Overview

CasWAF is an open-source Web Application Firewall (WAF) software developed by Go and React.

CasWAF features

1. Front-end and back-end separate architecture, developed by Golang, CasWAF provides web-based managing UI and supports multiple languages(Chinese, English).

2. Databases. CasWAF supports mainstream databases: MySQL, PostgreSQL, SQL Server, etc.

3. Casdoor SSO. CasWAF uses Casdoor as the Identity Provider (IdP) for OAuth login.

4. Reverse proxy. CasWAF supports reverse proxy, which can be used as a reverse proxy server to protect the backend server.

5. OAuth proxy. CasWAF supports OAuth login, which can be integrated with the existing OAuth system such as Casdoor.

6. Firewall. CasWAF uses Coraza as the firewall engine, which can protect the website from many common attack categories. Supports customized WAF rules.

How it works

CasWAF has both reverse proxy and OAuth proxy functionalities. If you haven't configured OAuth for your website, it will function solely as a reverse proxy server.

Reverse proxy

CasWAF appears externally as a reverse proxy server, providing an additional layer of security for your web servers and applications.

It sits between the users and web servers, acting as an intermediary, receiving requests from users, and forwarding them to the target web servers.

OAuth proxy

CasWAF OAuth proxy acts as an Identity Provider (IdP) and collaborates with your application to perform authentication and authorization through the OAuth protocol.

When a user attempts to access a resource that requires authentication, CasWAF will redirect the user to the real Identity Provider (IdP).

Subsequently, the OAuth proxy will guide the user to the configured Identity Provider (e.g., Google, Facebook, or an internal authentication service within the company) for authentication.

In CasWAF, we use Casdoor as the Identity Provider (IdP). More information of Casdoor SSO can be found here.

Online Demo

Here is an online demo:

- Deployed site: https://door.caswaf.com

Global admin login:

- Username: `admin`
- Password: `123`

Architecture

Caswaf contains 2 parts:

Name	Description	Language	Source code
Frontend	Web frontend UI for CasWAF	Javascript + React	https://github.com/casbin/caswaf/tree/master/web
Backend	RESTful API backend for CAsWAF	Golang + Beego + MySQL	https://github.com/casbin/caswaf

Core Concepts

As CasWAF's administrator, you should get familiar with at least 2 core concepts: `Site`, `Cert`.

Site

In CasWAF, `Site` is representing the real applications or websites you wish to protect. Each Site is associated with specific domain names or IP addresses, and you can configure multiple Sites according to different needs to ensure comprehensive security protection for all your applications.

The Site class definition is shown as follows:

```
type Site struct {
    Owner       string `xorm:"varchar(100) notnull pk"
json:"owner"`
    Name        string `xorm:"varchar(100) notnull pk" json:"name"`
    CreatedTime string `xorm:"varchar(100)" json:"createdTime"`
    UpdatedTime string `xorm:"varchar(100)" json:"updatedTime"`
    DisplayName string `xorm:"varchar(100)" json:"displayName"`

    Tag          string     `xorm:"varchar(100)" json:"tag"`
    Domain       string     `xorm:"varchar(100)" json:"domain"`
    OtherDomains []string   `xorm:"varchar(500)"
json:"otherDomains"`
    NeedRedirect bool       `json:"needRedirect"`
    EnableWaf    bool       `json:"enableWaf"`
    Waf          coraza.WAF `xorm:"-" json:"-"`
    Challenges   []string   `xorm:"mediumtext" json:"challenges"`
```

Cert

In CasWAF, `Cert` is representing the certificates used for HTTPS authentication. By configuring Cert, CasWAF can establish secure encrypted connections between itself and the clients, ensuring the confidentiality and integrity of data and preventing information leakage and tampering attacks.

The Cert class definition is shown as follows:

```
type Cert struct {
    Owner       string `xorm:"varchar(100) notnull pk"
json:"owner"`
    Name        string `xorm:"varchar(100) notnull pk" json:"name"`
    CreatedTime string `xorm:"varchar(100)" json:"createdTime"`
    DisplayName string `xorm:"varchar(100)" json:"displayName"`

    Type            string `xorm:"varchar(100)" json:"type"`
    CryptoAlgorithm string `xorm:"varchar(100)"
json:"cryptoAlgorithm"`
    ExpireTime      string `xorm:"varchar(100)" json:"expireTime"`

    Certificate string `xorm:"mediumtext" json:"certificate"`
    PrivateKey  string `xorm:"mediumtext" json:"privateKey"`
}
```

Installation

CasWAF uses Casdoor to manage members. So you need to create an organization and an application for CasWAF in a Casdoor instance.

Requirements

OS

All major operating systems including Windows, Linux and macOS are supported.

Environments

- Go 1.17+
- Node.js LTS (16 or 14)
- Yarn 1.x

> **信息**
>
> We strongly suggest you use Yarn 1.x to run & build CasWAF frontend, using NPM might cause UI styling issues, see more details at: casdoor#294

> **警告**
>
> For **Chinese** users, in order to download the Go dependency packages successfully, you need to use a Go proxy by Configuring the GOPROXY environment variable. We strongly recommend: https://goproxy.cn/

Download

The source code of CasWAF is hosted at GitHub: https://github.com/casbin/caswaf. Both the Go backend code and React frontend code are inside the single repository.

