Universidad Nacional
Autónoma de México

Plan de Becarios en Seguridad Informática

Seguridad Perimetral

Exámen Práctico

Autores:
López Matías Alexis Brayan
Luna Castañeda Abraham Iván
Martínez Ríos Ricardo

Profesores:
Sergio Anduin Tovar Balderas

21 de octubre de 2020

Índice

1. Syslog

Pare empezar, debemos instalar el paquete rsyslog o corroborar que ya se tiene instalado

```
root@syslog:~# apt install rsyslog
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
rsyslog is already the newest version (8.1901.0-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Después procedemos a crear el archivo de configuración para la recepción de las bitácoras de los servidores. Se creará en /etc/rsyslog.d/01-logs.conf.
Definimos que se usarán los protocolos TCP y UDP, por el puerto 514, el archivo en donde se escribirá si se detectan las etiquetas que identifican a cada servidor con el log que recibimos de este y que además se almacenará todo lo que se reciba en una bitácora llamada /opt/bitacoras/$NombreDeEquipo/$NombreDeLog.log.

```
root@syslog:~# cat /etc/rsyslog.d/01-logs.conf
module(load="imtcp")
input(type="imtcp" port="514")
module(load="imudp")
input(type="imudp" port="514")

$template AllLogs, "/opt/bitacoras/all.log"
if $programname == 'ServidorWeb1-httpd' then /opt/bitacoras/ServidorWeb1/httpd.log
if $programname == 'ServidorWeb2-httpd' then /opt/bitacoras/ServidorWeb2/httpd.log
if $programname == 'ServidorAuth-ldap' then /opt/bitacoras/ServidorAuth/ldap.log
if $programname == 'ServidorMail-error' then /opt/bitacoras/ServidorMail/error.log
if $programname == 'ServidorDB-error' then /opt/bitacoras/ServidorDB/error.log
if $programname == 'ServidorDHCP-error' then /opt/bitacoras/ServidorDHCP/error.log
if $programname == 'ServidorBalanceador-waf' then /opt/bitacoras/ServidorBalanceador/waf.log
```

Reiniciamos el servicio, y corroboramos que este funcionando

```
root@syslog:~# systemctl status rsyslog
* rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-10-20 23:23:40 CDT; 43s ago
     Docs: man:rsyslogd(8)
           https://www.rsyslog.com/doc/
 Main PID: 535 (rsyslogd)
    Tasks: 10 (limit: 515)
   Memory: 1.4M
   CGroup: /system.slice/rsyslog.service
           `-535 /usr/sbin/rsyslogd -n -iNONE

Oct 20 23:23:40 syslog.local systemd[1]: Starting System Logging Service...
Oct 20 23:23:40 syslog.local systemd[1]: Started System Logging Service.
Oct 20 23:23:40 syslog.local rsyslogd[535]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd.  [v8.1901.0]
Oct 20 23:23:40 syslog.local rsyslogd[535]:  [origin software="rsyslogd" swVersion="8.1901.0" x-pid="535" x-info="https://www.rsyslog.com"] start
```

1.1. Clientes

Debemos crear el archivo de configuración /etc/rsyslog.d/01-client.conf y definir el log que se tomará, la etiqueta que se le pondrá y a donde se mandará y con que protocolo (@@ TCP @ UDP). En el siguiente ejemplo usaremos al servidor web

```
module(load="imfile" PollingInterval="10")

input(type="imfile"
File="/var/log/apache2/error.log"
Tag="ServidorWeb1-httpd"
Severity="info")

if $programname == 'ServidorWeb1-httpd' then @@192.168.50.30:514
```

Reiniciamos rsyslog y notamos que nuestro servidor ya esta recibiendo los logs

```
root@syslog:/opt/bitacoras# cat ServidorWeb1/httpd.log
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:33.419794 2020] [mpm_event:notice]
 [pid 770:tid 139918236820608] AH00489: Apache/2.4.38 (Debian) configured -- resuming normal operati
ons
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:33.419869 2020] [mpm_event:info] [
pid 770:tid 139918236820608] AH00490: Server built: 2020-08-25T20:08:29
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:33.419879 2020] [core:notice] [pid
 770:tid 139918236820608] AH00094: Command line: '/usr/sbin/apache2'
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:43.011163 2020] [core:info] [pid 7
70:tid 139918236820608] AH00096: removed PID file /var/run/apache2/apache2.pid (pid=770)
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:43.011179 2020] [mpm_event:notice]
 [pid 770:tid 139918236820608] AH00491: caught SIGTERM, shutting down
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:43.047975 2020] [mpm_event:notice]
 [pid 840:tid 140236690486400] AH00489: Apache/2.4.38 (Debian) configured -- resuming normal operati
ons
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:43.048044 2020] [mpm_event:info] [
pid 840:tid 140236690486400] AH00490: Server built: 2020-08-25T20:08:29
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:24:43.048053 2020] [core:notice] [pid
 840:tid 140236690486400] AH00094: Command line: '/usr/sbin/apache2'
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:28:34.803774 2020] [core:info] [pid 8
40:tid 140236690486400] AH00096: removed PID file /var/run/apache2/apache2.pid (pid=840)
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:28:34.803794 2020] [mpm_event:notice]
 [pid 840:tid 140236690486400] AH00491: caught SIGTERM, shutting down
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:28:34.842951 2020] [mpm_event:notice]
 [pid 924:tid 140469455930496] AH00489: Apache/2.4.38 (Debian) configured -- resuming normal operati
ons
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:28:34.843020 2020] [mpm_event:info] [
pid 924:tid 140469455930496] AH00490: Server built: 2020-08-25T20:08:29
Oct 21 02:44:41 abraham-luna ServidorWeb1-httpd [Wed Oct 21 02:28:34.843031 2020] [core:notice] [pid
 924:tid 140469455930496] AH00094: Command line: '/usr/sbin/apache2'
```

Tomando ahora como ejemplo a LDAP debemos crear el archivo de configuración /etc/rsyslog.d/01-client.conf y definir el log que se tomará, en este caso mandaremos todos los que contengan ldap en el tag de la bitácora

```
root@ldap:~# cat /etc/rsyslog.d/01-client.conf
if $programname contains 'ldap' then @@192.168.50.30:514
```

El server recibe los logs

```
root@syslog:/opt/bitacoras# cat ServidorAuth/ldap.log
Oct 21 03:27:37 ldap ldapwhoami: DIGEST-MD5 common mech free
```

De esta forma se configuran todos los clientes para mandar las bitácoras que se deseen al servidor de syslog

2. Nagios

Empezaremos por instalar algunos paquetes en nuestro servidor que son necesarios para tener funcionando al Nagios, descomprimir paquetes descargados, para compilar, para tener el servidor web y los módulos de php que se requieren.

apt-get install build-essential unzip libssl-dev apache2 php libapache2-mod-php php-gd libgd-dev

Después descargaremos el paquete Nagios en /tmp y lo descomprimiremos

wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz -P /tmp
tar xzf nagios-4.4.6.tar.gz

```
root@nagios:/tmp# wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2020-10-20 00:22:10--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 2600:3c00::f03c:91ff:fedf:b821, 72.14.181.71
Connecting to assets.nagios.com (assets.nagios.com)|2600:3c00::f03c:91ff:fedf:b821|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz              100%[=============================================================>]  10.81M  26.2MB/s    in 0.4s

2020-10-20 00:22:11 (26.2 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

root@nagios:/tmp# ls
nagios-4.4.6.tar.gz
```

```
root@nagios:/tmp# tar xzf nagios-4.4.6.tar.gz
root@nagios:/tmp# cd nagios-4.4.6/
```

Ahora entraremos al directorio de nagios y empezaremos a compilar Nagios. Lo primero es configurarlo con

./configure –with-httpd-conf=/etc/apache2/sites-enabled

```
Creating sample config files in sample-config/ ...


*** Configuration summary for nagios 4.4.6 2020-04-28 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagios
            Event Broker:  yes
         Install ${prefix}:  /usr/local/nagios
     Install ${includedir}:  /usr/local/nagios/include/nagios
                 Lock file:  /run/nagios.lock
    Check result directory:  /usr/local/nagios/var/spool/checkresults
            Init directory:  /lib/systemd/system
  Apache conf.d directory:  /etc/apache2/sites-enabled
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/sbin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

Ahora, compilaremos con

make all

```
*** Support Notes *******************************************

If you have questions about configuring or running Nagios,
please make sure that you:

     - Look at the sample config files
     - Read the documentation on the Nagios Library at:
          https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

     - What version of Nagios you are using
     - What version of the plugins you are using
     - Relevant snippets from your config files
     - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

************************************************************

Enjoy.
```

Para crear el grupo y usuario nagios y añadir al usuario que identifica al servidor web al grupo nagios recién creado con usamos

make install-groups-users

```
root@nagios:/tmp/nagios-4.4.6# make install-groups-users
groupadd -r nagios
useradd -g nagios nagios
```

Instalamos los archivos de Nagios core

make install

```
*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme

make[1]: Leaving directory '/tmp/nagios-4.4.6'
make install-basic
make[1]: Entering directory '/tmp/nagios-4.4.6'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/archives
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/spool/checkresults
chmod g+s /usr/local/nagios/var/spool/checkresults

*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc
```

Instalamos los archivos con la configuración por defecto de Nagios Core

make install-config

```
root@nagios:/tmp/nagios-4.4.6# make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cf
g
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.
```

Instalamos los scripts del servicio

make install-init

```
root@nagios:/tmp/nagios-4.4.6# make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
```

Habilitamos el servicio Nagios para su inicio automático con cada arranque

make install-daemoninit

```
root@nagios:/tmp/nagios-4.4.6# make install-daemoninit
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /lib/systemd/system/nagios.service.

*** Init script installed ***
```

Configuramos el directorio para comandos externos

make install-commandmode

```
root@nagios:/tmp/nagios-4.4.6# make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

Añadimos los archivos de configuración necesarios para el servidor web

make install-webconf

```
root@nagios:/tmp/nagios-4.4.6# make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/apache2/sites-enabled/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***
```

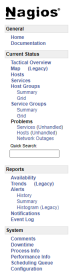
Debemos habilitar el modulo cgi de apache

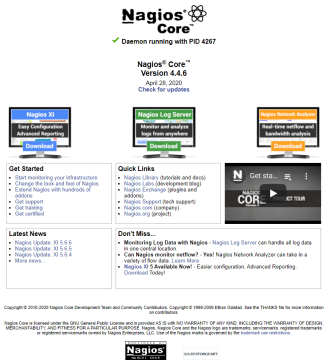
```
root@nagios:/tmp/nagios-4.4.6# a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Ahora debemos crear un usuario y contraseña para administrar el nagios

```
root@nagios:/tmp/nagios-4.4.6# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Al entrar a https://NagiosIP/nagios nos autenticaremos y veremos nuestro nagios funcionando





Ahora pasaremos a instalar nagiosgraph para poder tener gráficas de nuestros servicios. Lo primero es descargar el comrpimido de nagios graph y descomprimirlo

wget
'https://downloads.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz' -O
nagiosgraph-1.5.2.tar.gz
tar -xvf nagiosgraph-1.5.2.tar.gz

```
root@nagios:~# wget 'https://downloads.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz' -O nagiosgraph-1.5.2.tar.gz
--2020-10-20 13:21:00--  https://downloads.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 216.105.38.13
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|216.105.38.13|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://iweb.dl.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz [following]
--2020-10-20 13:21:01--  https://iweb.dl.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz
Resolving iweb.dl.sourceforge.net (iweb.dl.sourceforge.net)... 2607:f748:10:12::5f:2, 192.175.120.182
Connecting to iweb.dl.sourceforge.net (iweb.dl.sourceforge.net)|2607:f748:10:12::5f:2|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 329978 (322K) [application/x-gzip]
Saving to: 'nagiosgraph-1.5.2.tar.gz'

nagiosgraph-1.5.2.tar.gz                100%[=======================================================================>] 322.24K   864KB/s    in 0.4s

2020-10-20 13:21:02 (864 KB/s) - 'nagiosgraph-1.5.2.tar.gz' saved [329978/329978]
```

```
root@nagios:~# tar -xvf nagiosgraph-1.5.2.tar.gz
```

Ahora debemos instalar los paquetes necesarios para que nagiosgraph funcione

apt-get install -y whois mrtg libcgi-pm-perl librrds-perl libgd-perl libnagios-object-perl

Definimos variables de ambiente que se usarán durante la instalación automatizada de nagiosgraph como la ruta del archivo de configuración de nagios, la ruta de la definición de comandos de nagios, la ruta de sitios de apache,etc...

