Chapter 2: Presentation of the existing and the proposed solutions

Introduction:

In this chapter, we'll study the network architecture of Data Era as well as the existing security solution and we'll introduce the proposed solution using open source and free to use security software and tools.

1 Study of the existing

In the following figure, we see the existing network architecture of the company. It is only based on an ISP router connected to a UMT FortiGate unit, spreading to a variety of devices used by the company members, and a web server for public access.

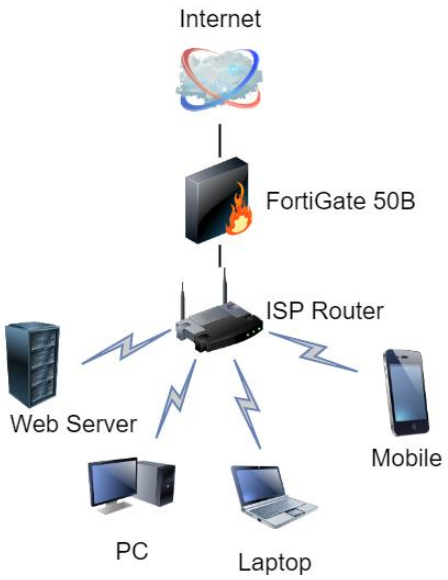

Figure 6: Simplified existing Data Era architecture

1.1 Existing Security Solution

Data Era is using an UTM called the FortiGate-50B, which offers dual WAN ports for load balancing or redundant internet connections. It provides three integrated switch ports for multi-user environments in a small remote office. It is ideally suited for remote offices, retail stores, broadband telecommuter sites and many other applications. The FortiGate-50B provides complete real-time network protection through a combination of network-based antivirus, web and email content filtering, firewall, VPN, network-based intrusion detection and prevention, and traffic shaping.

Figure 7: FortiGate 50B

FortiOS 4.0 is a software of FortiGate multi-threat security platforms.

Here are the technical specifications of a FortiGate 50B

Table 1: FortiGate 50B specifications [20]

Hardware Specifications	
10/100 WAN Interfaces (Copper, RJ-45)	2
10/100 Internal Switch Interfaces (Copper, RJ-45)	3
Console (Copper, RJ-45)	1
USB Interfaces	2
System Performance	
Firewall Throughput (1518 byte UDP packets)	100 Mbps
Firewall Throughput (512 byte UDP packets)	50 Mbps
IPsec VPN Throughput	48 Mbps
IPS Throughput	50 Mbps
Antivirus Throughput	19 Mbps
Gateway-to-Gateway IPsec VPN Tunnels	20
Client-to-Gateway IPsec VPN Tunnels	20
Concurrent Sessions	25,000
New Sessions/Sec	2,000
Concurrent SSL-VPN Users (Recommended Max)	20
SSL-VPN Throughput	15 Mbps
Firewall Policies (Max)	2,000
Virtual Domains (Max / Default	10 / 10
Unlimited User Licenses	Yes
Mean Time Between Failures	More than 5 years
Dimensions	
Height x Width x Length (in)	1.38 x 8.63 x 5.8 in
Height x Width x Length (cm)	3.51 x 21.92 x 14.73cm
Weight	1.5 lbs (0.68 kg)
Wall Mountable	Yes
Environment	
Power Required	100-240 VAC, 50-60 Hz, 0.8 Amp (Max)
Power Consumption (AVG)	6W
Heat Dissipation	42 BTU
Operating Temperature	32 – 104 deg F (0 – 40 deg C)
Storage Temperature	-13 to 158 deg F (-25 to 70 deg C)
Humidity	5 to 95% non-condensing

1.2 Critique of the existing solution

Why is Data Era giving up this solution?

Although there are many advantages of using an UTM, there are also a few disadvantages. Combining every security component together into one appliance allows system administrators to manage everything in one dashboard. It also presents a potential single point of failure, allowing a network to be completely exposed if the UTM appliance fails. This can be averted through redundancy, by deploying a high availability configuration, but will increase setup and running costs.

One of the most prominent weaknesses that UTMs suffer from is the fact that updates are spaced quite far apart. Generally, depending upon the company that has created the UTM, the updates can range anywhere from between a couple of months to a year, per se. In the internet world, this translates to roughly a century. That is not acceptable at all. Often times, updates are usually released after proper penetration testing and vulnerability scanning has been carried out in order to determine the areas in the network that can be exploited. And when these reports/ findings are given out to the company for remedial purposes, it takes anywhere between one to three months to create a specific patch.

Also to mention that upgrading to a newer version means cutting off all support to previous one, which makes the maintenance cost very high by obligating the client to pay for an upgrade. Renewing licenses and paying them over and over was a huge no-no for DataEra.

Another risk that must be assessed when deploying an UTM solution is that the security of your network depends on a single vendor. Vendor diversity is considered a best practice for network security, as many vendors use the same malware detection algorithms in their product set. If something gets missed by one vendor's product, there's a chance that it'll get missed by other products from that vendor also. If you have used Provider A for UTM and Provider B on the product endpoints, you are less likely to miss a specific malicious item because each provider runs different detection algorithms.

2 Proposed Solutions

In this section, we'll present the proposed solutions that we need to implement to secure the company's network.

2.1 pfSense

pfSense software is a free and open-source customized distribution of FreeBSD specifically made for the use as a firewall/router that is entirely managed via web interface. In addition to being a powerful, flexible firewalling and routing platform, it includes a long list of related features and packages allowing further expandability without adding bloat and potential security vulnerabilities to the base distribution. The pfSense project is hosted and developed by Rubicon Communications, LLC (NetGate).[21]

Figure 8: pfSense Logo

The variety of firewall applications is vast, but what made us choose pfSense over other options are its:

- Web User Interface: With new web UI, based on Bootstrap framework, you can control your pfSense from everywhere.
- Scalability: It's an all-in-one solution useful for every kind of company. It's also very easy to set up rules and NAT, and it has several modules like transparent proxy, VPN, and traffic shaping.
- Community: There's a large community behind pfSense so you can find a lot of documentation, tutorials, and how-to and also support from the official forum.

2.2 Snort

Snort is the foremost Open-Source Intrusion Prevention System (IPS) in the world. Snort IPS uses a series of rules that help identify malicious network activity and uses those chosen rules to find packets that match against them and creates alerts for the users or deployed inline to stop these packets as well.

Snort has 3 primary uses: As a packet sniffer like tcpdump, as a packet logger, which is useful for network traffic debugging, or it can be used as a full-blown network intrusion prevention system.[22]

Figure 9: Snort logo

Snort was an obvious choice of ours thanks to its advantages like:

- The threat intelligence from Cisco TALOS is unparalleled. This is grafted into the Sourcefire application which greatly improves security visibility.
- In depth information, we are able to do more than just see alerts, we can view the full information and packets that lead to the conclusion, though the conclusion is prepared in advance for us.
- High flexibility and compatibility with pfSense firewall.

2.3 FreeIPA

FreeIPA is an integrated security information management solution combining Linux (Fedora), MIT Kerberos, NTP, DNS, 389 Directory Server, Dogtag (Certificate System). It consists of a web interface and command-line administration tools.

FreeIPA is an integrated Identity and Authentication solution for Linux/UNIX networked environments. A FreeIPA server provides centralized authentication, authorization and account information by storing data about user, groups, hosts and other objects necessary to manage the security aspects of a network of computers.[23]

Figure 10: FreeIPA logo

Some of the features that made us choose FreeIPA over other LDAP solutions are:

- Allows central management of security mechanisms like passwords, SSH Public Keys, SUDO rules, Key tabs, Access Control Rules
- Enables delegation of selected administrative tasks to other power users
- Integrates into Active Directory environment

2.4 Zabbix

Zabbix is an enterprise-class open source distributed monitoring solution.

Zabbix is a software that monitors numerous parameters of a network and the health and integrity of servers, applications, services, virtual machines, databases, websites, the cloud and much more. Zabbix uses a flexible notification mechanism that allows administrators to make mail-based alerts for virtually any chosen event. This allows a fast reaction to server problems and resolution. Zabbix also offers excellent reporting and data visualization features based on the obtained stored data. This makes it ideal for capacity planning.[24]

Figure 11: Zabbix logo

Zabbix was an obvious choice to integrate in our open-source security solution, thanks to some of its advantageous points:

- Very complete documentation. Almost every aspect of Zabbix has been documented and reported on.
- Template system is really great, making it super easy to add new services and monitoring, graphs, etc. to any server.
- Provides a "plugin architecture" (via XML templates) to allow end users to extend it to monitor all kinds of equipment, software, or other metrics that are not already added into the software already.

3 Proposed network

Using the servers previously discussed, we are opting to configure new network as shown in the next figure.

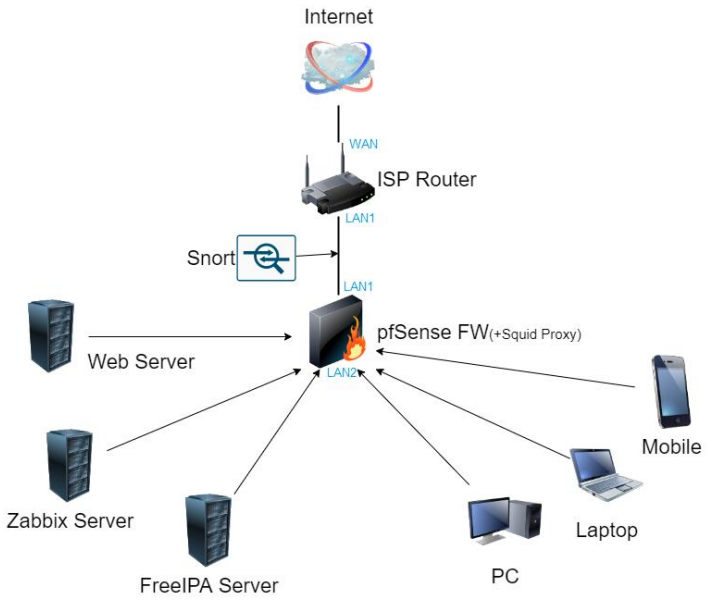

Figure 12: Proposed network

Conclusion:

In this chapter, we have presented both the existing network architecture and the new proposed one that will replace it, mentioning the main tools that we will be using to strengthen the security of the company. In the next chapter, we will be diving into the basic but most important configurations that are made.

Chapter 3 : Implementation of the security solutions

Introduction:

In this chapter we will present how we implemented our chosen solutions, according to the company's needs and demands. This chapter contains three parts: First of all, pfSense , which could be considered as the main focus of our project's work, thanks to the values and utilities it brings. Secondly, we will pass to Zabbix and finally we will briefly take a look at FreeIPA.

PLEASE NOTE: Not everything is going to be included in this report, only the necessary configurations, respecting the company's confidentiality and privacy. Some settings or information will be blurred or cut off.

1 pfSense

After the installation of pfSense as a virtual machine on our host running ESXi 6.7, we configured both its interfaces as follow:

WAN interface IP address 192.168.100.10; GW 192.168.100.1 (ISP router IP address)

LAN interface IP address 192.168.10.1

IPv6 is disabled.

To access the pfSense GUI, we type in its LAN address in any web browser from a machine connected to our LAN network. After logging in, we can check what are most of pfSense's functionalities in our network, simply by going to the Status→Services tab.

Status / Services

Services

	Service	Description	Status	Actions
1	bandwidthd	BandwidthD bandwidth monitoring daemon	✓	
2	c-icap	ICAP Interface for Squid and ClamAV integration	✓	
3	clamd	ClamAV Antivirus	✓	
4	dhcpd	DHCP Service	✓	
5	dpinger	Gateway Monitoring Daemon	✓	
6	lightsquid_web	Lightsquid Web Server	✓	
7	openvpn	OpenVPN server: Serveur VPN Société	✓	
8	pcscd	PC/SC Smart Card Daemon	✓	
9	pfb_dnsbl	pfBlockerNG DNSBL service	✓	
10	pfb_filter	pfBlockerNG firewall filter service	✓	
11	snort	Snort IDS/IPS Daemon	✓	
12	squid	Squid Proxy Server Service	✓	
13	squidGuard	Proxy server filter Service	✓	
14	syslogd	System Logger Daemon	✓	
15	unbound	DNS Resolver	✓	
16	zabbix_agentd	Zabbix Agent Host Monitor Daemon	✓	

Figure 13 : Running services on pfSense

Some services are built in by default into the installation of pfSense while others are obtained as add-ons and be downloaded from the System→Package manager. A variety of packages are available for multiple use, depending on the admin's preferences.

We will explain the purpose of each running service, some briefly and more detailed for others, depending on their complexity and their use.

1.1 Bandwidthd

Is a added package, it tracks usage of TCP/IP network subnets and builds html files with graphs to display utilization.

Charts are built by individual IPs, and by default display utilization over 2 days, 8 days, 40 days, and 400 days periods. Furthermore, each IP address's utilization can be logged out in CDF format, or to a backend database server. HTTP, TCP, UDP, ICMP, VPN, and P2P traffic are color coded.

Bandwidthd can be found under Diagnostics→BandwidthD tab.

Ip and Name	Total	Total Sent	Total Received	FTP	HTTP	P2P	TCP	UDP	ICMP
Total	811.3M	80.6M	730.6M	672	654.4M	3.0K	717.5M	93.8M	34.8K
192.168.10.18	200.1M	7.2M	192.9M	0	182.7M	0	199.8M	358.7K	1.1K
192.168.10.26	170.3M	3.0M	167.3M	0	170.2M	0	170.2M	82.6K	0
192.168.10.28	146.6M	7.5M	139.2M	0	129.0M	0	146.6M	66.0K	5.7K
192.168.10.200	103.0M	9.8M	93.1M	336	88.4M	1.5K	102.6M	334.3K	7.6K
192.168.10.25	102.3M	37.5M	64.8M	0	60.7M	0	60.7M	41.6M	0
192.168.10.23	36.7M	3.1M	33.6M	0	2.6M	0	2.7M	34.1M	0
192.168.10.24	17.5M	1000.0K	16.6M	0	1.5M	0	1.6M	16.0M	1.7K
192.168.10.1	16.2M	8.4M	7.8M	336	1.2M	1.5K	15.3M	883.1K	17.4K
192.168.10.16	13.0M	768.1K	12.2M	0	12.7M	0	12.8M	184.0K	1.2K
192.168.10.30	4.2M	1.8M	2.4M	0	4.1M	0	4.1M	32.1K	0
192.168.10.15	729.4K	367.2K	362.1K	0	691.3K	0	705.4K	24.0K	0
192.168.10.9	444.1K	162.4K	281.7K	0	320.9K	0	320.9K	123.2K	0
192.168.10.100	140.2K	55.7K	84.5K	0	97.5K	0	97.5K	42.7K	0
192.168.10.255	46.7K	0	46.7K	0	0	0	0	46.7K	0
192.168.10.13	5.3K	2.2K	3.1K	0	2.8K	0	2.8K	2.4K	56

Figure 14 : BandwidthD index page

1.2 C-icap

Is installed with the Squid package. C-icap is an implementation of an ICAP server. It can be used with HTTP proxies that support the ICAP protocol to implement content adaptation and filtering services.

Most of the commercial HTTP proxies must support the ICAP protocol. The open source Squid 3.x proxy server supports it, web antivirus service, using the clamav open-source antivirus engine.

1.3 Clamd

Is installed with the Squid package. SquidClamav is an antivirus for Squid proxy based on the Awards winnings ClamAv anti-virus toolkit.SquidClamav is the most efficient Squid Redirector and ICAP service antivirus tool for HTTP traffic available for free, it is written in C and can handle thousands of connections.

The antivirus interface can be found under Services→Squid proxy server→Antivirus.

We set it up to update its database every 24h from the official ClamAV db and from two other unofficial signatures, URLhaus and InterServer.

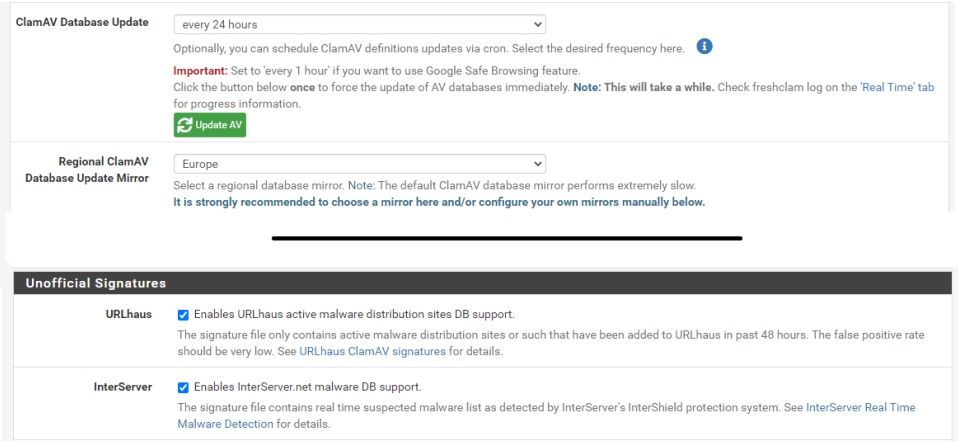

Figure 15 : ClamAV essential configuration

1.4 DHCPD

Is installed by default. With this daemon pfSense can act as a DHCP Server, which a network server that automatically provides and assigns IP addresses, default gateways and other network parameters to client devices. It relies on the standard protocol known as Dynamic Host Configuration Protocol or DHCP to respond to broadcast queries by clients.

The daemon settings are in Services→DHCP Server. We enabled the server on the LAN interface. The following shows the necessary configuration of the server.

