

FreeRadius + MySQL + Mikrotik

Danilo Bedani – (dbedani@gmail.com)

Vamos configurar a distribuição de endereços IP através do Pool do FreeRadius.

Neste documento, não irei abordar a instalação e configuração do FreeRadius e MySQL, então subentende-se que a autenticação de PPPoE no Mikrotik já esteja rodando com Radius e o MySQL.

Dentro do diretório do Radius, no arquivo radiusd.conf:

```
modules {  
    ...  
    $INCLUDE sqlippool.conf  
}
```

Dentro do diretório do Radius, no arquivo sqlippool.conf:

```
sqlippool {  
  
    # pool-name = "main_pool"  
    sql-instance-name = "sql"  
    ippool_table = "radippool"  
  
    ## IP lease duration. (Leases expire even if Acct Stop packet is  
    lost)  
    ## 30min  
    lease-duration = 1800  
  
    pool-key = "%{NAS-Port}"  
  
    $INCLUDE sql/mysql/ippool.conf  
  
    sqlippool_log_exists = "Existing IP: %{reply:Framed-IP-Address} \  
        (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}  
        user %{User-Name}))"  
  
    sqlippool_log_success = "Allocated IP: %{reply:Framed-IP-Address}  
        from %{control:Pool-Name} \  
        (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}  
        user %{User-Name}))"  
  
    sqlippool_log_clear = "Released IP %{Framed-IP-Address}\  
        (did %{Called-Station-Id} cli %{Calling-Station-Id} user %{User-  
        Name}))"  
  
    sqlippool_log_failed = "IP Allocation FAILED from %{control:Pool-  
        Name} \  
        (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}  
        user %{User-Name}))"  
  
    sqlippool_log_nopool = "No Pool-Name defined \  
        (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}  
        user %{User-Name}))"  
  
}
```

Arquivo sql/mysql/ippool.conf

```
allocate-clear = "UPDATE ${ippool_table} \
SET nasipaddress = '', pool_key = 0, \
callingstationid = '', username = '', \
expiry_time = '0000-00-00 00:00:00' \
WHERE expiry_time <= NOW() - INTERVAL 1 SECOND \
AND nasipaddress = '%{Nas-IP-Address}'"

allocate-find = "SELECT framedipaddress FROM ${ippool_table} \
WHERE pool_name = '%{control:Pool-Name}' AND (expiry_time < NOW() OR \
expiry_time = '0000-00-00 00:00:00') \
ORDER BY (username <> '%{User-Name}'), \
(callingstationid <> '%{Calling-Station-Id}'), \
expiry_time \
LIMIT 1 \
FOR UPDATE"

pool-check = "SELECT id FROM ${ippool_table} \
WHERE pool_name='%{control:Pool-Name}' LIMIT 1"

allocate-update = "UPDATE ${ippool_table} \
SET nasipaddress = '%{NAS-IP-Address}', pool_key = '${pool-key}', \
callingstationid = '%{Calling-Station-Id}', username = '%{User- \
Name}', \
expiry_time = NOW() + INTERVAL ${lease-duration} SECOND \
WHERE framedipaddress = '%I' AND expiry_time IS NULL"

start-update = "UPDATE ${ippool_table} \
SET expiry_time = NOW() + INTERVAL ${lease-duration} SECOND \
WHERE nasipaddress = '%{NAS-IP-Address}' AND pool_key = '${pool- \
key}' \
AND username = '%{User-Name}' \
AND callingstationid = '%{Calling-Station-Id}' \
AND framedipaddress = '%{Framed-IP-Address}'"

stop-clear = "UPDATE ${ippool_table} \
SET nasipaddress = '', pool_key = 0, callingstationid = '', username \
= '', \
expiry_time = '0000-00-00 00:00:00' \
WHERE nasipaddress = '%{Nas-IP-Address}' AND pool_key = '${pool-key}' \
AND username = '%{User-Name}' \
AND callingstationid = '%{Calling-Station-Id}' \
AND framedipaddress = '%{Framed-IP-Address}'"

alive-update = "UPDATE ${ippool_table} \
SET expiry_time = NOW() + INTERVAL ${lease-duration} SECOND \
WHERE nasipaddress = '%{Nas-IP-Address}' AND pool_key = '${pool-key}' \
AND username = '%{User-Name}' \
AND callingstationid = '%{Calling-Station-Id}' \
AND framedipaddress = '%{Framed-IP-Address}'"

on-clear = "UPDATE ${ippool_table} \
SET nasipaddress = '', pool_key = 0, callingstationid = '', username \
= '', \
expiry_time = '0000-00-00 00:00:00' \
WHERE nasipaddress = '%{Nas-IP-Address}'"
```

```

off-clear = "UPDATE ${ippool_table} \
SET nasipaddress = '', pool_key = 0, callingstationid = '', username
= '', \
expiry_time = '0000-00-00 00:00:00' \
WHERE nasipaddress = '%{Nas-IP-Address}'"

```

Arquivo sites-enabled/default

```

post-auth {
    sqlippool
}

```

Tendo estes ajustes efetuados, vamos inserir as informações no banco de dados MySQL. Primeiro de tudo, criar uma tabela no banco de dados que o radius está se conectando, chamada **radippool**.

```

--
-- Estrutura da tabela `radippool`
--
CREATE TABLE IF NOT EXISTS `radippool` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `pool_name` varchar(30) NOT NULL DEFAULT 'main_pool',
  `FramedIPAddress` varchar(15) NOT NULL DEFAULT '',
  `NASIPAddress` varchar(15) NOT NULL DEFAULT '',
  `CalledStationId` varchar(30) NOT NULL,
  `CallingStationID` varchar(30) NOT NULL,
  `expiry_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `username` varchar(64) NOT NULL DEFAULT '',
  `pool_key` varchar(30) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=0 ;

```

Após a tabela criada, vamos inserir os IPs que queremos alocar, lembrando que o campo *expiry_time* SEMPRE deverá ser inserido como valor padrão '0000-00-00 00:00:00', pois é isto que identificará se o IP está disponível ou não.

Criaremos uma pool chamada "main_pool" e adicionaremos os IPs com este valor. Você pode criar outros nomes de *pool_name*, para diferentes perfis de clientes.

```

INSERT INTO `raddb-pppoe`.`radippool` (`id`, `pool_name`,
`FramedIPAddress`, `NASIPAddress`, `CalledStationId`,
`CallingStationID`, `expiry_time`, `username`, `pool_key`)
VALUES (NULL, 'main_pool', '192.168.3.1', '', '', '', '0000-00-
00 00:00:00', '', '');

```

Vá inserindo todos os IPs que deseja disponibilizar aos seus usuários.

Feito isso, vamos informar ao Radius a qual *pool_name* os logins autenticados fazem parte, para que o Radius possa atribuir um endereço IP disponível.

Na tabela *radcheck*, existem os usuários e senhas que serão autenticados:

id	UserName	Attribute	op	Value
27307	eder.sabino	User-Password	==	eder2015

Nesta mesma tabela, insira mais um atributo para este mesmo usuário:

UserName	Attribute	op	Value
eder.sabino	Pool-Name	:=	main_pool

Desta forma, quando este usuário autenticar, o Radius vai fazer um Check no password e no Pool-Name definido.

É neste momento que você pode criar outros nomes de ip pools, sempre na mesma tabela *radippool*, mas de um range distinto, e atribuir aos usuários.