DKIM

Esta son las pruebas de implementación de DKIM que he seguido en un servidor de pruebas con SuSE Linux, Amavis y SpamAssassin.

Se trata de un caso muy particular, aplicado a nuestro servidor para unos requerimientos mucho menores de lo que probablemente habrá en instituciones realmente grandes, como las Universidades. Pero, si puede servir de guía para alguien que lo quiera implementar, sería suficiente.

1. Introducción

Más o menos todos sabemos lo que es DKIM. Brevemente, es un sistema de autenticación que permite verificar que el correo llega realmente de quien dice ser el remitente. La idea es similar a la del conocido SPF. En éste caso, lo que comprobamos es que el servidor desde donde proviene el correo está autorizado para enviar correo del dominio desde donde llega. DKIM persigue un objetivo similar, pero de forma diferente. Se utiliza un sistema de criptografía asimétrica y funciones de hash. Al igual que SPF, utiliza el DNS para su propósito. Pero en este caso, lo que DKIM hace es publicar mediante el DNS su clave pública para que servidores remotos puedan verificar los correos firmados con la clave privada.

De esta forma, el servidor remitente calcula un hash utilizando el cuerpo del mensaje y alguna de sus cabeceras y lo cifra mediante la clave privada. Cualquier servidor remoto que quiera verificar el correo, necesitará de la clave pública obtenida a través del DNS del dominio de origen.

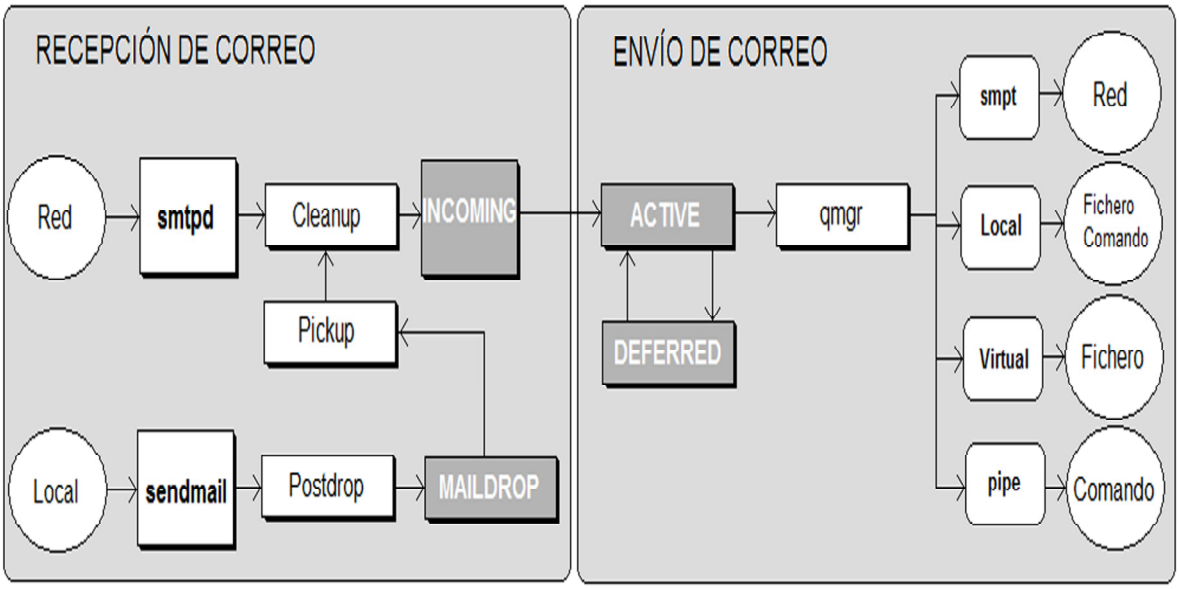
Como ventajas frente a SPF tiene el hecho de evitar los problemas de los fordward, ya que no se mira quién nos entrega el mensaje, sino quién lo originó. Sin embargo, sí es sensible a las modificaciones que pueda sufrir el mensaje durante su transmisión (por ejemplo con disclaimers posteriores a la firma)

2. Pasos previos a la implementación

La forma que hemos utilizado en nuestro caso ha sido mediante Amavis y Postfix. Para la verificación de firmas que nos llegan, utilizaremos SpamAssassin. Para éste último, nos va a ser necesario tener instalado **perl-Mail-DKIM** (así es como se llama en SuSE y puede ser instalado desde YaST, pero en otras versiones de linux, el paquete es **libmail-dkim-perl**). Igualmente, necesitamos habilitar el plugin en SpamAssassin: **Mail::SpamAssassin::Plugin::DKIM**, comentando la línea que hace referencia al mismo en el fichero **/etc/mail/spamassassin/v312.pre** (en otras distribuciones, el fichero puede estar en caminos diferente, como /etc/spamassassin/v312.pre). Supongo que cada uno deberá buscar estos requerimientos adecuados a sus distribuciones e instalaciones propias.

