Zoneminder 1.30 FreeBSD Installation and configuration

*Notes: All commands used assume you are logged as root; otherwise add **sudo** in front of the commands, where applicable.*
I use joe to edit files, you can use whatever you feel confortable with (i.e.: vi, nano, etc)
This document is a WIP.
Version: 1.00
zoneminder installed and working
Cameras working

1. First things first: Update your ports to ensure you will be installing the updated version of zoneminder and its depencies.

root@freebsd-1:~ # **portsnap fetch extract**

That will download a compressed version of all ports and extract them. It might take awhile, so grab a cup of coffee.

2. Compile zoneminder from ports

root@freebsd-1:~ # **cd /usr/ports/multimedia/zoneminder**
root@freebsd-1:~ # **make install clean**

Have a mentioned you needed a cup of coffee for the previous step? Well … compile from ports will take a long time and will have several steps to select the options for installations, so keep an eye on it from time to time. (* Note to self: check for anatended make options *)

In my case it took over two hours to finish:

```
===>  Cleaning for p5-DBI-1.636
===>  Cleaning for p5-DBD-mysql-4.033_1
===>  Cleaning for p5-Date-Manip-6.53
===>  Cleaning for p5-Test-LWP-UserAgent-0.030
===>  Cleaning for p5-Sys-Mmap-0.17_1
===>  Cleaning for p5-LWP-Protocol-https-6.06_1
===>  Cleaning for p5-Sys-CPU-0.52_1
===>  Cleaning for p5-Sys-MemInfo-0.91_1
===>  Cleaning for p5-Data-Dump-1.23_1
===>  Cleaning for p5-SOAP-WSDL-3.003
===>  Cleaning for p5-Data-UUID-1.221
===>  Cleaning for p5-IO-Socket-Multicast-1.12_2
===>  Cleaning for ffmpeg-2.8.7_1,1
===>  Cleaning for cmake-3.5.2_1
===>  Cleaning for gettext-tools-0.19.8.1
===>  Cleaning for jpeg-turbo-1.4.2
===>  Cleaning for mysql56-client-5.6.30
===>  Cleaning for sudo-1.8.16_2
===>  Cleaning for zip-3.0_1
===>  Cleaning for zoneminder-1.30.0
root@freebsd-1:/usr/ports/multimedia/zoneminder #
```

Just in case, checked the installed packages:

root@freebsd-1:/usr/ports/multimedia/zoneminder # **pkg info**

```
.
.
.
zoneminder-1.30.0            Complete security camera solution, fully web based with image
analysis
root@freebsd-1:/usr/ports/multimedia/zoneminder #
```

One of the assutions of the port maintener (or creator, sorry not a FreeBSD guy) is that this will run with an existent HTTP/SQL implementation.

For this how-to I'll be configuring a stand alone VM and will install FEMP on it. You can adapt the steps from https://raw.githubusercontent.com/freebsd/freebsd-ports/master/multimedia/zoneminder/files/README.FreeBSD to your particular need.

3. FEMP Installation and configuration

3.1. Installation with pkg

root@freebsd-1:~ # **pkg install nginx mysql56-server php56-mysql php56-pdo_mysql php56-gd fcgiwrap php56-session**
*** Update outputs below

```
root@freebsd-1:~ # pkg install nginx mysql56-server php56-mysql php56-pdo_mysql php56-gd fcgiwrap
Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 20 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        nginx: 1.8.1_3,2
        mysql56-server: 5.6.30
        php56-mysql: 5.6.22
        php56-pdo_mysql: 5.6.22
        php56-gd: 5.6.22
        fcgiwrap: 1.1.0_3
        pcre: 8.38_1
        libXmu: 1.1.2_3,1
        libXt: 1.1.5,1
        libSM: 1.2.2_3,1
        libICE: 1.0.9_1,1
        libXaw: 1.0.13,2
        libXpm: 3.5.11_4
        printproto: 1.0.5
        libXp: 1.0.3,1
        png: 1.6.21
        php56: 5.6.22
        php56-pdo: 5.6.22
        t1lib: 5.1.2_4,1
        fcgi-devkit: 2.4.0_5

The process will require 117 MiB more space.
14 MiB to be downloaded.

Proceed with this action? [y/N]: Y ← type Y and press ENTER
```

