

ISPProtect Web Security for Contao

ISPProtect Web Security for Contao: Manual

Version 1.0.0

2026-11-23

Contents

1. **Introduction**
   - What the bundle does
   - What the bundle does not do
   - Free and paid parts
   - Requirements
   - Who can use it
   - How this manual is arranged
2. **Installation and first steps**
   - Install the bundle
   - Create the database tables
   - Install the assets
   - Open the pages
   - First steps
   - Behind a reverse proxy or CDN
   - Scheduling with cron
3. **Overview**
   - Menu entry and tabs
   - Dashboard
   - If the tables are missing
4. **Firewall**
   - How the firewall decides
   - Firewall modes
   - What a refused visitor sees
   - Bans
   - The administrator’s own address
   - Address lists
   - Custom rules
   - Trap paths and paths without request limit
   - Protection against locking yourself out
   - If the firewall itself fails
   - Limits of the firewall
   - OWASP CRS rules
5. **File scanner**
   - What a scan looks at
   - Which files are scanned
   - Starting and stopping a scan
   - Scheduled scans
   - The list of findings
   - What you can do with a finding
   - Details of a finding
   - Known false alarms
   - Vulnerabilities of installed packages
   - Limits of the scanner
6. **Remote scan and licence**
   - Modes
   - What leaves the server
   - Use the local scan service
   - Use your own scan service

- Use the hosted scan service with a licence
- Test connection
- Result of the last scan
- When the service refuses or is not available
- How credentials are stored
- Limits of the remote scan

7. **Sign-in protection**
   - What it counts
   - Details
   - What it does not cover
   - The tab Sign-in protection
   - Lift a lock of a user name
   - Lift a lock of an address
   - Shortened addresses in the log
8. **Two-factor authentication**
   - Require the second step for administrators
   - What the setting does not do
   - After a lost device
9. **Security checkup**
   - Score and status
   - The checks
   - Listed files and the permission fix
   - Security headers
10. **Audit log and notifications**
    - The audit log
    - Notifications
11. **Settings reference**
    - Changing settings in the back end
    - Changing settings on the console
    - Value types
    - Firewall group
    - Request rate limit group
    - Missing pages group
    - Trap paths group
    - Custom rules group
    - Bans group
    - Sign-in protection group
    - Two-factor authentication group
    - Audit log group
    - Notifications group
    - Scanner group
    - Remote scan group
    - Security headers group
    - OWASP CRS rules group
12. **Command line**
    - List of commands
    - Status and checkup
    - Scan and findings
    - Audit log
    - Addresses
    - Sign-in locks

- Settings
- Maintenance
- Remote scan commands
- Removal

1.3. Getting back in and troubleshooting

- You are banned or blocked by the firewall
- You are locked by the sign-in protection
- A trap path or a rule blocks the wrong requests
- Two-factor authentication is required and an administrator cannot use it
- The pages show tables are missing
- The pages of the bundle have no layout
- All visitors have the same address
- Scans or mails do not happen
- The remote scan or the licence does not work
- Something else
- The OWASP CRS rules refuse a normal request

1.4. Privacy and external services

- Contact with outside services
- Data in your database

1.5. Removing the bundle

- Delete the data
- Remove the code

Introduction

ISPProtect Web Security is a security bundle for Contao. You manage it on pages in the Contao back end and with console commands. The pages are available in English and German.

What the bundle does

- **Request firewall.** It looks at every request before Contao routes it. It works with address lists, request rates, trap paths and rules that you write yourself. See Firewall.
- **Sign-in protection.** It counts failed sign-ins in the back end and the front end, bans addresses and locks user names. See Sign-in protection.
- **Two-factor authentication.** It can require administrators to set up the two-factor authentication that Contao already has. See Two-factor authentication.
- **File scanner.** It does basic checks of the files of your installation: changes, PHP in upload folders, leftovers and a few generic suspicious-code checks. The remote scan is the thorough one. It only reads files. It never deletes, moves, repairs or quarantines a file. See File scanner.
- **Remote scan.** Optionally, a scan service checks the code files by their checksums. See Remote scan and licence.
- **Security checkup.** 20 checks give a score from 0 to 100. See Security checkup.
- **Audit log and notifications.** Security-relevant events are recorded, and serious events are sent to you by e-mail. See Audit log and notifications.
- **Security headers.** Three response headers can be set or removed. See Security headers.

What the bundle does not do

- The firewall has no list of attack patterns. It does not look for SQL injection or cross-site scripting in requests, and it is not a web application firewall. It decides by address, request rate, path and your own rules.
- The local scanner does basic checks and finds changes. It is no guarantee that a site is clean.
- The bundle does not repair or remove infected files. You do that yourself.

Free and paid parts

Everything in the bundle works without a licence key, with one exception: the remote scan in the mode **Hosted scan service (licence)** needs a licence key. The remote scan modes that use a scan service on your own server or a scan service that you run yourself need no licence key from this bundle. See Remote scan and licence.

Requirements

- PHP 8.3 or newer
- Contao 5.7 (`contao/core-bundle` ^5.7)
- The database that Contao uses

The package name is `ispprotect/contao-web-security-bundle`. The bundle is published under the LGPL-3.0-or-later licence.

Who can use it

Only back end users with the administrator role can open the pages of the bundle. The menu entry is not shown to other users, and every page refuses them. Every change on the pages needs a valid Contao request token. If the token is missing or expired, nothing is changed and the page asks you to reload and try again.

How this manual is arranged

- Installation and first steps to set the bundle up.
- The chapters from Overview to Audit log and notifications describe one function each.
- Settings reference lists every setting, and Command line every console command.
- Getting back in and troubleshooting helps when you are locked out.
- Privacy and external services names every outside service the bundle can contact.
- Removing the bundle describes the removal.

In this manual, console commands start with `vendor/bin/contao-console`. Run them in the project directory of your Contao installation, with the PHP binary of the site.

Installation and first steps

Install the bundle

Install the bundle with Composer:

```
composer require ispprotect/contao-web-security-bundle
```

You can also install it with the Contao Manager: search for `ispprotect/contao-web-security-bundle`. The bundle registers itself with the Contao Manager, so you need no further configuration.

If you received the bundle as a zip file, upload the file on the Packages page of the Contao Manager with “Upload Packages”, confirm the upload with “Confirm Uploads” and install it there. With Composer alone, put the file into a folder that `composer.json` lists as a repository of the type `artifact`, then run the `composer require` command above.

Create the database tables

Create the tables of the bundle with the Contao migration:

```
vendor/bin/contao-console contao:migrate
```

In the Contao Manager, run the database update instead. The bundle stores its settings, address lists, event log, scan results and counters in tables whose names start with `ispprotect_`.

Until you have run the migration, the pages of the bundle show this notice: “The ISPProtect database tables do not exist yet. Run “contao:migrate” on the command line or in the Contao Manager.” The firewall, the sign-in protection and the audit log do nothing in that time. They do not break Contao.

Install the assets

The pages load a stylesheet and a script from the public assets of the bundle. Contao’s install scripts run `assets:install` after `composer require`. If the pages look unstyled, run it yourself:

```
vendor/bin/contao-console assets:install
```

Open the pages

Sign in to the back end as an administrator. You find the pages under **System > ISPProtect Web Security**. Other users do not see this entry. See Overview for the tabs.

First steps

Work through these steps in this order on a new installation.

1. Open the tab **Checkup** and work through the warnings. The score shows how many checks pass. See Security checkup.
2. Open the tab **Scanner** and click **Start scan**. A scan reads every file once. The first complete scan becomes the baseline, the recorded state of your files that later scans compare against. It reports no changed or added files, because there is nothing to compare with yet. Look at the content findings of this first scan, because they do not depend on the baseline. See File scanner.

3. Check the firewall mode. The default is **On**: the firewall refuses requests that break a rule. If you want to see what it would do first, set **Firewall mode** to **Log only** on the tab **Firewall**. Then it records the requests it would refuse and refuses none. After a few days, look at the tab **Events**. If only requests that you want to stop show up, set the mode to **On**. See Firewall.
4. Enter an e-mail address in **Notification address** on the tab **Settings**, or check that Contao's administrator address is set. Without a receiver, no alert or summary mail is sent. See Audit log and notifications.
5. If the site is behind a reverse proxy or a CDN, set `TRUSTED_PROXIES`. See Behind a reverse proxy or CDN.
6. Set up the system cron. See Scheduling with cron.

Behind a reverse proxy or CDN

The bundle uses the client address that Contao sees. Contao takes it from headers such as `X-Forwarded-For` only if the request comes from a trusted proxy. If your site runs behind a reverse proxy, a load balancer or a CDN, list the addresses of these proxies, and only these, in the environment variable `TRUSTED_PROXIES`. Write them as a comma-separated list, for example in `.env.local`.

What goes wrong otherwise:

- If a proxy is missing from the list, every visitor has the address of the proxy. The rate limit, the bans and the sign-in locks then hit all visitors together: one visitor who triggers a ban blocks everybody.
- If the list trusts everybody (an entry ending in `/0`, or `REMOTE_ADDR` on a site with public visitors), any visitor can choose the address that the firewall, the sign-in protection and the event log see. That visitor can avoid bans, or have the address of somebody else banned.

The checkup has a check for this: it fails if the list trusts every address, and warns if a request carried a forwarding header although it did not come from a trusted proxy. On the command line it can only judge the configured list.

Scheduling with cron

The scan and the maintenance work (alert mails, removing expired data, the summary mail) advance only when Contao's cron runs. The bundle registers a job that runs every minute.

Set up the system cron so that Contao's command runs every minute:

```
* * * * * /usr/bin/php /path/to/project/vendor/bin/contao-console contao:cron
```

Use the PHP binary and the path of your server. With the system cron, each run may scan for up to 20 seconds. A database lock keeps two runs from overlapping. A failing step is logged and tried again in the next minute.

Without a system cron, Contao runs its scheduler after page views (the web cron). This works, but a run may scan only for 5 seconds, and nothing runs while nobody visits the site. A scan then advances in short slices and takes much longer. Use the system cron.

You can run the same work by hand with `ispprotect:maintenance`, or run a scan on the console with `ispprotect:scan`. See Command line.

Overview

Menu entry and tabs

The bundle adds the entry **ISPProtect Web Security** to the menu **System** of the Contao back end. Only administrators see it. The pages are under `/contao/ispprotect`, with your configured back end prefix. They have nine tabs:

Tab	What it shows
Dashboard	The state at a glance.
Firewall	Address lists, bans and the firewall settings. See Firewall.
OWASP CRS rules	Install, update and remove the OWASP CRS rules, disable a rule and the CRS settings. See OWASP CRS rules.
Sign-in protection	Locks, sign-in events and the sign-in settings. See Sign-in protection.
Scanner	Start and stop a scan, the findings, the remote scan section. See File scanner.
Events	The audit log with filters. See Audit log and notifications.
Checkup	All check results. See Security checkup.
Settings	General settings: security headers, audit log, notifications, scanner and remote scan.
Manual	This manual, with a link to download it as a PDF.

