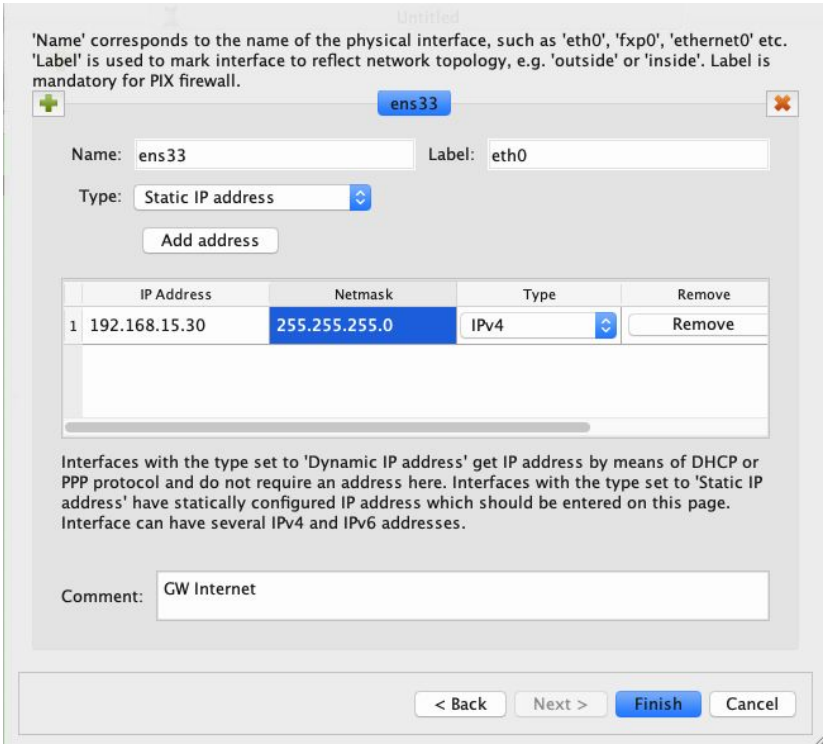


Image 2.8 - Interface GW Internet

After creating the first interface, the others are created by pressing the right mouse button on the FW object.

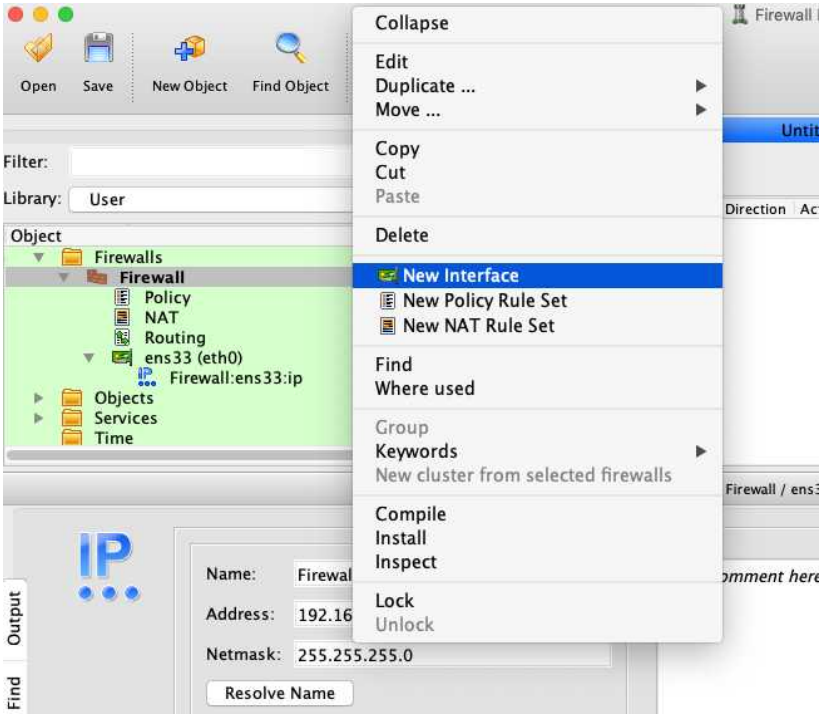


Image 2.9 - New network interfaces

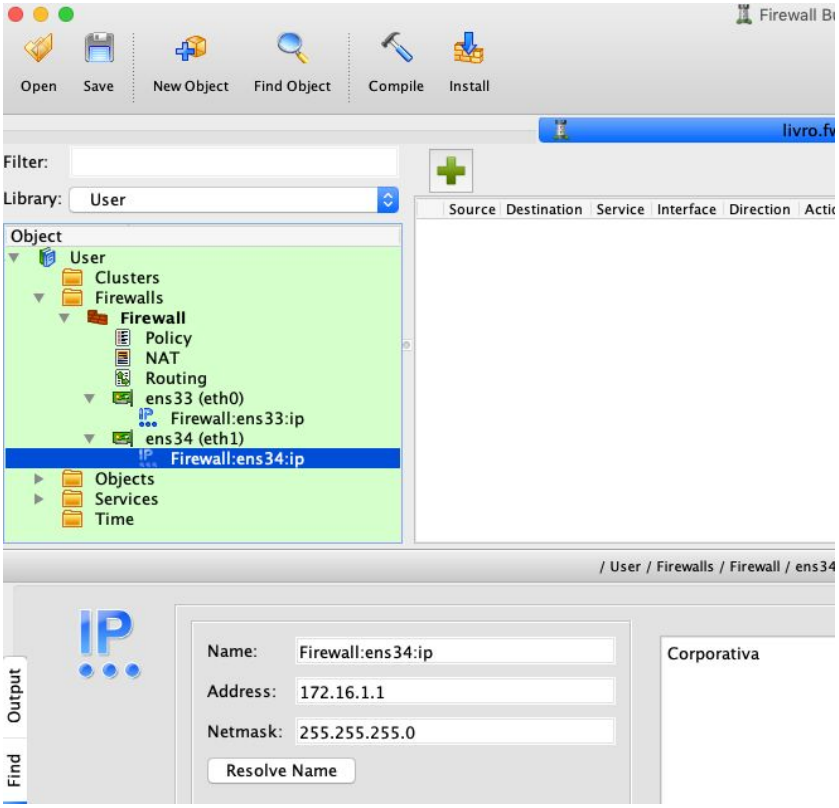


Image 2.10 - Corporate network

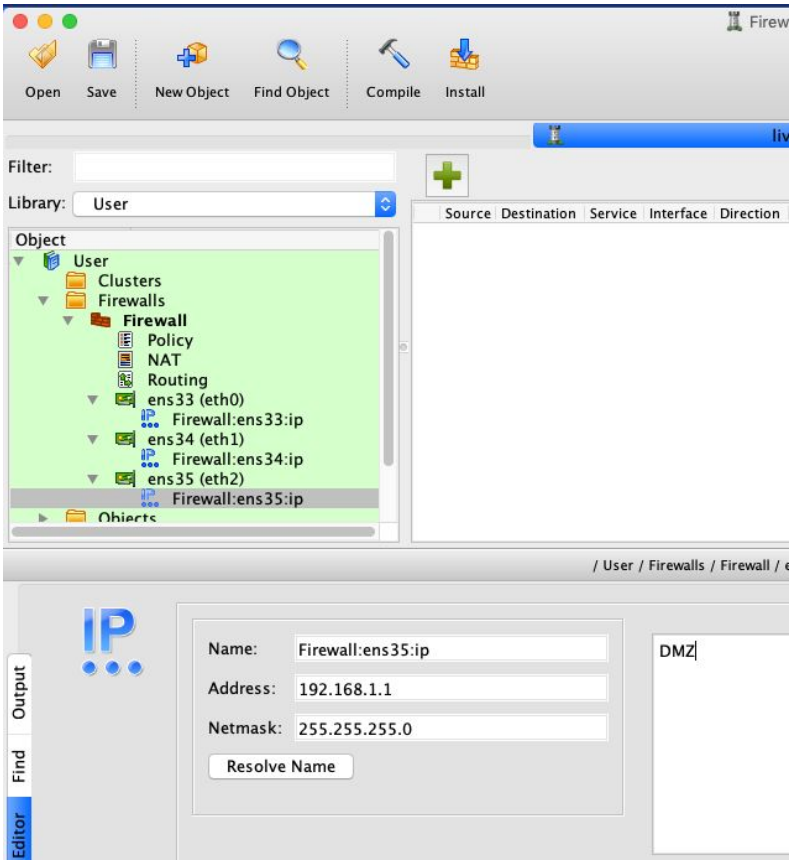


Image 2.11 - DMZ

With the network interfaces created, let's face the rules in the Policy table. But first, double-click the left mouse button on the ens33 network interface (eth0) and enable the field **Management interface**. This causes the FW rules to be transmitted from Firewall Builder to FW through this interface.

In my opinion, the best FW policy is denying everything and releasing only what is necessary. So our first rule will be precisely that of denial. Just click on the Policy table and then on  . See that the default rule created is exactly the one we wanted, denying EVERYTHING, with all fields in ANY and Action in 

. Another good practice is to always leave the Options field with Log enabled, to allow forensic analysis or even to check FW traffic.

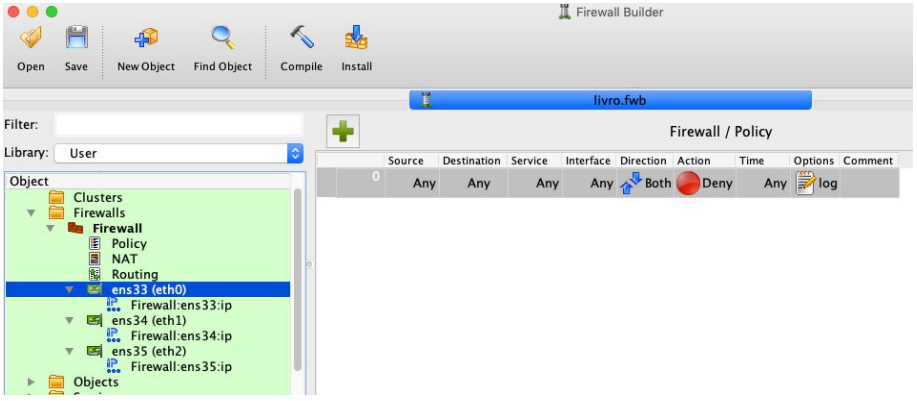


Image 2.12 - Policy1

We will create the basic rules so that we can, for example, make a ssh in the FW to manage it. Just press the  or with the right mouse button in the left corner of any FW rule and ***Insert New Rule***. We will use the Firewall Builder facilities to create a rule that releases the SSH (Secure Shell) service with any source (any) for FW. Just go to Find Object 

and look for the service. You can search for the SSH name or port 22. When you find it, just drag the object 

for the Service column of the rule.

We already have the service, now the object that will occupy the Destination column is missing, which in our example is FW. Just drag the object 

or just the IP of the FW that we want to enable SSH access, being a safer option, because if we drag the entire FW, we will enable SSH on all of its interfaces. I will just drag the IP of the interface I called ETH0, as it is the IP that is on the same network as my wi-fi router and I will be able to administer my machine external to the virtual environment. To finish, just enable the Action column to 

. The ready rule will look like this:

Image 2.13 - Rule ssh

See that I put a comment on the rule, also a good practice. The rules must be above the rule 

, because the FW reads the rules from top to bottom. Let's try it out. For that, it is necessary to install these Firewall Builder policies on the Debian machine that will be our FW. We will do everything as root, but just to make things easier in the virtualization environment for this book. In production it is recommended to create a user with sudo powers in Debian and install the FW with it.

That said, click 

, to compile and check for errors, and then 

.

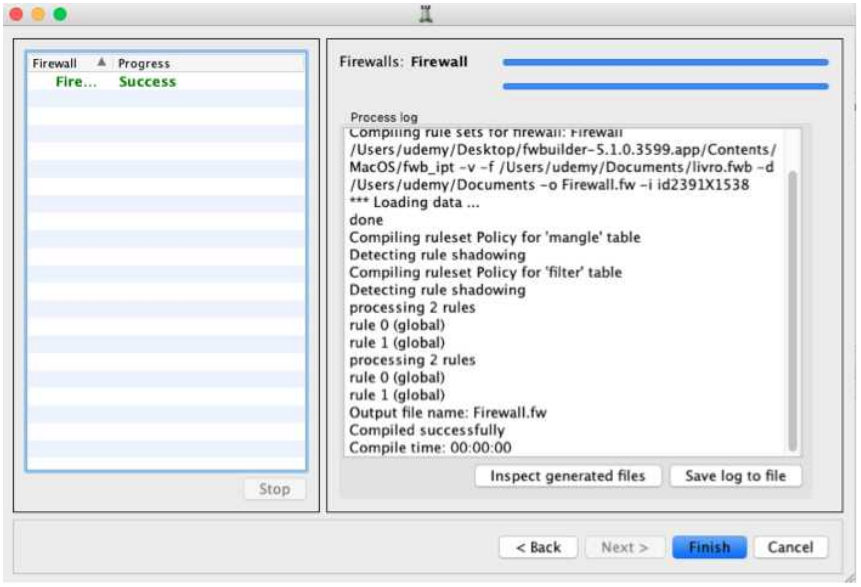


Image 2.14 - Expected build result

Image 2.15 - Fw installation screen

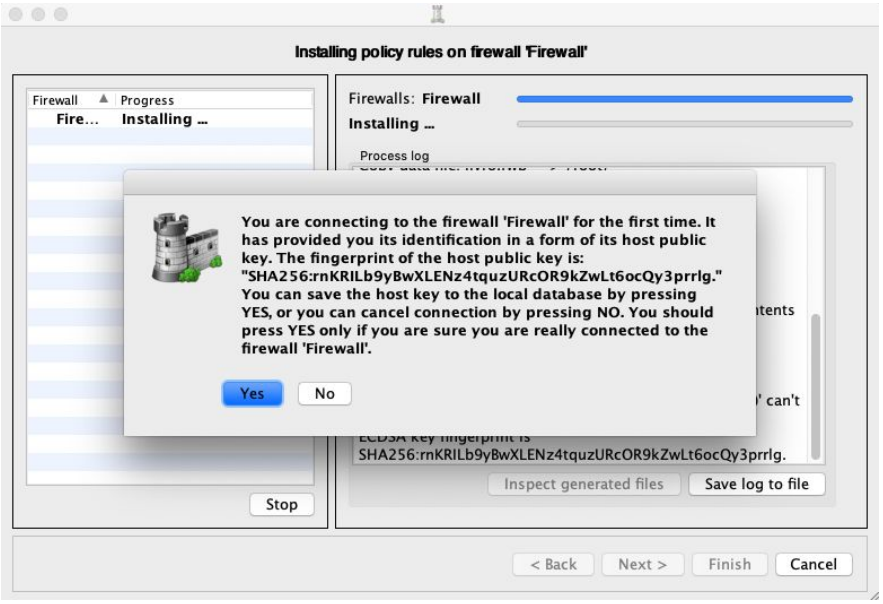


Image 2.16 - Establishing ssh to install FW

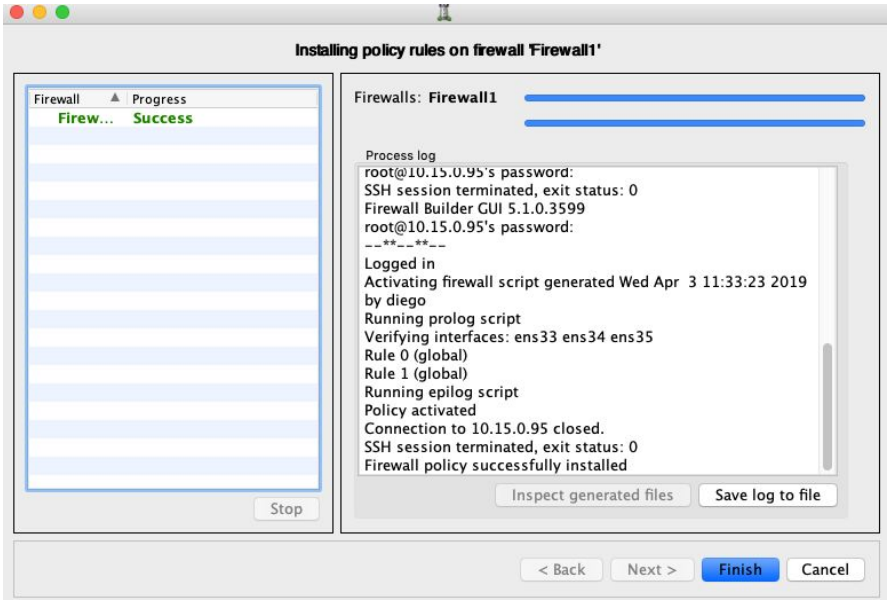


Image 2.17 - Successful installation of FW

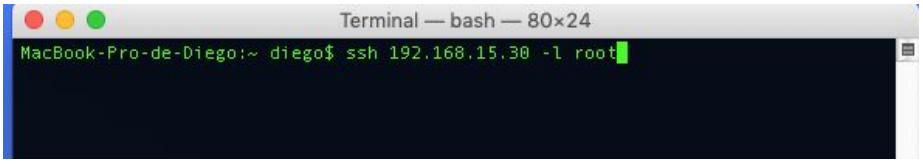


Image 2.18 - Login ssh on FW

If everything went well, it is now possible to access SSH on your FW. If not, some configuration was missing and redo the procedures described in this chapter.

Only SSH is enabled, try other services like Telnet (port 23) or a PING. They will be blocked by the FW in the last rule. To see the FW logs, do the command: **tailf /var/log/syslog**

Another interesting command to do in FW is the **iptables -L**. This command lists the FW rules and sees how many rows and FW tables you would have to manipulate by hand (editing a shell script). It's way easier for Firewall Builder.

To test the NAT table, we will create a client machine on the 172.16.1.0/24 network and make it go out to the internet through NAT in FW.
I will create, in the virtual environment, a Debian with a graphical environment to be the client machine. It will have the ip 172.16.1.10.

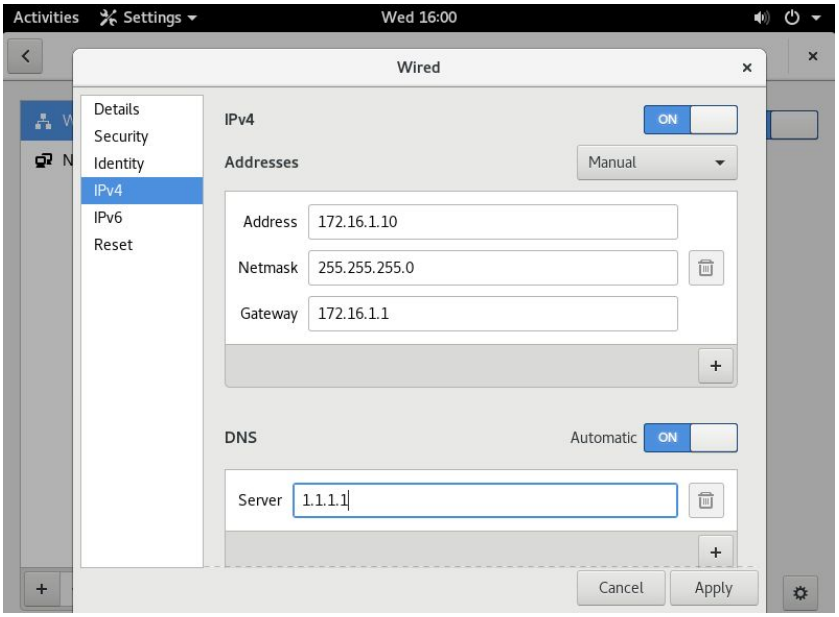


Image 2.19 - Network configuration of the client machine

See that I put the IP of the FW as a gateway. The DNS got an IP from the internet. Let's make the rules in FW for this machine to surf the internet.

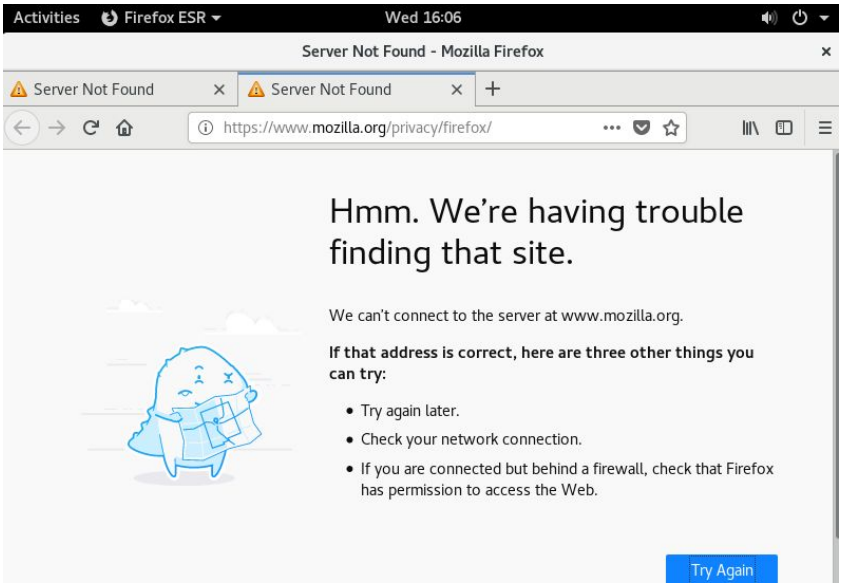


Image 2.20 - Without FW rules, without internet access

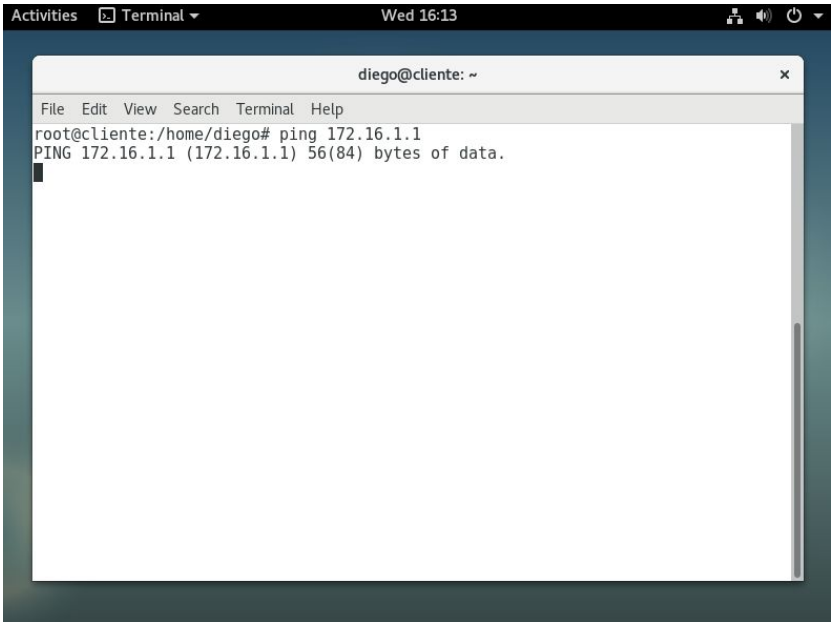


Image 2.21 - PING attempt at FW

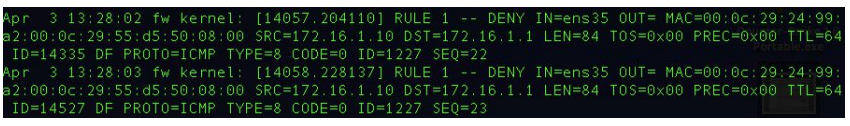


Image 2.22 - FW log denying PING (ICMP type 8)

Let's go to the rules in Firewall Builder:

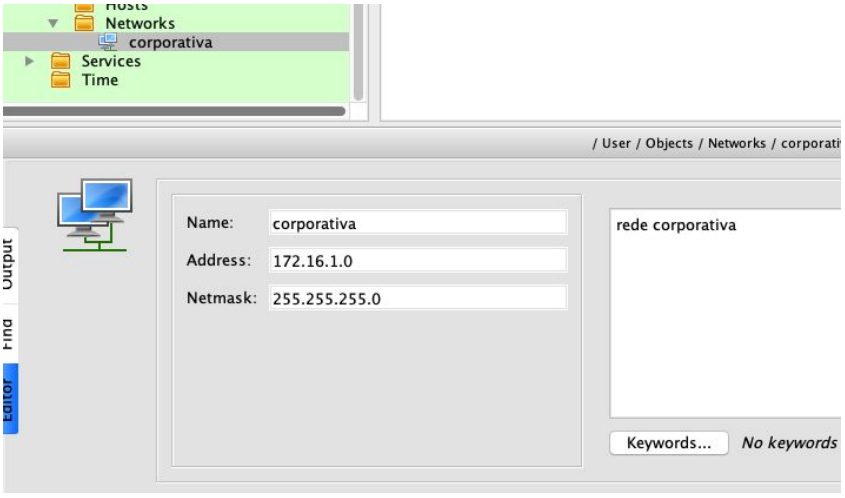


Image 2.23 - Network object creation

In rule number 1 below, we put the corporate object in SOURCE and in DESTINATION the rfc1918-nets object. This object refers to local networks, with IPs that are not routed on the Internet. See that there is the option **Negate** by right-clicking the object. When enabling this option, an **x** will appear in the lower left corner of the object, as shown in the image below. By denying the rfc1918 object, we are saying that this rule will only apply when the corporate network tries to access an IP that is not internal, therefore an Internet IP.

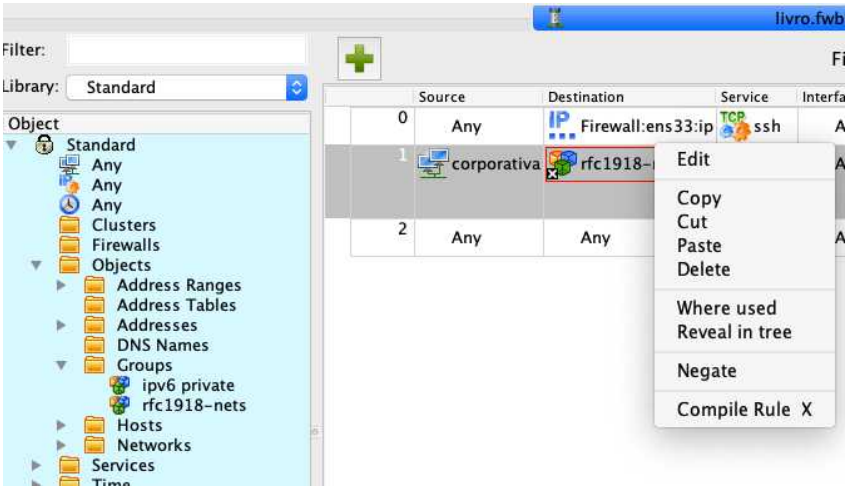


Image 2.24 - Object RFC1918-nets

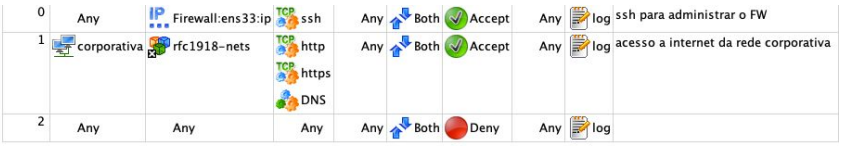


Image 2.25 - Rule 1 ready

The corporate network Internet access rule is listed above. We released HTTP, HTTPS and DNS services. However, NAT is still missing, because without it, you will not be able to navigate.

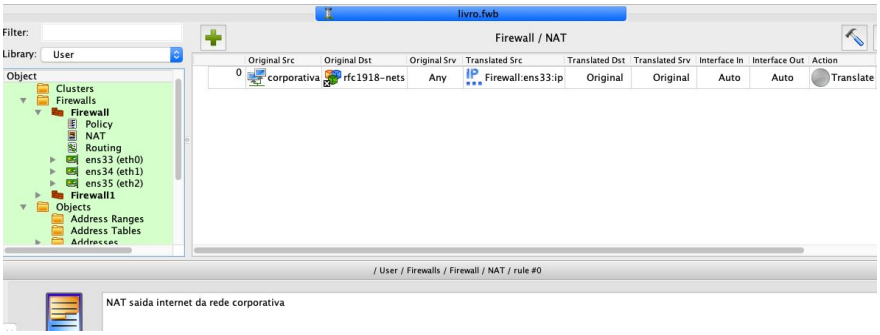


Image 2.26 - Rule NAT ready

The NAT rule was as above. Origin to corporate network, destination rfc1918 denied and translating origin to IP 192.168.15.30 (IP of the FW eth0 interface). Now it's compiling, installing and testing.

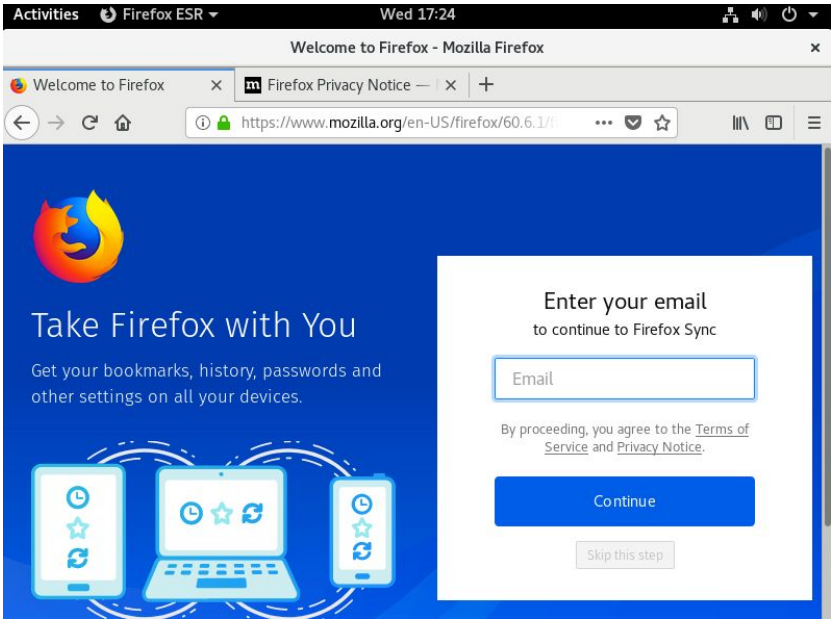


Image 2.27 - Client machine browsing the internet

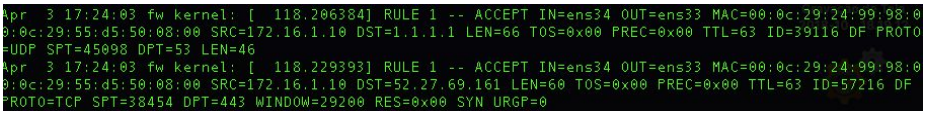


Image 2.28 - FW log accepting https and dns

2.2 Blocking Countries

One of the tools that state-of-the-art Firewalls (e.g. Checkpoint and Palo Alto) offer, with their so-called Next Generation Firewalls, is blocking of countries, which is often very useful for the Network Administrator to avoid having troubles. In order to implement this tool in our FW, we will use the tool **ipset**, a text file with the IP of all countries (https://pkgstore.datahub.io/core/geoip2-ipv4/geoip2-ipv4_csv/data/5ecd20f7df0f626a2270b71d4c725630/geoip2-ipv4_csv.csv) and a **shell script**.