3. Breve vistazo al funcionamiento de Postfix

Un esquema muy básico de cómo actúa Postfix al recibir un correo es el que sigue:



Los cuadros blancos representan programas y servicios, mientras que los oscuros son las colas. Cuando decimos que algo viene de la **red**, es cualquier correo que entra en el servidor desde la red. Es decir, si el relay de una máquina de nuestra propia red es el servidor, enviará todos los correos a dicho servidor y este los verá como provinientes de la **red**. Solo los correos generados en el propio servidor, serán **locales**.

3.a Los correos que llegan desde la Red

En **/etc/postfix/master.cf** esto se puede controlar muy bien. Las líneas implicadas, en nuestro caso, son:

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
#
submission    inet   n      -      n      -      10     smtpd
              -o smtpd_sasl_auth_enable=yes
              -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
              -o smtpd_enforce_tls=yes
#
#------------------------------------------------------------------------
#
smtps         inet   n      -      n      -      10     smtpd
              -o smtpd_sasl_auth_enable=yes
              -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
              -o smtpd_tls_auth_only=yes
              -o smtpd_tls_wrappermode=yes
#
#------------------------------------------------------------------------
#
smtp          inet   n      -      n      -      10     smtpd
#
```

Si nos fijamos, veremos que tenemos los 3 tipos posibles de entradas: **submission, smtps** y **smtp**, los tres relacionados con el servicio smtpd que es el que se encarga de entregar el correo. Esto significa que dependiendo del puerto al que enviemos el correo, usaremos submission (**587**), smtps (**465**) o smtp (**25**). Esto es lo que configuramos en los clientes, diciendo que utilizaremos el servidor 'fulanito' para smtp por el puerto 465 (smtps por ejemplo). Para usar los dos primeros (submission y smtps) es necesario estar autentificado, es decir, tener una cuenta válida, estar dado de alta en **sasl_senders** y configurar correctamente el cliente. De esta forma, cualquier cuenta válida puede autentificarse y siempre entrará en caserv por submission o smtps. Sin embargo, un correo que venga del exterior (desde alguien que no tenga cuenta en Calar Alto o no esté autentificado y usando un puerto seguro en su cliente), siempre lo hará por la tercera opción, es decir, por el smtp, puerto 25.

3.b Correos considerados Locales

En este caso es un correo originado directamente en el propio servidor. No utilizará el servicio smtpd. Esto solo se aplica a correos originados en el propio servidor de correo

4. Amavis

El Amavis funciona en base a puertos. Por unos puertos recibe información (el correo) y por otros manda la respuesta (una vez aplicados spam, anti-virus y firmas). El puerto por el que hacemos escuchar a Amavis es el **10024** y la información retornada aparecerá en el puerto **10025**. En nuestro caso, hasta la fecha, esto ha sido así, tanto si recibíamos o enviábamos un correo (independientemente de por dónde se recibiese en Postfix el correo). No se hacía discriminación alguna del lugar de procedencia del correo. Para analizar todos los correos, se utilizaba la siguiente entrada en el otro fichero de configuración de Postfix, **/etc/postfix/main.cf**:

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

De esta forma, **todos** los correos, de red o locales, eran analizados. En el fichero **/etc/postfix/master.cf** se configuraba el puerto de escucha de la respuesta de amavis:

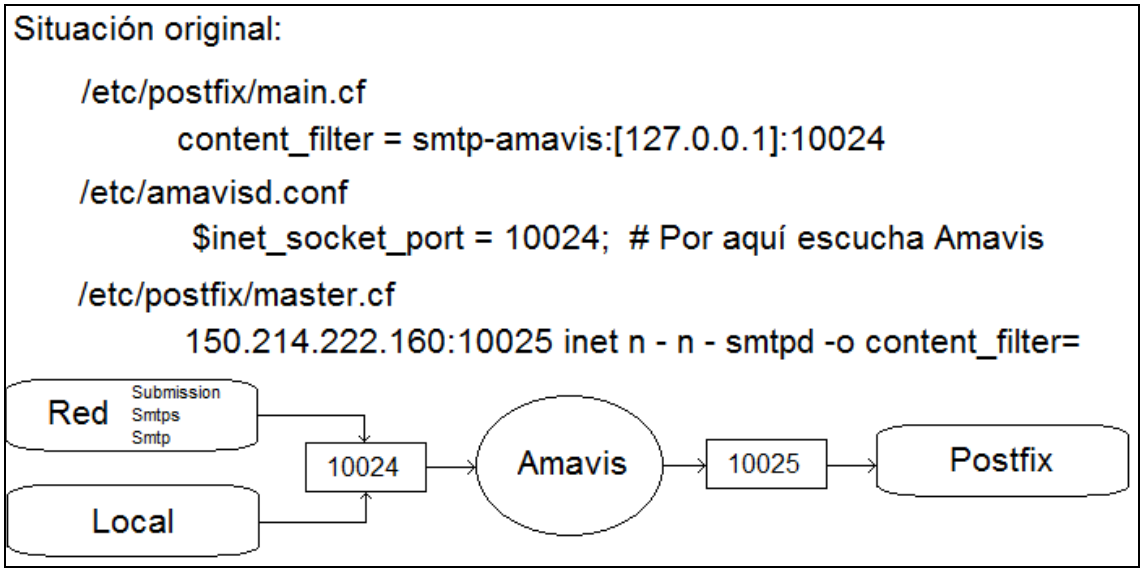
```
# Escucha en el 10025 la respuesta del Filtro-Amavis
150.214.222.160:10025 inet n - n -  - smtpd -o content_filter=
#
# Llamada al filtro - Amavis (a traves del 100024)
smtp-amavis unix -      -       n     -       15  lmtp
    -o lmtp_data_done_timeout=1200
    -o lmtp_send_xforward_command=yes
```

Finalmente, en el fichero de configuración de Amavis: **/etc/amavisd.conf** definíamos que amavis solo escuchaba en el puerto 10024 y no definíamos ningún tipo de acción asociada a ese puerto, para que amavis realizase las que son por defecto: anti-spam y anti-virus.