Each settings group has its own place:

Tab	Settings groups
Firewall	Firewall, Request rate limit, Missing pages, Trap paths, Custom rules, Bans
OWASP CRS rules	OWASP CRS rules
Sign-in protection	Sign-in protection, Two-factor authentication
Settings	Security headers, Audit log, Notifications, Scanner, Remote scan

The groups are listed with all their settings in Settings reference. Each group has a button **Reset** followed by the name of the group below the form that sets the group back to its defaults, after you confirm. The values you typed in the form above are not saved by it.

Dashboard

The dashboard shows:

- **Security checkup**: the score and the most important results. The link **All results** leads to the tab **Checkup**. When nothing needs attention, it says “No open problems.”
- **Firewall**: the firewall mode, and a hint when the mode is **Off** (“The firewall is switched off. Requests are neither checked nor refused.”) or **Log only** (“The firewall only logs. Requests that break a rule are not refused yet.”).
- **Active bans**: how many bans are active.
- **Open scan findings**: the open findings by level.
- **Last scan**: when the last scan finished, or “No scan has finished yet.” A button **Start scan** starts one.
- **Events**: the number of events in the last 24 hours and the last 7 days, and the newest events.

If the tables are missing

Before you run `contao:migrate`, the pages show a notice about the missing tables. See Installation and first steps.

Firewall

The firewall looks at every request before Contao routes it, starts a session or loads a page. The settings are on the tab **Firewall** and listed in Settings reference.

How the firewall decides

The firewall decides in this order:

1. Mode **Off**: the request passes.
2. An address on the allow list: the request passes.
3. An address on the block list or the ban list: the request is refused with status 403.
4. A request to a trap path: the address is banned and the request is refused with status 403.
5. Your custom rules, in order. The first matching rule with the action `block` refuses the request with status 403. A rule with the action `log` only records the request.
6. The request rate limit: an address that sends more requests than allowed in the time window gets status 429. The sign-in paths of the back end (`/contao/login` and `/contao/login-link`, with your configured back end prefix) have a stricter limit of their own. After the number of refusals set in **Ban after refusals** within 10 minutes, the address is banned.

After the response, the firewall counts “not found” answers (status 404) per address. An address that causes too many of them in the time window is banned.

The firewall itself works with addresses, request rates, paths and the rules you write yourself. The search for attack patterns such as SQL injection or cross-site scripting is a separate part, the OWASP CRS rules. See OWASP CRS rules.

Firewall modes

The setting **Firewall mode** has three values:

Value	Meaning
On	Requests that break a rule are refused. This is the default.
Log only	The firewall decides and records, but never refuses. Use it to see what the firewall would do.
Off	The firewall does nothing.

What a refused visitor sees

A refused request gets a small page without external resources, in English or German. The response is marked `no-store` and carries a `Retry-After` header.

Bans

A ban is a block of an address that ends after some time. The first ban of an address lasts as many minutes as **Ban length (minutes)** says. Each repeated ban of the same address within 30 days lasts four times as long, up to **Longest ban (minutes)**.

- IPv6 addresses are counted and banned per /64 range.
- Loopback addresses (for example 127.0.0.1) cannot be blocked or banned.
- Request counters live in the bundle’s cache pool. They are approximate when many requests arrive at the same time.
- Requests to `/_fragment`, `/_wdt` and `/_profiler` are not counted by the rate limit. You can add more paths in **Paths without request limit**.

The administrator's own address

When an administrator signs in, that address (IPv6: its /64 range) is protected for 24 hours, as long as **Protect administrator addresses** is on. A protected address skips only the automatic measures:

- Bans do not apply to it, also bans made before the protection began.
- It is not checked against the trap paths, the rate limit and the "not found" count.

The block list and your custom rules apply to it like to any other address. Only the allow list lets an address skip every step. Too many failed sign-ins from a protected address still lock the sign-in from it for a while. See Sign-in protection.

If you turn **Protect administrator addresses** off, the page warns you that your address can be banned like any other and that this can lock you out of the back end.

Address lists

The page shows "Your address as seen by this server". Check this address first when the site is behind a proxy: if it is not your own address, see Behind a reverse proxy or CDN.

The **Address list** shows every entry with its kind (**Allowed**, **Blocked** or **Banned**), the address or range, the source, the reason, the creation time and the end time. The source says where the entry came from: **Manual**, **Sign-in protection**, **Rate limit**, **Missing pages**, **Trap path** or **Rule**. An entry of your own address is marked "(your address)".

Under **Add an entry** you add an address or range, for example `203.0.113.7` or `203.0.113.0/24`:

1. Choose the kind: **Allowed** or **Blocked**. Bans are made by the firewall itself. You can add a ban on the console with `ispprotect:ip:add`.
2. Type the address or range.
3. Optionally type a lifetime in minutes. Without a lifetime the entry has no end. The longest lifetime is 5256000 minutes.
4. Optionally type a reason, up to 190 characters.
5. Click **Add**.

If the entry would block your own address, the page refuses it until you tick "I know that this range may include my own address and lock me out." A blocked range must not include the local loopback address.

The button **Remove** deletes an allowed or blocked entry. The button **Lift ban** ends a ban. Changes of the lists are recorded in the audit log.

Custom rules

A custom rule tests one part of the request with one comparison, then blocks or logs.

Part	Meaning
`path`	The path of the request
`query`	The query string
`user_agent`	The browser identification
`referer`	The referrer
`method`	The HTTP method
`host`	The host name

The comparisons are `equals`, `prefix`, `suffix`, `contains` and `regex`. A regular expression is written without delimiters and matches case-insensitively. A rule value must not be empty and can be up to 500 bytes. You can define up to 100 rules.

On the page, the field **Custom rules** holds the rules as a JSON list. On the console, `rules.custom` holds the same list. Each rule has these keys:

Key	Meaning
`id`	Optional. Letters, digits, `-` and `_`, up to 32 characters. Without an id, the bundle makes one
`field`	One of the parts in the table above
`op`	One of the comparisons above
`value`	The text to compare with
`action`	`block` or `log`
`enabled`	Optional. `true` to use the rule (the default), `false` to switch it off

Example:

```
vendor/bin/contao-console ispprotect:settings:set rules.custom '[{"id":"no-env","field":"path","op":"suffix","v
alue":".env","action":"block","enabled":true}]'
```

Trap paths and paths without request limit

Trap paths and **Paths without request limit** are lists of paths, one per line. Entries are compared in lower case. An entry matches the whole path, or the start of the path when it ends with `*`: `/backup/*` matches `/backup` and `/backup/x`, but not `/backups`. An entry that would cover the whole site is refused.

Protection against locking yourself out

So that nobody locks themselves out of the back end:

- A trap path that matches the back end (`/contao`, with your back end prefix) or its sign-in paths is refused when you save it. This holds on the pages and for `ispprotect:settings:set`.
- An enabled custom rule with the action `block` that matches a sign-in request is refused in the same way. A sign-in request here is one of the two sign-in paths, with `GET` or `POST`, without a query string, at the host of the website (the domain of the first published website root, else the configured router host). So rules on `path`, `query`, `method` and `host` are checked. A rule on `host` is not checked while the host of the website is unknown, and only that one domain is checked when the site has several website roots. Rules on `user_agent` and `referer` depend on the visitor and are not refused.
- After a save on the tab **Firewall**, the page warns if an enabled block rule matches your own sign-in request (your address and browser, the host you use and the host of the website).

The check runs when a setting is saved. If you change the back end prefix (`contao.backend.route_prefix`) later, stored entries are not checked again until the next save. If you are locked out anyway, see Getting back in and troubleshooting.

If the firewall itself fails

If the database or the cache fails, the firewall lets the request through and logs the problem, at most once in five minutes. It does not lock everybody out because of its own fault.

Limits of the firewall

- A page that Contao's HTTP cache answers does not reach the kernel, so it does not reach the firewall. Rate limits and bans do not apply to cache hits. To stop an address from fetching cached pages, you need a rule in your proxy or web server.

- The Contao Manager (`contao-manager.phar.php`) is a separate application. The firewall, the sign-in protection and the audit log do not cover it.

OWASP CRS rules

The OWASP Core Rule Set (CRS) is a widely used, openly maintained set of rules that recognises typical attacks on web applications, for example SQL injection, cross-site scripting, attempts to read files or to run commands, and requests of scanners and attack tools. In the back end they have their own tab, **OWASP CRS rules**, between **Firewall** and **Sign-in protection**. The CRS rules are the part of the firewall that looks for attack patterns; the bundle adds only four rules of its own, for requests the CRS rules do not recognise (see ISPProtect rules). The rules run with an engine of the plugin written in PHP, so no extra program or server module is needed. They are off until you install them.

The tab **OWASP CRS rules** shows the state, the mode, the version, the SHA-256 of the download, the time of the installation, the numbers of rules and the rules left out. It has the buttons **Install rules**, **Update rules**, **Remove rules** and **Licence text**, the form **Disable a rule** and the settings of the CRS rules (see Settings reference).

Installing the rules

1. Open the tab **OWASP CRS rules** and click **Install rules**.
2. The plugin downloads one fixed version of the rules from GitHub (the CRS project, version 4.25.2 in this release of the plugin) over `https`. Your server needs outgoing `https` access to `github.com` and to the download host that GitHub redirects to, and PHP needs `allow_url_fopen` and OpenSSL. The plugin compares the download with a checksum (SHA-256) that is built into the plugin and refuses anything else. A newer version of the rules arrives only with a newer version of the plugin; nothing else is ever downloaded or run.
3. The first time you install, the plugin sets the mode (`waf.mode`) to **Log only**.

The page then shows the version, the checksum, the time of the installation, the number of rules in use and the numbers of the rules that are not supported (see What the CRS rules do not cover). After a bundle update the pinned version of the rules may change. After a bundle update, **Update rules** brings the rules to the version that belongs to the new bundle. **Remove rules** deletes them again. The rules are stored in `var/ispprotect/waf`, outside the public folder.

Start with logging

Leave the mode on **Log only** first. In this mode a request that matches is only written to the audit log and is never refused. Look at the entries for a few days, in which all kinds of pages, forms and the back end are used, disable the rules that report normal requests of your site (see below) and only then set the mode to **Block**.

Value	Effect
Off (`off`)	Requests are not checked with the CRS rules. This is the state until you install the rules.
Log only (`log`)	A request that reaches the threshold is recorded as **Request logged by the firewall** and goes through.
Block (`block`)	A request that reaches the threshold is refused with status 403 and recorded as **Request refused by the firewall**.

The CRS check has its own mode (`waf.mode`) and does not depend on the firewall mode. It runs after the address lists, bans and limits of the firewall: a request from an address on the allow list is never checked with the CRS rules. Everybody else is checked, signed-in administrators too.

Reading an entry

A request that reached the threshold is recorded in the audit log with the event **Request logged by the firewall** or **Request refused by the firewall**. The details hold:

- the numbers and the messages of the rules that matched, for example `942190` and the name of the attack it suspects,
- for each rule the name of the part of the request that matched (for example `ARGS:comment`, `REQUEST_COOKIES:name` or `REQUEST_URI`) and a short excerpt of the value with the sensitive part masked,

- the total score, the threshold, the paranoia level, the method, the path and the answer (never the query string, the headers or the body as such).

