DANIMASTER · FULL SECURITY AUDIT REPORT

example.com

Scanned November 10, 2026 · report SAMPLE · SAMPLE REPORT

www.danimaster.com/security-audit

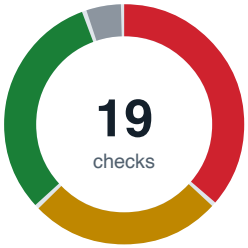
This is a sample report. It shows a real scan of `example.com`, the public demonstration domain reserved by IANA, so no private information appears here. Your own report has the same sections for your website, with the fix steps matched to what we found.

SCORE



Grade F · Poor

ALL 19 CHECKS



Failed	7
Warnings	5
Passed	6
Check by hand	1
Could not tell	0

OPEN FINDINGS BY SEVERITY

Critical	0
High	4
Medium	5
Low	3

Serious gaps found. These should be fixed soon. 12 items need attention; 6 checks passed.

About this website

IP address	104.20.23.154, 172.66.147.243
Network	CLOUDFLARENET - Cloudflare, Inc. (AS13335)
Behind a protection network	Cloudflare: the address above is the network's, not your own server's
Server software	cloudflare
Home page	answers with status 200
Page title	Example Domain
HTTPS certificate	issued by SSL Corporation on 2026-09-26, expires 2026-12-25 (45 days left)
Domain created	1995-08-14 (31.2 years ago)
Domain expires	2027-08-13
Registrar	RESERVED-Internet Assigned Numbers Authority
Name servers	elliott.ns.cloudflare.com, hera.ns.cloudflare.com

Public information only: DNS, the certificate the site presents, RIPEstat for the network and place, and RDAP for the domain registration.

Fix these first

Ordered by how serious the problem is. Each one has the full steps further down.

1. high **Strict-Transport-Security is sent on every HTTPS response** (HDR-001 · Cloudflare or web server)
2. high **A Content-Security-Policy limits where scripts and other resources load from** (HDR-002 · Cloudflare or application)
3. high **Only TLS 1.2 and TLS 1.3 are accepted** (TLS-001 · Cloudflare and web server)
4. high **Plain HTTP redirects to HTTPS, with no mixed content** (TLS-004 · Cloudflare)
5. medium **At least two nameservers on independent infrastructure answer for the domain** (DNS-004 · DNS provider)
6. medium **Pages cannot be framed by other sites (clickjacking protection)** (HDR-003 · Cloudflare or web server)

This site sits behind Cloudflare. **8** of the 12 open findings can be fixed there, and 3 more can be helped by it.

Results by area

DNS



1 fail · 1 warn · 1 pass

Browser security headers



4 fail · 2 warn · 4 pass

Encryption and certificate



2 fail · 2 warn · 1 pass

Findings and how to fix them

Fail **Strict-Transport-Security is sent on every HTTPS response** high

`HDR-001 · Browser security headers`

What we found: header missing

WHY IT MATTERS

HSTS tells the browser to use HTTPS only for the site, which closes the window in which a first HTTP request could be intercepted and blocks click-through on certificate errors.

HOW TO FIX IT

`Strict-Transport-Security: max-age=31536000; includeSubDomains`. Start with a short max-age (for example 300) for a day, confirm nothing breaks, then raise it. Add `always` in nginx so it is also sent on error responses.

Fix in Cloudflare **Where to fix:** Cloudflare or web server
SSL/TLS > Edge Certificates > HTTP Strict Transport Security (HSTS). Start with a short max-age.

HOW TO CHECK IT YOURSELF

```
curl -sI https://example.com/ | grep -i strict-transport-security
```

WHAT A PASS LOOKS LIKE

The header is present with `max-age` of at least 15552000 (180 days); one year (31536000) is the usual target. `includeSubDomains` is present once every subdomain serves HTTPS.

RISK AND HOW TO UNDO IT

Browsers cache the policy; if HTTPS breaks later, visitors cannot bypass it until max-age ends. Do not add `preload` until you accept that removal from the preload list takes months.

Fail **A Content-Security-Policy limits where scripts and other resources load from** high
HDR-002 · Browser security headers

What we found: no Content-Security-Policy header

WHY IT MATTERS

CSP is the strongest browser-side defence against cross-site scripting: even if an attacker injects markup, the browser refuses to run scripts the policy does not allow.

