SEG12 - Semana 1 - Sessão 11

Francisco Marcelo, Marcelo Karam e Felipe Scarel

06-08-2018

Correio Eletrônico — SMTP

As atividades desta sessão serão realizadas na máquina virtual *Server_Linux*.

Neste capítulo iremos realizar a configuração da primeira parte de um serviço de correio eletrônico: o envio e recebimento de emails entre domínios através do protocolo *Simple Mail Transfer Protocol* (SMTP). Iremos instalar e configurar o Postfix, uma dos servidores SMTP *open source* mais populares do mundo. Juntamente com o Postfix iremos instalar também o Cyrus SASL, um programa que provê módulos de autenticação plugáveis para verificarmos usuários e senhas via acesso cifrado, com criptografia TLS.

1) Instalação do servidor SMTP Postfix

Antes de instalar o Postfix, temos que corrigir alguns aspectos da nossa instalação atual. Como você se recorda da sessão 7 — DNS e NFS, configuramos a máquina *Server_Linux* com o nome de domínio `servidor.empresa.com.br`, no IP 192.168.0.10. Da mesma forma, inserimos uma entrada fictícia no DNS para uma máquina `email.empresa.com.br` no IP 192.168.0.15, que não existe em nossa topologia de rede.

Já que vamos instalar o Posfix + Cyrus na máquina *Server_Linux*, temos que apontar o nome `email.empresa.com.br` para o IP 192.168.0.10.

Contudo, não podemos tomar o caminho mais fácil, que seria criar um registro de *alias* `CNAME` do nome `email.empresa.com.br` para o nome `servidor.empresa.com.br` — a RFC 2181, seção 10.3 (https://tools.ietf.org/html/rfc2181) proíbe uso de `CNAME` para apontamentos `MX`, exigindo que esses apontamentos sejam feitos diretamente por registros `A`.

Isso exige uma série de alterações ao registro direto do domínio `empresa.com.br`, no arquivo `/etc/bind/db.empresa.com.br`, que fica como se segue:

```
$TTL 86400 ; (1 day)
$ORIGIN empresa.com.br.
@         IN    SOA           email.empresa.com.br. admin.empresa.com.br. (
                2018081200    ;Serial (YYYYMMDDnn)
                14400         ;Refresh (4 hours)
                1800          ;Retry (30 minutes)
                1209600       ;Expire (2 weeks)
                3600          ;Negative Cache TTL (1 hour)
)

@         IN    NS            email.empresa.com.br.

@         IN    MX      10    email.empresa.com.br.

email     IN    A             192.168.0.10
cliente   IN    A             192.168.0.20
windows   IN    A             192.168.0.25

meusite   IN    CNAME         email
pop       IN    CNAME         email
servidor  IN    CNAME         email
smtp      IN    CNAME         email
www       IN    CNAME         email
```

Da mesma forma, surge um problema também na resolução de registros reversos do domínio. Não é recomendado que haja múltiplos apontamentos `PTR` para o mesmo endereço IP, sob pena de obter respostas diferentes em duas *queries* DNS distintas. Vamos alterar o registro reverso no arquivo `/etc/bind/db.0.168.192`, deixando-o assim:

```
$TTL 86400 ; (1 day)
$ORIGIN 0.168.192.in-addr.arpa.
@         IN    SOA           email.empresa.com.br. admin.empresa.com.br. (
                2018081200    ;Serial (YYYYMMDDnn)
                14400         ;Refresh (4 hours)
                1800          ;Retry (30 minutes)
                1209600       ;Expire (2 weeks)
                3600          ;Negative Cache TTL (1 hour)
)

@         IN    NS            email.empresa.com.br.

@         IN    MX      10    email.empresa.com.br.

10        IN    PTR           email.empresa.com.br.
20        IN    PTR           cliente.empresa.com.br.
25        IN    PTR           windows.empresa.com.br.
```

Agora, vamos testar. Reinicie o serviço `bind` e verifique se o DNS que responde pelo domínio

empresa.com.br é, de fato, a máquina email.empresa.com.br:

```
# systemctl restart bind9.service

# dig -t NS empresa.com.br

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> -t NS empresa.com.br
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35860
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;empresa.com.br.                        IN      NS

;; ANSWER SECTION:
empresa.com.br.         86400   IN      NS      email.empresa.com.br.

;; ADDITIONAL SECTION:
email.empresa.com.br.   86400   IN      A       192.168.0.10

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Aug 12 14:50:45 -03 2018
;; MSG SIZE  rcvd: 79
```