```
root@nagios:~# export NG_PREFIX=/etc/nagiosgraph
root@nagios:~# export NG_MODIFY_NAGIOS_CONFIG=y
root@nagios:~# export NG_NAGIOS_CONFIG_FILE=/usr/local/nagios/etc/nagios.cfg
root@nagios:~# export NG_NAGIOS_COMMANDS_FILE=/usr/local/nagios/etc/objects/commands.cfg
root@nagios:~# export NG_MODIFY_APACHE_CONFIG=y
root@nagios:~# export NG_APACHE_CONFIG_DIR=/etc/apache2/sites-available
root@nagios:~# export NG_APACHE_CONFIG_FILE=nagiosgraph.conf
```

Entramos a la carpeta que fue descomprimida e iniciamos la instalación automatizada

./install.pl

```
root@nagios:~/nagiosgraph-1.5.2# ./install.pl
checking required PERL modules
  Carp ... 1.50
  CGI ... 4.40
  Data::Dumper ... 2.170
  Digest::MD5 ... 2.55
  File::Basename ... 2.85
  File::Find ... 1.34
  MIME::Base64 ... 3.15
  POSIX ... 1.84
  RRDs ... 1.5001
  Time::HiRes ... 1.9759
checking optional PERL modules
  GD ... 2.71
  Nagios::Config ... 36
checking nagios installation
  found nagios exectuable at /usr/local/nagios/bin/nagios
checking web server installation
  found apache executable at /usr/sbin/apache2
```

Ahora debemos configurar nuestro web server para nagiosgraph, la configuración se hace en /etc/apache2/sites-available/nagiosgraph.conf y queda de la siguiente manera

```
root@nagios:~# cat /etc/apache2/sites-available/nagiosgraph.conf

ScriptAlias /nagiosgraph/cgi-bin /etc/nagiosgraph/cgi
<Directory /etc/nagiosgraph/cgi>
    Options ExecCGI
    AllowOverride None
    Require all granted
</Directory>
Alias /nagiosgraph /etc/nagiosgraph/share
<Directory /etc/nagiosgraph/share>
    Options None
    AllowOverride None
    Require all granted
</Directory>
```

Habilitamos el sitio

```
root@nagios:~# a2ensite nagiosgraph.conf
Enabling site nagiosgraph.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

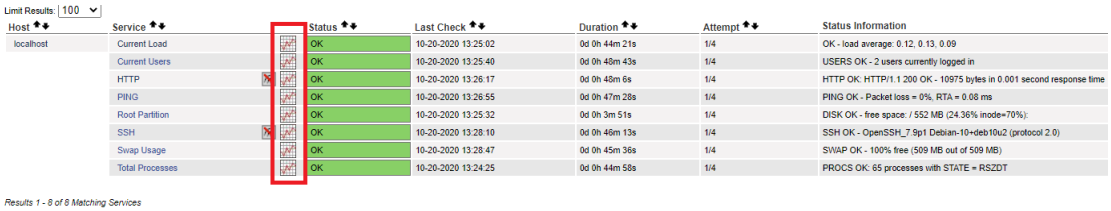
Debemos definir la plantilla para usar la graficación en el monitoreo de servicios. Esto se hace en /usr/local/nagios/etc/objects/templates.cfg y queda de la siguiente manera

```
define service {
    name            graphed-service
    action_url      /nagiosgraph/cgi-bin/show.cgi?host=$HOSTNAME$&service=$SERVICEDESC$' onMouseOver='showGraphPopup(this)' onMouseOut='hideGraphPopup()' rel='/nagiosgraph/cgi-bin/showgraph.cgi?host=$HOSTNAME$&service=$SERVICEDESC$&period=week&rrdopts=-w+450+-j
    register        0
}
```

Para añadir la graficación a un servicio solo debemos añadir esta plantilla creada en la definición del servicio que se hace en /usr/local/nagios/etc/objects/$HOSTFILE.cfg

```
# Define a service to check the load on the local machine.

define service {

    use                     local-service,graphed-service          ; Name of service template to use
    host_name               localhost
    service_description     Current Load
    check_command           check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}
```

Reiniciamos apache y nagios. Ahora notamos que nuestros servicios muestran un icono que indica que están siendo graficados



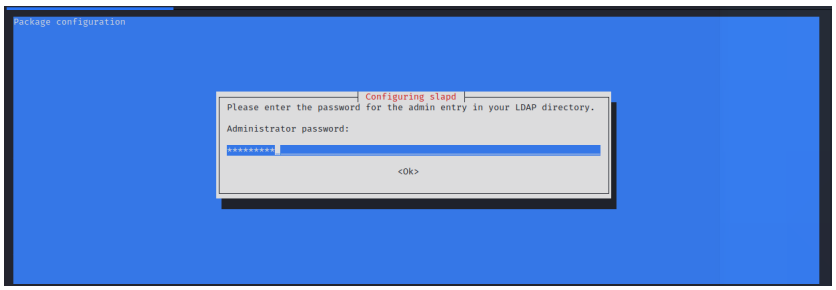
3. LDAP & LDAPS

3.1. LDAP

Se instalan paquetes LDAP con

```
apt -y install slapd ldap-utils ldapscripts
```

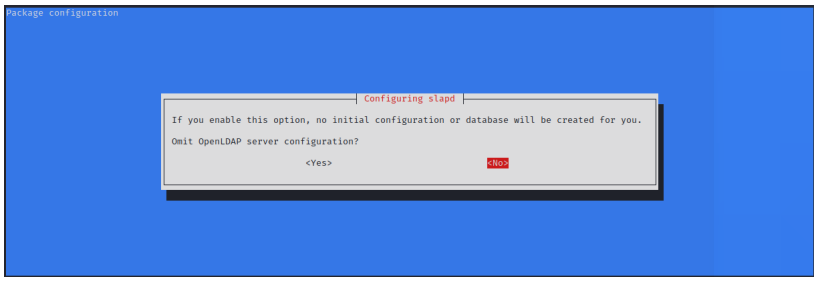
Durante la instalación, se le solicitará que se configure la contraseña de administrador LDAP



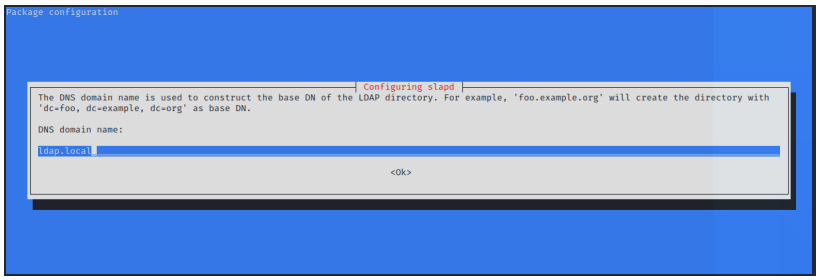
Se reconfigura slapd con

dpkg-reconfigure slapd

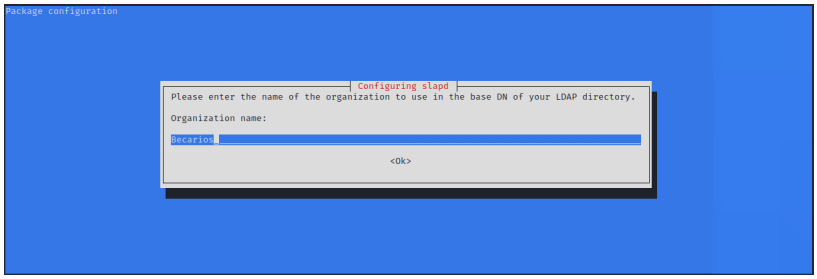
Cuando se ejecuta el comando, se pregunta si debe omitir la configuración del servidor OpenLDAP. Seleccionamos No para que se cree la configuración.



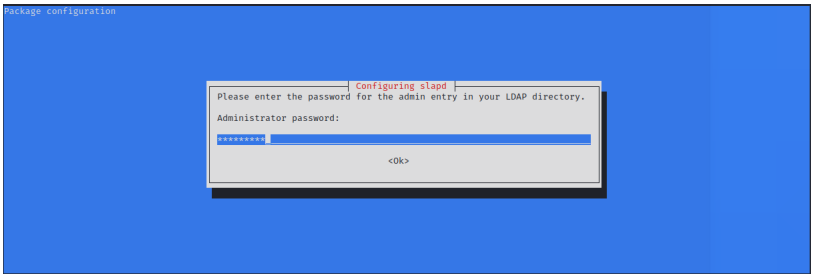
A continuación, configuramos el nombre de dominio completo del servidor OpenLDAP que se utilizará para crear el DN base.



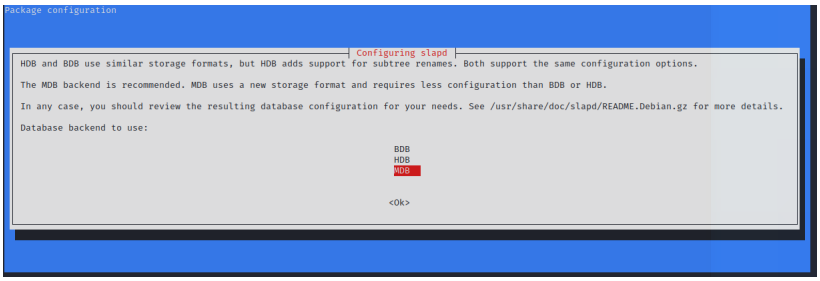
Establecemos el nombre de la organización

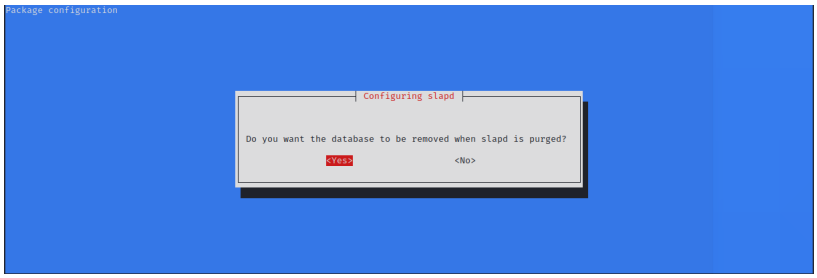


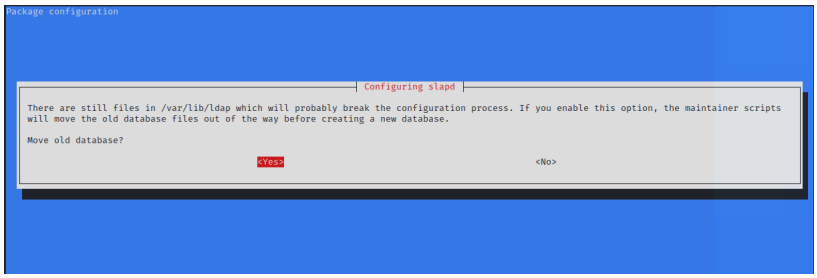
Configuramos y verificamos el password de administrador.



