

DNS security records cheat sheet

The DNS records that protect a domain: what each one does, what can go wrong, and how to check it is working.

MAIL AUTHENTICATION · TRANSPORT ENCRYPTION · CERTIFICATE CONTROL · DNS INTEGRITY

General educational guidance, not affiliated with or endorsed by any vendor named. References follow the RFCs named inline, NIST SP 800-81r3 for DNSSEC, and each vendor's own documentation. Records shown use example.com; substitute your domain. Published 2026-09-09.

STEP	RECORD	LIVES AT	PROTECTS AGAINST	QUICK CHECK	RFC
1	SPF	apex (@)	mail sent from unauthorized servers	dig +short example.com TXT grep spf1	7208
2	DKIM	<selector>._domainkey	tampered or unsigned mail	dig +short <selector>._domainkey.example.com TXT	6376
3	DMARC	_dmarc	mail forging your From address	dig +short _dmarc.example.com TXT	9989
4	TLS-RPT	_smtp._tls	silent TLS failures	dig +short _smtp._tls.example.com TXT	8460
5	MTA-STS	_mta-sts + file	inbound plaintext fallback	dig +short _mta-sts.example.com TXT	8461
6	DNSSEC	zone + parent	forged DNS answers	dig @1.1.1.1 example.com A +dnssec grep flags	4033-4035
7	CAA	apex (@)	certificates from unlisted CAs	dig +short example.com CAA	8659
8	parked domains	each parked domain	spoofing of domains that send no mail	dig +short _dmarc.<parked-domain> TXT	9989 · 7505

By default, anyone can send mail as your domain (spoofing), any certificate authority that is tricked into validating your domain can issue a certificate for it, and its DNS answers can be forged. The records above close those gaps.

Sections 1-8 follow the rollout order: steps 1-3 (email authentication) build on each other, 4 comes before 5, and the rest can be added any time. Sections 9 and 10 hold no records; they cover the controls around DNS and how to test from outside, and a box after section 5 covers DANE and BIML.

1 · SPF: WHO MAY SEND AS THE DOMAIN

A TXT record at the apex (the bare domain, example.com with nothing in front) listing the servers allowed to send as the domain; receivers check the connecting IP against it.

```
"v=spf1 include:spf.provider.example ~all"
# include: your mail provider's server list; ~all: soft-fail the rest
```

Checking an SPF record may use at most 10 DNS lookups (each `include`, `a`, `mx`, `ptr`, `exists`, and `redirect` counts); the 11th returns a permanent error (permerror) and mail starts failing. Lookups that come back empty (void lookups) have

their own limit of two, so dead `include:` entries fail a record well inside the count of 10. Count against the lookup limit before adding a mail service, and keep exactly one `v=spf1` record: a second one fails every check the same way.

With DMARC in place, `~all` is the usual choice over the hard-fail `-all`, which can drop forwarded mail before DMARC evaluates it and hide those senders from your reports (RFC 9989 section 7.1). Microsoft recommends `-all` for Microsoft 365 domains, so DMARC still acts on unsigned mail that fails SPF.

```
dig +short example.com TXT | grep spf1
```

2 • DKIM: A SIGNATURE ON EVERY MESSAGE

Your provider signs each outgoing message with a private key; the public key in DNS lets any receiver verify the message arrived unmodified. The selector in the record name comes from your provider and lets it move to a new key without breaking mail.

Google Workspace uses the selector `google`, has you publish the key in a TXT record, and offers 2048 bits when you generate it. Microsoft 365 uses `selector1` and `selector2`, has you publish two CNAMEs and keeps the key in its own zone, and issues 1024-bit keys until you run `Rotate-DkimSigningConfig -KeySize 2048`. Use 2048-bit RSA either way (RFC 8301).

```
dig +short <selector>._domainkey.example.com TXT
# expect: v=DKIM1; k=rsa; p=<long base64 key>, split across two
# strings (DNS caps each at 255); on Microsoft 365, a CNAME first
```

Some dashboards split the key for you and some reject a value that long. Your provider's page gives the exact records; re-run the check after any DNS change.

3 • DMARC: THE ENFORCEMENT POLICY

SPF and DKIM each pass or fail on their own; DMARC ties them to the From address the reader sees and tells receivers how you want failures handled: `p=none` asks for no action, `p=quarantine` for spam, `p=reject` for refusal. Receivers weigh that with their own analysis, so some spoofed mail lands even at reject (RFC 9989 section 7.4). They also mail you aggregate reports of who is sending as your domain; they arrive as gzipped XML meant for a parsing service, so plan to feed them to one rather than read them by hand.

```
"v=DMARC1; p=reject; rua=mailto:dmarc@example.com"
```

On a domain that already sends mail, roll out in stages: start at `p=none` and read the reports until every legitimate sender shows up as passing, then move to quarantine, then to reject. `t=y` asks receivers to apply one level below what you publish, so `p=reject; t=y` is treated as quarantine while reports keep arriving.

