Security report for

harbourline.example

Executive summary

C

Overall grade

Weighted average across 4 security checks — the latest result of each type.

Produced 17 October 2026, 19:36 UTC. Every result in this report is the most recent completed run of that check. A description of each check, and how to read its grade, is at the end.

Contents

Executive summary . . . 3
Controls tested . . . 4
Overview and checks performed . . . 7
- Web App Scan — Crawl & Attack . . . 7
- SSL/TLS Check . . . 7
- Email Security Check . . . 7
- HTTP Security Headers . . . 7

What these checks are . . . 7
Findings . . . 7
- Critical (1) . . . 7
- High (3) . . . 8
- Medium (4) . . . 8
- Low (7) . . . 8
- Informational (1) . . . 9

Reference: what each check is . . . 9

C

Overall grade C (50/100)

Weighted average of 4 security checks — the latest completed run of each. The score is the same average on a 0–100 scale, so a month that fixes four issues without changing the letter still shows movement.

Controls tested

40	13	4	5
pass	need attention	failing	not checked

40 of 62 controls pass. Every control we tested is listed on the next page with what we saw, including the ones that passed — a report that only lists problems cannot tell you the difference between a control that is healthy and one nobody looked at.

What to deal with first

Most severe first. The CWE reference is the industry identifier for the class of weakness, for pasting into a ticket or mapping to a control framework.

- **SQL Injection — Input is placed directly into a database query**
  CRITICAL · CWE-89 · confirmed
- **Reflected XSS — Input is reflected with the characters needed to inject markup**
  HIGH · CWE-79 · confirmed
- **Directory Listing — Anyone can browse the files in one of your folders**
  HIGH · CWE-548 · inferred
- **DMARC Policy Not Enforced — Impersonation is being watched, not blocked**
  HIGH · CWE-290 · inferred
- **Content Security Policy Allows Inline Script — Your content rules allow scripts written directly into the page**
  MEDIUM · CWE-693 · confirmed
- **Expiring Certificate — Your certificate expires in 16 days**
  MEDIUM · CWE-298 · confirmed

What this covers

This report covers the checks that have been run on this asset: Web App Scan — Crawl & Attack, SSL/TLS Check, Email Security Check, HTTP Security Headers. A check that has never been run on this asset does not appear here at all, so an empty area means untested rather than clean.

Every control this asset's checks tested, with what we saw. A control that passed is listed for the same reason as one that failed: without it, the absence of a problem is indistinguishable from the absence of a check.

Web application

25 of 37 pass

Control	Verdict	What we saw
Transport Encryption — The application answers only over an encrypted connection	Pass	every page we read came back over https
Mixed Content — No encrypted page pulls in unencrypted content	Pass	no page mixed plain http resources into an https page
Security Headers — Security headers are set, on every page rather than on one	Attention	Missing security header: X-Content-Type-Options
Content Security Policy — The content policy restricts what a page may run	Attention	Your content rules allow scripts written directly into the page
Cookie Security Attributes — Cookies carry their protections	Attention	The cookie PHPSESSID can be read by scripts in the page
Session Token in URL — No session token travels in a web address	Pass	no address carried something that identifies a session
Cleartext Password Submission — No password is submitted where it can be read or logged	Pass	every password field posted over https and in the body rather than the address
Cacheable Session Response — Pages that sign someone in are kept out of shared caches	Pass	no response we read set a session without keeping it out of shared caches
Password Field Autocomplete — Password boxes tell the browser not to store what is typed	Pass	no password box we read was left free to be remembered and filled in again
Cross-Site Request Forgery — Forms that change something carry an anti-forgery token	Pass	every form that changes something carried a token another site cannot guess
CSRF Origin Validation — A form cannot be submitted from another site	Pass	no form accepted a submission claiming to come from somewhere else
Default Credentials — No password that ships with the software still works	Not checked	none of the forms the crawl found could be tried with a default password
Object Level Authorization — One account cannot read another account's records	Not checked	this scan was not given two accounts to compare
Function Level Authorization — An ordinary account cannot reach an administrator's pages	Not checked	this scan was not given two accounts to compare
Cross-Domain Form Submission — No form posts to another company's server	Pass	every form posted back to a host you control
Cross-Origin Resource Sharing — Only the sites you chose may read your pages	Pass	no answer to a request claiming another origin opened the page to that origin
JSON Web Token Handling — A sign-in token is signed, strong, and expires	Pass	no sign-in token we were given was unsigned, signed with a guessable key or left without an expiry
JSON Web Token Signature Verification — A forged sign-in token is refused	Not checked	this scan had no sign-in token to test, or found nothing behind a login to test one against
GraphQL Introspection — No GraphQL interface describes itself to strangers	Pass	no interface we found would hand over its schema
Directory Listing — No folder lists its contents	Fail	Anyone can browse the files in one of your folders
Backup and Configuration Files — Nothing is served that was not meant to be	Pass	none of the conventional addresses for a configuration file, a backup or a repository answered
Information Disclosure — Responses do not name files, versions or paths	Attention	This application announces the exact software and versions it runs

