ALIBABA CLOUD

Alibaba Cloud

Web应用防火墙

Website Protection Settings

Document Version: 20201112

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Overview

This topic describes the website protection features supported by Web Application Firewall (WAF).

Module	Feature	Description	Enabling method	Reference
Web Security	RegEx Protection Engine	The feature protects your websites against common web attacks based on built-in rule groups. The common web attacks include SQL injection, XSS, webshell upload, command injection, backdoor isolation, invalid file requests, path traversing, and common application attacks.	The feature is enabled by default after you add a domain name.	Configure the RegEx Protection Engine Best practices for using RegEx Protection Engine
	Protection Rule Group	The feature allows you to combine protection rules to create a custom rule group and apply the group to specific websites as needed. Note You can create a custom rule group for only RegEx Protection Engine.	You need to enable it after you add a domain name.	Customize protection rule groups Best practices for using custom rule groups to provide enhanced protection
	Big Data Deep Learning Engine	The feature is based on the deep neural network system of Alibaba Cloud. It classifies all web attack data and normal business data in the cloud and then creates a data model. This way, potential attacks can be blocked in real time.	You need to enable it after you add a domain name.	Configure the Big Data Deep Learning Engine
	Website Tamper-proofing	The feature helps you lock specific web pages, such as those that contain sensitive information. When a locked web page is requested, the page cached in WAF is returned. This prevents the tampering of the web pages.	You need to enable it after you add a domain name.	Configure website tamper-proofing
	Data Leakage Prevention	The feature filters content, such as abnormal pages and keywords, returned from the servers to websites and masks sensitive information, such as identity card numbers, bank card numbers, phone numbers, and sensitive words. WAF then returns masked information or default error pages to visitors.	You need to enable it after you add a domain name.	Configure data leakage prevention

Module	Feature	Description	Enabling method	Reference
	Positive Security Model	The feature uses Alibaba Cloud machine learning algorithms to automatically analyze the normal network traffic of a website. It then generates security protection policies tailored for the website based on the collected data.	You need to enable it after you add a domain name.	Configure the positive security model
Bot Management	**Allowed Crawlers**	The feature maintains a whitelist for authorized search engines, such as Google, Bing, Baidu, Sogou, 360, and Yandex. The crawlers of these search engines are allowed to access specified domain names.	You need to enable it after you add a domain name.	Configure the allowed crawlers function
	Bot Threat Intelligence	The feature provides information about suspicious IP addresses of dialers, on-premises data centers, and malicious scanners based on the powerful computing capabilities of Alibaba Cloud. This feature also maintains a dynamic IP library of malicious crawlers and prevents crawlers from accessing your websites or specific directories.	You need to enable it after you add a domain name.	Set a bot threat intelligence rule
	Data Risk Control	The feature protects crucial website services, such as registrations, logons, campaigns, and forums, against fraud.	You need to enable it after you add a domain name.	Configure data risk control
	App Protection	The feature provides secure connections and anti-bot protection for native applications. This feature also identifies proxies, emulators, and requests with invalid signatures.	You need to enable it after you add a domain name.	Configure application protection
	HTTP Flood Protection	This feature helps you defend against HTTP flood attacks and provides protection policies in different modes.	The feature is enabled by default after you add a domain name.	Configure HTTP flood protection Best practices for preventing HTTP flood attacks

Module	Feature	Description	Enabling method	Reference
Access Control/Throttling	IP Blacklist	The feature blocks access requests from specified IP addresses, CIDR blocks, and IP addresses in specified regions.	You need to enable it after you add a domain name.	Configure a blacklist
	Scan Protection	The feature automatically blocks access requests that have specific characteristics. For example, if the source IP address of requests initiates multiple web attacks or targeted directory traversal attacks in a short period of time, WAF automatically blocks the requests. Source IP addresses are also blocked if they are from common scan tools or the Alibaba Cloud malicious IP library.	You need to enable it after you add a domain name.	Configure scan protection
	Custom Protection Policy	The feature allows you to customize ACL rules and configure rate limiting based on precise match conditions.	You need to enable it after you add a domain name.	Create a custom protection policy
Protection Lab	Account Security	The feature allows you to monitor user authentication-related interfaces, such as the endpoints used for registration and logon, and to detect events that may pose a threat to user credentials. These threats include credential stuffing, brute-force attacks, spam registration, weak password sniffing, and SMS flood attacks.	You need to enable it after you add a domain name.	Configure account security Account security best practices
	API Request Security	The feature allows you to upload a custom API rule file to execute only requests that comply with the rules. This protects your website assets against threats such as tampering and replay attacks.	You need to enable it after you add a domain name.	Enable API request security
	Website Whitelisting	After you configure a rule, requests that match the rule bypass all protection features and are directly forwarded to origin servers.	You need to enable it after you add a domain name.	Configure a website whitelist

Module	Feature	Description	Enabling method	Reference
Whitelists	**Whitelisting Rules in Web Intrusion Prevention**	After you configure a rule, requests that match the rule bypass specified protection features, such as RegEx Protection Engine and Big Data Deep Learning Engine.	You need to enable it after you add a domain name.	Configure a whitelist for Web Intrusion Prevention
	Whitelisting Rules in Data Security	After you configure a rule, requests that match the rule bypass specified protection features, such as website tamper-proofing, data leak prevention, and account security.	You need to enable it after you add a domain name.	Configure a whitelist for Data Security
	Whitelisting Rules in Bot Management	After you configure a rule, requests that match the rule bypass specified protection features, such as bot threat intelligence, data risk control, intelligent algorithm, and application protection.	You need to enable it after you add a domain name.	Configure a whitelist for Bot Management
	Whitelisting Rules in Access Control/Throttling	After you configure a rule, requests that match the rule bypass specified protection features, such as HTTP flood protection, blacklist, scan protection, and custom protection policy.	You need to enable it after you add a domain name.	Configure a whitelist for Access Control/Throttling

2.Web security

2.1. Configure the RegEx Protection Engine

After you add a website to Web Application Firewall (WAF), the RegEx Protection Engine is enabled by default. The RegEx Protection Engine is based on built-in expert rule groups. It automatically protects the website against common application vulnerability attacks such as SQL injection, XSS cross-site, webshell upload, command injection, backdoor isolation, invalid file requests, path traversing, common web attacks. You can adjust the protection policies of the RegEx Protection Engine as needed.

Prerequisite

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Web Security** tab, find the **RegEx Protection Engine** section, and then configure the following parameters.

Parameter	Description
Status	Enable or disable the RegEx Protection Engine. By default, the RegEx Protection Engine is enabled after you add a website to WAF. **Note** After the RegEx Protection Engine is enabled, all requests destined for the website are checked by the function. You can configure a Web Intrusion Prevention rule so that requests that meet the rule bypass the check. For more information, see Configure a whitelist for Web Intrusion Prevention.
Mode	The action that is taken on attack requests when they are detected. Valid values: ○ **Block**: blocks requests. ○ **Warn**: triggers alerts but does not block the attack requests.

Parameter	Description
Protection Rule Group	The protection rule group you want to use. Built-in rule groups and custom rule groups are supported. Built-in rule groups include: ○ **Medium rule group**: This rule group detects common web application attacks and default applications in a standard way. This rule group is applied by default. ○ **Strict rule group**: This rule group detects web application attacks such as path traversal, SQL injections, and command executions in a strict way. ○ **Loose rule group**: This rule group detects common web application attacks in a loose way. If you find a high false positive rate with the medium rule group or your business has a high amount of uncontrollable user input such as rich text editors and technical forums, we recommend that you select this rule group. Click **Settings** to go to the **Protection Rule Group** page. On this page you can create custom rule groups or select built-in rule groups as needed. For more information, see Customize protection rule groups.
Decoding Settings	The data formats that need to be decoded and analyzed by the RegEx Protection Engine. To ensure higher performance, the RegEx Protection Engine decodes and analyzes the request content of all formats by default. If the RegEx Protection Engine blocks normal requests that contain content of specified formats, you can clear the format to reduce the false positive rate. Procedure i. Unfold the configuration menu.  ii. Select or clear the format that you want to decode. ■ You cannot clear the following formats: **URL Decoding**, **JavaScript Unicode Decoding**, **Hex Decoding**, **Comment Processing**, and **Space Compression**. ■ You can clear the following format: **Multipart Data Parsing**, **JSON Data Parsing**, **XML Data Parsing**, **Serialized PHP Data Decoding**, **HTML Entity Decoding**, **UTF-7 decoding**, **Base64 Decoding**, and **Form Data Parsing**. iii. Click **Confirm**.

2.2. Configure the Big Data Deep Learning Engine

After you add a website to Web Application Firewall (WAF), you can enable the Big Data Deep Learning Engine for your website. The Big Data Deep Learning Engine is based on the deep neural network system of Alibaba Cloud. It performs classification training on all web attack data and normal business data in the cloud. This way, potential attacks can be blocked in real time. You can adjust the protection policies of the Big Data Deep Learning Engine based on your requirements.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
  - The instance is deployed in **mainland China**.

    > **Note** Instances that are deployed **outside mainland China** do not support the Big Data Deep Learning Engine.

  - The instance must be of the **Business** edition or higher. For more information, see Editions and features.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Web attack methods keep evolving as the Internet develops. Traditional single-method protection no longer meets the security requirements of complex Internet services. Collaborative protection that uses multiple detection engines is more powerful.

Based on massive operations data of Alibaba Cloud, the Big Data Deep Learning Engine trains models for normal web applications and identifies abnormalities from these models. It also refines attack models from various web application attacks. The Big Data Deep Learning Engine uses these models to detect zero-day vulnerabilities. It also blocks potential attacks online in real time to make up for the deficiencies of other protection engines. When WAF is used to prevent web attacks, protected traffic is forwarded to the RegEx Protection Engine. Then, the traffic is forwarded to the Big Data Deep Learning Engine. The two engines complement each other.

Scenarios

The Big Data Deep Learning Engine targets web requests with weak attack characteristics rather than HTTP flood attacks. If you have more precise requirements on web attack protection, we recommend that you enable the Big Data Deep Learning Engine.

The RegEx Protection Engine uses strong regular expression rules. It provides optimal protection against requests with strong attack characteristics. The RegEx Protection Engine may fail to detect potential risks from requests with weak attack characteristics, such as cross-site scripting (XSS) attacks. It may also fail to detect these attacks even in strict mode. In this case, you can enable the Big Data Deep Learning Engine to identify and block requests with weak attack characteristics that cannot be detected based on strict rules of the RegEx Protection Engine.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Web Security** tag, find the **Big Data Deep Learning Engine** section, and configure following parameters.

Big Data Deep Learning Engine

Parameter	Description
Status	Enable or disable the Big Data Deep Learning Engine. **Note** After the Big Data Deep Learning Engine is enabled, all requests destined for your website are checked by the function. You can configure a Web Intrusion Prevention rule so that requests that meet the rule bypass the check. For more information, see Configure a whitelist for Web Intrusion Prevention.
Mode	The action that is taken on attack requests when they are detected. Valid values: ○ **Block**: blocks requests. ○ **Warn**: triggers alerts but does not block requests.
Attack Probability	Threshold of the probability that a request is identified as an attack under deep learning. The value is an integer ranging from 50 to 100. If the parameter value is large, the standard for determining that a request is an attack is strict, and the Big Data Deep Learning Engine blocks real attacks more accurately. However, the engine may also leave more potential risks unblocked. If the parameter value is small, the standard for determining that a request is an attack is not strict, and the Big Data Deep Learning Engine blocks more suspicious requests. However, the engine may also block some normal requests.

2.3. Configure website tamper-proofing

After you add a website to Web Application Firewall (WAF), you can enable the website tamper-proofing function to protect the website from website defacement. Website tamper-proofing helps you lock specific web pages, such as those that contain sensitive information. When a locked web page is requested, the page cached in WAF is returned. This prevents web pages from being maliciously modified. You can customize website tamper-proofing rules as needed.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
    - If the instance is deployed in **mainland China**, the instance must be of the **Pro** edition or higher.
    - If the instance is deployed **outside mainland China**, the instance must be of the **Enterprise** edition or higher.

    For more information, see Editions and features.

For more information, see Purchase a WAF instance.

- Your website is added to the WAF console. For more information, see Add websites.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Web Security** tab and find the **Website Tamper-proofing** section. Then, turn on **Status** and click **Settings**.

> Note
> - You must enable website tamper-proofing before you create protection rules.
> - After website tamper-proofing is enabled, all requests destined for your website are checked by the function. You can configure a Data Security rule so that the requests that match the rule bypass the check. For more information, see Configure a whitelist for Data Security.

6. Create a tamper-proofing rule.
   i. On the **Website Tamper-proofing** page, click **Add Rule**.
   ii. In the **Add Rule** dialog box, specify the **Service Name** and **URL** parameters of the web page that you want to protect.
   - **Service Name**: Specify the name of the service that the web page provides.
   - **URL**: Enter an exact path. Wildcard characters such as `/*`, or parameters such as `/abc? xxx=` are not supported. Text data, HTML pages, and images under the specified path are protected.

   iii. Click **Confirm**.

