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

20 de octubre de 2020

Índice

1. 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

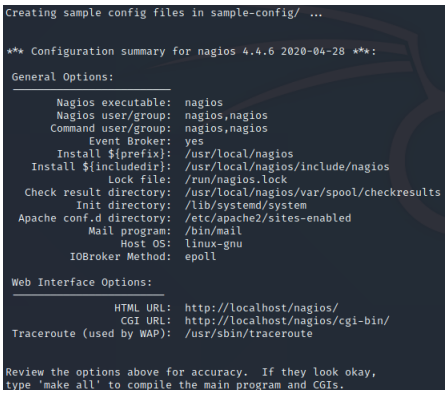
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz -P /tmp
tar xzf nagios-4.4.6.tar.gz

```
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/
```

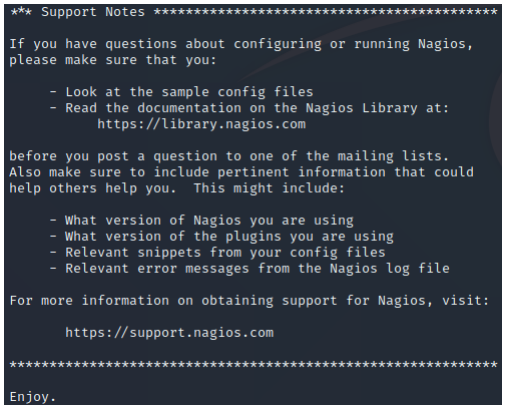
Ahora entraremos al directorio de nagios y empezaremos a compilar Nagios. Lo primero es configurarlo con

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



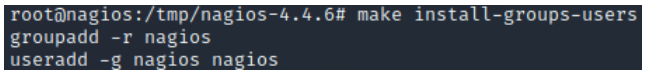
Ahora, compilaremos con

make all



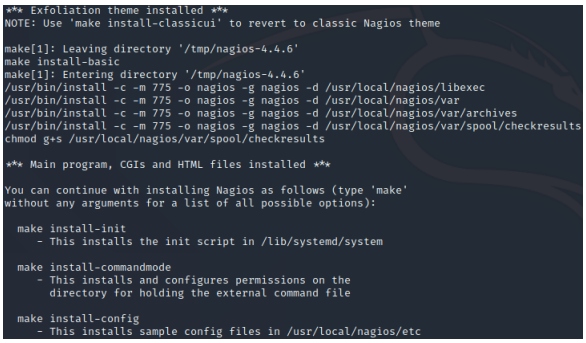
Para crear el grupo y usuario nagios y añadir al usuario que identifica al servidor web al grupo nagios recién creado con usamos

make install-groups-users



Instalamos los archivos de Nagios core

make install



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 ***
```

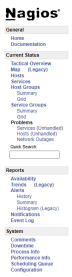
Debemos habilitar el modulo cgi de apache

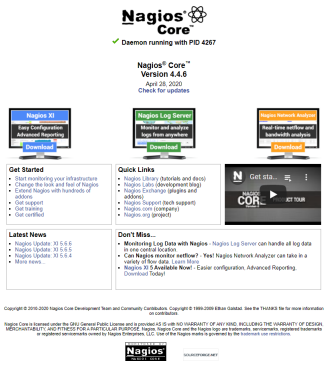
```
root@nagios:/tmp/nagios-4.4.6# a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

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

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-20-2020 13:25:02	0d 0h 44m 21s	1/4	OK - load average: 0.12, 0.13, 0.09
	Current Users	OK	10-20-2020 13:25:40	0d 0h 48m 43s	1/4	USERS OK - 2 users currently logged in
	HTTP	OK	10-20-2020 13:26:17	0d 0h 48m 6s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
	PING	OK	10-20-2020 13:26:55	0d 0h 47m 28s	1/4	PING OK - Packet loss = 0%, RTA = 0.08 ms
	Root Partition	OK	10-20-2020 13:25:32	0d 0h 3m 51s	1/4	DISK OK - free space: / 552 MB (24.36% inode=70%):
	SSH	OK	10-20-2020 13:28:10	0d 0h 46m 13s	1/4	SSH OK - OpenSSH_7.9p1 Debian-10+deb10u2 (protocol 2.0)
	Swap Usage	OK	10-20-2020 13:28:47	0d 0h 45m 36s	1/4	SWAP OK - 100% free (509 MB out of 509 MB)
	Total Processes	OK	10-20-2020 13:24:25	0d 0h 44m 58s	1/4	PROCS OK: 65 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services