A request counts as an attack when the scores of all rules that matched add up to the **Anomaly threshold**. Every rule adds a score by its severity: a critical match adds 5, an error 4, a warning 3 and a notice 2. With the default threshold 5, one critical match is enough; with a higher value, several rules have to match.

If the CRS rules themselves have a problem (for example the compiled rules cannot be read), the request goes through and the event **Problem with the CRS rules** is recorded, at most once an hour per kind of problem.

A rule reports a normal request

If a rule reports requests that are normal for your site (a false positive), disable that rule:

1. Open the entry in the audit log and note the rule number and the message.
2. On the tab **OWASP CRS rules**, enter the number in the field **Rule id** of the form **Disable a rule** and click **Disable rule**.

You can also enter rule numbers by hand under **Disabled rules** (`waf.disabled_rules`), one per line. A disabled rule no longer runs for any request, also not for the ones it was right about; disable only the rule that is wrong, not a whole group. If you disabled a rule by mistake, remove its number from the list.

Paranoia level and threshold

The **Paranoia level** (`waf.paranoia`, 1 to 4, default 1) decides how many rules run. Level 1 contains the rules with very few false positives and is recommended for most sites. Each higher level adds stricter rules that find more attacks and also report more normal requests, mainly in forms with free text, in editors and in page content. Raise the level only after the entries at the level before are clean, one step at a time, and start with logging again.

The **Anomaly threshold** (`waf.threshold`, 1 to 100, default 5) is the score at which a request counts as an attack. Raise it only if you accept that single rule matches no longer count.

Requests that cannot be fully inspected

Some requests cannot be checked completely: the body is larger than the engine reads, the request has more fields than it looks at, the body cannot be read, or the check takes too long. An attacker could use such a request to get past the rules, so by default the plugin treats it like an attack: in mode **Block** it is refused, in mode **Log only** it is recorded. The entry shows one of these numbers:

Number	Meaning	Answer when refused
`200002`	The body could not be read (for example invalid JSON or XML).	400
`200004`	The body is a form upload without its closing boundary.	400
`200005`	A pattern of a rule ran into the regular expression limits of PHP.	403
`200007`	More fields, headers, cookies or parts than the engine inspects (1000).	403
`200100`	The time limit for the request was used up before all rules ran.	403
`200101`	The body is larger than **Inspected body size**.	413
`200102`	A value is longer than the engine inspects (65536 bytes).	403

You can disable these numbers like any other rule. Two settings are for the two usual cases:

- **Inspected body size** (`waf.body_limit`, 1024 to 1048576 bytes, default 131072 = 128 KiB). A larger request cannot be fully inspected. A save with a large amount of content can be bigger than that and is then answered with status 413 and the number `200101`. If large saves fail in mode **Block**, raise the value, for example to 524288, or let such requests pass (see below).
- **Time limit per request** (`waf.time_budget_ms`, 5 to 10000 milliseconds, default 200). The time the rules may take for one request. A normal page or save needs 2 to 10 milliseconds, a save of 42 KB of content about 100 milliseconds on the server where it was measured. A request near the body limit at paranoia level 2 can reach the default. If requests

stop with `200100`, raise the value. Keep it as low as your site allows: the limit is also what an attacker can make your server spend on one request.

- **Program code in a form field.** Program code pasted into a form field (indented PHP or Python, about 20 to 40 KB) can use up the default time limit of 200 milliseconds, and the request is refused with `200100` in block mode. For a site where program code is pasted into forms, raise the time limit, switch off `200100`, or let requests pass that cannot be fully inspected. Normal text, HTML and the editor saves that the exclusions cover are not affected.
- **Pattern limits.** Rarely, a long field with program code makes a pattern of a CRS rule exceed the regular expression limits of PHP, and the request is refused with `200005`. The answers are the same: switch off `200005` or let requests pass that cannot be fully inspected.

Let requests pass that cannot be fully inspected (`waf.pass_uninspected`, default off) lets such requests through (they are still recorded). Big saves keep working then, but a crafted request that is too large or too slow is no longer checked completely. Use it only when you cannot raise the limits, or disable the single number that causes the trouble, for example `200100`.

An XML request body that declares an entity or names an external DTD is refused with `200002` (HTTP 400) in mode **Block** even when **Let requests pass that cannot be fully inspected** is on. Downloaded CRS exclusions cannot remove `200002` or the missing-multipart-boundary rule `200004`; you can still disable either number explicitly in the settings.

Exclusions for Contao

The bundle comes with its own exclusion rules in `rules/contao-exclusions.conf`. They are loaded before the CRS rules and have the numbers 10000 to 10999. Each one is limited to a path or a form and to named fields, and none of them switches off a whole kind of attack. They remove false positives that normal Contao work caused at paranoia levels 1 and 2:

- the cookie `sf_redirect` of the Symfony web profiler (only the `dev` environment reads it),
- the field `_target_path` of the sign-in forms, as long as it holds plain base64,
- the rich text, HTML and code fields of content elements and the teaser texts of articles, news and events,
- the meta description and the page title in the page settings,
- the lists of allowed HTML tags and attributes in the system settings,
- the `source` field of the source editor and the template editor,
- the text settings of the pages of ISPProtect Web Security (trap paths, custom rules, header values), on their save routes only.

The CRS rules run before Contao checks the session, so an unauthenticated request can name these forms too. Contao itself refuses such a request, so the skipped fields are only processed for signed-in back end users. All other fields, headers and cookies are still checked.

> The path of the back end is your back end prefix (`contao.backend.route_prefix`, default `/contao`). The bundle puts the prefix into the file when it loads it, so a changed prefix is followed. A site in a subdirectory (paths such as `/sub/contao`) or a prefix that is not a plain path gets none of the back end exclusions. You then see more entries, but the protection is not weaker.

ISPProtect rules

The bundle ships four rules of its own in `rules/ispprotect-contao.conf`, with the numbers 12000 to 12999. They report requests that the CRS rules do not recognise at paranoia level 1. They run with the CRS rules, so only while the CRS rules are installed and the mode is **Log only** or **Block**: after the CRS rules, before the rules of your own file and before the decision on the threshold. Each match adds a critical score (5), so with the default threshold one match is enough. The tab **OWASP CRS rules** shows how many of them are in use in the line **ISPProtect rules in use**.

Rule	Reports
`12001`	The version control folders of Bazaar (`.bzr/`) and Darcs (`_darcs/`) and the files `CVS/Entries`, `CVS/Root` and `CVS/Repository`. The CRS rules report `.git/`, `.svn/`, `.hg/` and `.env` themselves.

Rule	Reports
`12002`	Copies of PHP and YAML files (also `.phtml`, `.phar`, `.inc`) with the ending `.orig` or `.rej`, a number (`config.yml.1`) or a numbered `.bak`, `.old` or `.save` (`index.php.bak1`), and the Emacs files `#name#` and `.#name`. The CRS rules report `.bak`, `.old`, `.save`, `.swp` and `~` themselves.
`12200`	A PHP script in the upload folder (`contao.upload_path`, default `files`), also `name.php.jpg` and `name.php/more`. The bundle puts the folder into the rule when it loads it, so a changed upload folder is followed. A site in a subdirectory or an upload folder that is not a plain path gets no match.
`12201`	The configuration file `localconfig.php` of older Contao versions, and copies of it. Contao 5 keeps it outside the public folder, so a request for it is always a search for its passwords.

When the rules help:

- The rules see only requests that reach Contao's front controller (`public/index.php`). With the `.htaccess` file of Contao, a request for a file that does not exist in the public folder goes to Contao, which answers "not found". The rules refuse such a request and record it with their number, so a search for leftover files shows in the log as what it is. Some servers answer a missing `.php` file themselves (nginx with `try_files $uri =404`, Apache with `ProxyPassMatch`); such a request never reaches Contao.
- A file that exists in the public folder is sent by the web server without Contao, and a PHP script that exists is run by PHP without Contao. The rules never see these requests and cannot stop them, also not a real copy `index.php.orig` or a script in the upload folder.

The real fix is the configuration of the web server and a clean public folder: the server refuses version control folders and copies of files and runs no PHP in the upload folder. The check **Public leftovers** of the Security checkup lists backups and debugging files it finds in the public folder.

The rules are disabled like CRS rules: one rule by its number (the form **Disable a rule**, or **Disabled rules** in the settings), all four with the line `SecRuleRemoveByTag ispprotect` in your own rules file (see below). The tags `ispprotect/source-disclosure` (`12001`, `12002`, `12201`) and `ispprotect/upload-script` (`12200`) disable a group. A bundle update can bring changed rules; the bundle compiles the rules again on the next request.

Your own CRS rules file

You can add your own rules in SecLang, the language of the CRS, in the file `var/ispprotect/rules/own.conf` of your installation. The plugin never creates the file. The page shows whether it is in use. Rules in it:

- need a number from 11000 to 11999; a rule with another number is skipped,
- run after the CRS rules and before the decision on the threshold, so a rule can add to the score,
- use the subset of SecLang that the engine supports: `SecRule`, `SecAction`, `SecMarker`, `SecRuleRemoveById`, `SecRuleRemoveByTag`, `SecRuleUpdateTargetById` and `SecRuleUpdateTargetByTag`, and chains. Rules that need something else are skipped and counted as not supported.

An example that adds a critical score for a PHP tag in a parameter:

```
SecRule ARGS "@rx (?i)<\?php" \
    "id:11001,phase:2,block,t:none,t:urlDecodeUni,\
    msg:'PHP open tag in a parameter',severity:'CRITICAL',\
    setvar:'tx.inbound_anomaly_score_pl1=+%{tx.critical_anomaly_score}'"
```

The plugin notices a changed file on the next request and compiles the rules again. If they cannot be compiled, the CRS check does not run, requests go through and the problem is recorded as **Problem with the CRS rules**.

What the CRS rules do not cover

- The plugin checks requests only. Rules for responses (the CRS "response" phases) are not run.
- Rules that need the libinjection library are not run, because PHP has no such library: `941100` and `941101` (XSS) and `942100` and `942101` (SQL injection). The other CRS rules for these attacks do run. The page lists the rules that are not supported.

- The part headers of a form upload (`MULTIPART_PART_HEADERS`) are not available, because PHP reads the upload before the plugin sees it. The file names and sizes are checked.
- The CRS check only looks at the request. It does not scan files and does not replace updates.

Cost of the check

On a normal request the check costs 2 to 10 milliseconds. A save with 42 KB of content took about 100 milliseconds. At paranoia level 2 and above, and for large bodies, the check takes longer; the time limit stops it at the value you set. Program code pasted into a form field (about 20 to 40 KB of indented PHP or Python) can use up the default time limit of 200 milliseconds; normal text, HTML and the editor saves that the exclusions cover do not.

If the CRS rules lock you out

Switch the CRS check off on the console:

```
vendor/bin/contao-console ispprotect:settings:set waf.mode off
```

Then sign in again, find the rule that caused the trouble in the audit log, disable it and set the mode back to **Log only**. See also Getting back in and troubleshooting.

