

# Using **ZeroShell** as a NetBalancer, QoS server & Captive Portal.

(Among other things)

By Jose Menendez

orallo at gmail dot com

## **Acknowledgments:**

Let me start this document by thanking all the people that works or contributes to this magnificent piece of software, starting by Fulvio Ricciardi and extending those thanks to all the people that contributed documents or participates on the ZeroShell forums, particularly Atheling for all the help he gave me personally and his fix for the NetBalancer/QoS combo.

## **My Scenario:**

I'm by no means a network guru, I am a programmer by training and by trade. I work at a small/medium company where we have no network administrator so the IT department (two programmers) has been put in charge of "anything and everything" computer related.

We have a LAN with about 60/70 computers. Off of those computers most do administrative tasks via a web interface, some do some FTP traffic and the most critical application on our office is that several times a week we broadcast a web conference to students.

Our internet connection is based on two aDSL lines one with 18MB download speed and 800kbps upload speed and the other line with 6MB download speed with 400kbps upload speed.

When we started we had manually assigned IPs on each computer some on router 1 and some our router 2. Basically we left the broadcasting studio alone on router 2 to "guarantee" that nobody on the LAN would "steal" their bandwidth in the middle of a class.

## **The Goals:**

We basically need to accomplish 3 things.

- First we need to provide internet service to all the users on the LAN consistently (regardless of the fact that one or the other DSL line might go down once in a while)
- Second we need to guarantee that the broadcasting studio has always enough bandwidth to broadcast the classes.
- And Lastly we need to implement some basic HTTP proxy with a blacklist of sites and a captive portal so all the users are aware that they are connected to the internet via a company computer.

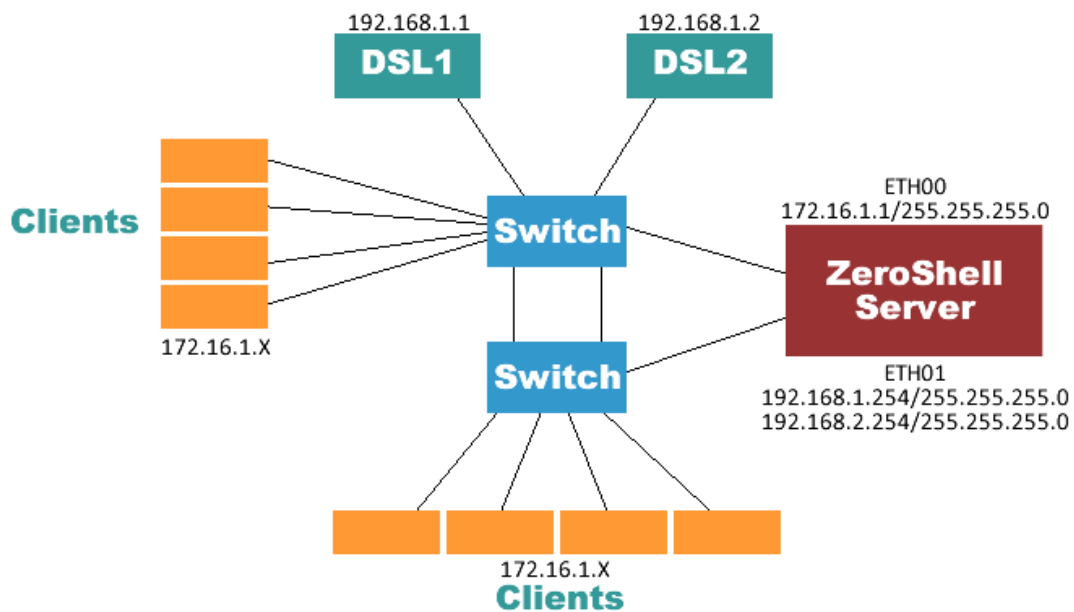
## The ZeroShell Solution:

I'm not going to go into much detail on the basic installation of ZeroShell, because it is simple enough and most important there is plenty of documentation about it.

We installed ZS on a computer with two inexpensive 100Mb/s full duplex RTL-8139/8139C/8139C+ network cards one facing the LAN and the other one facing the WAN.