Seleccionamos el backend de la base de datos OpenLDAP. MDB es el tipo recomendado







Para verificar la reconfiguración, simplemente ejecutamos *slapcat*.

```
root@debian:~# slapcat
dn: dc=ldap,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: Becarios
dc: ldap
structuralObjectClass: organization
entryUUID: b61b90ee-9c82-103a-8155-473ff0c53518
creatorsName: cn=admin,dc=ldap,dc=local
createTimestamp: 20201007004933Z
entryCSN: 20201007004933.915901Z#000000#000#000000
modifiersName: cn=admin,dc=ldap,dc=local
modifyTimestamp: 20201007004933Z

dn: cn=admin,dc=ldap,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9TTZiQ3cwNVZvUXlEdTJheSs2N3pucThTc3NzZHdSeTk=
structuralObjectClass: organizationalRole
entryUUID: b61bc208-9c82-103a-8156-473ff0c53518
creatorsName: cn=admin,dc=ldap,dc=local
createTimestamp: 20201007004933Z
entryCSN: 20201007004933.917200Z#000000#000#000000
modifiersName: cn=admin,dc=ldap,dc=local
modifyTimestamp: 20201007004933Z
```

En este momento ya tenemos un servidor LDAP funcional. Depende del lector dar de alta usuarios con contraseñas

3.2. LDAPS

Se usaron certificados autofirmados para este proyecto.
Para configurar el servidor OpeLDAP con certificado SSL / TLS, necesita un certificado CA, un certificado de servidor y un archivo de clave de certificado de servidor.
Creamos directorios para almacenar dichos certificados

mkdir -p /etc/ssl/openldap/{private,certs,newcerts}

Una vez creados los directorios anteriores, editamos el archivo de configuración /usr/lib/ssl/openssl.cnf y configuramos el directorio para almacenar certificados y claves SSL / TLS en la sección [CA_default].

```
[ CA_default ]

#dir            = ./demoCA              # Where everything is kept
dir             = /etc/ssl/openldap
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
```

También necesitamos algunos archivos para rastrear los certificados firmados.

echo ”1001” > /etc/ssl/openldap/serial
touch /etc/ssl/openldap/index.txt

Creamos un archivo de clave CA con

openssl genrsa -aes256 -out /etc/ssl/openldap/private/cakey.pem 2048

Cuando se le solicite, ingresamos la frase de contraseña. Esta la eliminaremos con

openssl rsa -in /etc/ssl/openldap/private/cakey.pem -out
/etc/ssl/openldap/private/cakey.pem

```
root@ldap:~# openssl genrsa -aes256 -out /etc/ssl/openldap/private/cakey.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................................................+++++
.............................+++++
e is 65537 (0×010001)
Enter pass phrase for /etc/ssl/openldap/private/cakey.pem:
Verifying - Enter pass phrase for /etc/ssl/openldap/private/cakey.pem:
```

```
root@ldap:~# openssl rsa -in /etc/ssl/openldap/private/cakey.pem -out /etc/ssl/openldap/private/cakey.pem
Enter pass phrase for /etc/ssl/openldap/private/cakey.pem:
writing RSA key
```

Creamos el certificado CA

openssl req -new -x509 -days 3650 -key /etc/ssl/openldap/private/cakey.pem -out
/etc/ssl/openldap/certs/cacert.pem

```
root@ldap:~# openssl req -new -x509 -days 3650 -key /etc/ssl/openldap/private/cakey.pem -out /etc/ssl/openldap/certs/cacert.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:CDMX
Locality Name (eg, city) []:CDMX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mafia
Organizational Unit Name (eg, section) []:Becario
Common Name (e.g. server FQDN or YOUR name) []:Bec
Email Address []:
```

Luego, la llave para el servidor LDAP

openssl genrsa -aes256 -out /etc/ssl/openldap/private/ldapserver-key.key 2048

```
root@ldap:~# openssl genrsa -aes256 -out /etc/ssl/openldap/private/ldapserver-key.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.........................+++++
........+++++
e is 65537 (0×010001)
Enter pass phrase for /etc/ssl/openldap/private/ldapserver-key.key:
Verifying - Enter pass phrase for /etc/ssl/openldap/private/ldapserver-key.key:
```

Removemos la frase de contraseña que pide

openssl rsa -in /etc/ssl/openldap/private/ldapserver-key.key -out
/etc/ssl/openldap/private/ldapserver-key.key

```
root@ldap:~# openssl rsa -in /etc/ssl/openldap/private/ldapserver-key.key -out /etc/ssl/openldap/private/ldapserver-key.key
Enter pass phrase for /etc/ssl/openldap/private/ldapserver-key.key:
writing RSA key
```

Generamos la solicitud de firma de certificado. Hay que configurar los mismos detalles que se utilizaron al generar el archivo de certificado de CA anterior

openssl req -new -key /etc/ssl/openldap/private/ldapserver-key.key -out
/etc/ssl/openldap/certs/ldapserver-cert.cs

```
root@ldap:~# openssl req -new -key /etc/ssl/openldap/private/ldapserver-key.key -out /etc/ssl/openldap/certs/ldapserver-cert.cs
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:CDMX
Locality Name (eg, city) []:CDMX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mafia
Organizational Unit Name (eg, section) []:Becario
Common Name (e.g. server FQDN or YOUR name) []:Bec
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Generamos el certificado del servidor LDAP y lo firmamos con la clave CA y el certificado generado anteriormente.

openssl ca -keyfile /etc/ssl/openldap/private/cakey.pem -cert /etc/ssl/openldap/certs/cacert.pem -in /etc/ssl/openldap/certs/ldapserver-cert.csr -out /etc/ssl/openldap/certs/ldapserver-cert.crt

```
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0×1001)
        Validity
            Not Before: Oct 21 05:20:58 2020 GMT
            Not After : Oct 21 05:20:58 2021 GMT
        Subject:
            countryName               = MX
            stateOrProvinceName       = CDMX
            organizationName          = Mafia
            organizationalUnitName    = Becarios
            commonName                = Becario
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                62:B8:C8:DF:41:6B:A3:28:54:C8:0E:EB:CF:26:EA:EB:58:09:7B:D4
            X509v3 Authority Key Identifier:
                keyid:97:8E:42:89:92:6C:45:32:53:D4:38:DD:AA:17:DD:6A:B9:43:7D:9C

Certificate is to be certified until Oct 21 05:20:58 2021 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
/etc/ssl/openldap/certs/ldapserver-cert.crt: OK
```

Verificamos el servidor LDAP con la CA

openssl verify -CAfile /etc/ssl/openldap/certs/cacert.pem /etc/ssl/openldap/certs/ldapserver-cert.crt

A continuación, establecemos la propiedad del directorio de certificados OpenLDAP a el usuario openldap

chown -R openldap: /etc/ssl/openldap/

Ahora debemos actualizar los certificados TLS de OpenLDAP Server. Por lo tanto, creamos un archivo LDIF para definir los atributos TLS.

vim ldap-tls.ldif

```
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/openldap/certs/cacert.pem
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/openldap/certs/ldapserver-cert.crt
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/openldap/private/ldapserver-key.key
```

Modificamos estos datos en LDAP con

ldapmodify -Y EXTERNAL -H ldapi:/// -f ldap-tls.ldif

Por último editamos el archivo de configuración /etc/ldap/ldap.conf y cambiamos la ubicación del certificado CA

```
...
# TLS certificates (needed for GnuTLS)
#TLS_CACERT     /etc/ssl/certs/ca-certificates.crt
TLS_CACERT      /etc/ssl/openldap/certs/cacert.pem
```

y reiniciamos el servicio

systemctl restart slapd

3.3. Verificar

Para verificar el funcionamiento de nuestro servidor podemos usar

ldapwhoami -H ldap:/$LDAP_IP-x -ZZ

```
root@ldap:/etc/ssl/openldap/certs# ldapwhoami -H ldap://localhost -x -ZZ
anonymous
```

Si se usa desde el mismo host, obtendremos *anonymous* como salida pues la autenticación anonima esta habilitada y esto indicará que nuestra conexión con certificados funciona. Si omitimos la bandera ZZ, la conexión que se probará será sin certificados.
Para crear usuarios primero debemos crear un grupo al cual añadirlos. Asi que creamos el archivo /etc/ldap/users.ldif y agregamos una unidad organizacional. Despues la añadimos a la base de datos ldap con el comando que se ve en la imagen

```
root@ldap:~# cat /etc/ldap/users.ldif
dn: ou=Servers,dc=mafia,dc=local
objectClass: organizationalUnit
ou: Servers
root@ldap:~# ldapadd -D "cn=admin,dc=mafia,dc=local" -W -H ldapi:/// -f /etc/ldap/users.ldif
Enter LDAP Password:
adding new entry "ou=Servers,dc=mafia,dc=local"
```

Ahora pasaremos a crear un usuario, para ello creamos el archivo /etc/ldap/new_users.ldif y lo llenamos con los datos del usuario a crear. Despues lo añadimos a la base con el comando mostrado

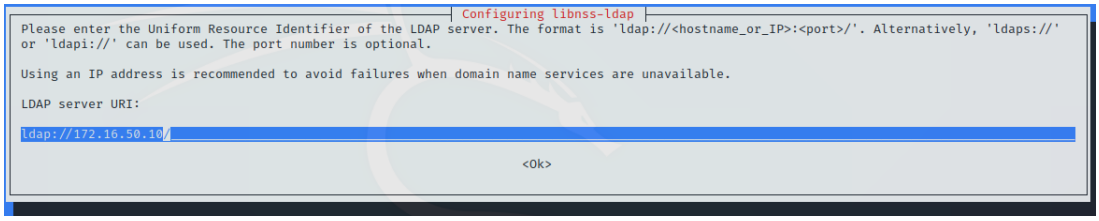
```
root@ldap:~# cat /etc/ldap/new_users.ldif
# Content of new_users LDIF file

dn: cn=abraham,ou=Servers,dc=mafia,dc=local
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: abraham
uid: abraham
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/abraham
userPassword: hola123.,
loginShell: /bin/bash
root@ldap:~# ldapadd -D "cn=admin,dc=mafia,dc=local" -W -H ldapi:/// -f /etc/ldap/new_users.ldif
Enter LDAP Password:
adding new entry "cn=abraham,ou=Servers,dc=mafia,dc=local"
```

3.4. Clientes

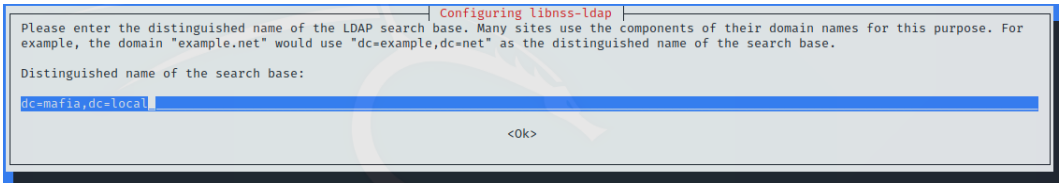
Para configurar un cliente para que use autenticación con ldap, primero debemos descargar los paquetes necesarios.