Licence and attribution

Uses the OWASP® Core Rule Set (CRS) 4.25.2, © Trustwave Holdings, Inc. and the OWASP CRS project contributors, licensed under the Apache License 2.0. The rules are downloaded directly from the CRS project on GitHub. OWASP® is a registered trademark of the OWASP Foundation; ISPProtect is not affiliated with or endorsed by OWASP.

The page shows the same text. The licence text of the CRS is part of the download; the button **Licence text** shows it.

File scanner

The scanner does basic checks of the files of your installation: changes, PHP in upload folders, leftovers and a few generic suspicious-code checks. The remote scan is the thorough one. It only reads files. It never deletes, moves, repairs or quarantines a file. You work with it on the tab **Scanner**. Its settings are in the groups Scanner and Remote scan, see Settings reference.

What a scan looks at

A scan produces findings of these kinds. The names are the ones shown in the column **Kind**.

Kind	Level	What it means
A file was changed since the last scan	Medium	A code file differs from the baseline.
A new file appeared	Medium	A code file is not in the baseline.
A file is gone	Low	A file of the baseline does not exist any more.
Suspicious content in a file	Medium or High	The basic content checks found a suspicious construct. They are generic and do not recognise known malware (use the remote scan for that). They search PHP code (including `.html5` and Twig templates and a few other template file types), `.htaccess` files and script code.
A PHP file in an upload directory	High	A PHP file in the upload directory, or PHP code hidden in a media file.
A file that should not be in the public directory	Medium	For example backup and debugging files.
An installed package has a known vulnerability	Low, Medium or High, by the severity of the advisory	Only if **Check for known vulnerabilities** is on.
The scan service recognised malware in a file	High	Only with the remote scan, see Remote scan and licence.

The baseline is the recorded state of your code files: the scanner stores a hash of each file and compares later scans with it. Files of packages that Composer changed since the last scan are not reported as changed. The first complete scan becomes the baseline and reports no changed, added or removed files.

The levels are **Low**, **Medium** and **High**.

Which files are scanned

- The project directory is checked against the baseline and for content.
- These generated directories and the upload directory are checked for content only: `assets`, `public/assets`, `public/bundles`, `public/share`, `system/cache`, `system/tmp` and `var`. Contao and Composer rewrite these on their own, so comparing them with the baseline would produce a flood of “changed” findings.
- The scanner skips `var/cache`, `var/log`, `var/logs`, `.git`, `node_modules`, symbolic links and the paths in **Excluded paths**.
- Files larger than **Largest scanned file (KB)** are not read completely.

Starting and stopping a scan

Click **Start scan** on the tab **Scanner**. A scan runs in slices of at most 200 files and saves its position after each slice, so a scan that is interrupted goes on where it stopped. While you keep the page open, it advances the scan in short steps and shows the state, the phase, the number of files and scripts, and the number of new findings. Click **Abort scan** to stop it.

Without JavaScript the progress is not shown live. The scan then advances with the Contao cron job. See Scheduling with cron.

On the console, `ispprotect:scan` does the same, see Command line.

Scheduled scans

File scan schedule can be **Off**, **Daily** or **Weekly**. A scheduled scan starts when the hour in **File scan hour** (in UTC) has passed and no scan has started since the start of the current day (daily) or in the last 7 days (weekly). Contao's cron drives the scan. Without a working cron, no scheduled scan starts.

The list of findings

The list below the remote scan section shows the findings. You can filter it by **Status** (**Open**, **Ignored**, **Fixed**), **Minimum level**, **Kind** and a search text, with the button **Filter**. **Reset filter** clears the filter. The columns are **Level**, **Kind**, **Path**, **Signature**, **First seen** and **Action**. A finding that a later scan no longer reports becomes **Fixed**.

What you can do with a finding

- **Ignore** hides a finding until the file changes again. **Stop ignoring** brings it back.
- **Accept** is offered for changed, added and removed files. It makes the current state of the file the new baseline. Use it after you have checked the change. If the file changed after the scan or cannot be read, the bundle refuses and asks you to scan again.
- **Accept all** does this for every changed, added and removed file. The link above the list shows how many findings it covers. A confirmation page tells you how many open and ignored findings are included. You must tick the confirmation box and click **Accept all**. If the findings changed in the meantime, nothing is accepted and you confirm again. Files that changed after the scan or cannot be read stay open. Files are never changed or deleted.

Findings of the other kinds cannot be accepted. Fix the cause, then scan again, or ignore the finding.

Malware that the scan service reports cannot be accepted as a new baseline, so **Accept all** does not cover it.

Details of a finding

Details in the column **Action** opens the details of a finding: level, kind, path, signature, line, status, and when it was first and last seen. Under **Lines of the file** you see the file as it is now, at most 4 lines before and after the line of the finding. The line of the finding is marked. When the finding names no line, the first lines of the file are shown. The page only reads the file and changes nothing. Only administrators can open it.

Instead of the lines, the page shows a note when:

- the file was removed or moved after the scan,
- the file can hold passwords or keys, for example `.env` files, `.htaccess`, `system/config/localconfig.php`, key files, database dumps and the folder `var`. These are the files that the remote scan never sends or sends only in redacted form, see Remote scan and licence. Backups and editor copies of such files count as well, for example `.env.bak` or `.env.swp`,
- the file is an archive or a database dump (for example `.sql`, `.sql.gz`, `.zip`, `.tar.gz`), or the finding is a file that should not be in the public directory (a backup or copy): their content is never shown,

- the path is a symbolic link, is not a regular file, has more than one hard link or leads out of the project directory,
- the file is larger than 5 MB.

A note also says when the file was changed after the finding was last seen, so the lines may differ from what the scan saw, and when a line was cut after 500 characters. Characters that cannot be shown safely (control characters, invisible characters and characters that change the direction of the text) appear as `?`.

Known false alarms

The scanner can report files of well-known packages although they are harmless, for example `vendor/league/uri/Uri.php`, `vendor/doctrine/inflector/src/Inflector.php` and `vendor/contao/core-bundle/src/Translation/Translator.php`. You can ignore them. Paths that you never want scanned go into **Excluded paths**.

Vulnerabilities of installed packages

If **Check for known vulnerabilities** is on, the scanner asks packagist.org for known security advisories of the installed packages. Only package names leave the server; the versions are compared on your server. See Privacy and external services.

Limits of the scanner

- The local scanner does basic checks and finds changes. It is no guarantee that a site is clean.
- It does not repair or remove anything.

Remote scan and licence

The remote scan is optional and off by default. With it, the scanner also asks a scan service about the code files of the site. The service knows checksums of files that are known to be harmless or malicious, so it can find malware that the local rules do not. Everything else of the scanner works without it, and a scan never depends on it: if the service cannot be reached, the local scan finishes all the same and the scan records that the remote part failed.

Modes

You choose the mode with the setting **Remote scan** on the tab **Settings**. You look after the connection in the section **Remote scan** on the tab **Scanner**. The mode is read when a scan starts.

Mode	Where the files are checked	What you need
Off	Nowhere. Nothing is sent. This is the default.	Nothing.
Local scan service (socket)	A scan service on the same server, reached through a Unix socket. Nothing leaves the server.	The socket path in **Socket of the local scan service**. No access token and no licence key.
Own scan service	A scan service that you run yourself, reached over HTTPS.	Its address in **Address of the scan service** and its access token.
Hosted scan service (licence)	The hosted ISPProtect scan service, reached over HTTPS. This is the one paid part of the bundle.	A licence key.

The default socket is `/run/ispprotect/scand.sock`. The bundle needs no site key for the local scan service of an ISPProtect server licence. Only the hosted service needs a licence key.

What leaves the server

For each non-empty code file with a UTF-8 file name, the bundle sends the SHA-256 checksum and a name hint (the file name, or only `file.<extension>`). It never sends a file name with a path. Only files that the service does not know are uploaded, and only within the limit **Upload limit per scan (MB)** of one scan and the limits of the service. With the value `0` only checksums are sent. Files larger than **Largest scanned file (KB)** are not sent either. In the mode **Local scan service (socket)** all of this stays on the server.

Never sent, although the local scanner still examines these files:

- `.htaccess`, `.user.ini` and `.env*` files;
- files whose name is that of a configuration or credential file, such as `parameters.yml`, `auth.json`, `.htpasswd`, `.npmrc`, `.netrc` and private key files (`id_rsa` and similar), and copies of them (names that start with `wp-config`, `localconfig`, `parameters.y` or `config.local`), except the PHP files named below;
- files with the extension `pem`, `key`, `crt`, `p12`, `pfx`, `jks`, `kdbx`, `sql`, `sqlite`, `db` or `ini`;
- files whose name starts or ends with a space or another blank, such as `localconfig.php`;
- files whose name only looks like a configuration file and that are not PHP files, such as `local_config.yml`, `wp_config.php.bak` or `wpconfig.txt`, also with an invisible or non-Latin character in the name;
- everything in `config/jwt`, `config/secrets`, `contao-manager`, `config`, `system/config`, `var`, `.git` and `.ssh`;
- files in `config/packages` with “secret” in the name;
- the paths in **Paths not sent to the scan service**.

Sent only in redacted form: PHP configuration files named `localconfig.php`, `config.local.php` or `wp-config.php`, copies whose name starts with `wp-config` and ends with `.php`, and PHP files whose name only looks like a configuration file (such as `local_config.php`, `localconfig-sample.php`, `wp_config.php` or `my-wp-config.php`, also with an invisible or non-Latin character, with an extension that PHP runs such as `.php`, `.phtml` or `.inc`), when they are not in one of the folders above. The `localconfig.php` of Contao itself is in `system/config` and so is never sent. Before the checksum is taken and before the upload, all strings, numbers and comments in such a file are replaced: every string by `ISPP_REDACTED` (also keys such as `dbPass` in `$GLOBALS['TL_CONFIG']['dbPass']`), every number by `0`, the text of every comment by `ISPP_REDACTED`, a

variable name inside a string (such as `"pa$word"`) by `$ISPP_REDACTED`, every name inside `{$...}` or `${...}` in a string by `ISPP_REDACTED`. Only the code structure, the names (functions, variables, constants) and the constant names in `define()` stay (in a file with a `namespace` declaration, a `use` import or PHP 8 attributes those names are replaced too), so passwords and other values never leave the server. Code that an attacker added can still be found by its structure: the service sees calls such as `eval()`, `base64_decode()` or `include`, but not their literal contents. The checksum that is looked up is the one of the redacted form; the one of the original file is never sent. The service can still tell an empty string or the leading blanks of a line in a string, but not read any value. A file that cannot be redacted safely (it does not parse, it is larger than 1 MB, it contains `__halt_compiler`, or it uses a short open tag `<?` or starts with `<?xml`, whose meaning depends on the PHP setting `short_open_tag`) is not sent at all. Malware found in the redacted form is reported for the real file, with the note that it was checked in redacted form.

The hosted service keeps a table of checksums with the verdict and the time. It does not keep uploaded files: they are deleted right after the scan, also when they are malware.

Enter only the address of a scan service you trust: the access credentials and the content of unknown files go to the address that is set.

Use the local scan service

1. Set **Remote scan** to **Local scan service (socket)**.
2. If the socket is not at the default path, set **Socket of the local scan service**.
3. Open the tab **Scanner** and click **Test connection**.