The first step then is to install the ipset. For this, the command ***apt-get update && apt-get install ipset***. But first it is necessary to release our FW to access the internet with apt-get. It's basically releasing http, https and DNS for our FW.

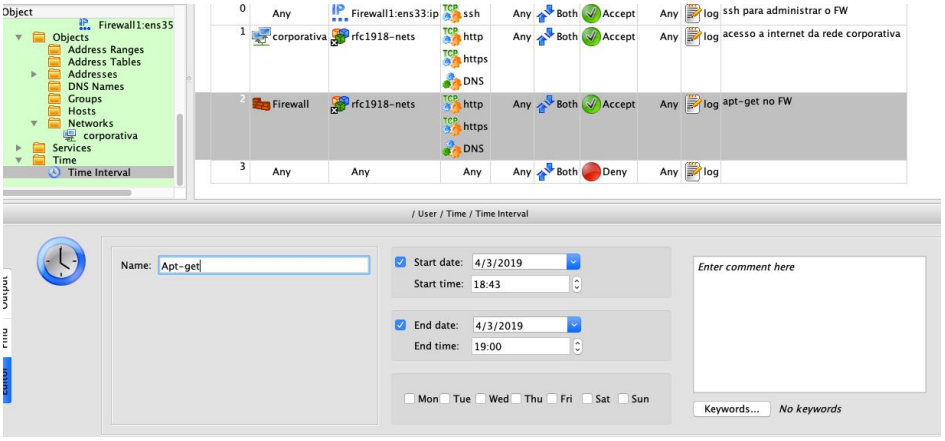


Image 2.29 - Rule apt-get with object TIME

I took the opportunity to use a time object, so the rule will be activated/deactivated on the date/time that I specified in the object. It can be useful in some cases.

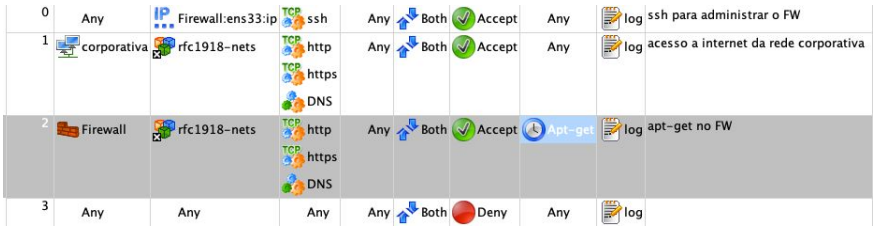


Image 2.30 - Insertion of rule 2

Ipset (http://ipset.netfilter.org) is an excellent tool that works inside the kernel to store IP, network and port. It makes use of HASH functions to expedite the consultation of this storage. The idea is, through the shell script, to search for the chosen country and store it in the ipset. Then we will insert the rules into iptables so that the FW will consult the ipset for each INPUT, FORWARD and OUTPUT. If the IP is in the ipset, the FW will block it.

Let's start by creating a table that will store the networks of the countries we want. The command is as follows:

- ipset create <name-of-list> hash:(net/ip/ip,port) hashsize 4096

We will use the list-name as blacklist-net, but it could be any name. **Blacklist** because the IPs or networks that are on the list are not welcome and **net** to identify that the list deals with network addresses. Then we have to decide what kind of data we want to store, we have three options (net, ip and ip / port). As we'll make a network list, we'll choose **net**. The last parameter concerns the size of our capacity to store networks, which is 4096. So, finally, the complete command will look like this:

- **ipset create blacklist-net hash:net hashsize 4096**

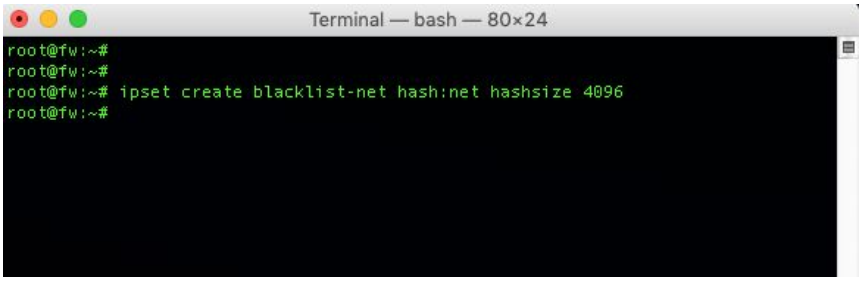


Image 2.31 - create blacklist-net

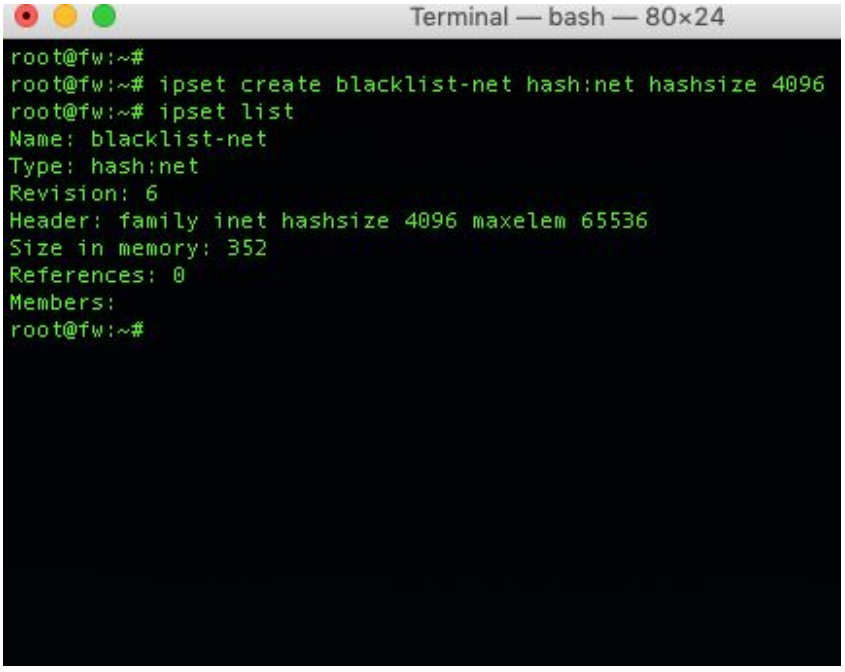


Image 2.32 - ipset list

After creating the blacklist-net, I used the command ***ipset list*** to list the created list. Of course it's still empty, so let's create Shell Script. Let's create a file called ***bloqueio-pais.sh*** on ***/root***.

Image 2.33 - bloqueio-pais.sh

Now put the Shell Script below:

```
#!/bin/bash
# Autor: Diego Brum
# 01/03/2019

##############
rm geoip2-ipv4_csv.*
wget https://pkgstore.datahub.io/core/geoip2-ipv4/geoip2-ipv4_csv/data/5ecd20f7df0f626a2270b71d4c725630/geoip2-ipv4_csv.csv
##############

PAIS_BLOCK_VAR="/root/PAIS-BLOCK-VAR.var"
TODOS_PAISES="/root/geoip2-ipv4_csv.csv"

echo
echo
echo "digite o nome do país:"
read pais
busca=$(cat "$TODOS_PAISES" |grep -i -w "\<$pais\>")

if [ ! -z "$busca" -a "$busca" != " " ]; then
        echo -e "Deseja \e[91mBloquear \e[39mou \e[94mLiberar \e[39m$pais ? (l/b)"
        read resp

        if [ "$resp" == "b" ]; then
                cat "$TODOS_PAISES" |grep -i -w "\<$pais\>" |awk -F',' '{print $1}' > "$PAIS_BLOCK_VAR"
                for i in $(cat "$PAIS_BLOCK_VAR"); do
                        ipset -A blacklist-net $i
                        echo -e "$pais  - $i -\e[91mBLOQUEADO"
                done
        elif [ "$resp" == "l" ]; then
                cat "$TODOS_PAISES" |grep -i -w "\<$pais\>" |awk -F',' '{print $1}' > "$PAIS_BLOCK_VAR"
                for i in $(cat "$PAIS_BLOCK_VAR"); do
                        ipset del blacklist-net $i
                        echo -e "$pais  - $i -\e[92mLIBERADO"
                done
        else
             exit 0
        fi
else
        echo "País não encontrado"
fi
echo -e "\e[39m "
exit 0
```

Image 2.34 - Shell Script Blocking Countries

```
root@fw:~# cd /root/
root@fw:~# ls
root@fw:~#
root@fw:~#
root@fw:~# nano bloqueio-pais.sh
root@fw:~#
root@fw:~# chmod u+x bloqueio-pais.sh
root@fw:~#
```

Image 2.35 - Execution permission to the owner

The Shell Script and the files used in this book are available for download at https://drive.google.com/drive/folders/1lc6iqZ19oFIhwAw8-qEKj6DDdjUdhnXS?usp=sharing.

To run, type ***./bloqueio-pais.sh***. Shell Script downloads the list of countries from the internet, asks which country to block or release (the name of the country has to be in English), searches for the country in the downloaded list and, if it finds it, blocks or releases it according to the choice.

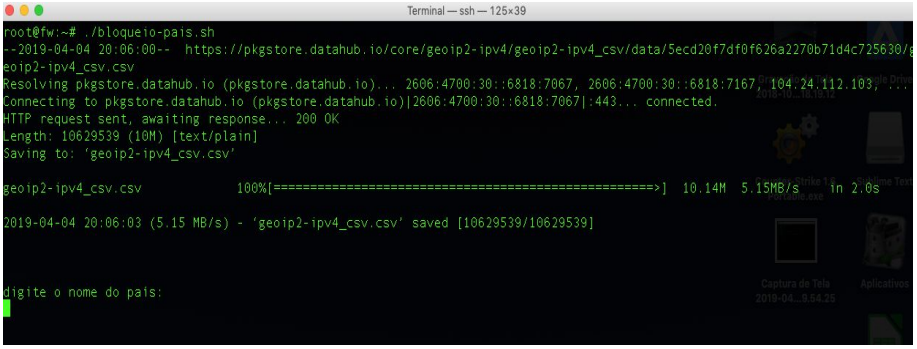


Image 2.36 - shell script execution

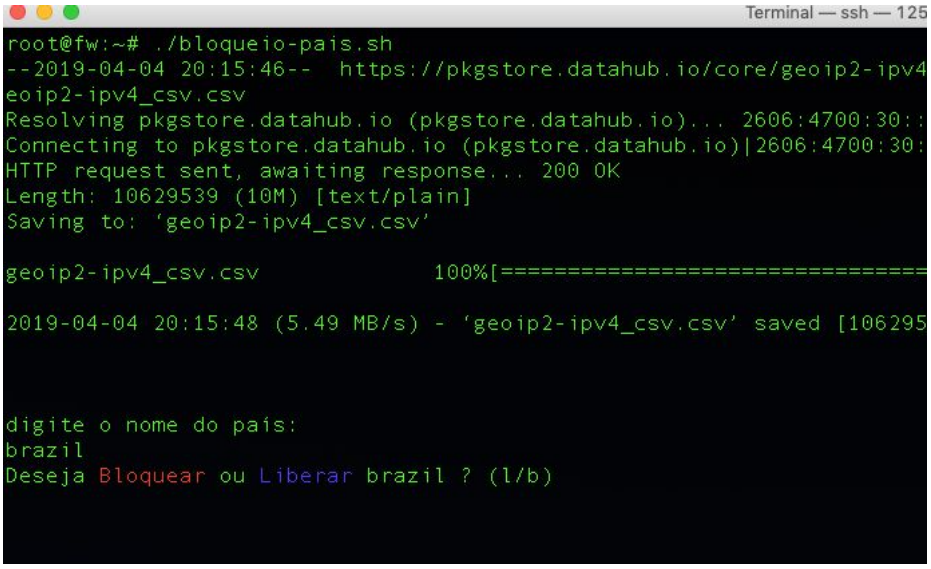


Image 2.37 - block Brazil

Image 2.38 - List of blocked IPs

If you want to release it, just run the script and put the letter l. Bearing in mind that for the time being this is not enough to block the country, the FW has yet to "tell" that it should consult the blacklist-net table for each INPUT, OUTPUT and FORWARD. For this it is necessary to execute the following commands:

iptables -I INPUT 1 -m set --match-set blacklist-net src -j DROP
iptables -I FORWARD 1 -m set --match-set blacklist-net src -j DROP
iptables -I OUTPUT 1 -m set --match-set blacklist-net dst -j DROP

See that the commands insert into iptables, at the top of the INPUT, FORWARD and OUTPUT chains, so number 1, rules that compare the

IPs that travel with our blacklist, if there is a match, the ip will be blocked (DROP).

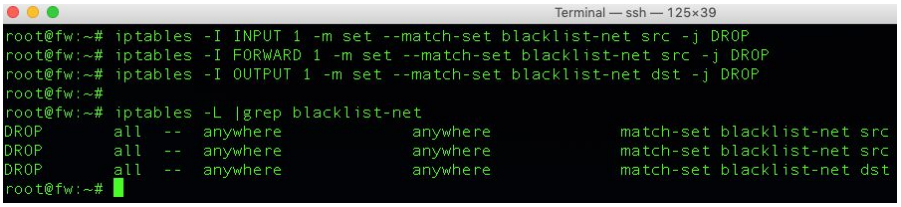


Image 2.39 - blacklist-net on iptables

I put the commands in and listed to check if they were in the FW. Now we are going to test on our client machine that was accessing the internet normally. Let's see what happens with the blockade in Brazil.

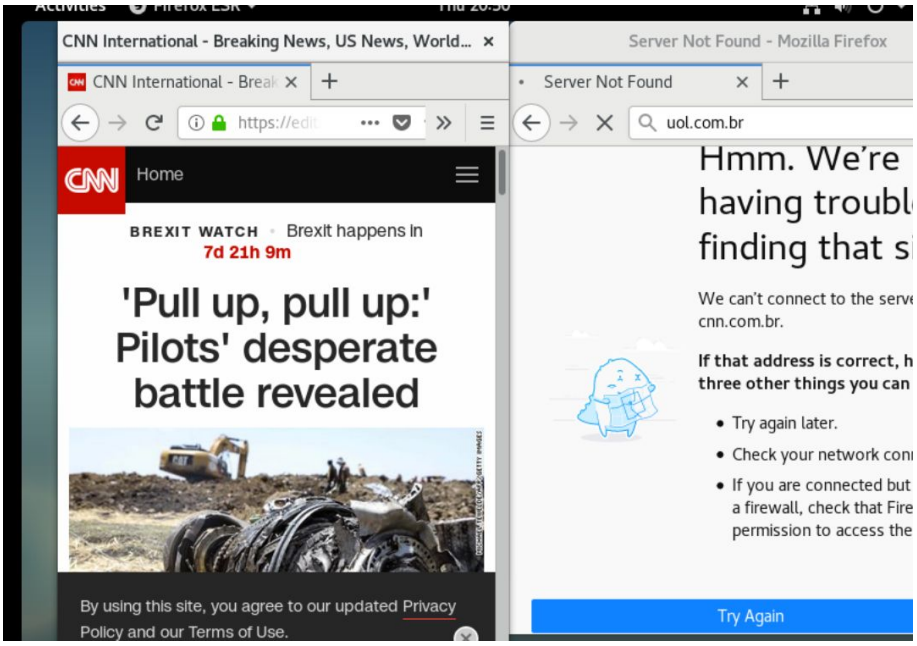


Image 2.40 - Brazilian website does not open

See that the Brazilian website will not open, as it got caught in the blockade of FW countries. We will release it and test again.

Image 2.41 - country release

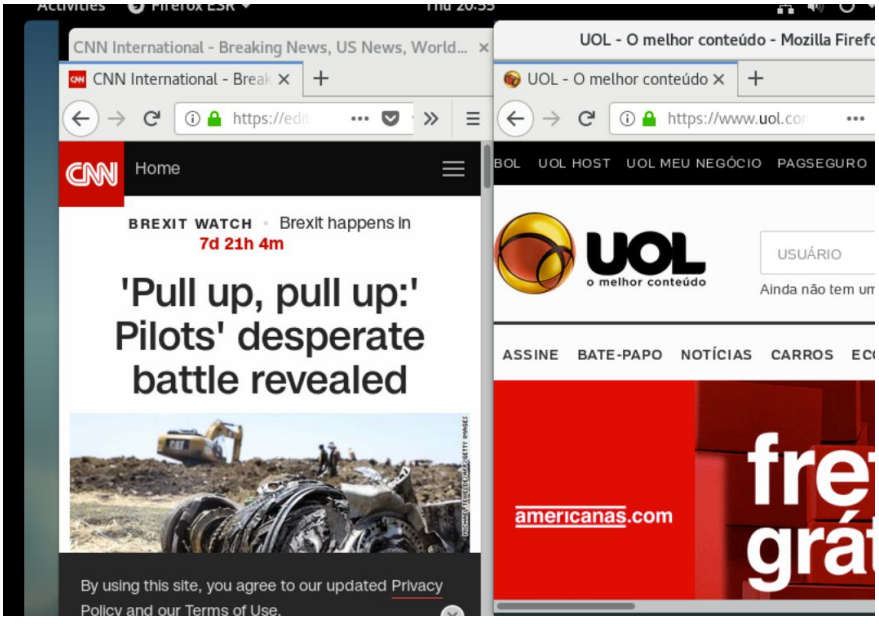


Image 2.42 - Normal access to the Brazilian website

2.3 Resume

The Firewall, even in IoT times, is still an important ally in network security. The less complex it is to configure it, the better for maintenance by Network Analysts and therefore more security. Firewall Builder brings this facility, making the task of creating rules and editing them simple. Just as we did in the password vault, I recommend using Gitlab to manage the versions of the FW file and keep access restricted to the project components only. In a production environment it will most likely be necessary to work with VLAN networks, in this case just install the vlan package on the FW (**apt-get install vlan**) and, when creating the network interfaces in Firewall Builder, put the number corresponding to the VLAN. Ex: If the ens35 interface (eth2), where the DMZ is, and the DMZ VLAN has TAG number 30, then the field **Name** interface will be **ens35.30**, that simple. We learned to block countries using an IP list provided on the internet, Shell Script and the ipset.

3 - HIDS

Host Intrusion Detection System or HIDS is a host IDS, that is, it constantly checks for possible malicious actions on a server. Unlike NIDS, whose scope is the network, HIDS operates at the Operating System level. Any modification of important files on your server, installation of new software, unsuccessful or successful login attempts via ssh, etc ... It is an important ally for the Security Analyst, after all, as I said in the Introduction, to know what happens in your network is vitally important.

The software I recommend is OSSEC (www.ossec.net). OSSEC is Open Source and Free, under the terms of the GNU - General Public License. Works with HASH to check the integrity of the files on the server. It is a very powerful software, being able to integrate with iptables and take some specific action. Send notifications by email and also via Syslog.

For this chapter, we will install two new virtual machines, one that will be our WEB server with OSSEC installed and another that will be our Syslog. Syslog is a very important service on your network, as it centralizes all the logs on your network in one place, facilitating the administration and backup of these logs, if applicable.

Image 3.1 - VM

At this point, we already have four virtual machines (FW, Client, Syslog and WebServer). The **FW** with three interfaces (192.168.15.30, 192.168.1.1 and 172.16.1.1), **Client** with an interface (172.16.1.10), **Syslog** with an interface (192.168.1.10) and **WebServer** with an interface (192.168.1.20). Remembering that the 172.16.1.0/24 network is the **corportiva** and 192.168.1.0/24 is the **DMZ**.

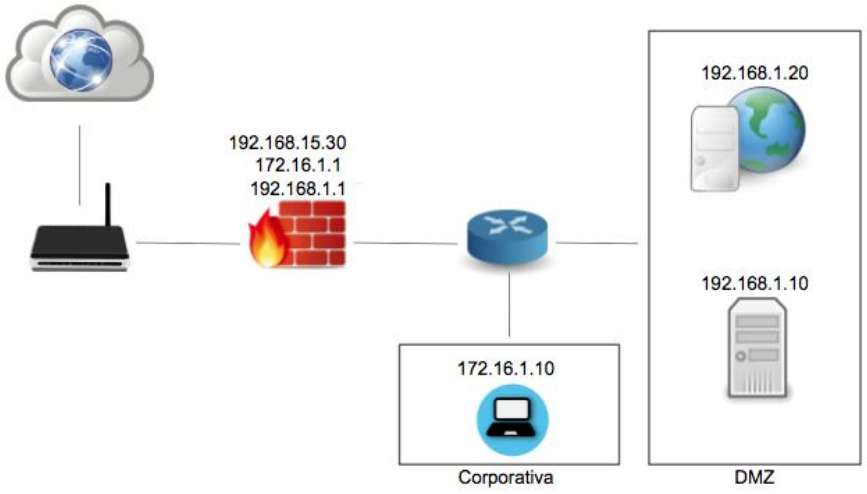


Image 3.2 - Network diagram

In order for us to install the necessary packages on the two new virtual machines, we will need to make the necessary releases on the FW.

0	Any	Firewall:ens33:ip	ssh	Any	Both	Accept	Any	log	ssh para administrar o FW
1	corporativa DMZ	rfc1918-nets	http https DNS	Any	Both	Accept	Any	log	acesso a internet das redes corporativa e DMZ
2	Firewall	rfc1918-nets	http https DNS	Any	Both	Accept	Any	log	apt-get no FW
3	Any	Any	Any	Any	Both	Deny	Any	log	

Image 3.3 - DMZ internet release

0	corporativa DMZ	rfc1918-nets	Any	Firewall:ens33:ip	Original	Original	Auto	Auto	Translate

Image 3.4 - DMZ release NAT for the internet

3.1 NTP

FW releases were made with rule 1 and NAT. We will start with the Syslog server. We will use the rsyslog package, which is a tool that works well and is easy to configure. Usually it is already installed in Debian, but if you need to install: ***apt-get update && apt-get install rsyslog***. It is not enough to have the logs if the date / time is incorrect, then we will install the NTP (Network Time Protocol) package, which is a protocol that synchronizes the clock so that it is always correct (***apt-get update && apt-get install ntp***).

If it is necessary to change the timezone, enter the command ***dpkg-reconfigure tzdata***.

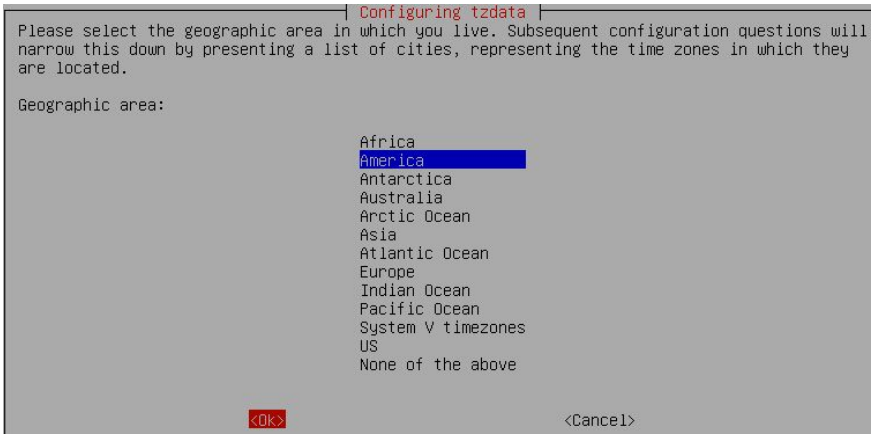


Image 3.5 - Configuração do timezone

In FW, ports 80, 443 and 53 were released, but not 123, which is the NTP port, without it there will be no clock synchronization. Let's release it.

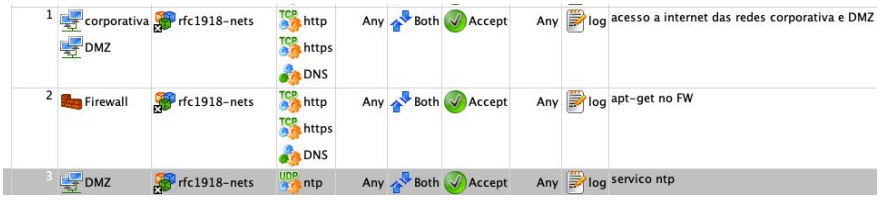


Image 3.6 - Release of NTP in the DMZ

After installing ntp and adjusting the timezone, perform the restart of the ntp service: ***systemctl restart ntp***. Sometimes, even following the previous steps, the date / time may not be correct, so to force synchronization, also install ntpdate (***apt-get update && apt-get install ntpdate***). To use, stop the ntp service (***systemctl stop ntp***) and execute ***ntpdate <servidor-ntp>.*** I will use the a.ntp.br server: ***ntpdate a.ntp.br***. Then come back with the ntp server: ***systemctl start ntp***. To check the date on the system, enter ***date***.

3.2 RSYSLOG

Now that we've taken care of setting the date / time, let's configure the rsyslog: ***nano /etc/rsyslog.conf.*** Uncomment the lines so that it looks the same below:

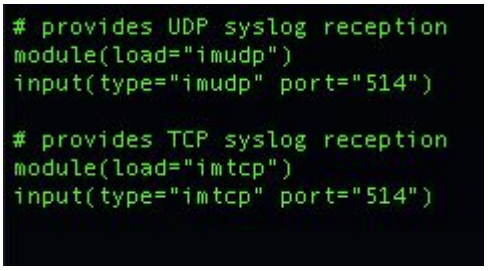


Image 3.7 - TCP and UDP port 514

The uncommented lines release rsyslog to receive the logs in TCP / UDP on port 514. After editing the rsyslog file, you must restore the application: ***systemctl restart rsyslog***.