Control	Verdict	What we saw
security.txt — A published security contact is current	Pass	the published contact has not expired
DOM-based XSS — No page builds itself out of a value the browser carries	Pass	no script we read takes part of the address, the referrer or the window name into the page or into a redirect
Cross-Site Scripting — Input is escaped where it reaches the page	Fail	Input is reflected with the characters needed to inject markup
Injection — Input does not reach the database, the shell, a template or eval	Fail	Input is placed directly into a database query
Input Validation — Values the application did not expect are handled	Pass	a number outside any plausible range and an empty value were answered rather than failed on
Path Traversal — Files cannot be read off the server's disk	Pass	no path we sent reached a file outside the application
Open Redirect — A redirect cannot be pointed at another site	Pass	no redirect followed an address we supplied
Server-Side Request Forgery — The server cannot be made to fetch an address of our choosing	Pass	nothing we sent made the application look up or fetch a host we control
XML External Entity — XML we send is parsed without following what it points at	Pass	no XML we sent made the parser fetch an entity from outside the document
Deserialization of Untrusted Data — Data from the request is not rebuilt into objects	Pass	nothing we sent was deserialized into a live object by the application
Host Header Injection — The address the application answers for is not trusted blindly	Pass	the application did not act on a host header we chose
HTTP Response Splitting — A request value cannot become a response header	Pass	nothing we sent ended a header line and started one of our own
HTTP Methods Allowed — The server refuses request methods it has no use for	Pass	it refused the method that echoes a request straight back, and it advertises none of the write methods a web server has no use for
Published API Definition — An API description is published deliberately, if at all	Pass	no machine-readable API description answered at a conventional address
Unrestricted File Upload — An upload is restricted, and what is uploaded is not served back	Not checked	no form the crawl could test took a file, so nothing was uploaded

Certificates and transport

8 of 9 pass

Control	Verdict	What we saw
Certificate Validity — Certificate is in date	Attention	expires in 16 days
Certificate Hostname Validation — Certificate matches the hostname	Pass	the name on the certificate covers harbourline.example
Certificate Chain of Trust — Certificate chain is trusted	Pass	issued by Example Trust CA R4
CA-Issued Certificate — Certificate is not self-signed	Pass	issued by Example Trust CA R4
Key Strength and Signature Algorithm — Key and signature are strong enough	Pass	ECDSA 256-bit, ECDSA-SHA256
TLS 1.3 — Modern TLS is offered	Pass	TLS 1.3 offered
Deprecated TLS Versions — Obsolete TLS versions are off	Pass	nothing older than TLS 1.2 is offered
Forward Secrecy — Past traffic stays private if the key is stolen	Pass	6 encryption methods are offered, and all of them keep past traffic private
Known TLS Vulnerabilities — No known TLS weaknesses	Pass	8 checks run, none matched

Email authentication

2 of 8 pass

Control	Verdict	What we saw
SPF — SPF record published	Attention	v=spf1 include:spf.mail.example.net ~all
SPF Hard Fail — SPF refuses unlisted senders	Attention	ends in ~all — unlisted senders are still delivered
DKIM — DKIM signing key published	Pass	key at selector selector1
DMARC — DMARC record published	Attention	v=DMARC1; p=none; rua=mailto:dmarc@harbourline.example
DMARC Policy — DMARC blocks impersonation	Fail	p=none — forged mail is delivered
DMARC Aggregate Reporting — DMARC reports collected	Pass	reports go to mailto:dmarc@harbourline.example
MTA-STS — Encrypted mail delivery is required	Attention	no MTA-STS policy published
TLS-RPT — Failed mail encryption is reported	Attention	no TLS-RPT record