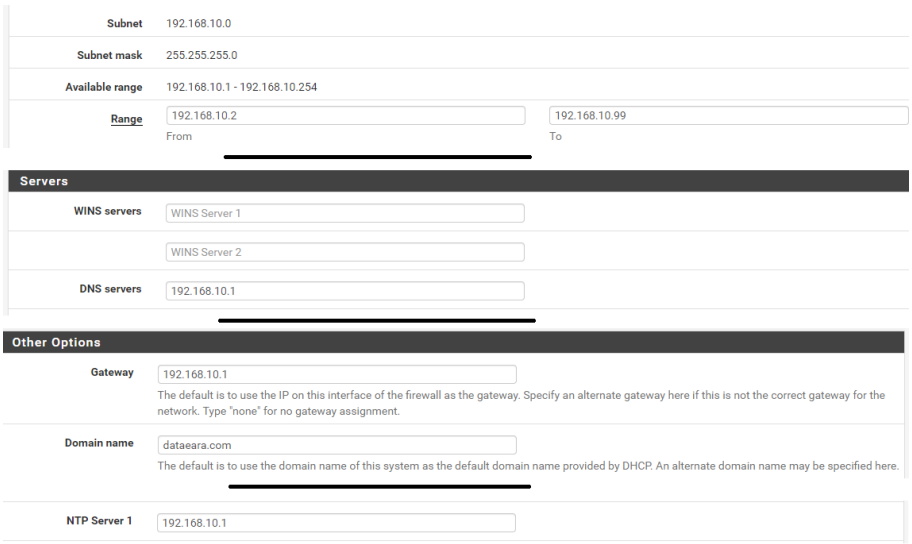

Figure 16 : DHCP Server essential configuration

1.5 dpinger

Is installed by default. To conserve bandwidth, the dpinger daemon sends a ping with a payload size of 0 by default so that no data is contained within the ICMP echo request. However, in rare circumstances a CPE, ISP router, or intermediate hop may drop or reject ICMP packets without a

payload. In these cases, set the payload size above 0. Usually, a size of 1 is enough to satisfy affected equipment. We have it setup to ping an external IP (a google DNS server) instead of the gateway of pfSense (the ISP router) because it keeps rejecting the ICMP packets. Still same functionality with a different end point.

The daemon can be found under Services→Gateways, as it continuously pings checking for internet accessibility.

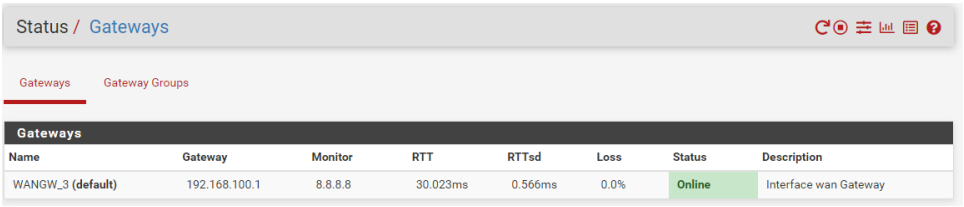

Figure 17 : dpinger service

1.6 lightsquid_web:

Is installed with the Lighsquid package. Lightsquid is used to create reports that detail the web history of computers that have accessed sites through the proxy. The look and feel of the reports may be customized by choosing the Language, Bar color, and Report Scheme. The Refresh scheduler option controls how often the report will be automatically updated, e.g. every 30 minutes.

In order to get data in the report, we have to check Enable Logging is set in Squid, and that the user traffic is going through the proxy as expected.

The report settings may be found under Status > Squid Proxy Reports.

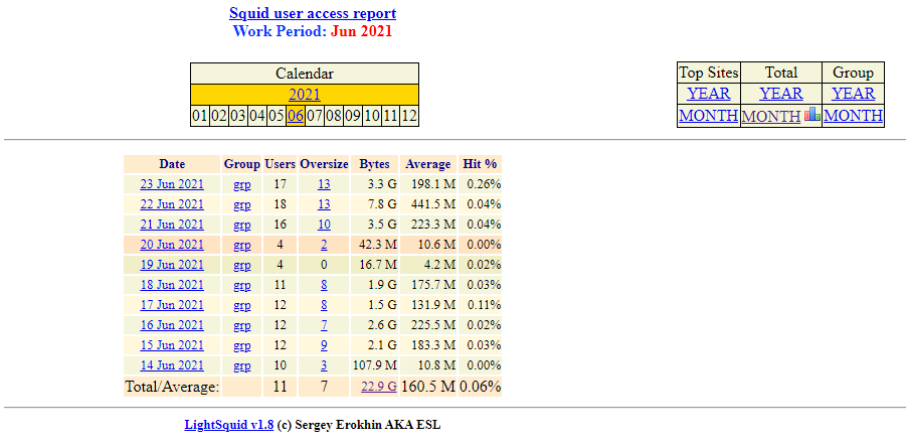

Figure 18: Lightsquid service front page

More detailed information can be viewed either by day or user, showing the visited sites and some cached files. More lists are available, such as top visited sites or top files by user or by period, and caching depends on the setting of the Squid proxy server.

Real time stats of the proxy downloads, not a live view of the log, is also found by clicking “sqstats” next to the light squid button.

1.7 OpenVPN

Is installed by default. OpenVPN is an open-source SSL VPN solution that can be used for remote access clients and site-to-site connectivity

Every OpenVPN connection, whether remote access or site-to-site, consists of a server and a client. In the case of site-to-site VPNs, one firewall acts as the server and the other as the client. It does not matter which firewall possesses these roles. Typically, the location of the primary firewall will provide server connectivity for all remote locations, like in our case.

The pfSense GUI includes a certificate management interface that is fully integrated with OpenVPN. Certificate authorities (CAs) and server certificates are managed in the Certificate Manager in the web interface, located at System → Cert Manager.

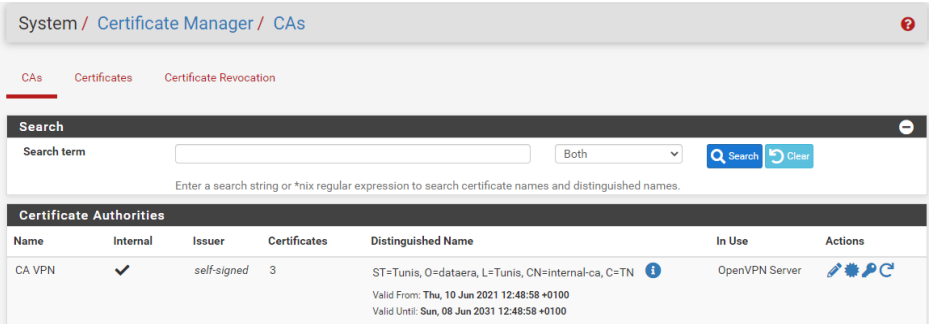

Figure 19 : OpenVPN self-signed CA

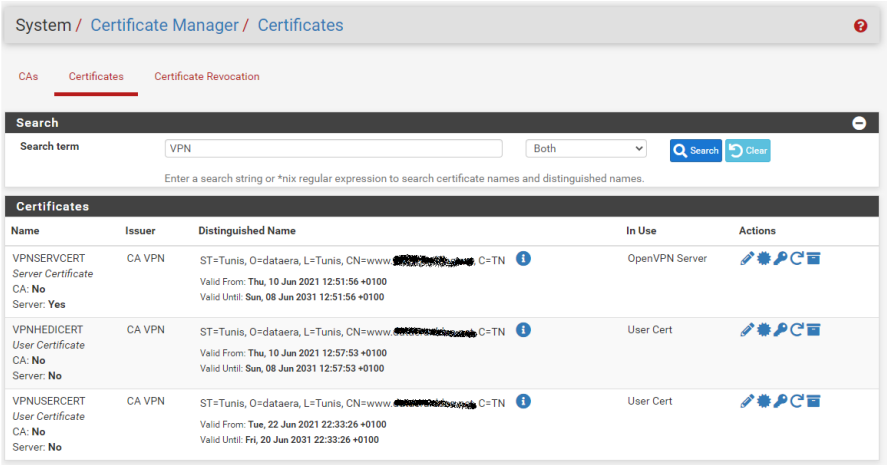

Figure 20 : Certificates issues by the VPN CA

The common name is the dynamic DNS hostname, that can be found under Services→Dynamic DNS, its being blurred for privacy reasons.

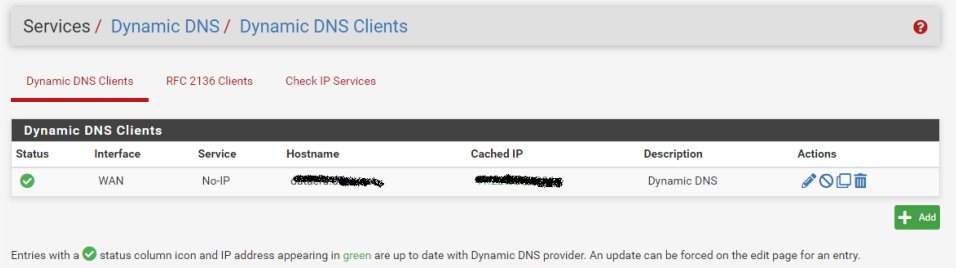

Figure 21 : DynDNS parameters

User certificates are also managed in the web interface, as a part of the built-in user manager found at System → User Manager. Certificates may be generated for any user account created locally on the firewall except for the default admin account. For the current company needs, 2 accounts should be issued, one for admin access (user Hedi), and another for a client access with limited access (user User1).

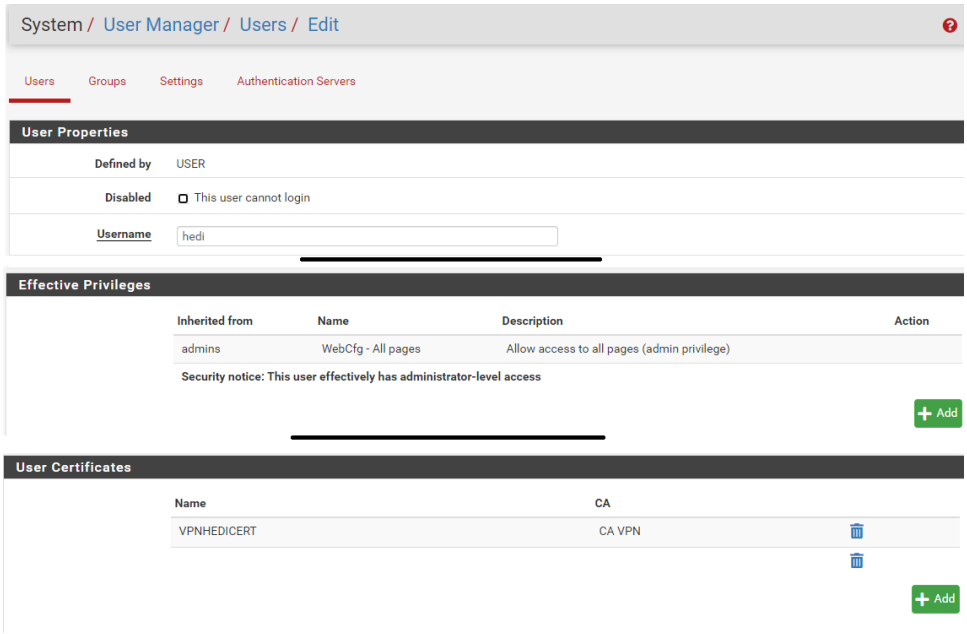

Figure 22: Admin user Hedi essential configuration

The server configuration options are available in one or more modes for OpenVPN server instances, managed from VPN > OpenVPN, on the Servers tab. We chose a 10.10.10.0/24 network with TCP4 protocol on port 8080 connection for our VPN tunnel network, looking from the WAN interface of pfSense, which routes to the ISP router then the internet.

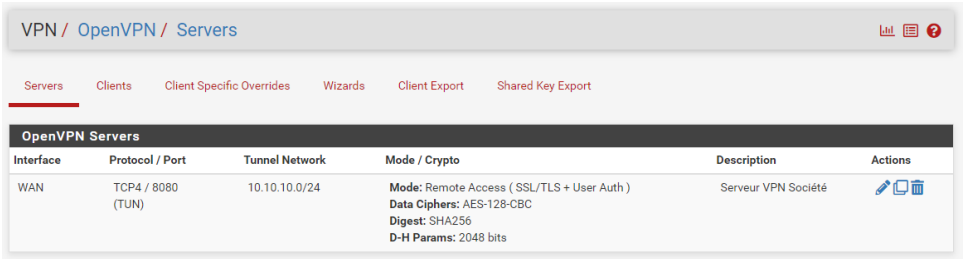

Figure 23 : OpenVPN server configuration

A rule will be automatically added to our pfSense firewall to accept any incoming traffic from the wan interface via IPv4TCP on port 8080 only.

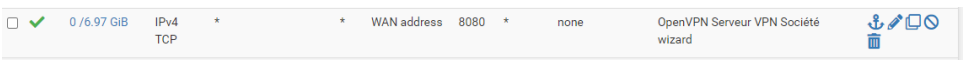

Figure 24 : Firewall rule accepting VPN connections

We now have to forward the incoming traffic from out external IP address from our ISP router facing outbound, to our pfSense where our OpenVPN server is installed, to complete the VPN tunnel link.

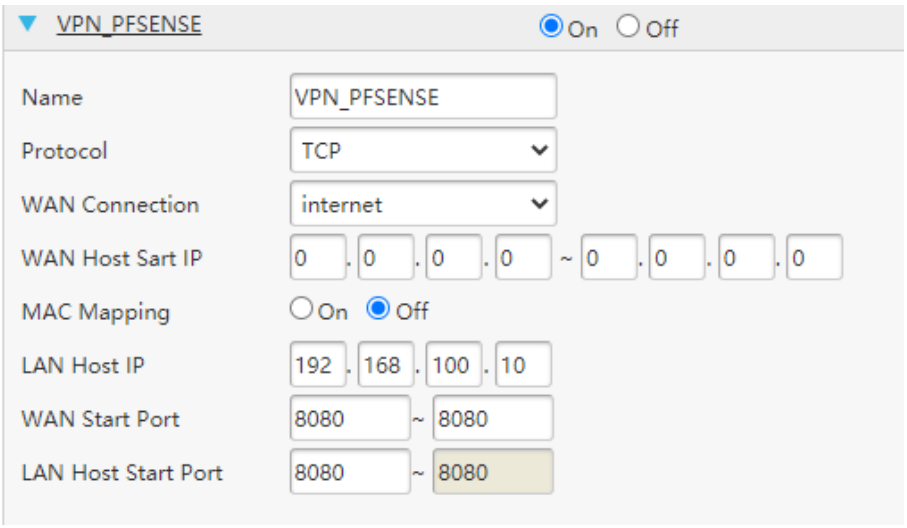

Figure 25 : ISP router port forwarding VPN configuration

Our server is now configured and setup, and all is left to do on the server side is to export the users parameters that will be installed on the client side using the package “OpenVPN-client-export", found under VPN→OpenVPN→Client export. We need to make sure to select the proper VPN server and the correct host name resolution (the dyndns hostname in our case).

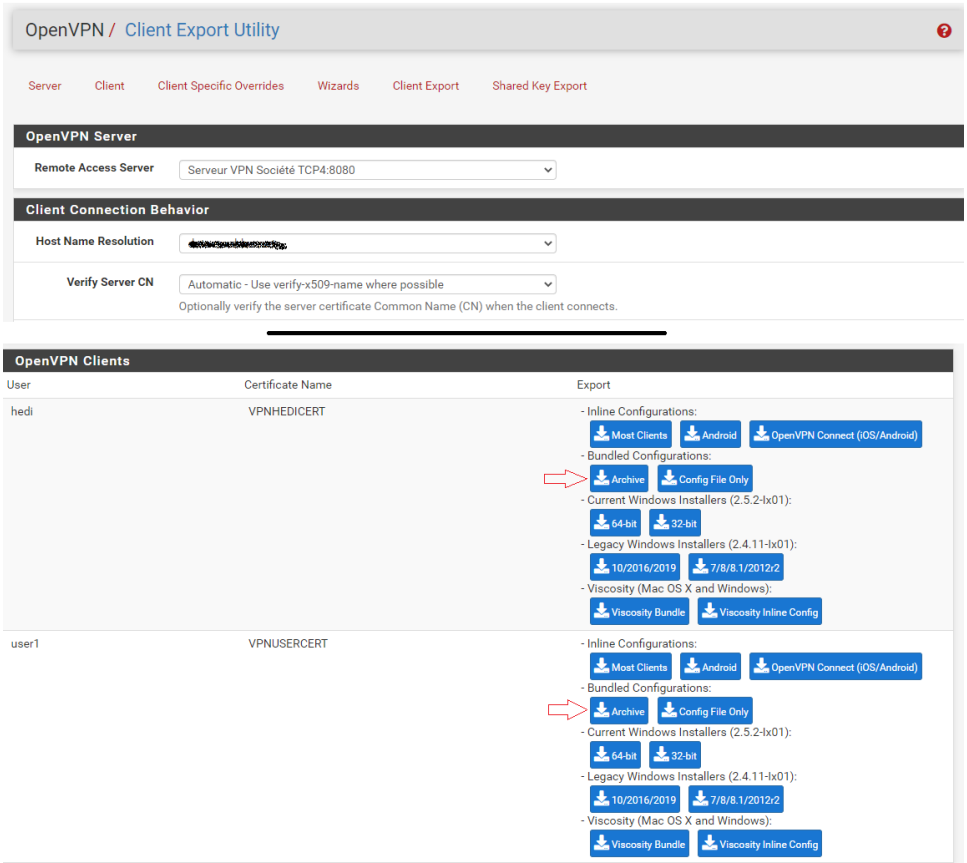

Figure 26 : Users config files export

For the client side, we need to download and install the OpenVPN client, then add the exported user settings to the “config” file of the OpenVPN application, then launch it and enter the credentials.