Use your own scan service

1. Set **Remote scan** to **Own scan service**.
2. Enter the HTTPS address in **Address of the scan service** and save.
3. Open the tab **Scanner**. In the section **Remote scan**, type the access token in **New access token** and click **Save token**. **Remove token** deletes it again.
4. Click **Test connection**.

On the console, use `isprotect:remote token:set` and `isprotect:remote token:clear`, see Command line. The token must have 1 to 4096 printable characters without spaces.

Use the hosted scan service with a licence

1. Set **Remote scan** to **Hosted scan service (licence)** and save.
2. The hosted scan service at `https://scan.isprotect.com` and the licence API at `https://license.isprotect.com` are built in; you do not enter an address. The setting **Address of the hosted scan service** (`remote.hosted_url`) is an optional override for testing and is normally left empty. Its `remote.portal_url` setting is an override for another portal deployment (console command `isprotect:settings:set`), not the account console at `https://console.isprotect.com`.
3. Open the tab **Scanner**. In the section **Remote scan**, type the key in **Licence key** and click **Connect**. On the console, use `isprotect:remote connect`.
4. Click **Test connection**.

The section then shows:

Field	Meaning
Licence	**Connected** or **Not connected**
Plan	The plan of the licence
Licence valid until	The end of the licence
Sites	How many sites use the licence, and how many it allows

Field	Meaning
Host sent to the licence portal	The host name of this site
Access valid until	The end of the current access of this site to the service
Automatic renewal	**About once a day**, the time of the next try, or **Stopped**
Last renewal	When the access was last renewed
Licence problem	The last reason the portal gave, if any

A connected site is counted at the licence portal. The bundle renews its access to the service by itself, about once a day, and again when the access runs out within two days. It also does this in the mode **Off** while a licence is connected. After a clear refusal by the portal, or when the clock of the server is wrong, the renewal stops. Fix the cause, then click **Check now** (console: `ispprotect:remote refresh`), or disconnect and connect again with a key.

Disconnect asks for confirmation (“Yes, disconnect”). It releases the site at the licence portal and removes the licence data on the server. The data is removed in every case, also when the portal could not be reached. Then the site may still count toward the licence until you release it in your account at the licence portal. Removing the bundle also releases the site, see Removing the bundle.

Test connection

Test connection asks the service for its status and shows the engine, the signatures, the number of signatures loaded, the largest file, the upload per day and the requests per minute that the service reports. It changes nothing.

Result of the last scan

The section shows the remote part of the last scan: its state and these counts.

Count	Meaning
Looked up	Files whose checksum was asked for
Clean	Files the service knows as harmless
Uploaded	Files that were uploaded
Malware	Files the service reported as malware
Skipped	Files that could not be read, or that changed while they were sent
Not examined	Files the service did not examine
Sent	The amount of data uploaded

Malware that the service reports is a finding of the kind “Malware found by the scan service”, with the name of the detection. See File scanner.

When the service refuses or is not available

The local scan always finishes. The scan records the reason, and the section shows it with the state of the last scan. The reasons are:

Reason	Meaning
The scan service could not be reached.	A network or socket problem.
The scan service did not accept the access credentials.	Wrong or expired token or licence.
The scan service is not ready yet.	For example, it is still loading its signatures.

Reason	Meaning
The scan service is busy, or asked to wait.	The scan will try again later.
The quota of the scan service is used up.	The limit of the service is reached.
A file was too large for the scan service.	The service has its own file size limit.
The scan service refused access.	The service does not allow this server user, or the access credentials do not include the remote scan.
The scan service could not scan several files in a row.	The scan will try again later.
The upload limit of this scan was reached.	The remaining files follow in the next scan.
No valid address or socket is set.	Check the settings.
An access token or a connected licence is missing.	Connect first.
The remote scan setting was changed while the scan was running.	The next scan uses the new setting.

Limits of the hosted service: the licence limits the number of sites, and the service has its own limits for the size of a file, uploads per day and requests per minute, which **Test connection** shows.

How credentials are stored

The access token and the licence key are stored encrypted in the database, with a key derived from the application secret of Contao. They are never shown again, never written to the log and never sent back to the browser. The page shows only whether they are set and, for an access token of at least 12 characters, its last four characters. For the licence key the page shows only that one is set. The command `ispprotect:remote` runs on the server and shows the last four characters of both when the value has at least 12 characters.

If the application secret is shorter than 16 bytes or no cipher is installed, secrets cannot be stored, and the page says so. If the application secret changes, the stored token and licence data can no longer be read. The page then says so, and you enter them again. Changes to the token and the licence are recorded in the audit log.

Limits of the remote scan

- The remote scan checks code files by their checksum and only uploads files that the service does not know. It finds what the service knows. It is no guarantee that a site is clean.

Sign-in protection

The sign-in protection watches the back end sign-in and the front end sign-in (members). Its settings are in the groups Sign-in protection and Two-factor authentication, see Settings reference.

What it counts

Wrong passwords, unknown user names and wrong or reused codes of the second step count as failures, per address and per user name. An attempt with an empty user name or an empty password is not counted. Stale form tokens are not counted either.

When a limit is reached within the time window:

- The address is banned (**Failed sign-ins per address**). A protected administrator address is not banned, see The administrator's own address. Instead only the sign-in from it is locked, and the rest of the site stays reachable. The first such lock lasts as many minutes as **Lock length (minutes)** says. If the previous lock of the address began within the last 30 days, the new one lasts four times as long as that one, up to **Longest ban (minutes)**.
- The user name is locked (**Failed sign-ins per user name**) for **Lock length (minutes)**.

Both checks run before the password or the code is checked. A refused attempt shows Contao's generic "too many login attempts" message. It does not tell the visitor whether the address or the name caused it. The reason is in the event log, event "Sign-in refused".

Details

- The second step is counted too, so a stolen password alone does not give unlimited tries at the code.
- A sign-in counts as successful only when it is complete, including the second step. A sign-in from a "remember me" cookie is ignored.
- User names are compared by key. Case, surrounding spaces and invisible characters do not matter, so all spellings of a name count against the same account.
- Contao keeps user names exactly as written, so "john" and "JOHN" (or the full-width "ｊｏｈｎ") can be two accounts in the same area that share one key. If another account's name may share the key (compared without regard to case, width and accents), a sign-in of the account does not clear the failure count of the name and does not make its address a known one for a name lock. It is not recorded as "Successful sign-in" or "Sign-in from a new address", and it does not protect an administrator's address. Otherwise one of these accounts could keep the name of the other from being locked.
- Allowed addresses are never counted, banned or locked out.
- Addresses protected by an administrator sign-in are counted like any other address, so a stolen session at such an address cannot be used for unlimited password guessing. Their bans do not apply to the sign-in, but block list entries do.
- IPv6 addresses are counted, banned, locked and protected per /64 range. Devices change their temporary IPv6 addresses within that range.
- A locked user name does not hold back an address from which that account completed a sign-in within the last 90 days (IPv6: the same /64 range). Otherwise anybody could keep the owner out of the account by typing wrong passwords for the name. The trade-off: someone else at such an address, for example in the same office network, can go on trying while the name is locked. The address limit still applies to them.
- When an administrator signs in from an address not seen for that user in the last 90 days, the bundle records the event "Sign-in from a new address" and, if **Alert on sign-ins from new addresses** is on, sends a mail.
- Sign-out and user switches are recorded in the event log.
- If the tables of the bundle are missing or the database fails, the sign-in goes on as if the bundle were not installed.

What it does not cover

Sign-ins with a login link or a “remember me” cookie, and other sign-ins without a user name and password, are not checked beforehand by the sign-in protection, and neither are passkeys. The firewall still refuses blocked and banned addresses on every request.

The tab Sign-in protection

The tab shows these parts:

- **Locked accounts**: locked user names with **Area** (back end or front end), **User name**, **Created** and **Expires**. The button **Unlock** lifts a lock.
- **Addresses locked for sign-in**: locks of protected administrator addresses, with **Area**, **Created**, **Expires** and **Minutes**. Only a hash of the address is stored, so the page cannot list the address. Each row has a field **Address** and a button **Unlock**.
- **Addresses banned after failed sign-ins**: the bans made by the sign-in protection, with a button **Lift ban**.
- **Recent sign-in events**.
- The settings of the groups Sign-in protection and Two-factor authentication.

Lift a lock of a user name

1. Open the tab **Sign-in protection**.
2. Find the account under **Locked accounts**.
3. Click **Unlock**.

On the console:

```
vendor/bin/contao-console ispprotect:unlock <username>
```

Replace `<username>` with the user name. The command lifts the lock in every area that has one. With `--realm=backend` or `--realm=frontend` it lifts only the one in that area.

Lift a lock of an address

An address that reached the failure limit while it was protected as the address of a signed-in administrator is not banned. It cannot sign in for a while instead. To lift this lock you must type the address.

1. Find the address. It is in the event “Sign-in locked for an address” on the tab **Events**: in the column **Address**, and in the details under `rule`. For IPv6, any address of the /64 range will do.
2. On the tab **Sign-in protection**, find the row under **Addresses locked for sign-in**, type the address into its field and click **Unlock**.

On the console:

```
vendor/bin/contao-console ispprotect:unlock --address=203.0.113.7
```

Give either a user name or `--address`, not both and not neither. The option `--realm=backend` or `--realm=frontend` limits the command to one area.

Lifting the lock also resets the failure count of the address in that area and removes the history that makes the next lock longer. It does not lift a firewall ban of the address. Lift that on the tab **Firewall** or with `ispprotect:ip:remove`. An entry on the allow list lets the address pass the sign-in check while the entry exists, but it does not remove the lock. When the entry is removed before the lock ends, the lock applies again.

When a lock is really lifted, the bundle records the event “User name unlocked” or “Sign-in unlocked for an address”. The event names the user name or the address, not who lifted the lock. Nothing is recorded when there was no lock. These events are not mailed as alerts.

Shortened addresses in the log

If **Shorten addresses in the log** is on, the event holds only the shortened address and no `rule`. You then cannot read the full address from the event. The full address is only in the access log of the web server, or you wait until the lock ends. With this setting on, the shortened addresses are also compared when the bundle checks whether an address is known for an account, so the whole /24 range (IPv4) or /48 range (IPv6) of a known address counts as known.

If you are locked out of the back end yourself, see Getting back in and troubleshooting.

Two-factor authentication

The bundle does not bring its own two-factor authentication. It uses the one that Contao already has. The bundle adds three things:

- The checkup reports active administrators that have not set up two-factor authentication. It counts accounts that are not disabled and inside their start and stop dates. See Security checkup.
- Wrong codes of the second step are counted by the sign-in protection and recorded as the event “Wrong code for the second sign-in step”. See Sign-in protection.
- The setting **Require a second sign-in step** can require two-factor authentication for administrators.

Require the second step for administrators

Require a second sign-in step is on the tab **Sign-in protection**. It is off by default. When you turn it on, administrators must set up a second sign-in step with an authenticator app.