We will now install a web server on our WebServer. We will install Apache (***apt-get update && apt-get install apache2***).

```
root@webserver:~# apt-get update && apt-get install apache2
Ign:1 http://ftp.br.debian.org/debian stretch InRelease
Get:2 http://ftp.br.debian.org/debian stretch-updates InRelease [91.0 kB]
Hit:3 http://security.debian.org/debian-security stretch/updates InRelease
Hit:4 http://ftp.br.debian.org/debian stretch Release
Fetched 91.0 kB in 1s (89.7 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,153 kB of archives.
After this operation, 7,158 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Image 3.8 - Apache installation

After installing, we can already test if the service is working. Just put the WebServer IP in the browser and, if everything is right, a standard Apache page will appear. But first we need to release the http service from the Corporate network to DMZ.

2	Firewall	rfc1918-nets	TCP http TCP https DNS	Any	Both	Accept	Any	log	apt-get no FW
3	DMZ	rfc1918-nets	UDP ntp	Any	Both	Accept	Any	log	servico ntp
4	corporativa	DMZ	TCP http	Any	Both	Accept	Any	log	acesso http na dmz

Image 3.9 - Rule 4 releasing http

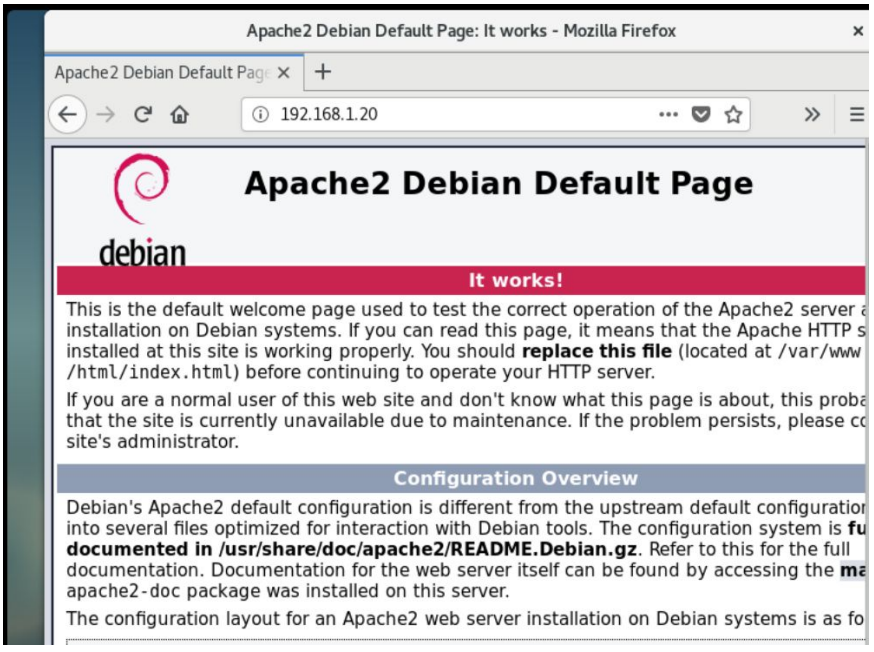


Image 3.10 - Client access to the WebServer page

We already have our Log server (Syslog) and the WebServer working, now we have to install OSSEC on the WebServer and see it working. First we will download the OSSEC installation package in the version compatible with Debian 9. The installation packages are in the https://updates.atomicorp.com/channels/ossec/debian/pool/main/o/ossec-hids-server/. We will download ossec-hids-server_3.3.0.6515stretch_amd64.deb using the wget command, which allows downloads via the http / https protocol.
wget https://updates.atomicorp.com/channels/ossec/debian/pool/main/o/ossec-hids-server/ossec-hids-server_3.3.0.6515stretch_amd64.deb

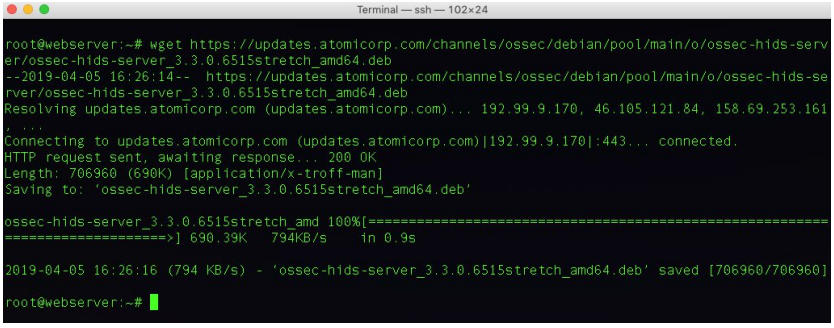


Image 3.11 - Download ossec package

Wget only downloads the Debian package. To install, at least here during the lab, some problems appeared, but if you follow the steps below at the end we will have ossec working:

1. ***dpkg -i ossec-hids-server_3.3.0.6515stretch_amd64.deb***.
2. ***apt-get install ossec-hids-server***
3. ***apt --fix-broken install***
4. ***apt-get install inotify-tools***

Image 3.12 - Ossec installed

Now let's edit the ossec configuration file:

nano /var/ossec/etc/ossec.conf. The settings on it are in XML format. Right in the header of the file we have the configuration for sending email. Very useful, but in this book we will make use of sending information only by syslog.

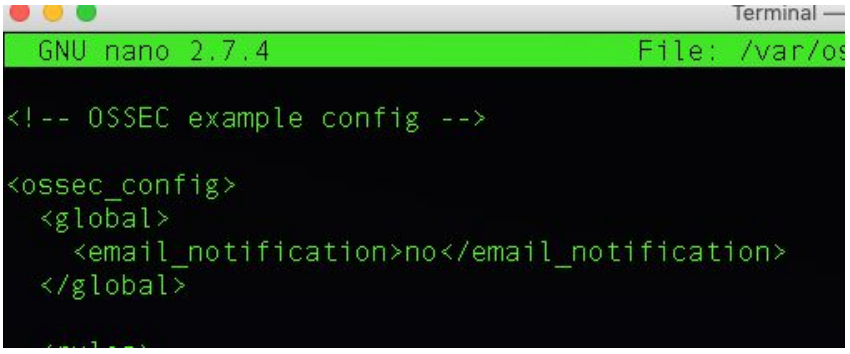


Image 3.13 - Email configuration

Let's go straight to the bottom of the configuration file to configure the syslog and see if it is working there on our log server. At the end of the file add the lines:

<syslog_output>
<server>192.168.1.10</server>
</syslog_output>

Image 3.14 - Syslog configuration

Save the document and run the following command: ***/var/ossec/bin/ossec-control enable client-syslog***. This will enable ossec to send messages via syslog. Then restart ossec: ***/var/ossec/bin/ossec-control restart***. See if when restarting ossec on the WebServer an alert appears on the Syslog. To do this, go to the

syslog machine (192.168.1.10) and execute the command ***tailf /var/log/message***.

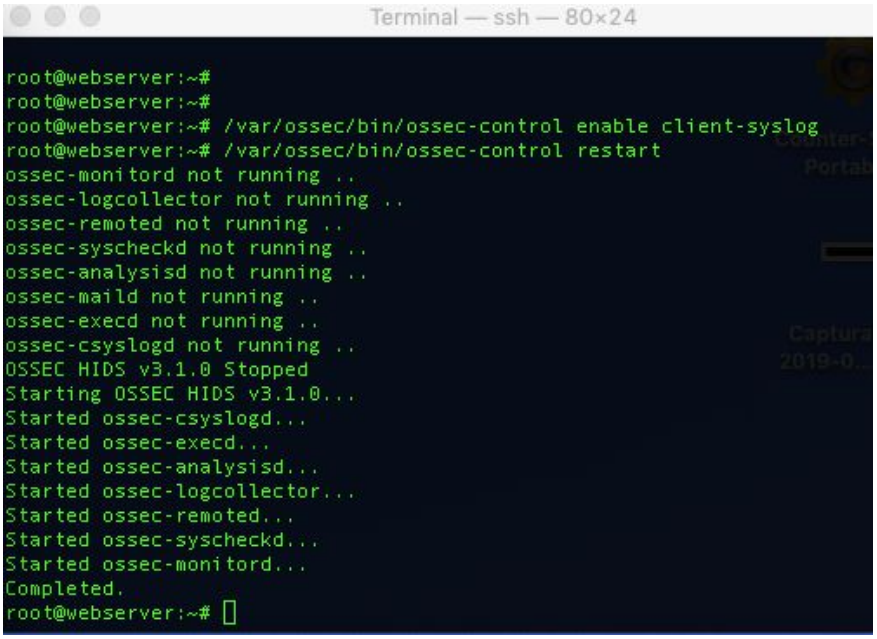


Image 3.15 - Restart ossec

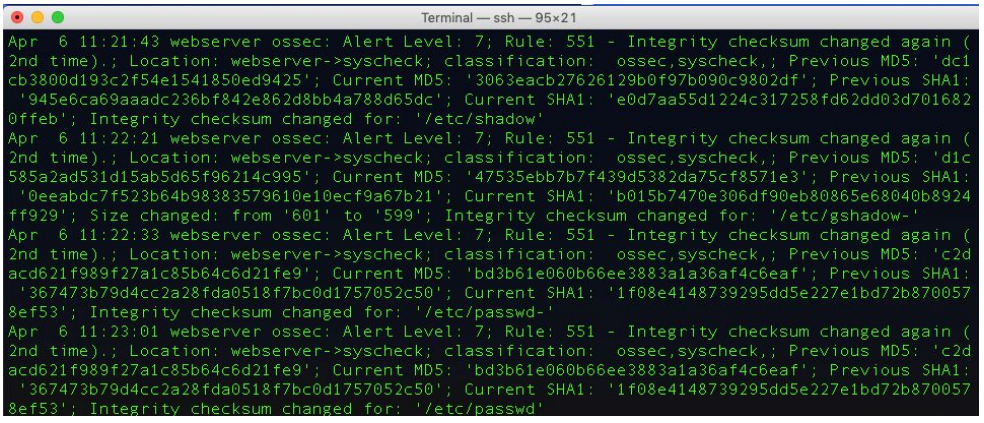


Image 3.16 - syslog receives alert from webserver

Right after configuring the ossec.conf file, enable ossec to send alerts via syslog and reset ossec-control, a few seconds later our syslog starts receiving alerts, as shown above. Note that there are changes alerts in the checksum of /etc/paswd, /etc/shadow, etc ... This happened because there were actually changes in these files when installing ossec, because it creates an ossec user in /etc/passwd and, for therefore, it also modifies /etc/shadow and /etc/group. See that ossec keeps you informed about everything that happens relevant on your server. Each alert has a number, which in these cases was 7 (“Bad word” matching.

They include words like “bad”, “error”, etc. These events are most of the time unclassified and may have some security relevance). The higher the number, the more concerned you should be.

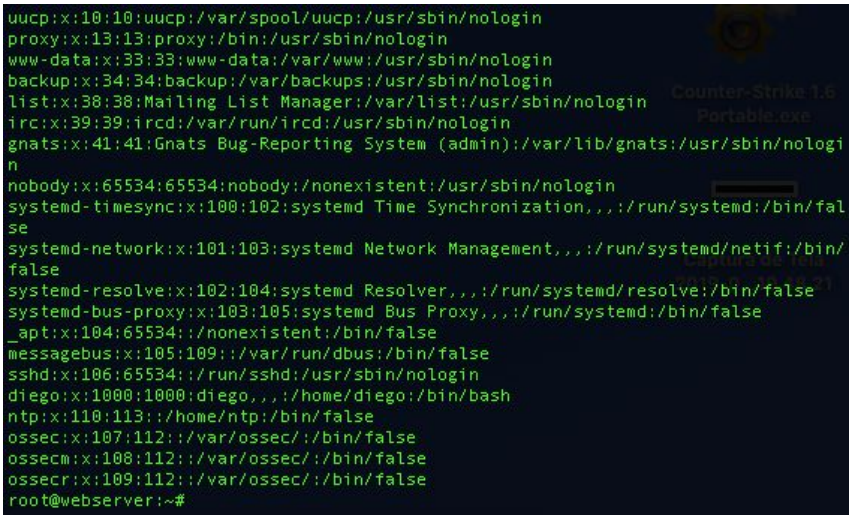


Image 3.17 - Ossec user in /etc/passwd

3.3 Rules Classification

Rules are classified at multiple levels. From the lowest level (00) to the maximum level 16. Some levels are not currently used. Other levels can be added between or after them.

The rules will be read from the highest level to the lowest level.

- 00 - Ignored - No action taken. Used to prevent false positives. These rules are checked before all others. They include events with no security relevance.

- 01 - None -

- 02 - Low priority system notification - System notification or status messages. They have no security relevance.

- 03 - Successful / authorized events - They include successful login attempts, firewall permission events, etc.

- 04 - Low system priority error - Errors related to incorrect settings or unused devices / applications. They have no security relevance and are usually caused by standard installations or software testing.

- 05 - User generated error - They include lost passwords, denied actions, etc. By themselves, they have no security relevance.

- 06 - Low relevance attack - They indicate a worm or a virus that does not affect the system (like red code for Apache servers, etc.). They also frequently include IDS events and often errors.

- 07 - Correspondence “bad word”. They include words like "bad", "error", etc. These events are most often not classified and may have some security relevance.

- 08 - First time seen - Include events seen for the first time. First time that an IDS event is triggered or the first time that a user has logged in. If you have just used OSSEC HIDS, these messages are likely to be frequent. After a while they should leave, it also includes relevant security actions (like starting a sniffer or something).

- 09 - Invalid source error - Include attempts to login as an unknown user or from an invalid source. It may have security relevance (especially if repeated). They also include errors related to the “admin” (root) account.

- 10 - Various user-generated errors - They include several bad passwords, several failed logins, etc. They can indicate an attack or a user may have forgotten their credentials.

- 11 - Health check warning - They include messages about modifying binaries or the presence of rootkits (by root check). If

you just modified your system configuration, you should be fine with the “syscheck” messages. They can indicate a successful attack. It also includes IDS events that will be ignored (high number of repetitions).

- 12 - Highly important event - They include error or warning messages from the system, kernel, etc. They can indicate an attack against a specific application.

- 13 - Unusual error (high importance) - Most of the time, it corresponds to a common attack pattern.

- 14 - Security event of high importance. Most often done with correlation and indicates an attack.

- 15 - Severe attack - There is no chance of false positives. Immediate attention is needed.

3.4 Rules Group

We can specify groups for specific rules. It is used for reasons of active response and for correlation.

We currently use the following groups:

- invalid_login
- authentication_success
- authentication_failed
- connection_attempt
- attacks
- adduser
- sshd
- ids
- firewall
- squid

- apache
- syslog

These data were taken from the official ossec documentation (https://ossec-docs.readthedocs.io/en/latest/manual/rules-decoders/rule-levels.html).

3.5 Web Server Configuration

Remember that we only configure ossec to send alerts via syslog and nothing else. As we have a Web Server, we want to adapt ossec to inform, for example, when someone creates a file with extension html, php, sh, etc ... Because it could be an attack in progress.

I created a basic website in the directory /var/www/html containing an index.html and an image Seguranca_open_source.jpg. I also created a folder called admin and inside it another index.html and the image Seguranca_open_source.jpg. The admin's idea is to simulate a restricted environment that we will use later.

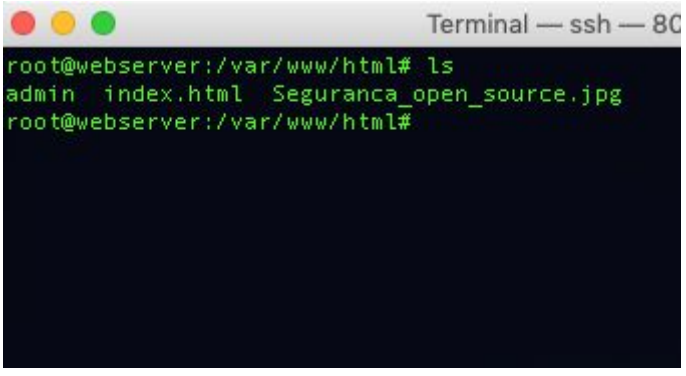


Image 3.18 - WebSite on /var/www/html

Image 3.19 - WebSite seen on client

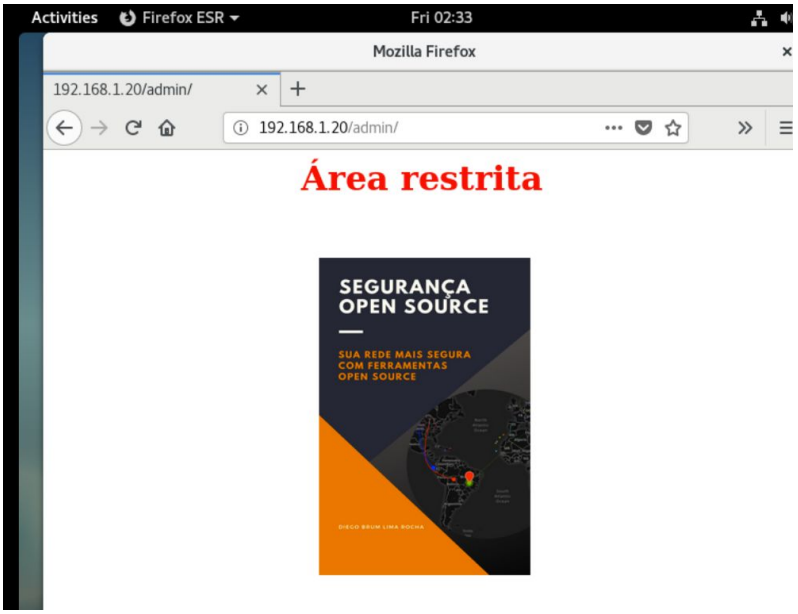


Image 3.20 - Admin seen on the client

With the website created, we will configure ossec to send messages when suspicious files are created in .php, .sh, .js, .html, .py formats in

the /var/www/html directory, or when modifying index.htm. First let's lower the ossec verification time to 3min and change some standard lines. Open /var/ossec/etc/ossec.conf and leave as below:

```
<syscheck>
  <!-- Frequency that syscheck is executed -- default every 20 hours -->
  <frequency>72000</frequency>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
  <directories check_all="yes">/bin,/sbin,/boot</directories>
```

Image 3.21 - ossec.conf before

```
<syscheck>
  <!-- Frequency that syscheck is executed -- default every 20 hours -->
  <frequency>180</frequency>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories report_changes="yes" realtime="yes" check_all="yes">/
etc,/usr/bin,/usr/sbin</directories>
  <directories report_changes="yes" realtime="yes" check_all="yes">/
bin,/sbin,/boot</directories>
```

Image 3.22 - ossec.conf after

This new configuration will alert changes in real time in the directories listed above. Now add below those rules the ones that will monitor /var /www/htm.

```
<alert_new_files>yes</alert_new_files>
<directories report_changes="yes" realtime="yes" check_all="yes">/
var/www/html</directories>
```

Image 3.23 - monitoring /var/www/html

Save the changes and there is still one more step to make everything work well. Ossec, even with the settings above, will still not send alerts, as there is a rule (<rule id = "554" level = "0">), in / var / ossec / rules, which is responsible for notifying new files, but it has level 0, and level 0 rules are not alerted. To resolve, we have to rewrite the rule in the /var/ossec/rules/local_rules.xml file. Add to the end of the file:

<rule id="554" level="7" overwrite="yes">
<category>ossec</category>
<decoded_as>syscheck_new_entry</decoded_as>
<description>File added to the system.</description>
<group>syscheck,</group>
</rule>

```
<rule id="554" level="7" overwrite="yes">
<category>ossec</category>
<decoded_as>syscheck_new_entry</decoded_as>
<description>File added to the system.</description>
<group>syscheck,</group>
</rule>

</group> <!-- SYSLOG,LOCAL -->



<!-- EOF -->
```

Image 3.24 - File local_rules.xml

With that done, let's restart ossec and create some "malicious" files and modify the index.html to see what happens. Leave the syslog open and with the command ***tailf /var/log/messages***.

```
Deleting PID file '/var/ossec/var/run/ossec-remoted-7886.pid' not used...
Killing ossec-monitord ..
Killing ossec-logcollector ..
ossec-remoted not running ..
Killing ossec-syscheckd ..
Killing ossec-analysisd ..
ossec-maild not running ..
Killing ossec-execd ..
Killing ossec-csyslogd ..
OSSEC HIDS v3.1.0 Stopped
Starting OSSEC HIDS v3.1.0 (by Trend Micro Inc.)...
Started ossec-csyslogd...
Started ossec-execd...
Started ossec-analysisd...
Started ossec-logcollector...
Started ossec-remoted...
Started ossec-syscheckd...
Started ossec-monitord...
Completed.
root@webserver:/var/www# date
Sat Apr  6 15:07:31 -03 2019
```

Image 3.25 - restart on ossec-control

I edited the index.html file and added the word HACKED, then created three "malicious" files in /var/www/html.

```
root@webserver:/var/www/html# nano index.html
root@webserver:/var/www/html#
root@webserver:/var/www/html#
root@webserver:/var/www/html# touch malicious.sh
root@webserver:/var/www/html# touch malicious.php
root@webserver:/var/www/html# touch malicious.html
```

Image 3.26 - creating files in the html folder

```
Apr  6 15:30:44 webserver ossec: Alert Level: 7; Rule: 554 - File added to the system.; Location: webserver->sysc
heck; classification:  local,syslog,syscheck,; New file '/var/www/html/malicious.sh' added to the file system.
Apr  6 15:30:44 webserver ossec: Alert Level: 7; Rule: 551 - Integrity checksum changed again (2nd time).; Locati
on: webserver->syscheck; classification:  ossec,syscheck,; Previous MD5: '42fbad178b4757aea976123360a0ec59'; Curr
ent MD5: '03d4c87da49efcb53363270e029f7fa7'; Previous SHA1: '7c3047f872106b110af38d5d91425dfff1a72e08'; Current S
HA1: '6d85c36ae51512e78aca51203efe1931cb9c4eda'; Size changed: from '244' to '311'; Integrity checksum changed fo
r: '/var/www/html/index.html'
Apr  6 15:30:44 webserver ossec: Alert Level: 7; Rule: 554 - File added to the system.; Location: webserver->sysc
heck; classification:  local,syslog,syscheck,; New file '/var/www/html/malicious.html' added to the file system.
Apr  6 15:30:44 webserver ossec: Alert Level: 7; Rule: 554 - File added to the system.; Location: webserver->sysc
heck; classification:  local,syslog,syscheck,; New file '/var/www/html/malicious.php' added to the file system.
```

Image 3.27 - syslog ossec

The figure above shows that ossec warned of the modification in index.html and the creation of "malicious" files.

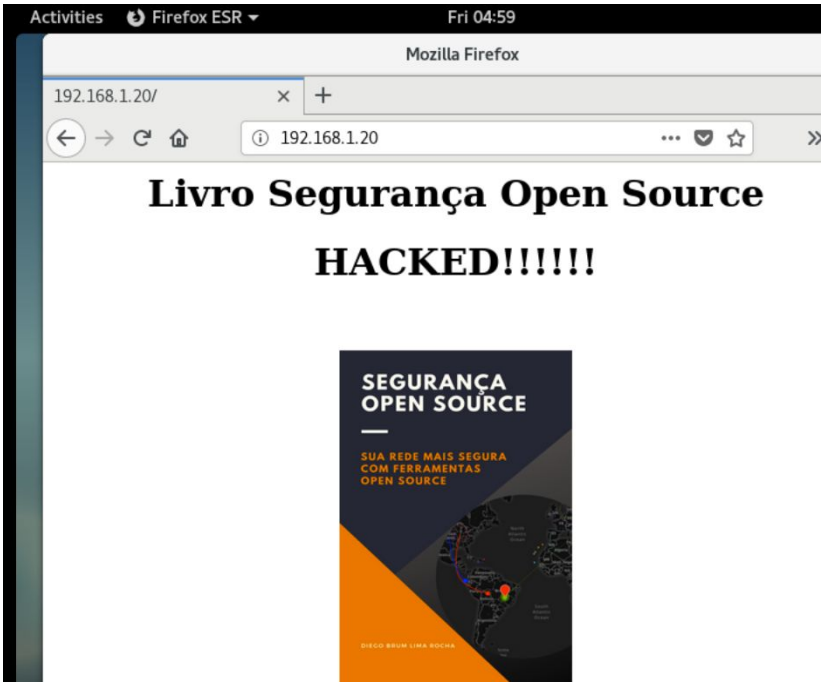


Image 3.28 - website "hacked"

To specify the types of files that ossec should check, just configure ossec.conf as shown below:

<directories report_changes="yes" realtime="yes" restrict=".php|.js|.py|.sh|.html" check_all="yes">/var/www/html</directories>

Be careful with the spaces when copying the codes shown here, as ossec may not work properly.

3.6 Resume

OSSEC is a HIDS that does its job well. The concepts exposed here were a small sample of what ossec is capable of. I think I would give another book if we were to go deeper into the tool. The idea is to show the tool's potential and encourage them to research and deepen the concepts to get the most out of it. We saw that ossec is like a "informer" on your network, warning you of things that happen at the Operating System level. We learned to receive alerts via syslog about changes and file creations on our Web Server. We also learned the importance of having an NTP service running, maintaining the correct times, after all, without it, the logs would not make much sense to the Security Analyst. We also learned to have a log server to centralize all the logs on the network, making it easier to analyze what happens on your network.

4 - Reverse Proxy

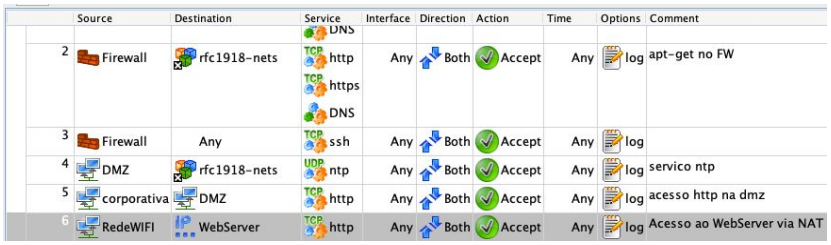
I like to think of information security as a series of layers that aim to hinder and prevent the network from suffering from malicious activities. Password Vault, Firewall, HIDS and the Reverse Proxy are part of these layers, all are important, so there is no hierarchy between them. Throughout the book I will still talk about other layers that all together form what I called Security Architecture in the Introduction to the Book. One of the pillars of this architecture is precisely the Reverse Proxy. Most already know the concept of a proxy, which is something that intermediates access to the internet, usually using software such as squid. In this case, if you want to access the internet, type the address in the browser and the request goes to the proxy, which makes the request on the internet site and returns it to you. The reverse proxy is exactly the "reverse", because the request comes from the internet and the website or server is on your network, it is up to the Reverse Proxy to receive the request from the internet and forward it to the web server that will normally be in your DMZ.

The interesting thing is that the Reverse Proxy creates a "barrier" between the Web Server and the Internet, because without it your Web Server would be "facing" the Internet and, therefore, extremely exposed. The Reverse Proxy also makes it possible to double your defenses, which can be implemented on both the Web Server and the Reverse Proxy.

The software we will use as a Reverse Proxy will be Apache. We have already used it for our Web Server and now we will upload an Apache that will receive requests and pass it on to our Web Server. But first, let's configure our Web Server to access without a reverse proxy. For this we will do a NAT and a new rule in the FW Policy, as below:

	Original Src	Original Dst	Original Srv	Translated Src	Translated Dst
0	corporativa DMZ	rfc1918-nets	Any	Firewall:ens33:ip	Original
1	RedeWIFI	Firewall:ens33:ip	http	Original	WebServer

Image 4.1 - Rule NAT number 1



	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
2	Firewall	rfc1918-nets	http https DNS	Any	Both	Accept	Any	log	apt-get no FW
3	Firewall	Any	ssh	Any	Both	Accept	Any	log	
4	DMZ	rfc1918-nets	ntp	Any	Both	Accept	Any	log	servico ntp
5	corporativa	DMZ	http	Any	Both	Accept	Any	log	acesso http na dmz
6	RedeWIFI	WebServer	http	Any	Both	Accept	Any	log	Acesso ao WebServer via NAT

Image 4.2 - Policy number 6

Image 4.3 - Access without reverse proxy

The configurations above made a NAT that allowed the access of my WIFI network (external to the virtualized environment) to the WebServer that is in the virtualized environment and behind the FW.

Now we will create another virtual machine that will be in the DMZ and will be our reverse Proxy. It will have the IP 192.168.1.30

.

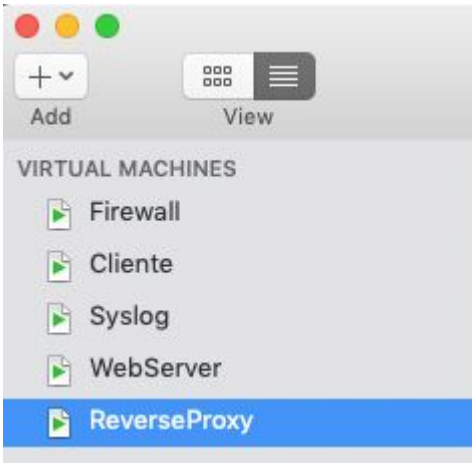


Image 4.4 - Created virtual machines

You will need to install Apache and enable two modules:

- mod_poxy
- mod_proxy_http

To install, follow the commands below:

- apt-get update && apt-get install apache2
- a2enmod proxy
- a2enmod proxy_http