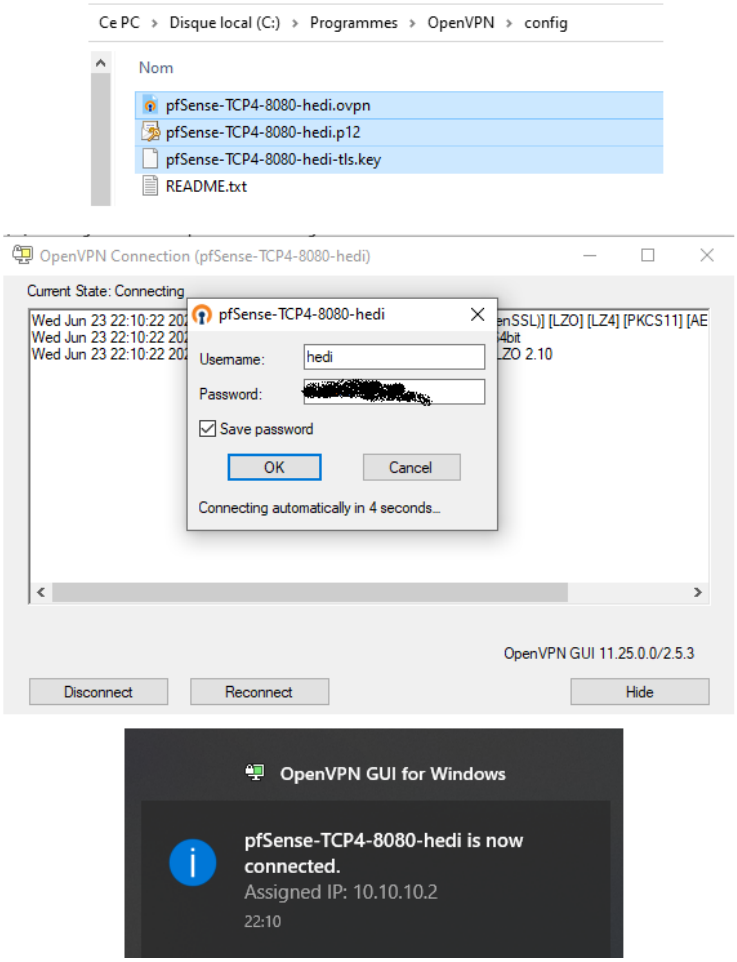

Figure 27 : Client side OpenVPN configuration

We are now logged in to our internal network via the VPN tunnel, but this is for the admin user Hedi who has total access to this network. And for our other user User1, who we want to have limited access, we need to get back to our server side and create these limitations. We have to assign this specific user a static IP address on our VPN network, to do that we go to VPN→OpenVPN→Client Specific Overrides.

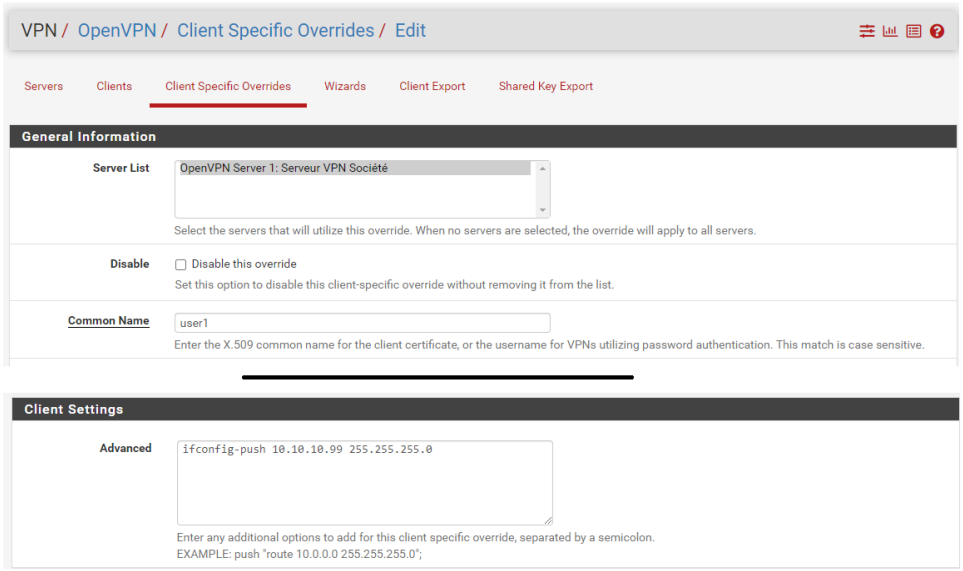

Figure 28 : VPN client specific overrides

Then we add rules on our firewall to restrict the access of that IP address Firewall→Rules →OpenVPN.

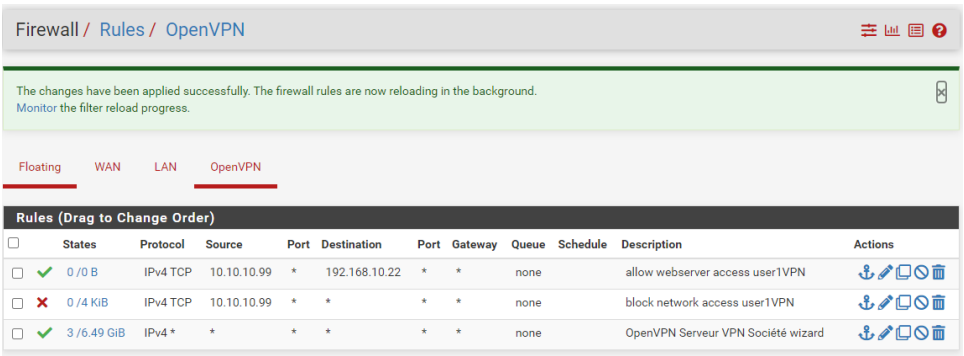

Figure 29 : Firewall rule on specific VPN client

1.8 pcscd

Is installed by default. PC/SC Smart Card Daemon, is the daemon program for pcsc-lite and the MuscleCard framework. It is a resource manager that coordinates communications with smart card readers and smart cards and cryptographic tokens that are connected to the system. We are currently not using this service.

1.9 pfb_dnsbl

Is installed with the pfBlockerNG-devel package. Domain Name System Blacklists, also known as DNSBL's or DNS Blacklists, are spam blocking lists that allow a website administrator to block messages from specific systems that have a history of sending spam. As their name implies, the lists are based on the Internet's Domain Name System.

In order for DNSBL to function, pfBlockerNG and DNS Resolver should both be enabled.

The configuration tab can be found under Firewall→pfBlockerNG→DNSBL.

Most of the settings are left in default, we just changed the virtual IP to 10.10.100.1 where rejected DNS Requests will be forwarded to it, and changed the list action to “Deny Both” which blocks all traffic in both directions, if the source or destination IP is in the block list.

Moving to DNSBL Groups, we can add black lists created by many different communities or security organizations. In this tab we can see our chosen list resources, with the action to unbound, meaning to enable 'Domain Name' blocking for this Alias, with a frequency of updates every day to keep up with newly added malicious domains.

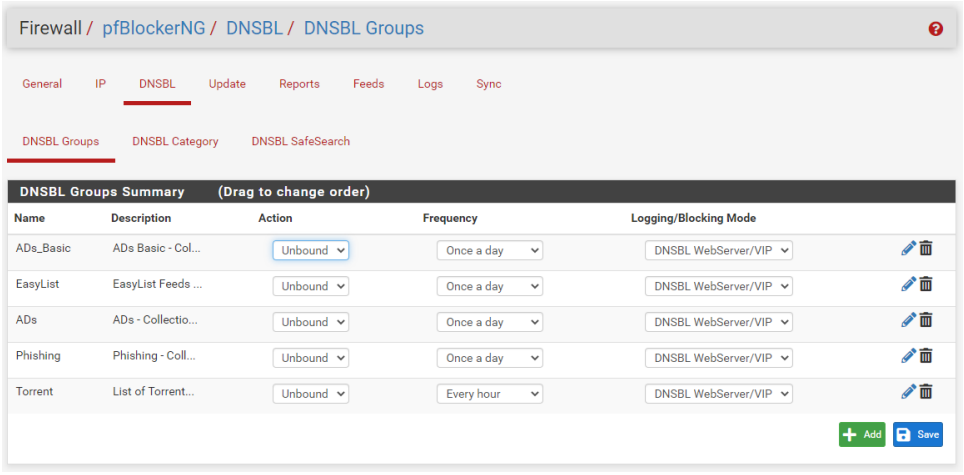

Figure 30 : DNSBL chosen groups

More groups can be added from the pfBlockerNG→Feeds tab, where we can find over a hundred lists, IPv4 and IPv6 , but we will discuss that later on the pfBlockerNG section.

Under DNSBL Category, we got lists organized by categories, we have 2 black listers from which we will take our blacklist categories, Shallalist and Université Toulouse 1 Capitole. We only chose certain categories regarding the security side such as spyware, tracking, ddos, malware, phishing. More can be added in the future.

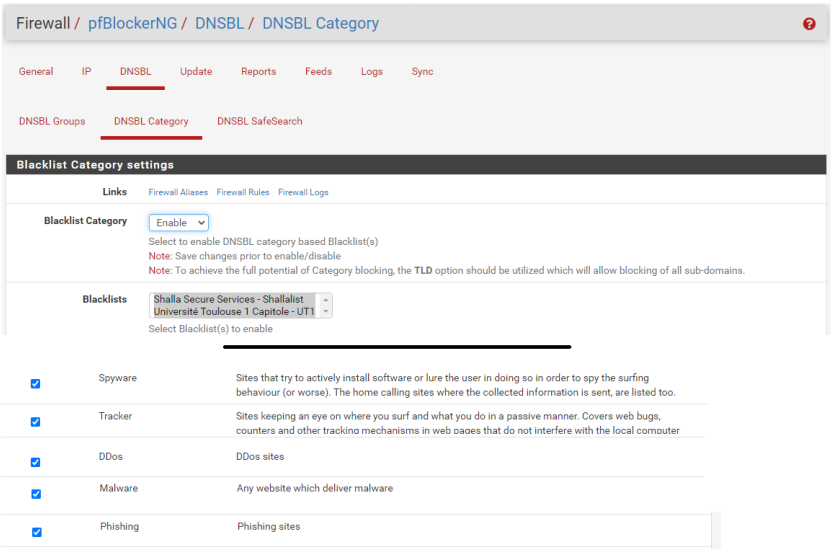

Figure 31 : DNSBL chosen categories

In the Reports→DNSBL Block Stats, we get a lot of bar and pie charts which can help us determine what are the most blocked sources or which lists are mostly being used.

Figure 32 : Available stats for DNSBL

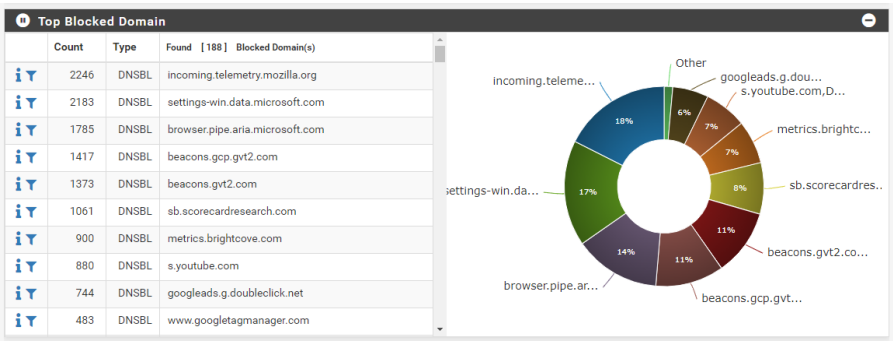

Figure 33 : Example of a pie chart for top blocked domains

1.10 pfb_filter

Is installed with pfBlockerNG-devel package. It's the pfBlockerNG firewall filter service, used to get more options to the basic configuration of the pfSense firewall rules. It allows for:

- Assigning many IP address URL lists from sites like I-blocklist to a single alias and then choose a rule action.
- Blocking countries and IP ranges.
- Replacement of both Country block and IP blocklist by providing the same functionality, and more, in one package.
- Uses native functions of pfSense software instead of file hacks and table manipulation.

This service can be accessed through Firewall→pfBlockerNG. General settings could be left to default. In the IP tab, we're interested in the IPv4 and the GeoIP blocking. Under IPv4 we can add a collection of feeds from the most reputable blocklist providers which can be found in the Feeds tab.

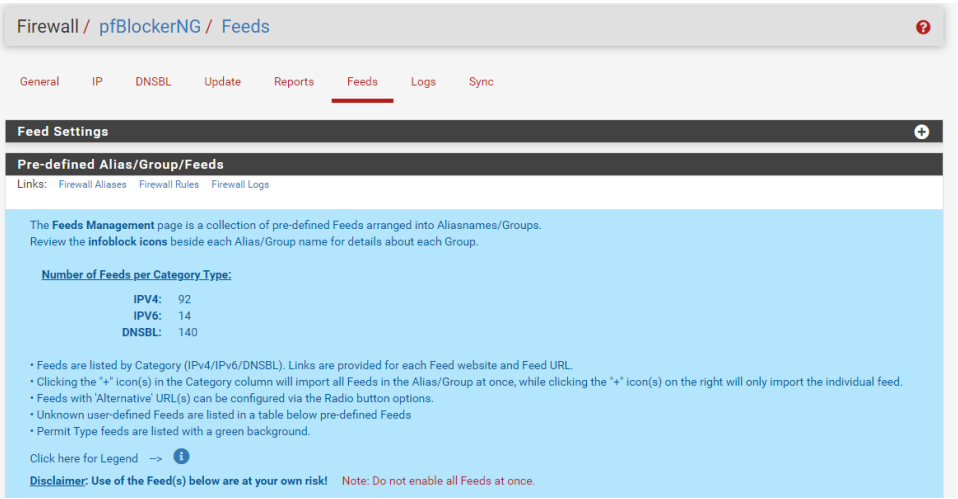

Figure 34 : pfBlockerNG Feeds tab

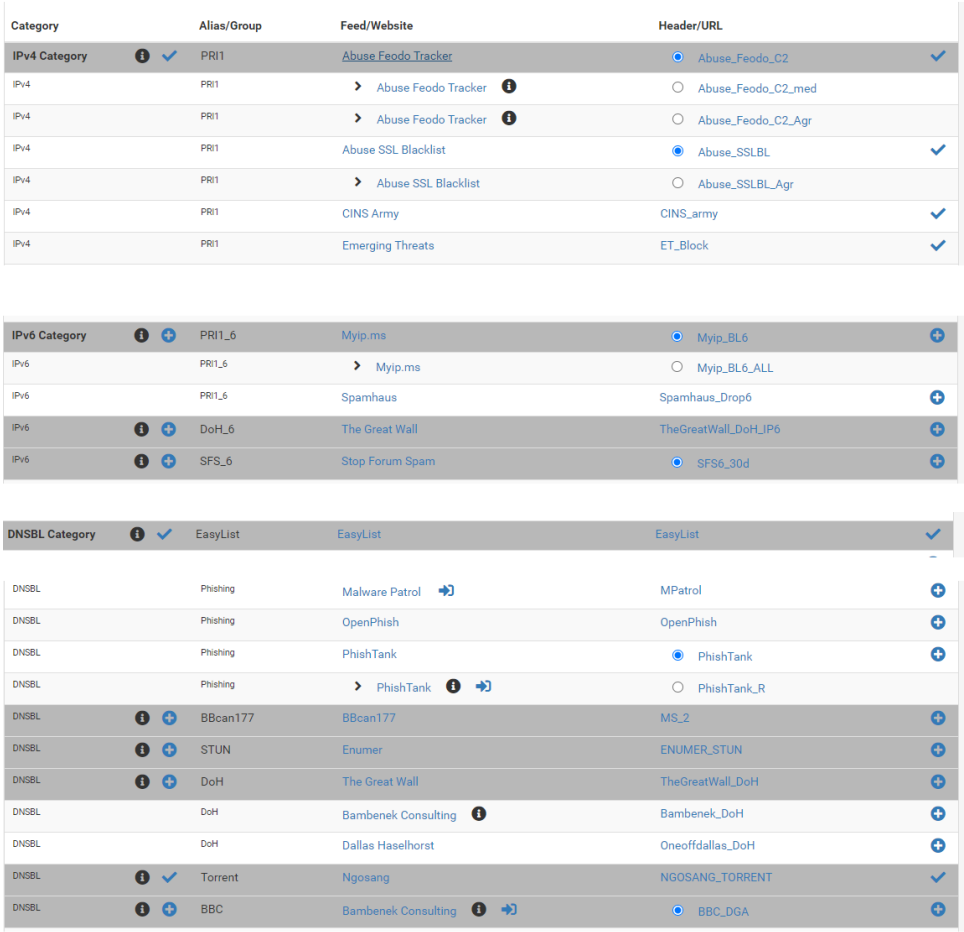

Figure 35 : Examples of blocklists available in the Feeds tab

For the IPv4 blocking, we chose few of the above-mentioned lists, mostly regarding security and torrent connections.