3.2. Enable all new services

```
root@freebsd-1:~ # sysrc mysql_server_enable="YES"
mysql_server_enable:  -> YES
root@freebsd-1:~ # sysrc fcgiwrap_enable="YES"
fcgiwrap_enable:  -> YES
root@freebsd-1:~ # sysrc zoneminder_enable="YES"
zoneminder_enable:  -> YES
root@freebsd-1:~ # sysrc mysql_enable="YES"
mysql_enable:  -> YES
root@freebsd-1:~ # sysrc nginx_enable="YES"
nginx_enable:  -> YES
root@freebsd-1:~ # sysrc php_fpm_enable="YES"
php_fpm_enable:  -> YES
```

3.4. Check rc.conf

root@freebsd-1:~ # less /etc/rc.conf

```
hostname="freebsd-1"
ifconfig_em0="DHCP"
sshd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
webmin_enable=YES
mysql_server_enable="YES"
fcgiwrap_enable="YES"
zoneminder_enable="YES"
mysql_enable="YES"
nginx_enable="YES"
php_fpm_enable="YES"
root@freebsd-1:~ #
```

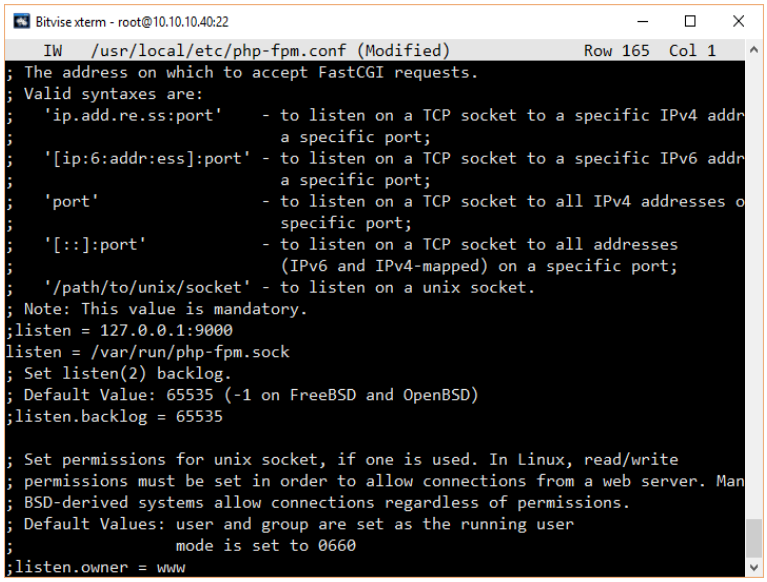
4. Configure php

4.1. Edit php-fpm.conf

root@freebsd-1:~ # **joe /usr/local/etc/php-fpm.conf**

- Find the line: **listen = 127.0.0.1:9000**
- Replace with: **listen = /var/run/php-fpm.sock**

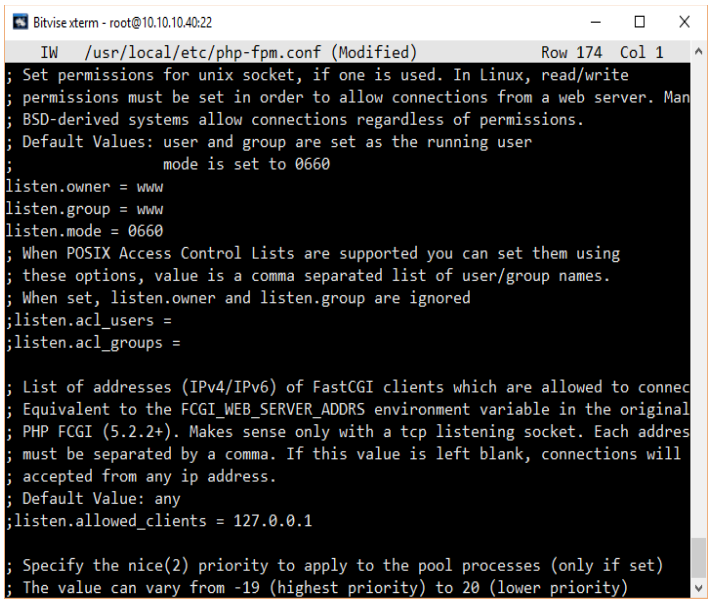
Will look like this:



- Two sessions below you will find the permission for unix socket, **uncomment the three lines below:**

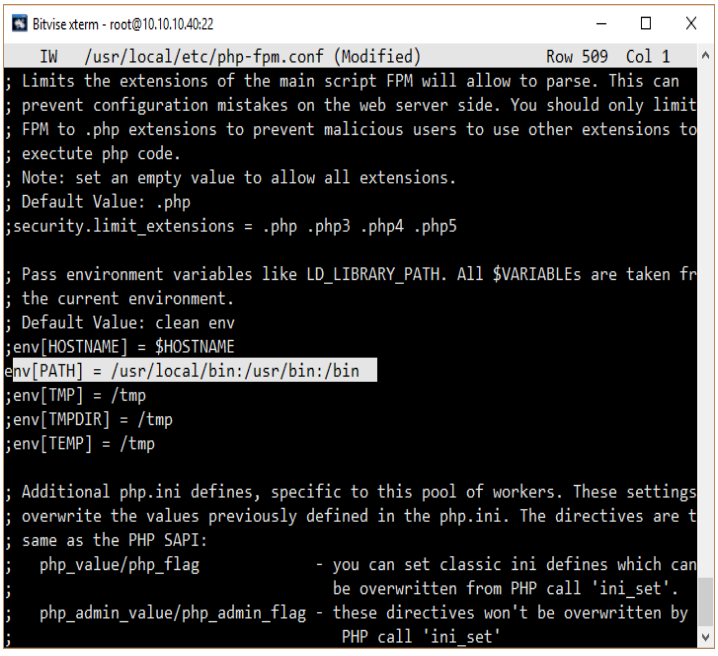
listen.owner = www
listen.group = www
listen.mode = 0660

Will look like this:



- Uncomment env[PATH] = /usr/local/bin:/usr/bin:/bin

Will look like this:



- Save and close the file

5. Create php.ini

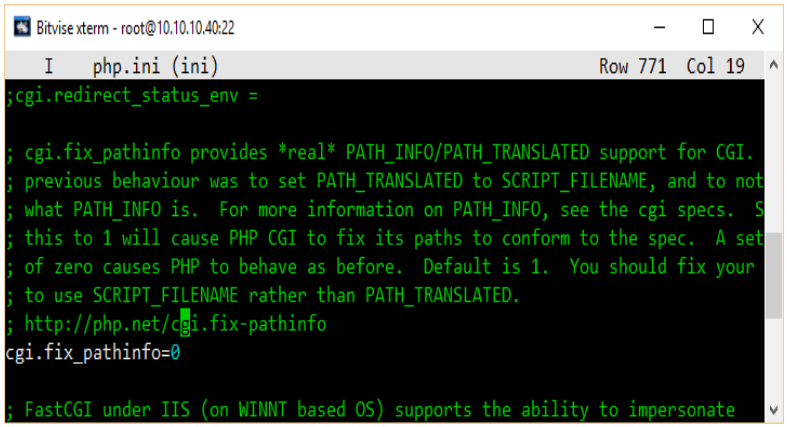
5.1. Copy the sample production file

root@freebsd-1:/ # **cd /usr/local/etc**
root@freebsd-1:/usr/local/etc # **cp php.ini-production php.ini**

5.2. Edit the file and set cgi.fix_pathinfo=0

root@freebsd-1:/ # **joe php.ini**

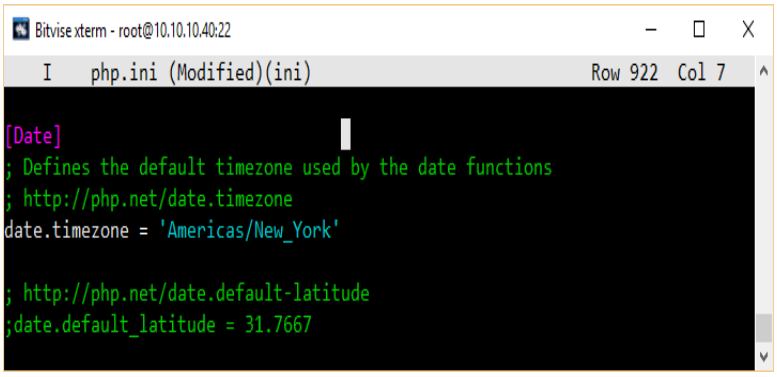
Will look like this:



5.3. Set your date time zone

- Uncomment date.timezone and type your time zone within single quotes (').