HOW TO FIX IT

Deploy first as `Content-Security-Policy-Report-Only` with a report endpoint, fix what the reports show, then enforce. Example starting point: `default-src 'self'; script-src 'self'; object-src 'none'; base-uri 'self'; frame-ancestors 'self'`.

Cloudflare helps **Where to fix:** Cloudflare or application
Rules > Transform Rules > Modify Response Header. Start with Content-Security-Policy-Report-Only. Pages that use per-request nonces must set it in the application.

HOW TO CHECK IT YOURSELF

```
curl -sI https://example.com/ | grep -i content-security-policy
```

Then read the policy for `script-src` containing `'unsafe-inline'`, `'unsafe-eval'`, bare `*`, or `data:`.

WHAT A PASS LOOKS LIKE

A policy is present. `script-src` uses nonces or hashes (not `'unsafe-inline'`), `object-src 'none'`, `base-uri 'self'`, and `frame-ancestors` is set.

RISK AND HOW TO UNDO IT

An enforced policy that is too tight breaks pages (third-party widgets, analytics, inline handlers). Report-only mode first is the rollback-free path.

Fail

Only TLS 1.2 and TLS 1.3 are accepted

high

`TLS-001 · Encryption and certificate`

What we found: accepts TLS 1.0 and 1.1

WHY IT MATTERS

TLS 1.0 and 1.1 were formally deprecated in 2021 because their handshake and cipher designs can no longer be made safe. A server that still accepts them lets an attacker on the network push a client down to the weak protocol where one is supported.

HOW TO FIX IT

nginx: `ssl_protocols TLSv1.2 TLSv1.3;`. Apache: `SSLProtocol -all +TLSv1.2 +TLSv1.3`. Reload the service and re-run the test. If the site sits behind Cloudflare, also set the zone's minimum TLS version to 1.2 (SSL/TLS, Edge Certificates); the edge, not your server, then answers the handshake.

Fix in Cloudflare **Where to fix:** Cloudflare and web server

Cloudflare: SSL/TLS > Edge Certificates > Minimum TLS Version = 1.2. Also set ssl_protocols TLSv1.2 TLSv1.3 on the origin web server.

HOW TO CHECK IT YOURSELF

```
openssl s_client -connect example.com:443 -servername example.com -tls1_1 </dev/null 2>&1
| grep -E 'Protocol|alert|handshake failure'
openssl s_client -connect example.com:443 -servername example.com -tls1_2 </dev/null 2>&1
| grep Protocol
openssl s_client -connect example.com:443 -servername example.com -tls1_3 </dev/null 2>&1
| grep Protocol
```

Some OpenSSL builds refuse to offer 1.0/1.1 at all; in that case use `nmap --script ssl-enum-ciphers -p 443 example.com`.

WHAT A PASS LOOKS LIKE

TLS 1.0 and 1.1 handshakes fail. TLS 1.2 succeeds. TLS 1.3 succeeds (or is simply not offered on very old stacks, which is a medium finding).

RISK AND HOW TO UNDO IT

Very old clients (Windows XP, Android 4, old embedded devices) stop connecting. Check access logs for the TLS version in use before the change if clients matter. Rollback: restore the previous directive and reload.

Fail

Plain HTTP redirects to HTTPS, with no mixed content

high

TLS-004 · Encryption and certificate

What we found: http:// answered 200 instead of redirecting to https

WHY IT MATTERS

If port 80 serves content, or redirects through an unencrypted hop to a different host, a network attacker can intercept the first request. Mixed content (an HTTPS page loading scripts or images over HTTP) lets an attacker alter the page.

HOW TO FIX IT

Add a port-80 server block that returns `301 https://$host$request_uri`. Change hard-coded `http://` URLs in templates and the database to `https://` or protocol-relative paths. Then add HSTS (HDR-001).

Fix in Cloudflare **Where to fix:** Cloudflare
SSL/TLS > Edge Certificates > Always Use HTTPS and Automatic HTTPS Rewrites.

HOW TO CHECK IT YOURSELF

```
curl -sI http://example.com/ | head -n 5
curl -sIL http://example.com/some/path | grep -iE '^(HTTP|location)'
curl -s https://example.com/ | grep -oE '(src|href)="http://[^"]+' | head
```

WHAT A PASS LOOKS LIKE

Port 80 answers with a single permanent redirect (301 or 308) to the HTTPS version of the same host and path. The page source contains no `http://` sub-resources.