Name	Description	Language	Source code
Frontend	Web frontend UI for CasWAF	Javascript + React	https://github.com/casbin/caswaf/tree/master/web
Backend	RESTful API backend for CAsWAF	Golang + Beego + MySQL	https://github.com/casbin/caswaf

CasWAF supports Go Modules. To download the code, you can just simply clone the code both via `go get` and `git`:

```
go get github.com/casbin/casdoor
go get github.com/casbin/caswaf
```

or

```
git clone https://github.com/casbin/casdoor
git clone https://github.com/casbin/caswaf
```

Necessary configuration

Set up database

CasWAF will store its users, nodes and topics information in a MySQL database named: `caswaf`. CasWAF will create it if not existed. The DB connection string can be specified at: https://github.com/casbin/caswaf/blob/master/conf/app.conf

```
dataSourceName = root:123@tcp(localhost:3306)/
```

CasWAF uses XORM to connect to DB, so all DBs supported by XORM can also be used.

Configure Casdoor

> **信息**
>
> In order not to affect Docker users, we temporarily chose to embed the WAF rules into the binary, if you need to change the default rules (`conf/waf.conf`), please do so before compiling.

After creating an organization and an application for CasWAF in a Casdoor, you need to update `clientID`, `clientSecret`, `casdoorOrganization` and `casdoorApplication` in app.conf and Conf.js to change the configuration.

- Backend (conf/app.conf)

```
casdoorEndpoint = <Your Casdoor endpoint>
```

- Frontend (web/src/Conf.js)

```
serverUrl: "<Your Casdoor endpoint>"
clientId: "<Your Casdoor application's client ID>"
appName: "<Your Casdoor application name>"
organizationName: "<Your Casdoor organization name>"
```

More details about Casdoor configuration can be found at: casdoor-sso

Run CasWAF

- Build backend of CasWAF
  - `go build github.com/casbin/caswaf`
- Build frontend of CasWAF
  - `yarn start`
- Now you can visit CasWAF configuration website at:
  - `http://localhost:16001/`

Optional configuration

Set up your WAF to enable some third-party login platform

CasWAF uses Casdoor to manage members. If you want to log in with oauth, you should see casdoor oauth configuration.

OSS, Mail, and SMS services

CasWAF uses Casdoor to upload files to cloud storage, send Emails and send SMSs. See Casdoor for more details.

Casdoor SSO

Introduction

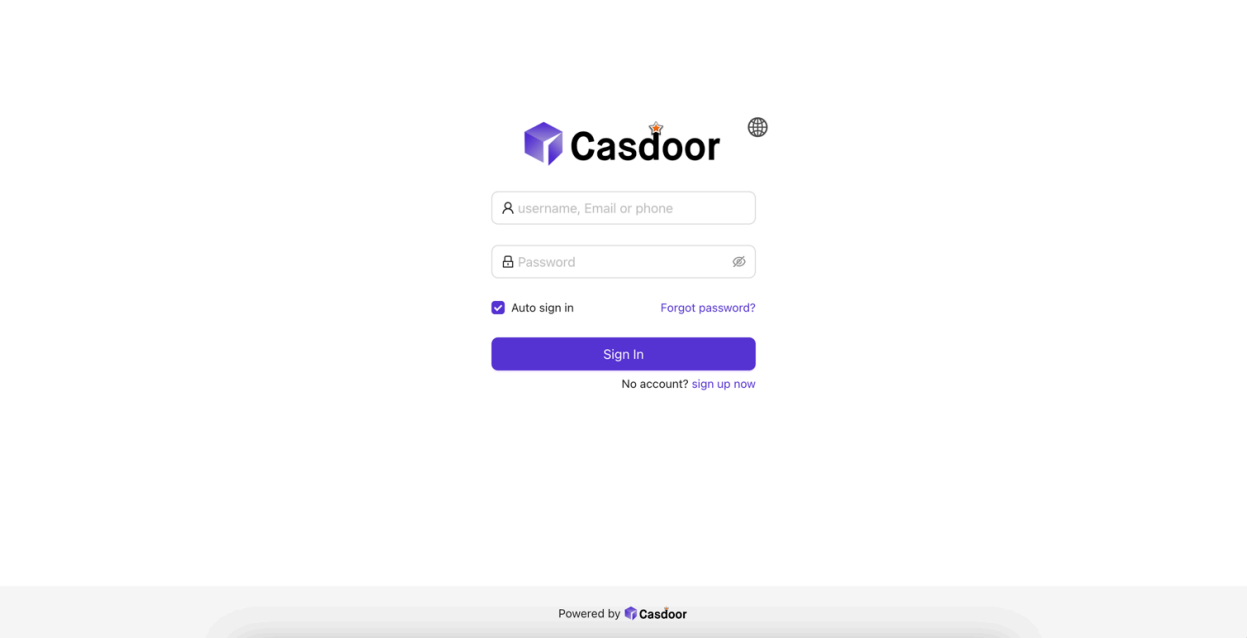
Casdoor is a UI-first centralized authentication / Single-Sign-On (SSO) platform based on OAuth 2.0 / OIDC. It provides OAuth 2.0 / OIDC based signup and login, as well as third-party login including GitHub, Google, QQ, WeChat, etc. CasWAF uses Casdoor as an SSO provider to manage users and permissions.