Notes: Check your time zone here: http://php.net/manual/en/timezones.php
I'm in EST, so will set to 'America/New_York' and will look like this:



- Save the file

5.4. Start php

root@freebsd-1:/usr/local/etc # **service php-fpm start**

```
Performing sanity check on php-fpm configuration:
[16-Jun-2016 19:41:24] NOTICE: configuration file /usr/local/etc/php-fpm.conf test is successful

Starting php_fpm.
```

6. Configure MySQL

6.1. Start the service:

root@freebsd-1:/usr/local/etc # **service mysql-server start**

```
Starting mysql.
root@freebsd-1:~ #
```

6.2. Execute installer

root@freebsd-1:/usr/local/etc # **mysql_secure_installation**

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current password for the root user. If you've just installed MySQL, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

Enter current password for root (enter for none):

- **Press ENTER and set your new password**

Output: (Instructions in **RED** below)
Change the root password? [Y/n] ***Y***
New password: ***<type your password>***
Re-enter new password: ***<retype your password for confirmation>***
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

- **Press ENTER on all following prompts** (Instructions in **RED** below)

Remove anonymous users? [Y/n] ***<ENTER>***
... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] ***<ENTER>***
... Success!

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] ***<ENTER>***
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] ***<ENTER>***
... Success!

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

Cleaning up...

6.3. Restart MySQL

root@freebsd-1:/usr/local/etc # **service mysql-server restart**

```
Stopping mysql.
Waiting for PIDS: 1427.
Starting mysql.
```

7. Configuring nginx

7.1. Start nginx

root@freebsd-1:/usr/local/etc # **service nginx start**

```
Performing sanity check on nginx configuration:
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
Starting nginx.
```

7.2. Create nginx.conf

Copy sample below to the new file been created:

```
user  www;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

pid        /var/run/nginx.pid;
events {
  worker_connections  1024;  ## Default: 1024
}
http {
      include mime.types;
      default_type  application/octet-stream;

      log_format  main  '$remote_addr - $remote_user [$time_local]
"$request" '
                        '$status $body_bytes_sent "$http_referer" '
                        '"$http_user_agent" "$http_x_forwarded_for"';

      sendfile on;
      tcp_nopush on;
      tcp_nodelay on;
      keepalive_timeout  65;
      server_tokens off;

      gzip  on;
      gzip_comp_level 4;
      gzip_min_length 1100;
```

```
      gzip_buffers 64 8k;
      gzip_http_version 1.1;
      gzip_proxied any;
      gzip_types text/plain application/xml application/x-javascript
text/css;

server {
        root /usr/local/www/zoneminder;
        try_files $uri $uri/ /index.php$is_args$args;
        index index.php;
        location = /cgi-bin/nph-zms {
           include fastcgi_params;
           fastcgi_param   SCRIPT_FILENAME
$document_root$fastcgi_script_name;
           fastcgi_pass    unix:/var/run/fcgiwrap/fcgiwrap.sock;
                                        }

         location ~ \.php$ {
           include fastcgi_params;
           fastcgi_param   SCRIPT_FILENAME
$document_root$fastcgi_script_name;
           fastcgi_pass    unix:/var/run/php-fpm.sock;
                             }

          location /api {
            rewrite ^/api/(.+)$ /api/index.php?p=$1 last;
                          }
       }
     }
```

root@freebsd-1:/usr/local/etc # **cd /usr/local/etc/nginx**
root@freebsd-1:/usr/local/etc/nginx # **mv nginx.conf nginx.conf-original**
root@freebsd-1:/usr/local/etc/nginx # **joe nginx.conf**

- Save the file

7.3. Create logs directory and empty files

root@freebsd-1:/usr/local/etc/nginx # **mkdir -p /var/log/nginx**
root@freebsd-1:/usr/local/etc/nginx # **touch /var/log/nginx/access.log**
root@freebsd-1:/usr/local/etc/nginx # **touch /var/log/nginx/error.log**