RISK AND HOW TO UNDO IT

Do not redirect to HTTPS before the certificate is valid on every hostname. Rollback: remove the redirect block.

Fail

At least two nameservers on independent infrastructure answer for the domain

medium

DNS-004 · DNS

What we found: 0 nameservers: (independence of the providers needs a manual look)

WHY IT MATTERS

If the only server that can answer for your zone goes down, every name under the domain stops resolving. RFC 2182 advises placing secondary servers in different networks and places so one outage cannot silence the zone. Two servers in the same rack, or under one account with one provider, share the same failure.

HOW TO FIX IT

Delegate to at least two servers on separate networks. For important domains add a second DNS provider and keep both in sync through zone transfer from a hidden primary or the providers' APIs. Update the nameserver list at the registrar so it matches the zone.

Where to fix: DNS provider
Use at least two independent name servers.

HOW TO CHECK IT YOURSELF

```
dig +short NS example.com
for ns in $(dig +short NS example.com); do
  echo "== $ns"; dig +short A "$ns"
  dig +norecurse +short SOA example.com @"$ns"
done
```

Every listed server should answer with the same SOA serial. Then run `whois` on each server address and compare the network owners to see whether they are really different.

WHAT A PASS LOOKS LIKE

At least two nameservers are delegated, each answers authoritatively with the same serial, and they sit on different networks. A single managed DNS provider with its own geographically spread network is a reasonable result if the dependency is documented; two servers of one small host are a finding.

RISK AND HOW TO UNDO IT

Two providers can disagree if a record is changed in only one, which gives intermittent wrong answers; automate the sync and compare both after every change. DNSSEC with two providers needs extra care. Rollback: remove the added server from the registrar delegation first, then from the zone.

Fail

Pages cannot be framed by other sites (clickjacking protection)

medium

HDR-003 · Browser security headers

What we found: frame-ancestors: no; X-Frame-Options: none

WHY IT MATTERS

If another site can place your page in an invisible frame, it can trick visitors into clicking buttons they cannot see. The `frame-ancestors` directive controls who may frame the page.

HOW TO FIX IT

Add `frame-ancestors 'self'` to the CSP and `X-Frame-Options: SAMEORIGIN`. If the site is meant to be embedded, list the exact embedding origins instead.

Fix in Cloudflare **Where to fix:** Cloudflare or web server
Transform Rules > Modify Response Header: X-Frame-Options or CSP frame-ancestors.

HOW TO CHECK IT YOURSELF

```
curl -sI https://example.com/ | grep -iE 'content-security-policy|x-frame-options'
```

WHAT A PASS LOOKS LIKE

`frame-ancestors 'none'` or `'self'` (or an explicit allow-list) is present in the CSP. `X-Frame-Options: DENY` or `SAMEORIGIN` is sent as a fallback for very old browsers.

RISK AND HOW TO UNDO IT

Setting `'none'` breaks legitimate embedding (for example inside your own admin or a partner's portal). Check for known embedders first.

Fail

X-Content-Type-Options: nosniff is sent

medium

HDR-004 · Browser security headers

What we found: header missing

WHY IT MATTERS

Without it, a browser may guess a file's type from its content and run an uploaded text or image file as a script. The header makes the browser trust the declared type.

HOW TO FIX IT

`add_header X-Content-Type-Options nosniff always;` (nginx) or `Header always set X-Content-Type-Options nosniff` (Apache). Make sure every resource is served with the correct Content-Type.

Fix in Cloudflare **Where to fix:** Cloudflare or web server

Transform Rules > Modify Response Header: X-Content-Type-Options: nosniff.

HOW TO CHECK IT YOURSELF

```
curl -sI https://example.com/ | grep -i x-content-type-options
```

WHAT A PASS LOOKS LIKE

The header is present with the exact value `nosniff` on HTML, script and stylesheet responses.

RISK AND HOW TO UNDO IT

Resources served with a wrong type (for example a script sent as text/plain) stop working. Fix the type rather than removing the header.

Warning

DNSSEC is enabled where the registrar and DNS host support it, and validates

medium

DNS-005 · DNS

What we found: no DS record: DNSSEC not enabled (record the decision if your registrar or DNS host cannot do it)

WHY IT MATTERS

DNSSEC lets a validating resolver check that an answer was signed by the zone's owner, which protects against forged or tampered DNS responses. It only works end to end when the zone is signed and the parent zone holds a matching DS record. A half-finished setup is worse than none: a DS record that does not match the signing keys makes validating resolvers return errors, and the domain appears to be down for a large share of users.

