ScanSentry

Security Assessment Report
Prepared by Brodie Bulman · getscansentry.com

Target	bobsbakery.example
Client	Bob's Bakery
Label	Security Health Check — Sample Report
Scan date	2026-08-11 03:04 UTC
Prepared by	Brodie Bulman — ScanSentry
Overall grade	F (A = strong posture, F = immediate action required)
Authorization	Confirmed by operator prior to scan

Executive Summary

This assessment identified **16** findings: **4** critical, **5** high, **3** medium, **2** low, **2** info. Findings are listed below in order of severity, each with plain-English remediation guidance.

CRITICAL	HIGH	MEDIUM	LOW	INFO
4	5	3	2	2

Findings & Remediation

CRITICAL Exposed Git repository metadata

exposure

Why it matters: Source code and history may be downloadable, leaking secrets and logic.
Evidence: `GET https://bobsbakery.example/.git/HEAD returned HTTP 200 (21 bytes).`
Remediation: Block access to all dotfiles at the web server; never deploy the .git directory to the public web root.

CRITICAL Exposed environment file

exposure

Why it matters: Environment files usually contain database passwords, API keys, and secrets.
Evidence: `GET https://bobsbakery.example/.env returned HTTP 200 (162 bytes).`
Remediation: Remove the file from the web root immediately, deny access via server config, and rotate every credential it contained.

CRITICAL AWS access key leaked in JavaScript bundle

secrets

Why it matters: Secrets shipped to the browser are public to anyone who views page source. Leaked keys are harvested by automated bots within minutes.
Evidence: `https://bobsbakery.example/wp-includes/js/site.js contains: AKIA…MPLE (redacted)`
Remediation: Rotate the exposed key IMMEDIATELY, remove it from client-side code, and move the operation behind a server-side API.

CRITICAL Redis allows unauthenticated access (CONFIRMED)

service

Why it matters: The Redis server answered a CONFIG command without any authentication. An attacker can read all data and, in many configurations, achieve remote code execution.

Evidence: `Sent 'CONFIG GET dir' to the server on port 6379 — full configuration returned without credentials.`
Remediation: Set requirepass, bind to localhost only, enable protected-mode, and firewall the port. Rotate any credentials stored in it.

HIGH — Database dump at predictable path

exposure

Why it matters: SQL dumps contain full application data — customers, orders, credentials.
Evidence: `GET https://bobsbakery.example/backup.sql returned HTTP 200 (1.4 MB).`
Remediation: Remove the dump from the web root, store backups outside public directories, and assume its contents are compromised.

HIGH — Open port 23 (Telnet)

port

Why it matters: Telnet is cleartext and obsolete — credentials cross the network readable by anyone listening.
Evidence: `TCP/23 open. Banner: bakery login:`
Remediation: Disable Telnet entirely; use SSH instead.

HIGH — Open port 3306 (MySQL)

port

Why it matters: Database service exposed to the network — a prime target for brute-force and exploit attempts.
Evidence: `TCP/3306 open. Banner: 5.5.5-10.4.32-MariaDB`
Remediation: Bind MySQL to localhost or a private network and firewall port 3306 from the internet.

HIGH — SMTP server is an open relay (CONFIRMED)

service

Why it matters: The mail server accepted mail from an external sender to an external recipient without authentication — spammers actively hunt for these.
Evidence: `RCPT TO:<external-recipient> accepted with '250 Ok' and no authentication.`
Remediation: Require authentication for relaying and restrict relay permissions to local domains only.

HIGH — SSL certificate validation failed

ssl

Why it matters: The certificate is expired, self-signed, or does not match the hostname.
Evidence: `certificate has expired (expired 41 days ago)`
Remediation: Install a valid certificate from a trusted CA (e.g. Let's Encrypt) and automate renewal so it never lapses again.

MEDIUM — CMS detected: WordPress 5.8

cms

Why it matters: WordPress 5.8 appears to be an outdated release. End-of-life CMS versions receive no security patches and are actively exploited.
Evidence: `Detected WordPress version 5.8 (meta generator tag)`
Remediation: Update WordPress core, themes, and plugins; enable automatic security updates; protect wp-login with MFA and rate limiting.

MEDIUM — phpinfo() page exposed

exposure

Why it matters: Leaks full PHP and server configuration to attackers.
Evidence: `GET https://bobsbakery.example/phpinfo.php returned HTTP 200 (118 bytes).`
Remediation: Remove phpinfo() and other diagnostic pages from production sites.

MEDIUM — Missing security header: Content-Security-Policy

headers

Why it matters: CSP restricts what scripts and resources can load, mitigating XSS and injection attacks.
Evidence: `GET https://bobsbakery.example — header 'Content-Security-Policy' not present in response.`

Remediation: Configure the web server or application to send a Content-Security-Policy header on all responses.

LOW **WordPress login page reachable** *exposure*

Why it matters: Login pages attract brute-force attacks; enable MFA/rate limiting.
Evidence: `GET https://bobsbakery.example/wp-login.php returned HTTP 200 (91 bytes).`
Remediation: Add MFA, rate limiting, and consider moving or renaming the login endpoint.

LOW **Version disclosure via 'server' header** *fingerprint*

Why it matters: Revealing server versions helps attackers match known vulnerabilities.
Evidence: `server: Apache/2.4.49 (Ubuntu)`
Remediation: Set 'ServerTokens Prod' in Apache to stop advertising the version.

INFO **robots.txt present** *exposure*

Why it matters: Not a vulnerability, but it can disclose hidden paths worth reviewing manually.
Evidence: `GET https://bobsbakery.example/robots.txt returned HTTP 200 (49 bytes).`
Remediation: Review the file for paths that should not be advertised.

INFO **TLS cipher configuration acceptable** *tls*

Why it matters: No weak cipher suites or deprecated protocol versions were accepted.
Evidence: `Probed 28 cipher suites and TLS 1.0/1.1; only modern suites accepted.`
Remediation: No action required. Re-audit after TLS configuration changes.

Evidence Snapshots

Pages captured exactly as an unauthenticated visitor sees them.

https://bobsbakery.example/.git/HEAD

```
ref: refs/heads/main
```

https://bobsbakery.example/.git/config

```
[core]
        repositoryformatversion = 0
        bare = false
[remote "origin"]
        url = git@github.com:bobsbakery/site.git
```

https://bobsbakery.example/.env

```
DB_HOST=127.0.0.1
DB_USER=bakery
DB_PASSWORD=supersecret123
AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```