```
root@reverse-proxy:~# apt-get update && apt-get install apache2
Get:1 http://security.debian.org/debian-security stretch/updates InRelease [94.3 kB]
Ign:2 http://ftp.br.debian.org/debian stretch InRelease
Get:3 http://ftp.br.debian.org/debian stretch-updates InRelease [91.0 kB]
Hit:4 http://ftp.br.debian.org/debian stretch Release
Fetched 185 kB in 0s (272 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-bla
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-
```

Image 4.5 - Installing Apache

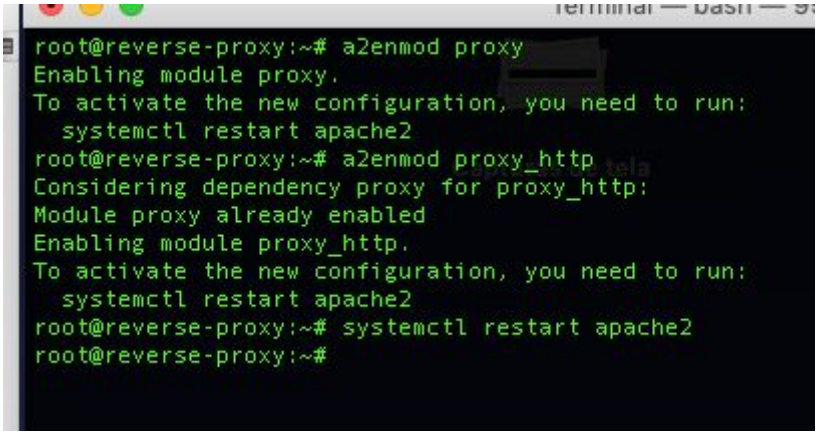


Image 4.6 - Enabling the proxy and proxy_http modules

Now we have to create a file called VirtualHost, which is where we will forward http requests to the WebServer. Let's go to /etc/apache2/sites-available. Then we will create in this directory a file with the name webserver.conf. And inside it put the following information:

<VirtualHost *:80>
ProxyPreserveHost On

ProxyPass / http://192.168.1.20/
ProxyPassReverse / http://192.168.1.20/
</VirtualHost>

Then save the file and type: ***a2ensite webserver.conf***. And finally: ***systemctl reload apache2***.

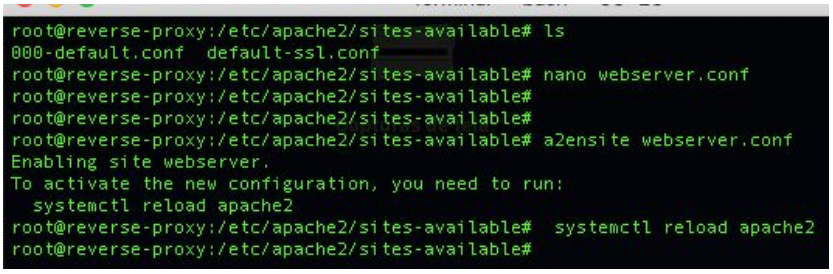


Image 4.7 - Configuration of webserver.conf

The a2ensite command creates a symbolic link in the sites-enabled directory.

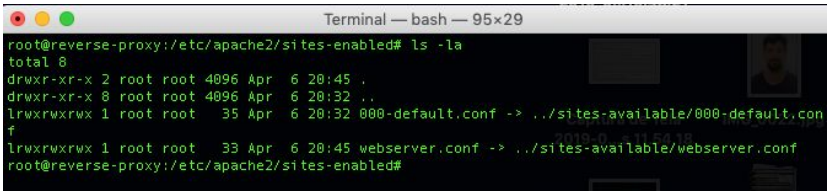


Image 4.8 - sites-enabled

We have to disable apache's default VirtualHost in /etc/apache2/sites-available/, 000-default.conf, as it is also listening on port 80 and will hinder the process. Then type the command ***a2dissite 000-default.conf && systemctl reload apache2***.

Our Reverse Proxy is basically ready, but it is still necessary to change the NAT and Policy rules to be able to pass http traffic through the Reverse Proxy.

Image 4.9 - NAT reverse proxy

Image 4.10 - Policy reverse proxy

Image 4.11 - Access to the webserver through the reverse proxy

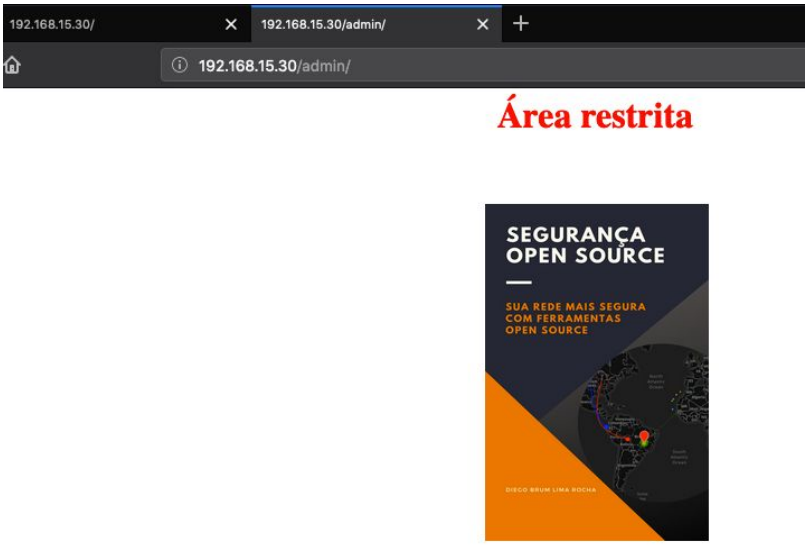


Image 4.12 - Access to the webserver /admin through the reverse proxy

We completed our Reverse Proxy, we gained security in the matter of having a machine between the internet and the WebServer, but there is still a lot to do, because as we saw above, the restricted area is accessible. It's time to start the HARDENING process.

4.1 HARDENING

On the Wikipedia website, Hardening is: "a process of mapping threats, mitigating risks and carrying out corrective activities, with a focus on infrastructure and the main objective of making it prepared to face attack attempts."

I would only add preventive activities as well. When removing services from a server that do not make sense, such as a service running on port 25 (e-mail) on a Web server, we are being preventive. Just like when configuring the linux file server to block files from running on certain partitions. But really many activities will also be remedial, like when you are attacked, check for vulnerability and make necessary adjustments.

There are many actions you can take to "harden" your server, here we will focus on the actions below:

- Automatic security upgrades
- checking for unnecessary services
- blocking access to the administrative area of your website
- blocking execution on certain partitions
- installation of forensic software
- limitation of HTTP protocol methods

There are numerous other hardening activities that can be done, but I believe that by doing the ones listed above, there will already be a huge increase in security on your web server.

4.1.1 Automatic Security Upgrades

When I started working with Information Security, I came across people who, oddly enough, were averse to updating servers. I had already read several books on security and it was obvious that not updating the servers sooner or later would cause problems. It didn't matter, it was kernel problems or exploitation of vulnerabilities happening frequently. Obviously, seeing that cracker banquet being served on the network every day, I started to implement the necessary updates throughout the network, even though it was against some people. But at the end of the process, "magically" the problems ended. Currently, with zero day exploits, you cannot afford not to update your systems as soon as a stable security patch comes out.

That said, we are going to implement an automatic security patch fix on your linux server. I will follow Gabriel Cánepa's tutorial, which is on the website www.tecmint.com. He says that the best system administrators are the lazy ones, in the sense of automating everything. I agree. Always be lazy when it comes to security, if you can automate an important process, do it!

We go to our WebServer, but the same process must be done on all your linux servers. To facilitate the process of having your Linux servers hardened since its conception, I recommend that you set up a Linux server with the techniques taught here and that it be the template (base) for creating all your servers. First you need to install the software below:

apt-get update -y && apt-get install unattended-upgrades apt-listchanges -y

Apt-listchanges will report what has been updated by unattended-upgrades. Then you need to edit the /etc/apt/apt.conf.d/50unattended-upgrades file and place the line below inside the Unattended-Upgrade :: Origins-Pattern:

Unattended-Upgrade::Mail "root";

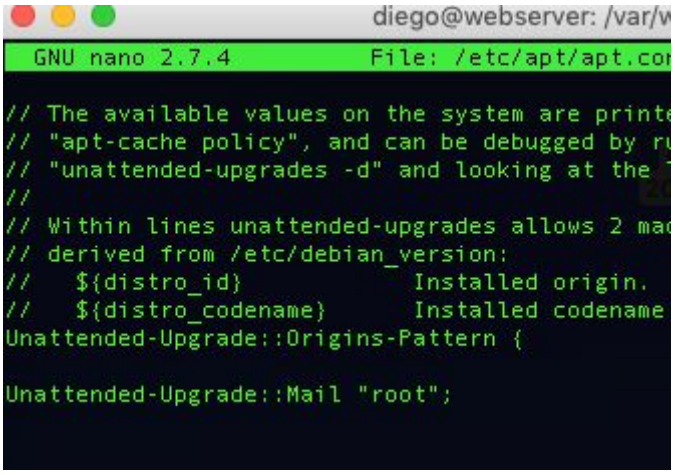


Image 4.13 - configuration of the 50unattended-upgrades

Then execute the command below:

dpkg-reconfigure -plow unattended-upgrades

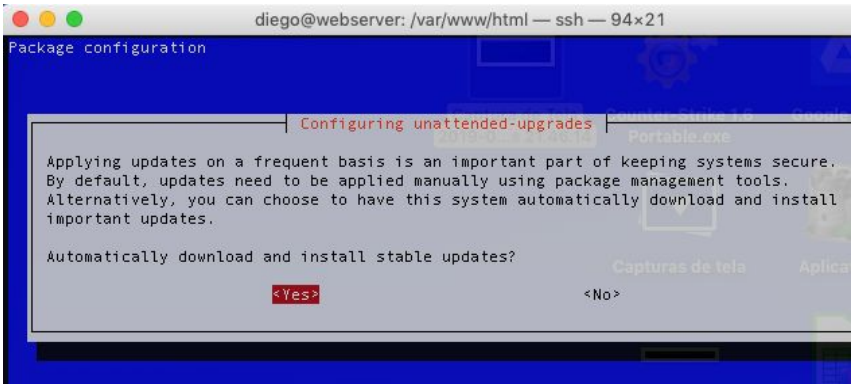


Image 4.14 - Automatically install upgrades

Automatically install stable updates? Sure!! It's actually **Yes**.

Then check if the /etc/apt/apt.conf.d/20auto-upgrades file has the lines below:

APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Unattended-Upgrade "1";

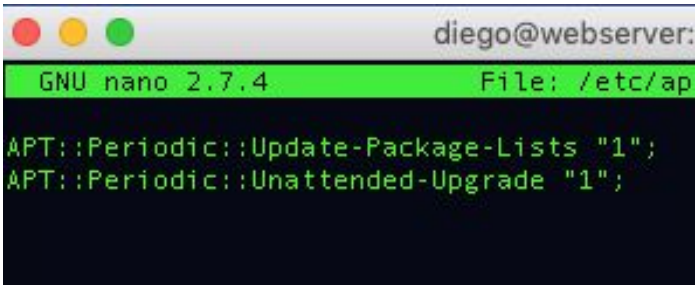


Image 4.15 - 20auto-upgrades

Add to /etc/apt/apt.conf.d/20auto-upgrades the line below:

APT::Periodic::Verbose "2";

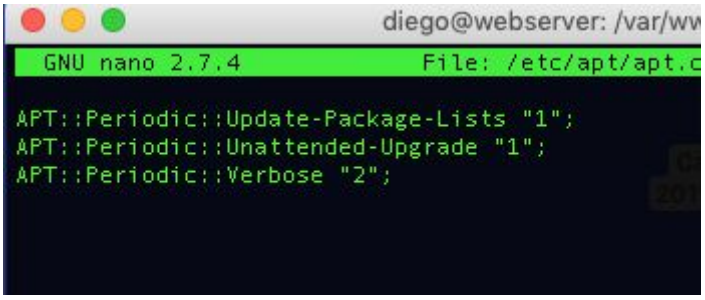


Image 4.16 - 20auto-upgrades new line

Ready! From now on your linux will be updated automatically and if you installed and configured your OSSEC correctly, you will receive notifications by email and/or syslog.

4.1.2 Blocking areas of your website

A good practice is to make the administrative areas of your website unavailable on the internet, as otherwise you will be the target of brute force attacks and at some point will unfortunately be successful. Of course, there are other ways to avoid the attack through Second Factor Authentication, blocking attempts, etc ... I prefer to simply block and leave administrative access to the intranet. Another important thing is not to let them browse your server through the Index, this makes the cracker happy to search the server for documents that you would not like to be exposed, etc…

Image 4.17 - WebSite indexing

Another thing we can omit is the version of our Apache. I'm not a big fan of security due to obscurity, which is precisely to hide information from others, but if it is to hinder the possible attacker, why not ?!

But let's go by parts. First we go to our webserver, in the webserver.conf file.

Image 4.18 - VirtualHost webserver

As it stands, everything is released. Let's leave it as shown below:

```
<VirtualHost *:80>
        <Directory /var/www/html/>
                Options -Indexes -Includes
                <LimitExcept HEAD>
                        Order deny,allow
                        allow from 172.16.1.0/24
```

```
            deny from all
        </LimitExcept>
    </Directory>

    <LocationMatch "^/admin">
        Order deny,allow
        allow from 172.16.1.0/24
        deny from all
    </LocationMatch>
</VirtualHost>
```

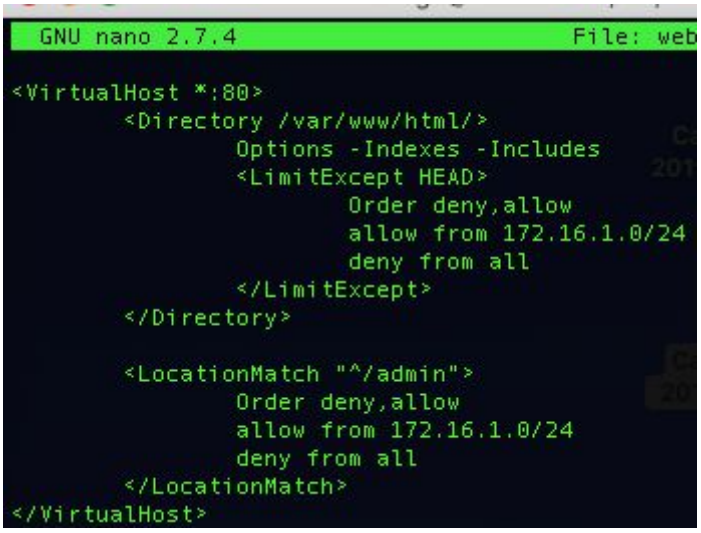


Image 4.19 - VirtualHost hardened

Now the Restricted Area (/admin) can only be accessed through the corporate network (172.16.1.0/24). In addition, it is not possible to index your site or include files. We also limited the HTTP protocol to use only the HEAD method. A lot of people don't know that HTTP has several methods and almost always you just need to enable the GET and POST methods. It is a tremendous vulnerability to leave all methods open if your website only needs, for example, GET to work. I've had problems with some HTTP methods that were used for attacks, so now I only release the methods that the site needs. Our webserver website is a simple static website and the HEAD method is sufficient for browsing. To

quell curiosity about HTTP methods, the list taken from the site follows https://developer.mozilla.org.

GET: The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

HEAD: The HEAD method requests a response identical to that of a GET request, but without the body of the response.

POST: The POST method is used to send an entity to the specified resource, often causing a change in state or side effects on the server.

PUT: The PUT method replaces all current representations of the target resource with the payload of the request.

DELETE: The DELETE method deletes the specified resource.

CONNECT: The CONNECT method establishes an encapsulation for the server identified by the target resource.

OPTIONS: The OPTIONS method is used to describe the communication options for the target resource.

TRACE: The TRACE method performs a message loopback test along the path to the target resource.

PATCH: The PATCH method is used to apply partial modifications to a resource.

Finally, we use LocationMatch to prevent the / admin from being accessed by a network other than the corporate network. Now we have increased the security of our webserver.

Image 4.20 - Normal access to the website

Image 4.21 - Access denied to /admin

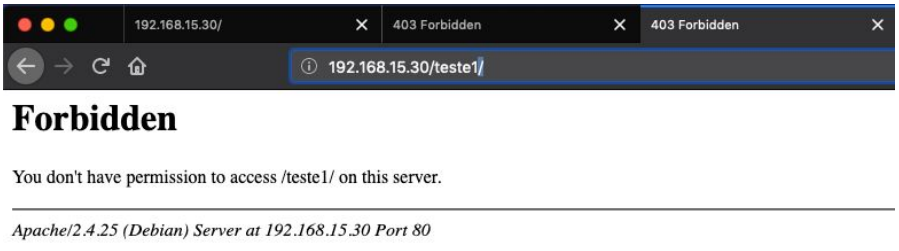


Image 4.22 - Access is denied to any other hidden directory

Image 4.23 - Normal network access 172.16.1.0/24

We can still improve further, we will create a website to replace this Forbbiden, which does not have a very scary effect on those who are eavesdropping. You can create a website on your network to alert in a more incisive way:

Image 4.24 - Modifying the Forbidden

To do this, add the line below to your virtualhost on the webserver:

ErrorDocument 403 http://192.168.1.20/aviso.html

Image 4.25 - ErrorDocument 403

We still need to remove the Apache version and do security by obscurity. To do this, just edit the file ***/etc/apache2/conf-available/security.conf*** and look for directives ***ServerTokens*** and ***ServerSignature***, putting ***Prod*** and ***Off*** respectively. After ***systemctl restart apache2***.

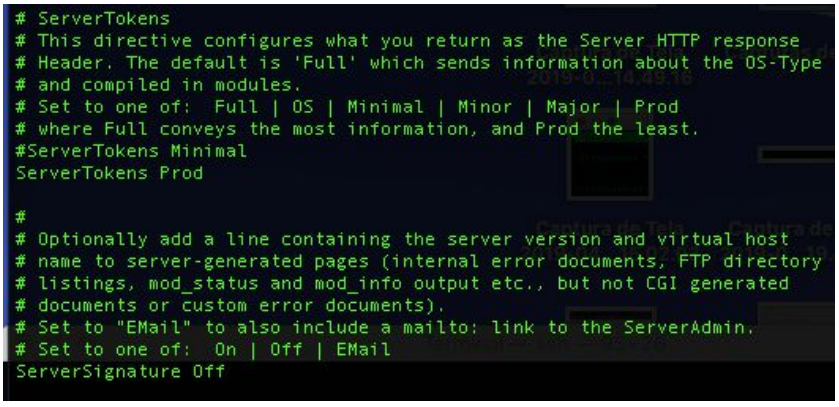


Image 4.26 - ServerTokens and ServerSignature

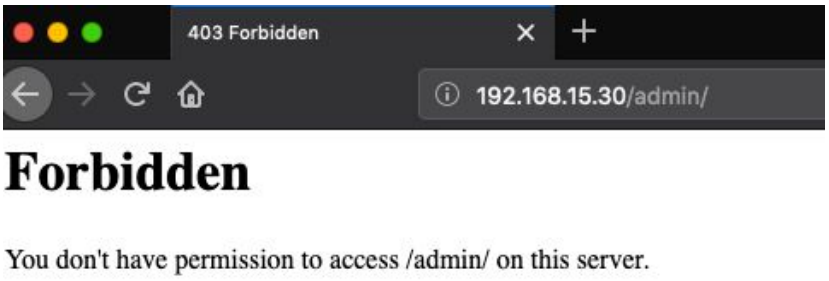


Image 4.27 - No information about Apache

4.1.3 Forensic Software Installation

When I talk about Forensics software I don't really mean exactly that, because the topic of Forensics is something that would make an entire book. The scope here, when using the term, actually refers to tools that help to discover gaps in your server, such as rootkits and outdated packages, as well as software that lets you know who is logged in, etc ... In the absence of a better term, I am using Forensics.

Let's start by installing the packages I mentioned:

apt-get install -y rkhunter chkrootkit unhide mtr-tiny apticron htop whowatch debsecan

Let's go to a basic explanation of each one:

- **rkhunter**: searches for rootkits on your operating system
- **chkrootkit**: another tool that searches for rootkits
- **unhide**: detects hidden processes
- **mtr-tiny**: powerful network diagnostic tool

- **apticron**: downloads the latest apt updates and leaves it in the cache
- **htop**: it is an excellent tool for viewing the machine's processes and resource usage. An improved version of the top.
- **whowatch**: valuable information about logged in users
- **debsecan**: scans known vulnerabilities in your installed packages.

Let's run each one to see how they work. Starting with: ***rkhunter --check***.

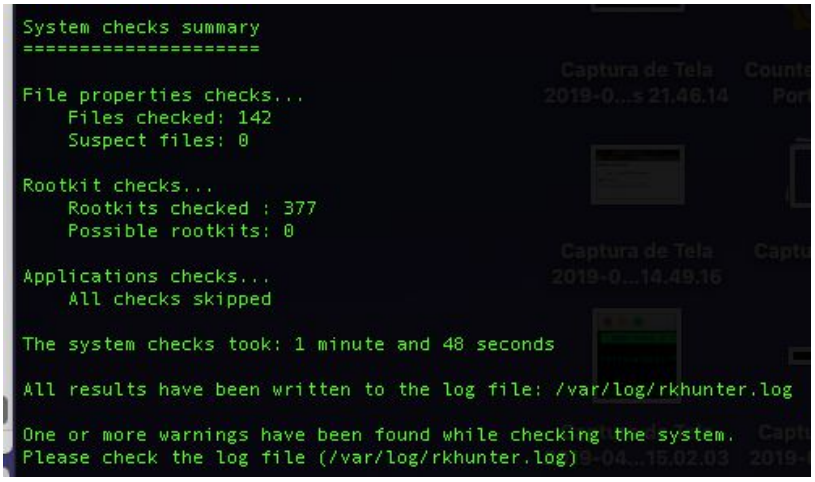


Image 4.28 - Analysis of rkhunter

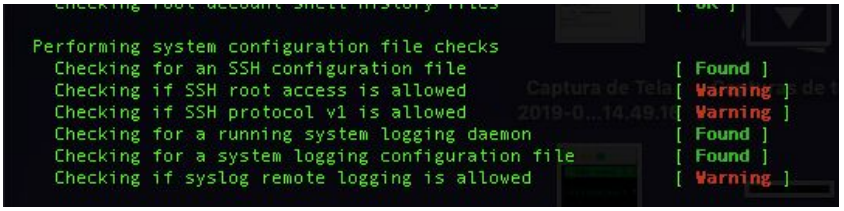


Image 4.29 - Found some warnings

Really, to facilitate access to the virtual machines for the book, I left root access for ssh, but it is not ideal and the tool warned about it. He also warned about the ssh protocol and syslog. Anyway, it gives you a view of what needs to be changed and you can adjust your server.

```
Searching for 64-bit Linux Rootkit ...                  nothing found
Searching for 64-bit Linux Rootkit modules...           nothing found
Searching for suspect PHP files...                      nothing found
Searching for anomalies in shell history files...       nothing found
Checking `asp'...                                       not infected
Checking `bindshell'...                                 not infected
Checking `lkm'...                                        OooPS, not expected 125092 valu
chkproc: Warning: Possible LKM Trojan installed
chkdirs: nothing detected
Checking `rexedcs'...                                   not found
Checking `sniffer'...                                   lo: not promisc and no packet sn
fer sockets
ens33: not promisc and no packet sniffer sockets
Checking `w55808'...                                    not infected
Checking `wted'...                                      chkwtmp: nothing deleted
Checking `scalper'...                                   not infected
Checking `slapper'...                                   not infected
Checking `z2'...                                        chklastlog: nothing deleted
Checking `chkutmp'...                                   chkutmp: nothing deleted
Checking `OSX_RSPLUG'...                                not infected
root@webserver:/var/www/html#
```

Image 4.30 - chkrootkit

A possible malware alert has appeared, but it is a false positive.

```
diego@webserver: /var/www/html — ssh — 94×24
root@webserver:/var/www/html# unhide sys
Unhide 20130526
Copyright © 2013 Yago Jesus & Patrick Gouin
License GPLv3+ : GNU GPL version 3 or later
http://www.unhide-forensics.info

NOTE : This version of unhide is for systems using Linux >= 2.6

Used options:
[*]Searching for Hidden processes through getpriority() scanning

[*]Searching for Hidden processes through getpgid() scanning

[*]Searching for Hidden processes through getsid() scanning

[*]Searching for Hidden processes through sched_getaffinity() scanning

[*]Searching for Hidden processes through sched_getparam() scanning

[*]Searching for Hidden processes through sched_getscheduler() scanning

[*]Searching for Hidden processes through sched_rr_get_interval() scanning

[*]Searching for Hidden processes through kill(..,0) scanning
```

Image 4.31 - unhide sys

