



SEGURIDAD DE REDES Y CN

Práctica 3.4 Instalación del servidor RADIUS mediante el paquete Freeradius.

OBJETIVO:

El alumno realizará la instalación de un servidor RADIUS y realizará la configuración básica para poner en marcha el servidor.

REQUERIMIENTOS:

Práctica individual

VM con CentOS 6.10 software instalado hasta la practica anterior

Tiempo Estimado: 1 -2 horas.

DESARROLLO:

RADIUS (**R**emote **A**uthentication **D**ial-In **U**ser **S**ervice) es un protocolo de autenticación, autorización y auditoría para el manejo de cuentas de usuario. Originalmente desarrollado por Livingston Enterprises y publicado en 1997 como los RFC 2058 y 2059. Es utilizado para administrar el acceso remoto y la movilidad IP, como ocurre en servicios de acceso por modem, DSL, servicios inalámbricos 802.11 o servicios de **VoIP**, este protocolo trabaja a través del puerto **1812 y 1813** de **UDP**.

La autenticación por este protocolo se realiza a través de un nombre de usuario y una clave de acceso. Esta información es procesada por un dispositivo **NAS** (**N**etwork **A**ccess **S**erver) a través de **PPP** (**P**oint to-**P**oint **P**rotocol) siendo posteriormente validada por un servidor **RADIUS** a través del protocolo correspondiente valiéndose de diversos esquemas de autenticación, como **PAP** (**P**assword **A**uthentication **P**rotocol o Protocolo de Autenticación de Clave de acceso), **CHAP** (**C**hallenge-**H**andshake **A**uthentication **P**rotocol) o **EAP** (**E**xtensible **A**uthentication **P**rotocol) y permitiendo el acceso al sistema. URL: <http://tools.ietf.org/html/rfc2058> y <http://tools.ietf.org/html/rfc2059>

Freeradius.

Este proyecto fue iniciado en 1999 por Alan DeKok y Miquel van Smoorenburg (quien colaboró anteriormente en el desarrollo de Cistron RADIUS), es una alternativa libre entre otros servidores RADIUS, siendo uno de los más completos y versátiles gracias a la variedad de módulos que lo integran. Puede operar tanto en sistemas con recursos limitados así como sistemas que atienden a millones de usuarios.

Freeradius inició como un proyecto de servidor RADIUS que permitiera una mayor colaboración de la comunidad y que pudiera cubrir las necesidades que otros servidores RADIUS no podían. Actualmente incluye soporte para LDAP, SQL y otras bases de datos, así como autenticación EAP, EAP-TTLS y PEAP que son los protocolos comunes de autenticación.

URL: <http://www.freeradius.org/>

Procedimiento de instalación:

Instalar los siguientes paquetes:

yum -y install freeradius

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
* epel: pubmirror1.math.uh.edu
base                                           | 3.7 kB    00:00
extras                                       | 3.4 kB    00:00
updates                                       | 3.4 kB    00:00
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Package freeradius.x86_64 0:2.2.6-7.el6_9 will be instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Paquete          Arquitectura  Versión          Repositorio      Tamaño
=====
Instalando:
freeradius        x86_64        2.2.6-7.el6_9    base              1.5 M

Resumen de la transacción
=====
Instalar          1 Paquete(s)

Tamaño total de la descarga: 1.5 M
Tamaño instalado: 6.3 M
Descargando paquetes:
freeradius-2.2.6-7.el6_9.x86_64.rpm          | 1.5 MB    00:00
Ejecutando el rpm_check_debug
Ejecutando prueba de transacción
La prueba de transacción ha sido exitosa
Ejecutando transacción
  Instalando   : freeradius-2.2.6-7.el6_9.x86_64          1/1
  Verifying    : freeradius-2.2.6-7.el6_9.x86_64          1/1

Instalado:
  freeradius.x86_64 0:2.2.6-7.el6_9

¡Listo!
[root@slinux1 ~]# _
```

yum -y install freeradius-mysql

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Repository centosplus is listed more than once in the configuration
Repository contrib is listed more than once in the configuration
Loading mirror speeds from cached hostfile
* epel: pubmirror1.math.uh.edu
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Package freeradius-mysql.x86_64 0:2.2.6-7.el6_9 will be instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Paquete          Arquitectura  Versión          Repositorio      Tamaño
=====
Instalando:
freeradius-mysql  x86_64        2.2.6-7.el6_9    base              60 k

Resumen de la transacción
=====
Instalar          1 Paquete(s)

Tamaño total de la descarga: 60 k
Tamaño instalado: 52 k
Descargando paquetes:
freeradius-mysql-2.2.6-7.el6_9.x86_64.rpm      | 60 kB     00:00
Ejecutando el rpm_check_debug
Ejecutando prueba de transacción
La prueba de transacción ha sido exitosa
Ejecutando transacción
  Instalando   : freeradius-mysql-2.2.6-7.el6_9.x86_64    1/1
  Verifying    : freeradius-mysql-2.2.6-7.el6_9.x86_64    1/1

Instalado:
  freeradius-mysql.x86_64 0:2.2.6-7.el6_9

¡Listo!
[root@slinux1 ~]# _
```