Login to CasWAF

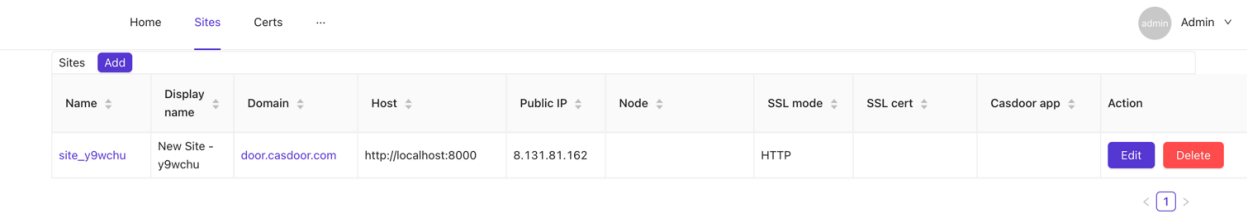
CasWAF supports logging in with Casdoor. You need to register CasWAF as a Casdoor application first.

About how to register CasWAF as a Casdoor application, please refer to Casdoor documentation.

After registering CasWAF as a Casdoor application, you can log in to CasWAF with Casdoor.



After logging in, you can manage the site and certificate according to the instructions provided in the Sites documentation and Cert documentation.



> TODO
>
> add content

Sites

Site List

Learn how to configure your site in CasWAF.

Site List

`Site` is representing the real applications or websites you wish to protect.

This section will provide a detailed explanation of the properties and usage of `Site`.

Site properties

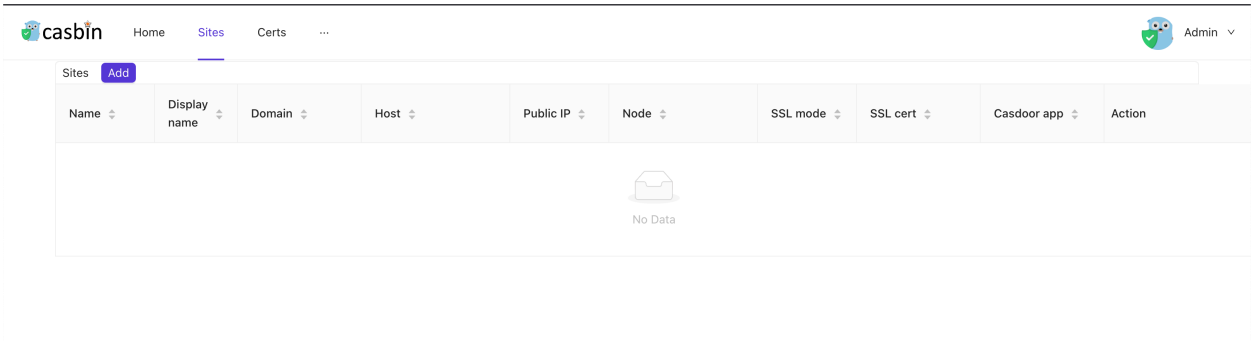
- `Name`: The name of the site.
- `DisplayName`: The display name of the site.
- `Domain`: The domain of the site.
  - e.g. `blog.example.com`
- `Host`: The host of the site.
  - e.g. `localhost:8080`
- `Public IP (Optional)`: The public IP of the site (if available).
- `Node (Optional)`: The name of the host on which the site is deployed.
- `Rules`: The rules used in the site to handle requestions. Users can select the rules from the dropdown list.
  - About how to add rules, please refer to Rule List.
- `SSL mode`: The SSL mode of the site. It can be `HTTP` or `HTTPS and HTTP` or `HTTPS Only`.
  - `HTTP`: The site is not using SSL. Users can access the site **only** via HTTP.
  - `HTTPS and HTTP`: The site is using SSL and HTTP. Users can access the site via **both** HTTP and HTTPS.
  - `HTTPS Only`: The site is using SSL only. Users can access the site **only**

via HTTPS. If users access the site via HTTP, they will be redirected to HTTPS.

- `SSL cert`: The SSL certificate of the site. User can select the SSL certificate from the dropdown list.
  - About how to add SSL certificate, please refer to SSL Certificates.
- `Casdoor app`: If yor site need OAuth login, you can select the Casdoor app from the dropdown list.
  - CasWAF uses Casdoor as the OAuth server. So you need to register your site as a Casdoor app first.
  - About how to add Casdoor app, please refer to Casdoor SSO.

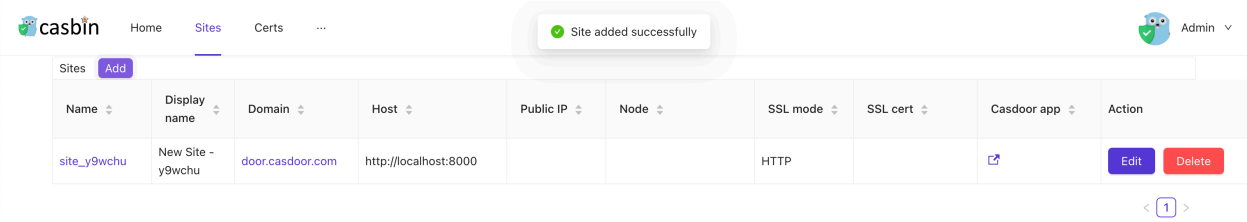
Usage

Manage sites



In the Sites page, you will see all sites you have created. You can create, edit, delete and view the site details. But now we don't have any sites.