```
$inet_socket_port = 10024;    # Puerto en el que escucha Amavis
```

5. Implementación de DKIM

El problema fundamental que nos encontramos con el planteamiento de más arriba es que todos los correos pasan a Amavis directamente, tanto los que vienen desde el exterior como los nuestros. Y una vez en amavis, se realizan las funciones por defecto. Esto lo podemos ver en la siguiente figura:



Por supuesto, a las funciones de anti-virus y anti-spam de amvis, podríamos añadir que se firmasen los correos, pero con la estructura de más arriba, se firmarían tanto los nuestros como los que lleguen del exterior. Y esto no es lo que queremos.

Afortunadamente, en Amavis podemos definir otros puertos que escuchen y realicen acciones alternativas. Para ello cambiamos la línea del fichero de configuración de amavis, **/etc/amavisd.conf**, por:

```
$inet_socket_port = [10024,10026];  # Escuchamos en dos puertos
```

Lo que queremos es **diferenciar entre correos que lleguen autentificados (nuestros) y correos que vengan de fuera o no autentificados**. Estos últimos pueden ser nuestros también y se pueden dar si no configuramos bien el cliente de correo o decidimos usar expresamente el puerto 25 en vez de uno seguro. Y una vez diferenciados, mandar por el puerto 10024 (los no autentificados) y por el 10026 (los autentificados). En cualquier caso, la respuesta de Amavis siembre volverá al 10025.

La forma de diferenciar desde dónde nos van a llegar los correos es mediante los puertos submission, smtps y smtp. Lo que necesitamos es hacer un par de modificaciones en Postfix. La primera es en el fichero **/etc/postfix/master.cf** para decir qué acción seguirá amavis (a qué puerto enviarle el correo) dependiendo desde donde le llegue el correo al Postfix:

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
#
submission inet n   -      n      -      10     smtpd
             -o smtpd_sasl_auth_enable=yes
             -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
             -o smtpd_enforce_tls=yes
             -o content_filter=smtp-amavis:[127.0.0.1]:10026
#
#------------------------------------------------------------------------
#
smtps    inet       n      -      n      -      10     smtpd
             -o smtpd_sasl_auth_enable=yes
             -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
             -o smtpd_tls_auth_only=yes
             -o smtpd_tls_wrappermode=yes
             -o content_filter=smtp-amavis:[127.0.0.1]:10026
#
#------------------------------------------------------------------------
#
smtp     inet n      -      n      -      10     smtpd
             -o content_filter=smtp-amavis:[127.0.0.1]:10024
```