```
apt install libnss-ldap libpam-ldap ldap-utils
```

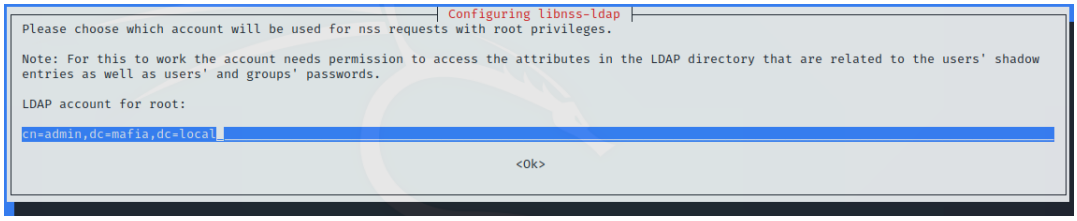
Esto iniciará un asistente donde tendremos que pedirá la ip del servidor ldap

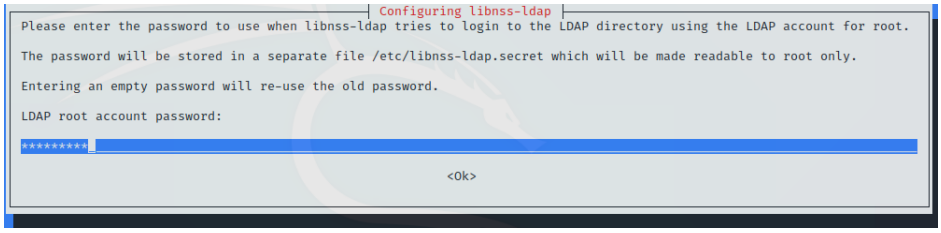


El distinguished name

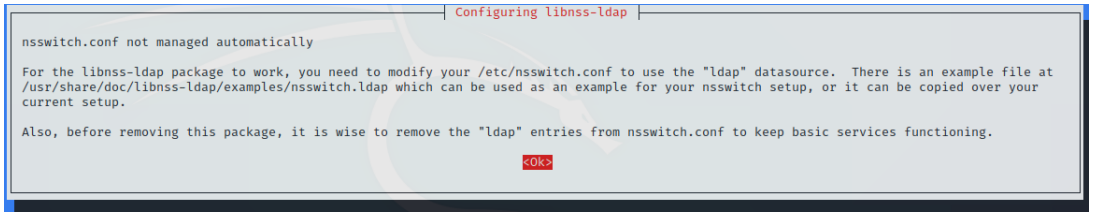


La cuenta con permisos administrativos de ldap y su contraseña

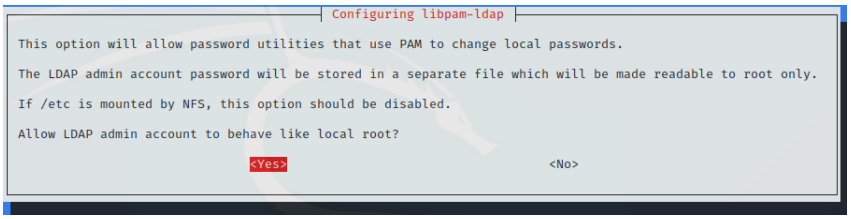




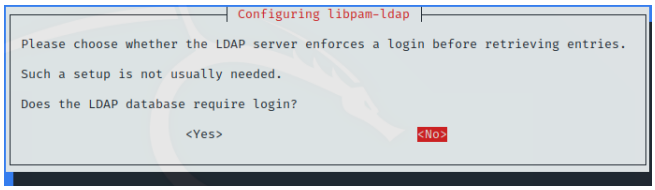
Segumos, aqui solo muestra una advertencia



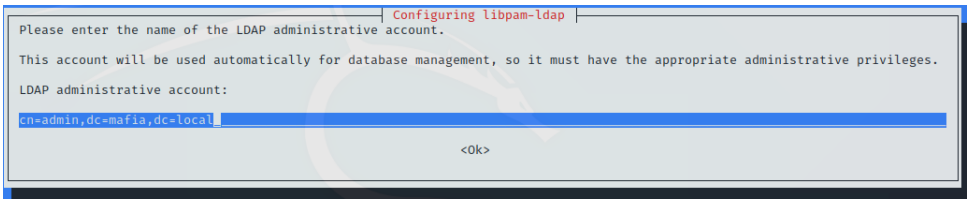
Se preguntará si deseamos que la raíz local sea el administrador de la base de datos. Ponemos Si en esta opción ya que queremos cambiar la contraseña de usuario directamente desde la máquina host. Con esta opción, podremos ejecutar passwd y modificar la contraseña directamente en el directorio LDAP.

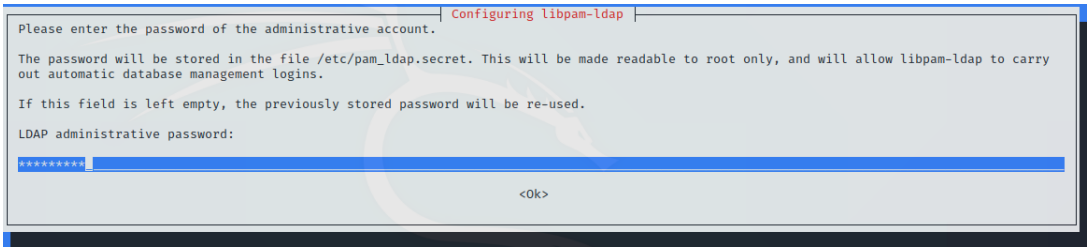


Por default no se requiere login, asi que lo dejamos asi



Volvemos a ingresar permisos administrativos de ldap





Para vincular la información de el cliente al directorio LDAP, debemos modificar el archivo nsswitch. Editamos el archivo /etc/nsswitch.conf y agregamos una entrada ldap a las primeras cuatro secciones

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd: files systemd ldap
group: files systemd ldap
shadow: files ldap
gshadow: files ldap

hosts: files dns
networks: files

protocols: db files
services: db files
ethers: db files
rpc: db files

netgroup: nis
```

También el archivo /etc/pam.d/common-session, añadiendo el modulo pam_mkhomedir para crear directorios hogar de usuarios ldap

```
# here are the per-package modules (the "Primary" block)
password        [success=2 default=ignore]      pam_unix.so obscure sha512
password        [success=1 user_unknown=ignore default=die]     pam_ldap.so use_authtok try_first_pass
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

Vemos como ingresamos al sistema con un usuario ldap y se crea el home

```
Debian GNU/Linux 10 debian tty1

debian login: abraham
Password:
Linux debian 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Creating directory '/home/abraham'.
abraham@debian:~$
```

4. DNS

El nombre del paquete del servidor DNS en Debian es bind9 y está disponible en el repositorio base.

sudo apt install bind9

/etc/bind es el directorio de configuración de bind9, contiene archivos de configuración y archivos de búsqueda de zona. El archivo de configuración global es /etc/bind/named.conf
Comenzamos por crear una foward zone y una reverse zone

sudo nano /etc/bind/named.conf.local

```
//
// Do any local configuration here
//
zone "mafia.local" IN { //Domain name

     type master; //Primary DNS

     file "/etc/bind/forward.mafia.local.db"; //Forward lookup file

     allow-update { none; }; // Since this is the primary DNS, it should be none.

};
zone "50.16.172.in-addr.arpa" IN { //Reverse lookup name, should match your network in reverse order

     type master; // Primary DNS

     file "/etc/bind/reverse.mafia.local.db"; //Reverse lookup file

     allow-update { none; }; //Since this is the primary DNS, it should be none.

};
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
~
```

Creamos una lista de control de acceso con los segmentos de red que podrán hacernos consultas

```
acl "trusted" {
        192.168.50.0/24;
        172.16.50.0/24;
        10.10.50.0/24;
        localhost;
        localnets;
};
```

También habilitamos la recursión

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        listen-on-v6 { any; };
        recursion yes;
        allow-query { any; };
        allow-recursion { trusted; };
        allow-query-cache { trusted; };
};
```

Una vez creadas las zonas, creamos archivos de datos de zona. Copiamos la plantilla de muestra al archivo de zona a forward.mafia.local.db y lo editamos

sudo cp /etc/bind/db.local /etc/bind/forward.mafia.local.db
sudo nano /etc/bind/forward.mafia.local.db

```
;
; BIND data file for local loopback interface
;
$ORIGIN mafia.local.
$TTL    604800

@       IN      SOA     ns1.mafia.local. root.mafia.local. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; Commentout below three lines
;@      IN      NS      localhost.
;@      IN      A       127.0.0.1
;@      IN      AAAA    ::1

;Name Server Information

@       IN      NS      ns.mafia.local.
@       IN      NS      ns1.mafia.local.

;IP address of Name Server

ns      IN      A       172.16.50.40
ns1     IN      A       172.16.50.40

;A - Record HostName To Ip Address

@       IN       A      172.16.50.50
mail    IN       A      172.16.50.20
ldap    IN       A      172.16.50.10
db      IN       A      172.16.50.30

;CNAME record

www     IN      CNAME   mafia.local.
ftp     IN      CNAME   ns.mafia.local.

;Mail Exchanger
; MX
@       IN       MX   138   mail.mafia.local.

; SPF
@       IN       TXT  "v=spf1 a mx ~all"
~
```

Para la reverse zone copiamos la plantilla de muestra al archivo de zona reverse.mafia.local.db y lo editamos

sudo cp /etc/bind/db.127 /etc/bind/reverse.mafia.local.db
sudo nano /etc/bind/reverse.mafia.local.db

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     mafia.local. root.mafia.local. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; Commentout below two lines

;@      IN      NS      localhost.
;1.0.0  IN      PTR     localhost.

;Name Server Information

@       IN      NS      ns1.mafia.local.

