DKIM Operation

DKIM permits a signing domain to assert responsibility for a message. It does this through a digital signature protocol where the public key is distributed by DNS.

DKIM signatures

DKIM-Signature is the added header field. Its domain, d=, and selector, s=, defined the DNS entry the recipient verifier will retrieve.

Field	DKIM-Signature header field tags
v	version - always 1
a	algorithm - rsa-sha256 is recommended. rsa-sha1 option if CPU is limited
c	canonicalization - relaxed/relaxed tolerate whitespace manipulation on header and

	body
t	timestamp - time message was signed
bh	hash of body content
h	headers which the dkim signature covers
b	the signature of the message headers including DKIM-Signature.

ADSP Overview

ADSP allows author domains, those in the From: address, to defined a policy for the recipient in the absence of a valid DKIM signature. The ADSP policy is available in DNS as a TXT record _adsp._domainkey.example.org.

DKIM Signing:

Key aspects:

- Sign at outermost mail boundary after all possible internal modification.
- Sign different message streams, user/list/transactional with domains i.e. subdomains of the signing domain (see DKIM and Mailing Lists draft RFC and DKIM Deployment RFC).
- Sign using c=relaxed/relaxed header/body canonicalization (whitespace tolerance)
- Set reporting address ("r={dkim-fail-address}") in DNS to be notified of DKIM verification failures
- Initially deploy in test mode (t=y)

Deployment:

1. Product Selection - OpenDKIM, Amavis-new, DKIMproxy(?) - others(?)
2. Create DKIM public/private keys - there will be a tool with the product. Selector names are at your convenience. Multiple selectors for different servers can be generated.
3. Deploy to DNS with low TTL - {selector}._domainkey.example.org IN TXT "v=DKIM1; g=*; t=y; k=rsa; r=dkim-fail; p=......", increment SOA and sync authoritative servers.
4. Verify key is retrieval from all authoritative servers. dig -t txt {selector}._domainkey.example.org @ns{1,2,3,4}.example.org
5. Install product selection with configuration recommendations above. Configure product to sign the right domains and from authenticated sources (IP addresses and via SMTP AUTH). Configure to not verify secondary mail servers.
6. Set MTA configuration to use product as per its install instructions.
7. Send an email to a gmail account and look at the Authenticated-Results header fields (or alternately find an autoresponder (rare)). Once all is working remove the testing flag and increase TTL.

Key Compromise/Roll over:

1. When the confidentiality of a key is compromised a new key will need to be generated.
2. Generate a new key on a different selector name.
3. Deploy and test new key. TTL values will determine vulnerability exposure.
4. Remove old key after a time period from DNS considering email still in transit.

DKIM Verification:

Key Points and Deployment:

- Verify as early as possible;
- Avoid verifying twice - ignore secondary mail servers that may have broken the signature after verification;
- SMTP/LMTP based DKIM verification product? - see Postfix workarounds to avoid Postfix breaking a DKIM signature before verification
- Milter based DKIM verification product? - use "-o receive_override_options=no_milters" on secondary smtpd processes in master.cf to prevent dual verification (or signing). Examine milter_*_timeout options to ensure that a broken milter does not overload your server.

- Set to send DKIM and ADSP reporting failures to originating domains;
- Set DKIM verification to add Authenticated-Results header field to incoming messages to allow end users to apply MUA filtering rules.

ADSP deployment:

ADSP informs recipients that a sender signs all emails. Emails domains that have users that send through email lists are not suitable for ADSP deployment. Transactional emails or email domains associated only with email lists are good candidates for ADSP as their signatures are unlikely to be broken. Where email forgery prevention is more important to the sender than the risk of legitimate email being dropped an ADSP policy of “adsp=discardable” may be appropriate.

Deploying a ADSP policy of “adsp=unknown” may have some DNS caching benefits to a recipient.

An ADSP policy of “adsp=all” was initially treated by some receivers as reject any broken/missing DKIM signature when all it really said was it was sent signed and mailing lists often broke these.

When deploying ADSP add a “r={adsp-fail-address}” to be notified of ADSP failures.

DKIM/ADSP Filtering

- DKIM is about domain validity - not anti-spam - don’t use DKIM results as a global spam score reduction.
- Use DKIM as a differentiator when statistics show it to be worthwhile.
- Statistics on domains will show what regularly succeeds and what fails
- Honour ADSP with care - recommending gathering statistics of potential rejections first.

References:

RFC 5585 DomainKeys Identified Mail (DKIM) Service Overview
RFC 5863 DomainKeys Identified Mail (DKIM) Development, Deployment, and Operations
RFC 4871 DomainKeys Identified Mail (DKIM) Signatures
RFC 5617 DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP)
draft-ietf-dkim-mailinglists-02 DKIM and Mailing Lists
RFC 5451 Message Header Field for Indicating Message Authentication Status
draft-ietf-marf-dkim-reporting-00 Reporting of DKIM Verification Failures
Australian Government IT Security Manual (ISM) September 2009 - ref: 6.6.178, 6.6.183-185
MAAWG Training Videos

By: Daniel Black <daniel.black@ovee.com.au>