The LAN is going to be on the 172.16.1.XXX segment and the routers each will stay with their current configuration, DSL1 will stay on 192.168.1.1 and DSL2 will stay on 192.168.2.1.

Here is a basic diagram of our network layout

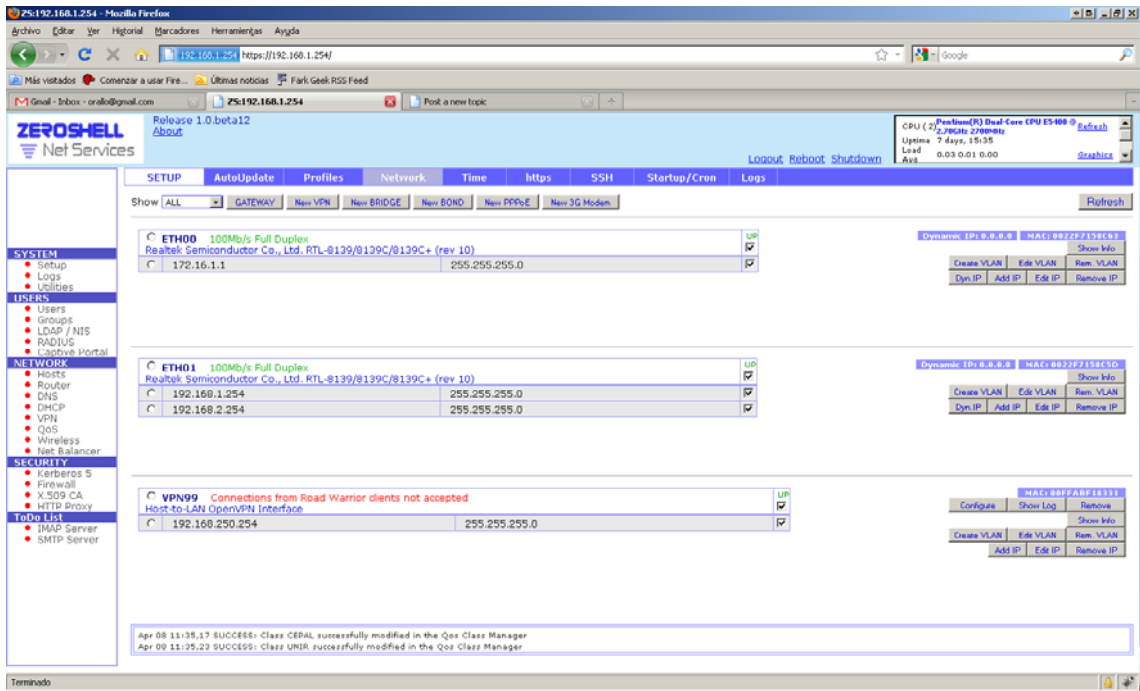


### Note 1

*It has been pointed out that ZS should be physically placed between the routers and the switches to make sure no client changes his/her network configuration and bypasses the ZS setup. We partially agree with this notion, but right now it is not an option for us, the only place where we can place the ZS box is in our office. And our users do not have administrator rights so they can't really change their network settings.*

First we proceed to the NIC configuration by going to the System/Setup/Network tab on the web interface, and we Add IP 172.16.1.1/255.255.255.0 to ETH00 and we add IPs 192.168.1.254/255.255.255.0 and 192.168.2.254/255.255.255.0 to ETH01.