;Reverse lookup for Name Server

138      IN      PTR    ns1.mafia.local.

;PTR Record IP address to HostName

100     IN      PTR     www.mafia.local.
~
```

Reiniciamos el servicio

sudo systemctl restart bind9

Después establecemos la ip de nuestro dns en el /etc/network/interfaces de las maquinas cliente. En la siguiente imagen se establece la resolución a nuestro dns desde un host en otra subred

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet static
    address 192.168.50.30
    netmask 255.255.255.0
    gateway 192.168.50.2
    dns-nameservers 172.16.50.40
    dns-search mafia.local
~
```

Prueba con nslookup desde la maquina que cuto /etc/network/interfaces acabamos de editar

```
root@monitor-syslog-ntp:~# nslookup unam.mx
Server:         172.16.50.40
Address:        172.16.50.40#53

Non-authoritative answer:
Name:   unam.mx
Address: 132.248.166.19
Name:   unam.mx
Address: 132.248.166.20
Name:   unam.mx
Address: 132.248.166.17
Name:   unam.mx
Address: 132.248.166.18
Name:   unam.mx
Address: 2001:1218:3000:180::18
Name:   unam.mx
Address: 2001:1218:3000:180::17
Name:   unam.mx
Address: 2001:1218:3000:180::19
Name:   unam.mx
Address: 2001:1218:3000:180::20

root@monitor-syslog-ntp:~#
```

5. FTP

Emezamos por instalar vsftp

apt install vsftpd

modificamos el archivo de configuración que se encuetra en /etc/vsftpd.conf para que quede de la siguiente forma

```
root@debian:~# cat /etc/vsftpd.conf
listen=NO
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
idle_session_timeout=600
data_connection_timeout=120
ftpd_banner=FTP de becarios
chroot_local_user=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
pasv_enable=YES
pasv_min_port=10000
pasv_max_port=11000
user_sub_token=$USER
local_root=/home/$USER/ftp
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
```

Reiniciamos el servicio y verificamos su status. Podemos notar que no hay errores en nuestra configuración y que el servidor esta en pleno funcionamiento

```
root@debian:~# systemctl status vsftpd
* vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-10-07 22:56:22 CDT; 4s ago
  Process: 1089 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
 Main PID: 1090 (vsftpd)
    Tasks: 1 (limit: 515)
   Memory: 740.0K
   CGroup: /system.slice/vsftpd.service
           `-1090 /usr/sbin/vsftpd /etc/vsftpd.conf

Oct 07 22:56:22 debian systemd[1]: Starting vsftpd FTP server...
Oct 07 22:56:22 debian systemd[1]: Started vsftpd FTP server.
```

Crearemos un usuario para despues añadirlo a la lista de usuarios que pueden usar ftp. También debemos crearle el directorio ftp a dicho usuario en su /home y cambiar estos permisos a nobody:nogroup

```
root@debian:~# adduser becario
Adding user `becario' ...
Adding new group `becario' (1001) ...
Adding new user `becario' (1001) with group `becario' ...
Creating home directory `/home/becario' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for becario
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

echo usuario_creado >> /etc/vsftpd.userlist
mkdir /home/usuario_creado/ftp
chown nobody:nogroup /home/usuario_creado/ftp

Comprobamos que el servicio es funcional de la siguiente manera

```
root@debian:~# ftp localhost
Connected to localhost.
220 FTP de becarios
Name (localhost:root): becario
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> exit
221 Goodbye.
root@debian:~#
```

6. Squid

6.1. Squid Proxy

Lo primero que se debe hacer es actualizar la información de paquetes:

```
root@dhcp:~# apt-get update
Hit:1 http://mmc.geofisica.unam.mx/debian buster InRelease
Hit:2 http://mmc.geofisica.unam.mx/debian buster-updates InRelease
Get:3 http://security.debian.org/debian-security buster/updates InRelease [65.4 kB]
Fetched 65.4 kB in 1s (81.2 kB/s)
Reading package lists... Done
root@dhcp:~# hostname
dhcp.mafia.local
root@dhcp:~#
```

Después podemos instalar el paquete principal de **squid**:

```
root@dhcp:~# apt-get install  -y squid
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates libdbi-perl libecap3 libgdbm-compat4 libgdbm6 libldap-2.4-2 libldap-common libltdl7 libperl5.28
  libsasl2-2 libsasl2-modules libsasl2-modules-db openssl perl perl-modules-5.28 squid-common squid-langpack
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl libsasl2-modules-gssapi-mit
  | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make libb-debug-perl liblocale-codes-perl squidclient
  squid-cgi squid-purge smbclient ufw winbind
The following NEW packages will be installed:
  ca-certificates libdbi-perl libecap3 libgdbm-compat4 libgdbm6 libldap-2.4-2 libldap-common libltdl7 libperl5.28
  libsasl2-2 libsasl2-modules libsasl2-modules-db openssl perl perl-modules-5.28 squid squid-common squid-langpack
0 upgraded, 18 newly installed, 0 to remove and 1 not upgraded.
Need to get 13.0 MB of archives.
After this operation, 66.0 MB of additional disk space will be used.
```

```
Updating certificates in /etc/ssl/certs...
126 added, 0 removed; done.
Setting up libgdbm-compat4:amd64 (1.18.1-4) ...
Setting up libperl5.28:amd64 (5.28.1-6+deb10u1) ...
Setting up perl (5.28.1-6+deb10u1) ...
Setting up libdbi-perl:amd64 (1.642-1+deb10u1) ...
Setting up squid (4.6-1+deb10u4) ...
Setcap worked! /usr/lib/squid/pinger is not suid!
Created symlink /etc/systemd/system/multi-user.target.wants/squid.service → /lib/systemd/system/squid.service.
Processing triggers for systemd (241-7~deb10u4) ...
Processing triggers for libc-bin (2.28-10) ...
Processing triggers for ca-certificates (20200601~deb10u1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@dhcp:~#
```

Y verificamos que el servicio está corriendo:

```
root@dhcp:~# systemctl status squid
● squid.service - Squid Web Proxy Server
   Loaded: loaded (/lib/systemd/system/squid.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-10-21 12:45:02 CDT; 51s ago
     Docs: man:squid(8)
 Main PID: 2699 (squid)
    Tasks: 4 (limit: 515)
   Memory: 15.8M
   CGroup: /system.slice/squid.service
           ├─2699 /usr/sbin/squid -sYC
           ├─2701 (squid-1) --kid squid-1 -sYC
           ├─2704 (logfile-daemon) /var/log/squid/access.log
           └─2721 (pinger)

Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Using Least Load store dir selection
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Set Current Directory to /var/spool/squid
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Finished loading MIME types and icons.
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: HTCP Disabled.
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Pinger socket opened on FD 14
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Squid plugin modules loaded: 0
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Adaptation support is off.
Oct 21 12:45:02 dhcp.mafia.local squid[2701]: Accepting HTTP Socket connections at local=[::]:3128 remote=[::] FD 12 fla
Oct 21 12:45:02 dhcp.mafia.local systemd[1]: /lib/systemd/system/squid.service:7: PIDFile= references path below legacy
Oct 21 12:45:03 dhcp.mafia.local squid[2701]: storeLateRelease: released 0 objects
lines 1-23/23 (END)
```

Para personalizar la configuración, es recomendable hacer un respaldo del archivo original de configuración, antes de cualquier cambio:

```
root@dhcp:~# cp /etc/squid/squid.conf{,.backup}
root@dhcp:~# ls /etc/squid/
conf.d  errorpage.css  squid.conf  squid.conf.backup
root@dhcp:~#
```

Ahora procedemos a cambiar el archivo **/etc/squid/squid.conf**

Por defecto **squid** se encuentra escuchando en el puerto **3128**:

```
# Squid normally listens to port 3128
http_port 3128
```

Se pueden usar listas de control de acceso (*ACL - Access Control List*), para definir el acceso a los recursos *Web* para los usuarios, para ello podemos usar un archivo que contenga las direcciones **IP** o rangos, en este caso lo generamos en **/etc/squid/allowed_ips.txt**:

```
root@dhcp:~# cat /etc/squid/allowed_ips.txt
10.10.50.0/24
root@dhcp:~#
```

Una vez definidas las direcciones **IP** y/o rangos, podemos agregar la *ACL* de esta manera a **/etc/squid/squid.conf**:

```
acl localnet src fc00::/7             # RFC 4193 local private network range
acl localnet src fe80::/10            # RFC 4291 link-local (directly plugged) machines

# New ACL
acl allowed_ips src "/etc/squid/allowed_ips.txt"

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
```

Y después añadimos la regla de acceso:

```
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
#http_access allow localnet
http_access allow localhost

# New Rule
http_access allow allowed_ips

# And finally deny all other access to this proxy
http_access deny all
```

Se deben poner las reglas de acceso antes de la regla `"http_access deny all"`, ya que

su comportamiento es similar a las reglas de un *firewall*.

Para que los cambios surtan efecto, se debe reiniciar el servicio:

```
root@dhcp:~# systemctl restart squid
root@dhcp:~# systemctl status squid
● squid.service - Squid Web Proxy Server
   Loaded: loaded (/lib/systemd/system/squid.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-10-21 12:57:16 CDT; 6s ago
     Docs: man:squid(8)
  Process: 3396 ExecStartPre=/usr/sbin/squid --foreground -z (code=exited, status=0/SUCCESS)
  Process: 3399 ExecStart=/usr/sbin/squid -sYC (code=exited, status=0/SUCCESS)
 Main PID: 3400 (squid)
    Tasks: 4 (limit: 515)
   Memory: 15.8M
   CGroup: /system.slice/squid.service
           ├─3400 /usr/sbin/squid -sYC
           ├─3402 (squid-1) --kid squid-1 -sYC
           ├─3403 (logfile-daemon) /var/log/squid/access.log
           └─3404 (pinger)

Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Using Least Load store dir selection
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Set Current Directory to /var/spool/squid
Oct 21 12:57:16 dhcp.mafia.local systemd[1]: Started Squid Web Proxy Server.
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Finished loading MIME types and icons.
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: HTCP Disabled.
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Pinger socket opened on FD 14
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Squid plugin modules loaded: 0
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Adaptation support is off.
Oct 21 12:57:16 dhcp.mafia.local squid[3402]: Accepting HTTP Socket connections at local=[::]:3128 remote=[::] FD 12 fla
Oct 21 12:57:17 dhcp.mafia.local squid[3402]: storeLateRelease: released 0 objects
lines 1-25/25 (END)
```

7. Web

Comenzamos por instalar todos los paquetes necesarios, desde el servidor web, clientes para conexion a base de datos hasta modulos de php para el CMS

apt-get install -y apache2 php libapache2-mod-php php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip curl ed dos2unix mariadb-client postgresql-client-11

Primero debemos crear el virtual host en /etc/apache/sites-aviable/mediawiki.conf

```
<VirtualHost *:80>
    ServerName web01.mafia.local
    ServerAlias www.web01.mafia.local
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/mediawiki
    Redirect permanent / https://web01.mafia.local
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

En ese mismo archivo debemos definir las directivas para poder usar certificados que permitan https. También debemos definir el direccionamiento a https y definir las rutas donde tenemos los certificados

```
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerName web01.mafia.local
    ServerAlias www.web01.mafia.local
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/mediawiki
    <Directory /var/www/html/>
      AllowOverride All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/becarios.pem
    SSLCertificateKeyFile /etc/ssl/private/becarios.key
    <FilesMatch \"\.(cgi|shtml|phtml|php)$\">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
  </VirtualHost>
</IfModule>
```

Habilitamos el sitio

```
root@web01:~# a2ensite mediawiki.conf
Enabling site mediawiki.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Deshabilitamos el sitio por defecto

```
root@web01:~# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Habilitamos el modulo de apache para ssl

```
root@web01:~# a2enmod rewrite ssl
Enabling module rewrite.
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Podemos notar que apache está corriendo sin problemas