Add a site



Just click the `Add` button, you will create a site. The created Site will have some default information that you can modify.

Edit site

After you create a site, you can click the `edit` button to edit the site.

Each field's meaning is as described above, and you can freely modify them according to the actual situation of your website.

After you have completed the modifications, you only need to click the `Save` button to save your settings.

Certs

Cert List

Learn how to add your server's certificates in CasWAF.

Cert List

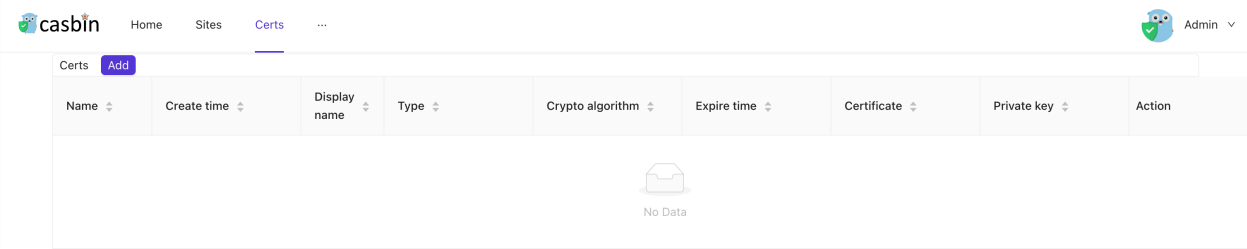
If your website supports HTTPS access, you need to configure an SSL certificate in CasWAF to enable it to establish a secure connection with your server as a reverse proxy.

This section will provide a detailed explanation of the properties and usage of `Cert`.

Cert properties

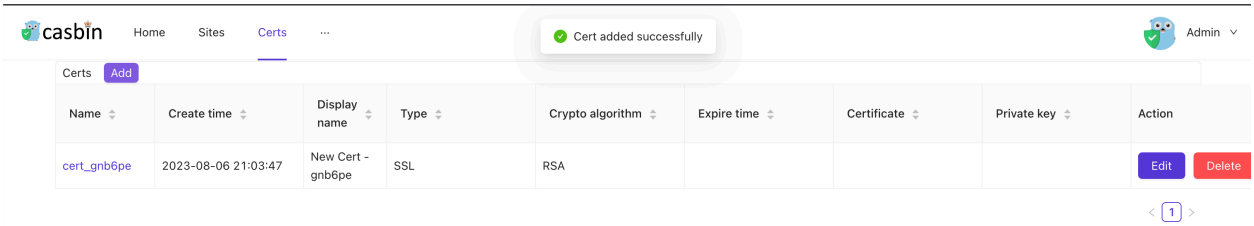
- `Name`: The name of the certificate.
- `Display name`: The display name of the certificate.
- `Type`: The type of the certificate. This field is usually set to `SSL` since SSL is the most commonly used option.
- `Crypto algorithm`: The crypto algorithm of the certificate. It can be set to `RSA` or `ECC` which is depend on algorithm your certificate used.
- `Expire time`: The expire time of the certificate. This field will be automatically filled in when you set certificate and private key.
- `Certificate`: The certificate content.
  - The certificate is a public key that has been authenticated by a Certificate Authority (CA).
- `Private key`: The private key content.
  - The private key is a secret key that is used to encrypt and decrypt data.

Manage Certificates



In the Certs page, you will see all certs you have created. You can create, edit, delete and view the cert details. But now we don't have any certs.

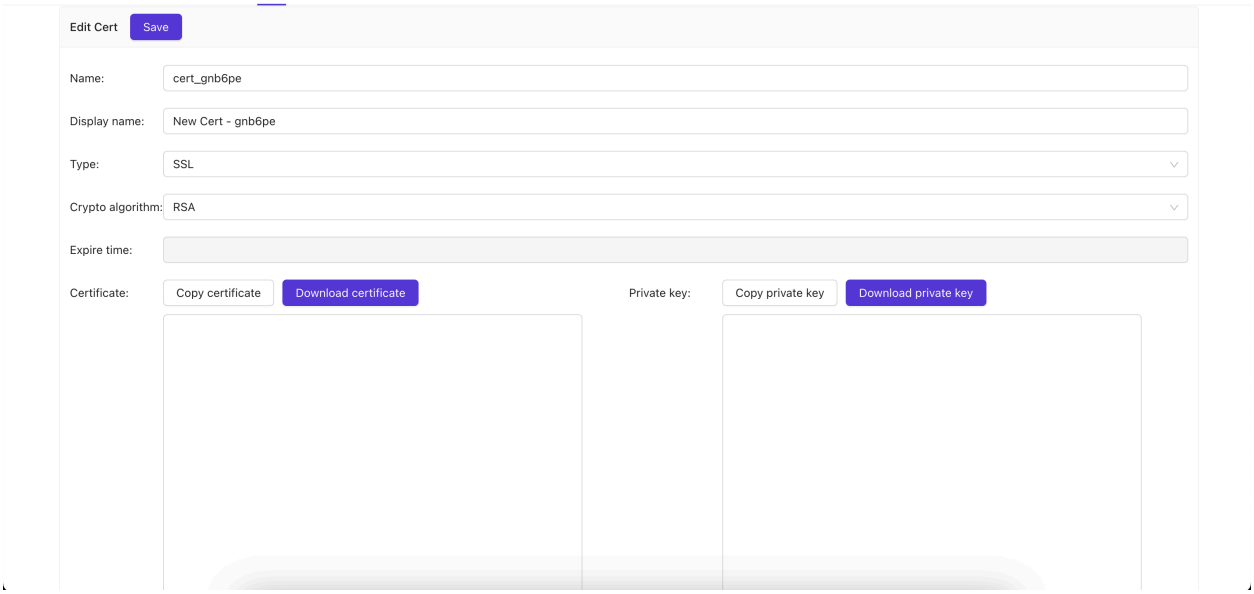
Add a cert



Just click the `Add` button, you will create a cert. The created cert will have some default information that you can modify.