An administrator who has not turned on two-factor authentication is then redirected to the “Security” page of the back end until they have done so. What a user sees:

- Every page of the back end leads to the “Security” page, where the user sets up the second step.
- The sign-in itself, the second sign-in step, the login link, the sign-out and the password change stay open. So does the “Security” page with its set-up form.
- Background requests (XHR, Turbo frames and streams) cannot follow a redirect. They are refused with status 403 and a short text, or JSON if the request asks for it.

The check covers every route of the back end, not only the main back end pages: popups, alerts, picker, confirmation, help, favorites, preview, jobs, search, the template studio and the pages of this bundle. GET and POST requests are redirected (status 303) before the controller runs, so nothing can be changed.

The setting applies to administrators only. Other back end users and front end members are not affected. Users who must change their password first, and sessions in which an administrator has switched to another user, are not redirected, as in Contao’s own option. If the check itself fails, for example because the database cannot be read, the request is let through and the error is written to the log, so that a fault does not lock every administrator out.

What the setting does not do

- It makes administrators set up the second step. It is not a second step for accounts that have none yet. Someone who has the password of such an administrator can still sign in, and can then set up two-factor authentication with a device of their own.
- It does not cover the console, other back end users or front end members.

Contao has its own option, `contao.security.two_factor.enforce_backend`, which covers all back end users. It is set in the Contao configuration. The advice of the checkup for administrators without two-factor authentication names both options.

After a lost device

A user who lost the device with the authenticator app cannot give the right code. Wrong codes count as failures, so the sign-in of that user name or address may be locked for a while. See Getting back in and troubleshooting.

Another administrator can turn off the two-factor authentication of the user in the user form of the Contao back end. The bundle records this as a warning, event “Second sign-in step turned off”. The user then signs in with the password and sets up the second step again; with **Require a second sign-in step** on, the “Security” page opens at once.

When a user turns two-factor authentication on or off in their own account settings, the bundle does not record it.

Security checkup

The checkup runs 20 checks and gives a score from 0 to 100. You find it on the tab **Checkup**, and the score and the main results on the **Dashboard**. On the console, `isprotect:checkup` prints the same.

Score and status

Each check has a weight and a status. The score is the weighted share of passed checks. A warning counts half, a problem counts nothing, and notes are not rated.

Status as shown	Meaning
In order	The check passed.
Needs attention	A warning.
Problem	The check failed.
Not known	The value could not be read. The check is not rated.

Each result comes with a short explanation of what to change. A check that stops with an error is shown as “A check could not be completed”; the other checks are not affected.

The checks

Checks about the server and the installation:

Check	What it looks at
PHP version	PHP still gets security fixes.
Application secret	The application secret is long and not a sample value.
`display_errors`	PHP does not show error messages to visitors.
`expose_php`	PHP does not announce its version.
`allow_url_include`	PHP does not include code from other servers.
Debug mode	Contao runs in production mode without debugging.
HTTPS	The site address starts with `https://`.
Files with passwords and keys	Other users cannot read them.
World-writable paths	No file or folder in the code can be changed by everybody.
Public leftovers	No backups or debugging files are in the public folder.
Disk space	Enough free space. Below 5 percent it is a warning, below 1 percent a problem.

The values of `display_errors`, `expose_php` and `allow_url_include` cannot be read reliably from the command line, because the web server can use another PHP configuration. Open the tab **Checkup** in the back end for them.

Without a configured host, the checkup does not rate the HTTPS check of the site address.

Checks about the bundle:

Check	What it looks at
The firewall is on	**Firewall mode** is **On**. **Log only** is a warning, **Off** a problem.
Sign-in protection is on	**Sign-in protection** is on. Off is a problem.
Alerts have a receiver	A notification address is available.

Check	What it looks at
The file scan ran	A scan finished in the last 8 days.
No open findings of a high level	An open finding of the level High is a problem, an open one of the level Medium a warning.

Checks specific to Contao:

Check	What it looks at
Administrators use two-factor authentication	Warns while active administrators have none.
Few administrator accounts, all in use	More than 5 active administrators, or an administrator who has not signed in for 180 days, is a warning.
Contao Manager is not left on the server	The check looks for `contao-manager.phar.php` in the public directory and reports it as a note if it is there.
Trusted proxies match the real setup	See Behind a reverse proxy or CDN.

Listed files and the permission fix

Three checks list what they found below their advice: **Files with passwords and keys**, **World-writable paths** and **Public leftovers**. The names are relative to the project folder, for public leftovers to the public folder. The lists are shown on the tab **Checkup**, on the **Dashboard** and by `isprotect:checkup`. The list of world-writable paths has at most 20 entries, the list of public leftovers at most 50. A longer list ends with “The list shows the first {count} entries only.”

When the check **Files with passwords and keys** shows **Needs attention**, the button **Fix permissions** removes all permissions of “others” from the listed files, the same as `chmod o-rwx`. The owner and the group keep theirs. The button takes the file names from the check itself, never from the browser, and only changes a file that

- is a regular file directly in the project folder, not a link or a folder, with no second name (hard link), and
- belongs to the user PHP runs as.

The button is shown when at least one listed file can be changed. For every other file the message names the command to run in the project folder, for example `chmod o-rwx .env`. Afterwards the check runs again and shows the new result. The change is recorded in the audit log as “File permissions changed by the checkup”, with the names of the changed and of the unchanged files.

A program that reads these files under another user account, for example a cron job, needs to be in the group of the file after the change.

The other two checks have no button. Remove the write permission of “others” from the listed paths, for example with `chmod o-w`, and delete the listed leftovers or move them out of the public folder.

Security headers

The bundle can change three headers of responses. The settings are in the group Security headers, see Settings reference.

- **Hide the PHP version** removes the `X-Powered-By` header. It is on by default.
- **Enforce HTTPS (HSTS)** adds `Strict-Transport-Security`, with the age from **HSTS duration (seconds)**. It is sent on HTTPS requests only and is off by default. Turn it on only when HTTPS works everywhere on the site, because browsers remember it for the whole duration.
- **Permissions-Policy header** sets the `Permissions-Policy` header to the text you enter. Leave it empty to send none.

Audit log and notifications

The audit log

The audit log is the tab **Events**. It records security-relevant events with time, level, address, user name and details. The levels are info, notice, warning and alert. The tab lists the events, newest first, and you can filter them:

Filter	Meaning
Event	One type of event
Minimum level	Only events of this level or higher
Address	An address
User name	A user name
Area	The area, `backend` or `frontend`
From (date) and **Until (date)**	A range of dates
Search	Text in the event

Click **Filter** to apply the filter and **Reset filter** to clear it. The columns are **Time**, **Level**, **Event**, **Address**, **User name**, **Area** and **Details**.

What is recorded

- Sign-ins: successful, failed, refused, sign-out, user switch, sign-in from a new address, locked user names and addresses, lifted locks, wrong codes of the second step.
- Firewall: refused and logged requests, bans, lifted bans, changes of the address lists.
- Scanner: started and finished scans, findings, changed packages, known vulnerabilities, and system notices such as a mail that could not be sent.
- Accounts: creation, change and deletion of back end users, back end user groups and members, and role changes, when done in the back end.
- Settings: changes to the settings of this bundle, made on the pages or with `ispprotect:settings:set` (event “Settings changed”).
- Remote scan: storing and removing the access token, connecting, refreshing and disconnecting the licence, and refusals by the licence portal.

For accounts, the log records the names of the changed fields, never the old or new values, with one exception: for back end users it records the old and new e-mail address. Whether a password changed is known from the field being submitted. The password and its hash are never read or stored. Creating an administrator, making a user an administrator and turning off a user’s two-factor authentication in the user form are warnings. Copying a record, restoring it from “undo” and restoring an older version are recorded too. Front end members who register, activate or close their account, or set a new password, are recorded as well.

A change to the settings of this bundle is recorded with the names of the changed settings, never their values.

E-mail addresses are stored in the audit log only for back end users, never for members.

Retention and shortened addresses

Keep log entries (days) says how long entries stay. Expired entries are removed once an hour. With **Shorten addresses in the log** on, IPv4 addresses are stored without the last of their four numbers (it is set to 0, so 192.0.2.17 is stored as 192.0.2.0); of IPv6 addresses only the first 48 bits are kept.

Gaps in the audit log

- Accounts created or changed on the command line or directly in the database are not recorded.

- A user who turns two-factor authentication on or off in their own account settings is not recorded. A change made by an administrator in the user form is recorded.
- Front end members who change their personal data other than the password are not recorded.
- Restoring a version records only that a version was restored and its number, not what changed.
- Settings changed directly in the database are not recorded.
- If saving an entry fails, the failure is logged and Contao saves the record anyway.

On the console, `ispprotect:events` shows the newest events, see Command line.

Notifications

The bundle sends mails through Contao's mailer, in plain text. The sender is Contao's system administrator address if one is set. There are three kinds:

- **Alerts** for events of the level alert, if **Immediate alerts** is on. There is at most one mail per event type and hour. After a failed mail, the bundle waits 15 minutes before it tries again.
- **Summary mail**: a daily or weekly list of the security events, as set in **Summary mail**. The value **Off** sends none.
- A mail about a sign-in from a new address, if **Alert on sign-ins from new addresses** is on.

The receiver is **Notification address**. If it is empty, the bundle uses the administrator address in the Contao settings, and without one the address of the first active administrator. Without any receiver, no alert or summary mail is sent. The checkup reports this.

The site address in the mails comes from the first root page with a domain, else from Contao's configured request context (`router.request_context.host` and `router.request_context.scheme`, or `framework.router.default_uri`). It is never taken from the current request, because the mails are often sent from the cron route `/_contao/cron`, which anybody can call with a Host header of their choice. Without a configured host, the mails leave the address out.

Mails are sent by the maintenance run, so they need the cron, see Scheduling with cron.

Settings reference

This chapter lists every setting of the bundle, grouped as the pages show them. For each setting you see the label as the page shows it, the key that the console uses, the default and what it does.

Changing settings in the back end

The settings are on four tabs:

Tab	Groups
Firewall	Firewall, Request rate limit, Missing pages, Trap paths, Custom rules, Bans
OWASP CRS rules	OWASP CRS rules
Sign-in protection	Sign-in protection, Two-factor authentication
Settings	Security headers, Audit log, Notifications, Scanner, Remote scan

Change the values and click **Save**. The page checks every value. A value that is not allowed is not saved, and the page shows the reason next to the field. Below the form, a button named **Reset** with the group name sets one group back to its defaults, after you confirm. The setting `remote.portal_url` is not on any page.

Changing settings on the console

Show one setting or all of them:

```
vendor/bin/contao-console ispprotect:settings:get firewall.mode
vendor/bin/contao-console ispprotect:settings:get
```

Change a setting:

```
vendor/bin/contao-console ispprotect:settings:set firewall.mode log
vendor/bin/contao-console ispprotect:settings:set trap.paths '["/wp-login.php","/xmlrpc.php"]'
```

The value is read as JSON when it can be, for example `true`, `5` or `["a","b"]`, and otherwise as text. A value that is not allowed is refused with exit code 2 and the reason. A change is recorded in the audit log with the key and without the value. See Command line.