Con esto le decimos al Postfix que cualquier mensaje que llegue al submission o al smtps (autentificados, por tanto), los pase al Amavis por el puerto 10026 y los que no lleguen así, los pase por el puerto 10024 como antiguamente. Además, la segunda modificación hay que hacerla en el otro fichero de configuración, **/etc/postfix/main.cf**, en donde debemos dejar en blanco la línea que antes mostrábamos con el content_filter:

```
             content_filter =
```

De esta manera, no se envían los mensajes, independientemente de su procedencia, por el 10024, sino que se tiene en cuenta desde dónde vienen y se decide en función de eso.

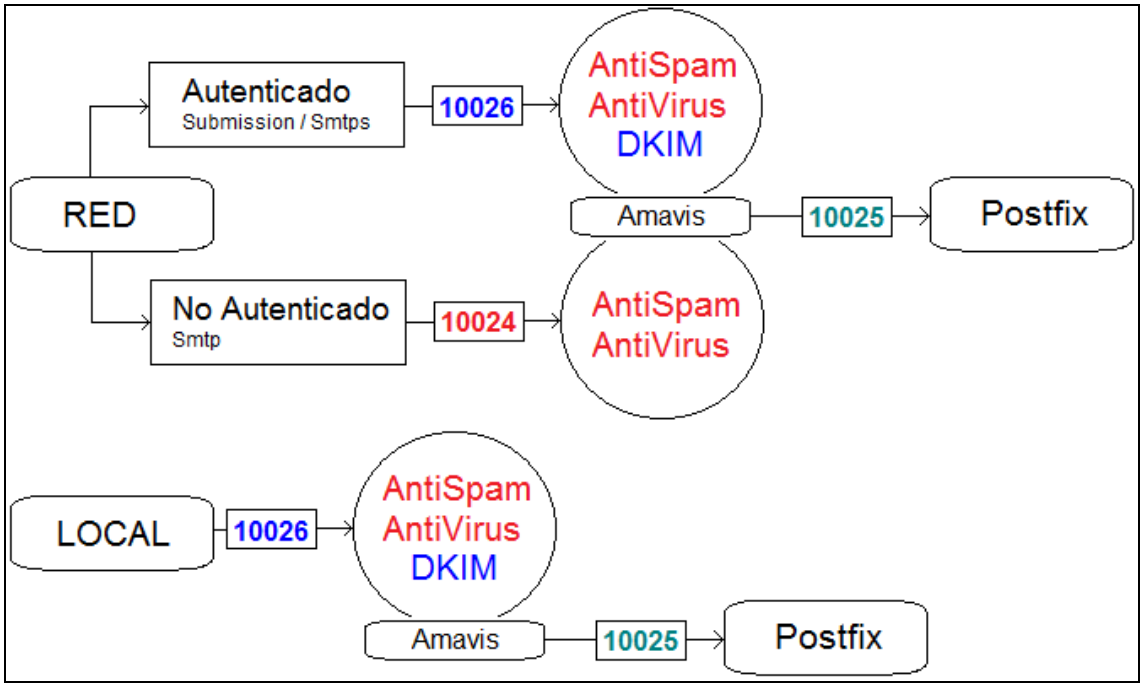
Una vez preparado esto, debemos modificar la configuración de Amavis en **/etc/amavisd.conf** para que por el 10026 realice algo más que lo que por defecto realiza en 10024. Esto es, firmar los mensajes. Para ello tenemos que definir una política para el puerto 10026:

```
       $interface_policy{'10026'} = 'FIRMA_DKIM';

       $policy_bank{'FIRMA_DKIM'} = {    # Si llega  a sub o smtps, firmamos
           bypass_spam_checks_maps => 0, # Que analice el mensaje
           originating => 1,             # Mail einviado por nuestro smtp
           smtpd_discard_ehlo_keywords => ['8BITMIME'], # Necesario para firmar
       };

       $enable_dkim_verification = 1;    # Chequeo de entrantes
       $enable_dkim_signing = 1;         # Firma
       dkim_key('caha.es', 'personal', '/etc/dkim/caha.es.key.pem'); # Clave
```

De esta forma, ya tenemos lo básico para obtener la configuración que íbamos buscando y que podemos ver en la imagen siguiente:



La última línea del fichero /etc/amavisd.conf:

```
dkim_key('caha.es', 'personal', '/etc/dkim/caha.es.key.pem'); # Clave
```

es muy importante y nos enlaza ya directamente con el DNS, ya que es donde vamos a definir el registro dkim y sus características. El selector 'personal' puede ser el nombre que se quiera. Esa línea también indica el fichero donde guardaremos la clave privada y que veremos cómo se genera en el siguiente punto.