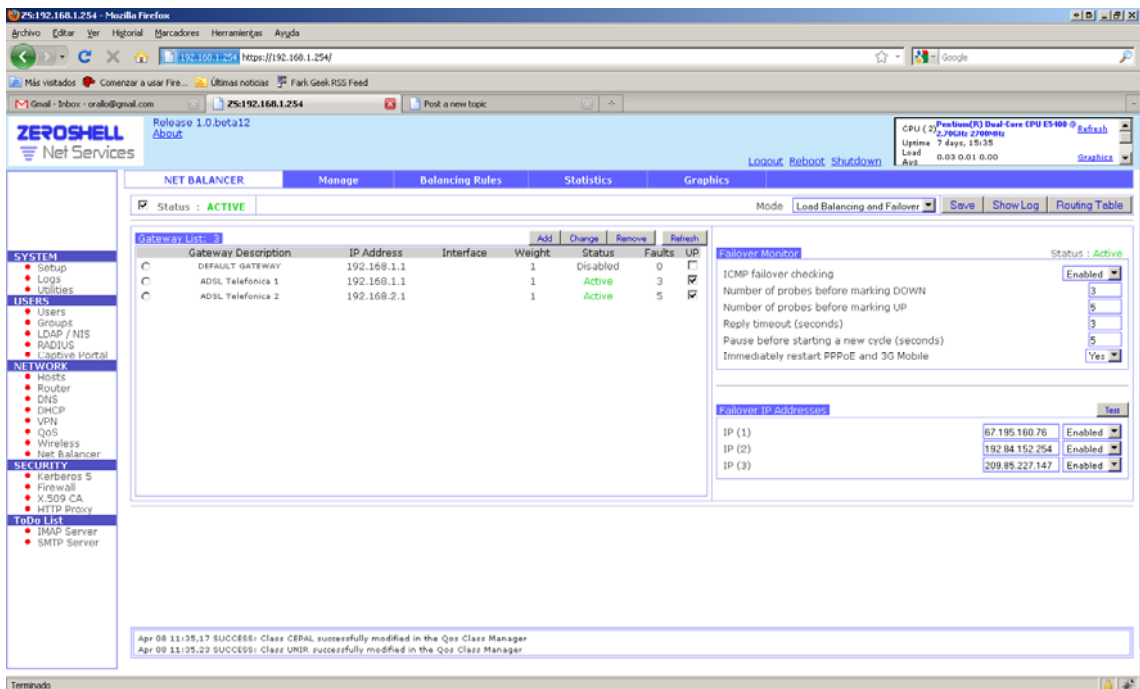
ETH00 will face the LAN and ETH01 will face the internet routers.



The first goal is to provide Net Balancing on our two internet connections so if for some reason one of the DSL connections goes down, all the users on the LAN can still work so we go to the Network/NetBalancer/Manage tab enabled the checkmark to make the net balancing active and added our two routers.

We clicked on Add, then we gave each router/gateway a name, entered the IP address of each router and clicked on Save. Then we disabled the default gateway that ZS adds by default.

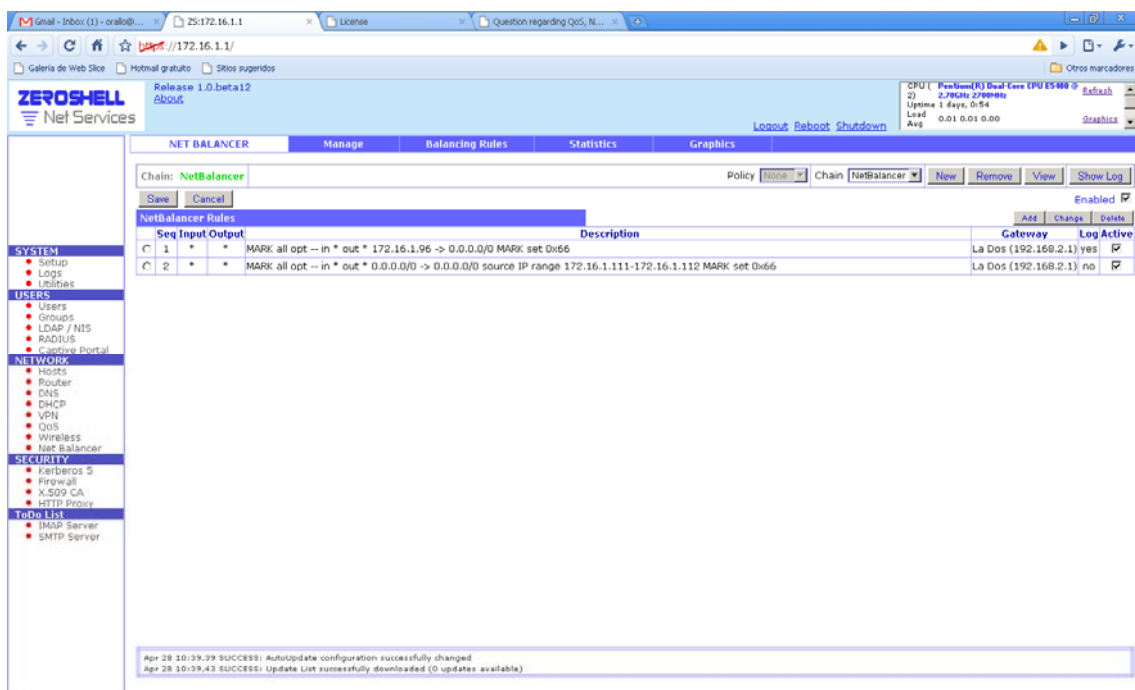
So now the NetBalancer looks like this:



Then we proceeded to enable the ICMP failover checking, to accomplish this, we added some IPs to the Failover IP Addresses list (we added, one of google's IPs, one of cnn.com's and one of yahoo's if I remember correctly).

One side note to the NetBalancing, since we had put NB in place, some users experienced some problems using some sites that require authentication, in particular I had problems managing some websites using Plesk as the interface and one user had problems using a particular web forum based on some software package called mvnForum.

The solution to this problem is going to the NetBalancing, balancing rules and creating a rule that sends those users always through the same gateway regardless. Apparently Plesk and mvnForum check the IP headers and if a user goes through one gateway one minute and the other gateway 5 minutes later they lose the authentication...



So first goal accomplished!!! On to the next goal, QoS.

Reading through the ZS forums we found out that there was a glitch on ZS that makes that you can't use Net Balancing AND QoS at the same time.

I had some contacts on the forum with Atheling, as he answered some of my posts. On one of his posts he said that he had created a patch to fix the QoS/NB conflict, so I asked him if I could get a copy. He was kind enough to send it to me, so I installed it and proceeded to test it, and it works like a charm!

The patch that Atheling created can be found on the ZeroShell forums on the following thread:

<http://www.zeroshell.net/eng/forum/viewtopic.php?t=2125>

#### Note 2

I installed the patch by first copying the patch to the kerbynet.cgi folder and issuing the following command:

```
patch -p0 < Zeroshell.3.patch
```

Then the system responded with:

```
patching file scripts/fw_initrules
patching file scripts/fw_makerule
patching file scripts/fw_start
patching file scripts/fw_viewchain
patching file scripts/nb_fw
patching file scripts/nb_setautomarking
```

Then to make the changes permanent, since most of the system resides on RAM and its reloaded every time the system is rebooted, I made a copy of the patched files and placed them on a folder I created under /Database which is on the HDD and therefore does not disappear on reboot, and I put the following script on the preboot script on the cron tab.

```
for file in /Database/custom/*
do
    cp ${file} /root/kerbynet.cgi/scripts/
done
```

So first we created some classes to group users by, we created seven groups based on the type of activities that each user does and the physical location within the building. We had segmented the LAN by location/IP so this came in handy. For example each classroom has its own consecutive IP range, the broadcast studio has another consecutive IP range, the reception area has its own consecutive IP range, etc, etc...

This is what our class manager looks like now:

QoS Class Manager - Mozilla Firefox

192.168.1.254 https://192.168.1.254/cgi-bin/kerbynet

### QoS - CLASS MANAGER

CEPAL Description: Tutores

Priority: Medium DSCP: Maximum: 1 Mbit/s Guaranteed: 300 Kbit/s

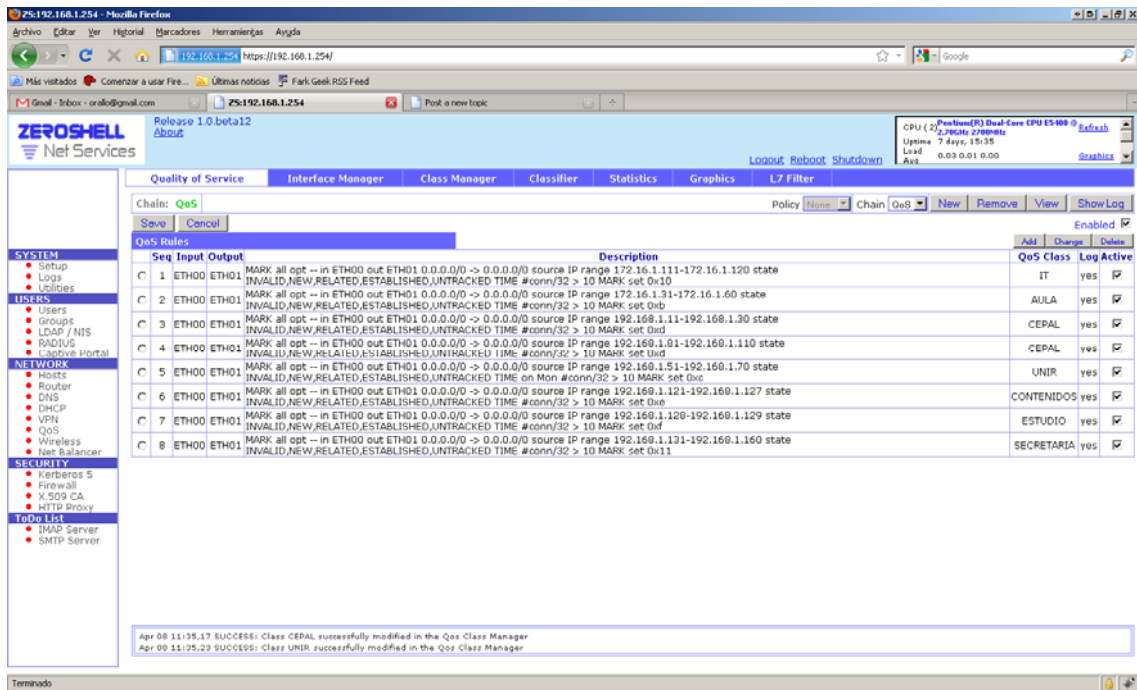
Class	Description	Priority	DSCP	Max Bandwidth	Guaranteed	On
<input type="radio"/> AULA	Trafico Del Aula SRE	Low		1Mbit/s	300Kbit/s	<input checked="" type="checkbox"/>
<input checked="" type="radio"/> CEPAL	Tutores	Medium		1Mbit/s	300Kbit/s	<input checked="" type="checkbox"/>
<input type="radio"/> CONTENIDOS	Editores De Contenido	Medium		1Mbit/s	300Kbit/s	<input checked="" type="checkbox"/>
<input type="radio"/> DEFAULT	Default class for unclassified traffic	Medium		1Mbit/s	100Kbit/s	<input checked="" type="checkbox"/>
<input type="radio"/> ESTUDIO	Estudio De Grabacion	High		1Mbit/s	1Mbit/s	<input checked="" type="checkbox"/>
<input type="radio"/> IT	Departamento De Informatica	High		1Mbit/s	1Mbit/s	<input checked="" type="checkbox"/>
<input type="radio"/> SECRETARIA	Secretaria y Recepcion	Medium		1Mbit/s	300Kbit/s	<input checked="" type="checkbox"/>
<input type="radio"/> UNIR	Profesores Tutores	Medium		1Mbit/s	300Kbit/s	<input checked="" type="checkbox"/>

Terminado

### Note 3

The bandwidths are managed manually for now on the interface manager until we fine tune the numbers via trial and error so the numbers on the class manager are not representative.

Next we added the classifying rules on the classifier, by clicking add, entering the IP ranges for each range, and assigning them to a TARGET CLASS at the bottom of the configuration screen.