   After a tamper-proofing rule is created, it is disabled by default. You can find the newly created rule in the rule list and **Protection Status** of the rule is turned off.

7. Enable the rule. Find the rule you want to enable in the rule list and turn on **Protection Status**.

   After the rule is enabled, if the specified web page is requested, the page cached in WAF is returned.

8. (Optional)Update cached data. Find the rule that is enabled in the rule list and click **Refresh Cache** in the **Protection Status** column.

> **Notice** If the protected web page is updated, you must click **Refresh Cache** to update the data cached in WAF. If you do not update the cached data after a page is updated, WAF returns the most recent page stored in the cache.

2.4. Configure data leakage prevention

After you add a website to Web Application Firewall (WAF), you can enable data leakage prevention for the website. Data leakage prevention filters content (abnormal pages and keywords) returned from the servers, and mask sensitive information, such as identity card numbers, bank card numbers, phone numbers, and sensitive words. WAF then returns masked information or default response pages to visitors. You can customize data leakage prevention rules as needed.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
    - If the instance is deployed in **mainland China**, the instance must be of the **Pro** edition or higher.
    - If the instance is deployed **outside mainland China**, the instance must be of the **Business** edition or higher.

    For more information, see Editions and features.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

WAF provides the data leakage prevention function to comply with the following regulations required by Cybersecurity Law of the People's Republic of China: Network operators shall adopt technological and other necessary measures to ensure the security of personal information they collect, and prevent information leaks, damage or loss. Where a situation of information leak, damage or loss occurs, or might occur, they shall promptly take remedial measures, timely notify users and report the matter to the authority according to regulations. Data leakage prevention masks sensitive information (phone numbers, identity card numbers, and bank card numbers) in website content and triggers alerts upon sensitive information. You can also use data leakage prevention to block a specific HTTP status code.

Features

Information maintained by a website may be leaked in the following scenarios: allowing unauthorized access to a URL, such as access to the backend management system, horizontal and vertical privilege escalation, and malicious crawlers retrieving sensitive information from web pages. To prevent common sensitive information leak, data leakage prevention provides the following functions:

- Detects and identifies personal information on web pages, masks the information, and triggers alerts to protect website data. Personal information includes but is not limited to identity card numbers, phone numbers, and bank card numbers.

  > **Note** Only phone numbers and landline numbers in mainland China can be identified.

- Masks sensitive server information, including web applications used by the website, the operating system, and the version of the server.

- Maintains a library that contains illicit and sensitive keywords to detect and mask illicit or sensitive website content, and trigger alerts.

How data leakage prevention works

Based on the specified protection rules, data leakage prevention detects whether a web page contains sensitive information, such as identity card numbers, phone numbers, and band card numbers. If a rule is matched, WAF triggers alerts or masks the information based on the action specified in the rule. Data leakage prevention masks sensitive information with asterisks (*).

Data leakage prevention allows you to set Content-Type to `text/*` , `image/*` , or `application/*` to protect web applications, native applications, and API operations.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Web Security** tab and find the **Anti Sensitive Information Leakage** section. Then, turn on **Status** and click **Settings**.

   > **Note**
   > - You must enable data leakage prevention before you can set protection rules.
   > - When the data leakage prevention function is enabled, all requests destined for the website are checked by the function. You can configure a Data Security rule so that the requests that match the rule bypass the check. For more information, see Configure a whitelist for Data Security.

6. Create a data leakage prevention rule.
   i. On the **Anti Sensitive Information Leakage** page, click **Add Rule**.
   ii. In the **Create Rule** dialog box that appears, configure the following parameters.

Parameter	Description
Rule name	The name of the rule that you want to create.

Parameter	Description
Matching conditions	The types of information that you need to detect. Supported types include: ■ **Status Code**: 400, 401, 402, 403, 404, 500, 501, 502, 503, 504, 405 to 499, and 505 to 599. ■ **Sensitive Info**: **ID Card**, **Credit Card**, **Telephone No.**, and **Default Sensitive Word** **Note** **Telephone No.** supports only phone numbers and landline numbers in mainland China. You can specify one or more HTTP status codes or sensitive information types. If you select the **and** check box, you can specify the **URL** that you want to check. In this case, WAF scans for sensitive information on only the specified page.
Matching Action	The action to be performed on detected sensitive information. ■ If you set the match condition to **Status Code**, supported actions include: ■ **Warn**: triggers alerts upon sensitive information leaks. ■ **Block**: blocks requests and returns the default page indicating that your requested website is blocked. ■ If you set the match condition to **Sensitive Info**, supported actions include: ■ **Warn**: triggers alerts upon sensitive information leaks. ■ **Sensitive information filtering**: masks sensitive information in responses.

Sample configurations

- Mask sensitive information: Web pages may contain sensitive information, such as phone numbers and identity card numbers. You can create rules to mask or trigger alerts upon sensitive information. The following example shows how to create a rule that masks phone numbers and identity card numbers.
  - Matching conditions: ID Card and Telephone No.
  - Matching Action: Sensitive information filtering

  After this rule is applied, all phone numbers and identity card numbers on the website are masked, as shown in the following figure.

  Notice Phone numbers that must be provided to the public to manage business affairs, such as business cooperation and complaints, may also be masked by data leakage prevention rules.

- Block HTTP status codes: You can create a rule to block or generate alerts upon specific HTTP status codes to prevent sensitive server information leaks. The following example shows how to create a rule that blocks the 404 HTTP code.
  - Matching conditions: 404
  - Matching Action: Block

  After this rule is applied, if the requested page does not exist, the specified page indicating that your requested website is blocked is returned, as shown in the following figure.
- Mask specific sensitive information on specific pages: You can create rules to mask or generate alerts upon specific sensitive information, such as phone numbers and identity card numbers, on specific pages. The following example shows how to create a rule that masks identity card numbers on pages whose URLs contain `admin.php`.
  - Matching conditions: ID card numbers on pages whose URLs contain `admin.php`
  - Matching Action: Sensitive information filtering

  After this rule is applied, identity card numbers on pages whose URLs contain admin.php are masked.

iii. Click **Confirm**.
After a data leakage prevention rule is created, it takes effect automatically. You can view newly created rules, and modify or delete rules in the rule list as needed.

What's next

After you enable data leakage prevention, you can view log data of filtered or blocked requests that triggered data leakage prevention rules. To view the log data, navigate to the **Security report** page and choose **Web Security** > **Data Leakage Prevention** to view the relevant security report. For more information, see View security reports.

2.5. Configure the positive security model

After you add a website to Web Application Firewall (WAF), you can enable the positive security model for your website. The positive security model of WAF uses Alibaba Cloud machine learning algorithms to automatically learn normal network traffic of a website. The positive security model then generates security rules tailored for the website based on the collected data. You can adjust the protection mode and rules of the positive security model based on your requirements.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
  - The instance is deployed in **mainland China**.

    > **Note** Instances deployed **outside mainland China** do not support the positive security model.

  - The instance must be of the **Enterprise** edition or higher. For more information, see Editions and features.

  For more information, see Purchase a WAF instance.

- Your website is added to the WAF console. For more information, see Add websites.

Background information

Traditional protection methods against web attacks are based on detection rules. The positive security model automatically learns the network traffic of a website and uses machine learning algorithms to generate a standard security score and grade different requests. Based on the request scores, the positive security model defines the baseline traffic of a website and customizes security policies for the website. The positive security model collaborates with other detection modules of WAF to detect attacks at different network layers.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.
5. On the **Web Security** tab, find the **Positive Security Model** section and configure the following parameters.

   Positive Security Model

Parameter	Description
Status	Enable or disable the positive security model.
Mode	The action that is taken on attack requests when they are detected. Valid values: ○ **Warn**: triggers alerts but does not block requests. ○ **Block**: blocks requests. **Note** By default, the positive security model is set to the warn mode. In this mode, WAF only reports requests that match the security rules but does not block the requests. We recommend that you study the data in security reports to make sure that the security rule does not cause false positives before you set the mode to Block.

   If this is your first time enabling the positive security model for a website, WAF automatically learns the network traffic history of the website based on machine learning algorithms. WAF then customizes security rules to protect the website. The initial machine learning process may take a long time to complete based on the total amount of network traffic data. In most cases, it takes about one hour for WAF to learn the network traffic history of the website and generate security rules. After WAF completes the learning process, it notifies you by using internal messages, text messages, and emails.

3.Bot management

3.1. Configure the allowed crawlers function

The allowed crawlers function maintains a whitelist of authorized search engines, such as Google, Bing, Baidu, Sogou, 360, and Yandex. The crawlers of these search engines are allowed to access all pages on domain names.

Prerequisites

- A WAF instance is purchased and the instance meets the following requirements:
  - The instance is billed on a subscription basis.
  - **Bot Management** is enabled. This feature is a value-added service.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Rules defined in the function allow requests from specific crawlers to the target domain name based on the Alibaba Cloud crawler library. The Alibaba Cloud crawler library is updated in real time based on the analysis of network traffic that flows through Alibaba Cloud, and captures the characteristics of requests that are initiated from crawlers. The crawler library is updated dynamically and contains crawler IP addresses of mainstream search engines, including Google, Baidu, Sogou, 360, Bing, and Yandex.

After you enable the allowed crawlers function, requests initiated from the crawler IP addresses of the authorized search engines are directly sent to the target domain names. The bot management module no longer detects these requests.

> **Note** To filter some requests from the crawler IP addresses, use the **Access Control/Throttling** module. For more information, see Create a custom protection policy.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. Click the **Bot Management** tab, find the **Allowed Crawlers** section. Then, turn on **Status** and click **Settings**.

6. In the **Allowed Crawlers** list, find the target rule by **Intelligence Name**, and turn on **Status**.

   The default rules only allow crawler requests from the following search engines: Google, Bing, Baidu, Sogou, 360, and Yandex. You can enable the **Legit Crawling Bots** rule to allow requests from all search engine crawlers.

3.2. Set a bot threat intelligence rule

Bot threat intelligence provides information about suspicious IP addresses used by dialers, on-premises data centers, and malicious scanners. This function also maintains an IP address library of malicious crawlers and prevents crawlers from accessing all pages under your domain name or specific directories.

Prerequisites

- A WAF instance is purchased and the instance meets the following requirements:
  - The instance is billed on a subscription basis.
  - **Bot Management** is enabled. This feature is a value-added service.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Bot threat intelligence rules can block requests from crawlers that are recorded in the Alibaba Cloud crawler library. The Alibaba Cloud crawler library is updated in real time based on the analysis of network traffic that flows through Alibaba Cloud. It can effectively detect IP addresses of malicious crawlers and provide the characteristics of requests that are initiated from the crawlers.

> **Note** The Alibaba Cloud crawler library covers public clouds and on-premises data centers.

You can set a bot threat intelligence rule that chooses different actions to manage different requests based on the type of the threat intelligence library. For example, you can set a rule that blocks certain requests, or requires JavaScript verification or CAPTCHA verification to verify certain requests. You can also use a bot threat intelligence rule to protect important endpoints against certain threats. This helps you minimize the negative impacts on the services.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Bot Management** tab, find the **Bot Threat Intelligence** section. Then, turn on **Status** and click **Settings**.

> **Note** After the bot threat intelligence function is enabled, all requests destined for your website are checked by the function. You can configure a Bot Management rule so that the requests that match the rule bypass the check. For more information, see Configure a whitelist for Bot Management.

Bot Threat Intelligence

6. In the **Bot Threat Intelligence** rule list, find the threat intelligence library you want to use by **Intelligence Name**, and turn on **Status**.

Bot threat intelligence rules

The following table lists the bot threat intelligence libraries that WAF supports.

Intelligence library	Description
Malicious Scanner Fingerprint Blacklist	This library contains characteristics of common scanners.
Malicious Scanner IP Blacklist	This library contains malicious IP addresses that are dynamically updated based on the source IP addresses of scan attacks detected on Alibaba Cloud.
Credential Stuffing IP Blacklist	This library contains malicious IP addresses that are dynamically updated based on the source IP addresses of credential stuffing and brute-force attacks detected on Alibaba Cloud.
Fake Crawler Blacklist	This library identifies crawlers that use the User-Agent of authorized search engines, such as BaiduSpider, to disguise as authorized programs. **Notice** Before you enable this library, make sure that you have configured a whitelist of crawlers. Otherwise, false positives may occur. For more information, see Configure the allowed crawlers function.
Malicious Crawler Blacklist	This library contains malicious IP addresses that are dynamically updated based on the source IP addresses of crawlers detected on Alibaba Cloud. This library is categorized into three severity levels: low, medium, and high. A higher severity indicates more IP addresses in the library and a higher false positive rate. **Note** We recommend that you set up two-factor authentication, such as CAPTCHA and JavaScript verification, for the high-severity library. In scenarios where two-factor authentication cannot be implemented, we recommend that you set threat intelligence rules based on the low-severity library.