```
Terminal — ssh — 80×21
root@fw:~# mtr
root@fw:~# mtr --report www.google.com
Start: Sun Apr  7 16:54:23 2019
HOST: fw                          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- 2804:7f3:888e:716c:9a97:d 10.0%    10    2.6  49.8   1.3 413.7 136.5
  2.|-- 2804:7f4:2000:1::d1       10.0%    10   23.9  52.9  21.1 269.6  81.5
  3.|-- 2804:7f4:2000:2004:7380:: 10.0%    10   22.0  36.0  21.9 125.2  33.7
  4.|-- 2001:12e0:100:5010:a090:5 30.0%    10   21.3  25.6  21.3  34.7   5.4
  5.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  6.|-- 2001:12e0:100:4021:a090:1 60.0%    10   40.1  43.2  40.1  45.8   2.4
  7.|-- 2001:12e0:100:1042:a090:1 90.0%    10   50.1  50.1  50.1  50.1   0.0
  8.|-- 2001:4860:1:1:0:49c1:0:1e  0.0%     9   57.9  44.8  37.5  59.2   8.2
  9.|-- 2001:4860:0:7a::1          0.0%     9   40.0  40.7  38.0  43.6   2.3
 10.|-- 2001:4860:0:1::d59         0.0%     9   57.4  46.2  38.9  74.3  12.1
 11.|-- 2800:3f0:4001:810::2004    0.0%     9   37.8  47.2  37.7  78.7  14.9
root@fw:~#
```

Image 4.32 - mtr --report www.google.com

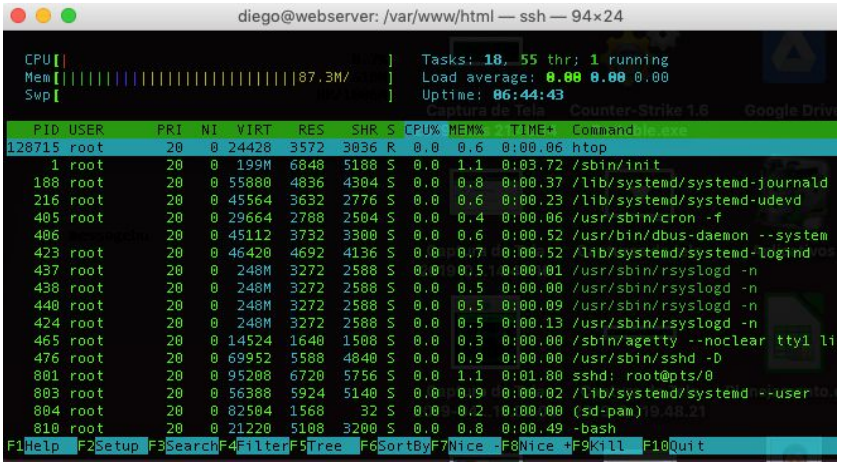


Image 4.33 - htop

Image 4.34 - whowatch

The whowatch tool makes it possible to click on the user's name and know which processes are running, among other information.

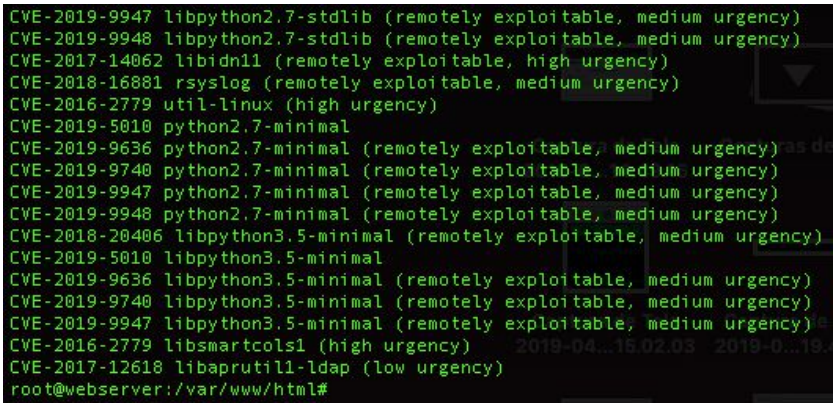


Image 4.35 -debsecan checks for vulnerabilities

Debsecan gives you a vulnerability diagnosis of the packages installed on your server. That way you can manage the risk and apply updates if applicable. You can use the command ***apt-get install -y $(debsecan debsecan --format packages)*** to update packages given as vulnerable by debsecan.

4.1.4 Unnecessary services and running on partitions

This topic closes the hardening section. Many of the software installed above already helps to know about unnecessary services, but a great tool is **netstat**. Always run netstat to see if strange services appear running on your server. If you have a web server, then you are expected to see web ports, such as 80 and 443.

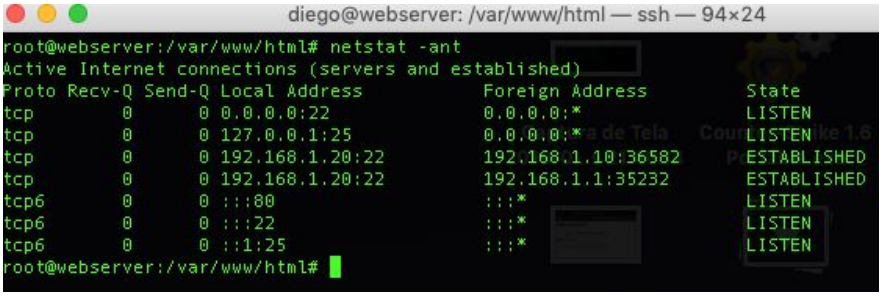


Image 4.36 - netstat -ant on WebServer

Note that ports 80 (http) and 22 (ssh) are listening, but an unwanted port 25 (email) has appeared. For more details, like knowing which application is using port 25, use **netstat -putona**.

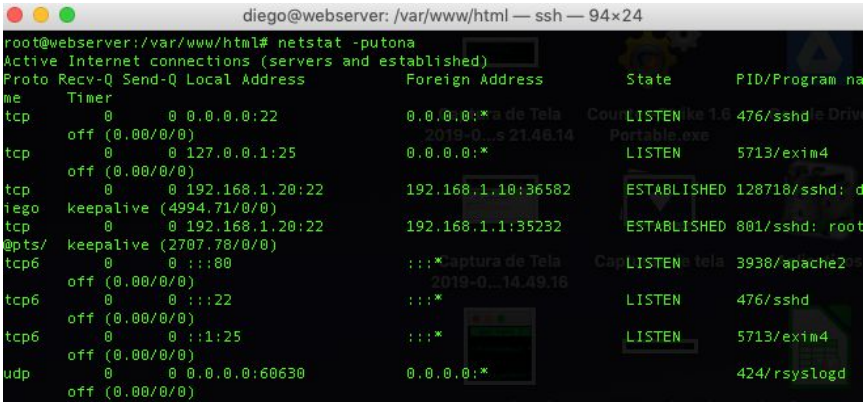


Image 4.37 - netstat -putona

Now it appears that the software that is using port 25 is such an exim4. Unfortunately this package comes with the standard Debian installation and I always have to remove it. To list the packages installed on your Debian, type ***dpkg -l***. To filter the output and find the exim4 package more easily, use grep. ***dpkg -l |grep exim4***.

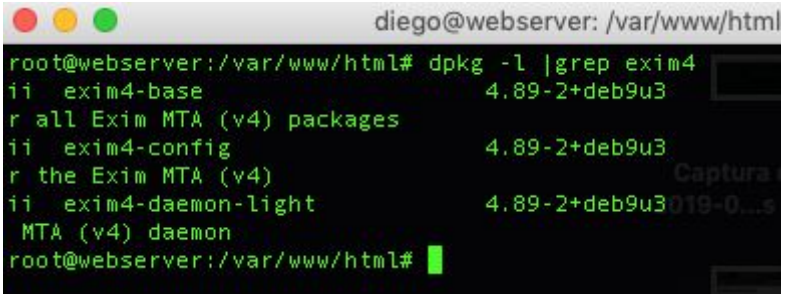


Image 4.38 - exim4

To delete the package: ***apt-get remove --purge exim4-base exim4-config exim4-daemon-light -y***.

Finally, it is sometimes interesting to block scripts from running or to run anything on certain partitions. An example is to block execution on the / tmp partition, as it is a partition where normally any user has full permission and malware or someone without administrator privileges may want to run something there. To block execution on the / home and / tmp partitions, simply type ***mount -o remount,rw,noexec /home /tmp***. This goes back to the partitions, but without power to execute (noexec).

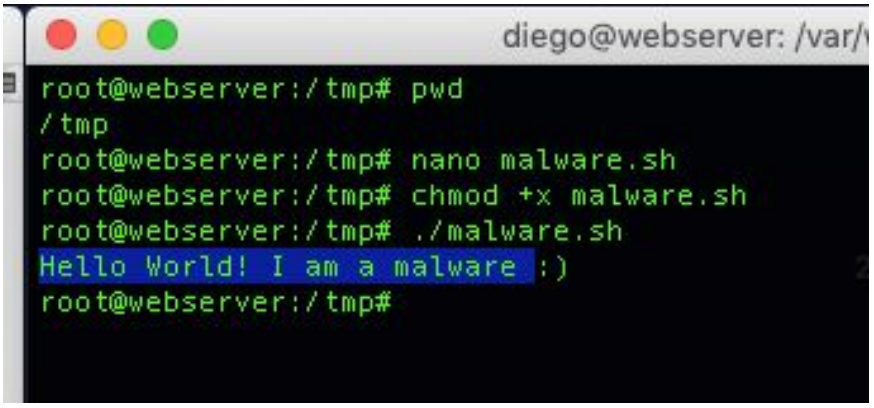


Image 4.39 - malware.sh on /tmp

I created and ran harmless malware called malware.sh in /tmp. Now let's run the command ***mount -o remount,rw,noexec /home /tmp***.

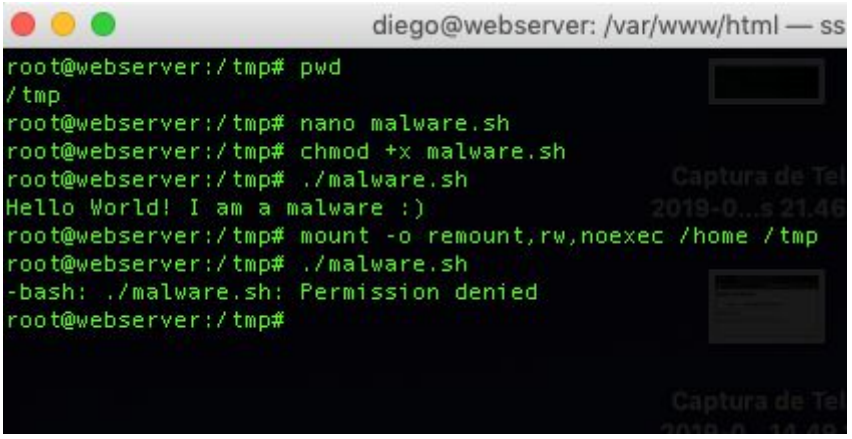


Image 4.40 - Permission denied

See that even though I am root I can no longer run anything in /tmp and /home.

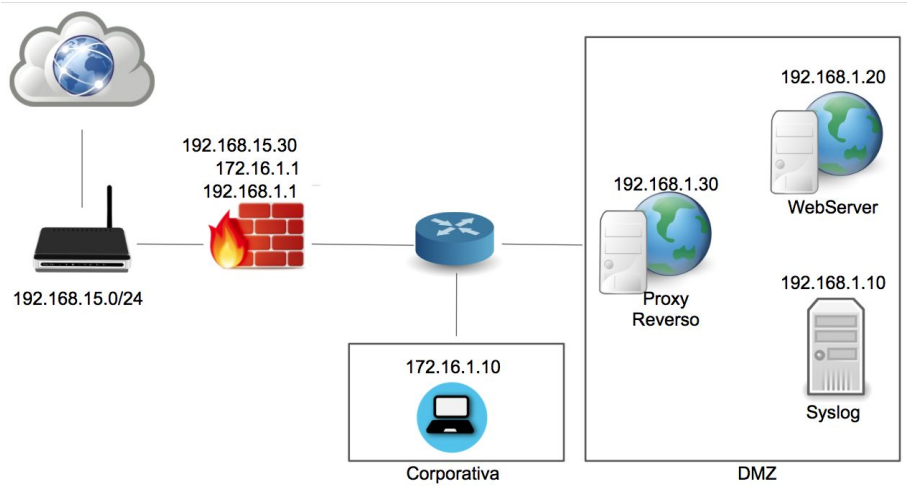


Image 50 - Network with reverse proxy

4.2 Resume

In this chapter we saw the importance of the reverse proxy and learned how to implement it. Then, we looked at the power to "harden" our server and the security gains we achieved in doing so. Recalling that this book is only providing an overview on the themes, so that there is still much to be studied in all subjects. But by placing a reverse proxy on your network and a hardening on your network templates, in order to implement security already in the server design, your network will make a big leap in security.

5 - Web Application Firewall

This tool is spectacular and essential. Below is a list of features and attacks mitigated by ModSecurity:

- SQL Injection (SQLi)
- Cross Site Scripting (XSS)
- Local File Inclusion (LFI)
- Remote File Inclusion (RFI)
- Remote Code Execution (RCE)
- PHP Code Injection
- HTTP Protocol Violations
- Shellshock
- Session Fixation
- Scanner Detection
- Metadata/Error Leakages
- Project Honey Pot Blacklist
- GeoIP Country Blocking

ModSecurity is an Apache module that together with signatures (OWASP ModSecurity Core Rule Set (CRS)) from attacks protect your web server. Subscriptions have the free version and the paid version. But the free version will already resolve 90% or more of the attacks on your website. Another cool tool is GeoIP Country Blocking. Just as we did in FW, in WAF (Web Application Firewall) there is also a way to block by countries. It happens that often the cracker is in the target country, but to hinder its discovery, it uses international proxies.

For this chapter I will use the linode tutorial (https://www.linode.com/docs/web-servers/apache-tips-and-tricks/configure-modsecurity-on-apache/).

First we will install Apache's modSecurity on our Reverse Proxy. But first, we have to resolve a bug that is happening when installing ModSecurity on Debian 9. I tried every way to make it work, but when installing the apache library, both libapache2-mod-security2 and libapache2-modsecurity do apache stops working and I was unable to verify the causes of the error messages. So, as often happens in computing, we have to find a workaround. And I found:

1. Install a new reverse proxy with Debian 8.11 (the latest version before 9).
2. Install Apache and configure it as shown in the Reverse Proxy chapter.
3. Install the new libapache2-mod-security2 library on the new reverse proxy with Debian 8.11. (***apt-get install libapache2-mod-security2***).
4. Upgrade from Debian 8 to 9.
    a. ***apt-get update && apt-get upgrade -y && apt-get dist-upgrade***
    b. edit the ***/etc/apt/sources.list***. Replace **jessie** with ***stretch***
    c. ***apt-get update && apt-get upgrade -y && apt-get dist-upgrade***

Following the steps above we will have our Debian 9, in the latest version, with ModSecurity and Apache working perfectly.

Resuming our ModSecurity installation process, we still need to download the signatures and make some settings. Type the command below:

mv /etc/modsecurity/modsecurity.conf-recommended modsecurity.conf

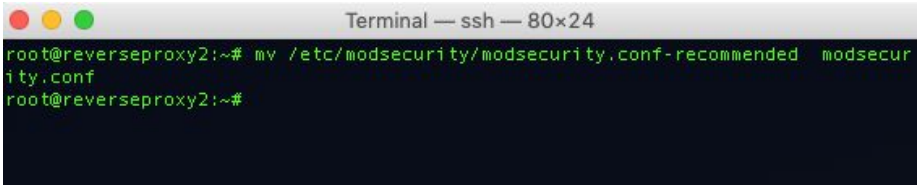


Image 5.1 - modsecurity.conf

Now we are going to install Git (distributed version control system), the same used by GitLab, which I talked about in the Password Vault and Firewall chapters. As modsecurity is maintained by a community that develops and contributes to the project, I suggest downloading git so that we can download subscriptions directly from the project website.

apt-get install git

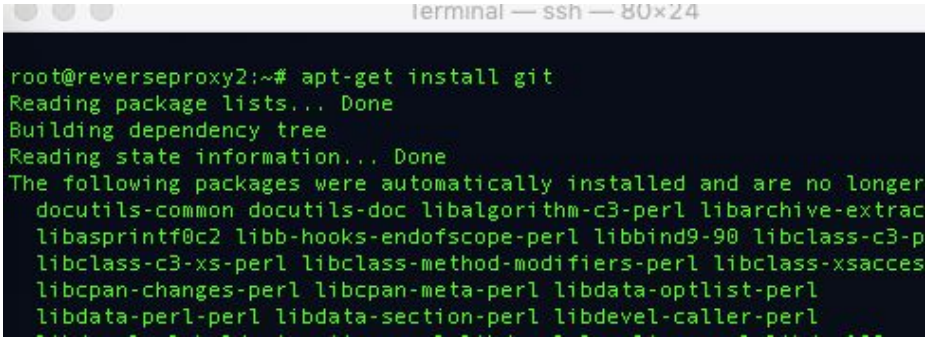


Image 5.2 - apt-get install git

git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git

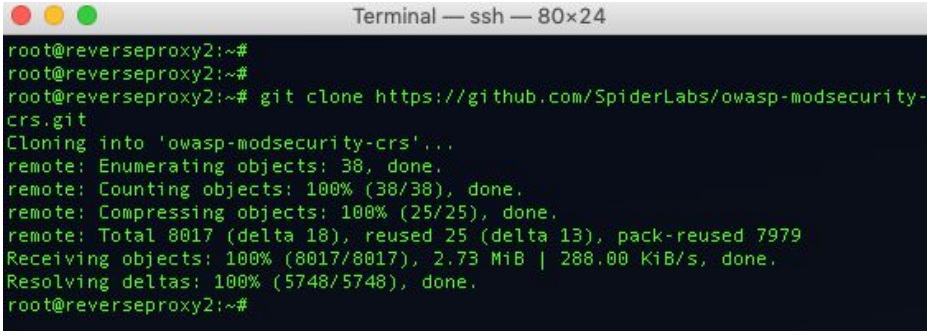


Image 5.3 - git clone

cd owasp-modsecurity-crs
mv crs-setup.conf.example /etc/modsecurity/crs-setup.conf
mv rules/ /etc/modsecurity/

Image 5.4 - owasp-modsecurity-crs

Now let's edit the file ***/etc/apache2/mods-available/security2.conf***. And do as below:

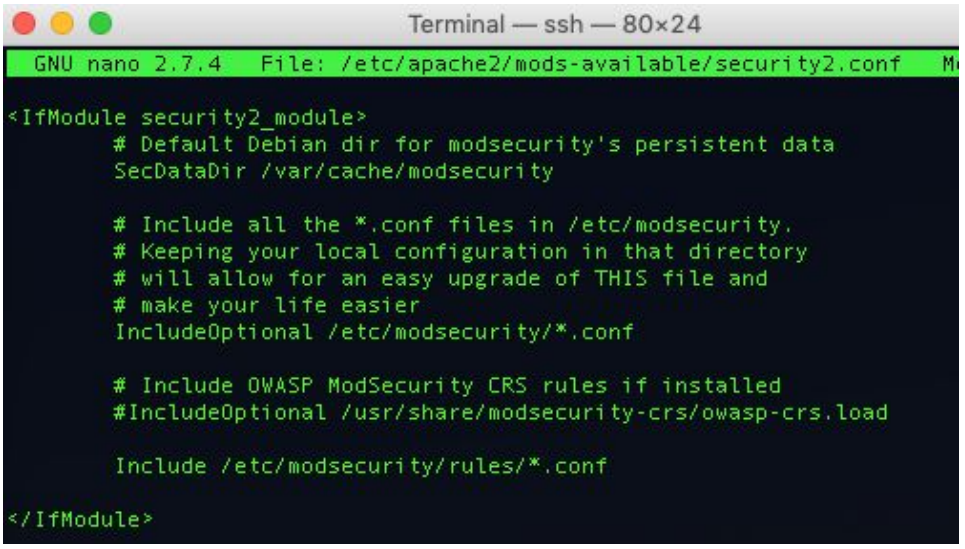


Image 5.5 - security2.conf

Then restart apache: ***systemctl restart apache2***.

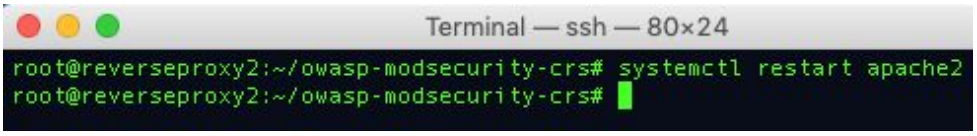


Image 5.6 - restart apache without error

Now modify the file /etc/apache2/sites-available/webserver.conf, so that it looks as below and then reload apache: ***systemctl reload apache2***.

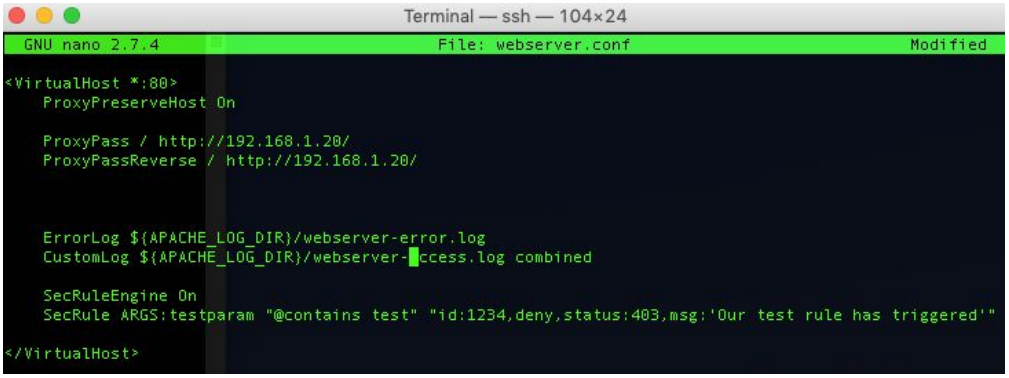


Image 5.7 - webserver.conf with ModSecurity

Added:

- Error and access log settings
- SecRuleEngine on, this enables ModSecurity
- SecRule ARGS, rewrites the log if it contains a certain string, which in the example is "test"

Before we do the security tests on ModSecurity, let's create a page, different from Forbidden, to create more impact to those who are attacking our site. In the webserver, directory / var / www / html, we will create an incident.html page. Mine was as below:

Image 5.8 - incidente.html

Now let's edit a modsecurity file. Before you need to rename the file below that is in /etc/modsecurity/rules:

mv RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf

The above command changes the length of .conf.example to .conf.

Now, edit the file and leave it as shown below:

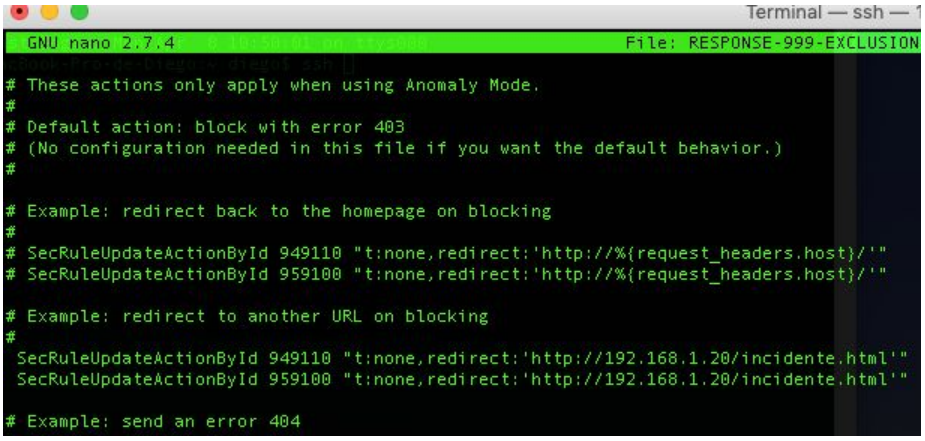


Image 5.9 - RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf

After saving, restart apache. See that I am pointing to the webserver, because it was there that we created the incident.html.

Now we are going to test a SQL Injection attack by inserting the string ***index.php?postid=6641%22%20or%20(1,2)=(select*from(select%20name_const(CHAR(111,108,111,108,111,115,104,101,114),1),name_const(CHAR(111,108,111,108,111,115,104,101,114),1))a)%20--%20%22x%22=%22x*** on site 192.168.1.30 through the client's machine (172.16.1.10), in what would be an internal attack. Leave the tailf /var/log/apache2/webserver-error.log command from the reverse proxy.

Image 5.10 - SQL Injection

Image 5.11 - WAF blocked the ataque

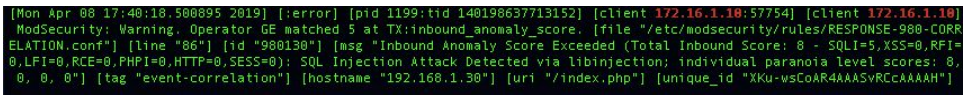


Image 5.12 - Logs modsecurity on reverse proxy

WAF detected the SQL Injection signature and returned the Forbidden (403) page. As we modified to return the incident.html page, it happened. In addition logs were generated containing the attack information and which IP it came from, in the case of 172.16.1.10.

We go to two more tests via terminal with the curl command. Open a terminal on your real machine and type the command: ***curl http://192.168.15.30/index.html?testparam=test***.

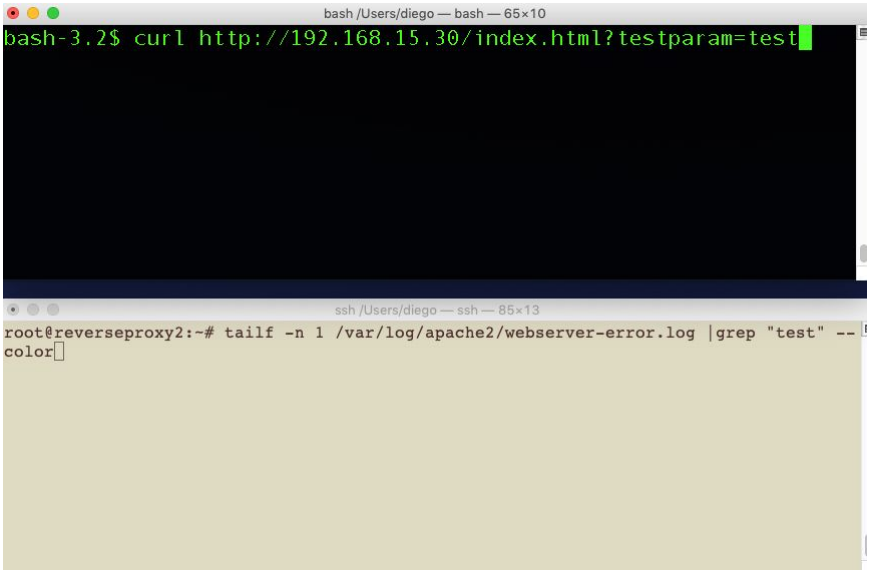


Image 5.13 - Shell with curl and reverse proxy log

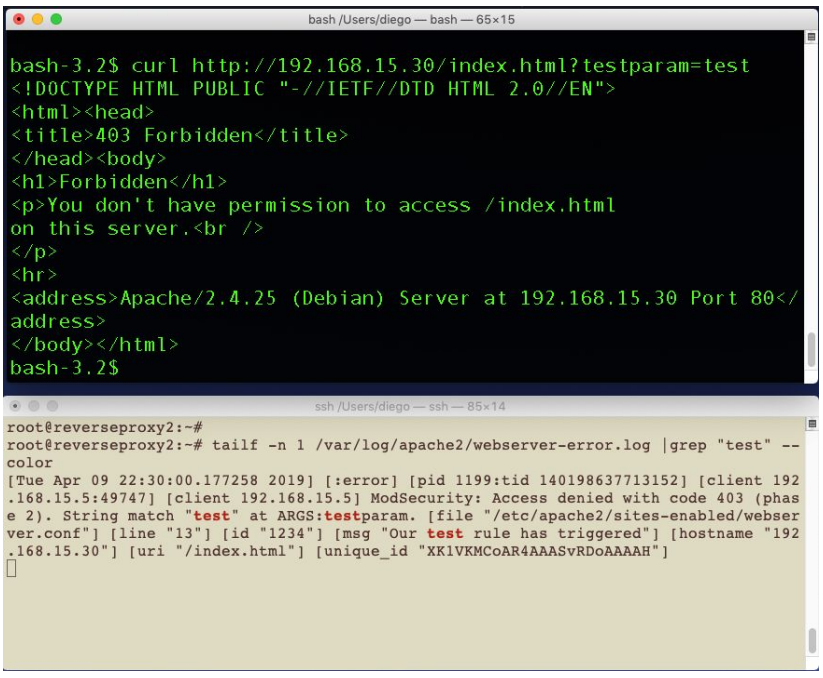


Image 5.14 - 403 Forbidden, mitigated attack

This attack activated SecRule ARGS, shown in figure 126. It is a good example of how you can easily adapt a ModSecurity rule. In this case, upon detecting the string "test", the rule was activated. We could improve this by adding a subscription that does not yet exist in ModSecurity, etc ...

Finally, let's go to another attack, now one trying to bash via url, trying to exploit a possible vulnerability: ***curl http://192.168.15.30/index.html?exec=/bin/bash***.

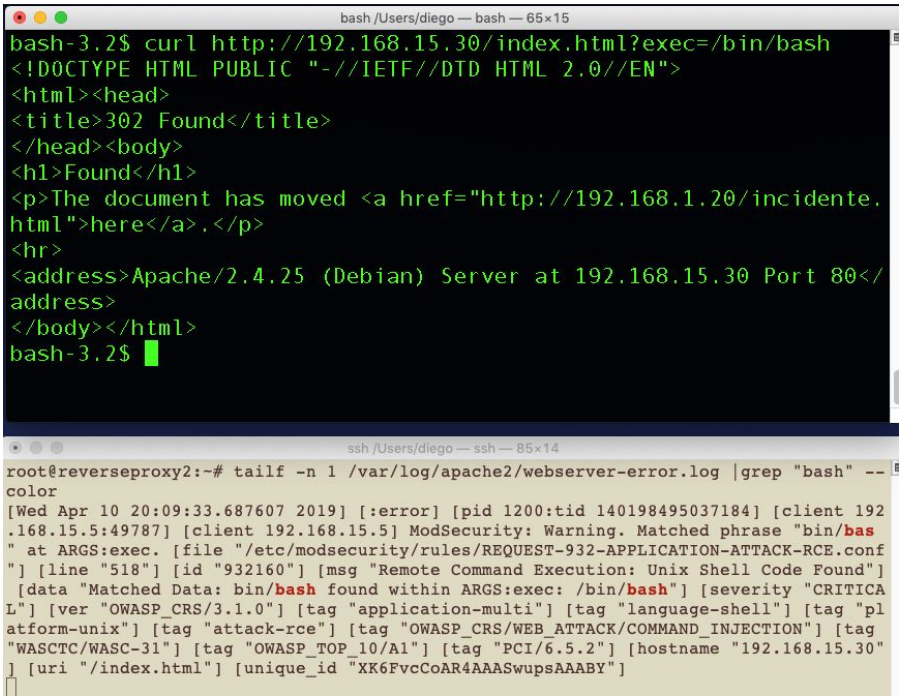


Image 5.15 - mitigated attack

5.1 Resume

Having a Web Application Firewall on your network is a must. The tests we did here were pretty basic. I recommend that you test your defenses using tools like Openvas, which we will see later. Make controlled attacks using the entire arsenal of the Kali Linux operating system and adjust your defenses. Schedule periodic and controlled tests, in an approval / testing environment, and you will be anticipating possible attacks that would compromise your network.

6 - SIEM

According to wikpedia: "Security Information and Event Management is a software solution that combines SIM (security information management) and SEM (security event manager).

In summary, SIEM (Security Event Management and Correlation) is software where you have the visibility of virtually everything that happens on your network in terms of security. In addition to visibility, it also offers tools for scanning vulnerabilities (openvas), network inventory (type of operating system, services, network, etc ...) and compliance, just to name a few.

With the SIEM concept, the OSSIM concept emerged, which is nothing more than the Open Source version of SIEM. And when it comes to OSSIM, I think of AlienVault (https://www.alienvault.com/products/ossim). Below are the security features listed on the website:

- Asset discovery
- Vulnerability assessment
- Intruder detection
- Monitoring network behavior
- Correlation of SIEM events

AlienVault will really revolutionize security on your network. The solution that I believe to be the most important is Intruder Detection, or for insiders, IDS. The junction of an IDS with a world map containing all the information about the attacks is the apex of the visibility of your network. We will do that in the next chapter.

AlienVault intrusion detection occurs through Suricata (https://suricata-ids.org), which is an excellent Open Source IDS / IPS. Despite being an IPS (Intrusion Prevention), in AlienVault it only acts as IDS, being positioned receiving all the traffic that the edge FW receives, through port mirroring on the switch. Due to the limitations of our laboratory with virtual machines, we will install a new virtual machine with AlienVault just to provide an overview of its functionalities.

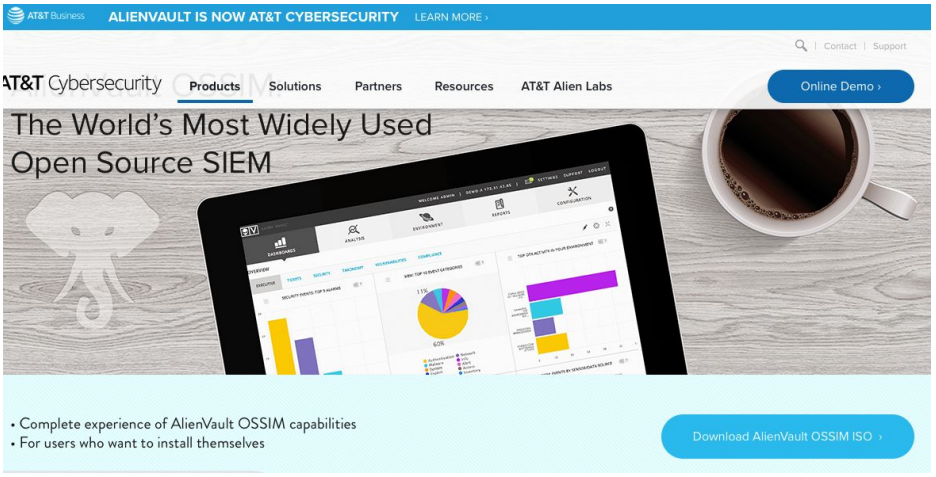


Image 6.1 - Download the ISO file

Normally, 3 network interfaces are required, one for Management, another for logs and vulnerability scans; the latter for port mirroring to work in promiscuous mode. We will create our machine with all three, one on my wifi router's network (192.168.15.40), for management, the other two on the DMZ for logs and vulnerability scans (192.168.1.40); and promiscuous mode (192.168.1.40), it seems strange, but AlienVault assigns the same IP to both interfaces.

Image 6.2 - Choice of type

Alienvault works in two ways. In Sensor mode, it works as a sensor that passes the data on to the AlienVault Server (which we will install). This makes it possible to spread multiple sensors on your network, increasing the visibility range of your network.

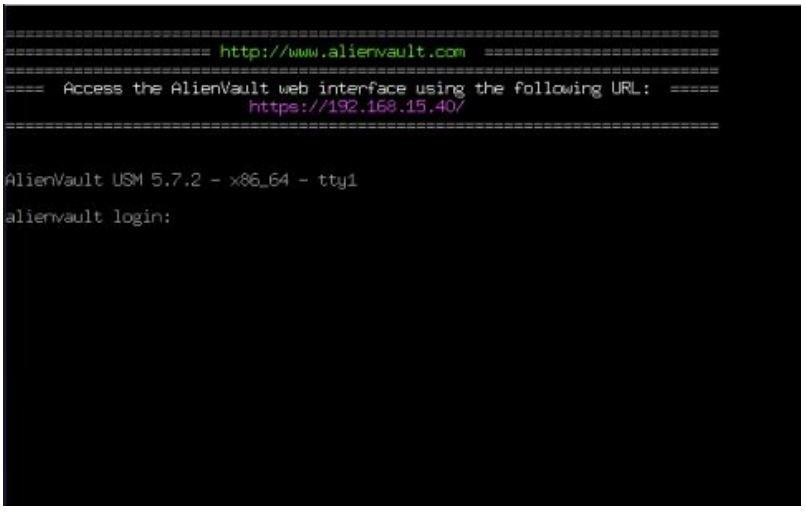


Image 6.3 - AlienVault installed

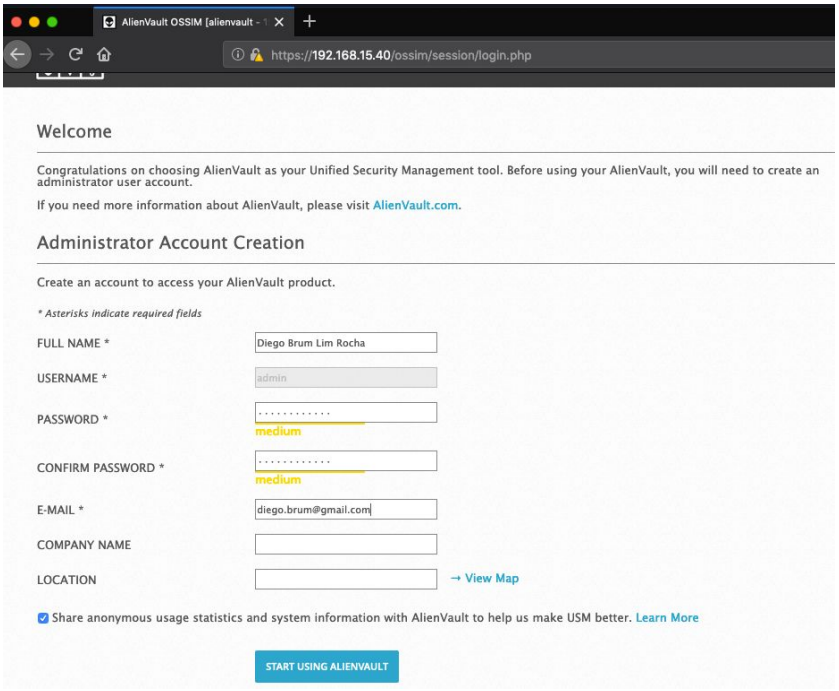


Image 6.4 - Welcome

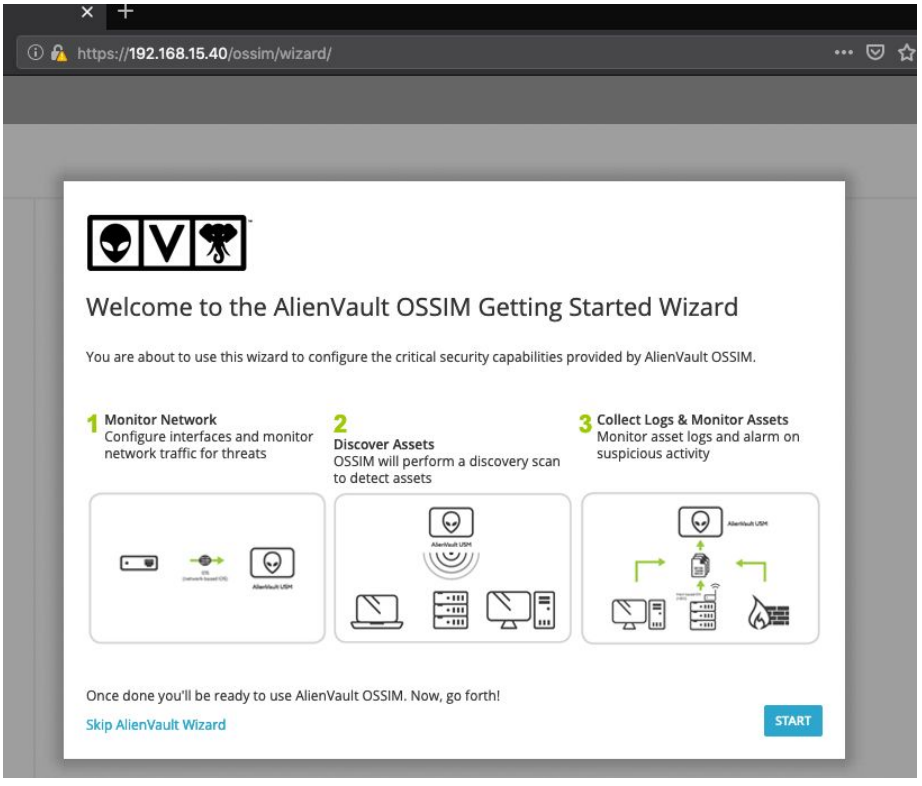


Image 6.5 - Start of configuration

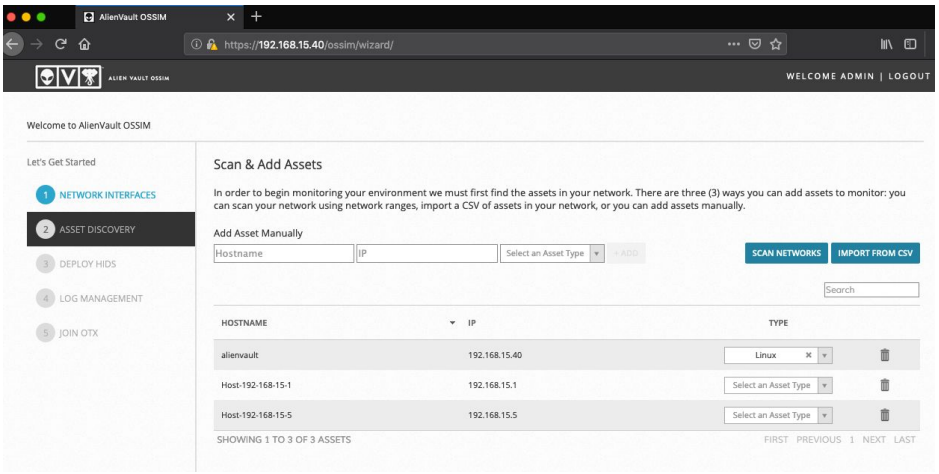


Image 6.6 - Discovery of hosts

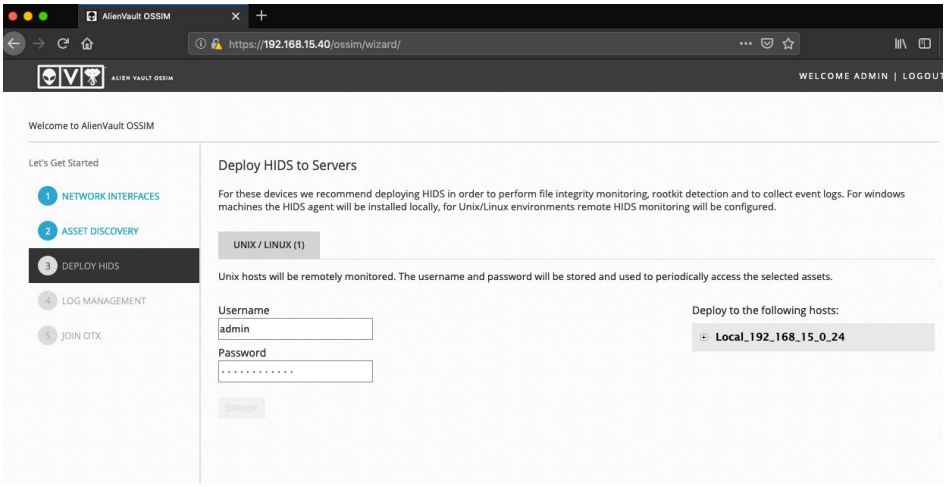


Image 6.7 - HIDS configuration

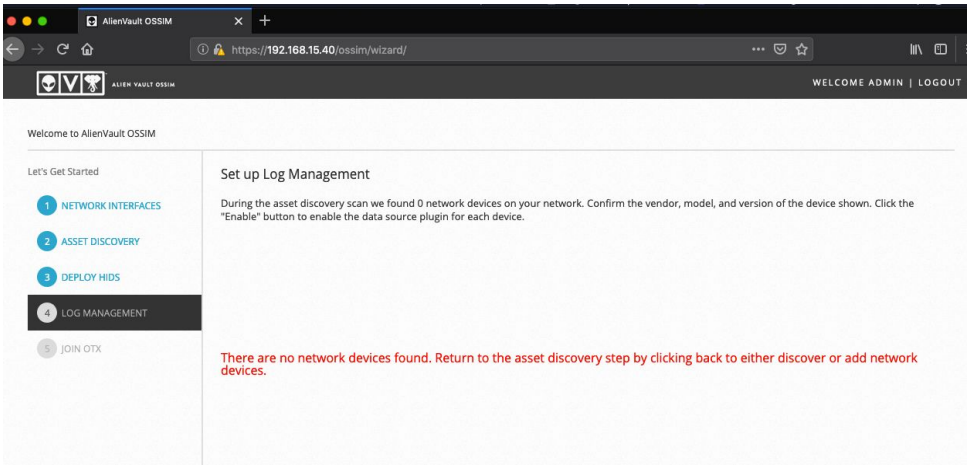


Image 6.8 - Log management (syslog)

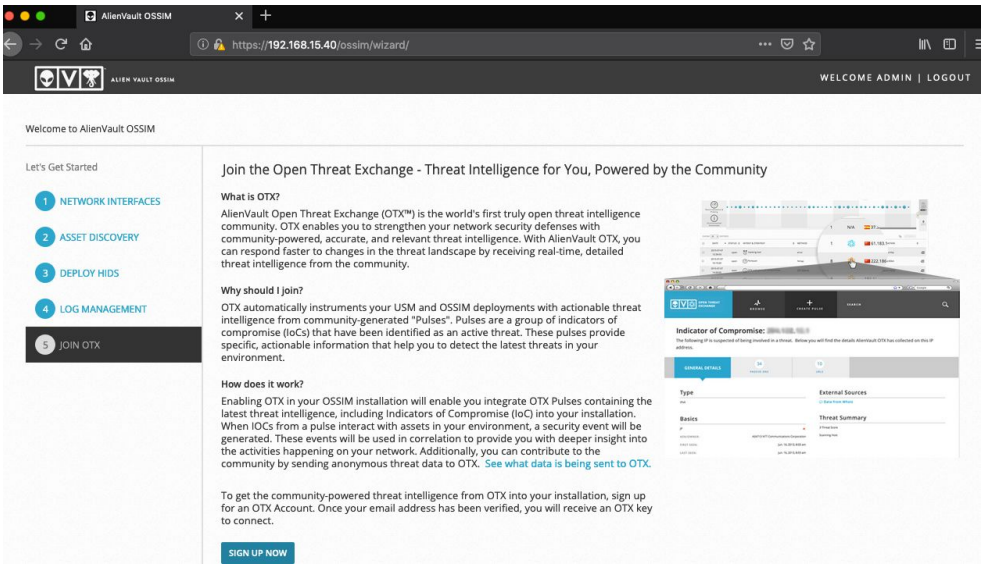


Image 6.9 - Open Threat Exchange

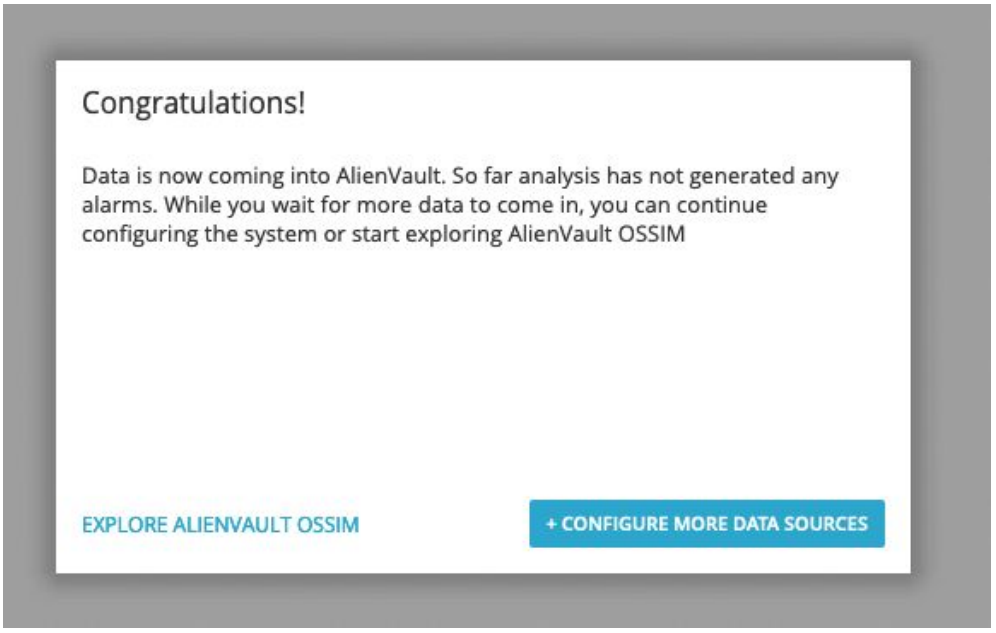


Image 6.10 - Click Explore AlienVault OSSIM

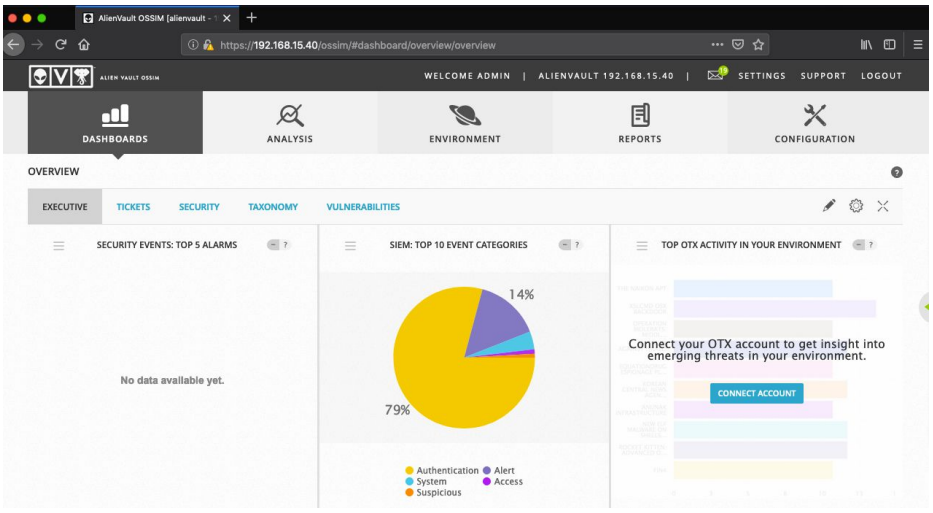


Image 6.11 - AlienVault now operational

Don't worry, all of the above settings can be done later. Figure 140 shows the host discovery feature, which is very useful for a complete inventory of your network. 141 deals with HIDS, which we talked about in chapter 3, and can also be configured in AlienVault. 142 is the syslog configuration. 143 is a community, which I recommend that you join, where you receive information in your AlienVault about new threats. See that AlienVault, like a good SIEM, tries to centralize all its security information in one place. This is to avoid having to look at countless different tools.

Note that there are five large menus in the website header (Dashboards, Analysis, Environment, Reports and Configuration). Let's go to the basic characteristics of each one:

- Dashboards: here are the charts that are subdivided into five specialties (Executive, Tickets, Security, Taxonomy and Vulnerabilities).

    - Executive: brings the most summarized graphics, making the most critical information stand out.
    - Tickets: tickets are events, which can receive a rating and be treated, as in incident management.

    - Security: it has graphs with the top events, those that should receive more attention from the security analyst.
    - Taxonomy: brings information about FW rules, malware, exploits and system events.
    - Vulnerabilities: the data here are the result of a vulnerability analysis carried out in the Environment -> Vulnerabilities menu.

- Analysis: here is the information that, after normalized, forms the Dashboards' graphics. It is possible to search and know with great granularity the events caused by all types of threats. The function I like a lot is Security Events (SIEM) -> Real Time, as it is possible to see in real time the threats that are traveling on your network, especially the information coming from the IDS, as they are what I replicate on the Map of Attack, which we'll talk about in the next chapter.

- Environment: the focus in this menu is on the devices that make up your network. It is possible to do vulnerability tests, check availability, information from distributed HIDS, Netflow and Traffic Capture.

- Reports: well, there is no mystery here, it's the reports.

- Configuration: are the AlienVault settings. In the Threat Intelligence menu, advanced settings and the compliance part are made. In Deployment, sensors and plugins are configured (to integrate the security tools you have on your network, such as the antivirus server).

At first glance, it seems complex to configure AlienVault, but when you install it, configure the 3 interfaces and do a port mirroring in the Network Monitoring interface, you will already have valuable

information made available by AlienVault. As you use OSSIM, it becomes easier to move to the most challenging settings.

AlienVault is a huge tool and full of peculiarities that, like many subjects described in this book, would give another book. As I said during the book, the idea is to introduce you to the concepts and tools to demystify them and help them get started.

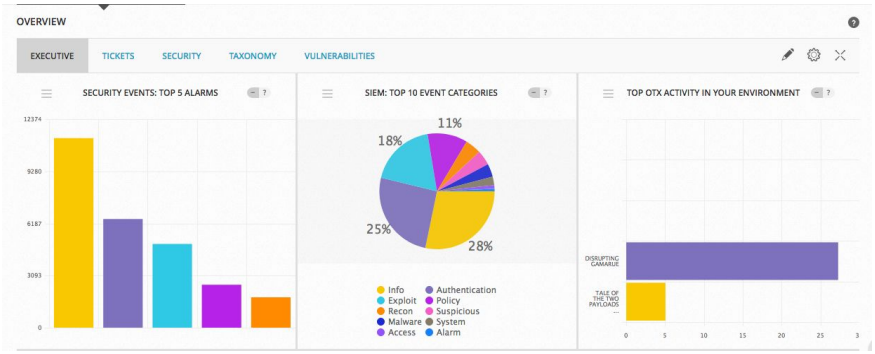


Image 6.12 - AlienVault images in production 1

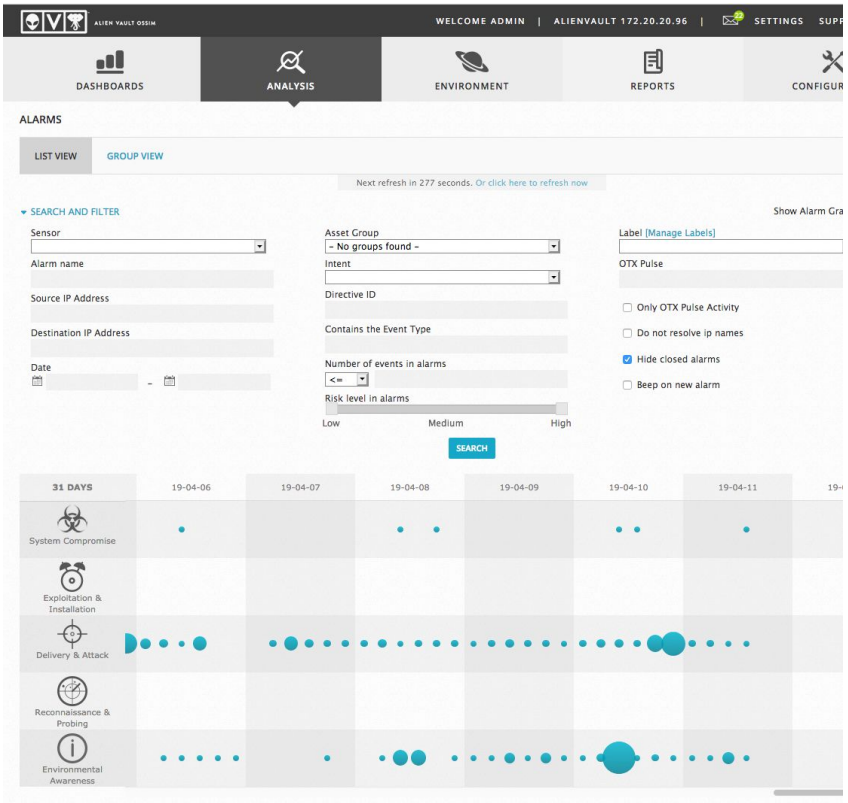


Image 6.13 - AlienVault images in production 2

6.1 Resume

Having a SIEM is mandatory in any network that wants to treat information security with due importance. It centralizes, through plugins, information from other security tools on your network (Firewall, HIDS, Antivirus, Network Assets, etc.), making administration much easier. In addition, it provides a vulnerability scanning tool and an excellent IDS. AlienVAult has even more features, many I'm still learning, but we've seen all the potential it offers for security on your network.

7 - Geoip Map Attack

When I first saw an attack map, with all that information in real time, I found it very interesting. That cyber war going on, colored lines representing the attacks like intercontinental missiles ... I was very impressed. My goal, since I saw this technology, was to implement something similar in my work network. So, I started to research Open Source projects on the internet, of course, that I could incorporate into the list of security tools that I had already implemented on the network. I was pleasantly surprised to find an excellent project by Matthew Clark May. He simply put together an attack map that, in addition to being simple to implement on the network, was stable and, most importantly, clearly presented the attacks that were taking place on the network. . As soon as I downloaded it, I made some adjustments and sent an email to Matthew, who really liked my improvements. From then on, I became part of the project (https://github.com/MatthewClarkMay/geoip-attack-map).

The improvements I made were:

- Placing information on the Type of Attack, Exploit and IP; all at the top of the map.
- Target information, the one at the bottom left of the map.
- Button to block the IP that is attacking and menu to check the reputation of the IP. This was done by Prof. MSc. João Victor de Araujo Oliveira (http://lattes.cnpq.br/6697354215628897).
- Animation that happens when pressing the Block IP button
- Improved IDS log management across the map, using fewer web browser features.

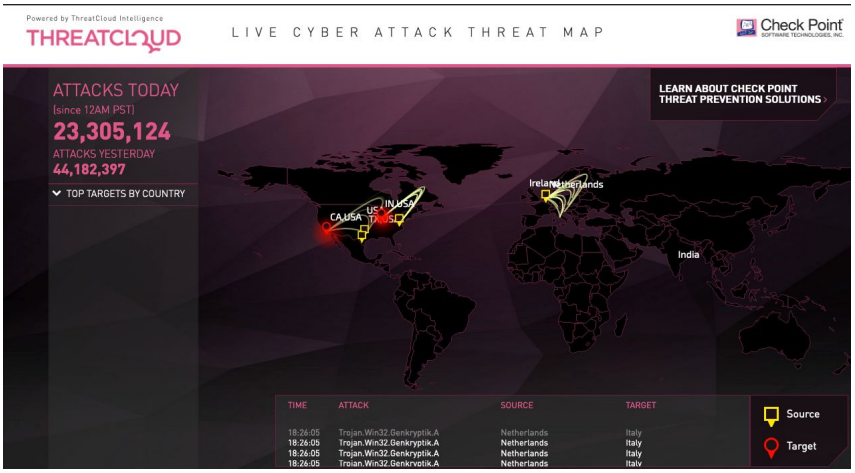


Image 7.1 - Check Point cyber attack map

Image 7.2 - Norse cyber attack map

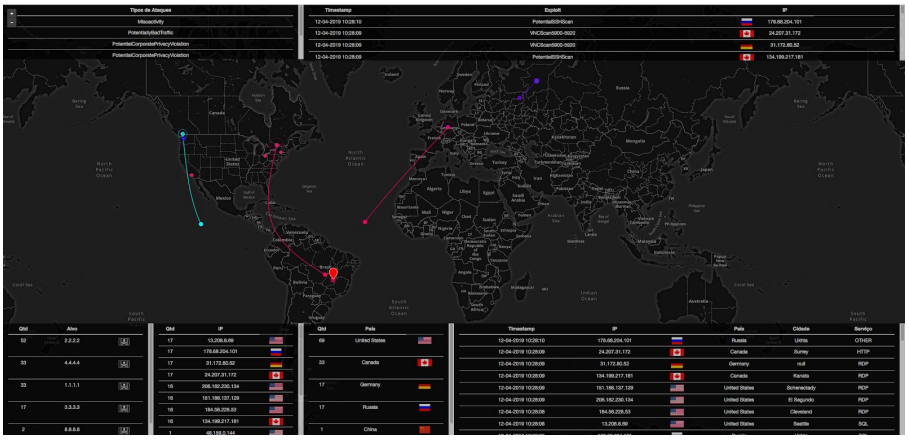


Image 7.3 - Matthew's cyber attack map project

See that the map of Matthew's project does not leave much to be desired compared to the maps shown by large companies. The fund, which is a world map, is from a company called MapBox (https://www.mapbox.com). Then you need to access the site and create an account to access the map. The company makes the map available for a limited number of accesses, but it is usually sufficient to use it for a long time. Depending on the need, it may be more advantageous to purchase a license. Then create an account and a token will be made available, which we will use in the project's index.html settings.

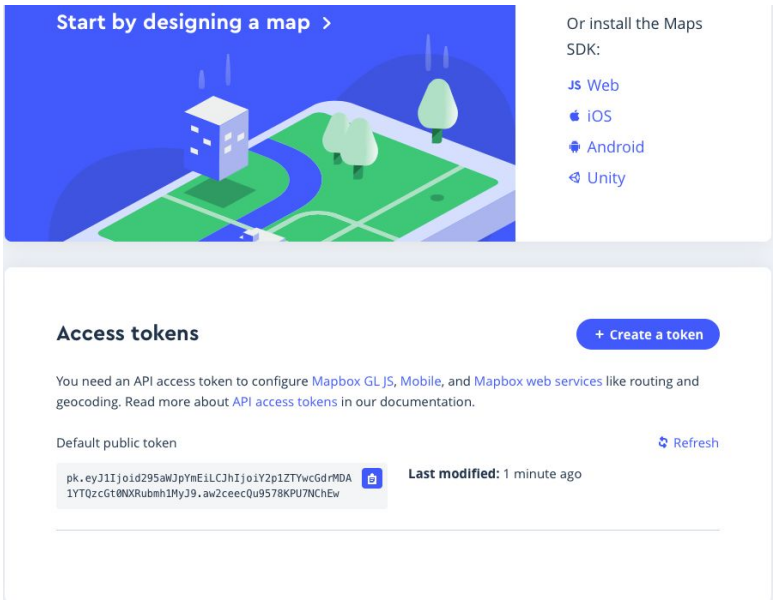


Image 7.4 - Token MapBox

We will install our map on the virtual machine called syslog, which we set up in the HIDS chapter.

Image 7.5 - machines used in this chapter

The idea is to install the map on the syslog and access the attack map through the external network, through a NAT on the FW. We will follow the step by step of Matthew's project (https://github.com/MatthewClarkMay/geoip-attack-map), but when downloading the map, I will leave two options:

- The available in the project (git clone https://github.com/matthewclarkmay/geoip-attack-map.git).

- The one provided by me (https://drive.google.com/drive/folders/1Ic6iqZ19oFIhwAw8-qEKj6DDdjUdhnXS?usp=sharing).

It just so happened that I made some changes and a while ago I stopped uploading the modifications to the project. Actually the main changes I already made available in the project, but if you want my version that is all in Portuguese, the choice is yours.

First, we have to basically understand how the map works. It was written in Python and has two systems, one for handling logs (DataServer) and another for handling data structures and making information available on the website, which is the AttackMapServer module. So there are two systems: DataServer and

AttackMapServer. They are distinct, being processes that can be turned on or off separately. The application server is Redis Server.

In addition, the project comes with a shell script that simulates the operation of the map with fictitious attacks. We will use it to verify that our map is working. The map works by reading a log file that must receive the data from an IDS, which in a production network would be our AlienVault. However, given the difficulties of running AlienVault in the lab environment of this book, I will only explain a few ways to normalize the AlienVault logs and make them available to the map.

We will then begin the installation process available on the project website:

baixar o geoip-attack-map_v4.tar.gz no link do livro ou no site do projeto. Abaixo eu baixei o exemplo do livro.

tar xvzf geoip-attack-map_v4.tar.gz

apt install python3-pip redis-server

cd geoip-attack-map_v4

pip3 install -U -r requirements.txt

editar o /etc/redis/redis.conf. No campo bind 127.0.0.1, mudar para bind 0.0.0.0

redis-server