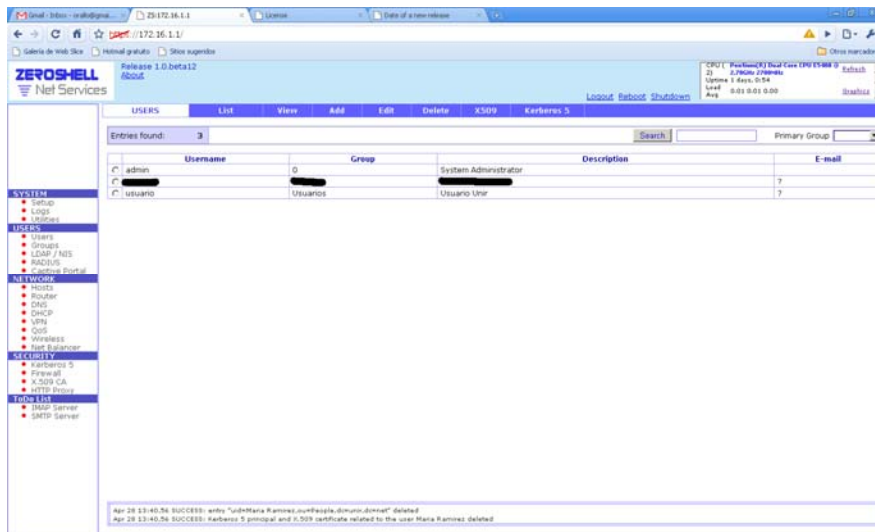


Then we went back to the Interface Manager of the QoS and enabled the newly created classes with their rules. And voilà, we had QoS up and running.

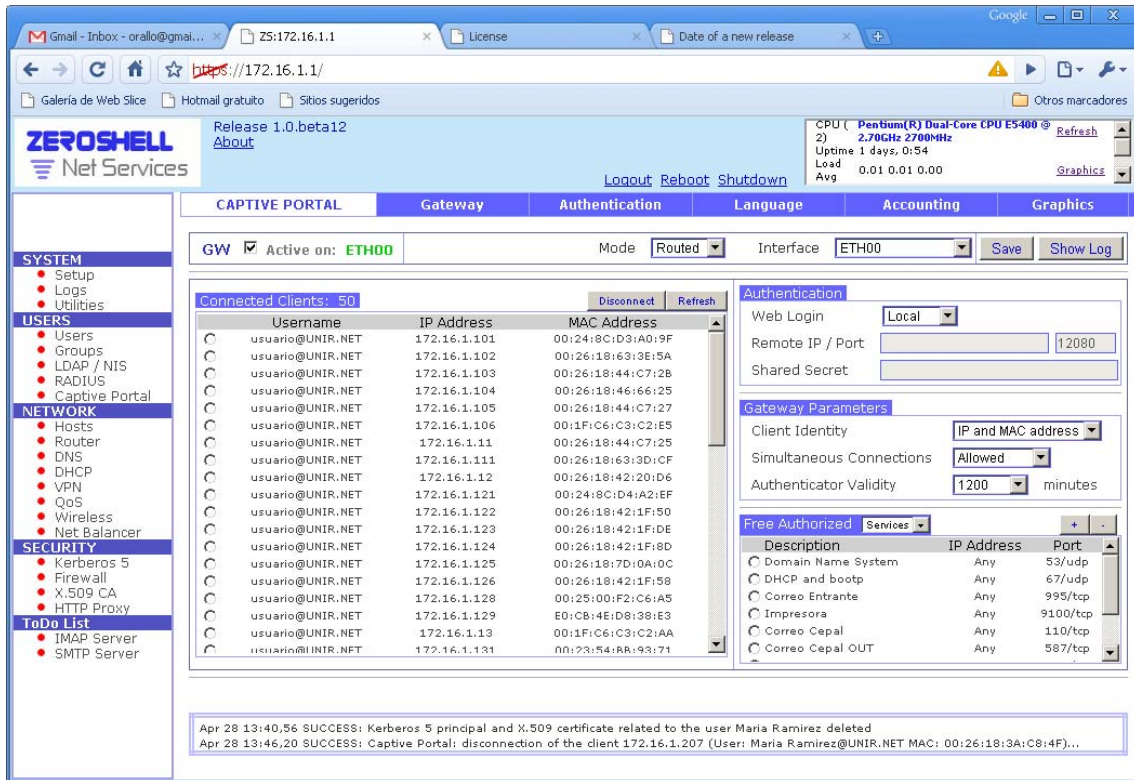
## Setting up the Captive Portal:

For our needs we don't need to create a user account for each user that logs into the system, management just want users to be aware that they are connected to the internet via a company computer in the hopes that they will feel monitored and they will be on their best behavior while on the internet at work.