Intelligence library	Description
IDC IP List	This library contains IP addresses of public clouds and on-premises data centers, including Alibaba Cloud, Tencent Cloud, Meituan Open Services, and 21Vianet. Attackers typically use CIDR blocks of public clouds or on-premises data centers to deploy crawlers or as proxies to access sites. Regular users rarely access sites in this way.

After you enable the default rule, requests initiated from IP addresses in the threat intelligence library to any directory of the protected domain name trigger the **Monitor** action. This action allows the requests to the destination directories and records the events.

If you need to modify the default rule, such as the protected URL or action, see the following step on how to customize a threat intelligence rule.

7. (Optional)Customize a threat intelligence rule.
   i. Find the target rule and click **Edit** in the Actions column.
   ii. In the **Edit Intelligence** dialog box that appears, set the following parameters.

Parameter	Description
Protected Path	The **URL** that you want to protect, such as /abc, /login/abc, or forward slash (/) that indicates all directories. You also need to select a value for **Matching**. Valid values: ■ **Precise Match**: The destination URL must be an exact match of the protected URL. ■ **Prefix Match**: The prefix of the destination URL matches the protected URL. ■ **Regex Match**:The destination URL matches the specified regular expression. You can click **Add Protected URL** to add more URLs. You can add up to 10 URLs.

Parameter	Description
Action	The action to be performed after the match conditions of the rule are met. Valid values: ■ **Monitor**: allows the request to the destination directory and records the event. ■ **Block**: blocks the request. ■ **JavaScript Validation**: requires JavaScript verification. Requests are forwarded to the destination directory only after they pass the verification. ■ **Captcha**: requires CAPTCHA verification on the client side. Requests are forwarded to the destination directory only after they pass the verification. **Note** CAPTCHA only supports synchronous requests. To verify asynchronous requests, such as Ajax requests, contact the Alibaba Cloud security team. If you cannot determine whether the protected URL supports CAPTCHA, we recommend that you create a custom protection policy, such as an ACL rule, to run a test. For more information, see Create a custom protection policy. ■ **Strict Captcha**: requires CAPTCHA verification on the client side. Requests are forwarded to the destination directory only after they pass the verification. CAPTCHA verification has a stricter standard to verify visitor identities.

iii. Click **Confirm**.

3.3. Configure data risk control

After you add a website to Web Application Firewall (WAF), you can enable data risk control. Data risk control helps you protect crucial website services, such as registrations, logons, campaigns, and forums, against attacks. You can customize data risk control rules based on your requirements.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
    - The instance is deployed in **mainland China**.

Note Instances deployed **outside mainland China** do not support data risk control.

  - The **Bot Management** feature is enabled.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Data risk control is based on Alibaba Cloud big data. It uses industry-leading engines for risk decision-making and integrates human-machine identification technologies to protect crucial services in different scenarios against attacks. To use data risk control, you only need to add your website to WAF, without the need to configure the server or client.

Data risk control is suitable for a wide array of scenarios, including but not limited to: spam registration,SMS verification code abuse, credential stuffing, brute-force attacks, fraud in flash sales, second kills, bargain manipulation, red envelope lucky draws, ticket snapping by using bots, vote rigging, and spamming.

The following figure shows how data risk control protects your website. For more information about scenarios and protection effects of data risk control, see Examples.

Compatibility

Data risk control is supported only by web pages or HTML5 environments. In some cases, the JavaScript plug-in inserted into web pages may be incompatible with the web pages and this causes errors during CAPTCHA verification. Web pages that may encounter incompatibility with data risk control include:

- Static web pages that can be accessed by using their URLs, such as HTML details page, shared pages, website homepages, and documents. Web pages that are redirected by modifying `location.href` or by using the `window.open` method and the anchor tag (<a>).
- Web pages where you can rewrite service code and submit requests by using request methods or custom methods, such as submitting forms, rewriting XMLHttpRequest (XHR), and sending custom Ajax requests.
- Service code that makes use of webhooks.

After you enable data risk control, we recommend that you choose the warn mode and use data risk control together with Log Service for WAF to run a compatibility test. For more information, see Overview.

If data risk control is incompatible with your website, use Alibaba Cloud Human-Machine Validation together with WAF to protect your website.

To protect native applications, we recommend that you use the Anti-Bot SDK. For more information, see Configure application protection.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. Click the **Bot Management** tab, find the **Data Risk Control** section, and then click **Settings**.

Data Risk Control

Parameter	Description
Status	Enable or disable data risk control. After you enable data risk control for a website, WAF inserts a JavaScript plug-in into specific or all web pages of the website. Reactive elements in the web pages are returned to visitors as compressed files that are not in the GZIP format. Even if your website uses non-standard ports, no further configurations are required. **Note** ◦ You must enable data risk control before you set the mode and configure protection rules. ◦ When data risk control is enabled, all requests destined for your website are checked by the function. You can configure a Bot Management rule so that the requests that match the rule bypass the check. For more information, see Configure a whitelist for Bot Management.
Mode	The mode for data risk control. Valid values: ◦ **Strict Interception**: If WAF detects that your website is under attack, requests are required to pass a strict multi-factor authentication. ◦ **Block**: If WAF detects that your website is under attack, requests are required to pass a multi-factor authentication. ◦ **Warn**: If WAF detects that the website is under attack, requests are forwarded to your website but events related to the requests are recorded. You can view detailed information in risk reports. **Note** The Mode parameter is set to Warn by default. In this mode, data risk control does not block requests. However, WAF inserts a JavaScript plug-in into static web pages to analyze client behaviors.

6. Add a protection request for data risk control.
   i. On the **Data Risk Control** page, click the **Protection Request** tab and click **Add Protection Request**.
   ii. In the **Add Protection Request** dialog box, enter the URL that you want to protect in the **Protection Request URL** field.For more information, see Introduction to a protected URL in a protection request.

   Add a protection request for data risk control

   iii. Click **Confirm**.

   A newly added protection request takes effect in about 10 minutes. You can view the newly added protection requests in the request list, and modify or delete them based on your requirements.
7. (Optional)Specify the web page into which you want to insert the JavaScript plug-in.Some code of web pages may be incompatible with the JavaScript plug-in. In this case, we recommend that you insert the JavaScript plug-in into specific pages that are compatible with the plug-in.

> **Note** When the JavaScript plug-in is inserted into specific web pages, data risk control may fail to obtain all visitor behaviors. This affects the effectiveness of data risk control.

i. On the **Data Risk Control** page, click the **Insert JavaScript into Webpage** tab.

ii. Select **Insert JavaScript into Specific Webpage** and click **Add Webpage**.

> **Note** You can insert the JavaScript plug-in into a maximum of 20 web pages.

iii. In the **Add URL** dialog box, enter the URL into which you want to insert the JavaScript plug-in and click **Confirm**. The URL must start with a forward slash (/).

After you add the URL, data risk control inserts the JavaScript plug-in into all web pages under this URL.

After data risk control is enabled, you can use Log Service for WAF to monitor the protection effect. For more information, see View protection results.

Introduction to a protected URL in a protection request

A protected URL in a protection request is the endpoint where operations are performed on a service. It is not the URL of the web page. As shown in the following figure, the URL of the registration page is `www.abc.com/new_user`. The endpoint where you can obtain verification codes is `www.abc.com/getsmscode`, and the endpoint where you can register is `www.abc.com/register.do`.

In this example, you must add protection requests to protect the endpoints `www.abc.com/getsmscode` and `www.abc.com/register.do`. This way, WAF protects these URLs from SMS interface abuse and spam registration. If you set the protected URL to `www.abc.com/new_user`, regular visitors are also required to pass CAPTCHA verification, which may adversely affect the user experience.

Usage notes of protected URLs

- The protected URL must be exact. Fuzzy match is not supported.

  If you set the protected URL to `www.test.com/test`, data risk control filters only requests that are sent to this URL. Data risk control does not filter requests sent to the sub-directories of this URL.

- Data risk control protects website directories.

  If you set the protected URL to `www.abc.com/book/*`, data risk control filters the requests that are sent to all web pages under `www.abc.com/book`. We recommend that you do not set data risk control to monitor the entire website. If you set the protected URL to `www.abc.com/*`, regular visitors need to pass CAPTCHA verification to visit the website homepage. This adversely affects the user experience.

- Requests that are sent to a protected URL always trigger CAPTCHA verification. Make sure that the protected URL is not directly requested by regular visitors in normal cases. Otherwise, regular visitors must pass multi-factor authentication before they can visit the URL.
- Data risk control does not apply to websites that provide API calling services. API calls are directly initiated machine actions and cannot pass the CAPTCHA verification of data risk control. However, if the API operation is called after regular visitors click a button on a page, you can implement data risk

control.

View protection results

You can use Log Service for WAF to view the protection results.

- Allowed requests

  The following figure shows a request that passes data risk control verification. The URL of a request from a regular visitor that passes data risk control verification includes a parameter that starts with u_a. The request is forwarded to the origin server by WAF and the origin server returns a response to the visitor.

- Blocked requests

  The following figure shows a request that is blocked by data risk control. Typically, a request directly sent to the URL of a service does not start with u_a, or starts with a forged u_a parameter. WAF blocks this type of request, and the origin server does not return responses.

After you enable Log Service for WAF, choose **Advanced Search > URL Key Words** and set the endpoints to be protected by data risk control. This function helps you monitor the status of data risk control and records blocked requests. For more information, see Use full logs.

Examples

User Tom has a website and the website domain name is `www.abc.com`. Regular visitors can register as website members at `www.abc.com/register.html`. Tom notices that attackers can use malicious scripts to submit registration requests and create accounts. These accounts are used to participate in prize draws held by the website. The registration requests are highly similar to normal requests, and the request rate is maintained at a normal level. The HTTP flood protection policy fails to identify this type of malicious request.

Sample configurations

Tom configures WAF for the website and enables data risk control for the domain name `www.abc.com`. The URL of the most crucial registration service is `www.abc.com/register.html`. Therefore, Tom sets the protected URL to this URL.

Protection results

After the configurations take effect, data risk control inserts a JavaScript plug-in into all web pages of the website to monitor and analyze the behaviors of each visitor that visits `www.abc.com`, including the homepage subpages. Data risk control then determines whether a visitor behavior is normal. Data risk control also determines whether a source IP address is malicious based on the Alibaba Cloud big data reputation library.

When a visitor sends a registration request to `www.abc.com/register.html`, WAF determines whether the visitor is a potential attacker based on the visitor behavioral data generated from the time the visitor visits the website to the time the visitor submits the registration request. For example, if a visitor directly submits a registration request without performing other operations in advance, the request is identified as suspicious.

- If data risk control determines that the request is from a regular visitor based on previous visitor behaviors, the visitor can register accounts without verification.

- If data risk control identifies a request as potentially malicious, or the source IP address has a record of sending malicious requests, CAPTCHA is triggered to verify the identity of the visitor. Only visitors that pass the verification can register accounts.
  - If CAPTCHA verification captures suspicious visitor behaviors, such as using scripts to simulate real visitor behaviors to pass CAPTCHA verification, data risk control requires two-factor authentication to verify the visitor identity until the visitor passes verification and is identified as a regular visitor.
  - If the visitor fails the verification, data risk control blocks the request.

During this process, data risk control is enabled for the entire website (`www.abc.com`). Data risk control inserts a JavaScript plug-in into all web pages of the website to analyze visitor behaviors. However, protection and verification are enabled for only `www.abc.com/register.html` where visitors submit registration requests. Data risk control is triggered only after a registration request is submitted.

3.4. Application protection

3.4.1. Overview

Web Application Firewall (WAF) provides the application protection feature that allows you to use SDKs to protect native applications. This feature secures connections and protects applications from bot scripts.

What security issues can be resolved by application protection

Application protection was developed based on years of Alibaba experience protecting against online attackers, exploiters, and speculators. After applications are integrated with the Anti-Bot SDK, they have the same capabilities as Tmall, Taobao, Alipay, and other Alibaba applications to maintain secure connections. The applications have access to the library of malicious device fingerprints accumulated by Alibaba Group against online attackers, exploiters, and speculators. This helps you fundamentally solve your application risks.

Application protection provides the following solutions to resolve security issues of native applications:

- Malicious registrations, credential stuffing, and brute-force attacks
- HTTP flood attacks against applications
- SMS and verification code API abuse
- Coupon hunting and snatching
- Malicious purchases of limited goods
- Malicious ticket checking and abuse such as air tickets or hotel booking
- Valuable information crawling such as prices, private credit information, financing, and fictions
- Vote rigging
- Spam and malicious comments

How to enable application protection

Take the following steps to enable application protection for your applications.

1. Activate the application protection module in the WAF console.

   Application protection is a value-added service provided by WAF. You must enable the module before you enable application protection. You can enable application protection in the following ways:

- If you have not activated WAF, you must activate WAF subscription and then purchase the **Mobile App Protection** service in the advanced configuration. For more information, see Activate Alibaba Cloud WAF.
- If you have already activated WAF, upgrade the WAF and purchase the **Mobile App Protection** service in the advanced configuration.