Con esto, la parte que viene de la Red estaría completa. Ahora nos faltaría tratar los mensajes locales, es decir, aquellos que vienen directamente desde el propio servidor.

Un punto donde podemos enviar el correo al Amavis es el **pickup**. Para ello, añadimos la línea siguiente al **/etc/postfix/master.cf:**

```
pickup      fifo    n    -    n    60    1    pickup
        -o content_filter=smtp-amavis:[127.0.0.1]:10026
```

De esta forma, cualquier mensaje enviado por el propio servidor será analizado y firmado, aunque se utilice el puerto 25 (con el mailx, por ejemplo). En nuestro caso de Calar Alto, esto es prácticamente irrelevante ya que nadie se puede logar en el servidor y éste sólo manda mensajes locales de los programas de control que nuestro Departamento usa.

6. DNS

Como se ha dicho más arriba, una de las partes implicadas en el sistema es el DNS, ya que es ahí donde publicaremos nuestra clave pública para que pueda ser consultada por los servidores externos.

Lo primero que debemos hacer es crear el fichero para contener la clave privada. Las claves pueden ser generadas con el comando **openssl genrsa** o con el comando propio de amavis:

amavisd genrsa <fichero_clave_privada> [num_bits]

En nuestro caso, utilizamos el propio amavis y dejamos el número de bits que utiliza por defecto (1023):

```
# mkdir /etc/dkim
# amavisd genrsa /etc/dkim/caha.es.key.pem
```

(cambiar caha.es por el dominio adecuado)

Ese fichero debe ser **vscan:vscan** (usuario para amavis) para que amavis lo pueda leer. Solo amavisd debe tener acceso a esta clave.

Y, acto seguido, preparamos nuestro servidor DNS con la clave pública para que pueda ser consultada por los servidores remotos:

```
# amavisd showkeys
; key#1, domain caha.es, /etc/dkim/caha.es.key.pem
personal._domainkey.caha.es.    3600 TXT (
"v=DKIM1; p="
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmWLdDJRaGDpe/tYr0Ue/af4JF"
"txqp4AVCPLYQ6OJAMsbjg6FKPHC9BkbIfF/Z7MIsxVgF7gKVUcDkY8Cn8g2HnGJC"
"ZTRlp3qQKdyi8OaSWy3IOv/RKvYYaoGz1bjg+kj2uasWwn54UGSsHfWGoJmBEfDm"
"T6Z7sVoEi6XS+M/nJQIDAQAB")
```

Esa salida es la que hay que añadir a la zona de nuestro dominio en el servidor DNS:

```
personal._domainkey.caha.es.    3600 TXT (
  "v=DKIM1; p="
  "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmWLdDJRaGDpe/tYr0Ue/af4JF"
  "txqp4AVCPLYQ6OJAMsbjg6FKPHC9BkbIfF/Z7MIsxVgF7gKVUcDkY8Cn8g2HnGJC"
  "ZTRlp3qQKdyi8OaSWy3IOv/RKvYYaoGz1bjg+kj2uasWwn54UGSsHfWGoJmBEfDm"
  "T6Z7sVoEi6XS+M/nJQIDAQAB")
```

Es muy importante que, para que el showkeys funcione correctamente, tengamos un línea como la que he explicado arriba: dkim_key(…) puesta en el /etc/amavisd.conf.

Obviamente, el fichero a modificar dentro del Bind dependerá de la instalación que cada uno tenga. A modo de ejemplo, en nuestro caso en el named.boot designamos el fichero db.caha como el fichero donde se definen todos los registros de nuestras máquinas. Y es en ese fichero db.caha donde hemos situado un include que apunta a otro fichero (spcl.caha) que es donde definimos los registros NS, MX y TXT, tanto para SPF como para DKIM.

