**Conformance Statement**

THE MP STANDARD IS AN INDEPENDENTLY DEVELOPED EXTERNAL SECURITY BASELINE. IT IS NOT AN ACCREDITED CERTIFICATION, PENETRATION TEST OR COMPREHENSIVE SECURITY ASSESSMENT. CONFORMANCE CONFIRMS ONLY THAT THE PUBLICLY OBSERVABLE CONTROLS DEFINED IN THIS DOCUMENT MET THE STATED REQUIREMENTS AT THE TIME OF ASSESSMENT.

# THE MP STANDARD

A domain security baseline for mail, DNS and web, with scoring rationale

| Field | Value |
|---|---|
| Reference | MP-STD-001 |
| Version | 1.1 |
| Issued | 7 September 2026 |
| Author | Martin Paul (MP) |
| Status | Issued. Correct at the time of publication |
| Review | Twelve months from issue, or on any change to the grader's scoring, whichever comes first |
| Implemented by | SecurityNut IP, ip.securitynut.co.uk, scoring engine at v2.2.6 at time of issue |
| Related | MP-DER-001 Digital Estate Register; MP Web Estate Reference Record v1.0 |

**Summary.** The MP Standard is my definition of the minimum security posture a public internet domain should present, across three pillars: the mail it sends, the DNS that describes it and the website it serves. It is deliberately opinionated. Every control in it is one I already run across my own domains, so the standard asks nothing of anyone that I don't do myself. This document names each protocol and policy, explains how it works, states the threat it addresses, sets the exact requirement and describes how conformance is measured and scored. The live examples throughout are taken from securitynut.co.uk, the domain the grader itself lives on.

*"Security should never come at the cost of convenience"*

Martin Paul

# Contents

# 1. Purpose, scope and principles

The MP Standard exists to answer one question consistently: does this domain do the basic things that stop it being impersonated, hijacked or served insecurely? It is a baseline, not a full security programme. It says nothing about application logic, authentication, patching or the people who run the domain. What it covers, it covers precisely.

## 1.1 Scope

The standard applies to a registrable domain (the zone apex, for example securitynut.co.uk) and the website served at that name. Three pillars are assessed:

- **Mail.** Whether mail claiming to come from the domain can be authenticated and whether mail sent to it is protected in transit: SPF, DKIM, DMARC, MTA-STS and TLS-RPT.
- **DNS.** Whether the domain's records can be trusted and whether certificate issuance is constrained: DNSSEC, CAA and nameserver resilience.
- **Web.** Whether the site forces encrypted transport and sends the response headers that protect visitors: HTTPS redirection, HSTS and a defined header set.

## 1.2 Principles

- **Practised, not preached.** Every required control is in force on the author's own zones. The benchmark domain that defined the standard was brought to full conformance before the standard was published.
- **Enforcement over monitoring.** Where a protocol offers a monitoring mode and an enforcement mode (DMARC p=none versus p=reject, MTA-STS testing versus enforce, SPF ~all versus -all), the standard requires enforcement. Monitoring modes are a stage on the way, not a destination.
- **Measured, not declared.** Conformance is determined from public DNS and one live HTTP fetch. Nothing is taken on trust and nothing requires access to the domain owner's systems.
- **Provider neutral.** Requirements describe the published result, not how any particular DNS host, edge provider or mail platform achieves it.
- **Honest about limits.** Where a check cannot be measured reliably (a site that refuses automated fetches, a DKIM selector that cannot be guessed) the standard says so and does not penalise.

## 1.3 Currency and review

Security protocols and the controls in this standard are added or changed as new threats emerge and new technologies become available. This document was correct at the time of publication. The grader at ip.securitynut.co.uk is authoritative for the current version of the standard, and the document history in section 10 records every change to the controls or their weights. The standard is reviewed at least every twelve months, and sooner if a control is deprecated, a new protocol reaches wide receiver or browser support, or a scoring change is made to the grader.

## 1.4 Terms

| Term | Meaning |
|---|---|
| Zone apex | The registrable domain itself, for example securitynut.co.uk, as opposed to a hostname beneath it. |
| Edge provider | The reverse proxy or CDN in front of a site that terminates TLS and can apply redirects, headers and security policy before requests reach the origin. |

| Term | Meaning |
|---|---|
| DNS host | The authoritative nameserver operator for the zone. |
| Registrar | The organisation through which the domain is registered, and the only party able to publish DS records at the parent zone. |
| Receiver | A mail server accepting inbound mail and evaluating SPF, DKIM and DMARC for it. |
| Resolver | A recursive DNS server answering queries on behalf of clients, and the party that performs DNSSEC validation. |

# 2. The scoring model

A domain is scored out of 100 across the three pillars. Each control has a fixed weight, partial credit is given for weaker but present configurations, and the total is converted to a letter grade. A domain that scores full marks on every required control is said to meet the MP Standard regardless of bonuses.

## 2.1 Weights

| Pillar | Control | Points | Full marks requires |
|---|---|---|---|
| Mail (40) | SPF | 10 | Record present, ends in -all, ten or fewer DNS lookups |
| | DKIM | 8 | Key found on a common selector, 1024 bits or more |
| | DMARC | 12 | p=reject, sp=reject, adkim=s, aspf=s, pct=100, rua present |
| | MTA-STS | 6 | DNS record, fetchable policy, mode: enforce |
| | TLS-RPT | 4 | Record present |
| DNS (20) | DNSSEC | 10 | DS at the parent and DNSKEY in the zone |
| | CAA | 5 | At least one record, plus an iodef contact |
| | Nameservers | 5 | Two or more, all resolving to addresses |
| Web (40) | HTTPS redirect | 8 | http:// answers 301, 302, 307 or 308 to an https:// location |
| | HSTS | 10 | Header present, max-age of one year or more, includeSubDomains |
| | X-Content-Type-Options | 5 | nosniff |
| | Referrer-Policy | 5 | Header present |
| | Clickjacking protection | 6 | X-Frame-Options DENY or SAMEORIGIN, or CSP frame-ancestors |
| | Permissions-Policy | 4 | Header present |
| | Version disclosure | 2 | No version string in Server, no X-Powered-By |