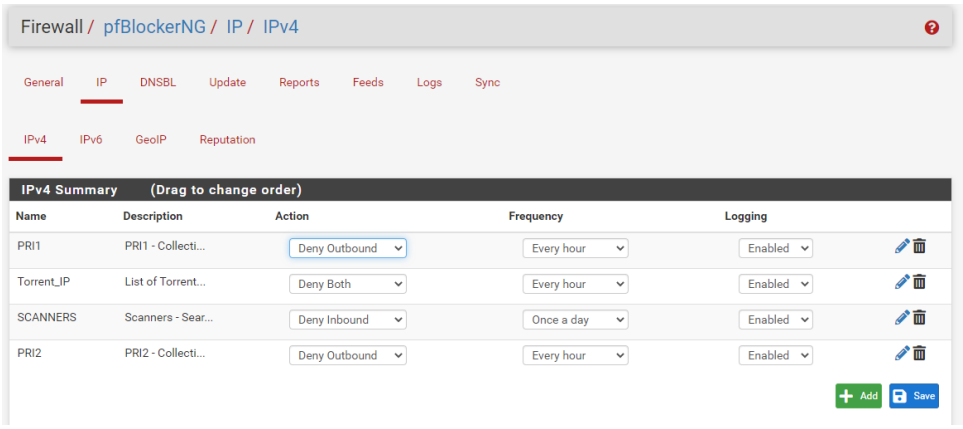

Figure 36 : pfBlockerNG IPv4 chosen lists

Other than the predefined lists, we can add our own blocking choices which are more advanced than the basic blocking rules found in pfSense's firewall.

For example, Facebook.com doesn't have a single IP address but many others, so pfBlockerNG gives the ability to block sites using their Autonomous System Number. An autonomous system is a collection of connected Internet Protocol routing prefixes under the control of one or more network operators on behalf of a single administrative entity or domain that presents a common, clearly defined routing policy to the Internet.

By adding a new IPv4 source definition, we select the ASN format and the source type as ASN and we deny the outbound traffic, and we can choose the update rate to once every day for example.

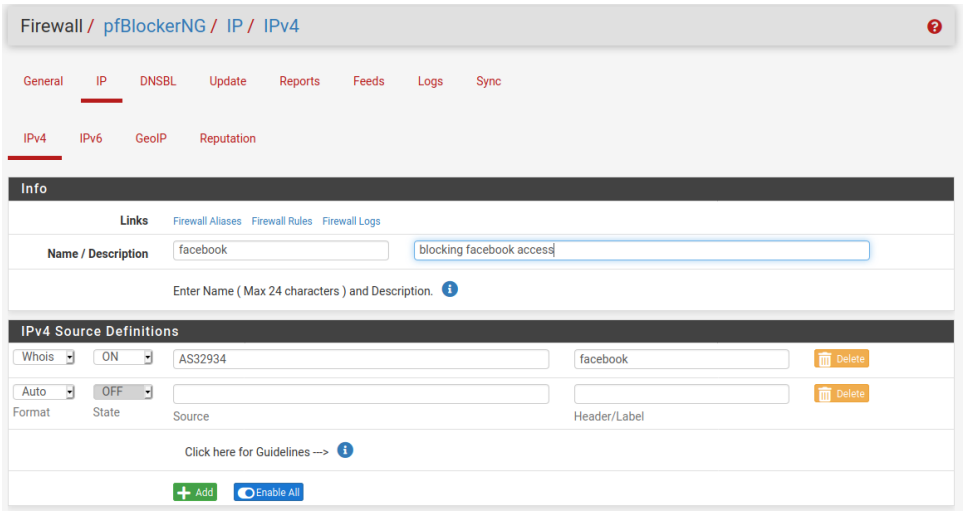

Figure 37 : Blocking using ASN

For the GeoIP we just selected the Top Spammers list with the Deny inbound action.

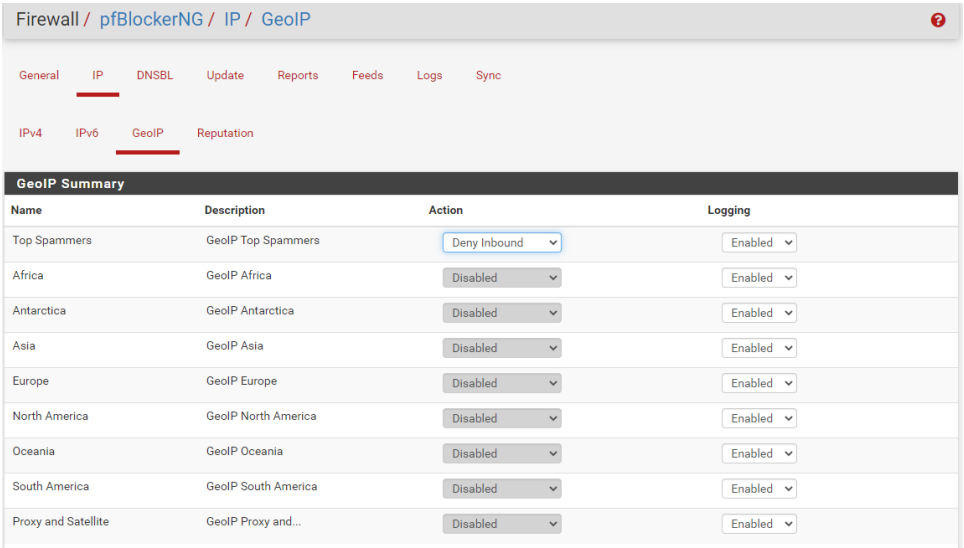

Figure 38 : pfBlockerNG GeoIP

Similar to DNSBL reports, we can find stats for IP addresses blocked, permitted or matched.

These stats are of course based off alerts generated by pfBlockerNG, that can be seen in the Reports→Alerts tab.

Figure 39 : Available stats for IP blocks

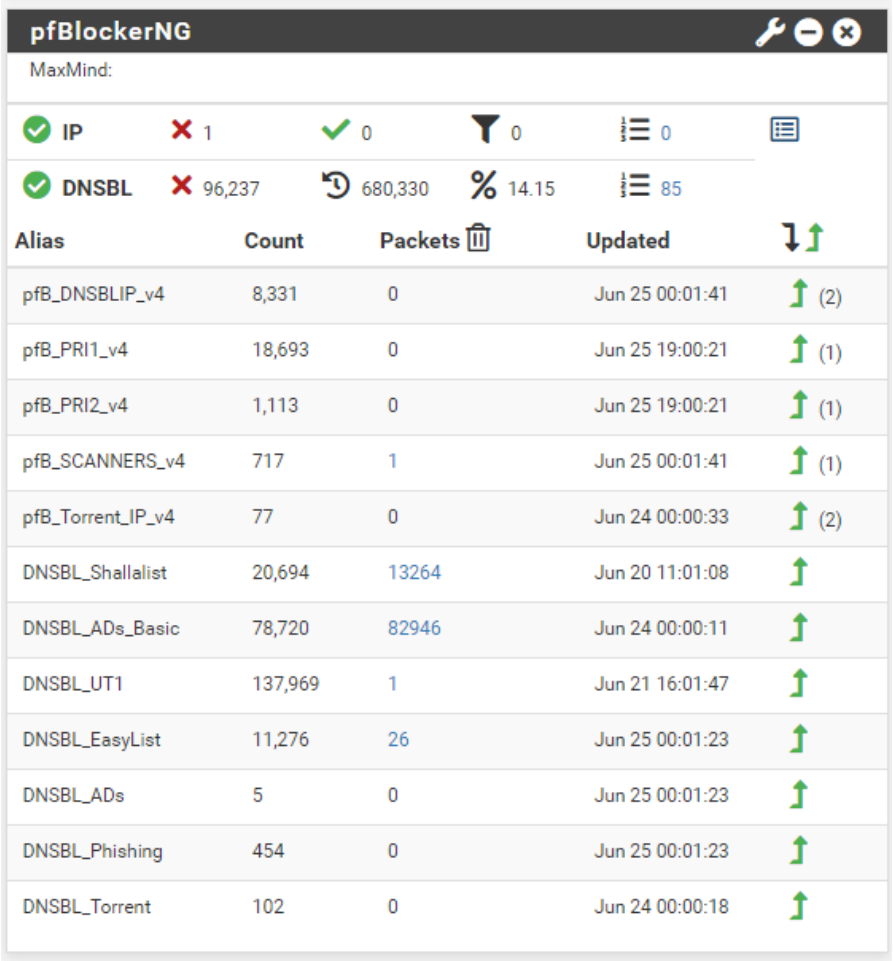

Figure 40 : pfBlockerNG Dashboard widget

1.11 Snort

Is installed with the Snort package. Snort is an intrusion detection and prevention system. It can be configured to simply log detected network events to both log and block them. Thanks to OpenAppID detectors and rules, Snort package enables application detection and filtering.

Snort operates using detection signatures called rules. Snort rules can be custom created by the user, or any of several pre-packaged rule sets can be enabled and downloaded. The service can be found under Services→Snort. It is enabled and set to inspect traffic on the WAN interface.

The Snort package currently offers support for these pre-packaged rules:

- Snort VRT (Vulnerability Research Team) rules
- Snort GPLv2 Community Rules
- Emerging Threats Open Rules
- Emerging Threats Pro Rules
- OpenAppID Open detectors and rules for application detection

We got all these packages selected, but not all rules are. After a trial period we got to decided which rules to keep on and which rules to disable. These rulesets are set to update every 24h to keep up with the changes.

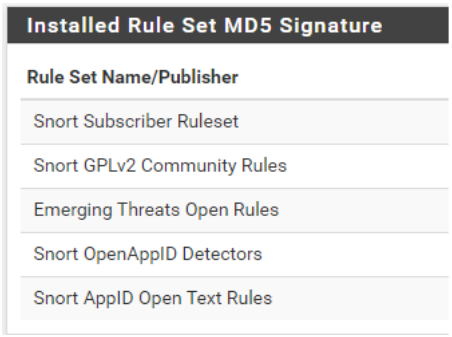

Figure 41: Installed Snort rulesets

The Alerts tab is where alerts generated by Snort may be viewed. If Snort is running on more than one interface, we can choose the interface to view alerts for in the drop-down selector. We use the DOWNLOAD button to download a gzip tar file containing all of the logged alerts to a local machine. The CLEAR button is used to erase the current alerts log.

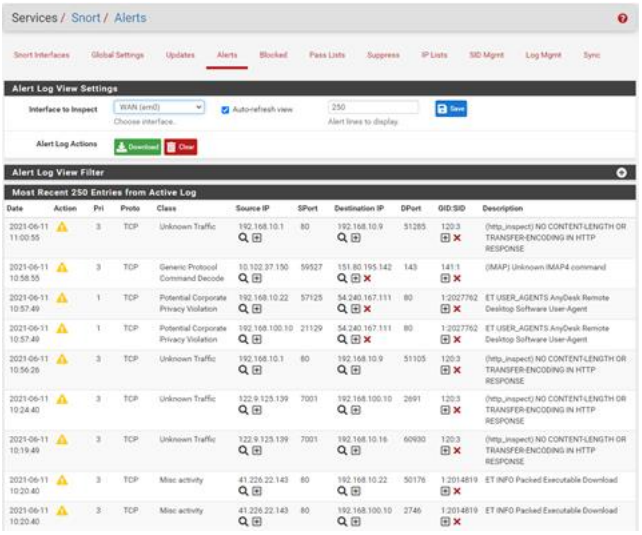

Figure 42: Snort alerts tab

Even with this setup, Snort should not be overlooked, some rules can have negative impact as they generate false positive alerts, that can spam our alerts table or wrongfully block offenders, that's where suppress lists come to a point. We can add preconfigured suppress lists from community members, or we can slowly grow ours.

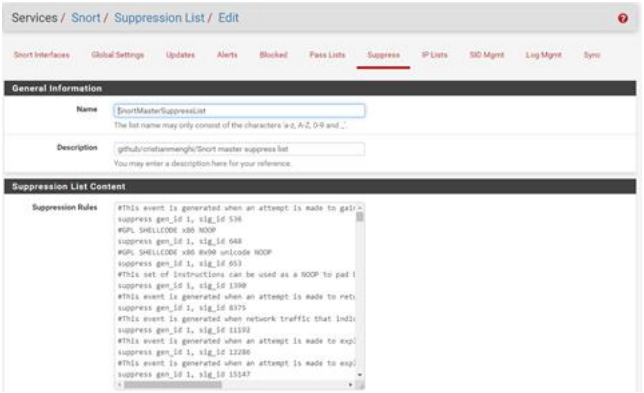

Figure 43: Configured Snort suppressions lists

IP Lists tab is to manage the IP lists files for the IP Reputation preprocessor. IP lists are text-format files containing one IP address or network (expressed in CIDR notation) per line.

We can upload an IP list file to the firewall, by opening the file upload dialog, then browsing to the file on the local machine.

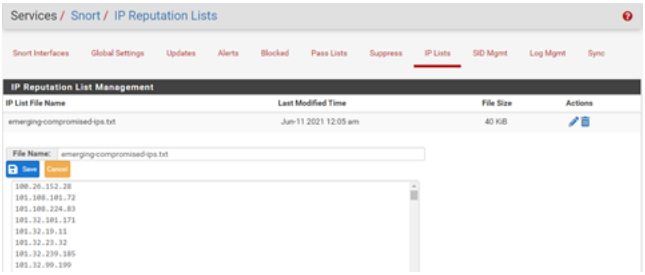

Figure 44: Snort IP reputation lists

1.12 Squid

Is installed with the Squid package. Squid is a caching proxy for the Web supporting HTTP, HTTPS, FTP, and more. It reduces bandwidth and improves response times by caching and reusing frequently-requested web pages.

We chose to run the proxy server as a logger of the users' activity inside the company's network with a little bit of security involved. We opted to run the proxy in transparent mode enabling SSL interception for HTTPS packets. The settings of Squid are under Services→Squid Proxy Server

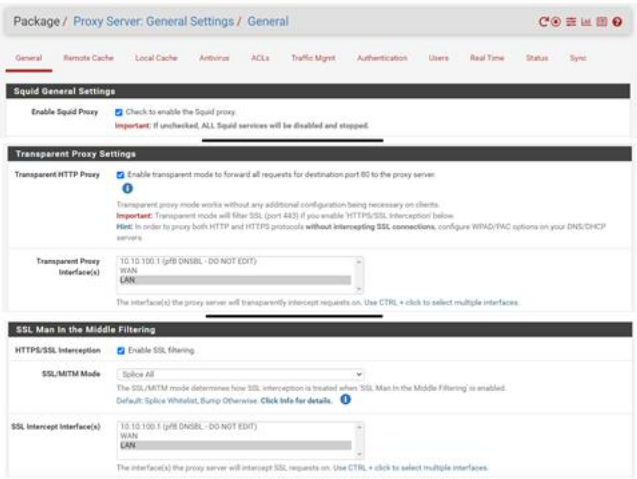

Figure 45: Squid proxy essentials settings

Caching settings are found in the "remote cache" and "local cache" tabs, we are only interested in local caching for the proxy server. Caching replacement policies and the hard disk caching system are left as default.

The real time tab follows multiple tracking table, consisting of Squid Access Table, Squid Cache Table, SquidGuard Table, C-ICAP Virus Table, C-ICAP Access Table, C-ICAP Server Table, freshclam Table and clamd Table.

1.13 squidGuard

Is installed with the squidGuard package. SquidGuard is a URL redirector used to use blacklists with the proxy software Squid. There are two big advantages to squidguard: it is fast and it is free. SquidGuard is published under GNU Public License.

Adding squidGuard on top of the previously discussed pfBlockerNG can be considered unnecessary and an overkill, but we are adding it for educational reasons, and a little more security option in case the others fail cannot hurt.

The configuration tab can be found under Services→SquidGuard Proxy Filter.

The general settings are left to default, while we chose the appropriate target rules from the CommonACL tab that SquidGuard will block from, similarity to pfb_dnsbl.

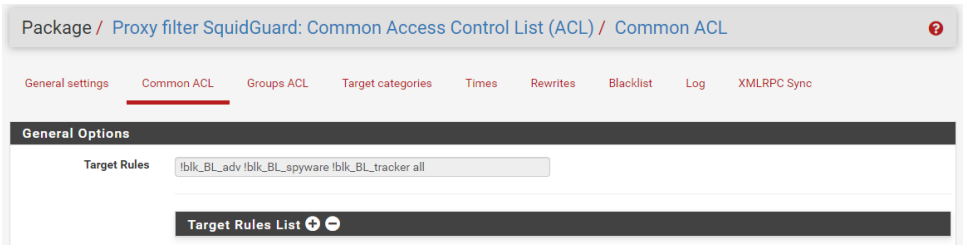

Figure 46 : SquidGuard blocking target rules

This target list was added and downloaded through the Blocklist tab, where we enter a URL of a blocklist delivered by known black lister like MESD blacklists, Shalla's Blacklists, Université Toulouse blacklist collection, URLBlacklist.com.

1.14 syslogd

Is installed by default. The syslogd daemon reads and logs messages to the system console, log files, and other machines as specified by its configuration file. The daemon reads its configuration file when it starts up and whenever it receives a hang-up signal. The Remote Logging options under Status → System Logs on the Settings tab allow syslog to copy log entries to a remote server.

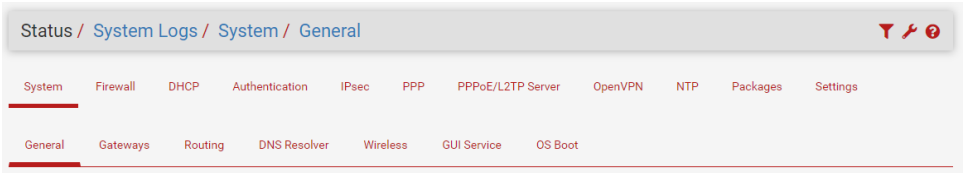

Figure 47: Available tabs by the Syslogger

1.15 unbound

Is installed by default. The DNS Resolver in pfSense utilizes unbound, which is a validating, recursive, caching DNS resolver that supports DNSSEC and a wide variety of options.

By default, the DNS Resolver queries the root DNS servers directly and does not use DNS servers configured under System → General Setup or those obtained automatically from a dynamic WAN.

