Security standards

Exponential against the OWASP Top 10 (2021), OWASP ASVS 4.0.3, Content Security Policy Level 3 and the OWASP cheat sheets, with the settings, classes and tests behind each mitigation.

Source file: `doc/eng-us/standards/security.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/standards/security.md

This page is for developers and administrators who need to relate Exponential 6.0.x to the security standards an auditor or a customer asks about. It maps the OWASP Top 10 to the defenses that ship with the kernel, states how far the OWASP Application Security Verification Standard is met, explains the Content Security Policy support, and lists which OWASP cheat sheet recommendations are reflected in code and settings. For the hands-on hardening steps, read Security hardening and chapter 13 of the installation guide.

Summary

Standard	Version	Status	Evidence (short)
OWASP Top 10	2021	**Partly complies**	mitigations for all ten categories; gaps in account lockout default, dependency scanning in CI, public-site CSP
OWASP Application Security Verification Standard (ASVS)	4.0.3	**Unverified**	no ASVS assessment made; many level 1 controls present (see below)
Content Security Policy	Level 3 (W3C Working Draft)	**Partly complies**	kernel policy with per-request nonces, `report-uri` and Reporting API, `csp/report` collector; report-only by default, off on sites
OWASP Cheat Sheet Series	current	**Partly complies**	CSRF tokens, password storage, session cookies, security headers, output encoding, upload validation
Coordinated vulnerability disclosure	ISO/IEC 29147 (practice)	**Partly complies**	`SECURITY.md` with a private reporting channel and supported versions; no `security.txt` (RFC 9116) shipped

OWASP Top 10 (2021)

What it is

The OWASP Top 10 is the Open Worldwide Application Security Project's awareness document of the ten most critical web application security risk categories. The current edition is 2021.

How Exponential complies

ID	Category	Mitigations in Exponential	Status

ID	Category	Mitigations in Exponential	Status
A01	Broken Access Control	Roles and policies with limitations checked by the kernel for every module view (`kernel/classes/ezrole.php`, `eZUser::hasAccessTo()`); content read access per node; admin links shown only when the policy allows (Admin links follow permissions); downloads only through `content/download`, which checks read access; the rewrite rules refuse PHP files below the document root; safe redirects (`eZRedirectManager`, `[SiteSettings] AllowedRedirectHosts`, Safe redirects); setup wizard locked on an installed site (`expSetupWizardGuard`)	Partly complies
A02	Cryptographic Failures	Passwords hashed with PHP's `password_hash()` (`[UserSettings] HashType=php_default`, bcrypt today) and upgraded from older hash types at sign-in (`UpdateHash=true`); REST/OAuth tokens and activation hashes from `random_bytes()` (findings F-01, F-02); session cookie `Secure` over HTTPS (`[Session] CookieSecure=auto`); HSTS available (`SecurityHeaders[Strict-Transport-Security]`, Velocity `[HTTPSSettings] HSTSMaxAge`)	Partly complies
A03	Injection	Template output escaped with the `wash` operator (`htmlspecialchars` with `ENT_QUOTES`); SQL built with `eZDBInterface::escapeString()` and integer casts, reviewed in the 6.0.13 hardening (SEC-01 to SEC-04); shell arguments through `escapeshellarg()` (SEC-05, SEC-06, F-05); CRLF removed from outgoing header values (F-03); datatype input checks (Datatype and input hardening)	Partly complies
A04	Insecure Design	Secure defaults since September 2026 (Security defaults); login that fails closed for accounts without settings (F-06); `ManagedByDeployment` to lock settings, templates and packages in production (Configuration managed by the deployment); attacker-oriented guide (Security hardening)	Partly complies
A05	Security Misconfiguration	Security headers on every page (`[HTTPHeaderSettings] SecurityHeaders[]`); debug output limited to listed addresses; installer files and initial password handling; **Setup > Upgrade check** compares files with `share/filelist.md5` (File consistency check)	Partly complies
A06	Vulnerable and Outdated Components	Bundled jQuery replaced by jQuery 4.0.0 and YUI removed (jQuery 4 and YUI removal); supported versions in `SECURITY.md`; `composer audit` recommended in the hardening guide	Partly complies
A07	Identification and Authentication Failures	Minimum password length 10 (`[UserSettings] MinPasswordLength`); same cost for unknown logins and wrong passwords; uniform "forgot password" answers; session id regenerated at sign-in; other sessions ended on password change (`[PasswordSettings] EndOtherSessions=enabled`); optional lockout (`MaxNumberOfFailedLogin`); two-factor sign-in with `sevenx_authentication_2fa` (Two-step sign-in)	Partly complies
A08	Software and Data Integrity Failures	Immutable release tags; `share/filelist.md5` manifest checked in CI (`.github/workflows/quality.yml`, "File manifest") and in the admin; CSRF tokens on every form (`ezformtoken`); SVG uploads sanitized (`expSvgSanitizer`); audit records hash-chained with signed manifests	Partly complies
A09	Security Logging and Monitoring Failures	Tamper-evident audit trail, on by default (`audit.ini [AuditSettings] Audit=enabled`) with alert rules such as `brute_force` (Audit trail); request classifier writing probes and attacks to `var/log/security.log` (Request logs); CSP violation reports in `var/log/csp.log`	Complies

