

Gestion de la distribución de los correos electrónicos

El servidor MTA almacena localmente los mensajes recibidos. El formato de almacenamiento de los mensajes puede ser `mbox` o `maildir`.

Después de haber recibido un correo electrónico, se puede configurar el servidor Postfix para que haga una llamada a un programa externo (LDA, *Local Distribution Agent*), encargado de filtrar y de clasificar antes de almacenar el correo electrónico.

1. Formatos de almacenamiento de los mensajes

El servidor MTA almacena localmente los mensajes recibidos. El formato de almacenamiento de los mensajes puede ser `mbox` o `maildir`.

a. El formato mbox

El formato `mbox` es el más antiguo. En este formato, todos los mensajes del usuario se concatenan en el interior de un único archivo de buzón de correo. El principio de cada mensaje está identificado por la secuencia de caracteres `From` al principio de la línea.



Como se trata de un archivo de texto único, se podría corromper fácilmente si dos procesos acceden en escritura simultáneamente a él.

Ejemplo

Archivo de buzón de correo de un usuario, en formato `mbox`:

vi /var/mail/aca

From yo@mydom.com Sun May 24 08:57:10 2020

Return-Path: <yo@mydom.com>

X-Original-To: aca

Delivered-To: aca@centos7.localdomain

Received: from mydom.com (localhost [IPv6:::1])

by centos7.localdomain (Postfix) **with** ESMTTP id 1B4408AC247

for <aca>; Sun, 24 May 2020 08:56:11 +0200 (CEST)

Message-Id: <20200524065632.1B4408AC247@centos7.localdomain>

Date: Sun, 24 May 2020 08:56:11 +0200 (CEST)

From: yo@mydom.com

Status: RO

X-Status: A

Hola Alejandro,

espero que todo vaya bien

From root@centos7.localdomain Sun May 24 09:01:54 2020

Return-Path: <root@centos7.localdomain>

X-Original-To: aca

Delivered-To: aca@centos7.localdomain

Received: by centos7.localdomain (Postfix, from userid 0)

id 5A0668AC1CE; Sun, 24 May 2020 09:01:54 +0200 (CEST)

Date: Sun, 24 May 2020 09:01:53 +0200

To: aca@centos7.localdomain

Subject: ess

User-Agent: Heirloom mailx 12.5 7/5/10

MIME-Version: 1.0

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

Message-Id: <20200524070154.5A0668AC1CE@centos7.localdomain>

From: root@centos7.localdomain (root)

Status: RO

hello from root

[...]

b. El formato maildir

El formato `maildir` usa una estructura de directorios para el almacenamiento de los correos electrónicos recibidos por un usuario. Cada mensaje está almacenado en un archivo independiente.

Un directorio de correo con formato `maildir` contiene tres subdirectorios: `tmp`, `new` y `cur`. Los mensajes se almacenan primero en `tmp`, y después se desplazan a `new`. Después de haberlos leído, se desplazan a `cur`. Cada correo electrónico está almacenado en un archivo cuyo nombre no guarda ninguna relación con el título o contenido del mensaje.

c. Uso del formato `maildir` por Postfix

Por defecto, el servidor Postfix usa el formato `mbox` para almacenar los correos electrónicos.

Si queremos usar el formato más reciente `maildir`, hay que configurarlo en el archivo `main.cf` usando la directiva siguiente:

```
home_mailbox = Maildir/
```

El directorio se creará en el directorio de conexión del usuario en cuanto se reciba el primer correo electrónico.



El comando `mail`, para la lectura de los mensajes, solo soporta el formato `mbox`.

Ejemplo

Directorio de recepción de correos electrónicos, con formato `maildir`, del usuario `aca`:

```
ls -l /home/aca/Maildir/*
```

/home/aca/Maildir/cur:

total 0

/home/aca/Maildir/new:

total 4

-rw-----. 1 aca aca 542 24 mayo 11:27
1590312435.Vfd00lce5f17M830163.centos7

/home/aca/Maildir/tmp:

total 0

Hay un mensaje que todavía no se ha leído:

cat /home/aca/Maildir/new/1590312435.Vfd00lce5f17M830163.centos7

Return-Path: <root@centos7.localdomain>

X-Original-To: aca

Delivered-To: aca@centos7.localdomain

Received: by centos7.localdomain (Postfix, from userid 0)

id BF4128A0DBF; Sun, 24 May 2020 11:27:15 +0200 (CEST)