By contacting the roots directly, it eliminates many issues typically encountered by users with incorrect local DNS configurations, and the DNS results are more trustworthy and verifiable with Domain Name System Security Extensions (DNSSEC).

This behavior may be changed, however, using the DNS Query Forwarding option, and that is what we chose to do by using Quad9 DNS server which does support DNSSEC and is largest of the three global public recursive DNS resolvers which protect users from malware and phishing, and is notable for consistently being found to be by far the most effective at doing so in independent evaluations.

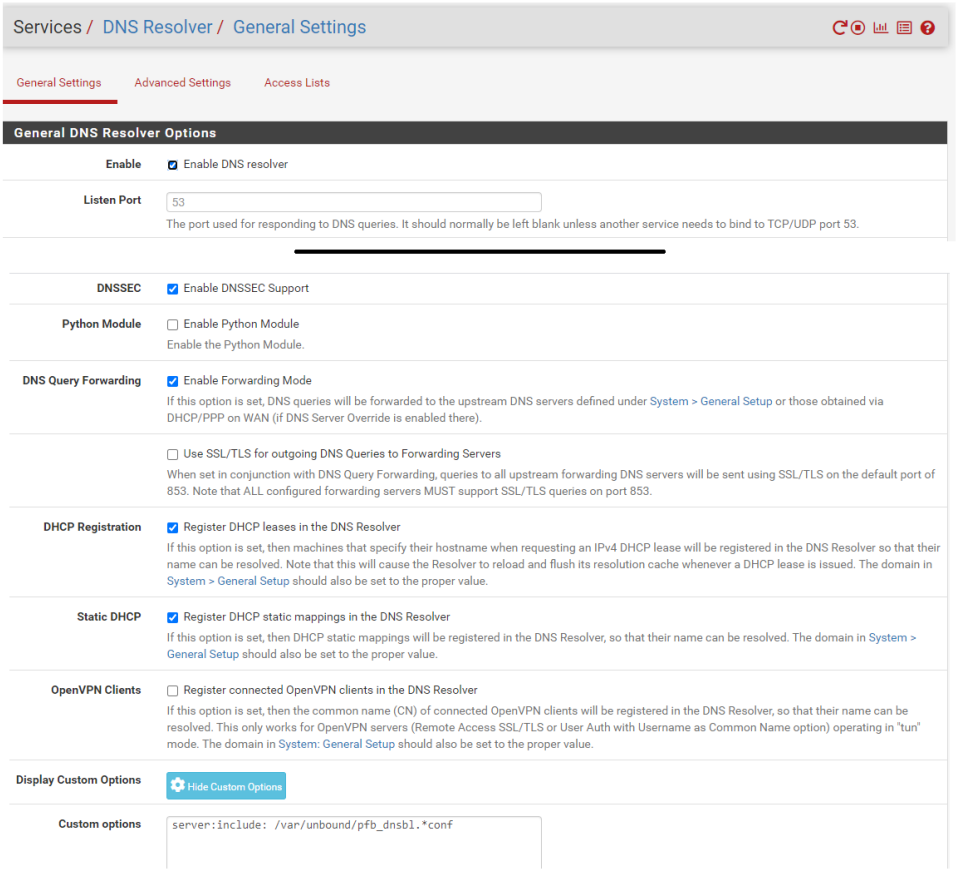

Figure 48 : DNS Resolver essential configuration

Using the custom options setting, we added an entry to allow pfb_dnsbl to intervien into the dns queries to be able to filter them.

1.16 Zabbix_agentd

Is installed with the Zabbix-agent5 package. Zabbix agent is deployed on a monitoring target to actively monitor local resources and applications (hard drives, memory, processor statistics etc). The agent gathers operational information locally and reports data to Zabbix server for further processing. In case of failures (such as a hard disk running full or a crashed service process), Zabbix server can actively alert the administrators of the particular machine that reported the failure.

The Zabbix agent configuration page is located under Services→Zabbix agent tab.

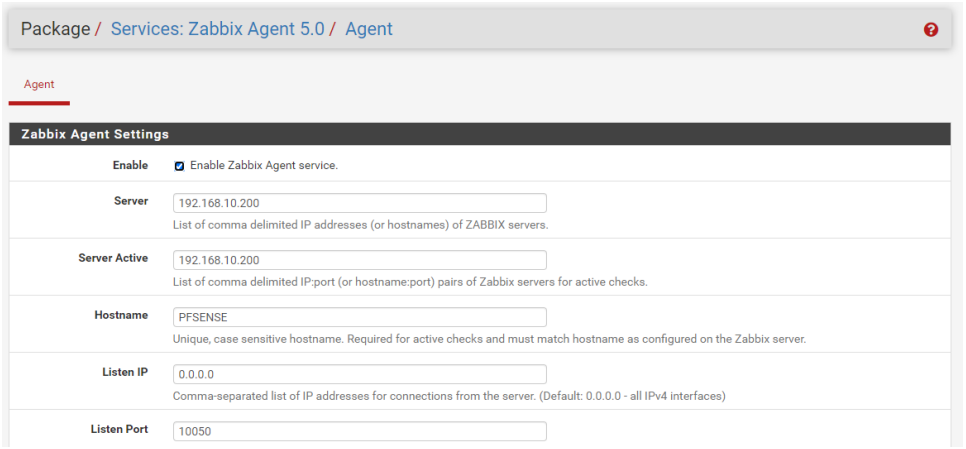

Figure 49 : Zabbix agent essentials configuration

We simply added the IP address and the listening port of our preconfigured Zabbix server, more on that to come.

1.17 NMAP tool

Is installed with the Nmap package. Is a powerful network scanner that provides port scanning, OS and service identification and more.

NAMP will be available at Diagnostics → Nmap as well as in the shell (SSH or Console). The Nmap package also installs an OUI database that is used by the pfSense web GUI to display manufacturer names on pages that list MAC addresses, such as the ARP Table, and DHCP Leases.

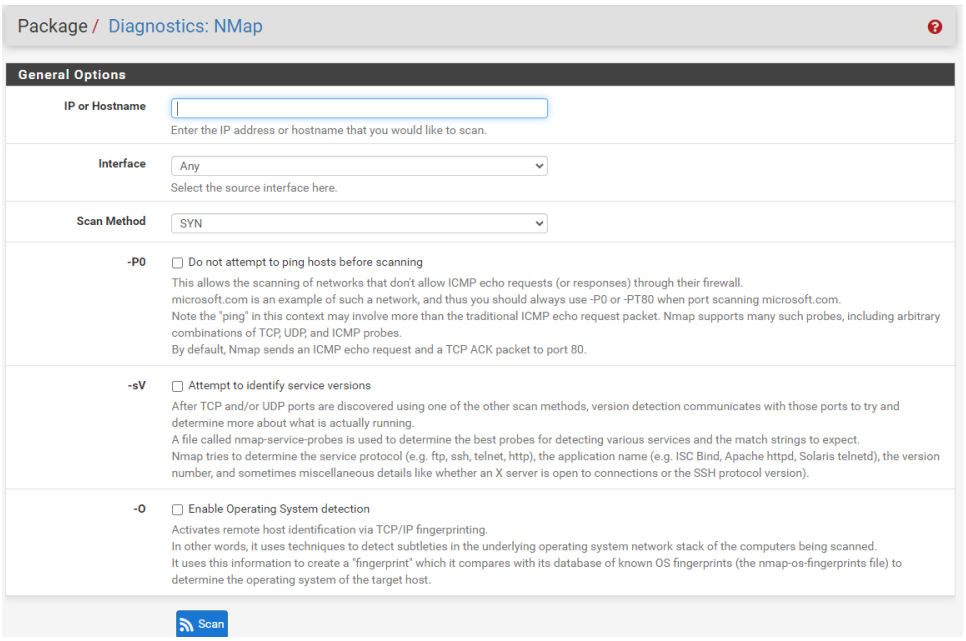

Figure 50 : Nmap interface

1.18 Web server port forwarding

To give the public access to the company's web server, which is fully protected in our local network, we're going to use port forwarding. Starting up from out ISP router, any incoming traffic from the internet on a certain port via TCP, will be redirected to pfSense.

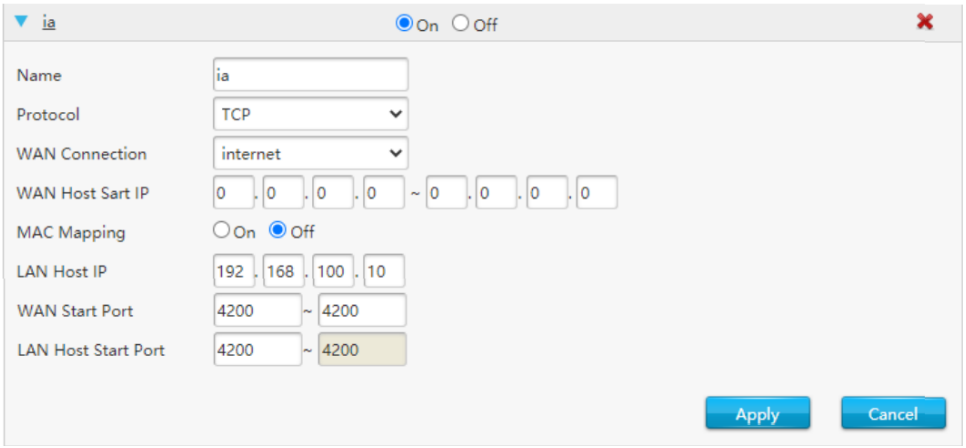

Figure 51: ISP router port forwarding configuration for web server

Now moving to the pfSense side, any incoming traffic from the pfSense WAN interface on port 4200 via TCP will be forwarded to the web server with the local IP address 192.168.10.22 on the respective port 4200. A firewall rule must be added to accept this incoming traffic from the outside.

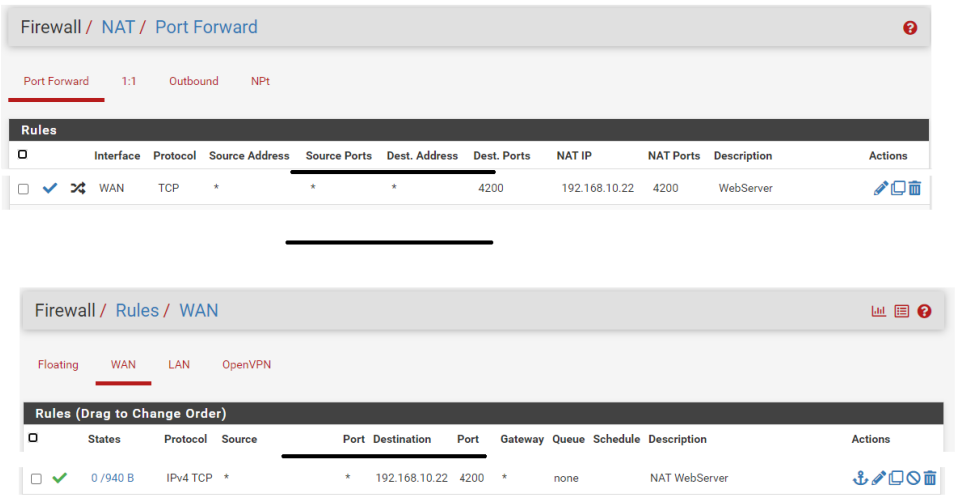

Figure 52: pfSense port forwarding configuration and firewall rule for web server

2 Zabbix

Zabbix works in a Client/Server model. The server communicates to the native software agents available for various Operating systems like Linux, NIX, and Windows. For systems without an agent, generic monitoring protocols such as Simple Network Management Protocol (SNMP) or Intelligent Platform Management Interface (IPMI) can be used.

We installed Zabbix 5.0 on our Ubuntu 20.04 LTS virtual machine, running in ESXi 6.7 on the same host as our pfSense machine.

Zabbix Server depends on the following software applications:

- Apache web server
- PHP with required extensions
- MySQL/ MariaDB database server

MySQL or MariaDB can be a remote server, but php and httpd need to be installed on the Zabbix server. Everything is installed locally.

We gave the Zabbix server a static IP address of 192.168.10.200, the web GUI can be access from the local network from “http://(Zabbix server’s hostname or IP address)/zabbix/”

First thing after the server installation, we get welcomed with a setup page to give the proper information and configuration of the server.

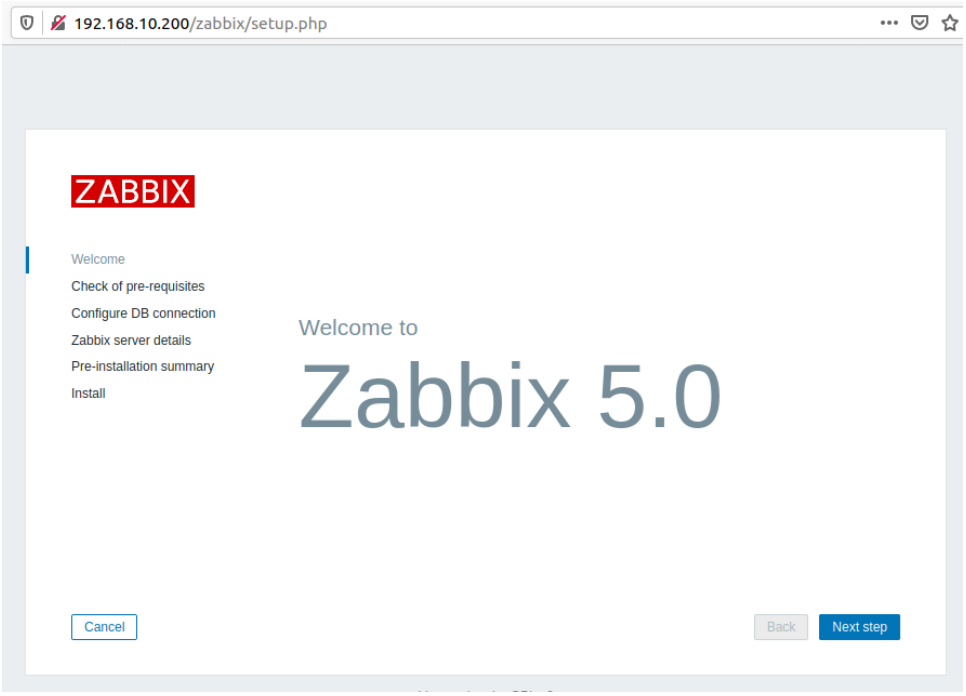

Figure 53: Preinstallation setup page of Zabbix

After that configuration, we are moved to the Zabbix index page, which defaults to the Dashboard.

The side bar contains multiple tabs for different pages, here is a cut off picture showing them.

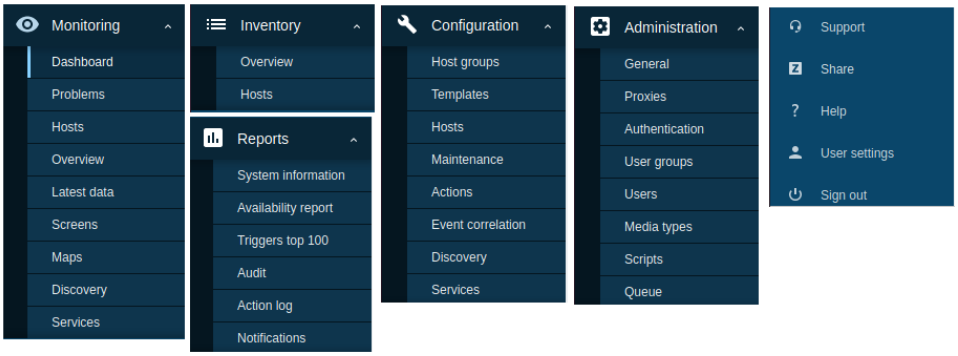

Figure 54 : Zabbix server side bar tabs

We are not going to discuss every single tab, Zabbix documentation is very detailed and informative on everything, we are only showing the essentials and our most used tabs.

First thing is to add a host to monitor, which is done by accessing Configuration→Hosts→Create Host from the top right corner. The host should have a unique name and be added to at least one host group. A new group can be created and linked to the host group by adding a non-existing group name. Also, the IP address of the host with the TCP/UDP port number, default values are: 10050 for Zabbix agent, 161 for SNMP agent, 12345 for JMX and 623 for IPMI.

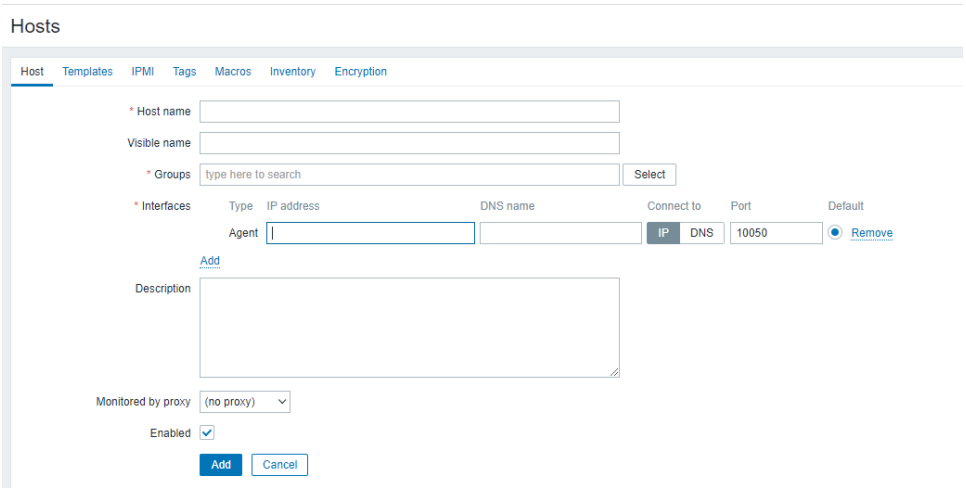

Figure 55: Adding a host to Zabbix server

After the necessary setting for a connection between the Zabbix server and our new host, we can add a predefined template. A template is a set of entities that can be conveniently applied to multiple hosts.