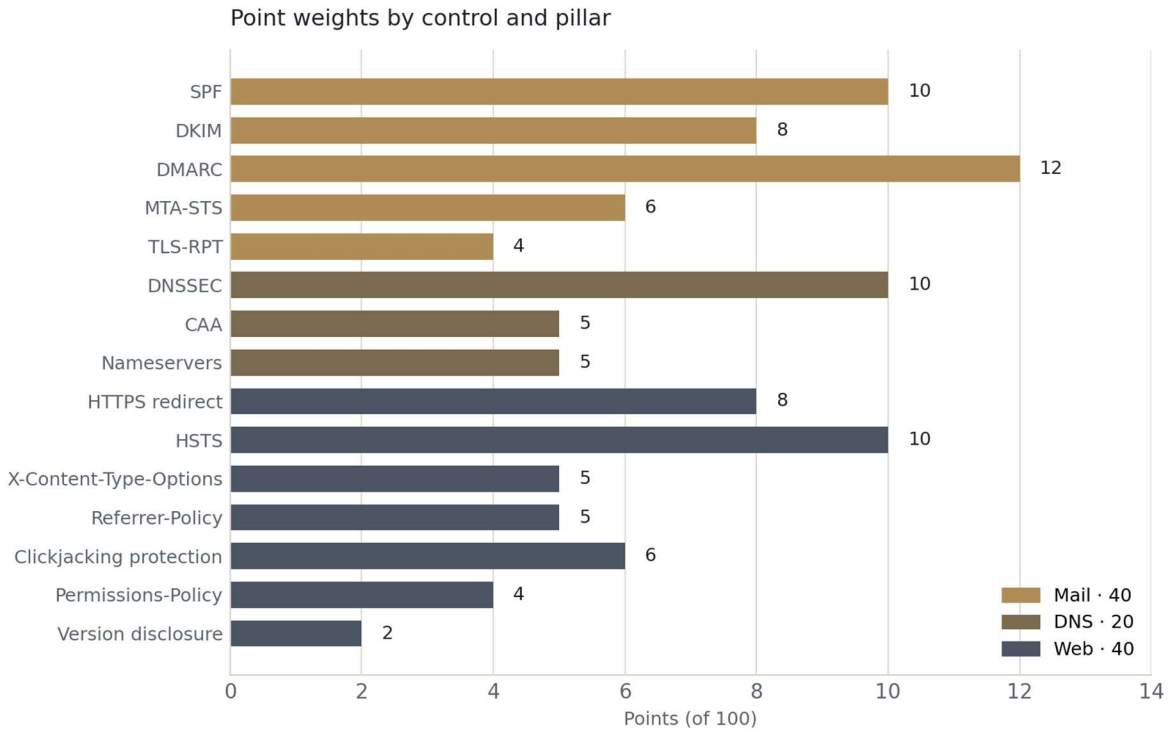


*Figure 1. Point weights by control and pillar. DMARC, HSTS, SPF and DNSSEC carry the most weight because each closes an entire class of attack on its own.*

## 2.2 Grade bands

| Grade | Score | Reading |
|---|---|---|
| A | 90 to 100 | Strong. Meets the standard at 100. |
| B | 75 to 89 | Good, with specific gaps. |
| C | 60 to 74 | Adequate, material gaps. |
| D | 40 to 59 | Weak. Impersonation or downgrade is likely to succeed somewhere. |
| F | below 40 | Largely unprotected. |


*Figure 2. Grade bands. Only a score of 100 with no findings meets the standard.*

## 2.3 Rules that keep the score fair

- **Web not assessed.** If the site refuses the automated fetch (bot protection, timeout, connection failure) the whole Web pillar is excluded and the grade is calculated out of 60. A refusal is reported, not punished.
- **Null mail domains.** A domain with no MX records and an SPF record of exactly v=spf1 -all is declaring that it sends no mail. DKIM, MTA-STS and TLS-RPT are then not applicable and are scored as met.

- **DKIM discovery.** DKIM keys live under selector names the domain owner chooses. The standard probes a list of 43 selectors used by common providers. A key on a private selector name will not be found, so the finding is worded as "not found on common selectors" and carries less weight than DMARC or SPF.
- **Partial credit.** Present but weaker settings earn part of the points (for example DMARC p=quarantine earns half the policy points, SPF ~all earns half the terminator points) so that progress is visible.
- **Bonuses never penalise.** Controls beyond the standard are listed but do not add to or subtract from the score.

## 2.4 What each control defends against

Every control earns its place by closing a specific threat. The matrix below maps the threats the standard is concerned with to the controls that address them. A tick means the control is a primary defence; a circle means it contributes.

| Threat | SPF | DKIM | DMARC | MTA-STS | TLS-RPT | DNSSEC | CAA | NS | Redirect | HSTS | Headers |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Sender spoofing (envelope) | ✓ | | ✓ | | | | | | | | |
| From header impersonation | ○ | ○ | ✓ | | | | | | | | |
| Message tampering in transit | | ✓ | ○ | | | | | | | | |
| Inbound mail downgrade to plaintext | | | | ✓ | ○ | | | | | | |
| Silent mail delivery failure | | | ○ | | ✓ | | | | | | |
| DNS forgery, cache poisoning | | | | | | ✓ | | | | | |
| Zone outage | | | | | | | | ✓ | | | |
| Certificate misissuance | | | | | | ○ | ✓ | | | | |
| Plaintext web session | | | | | | | | | ✓ | ✓ | |
| SSL stripping, downgrade | | | | | | | | | ○ | ✓ | |
| Clickjacking | | | | | | | | | | | ✓ |
| MIME confusion, content sniffing | | | | | | | | | | | ✓ |
| URL and token leakage via Referer | | | | | | | | | | | ✓ |
| Abuse of device features by embeds | | | | | | | | | | | ✓ |
| Targeted exploitation via version fingerprint | | | | | | | | | | | ○ |

### 2.5 Beyond the standard

The following are recognised as good practice and reported when present, but are not required because they depend on the provider, the mail platform or a commitment the standard should not impose on others: BIMI, DANE TLSA on MX hosts, the HSTS preload directive, a Content-Security-Policy and DMARC forensic reporting (ruf).

## 3. Mail pillar

Mail is the most impersonated channel on the internet. The mail pillar establishes who may send as the domain, proves that messages were not altered, tells receivers what to do when those checks fail, and protects mail in transit to the domain. The five controls are designed to work together; each closes a gap the others leave.

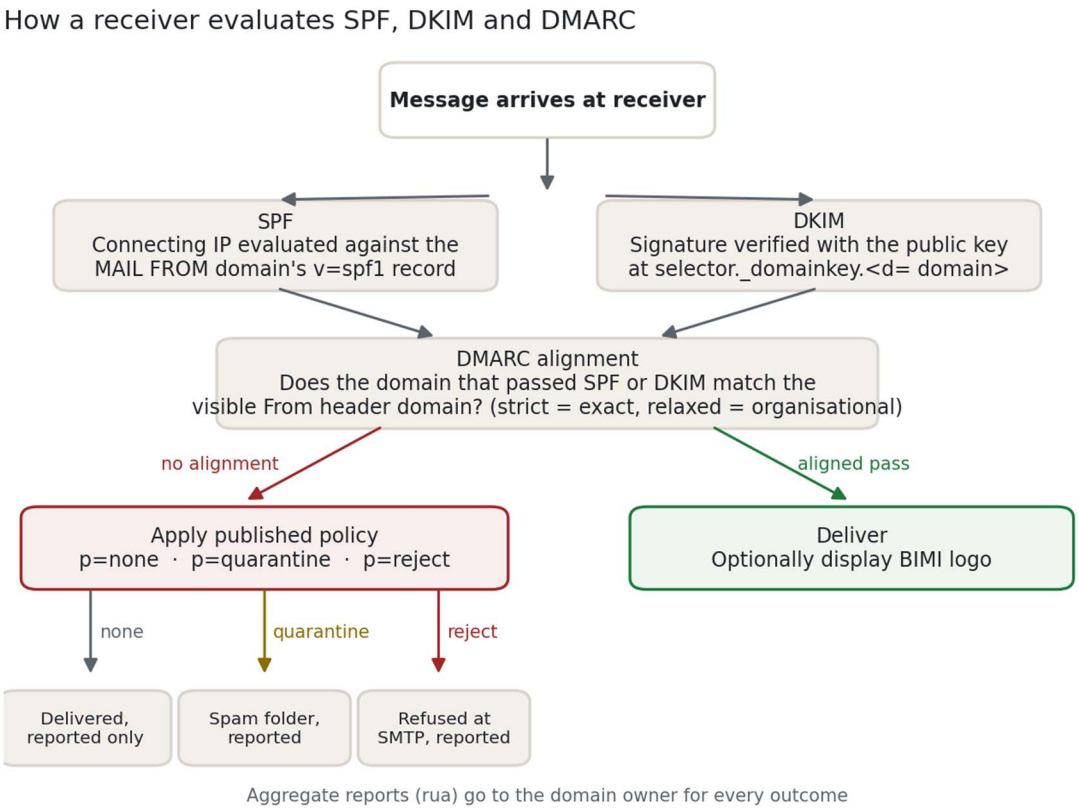


*Figure 3. The receiver side evaluation that SPF, DKIM and DMARC together define. Alignment is the step that binds the technical checks to the address a human sees.*

### 3.1 SPF, Sender Policy Framework

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Mail | 10 | RFC 7208 | TXT at the apex beginning v=spf1, terminating in -all, evaluating within ten DNS lookups |

**What it is**

SPF is a DNS TXT record at the zone apex that lists the servers permitted to send mail using the domain in the SMTP envelope sender (the MAIL FROM address, also called the return path). A receiving server looks up the record, evaluates the connecting IP against it and reaches a result of pass, fail, softfail, neutral or an error.