RFC 9989 retired `pct`: a record still carrying it below 100 now gets its full policy on every failing message.

Forwarding breaks SPF, so a domain at reject has to be DKIM signing everything, and RFC 9989 section 7.4 advises staying at quarantine where staff might post to mailing lists.

Leave the alignment tags at their defaults, which suit most domains: alignment checks that the domain approved by SPF or DKIM matches the From domain, and the default counts a subdomain as a match, while the strict setting (`adkim=s`) requires an exact match and breaks mail signed from a subdomain. Subdomains follow the main policy unless `sp=` sets a different one, and `np=reject` covers subdomains that do not exist, the names a spoofer invents (RFC 9989).

Reports can go to an address outside your domain only when that outside domain publishes an authorization record (RFC 9990 section 4); without it, the reports silently never arrive:

```
dig +short _dmarc.example.com TXT
# reports going to a vendor? check their authorization:
dig +short example.com._report._dmarc.<vendor> TXT
```

4 • TLS-RPT: THE FAILURE REPORTS

One TXT record asking senders for a daily JSON summary of their TLS sessions with your mail server, successes included. An MTA-STS policy in enforce mode refuses bad connections silently, so these reports are how an attack or a

misconfiguration becomes visible. Publish it before the MTA-STS policy goes live.

```
_smtp._tls TXT "v=TLSRPTv1; rua=mailto:tlsrpt@example.com"
dig +short _smtp._tls.example.com TXT
```

Reports arrive roughly once a day as JSON attachments, usually compressed (RFC 8460); a healthy report shows zero failures.

5 • MTA-STS: TLS REQUIRED INBOUND

SMTP encryption is opportunistic: an attacker on the network path can strip the STARTTLS offer, and a sender that sees no offer delivers in plaintext. MTA-STS makes TLS mandatory through a policy you publish: senders that honor it refuse the plaintext fallback, require a valid certificate, and deliver only to the MX hosts the policy names. Three pieces have to exist and stay consistent:

_mta-sts TXT the record that tells senders a policy exists. Its **id** value gives the policy version; a changed id triggers a re-fetch.

the policy file the rules, served over HTTPS at `https://mta-sts.<domain>/.well-known/mta-sts.txt`

mta-sts subdomain the fixed hostname senders fetch from; it needs a valid certificate of its own.

```
version: STSv1
mode: enforce
mx: mail.example.com
mx: mail2.example.com
max_age: 1209600
# list every MX host your domain publishes.
# or one mx: *.example.com in place of both: it matches one label
```

Senders cache the policy for **max_age** seconds (RFC 8461 expects weeks or greater) and re-check the id before bouncing any message, so a corrected policy with a new id repairs a mistake. **mode: testing** reports what enforce would have blocked; a domain whose MX hosts belong to a major provider can usually go straight to enforce, because those hosts already serve certificates valid for the names they publish.

Every edit to the policy file needs a new id in the TXT record, or senders keep the cached copy and never see the change. Update the policy before changing MX hosts: senders holding the old policy refuse a new MX it does not name. To retire MTA-STS cleanly, publish mode: none with a small max_age, bump the id, wait out the old lifetime, then remove the records (RFC 8461 section 8.3).

```
_mta-sts TXT "v=STSv1; id=20260906120000"
dig +short _mta-sts.example.com TXT
curl -s https://mta-sts.example.com/.well-known/mta-sts.txt
```

DANE AND BIMI

DANE (RFC 7672) is the DNSSEC-based way to require TLS on inbound mail: TLSA records pin the receiving server's certificate, verified through the signed zone. It needs DNSSEC end to end and a mail host that publishes TLSA records (Microsoft 365 supports it inbound, switched on per domain rather than by default); MTA-STS covers setups without it.

BIMI (a TXT record at **default._bimi**) shows your logo beside authenticated mail in some clients. It requires a DMARC policy at quarantine or reject and a paid mark certificate renewed annually: a Verified Mark Certificate needs a registered trademark or a government mark and earns the checkmark in Gmail, while a Common Mark Certificate needs no trademark and earns no checkmark. Most small domains skip it.

6 • DNSSEC: SIGNED DNS ANSWERS

DNSSEC signs your DNS records so a resolver can verify every answer came from you and was not altered on the way (NIST SP 800-81r3). Without it, a forged answer can redirect mail or visitors, and every record above can itself be forged in transit. Enable signing at the DNS host, then publish the DS record it gives you at the registrar; that links your domain

into the global chain of trust. Pick ECDSA P-256 (algorithm 13) where offered; its signatures are smaller than RSA's (SP 800-81r3).

```
dig @1.1.1.1 example.com A +dnssec | grep flags
# two lines come back; "ad" on the first means a validating
# resolver checked the chain, so name one that does
dig +short example.com DS
# the DS at the parent completes the chain
```

Moving DNS hosts or registrars while the old DS record stands makes every validating resolver reject the whole domain: remove the DS first, wait out its TTL, then move.