#yum -y install freeradius-utils

```
Resumen de la transacción
=====
Instalar      2 Paquete(s)

Tamaño total de la descarga: 259 k
Tamaño instalado: 550 k
Descargando paquetes:
(1/2): freeradius-utils-2.2.6-7.el6_9.x86_64.rpm           | 128 kB    00:00
(2/2): libpcap-1.4.0-4.20130826git2dbcaa1.el6.x86_64.rpm   | 131 kB    00:00
-----
Total                                                426 kB/s | 259 kB    00:00
Ejecutando el rpm_check_debug
Ejecutando prueba de transacción
La prueba de transacción ha sido exitosa
Ejecutando transacción
  Instalando      : 14:libpcap-1.4.0-4.20130826git2dbcaa1.el6.x86_64      1/2
  Instalando      : freeradius-utils-2.2.6-7.el6_9.x86_64                2/2
  Verifying       : 14:libpcap-1.4.0-4.20130826git2dbcaa1.el6.x86_64      1/2
  Verifying       : freeradius-utils-2.2.6-7.el6_9.x86_64                2/2

Instalado:
  freeradius-utils.x86_64 0:2.2.6-7.el6_9

Dependencia(s) instalada(s):
  libpcap.x86_64 14:1.4.0-4.20130826git2dbcaa1.el6

¡Listo!
[root@slinux1 ~]#
```

Verifique que el servicio de mysqld este ejecutándose, en caso contrario inícielo:

service mysqld status

Deberá aparecer un mensaje que muestra el número del proceso de mysqld ejecutándose.

Acceda al intérprete de comandos de MySQL:

mysql -uroot -p

Proporcione el password que estableció (practica anterior) para el usuario root de mysql

Ahora hay que crear una nueva base de datos denominada **radius**, puede hacerlo mediante la interfaz de phpMyAdmin o desde la línea de comandos de la terminal:

mysql> CREATE DATABASE radius;

Otorgue privilegios al usuario **radius** sobre la base de datos recién creada:

mysql> GRANT all ON radius.* TO radius@localhost IDENTIFIED BY 'Rad123'; mysql> flush privileges;

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Instalado:
  freeradius-utils.x86_64 0:2.2.6-7.el6_9

Dependencia(s) instalada(s):
  libpcap.x86_64 14:1.4.0-4.20130826git2dbcaa1.el6

¡Listo!
[root@slinux1 ~]# service mysqld status
Se está ejecutando mysqld (pid 3195)...
[root@slinux1 ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE DATABASE radius;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT all ON radius.* TO radius@localhost IDENTIFIED BY 'Rad123';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Ahora crearemos las tablas que requerimos para usar el software, para ello usaremos los script que nos proporcionan para generar el esquema de la base de datos del radius, el cuál está en la ruta **/etc/raddb/sql/mysqsl/**, solo usaremos el script **schema.sql** y **nas.sql**, se puede hacer desde la interfaz de mysql: Primero nos conectamos a mysql con el usuario y contraseña que acabamos de crear: **mysql> mysql -uradius**



-pRad123

Seleccionamos la base de datos con la que trabajaremos:

```
mysql> use radius;
```

Verificamos que la base de datos esta vacía, aún no tiene tablas creadas:

```
mysql> show tables;
```

Ejecutamos el script para crear las tablas de la base de datos radius:

```
mysql> SOURCE /etc/raddb/sql/mysql/schema.sql;
```

```
[root@linux1 ~]# mysql -uradius -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use radius;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> SOURCE /etc/raddb/sql/mysql/schema.sql;_
```

Verificamos que se hayan creado las tablas en la base de datos radius:

```
mysql> show tables;
```

Debe mostrarnos 7 tablas creadas (radacct, radcheck, radgroupcheck, radgroupreply, radpostauth, radreply, radusergroup)

Creamos la tabla faltante que vamos a requerir:

```
mysql> SOURCE /etc/raddb/sql/mysql/nas.sql;
```

Verificamos que se haya creado una tabla adicional (nas):

```
sql> show tables;
```