Value types

- Switches are shown as check boxes. On the console they accept `1`, `0`, `true`, `false`, `on`, `off`, `yes` and `no`. The tables show them as `true` and `false`.
- Numbers have a smallest and a largest value. The tables name them.
- Lists take up to 500 entries of up to 255 bytes each, unique. In the forms and with `ispprotect:settings:get` they are one entry per line; on the console `ispprotect:settings:set` takes a JSON list.
- Lists of paths (**Trap paths**, **Paths without request limit**) refuse an entry that would cover the whole site.

Firewall group

Label	Key	Default	What it does
Firewall mode	`firewall.mode`	`on`	`on` refuses requests that break a rule, `log` only records them, `off` turns the firewall off.

Label	Key	Default	What it does
Protect administrator addresses	`firewall.allow_admin_ip`	`true`	After an administrator signs in, that address cannot be banned for 24 hours.

Request rate limit group

Label	Key	Default	What it does
Limit requests per address	`rate.enabled`	`true`	Refuses addresses that send more requests than allowed within the time window.
Requests per window	`rate.limit`	`600`	How many requests one address may send within the time window, 30 to 100000.
Time window (seconds)	`rate.window`	`60`	Length of the time window for the request limits, 10 to 3600.
Sign-in requests per window	`rate.signin_limit`	`30`	How many requests to the sign-in paths one address may send within the time window, 5 to 1000.
Paths without request limit	`rate.exempt_prefixes`	empty	Request paths that are not counted, in addition to the bundle's own list.
Ban after refusals	`rate.ban_after`	`3`	Bans an address after this many refusals within 10 minutes, 0 to 100. `0` never bans.

Missing pages group

Label	Key	Default	What it does
Watch "not found" answers	`notfound.enabled`	`true`	Bans addresses that cause many "not found" answers, which is typical for scanners.
"Not found" answers per window	`notfound.limit`	`40`	How many such answers one address may cause within the window before it is banned, 5 to 10000.
"Not found" time window (seconds)	`notfound.window`	`300`	Length of the window, 60 to 3600.

Trap paths group

Label	Key	Default	What it does
Trap paths	`trap.paths`	empty	A request to one of these paths bans the address at once.

Custom rules group

Label	Key	Default	What it does
Custom rules	`rules.custom`	empty	Your own rules that block or log requests by path, query, browser identification, referrer, method or host. Up to 100 rules, see Custom rules.

Bans group

Label	Key	Default	What it does
Ban length (minutes)	`ban.base_minutes`	`15`	Length of the first ban of an address, 1 to 10080.

Label	Key	Default	What it does
Longest ban (minutes)	`ban.max_minutes`	`10080`	The longest ban or sign-in lock, 1 to 525600. Each repeated ban within 30 days lasts four times as long, up to this length.

Sign-in protection group

Label	Key	Default	What it does
Sign-in protection	`login.enabled`	`true`	Counts failed sign-ins, bans addresses and locks user names when the limits are reached.
Failed sign-ins per address	`login.ip_limit`	`5`	Bans an address after this many failed sign-ins within the time window, 2 to 1000.
Failed sign-ins per user name	`login.user_limit`	`10`	Locks a user name after this many failed sign-ins within the time window, 2 to 1000.
Sign-in time window (minutes)	`login.window_minutes`	`15`	Length of the window in which failed sign-ins are counted, 1 to 1440.
Lock length (minutes)	`login.lock_minutes`	`15`	How long a locked user name stays locked, 1 to 1440. It is also the length of the first sign-in lock of a protected administrator address.
Alert on sign-ins from new addresses	`login.alert_new_address`	`true`	Sends a mail when an administrator signs in from an address not seen for that user in 90 days.

Two-factor authentication group

Label	Key	Default	What it does
Require a second sign-in step	`twofactor.required`	`false`	Administrators must set up a second sign-in step with an authenticator app. See Two-factor authentication.

Audit log group

Label	Key	Default	What it does
Keep log entries (days)	`audit.retention_days`	`90`	Log entries older than this are removed, 1 to 3650.
Shorten addresses in the log	`audit.anonymize_ip`	`false`	Stores IPv4 addresses without the last of their four numbers (set to 0) and keeps only the first 48 bits of IPv6 addresses.

Notifications group

Label	Key	Default	What it does
Notification address	`notify.email`	empty	Receives alerts and summaries. A valid e-mail address of up to 254 characters, or empty to use the administrator address of Contao.
Immediate alerts	`notify.alerts`	`true`	Sends a mail for the most serious events, at most one per event type and hour.
Summary mail	`notify.digest`	`daily`	Sends a summary of the security events: `off`, `daily` or `weekly`.

Scanner group

Label	Key	Default	What it does
File scan schedule	`scanner.schedule`	`daily`	How often the files are scanned automatically: `off`, `daily` or `weekly`.
File scan hour	`scanner.hour`	`3`	Hour of the day, in UTC, from which the scheduled scan may start, 0 to 23.
Detect changed files	`scanner.baseline`	`true`	Compares code files with the last scan and reports changed, added and removed files.
Search for malicious code	`scanner.content`	`true`	Runs a small set of generic suspicious-code checks on file contents.
Check for known vulnerabilities	`scanner.advisories`	`true`	Asks packagist.org for known vulnerabilities of the installed packages. Only package names leave the server.
Excluded paths	`scanner.exclude`	empty	Paths relative to the project directory that the scan skips.
Largest scanned file (KB)	`scanner.max_file_kb`	`5120`	Largest file that is read completely, 64 to 102400. Larger files are not read completely.

Remote scan group

Label	Key	Default	What it does
Remote scan	`remote.mode`	`off`	The mode: `off`, `daemon` (**Local scan service (socket)**), `service` (**Own scan service**) or `license` (**Hosted scan service (licence)**). See Remote scan and licence.
Socket of the local scan service	`remote.socket`	`/run/ispprotect/scand.sock`	Socket of the local scan service, used in the mode `daemon`.
Address of the scan service	`remote.service_url`	empty	HTTPS address of your own scan service, used in the mode `service`.
Address of the hosted scan service	`remote.hosted_url`	empty	Optional override of the built-in hosted scan service `https://scan.ispprotect.com` (mode `license`). Normally left empty; set an HTTPS address only to use another deployment, for example for testing.
Address of the licence portal	`remote.portal_url`	empty (uses `https://license.ispprotect.com`)	HTTPS address of a different licence API deployment. It is not on any page; override it on the console only when needed.
Upload limit per scan (MB)	`remote.upload_mb`	`200`	The most data that one scan uploads, 0 to 4096. `0` sends checksums only.
Paths not sent to the scan service	`remote.exclude`	empty	Paths relative to the project directory that are never sent to the service.

Security headers group

Label	Key	Default	What it does
Hide the PHP version	`headers.hide_php_version`	`true`	Removes the `X-Powered-By` header from responses.

Label	Key	Default	What it does
Enforce HTTPS (HSTS)	`headers.hsts`	`false`	Sends `Strict-Transport-Security` on HTTPS requests. Turn it on only when HTTPS works everywhere.
HSTS duration (seconds)	`headers.hsts_max_age`	`31536000`	How long browsers remember to use only HTTPS, 300 to 63072000.
Permissions-Policy header	`headers.permissions_policy`	empty	Value of the `Permissions-Policy` header, up to 1024 bytes without control characters. Empty sends no header.

OWASP CRS rules group

The group is on its own tab, **OWASP CRS rules**. The rules must be installed before the mode **Log only** or **Block** has an effect. See OWASP CRS rules.

Label	Key	Default	What it does
CRS rules mode	`waf.mode`	`off`	`off`, `log` or `block`. The first installation of the rules sets `log`. See Start with logging.
Paranoia level	`waf.paranoia`	`1`	1 to 4. Higher levels run more rules and report more normal requests. See Paranoia level and threshold.
Anomaly threshold	`waf.threshold`	`5`	Score at which a request counts as an attack (1 to 100).
Disabled rules	`waf.disabled_rules`	empty	Rule numbers that are not run, one per line (up to 500). A rule number is a whole number, for example `942190`.
Inspected body size	`waf.body_limit`	`131072`	Bytes of a request body that are inspected (1024 to 1048576). A larger body is refused with status 413 unless uninspected requests may pass.
Time limit per request	`waf.time_budget_ms`	`200`	Most milliseconds the rules may take for one request (5 to 10000).
Let requests pass that cannot be fully inspected	`waf.pass_uninspected`	true	Lets requests through that are too large, too slow or cannot be read. They are still recorded. See Requests that cannot be fully inspected.

Command line

All commands start with `vendor/bin/contao-console` and run in the project directory, as the user that owns the files. The examples leave out the prefix when they show only the command name.

The exit code is 0 on success. Exit code 1 means that the command ran and found a problem. Exit code 2 means that an argument or option was not valid. Requests made through the console are not checked by the firewall.

List of commands

Command	What it does
`ispprotect:status`	Shows the state of the protection
`ispprotect:checkup`	Runs the security checkup
`ispprotect:scan`	Runs a file scan
`ispprotect:findings`	Lists and accepts findings
`ispprotect:events`	Shows the audit log
`ispprotect:ip:list`	Lists the address lists and bans
`ispprotect:ip:add`	Adds an address to a list
`ispprotect:ip:remove`	Removes an address from all lists
`ispprotect:unlock`	Lifts a sign-in lock
`ispprotect:settings:get`	Shows settings
`ispprotect:settings:set`	Changes a setting
`ispprotect:maintenance`	Runs the periodic work
`ispprotect:remote`	Looks after the remote scan and the licence
`ispprotect:uninstall`	Removes the data of the bundle

Status and checkup

`ispprotect:status` shows the firewall mode, the sign-in protection, the number of active bans and locks, the open findings, the last complete scan and the number of events of the last 24 hours with level warning or higher. It always ends with exit code 0.

`ispprotect:checkup` prints the score and every check with its status. It ends with exit code 1 if at least one check has the status **Problem**, so you can use it in monitoring.

Scan and findings

```
ispprotect:scan [--budget=SECONDS] [--no-wait]
```

Starts a scan, or continues one that is running, and works in steps of 5 seconds until it is finished.

Option	Meaning
`--budget=SECONDS`	Stops after this many seconds, at most 86400. The scan continues the next time. A value that is not a number or too large gives exit code 2.
`--no-wait`	Does not wait for a scan that another process is running.

Exit code 1 means that the scan failed, or that open findings of level Medium or High exist after the scan finished.

```
ispprotect:findings [--all] [--accept=ID] [--accept-all]
```

Lists the open findings, at most 500 rows. `--all` includes findings that are not open. `--accept=ID` accepts the changed, added or removed file of one finding as the new baseline, `--accept-all` does so for every such finding. A finding that changed since it was listed, or that does not exist, gives exit code 1. An invalid ID gives exit code 2. Without an accept option, the exit code is 1 when open findings of level Medium or High exist; with a clean list it is 0.

Audit log

```
ispprotect:events [--type=NAME] [--since=MINUTES|DATE] [--limit=N]
```

Option	Meaning
`--type`	Only this type of event
`--since`	Events of the last given minutes, or since a date and time in UTC
`--limit`	Number of events, 1 to 500, default 50

Addresses

```
ispprotect:ip:list
ispprotect:ip:add KIND ADDRESS [--minutes=N] [--reason=TEXT]
ispprotect:ip:remove ADDRESS
```

`ispprotect:ip:list` shows the allow list, the block list and the active bans.