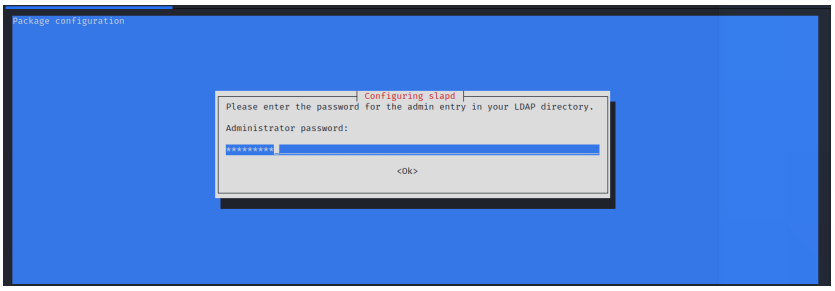
2. LDAP & LDAPS

2.1. 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

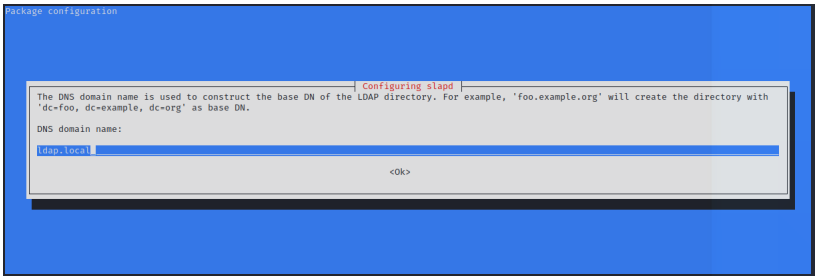


Se reconfigura slapd con

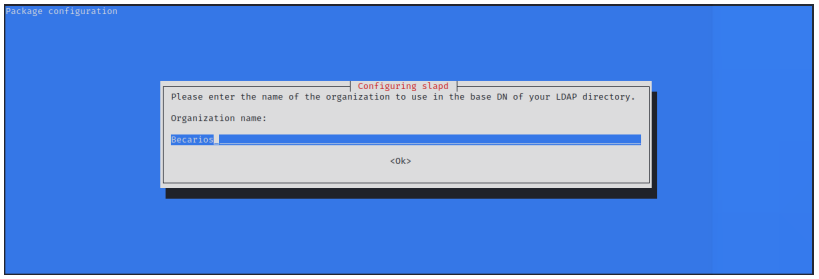
dpkg-reconfigure slapd

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

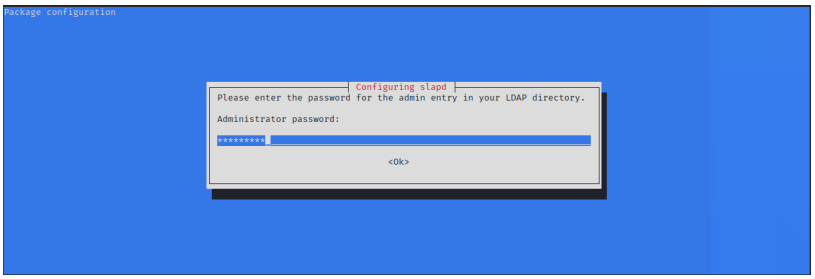
A continuación, configuramos el nombre de dominio completo del servidor OpenLDAP que se utilizará para crear el DN base.



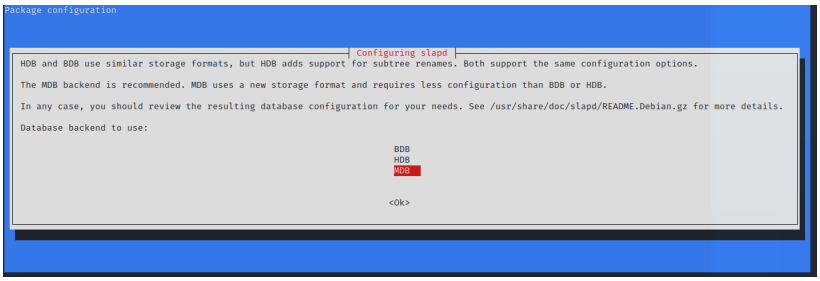
Establecemos el nombre de la organización

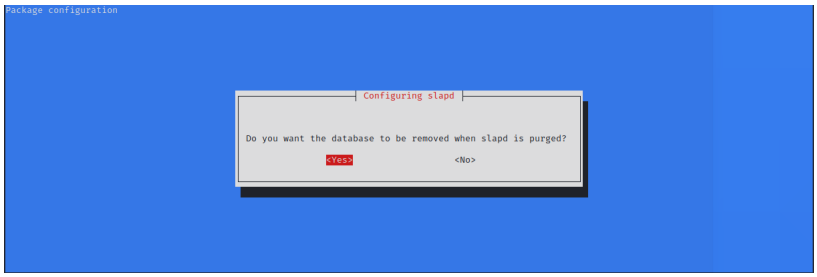


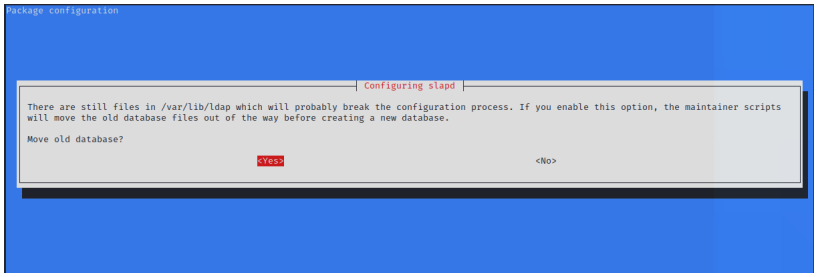
Configuramos y verificamos el password de administrador.