Salimos de la interfaz de mysql:

```
mysql> exit
```

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_radius |
+-----+
| radacct          |
| radcheck         |
| radgroupcheck    |
| radgroupreply    |
| radpostauth      |
| radreply         |
| radusergroup     |
+-----+
7 rows in set (0.00 sec)

mysql> SOURCE /etc/raddb/sql/mysql/nas.sql;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_radius |
+-----+
| nas              |
| radacct          |
| radcheck         |
| radgroupcheck    |
| radgroupreply    |
| radpostauth      |
| radreply         |
| radusergroup     |
+-----+
8 rows in set (0.00 sec)

mysql> exit
Bye
[root@linux1 ~]# _
```

También podemos hacerlo desde la línea de comandos, ejecutamos los scripts de creación de tablas usando el





redireccionamiento para proporcionar como entrada a mysql dichos scripts:

```
# mysql -uradius -pRad123 radius < /etc/raddb/sql/mysql/schema.sql
```

```
# mysql -uradius -pRad123 radius < /etc/raddb/sql/mysql/nas.sql
```

Edite el archivo de configuración del radius, pero antes haga una copia de seguridad:

```
# cp /etc/raddb/radiusd.conf /etc/raddb/radiusd.conf.old
```

```
# nano -c /etc/raddb/radiusd.conf
```

Asegúrese que este descomentada la línea 735 que dice:

```
$INCLUDE sql.conf
```

Guarde la configuración y salga de nano.

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.8.9 Fichero: /etc/raddb/radiusd.conf

# authenticate, accounting, pre/post-proxy, etc.
$INCLUDE ${confdir}/modules/

# Extensible Authentication Protocol
# For all EAP related authentications.
# Now in another file, because it is very large.
#
$INCLUDE eap.conf

# Include another file that has the SQL-related configuration.
# This is another file only because it tends to be big.
#
$INCLUDE sql.conf

# This module is an SQL enabled version of the counter module.
# Rather than maintaining separate (GDRM) databases of
# accounting info for each counter, this module uses the data
# stored in the raddacct table by the sql modules. This
# module NEVER does any database INSERTS or UPDATES. It is
# totally dependent on the SQL module to process Accounting
# packets.
#
$INCLUDE sql/mysql/counter.conf

# IP addresses managed in an SQL table.
#
$INCLUDE sqlippool.conf

[ 857 líneas escritas ]

[root@linux1 ~]# _
```

Edite el archivo de configuración que usa el radius para acceder a la base de datos mysql:

```
# nano -c /etc/raddb/sql.conf
```

En la sección de información para conectarse deje los parámetros como sigue:

```
# Connection info:
```

```
server = "localhost"
```

```
port = 3306
```

```
login = "radius"
```

```
password = "Rad123"
```

Habilite (descomente) la línea que contiene el parámetro **readclients = yes**

Guarde la configuración y salga de nano.



```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.8.9 Fichero: /etc/raddb/sql.conf Modificado

#
# Setting this to MORE than the number of threads means
# that there are more connections than necessary.
#
num_sql_socks = ${thread[pool].max_servers}

# number of seconds to delay retrying on a failed database
# connection (per socket)
connect_failure_retry_delay = 60

# lifetime of an SQL socket. If you are having network issues
# such as TCP sessions expiring, you may need to set the socket
# lifetime. If set to non-zero, any open connections will be
# closed "lifetime" seconds after they were first opened.
lifetime = 0

# Maximum number of queries used by an SQL socket. If you are
# having issues with SQL sockets lasting "too long", you can
# limit the number of queries performed over one socket. After
# "max_queries", the socket will be closed. Use 0 for "no limit".
max_queries = 0

# Set to 'yes' to read radius clients from the database ('nas' table)
# Clients will ONLY be read on server startup. For performance
# and security reasons, finding clients via SQL queries CANNOT
# be done "live" while the server is running.
#
readclients = yes

# Table to keep radius client info
nas_table = "nas"

[ línea 188/116 (93%), col 9/26 (34%), car 3251/3404 (95%) ]
Ver ayuda Guardar Leer Fich Pág Ant CortarTxt Pos actual
Salir Justificar Buscar Pág Sig PegarTxt Ortografía
CTRL DERECHA
```

Ahora edite el archivo `/etc/raddb/sites-enabled/default` para habilitar el sql:

`nano /etc/raddb/sites-enabled/default`

Ahora debemos habilitar el soporte para MySQL:

- Habilitar en la sección **authorize** {...}, la línea que contiene **sql** (línea 177 aprox.).
- Habilitar en la sección **accounting** {...}, la línea que contiene **sql** (línea 406 aprox.).
- Habilitar en la sección **session** {...}, la línea que contiene **sql** (línea 454 aprox.).