Genéricamente, en named boot debe contener algo así:

named.boot

```
directory     <directorio_para_named>
cache         .          <fichero_cache>     → NS autoritativos para la zona root
primary       <dominio>  <fichero_maquinas>  → Aquí o en include desde aquí
```

En el caso particular del Calar Alto:

```
Primary para caha.es → db.caha → (include) → spcl.caha
```

Como se ve, el primario para el dominio caha.es es el fichero db.caha y dentro de ése fichero hemos definido un include que apunta a otro fichero donde tenemos todos los registros NS, MX y TXT.

7. Pruebas

Una vez hecho, tenemos 3 posibles pruebas. **La primera** es la más sencilla:

```
# amavisd testkeys
TESTING#1: personal._domainkey.caha.es       => pass
```

La segunda prueba que podemos hacer si tenemos cuenta en algún servidor que chequee DKIM (como gmail), es mandarnos un correo a dicha cuenta. Si miramos el mensaje con cabeceras, deberíamos ver algo así:

```
Authentication-Results: mx.google.com; spf=pass (google.com: domain of
guindos@caha.es designates 150.214.222.10 as permitted sender)
smtp.mail=guindos@caha.es; dkim=pass header.i=@caha.es
Received: from localhost (caserv.caha.es [150.214.222.10])
        by caserv.caha.es (Postfix) with ESMTP id 29D3CC0
        for <eguindos@gmail.com>; Thu,  9 Dec 2010 14:58:11 +0000 (GMT)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=caha.es; h=
        x-mailer:content-type:content-type:mime-version:message-id:date
        :date:subject:subject:from:from:received:received; s=personal;
         t=1291906686; bh=dy9hTr+UewkzfUC9xlL2Mr5OmBdSGpgE8hHKyJCwMNs=; b=
        QvBs3+/F5IezrVfniSd4TdtBvM8nFLtdL+8OeZoLQOCfSK/DgnA6tM1ZfJjN6+4n
        vVz5mCKmgjg8a+L3wL/Bm7RbptpklIXHniYtCm9MRzBeIfDTUSCicitoPZF5vm5+
        RXXvOvKWgGfEVp3ZAFhTVF+ANT/OxZXLmDMRmX1xph8=
```

Finalmente, la **última prueba** es la más maja de todas, y consiste en enviar un mensaje (sin asunto ni cuerpo) a la siguiente varias posibles direcciones de auto-chequeo. Una de ellas, puede ser: **check-auth2@verifier.port25.com.** Este correo nos responderá automáticamente con un exhaustivo análisis del estado de nuestro servidor, en donde deberemos comprobar que el DKIM está funcionando. En nuestro caso, con el servidor de pruebas, lo que recibimos concerniente al DKIM fue:

```
------------------------------------------------------------
DKIM check details:
------------------------------------------------------------
Result:          pass (matches From: guindos@caha.es)
ID(s) verified: header.d=caha.es
Canonicalized Headers:
    from:guindos@caha.es'20'(Enrique'20'de'20'Guindos)'0D''0A'
    message-id:<20101013083753.52EACB804E8@oxserv.caha.es>'0D''0A'
    content-transfer-encoding:quoted-printable'0D''0A'
    content-type:text/plain;'20'charset=us-ascii'0D''0A'
    mime-version:1.0'0D''0A'
    user-agent:Heirloom'20'mailx'20'12.2'20'01/07/07'0D''0A'
    date:Wed,'20'13'20'Oct'20'2010'20'10:37:53'20'+0200'0D''0A'

received:by'20'oxserv.caha.es'20'(Postfix,'20'from'20'userid'20'1000)'20'id'20
'52EACB804E8;'20'Wed,'20'13'20'Oct'20'2010'20'10:37:53'20'+0200'20'(CEST)'0D''
0A'

received:from'20'oxserv.caha.es'20'([127.0.0.1])'20'by'20'localhost'20'(oxserv
.caha.es'20'[127.0.0.1])'20'(amavisd-
new,'20'port'20'10026)'20'with'20'LMTP'20'id'20'Y2zoXqSwbxgA'20'for'20'<check-
auth2@verifier.port25.com>;'20'Wed,'20'13'20'Oct'20'2010'20'10:37:53'20'+0200'
20'(CEST)'0D''0A'
    dkim-signature:v=1;'20'a=rsa-
sha256;'20'c=relaxed/simple;'20'd=caha.es;'20'h=from'20':from:message-
id:content-transfer-encoding:content-type'20':content-type:mime-version:user-
agent:date:date:received'20':received;'20's=personal;'20't=1286959073;'20'bh=5
g22i38m5ab1CDk2XO5BVyX7'20'c5UtL2kHd7C21wyYSZE=;'20'b=

Canonicalized Body:
    ='0D''0A'

DNS record(s):
    personal._domainkey.caha.es. 3600 IN TXT "v=DKIM1;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmWLdDJRaGDpe/tYr0Ue/af4JFtxqp4AVCPLYQ
6OJAMsbjg6FKPHC9BkbIfF/Z7MIsxVgF7gKVUcDkY8Cn8g2HnGJCZTRlp3qQKdyi8OaSWy3IOv/RKv
YYaoGz1bjg+kj2uasWwn54UGSsHfWGoJmBEfDmT6Z7sVoEi6XS+M/nJQIDAQAB"

NOTE: DKIM checking has been performed based on the latest DKIM specs (RFC
4871 or draft-ietf-dkim-base-10) and verification may fail for older versions.
If you are using Port25's PowerMTA, you need to use version 3.2r11 or later to
get a compatible version of DKIM.
```