7.4. Configuring document root

root@freebsd-1:/usr/local/etc/nginx # **rm /usr/local/www/nginx**
root@freebsd-1:/usr/local/etc/nginx # **mkdir /usr/local/www/nginx**
root@freebsd-1:/usr/local/etc/nginx # **cp /usr/local/www/nginx-dist/index.html /usr/local/www/nginx**

7.5. Test nginx con

root@freebsd-1:/usr/local/etc/nginx # **nginx -t**

nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful

7.6. Restart nginx

root@freebsd-1:/usr/local/etc/nginx # **service nginx restart**

```
Performing sanity check on nginx configuration:
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
Stopping nginx.
Waiting for PIDS: 1659.
Performing sanity check on nginx configuration:
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
Starting nginx.
```

8. Configuring Zoneminder

8.1. Copy and paste session highlighted in yellow to the new file: /var/db/mysql/my.cnf

```
[server]
        skip-networking
        skip-name-resolve
        innodb_flush_method = O_DIRECT
        skip-innodb_doublewrite
        innodb_file_per_table
```

root@freebsd-1:/ # **joe /var/db/mysql/my.cnf**

8.2. Restart MySQL

root@freebsd-1:/ # **service mysql-server restart**

```
Stopping mysql.
Waiting for PIDS: 1652.
Starting mysql.
```

8.3. Configure fcgiwrap

Note: I'll be using 10 cameras, so setting sysrc fcgiwrap_flags="-c 10”

```
root@freebsd-1:/usr/local/etc/nginx # sysrc fcgiwrap_user="www"
fcgiwrap_user:  -> www
root@freebsd-1:/usr/local/etc/nginx # sysrc fcgiwrap_flags="-c 10"
fcgiwrap_flags:  -> -c 10
root@freebsd-1:/usr/local/etc/nginx # service fcgiwrap restart
fcgiwrap not running? (check /var/run/fcgiwrap/fcgiwrap.pid).
Starting fcgiwrap.
root@freebsd-1:/usr/local/etc/nginx #
```

8.4. Map /tmp to tmpfs

Edit /etc/fstab and add the line below at the end of the file:

```
tmpfs                   /tmp            tmpfs   rw,nosuid,mode=01777    0       0
```

root@freebsd-1:/ # **joe /etc/fstab**

```
# Device        Mountpoint      FStype  Options Dump    Pass#
/dev/ada0p2     /               ufs     rw      1       1
/dev/ada0p3     none            swap    sw      0       0
tmpfs               /tmp            tmpfs   rw,nosuid,mode=01777    0       0
```

8.5. Create user and database *(Note the for this example I'm not changing the user or password. You should change it to something only your know, but there are other steps involved to change other configuration files I won't be covering here)*

root@freebsd-1:/ # **mysql -u root -p**
Enter password: **← type your sql password**

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.30 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> **CREATE DATABASE zm;**
Query OK, 1 row affected (0.01 sec)
mysql> **GRANT ALL PRIVILEGES ON zm.* TO 'zmuser'@'localhost' IDENTIFIED BY 'zmpass';**
Query OK, 0 rows affected (0.02 sec)
mysql> **FLUSH PRIVILEGES;**
Query OK, 0 rows affected (0.00 sec)
mysql> **quit;**
Bye
root@freebsd-1:/ #

8.6. Populate the database

root@freebsd-1:/ # **mysql -u root -p zm < /usr/local/share/zoneminder/db/zm_create.sql**
Enter password:
root@freebsd-1:/ #