2. Add the domain name of your application to WAF to activate application protection. For more information, see Add a domain.
3. Update the DNS settings of the domain name to resolve the domain name to the corresponding CNAME address of WAF. For more information, see Modify DNS settings.
4. Contact Alibaba Cloud technical support to obtain the Anti-Bot SDK package and integrate the SDK package into your application. For more information, see the following topics:
   - Integrate the Anti-Bot SDK into iOS applications
   - Integrate the Anti-Bot SDK into Android applications

   > **Note** SDK integration may take one or two days.
5. After you finish integrating the Anti-Bot SDK, configure application protection in the WAF console. You can also customize the endpoints that need to be protected and enable version protection as needed. For more information, see Configure application protection.
6. Use SDK-integrated applications to send test requests, and debug errors and exceptions based on the responses and log data until the SDK integration is verified correct.
7. Enable application protection in the WAF console after you release the latest version of the SDK-integrated application. For more information, see Configure application protection.

   > **Note** We recommend that you perform an update when you release a new version of your application. Otherwise, the old version still contains security risks.

3.4.2. Integrate the Anti-Bot SDK into iOS applications

This topic describes how to integrate the Anti-Bot SDK into iOS applications.

SDK files for iOS applications

Contact Alibaba Cloud technical support to obtain the SDK package, and decompress it on your local machine. The following table describes the files contained in the *sdk-iOS* folder.

File name	Description
SGMain.framework	The main framework.
SecurityGuardSDK.framework	The basic security plug-in.
SGSecurityBody.framework	The bot recognition plug-in.

File name	Description
SGAVMP.framework	The virtual machine engine plug-in.
yw_1222_0335_mwua.jpg	Configuration files.

Configure an iOS project

1. Import the SDK dependency files. Import the following four .framework files extracted from the SDK package to the dependency library in an iOS project. The dependency library locates in the **Link Binary With Libraries** menu on the **Build Phases** tab.
   - *SGMain.framework*
   - *SecurityGuardSDK.framework*
   - *SGSecurityBody.framework*
   - *SGAVMP.framework*

   
2. Add link options. On the **Build Settings** tab, choose **Linking > Other Linker Flags** to set the value to **-ObjC**.

   
3. Import system dependency files. Import these files to the dependency library of an iOS project:
   - *CoreFoundation.framework*
   - *CoreLocation.framework*
   - *AdSupport.framework*
   - *CoreTelephony.framework*
   - *CoreMotion.framework*
   - *SystemConfiguration.framework*

   
4. Import the configuration file. Add the *yw_1222_0335_mwua.jpg* configuration file in the SDK package to the *mainbundle* directory.

   > **Notice** When the application integrates multiple targets, make sure to add the `yw_1222_0335_mwua.jpg` configuration file to the correct target membership.

Call the SDK

Step 1: Initialize the SDK

Endpoint: `+ (BOOL) initialize;`

Function: Initializes the SDK.

Parameters: None.

Responses: Boolean. YES is returned if the initialization is successful. NO is returned if the initialization fails.

Call methods: `[JAQAVMPSignature initialize];`

Sample code

```
static BOOL avmpInit = NO;
- (BOOL) initAVMP{
 @synchronized(self) { // just initialize once
  if(avmpInit == YES){
   return YES;
  }
  avmpInit = [JAQAVMPSignature initialize];
  return avmpInit;
 }
}
```

Step 2: Sign the request

Endpoints: `+ (NSData*) avmpSign: (NSInteger) signType input: (NSData*) input;`

Function: Signs the input data by using the AVMP technique, and returns a signature string.

Warning The signed request body must be the same as the request body that is sent by the client. That is, the string coding format, spaces, special characters, and parameter sequence of the signed request body must be the same as those of the request body sent by the client. Otherwise, signature verification may fail.

Request parameters

Parameter	Type	Required	Description
signType	NSInteger	Yes	The algorithm used to sign the request. Set the value to `3`.
input	NSData*	No	The data to be signed, which is typically the entire request body. **Note** If the request body is empty, for example, an empty POST or GET request body, enter null or the value of the Bytes parameter.

Responses: A signature string is returned.

Call methods: `[JAQAVMPSignature avmpSign: 3 input: request_body];`

Sample code

> **Note** When the client sends data to the server, you must call the avmpSign operation to sign the entire request body. Then, you will obtain a wToken signature string.

```
# define VMP_SIGN_WITH_GENERAL_WUA2 (3)
- (NSString*) avmpSign{
 @synchronized(self) {
  NSString* request_body = @"i am the request body, encrypted or not!" ;
  if(![ self initAVMP]){
   [self toast:@"Error: init failed"];
    return nil;
  }
  NSString* wToken = null;
  NSData* data = [request_body dataUsingEncoding:NSUTF8StringEncoding];
  NSData* sign = [JAQAVMPSignature avmpSign: VMP_SIGN_WITH_GENERAL_WUA2 input:data];
  if(sign == nil || sign.length <= 0){
   return nil;
  }else{
   wToken = [[NSString alloc] initWithData:sign encoding: NSUTF8StringEncoding];
   return wToken;
  }
 }
}
```

If the request body is empty, you must call the avmpSign operation to generate the wToken signature string. When you call this operation, set the value of the second parameter to null. Examples:

```
NSData* sign = [JAQAVMPSignature avmpSign: VMP_SIGN_WITH_GENERAL_WUA2 input:null];
```

Step 3: Add wToken to the protocol header

Sample code

```
#define VMP_SIGN_WITH_GENERAL_WUA2 (3)
-(void)setHeader
{ NSString* request_body = @"i am the request body, encrypted or not!" ;
 NSData* body_data = [request_body dataUsingEncoding:NSUTF8StringEncoding];
 NSString* wToken = nil;
 NSData* sign = [JAQAVMPSignature avmpSign: VMP_SIGN_WITH_GENERAL_WUA2 input:body_data];
 wToken = [[NSString alloc] initWithData:sign encoding: NSUTF8StringEncoding];
 NSString *strUrl = [NSString stringWithFormat:@"http://www.xxx.com/login"];
 NSURL *url = [NSURL URLWithString:strUrl];
 NSMutableURLRequest *request =
  [[NSMutableURLRequest alloc]initWithURL:url cachePolicy:NSURLRequestReloadIgnoringCacheData time
outInterval:20];
 [request setHTTPMethod:@"POST"];
 // set request body info
 [request setHTTPBody:body_data];
 // set wToken info to header
 [request setValue:wToken forHTTPHeaderField:@"wToken"];
 NSURLConnection *mConn = [[NSURLConnection alloc]initWithRequest:request delegate:self startImmedia
tely:true];
 [mConn start];
 // ...
}
```

Step 4: Send data to the server

Send the data with the modified protocol header to Alibaba Cloud Security, which analyzes the wToken for risk identification and malicious request interception, and then forwards valid requests to the origin server.

Error code

Errors may occur when you call the initialize and avmpSign operations. If the system fails to generate a valid signature string, see the information about security guard errors in the console.

The following table lists the common error codes and their descriptions.

Error code	Description
1901	The error code returned because the parameters are invalid. Check the parameters.
1902	The error code returned because the image file is invalid. The image may not match the bundle ID.
1903	The error code returned because the format of the image file is invalid.

Error code	Description
1904	Upgrade the image version. The AVMP signature function only supports v5 images.
1905	The error code returned because the specified image file cannot be found. Make sure that the yw_1222_0335_mwua.jpg image file has been correctly added to the project.
1906	The error code returned because the AVMP signature of the image does not have the required bytecode. Check whether the image is invalid.
1907	The error code returned because the initialization of AVMP failed. Try again later.
1910	The error code returned because the AVMP instance is invalid. Possible causes include: • The AVMP instance is destroyed before InvokeAVMP is called. • The version of the bytecode of the image does not match the SDK.
1911	The byteCode of the encrypted image does not have the corresponding export function.
1912	The error code returned because the system failed to call AVMP. Contact Alibaba Cloud technical support.
1913	The error code returned because the InvokeAVMP method was called after the AVMP instance had been destroyed.
1915	The error code returned because the memory resources of the AVMP instance are insufficient. Try again later.
1999	The error code returned because an unknown error occurred. Try again later.

3.4.3. Integrate the Anti-Bot SDK into Android applications

This topic describes how to integrate the Anti-Bot SDK into Android applications.

SDK files for Android applications

Contact Alibaba Cloud technical support to obtain the SDK package, and decompress it on your local machine. The following table describes the files contained in the *sdk-Android* folder.

File name	Description
SecurityGuardSDK-xxx.aar	The main framework.
AVMPSDK-xxx.aar	The virtual machine engine plug-in.
SecurityBodySDK-xxx.aar	The bot recognition plug-in.
yw_1222_0335_mwua.jpg	The configuration file of the virtual machine.

Configure an Android project

1. Import the AAR files from the decompressed SDK package to Android Studio. Copy all the AAR files from the sdk-Android folder to the *libs* directory of the Android application project.

   > **Note** If the *libs* directory does not exist in the current project, manually create a folder named *libs* in the specified path.

2. Modify the configurations. Open the *build.gradle* file of the project and modify the configuration as follows.
   - Add the *libs* directory as the source for searching dependencies.

     ```
     repositories{
       flatDir {
         dirs 'libs'
       }
     }
     ```

   - Add compilation dependencies.

     > **Note** The versions of the AAR files in this example may be different from those of the files you downloaded.

     ```
     dependencies {
       compile fileTree(include: ['*.jar'], dir: 'libs')
       compile ('com.android.support:appcompat-v7:23.0.0')
       compile (name:'AVMPSDK-external-release-xxx', ext:'aar')
       compile (name:'SecurityBodySDK-external-release-xxx', ext:'aar')
       compile (name:'SecurityGuardSDK-external-release-xxx', ext:'aar')
     }
     ```

3. Add the JPG configuration file from the decompressed SDK package to the *drawable* directory. Copy the *yw_1222_0335_mwua.jpg* configuration file in the *sdk-Android* folder to the *drawable* directory of the Android application project.

> **Note** If the *drawable* directory does not exist in the project, create a folder named *drawable* in the specified path.

4. Remove redundant application binary interfaces (ABIs) because they require SO files. Currently, the Anti-Bot SDK only provides SO files for the following ABIs: armeabi, armeabi-v7a, and arm64-v8a.

   > **Warning** Therefore, you must filter out redundant ABIs. Otherwise, the application may crash.

   i. In the *libs* directory of the Android application project, delete all CPU architecture files other than *armeabi*, *armeabi-v7a*, and *arm64-v8a*, including *x86*, *x86_64*, *mips*, and *mips64*. Keep the *armeabi*, *armeabi-v7a*, and *arm64-v8a* folders only.

   ii. As shown in the following sample code, add filter rules to the *build.gradle* configuration file of the application project. Architectures specified by abiFilters are included in the Android application package (APK) file.

   > **Note** In this example, abiFilters only specifies the armeabi architecture. You can also specify the armeabi-v7a and arm64-v8 architectures as needed.

```
defaultConfig{
 applicationId "com.xx.yy"
 minSdkVersion xx
 targetSdkVersion xx
 versionCode xx
 versionName "x.x.x"
 ndk{
  abiFilters "armeabi"
  // abiFilters "armeabi-v7a"
  // abiFilters "arm64-v8a"
 }
}
```

   > **Note** If you keep the SO files of the armeabi architecture only, you can significantly reduce the size of the application without affecting its compatibility.

5. Grant permissions to the application.
   - If you use an Android Studio project and AAR files to integrate the SDK, required permissions are already specified in the AAR files. You do not need to grant permissions to the application in the project.
   - If you use an Eclipse project, you must add the following permissions to the *AndroidMenifest.xml* file:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
```

6. Add ProGuard configurations.

   Note If you need to use ProGuard to obfuscate code, you must add ProGuard configurations. Methods to configure ProGuard in Android Studio and Eclipse are different.

   - Android Studio

     If you have set the proguardFiles parameter and the minifyEnabled parameter is set to true in the *build.gradle* file, the *proguard-rules.pro* file is used to obfuscate code.

   - Eclipse

     If you have configured ProGuard in the *project.properties* file, such as adding the `proguard.config=proguard.cfg` statement to the *project.properties* file, ProGuard is used to obfuscate code.

     Note Obfuscation configurations are specified in the *proguard.cfg* file.

   Add keep rules

   To guarantee that certain classes are not obfuscated, you must add the following rules to the ProGuard configuration file.