Edit cert

After you create a site, you can click the `edit` button to edit the site.



Each field's meaning is as described above, and you can freely modify them according to the actual situation of your website.

After you have completed the modifications, you only need to click the `Save` button to save your settings.

Rules

Rule List

Learn how to add your server's rule in CasWAF.

IP Rule

Learn how to config IP rules in CasWAF.

User-Agent Rule

Learn how to config User-Agent rules in CasWAF.

WAF Rule

Learn how to config WAF rules in CasWAF.

IP Rate Rule

Learn how to config IP Rate rules in CasWAF.

Compound Rule

Learn how to config Compound rules in CasWAF.

Rule List

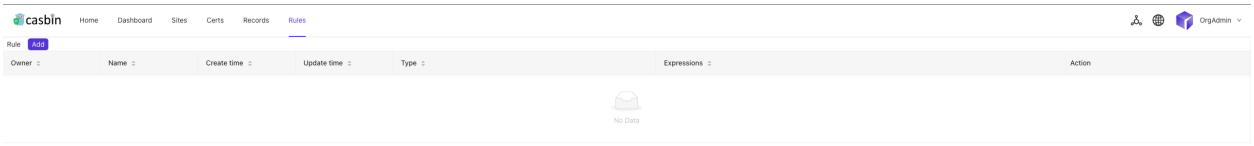
If you want your website under protection, you can add your rules in CasWAF.

This section will provide a detailed explanation of the properties and usage of `Rule`.

Rule properties

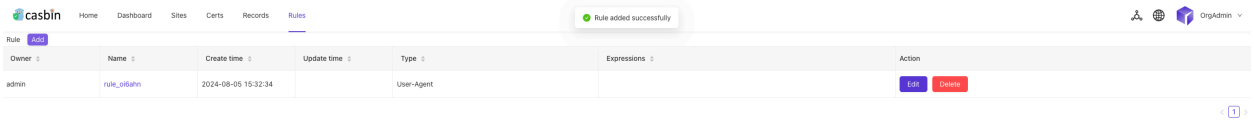
- `Name`: The name of the rule.
- `Type`: The type of the rule. It can be set to `IP`, `User-Agent` and `WAF`.
- `Expressions`: The expressions contains the rule's conditions, including `operator` and `value`.
- `Action`: The action of the rule. It can be set to `Allow` or `Block`.
- `Reason`: The reason of the rule. When the rule is hitted and its action is `Block`, CasWAF would reply a 403 response with the reason.

Manage Rules



In the Rules page, you will see all rules you have created. You can create, edit, delete and view the rule details. But now we don't have any rules.

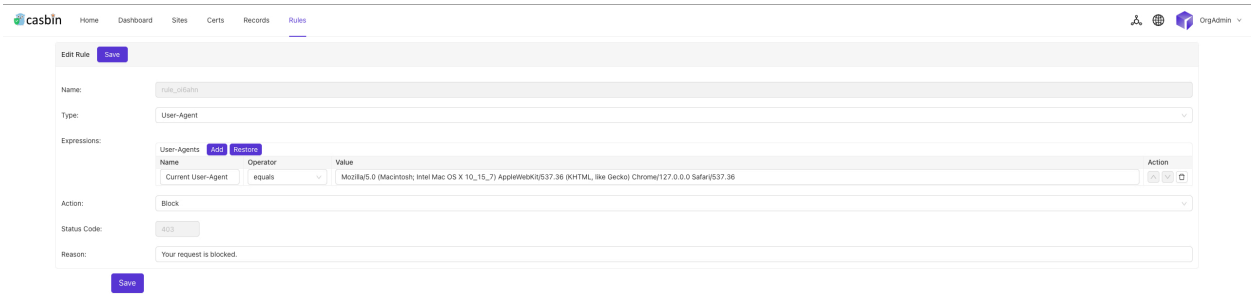
Add a rule



Just click the `Add` button, you will create a rule. The created rule will have some default information that you can modify.