```
root@syslog:~/geoip-attack-map_v4# service redis-server status
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor pre
   Active: active (running) since Thu 2019-04-11 19:33:25 -03; 51min ago
     Docs: http://redis.io/documentation,
           man:redis-server(1)
 Main PID: 6226 (redis-server)
   CGroup: /system.slice/redis-server.service
           └─6226 /usr/bin/redis-server 127.0.0.1:6379

Apr 11 19:33:25 syslog systemd[1]: Starting Advanced key-value store...
Apr 11 19:33:25 syslog run-parts[6221]: run-parts: executing /etc/redis/redis-se
Apr 11 19:33:25 syslog run-parts[6227]: run-parts: executing /etc/redis/redis-se
Apr 11 19:33:25 syslog systemd[1]: Started Advanced key-value store.
```

Image 7.6 - redis-server running

Create the log file for the DataServer.py: ***touch /var/log/suricata_geoip.log***

cd DataServer

python3 DataServer.py

```
root@syslog:~/geoip-attack-map_v4# touch /var/log/suricata_geoip.log
root@syslog:~/geoip-attack-map_v4# cd DataServer
root@syslog:~/geoip-attack-map_v4/DataServer# python3 DataServer.py
```

Image 7.7 - DataServer.py running

Now that the DataServer is running perfectly, open another terminal, since the terminal is stuck running the DataServer, and execute the commands below:

Go to the directory geoip-attack-map_v4, then to DataServer.

running the Shell Script ***./syslog-gen.sh &***

```
Terminal — ssh — 114×26
root@syslog:~/geoip-attack-map_v4/DataServer#
root@syslog:~/geoip-attack-map_v4/DataServer#
root@syslog:~/geoip-attack-map_v4/DataServer# ./syslog-gen.sh &
[1] 6568
root@syslog:~/geoip-attack-map_v4/DataServer#
```

Image 7.8 - ./syslog-gen.sh &

The Shell Script above writes fictitious attack logs to the log we created: /var/log/suricata_geoip.log. And it's running in the background, because this the &.

cd ../AttackMapServer/

Inside AttackMapServer, edit the AttackMapServer.py. In the field self.client = tornadoredis.Client('XXX.XXX.XXX.XXX'), put the syslog IP 192.168.1.10, as shown below:

```
try:
    # This is the IP address of the DataServer
    self.client = tornadoredis.Client('192.168.1.10')
    self.client.connect()
```

Image 7.9 - IP of the machine where the map is

Now, edit the index.html. Put, as below, the NAT ip that we will use to access the map: 192.168.15.30.

```
// To access by a browser in another computer, use the external IP of machine running AttackMapServer
// from the same computer(only), you can use the internal IP.
// Example:
// - AttackMapServer machine:
//   - Internal IP: 127.0.0.1
//   - External IP: 192.168.11.106
var webSock = new WebSocket("ws:/192.168.15.30:8888/websocket");
```

Image 7.10 - websocket with IP from NAT

The token field must have what you obtained when registering with MapBox. In addition to the dark background map, MapBox offers two more interesting options: satellite and streets.

```
L.mapbox.accessToken = "pk.eyJ1Ijoid295aWJpYmEiLCJhIjoiY2p1ZTYwcGdrMDA1YTQzcGt0NXRubmh1MyJ9.aw2ceecQu9578KPU7NChEw"


//mapbox.streets,mapbox.satellite,mapbox.dark

        var map = L.mapbox.map("map", "mapbox.dark", {
                center: [9.8302209,12.4282347], // lat, long
                zoom: 3
                });
```

Image 7.11 - configuration of the index.html

Enter the latitude and longitude of your location in the field below.