### How it works

The record is a single string of mechanisms evaluated left to right.

```
securitynut.co.uk  TXT  "v=spf1 include:spf.protection.outlook.com -all"
```

Mechanisms include ip4 and ip6 (literal address ranges), a and mx (the domain's own A or MX hosts), include (evaluate another domain's SPF record and pass if it passes) and all (matches everything, used last). Each mechanism carries a qualifier: + pass (the default), - fail, ~ softfail and ? neutral. The terminating all mechanism therefore sets the default verdict for any sender not matched above it.

Every include, a, mx, ptr, exists and redirect term costs a DNS lookup, and RFC 7208 caps a single evaluation at ten. Exceeding the limit yields a permanent error, which receivers treat as no SPF at all. Nested includes from large providers consume the budget quickly.

### Why it matters

Without SPF anyone can connect to a receiver and announce a return path at the domain. With SPF and a -all terminator, a message from an unlisted server fails, and a DMARC policy can then instruct the receiver to reject it. SPF alone does not protect the visible From header, which is why DMARC is needed to bind the two together.

The ~all softfail terminator is the most common weakness. It tells receivers "this is probably not authorised, but accept it and mark it", which most receivers translate into delivery to the inbox with no visible warning. It was designed as a transition state and is frequently left in place for years.

### What the standard requires and how it is scored

Ten points. Four for a syntactically valid record at the apex, six for a -all terminator (three for ~all, none for ?all, +all or a missing terminator). Three points are deducted if the lookup count exceeds ten, because the record then fails in practice however well intentioned.

Measurement follows include and redirect terms recursively, counting lookup mechanisms and stopping at fifteen queries to protect the assessor. The count and any cap are reported.

### Common failure modes

- ~all left in place after a migration.
- Two SPF records at the apex, which is a permanent error under RFC 7208 (merge them into one).
- Lookup limit exceeded through stacked provider includes (flatten or remove unused includes).
- SPF published only on a subdomain used for bulk mail while the apex has none.

### Implementation notes

Publish exactly one TXT record at the apex. List only providers that actually send as the domain. Use include for providers that publish their own SPF, ip4 or ip6 for fixed relays, and end with -all. Review the record whenever a mail platform is added or retired.

## 3.2 DKIM, DomainKeys Identified Mail

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Mail | 8 | RFC 6376 | At least one public key published under a common selector, 1024 bits or larger (2048 recommended) |

### What it is

DKIM lets a sending system sign each message with a private key and publish the matching public key in DNS. Receivers fetch the key and verify the signature, proving that the message body and selected headers were not modified in transit and that the signer controls the domain's DNS.

### How it works

The sender adds a DKIM-Signature header carrying the domain (d=), a selector (s=), the list of signed headers (h=), a hash of the body (bh=) and the signature itself (b=). The receiver builds the DNS name selector._domainkey.domain, fetches the TXT record, extracts the RSA or Ed25519 public key (p=) and verifies the signature.

```
selector1._domainkey.securitynut.co.uk  TXT
"v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0pfO2Nl4BpfgOKAMpA7iUA3ZY2bF...
 ...UOcn0x1dk7+vY3eIof/heAn6hwrsJnx1de9p/dE3NE3yRx0pBfGOGXwxoj7tAiDQIDAQAB"  (2048-bit RSA,
truncated)

selector2._domainkey.securitynut.co.uk  TXT
"v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3oBniXRKnmZSoZlIxUeRaDqd9S9L..."  (2048-bit
RSA, truncated)
```

Selectors allow several keys to coexist so that keys can be rotated and different platforms can sign with their own. Key length is inferred from the base64 length of p=: a 2048-bit RSA key produces roughly 392 characters of base64; 1024-bit roughly 216.

### Why it matters

SPF checks the connection; DKIM checks the message. A DKIM signature survives forwarding, which SPF does not, and it is the only one of the two that can be tied to the visible From header via DMARC alignment. A domain with DMARC at p=reject and no DKIM will see legitimate forwarded mail rejected, so DKIM is what makes strict DMARC safe to deploy.

Key length matters because a 1024-bit RSA key is within reach of a well resourced attacker and was deprecated by major receivers years ago. 2048 bits is the current norm.

### What the standard requires and how it is scored

Eight points for a key found on any of 43 common selectors with a length of 1024 bits or more; four points if a key is found but is shorter. No points if nothing is found. The selectors found and the inferred key length are reported.

Because selector names are chosen by the domain owner, a not found result may be a false negative. The finding says so explicitly, which is why DKIM carries less weight than SPF and DMARC.

### Common failure modes

- Provider DKIM never enabled, so mail is unsigned and DMARC relies on SPF alone.
- CNAME based provider keys pointing at records that have since been removed.
- Legacy 1024-bit keys still in service.
- Key published under the wrong domain when a platform signs with its own domain rather than the customer's.

### Implementation notes

Enable DKIM signing in every platform that sends as the domain and publish each platform's keys. Rotate keys periodically by publishing a new selector, switching signing, then removing the old record after a grace period.

## 3.3 DMARC, Domain-based Message Authentication, Reporting and Conformance

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Mail | 12 | RFC 9989 | TXT at _dmarc with p=reject, sp=reject, adkim=s, aspf=s, pct=100 and a rua reporting address |

### What it is

DMARC ties SPF and DKIM to the address the recipient actually sees, the From header, and tells receivers what to do when neither check aligns. It also asks receivers to send aggregate reports so the domain owner can see who is sending as them, legitimately or otherwise.

### How it works

The policy lives in a TXT record at _dmarc.domain.

```
_dmarc.securitynut.co.uk  TXT
"v=DMARC1; p=reject; pct=100; sp=reject; ri=86400; adkim=s; aspf=s;
 rua=mailto:151295f7e6044763a804ed784603cbae@dmarc-reports.cloudflare.net;"
```

A message passes DMARC if either SPF or DKIM passes and the domain that passed aligns with the From header domain. Alignment can be relaxed (r, organisational domain matches) or strict (s, exact match). The policy tag p= applies to the domain; sp= applies to subdomains and defaults to p if absent. pct= samples the policy to a percentage of failing mail. rua= receives aggregate XML reports, typically daily, from every participating receiver; ruf= optionally receives forensic samples.

Policies escalate: none (monitor only), quarantine (deliver to spam) and reject (refuse at SMTP time).

### Why it matters

Neither SPF nor DKIM on its own stops an attacker putting the domain in the From header while passing checks for some other domain they control. DMARC alignment closes that gap. At p=reject, a message that fails alignment never reaches a human, which removes the impersonation route used in most business email compromise and invoice fraud.

Strict alignment (adkim=s, aspf=s) ensures a lookalike subdomain or a cousin domain in the organisational tree cannot be used to pass. sp=reject prevents an attacker inventing a subdomain that has no policy of its own. pct=100 ensures the policy actually applies to every message rather than a sample. rua ensures the owner sees what is happening; a policy without reporting is flying blind.

### What the standard requires and how it is scored

Twelve points, the heaviest single control. Three for a valid record; six for p=reject (three for quarantine, none for none); one for sp=reject (inherited from p if sp is absent); half a point each for adkim=s and aspf=s; one for a rua address. Two points are deducted if pct is anything other than 100. Every tag read is reported so the exact shortfall is visible.

