request-shield and request rules

The three request layers a host can use with Exponential: request-shield, a firewall that decides before the kernel starts; the kernel's request classifier, which only classifies into security.log; and the kernel's...

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

This page is for system administrators and security staff at a host. Three layers look at an Exponential request before or while the kernel handles it, and they are often confused. This page says what each does, where it runs and how a host switches it on. The facts about request-shield come from Security hardening: A shield in front; the classifier from Request logs; the request rules from Request rules.

Contents

1. The three layers
2. request-shield
3. The request classifier
4. Request rules
5. Offering it to customers
6. Checklist

The three layers

	request-shield	Request classifier	Request rules
What it does	**decides**: refuses, slows down, challenges, bans for a while, or only logs (monitor mode)	**classifies** only: probes, traversal, injection, attack tools, bursts of 404s	**decides** per module view: redirect, not found, access denied, sign in first, rewrite, log
Where it runs	before the kernel starts	inside the kernel, at the end of every request	inside the kernel, just before a module view runs
Part of Exponential	no; added by the host or site (MIT license, by the CJW Network)	yes, on by default	yes; three rules ship defined and off
Settings	its own rule files	`settings/requestclassifier.ini`	`settings/requestrules.ini`
Its log	its own log, and a live view under `/rs/`	`var/log/security.log`	as a rule's action decides

A request the shield refuses never reaches the kernel, so it is not in `security.log`. In the shield's monitor mode the same probe can be in both logs. The classifier's probe and injection rule names follow request-shield's names in lower case, so a line in one can be found in the

other.

request-shield

request-shield (source: cjw-network/request-shield) looks at every request before the kernel starts. Real visitors pass in microseconds; a refused request costs about 30 µs instead of a rendered page.

Installing

It is not part of Exponential's `composer.json`. Either:

Shell

```
composer require cjw-network/request-shield
```

or a checkout or release of the library in a directory **outside** the document root. Then copy the Exponential example rules from the library's `examples/exponential/` directory to a directory the web server never hands out (the library suggests `settings/request-shield/`), and choose the main file:

File	When
`exponential.rules`	the frontend rules; included by one of the two below
`exponential-admin-uri.rules`	the administration is the siteaccess `/admin` on the same host
`exponential-admin-host.rules`	the administration has a host of its own

Switching it on

The shield must run at the start of **every** request:

- **Apache or nginx with PHP-FPM**: a `Shield::protectFile()` call at the top of `config.php`, or PHP's `auto_prepend_file` pointing to the library's `bootstrap.php`. With `auto_prepend_file` in the pool configuration a host can switch it on per customer without touching the customer's files.
- **Velocity**: `config.php` runs once per worker, not per request. Since 6.0.15, `index.php` runs the file that `config.php` names in the constant `EXP_REQUEST_PREPEND_FILE` at the start of every request; `config.php-RECOMMENDED` shows the pattern. Velocity answers cached pages before a worker starts, so the shield sees only what reaches a worker.

Then, in this order:

1. Start in **monitor mode** (`set mode monitor`, as the examples ship) and read its log for a few days while editors work.
2. Adapt the site-specific lines: URI siteaccesses (`EXP-MODULES`), extension modules that take forms (`EXP-POST`), host names and load balancer (`host`, `trust`).
3. Check after every change:

   Shell

   ```
   bin/request-shield check <main file>
   bin/request-shield test <main file>
   bin/request-shield trace <main file> "GET https://www.example.com/user/login"
   ```

4. Switch to `set mode enforce`.
5. To switch it off, use `set mode off` rather than removing the include; under Velocity a persistent worker would otherwise keep the last request's marks.

The example rules hide system URLs and administration modules on the public site, refuse internal paths, put a browser check before the sign-in and the administration, and ban an address for an hour after ten refusals in ten minutes (verified search engines are never banned). The shield cannot tell a right password from a wrong one, and does not see static files the web server serves itself: account lockout stays Exponential's job (`MaxNumberOfFailedLogin`, two-step sign-in).

The request classifier

The kernel classifies every request once and writes what looks like an attack to `var/log/security.log` with category, rule and confidence, whatever the status. Rule sets: `probes`, `traversal`, `injection`, `scanner`, `threshold` (`Max404PerMinute=30` within `WindowSeconds=60`).

- It **never blocks**. Blocking belongs to the shield, the web server or a listener class registered in `[ClassifierSettings] Listeners[]` (none by default).
- Addresses are masked, the user agent is hashed and shortened; query values are kept in `security.log` as evidence, with secrets, e-mail and IP addresses masked.
- `Classification=disabled` switches it off; leave a rule set out of `RuleSets[]` to switch one off.

Reading and forwarding it: Security log and Request logs.

Request rules

The request rules decide just before a module view runs: *when* (typed as a system URL, full view, the user lacks a policy, below a node, from a network) and *what* (redirect to the URL alias, not found, access denied, sign in first, redirect, rewrite, log). Exemptions are policies, so they are granted in the role editor. Nothing changes until a rule is listed in `RuleList[]`.

The common case, keeping anonymous visitors off `/content/view/full/<node>` while the URL alias keeps working, is one shipped rule switched on per public siteaccess:

INI

```ini
# settings/siteaccess/<public siteaccess>/requestrules.ini.append.php
[RequestRuleSettings]
RuleList[]
RuleList[]=system_url_full_view_to_alias
```

Leave the administration and editor siteaccesses alone: editors use system URLs there. Check and explain:

Shell

```shell
php bin/php/ezrequestrules.php -s <siteaccess> --list  --allow-root-user
php bin/php/ezrequestrules.php -s <siteaccess> --check --allow-root-user   # exit code 1 on probl
ems, for CI
php bin/php/ezrequestrules.php -s <siteaccess> --uri=content/view/full/2 --allow-root-user
```

> **Warning**
> Rules restrict *how* allowed content may be reached; they are not a replacement for `content/read` limitations. An `ip` condition is only as trustworthy as the visitor address: behind a proxy, configure `TrustedProxies[]` and `ClientIpByCustomHTTPHeader` first (TLS and HTTPS).

Offering it to customers

These are options a host can choose:

- request-shield in monitor mode on every new site, with enforcement after the first week, as part of a managed plan;
- the administration behind the shield's browser check or limited to the customer's network (`EXP-ADMIN-NET`);
- the system-URL rule on public siteaccesses as a default of the server template;
- a weekly look at `security.log` and the shield's log, with a report to the customer when a rule rises sharply.

Checklist

☐ request-shield installed outside the document root, rules in a directory never served.

☐ Hooked into every request: `config.php` or `auto_prepend_file` (PHP-FPM), `EXP_REQUEST_PREPEND_FILE` (Velocity).

☐ Monitor mode first; `check` and `test` pass; then `enforce`.

☐ `security.log` read weekly; the classifier left on.

☐ Request rules checked with `--check` on every siteaccess that uses them.

☐ Proxies trusted correctly before any address-based rule is relied on.

Related pages

- Security for hosts: overview
- Security hardening
- Restricting the administration
- Security log
- HTTP cache: With request-shield in front
- 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.