ID	Category	Mitigations in Exponential	Status
A10	Server-Side Request Forgery	The online editor's "upload from URL" accepts only `http`/`https`, resolves the host once, refuses loopback, private, link-local, multicast and reserved addresses, and re-checks every redirect (TinyMCE 8); the link checker (`kernel/classes/explinkcheck.php`) tests private hosts only when listed in `AllowedPrivateHosts`	Partly complies

Status

Partly complies. Every category has built-in mitigations, and the kernel has received three documented hardening passes (6.0.13, August 2026, September 2026 defaults). "Complies" would need an external assessment, which has not been made.

Known gaps

- **Account lockout is off by default**: `settings/site.ini [UserSettings] MaxNumberOfFailedLogin=0`. Brute force is detected by the audit alert rules, not throttled by the kernel. Rate limiting is left to the web server or a request filter such as request-shield (Security hardening).
- **Two-factor sign-in is opt-in** (an extension and a login handler to configure).
- **No prepared statements** in the database layer: `lib/ezdb/classes/` escapes values (`escapeString()`) rather than binding parameters. Code in extensions must follow the same discipline; the kernel cannot enforce it.
- **Output escaping is not automatic**: a template must apply `|wash`; tests such as `tests/tests/design/AdminStoredValueEscapingTest.php` and `AdminTreeMenuEscapingTest.php` hold specific places.
- **Older password hash types** (`md5_password`, `md5_user`, `md5_site`, `mysql`, `plaintext`) can still be read for migrated accounts until they sign in again and `UpdateHash` replaces them.
- **No dependency vulnerability scan in CI**: the workflows in `.github/workflows/` run PHPStan, PHPUnit, syntax, code style and manifest checks, but no `composer audit`, Dependabot or similar.
- **HSTS** is empty by default in `site.ini` (`SecurityHeaders[Strict-Transport-Security]=`) and has to be set by the operator once the whole site is HTTPS-only.

OWASP ASVS 4.0.3

What it is

The OWASP Application Security Verification Standard 4.0.3 (October 2021) is a catalog of testable security requirements in three levels: level 1 for all software, level 2 for applications that handle sensitive data, level 3 for the most critical ones. (OWASP has since published ASVS 5.0; this page uses 4.0.3 as asked by most current questionnaires.)

How Exponential complies

No requirement-by-requirement assessment has been made. Controls that map to ASVS chapters and exist in the code:

ASVS chapter	Present in Exponential
V2 Authentication	`password_hash()` storage (V2.4), minimum length 10 (V2.1.1 asks 12), optional TOTP 2FA (V2.8)
V3 Session management	`HttpOnly`, `SameSite=Lax`, `Secure` over HTTPS (V3.4); new session id at sign-in (V3.2.1); sessions ended on password change (V3.3.3)
V4 Access control	deny by default through policies; module views need a policy unless listed in `[RoleSettings] PolicyOmitList[]`
V5 Validation, sanitization and encoding	`wash` operator; image uploads checked by content (`expImageUploadValidator`, `image.ini [ImageUploadSettings] AllowedMIMETypes[]`); SVG sanitizer
V7 Error handling and logging	audit trail without passwords or tokens; log viewer masks e-mail and IP addresses and secrets
V12 Files and resources	no PHP execution below the document root by the shipped rewrite rules; downloads through `content/download`
V13 API	REST tokens from `random_bytes()`
V14 Configuration	security headers, CSP, debug output restricted, file manifest

Status

Unverified. The default minimum password length (10) is below ASVS 4.0.3 V2.1.1 (12), so level 1 is not met without changing `MinPasswordLength`. Levels 2 and 3 have not been assessed.

Content Security Policy Level 3

What it is

Content Security Policy Level 3 is a W3C Working Draft (Web Application Security Working Group) that lets a site tell the browser which sources of scripts, styles, images and connections a page may use, with nonces, hashes, `strict-dynamic` and the Reporting API.