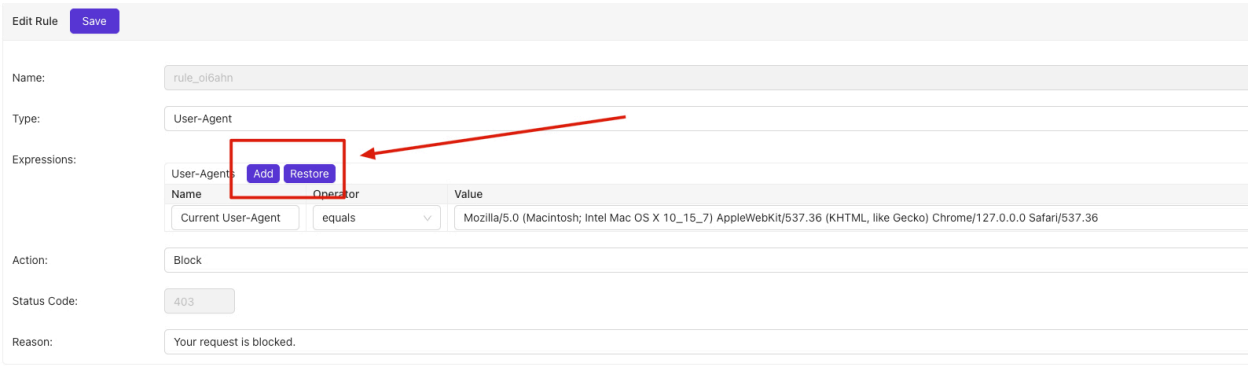
Edit rule

After you create a rule, you can click the `edit` button to edit the rule.



Each field's meaning is as described above, and you can freely modify them according to the actual situation of your website.

Expressions



Click the `Add` button in the `Expressions` section, you Kcan add a new expression to the rule.

Click the `Restore` button in the `Expressions` section, you can restore the default expressions of the rule.

IP Rule

CasWAF provides a powerful IP rule feature to help you control the access of your website. You can add IP rules to allow or block specific IP addresses or IP ranges (CIDR only).

IP Expression properties

- `Name`: The name of the expression. It is used to identify the rule and have no effect on the rule itself.
- `Operator`: The operator of the expression. It can be set to `is in` or `is not in`.
- `Value`: The value of the expression. It can be set to IPv4 or IPv6 addresses or CIDR ranges (even mixed).

User-Agent Rule

CasWAF provides a User-Agent rule feature to help you control the access of your website. You can add User-Agent rules to allow or block specific User-Agents.

User-Agent Expression properties

- `Name`: The name of the expression. It is used to identify the rule and have no effect on the rule itself.
- `Operator`: The operator of the expression. It can be set to `equals`, `does not equals`, `contains`, `does not contain` and `regex match`.
- `Value`: The value of the expression. It can be set to any string. The default value is your current User-Agent. If you are using `regex match`, the value should be a regular expression.

WAF Rule

CasWAF use coraza WAF as the WAF engine, you can follow the coraza seclang documents to write your WAF rules.

WAF Expression properties

- `Value`: The value of the rule. It should be a valid coraza WAF rule.

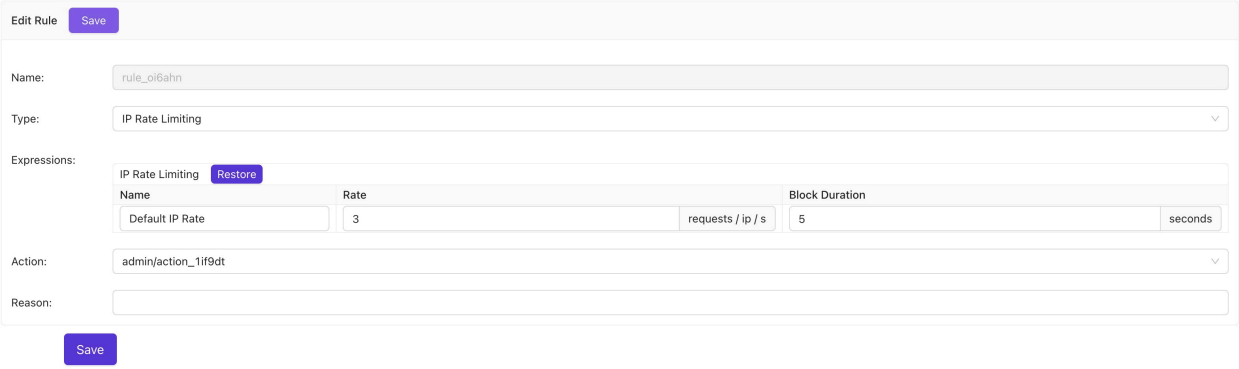
Supported disruptive actions

Seclang action	CasWAF action	Status code
allow	Allow	200
block	Block	403
deny	Deny	403
drop	Drop	400

> **信息**
>
> We recommend that you have a good understanding of the coraza WAF rules before you write your own rules.

IP Rate Rule

CasWAF provides a powerful IP Rate rule feature to help you control the access of your website. You can add IP Rate rules to limit the access rate to your website based on IP addresses.