8.7. Start zoneminder

root@freebsd-1:/usr/local/etc/nginx # **service zoneminder start**
root@freebsd-1:/usr/local/etc/nginx #

root@freebsd-1:/usr/local/etc/nginx # service zoneminder status
running

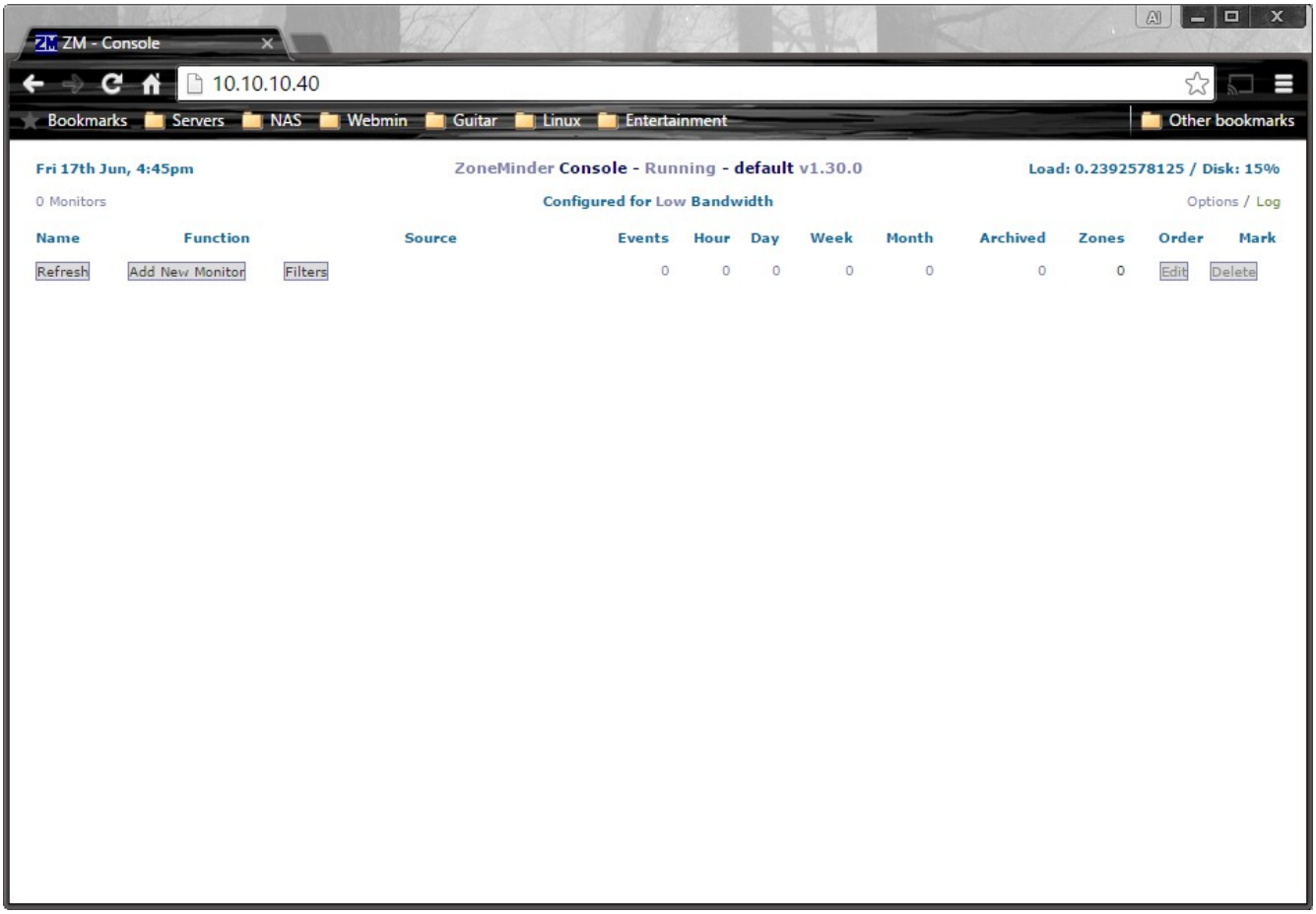
9. Accesssing zoneminder

9.1. Obtain the IP address for your server

You can use any method for this. Below ifconfig showing the interface ID address.

```
root@freebsd-1:/ # ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:8e:96:13
        inet 10.10.10.40 netmask 0xffffff00 broadcast 10.10.10.255
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=600003<RXCSUM,TXCSUM,RXCSUM_IPV6,TXCSUM_IPV6>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
```