```
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-10-21 12:50:19 CDT; 10min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 770 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 774 (apache2)
    Tasks: 7 (limit: 515)
   Memory: 39.6M
   CGroup: /system.slice/apache2.service
           ├─774 /usr/sbin/apache2 -k start
           ├─775 /usr/sbin/apache2 -k start
           ├─776 /usr/sbin/apache2 -k start
           ├─777 /usr/sbin/apache2 -k start
           ├─778 /usr/sbin/apache2 -k start
           ├─779 /usr/sbin/apache2 -k start
           `-783 /usr/sbin/apache2 -k start
```

Ahora pasaremos a instalar mediawiki, para ello descargamos la última versión

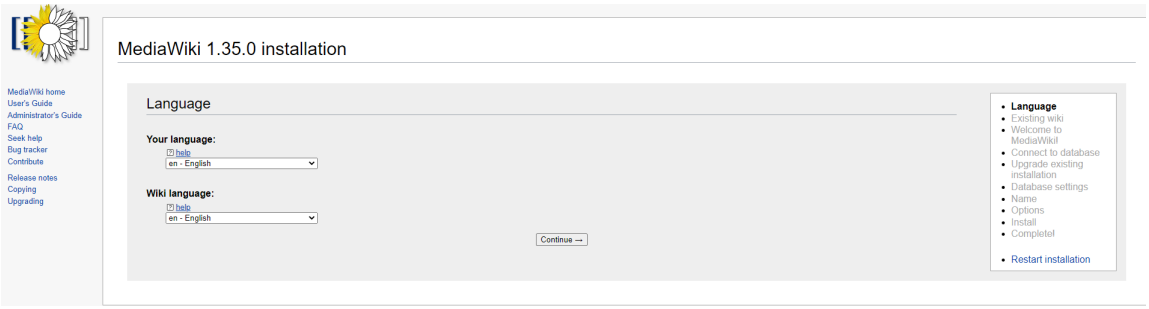
```
root@web01:~# wget https://releases.wikimedia.org/mediawiki/1.35/mediawiki-1.35.0.tar.gz
--2020-10-21 13:03:39--  https://releases.wikimedia.org/mediawiki/1.35/mediawiki-1.35.0.tar.gz
Resolving releases.wikimedia.org (releases.wikimedia.org)... 2620:0:860:ed1a::1, 208.80.153.224
Connecting to releases.wikimedia.org (releases.wikimedia.org)|2620:0:860:ed1a::1|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48039474 (46M) [application/x-gzip]
Saving to: 'mediawiki-1.35.0.tar.gz'

mediawiki-1.35.0.tar.gz             100%[===================================================>]  45.81M  23.8MB/s    in 1.9s

2020-10-21 13:03:41 (23.8 MB/s) - 'mediawiki-1.35.0.tar.gz' saved [48039474/48039474]
```

Descomprimimos el archivo descargado y lo pasamos a /var/www/html

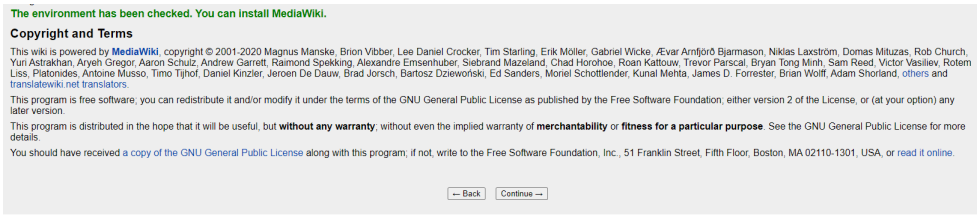
```
root@web01:~# tar xzf mediawiki-1.35.0.tar.gz
root@web01:~# ls m*
mediawiki-1.35.0.tar.gz

mediawiki-1.35.0:
CODE_OF_CONDUCT.md  INSTALL            api.php                  docs          index.php    mw-config            tests
COPYING             README.md          autoload.php             extensions    jsduck.json  opensearch_desc.php  thumb.php
CREDITS             RELEASE-NOTES-1.35 cache                    images        languages    resources            thumb_handler.php
FAQ                 SECURITY           composer.json            img_auth.php  load.php     rest.php             vendor
HISTORY             UPGRADE            composer.local.json-sample includes    maintenance  skins
```

Notemos que ya esta lista la página que nos asistirá a instalar Mediawiki y además esta en https

La instalación del CMS es muy sencilla. Hay que ir a http://$IPWebServer/mediawiki y una vez ahi empezaremos la instalación.
Comenzamos por seleccionar el lenguaje

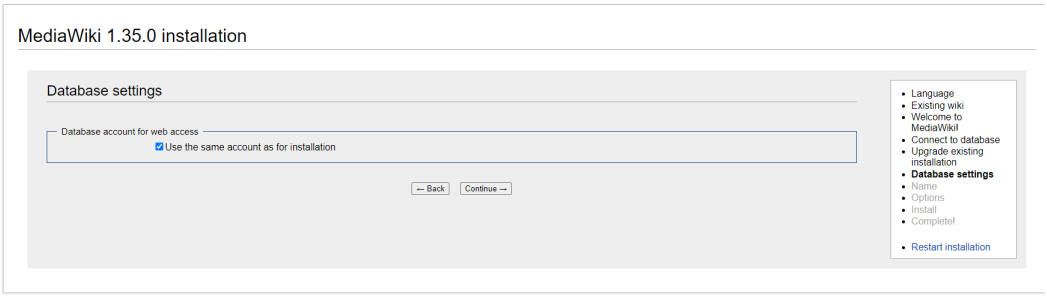


Aceptamos los términos

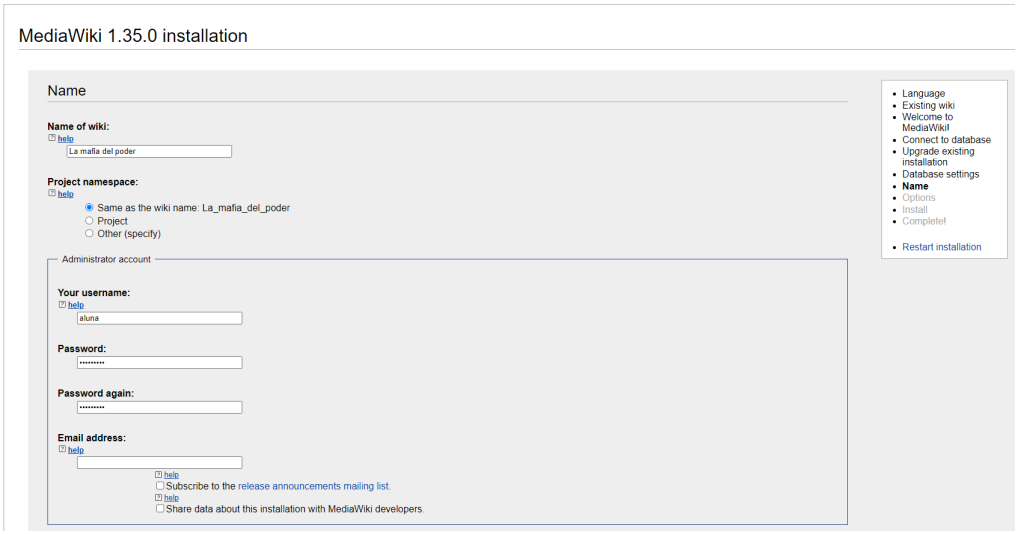


Ingresamos los datos de la base de datos a utilizar y las credenciales para acceder

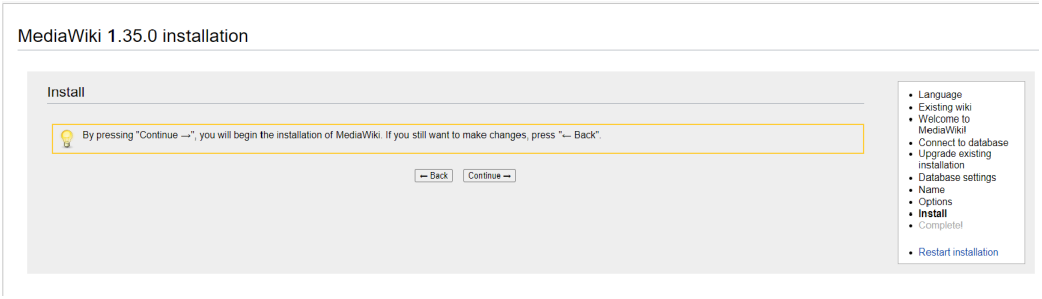
Y aceptamos usar la misma cuenta para instalación



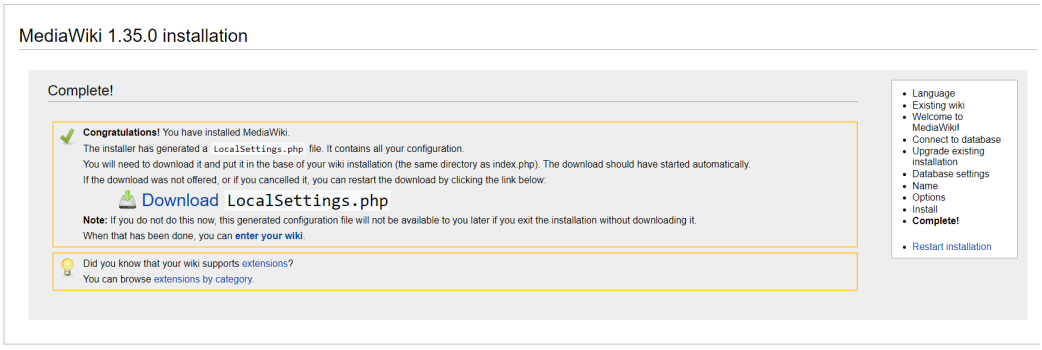
Ahora llenaremos datos para acceso al panel de administración del CMS así como un nombre para este



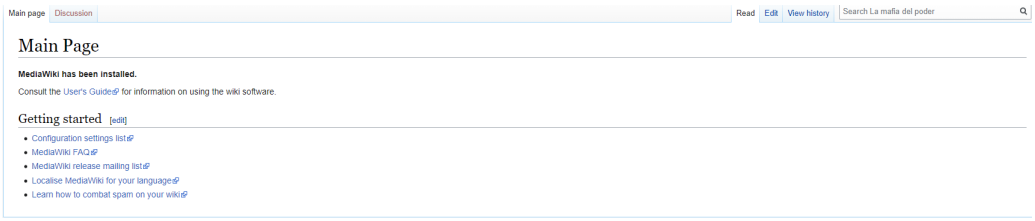
Continuamos con la instalación



La instalación ha sido un éxito. Hay que copiar el archivo que se descargó en el mismo directorio donde esta el index de mediawiki



A continuación, la página principal



8. Bases de datos

Comenzamos por instalar las dependencias necesarias

apt-get install -y mariadb-server mariadb-client

Ahora comenzamos la instalación de mariadb con

mysql_secure_installation

Esto iniciara la configuración e instalación de MariaDB. Nos pedirá una contraseña nueva para root

```
root@database:~# mysql_secure_installation

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

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, 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):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.
```

Ahora debemos remover usuarios anonimos

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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.

Remove anonymous users? [Y/n] y
 ... Success!
```

Permitimos que root solo se conecte desde el localhost

```
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] y
 ... Success!
```

Borramos la base de datos test que viene por defecto y ya hemos instalado MariaDB

```
By default, MariaDB 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] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
```