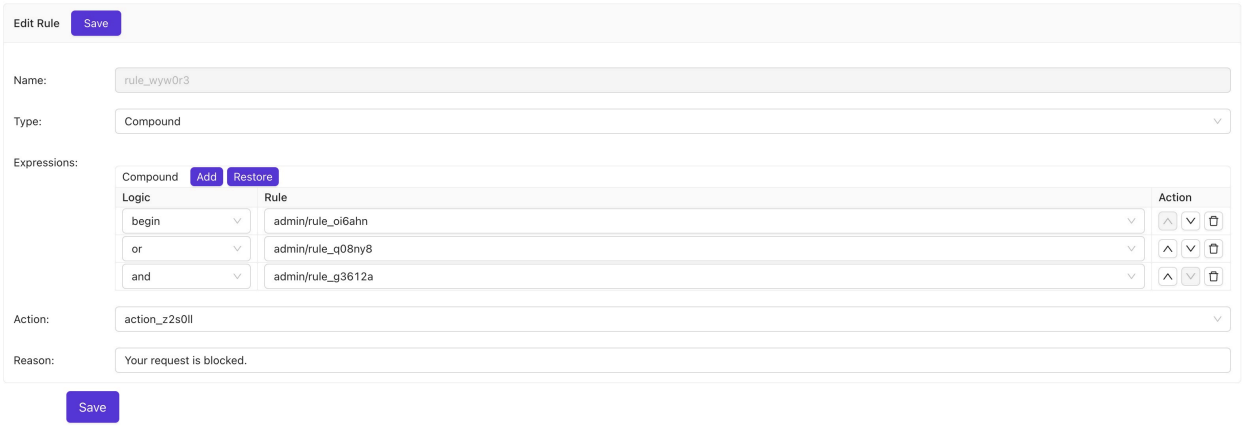


IP Rate Expression properties

- `Name`: The name of the expression. It is used to identify the rule and have no effect on the rule itself.
- `Rate`: The max request rate for a single IP address. It should be a positive integer.
- `Block Duration`: The duration of blocking the IP address when the rate limit is exceeded. It should be a positive integer in seconds.

Compound Rule

CasWAF provides a Compound rule feature to help you control the access of your website. With Compound rules, you can combine multiple rules with logical operators to create complex access control policies.



Compound Expression properties

- `Logic`: The logical operator of the compound rule. It can be set to `and` or `or`, and the first rule in the compound rule will be ignored if the logical operator is `begin`. During expression evaluation, `and` has a higher priority than `or`. For example, `A and B or C` is equivalent to `(A and B) or C`.
- `Rule`: The rules that need to be combined.

Actions

Action List

Learn how to configure your actions in CasWAF.

Allow Action

Learn how to configure your Allow Action in CasWAF.

Block Action

Learn how to configure your Block Action in CasWAF.

Captcha Action

Learn how to configure your Captcha Action in CasWAF.

Action List

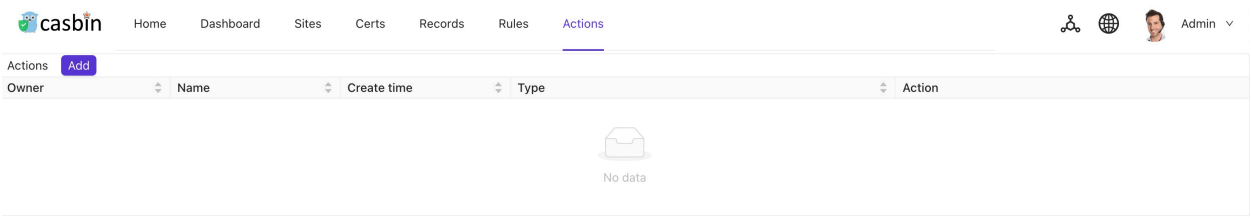
`Action` is related to the rules used in the site to handle requests.

This section will provide a detailed explanation of the properties and usage of `Action`.

Action properties

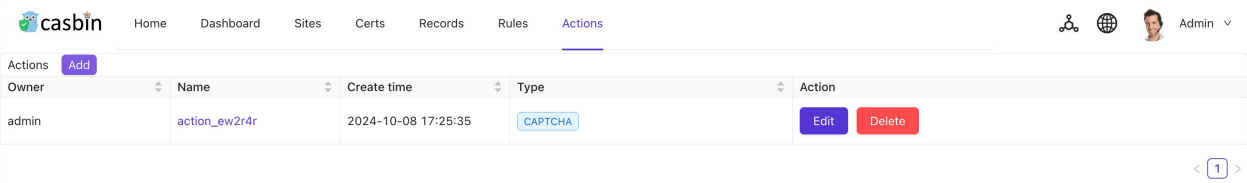
- `Name`: The name of the action.
- `Type`: The type of the action. It can be set to `Allow`, `Block` and `Captcha`.
- `Status Code`: The status code of the action. It can be set to any HTTP status code.

Manage Actions



In the Actions page, you will see all actions you have created. You can create, edit, delete and view the action details. But now we don't have any actions.

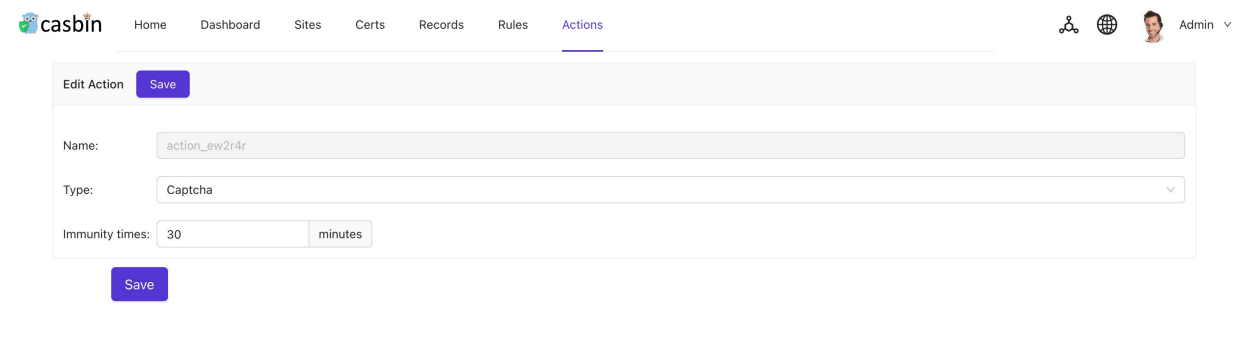
Add an action



Just click the `Add` button, you will create an action. The created action will have some default information that you can modify.