Guarde la configuración y salga del editor.

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.8.9 Fichero: /etc/raddb/sites-enabled/default

# Filter attributes from the accounting response.
attr_filter.accounting_response

#
# See "Autz-Type Status-Server" for how this works.
#
Acct-Type Status-Server {
}

#
# Session database, used for checking Simultaneous-Use. Either the radutmp
# or rlm_sql module can handle this.
# The rlm_sql module is *much* faster
session {
    radutmp

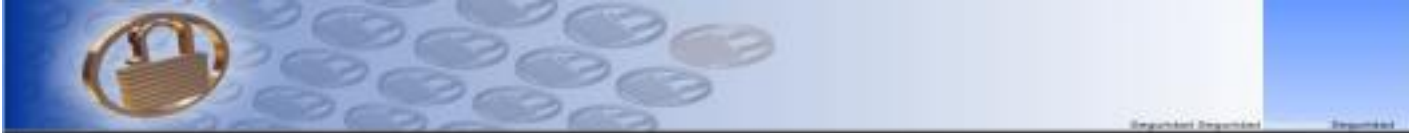
    #
    # See "Simultaneous Use Checking Queries" in sql.conf
    sql
}

# Post-Authentication
# Once we KNOW that the user has been authenticated, there are
# additional steps we can take.
post-auth {
    # Get an address from the IP Pool.
    main_pool
}

[ 660 líneas escritas ]

[root@linux1 ~]#
```

Por último edite `/etc/raddb/sites-available/inner-tunnel` y habilite la línea que contiene **sql** en la sección **authorize** {...} (línea 131 aprox.) y en la sección **session** {...} (línea 255 aprox.).



```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.8.9          Fichero: /etc/raddb/sites-available/inner-tunnel  Modificado

#
#   There are no accounting requests inside of EAP-TTLS or PEAP
#   tunnels.
#
#####

#
#   Session database, used for checking Simultaneous-Use. Either the radutmp
#   or rlm_sql module can handle this.
#   The rlm_sql module is *much* faster
session {
    radutmp

    #
    #   See "Simultaneous Use Checking Queries" in sql.conf
    sql
}

#
#   Post-authentication
#   Once we KNOW that the user has been authenticated, there are
#   additional steps we can take.
post-auth {
    # Note that we do NOT assign IP addresses here.
    # If you try to assign IP addresses for EAP authentication types,
    # it WILL NOT WORK. You MUST use DHCP.

    #
    #   If you want to have a log of authentication replies,
    #   un-comment the following line, and the 'detail reply_log'
    #   section, above.
    reply_log
}

#
#   [ línea 256/422 (68%), col 1/12 (8%), car 7250/12326 (58%) ]
#   Guardar Leer Fich Páq Ant CortarTxt Pos actual
#   Salir Justificar Buscar Páq Sig PegarTxt Ortografía
#   CTRL DERECHA
```

Regrese al símbolo de sistema y acceda a MySQL para agregar o dar de alta un usuario de prueba:

mysql -uradius -pRad123

mysql> use radius;

mysql> describe radcheck;

mysql> INSERT INTO radcheck (username, attribute, value) VALUES ('usuario1', 'Password', '123456');

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
GNU nano 2.8.9          Fichero: /etc/raddb/sites-available/inner-tunnel  Modificado

#
#   reply_log
#   [ 421 líneas escritas ]

[root@linux1 ~]# mysql -uradius -pRad123
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use radius;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> describe radcheck;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11) unsigned | NO | PRI | NULL | auto_increment |
| username | varchar(64) | NO | MUL | | |
| attribute | varchar(64) | NO | | | |
| op    | char(2) | NO | | = | |
| value | varchar(253) | NO | | | |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> INSERT INTO radcheck (username, attribute, value) VALUES ('usuario1', 'Password', '123456');
Query OK, 1 row affected (0.00 sec)

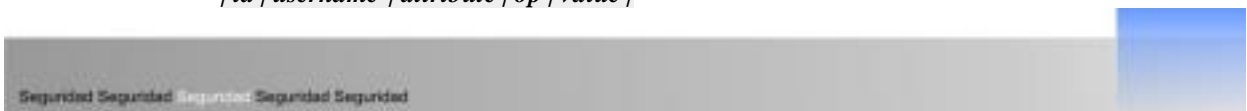
mysql> _
```

Ahora verifique que el usuario se agregó correctamente:

select * from radcheck where username='usuario1';

Debe regresar algo similar a los siguiente:

```
+-----+-----+-----+-----+-----+
| id | username | attribute | op | value |
+-----+-----+-----+-----+-----+
```





```

+-----+-----+-----+
/ 6 / usuario1 / Password / == / 123456 /
+-----+-----+-----+
1 row in set (0.00 sec)

```

Salga de la interfaz de mysql:

mysql> exit

Lo anterior equivale a añadir **usuario1 Cleartext-Password := "123456"** en el archivo **/etc/raddb/users**, pero esta es una configuración estática y nosotros agregaremos dinámicamente los usuarios requeridos a nuestra base de datos.