Otro de los servidores que hemos probado es: **auth-results@verifier.port25.com**

Y su mensaje de respuesta sería el siguiente:

```
============================================================
Summary of Results
============================================================
SPF check:          pass
DomainKeys check:   neutral
DKIM check:         pass
Sender-ID check:    pass
SpamAssassin check: ham

============================================================
Details:
============================================================

HELO hostname:  caserv.caha.es
Source IP:      150.214.222.10
mail-from:      guindos@caha.es

------------------------------------------------------------
```

```
SPF check details:
----------------------------------------------------------
Result:         pass
ID(s) verified: smtp.mail=guindos@caha.es DNS record(s):
    caha.es. 86400 IN TXT "v=spf1 mx a:caserv.caha.es -all"
    caha.es. 86400 IN MX 15 caserv.caha.es.
    caha.es. 86400 IN MX 10 wdxnp.caha.es.
    caserv.caha.es. 86400 IN A 150.214.222.10

----------------------------------------------------------
DomainKeys check details:
----------------------------------------------------------
Result:         neutral (message not signed)
ID(s) verified: header.From=guindos@caha.es DNS record(s):

----------------------------------------------------------
DKIM check details:
----------------------------------------------------------
Result:         pass (matches From: guindos@caha.es)
ID(s) verified: header.d=caha.es
Canonicalized Headers:
    x-mailer:Microsoft'20'Office'20'Outlook'20'11'0D''0A'
    content-transfer-encoding:quoted-printable'0D''0A'
    content-type:text/plain;'20'charset="iso-8859-1"'0D''0A'
    mime-version:1.0'0D''0A'
    message-id:<58779E4CD127461DA86CFD4D523D347B@ike>'0D''0A'
    date:Mon,'20'25'20'Oct'20'2010'20'08:52:57'20'+0200'0D''0A'
    subject:'0D''0A'
    from:"Enrique'20'de'20'Guindos'20'Carretero"'20'<guindos@caha.es>'0D''0A'

received:from'20'ike'20'(ike.caha.es'20'[150.214.222.77])'20'by'20'caserv.caha
.es'20'(Postfix)'20'with'20'ESMTPSA'20'id'20'36744BD'20'for'20'<check-
auth2@verifier.port25.com>;'20'Mon,'20'25'20'Oct'20'2010'20'06:52:17'20'+0000'
20'(GMT)'0D''0A'

received:from'20'caserv.caha.es'20'([127.0.0.1])'20'by'20'localhost'20'(caserv
.caha.es'20'[127.0.0.1])'20'(amavisd-
new,'20'port'20'10026)'20'with'20'LMTP'20'id'20'wdRjEYEh2tEu'20'for'20'<check-
auth2@verifier.port25.com>;'20'Mon,'20'25'20'Oct'20'2010'20'06:52:17'20'+0000'
20'(GMT)'0D''0A'
    dkim-signature:v=1;'20'a=rsa-
sha256;'20'c=relaxed/simple;'20'd=caha.es;'20'h='20'x-mailer:content-transfer-
encoding:content-type:content-type'20':mime-version:message-
id:date:date:subject:subject:from:from'20':received:received;'20's=personal;'2
0't=1287989537;'20'bh=7ipdI8sx9mvdCRU'20'x2ul3EW1cbK2geh2aZAMkULu2KAw=;'20'b=

Canonicalized Body:
    '0D''0A'
    '0D''0A'
    ---'0D''0A'
    Enrique'20'de'20'Guindos'20'Carretero'0D''0A'
    Departamento'20'de'20'Inform=E1tica'0D''0A'
    Tel.'20'950632517'20'-'20'Fax'20'950632504'0D''0A'
    Centro'20'Astron=F3mico'20'Hispano-Alem=E1n'20'A.I.E.'0D''0A'
    Calar'20'Alto'20'-'20'Almer=EDa=20'0D''0A'
    =20'0D''0A'
    =20'0D''0A'


DNS record(s):
    personal._domainkey.caha.es. 3600 IN TXT "v=DKIM1;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC8RpVXB/0fSWYG0idkodr+sVsrcnfA3nTBFMqk
z/tyG+CHWnl1zN9TatvY/MAVbRlrEe+F4cS3YkyhGdPOoomx02jcg54/sy2TKOIzHcYRCz495dZqjj
0t2Opg93g6jaU0v2aMF9Zr5vKQ6bqBCYCCOC4PJ3cn5OATsHzQPeLRPwIDAQAB"

NOTE: DKIM checking has been performed based on the latest DKIM specs (RFC
4871 or draft-ietf-dkim-base-10) and verification may fail for older versions.
```

```
If you are using Port25's PowerMTA, you need to use version 3.2r11 or later to
get a compatible version of DKIM.

---------------------------------------------------------
Sender-ID check details:
---------------------------------------------------------
Result:         pass
ID(s) verified: header.From=guindos@caha.es DNS record(s):
    caha.es. 86400 IN TXT "v=spf1 mx a:caserv.caha.es -all"
    caha.es. 86400 IN MX 15 caserv.caha.es.
    caha.es. 86400 IN MX 10 wdxnp.caha.es.
    caserv.caha.es. 86400 IN A 150.214.222.10

---------------------------------------------------------
SpamAssassin check details:
---------------------------------------------------------
SpamAssassin v3.2.5 (2008-06-10)

Result:         ham  (-0.7 points, 5.0 required)

 pts rule name              description
---- ---------------------- --------------------------------------------------
-4.0 RCVD_IN_DNSWL_MED      RBL: Sender listed at http://www.dnswl.org/,
medium
                             trust
                            [150.214.222.10 listed in list.dnswl.org]
-0.0 SPF_PASS               SPF: sender matches SPF record
-2.6 BAYES_00               BODY: Bayesian spam probability is 0 to 1%
                            [score: 0.0000]
 1.8 MISSING_SUBJECT        Missing Subject: header
 4.2 AWL                    AWL: From: address is in the auto white-list

=========================================================
Explanation of the possible results (adapted from
draft-kucherawy-sender-auth-header-04.txt):
=========================================================
```