```
-keep class com.taobao.securityjni.**{*;}
-keep class com.taobao.wireless.security.**{*;}
-keep class com.ut.secbody.**{*;}
-keep class com.taobao.dp.**{*;}
-keep class com.alibaba.wireless.security. **{*;}
```

Call the SDK

Step 1: Import packages

```
import com.alibaba.wireless.security.jaq.JAQException;
import com.alibaba.wireless.security.jaq.avmp.IJAQAVMPSignComponent;
import com.alibaba.wireless.security.open.SecurityGuardManager;
import com.alibaba.wireless.security.open.avmp.IAVMPGenericComponent;
```

Step 2: Initialize the SDK

Endpoints: `boolean initialize();`

Function: Initializes the SDK.

Parameters: None.

Responses: Boolean values. If the initialization is successful, true is returned. If the initialization fails, false is returned.

Sample code

```
IJAQAVMPSignComponent jaqVMPComp = SecurityGuardManager.getInstance(getApplicationContext()).get
Interface(IJAQAVMPSignComponent.class);
boolean result = jaqVMPComp.initialize();
```

Step 3: Sign requests

Endpoints: `byte[] avmpSign(int signType, byte[] input);`

Function: Signs the input data by using the Ali Virtual Machine Protect (AVMP) technique, and returns a signature string.

Parameters

Parameter	Type	Required	Description
signType	int	Yes	The algorithm used to sign requests. Set the value to `3`.
input	byte[]	No	The data to be signed, which is typically the entire request body. **Note** If the request body is empty, for example, an empty POST or GET request body, enter null or the value of the Bytes parameter, such as "".getBytes("UTF-8").

Responses: A signature string of the byte[] data type.

Sample code: When the client sends data to the server, it must call the avmpSign method to sign the entire request body. A wToken signature string is returned.

```
int VMP_SIGN_WITH_GENERAL_WUA2 = 3;
String request_body = "i am the request body, encrypted or not!" ;
byte[] result = jaqVMPComp.avmpSign(VMP_SIGN_WITH_GENERAL_WUA2, request_body.getBytes("UTF-8"))
;
String wToken = new String(result, "UTF-8");
Log.d("wToken", wToken);
```

Step 4: Add wToken to the protocol header

Add the content of the wToken field to the object of the HttpURLConnection class.

Sample code

```
String request_body = "i am the request body, encrypted or not!" ;
URL url = new URL("http://www.xxx.com");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("POST");
// set wToken info to header
conn.setRequestProperty("wToken", wToken);
OutputStream os = conn.getOutputStream();
// set request body info
byte[] requestBody = request_body.getBytes("UTF-8");
os.write(requestBody);
os.flush();
os.close();
```

Step 5: Send data to the server

Send data with the modified protocol header to the server of the application. Anti-Bot Service captures the data and parses the wToken to identify risks.

> **Warning** The signed request body must be the same as the original request body that is sent by the client. The string encoding format, spaces, special characters, and parameter sequence of the signed request body must be the same as those of the original request body sent by the client. Otherwise, the request fails to pass signature verification.

Error codes

Errors may occur when you call the initialize and avmpSigni methods. If an error occurs or the SDK fails to generate a signature string, use the keyword `SecException` to search for relevant information in the log data.

The following table describes common error codes.

Error code	Description

Error code	Description
1901	The error code returned because the parameters are invalid. Check the parameters.
1902	The error code returned because the image file is invalid. The APK signature used to retrieve the image file is not the same as that of the application. Use the APK signature of the application to generate a new image.
1903	The error code returned because the format of the image file is invalid.
1904	Upgrade the image version. The AVMP signature function only supports v5 images.
1905	The error code returned because the specified image file cannot be found. Make sure that the image file is in the *res\drawable* directory, and AVMP images are in the *yw_1222_0335_mwua.jpg* file.
1906	The error code returned because the AVMP signature of the image does not have the required bytecode. Check whether the image is invalid.
1907	The error code returned because the initialization of AVMP failed. Try again later.
1910	The error code returned because the AVMP instance is invalid. Possible causes include: • The InvokeAVMP method was called after the AVMP instance had been destroyed. • The version of the bytecode of the image does not match the SDK.
1911	The error code returned because the bytecode of the encrypted image does not have the required export function.
1912	The error code returned because the system failed to call AVMP. Contact Alibaba Cloud technical support.
1913	The error code returned because the InvokeAVMP method was called after the AVMP instance had been destroyed.
1915	The error code returned because the memory resources of the AVMP instance are insufficient. Try again later.

Error code	Description
1999	The error code returned because an unknown error occurred. Try again later.

Verify the integration

Take the following steps to verify that the Anti-Bot SDK has been correctly integrated into the application.

1. Convert the packaged APK file into a ZIP file by modifying the file name extension, and decompress the file on your local machine.
2. Go to the *libs* directory of the project, and make sure that the folder only contains the *armeabi*, *armeabi-v7a*, and *arm64-v8a* sub-folders.

   > **Note** If any other architecture file exists, delete it. For more information, see Configure an Android project.

3. Go to the *res/drawable* directory of the project, and make sure that the *yw_1222_0335_mwua.jpg* file exists and its size is not 0.
4. Print the log, and make sure that the correct signature information can be generated after the avmpSign method is called.

   > **Note** If signature information cannot be generated, see the error codes and descriptions to troubleshoot.

FAQ

Why is the key image incorrectly optimized after shrinkResources is set to true?

In Android Studio, if shrinkResources is set to true, resource files that are not referenced in the code may be optimized during project compilation. After shrinkResources is set to true, JPG files in the Anti-Bot SDK may not work as expected. If the size of the *yw_1222_0335.jpg* configuration file in the packaged APK is 0 KB, it indicates that the image file has been optimized.

Solutions

1. Create a directory named *raw* in the *res* directory of the project, and create a file named *keep.xml* in the raw directory.
2. Add the following content to the *keep.xml file*:

```
<? xml version="1.0" encoding="utf-8"? >
<resources xmlns:tools="http://schemas.android.com/tools"
tools:keep="@drawable/yw_1222_0335.jpg,@drawable/yw_1222_0335_mwua.jpg" />
```

3. After you add the content, compile the project APK again.

3.4.4. Configure application protection

Application protection provides secure connections and anti-bot protection for native applications. This function identifies proxies, emulators, and requests with invalid signatures. This topic describes how to configure and enable application protection in the Web Application Firewall (WAF) console after you integrate the Anti-Bot SDK into an application.

Prerequisites

- A WAF instance is purchased and the instance meets the following requirements:
  - The instance is billed on a subscription basis.
  - **App Protection** is enabled. This feature is a value-added service.

  For more information, see Purchase a WAF instance.
- You have integrated the Anti-Bot SDK into the target application. For more information, see Overview.

Procedure

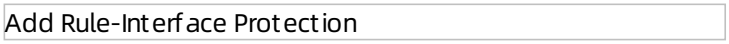
1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. Click the **Bot Management** tab, find the **App Protection** section, and then click **Settings**.

   > **Note** After application protection is enabled, all service requests are checked by the function. You can configure a Bot Management rule so that the requests that match the rule bypass the check. For more information, see Configure a whitelist for Bot Management.

6. Create a path protection rule.
   i. On the **App Protection** page, find the **Interface Protection** section, and click **Add Rule**.
   ii. In the **Add Rule** dialog box that appears, set the following parameters.

   > **Note** In the test phase, we recommend that you set **Path** to a forward slash (/) and **Matching** to **Prefix Match** to match all paths. You can set **Action** to **Monitor**. If the target domain name is a test domain name, you can set Action to **Block**. This allows you to debug the application without affecting your online workloads.

Parameter	Description
Rule Name	The name of the rule that you want to create.

Parameter	Description
Path Protection Settings	The path that you need to protect. The following parameters are required: ■ **Path**: The path that you need to protect. A forward slash (/) indicates all paths. **Note** Signature verification may fail when the body of a POST request exceeds 8 KB. We recommend that you disable SDK protection for API operations that do not require protection, for example, the API operation that is used to upload large images. If you do need to enable SDK protection for an API operation, use a user-defined field. ■ **Matching**: **Prefix Match**, **Precise Match**, and **Regular Expression Match** are supported. If you set the value to Prefix Match, all endpoints under the specified path are considered matches. If you set the value to Precise Match, only the specified path is considered a match. If you set the value to Regular Expression Match, paths specified by the regular express are considered matches. ■ **Parameter**: The parameters that need to be matched if the protected path contains invariable parameters. WAF can use these parameters to filter endpoints more precisely. The parameters are the parts following the question mark (?) in the request URL. Example: The protected URL contains `domain name/? action=login&name=test`. In this case, set **Path** to a forward slash (/), **Matching** to Prefix Match, and **Parameter** to name, login, name=test, and action=login.
Protection Policy	Protection policy that you want to use. ■ **Invalid Signature**: This policy is selected by default and cannot be cleared. The system checks whether the signatures of requests sent to the specified path are correct. The rule is matched if a signature is incorrect. ■ **Simulator**: If this policy is selected, the system checks whether the user uses an emulator to initiate requests to the specified path. The rule is matched if a request is initiated from an emulator. ■ **Proxy**: If this policy is selected, the system checks whether the user uses a proxy to initiate requests to the specified path. We recommend that you select this option. The rule is matched if a request is initiated from a proxy.

Parameter	Description
Action	The action to be performed on requests that match the rule. ■ **Monitor**: records the request but does not block the request. ■ **Block**: blocks the request and returns a 405 HTTP status code. **Notice** Before the SDK integration or debugging is completed, do not set Action to Block for domain names used in a production environment. Otherwise, valid requests may be blocked because the SDK is not properly integrated into the application. In the test phase, you can set Action to Monitor to debug the SDK-integrated application based on log data.
User-defined Field	When a user-defined field is used, the system verifies the request signature based on the specified request field and field value. By default, the system verifies the signature based on the request body. The verification may fail if the length of the request body exceeds 8 KB. In this case, you can specify a user-defined field to replace the default field for signature verification. If you select User-defined Field, you can choose Header, Parameter, or Cookie, and then specify the field that is used to verify the request signature. For example, you can choose **Cookie** and then enter DG_ZUID. This replaces the default body field with the DG_ZUID field in the request cookie as the field used for signature verification.

iii. Click **Confirm**.

7. Enable version protection.You can configure version protection to block requests from non-official applications. You can also use this function to verify the validity of an application.

Note A version protection policy is required only when you need to verify the validity of an application.

i. On the **App Protection** page, find the **Version Protection** section and turn on **Allow Specified Version Requests**.

ii. In the **Add Rule** dialog box that appears, set the following parameters.

Parameter	Description
Rule Name	The name of the rule that you want to create.
Valid Version	The valid versions of an application. ■ **Enter the legal package name**: Enter the name of the valid application package, for example, *com.aliyundemo.example*. ■ **Package Signature**: Contact Alibaba Cloud technical support to obtain the package signature. This parameter is optional if the package signature does not need to be verified. In this case, the system verifies only the package name. **Notice** The **Package Signature** is not the signature of the application certificate. Click **Add Valid Version** to add more valid versions. You can add a maximum of five valid versions. Package names must be unique. Currently, both iOS and Android applications are supported. You can enter multiple valid versions to match the package names.
Disposal Method for Illegal Version	■ **Monitor**: records the request but does not block the request. ■ **Block**: blocks the request and returns a 405 HTTP status code.

iii. Click **Confirm**.

8. Enable application protection. In the **App Protection** section, turn on **Status**.

Note We recommend that you integrate the Anti-Bot SDK into the application, debug the application, and release the new version before you enable application protection to make sure that the protection settings take effect.

4.Access control and throttling

4.1. Configure HTTP flood protection

After you add a website to Web Application Firewall (WAF), HTTP flood protection is enabled by default and protects your website against HTTP flood attacks. When HTTP flood attacks are blocked, the 405 Method Not Allowed error code is returned. You can adjust HTTP flood protection policies as needed.

Prerequisite

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. On the **Access Control/Throttling** tab, find the **HTTP Flood Protection** section and configure the following parameters.

Parameter	Description
Status	Enable or disable HTTP flood protection. By default, HTTP flood protection is enabled after you add a website to WAF. **Note** After HTTP flood protection is enabled, all requests destined for your website are checked by this function. You can configure a **Access Control/Throttling** rule so that requests that match the rule bypass the check. For more information, see Configure a whitelist for Access Control/Throttling.

Parameter	Description
Mode	The protection mode. Valid values: ○ **Protection**: This mode blocks only suspicious requests and maintains a low false positive rate. We recommend that you apply this mode when no abnormal traffic is detected on the website to avoid false positives. ○ **Protection-emergency**: This mode effectively blocks HTTP flood attacks but maintains a high false positive rate. You can apply this mode if the Protection mode fails to block HTTP flood attacks, the website responds slowly, and indicators such as traffic, CPU, and memory are abnormal. **Note** The Protection-emergency mode is applicable to web pages and HTML5 pages. This mode is not suitable for APIs or native applications because a large number of false positives may occur. We recommend that you create custom protection policies for API and native applications scenarios. For more information, see Create a custom protection policy.

References

- If you find that the **Protection-emergency** mode cannot block a large number of attacks, we recommend that you check whether the attacks come from back-to-origin IP addresses of WAF. If the origin server is directly attacked, you can change the settings to allow only requests from the back-to-origin IP addresses of WAF. For more information, see Configure protection for an origin server.
- If you need to reinforce protection and maintain a low false positive rate, we recommend that you use the custom protection policy. For more information, see Create a custom protection policy.

4.2. Configure a blacklist

After you add a website to Web Application Firewall (WAF), you can enable the blacklists feature. This feature blocks access requests from specified IP addresses, Classless Inter-Domain Routing (CIDR) blocks, and IP addresses in specified regions. You can specify either an IP address blacklist or a region blacklist based on your requirements.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.

    If your WAF instance is of the **Business** edition or lower, you can use only an **IP address blacklist**. If you want to use a **region blacklist**, make sure that your WAF instance is of the **Enterprise** edition or higher.

    For more information, see Editions and features.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

WAF supports both IP address and region blacklists.

- An IP address blacklist blocks access requests from specified IP addresses and CIDR blocks.
- A region blacklist blocks the access requests from specified regions inside or outside China. You can specify a total of 247 regions inside and outside China as blocked regions.

  You can use the IP address library of Taobao to query the source region of an IP address. For more information, visit the IP address library of Taobao.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. On the **Access Control/Throttling** tab, find the **IP Blacklist** section. Turn on **Status** and click **Settings**.

   > **Note** If you specify an IP address blacklist, all requests destined for your website are checked by this blacklist. You can also configure a whitelist for **Access Control/Throttling** to allow qualified requests to bypass the check. For more information, see Configure a whitelist for Access Control/Throttling.

6. On the **IP Blacklist** page, configure **IP Blacklist** and **Area-based IP Blacklist**.
   - **IP Blacklist**: Enter IP addresses that you want to block and click **Save** in the lower part of the page. Separate multiple IP addresses with commas (,). You can add a maximum of 200 IP addresses.

   - **Area-based IP Blacklist**: Select regions that you want to block from the **Inside China** and **Outside China** tabs. Then, click **Save** in the lower part of the page.

     After the blacklists feature is enabled, all the access requests from IP addresses and regions in the blacklists are blocked.

References

- If you need more precise access control based on blacklists, we recommend that you use a custom protection policy. For more information, see Create a custom protection policy.
- If you want to allow access requests only from specified IP addresses, we recommend that you configure a whitelist for **Access Control/Throttling**. For more information, see Configure a whitelist for Access Control/Throttling.

4.3. Configure scan protection

After you add a website to Web Application Firewall (WAF), you can enable the scan protection function for your website. If you do that, access requests from specific IP addresses are automatically blocked. These IP addresses include source IP addresses that initiate high-frequency web attacks and malicious directory traversal attacks, and IP addresses defined in common scanning tools or the Alibaba Cloud malicious IP library. You can customize scan protection policies based on your requirements.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - If the instance is billed on a subscription basis, the instance must be of the **Pro** edition or higher.

    If you need to customize policies of **Blocking IPs Initiating High-frequency Web Attacks** and **Directory Traversal Prevention**, the instance must be of the **Business** edition or higher. The WAF instance of the **Pro** edition or lower can use only the default scan protection policies. For more information, see Editions and features.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

The scan protection function provides the following policies:

- Blocking IPs Initiating High-frequency Web Attacks: automatically blocks client IP addresses that initiate multiple web attacks within a short period of time. You can customize the protection policy and manually unblock a blocked IP address.
- Directory Traversal Prevention: automatically blocks client IP addresses that initiate multiple directory traversal attacks in a short period of time. You can customize the protection policy and manually unblock a blocked IP address.
- Scanning Tool Blocking: automatically blocks access requests from IP addresses defined in common scanning tools. The scanning tools include sqlmap, AWVS, Nessus, AppScan, WebInspect, Netsparker, Nikto, and RSAS.
- Collaborative Defense: automatically blocks access requests from IP addresses in the Alibaba Cloud malicious IP library.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. On the **Access Control/Throttling** tab, find the **Scan Protection** section and configure the following settings:

> **Note** By default, all requests destined for your website are checked by scan protection when any policy in this section is enabled. You can configure a **Access Control/Throttling** rule so that requests that match the rule bypass the check. For more information, see Configure a whitelist for Access Control/Throttling.

- **Blocking IPs Initiating High-frequency Web Attacks**: You can enable or disable it.

  Configure the protection policy.

  a. Turn on Blocking IPs Initiating High-frequency Web Attacks.

  b. Click **Settings**.

  c. In the **Rule Setting** dialog box, specify the following parameters: **Inspection Time Range**, **The number of attacks exceeds**, and **Blocked IP Addresses**.

  If the number of web attacks initiated from a client IP address in the specified inspection time range exceeds a specific number, the access requests from this IP address are blocked for the specified time range.**Inspection Time RangeThe number of attacks exceedsBlocked IP Addresses**

  > **Note** We recommend that you select a built-in configuration mode from **Flexible Mode**, **Strict Mode**, and **Normal Mode** in the **Mode** section. You can modify the parameters based on your requirements.

  d. Click **Confirm**.

  You can click **Unblock IP Address** to unblock IP addresses that are blocked by the policy.

- **Directory Traversal Prevention**: You can enable or disable it.

  Configure the protection policy.

  a. Turn on Directory Traversal Prevention.

  b. Click **Settings**.

  c. In the **Rule Setting** dialog box, specify the following parameters: **Inspection Time Range**, **The total requests exceed**, **And the percentage of responses with 404 exceeds**, **Blocked IP Addresses**, and **Directory number**.

  If the total number of requests initiated from a client IP address in the specified inspection time range exceeds a specific number and the proportion of the requests for which the HTTP status code 404 is returned to the total requests exceeds a specific proportion, or the number of directories to which requests are sent within the specified inspection time range exceeds a specific number, the access requests from this IP address are blocked for the specified time range.**Inspection Time RangeThe total requests exceedInspection Time RangeBlocked IP Addresses**

  > **Note** We recommend that you select a built-in configuration mode from **Flexible Mode**, **Strict Mode**, and **Normal Mode** in the **Mode** section. You can modify the parameters based on your requirements.

  d. Click **Confirm**.

You can click **Unblock IP Address** to unblock IP addresses that are blocked by the policy.

- **Scanning Tool Blocking**: You can enable or disable it.

  After you enable Scanning Tool Blocking, the behaviors of common scanning tools are automatically detected. If an access request meets the characteristics of scanning, this request is always blocked. If you disable Scanning Tool Blocking, scanning behaviors are no longer blocked.

- **Collaborative Defense**: You can enable or disable it.

  After you enable Collaborative Defense, all access requests from the IP addresses in the Alibaba Cloud malicious IP library are blocked.

4.4. Create a custom protection policy

After you add a website to Web Application Firewall (WAF), you can enable the custom protection policy function to protect the website. This function allows you to customize ACL rules based on precise match conditions and configure rate limiting. Custom protection policy can be tailored for different scenarios, such as hotlink protection and website backend protection. You can create custom protection rules as needed.

Prerequisites

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

The custom protection policy function is implemented by using custom protection rules. Custom protection rules include ACL rules and HTTP flood protection rules.

- An ACL rule filters requests based on precise match conditions such as client IP addresses, request URLs, and common request headers.
- An HTTP flood protection rule filters requests based on the precise match conditions and rate limiting you have configured.

Limits

Subscription WAF instances have the following limits on custom protection policies.

Specification	Description	Enterprise	Business	Pro
Number of custom protection rules	The maximum number of custom protection rules that you can create.	200	100	100
Advanced match fields	The advanced match fields other than IP addresses and URLs that you can specify in custom protection rules.	Supported	Supported	Not supported
Rate limiting	The rate limiting settings in a custom protection policy. The settings define an HTTP flood protection rule.	Supported	Supported	Not supported

Specification	Description	Enterprise	Business	Pro
Custom statistical objects	The custom statistical objects other than IP addresses and sessions that can be used to configure rate limiting.	Supported	Not supported	Not supported

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Access Control/Throttling** tab and find the **Custom Protection Policy** section. Then, turn on **Status** and click **Settings**.

   > **Note** When the custom protection policy function is enabled, all requests destined for your website are checked by the function. You can configure a Access Control/Throttling rule so that requests that match the rule bypass the check. For more information, see Configure a whitelist for Access Control/Throttling.