Para habilitar la conexión de usuarios desde host remotos debemos crearlos en la base de taos y darles permisos

```
MariaDB [(none)]> CREATE DATABASE wikidb;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wikiuser'@'localhost' IDENTIFIED BY '[redacted]';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wikidb.* TO 'wikiuser'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wikidb.* TO 'wikiuser'@'web01.mafia.local' IDENTIFIED BY '[redacted]';
Query OK, 0 rows affected (0.000 sec)
```

Por último debemos editar el archivo de configuración /etc/mysql/mariadb.conf.d/50-server.cnf para abrir el puerto de escucha del demonio y habilitar que escuche desde todas las ip's. Para restringir el acceso se usa el firewall

```
root@database:~# cat /etc/mysql/mariadb.conf.d/50-server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#
user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
socket                  = /run/mysqld/mysqld.sock
port                    = 3306
basedir                 = /usr
datadir                 = /var/lib/mysql
tmpdir                  = /tmp
lc-messages-dir         = /usr/share/mysql
#skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

Para Postgresql descargamos los paquetes necesarios

apt-get install -y postgresql-11 postgresql-client-11

Para habilitar la conexión remota de usuarios debemos editar 2 archivos. El primero es /etc/postgresql/11/main/postgresql.conf donde habilitaremos la escucha de todas las direcciones

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
port = 5432                     # (change requires restart)
```

El segundo es /etc/postgresql/11/main/pg_hba.conf donde habilitaremos el método, la autenticación, la base de datos a la que se conectan los usuarios y de donde

```
# IPv4 local connections:
host    all             all             0.0.0.0/0               password
```

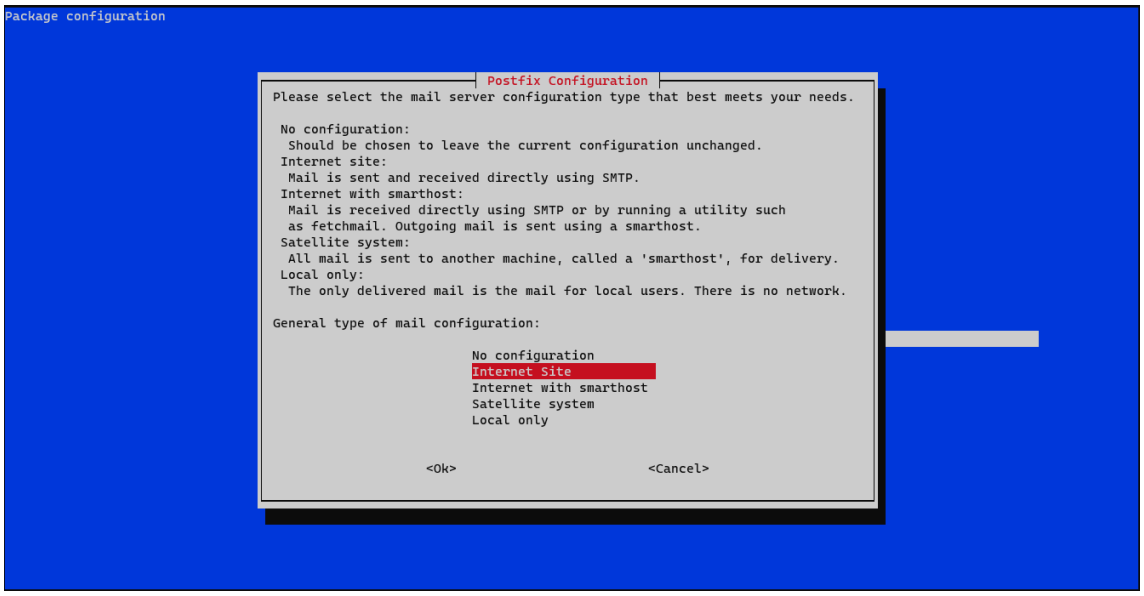
9. Mail

Lo primero que debemos hacer es instalar **`postfix`** a través del gestor de paquetes para proveer el servicio de SMTP.

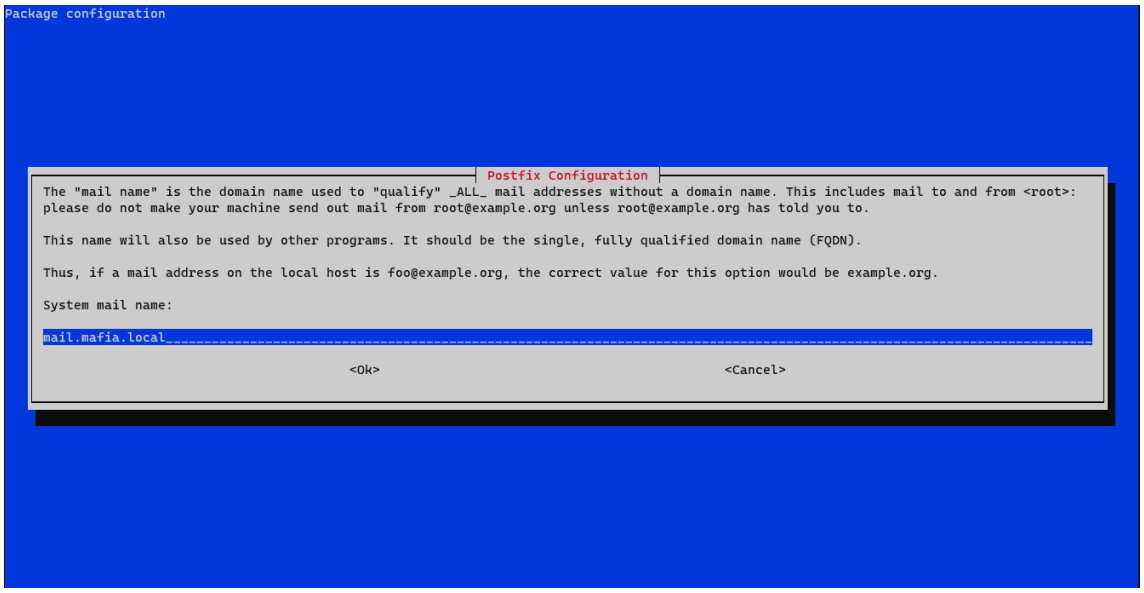
apt-get install -y postfix

```
root@mail:~# apt-get install -y postfix
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Una vez comience el proceso de instalación, se nos mostrará una pantalla en la cual debemos indicar la configuración general de nuestro servidor SMTP. En este caso deberemos seleccionar **`Internet Site`**, ya que estamos emulando un servicio de correo en internet.



Posteriormente deberemos indicar el nombre del equipo que estamos configurando.



Una vez realizada la instalación, debemos modificar el archivo /etc/postfix/main.cf y poner el siguiente contenido.

```
myhostname = mail.mafia.local
mydomain = mafia.local
mynetworks = 127.0.0.0/8 172.16.50.0/24
```

```
37 myhostname = mail.mafia.local
38 mydomain = mafia.local
```

```
43 relayhost =
44 mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
45 mailbox_size_limit = 0
```

De esta manera indicamos el nombre del equipo, el dominio y las redes utilizadas.

Luego debemos indicar la ruta a nuestro certificado y llave SSL.

```
26 # TLS parameters
27 smtpd_tls_cert_file = /etc/ssl/certs/mafia.pem
28 smtpd_tls_key_file = /etc/ssl/private/mafia.key
29 smtpd_use_tls=yes
```

En caso de que no contemos con estos archivos, podemos generarlos con el siguiente comando.

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/mafia.key -out /etc/ssl/certs/mafia.pem

```
root@mail:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
>   -keyout /etc/ssl/private/mafia.key -out /etc/ssl/certs/mafia.pem
Generating a RSA private key
```

10. DHCP

Para proveer el servicio de DHCP debemos instalar el paquete **isc-dhcp-server** a través del gestor de paquetes de Debian.

```
alopez@debian:~$ sudo apt-get install -y isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libirs-export161 libisccfg-export163 policycoreutils selinux-utils
Suggested packages:
  policykit-1 isc-dhcp-server-ldap
The following NEW packages will be installed:
  isc-dhcp-server libirs-export161 libisccfg-export163 policycoreutils selinux-utils
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
```

Una vez que se instale se iniciará el servicio de forma automática, por lo que se mostrará un mensaje de error debido a que aún no se realizan las configuraciones correspondientes a una interfaz de red; de momento lo podemos ignorar.

```
Job for isc-dhcp-server.service failed because the control process exited with error code.
See "systemctl status isc-dhcp-server.service" and "journalctl -xe" for details.
invoke-rc.d: initscript isc-dhcp-server, action "start" failed.
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: failed (Result: exit-code) since Wed 2020-10-21 06:26:08 CDT; 20ms ago
     Docs: man:systemd-sysv-generator(8)
  Process: 840 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=1/FAILURE)

Oct 21 06:26:06 debian dhcpd[852]: bugs on either our web page at www.isc.org or in the README file
Oct 21 06:26:06 debian dhcpd[852]: before submitting a bug.  These pages explain the proper
Oct 21 06:26:06 debian dhcpd[852]: process and the information we find helpful for debugging.
Oct 21 06:26:06 debian dhcpd[852]:
Oct 21 06:26:06 debian dhcpd[852]: exiting.
Oct 21 06:26:08 debian isc-dhcp-server[840]: Starting ISC DHCPv4 server: dhcpdcheck syslog for diagnostics. ... failed!
Oct 21 06:26:08 debian isc-dhcp-server[840]:  failed!
Oct 21 06:26:08 debian systemd[1]: isc-dhcp-server.service: Control process exited, code=exited, status=1/FAILURE
Oct 21 06:26:08 debian systemd[1]: isc-dhcp-server.service: Failed with result 'exit-code'.
Oct 21 06:26:08 debian systemd[1]: Failed to start LSB: DHCP server.
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10) ...
Processing triggers for systemd (241-7~deb10u4) ...
```

Una vez hecho lo anterior, procedemos a modificar el archivo /etc/default/isc-dhcp-server. Lo primero que haremos será quitar el comentario de la línea **4** para indicar el archivo de configuración utilizado para la distribución de direcciones IPv4.

```
3 # Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
4 DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
5 #DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf
```

Posteriormente deberemos modificar la línea **17** para indicar la interfaz de la red interna que se utilizará para distribuir las direcciones IPv4.