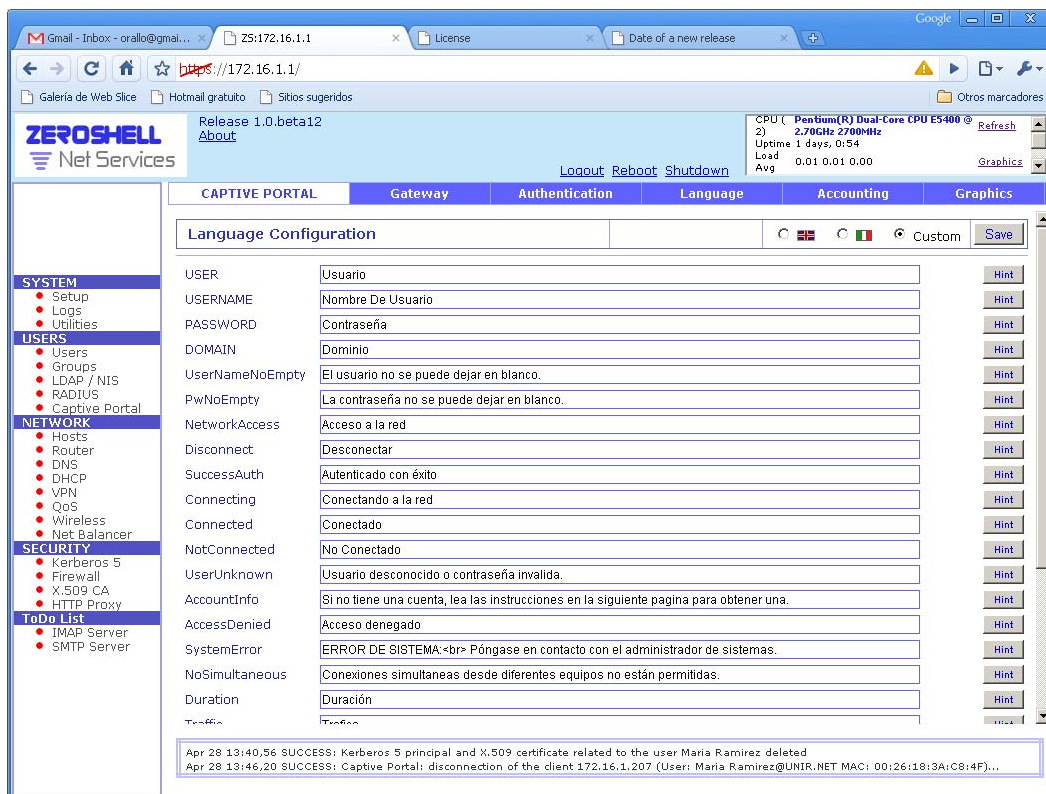
So we created a generic "usuario" account that all the users on the LAN share and use to connect to the internet via the captive portal.



Then we enable local authentication on the captive portal and added some items to the “Free Authorized Services” area at the bottom of the captive portal. We added one for the office printer opening port 9100 for it and some other ports that are used to connect to our mail servers.