How Exponential complies

- The policy is built by `kernel/private/classes/expcontentsecuritypolicy.php` from `settings/site.ini [ContentSecurityPolicy]`: `Mode` (`disabled`, `report-only`, `enforce`, `auto`), `Directives[]` (default `script-src 'self'`, `style-src 'self'`, `object-src 'none'`, `base-uri 'self'`, `form-action 'self'`, `frame-ancestors 'self'`).
- **Nonces**: `Nonce=enabled` and `NonceDirectives[]=script-src` add a nonce of 16 random bytes per request; templates read it with the `csp_nonce()` operator (`kernel/common/expcontentsecuritypolicyoperator.php`). The nonce is not added where a directive allows `'unsafe-inline'`.
- **Reporting**: the `csp/report` view accepts `application/csp-report` and `application/reports+json` and writes one line per violation to `var/log/csp.log`, listed by the log viewer. `ReportEndpoint=enabled` points `report-uri` and `Reporting-Endpoints` at it; `ReportURI` names an external collector.
- **Templates without inline code**: `tests/tests/design/AdminContentSecurityPolicyTest.php` fails on inline `<script>`, event handler attributes, `javascript:` URLs and `style=""` in the admin4 and admin4l

templates; `tests/tests/design/AdminStylingRulesTest.php` forbids inline CSS in every admin design.

- Details: Content Security Policy.

Status

Partly complies. The mechanism follows CSP Level 3. As shipped, `Mode=auto` sends a **report-only** policy on administration siteaccesses and **no policy** on public sites; the classic designs listed in `ReportOnlyDesigns[]` stay report-only even when enforced.

Known gaps

- Enforcement is a per-siteaccess decision of the operator.
- Public designs and rich text output (`style=""` from content, embedded players) are not checked by the test.
- `strict-dynamic` and hash sources are not configured by default.

OWASP cheat sheets reflected in the code

Cheat sheet	What Exponential does	Where
Cross-Site Request Forgery Prevention	Synchronizer token per session on every form, checked on every POST; refused tokens get the "form expired" page; SameSite=Lax as defense in depth	`extension/ezformtoken`, activated by the setup wizard (`kernel/setup/steps/ezstep_create_sites.php`); Form expired page
Password Storage	`password_hash()` with PHP's default algorithm (bcrypt), verification with `password_verify()`, rehash on sign-in	`kernel/classes/datatypes/ezuser/ezuser.php` (`PASSWORD_HASH_PHP_DEFAULT`), `[UserSettings] HashType`, `UpdateHash`
Session Management	`HttpOnly` (default true), `SameSite` (default Lax), `Secure` (auto), new id at sign-in, idle timeout	`settings/site.ini [Session]` (`CookieHttponly`, `CookieSameSite`, `CookieSecure`, `SessionTimeout`, `ActivityTimeout`), `lib/ezsession/classes/ezsession.php`
Cross Site Scripting Prevention	Context-aware escaping with `wash` (`xhtml`, `email`, `javascript` types), rich text stored as XML and rendered through templates, CSP	`lib/eztemplate/classes/eztemplatestringoperator.php`, Content Security Policy
SQL Injection Prevention	Escaping and integer casts, persistent objects instead of hand-written SQL	`lib/ezdb/classes/`, `eZPersistentObject`; 6.0.13 hardening
File Upload	Type read from the file content with fileinfo, allow-list of MIME types, SVG sanitized, no execution below the root	`image.ini [ImageUploadSettings]`, SVG images
HTTP Security Response Headers	`X-Content-Type-Options: nosniff`, `X-Frame-Options: SAMEORIGIN`, `Content-Security-Policy: frame-ancestors 'self'`, `Referrer-Policy: strict-origin-when-cross-origin`, `Permissions-Policy: camera=(), microphone=(), payment=(), usb=()`	`settings/site.ini [HTTPHeaderSettings] SecurityHeaders[]`, `settings/velocity.ini [ServerSettings] ResponseHeaders[]`

Cheat sheet	What Exponential does	Where
Clickjacking Defense	`X-Frame-Options: SAMEORIGIN` and `frame-ancestors 'self'` on every page, the administration included	same as above
Authentication	uniform error answers, MFA available, optional lockout with `TrustedIPList[]`	`[UserSettings]`, `sevenx_authentication_2fa`
Logging	security events recorded, secrets never written, addresses truncated	Audit trail, Log viewer
Unvalidated Redirects and Forwards	external targets refused unless allowed	`eZRedirectManager`, Safe redirects

Vulnerability disclosure

`SECURITY.md` at the repository root names the supported versions (the 6.0.x line, fixed on `main` and released as the next tag) and asks for private reports by e-mail, not in the public issue tracker; see also Reporting a vulnerability. Published release tags are never changed: a fix arrives as the next version.

Gap: no `/.well-known/security.txt` (RFC 9116) is shipped; an operator can add one.

Tests and CI

- `phpunit.xml` testsuite `security` (`tests/tests/kernel/classes/security/`: `eZSecurityHardeningTest.php`, `eZRSSSecurityTest.php`, `expFileModeLimitsTest.php`, `expIniPhpWrapperTest.php`); the PHPUnit workflow fails if the suite is missing from `phpunit.xml`.
- Audit trail tests in `tests/tests/kernel/classes/audit/`.
- CSP and escaping tests in `tests/tests/design/`.
- PHPStan with a baseline (`.github/workflows/quality.yml`).

Related pages

- Accessibility standards, Privacy standards, Web platform standards
- Security hardening
- Security and audit guide
- Velocity HTTP/2 and security

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.