De igual forma, verifique o registro reverso do IP 192.168.0.10, que deve retornar o nome email.empresa.com.br. Finalmente, o nome servidor.empresa.com.br torna-se agora um *alias* do CNAME email.empresa.com.br.

```
# nslookup 192.168.0.10
Server:         127.0.0.1
Address:        127.0.0.1#53

10.0.168.192.in-addr.arpa       name = email.empresa.com.br.

# nslookup servidor.empresa.com.br
Server:         127.0.0.1
Address:        127.0.0.1#53

servidor.empresa.com.br canonical name = email.empresa.com.br.
Name:   email.empresa.com.br
Address: 192.168.0.10
```

Ainda falta alterar os registros locais de nomes, nos arquivos /etc/hostname, /etc/mailname e /etc/hosts. Altere-os como mostrado a seguir:

```
# cat /etc/hostname
email

# cat /etc/mailname
email.empresa.com.br

# cat /etc/hosts
127.0.0.1                       localhost
127.0.1.1    email.empresa.com.br email
192.168.0.10 email.empresa.com.br email

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Finalmente, reinicie a máquina *Server_Linux*. No próximo login, o nome mostrado pelo *prompt* do shell deve ser `USERNAME@email:~$`.

```
# reboot

(...)

$ ssh aluno@192.168.0.10
You have new mail.
Last login: Sun Aug 12 18:00:53 2018 from 192.168.0.254
aluno@email:~$
```

Isso feito, podemos começar a atividade. Instale o Postfix + Cyrus SASL na máquina *Server_Linux* (pacotes `postfix`, `sasl2-bin` e `mailutils`). Em seguida, reconfigure o Postfix (comando `dpkg-reconfigure postfix`) de acordo com as informações da tabela abaixo:

Tabela 1. Configurações do Postfix

Parâmetro	**Valor**
Tipo geral de configuração de e-mail	Site da internet
Nome de e-mail do sistema	email.empresa.com.br
Destinatário das mensagens para root e postmaster	Em branco
Outros destinos para os quais deve aceitar mensagens	email.empresa.com.br, localhost.empresa.com.br, empresa.com.br, localhost
Forçar atualizações síncronas na fila de mensagem	Não
Redes locais	127.0.0.0/8, 192.168.1.0/24, 172.16.0.0/24, [::ffff:127.0.0.0]/104, [::1]/128

Parâmetro	Valor
Usar procmail para entrega local	Sim
Limite de tamanho da caixa postal	0
Caractere de extensão de endereço local	+
Protocolos de internet para usar	Todos

Crie um par de chaves RSA de 4096 bits e validade de dois anos para permitir conexões TLS ao seu servidor, com chave pública em `/etc/ssl/certs/smtpd.crt` e chave privada em `/etc/ssl/private/smtpd.key`. Feito isso, configure o Postfix, editando o arquivo `/etc/postfix/main.cf`, e:

- Habilite criptografia TLS em conexões oriundas dos clientes, de forma opcional;
- Use as chaves assimétricas criadas acima para implementar a cifragem TLS;
- Habilite autenticação SASL dos tipos PLAIN e LOGIN, comunicando-se com o *daemon* `saslauthd` do Cyrus — deve-se consultar a base de usuários locais via PAM para autenticação.

Atente-se para o fato de que, por padrão, o Postfix opera dentro de um ambiente `chroot`. Será necessário editar opções padrão do `saslauthd` no arquivo `/etc/default/saslauthd` para adaptar-se a esse cenário. Mais além, adicione o usuário do `postfix` ao grupo `sasl` para permitir comunicação entre os dois *daemons*.

Ao final do processo, use o comando `telnet` para testar a configuração realizada, logando no servidor SMTP com usuário `aluno` e senha `rnpesr` pelo método PLAIN.

1. Instale o servidor SMTP Postfix e o Cyrus SASL:

   ```
   # apt-get install postfix sasl2-bin mailutils
   ```

2. Reconfigure o Postfix de acordo com os dados apontados na tabela acima:

   ```
   # dpkg-reconfigure postfix
   ```

3. Gere as chaves assimétricas usando o comando `openssl`. Para gerar um par de chaves com os parâmetros solicitados, basta usar as opções `-days 730` e `-newkey rsa:4096`.