6. Create a custom protection rule.
   i. On the **Custom Protection Policy** page, click **Create Custom Rule**.
   ii. In the **Create Rule** dialog box that appears, configure the following parameters.

Parameter	Description
Rule name	The name of the rule that you want to create.
Matching Condition	The detection logic of the rule. The rule is triggered only when match conditions are met. Click **Add rule** to add more match conditions. You can specify a maximum of five match conditions. If you have configured multiple conditions, the rule takes effect only when all of them are met. For more information about match conditions, see Fields in match conditions.
Rate Limiting	Enable or disable rate limiting. WAF starts calculating the request rate only when match conditions are met. When you enable rate limiting, you need to configure the parameters to specify the object to be calculated.  For more information about rate limiting parameters, see Rate limiting parameters.

Parameter	Description
Action	The action to be performed after the rule is triggered. Valid values: ■ **Monitor**: triggers alerts but does not block requests. ■ **Block**: blocks requests. ■ **CAPTCHA**: redirects requests to another page to implement CAPTCHA verification. ■ **Strict Captcha**: redirects requests to another page to implement strict CAPTCHA verification. ■ **JavaScript Validation**: triggers JavaScript verification. If you enable **Rate Limiting**, you must specify the **TTL (Seconds)**, during which the action takes effect. **Note** A certain latency may exist in the statistical process because WAF collects data from multiple servers in a cluster to calculate the request rate.
Protection Type	The type of the rule. This parameter is automatically set based on the status of **Rate Limiting**. ■ If rate limiting is enabled, the value is set to **HTTP Flood Protection**. ■ If rate limiting is disabled, the value is set to **ACL**.

The parameters required to configure rate limiting are described in the following table.

Parameter	Description
Statistical Object	The object based on which request rate is calculated. Valid values: ■ **IP**: calculates the number of requests from a specific IP address. ■ **Session**: calculates the number of requests transmitted over a specific session. ■ **Custom-Header**: calculates the number of requests with the same specified header content. ■ **Custom-Param**: calculates the number of requests with the same specified parameter content. ■ **Custom-Cookie**: calculates the number of requests with the same specified cookie content.
Interval (Seconds)	The time period during which the number of requests is calculated.
Threshold (Occurrences)	The maximum number of the statistical objects that are allowed during the specified time period. If this limit is exceeded, rate limiting is triggered.

Parameter	Description
Status Code	After the detection logic takes effect, the number or percentage of the specified **Status Code** within the specified time period is calculated. Select either the amount or the percentage. ■ **Amount**: The maximum number of the specified status code. ■ **Percentage (%)**: The maximum percentage of the requests for which the specified status code is returned in the total request.
Take Effect For	The objects to which rate limiting is applied. ■ **Feature Matching Objects** ■ **Applied Domains**

iii. Click **Save**.

After a custom protection rule is created, it is automatically enabled. You can view the newly created rule, and disable, modify, or delete the rule in the rule list as needed.

References

Fields in match conditions

5.Whitelist

5.1. Configure a website whitelist

After you add a website to Web Application Firewall (WAF), you can configure a website whitelist to allow trusted access requests of the website to be directly routed to the origin server. Trusted access requests include requests from trusted vulnerability scan tools and trusted authenticated third-party system endpoints.

Prerequisites

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

WAF provides multiple detection modules. If a website is added to WAF, all requests to this website are automatically detected by the modules that are enabled. To directly route trustworthy requests to your origin server, you can configure a website whitelist that allows the requests to bypass all detection modules of WAF.

You can also configure a whitelist for a specific detection module. This allows trusted access requests to bypass the detection of the specific detection module. You can configure the following types of whitelists:

- Whitelist for Web Intrusion Prevention: Trusted access requests are not detected by RegEx Protection Engine or Big Data Deep Learning Engine.
- Whitelist for Data Security: Trusted access requests are not detected by Data Leakage Prevention, Website Tamper-proofing, or Account Security.
- Whitelist for Bot Management: Trusted access requests are not detected by Bot Threat Intelligence, Data Risk Control, Intelligent Algorithm, or App Protection.
- Whitelist for Access Control/Throttling: Trusted access requests are not detected by HTTP Flood Protection, IP Blacklist, Scan Protection, or Custom Protection Policy.

> **Note** We recommend that you create a whitelist for a specific detection module as required. A whitelist with more precise rules improves website security. A whitelist for a detection module provides better security protection than a website whitelist.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. In the upper-right corner, click **Website Whitelisting**.

6. Create a website whitelist.
   i. On the **Website Whitelisting** page, click **Create Rule**.
   ii. In the **Create Rule** dialog box, configure the following parameters.

   Website Whitelisting

Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify match conditions for the rule. Click **Add rule** to add more match conditions. A maximum of five match conditions are allowed. If you specify multiple match conditions, the rule is triggered only after all the match conditions are met. For more information about match conditions, see Fields in match conditions.

   iii. Click **Save**.

   After you create rules for the whitelist, the rules are automatically enabled. You can view created rules in the rule list. You can also disable, edit, or delete rules as required.

References

Fields in match conditions

5.2. Configure a whitelist for Web Intrusion Prevention

After you add a website to Web Application Firewall (WAF), you can configure a whitelist for Web Intrusion Prevention to allow trusted access requests of the website to bypass the detection of RegEx Protection Engine and Big Data Deep Learning Engine. This whitelist is used to allow access requests that are blocked by mistake.

Prerequisites

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Web Intrusion Prevention protects your website against common web attacks and zero-day vulnerabilities. It provides the following detection modules:

- RegEx Protection Engine
- Big Data Deep Learning Engine

After the preceding detection modules are enabled, normal access requests may be blocked by mistake. In this case, you can configure a whitelist to allow trusted access requests to bypass the detection of a specific module in Web Intrusion Prevention.

We recommend that you specify rules for the whitelist as precisely as possible to ensure that only trusted access requests are allowed.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. Click the **Web Security** tab, find the **Web Intrusion Prevention** section, and then click **Settings**.
6. Create a whitelist for Web Intrusion Prevention.
   i. On the **Web Intrusion Prevention - Whitelisting** page, click **Create Rule**.
   ii. In the **Create Rule** dialog box, configure the following parameters.

   Web Intrusion Prevention - Whitelisting

Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify match conditions for the rule. Click **Add rule** to add more match conditions. A maximum of five match conditions are allowed. If you specify multiple match conditions, the rule is triggered only after all the match conditions are met. For more information about match conditions, see Fields in match conditions.
Modules Bypassing Check	Select the detection modules to bypass after the match conditions are met. Valid Values: ■ **RegEx Protection Engine** ■ **Big Data Deep Learning Engine**

   iii. Click **Save**.

   After you create rules for the whitelist, the rules are automatically enabled. You can view created rules in the rule list. You can also disable, edit, or delete rules as required.

References

Fields in match conditions

5.3. Configure a whitelist for Data Security

After you add a website to Web Application Firewall (WAF), you can configure a whitelist for Data Security to allow trusted access requests of the website to bypass the detection of Website Tamper-proofing, Data Leakage Prevention, and Account Security. This whitelist is used to allow access requests that are blocked by mistake.

Prerequisites

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Data Security protects your website against page content leaks and tampering to ensure the integrity and confidentiality of website data. It provides the following detection modules:

- Website Tamper-proofing
- Data Leakage Prevention
- Account Security

After the preceding detection modules are enabled, normal access requests may be blocked by mistake. In this case, you can configure a whitelist to allow trusted access requests to bypass the detection of a specific module in Data Security.

We recommend that you specify rules for the whitelist as precisely as possible to ensure that only trusted access requests are allowed.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. Click the **Web Security** tab, find the **Data Security** section, and then click **Settings**.
6. Create the whitelist for Data Security.
   i. On the **Data Risk Control - Whitelisting** page, click **Create Rule**.

ii. In the **Create Rule** dialog box, configure the following parameters.

Data Security Control - Whitelisting

Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify match conditions for the rule. Click **Add rule** to add more match conditions. A maximum of five match conditions are allowed. If you specify multiple match conditions, the rule is triggered only after all the match conditions are met. For more information about match conditions, see Fields in match conditions.
Modules Bypassing Check	Select the detection modules to bypass after the match conditions are met. Valid Values: ■ **Data Leakage Prevention** ■ **Website Tamper-proofing** ■ **Account Security**

iii. Click **Save**.

After you create rules for the whitelist, the rules are automatically enabled. You can view created rules in the rule list. You can also disable, edit, or delete rules as required.

References

Fields in match conditions

5.4. Configure a whitelist for Bot Management

After you add a website to Web Application Firewall (WAF), you can configure a whitelist for Bot Management to allow trusted access requests of the website to bypass the detection of Bot Threat Intelligence, Data Risk Control, Intelligent Algorithm, and App Protection. This whitelist is used to allow access requests that are blocked by mistake.

Prerequisites

- A WAF instance is purchased and the instance meets the following requirements:
  - The instance is billed on a subscription basis.
  - **Bot Management** is enabled. This feature is a value-added service.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Bot Management protects web applications, native applications, and APIs from malicious crawlers. It provides the following detection modules:

- Allowed Crawlers
- Bot Threat Intelligence
- Data Risk Control
- App Protection
- Intelligent Algorithm

After the preceding detection modules but Allowed Crawlers are enabled, normal access requests may be blocked by mistake. In this case, you can configure a whitelist to allow trusted access requests to bypass the detection of a specific module in Bot Management.

We recommend that you specify rules for the whitelist as precisely as possible to ensure that only trusted access requests are allowed.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

   
5. Click the **Bot Management** tab, find the **Bot Management** section, and then click **Settings**.
6. Create a whitelist for Bot Management.
   i. On the **Bot Management - Whitelist** page, click **Create Rule**.
   ii. In the **Create Rule** dialog box, configure the following parameters.

   Bot Management - Whitelist

Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify match conditions for the rule. Click **Add rule** to add more match conditions. A maximum of five match conditions are allowed. If you specify multiple match conditions, the rule is triggered only after all the match conditions are met. For more information about match conditions, see Fields in match conditions.
Modules Bypassing Check	Select the detection modules to bypass after the match conditions are met. Valid Values: ■ **Bot Threat Intelligence** ■ **Data Risk Control** ■ **Algorithm Model** ■ **App Protection**

iii. Click **Save**.

After you create rules for the whitelist, the rules are automatically enabled. You can view created rules in the rule list. You can also disable, edit, or delete rules as required.

References

Fields in match conditions

5.5. Configure a whitelist for Access Control/Throttling

After you add a website to Web Application Firewall (WAF), you can configure a whitelist for Access Control/Throttling to allow trusted access requests of the website to bypass the detection of HTTP Flood Protection, IP Blacklist, Scan Protection, and Custom Protection Policy. This whitelist is used to allow access requests that are blocked by mistake.

Prerequisites

- A WAF instance is purchased. For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Background information

Access Control/Throttling provides custom access control policies and traffic management policies at the application layer to ensure website accessibility. It provides the following detection modules:

- HTTP Flood Protection
- IP Blacklist
- Scan Protection
- Custom Protection Policy

After the preceding detection modules are enabled, normal access requests may be blocked by mistake. In this case, you can configure a whitelist to allow trusted access requests to bypass the detection of a specific module in Access Control/Throttling.

We recommend that you specify rules for the whitelist as precisely as possible to ensure that only trusted access requests are allowed.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

5. Click the **Access Control/Throttling** tab, find the **Access Control/Throttling** section, and then click **Settings**.
6. Create a whitelist for Access Control/Throttling.

i. On the **Access Control/Throttling - Whitelisting** page, click **Create Rule**.

ii. In the **Create Rule** dialog box, configure the following parameters.

Access Control/Throttling - Whitelisting

Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify match conditions for the rule. Click **Add rule** to add more match conditions. A maximum of five match conditions are allowed. If you specify multiple match conditions, the rule is triggered only after all the match conditions are met. For more information about match conditions, see Fields in match conditions.
Modules Bypassing Check	Select the detection modules to bypass after the match conditions are met. Valid Values: ■ **HTTP Flood Protection** ■ **Custom Rules** ■ **IP Blacklist** ■ **Anti-Scan**

iii. Click **Save**.

After you create rules for the whitelist, the rules are automatically enabled. You can view created rules in the rule list. You can also disable, edit, or delete rules as required.

References

Fields in match conditions

6.Fields in match conditions

You must add match conditions to rules when you configure a whitelist and customize protection policies for Web Application Firewall (WAF). This topic describes the fields that you can use in the match conditions and their descriptions.

Match conditions and actions

In the WAF console, you can customize rules for whitelists and protection policies. A custom rule consists of match conditions and actions. When you create a rule, you must specify the match fields, logical operators, and match content to add match conditions. You also need to select an action that is triggered when requests match the conditions you specify.

- **Match conditions**

  Each match condition consists of a match field, logical operator, and match content. The match content does not support regular expressions. You can add a maximum of five match conditions to a custom rule, and the logical relation among the conditions is AND. The custom rule works only when all the match conditions are met.

- **Action**

  When you configure a whitelist rule, you must select features for Modules Bypassing Check so that requests that meet match conditions bypass the corresponding checks. When you configure custom protection policies, you must select an action that is triggered for requests that meet match conditions. For more information, see the following topics:

  - Configure a website whitelist
  - Configure a whitelist for Web Intrusion Prevention
  - Configure a whitelist for Access Control/Throttling
  - Configure a whitelist for Bot Management
  - Configure a whitelist for Data Security
  - Create a custom protection policy

Supported match fields

The following table lists the match fields that are supported in match conditions.

Match field	Edition	Logical operator	Description
IP	Pro edition or higher	Has and Does not have	The source IP address of an access request. You can enter IP addresses or CIDR blocks, for example, 1.1.1.1/24. **Note** You can enter a maximum of 50 IP addresses or CIDR blocks. Separate them with commas (,).

Match field	Edition	Logical operator	Description
URL	Pro edition or higher	• Includes and Does not include • Equals and Does not equal • URI Path Match • Regular Expression	The URL of an access request.
Referer	Pro edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than • Does not exist	The URL of the source page from which the access request is redirected.
User-Agent	Pro edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than	The browser information of the client that initiates access requests. The information includes the browser, rendering engine, and version.
Params	Pro edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than	The parameter part in the request URL, usually the part that follows the question mark (?) in the URL. For example, in `www.abc.com/index.html?action=login`, `action=login` is the parameter part.

Match field	Edition	Logical operator	Description
Cookie	Business edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than • Does not exist	The cookie information in an access request.
Content-Type	Business edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than	The HTTP content type (MIME) in the response.
Content-Length	Business edition or higher	Value less than, Value equals, and Value more than	The number of bytes in the response.
X-Forwarded-For	Business edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than • Does not exist	The actual IP address of the client that initiates access requests. X-Forwarded-For (XFF) is an HTTP header field. It is used to identify the originating IP address of a client that connects to the server through an HTTP proxy or a Server Load Balancer (SLB) instance. XFF is only included in the access requests that are forwarded by the HTTP proxy or SLB instance.
Post-Body	Business edition or higher	• Includes and Does not include • Equals and Does not equal	The content of an access request.
Http-Method	Business edition or higher	Equals and Does not equal	The request method, such as GET, POST, DELETE, PUT, and OPTIONS.

Match field	Edition	Logical operator	Description
Header	Business edition or higher	• Includes and Does not include • Equals and Does not equal • Length equals, Length more than, and Length less than • Does not exist	The header of an access request, which is used to customize the HTTP header.
