Security for hosts: overview

The security section of the hosting help: what Exponential protects by default, what a host must add on the server, and one page per topic, from hardening, TLS and the Content Security Policy to isolation between...

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

This section is for the security lead, the system administrators and the product managers of a hosting company that runs Exponential sites for customers. It answers two questions: what does Exponential already do by itself, and what is the host's part of the work on the server, in the panel and in the contract.

> **Warning**
>
> `.htaccess` **slows down every request.** When `.htaccess` files are enabled (`AllowOverride` anything but `None`), let alone used, Apache looks for them and reads and evaluates them on every request, in every directory of the path, and page loading times go up drastically (measured). The only real fix is to switch the feature off for the virtual host: `AllowOverride None` in the HTTPS virtual host, and the rules in the virtual host's own configuration as mod_rewrite rules. That is far faster in production speed tests and well worth the time it takes to optimize a production configuration. Switch over right away: copy the rules of the example HTTPS virtual host into your virtual host (with the static cache, also `staticcache_offload.conf`), set `AllowOverride None`, run `apachectl configtest`, reload Apache and test the site. The same rules for other servers: nginx, LiteSpeed, Caddy, IIS, Velocity (built in); where virtual host rules go in a control panel: Plesk, cPanel, DirectAdmin, ISPConfig. **On LiteSpeed or OpenLiteSpeed,** the server's own page cache can also answer anonymous visitors and crawlers without PHP: Page caching with LiteSpeed and OpenLiteSpeed.

Exponential 6 is built to be secure by default: a fresh installation serves only a fixed list of files, sends security headers, keeps debug output off, protects forms with tokens, hashes passwords with bcrypt, keeps a tamper-evident audit trail and refuses to run its setup wizard on an installed site. The reasoning behind each default is on Security hardening and the setting-by-setting reference is chapter 13 of the installation book. The pages here take the host's view: many sites, many customers, one team.

Back to Hosting companies: start here.

Contents

1. Pages in this section
2. Who does what
3. Where to start
4. Related pages

Pages in this section

Page	What it answers
Hardening for hosts	the hardening a host applies to every Exponential site, layer by layer
TLS and HTTPS	HTTPS on every name, HSTS, secure cookies, and proxies that terminate TLS
Content Security Policy	the shipped report-only policy, the reports, and moving a siteaccess to enforce
request-shield and request rules	the firewall in front of the kernel, the kernel's request classifier and its request rules
Restricting the administration	its own host name, network limits, least privilege and `ManagedByDeployment`
First sign-in and default passwords	what each installer sets as the administrator password, and what a host does after provisioning
Demo mode for hosts	running a public demo with `EXP_DEMO_MODE=1`, and keeping it away from every production site
Two-step sign-in	the sevenx_authentication_2fa extension: turning it on, requiring it, lost phones
Audit trail	what is recorded, verifying the hash chain, alerts, sinks and keys
Security updates and disclosure	supported versions, following releases, updating, and how to report a vulnerability
Multi-tenant isolation	how far one customer's site is kept from another's, and which setup gives which boundary
Per-customer users and pools	one operating-system user, one PHP-FPM pool or Velocity instance, one database user per customer
Compliance and GDPR	what the software provides for data protection and e-mail law, and what stays with host and customer
Log and data retention	how long each log and record is kept, and the settings that change it

Who does what

Security is shared between the software, the host and the customer. A clear split avoids gaps that each side assumes the other covers.

Area	Exponential provides	The host provides	The customer decides
Served files	the rewrite rules of `.htaccess_root` and Velocity's list; `exp:setup:htaccess` to copy them and to show drift	a web server configuration that keeps them	nothing, unless they bring their own rules
`config.php`	`exp:config:*` with masked secrets, backups and a `php -l` check	mode `0640`, out of version control, `var/backup/` protected and cleaned	the values their site needs
Transport	HSTS and certificate handling in Velocity; secure cookies	certificates, renewal, redirects, the TLS terminator	which host names exist
Accounts	bcrypt hashes, password rules, the weak password check, two-step sign-in	a provisioning flow that never leaves a known password	who gets which role

Area	Exponential provides	The host provides	The customer decides
Isolation	nothing between installations: that is the operating system's job	one user, one pool, one database user per customer	nothing
Logging	`security.log`, `auth.log`, `csp.log`, the audit trail with hash chains	disk space, forwarding off the machine, retention in backups	retention values that their privacy notice states
Updates	immutable releases, changelogs, the upgrade check	a routine that applies them	approval of a maintenance window, where the contract asks for it

> **Note**
>
> This split is a starting point for your own service description, not a statement of what any contract must say. Plans, SLAs and support tiers are options a host chooses; see The hosting business.

Where to start

For a host that is setting up its first Exponential offer:

1. Read Hardening for hosts and turn it into your server template.
2. Decide the isolation model with Multi-tenant isolation and Per-customer users and pools.
3. Make provisioning safe: First sign-in and default passwords.
4. Make watching possible: Audit trail and the Security log.
5. Write the update routine: Security updates and disclosure.
6. Before go-live and every quarter, run the Security review checklist.

When something has already gone wrong, the Compromised site runbook is the page to open.

Related pages

- Hosting companies: start here
- Security hardening
- Chapter 13: Security hardening for production
- Security review checklist
- Compromised site
- Setting up .htaccess and config.php from the console
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and

> land new features faster.
>
> Partners: start here | How to apply

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.