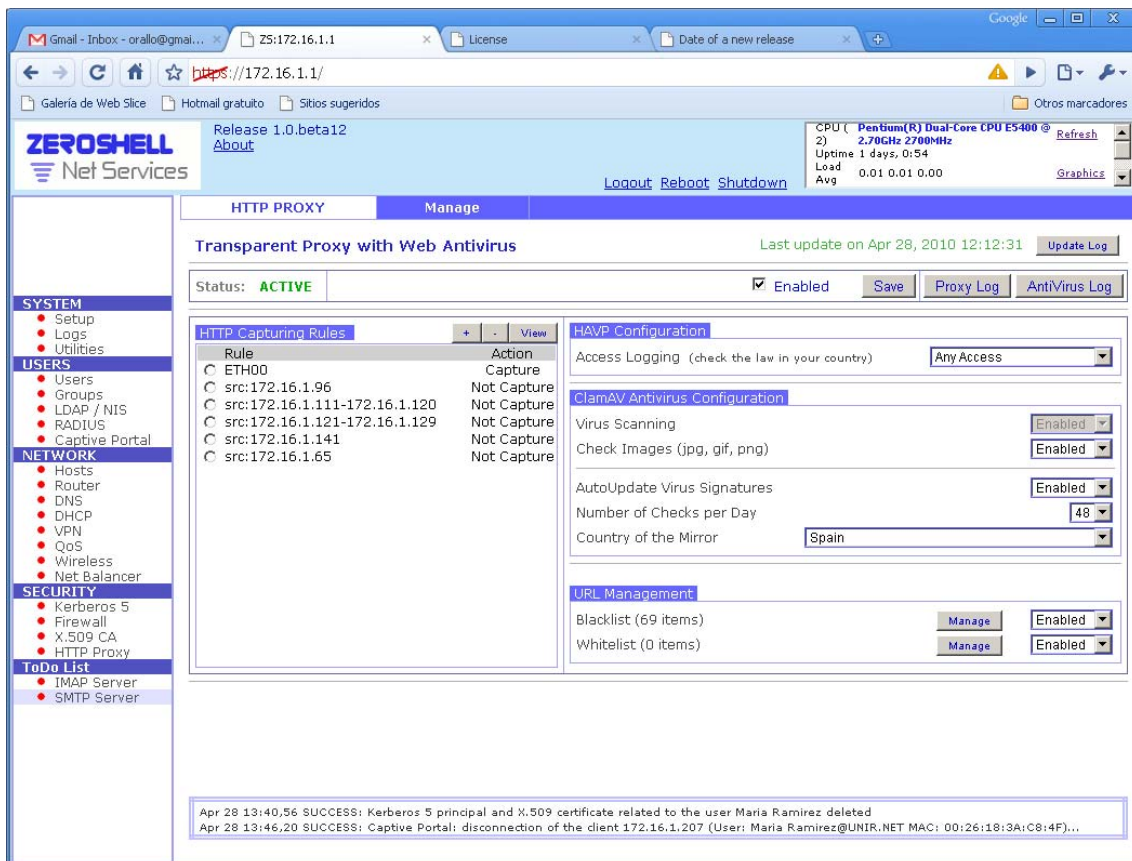


Later we went on to the Language tab and translated the portal to Spanish.



Finally we needed to setup a list of sites that are not to be accessed from our LAN. So we went to the HTTP Proxy area of ZS, enabled it and went to the Blacklist manager and entered the list of sites that management wants blocked.

Then we added a general rule to capture the LAN traffic on ETH00 and some exceptions (not capture) for the people that are authorized to be off the proxy and get unrestricted web access and we are done with our third and last goal.



#### Note 4

Since we were at it, we added ClamAV protection to our LAN by simply enabling the Virus Scanning via the drop list, selecting the number of updates per day and selecting the mirror for our country.

## Conclusions:

In general ZeroShell was a breeze to install, we had some minor issues with the storage, but pretty much everything is documented on the ZeroShell site and/or the ZeroShell forums.

Chances are that if you run into a problem while installing it, someone has had the problem before, fixed it and documented it.

Configuring ZeroShell was also easy, we think that if we had a better background in network stuff we wouldn't have had most of the problems we had, and even so, it was just a matter of

reading a lot of documentation and posts of the forums to find out what each thing was supposed to do, and how it did it and then just translating that to ZeroShell.

I hope this manual helps people with goals similar to the ones we had, and if anything needs to be clarified, corrected or removed from this document, do not hesitate to contact me at orallo at gmail dot com and will do so as soon as possible.

Finally I would like to reiterate our thanks to Mr. Fulvio Ricciardi for this excellent piece of software and we look forward to future updates that will hopefully include even more and better features if that's even possible.

Thank You.