Instalación de servidor de correo con SPF, DKIM y DMARC

Volodimir Yarmash Yarmash

Introducción

El correo electrónico es uno de los servicios más utilizados en Internet y, a su vez, uno de los principales vectores de ataque (spoofing, phishing, spam). Para mitigar estos riesgos, existen mecanismos de autenticación y validación del correo como SPF, DKIM y DMARC, ampliamente utilizados en entornos profesionales.

En esta práctica, el alumnado configurará un servidor de correo seguro en Ubuntu, implementando dichos mecanismos y verificando su correcto funcionamiento.

Instalación del servidor de correo

Instalar y configurar un MTA (por ejemplo, Postfix).

Verificar el envío básico de correos desde el servidor.

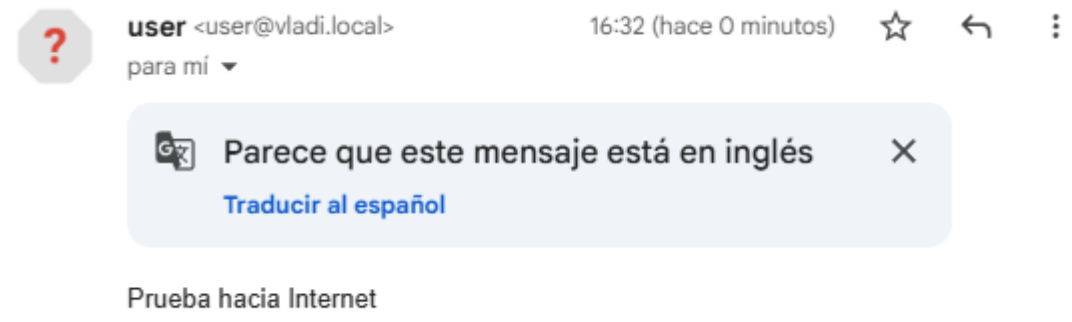
Después de haber realizado la instalación, nos dirigimos a main.cf, comprobamos que nuestro host está bien configurado.

```
  GNU nano 6.2                    /etc/postfix/main.cf
smtpd_tls_security_level=may

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache


smtpd_relay_restrictions = permit mynetworks permit sasl_authenticated defer_un>
myhostname = vladi.local
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, vladi.local, localhost, localhost.localdomain, loc>
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

Si está bien, realizamos una prueba enviando un mail.

```
user@user-Standard-PC-i440FX-PIIX-19echo "Primera prueba ." | mail -s "Prueba Local" u
ser@vladi.local.local
user@user-Standard-PC-i440FX-PIIX-1996:~$ cat /var/mail/$(whoami)
From user@vladi.local  Fri Jan 30 16:28:25 2026
Return-Path: <user@vladi.local>
X-Original-To: user@vladi.local
Delivered-To: user@vladi.local
Received: by vladi.local (Postfix, from userid 1000)
        id A5B82420362; Fri, 30 Jan 2026 16:28:25 +0100 (CET)
Subject: Prueba Local
To: <user@vladi.local>
User-Agent: mail (GNU Mailutils 3.14)
Date: Fri, 30 Jan 2026 16:28:25 +0100
Message-Id: <20260130152825.A5B82420362@vladi.local>
From: user <user@vladi.local>

Primera prueba .
```

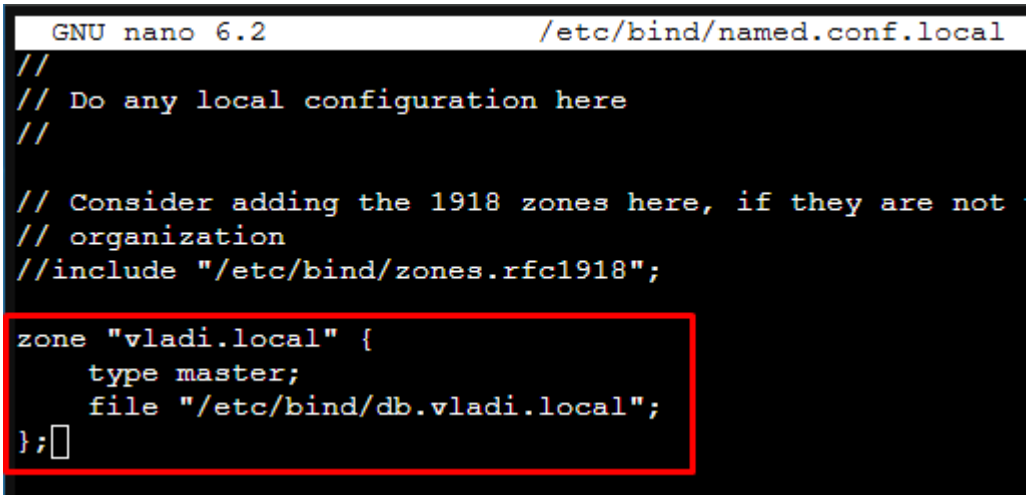
Ahora hacia un correo real.



Configuración de DNS básica

Configurar los registros necesarios para el funcionamiento del correo (MX, A, PTR si aplica).

Para hacer este paso, necesitamos instalar Bind9 (servidor dns) para realizar la práctica.
Creamos una zona local, para ello:
Nos dirigimos a /etc/bind/named.conf.local
Agregamos esta configuración



Implementación de SPF

Crear y configurar el registro SPF en DNS.

Verificar que los correos enviados cumplen la política definida.

Entramos a /etc/bind/db.vladi.local(si no existe, lo copiamos del local)
Aplicamos esta configuración:

```
  GNU nano 6.2                  /etc/bind/db.vladi.local *
;
$TTL    604800
@       IN      SOA     vladi.local. root.vladi.local. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; Registros NS (Nameserver)
@       IN      NS      ns.vladi.local.
; Registros A (Direcciones IP)
@       IN      A       10.4.0.40
ns      IN      A       10.4.0.40
mail    IN      A       10.4.0.40
; Registro MX (Mail Exchange)
@       IN      MX      10 mail.vladi.local.
; --- IMPLEMENTACIÓN SPF ---
@       IN      TXT     "v=spf1 mx -all"
```

Implementación de DKIM

Generar las claves DKIM.

Configurar el servidor de correo para firmar los mensajes.

Publicar la clave pública DKIM en DNS.

Verificar la firma DKIM en los correos enviados.

Creamos la ruta /etc/opendkim/keys y le damos permiso 750
Generamos la llave con el comando sudo opendkim-genkey -D /etc/opendkim/keys/ -s mail -d vladi.local

```
user@user-Standard-PC-i440FX-PIIX-1996:~$ sudo ls -l /etc/opendkim/keys/
sudo: unable to resolve host vladi.local: Name or service not known
total 8
-rw------- 1 opendkim opendkim 1704 ene 30 17:14 mail.private
-rw------- 1 root     root      501 ene 30 17:14 mail.txt
```

En /etc/opendkim/TrustedHosts agregamos los hosts de confianza para que OpenDKIM sepa qué dominios firmar.

Ahora en /etc/opendkim/KeyTable indicamos la localizacion de la llave. Agregamos este texto: mail._domainkey.vladi.local vladi.local:mail:/etc/opendkim/keys/mail.private

Vamos a indicar que todos los usuarios de @vladi.local, puedan usar la llave Abrimos /etc/opendkim/SigningTable y agregamos *@vladi.local mail._domainkey.vladi.local

Para configurar el servidor, entramos en /etc/opendkim.conf y nos aseguramos de que tenemos habilitados y bien configurados estos parámetros:

```
Syslog                  yes
UMask                   002
Mode                    sv
KeyTable                refile:/etc/opendkim/KeyTable
SigningTable            refile:/etc/opendkim/SigningTable
ExternalIgnoreList      refile:/etc/opendkim/TrustedHosts
InternalHosts           refile:/etc/opendkim/TrustedHosts
Socket                  inet:8891@localhost
```

Forzamos el puerto en /etc/default/opendkim agregando SOCKET="inet:8891@localhost"

Para poder usar la firma, entramos en config de postfix /etc/postfix/main.cf y agregamos la config DKIM:

```
milter_protocol = 2
milter_default_action = accept
smtpd_milters = inet:localhost:8891
non_smtpd_milters = inet:localhost:8891
```

cojemos la clave pública de cat /etc/opendkim/keys/mail.txt y la agregamos a nuestro DNS en /etc/bind/db.vladi.local, agregamos este texto:
mail._domainkey IN TXT ("lo que nos de")

Reiniciamos todos los servicios y hacemos un envío de prueba con el comando mail -s "Correo Firmado DKIM" user@vladi.local

Entramos y podemos ver que lo hemos hecho bien:

```
  GNU nano 6.2                          /var/mail/user
Delivered-To: user@vladi.local
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=vladi.local; s=mail;
        t=1769793950; bh=uLKcC1rU5ulMGfA5xupjpzd42FK2iyuM9WHqp7DONws=;
        h=Subject:To:Date:From:From;
        b=i0HzddkWmkmEC1OtVyOyH5BbR/T52E4D5DFJIYaNgFZFuT4udAvXSZTj0FQLCk6S5
         tJqQAEnZQUwXhWwVcm7h83PaCc4bTnvV0GKrduebfcveoXHQsfvbvaDwR3tuoUGAHu
         cJnQ6tX12zrLPNHMdqpgBcf7xHik3dN7ZwHECuy446QQI9C37ZrtLhS2KTTUmUJBuB
         p4xalujSHRCvoxdaosBIvBeJkZEMDRPdbBmL5z/n8VcDcP9boBYRCAeazpVydphAzG
         KbWdi2VWhdrEfTIJ6+FLtjMzIqGPNPRjmvhhGcrnyOBpnom2RF7OatcrC7oP1kimwn
         G7kVRCjr/JNtg==
Received: by vladi.local (Postfix, from userid 1000)
        id F30F0420DD7; Fri, 30 Jan 2026 18:25:49 +0100 (CET)
Subject: Correo Firmado DKIM
To: <user@vladi.local>
User-Agent: mail (GNU Mailutils 3.14)
Date: Fri, 30 Jan 2026 18:25:49 +0100
Message-Id: <20260130172549.F30F0420DD7@vladi.local>
From: user <user@vladi.local>

Pruebadkim
```

Implementación de DMARC

Crear una política DMARC adecuada.

Configurar el registro DMARC en DNS.

Analizar los resultados de autenticación SPF y DKIM mediante DMARC.

Vamos a configurar el registro DMARC en DNS, entramos en /etc/bind/db.vladi.local y tenemos que cambiar Serial a 4 y agregar al final esta línea de código:
_dmarc IN TXT "v=DMARC1; p=quarantine; rua=mailto:user@vladi.local"

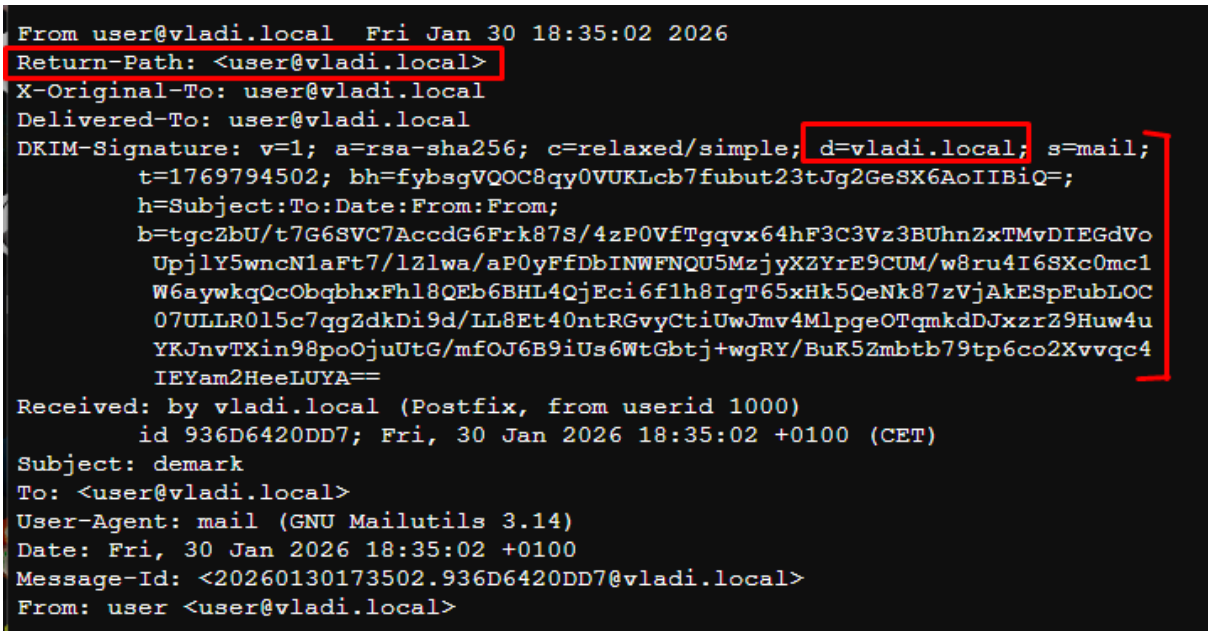
Pruebas y validación

Enviar correos a servicios externos o herramientas de verificación.

Analizar cabeceras de correo para comprobar SPF, DKIM y DMARC.

Documentar los resultados obtenidos.

Enviamos un correo electrónico y comprobamos el mailbox
mail -s "demark" user@vladi.local



SPF pasa ya que Return-Path: <user@vladi.local>
DKIM pasa ya que vemos el dominio d=vladi.local y la clave cifrada
DMARK como coinciden el From y el dominio de la firma, pasa tambien.