Security headers

5 of 8 pass

Control	Verdict	What we saw
Strict-Transport-Security	Pass	max-age=31536000
Content-Security-Policy	Pass	default-src 'self'; script-src 'self' 'unsafe-inline'; frame-ancestors 'self'
X-Frame-Options	Pass	SAMEORIGIN
X-Content-Type-Options	Attention	Missing.
Referrer-Policy	Attention	Missing.
Permissions-Policy	Pass	camera=(), microphone=(), geolocation=()
X-XSS-Protection	Pass	Not set (or disabled), which is correct — it is deprecated.
Server	Attention	nginx/1.24.0

harbourline.example

Latest result of 4 security checks | 17 October 2026

C

Overall grade
Average across the 4 checks below — the latest result of each type.

Checks performed

F **Web App Scan — Crawl & Attack**
Input on the shipment tracking page reaches the database unescaped.

B **SSL/TLS Check**
The certificate is valid and the protocols are current, but it expires in 16 days.

C **Email Security Check**
Anyone can send email that looks like it came from this domain and receivers are asked to deliver it anyway.

C **HTTP Security Headers**
4 of 6 core security headers are set; X-Content-Type-Options and Referrer-Policy are missing.

What these checks are

One line each. The full description of every check, and how to read its grade, is at the end of this report.

Web App Scan — Crawl & Attack
Everything the crawl does, and then we test what it found by sending it crafted values.

SSL/TLS Check
The certificate this server presents, and the encryption it agrees to use.

Email Security Check
The published DNS records that decide who may send email as this domain.

HTTP Security Headers
The security instructions this site sends to a browser with each page.

1	3	4	7	1
CRITICAL	HIGH	MEDIUM	LOW	INFO

Findings (16)

Critical (1)

CRITICAL Web App Scan — Crawl & Attack · CWE-89

SQL Injection — Input is placed directly into a database query

High (3)

HIGH Web App Scan — Crawl & Attack · CWE-79

Reflected XSS — Input is reflected with the characters needed to inject markup

HIGH Web App Scan — Crawl & Attack · CWE-548

Directory Listing — Anyone can browse the files in one of your folders

HIGH Email Security Check · CWE-290

DMARC Policy Not Enforced — Impersonation is being watched, not blocked

Medium (4)

MEDIUM Web App Scan — Crawl & Attack · CWE-693

Content Security Policy Allows Inline Script — Your content rules allow scripts written directly into the page

MEDIUM SSL/TLS Check · CWE-298

Expiring Certificate — Your certificate expires in 16 days

MEDIUM Email Security Check · CWE-290

SPF Soft Fail — Your SPF record does not ask for impersonated mail to be refused

MEDIUM Email Security Check · CWE-319

Missing MTA-STS — Mail to this domain can be delivered unencrypted

Low (7)

LOW Web App Scan — Crawl & Attack · CWE-1004

Cookie Without HttpOnly Flag — The cookie PHPSESSID can be read by scripts in the page

LOW Web App Scan — Crawl & Attack · CWE-693

Missing Security Header — Missing security header: X-Content-Type-Options

LOW Web App Scan — Crawl & Attack · CWE-693

Missing Security Header — Missing security header: Referrer-Policy

LOW Email Security Check · CWE-16

Missing TLS-RPT — No reports when encrypted delivery fails

LOW HTTP Security Headers · CWE-693

Missing Security Header — Missing security header: X-Content-Type-Options

LOW HTTP Security Headers · CWE-693

Missing Security Header — Missing security header: Referrer-Policy

LOW HTTP Security Headers · CWE-200

Software Version Disclosure — Server discloses software: Server

Informational (1)

INFO Web App Scan — Crawl & Attack · CWE-200

Software Version Disclosure — This application announces the exact software and versions it runs

Reference: what each check is

For a reader who did not run these checks: what each one examined, why it matters, and how its grade is arrived at.