Iniciamos el servicio **radiusd** y después añadimos el servicio al arranque del sistema:

service radiusd start

chkconfig --level 235 radiusd on

Ahora verificaremos que el servicio **radiusd** puede autenticar a través de MySQL, para ello ejecutamos el comando **radtest**, el cual viene incluido en el paquete **freeradius-utils**, la sintaxis del comando es la siguiente: **radtest usuario password server-name puerto-nas clave compartida**.

radtest usuario1 123456 localhost 1812 testing123

Lo anterior debe devolver algo similar como lo siguiente:

```

Sending Access-Request of id 222 to 127.0.0.1 port 1812
  User-Name = "usuario1"
  User-Password = "123456"
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 1812
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=222, length=20

mysql> INSERT INTO radcheck (username, attribute, value) VALUES ('usuario1','Password','123456');
Query OK, 1 row affected (0.00 sec)

mysql> select * from radcheck where username='usuario1';
+-----+-----+-----+
| id | username | attribute | op | value |
+-----+-----+-----+
| 1 | usuario1 | Password | == | 123456 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> exit
Bye
[root@linux1 ~]# service radiusd start
Iniciando radiusd: [ OK ]
[root@linux1 ~]# chkconfig --level 235 radiusd on
[root@linux1 ~]# radtest usuario1 123456 localhost 1812 testing123
Sending Access-Request of id 19 to 127.0.0.1 port 1812
  User-Name = "usuario1"
  User-Password = "123456"
  NAS-IP-Address = 208.91.197.27
  NAS-Port = 1812
  Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=19, length=20
[root@linux1 ~]#

```

En este punto ya tenemos funcionando nuestro servidor RADIUS con autenticación local.

Habilitar la conectividad remota:

Para permitir el acceso remoto al servidor RADIUS, agregaremos unas reglas al firewall para que permita las peticiones de autenticación y auditoría:

nano /etc/sysconfig/iptables

Agregamos las siguientes reglas para permitir el tráfico UDP en los puertos de escucha de peticiones de autorización y los puertos de comunicación para funciones de auditoría de RADIUS, las agregaremos inmediatamente después de la regla del puerto 80:

-A INPUT -p udp --dport 1812 -j ACCEPT

-A INPUT -p udp --dport 1813 -j ACCEPT

-A INPUT -p udp --dport 1814 -j ACCEPT

Guardamos los cambios y reiniciamos el servicio de firewall:



service iptables restart

```
sradius (pract 3.3 error https) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp --dport 21 -m conntrack --ctstate ESTABLISHED,NEW -j ACCEPT -m comment --comment $
-A OUTPUT -p tcp -m tcp --dport 21 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT -m comment --comment $
-A INPUT -p tcp -m tcp --dport 20 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT -m comment --comment $
-A OUTPUT -p tcp -m tcp --dport 20 -m conntrack --ctstate ESTABLISHED -j ACCEPT -m comment --comment $
-A INPUT -p tcp -m tcp --sport 1024: --dport 1024: -m conntrack --ctstate ESTABLISHED -j ACCEPT -m comment --comment $
-A OUTPUT -p tcp -m tcp --sport 1024: --dport 1024: -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT -m comment --comment $
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -p udp --dport 1812 -j ACCEPT
-A INPUT -p udp --dport 1813 -j ACCEPT
-A INPUT -p udp --dport 1814 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

[ 25 líneas escritas ]

[root@slinux1 ~]# service iptables restart
iptables: Poniendo las cadenas de la política ACCEPT: filter OK
iptables: Guardando las reglas del cortafuegos: OK
iptables: Descargando módulos: OK
iptables: Aplicando reglas del cortafuegos: OK
[root@slinux1 ~]#
```

Ahora nuestro servidor RADIUS está listo para recibir peticiones de autenticación y autorización remota y de auditoría, pero aún falta registrar los clientes o dispositivos (NAS) que podrán autenticarse, así como los usuarios, esto lo haremos desde la interfaz de administración web de Doloradius.