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

2.2. 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 creadoslos 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            = ./demcCA              # 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

Creamos el certificado CA

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

Luego, la llave para el servidor LDAP

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

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

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

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

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

2.3. Verificar

Para verificar el funcionamiento de nuestro servidor podemos usar

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

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.

3. 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

```
 GNU nano 3.2                                    /etc/bind/named.conf.local

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

     type master; //Primary DNS

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

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

};
zone "200.168.192.in-addr.arpa" IN { //Reverse lookup name, should match your network in reverse order

     type master; // Primary DNS

     file "/etc/bind/reverse.becarios.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";
```

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

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

```
 GNU nano 3.2                                                    /et

;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.becarios.local. root.becarios.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      ns1.becarios.local.

;IP address of Name Server

ns1     IN      A       192.168.200.138

;Mail Exchanger

;becarios.local.   IN     MX   138   mail.becarios.local.

;A - Record HostName To Ip Address

www     IN       A      192.168.200.138

;CNAME record

ftp     IN      CNAME   www.becarios.local.
```

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

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

```
  GNU nano 3.2

;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     becarios.local. root.becarios.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.becarios.local.

;Reverse lookup for Name Server

138     IN      PTR    ns1.becarios.local.

;PTR Record IP address to HostName

100     IN      PTR    www.becarios.local.
```

Reiniciamos el servicio y despues checamos el estatus del mismo

sudo systemctl restart bind9
sudo systemctl status bind9

```
root@debian:~# sudo systemctl status bind9
* bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-10-07 18:31:42 CDT; 2min 56s ago
     Docs: man:named(8)
  Process: 1299 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 1300 (named)
    Tasks: 4 (limit: 515)
   Memory: 12.1M
   CGroup: /system.slice/bind9.service
           `-1300 /usr/sbin/named -u bind

Oct 07 18:31:42 debian named[1300]: zone 127.in-addr.arpa/IN: loaded serial 1
Oct 07 18:31:42 debian named[1300]: zone 200.168.192.in-addr.arpa/IN: loaded serial 2
Oct 07 18:31:42 debian named[1300]: zone becarios.local/IN: loaded serial 2
Oct 07 18:31:42 debian named[1300]: zone localhost/IN: loaded serial 2
Oct 07 18:31:42 debian named[1300]: all zones loaded
Oct 07 18:31:42 debian systemd[1]: Started BIND Domain Name Server.
Oct 07 18:31:42 debian named[1300]: running
Oct 07 18:31:42 debian named[1300]: zone 200.168.192.in-addr.arpa/IN: sending notifies (serial 2)
Oct 07 18:31:42 debian named[1300]: managed-keys-zone: Key 20326 for zone . acceptance timer complete: key now trusted
Oct 07 18:31:42 debian named[1300]: resolver priming query complete
root@debian:~#
```

Podemos ver que el servicio está funcionando

4. 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:~#
```

5. Squid + Dansguardian

5.1. Squid Proxy

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

```
rmrios@debian10test:~$ sudo apt-get update
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://mmc.geofisica.unam.mx/debian buster InRelease
Hit:3 http://mmc.geofisica.unam.mx/debian buster-updates InRelease
Reading package lists... Done
```

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

```
rmrios@debian10test:~$ sudo 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 libicu63 libldap-2.4-2 libldap-common libltdl7
  libperl5.28 libsasl2-2 libsasl2-modules libsasl2-modules-db libxml2 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 resolvconf smbclient ufw winbind
The following NEW packages will be installed:
  ca-certificates libdbi-perl libecap3 libgdbm-compat4 libgdbm6 libicu63 libldap-2.4-2 libldap-common libltdl7
  libperl5.28 libsasl2-2 libsasl2-modules libsasl2-modules-db libxml2 openssl perl perl-modules-5.28 squid
  squid-common squid-langpack
0 upgraded, 20 newly installed, 0 to remove and 0 not upgraded.
Need to get 22.0 MB of archives.
After this operation, 100 MB of additional disk space will be used.
Get:1 http://mmc.geofisica.unam.mx/debian buster/main amd64 perl-modules-5.28 all 5.28.1-6+deb10u1 [2,873 kB]
```