### Common failure modes

- p=none left in place indefinitely after an initial monitoring period.
- rua pointing at a mailbox nobody reads, or at a third party that no longer accepts reports for the domain (external destinations must publish an authorisation record).
- pct=10 or similar sampling left over from a staged rollout.
- No sp tag on a domain with p=quarantine, leaving subdomains under a weaker policy than intended.

### Implementation notes

Start at p=none with rua to learn who sends as the domain, fix SPF and DKIM for every legitimate source, move to p=quarantine, then p=reject. Set adkim=s and aspf=s once all sources sign and send from the exact domain. Route reports to a mailbox that is monitored or to an aggregation service.

## 3.4 MTA-STS, SMTP MTA Strict Transport Security

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Mail | 6 | RFC 8461 | TXT at _mta-sts, a policy file at https://mta-sts.domain/.well-known/mta-sts.txt, mode: enforce |

### What it is

SMTP was designed without encryption. Opportunistic TLS (STARTTLS) was added later, but a sending server that is told "no TLS here" simply falls back to plaintext, and an attacker on the path can strip the STARTTLS offer or present any certificate. MTA-STS lets a receiving domain publish a policy that says its mail servers always support TLS with a valid certificate, so senders must refuse to deliver in the clear.

### How it works

Two parts. A DNS TXT record at _mta-sts.domain signals that a policy exists and carries an id that changes whenever the policy changes.

```
_mta-sts.securitynut.co.uk  TXT  "v=STSv1; id=20230615T153000;"
```

The policy itself is fetched over HTTPS from a fixed URL on the mta-sts subdomain.

```
https://mta-sts.securitynut.co.uk/.well-known/mta-sts.txt

version: STSv1
mode: enforce
mx: securitynut-co-uk.mail.protection.outlook.com
max_age: 604800
```

Sending servers cache the policy for max_age seconds and, in enforce mode, will only deliver over TLS to an MX host that matches the mx patterns and presents a certificate valid for that name. Mode testing reports failures without blocking delivery; mode none withdraws a policy.

### Why it matters

Without MTA-STS, mail to the domain can be silently downgraded to plaintext by anyone between the sender and the MX, read or altered, and the sender will never know. MTA-STS makes downgrade a delivery failure instead, and because the policy is fetched over HTTPS from a name protected by the web PKI, it does not depend on DNSSEC. It pairs with TLS-RPT so the domain learns when a sender could not comply.

### What the standard requires and how it is scored

Six points. Two for the DNS record, two for a policy file that is fetchable and parses, two for mode: enforce (one for testing, none for none). The mode found is reported.

### Common failure modes

- DNS record published but the mta-sts host has no certificate or serves the wrong content type (must be text/plain).
- Policy left in testing mode after the trial period.
- mx patterns that do not match the certificate names presented by the actual MX hosts, so compliant senders refuse delivery.

- Changing MX hosts without updating the policy and the id.

#### Implementation notes

Serve the policy from a small static host or edge function on mta-sts.domain with a valid certificate. Start in testing with TLS-RPT enabled, watch the reports for a week, then switch to enforce and bump the id. Update mx and id whenever mail hosting changes.

### 3.5 TLS-RPT, SMTP TLS Reporting

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Mail | 4 | RFC 8460 | TXT at _smtp._tls with a rua reporting address |

#### What it is

TLS-RPT is the reporting companion to MTA-STS and DANE. Sending servers that attempted to deliver to the domain report, once a day, how many sessions succeeded, how many failed to negotiate TLS and why, and which policy they were applying.

#### How it works

```
_smtp._tls.securitynut.co.uk  TXT  "v=TLSRPTv1; rua=mailto:tls-reports@securitynut.co.uk"
```

Reports arrive as JSON, usually gzipped, to the mailto or https destination given. They identify the sending organisation, the policy type (sts, tlsa or no-policy-found), the MX hosts involved and a count of successful and failed sessions with failure reason codes such as certificate-host-mismatch or starttls-not-supported.

#### Why it matters

MTA-STS in enforce mode will cause compliant senders to refuse delivery when something is wrong with the receiving side's TLS. Without TLS-RPT the domain owner only discovers this when someone phones to say their mail bounced. With it, a misconfiguration surfaces in the next day's report, and the transition from testing to enforce can be made with evidence rather than hope.

#### What the standard requires and how it is scored

Four points for a valid record with a rua destination. It is a small control on its own but it is what makes the MTA-STS enforce requirement responsible.

#### Common failure modes

- Record absent because MTA-STS was set up from a guide that omitted it.
- rua pointing at a mailbox that filters JSON attachments as spam.

#### Implementation notes

Point rua at a dedicated alias on a monitored mailbox, or at a report processing service. The same mailbox can also receive DMARC aggregate reports and CAA violation reports to keep all security reporting in one place.

## 4. DNS pillar

Everything else in the standard is published through DNS, so DNS must itself be trustworthy and resilient. The DNS pillar asks whether answers can be verified, whether certificate issuance is constrained, and whether the zone would survive the loss of a nameserver.

## 4.1 DNSSEC, DNS Security Extensions

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| DNS | 10 | RFC 4033, 4034, 4035, 6840 | Zone signed with DNSKEY published, and the matching DS record published at the parent through the registrar |

### What it is

DNSSEC adds digital signatures to DNS records so that a validating resolver can prove an answer came from the zone owner and was not altered or forged. It establishes a chain of trust from the root, through the top level domain, down to the zone.

### How it works

The DNS host signs every record set in the zone with a zone signing key and publishes the public keys in DNSKEY records. A DS (delegation signer) record, containing a hash of the key signing key, is published in the parent zone (for example co.uk) by the registrar. A validating resolver fetches the DS from the parent, checks it against the zone's DNSKEY, then checks each record's RRSIG signature. Any mismatch is treated as a failure and the answer is discarded.

```
securitynut.co.uk  DS      2371 13 2
8507450b3a2b0b8425774a8f737056fc0decc3f09aff68bdfef4b384e656f725
securitynut.co.uk  DNSKEY  257 3 13
mdsswUyr3DPW132mOi8V9xESWE8jTo0dxCjjnopKl+GqJxpVXckHAeF+KkxLbxILfDLUT0rAK9iUzy1L53eKGQ==
securitynut.co.uk  DNSKEY  256 3 13
oJMRESz5E4gYzS/q6XDrvU1qMPYIjCWzJaOau8XNEZeqCYKD5ar0IRd8KqXXFJkqmVfRvMGPmM1x8fGAa2XhSA==
```

Algorithm 13 (ECDSA P-256 with SHA-256) is the current norm: small signatures, fast validation. The chain requires both halves. DNSKEY without a DS at the parent is an island of security nobody can reach; a DS without a matching DNSKEY makes the zone fail validation entirely.

### Why it matters

Without DNSSEC, a resolver has no way to know that the MX, the SPF record, the DMARC policy or the address of the website are the ones the owner published. Cache poisoning and on-path manipulation of DNS answers undermine every other control in this standard. DNSSEC is the mechanism that makes the rest of the DNS trustworthy, and it is a prerequisite for DANE.

### What the standard requires and how it is scored

Ten points, split evenly: five for a DS record at the parent and five for DNSKEY records in the zone. The two halves are checked independently so that a broken chain (one without the other) is reported as exactly that.

### Common failure modes

- DNSSEC enabled at the DNS host but the DS never added at the registrar, so nothing validates.
- DS left at the registrar after changing DNS host, breaking the zone until removed.
- Registrar that does not support DS records for the TLD, requiring a transfer.

### Implementation notes

Enable signing at the DNS host, copy the DS record it produces to the registrar, confirm with an external validator that the chain is complete. When changing DNS host, remove the old DS first, move, then publish the new DS.