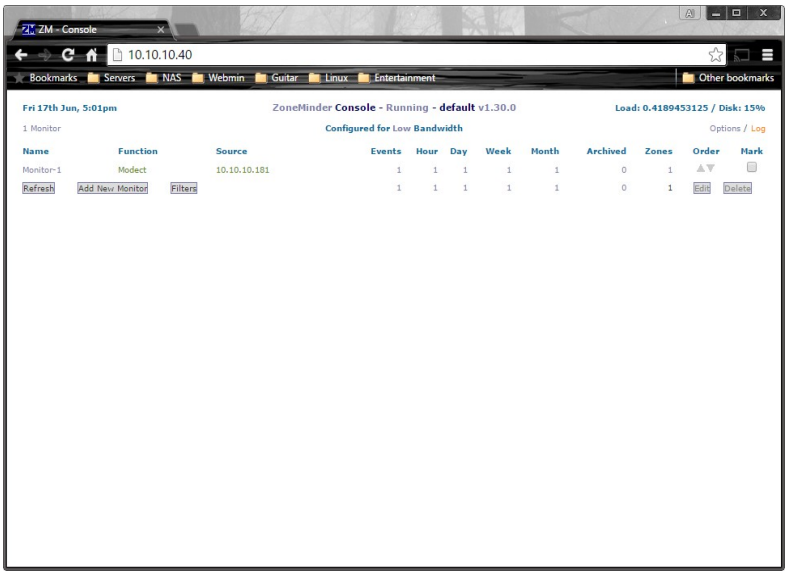
9.2. Type the address on your browser. You should get the screen below:



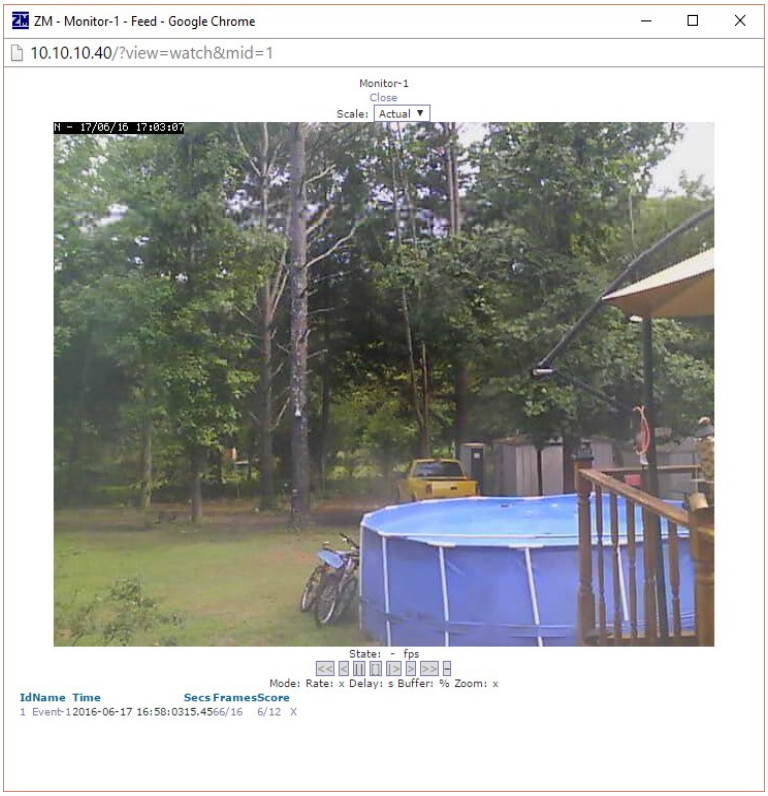
zoneminder running with one camera added

Screens

zoneminder showing one event forced by me moving the camera on my windows



zoneminder camera-1 screen with captured motion event



Final thoughts:

My thanks to Abi (bsd@abinet.ru) and Josua for putting up with my stupid questions while writing this who-to.
Special thanks to Abi that spent couple hours chatting in skype to get the final version running.

This how-to was written for someone, from someone, with no knowledge of FreeBSD like me, so some steps will make people LOL. Have in mind that you still need to know how to install and access a FreeBSD server and be comfortable working on its shell. If not google is your best friend, as it is mine.

The main reason it didn't run the first time were:

- Missing dependencies: That was fixed with the addition of all on item 3.1.
- nginx.conf: Missing configuration added to final version on current how-to.
- php56-gd: Added to allow communication with zmninja