```
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.
rmrios@debian10test:~$
```

Y verificamos que el servicio está corriendo:

```
rmrios@debian10test:~$ sudo systemctl status squid
● squid.service - Squid Web Proxy Server
   Loaded: loaded (/lib/systemd/system/squid.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-10-09 12:00:29 CDT; 2min 56s ago
     Docs: man:squid(8)
 Main PID: 2974 (squid)
    Tasks: 4 (limit: 2330)
   Memory: 16.0M
   CGroup: /system.slice/squid.service
           ├─2974 /usr/sbin/squid -sYC
           ├─2976 (squid-1) --kid squid-1 -sYC
           ├─2978 (logfile-daemon) /var/log/squid/access.log
           └─2980 (pinger)

Oct 09 12:00:29 debian10test squid[2976]: Using Least Load store dir selection
Oct 09 12:00:29 debian10test squid[2976]: Set Current Directory to /var/spool/squid
Oct 09 12:00:29 debian10test squid[2976]: Finished loading MIME types and icons.
Oct 09 12:00:29 debian10test squid[2976]: HTCP Disabled.
Oct 09 12:00:29 debian10test squid[2976]: Pinger socket opened on FD 14
Oct 09 12:00:29 debian10test squid[2976]: Squid plugin modules loaded: 0
Oct 09 12:00:29 debian10test squid[2976]: Adaptation support is off.
Oct 09 12:00:29 debian10test squid[2976]: Accepting HTTP Socket connections at local=[::]:3128 remote=[::] FD 12 flags=9
Oct 09 12:00:30 debian10test systemd[1]: /lib/systemd/system/squid.service:7: PIDFile= references path below legacy dire
Oct 09 12:00:30 debian10test squid[2976]: 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:

```
rmrios@debian10test:~$ ls -l /etc/squid/
total 320
drwxr-xr-x 2 root root   4096 Oct  9 12:00 conf.d
-rw-r--r-- 1 root root   1800 Aug 26 05:35 errorpage.css
-rw-r--r-- 1 root root 316146 Aug 26 05:35 squid.conf
rmrios@debian10test:~$ sudo cp /etc/squid/squid.conf{,.backup}
rmrios@debian10test:~$ ls -l /etc/squid/
total 632
drwxr-xr-x 2 root root   4096 Oct  9 12:00 conf.d
-rw-r--r-- 1 root root   1800 Aug 26 05:35 errorpage.css
-rw-r--r-- 1 root root 316146 Aug 26 05:35 squid.conf
-rw-r--r-- 1 root root 316146 Oct  9 12:07 squid.conf.backup
rmrios@debian10test:~$
```

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**:

```
rmrios@debian10test:~$ cat /etc/squid/allowed_ips.txt
192.168.20.0/24
192.168.192.0/24
```

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

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

# Nueva 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

# Nueva Regla
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:

```
rmrios@debian10test:~$ sudo systemctl restart squid
rmrios@debian10test:~$ sudo systemctl status squid
● squid.service - Squid Web Proxy Server
   Loaded: loaded (/lib/systemd/system/squid.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-10-09 12:50:14 CDT; 28s ago
     Docs: man:squid(8)
  Process: 3740 ExecStartPre=/usr/sbin/squid --foreground -z (code=exited, status=0/SUCCESS)
  Process: 3743 ExecStart=/usr/sbin/squid -sYC (code=exited, status=0/SUCCESS)
 Main PID: 3744 (squid)
    Tasks: 4 (limit: 2330)
   Memory: 15.8M
   CGroup: /system.slice/squid.service
           ├─3744 /usr/sbin/squid -sYC
           ├─3746 (squid-1) --kid squid-1 -sYC
           ├─3747 (logfile-daemon) /var/log/squid/access.log
           └─3748 (pinger)

Oct 09 12:50:14 debian10test squid[3746]: Using Least Load store dir selection
Oct 09 12:50:14 debian10test squid[3746]: Set Current Directory to /var/spool/squid
Oct 09 12:50:14 debian10test systemd[1]: Started Squid Web Proxy Server.
Oct 09 12:50:14 debian10test squid[3746]: Finished loading MIME types and icons.
Oct 09 12:50:14 debian10test squid[3746]: HTCP Disabled.
Oct 09 12:50:14 debian10test squid[3746]: Pinger socket opened on FD 14
Oct 09 12:50:14 debian10test squid[3746]: Squid plugin modules loaded: 0
Oct 09 12:50:14 debian10test squid[3746]: Adaptation support is off.
Oct 09 12:50:14 debian10test squid[3746]: Accepting HTTP Socket connections at local=[::]:3128 remote=[::] FD 12 flags=9
Oct 09 12:50:15 debian10test squid[3746]: storeLateRelease: released 0 objects
rmrios@debian10test:~$
```

5.2. Dansguardian