`ispprotect:ip:add` takes the kind `allow`, `block` or `ban` and an address or a range in CIDR notation. `--minutes` (1 to 5256000) sets the length of a ban, `--reason` a note. An invalid kind, address or number gives exit code 2. An address that cannot be stored, such as a loopback address, gives exit code 1.

`ispprotect:ip:remove` removes the address from every list, including bans. An address that is on no list gives exit code 1, an invalid address exit code 2.

Sign-in locks

```
ispprotect:unlock [USERNAME] [--address=IP] [--realm=backend|frontend]
```

Give exactly one of the user name and `--address`; otherwise exit code 2. An invalid address gives exit code 2. If nothing was locked, the exit code is 1. See Sign-in protection. The command does not lift a firewall ban and does not reset the count of wrong codes of the second step for a user name.

Settings

```
ispprotect:settings:get [KEY]
ispprotect:settings:set KEY VALUE
```

An unknown key gives exit code 2. `settings:set` reads the value as JSON if it can, otherwise as text. A refused value gives exit code 2 and the reason. See Settings reference.

Maintenance

`ispprotect:maintenance` runs the periodic work: scheduled scans, removing old log entries, mails, renewing the licence. It ends with exit code 1 if a step failed. You normally leave it to the cron, see Scheduling with cron.

Remote scan commands

```
ispprotect:remote [ACTION]
```

Without an action, it shows the status. The actions are:

Action	What it does
`status`	Shows mode, connection and the last scan
`test`	Tests the connection
`token:set`	Stores the access token
`token:clear`	Removes the access token
`connect`	Connects a licence key
`refresh`	Renews the access of a connected licence
`disconnect`	Releases the licence and removes its data

The access token and the licence key are never taken as arguments, so they do not appear in the process list or the shell history. Pipe them in, or type them at the hidden prompt:

```
printf '%s' "$TOKEN" | vendor/bin/contao-console ispprotect:remote token:set
```

An extra argument or an unknown action gives exit code 2, as does an empty secret or one that is too long. If no hidden input is possible, the exit code is 1. A failed test, connection, renewal or storing gives exit code 1, as does `connect` when a licence is already connected. `disconnect` ends with exit code 0 even if the portal could not be reached; the local data is removed in every case. See Remote scan and licence.

Removal

`ispprotect:uninstall` is described in Removing the bundle.

Getting back in and troubleshooting

You are banned or blocked by the firewall

The firewall refuses requests of a banned or blocked address, also for the back end. Requests on the console are not affected, so you can fix it there:

```
vendor/bin/contao-console ispprotect:ip:list
vendor/bin/contao-console ispprotect:ip:remove 203.0.113.7
```

`ispprotect:ip:list` shows the bans and lists, `ispprotect:ip:remove` removes the address from all of them. To switch the firewall off for a moment, set the mode to `off`:

```
vendor/bin/contao-console ispprotect:settings:set firewall.mode off
```

Turn it on again afterwards with `on`. See Firewall.

If the whole site is unreachable for all visitors, check first whether the address the bundle sees is the one of the visitor, see Behind a reverse proxy or CDN.

You are locked by the sign-in protection

After too many failed sign-ins, the user name or the address is locked. Lift the lock on the tab **Sign-in protection** from another browser or address, or on the console:

```
vendor/bin/contao-console ispprotect:unlock USERNAME
vendor/bin/contao-console ispprotect:unlock --address=203.0.113.7
```

Add `--realm=backend` or `--realm=frontend` to limit the command to one area. A lock ends by itself after **Lock length (minutes)**. See Sign-in protection.

A trap path or a rule blocks the wrong requests

A trap path or a custom rule that is too wide can ban or block visitors, also you. Remove them on the console:

```
vendor/bin/contao-console ispprotect:settings:set trap.paths '[]'
vendor/bin/contao-console ispprotect:settings:set rules.custom '[]'
```

Then lift your own ban as described above. To test a rule first, turn on **Firewall mode Log only**, which only records what would be refused.

Two-factor authentication is required and an administrator cannot use it

Turn off **Require a second sign-in step** on the tab **Sign-in protection**, or on the console:

```
vendor/bin/contao-console ispprotect:settings:set twofactor.required false
```

Contao's own option `contao.security.two_factor.enforce_backend` is not changed by this. See Two-factor authentication.

The pages show tables are missing

The page says that the tables are missing when the database was not updated after the installation. Run `contao:migrate` or use the Contao Manager, see Installation and first steps and If the tables are missing.

The pages of the bundle have no layout

The assets are not installed. Run:

```
vendor/bin/contao-console assets:install
```

All visitors have the same address

The bundle sees only the address of your proxy or CDN. Enter its addresses as trusted proxies in the Contao configuration, see Behind a reverse proxy or CDN. The checkup reports a wrong setting as “Trusted proxies match the real setup”.

Scans or mails do not happen

Scheduled scans, the summary mails and the removal of old log entries need the cron of Contao. Check that the system cron calls the Contao cron every minute, see Scheduling with cron. You can also run the periodic work by hand:

```
vendor/bin/contao-console ispprotect:maintenance
```

A scan that does not seem to move on continues in steps at each run. `ispprotect:scan` finishes it on the console.

If no mails arrive, check **Notification address**, the mail transport of Contao and the events of the type “System notice” on the tab **Events**. The checkup tells you if no receiver exists. See Audit log and notifications.

The remote scan or the licence does not work

- Click **Test connection** on the tab **Scanner**. The section **Remote scan** shows the reason for a refusal, see Remote scan and licence.
- If the licence section shows a problem, fix the cause and click **Check now**, or run `ispprotect:remote refresh`.
- If the application secret of Contao was changed, the stored token and licence data cannot be read any more. Enter the token or the licence key again.
- No **Connect** button: check that the built-in licence API address has not been replaced by an invalid `remote.portal_url` override; see Remote scan and licence.

Something else

The tab **Events** shows what the bundle did and why. Filter by the address or the user name that has the problem. If the pages themselves fail, the bundle lets requests pass instead of blocking them: a failing firewall does not take the site down. The error is in the log of Contao.

The OWASP CRS rules refuse a normal request

The OWASP CRS rules can refuse a normal request (a false positive). The request is answered with status 403, or with 413 or 400 for a request that cannot be inspected completely. See OWASP CRS rules for the background.

1. Open the tab **Events** and find the entry **Request refused by the firewall** (or **Request logged by the firewall**, in mode **Log only**). Note the rule numbers in the details.

2. Disable the rule that is wrong: enter its number in the form **Disable a rule** on the tab **OWASP CRS rules**. A rule that you disable no longer runs for any request, so disable only the rule that caused the trouble.
3. If a save of a large amount of content fails with status 413 and the number `200101`, raise **Inspected body size** (`waf.body_limit`), for example to 524288.
4. If requests stop with the number `200100`, raise **Time limit per request** (`waf.time_budget_ms`), or disable the number `200100`. Program code pasted into a form field (about 20 to 40 KB) can use up the default limit of 200 milliseconds. Requests refused with the number `200005` (rarely, a long field with program code exceeds the regular expression limits of PHP) are answered the same way: disable the number `200005` or let requests pass that cannot be fully inspected.

To find the cause without losing visitors, set **CRS rules mode** to **Log only**: requests are then only recorded.

If you cannot sign in any more, switch the CRS rules off on the console, then sign in and disable the rule:

```
vendor/bin/contao-console ispprotect:settings:set waf.mode off
```

Privacy and external services

The bundle runs on your server and stores its data in your database. This chapter lists everything that leaves the server.

Contact with outside services

Service	When	What is sent
packagist.org	**Check for known vulnerabilities** is on	The names of the installed packages. No versions: they are compared on your server.
Scan service	**Remote scan** is not **Off**	See below.
OWASP CRS rules (GitHub)	You click **Install rules** or **Update rules**	Only the download request for the rules archive of the CRS project, to `github.com` and the download host that GitHub redirects to. Nothing about your website, its visitors or their requests. The download is checked against a checksum built into the bundle.
Licence portal	In the mode **Hosted scan service (licence)**, when you connect, renew or disconnect a licence, and when you remove the bundle	See below.

The bundle makes no other contact with outside services. It sends no usage data and no data to its authors.

Scan service

With **Local scan service (socket)**, nothing leaves the server. With **Own scan service** the bundle sends to the address you set, and with **Hosted scan service (licence)** to the ISPProtect scan service at `https://scan.ispprotect.com`. In both modes it sends for each code file the SHA-256 checksum and the file name or only `file.` with the extension. It uploads files that the service does not know, up to **Upload limit per scan (MB)**. Files with secrets, such as keys, `.env` files and database dumps, are never sent. A few PHP configuration files are sent only in redacted form, with all strings, numbers and comments replaced. See Remote scan and licence for the list.

Licence portal

Action	Data sent
Connect	The licence key, the product name, a random installation ID, the host name of the site, the platform `contao` and the version of the bundle
Renew	The licence key, the installation ID, the host name and the version
Disconnect, remove	The licence key and the installation ID

The installation ID is a random value created by the bundle. Licence requests go to `https://license.ispprotect.com`, not the account console at `https://console.ispprotect.com`. They wait 15 seconds, and 5 seconds when the bundle is removed.

Data in your database

- The audit log holds events with time, address, user name and details. E-mail addresses are stored only for back end users. Entries are removed after **Keep log entries (days)**. **Shorten addresses in the log** shortens addresses before they are stored: the last of the four numbers of an IPv4 address is set to 0, of an IPv6 address only the first 48 bits are kept.
- The counters for the sign-in protection hold addresses and user names. Locks of protected administrator addresses hold only a hash of the address.
- Accounts of back end users, groups and members are logged by field names only, without values.
- Passwords and password hashes are never read or stored.

- The access token and the licence key are stored encrypted, see Remote scan and licence.

When you remove the bundle, all of this data is deleted, see Removing the bundle.

Removing the bundle

Remove the bundle in two steps: first its data, then the code.

Delete the data

```
vendor/bin/contao-console ispprotect:uninstall
```

The command does this:

1. If a licence key is stored, it releases the site at the licence portal and says whether that worked. If the portal could not be reached, the site still counts toward the site limit of the licence until you release it in your account at the licence portal.
2. It deletes all tables of the bundle: settings, address lists, locks, counters, audit log, scans, files, findings, the table for two-factor data and the remote scan data.
3. It clears the cache pool of the bundle.

It removes no files and changes nothing else in Contao. The command asks for confirmation. Without a terminal, it needs `--force` (or `-f`) and otherwise stops with exit code 1; if you decline, it deletes nothing and also ends with exit code 1.

> The deletion cannot be undone. The audit log and the findings are gone afterwards.

Do not run `contao:migrate` between the two steps. As long as the bundle is installed, it creates the empty tables again.

Remove the code

```
composer remove ispprotect/contao-web-security-bundle
```

Then clear the cache of Contao. You can also remove the bundle with the Contao Manager.