Web App Scan — Crawl & Attack

What we checked

We do the whole of the crawling scan first — follow the links, record every address, parameter and form, and read what comes back. Then we go back to what we found and TEST it, by submitting values chosen to make a badly built application misbehave. We try to make a database answer a question it was never asked, to get a command run on the server, to get our own markup rendered back to another visitor, to read a file outside the application's own folder, and to make the application fetch an address of our choosing. We try published default passwords on the sign-in forms we are able to submit to — at most three per account name, so we cannot lock out a real administrator. A form that carries an anti-forgery token is left alone and reported as NOT TESTED rather than as clean, and on a modern framework that is most login forms. Where you have given us two sign-ins AND told us which of them is the higher-privileged one, we ask for the same pages as each and compare the answers. Two accounts you have not ranked, or two at the same level, are not compared at all — we will not invent an order between accounts you consider equal — and the report says so rather than reporting nothing found. Two things follow from all that and are said plainly rather than buried. THIS SCAN CHANGES DATA on the application it is pointed at: where a form stores what it is given, we store an inert marker in it, and the report lists the places a marking rule submitted to. One gap in that list we would rather state than have you discover — a file left by the upload test is not yet enumerated there. And we deliberately submit NOTHING to forms that look destructive — a delete, a cancel, a reset. Those stay on the map and the report counts them; they are simply never given a value of ours. Point this at a target you are ready to have tested.

Why it matters

Reading an application tells you what it exposes. Only submitting to it tells you whether it mishandles what it is given — and that is where the breaches come from. A form that passes what you type straight into a database query, a page that renders another visitor's markup, an address that hands over another customer's record because nothing checked whose it was: none of these is visible from the outside, and none of them appears in a scan that only looks. They are also the findings that matter most, because each one is a working way in rather than a weakness somebody might combine with something else.

How to read the grade

The letter is driven by the worst finding in the run, and the report names which one drove it. Everything the ATTACKING pass claims, it demonstrates — a value that came back rendered, a database error that appears only in the tampered request, an address that answered for an account that should not have reached it — so an attacking finding is not a suspicion we are passing on to you. Be aware that this run also carries the whole of the crawling scan's reading, and a few of those are inferred rather than demonstrated, so the letter can be driven by one of them; every finding says which kind it is. An A means everything this scan was able to test answered the way it should. It does not mean the application is safe: what we could not test is reported as not tested rather than counted as a pass, which is why a run that stopped at its budget, a sign-in we could not complete, or a target with only one account says so in as many words. Checks that need our out-of-band listener report that they were not checked when it is not configured, rather than reporting clean.

SSL/TLS Check

What we checked

We connected to this server the way a browser does and examined the certificate it presented and the encryption it agreed to use: who issued the certificate, whether it is still valid and matches the name, which protocol versions the server still accepts, and which encryption ciphers it will agree to. We also probed for a set of named weaknesses by name.

Why it matters

A certificate that has expired or does not match the name makes browsers warn your visitors, and most of them will leave. Obsolete protocols and broken ciphers are worse and quieter: the padlock still appears, while somebody positioned between your visitor and your server can read or alter what passes.

How to read the grade

The grade follows the most serious problem found, not an average. Genuinely broken encryption is an F here — a padlock that means nothing is not a C. A clean setup that also tells browsers to refuse plain HTTP for at least six months earns A+.

Email Security Check

What we checked

We read the public DNS records that govern who is allowed to send email using this domain name — SPF, DKIM and DMARC — along with the mail servers it publishes. Nothing was sent and no mailbox was touched.

Why it matters

These three records are what stops anyone on the internet putting your domain in the From line of an email. Without them, a message that appears to come from your company can be delivered to your customers, your staff or your suppliers, and nothing in the message will look wrong to them. This is how most invoice fraud begins.

How to read the grade

Each record contributes points, weighted by how much protection it actually provides — a DMARC policy set to take no action scores far below one that rejects. The grade explanation names the heaviest deductions.

HTTP Security Headers

What we checked

We requested one page over HTTPS and read the security instructions the server sent back with it. These headers tell a browser what it may and may not do with your site — whether to allow it inside a frame, where scripts may be loaded from, whether to guess at file types.

Why it matters

Without them a browser falls back to permissive defaults. That is what makes clickjacking possible (your page loaded invisibly inside somebody else's), and it removes the last line of defence against a script injected into your page. It also shows when a server volunteers its exact software version, which tells an attacker which known flaws to try first.

How to read the grade

The grade is driven by how many of the six core protection headers are present and correct. The additional headers and the information-disclosure checks are reported but do not by themselves move the letter.

Report generated 17 October 2026, 19:36 UTC for harbourline.example.