Autenticació de correu electrònic

Índex

Cicle formatiu: CFGM Sistemes Microinformàtics i Xarxes (SMX) / CFGS Administració de sistemes informàtics en xarxa (ASIX)

Mòdul: 0226 --- Seguretat Informàtica / 0227 - Serveis de xarxa / 0375 - Serveis de xarxa i Internet

Sistema operatiu: Ubuntu Server 26.04 LTS

El protocol SMTP, dissenyat als anys 80, no verifica qui envia realment un correu: qualsevol servidor pot afirmar que envia en nom de qualsevol domini, sense cap comprovació. Això fa possible el *spoofing* (suplantar un remitent) i el *phishing*: un atacant pot enviar-te un correu que sembli venir del teu banc o de la teva empresa sense tenir-hi cap relació real --- per exemple, un missatge que sembli venir de `banc@elteubanc.com` sense que el banc hi tingui res a veure.

L'autenticació de correu (SPF, DKIM i DMARC) tanca aquest forat: verifica que el servidor emissor estigui autoritzat, que el missatge no s'hagi alterat pel camí, i defineix què fer si alguna comprovació falla. Sense això, els destinataris no tenen cap manera fiable de distingir correu legítim de suplantat.

1. SPF

SPF (*Sender Policy Framework*) és un mecanisme que permet a un domini de correu dir, públicament i verificable, **“aquestes són les úniques màquines autoritzades a enviar correu en el meu nom”**. Ho fa publicant un registre TXT al DNS del domini amb la llista d'IPs/servidors autoritzats; el destinatari, en rebre un correu, consulta aquest registre i comprova si la IP d'origen hi coincideix.

1.1. Una limitació important

SPF **només verifica la IP del servidor emissor**, no el contingut ni la identitat real de qui ha escrit el correu --- per això sol combinar-se amb **DKIM** (signatura criptogràfica del missatge) i **DMARC** (la política que uneix tots dos i li diu al destinatari què fer si algun falla). Els tres junts formen l'estàndard actual d'autenticació de correu.

2. DKIM

DKIM (*DomainKeys Identified Mail*) permet a un domini remitent **signar criptogràficament** els seus missatges de sortida, perquè el destinatari pugui verificar dues coses:

1. Que el correu **realment prové** del domini que diu (autenticitat)
2. Que el contingut **no s'ha modificat** pel camí (integritat)

A diferència de l'SPF (que només comprova la IP del servidor), DKIM verifica el **missatge en si mateix**, cosa que el fa més robust.

2.1. Com funciona?

Es basa en **criptografia de clau pública/privada**, igual que un certificat TLS:

1. El servidor de correu genera un parell de claus: una **privada** (secreta, es queda al servidor) i una **pública** (es publica al DNS).
2. Quan surt un correu, el servidor **el signa** amb la clau privada, generant un hash xifrat d'algunes capçaleres i del cos del missatge. Aquesta signatura s'afegeix com una capçalera nova al correu:

   ```
   DKIM-Signature: v=1; a=rsa-sha256; d=thos.local;
    ↪  s=mail; ...; b=Kj3f9s...
   ```

3. El destinatari (Gmail, per exemple) rep el correu, **consulta al DNS** la clau pública del domini remitent (a `mail._domainkey.thos.local`), i amb ella **desxifra i comprova** que la signatura coincideix amb el contingut real del missatge.
4. Si coincideix → el correu no s'ha tocat pel camí i realment prové d'algú amb accés a la clau privada d'aquest domini (és a dir, el servidor legítim).

2.2. Elements clau

Terme	Significat
Selector (`s=mail`)	Un nom arbitrari que identifica quina clau concreta s'ha fet servir (útil si tens diverses claus o vols rotar-les sense trencar res)
Domini (`d=thos.local`)	El domini que reivindica haver enviat el correu
Clau privada	Es queda al servidor (per exemple, `mail.private`), mai es comparteix
Clau pública	Es publica com a registre TXT al DNS: `mail._domainkey.thos.local`

2.3. Diferència pràctica amb SPF

	SPF	DKIM
Què verifica	La IP del servidor emissor	El contingut i l'autenticitat del missatge
Es trenca si…	El correu es reenvia des d'un altre servidor	El contingut es modifica pel camí (per exemple, un servidor intermedi hi afegeix una capçalera)
Mecanisme	Consulta DNS simple	Criptografia de clau pública

Fer servir **tots dos junts** (i sovint DMARC per sobre) és el que dona una protecció real contra la suplantació --- cadascun cobreix un vector d'atac diferent que l'altre no cobreix.

3. DMARC

DMARC (*Domain-based Message Authentication, Reporting & Conformance*) és el mecanisme que **uneix SPF i DKIM sota una política única**, i li diu al destinatari (Gmail, Outlook, etc.) **què ha de fer exactament** si un correu falla aquestes comprovacions. SPF i DKIM, per si sols, **verifiquen** coses, però no diuen què fer amb el resultat --- cada proveïdor decideix pel seu compte. DMARC treu aquesta ambigüitat: el propietari del domini **especifica explícitament la política** a seguir.