```
15 # On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
16 #       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
17 INTERFACESv4="ens37"
18 INTERFACESv6=""
```

Ahora debemos editar el archivo /etc/dhcp/dhcpd.conf y agregar el siguiente contenido:

```
## Red interna
subnet 10.10.50.0 netmask 255.255.255.0 {
  authoritative;
  range 10.10.50.32 10.10.50.63;
  default-lease-time 3600;
  max-lease-time 7200;
  option subnet-mask 255.255.255.0;
  option broadcast-address 10.10.50.255;
  option routers 10.10.50.10;
  option domain-name-servers 172.16.50.40;
  option domain-name "mafia.local";

  host client01 {
    hardware ethernet 00:0c:29:b3:03:85;
    fixed-address 10.10.50.31;
  }

  host client02 {
    hardware ethernet 00:0c:29:fb:30:0f;
    fixed-address 10.10.50.32;
  }
}
```

```
## Red interna
subnet 10.10.50.0 netmask 255.255.255.0 {
  authoritative;
  range 10.10.50.32 10.10.50.63;
  default-lease-time 3600;
  max-lease-time 7200;
  option subnet-mask 255.255.255.0;
  option broadcast-address 10.10.50.255;
  option routers 10.10.50.10;
  option domain-name-servers 172.16.50.40;
  option domain-name "mafia.local";

  host client01 {
    hardware ethernet 00:0c:29:b3:03:85;
    fixed-address 10.10.50.30;
  }

  host client02 {
    hardware ethernet 00:0c:29:fb:30:0f;
    fixed-address 10.10.50.31;
  }
}
```

De esta manera, creamos una subred `10.10.50.0/24` en la cual el rango de direcciones va desde la `10.10.50.32` a la `10.10.50.63`. El tiempo de asignación de dirección IPv4

por defecto es de 3600 segundos (1 hora) y el máximo de 7200 segundos (2 horas). Se utiliza la máscara de red `255.255.255.0`, dirección de broadcast `10.10.50.255`, gatew `10.10.50.10` (DHCP), servidor DNS `172.16.50.40` (red protegida) y nombre de dominio `mafia.local`.

Adicionalmente se establecen direcciones IP estáticas `10.10.50.30` y `10.10.50.31` para las MAC `00:0c:29:b3:03:85` y `00:0c:29:fb:30:0f` de forma correspondiente.

Finalmente reiniciamos el servicio de DHCP.

```
root@dhcp:~# systemctl restart isc-dhcp-server.service
root@dhcp:~#
```

Verificamos que los clientes poseen una dirección IP del segmento de la red interna.

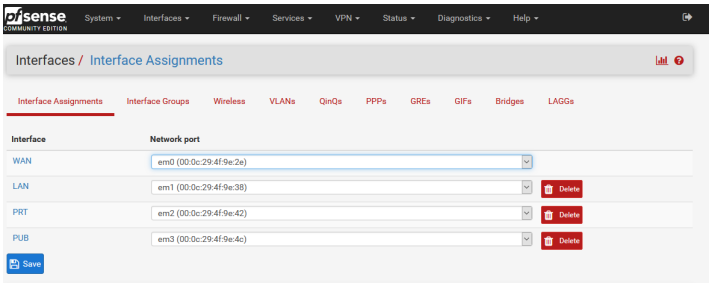
```
Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : mafia.local
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-FB-30-0F
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::9daf:b57d:5c02:94bf%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.50.31(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, October 21, 2020 7:27:26 AM
   Lease Expires . . . . . . . . . . : Wednesday, October 21, 2020 8:27:26 AM
   Default Gateway . . . . . . . . . : 10.10.50.10
   DHCP Server . . . . . . . . . . . : 10.10.50.10
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-27-21-DC-5C-00-0C-29-FB-30-0F
   DNS Servers . . . . . . . . . . . : 172.16.50.40
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

```
alopez@debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 10
00
    link/ether 00:0c:29:b3:03:85 brd ff:ff:ff:ff:ff:ff
    inet 10.10.50.30/24 brd 10.10.50.255 scope global dynamic ens33
       valid_lft 3400sec preferred_lft 3400sec
    inet6 fe80::20c:29ff:feb3:385/64 scope link
       valid_lft forever preferred_lft forever
alopez@debian:~$
```

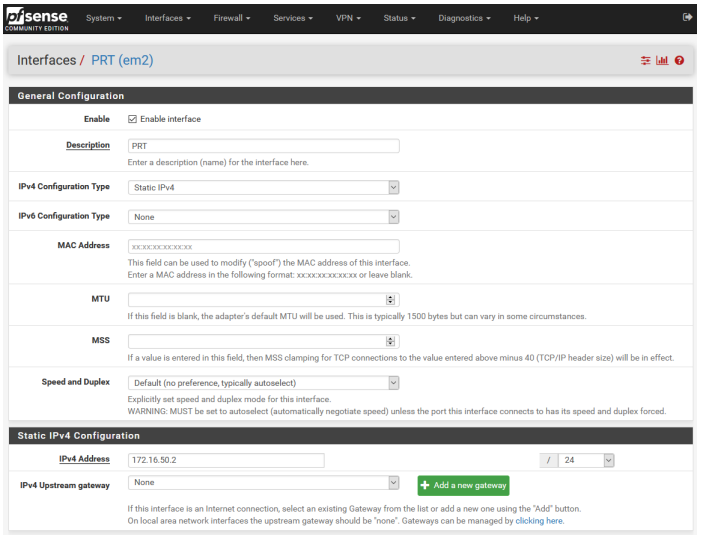
11. pfSense

La maquina pfsense tiene 4 interfaces, si se instala nuevamente es posible ponerle solo 3 para que se tenga acceso a la consola, en este caso se tienen 2 interfaces con nat y dos con vmnet10 y 11 en host only. La 10 es para el segmento 172.16.0.0 y la 11 para el segmento 192.168.50.0.

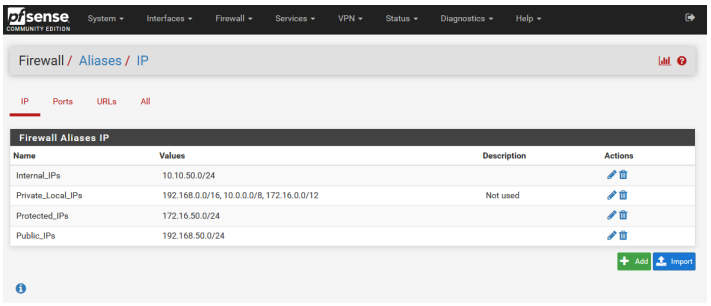
Comenzamos la configuración añadiendo las interfaces



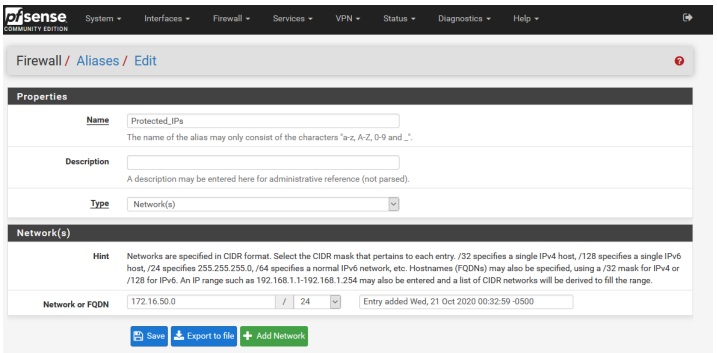
Se define la interfaz con ip estatica, tiene que ser la .2 para que actue como gateway y no se encime con la .1 que usa windows en los adaptadores vmnet



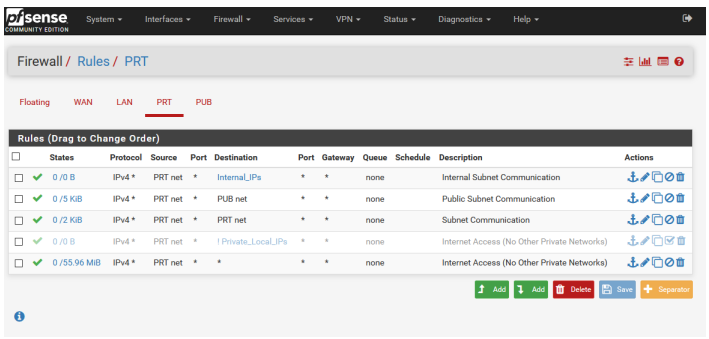
Se crean alias de rangos de subredes para usarlos mas adelante en las reglas de firewall



Se indica el tipo como red y se pone en notacion CIDR



La consola donde se añaden las reglas. pfSense procesa sus reglas de arriba a abajo, el primer match gana



12. NTP

Comenzaremos por instalar ntp tanto en los servidores web como en el servidor NTP.

apt install ntp

Ahora procederemos a editar el archivo de configuración /etc/ntp.conf del servidor NTP. Agregamos el servidor NTP del CENAM a nuestra lista de servidores NTP a los cuales se va a sincronizar nuestro servidor, comentamos los permisos por default, las entradas pool, el permiso local con IPv6 y se agrega la restricción para que solo puedan sincronizarse los dispositivos de la red que hemos creado

```
 GNU nano 3.2                          /etc/ntp.conf

# You do need to talk to an NTP server or two (or three).
#server ntp.your-provider.example
server cronos.cenam.gob.mx prefer iburst
# pool.ntp.org maps to about 1000 low-stratum NTP servers.  Your server will
# pick a different set every time it starts up.  Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
#pool 0.debian.pool.ntp.org iburst
#pool 1.debian.pool.ntp.org iburst
#pool 2.debian.pool.ntp.org iburst
#pool 3.debian.pool.ntp.org iburst


# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
# details.  The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
#restrict -4 default kod notrap nomodify nopeer noquery limited
#restrict -6 default kod notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 192.168.50.0 mask 255.255.255.0 nomodify notrap
restrict 172.16.50.0 mask 255.255.255.0 nomodify notrap
restrict 10.10.50.0 mask 255.255.255.0 nomodify notrap
restrict 127.0.0.1
#restrict ::1

# Needed for adding pool entries
```

Podemos notar que está sincronizado con el servidor del cenam

```
root@monitor-syslog-ntp:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 cronos.cenam.mx .STEP.          16 u   17   64    0    0.000    0.000   0.000
root@monitor-syslog-ntp:~# _
```

Despues de entre 5 y 10 minutos, la sincronización estará completa y recibiremos codigo de estado exitoso al ejecutar nptime

```
ntp_gettime() returns code 0 (OK)
  time e337176f.e437b4a4  Sun, Oct 18 2020 14:35:11.891, (.891475287),
  maximum error 248999 us, estimated error 1105 us, TAI offset 37
ntp_adjtime() returns code 0 (OK)
  modes 0x0 (),
  offset 122.761 us, frequency -5.922 ppm, interval 1 s,
  maximum error 248999 us, estimated error 1105 us,
  status 0x2001 (PLL,NANO),
  time constant 6, precision 0.001 us, tolerance 500 ppm,
```

12.1. Clientes

Para sincronizar maquinas con nuestro servidor NTP vamos al archivo de configuración /etc/ntp.conf y ahora solo indicamos que el servidor al cual nos sincronizaremos sera el nuestro (cuya ip es 192.168.50.30), comentamos las mismas lineas que en el archivo del servidor y no añadimos restricciones pues somos solo clientes.

```
 GNU nano 3.2                        /etc/ntp.conf                        Modified

# You do need to talk to an NTP server or two (or three).
#server ntp.your-provider.example
server monitor-syslog-ntp.mafia.local_
# pool.ntp.org maps to about 1000 low-stratum NTP servers.  Your server will
# pick a different set every time it starts up.  Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
#pool 0.debian.pool.ntp.org iburst
#pool 1.debian.pool.ntp.org iburst
#pool 2.debian.pool.ntp.org iburst
#pool 3.debian.pool.ntp.org iburst


# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
# details.  The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
#restrict -4 default kod notrap nomodify nopeer noquery limited
#restrict -6 default kod notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
#restrict ::1

# Needed for adding pool entries
#restrict source notrap nomodify noquery

# Clients from this (example!) subnet have unlimited access, but only if
```

Notemos que la sincronización está hecha con nuestro servidor NTP

```
root@web01:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 monitor-syslog- .INIT.          16 u    -   64    0    0.000    0.000   0.000
root@web01:~#
```