The entities may be:

- items
- triggers

- graphs
- applications
- dashboards
- low-level discovery rules
- web scenarios

As many hosts in real life are identical or fairly similar so it naturally follows that the set of entities (items, triggers, graphs…) we have created for one host, may be useful for many. Of course, we could copy them to each new host, but that would be a lot of manual work. Instead, with templates we can copy them to one template and then apply the template to as many hosts as needed.

When a template is linked to a host, all entities (items, triggers, graphs…) of the template are added to the host. Templates are assigned to each individual host directly (and not to a host group).

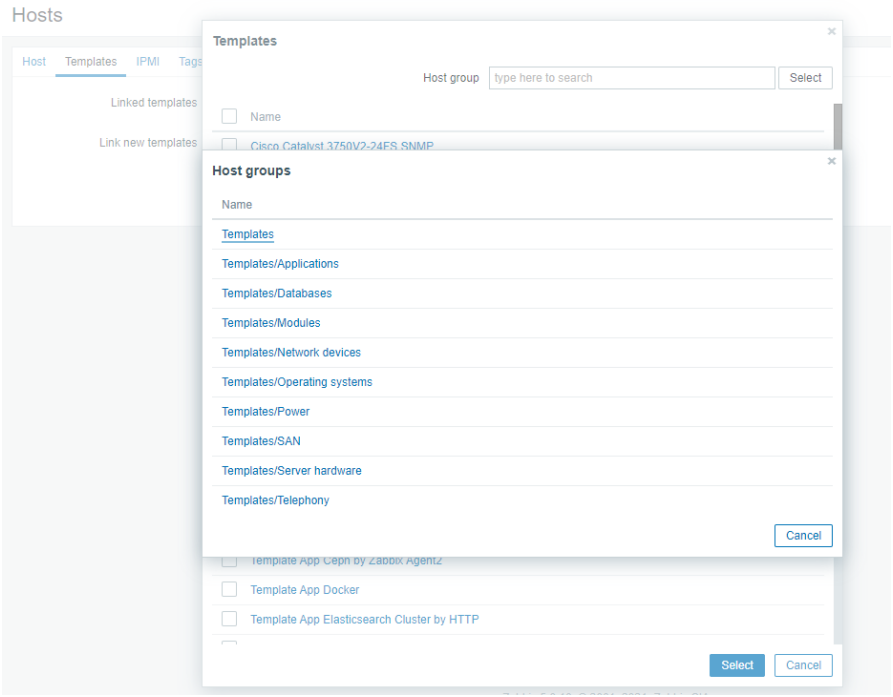

Figure 56 : List of predefined templates when adding new host

Once our new host is added, it now appears in the Zabbix dashboard and hosts tabs. To add more monitoring options, we can head to Configuration→Hosts where we'll select the desired host we want to customize more. There we can add more applications, items, triggers, graphs, discovery rules and web scenarios, other than the ones found in the selected predefined template we chose in the creation of the host.

In our case, we want to monitor certain ports of our web server, so we start off by heading to Items→Create Item from the top right. In an individual item we specify what sort of data will be gathered from the host using what Zabbix calls a key, and an update interval. We should add the item to an application which might contain other items. Here we created a new application called Ports.

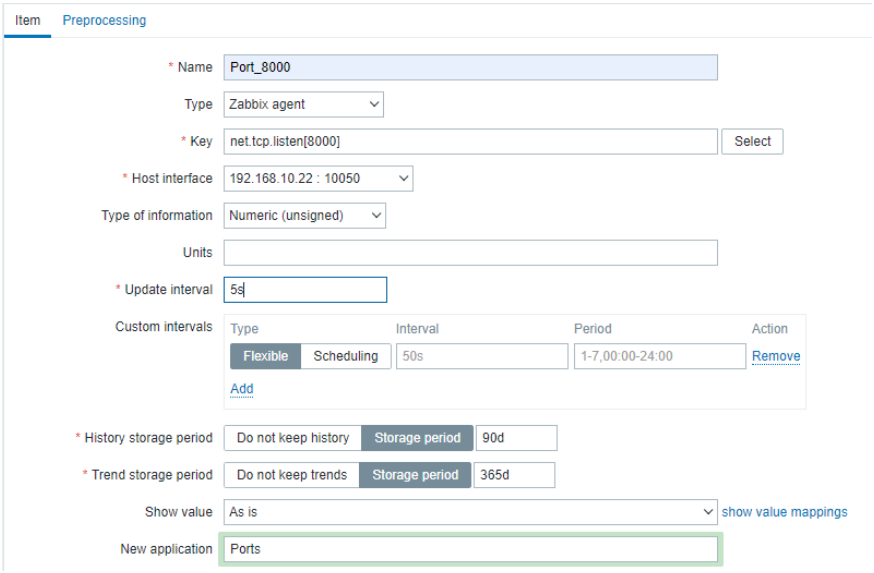

Figure 57: Adding new Item

Graphs of the items data can be visualized, we will create a graph representing the availability of our newly created items that follow our web server's selected ports.

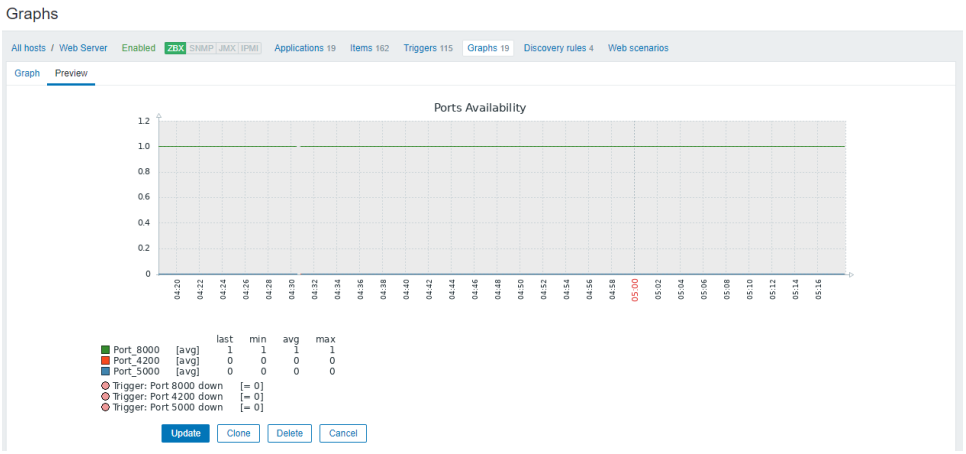

Figure 58: Graph example of our custom items

We can manually check the incoming data of this newly added item from the Monitoring tab→Latest data. By specifying which host/hosts we want to take data from and which application are the items in. For our item, based on its key, if the latest value is 1 that means the port is up, if the value is 0 the port is down.

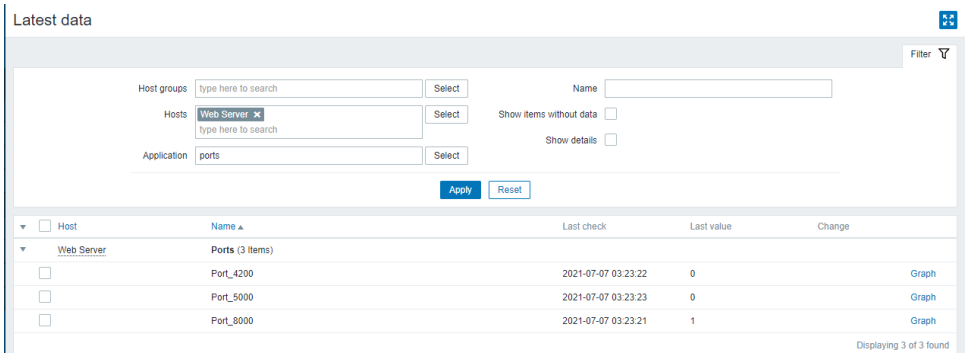

Figure 59 : Monitoring latest data output

If we want Zabbix to automatically inform us if a port goes down, we need a add a new trigger. After adding a name to our trigger, we need to select its severity ranging from Not Classified to Disaster, and the function that this trigger will run on. The function depends on an existing item, followed by some parameters.

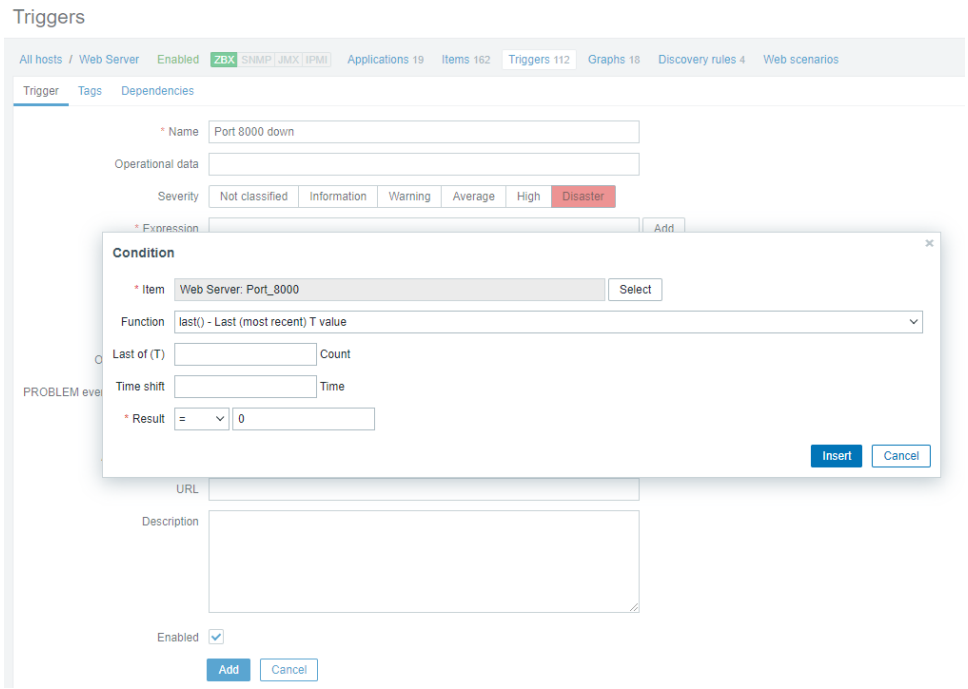

Figure 60 : Adding a trigger

Now once Zabbix detects that a port is down, it will automatically show in the dashboard/problems feed. Triggers can be disabled so they do not show up in the feed, if our host is in a maintenance mode or if we don't want to use some of the triggers found in the predefined template.

Zabbix gives us the opportunity to further keep us updated, by sending the system administrator a notification via a large set of media types, such as Emails, SMS and Webhooks (Jira, Slack, Discord...). We chose to use emails for now.

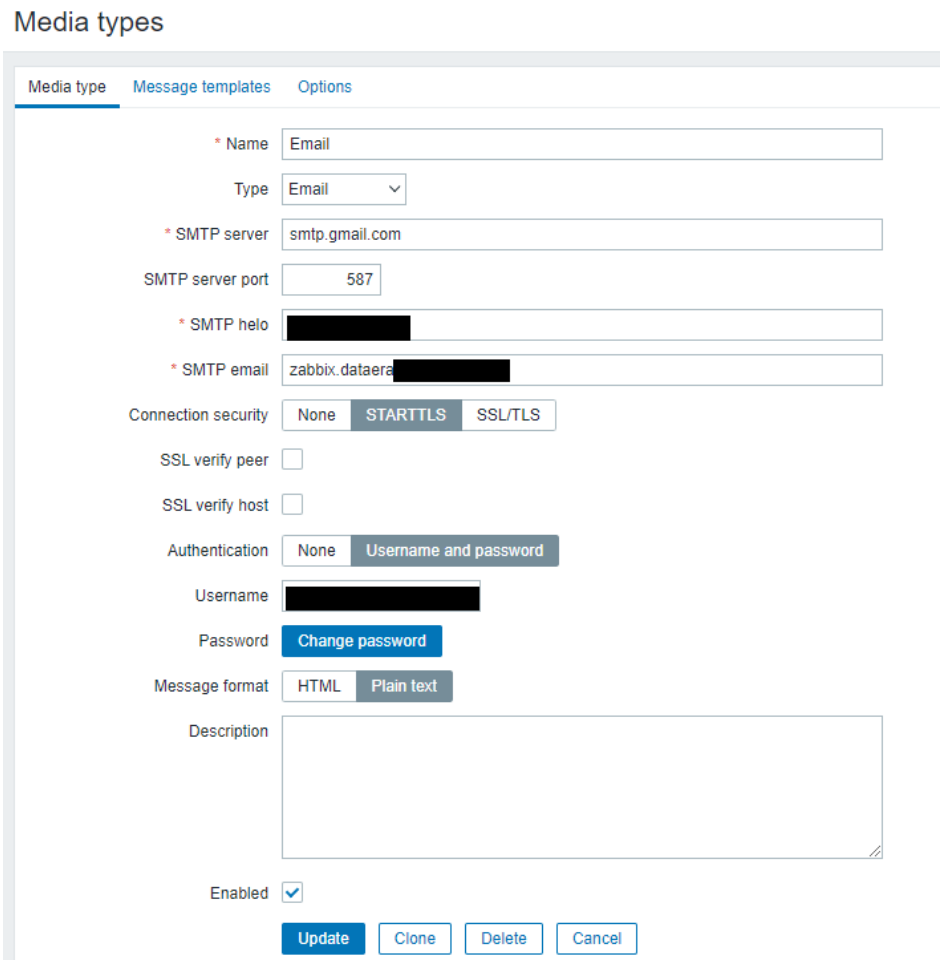

Figure 61: Zabbix email media type config

After that, we need to go to Configuration→Actions tab to enable Zabbix to send notifications to the administrators via the selected media types. Zabbix can send each and every alert generated if we want to with the predefined “Report problems to Zabbix administrators” action, but we can limit it to only certain alerts. We’re going to add the previously created trigger to be send by email in case it happens to a new action called “Port down email notification”.

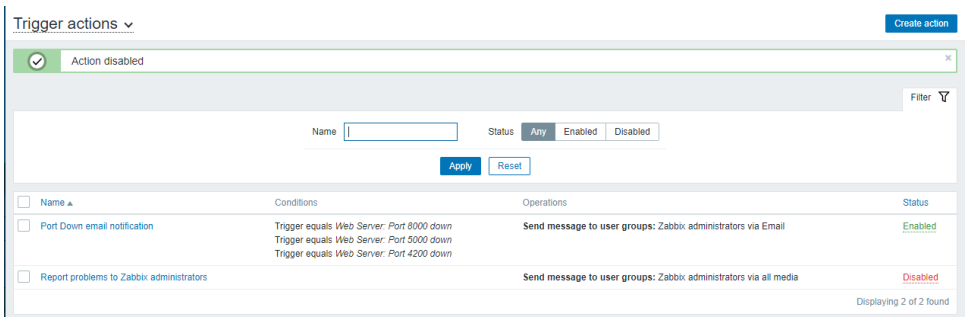

Figure 62: Trigger actions

Finally, we need to add the appropriate email to the Zabbix user admin or admins group so they receive the emails of the generated triggers. We edited the administrator user settings, selected the Email media type and chose which severity will Zabbix send to us.

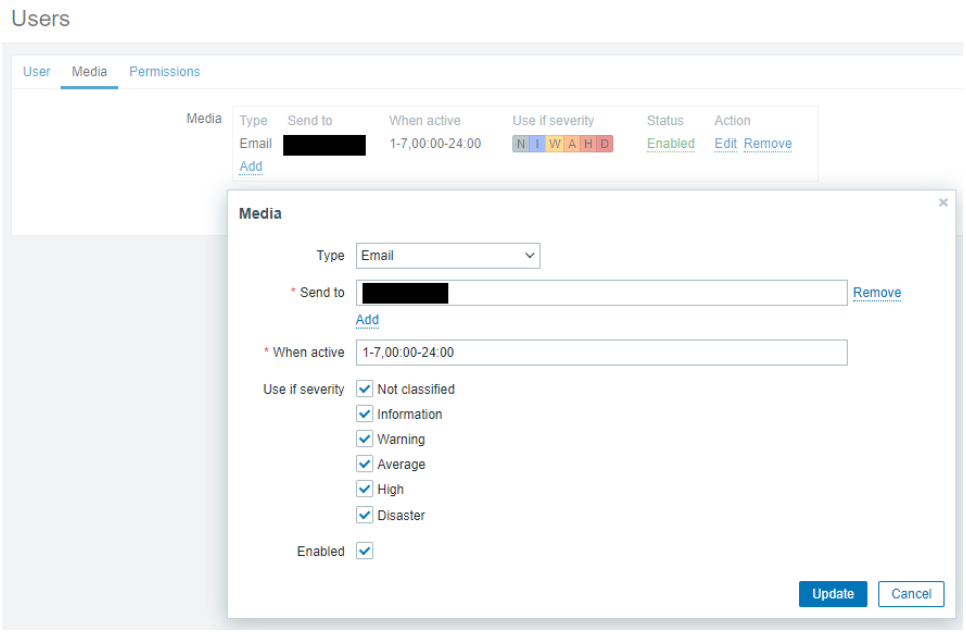

Figure 63: User email notification settings

These steps showcase the most used use case scenario for Zabbix server. Next, we'll discuss briefly some other unmentioned used tabs.

2.1 Monitoring

Zabbix Dashboard is a central place in the web frontend and provides high-level personalized details about the monitored environment:

- Favorite maps
- Favorite graphs
- Favorite screens
- Last 20 issues
- System status
- Host status
- Status of Zabbix server
- Discovery status
- Web monitoring

In Monitoring → Problems we can see what problems we currently have. Problems are those triggers that are in the "Problem" state.