HOW TO FIX IT

Enable signing at the DNS host, then copy the DS data it shows into the registrar's DNSSEC settings. Prefer a host that rolls keys automatically (operational practices are in RFC 6781). Before moving DNS to another provider, plan the DS record: remove it, wait for its TTL to pass, and add a new one only after the new provider signs the zone.

Cloudflare helps **Where to fix:** DNS and registrar
Cloudflare: DNS > Settings > DNSSEC > Enable, then add the DS record it shows at your registrar.

HOW TO CHECK IT YOURSELF

```
dig +short DS example.com
dig +short DNSKEY example.com
dig +noall +comments example.com A @1.1.1.1 | grep -E 'flags|status'
dig +cd +noall +comments example.com A @1.1.1.1 | grep status
```

A signed domain has DS and DNSKEY records, and the validating resolver's reply carries the `ad` flag. If the normal query returns `SERVFAIL` but the `+cd` query (validation switched off) returns `NOERROR`, the DNSSEC chain is broken.

WHAT A PASS LOOKS LIKE

Either the zone is signed, the DS at the parent matches a published key, and validation succeeds, or there is no DS at all and the lack of DNSSEC is a recorded decision because the registrar or DNS host does not support it. A DS with no matching signed zone is a failure.

RISK AND HOW TO UNDO IT

Mistakes take the domain offline for validating resolvers, so change DS records only with the key material in hand and re-run the test right after. Rollback: delete the DS record at the registrar; the zone then resolves unsigned once cached DS answers expire.

Warning

Cipher suites are modern and give forward secrecy

medium

TLS-003 · Encryption and certificate

What we found: accepts static-RSA key exchange (no forward secrecy): AES128-GCM-SHA256

WHY IT MATTERS

Forward secrecy means that recorded traffic cannot be decrypted later if the server's private key leaks. Suites with RC4, 3DES, export-grade keys, NULL encryption or static RSA key exchange do not meet current guidance.

HOW TO FIX IT

Prefer a maintained generator for the server and OpenSSL version in use rather than hand-written lists, and re-generate when the server software is upgraded. Keep the list short; do not re-enable legacy suites to please one old client.

Cloudflare helps **Where to fix:** Web server
Cloudflare manages its edge ciphers. Set modern ciphers on the origin web server.

HOW TO CHECK IT YOURSELF

```
nmap --script ssl-enum-ciphers -p 443 example.com
```

Look for any suite rated below grade A, any suite using RC4, 3DES, NULL, EXPORT or anonymous (`anon`) key exchange, and any TLS 1.2 suite that does not begin with ECDHE or DHE.

WHAT A PASS LOOKS LIKE

Every offered TLS 1.2 suite uses ECDHE (or DHE with at least 2048-bit parameters) and an AEAD cipher (AES-GCM or ChaCha20-Poly1305). TLS 1.3 suites are used as offered by the library.

RISK AND HOW TO UNDO IT

Removing suites can lock out old clients. Test with the oldest client you must support. Rollback: restore the previous cipher directive.

Warning **Referrer-Policy limits what URLs leak to other sites** low

HDR-005 · Browser security headers

What we found: header missing (browser default applies)

WHY IT MATTERS

The browser sends the previous page's address to the next site. URLs often contain tokens, search terms or account identifiers that should not leave your origin.

HOW TO FIX IT

`Referrer-Policy: strict-origin-when-cross-origin` as the general choice; `no-referrer` for sensitive areas such as account and billing pages.

Fix in Cloudflare **Where to fix:** Cloudflare or web server

Transform Rules > Modify Response Header: Referrer-Policy.

HOW TO CHECK IT YOURSELF

```
curl -sI https://example.com/ | grep -i referrer-policy
```

WHAT A PASS LOOKS LIKE

A policy is present and is `strict-origin-when-cross-origin` , `same-origin` , or `no-referrer` . `unsafe-url` and `no-referrer-when-downgrade` fail.

RISK AND HOW TO UNDO IT

Some analytics and affiliate tools rely on full referrers; check reports after the change.

Warning **Permissions-Policy switches off browser features the site does not use** low

HDR-006 · Browser security headers

What we found: header missing

WHY IT MATTERS

If your site does not need the camera, microphone, geolocation or payment APIs, an injected script or an embedded third party should not be able to request them either.