Date: Sun, 24 May 2020 11:27:15 +0200

To: aca@centos7.localdomain

Subject: prueba mdir

User-Agent: Heirloom mailx 12.5 7/5/10

MIME-Version: 1.0

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

Message-Id: <20200524092715.BF4128A0DBF@centos7.localdomain>

From: root@centos7.localdomain (root)

Hola de parte de root

2. procmail

Podemos configurar el servidor Postfix para que use un programa externo para tratar los correos electrónicos recibidos antes de almacenarlos. Los programas más utilizados son procmail y, más recientes todavía, las aplicaciones que usan el lenguaje de filtrado Sieve.

Los tratamientos permiten filtrar mensajes y clasificarlos aplicando diferentes criterios.

a. Configuración de Postfix

La directiva `mailbox_command` del archivo de configuración Postfix define el programa que se usará para tratar los mensajes entrantes (es el caso de procmail).

Sintaxis

```
mailbox_command = RutaPrograma
```

Para configurar una aplicación LDA (*Local Delivery Agent*), hay que usar la directiva:

```
virtual_transport .
```

Sintaxis

```
virtual_transport = NombreLDA
```

b. Configuración de procmail

La configuración de procmail, muy completa pero compleja, ya no se incluye en el programa de la certificación LPIC-2, pero algunos ejemplos permiten conocer sus principios de base.

procmail usa un archivo de configuración, `.procmailrc`, que se encuentra en el directorio de conexión del usuario. Este archivo contiene reglas que se tienen que aplicar, en su orden de aparición, a todos los correos electrónicos entrantes. El tratamiento se termina en cuanto una regla es aplicable.

Formato de una regla

```
:0 [Tipo]
Condición
Acción
```

Donde:

<code>:0 [Tipo]</code>	Principio de la regla + tipo de regla (H encabezado y B cuerpo del mensaje).
Condición	Criterio de selección del mensaje (expresión regular).
Acción	Tratamiento que se aplicará al mensaje.

Ejemplos

Almacenar en el directorio *todosmisamigos/borja* todos los mensajes entrantes cuyo nombre de dominio termine por *borja*:

```
:0
* ^From.*borja
todosmisamigos/borja
```

Imprimir todos los correos electrónicos con un tamaño inferior a 1000 bytes.

```
:0
* < 1000
| /usr/bin/lp
```

3. Sieve

Sieve (« colador » en inglés) se trata de un lenguaje especializado en el filtrado de correos electrónicos definido en la RFC 5228. Existen programas de distribución local de correos electrónicos (LDA, *Local Delivery Agent*) que lo implementan y que hacen competencia a procmail. Es el caso, por ejemplo, de Dovecot.

El principio de concepción del lenguaje Sieve ha sido limitar sus funcionalidades para

hacerlo simple y robusto, para poder garantizar en todo lo posible su seguridad de uso.

a. Componentes de un filtro Sieve

Un filtro Sieve es un script que contiene una lista de comandos.

El comando `require` permite declarar el uso de módulos opcionales, identificados por palabras clave. Su sintaxis es la siguiente:

```
require ["PalabraClave1", ..., "PalabraClaveN"];
```

Una cadena de caracteres debe estar rodeada por el carácter `"`.

```
["Cadena1", ..., "CadenaN"]
```

Esta sintaxis permite definir una lista de cadenas de caracteres, que se puede utilizar con la mayoría de los comandos.

Los comentarios empiezan con el carácter `#` al principio de línea hasta el final de la línea, o se encuentran en una o en varias líneas rodeadas por `/ * y */`.

b. La estructura if

Los comandos se agrupan, en la mayoría de casos, en estructuras `if`, según la sintaxis siguiente:

```
if CriterioSelección {
  Comando1
  ...
  ComandoN
}
```

La estructura asocia uno varios comandos que se aplican secuencialmente al correo electrónico objeto del análisis, si corresponde a los criterios de selección especificados.

Un filtro Sieve puede estar compuesto de distintas estructuras `if`, que se ejecutarán siguiendo su orden de aparición. Si una estructura no se aplica al correo electrónico objeto del análisis, se pasará a la siguiente.

Si no se aplica ninguna estructura `if`, el email objeto del análisis será aceptado.