## 4.2 CAA, Certification Authority Authorization

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| DNS | 5 | RFC 8659 | At least one CAA record set at the apex, including an iodef contact |

### What it is

CAA records declare which certificate authorities are permitted to issue certificates for the domain. Every publicly trusted CA is required by the CA/Browser Forum baseline requirements to check CAA before issuing and to refuse if it is not listed. An iodef record gives CAs an address to report attempted violations.

### How it works

```
securitynut.co.uk  CAA  0 iodef "mailto:security@securitynut.co.uk"
securitynut.co.uk  CAA  0 issue "letsencrypt.org"
securitynut.co.uk  CAA  0 issue "pki.goog; cansignhttpexchanges=yes"
securitynut.co.uk  CAA  0 issue "digicert.com; cansignhttpexchanges=yes"
securitynut.co.uk  CAA  0 issue "comodoca.com"
securitynut.co.uk  CAA  0 issue "ssl.com"
securitynut.co.uk  CAA  0 issuewild ... (one issuewild record per CA above)
```

issue names a CA permitted to issue ordinary certificates; issuewild governs wildcard certificates; iodef gives a mailto or https reporting endpoint. The CA looks up CAA at the exact name and climbs towards the apex until it finds a record set, so records at the apex govern every subdomain unless overridden. An empty issue value (issue ";") forbids issuance entirely.

### Why it matters

Certificate misissuance, whether through a compromised CA account, a validation flaw or social engineering, gives an attacker a certificate browsers will trust for the domain. CAA shrinks the set of CAs that could be tricked from all of them to the few the owner actually uses, and iodef means an attempt is reported rather than silent. It does not replace Certificate Transparency monitoring, which detects misissuance after the fact; the two complement each other.

Where an edge provider manages certificates for the domain and rotates between several CAs, the CAA set will name those CAs. The protection is then "only CAs my provider uses" rather than a single CA of choice, which is still a meaningful reduction.

### What the standard requires and how it is scored

Five points: four for any CAA record set at the apex, one more when an iodef record is present. CAA at a subdomain only, or none at all, scores zero.

### Common failure modes

- No CAA published because "it might break renewal" (a CA already in use is simply listed).
- issue records for a CA no longer used, blocking a new provider's issuance.
- iodef pointing at a mailbox that does not exist.

### Implementation notes

List every CA that issues for the domain, including those used by the edge provider or hosting platform, add issuewild if wildcards are used, and add iodef pointing at a monitored security mailbox. Where the edge provider publishes CA records automatically once any CAA exists, a single iodef record is sufficient to trigger that.

### 4.3 Nameserver resilience

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| DNS | 5 | RFC 2182, RFC 1034 | Two or more nameservers, each resolving to at least one address |

**What it is**

The NS records at the apex name the authoritative servers for the zone. Resolvers pick among them; if none answer, the domain and everything under it disappears from the internet, including the MX, the website and the DMARC policy.

**How it works**

NS records are published both in the parent zone (the delegation) and in the zone itself. Each nameserver hostname must resolve to A or AAAA addresses, either through normal resolution or through glue records supplied by the parent when the nameserver sits inside the zone it serves. Resolvers query whichever server answers fastest and fail over to others on timeout.

**Why it matters**

A single nameserver is a single point of failure for every other control in this standard. Two or more on separate networks, ideally with anycast distribution, mean a server failure, network partition or denial of service against one does not take the zone offline. Nameservers that do not resolve to an address are worse than absent because resolvers waste time trying them.

**What the standard requires and how it is scored**

Five points: three for two or more NS records (one if only one exists), two more when every nameserver resolves to at least one address.

**Common failure modes**

- Stale NS record for a decommissioned server.
- Nameservers all within one provider network with no diversity.
- Glue records at the registrar not updated after an address change.

**Implementation notes**

Use a DNS host that provides at least two anycast nameservers on distinct networks. Verify each NS hostname resolves. Keep the delegation at the registrar in step with the zone.

## 5. Web pillar

The web pillar assesses the site served at the domain: whether it insists on encrypted transport and whether the response headers protect visitors from the browser side attacks that transport encryption does not address. Headers are measured from one GET request to the site over HTTPS; the redirect is measured from one GET over plain HTTP with redirects held so the response can be inspected.

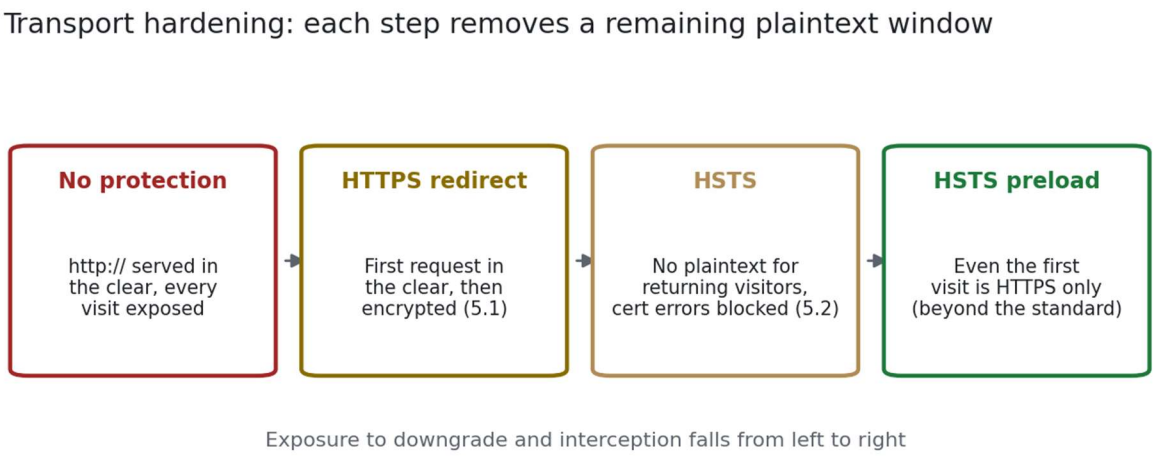


*Figure 4. Transport hardening as a ladder. The standard requires the first two rungs and records the third as beyond the standard.*

## 5.1 HTTPS redirection

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 8 | RFC 9110 (status codes), RFC 8446 (TLS 1.3) | A request to http://domain/ answers 301, 302, 307 or 308 with an https:// Location |

### What it is

Any request arriving over plain HTTP must be redirected to the HTTPS equivalent before any content is served. This is the first line: without it, a visitor who types the bare domain, or follows an old http link, receives an unencrypted page an attacker on the path can read or rewrite.

### How it works

The server or edge provider inspects the scheme of the incoming request and responds with a redirect status and a Location header pointing at the same path on https://. 301 and 308 are permanent and cacheable, which is preferable. 307 and 302 are accepted. The redirect should preserve the path and query string.

### Why it matters

Plaintext HTTP allows passive reading and active modification of everything on the page, including injecting scripts or replacing download links. A redirect ensures the plaintext exposure is limited to the first request and that request carries no content. HSTS then removes even that first request for returning visitors.

### What the standard requires and how it is scored

Eight points for a proper redirect. Four if the plain HTTP request could not be completed but the HTTPS site loads (the redirect is probably present but unmeasurable). Zero if http:// serves content with a 200. The status returned is reported.

### Common failure modes

- HTTP serving a full copy of the site, so both versions exist and get indexed separately.
- Redirect chains that bounce through www and back before reaching HTTPS.
- Redirect on the apex but not on subdomains.

### Implementation notes

Enable the redirect at the edge for the entire zone so it applies to every hostname. Test with a plain HTTP request using a client that does not follow redirects; browsers with HSTS cached will not show the real behaviour.