Signatures carry an expiry date, so a signing process that stalls unnoticed has the same effect with no attacker involved: validating resolvers start rejecting the domain. Where the DNS host and the registrar both support CDS and CDNSKEY records, the DS updates itself and nobody carries it across by hand.

7 • CAA: WHO MAY ISSUE CERTIFICATES

CAA records name the certificate authorities allowed to issue certificates for the domain; any CA not on the list must refuse the request. An `issue` record covers every certificate for the domain, wildcards included. `issuewild` overrides `issue` for wildcard names only, so add it to allow a different CA there, or as `issuewild ";"` to block wildcards outright (RFC 8659 section 4.3). `iodef` gives CAs an address to report refused or suspicious requests.

```
dig +short example.com CAA
# 0 issue "letsencrypt.org"
# 0 issuewild ";" no wildcard certificate from anyone
# 0 iodef "mailto:security@example.com"
```

Before adding any service that will hold a certificate for your domain (a mail gateway, a second CDN, an app on a subdomain, including the `mta-sts` host in section 5), put that service's CA on this list. Otherwise the request fails at that service. An ACME client names the cause, because RFC 8555 defines a `caa` error type for it; a managed service may report only that the certificate failed.

Check the records with dig rather than your DNS dashboard: some DNS hosts add CAA entries for their own certificate partners without telling you, so the dashboard can show fewer records than the internet sees.

CAA blocks issuance from unlisted CAs; certificate transparency (CT) reports what was issued anyway. Browsers reject a publicly trusted certificate that is missing from the public logs, so a CT monitoring service emails you when one appears for your domain, and crt.sh shows the history on demand.

8 • PARKED DOMAINS NEED DMARC TOO

Each domain needs its own DMARC record; the policy on your main domain covers only that domain and its subdomains. A parked domain with no mailboxes still gets spoofed, and with no policy published you have given receivers nothing to act on. On every domain that sends no mail, publish the records below; the null MX line belongs only when the domain receives no mail either:

```
_dmarc TXT "v=DMARC1; p=reject;"
@      TXT "v=spf1 -all"
@      MX  0 .
# @ = the apex in most DNS dashboards; -all is right here:
# nothing legitimate is ever sent from a parked domain
# null MX (RFC 7505): accepts no mail. skip it where email
# forwarding exists; it cannot sit beside a real MX on one name
```

Publish no DKIM record here: with no public key in DNS, a forged signature cannot validate. M3AAWG's parked domains guidance recommended a wildcard revocation record (`*._domainkey` with an empty `p=`, RFC 6376 section 3.6.1) until June 2022 and now says to publish none.

registrar lock blocks domain transfers until you unlock; the domain's RDAP record (lookup.icann.org) should show `clientTransferProhibited`. Most country-code domains still answer `whois` instead; for those, check the registry's own lookup.

registry lock a paid, stronger lock applied at the registry itself, setting `serverTransferProhibited`, `serverUpdateProhibited`, and `serverDeleteProhibited` so nobody can move the domain, change its nameservers, or delete it. Even a hijacked registrar account cannot lift it, because removal goes through the registry operator by hand. Aimed at high-value domains.

expiry + auto-renew when a domain lapses, the site and the mail stop working.

MFA on registrar and DNS host either account can redirect the domain's mail and traffic, so use phishing-resistant MFA where they offer it.

recovery address off-domain a recovery email on the domain itself stops working exactly when the domain is lost.

no wildcard record a wildcard (`*.example.com`) makes every subdomain resolve, even ones you never created. If its target is ever a shared or abandoned service, whoever claims it gets a working site under your name, with a valid certificate, and can serve phishing pages from it (a subdomain takeover).

zone transfers refused `dig example.com AXFR @ns1` should fail; a successful transfer hands an attacker every name in your DNS.

10 • TEST IT FROM OUTSIDE

Send a message to a mailbox you control elsewhere, open its headers, and read the Authentication-Results line. A healthy result shows three passes naming your domain:

```
Authentication-Results: ...
    spf=pass ... dkim=pass header.d=example.com
    dmarc=pass header.from=example.com
```

internet.nl's email test (internet.nl/test-mail) covers SPF, DKIM, DMARC, STARTTLS, DANE, and DNSSEC in one pass, without an account, and gives you a link you can send to your mail host. Google Admin Toolbox Check MX (toolbox.googleapps.com/apps/checkmx) reads the mail records without an account as well.