Una estructura `if` acepta las palabras clave `elsif` y `else` para efectuar pruebas anidadas:

```
if CriterioSelección {}
elsif CriterioSelección {}
else {}
```

c. Los comandos de acción

Dentro de una estructura `if`, podemos encontrar los comandos habituales siguientes:

<code>discard</code>	Rechaza silenciosamente el correo electrónico objeto del análisis.
<code>reject Motivo</code>	Rechaza el correo electrónico objeto del análisis enviando el mensaje <code>Motivo</code> al emisor.
<code>fileinto Directorio</code>	Almacena el correo electrónico objeto del análisis en el directorio especificado.
<code>Redirect Dirección</code>	Transfiere el correo electrónico objeto del análisis hacia la dirección especificada.
<code>stop</code>	Termina la ejecución del script, sin tratar los otros comandos.
<code>keep</code>	Acepta el correo electrónico objeto del análisis (comando por defecto).

d. Los criterios de selección

Los criterios de selección se pueden aplicar a distintas partes de un mensaje.

Direcciones

Las direcciones se pueden comprobar, con la palabra clave `address`, a partir de los campos `From:`, `To:` y `Sender:` del mensaje.

Una dirección de correo electrónico está compuesta por dos partes, que se pueden comparar de manera separada, según el formato siguiente:

`:localpart@:domain`

También se puede comprobar el conjunto de la dirección, usando la palabra clave `:all`.

La sintaxis del comando es la siguiente:

`address OperatorComp ParteDir CampoDir Cadena`

Donde:

`OperatorComp` Operador de comparación (`:is`, `:contains`, `:matches`).

`ParteDir` `:all`, `:domain` o `:local` (defecto: `:all`).

`CampoDir` `"to"`, `"from"`, `"sender"`.

`Cadena` Cadena para la comparación.

Campos de encabezado

Se puede comprobar el encabezado, usando la palabra clave `header`. También se puede especificar un campo particular del encabezado para limitar la comprobación.

e. Los operadores de comparación

Los operadores de comparación con una cadena de caracteres son los siguientes:

<code>:is</code>	Correspondencia completa.
<code>:contains</code>	Correspondencia con una parte de la cadena.
<code>:matches</code>	Correspondencia con un modelo con caracteres joker (<code>?</code> y <code>*</code>).
<code>allof (Lista)</code>	Si la cadena corresponde al conjunto de pruebas de la lista.
<code>anyof (Lista)</code>	Si la cadena corresponde a una prueba de la lista.
<code>Size [Oper]</code> <code>Val[K M]</code>	Si el tamaño del mensaje es (<code>:over</code> o <code>:under</code>) del tamaño indicado.

f. La extensión vacation

Esta extensión del lenguaje Sieve, implementada en particular por Dovecot, permite gestionar respuestas automáticas a los correos electrónicos recibidos en ausencia del usuario.

La extensión usa el comando `vacation` y distintos parámetros.

Ejemplo

Script Sieve para dar una respuesta automática en caso de ausencia:

```
require ["vacation"];
vacation
# Se dará solamente una respuesta cada día al mismo emisor
```

```

:days 1
:subject "Respuesta en ausencia "
# Direcciones de los destinatarios que podrán provocar una respuesta automática.
:addresses ["aca@mindns.es", "admin@mindns.es"]
"No estoy disponible. Por favor, contacte con el servicio de atención al cliente.
Atentamente,
Alejandro";

```

g. Ejemplo

Este script presenta algunos comandos Sieve para filtrar correos electrónicos.



La implementación de filtros Sieve no es suficiente para proteger de manera eficaz a un servidor de correo electrónico contra el spam, se aconseja el uso de un programa especializado.

```

# Ejemplo simple de filtro Sieve

# Declaración de los comandos opcionales usados.
require ["fileinto", "reject"];

# Se rechazarán los mensajes muy grandes (> 10 MB)
if size:over 10000K {
    reject "No se aceptarán los mensajes que superen un tamaño de 10 MB.";
}

# Los mensajes enviados desde la lista de difusión intra se almacenarán
en un directorio especial
elsif address:is ["From", "To"] "mailinglist.intra@mindns.es" {
    fileinto "INBOX.ListaIntra";
}

# Se enviarán los mensajes no profesionales a spam

```

```
elseif header:matches "Subject" ["*sexo*", "*Viagra*", "*herencia*"] {  
    fileinto "INBOX.spam";  
}
```