## 5.2 HSTS, HTTP Strict Transport Security

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 10 | RFC 6797 | Strict-Transport-Security with max-age of at least 31536000 and includeSubDomains |

### What it is

HSTS is a response header that tells the browser to remember, for a set period, that this domain is HTTPS only. For the lifetime of that memory the browser rewrites any http:// URL for the domain to https:// before sending it, and refuses to let the user click past a certificate error.

### How it works

```
strict-transport-security: max-age=31536000; includeSubDomains; preload   (as served by securitynut.co.uk)
```

max-age is the retention period in seconds; one year is the accepted minimum for a domain that is committed to HTTPS. includeSubDomains extends the rule to every subdomain, closing the gap where a cookie scoped to the parent could be leaked over HTTP to an insecure subdomain. preload is a flag indicating willingness to be included in the browser preload lists, which hard code the domain as HTTPS only in the browser itself so that even the very first visit is protected. Submission to the preload list is a separate, deliberate step and is not required by the standard.

The header is only honoured when received over HTTPS with a valid certificate, which is why the redirect in 5.1 must come first.

### Why it matters

The redirect protects a visitor only after the first plaintext request has been made, and that request can be intercepted. SSL stripping attacks work exactly there: the attacker answers the plaintext request themselves and never lets the browser reach HTTPS. HSTS removes the plaintext request for anyone who has visited before, and blocks the certificate error click through that other downgrade attacks depend on. includeSubDomains matters because attackers target the weakest subdomain, not the one the owner is thinking about.

### What the standard requires and how it is scored

Ten points: four for the header being present, three for max-age of one year or more, three for includeSubDomains. preload is noted as beyond the standard. The header value is reported verbatim.

### Common failure modes

- Header set on the apex but not on subdomains, or without includeSubDomains.
- max-age of a few hours or days left over from testing.
- Header sent over HTTP as well, where browsers ignore it.

### Implementation notes

Enable at the edge for the whole zone. Confirm every subdomain serves HTTPS before adding includeSubDomains, because a subdomain that cannot serve HTTPS becomes unreachable for a year. Consider preload submission only once the estate has been stable for some time; it is very hard to reverse.

## 5.3 X-Content-Type-Options

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 5 | WHATWG Fetch Standard | X-Content-Type-Options: nosniff |

### What it is

A header that instructs the browser to trust the Content-Type the server declares and not to guess (sniff) the type from the content. Sniffing was introduced to cope with badly configured servers and became an attack surface.

### How it works

```
X-Content-Type-Options: nosniff
```

With nosniff, a response served as text/plain will not be executed as JavaScript even if it looks like JavaScript, a response served as an image will not be interpreted as HTML, and cross origin script and style requests are blocked unless the type is correct.

### Why it matters

Content sniffing lets an attacker who can get a file onto the origin (an upload, a comment, a user profile) have it interpreted as something more dangerous than the server intended. The header also enables Cross-Origin Read Blocking in modern browsers, which protects sensitive responses from being read by cross site attacks. It is the cheapest header in the set and there is no legitimate reason to omit it.

### What the standard requires and how it is scored

Five points for the header with value nosniff. No partial credit.

### Common failure modes

- Set on HTML responses but not on static assets or API responses.
- Edge and origin both setting it with different casing, which is harmless but untidy.

### Implementation notes

Send on every response, including JSON, images, downloads and error pages. Where the edge provider offers a zone wide toggle, use it and also set it in application code so the protection does not depend on the edge.

## 5.4 Referrer-Policy

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 5 | W3C Referrer Policy | Referrer-Policy header present (strict-origin-when-cross-origin or stricter recommended) |

### What it is

Controls what the browser puts in the Referer header when a visitor follows a link or loads a resource from the site. Without a policy, browsers historically sent the full URL of the current page to the destination, including path and query string.

### How it works

```
Referrer-Policy: strict-origin-when-cross-origin
```

The recommended value sends the full URL for same origin requests, only the origin (scheme and host) for cross origin requests, and nothing at all when going from HTTPS to HTTP. Stricter options are same-origin (nothing cross origin) and no-referrer (nothing ever). A meta tag can carry the same policy but the header is authoritative and applies to every response.

### Why it matters

URLs leak. A path can identify a document, a query string can carry a session token, a search term or a reset code. Sending those to third party origins (analytics, fonts, embedded content, outbound links) discloses them to parties who have no need of them and who may log them indefinitely. The recommended policy keeps analytics referrer data useful at the origin level while preventing path and parameter leakage.

### What the standard requires and how it is scored

Five points for any Referrer-Policy header. The value is reported so a weak choice (unsafe-url, no-referrer-when-downgrade) is visible even though it scores.

### Common failure modes

- Policy set only via meta tag on the home page and not on other responses.
- unsafe-url set to keep a legacy analytics report working.

### Implementation notes

Send strict-origin-when-cross-origin on every response. Tighten to same-origin or no-referrer for private tools where even the origin should not be disclosed.

## 5.5 Clickjacking protection

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 6 | RFC 7034 (X-Frame-Options), W3C CSP Level 3 (frame-ancestors) | X-Frame-Options DENY or SAMEORIGIN, or a Content-Security-Policy containing frame-ancestors |

### What it is

Prevents the site from being loaded inside an iframe on another site. Clickjacking places a transparent frame of the target over decoy content so that a visitor's clicks on the decoy land on real controls in the framed site.

### How it works

```
X-Frame-Options: DENY

Content-Security-Policy: frame-ancestors 'none'
```

X-Frame-Options is the older header with two useful values, DENY (never frame) and SAMEORIGIN (only the site itself may frame it). frame-ancestors in a Content-Security-Policy is the modern replacement, supports a list of permitted framing origins, and takes precedence where both are present. Sending both is harmless and covers older browsers.

### Why it matters

Any site with a button that does something on the visitor's behalf (submit, pay, follow, delete, grant) can be abused through a frame. Even a site with no such controls can be framed to lend its appearance to a phishing page. Denying framing removes the whole class of attack at the cost of one header.

### What the standard requires and how it is scored

Six points for X-Frame-Options DENY or SAMEORIGIN, or for any CSP that includes frame-ancestors. No partial credit; the mechanism found is reported.

### Common failure modes

- Header set only on the login page.
- X-Frame-Options ALLOW-FROM, which browsers never widely supported and is now ignored.

### Implementation notes

Send X-Frame-Options DENY on every response unless the site genuinely needs to be framed by itself, in which case SAMEORIGIN. Add frame-ancestors to the CSP when one is introduced.

## 5.6 Permissions-Policy

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 4 | W3C Permissions Policy | Permissions-Policy header present |

### What it is

Declares which powerful browser features the site and anything it embeds may use: camera, microphone, geolocation, payment, USB, sensors and so on. A feature that is disabled cannot be enabled by a script on the page, including a third party script.

### How it works

```
Permissions-Policy: camera=(), microphone=(), geolocation=()
```

Each directive names a feature and an allowlist. An empty allowlist () disables the feature for the page and all frames within it. self allows the origin only; a quoted origin allows a specific embedded site. The policy is inherited by iframes, so a compromised or malicious embed cannot ask the visitor for permissions the top level page has switched off.

### Why it matters

Most sites have no reason to access the camera, microphone or location. Declaring that up front means a supply chain compromise of an embedded script, or an injected script, cannot prompt the visitor for them. It is defence in depth for the visitor's device rather than the site's data, and it signals intent clearly in the response.

### What the standard requires and how it is scored

Four points for any Permissions-Policy header. The standard requires the header to exist and does not mandate particular directives, because the right set depends on what the site actually does.

### Common failure modes

- Header absent because nothing on the site uses those features, which is precisely when it costs nothing to send.
- Legacy Feature-Policy header only, which current browsers no longer honour.

### Implementation notes

Send at minimum camera=(), microphone=(), geolocation=(). Add payment=(), usb=(), and others as appropriate to the site.