The Monitoring → Hosts section gives a neat overview of monitored hosts with detailed information about host interface, availability, tags, current problems, status (enabled/disabled), and links to easily navigate to the host's latest data, problem history, graphs, dashboards and web scenarios.

The section in Monitoring → Latest data can be used to view latest values gathered by items as well as to access various graphs for the items. Items are displayed with their name, last check time, last value, change amount, tags and a link to a simple graph/history of item values.

In the Monitoring → Discovery section results of network discovery are shown. Discovered devices are sorted by the discovery rule. With nothing selected in the filter, all enabled discovery rules are displayed. To select a specific discovery rule for display, start typing its name in the filter. All matching enabled discovery rules will be listed for selection. More than one discovery rule can be selected.

2.2 Inventory

Here we can keep the inventory of manually chosen networked devices in Zabbix.

There is a special Inventory menu in the Zabbix frontend. However, we will not see any data there initially and it is not where we enter data. Building inventory data is done manually when configuring a host or automatically by using some automatic population options. We are not using this tab seeing that we don’t have a big number of hosts monitored by Zabbix.

2.3 Reports

The Reports menu features several sections that contain a variety of predefined and user-customizable reports focused on displaying an overview of such parameters as system information, triggers and gathered data.

In Reports → System information a summary of key Zabbix system data is displayed.

In Reports → Availability report we can see what proportion of time each trigger has been in problem/ok state. The percentage of time for each state is displayed.

Thus, it is easy to determine the availability situation of various elements on our system

In the Reports → Audit section users can view records of changes made in the frontend.

2.4 Configuration

The Configuration menu contains sections for setting up major Zabbix functions, such as hosts and host groups, data gathering, data thresholds, sending problem notifications, creating data visualization and others.

In the Configuration → Host groups section users can maintain host groups. A host group can contain both templates and hosts. A listing of existing host groups with their details is displayed. We can search and filter host groups by name.

In the Configuration → Templates section users can maintain templates. A listing of existing templates with their details is displayed.

2.5 Administration

The Administration menu is for administrative functions of Zabbix. This menu is available to users of Super Administrators type only.

in the Administration → Scripts section user-defined global scripts found on the Zabbix server can be configured and maintained. Each script can be applied to different hosts as needed. Depending on the set user permission, the scripts are available for execution by clicking on the host in various frontend locations (Dashboard, Problems, Latest data, Maps) and can also be run as an action operation.

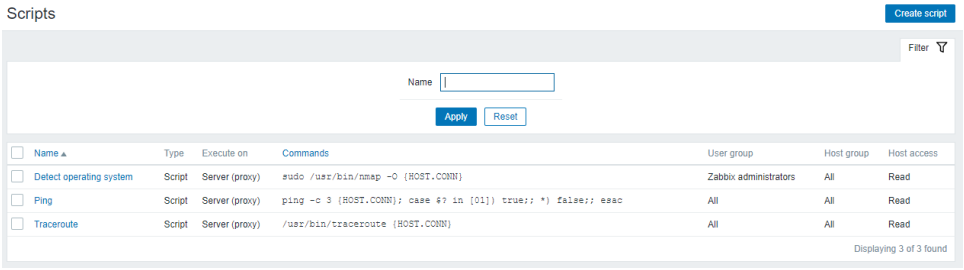

Figure 64: Available Zabbix scripts

3 FreeIPA

FreeIPA is a free and open-source identity management system for centrally managed users and computers in your network. We installed the server on a 16.04 LTS Linux machine, running with the previous server on ESXi 6.7 host.

PLEASE NOTE: Due to the Coronavirus outbreak, DataEra is currently working with BYOD (Bring Your Own Device) which is an approach allowing to use one's personally owned device, rather than being required to use an officially provided device, to make working from home easier. For that, we are only going to use FreeIPA as a users repository, no hosts will be added.

The index page of FreeIPA defaults to the users page, which is the main use of our FreeIPA server.

Personal information of the users has been blocked due to privacy reasons.

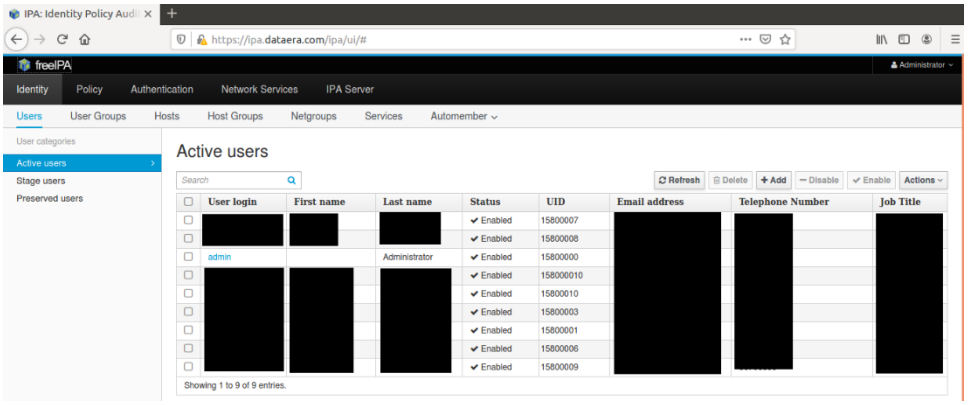

Figure 65 : FreeIPA users tab

To add a new user, we simply click on the +Add button and a new form window appears asking for primary information about the user.

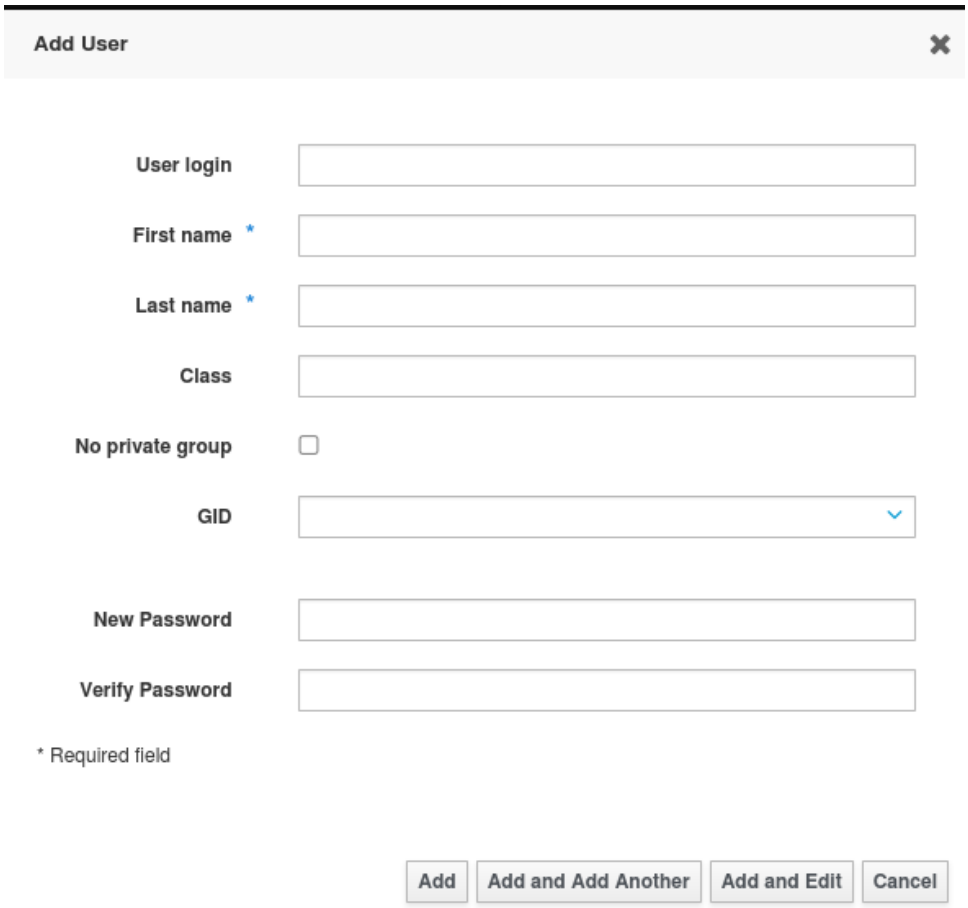

Figure 66 : FreeIPA add user interface

Once the user has been added, more personal information can be added by clicking on the user login from the users list. The user can be added then to any existing users' groups.

Future plans with FreeIPA:

Right now, this is the extend of our use of the FreeIPA server, but for future use, we can add more restraints by using another functionality of an LDAP server, configuring Host-Based access control.

FreeIPA can control access to both machines and the services on those machines within the FreeIPA domain. The rules define who can access what within the domain, not the level of access (which are defined by system or application settings). These access control rules grant access, with all other users and hosts implicitly denied.

This is called host-based access control because the rule defines what hosts (source) are allowed to access other hosts (targets) within the domain. This access can be further broken down to users and services.

Conclusion:

In this chapter we showcased the essential configuration that have been made to our three servers to give the best security possible and that meets the company's demands. Again, we couldn't show everything that has been setup but we made sure we included most of the work that has been done.

For the next chapter, we will briefly show the environment that the servers are running on.

Chapter 4: Virtualization of the Servers

Introduction:

In this final chapter, we will be showing what our new integrated servers are running on. We won't be showing every single aspect of this virtualization process because this project doesn't focus on it but we can't just ignore it as it is essential for the existence of our servers.

VMware vSphere is a software suite that includes components like ESXi, vCenter Server, vSphere Client, vCenter Orchestrator, vSphere Update Manager, etc. vSphere components provide virtualization, management, resource optimization and many other features useful for a virtual environment. vSphere is used to virtualize and aggregate the underlying physical hardware resources and to provide the pool of virtual resources to the data center. It also supports some advanced virtualization features such as disaster recovery, high availability, fault tolerance, dynamic resource allocation, etc.

People new to the VMware's virtualization platform sometimes get confused in dealing with vSphere and its components. Remember that vSphere is a suite of products, just like Microsoft Office (a suite of office products such as Word, Excel, Access), and not a single product that you can install in your environment.

Here is a list and description of the most important components included in the vSphere product suite:

- ESXi – a type 1 hypervisor. A hypervisor is a piece of software that creates and runs virtual machines. In vSphere, virtual machines are installed on ESXi servers.
- vCenter Server – a centralized management platform and framework that lets you manage virtual machines and ESXi hosts centrally.
- vSphere Update Manager – an add-on package for vCenter Server that helps you keep your ESXi hosts and VMs patched with the latest updates.
- vSphere Web Client – a web-based user interface used for managing a virtual infrastructure.
- vSphere Client – a locally installed Windows application with a graphical user interface (GUI) for all day-to-day management tasks and for the advanced configuration of a virtual infrastructure.[25]

1 ESXi

VMware ESXi is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers. As a type-1 hypervisor, ESXi is not a software application that is installed on an operating system; instead, it includes and integrates vital OS components

ESXi provides a virtualization layer that abstracts the CPU, storage, memory and networking resources of the physical host into multiple virtual machines. That means that applications running in virtual machines can access these resources without direct access to the underlying hardware. VMware refers to the hypervisor used by VMware ESXi as vmkernel. vmkernel receives requests from virtual machines for resources and presents the requests to the physical hardware.

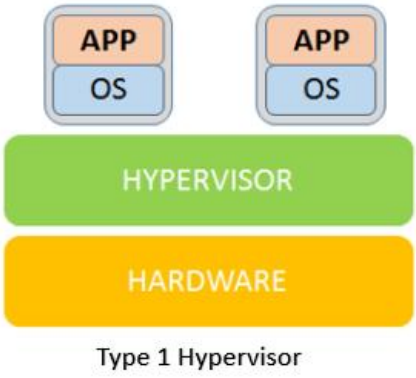

Figure 67 : Type 1 hypervisor

1.1 ESXi host 1

After the installation of VMware ESXi on the designated host, we access its Direct Console User Interface (DCUI) to configure the ESXi host and set up an IP address. DCUI enables you to modify its settings and troubleshoot ESXi networking on your hosts without using the ESXi web GUI or the vSphere Client.

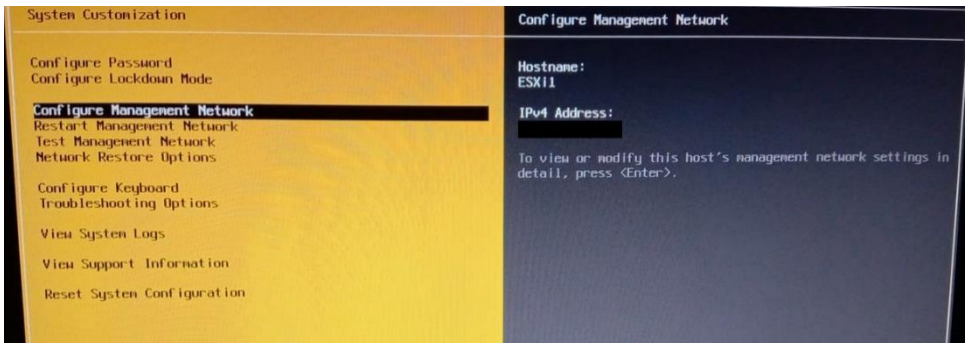

Figure 68: ESXi DCUI

Now by accessing the web interface of the ESXi1 host and after logging in as the root user, we are greeted by the index page showing basic information about the host.

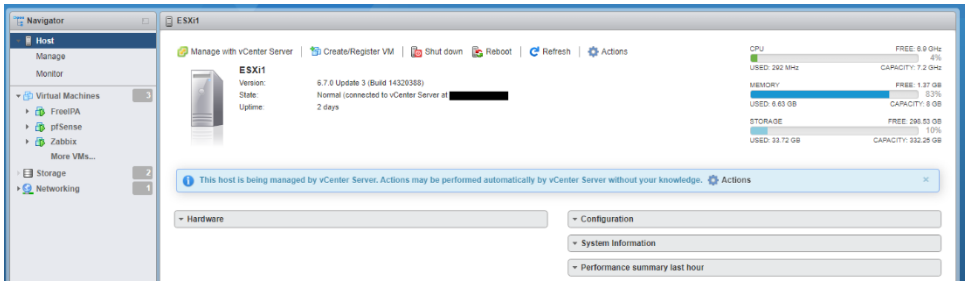

Figure 69 : ESXi1 index page

From this interface, virtual machines can be either created directly on the host or also exported, in our case we exported the VMDK and OVF files of the servers from VMware Workstation after the testing phase.

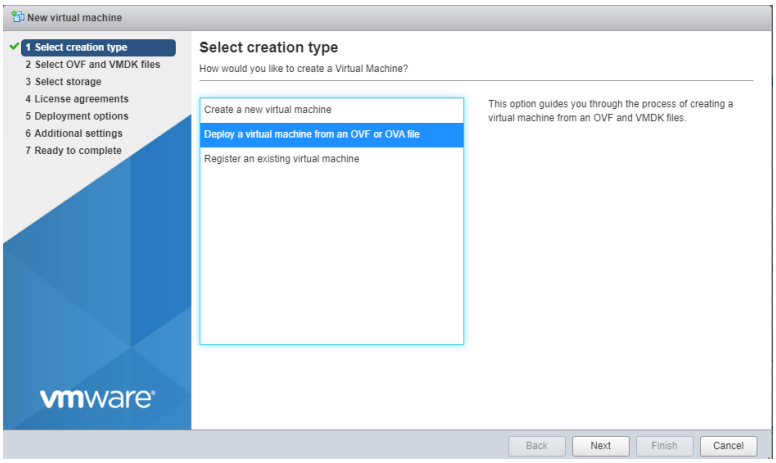

Figure 70: Adding a VM to the ESXi host

Our servers are now added into the ESXi1 host and can be accessed using a console.

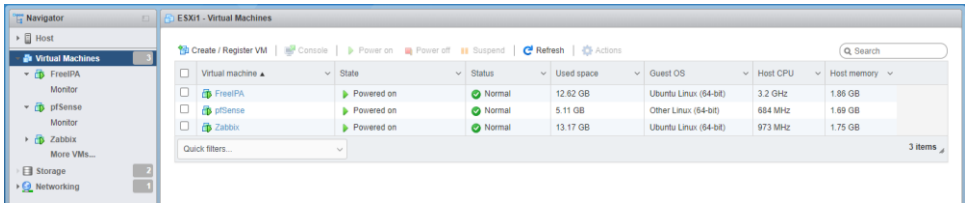

Figure 71: Our servers in ESXi1

This translates to the following architecture.

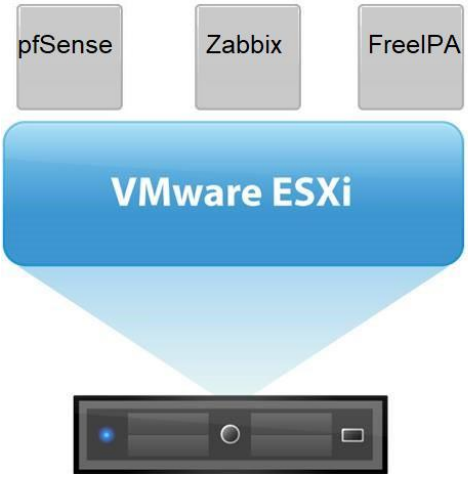

Figure 72 : ESXi1 architecture

1.2 ESXi host 2

The second ESXi host is created to host the VMware vCenter Server Appliance. vCenter Server Appliance comes as an Open Virtualization Format (OVF) template. The appliance is imported to an ESXi host and configured through the web-based interface. It comes pre-installed with all the components needed to run a vCenter Server, including vCenter SSO (Single Sign-on), Inventory Service, vSphere Web Client and the vCenter Server itself.