3.1. Com funciona?

1. Es publica un registre TXT addicional al DNS:

```
_dmarc.thos.local. IN TXT "v=DMARC1; p=reject;
 ↪  rua=mailto:dmarc-reports@thos.local"
```

2. Quan arriba un correu, el destinatari comprova **l'alineació** (*alignment*): que el domini del `From:` visible coincideixi amb el domini que ha passat SPF o DKIM (no cal que passin els dos, però almenys un ha d'estar "alineat" amb el domini visible).
3. Segons el resultat i la política publicada (p=), el destinatari decideix:

Política	Significat
`p=none`	Només informa (mode d'observació, no bloqueja res)
`p=quarantine`	Marca'l com a sospitós (típicament, carpeta spam)
`p=reject`	Rebutja'l directament, no l'entreguis

3.2. L'element diferencial: informes (*reports*)

A diferència de SPF/DKIM, DMARC inclou un mecanisme de **retroalimentació**: amb `rua=mailto:...`, els grans proveïdors (Gmail, Yahoo, Microsoft) t'envien periòdicament informes agregats (XML) dient-te *qui* està enviant correu en nom del teu domini i si passa o falla les comprovacions --- útil per detectar suplantacions o servidors legítims que t'has deixat sense autoritzar a l'SPF.

4. Relació entre els tres mecanismes

Mecanisme	Què verifica	Qui decideix què fer
SPF	La IP del servidor emissor	Cada destinatari, pel seu compte
DKIM	La signatura/integritat del missatge	Cada destinatari, pel seu compte
DMARC	Que SPF o DKIM estiguin alineats amb el domini visible	**El domini remitent**, explícitament

En resum: **SPF i DKIM són els mecanismes de verificació; DMARC és la política que decideix les conseqüències**. Els proveïdors grans com Gmail donen molt més pes a un domini que té DMARC configurat (especialment amb `p=reject`) que a un que només té SPF/DKIM solts sense cap política que els lligui.

5. Configuració d'un stack d'autenticació de domini de correu

Abans de començar, un aclariment important: **SPF i DKIM funcionen publicant registres al DNS *públic* del domini remitent**, perquè els servidors externs (Gmail, etc.) puguin consultar-los. Com que `thos.local` és un domini de laboratori, no resoluble des d'Internet, cap servidor extern real podrà mai verificar aquests registres per a `thos.local` --- per tant, **Gmail o qualsevol altre destinatari extern seguirà sense poder validar SPF/DKIM dels teus correus**, independentment del que configuris.

Dit això, val molt la pena fer-ho igualment amb finalitat **didàctica**: podràs veure el mecanisme complet funcionant dins del laboratori (publicant els registres a la teva pròpia zona BIND de `thos.local`, i verificant-los amb `dig`/`opendkim-testkey` des d'una altra màquina interna). Si en el futur vols que sigui efectiu de debò cap a l'exterior, caldria fer-ho sobre un domini públic real.

5.1. SPF (senzill, només DNS)

SPF només necessita un registre **TXT** indicant quines IP poden enviar correu en nom del domini. Si ja tens BIND9 gestionant la zona `thos.local`, afegeix-hi:

```
thos.local.   IN   TXT   "v=spf1 mx a:correu.thos.local ip4:10.0.2.10
 ↪  -all"
```

- `mx` i `a:correu.thos.local` → autoritza el mateix servidor de correu
- `ip4:10.0.2.10` → autoritza explícitament la IP
- `-all` → rebutja qualsevol altra font (política estricta, correcta per a un domini que només té un servidor de sortida)

Recarrega la zona:

```
sudo rndc reload thos.local
dig TXT thos.local +short
```

5.2. DKIM (signatura criptogràfica, cal OpenDKIM)

5.2.1. Instal·la OpenDKIM

```
sudo apt install opendkim opendkim-tools
```

5.2.2. Genera el parell de claus

```
sudo mkdir -p /etc/opendkim/keys/thos.local
sudo opendkim-genkey -b 2048 -d thos.local -D
 ↪  /etc/opendkim/keys/thos.local -s mail -v
sudo chown -R opendkim:opendkim /etc/opendkim/keys
sudo chmod 700 /etc/opendkim/keys/thos.local
sudo chmod 600 /etc/opendkim/keys/thos.local/mail.private
```

Això genera `mail.private` (clau privada, per signar) i `mail.txt` (el registre DNS que has de publicar).

5.2.3. Configura `/etc/opendkim.conf`

```
sudo nano /etc/opendkim.conf
```

```
Syslog yes
UMask 002
Canonicalization relaxed/simple
Mode sv
Domain thos.local
KeyFile /etc/opendkim/keys/thos.local/mail.private
Selector mail
Socket local:/run/opendkim/opendkim.sock
UserID opendkim:opendkim
```

5.2.4. Assegura el directori del socket (i que sobrevisqui a un reinici)

```
sudo mkdir -p /run/opendkim
sudo chown opendkim:postfix /run/opendkim
```

IMPORTANT

`/run/` és un sistema de fitxers en memòria (`tmpfs`) que es buida completament en cada arrencada del servidor. Si només fas el `mkdir` manual, el directori **desapareixerà després del pròxim reinici** i OpenDKIM fallarà en arrencar. Per evitar-ho, defineix-lo perquè es recreï automàticament a cada arrencada:

```
sudo nano /etc/tmpfiles.d/opendkim.conf
```

```
d /run/opendkim 0750 opendkim postfix -
```

Això li diu a `systemd-tmpfiles` que recreï el directori amb els permisos correctes cada cop que arrenqui la màquina, sense dependre que ho facis manualment.