### 5.7 Version disclosure

| Pillar | Weight | Reference | Requirement |
|---|---|---|---|
| Web | 2 | Good practice (OWASP ASVS 14.3) | No version number in the Server header and no X-Powered-By header |

#### What it is

Servers and frameworks often announce themselves in response headers: Server: Apache/2.4.41, X-Powered-By: PHP/7.4.3. This tells an attacker exactly which vulnerability list to consult.

#### How it works

The Server header may name the software but should not carry a version. X-Powered-By should be removed entirely. Edge providers typically replace Server with their own name, which is acceptable.

#### Why it matters

Version disclosure does not create a vulnerability but it removes the attacker's need to guess, and it advertises when a site has fallen behind on patching. Removing it is trivial and is a standard item on any hardening checklist.

#### What the standard requires and how it is scored

Two points, lost if Server contains a version pattern (digits, dot, digits) or if X-Powered-By is present at all.

#### Common failure modes

- Framework defaults never changed.
- Origin headers passing through an edge provider that does not strip them.

#### Implementation notes

Configure the server or framework to omit version tokens and remove X-Powered-By. Verify at the edge, not just at the origin.

## 6. Beyond the standard

These controls are reported when present. They are not required because each depends on a factor outside the domain owner's sole control or represents a commitment the standard should not impose.

| Control | Reference | What it adds | Why not required |
|---|---|---|---|
| BIMI | IETF draft (bimi-overview) | Displays the sender's verified logo in supporting mail clients when DMARC is at enforcement. | Requires a Verified Mark Certificate and a registered trademark for most receivers to display it; a commercial rather than security control. |
| DANE TLSA on MX | RFC 7672, RFC 6698 | Pins the MX hosts' certificates in DNS, protected by DNSSEC, so senders can verify TLS without the web PKI. | Requires DNSSEC and MX hosts whose certificates the domain owner controls; many mail platforms do not support it. |
| HSTS preload | Browser preload lists | Hard codes the domain as HTTPS only in the browser, protecting even the first visit. | Effectively irreversible for the lifetime of browser releases; a commitment each owner should make deliberately. |

| Control | Reference | What it adds | Why not required |
|---|---|---|---|
| Content-Security-Policy | W3C CSP Level 3 | Restricts where scripts, styles, images and frames may load from, defeating most injected script attacks. | A correct policy is specific to each site's resources; a generic requirement would produce either broken sites or meaningless policies. |
| DMARC ruf | RFC 9989 | Forensic samples of individual failing messages. | Many receivers no longer send them for privacy reasons; aggregate reports are sufficient for the standard. |

# 7. Complementary controls not scored

The following are part of the author's own posture and are recommended alongside the standard. They are not scored because they cannot be measured reliably from outside, or because they concern operations rather than the published result.

- **Minimum TLS 1.2 with TLS 1.3 enabled.** TLS 1.0 and 1.1 are deprecated (RFC 8996). An external assessor cannot force a downgrade from a serverless environment, so this is verified by hand rather than by the grader.
- **Certificate Transparency monitoring.** Every publicly trusted certificate is logged; a monitor alerts when one is issued for the domain. CAA constrains who may issue; CT monitoring detects when someone did.
- **A single security reporting mailbox.** DMARC rua, TLS-RPT rua, CAA iodef and CT alerts for every domain routed to one monitored mailbox, with an alias per domain so each record can name its own address.
- **Consistent headers in code, not only at the edge.** Edge toggles are convenient, but setting the header block in the application means a change of provider does not silently remove protection.
- **Canonical hostname discipline.** One canonical host per site, with www and alternative TLDs redirecting to it, so that policies, headers and indexing all apply to a single name.

# 8. Assessment method

A conformance assessment is performed entirely from public sources in a single pass. The assessment application does not retain the submitted domain or assessment results. Normal infrastructure security and operational logs may still be processed by the hosting provider.

- DNS queries over DNS-over-HTTPS for A, AAAA, MX, NS, SOA, TXT, CAA, DS, DNSKEY, the _dmarc, _mta-sts, _smtp._tls and default._bimi names, TLSA for _25._tcp on each MX host, glue addresses for each nameserver, and TXT for each of 43 DKIM selectors under _domainkey.
- A recursive walk of SPF include and redirect terms, counting lookup mechanisms and capped at fifteen queries.
- An HTTPS fetch of the MTA-STS policy file at the well-known location.
- One GET of https://domain/ following redirects, reading the final URL and the security response headers.
- One GET of http://domain/ with redirects held, reading the status and Location header.

Each control is scored as described in sections 3 to 5, findings are ordered by points lost, and a scorecard, a prioritised fix list and the underlying evidence are presented. The assessment is repeatable and two runs against an unchanged domain produce the same result.

### 8.1 Conformance statement

A domain that scores 100 with no findings may be described as meeting the MP Standard, version 1.1. A domain scoring below 100 is described by its grade and the number of shortfalls. Bonuses are reported separately and do not form part of the statement.

## 9. References

| Reference | Title |
|---|---|
| RFC 7208 | Sender Policy Framework (SPF) for Authorizing Use of Domains in Email |
| RFC 6376 | DomainKeys Identified Mail (DKIM) Signatures |
| RFC 9989 | Domain-based Message Authentication, Reporting, and Conformance (DMARC) |
| RFC 8461 | SMTP MTA Strict Transport Security (MTA-STS) |
| RFC 8460 | SMTP TLS Reporting |
| RFC 4033, 4034, 4035 | DNS Security Introduction and Requirements; Resource Records for the DNS Security Extensions; Protocol Modifications for the DNS Security Extensions |
| RFC 6840 | Clarifications and Implementation Notes for DNS Security (DNSSEC) |
| RFC 8659 | DNS Certification Authority Authorization (CAA) Resource Record |
| RFC 2182 | Selection and Operation of Secondary DNS Servers |
| RFC 6797 | HTTP Strict Transport Security (HSTS) |
| RFC 7034 | HTTP Header Field X-Frame-Options |
| RFC 7672 | SMTP Security via Opportunistic DANE TLS |
| RFC 8996 | Deprecating TLS 1.0 and TLS 1.1 |
| RFC 9110 | HTTP Semantics |
| W3C | Content Security Policy Level 3; Referrer Policy; Permissions Policy |
| WHATWG | Fetch Standard (nosniff, CORB) |
| CA/Browser Forum | Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates (CAA checking) |

## 10. Document history

| Version | Date | Change |
|---|---|---|
| 1.0 | 6 September 2026 | First issue. Codifies the standard as implemented in SecurityNut IP v2.1.0 to v2.2.6, following the benchmark domain reaching full conformance on 6 September 2026. |
| 1.1 | 7 September 2026 | Live records from securitynut.co.uk. Currency and review statement added (1.3). Figures added: weights, grade bands, mail evaluation flow, transport ladder. Threat matrix added (2.4). Appendices added: quick reference, glossary, live report. No change to controls or weights. |

# Appendix A. Quick reference: a conforming domain

The complete set of records and headers a domain publishes to meet the standard, using securitynut.co.uk as the worked example. Substitute the domain, the mail provider's include and MX names, the reporting addresses and the certificate authorities in use.