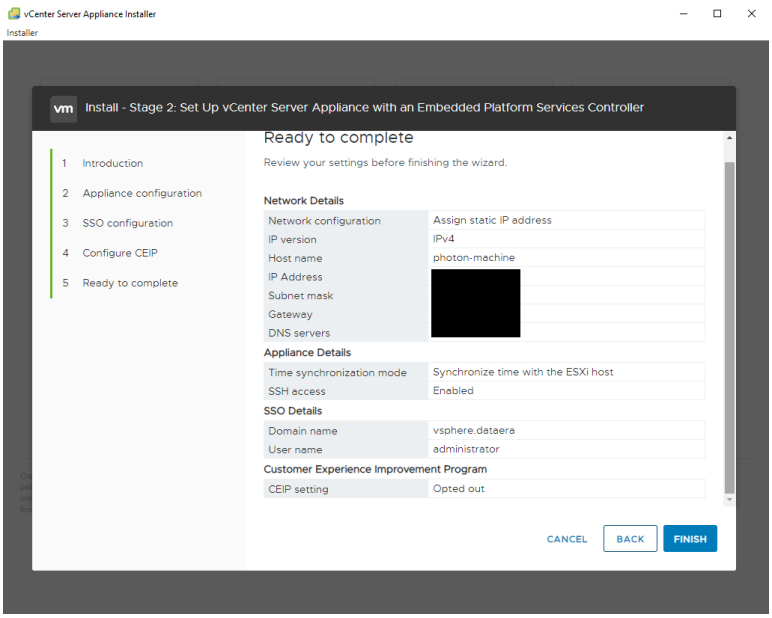

Figure 73 : Final step of vCenter Server Appliance

After the installation, we can find it ready to boot in its targeted host, our ESXi 2 host.

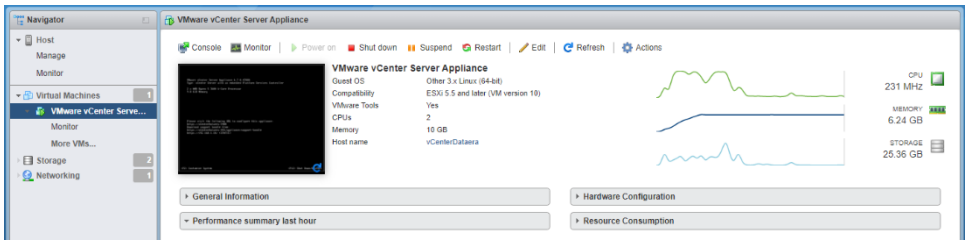

Figure 74: VSA running on ESXi2

Once we start the vCenter server appliance from the ESXi2 host, we can access the Appliance Management web interface and the vSphere client.

2 vCenter Appliance Management

The vCenter Server Appliance Management Interface (VAMI) is the administration Web interface for the vCenter Server Appliance (VCSA), and is used to perform basic administrative tasks such as monitoring the VCSA, changing the host name and the network configuration, NTP configuration, and applying patches and updates.

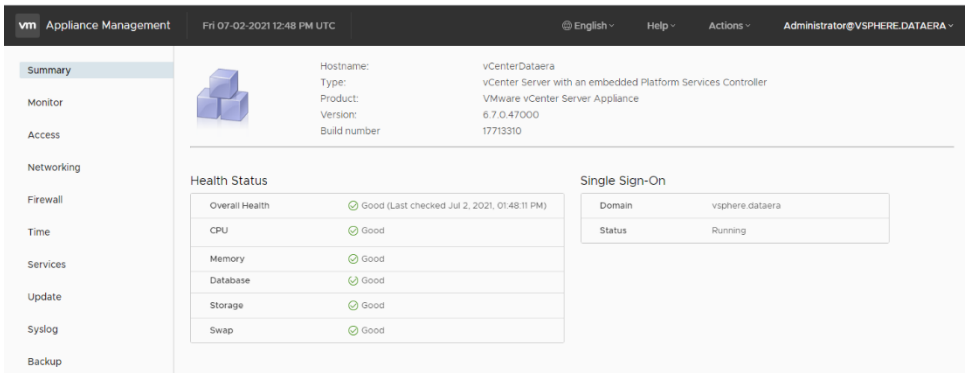

Figure 75: vCenter Appliance management index page

From this web interface we are able to configure the vCenter server, some of the settings that we can edit are:

- View the vCenter server appliance health status
- Create a support bundle enable or disable SSH and bash shell access
- Start, stop, and restart services
- Update settings
- Forward vCenter server appliance log files to remote syslog server

3 vSphere

vSphere Web Client is essentially an administrative interface that VMware administrators can use to access VMware hosts. It allows administrators to create new virtual machines and manage existing ones and their resources.

We as system administrators can access vCenter Server remotely to create, clone and manage virtual machines through vSphere Client.

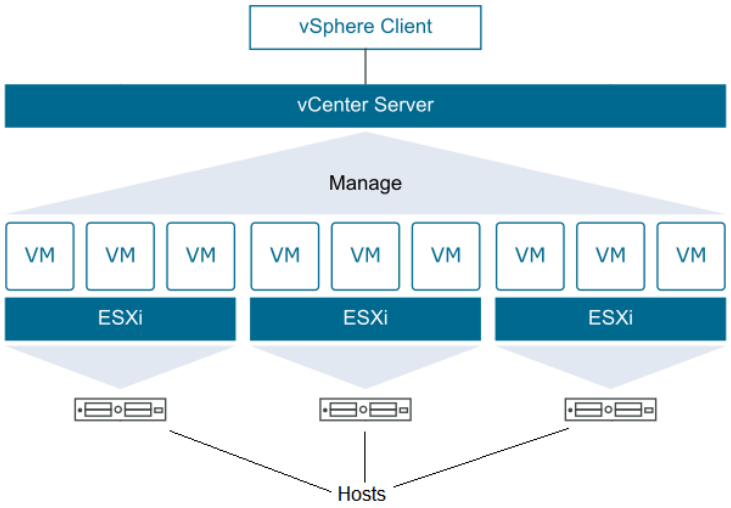

Figure 76: vSphere complete architecture

After verifying from the vCenter appliance management web interface that the vCenter server and all services are started and running, we can access the vSphere web client.

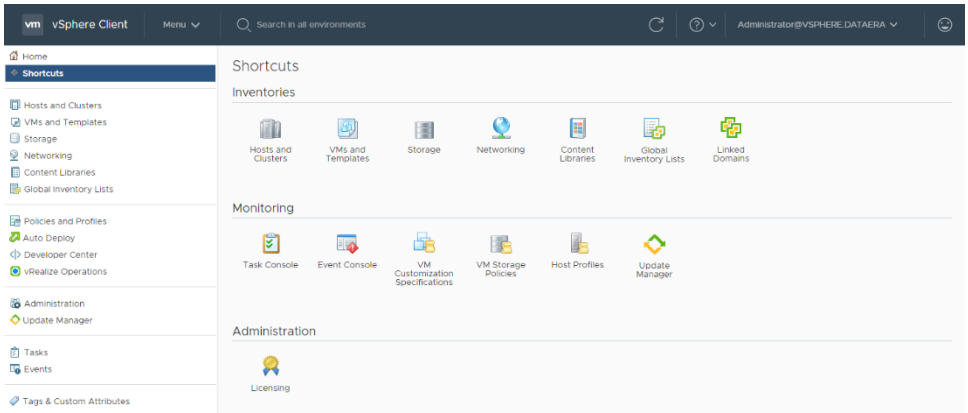

Figure 77 : vSphere web client

After creating our datacenter and our cluster, and adding our two ESXi hosts, the vCenter server detects automatically the VMs running on those hosts and adds them to the cluster. A cluster in vSphere is a collection of ESXi hosts configured to share their resources.

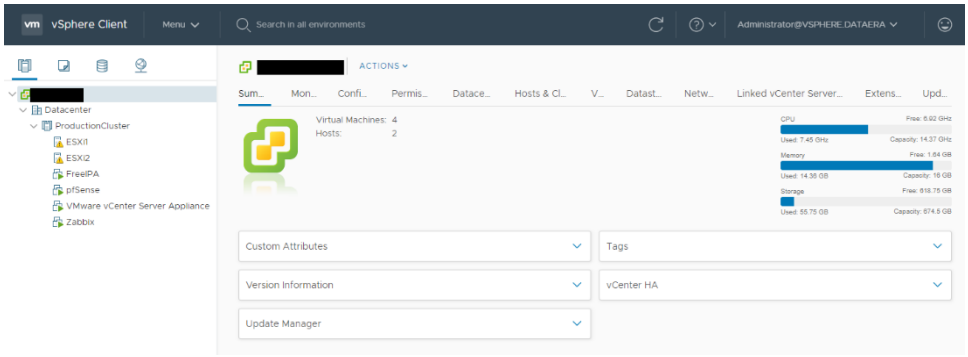

Figure 78: vSphere hosts and cluster

From now on, the VMs can be accessed from this web interface instead of their respective ESXi host interface, which makes managing multiples VMs on different hosts much easier. Also, clusters are used to enable some of the more powerful features in vSphere, such as High Availability (HA), Distributed Resource Scheduler (DRS), vMotion and Fault Tolerance (FT).

3.1 High Availability

vSphere HA provides high availability for virtual machines by pooling the virtual machines and the hosts they reside on into a cluster. Hosts in the cluster are monitored and in the event of a failure, the virtual machines on a failed host are restarted on alternate hosts.

Once a cluster is created, a single host is automatically elected as the primary host, for us it is the ESXi2 host because the vCenter server was installed on it. The primary host communicates with vCenter Server and monitors the state of all protected virtual machines and of the secondary hosts. Different types of host failures are possible, and the primary host must detect and appropriately deal with the failure. The primary host must distinguish between a failed host and one that is in a network partition or that has become network isolated. The primary host uses network and datastore heart beating to determine the type of failure.

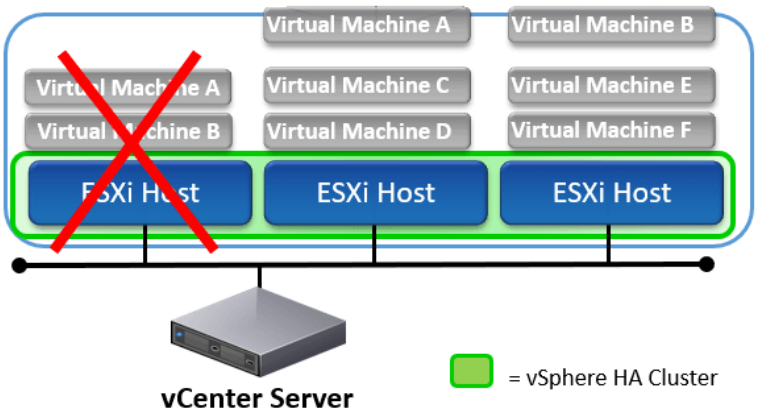

Figure 79: vSphere HA visualized

The HA settings can be found under Cluster→Configure→Services→vSphere Availability

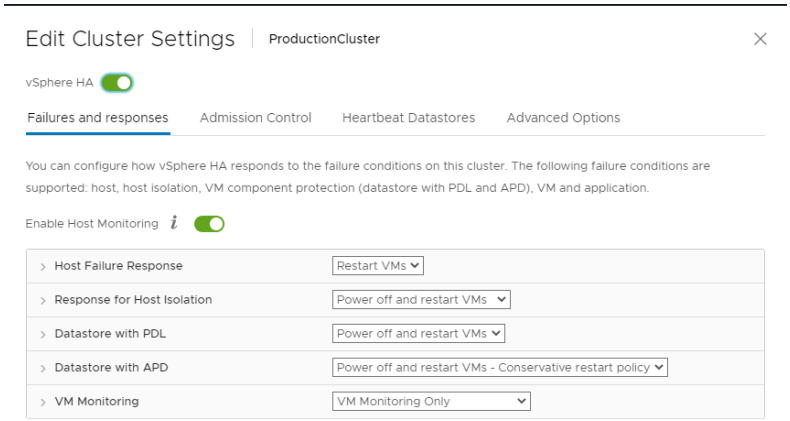

Figure 80: vSphere HA settings

3.2 Distributed Resource Scheduler

VMware vSphere Distributed Resource Scheduler (DRS) is a feature that enables a virtual environment to automatically balance itself across the ESXi hosts in a cluster in an effort to eliminate resource contention. The goals of DRS are:

- At startup, DRS attempts to place each VM on the host that is best suited to run that virtual machine.
- While a VM is running, DRS seeks to provide that VM with the required hardware resources while minimizing the amount of contention for those resources in an effort to maintain balanced utilization levels.

If a DRS cluster becomes unbalanced, DRS can migrate VMs from overutilized ESXi hosts to underutilized hosts. DRS performs these migrations of VMs across hosts in the cluster without any downtime by using vMotion. You can determine whether DRS will just display migration recommendations or automatically perform the migration when the cluster becomes unbalanced by defining the automation level.

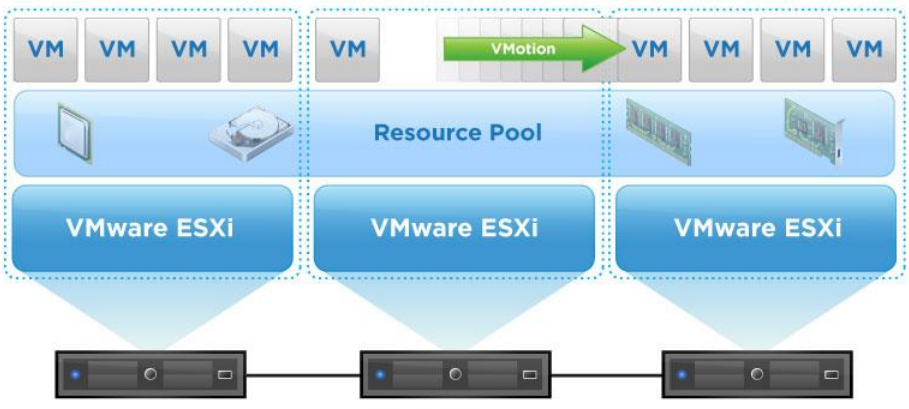

Figure 81: vSphere DRS visualized

The DRS setting is at Cluster→Configure→Services→ vSphere DRS

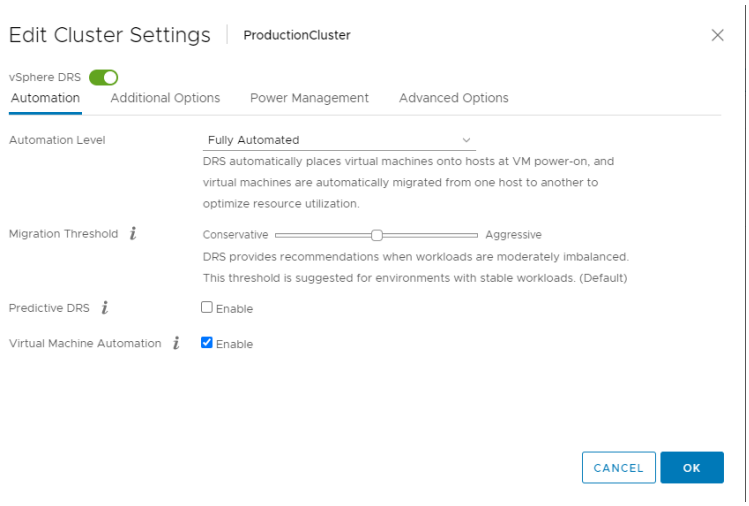

Figure 82: vSphere DRS settings

3.3 Fault Tolerance

vSphere Fault Tolerance (FT) provides a higher level of business continuity than vSphere HA. It works by creating a duplicate (secondary) copy of the virtual machine on a different host and keeping the two VMs in sync. The secondary VM can immediately take over in the event of an ESXi host failure and the entire state of the virtual machine will be preserved.

Because FT provides zero downtime and zero data loss, it is usually used for business-critical applications that must be available all the time. It is also sometimes used for applications that have no native capability for clustering.

vSphere FT also has some disadvantages. Here are the main ones:

- Increased resource usage. An FT-protected VM will use twice as much resources. For example, if the primary VM uses 2GB of RAM, the secondary VM will also use 2GB of RAM.
- Only virtual machines with a single vCPU are compatible with Fault Tolerance.
- Hosts must be licensed for vSphere FT.
- The VM must not have any snapshots.

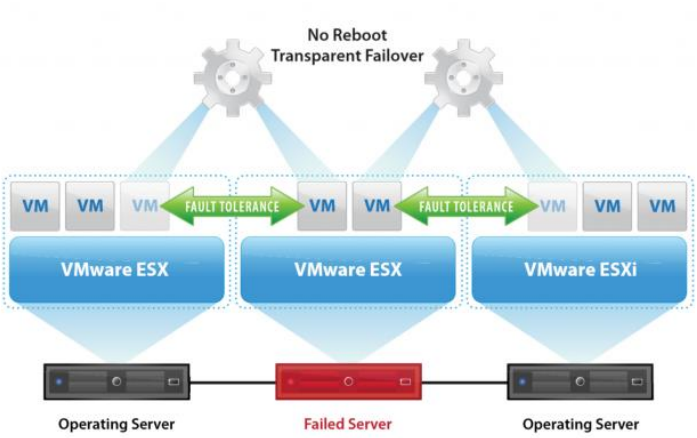

Figure 83: vSphere FT visualized

Conclusion:

In this final chapter, we showed how our servers are functioning in a virtualized environment. The most important feature of virtualization is the capability of running multiple operating systems and applications on a single computer or server which can usually improve overall application performance due to technology that can balance resources, and provide only what the user needs.