5.2.5. Afegeix `postfix` al grup `opendkim` (perquè pugui llegir el socket)

```
sudo usermod -aG opendkim postfix
```

5.2.6. Connecta Postfix a OpenDKIM com a milter

```
sudo postconf -e 'milter_default_action = accept'
sudo postconf -e 'milter_protocol = 6'
sudo postconf -e 'smtpd_milters = local:/run/opendkim/opendkim.sock'
sudo postconf -e 'non_smtpd_milters =
 ↪  local:/run/opendkim/opendkim.sock'
```

5.2.7. Reinicia tots dos serveis

```
sudo systemctl restart opendkim
sudo systemctl restart postfix
```

5.2.8. Publica el registre a la teva zona BIND de `thos.local`

```
cat /etc/opendkim/keys/thos.local/mail.txt
```

Copia el contingut (format `mail._domainkey IN TXT "v=DKIM1; h=sha256; k=rsa; p=..."`) a la teva zona:

```
mail._domainkey.thos.local.   IN   TXT   "v=DKIM1; h=sha256; k=rsa;
 ↪  p=MIIBIjANBg..."
```

```
sudo rndc reload thos.local
```

5.2.9. Verificació

```
sudo opendkim-testkey -d thos.local -s mail -vvv
```

Hauria de respondre `key OK`.

Envia un missatge de prova (per exemple, `joan → maria`) i revisa les capçaleres del missatge rebut (a SOGo: obre el missatge → menú de tres punts → "Mostra el codi font" o similar) --- hauries de veure una capçalera `DKIM-Signature:` afegida automàticament.

5.3. Publica el registre DMARC a la teva zona BIND

Afegeix a la zona `thos.local` (el mateix fitxer on ja tens el registre SPF i el `mail._domainkey`):

```
_dmarc.thos.local.   IN   TXT   "v=DMARC1; p=quarantine;
 ↪  rua=mailto:dmarc-reports@thos.local; pct=100"
```

Explicació de cada paràmetre:

Paràmetre	Significat
`v=DMARC1`	Versió del protocol (sempre aquest valor)
`p=quarantine`	Política: si falla l'alineació, marca'l com a sospitós (es recomana començar aquí, no directament amb `reject`)
`rua=mailto:...`	On enviar els informes agregats (XML) --- cal que aquesta bústia existeixi realment al teu servidor
`pct=100`	Aplica la política al 100% dels correus (útil posar-ho més baix, per exemple `pct=10`, durant les primeres proves)

Recarrega la zona:

```
sudo rndc reload thos.local
dig TXT _dmarc.thos.local +short
```

5.4. Crea la bústia per rebre els informes (opcional però recomanat)

Si vols que `rua=mailto:dmarc-reports@thos.local` funcioni de debò dins del laboratori, cal que aquest usuari existeixi al teu LDAP, igual que `joan`/`maria`:

```
ldapadd -x -H ldap://ldapserver.thos.local -D
 ↪  "cn=admin,dc=thos,dc=local" -w CONTRASENYA <<'EOF'
dn: uid=dmarc-reports,ou=Usuaris,dc=thos,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: dmarc-reports
cn: DMARC Reports
sn: Reports
```

```
mail: dmarc-reports@thos.local
uidNumber: 10003
gidNumber: 10001
homeDirectory: /home/dmarc-reports
loginShell: /usr/sbin/nologin
userPassword: {SSHA}...
EOF
```

Com que ja tens Postfix configurat amb `virtual_mailbox_domains`/LMTP, aquesta bústia es crearà automàticament a `/var/mail/vhosts/dmarc-reports/` en rebre el primer correu, sense cap pas addicional.

5.5. Verificació completa dels tres registres

```
dig TXT thos.local +short
dig TXT mail._domainkey.thos.local +short
dig TXT _dmarc.thos.local +short
```

Tots tres haurien de respondre amb el contingut esperat.

5.6. Progressió recomanada de política (quan sigui un domini real)

En un domini real, la pràctica habitual és començar amb `p=none` (només observar via informes), confirmar durant unes setmanes que tot el correu legítim passa SPF/DKIM correctament, i només llavors pujar a `p=quarantine` i finalment `p=reject`. Per al teu entorn de laboratori, pots saltar-te directament a `p=quarantine` o `p=reject` per veure l'efecte immediatament, ja que no hi ha trànsit de producció real que es pugui veure afectat.

Versions d'aquest document

- HTML
- PDF
- ODT
- MD