### DNS records

```
@                     TXT  "v=spf1 include:<mail provider SPF> -all"
<selector>._domainkey TXT  "v=DKIM1; k=rsa; p=<2048-bit public key>"   (one per signing
platform)
_dmarc                TXT  "v=DMARC1; p=reject; sp=reject; adkim=s; aspf=s; pct=100;
rua=mailto:<reports>"
_mta-sts              TXT  "v=STSv1; id=<YYYYMMDDTHHMMSS>"
_smtp._tls            TXT  "v=TLSRPTv1; rua=mailto:<reports>"
@                     CAA  0 issue "<each CA in use>"
@                     CAA  0 issuewild "<each CA in use>"   (if wildcards are issued)
@                     CAA  0 iodef "mailto:security@<domain>"
@                     NS   <at least two nameservers on distinct networks>
DNSSEC                DNSKEY in the zone, DS at the parent via the registrar
```

### MTA-STS policy file

```
https://mta-sts.<domain>/.well-known/mta-sts.txt   (text/plain, valid certificate)

version: STSv1
mode: enforce
mx: <each MX hostname or pattern>
max_age: 604800
```

### HTTP behaviour and response headers

```
http://<domain>/*  ->  301 https://<domain>/*

Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Content-Type-Options: nosniff
Referrer-Policy: strict-origin-when-cross-origin
X-Frame-Options: DENY
Permissions-Policy: camera=(), microphone=(), geolocation=()
Server: <no version>   and no X-Powered-By
```

### Verify

Run the domain through ip.securitynut.co.uk. A conforming domain shows A, 100 / 100, "Meets the MP Standard" and an empty What to fix list. Use Print report to keep the evidence.

# Appendix B. Glossary

| Term | Meaning |
|---|---|
| Alignment (DMARC) | The requirement that the domain which passed SPF or DKIM matches the From header domain, either exactly (strict) or at the organisational level (relaxed). |
| BIMI | Brand Indicators for Message Identification. Publishes a logo for mail clients to display once DMARC is enforced. |
| CAA | Certification Authority Authorization. DNS records naming the CAs permitted to issue certificates for the domain. |
| CT | Certificate Transparency. Public append-only logs of every issued certificate. |
| DANE | DNS-based Authentication of Named Entities. Pins TLS certificates in DNS via TLSA records, protected by DNSSEC. |

| Term | Meaning |
|---|---|
| DKIM | DomainKeys Identified Mail. Cryptographic signing of messages with the public key published in DNS. |
| DMARC | Domain-based Message Authentication, Reporting and Conformance. Policy and reporting layer over SPF and DKIM. |
| DNSKEY | The public keys used to sign a DNSSEC zone. |
| DNSSEC | DNS Security Extensions. Signed DNS records validated along a chain of trust from the root. |
| DS | Delegation Signer. A hash of the zone's key signing key, published in the parent zone. |
| Edge provider | The reverse proxy or CDN that terminates TLS in front of a site. |
| HSTS | HTTP Strict Transport Security. A header instructing browsers to use HTTPS only for a stated period. |
| iodef | Incident Object Description Exchange Format. In CAA, the reporting address for issuance violations. |
| MTA-STS | SMTP MTA Strict Transport Security. A policy requiring TLS with a valid certificate for inbound mail. |
| MX | Mail Exchanger. DNS records naming the servers that accept mail for a domain. |
| Preload (HSTS) | Inclusion of a domain in browsers' built-in HTTPS-only lists. |
| RDAP | Registration Data Access Protocol. The structured successor to WHOIS. |
| RPKI | Resource Public Key Infrastructure. Cryptographic validation of BGP route origins. |
| rua / ruf | DMARC and TLS-RPT aggregate report address / DMARC forensic report address. |
| Selector | The label under _domainkey that identifies a specific DKIM key. |
| SPF | Sender Policy Framework. DNS record listing the servers permitted to send mail for a domain. |
| STARTTLS | The SMTP command that upgrades a plaintext connection to TLS opportunistically. |
| TLS-RPT | SMTP TLS Reporting. Daily reports from senders about TLS success and failure to the domain's MX. |
| TLSA | The DNS record type used by DANE to publish certificate associations. |
| Zone apex | The registrable domain itself, as opposed to a hostname beneath it. |

# Appendix C. Live report, securitynut.co.uk

The conformance report for securitynut.co.uk produced by ip.securitynut.co.uk v2.2.6 on 7 September 2026. The benchmark domain meets the standard with no findings.

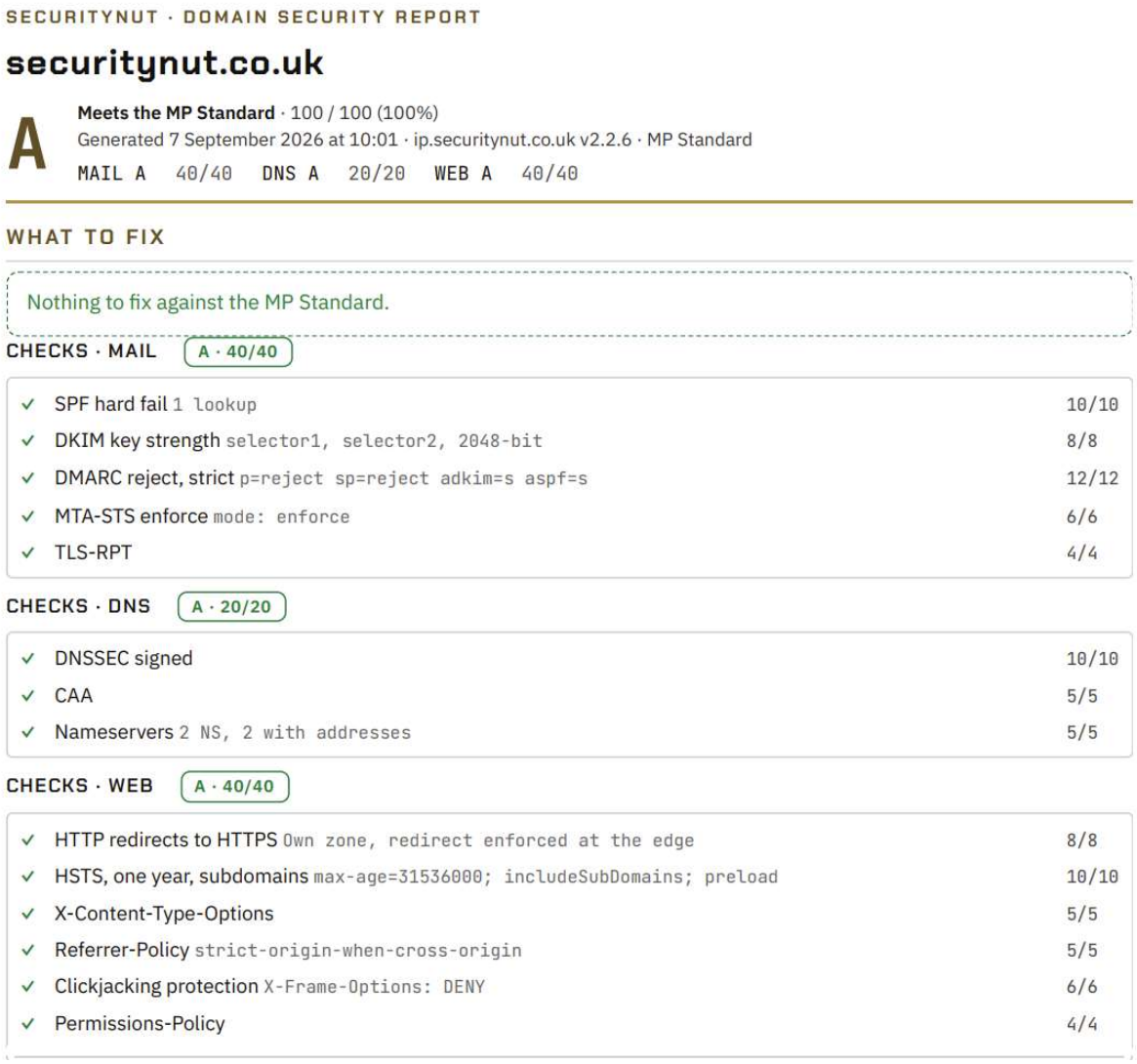


*Figure 5. Domain Security Report for securitynut.co.uk, A, 100 / 100.*

*<eof>*