```
# openssl req -x509 -nodes -days 730 -newkey rsa:4096 -keyout
/etc/ssl/private/smtpd.key -out /etc/ssl/certs/smtpd.crt
Generating a 4096 bit RSA private key
................++
................................++
writing new private key to '/etc/ssl/private/smtpd.key'
-----
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]:BR
State or Province Name (full name) [Some-State]:DF
Locality Name (eg, city) []:Brasilia
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Empresa
Organizational Unit Name (eg, section) []:TI
Common Name (e.g. server FQDN or YOUR name) []:email.empresa.com.br
Email Address []:postmaster@empresa.com.br
```

4. Observe que a permissão da chave privada gerada pelo comando acima é muito leniente — utilize o comando `chmod 600` para corrigir isso.

```
# ls -ld /etc/ssl/private/smtpd.key
-rw-r--r-- 1 root root 3272 Ago 12 15:31 /etc/ssl/private/smtpd.key

# chmod 600 /etc/ssl/private/smtpd.key
```

5. Antes de editar o arquivo de configuração do Postfix, faça o *backup* da versão original em caso de necessidade de *rollback*:

```
# cp /etc/postfix/main.cf /etc/postfix/main.cf.orig
```

6. Edite o arquivo principal de configuração do Postfix, `/etc/postfix/main.cf`, da seguinte forma:

```
# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/smtpd.crt
smtpd_tls_key_file=/etc/ssl/private/smtpd.key

smtpd_use_tls=yes
smtpd_tls_auth_only = no
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache

smtp_use_tls = yes
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# SASL parameters
smtpd_sasl_path = smtpd
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = empresa.com.br

biff = no
append_dot_mydomain = no
readme_directory = no

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
defer_unauth_destination
myhostname = email.empresa.com.br
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = email.empresa.com.br, localhost.empresa.com.br, empresa.com.br,
localhost
relayhost =
mynetworks = 127.0.0.0/8, 192.168.0.0/24, 172.16.0.0/24, [::ffff:127.0.0.0]/104,
[::1]/128
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

7. A configuração de autenticação SASL fica no arquivo `/etc/postfix/sasl/smtpd.conf`, como se segue:

```
pwcheck_method: saslauthd
mech_list: PLAIN LOGIN
```

8. Precisamos ativar o *daemon* `saslauthd`, bem como configurá-lo para operar com o Postfix sob `chroot` no diretório `/var/spool/postfix`:

```
# cat /etc/default/saslauthd | grep '^START=\|^OPTIONS='
START=yes
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd"
```

9. Para que o Postfix consiga se comunicar com o `saslauthd` e autenticar usuários, é necessário adicioná-lo ao grupo deste:

```
# adduser postfix sasl
```

10. Reinicie ambos os *daemons* — em caso de erros, verifique os arquivos `/var/log/syslog` e `/var/log/daemon.log`:

```
# systemctl restart postfix.service
# systemctl restart saslauthd.service
```

11. Agora, basta testar o funcionamento da conexão. O único impeditivo final é que, no método PLAIN, o servidor SMTP espera o envio da combinação usuário/senha em um formato específico — `\0username\0password` — e codificado em base64. Podemos fazer isso usando o comando `openssl`, como se segue:

```
# echo -ne '\000aluno\000rnpesr' | openssl base64
AGFsdW5vAHJucGVzcg==
```

12. Finalmente, basta conectar-se ao servidor SMTP via `telnet` e fornecer as informações de autenticação obtidas acima:

```
# telnet localhost 25
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 email.empresa.com.br ESMTP Postfix

EHLO localhost
250-email.empresa.com.br
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

AUTH PLAIN AGFsdW5vAHJucGVzcg==
235 2.7.0 Authentication successful
```

2) Envio e recebimento de mensagens por *telnet*

Vamos agora testar o envio de mensagens usando o comando `telnet`, diretamente a partir do servidor SMTP. Este teste visa averiguar o funcionamento do servidor de e-mail sem a influência de configurações de clientes de e-mail (*Mail User Agents* — MUA).

Conecte-se ao servidor SMTP por `telnet` com um usuário qualquer existente na base local de usuários ou LDAP e envie email para outro usuário usando os comandos `MAIL` e `RCPT TO` do SMTP. Logue na conta do destinatário e verifique que a mensagem foi recebida.