Antes de acabar, me gustaría también dejar aquí un log de un mensaje enviado por un usuario nuestro y otro recibido desde el exterior:

```
Nov 30 07:33:16 caserv amavis[32732]: (32732-17) Passed CLEAN,
FIRMA_DKIM/MYNETS LOCAL [192.168.222.73] [192.168.222.73] <guindos@caha.es> ->
<p_pino_perez@prevencionfremap.es>, Message-ID: <4CF4A9F1.2060706@caha.es>,
mail_id: jHNmVafbi79x, Hits: -1.02, size: 12719160, queued_as: 5FE4ABD, 6750
ms
Nov 30 07:33:16 caserv postfix/lmtp[10792]: 57ADEBE:
to=<p_pino_perez@prevencionfremap.es>, relay=127.0.0.1[127.0.0.1]:10026,
delay=7.5, delays=0.78/0/0/6.8, dsn=2.0.0, status=sent (250 2.0.0 Ok,
id=32732-17, from MTA([150.214.222.10]:10025): 250 2.0.0 Ok: queued as
5FE4ABD)
Nov 30 07:34:44 caserv amavis[32732]: (32732-18) Passed CLEAN, [213.0.54.193]
[213.0.54.193] <p_pino_perez@prevencionfremap.es> -> <guindos@caha.es>,
Message-ID:
<2C4493A0B4DB344B9E5D364775D6038401ED317D@EXCHANGESPF.SPFREMAP.INT>, mail_id:
ckHIJjHJhucl, Hits: -0.02, size: 6115, queued_as: 31C91C0, 4154 ms
```

En este ejemplo se puede ver que el mensaje enviado por guindos@caha.es ha pasado por el puerto 10026 al amavis, puerto asociado a la política FIRMA_DKIM y que lo que hace es, además de comprobar el correo, lo firma. En cambio, al correo recibido se le ha aplicado la política estándar que solo realiza el control del correo, sin firmarlo.