```
    var marker = L.marker([-15.753912, -47.8809987], {
  icon: L.mapbox.marker.icon({
    'marker-color': '#FF0000'
  })
})
.bindPopup('<button class="trigger">Brasília</button>')
.addTo(map);
```

Image 7.12 - lat / long configuration

python3 AttackMapServer.py

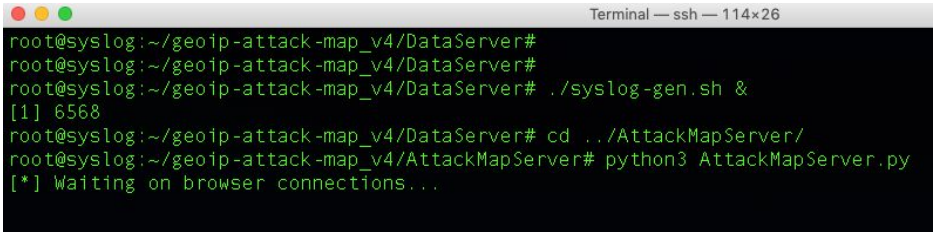


Image 7.13 - AttackMapServer running

NOTE: AttackMapServer is listening on port 8888, which is not released on the FW, so let's release it.

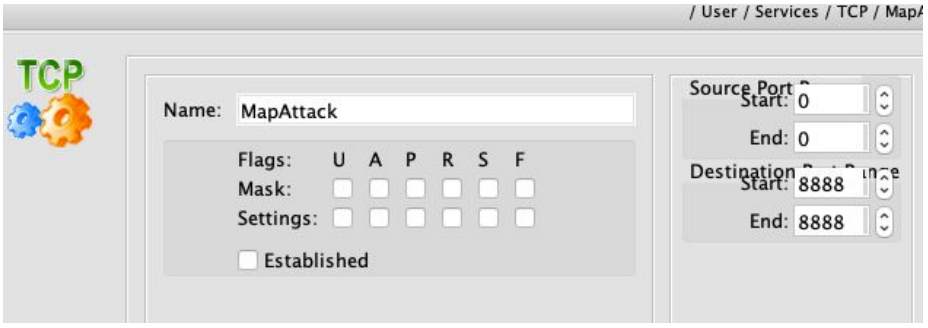


Image 7.14 - Creation of object 8888

Image 7.15 - Policy rule created

NAT is also missing:

Image 7.16 - NAT rule map access

It is now possible to access the attack map through your real machine. http://192.168.15.30:8888.

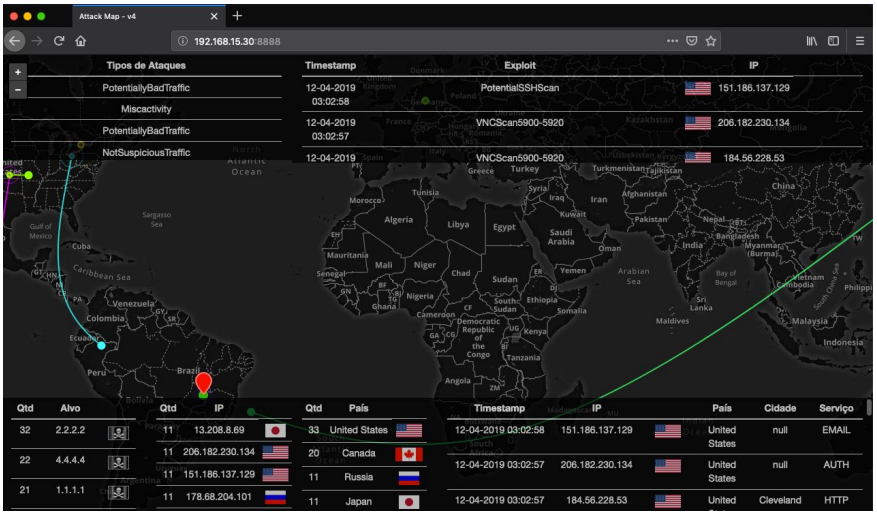


Image 7.17 - running attack map

NOTE: If it didn't work, see the possibilities below:

- The map (background) does not appear. See if you have placed your account token on the MapBox.

- Nothing appears in the browser. Make sure that the FW has the release in the Policy and that there is NAT.

- there is no attack data. Check that the syslog-gen.sh script is running. One way to check is to check the log suricata_geoip.log is receiving information.

- Make sure that the four required applications are running: redis-server, AttackMapServer.py, DataServer.py and syslog-gen.sh.

- Review all settings.

Let's test some features of the map. In the IP column, which is next to the Target column, click on any IP and a menu will appear.

Image 7.18 - Map Menu

The IP block function is commented on in the code, but the idea is to ssh the FW and using the ipset, which we saw in the FW chapter, to block it.

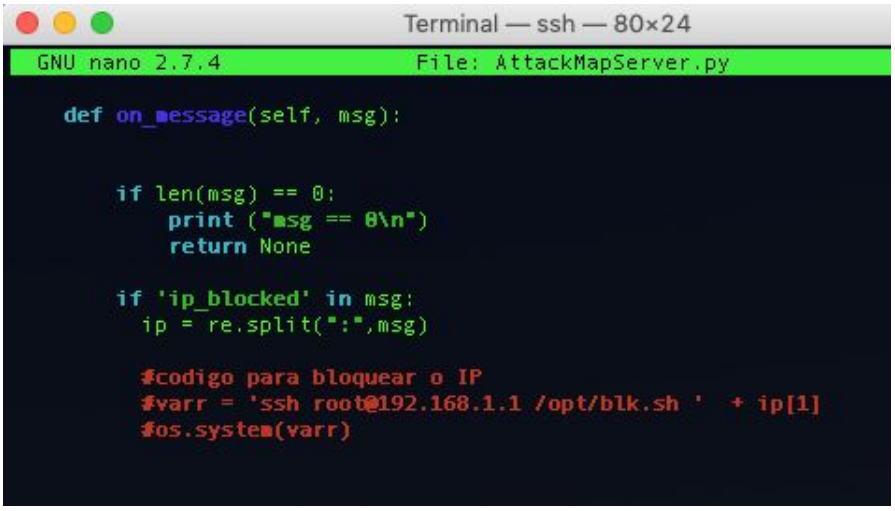


Image 7.19 - lock code

The code uses a Python function to make Operating System commands and makes a ssh that triggers a Shell Script that is in 192.168.1.1, /opt, called blk.sh; the IP that was clicked on the map is passed as a parameter to blk.sh. Shell Script receives the IP and makes a command **ipset add blacklist-ip 104.4.162.185**. We already learned how to use

ipset in the FW chapter and there is an appendix in this book that will help you develop your scripts.

When you click on the Senderbase or Anti-abuse link, a window will open and the reputation and other information about the IP will be checked.

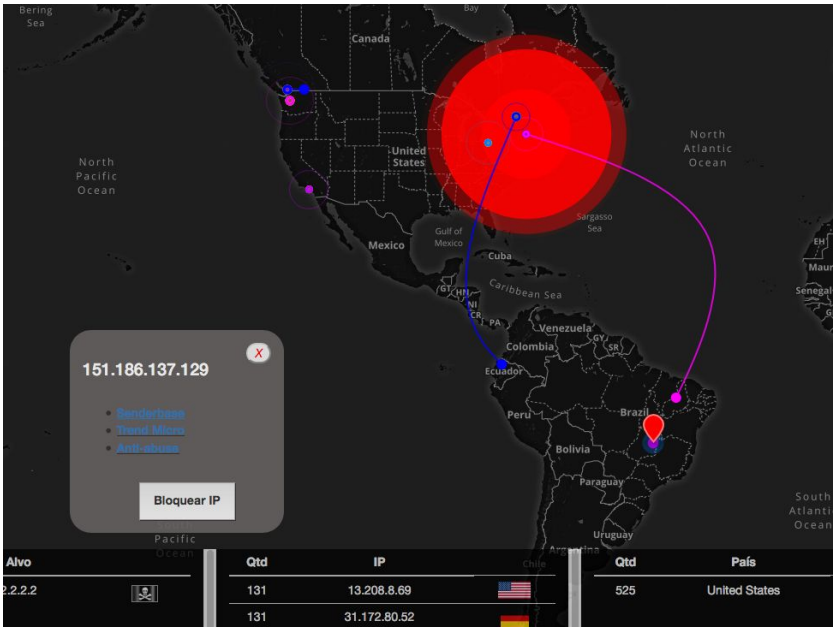


Image 7.20 - Animation that happens when pressing the button (Block IP).

7.1 Log normalization

We already have our attack map working, but only with fictitious data. To receive the data from an IDS, it takes a few steps to understand how the attack map expects to receive the data. As it currently stands, the map expects to receive in its log file (/var/log/suricata_geoip.log) six pieces of information in the following sequence and without spaces:

- Source IP
- Destination IP
- Source port
- Destination port
- Attack type
- Exploit. Although it is called an exploit, it is actually a field that details the attack more, it is not necessarily an exploit.

src_ip,dst_ip,src_port,dst_port,attack,exploit

It is exactly as described above. The attack map expects to receive the data. This can be improved in the code, which is precisely the interesting part of using an Open Source tool.

I use a Shell Script in production. It reads the AlienVault logs, more specifically the Suricata logs, and format as the map expects to receive. To be more precise, on the syslog server I do a ssh with the command tail -f in the log of the meerkat that is in the Alien Vault. The tail output is to a syslog log file. It is in this file, which is a copy of the Alien Vault log, that the shell script acts by translating it into the map format.

Another solution would be to configure AlienVault to send the IDS logs via syslog and on the syslog server to normalize.

I believe that the best solution would be to improve the DataServer.py code to receive the data in the IDS format that we are using. I intend to work on it and improve the project.

7.2 Resume

The attack map is a tool that makes all the difference in the visibility of your network. It is possible to take action as soon as an attack is seen in progress. If not present, the other open source tools that we have learned to use will contain the attacks. The map, despite having few active users contributing to the project, is quite satisfactory for use in production. Be one of the members and contribute to the project, which in my view has excellent potential.

Final considerations

Throughout the book we learned about some Open Source tools that can assist in the cyber defense of your network, but these tools do not represent everything available. There are still excellent tools like ELK (Elasticsearch, Logstash and Kibana), https://www.elastic.co/pt/elk-stack, to store the logs for the entire network and generate valuable information. There is also SELKS, a solution from Stamus Networks, https://www.stamus-networks.com/open-source/. It has excellent potential as an IPS Suricata and the ELK battery, all together. There is also the Pfsense Firewall, https://www.pfsense.org/download/, which in addition to FW, also offers a huge amount of software that works in conjunction with FW. Of course, there must still be many other excellent Open Source solutions to be studied.

I would like to thank my family members who motivated me to write this book. To thank the coworkers who helped me to put into practice the solutions described here. To Edimaria Cerqueira Rodrigues Lamounier and Erlan Pereira Frade Tostes for helping me with this project.

I really hope that this knowledge helps you in some way.

Files used in the book:
https://drive.google.com/drive/folders/1lc6iqZ19oFIhwAw8-qEKj6DDdjUdhnXS?usp=sharing

Facebook: https://www.facebook.com/LivroSegurancaOpenSource/
Instgram: https://www.instagram.com/segurancaopensource/
Blog: https://segurancaopensource.blogspot.com
Site: https://segurancaopensource.com
Linkedin: https://www.linkedin.com/in/diego-brum-lima-rocha-7ba78a22

Appendix 1

Basic Operating System Linux

1 - Installing Debian

First step is to download the ISO on the website https://www.debian.org.
I will use Debian version 8 as an example.

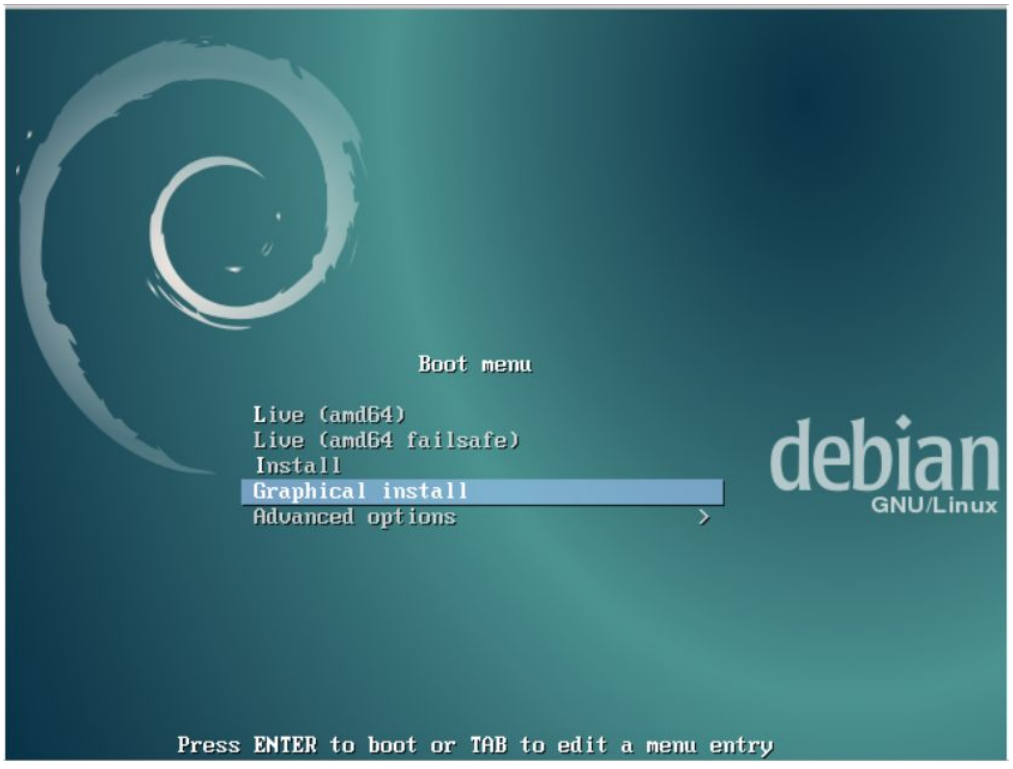


Image 1 - Graphical install

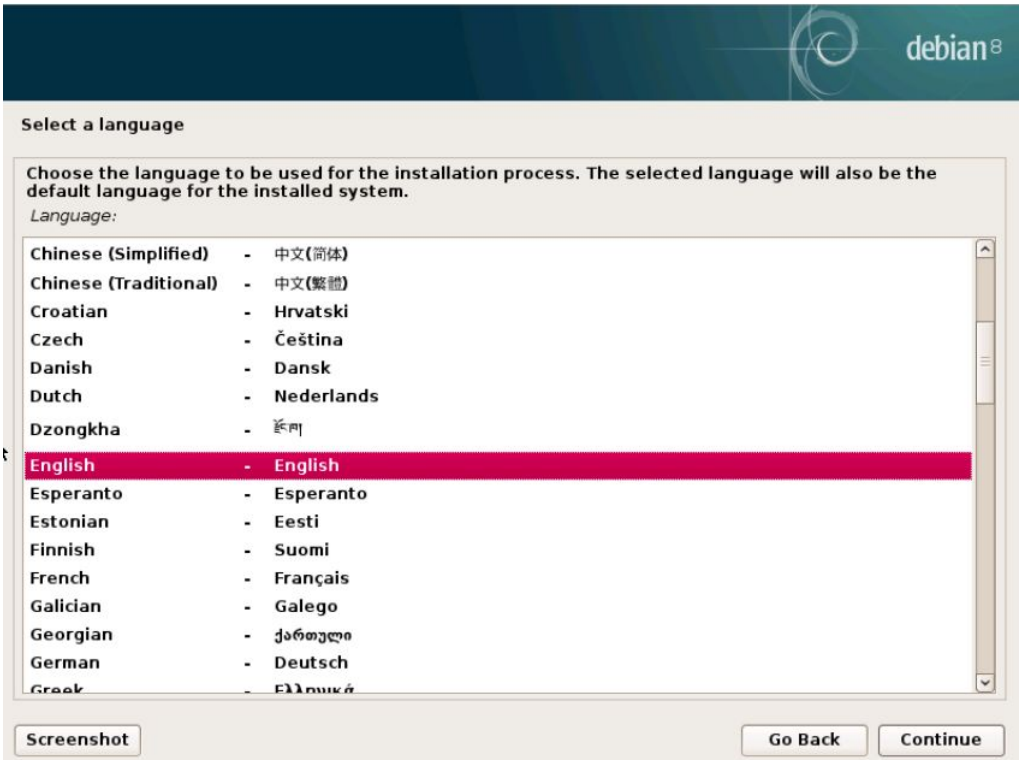


Image 2 - Choose the installation language

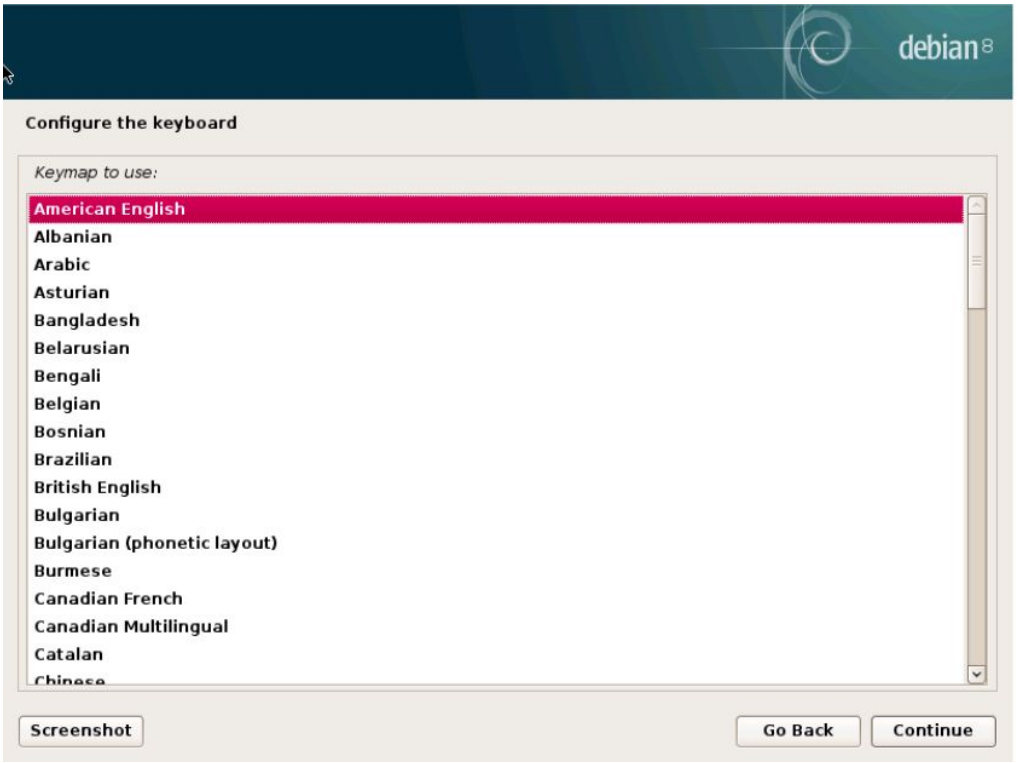


Image 3 - Keyboard language

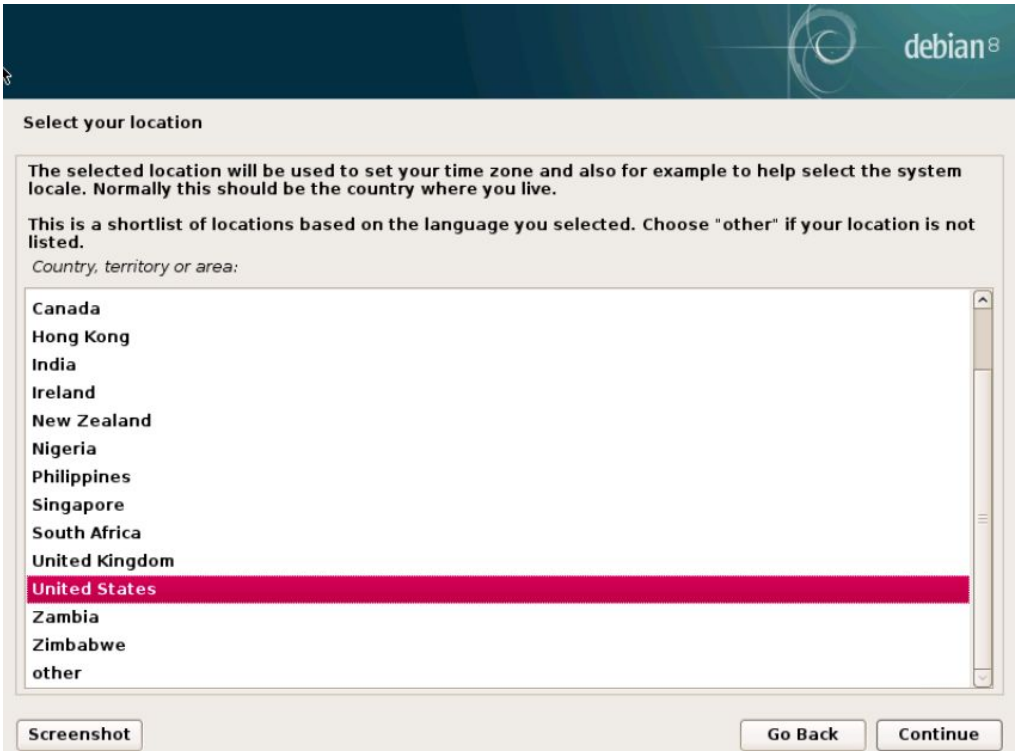


Image 4 - Your location

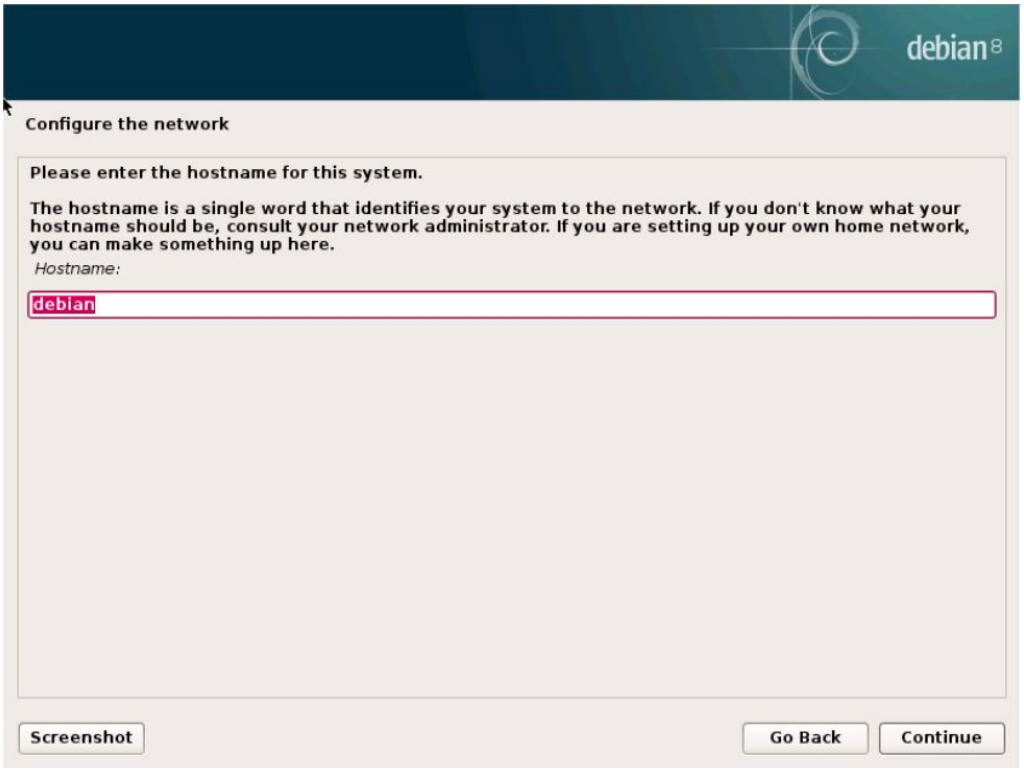


Image 5 - hostname

Image 6 - network domain

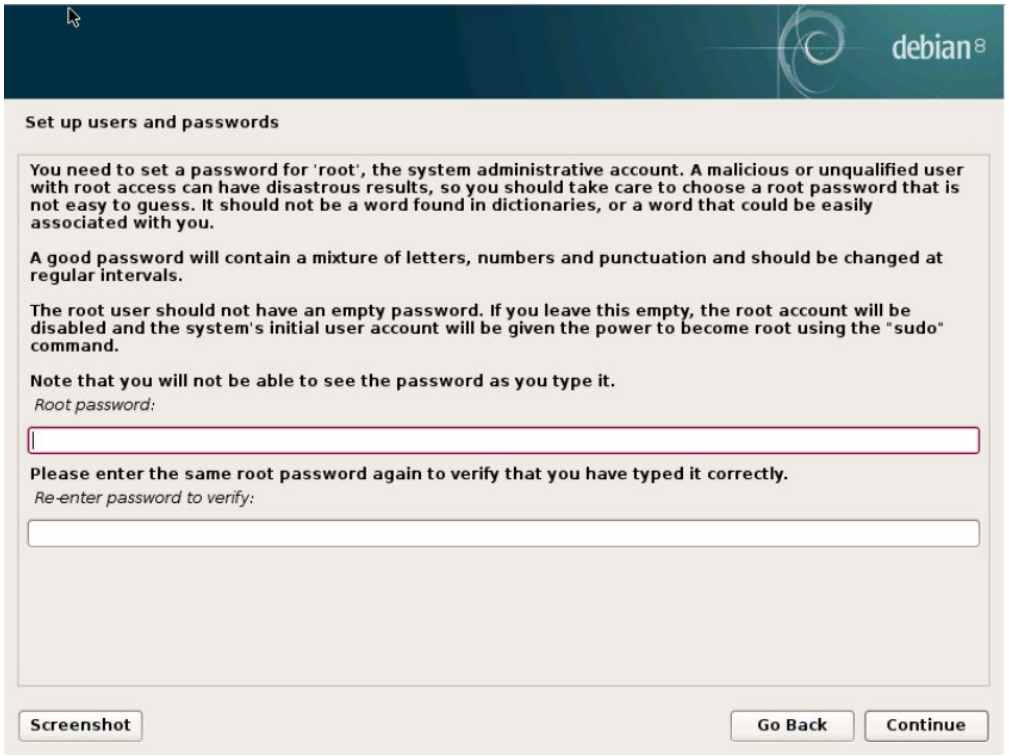


Image 7 - Root password (get from the password vault)

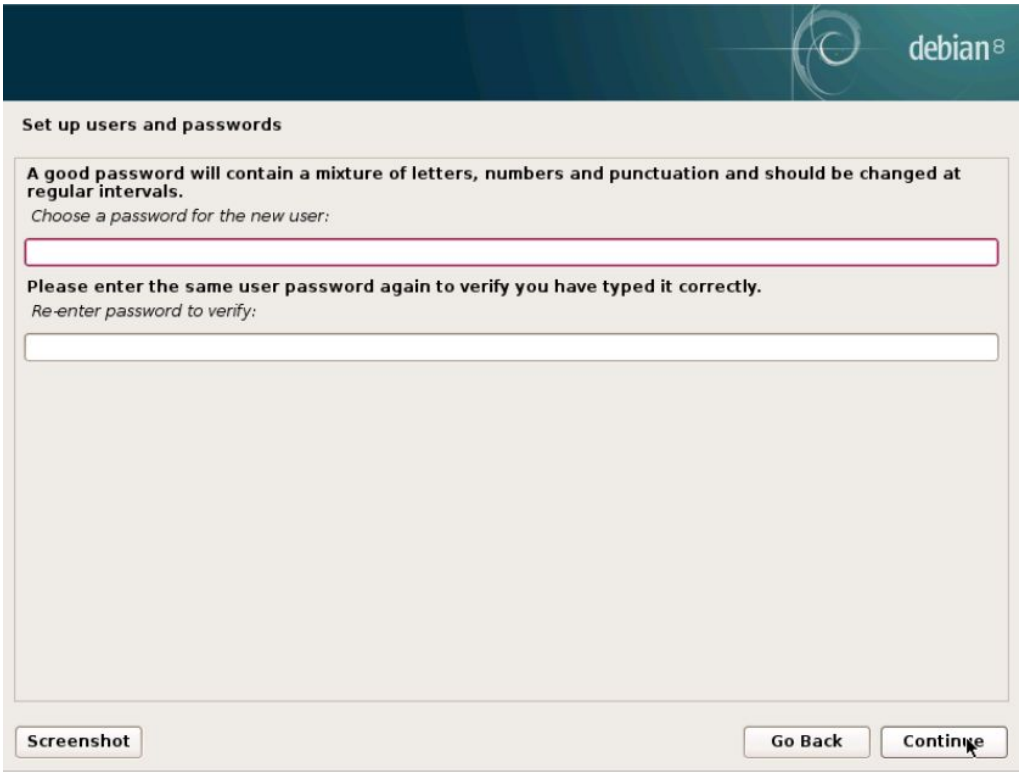


Image 8 - User password (non-root)

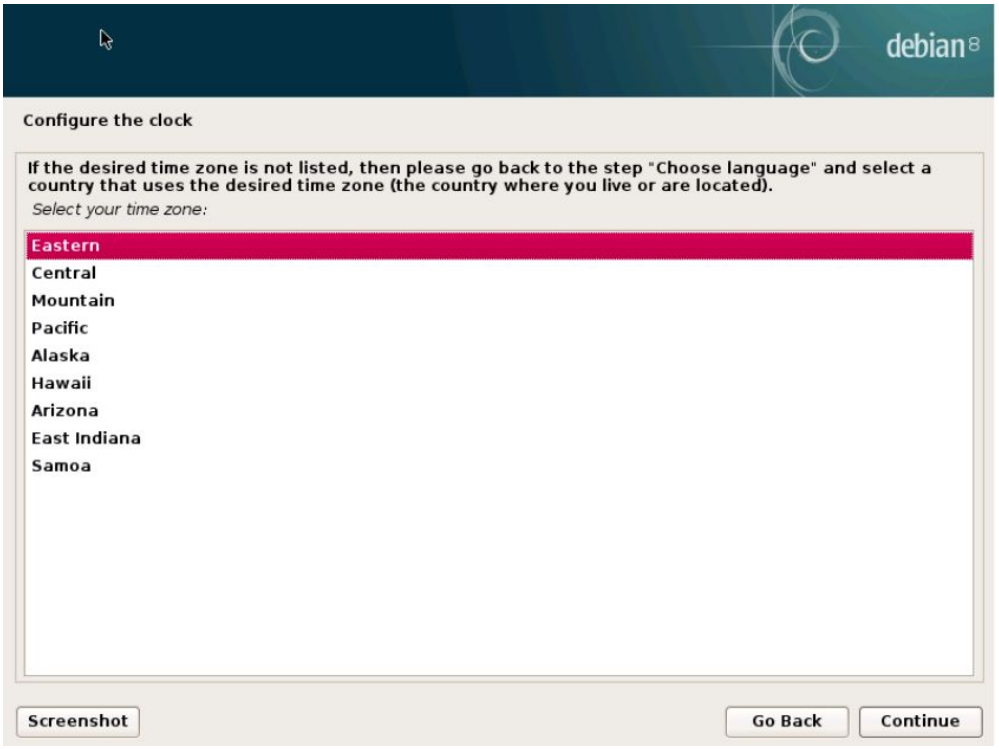


Image 9 - timezone

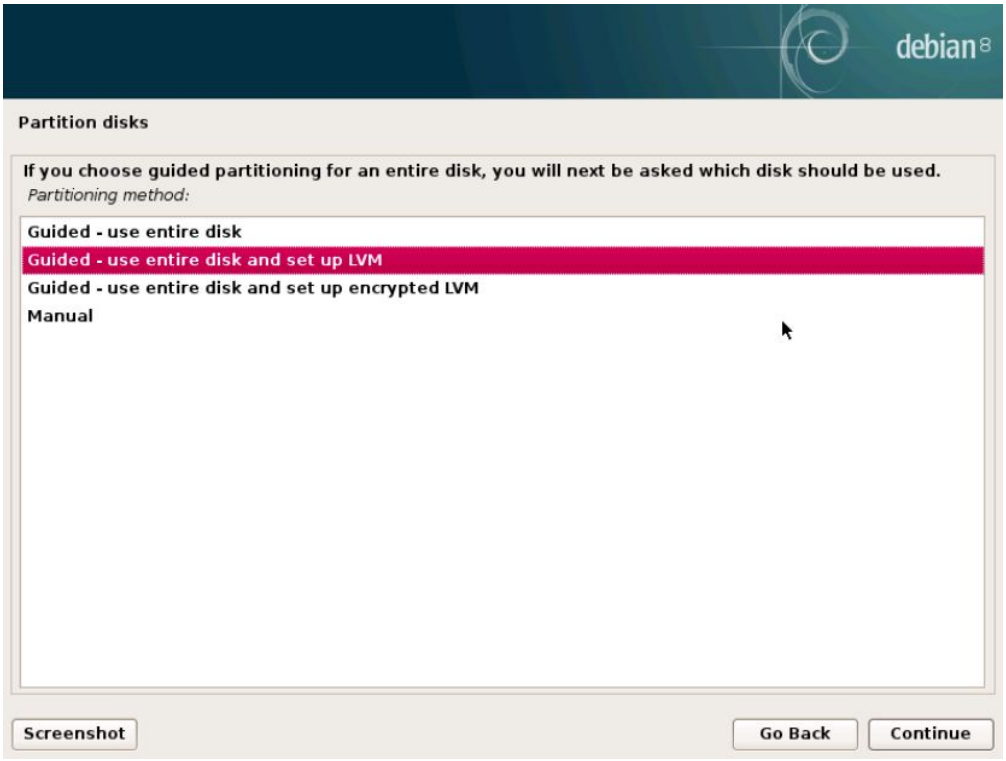


Image 10 - Always use LVM, so you can adjust the size of the hot partitions.

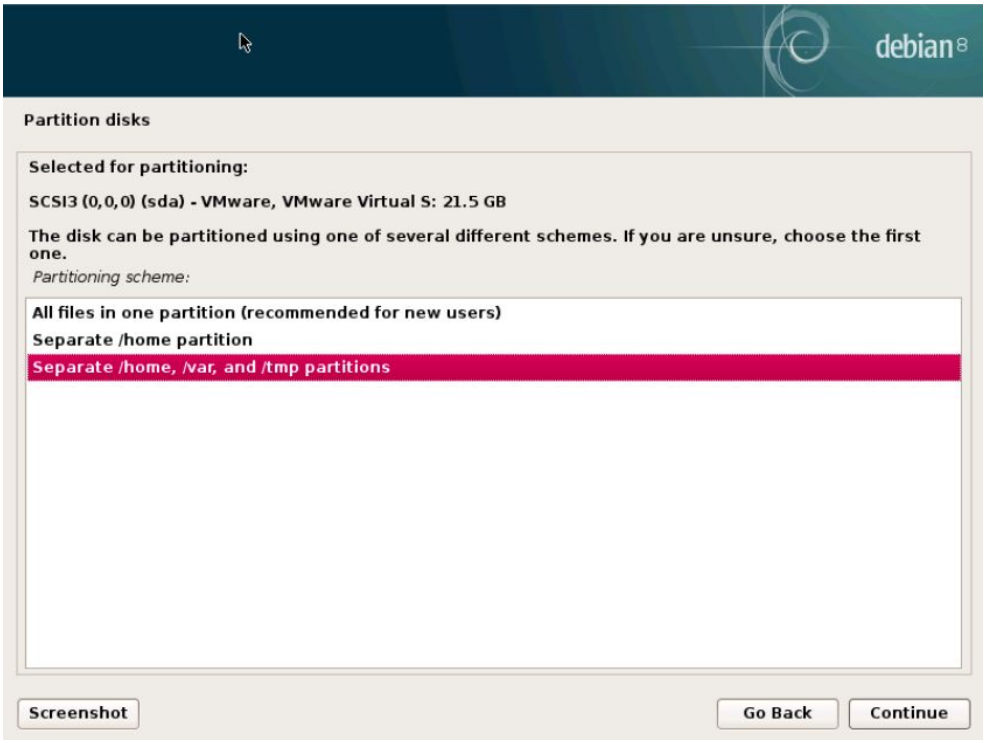


Image 11 - Partition the disk

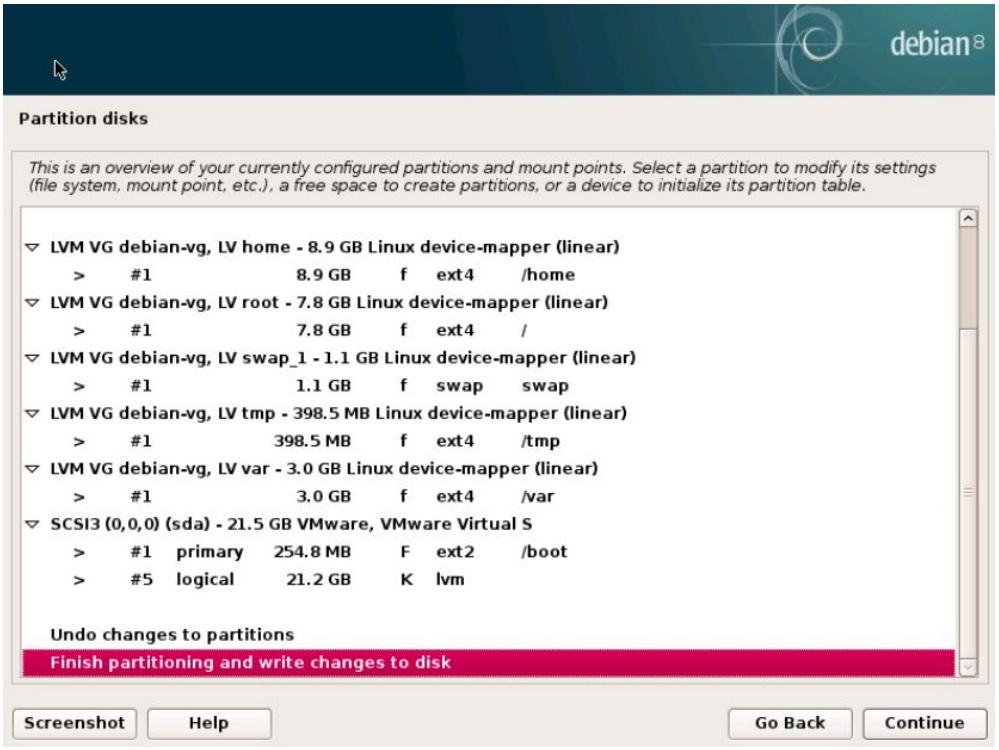


Image 12 - Finalizando o particionamento

Image 13 - Mirror server with Debian packages

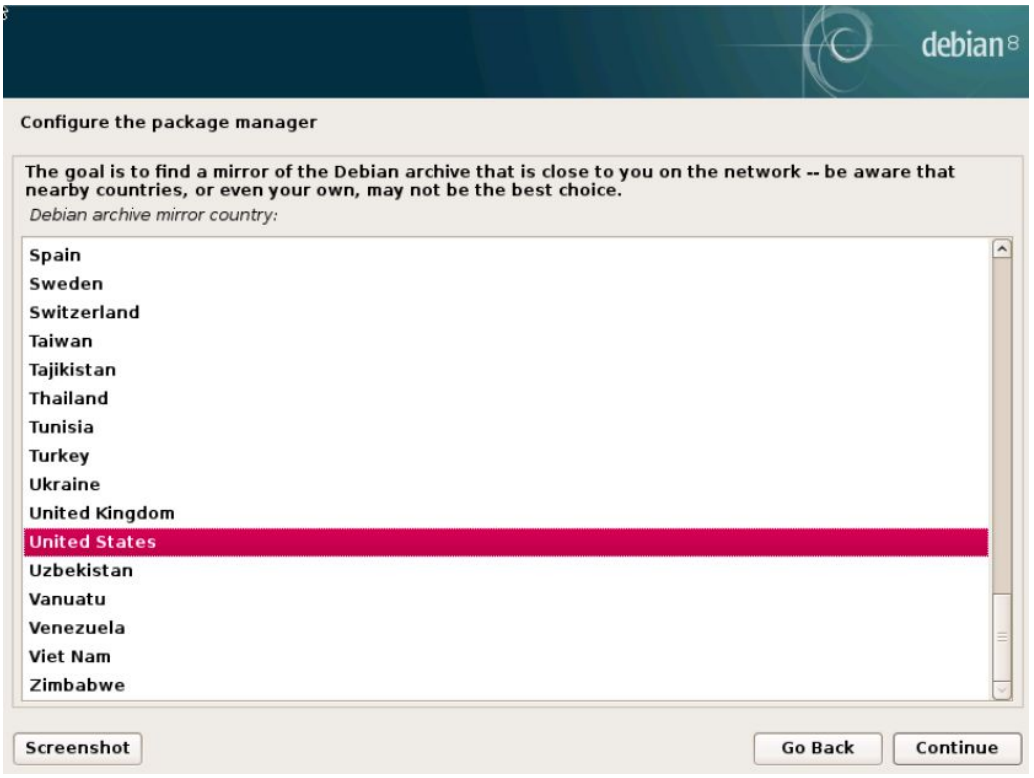


Image 14 - Mirror server location

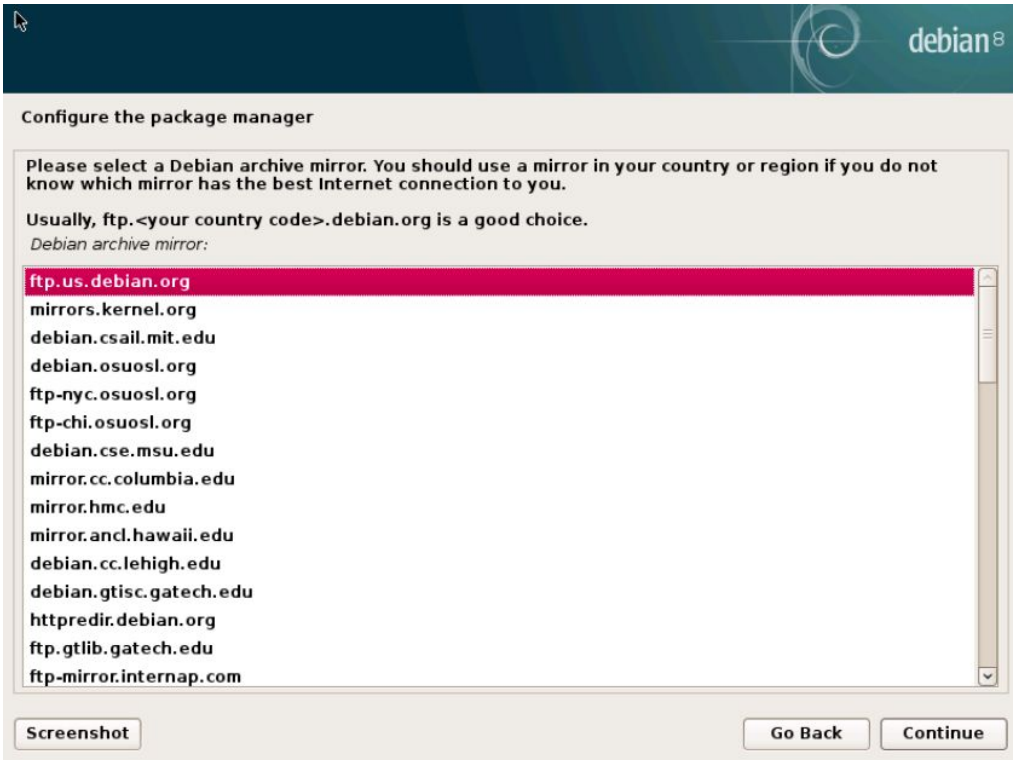


Image 15 - server choice

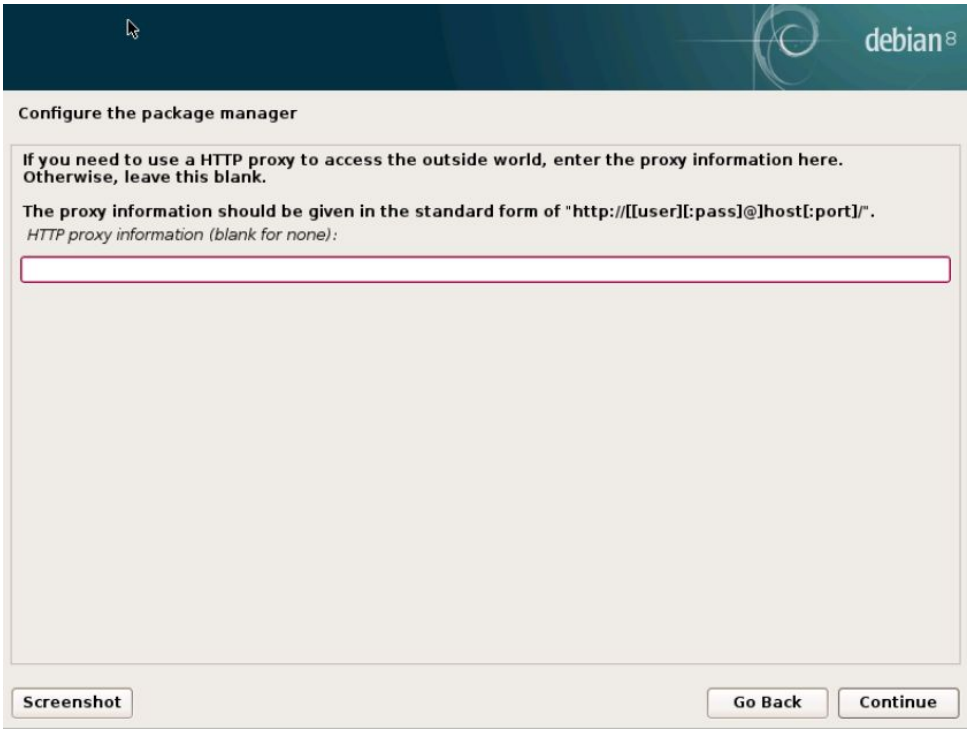


Image 16 - proxy information

Image 17 - GRUB installation

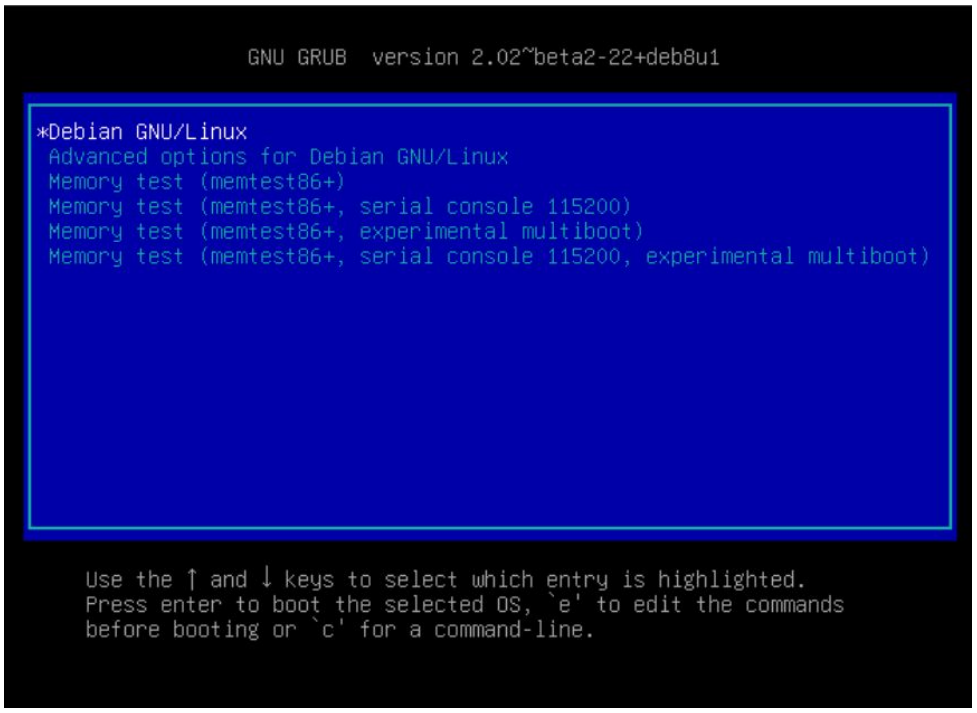


Image 18 - Linux boot (GRUB)

Image 19 - Login Linux

2 - Basic commands

command	description
pwd	show the name of the current directory
whoami	show username show the name of the current directory
id	show the identity of the current user (name, uid, gid, and associated groups)
file <foo>	show the file type for the file "<foo>"
type -p <nome_do_comando>	show the location of a file for the command "<command_name>"
which <nome_do_comando>	command location
type <nome_do_comando>	show information for the command "<command_name>"
apropos <palavra_chave>	show commands related to "<keyword>"
man -k <palavra_chave>	search for applications based on the keyword

whatis <nome_do_comando>	show a one-line explanation for the command "<command_name>"
man -a <nome_do_comando>	show explanation of the command "<command_name>" (Unix style)
info <nome_do_comando>	show a long explanation of the command "<command_name>" (GNU style)
ls	list the contents of the directory (files and directories not hidden)
ls -a	list the contents of the directory (all files and directories)
ls -A	list the contents of the directory (almost all files and directories, that is, skip the ".." and ".")
ls -la	list the entire contents of the directory with detailed information
ls -lai	list the entire contents of the directory with inode number and detailed information
ls -d	list all directories under the current directory
tree	show the contents of the file tree
lsof <foo>	list the open status of the file "<foo>"
lsof -p <pid>	list files opened by the ID process: "<pid>"
mkdir <foo>	create a new "<foo>" directory in the current directory
rmdir <foo>	remove a "<foo>" directory in the current directory

cd <foo>	change the directory to the "<foo>" directory in the current directory or in the directory listed in the "$ CDPATH" variable
cd /	change the directory to the root directory
cd	switch to the current user's home directory
cd /<foo>	switch to the absolute path directory "/ <foo>"
cd ..	switch to the parent directory
cd ~<foo>	change to the home directory of user "<foo>"
cd -	switch to the previous directory
</etc/motd pager	show the contents of "/ etc / motd" using the default pager
touch <junkfile>	create an empty file "<junkfile>"
cp <foo> <bar>	copy an existing "<foo>" file to a new "<bar>" file
rm <junkfile>	remove a "<junkfile>" file
mv <foo> <bar>	rename an existing "<foo>" file to a new name "<bar>" ("<bar>" cannot exist)
mv <foo> <bar>	move an existing "<foo>" file to a new location <bar> / <foo> "(the directory" <bar> "must exist)
mv <foo> <bar>/<baz>	move an existing "<foo>" file to a new location with a new name "<bar> / <baz>" (the "<bar>" directory must exist but the "<bar> / <baz>" directory does not may exist)

chmod 600 <foo>	make an existing file "<foo>" prohibited from being read and written by others (not executable for everyone)
chmod 644 <foo>	make an existing file "<foo>" permissible to be read but forbidden to be written by others (not executable for everyone)
chmod 755 <foo>	make an existing "<foo>" file permissible to be read but forbidden to be written by others (executable for everyone)
find . -name <padrão>	search for matching filenames using a "<standard>" shell (slow)
locate -d . <padrão>	search for matching file names using a "<standard>" shell (faster using a regularly generated database)
grep -e "<padrão>" *.html	look for a "<standard>" in all files ending with ".html" in the current directory and show them all
top	show process information using full screen, press "q" to exit
ps aux | pager	show information of running processes using BSD-style output
ps -ef | pager	show information of running processes using Unix system-V style output
ps aux | grep -e "[e]xim4*"	show all processes running "exim" and "exim4"
ps axf | pager	show the information of all the processes running with output in ASCII art

kill <1234>	kill all processes identified by the process ID: "<1234>"
gzip <foo>	compress "<foo>" to create "<foo> .gz" using Lempel-Ziv encoding (LZ77)
gunzip <foo>.gz	decompress "<foo> .gz" to create "<foo>"
bzip2 <foo>	compress "<foo>" to create "<foo> .bz2" using the text compression algorithm organized in Burrows-Wheeler blocks, and Huffman encoding (better compression than gzip)
bunzip2 <foo>.bz2	decompress "<foo> .bz2" to create "<foo>"
xz <foo>	compress "<foo>" to create "<foo> .xz" using the Lempel – Ziv – Markov chain algorithm (better compression than bzip2)
unxz <foo>.xz	decompress "<foo> .xz" to create "<foo>"
tar -xvf <foo>.tar	extract files from the "<foo> .tar" file
tar -xvzf <foo>.tar.gz	extract files from the gzipped archive "<foo> .tar.gz"
tar -xvjf <foo>.tar.bz2	extract files from the "<foo> .tar.bz2" file
tar -xvJf <foo>.tar.xz	extract files from the "<foo> .tar.xz" file
tar -cvf <foo>.tar <bar>/	archive the contents of the "<bar> /" folder in the "<foo> .tar" file

tar -cvzf <foo>.tar.gz <bar>/	archive the contents of the folder "<bar> /" in the compressed file "<foo> .tar.gz"
tar -cvjf <foo>.tar.bz2 <bar>/	archive the contents of the folder "<bar> /" in the file "<foo> .tar.bz2"
tar -cvJf <foo>.tar.xz <bar>/	archive the contents of the folder "<bar> /" in the file "<foo> .tar.xz"
zcat README.gz | pager	show the contents of "README.gz" compressed using the default pager
zcat README.gz > foo	create the file "foo" with the unzipped content of "README.gz"
zcat README.gz >> foo	add the unzipped content of "README.gz" to the end of the "foo" file (if it doesn't exist, it is created first)

Appendix 2

Basic Shell Script Language

Shell Script is a relatively simple and powerful language. Incredible things that can be automated with simple scripts. To create a shell script just open a text editor (**nano meuscript.sh**) and put the header **#!/Bin/bash**, which is not mandatory, but it is good practice. It means to say which interpreter will be used to execute your code, in this case bash. The file extension (.sh) is also not mandatory.

Image 1 - My First shell script

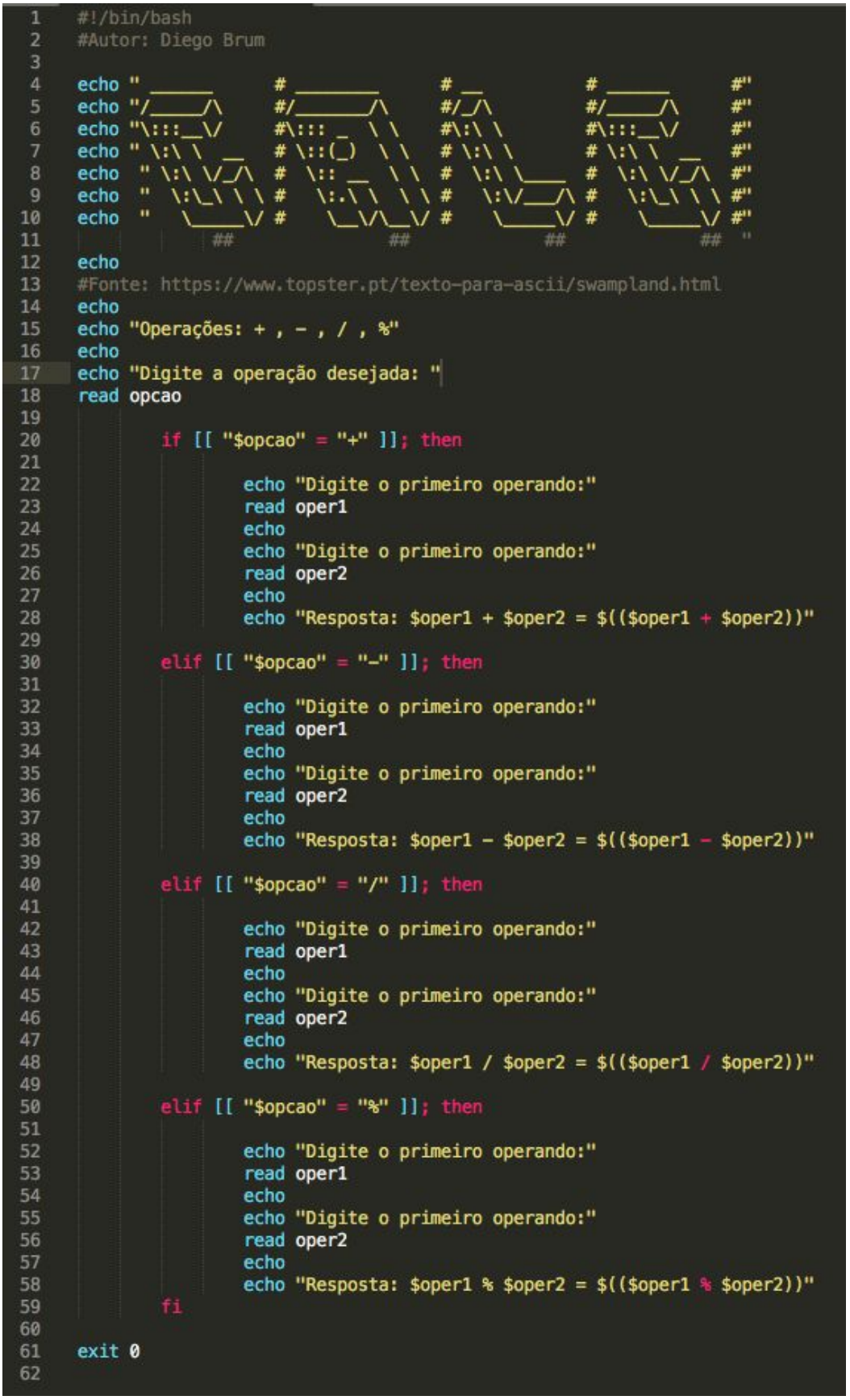
To make it executable, type the command ***chmod +x meuscript.sh***.

To execute: ***./meuscript.sh***

Image 2 - MyScript.sh output

I noticed that the echo command displays a string on the screen. The exit command is to terminate the shell script. Implicitly, even if the exit is not set, it will exist, but I like to put it together with parameter 0 to mean that the program ended without errors. Inheritance when I programmed in Java (System.exit (0)).

Let's make a simple calculator:



```
#!/bin/bash
#Autor: Diego Brum

echo " _______      # ________      # __          # ______      #"
echo "/_____/\     #/_______/\     #/_/\         #/_____/\     #"
echo "\:::__\/     #\::: _  \ \    #\:\ \        #\:::__\/     #"
echo " \:\ \  __   # \::(_)  \ \   # \:\ \       # \:\ \  __   #"
echo "  \:\ \/_/\  #  \:: __  \ \  #  \:\ \____  #  \:\ \/_/\  #"
echo "   \:\_\ \ \ #   \:.\ \  \ \ #   \:\/___/\ #   \:\_\ \ \ #"
echo "    \_____\/ #    \__\/\__\/ #    \_____\/ #    \_____\/ #"
                ##             ##             ##             ##
echo
#Fonte: https://www.topster.pt/texto-para-ascii/swampland.html
echo
echo "Operações: + , - , / , %"
echo
echo "Digite a operação desejada: "
read opcao

        if [[ "$opcao" = "+" ]]; then

                echo "Digite o primeiro operando:"
                read oper1
                echo
                echo "Digite o primeiro operando:"
                read oper2
                echo
                echo "Resposta: $oper1 + $oper2 = $(($oper1 + $oper2))"

        elif [[ "$opcao" = "-" ]]; then

                echo "Digite o primeiro operando:"
                read oper1
                echo
                echo "Digite o primeiro operando:"
                read oper2
                echo
                echo "Resposta: $oper1 - $oper2 = $(($oper1 - $oper2))"

        elif [[ "$opcao" = "/" ]]; then

                echo "Digite o primeiro operando:"
                read oper1
                echo
                echo "Digite o primeiro operando:"
                read oper2
                echo
                echo "Resposta: $oper1 / $oper2 = $(($oper1 / $oper2))"

        elif [[ "$opcao" = "%" ]]; then

                echo "Digite o primeiro operando:"
                read oper1
                echo
                echo "Digite o primeiro operando:"
                read oper2
                echo
                echo "Resposta: $oper1 % $oper2 = $(($oper1 % $oper2))"
        fi

exit 0
```

Image 3 - calculadora.sh

In this calculator project we use **if** and **elif**, which is an else if. We use read to read what the user types. To do mathematical operations, one puts **$(( ))**.

Let's make a shell script that generates 6 random numbers as suggested by the Mega Sena. I will adapt one that I did and it is available in: http://terminalroot.com.br/2015/01/gerando-numeros-para-mega-sena-com.html.

```
#!/bin/bash
# Autor: Diego Brum
# email: diego.brum@gmail.com
# Numeros sugeridos da mega sena
n1=0
n2=0
n3=0
n4=0
n5=0
n6=0

function ordenador {
    echo " " > var1.txt
    printf "%s\n" $@ | sort -n >> var1.txt
    todos=""
    for y in $(cat var1.txt); do
        todos="$todos $y"

    done
    echo -en " \033[42;1;34m $todos \033[0m \n"
}

function sugestao {
echo "Sugestão de jogo:"
for x in {1..6}; do
    num=$(echo "$((RANDOM % 61))")
    while [[ $num -eq 0 ]] || [[ $num -eq $n1 ]] || [[ $num -eq $n2 ]] || [[ $num -eq $n3 ]] \
|| [[ $num -eq $n4 ]] || [[ $num -eq $n5 ]] || [[ $num -eq $n6 ]]
    do
        num=$(echo "$((RANDOM % 61))")
    done
    case "$x" in

        1) n1=$num
            ;;
        2) n2=$num
            ;;
        3) n3=$num
            ;;
        4) n4=$num
            ;;
        5) n5=$num
            ;;
        6) n6=$num
            ;;
    esac
done
ordenador "$n1 $n2 $n3 $n4 $n5 $n6"
}

echo
sugestao
echo
exit 0
```

Image 4 - megasena.sh

In the Mega Sena project we use:

- functions to make the code more efficient;
- I made use of printf, which is equivalent to echo.
- use of variables in text files (var1.txt)
- colors in text output using echo -e
- for and while loop
- conditional case structure

Foreground (text)

Code	Color	Example	Preview
39	Default foreground color	`echo -e "Default \e[39mDefault"`	Default Default
30	Black	`echo -e "Default \e[30mBlack"`	Default Black
31	Red	`echo -e "Default \e[31mRed"`	Default Red
32	Green	`echo -e "Default \e[32mGreen"`	Default Green
33	Yellow	`echo -e "Default \e[33mYellow"`	Default Yellow
34	Blue	`echo -e "Default \e[34mBlue"`	Default Blue
35	Magenta	`echo -e "Default \e[35mMagenta"`	Default Magenta
36	Cyan	`echo -e "Default \e[36mCyan"`	Default Cyan
37	Light gray	`echo -e "Default \e[37mLight gray"`	Default Light gray
90	Dark gray	`echo -e "Default \e[90mDark gray"`	Default Dark gray
91	Light red	`echo -e "Default \e[91mLight red"`	Default Light red
92	Light green	`echo -e "Default \e[92mLight green"`	Default Light green
93	Light yellow	`echo -e "Default \e[93mLight yellow"`	Default Light yellow
94	Light blue	`echo -e "Default \e[94mLight blue"`	Default Light blue
95	Light magenta	`echo -e "Default \e[95mLight magenta"`	Default Light magenta
96	Light cyan	`echo -e "Default \e[96mLight cyan"`	Default Light cyan
97	White	`echo -e "Default \e[97mWhite"`	Default White

Image 5 - Foreground color

Background

Code	Color	Example	Preview
49	Default background color	`echo -e "Default \e[49mDefault"`	Default Default
40	Black	`echo -e "Default \e[40mBlack"`	Default Black
41	Red	`echo -e "Default \e[41mRed"`	Default Red
42	Green	`echo -e "Default \e[42mGreen"`	Default Green
43	Yellow	`echo -e "Default \e[43mYellow"`	Default Yellow
44	Blue	`echo -e "Default \e[44mBlue"`	Default Blue
45	Magenta	`echo -e "Default \e[45mMagenta"`	Default Magenta
46	Cyan	`echo -e "Default \e[46mCyan"`	Default Cyan
47	Light gray	`echo -e "Default \e[47mLight gray"`	Default Light gray
100	Dark gray	`echo -e "Default \e[100mDark gray"`	Default Dark gray
101	Light red	`echo -e "Default \e[101mLight red"`	Default Light red
102	Light green	`echo -e "Default \e[102mLight green"`	Default Light green
103	Light yellow	`echo -e "Default \e[103mLight yellow"`	Default Light yellow
104	Light blue	`echo -e "Default \e[104mLight blue"`	Default Light blue
105	Light magenta	`echo -e "Default \e[105mLight magenta"`	Default Light magenta
106	Light cyan	`echo -e "Default \e[106mLight cyan"`	Default Light cyan
107	White	`echo -e "Default \e[107mWhite"`	Default

Image 6 - Background color

With the projects presented, you will be able to make good scripts. Good luck!

Bibliography

https://e-tinet.com/linux/tabelas-do-iptables-firewall-linux/

https://wiki.sj.ifsc.edu.br/wiki/index.php/Tabelas_de_uso_do_IPTables

https://pt.wikibooks.org/wiki/Guia_do_Linux/Avançado/Firewall_iptable/A_tabela_mangle

https://keepass.info/%0D/download.html

http://fwbuilder.sourceforge.net

https://www.debian.org

https://ossec-docs.readthedocs.io/en/latest/manual/output/syslog-output.html

https://ossec-docs.readthedocs.io/en/latest/manual/rules-decoders/rule-levels.html

https://blog.wpscans.com/using-ossec-to-monitor-directory-and-file-changes-in-wordpress/

https://www.digitalocean.com/community/tutorials/how-to-use-apache-as-a-reverse-proxy-with-mod_proxy-on-ubuntu-16-04

https://pt.wikipedia.org/wiki/Hardening

https://www.tecmint.com/auto-install-security-updates-on-debian-and-ubuntu/

https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

https://www.tecmint.com/hide-apache-web-server-version-information/

https://debian-handbook.info/browse/pt-BR/stable/sect.regular-upgrades.html

http://rkhunter.sourceforge.net

http://www.chkrootkit.org

http://www.unhide-forensics.info/?Linux

https://packages.debian.org/sid/mtr-tiny

https://linux.die.net/man/1/whowatch

https://servidordebian.org/pt/wheezy/security/audit/debsecan

https://modsecurity.org/about.html

https://www.linode.com/docs/web-servers/apache-tips-and-tricks/configure-modsecurity-on-apache/

https://pt.wikipedia.org/wiki/Git

https://www.alienvault.com/products/ossim

https://pt.wikipedia.org/wiki/Gerenciamento_e_Correlação_de_Eventos_de_Segurança

https://www.alienvault.com/products/ossim

https://networkhop.wordpress.com/2016/04/27/port-mirroring-with-iptables/

https://cybermap.kaspersky.com

https://github.com/MatthewClarkMay/geoip-attack-map

https://threatmap.checkpoint.com/ThreatPortal/livemap.html

http://www.norse-corp.com

https://www.mapbox.com

https://www.elastic.co/pt/elk-stack

https://www.stamus-networks.com/open-source/

https://www.debian.org/doc/manuals/debian-reference/ch01.pt.html#_midnight_commander_mc

https://misc.flogisoft.com/bash/tip_colors_and_formatting