HOW TO FIX IT

`Permissions-Policy: camera=(), microphone=(), geolocation=(), payment=()`. Allow a feature only for `self` or a named origin if the site really uses it.

Fix in Cloudflare **Where to fix:** Cloudflare or web server
Transform Rules > Modify Response Header: Permissions-Policy.

HOW TO CHECK IT YOURSELF

```
curl -sI https://example.com/ | grep -i permissions-policy
```

WHAT A PASS LOOKS LIKE

A policy is present and disables the features the site does not use, for example `camera=(), microphone=(), geolocation=()`.

RISK AND HOW TO UNDO IT

Disabling a feature the site does use (video calls, maps, payment buttons) breaks it. List the features in use before choosing the policy.

Warning

DNS CAA records restrict which authorities may issue certificates

low

TLS-005 · Encryption and certificate

What we found: no CAA record found (low severity)

WHY IT MATTERS

A CAA record tells certificate authorities that only the listed ones may issue for the domain. It does not stop a determined attacker, but it blocks accidental or opportunistic mis-issuance and gives a clear audit trail.

HOW TO FIX IT

Add for example `example.com. CAA 0 issue "letsencrypt.org"` (replace with the CA you use). If you use several CAs list each one.

Fix in Cloudflare **Where to fix:** DNS
DNS > Records > add CAA records for the certificate authorities you use. Confirm the certificate still renews afterwards.

HOW TO CHECK IT YOURSELF

```
dig +short CAA example.com
dig +short CAA www.example.com
```

WHAT A PASS LOOKS LIKE

At least one CAA `issue` record exists for the apex and names only the authority actually in use. An `iodef` record with a monitored mailbox is a bonus.

RISK AND HOW TO UNDO IT

A CAA record for the wrong CA will make the next renewal fail. Add the record for the current CA first, and test a renewal in a staging window.

Checks to do by hand or that were inconclusive

The scanner could not judge these from the outside. They are not failures.

Check by hand **Certificate renewal is automatic and monitored** high

TLS-006 · Encryption and certificate

What we found: renewal automation lives on the server; see kb/tls/TLS-006.md

WHY IT MATTERS

Most certificate outages come from a renewal job that silently stopped, not from a hard attack. Automation using ACME removes the human step; monitoring catches the day automation breaks.

HOW TO FIX IT

Enable the ACME client's timer or cron job, and add an expiry check from outside the server (a monitor that connects over TLS and reads the notAfter date).

Where to fix: Web server
Automate renewal on the origin and monitor expiry.

HOW TO CHECK IT YOURSELF

```
systemctl list-timers | grep -iE 'certbot|acme'
certbot certificates 2>/dev/null
crontab -l 2>/dev/null | grep -iE 'certbot|acme'
certbot renew --dry-run
```

Use the equivalent for the ACME client in use (acme.sh, Caddy, Traefik, lego).

WHAT A PASS LOOKS LIKE

A scheduled renewal exists, a dry run succeeds, and an external check alerts when any certificate has fewer than 14 days left.

RISK AND HOW TO UNDO IT

Renewal hooks that reload the web server must be tested; a failing hook leaves a renewed certificate unused. Keep the old certificate path in place until a reload is confirmed.

Everything that passed (6)

CHECK	AREA	SEVERITY
DNS-009 Public DNS does not expose internal hostnames or private addresses	DNS	low

CHECK		AREA	SEVERITY
HDR-007	Session and authentication cookies carry Secure, HttpOnly and SameSite	Browser security headers	high
HDR-008	Responses with private data are not cached by browsers or shared caches	Browser security headers	medium
HDR-009	Server and framework version banners are not exposed	Browser security headers	low
HDR-010	CORS allows only named origins and never combines a wildcard with credentials	Browser security headers	high
TLS-002	Certificate is valid, complete, matches the name, and is not close to expiry	Encryption and certificate	high

Not covered yet

> These areas are checked only after you prove the website is yours, so this report does not include them: **Email authentication, Mail server connection, DNS server hardening**. Verification is the next step if you want them covered.

How to read this report

A **fail** is a clear gap we observed. A **warning** is weaker or partial. A **pass** met the test. The score starts at 100 and drops by more for serious findings. A finding can be wrong, and a clean result is not proof of safety, so confirm each one with the check-it-yourself step before changing anything. Make a backup before changes to a live server. This report is general information and is not legal, privacy or compliance advice.