1. Já que utilizamos o usuário `aluno` no teste da atividade anterior, vamos tentar logar com um usuário diferente. O usuário `esr`, que foi criado anteriormente na base LDAP, será nosso remetente:

```
# getent passwd | grep '^esr:'
esr:x:5000:5000:esr,,,:/home/esr:/bin/bash
```

2. Temos que gerar a *string* de autenticação base64, como feito anteriormente:

```
# echo -ne '\000esr\000rnpesr' | openssl base64
AGVzcgBybnBlc3I=
```

3. Agora, basta logar no servidor SMTP e enviar a mensagem. Usa-se, em ordem, os comandos `EHLO`, `AUTH PLAIN`, `MAIL FROM`, `RCPT TO`, `DATA` e `QUIT`, como mostrado abaixo:

```
# telnet localhost 25
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 email.empresa.com.br ESMTP Postfix

EHLO localhost
250-email.empresa.com.br
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

AUTH PLAIN AGVzcgBybnBlc3I=
235 2.7.0 Authentication successful

MAIL FROM:esr@empresa.com.br
250 2.1.0 Ok

RCPT TO:aluno@empresa.com.br
250 2.1.5 Ok

DATA
354 End data with <CR><LF>.<CR><LF>
Mensagem de teste
.
250 2.0.0 Ok: queued as C0465A075C

QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

4. Vamos finalmente entrar como o usuário `aluno` e verificar a caixa de entrada:

```
# su - aluno

$ mail
"/var/mail/aluno": 1 message 1 unread
>U   1 esr@empresa.com.br Dom Ago 12 17:29  14/475
? 1
Return-Path: <esr@empresa.com.br>
X-Original-To: aluno@empresa.com.br
Delivered-To: aluno@empresa.com.br
Received: from localhost (localhost [IPv6:::1])
        by email.empresa.com.br (Postfix) with ESMTPA id C0465A075C
        for <aluno@empresa.com.br>; Sun, 12 Aug 2018 17:29:34 -0300 (-03)
Message-Id: <20180812202940.C0465A075C@email.empresa.com.br>
Date: Sun, 12 Aug 2018 17:29:34 -0300 (-03)
From: esr@empresa.com.br
X-IMAPbase: 1534106068 10
Status: O
X-UID: 9

Mensagem de teste
```

3) Análise do log de envio

Envie uma nova mensagem de email usando o `telnet`, e monitore ao mesmo tempo o arquivo `/var/log/mail.log` por alterações. Responda, apontando a excerto do log que identifica a informação:

- Qual é o IP de origem da conexão SMTP?
- Qual o nome do usuário que efetuou login?
- Qual o endereço do destinatário da mensagem?
- Qual o método de entrega da mensagem para a caixa do usuário?

Vamos abrir um terminal monitorando por mudanças no arquivo de log do servidor SMTP com o comando `tail -f -n0 /var/log/mail.log`. Em outro terminal, vamos executar uma nova sessão de envio de email usando o comando `telnet`, como feito anteriormente.

1. Assim que a conexão é aberta, visualiza-se a mensagem:

   ```
   Aug 12 17:57:02 email postfix/smtpd[6039]: connect from localhost[::1]
   ```

   Logo, o IP de origem da conexão é `localhost`, ou 127.0.0.1.

2. Assim que o comando `RCPT TO` é enviado, surge uma nova mensagem:

```
Aug 12 18:02:15 email postfix/smtpd[6079]: F0D56A0282: client=localhost[::1],
sasl_method=PLAIN, sasl_username=esr@empresa.com.br
```

Assim, o usuário que efetuou login e deseja enviar a mensagem é o `esr@empresa.com.br`.

3. Quando o caractere ".", que delimita o final da mensagem, é enviado, vemos novas mensagens no log:

```
Aug 12 18:03:31 email postfix/cleanup[6082]: F0D56A0282: message-
id=<20180812210215.F0D56A0282@email.empresa.com.br>
Aug 12 18:03:31 email postfix/qmgr[5983]: F0D56A0282: from=<esr@empresa.com.br>,
size=333, nrcpt=1 (queue active)
Aug 12 18:03:31 email postfix/local[6091]: F0D56A0282: to=<aluno@empresa.com.br>,
relay=local, delay=81, delays=81/0/0/0, dsn=2.0.0, status=sent (delivered to
command: procmail -a "$EXTENSION")
Aug 12 18:03:31 email postfix/qmgr[5983]: F0D56A0282: removed
```

No campo `to=` da terceira linha do log acima pode-se observar que o endereço do destinatário é `aluno@empresa.com.br`. Na mesma linha, vê-se que o método de entrega é via *relay* local, usando o programa auxiliar `procmail`.