Edit action

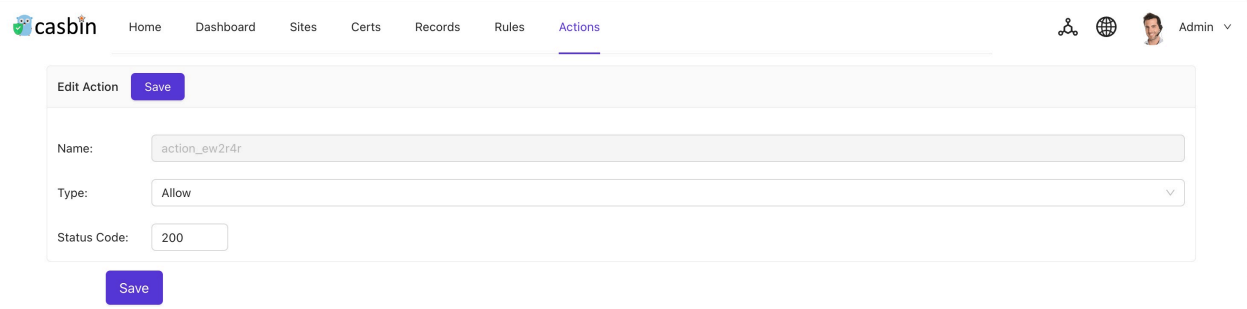
After you create an action, you can click the `edit` button to edit the action.



Each field's meaning is as described above, and you can freely modify them according to the actual situation of your website.

Allow Action

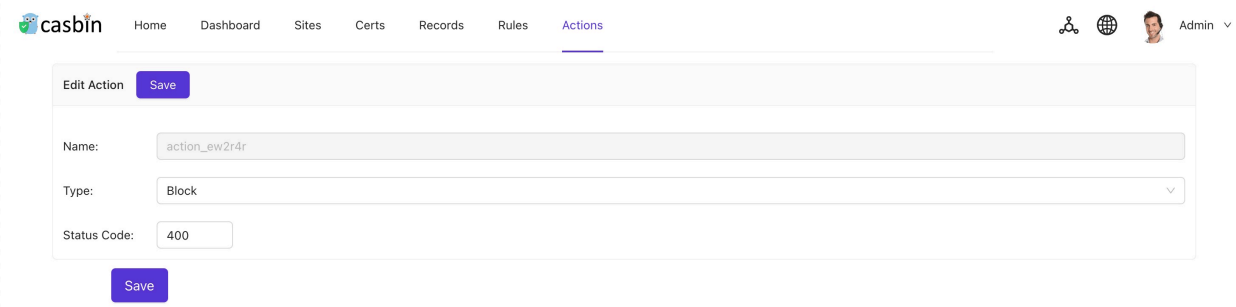
CasWAF provides a easy-to-use Allow Action feature to help you control the access of your website. You can add Allow Actions to allow specific requests to your website.



It is recommended to set the `Status Code` to `200` for the `Allow` action. When the rule is hitted and its action is `Allow`, CasWAF will respond with a status code of `200`.

Block Action

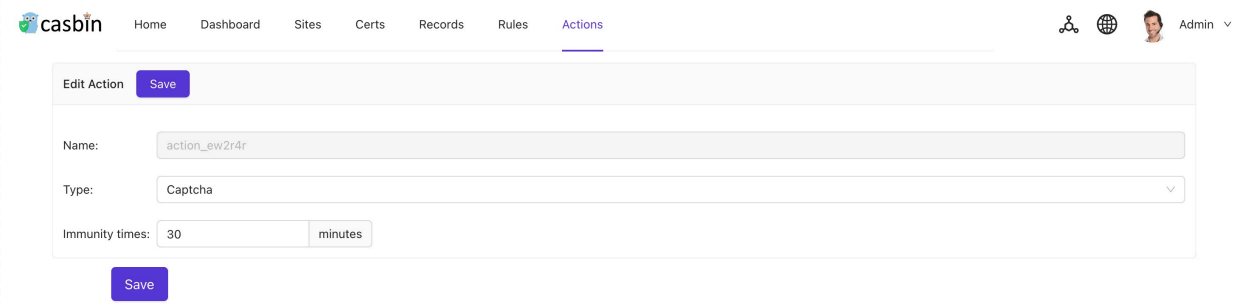
CasWAF provides a easy-to-use Block Action feature to help you control the access of your website. You can add Block Actions to block specific requests to your website.



It is recommended to follow the HTTP status code standard when setting the `Status Code` for the `Block` action. When the rule is hitted and its action is `Block`, CasWAF will respond with the specified status code.

Captcha Action

CasWAF provides a easy-to-use Captcha Action feature to help you control the access of your website. You can add Captcha Actions to require users to pass a CAPTCHA verification before accessing your website.



Before adding a Captcha Action, you need to make sure that you have added a Captcha applications in your Casdoor instance and your Casdoor version is greater than or equal to `v1.681.0`.

Captcha Action properties

- `Immunity times`: Used in the `Captcha` type. The times that the user can access the site without captcha verification after passing the captcha verification. It should be a positive integer.