URLPath	Business edition or higher	• Includes and Does not include • Equals and Does not equal • URI Path Match • Regular Expression	The URL path of an access request.

Logical operators

Logical operator	Description
Has and Does not have	Whether the match field has the match content.
Includes and Does not include	Whether the match field includes the match content.
Equals and Does not equal	Whether the match field equals the match content.
Length equals, Length more than, and Length less than	The length of the match field is equal to, greater than, or less than that of the match content.
Does not exist	The match field does not exist.
Value less than, Value equals, and Value more than	The value of the match field is less than, equal to, or greater than that of the match content.
URI Path Match	The prefix of the match field contains the match content.
Regular Expression	The match field matches the regular expression defined in the match content.

7.Customize protection rule groups

You can use default protection rule groups provided by Web Application Firewall (WAF) to customize your rule groups for a specific protection feature, such as web application protection, known as RegEx Protection Engine. If the default protection rule groups cannot meet your business requirements, we recommend that you customize protection rule groups to protect your website.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
  - If the instance is deployed in **mainland China**, the instance must be of the **Business** edition or higher.
  - If the instance is deployed **outside mainland China**, the instance must be of the **Enterprise** edition or higher.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Context

You can customize protection rule groups only for the **RegEx Protection Engine** feature. For more information, see Configure the RegEx Protection Engine.

Use a custom rule group

To use a custom rule group, you must complete the following steps:

1. Create a rule group: Create a custom rule group for a specific protection feature.
2. Apply the rule group: Apply the created rule group to your website.

Create a rule group

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **System Management > Protection Rule Group**.
4. (Optional)On the **Protection Rule Group** page, click the tab of the protection feature that you want to manage.

   > **Note** You can skip this step because only the **RegEx Protection Engine** feature supports protection rule groups. You are directly redirected to the **Web Application Protection** tab.

   The **Web Application Protection** tab displays the default and custom rule groups.

   - Default rule group: The name of a default rule group can be **Loose rule group**, **Medium rule group**, or **Strict rule group**.

You can click the number in the **Built-in Rule Number** column to view information about the built-in rules.

> **Note** Default rule groups cannot be edited or deleted.

- Custom rule group: You can create a rule group on the **Protection Rule Group** page.

5. Click **Create Rule Group**.

> **Note** You can create a maximum of 10 rule groups for the web application protection feature.

6. Specify the parameters in the **Create Rule Group** wizard.
   i. **Specify rule information**. Configure the following parameters and click **Next: Apply to Websites**.

Create Rule Group

Parameter	Description
Rule Group Name	Enter a name for the rule group. The rule group name is used to identify the rule group. We recommend that you enter an informative name.
Rule Group Template	Select a rule group template from which you want to select rules for the rule group. Valid values: ■ **Strict rule group**: contains 1,068 rules by default. ■ **Medium rule group**: contains 1,039 rules by default. ■ **Loose rule group**: contains 1,031 rules by default. Different rule group templates contain different rules. After you select the rule group template and turn on the Automatic Update switch, each time a rule in the rule group template is updated, the rule is also updated in the created rule group.
Description	Enter a description for the rule group.
Automatic Update	If you turn on this switch, each time a rule in the rule group template is updated, the rule is also updated in the created rule group. **Note** Some custom rule groups do not support the automatic update feature. In this case, we recommend that you create rule groups to replace these rule groups.

Parameter	Description
Select Rule	Select a rule for the rule group. The **Selected Rules** tab lists all rules in the rule group template that you select. You must select rules that are not applicable or may cause false positives, and then click **Remove Selected Rules**. You can use the filter or search feature to find rules you want to manage. You can filter rules by **Protection Type**, **Application Type**, or **Risk Level** or enter a rule name or ID to search for a rule. ■ **Risk Level**: indicates the risk level of web attacks. Valid values: **High**, **Medium**, and **Low**. ■ **Protection Type**: indicates the type of web attacks. Valid values: **SQL Injection**, **Cross-site Script**, **Code Execution**, **CRLF**, **Local File Inclusion**, **Remote File Inclusion**, **Webshell**, **CSRF**, and **Others**. ■ **Application Type**: indicates the type of the protected web application. Valid values: **Common**, **Wordpress**, **Dedecms**, **Discuz**, **Phpcms**, **Ecshop**, **Shopex**, **Drupal**, **Joomla**, **Metinfo**, **Struts2**, **Spring Boot**, **Jboss**, **Weblogic**, **Websphere**, **Tomcat**, **Elastic Search**, **Thinkphp**, **Fastjson**, **ImageMagick**, **PHPwind**, **phpMyAdmin**, and **Others**.

> **Note** If you do not want to apply a rule group immediately after you create it, click **Save**. You can edit the group again after you complete the step.

ii. (Optional)**Apply to a website.** Select the website to which you want to apply the new rule group from the **Websites not Added to WAF** section and add them to the **Websites Added to WAF** section.

> **Notice** You must apply one rule group to each website.

Apply to Website

iii. Click **Save**.

You can view the new rule group in the rule group list and select the website to which you want to apply the rule group. For more information, see Apply the rule group.

After you create the rule group, you can view the creation time of a rule group in the **Updated On** column on the **Protection Rule Group** page and determine whether to update the rule group.

Apply the rule group

After you create a custom rule group, you can apply it by using one of the following methods:

- On the **Protection Rule Group** page, apply the rule group to a website. The following steps are provided for this scenario.
- On the **Website Protection** page, select a custom rule group from the Protection Rule Group drop-down list in the RegEx Protection Engine card.

For more information, see Configure the RegEx Protection Engine.

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **System Management > Protection Rule Group**.
4. (Optional)On the **Protection Rule Group** page, click the tab of protection feature you want to manage.

   > **Note** You can skip this step because only the **web application protection** feature supports the rule group. You are directly redirected to the **Web Application Protection** tab.

5. In the **Protection Rule Group** list on the **Web Application Protection** tab, find the rule group that you want to apply and click **Apply to Website** in the Action column.
6. On the **Apply to Website** page, select the website to which you want to apply the rule group from the **Websites not Added to WAF** section, add them to the **Websites Added to WAF** section, and then click **Save**.

   > **Notice** You must apply one rule group to each website.

   After you complete the operation, you can view the website in the **Website** column on the **Protection Rule Group** page.

Related operations

You can perform the following operations to manage the created rule group on the **Protection Rule Group** page:

- **Copy**: allows you to copy the configurations of the rule group.

  The following figure shows the Copy Rule Group page. On this page, you can modify **Rule Group Name**, **Description**, and **Automatic Update**, but cannot modify **Rule Group Template** and rule settings. If you need to modify the rule settings, we recommend that you copy the rule group and modify the rule settings in the copied rule group.

  > **Note** Some custom rule groups cannot be copied because they do not support the automatic update feature. In this case, we recommend that you create rule groups to replace these rule groups.

- **Edit**: allows you to modify the name, description, and rule settings of the rule group. Default rules cannot be edited.
- **Delete**: allows you to delete the rule group. Default rules cannot be deleted.

Before you delete a custom rule group, make sure that it is not applied to any website. If the rule group is applied to a website, apply a different rule group to the website before you delete the rule group.

8.Best practices for protection settings

8.1. Best practices for website protection

If this is the first time you add a domain name to WAF, we recommend that you learn more about website protection. This topic describes how to select protection modules and configure protection policies of WAF from the perspective of different roles to meet business requirements in different scenarios. By reading this topic, you can understand the protection logic of WAF.

Prerequisites

Your website configurations are added to WAF. For more information, see Add websites.

Usage notes

All the descriptions in this topic are based on the fact that you have enabled the recommended website protection features. If you have not enabled such features, enable and configure them based on the feature descriptions.

Unless otherwise specified, the recommended website protection features are configured on the **Website Protection** page. Perform the following operations to go to the **Website Protection** page:

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Settings > Website Protection**.
4. In the upper part of the **Website Protection** page, select the domain name for which you want to configure the whitelist.

Overview

This topic provides the recommended website protection features based on roles and business requirements. You can decide which features to enable based on your business requirements.

- I am new to WAF. I am unsure of my security needs
- I am an O&M engineer. I require reliable services and convenient troubleshooting
- I am a security engineer. I need to comprehensively prevent web intrusion
- I want to achieve the strongest protection and radically block attacks
- My website is often crawled and is at risk of data breach or tampering

I am new to WAF. I am unsure of my security needs

You may have purchased a WAF instance based on a need for classified protection or the intention to improve the security level of your enterprise. In either case, you can add your website configurations to WAF and then use the default protection settings of WAF. The default protection settings are sufficient to protect your website from the majority of basic web threats.

We recommend that you browse the **Overview** and **Security report** pages in the Web Application Firewall console to understand the security situations of your business and the attacks it may face. For more information, see the following topics:

- View overall information
- View security reports

I am an O&M engineer. I require reliable services and convenient troubleshooting

We recommend that you enable the following website protection features after you add your website configurations to WAF:

- **Website Whitelisting**: You can configure a whitelist to allow requests that meet the specific conditions without the need to perform a check.

  Operations: On the **Website Protection** page, click **Website Whitelisting** in the upper-right corner. On the Website Whitelisting page, create a whitelist. For more information, see Configure a website whitelist.

  Website Whitelisting

  To implement more precise protection, you can also configure a whitelist for a specific protection module. For more information, see the following topics:

  - Whitelist for Web Intrusion Prevention: Trusted access requests are not detected by RegEx Protection Engine or Big Data Deep Learning Engine.
  - Whitelist for Data Security: Trusted access requests are not detected by Data Leakage Prevention, Website Tamper-proofing, or Account Security.
  - Whitelist for Bot Management: Trusted access requests are not detected by Bot Threat Intelligence, Data Risk Control, Intelligent Algorithm, or App Protection.
  - Whitelist for Access Control/Throttling: Trusted access requests are not detected by HTTP Flood Protection, IP Blacklist, Scan Protection, or Custom Protection Policy.

- **IP Blacklist**: This feature allows you to configure an IP address blacklist to block requests from IP addresses and CIDR blocks that are irrelevant to your business and from IP addresses in specific regions. For example, if a local government forum is accessed only by local IP addresses, you can add IP addresses from other regions to a regional blacklist. If your website does not have users outside China, you can add all the regions outside China to a regional blacklist.

  IP Blacklist

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **IP Blacklist** card and configure the required parameters. For more information, see Configure a blacklist.

- **Custom Protection Policy**: This feature allows you to customize access control lists (ACLs) or throttling policies. For example, you can allow access to an API only from specific IP addresses or user agents and configure an upper limit for specific types of requests. You can also use this feature to defend against HTTP flood attacks, crawler attacks, and some special web attacks.

  Custom Protection Policy

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **Custom Protection Policy** card and configure the required parameters. For more information, see Create a custom protection policy.

- **Account Security**: This feature allows you to monitor user authentication-related interfaces, such as the interfaces used for registration and logon, to detect events that may pose a threat to user credentials. These threats include credential stuffing, brute-force attacks, account registrations launched by bots, weak password sniffing, and SMS interface abuse.

  Account Security tab

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Account Security**. In the Account Security card, click **Settings** and configure the required parameters. For more information, see Configure account security.

I am a security engineer. I need to comprehensively prevent web intrusion

We recommend that you enable the following website protection features after you add your website configurations to WAF:

- **Decoding Settings**: This feature allows you to specify a decoding method for the WAF engine based on your business coding scheme to maximize protection for your website. A proper decoding method allows the WAF engine to effectively identify traffic and achieve precise prevention. WAF uses all the 13 decoding methods by default. You can filter out unnecessary methods to avoid unnecessary parsing and false blocking.

  Decoding Settings

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Web Intrusion Prevention** section, find **RegEx Protection Engine**. In the RegEx Protection Engine card, specify **Decoding Settings**. For more information, see Configure the RegEx Protection Engine.

- **Protection Rule Group**: This feature allows you to select protection rules from a built-in protection rule set based on the form, framework, and middleware of your business system. You can use these rules to customize a rule group to prevent web attacks and apply the rule group to your website. We recommend that you use this feature to configure web intrusion prevention policies for your website. If you want to configure prevention policies for a single URL, we recommend that you use the Custom Protection Policy feature.

  Operations: Log on to the Web Application Firewall console and choose **System Management** > **Protection Rule Group**. On the Protection Rule Group page, customize the rule group for web attack prevention and apply the rule group to your website. For more information, see Customize protection rule groups.

  Default rule groups for web application protection

- **Custom Protection Policy**: This feature allows you to customize access control lists (ACLs) or throttling policies. For example, you can allow access to an API only from specific IP addresses or user agents and configure an upper limit for specific types of requests. You can also use this feature to defend against HTTP flood attacks, crawler attacks, and some special web attacks.

  Custom Protection Policy

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **Custom Protection Policy** card and configure the required parameters. For more information, see Create a custom protection policy.

- **Big Data Deep Learning Engine** (**Warn** mode): The Big Data Deep Learning Engine is trained based on the intelligence of hundreds of millions of samples generated on the cloud every day. This makes up for the weaknesses of the RegEx Protection Engine, especially in terms of defense against deformed or unknown attacks. We recommend that you enable the Big Data Deep Learning Engine in

Warn mode. Then, observe the anomalies that are detected by the engine over a period of one to two weeks. If the engine works properly, switch to the **Block** mode.

Big Data Deep Learning Engine

Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Web Intrusion Prevention** section, find **Big Data Deep Learning Engine**. In the Big Data Deep Learning Engine card, turn on **Status** and set **Mode** to **Warn**. For more information, see Configure the Big Data Deep Learning Engine.

- **Positive Security Model** (**Warn** mode): The positive security model is built based on the learning of the traffic in the current domain name. The model specifies the types and lengths of request parameters and whether the parameters are required. After the model is built, if a request does not match the characteristics described in the model, an alert is generated. The positive security model in **Warn** mode allows you to effectively detect anomalies and threats to your business. If the detected requests are useless to your business, you can enable the **Block** mode.

  Positive Security Model

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Advanced protection** section, find **Positive Security Model**. In the Positive Security Model card, turn on **Status** and set **Mode** to **Warn**. For more information, see Configure the positive security model.

- **Scan Protection** (**Blocking IPs Initiating High-frequency Web Attacks**, **Directory Traversal Prevention**, **Scanning Tool Blocking**, and **Collaborative Defense**): This feature helps reduce the threats generated by your scanner from multiple dimensions, such as intelligence, scanner features, and scan behavior.

  Scan protection

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. In the **Scan Protection** card, enable all functions and specify appropriate thresholds. For more information, see Configure scan protection.

I want to achieve the strongest protection and radically block attacks

We recommend that you enable the following website protection features after you add your website configurations to WAF:

- **RegEx Protection Engine** (**Strict rule group**)

  RegEx Protection Engine - Strict rule group

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. In the **Web Intrusion Prevention** section, find **RegEx Protection Engine**. In the RegEx Protection Engine card, set **Protection Rule Group** to **Strict rule group**. For more information, see Create a custom protection policy.

- **Big Data Deep Learning Engine** (**Block** mode): The Big Data Deep Learning Engine is trained based on the intelligence of hundreds of millions of samples generated on the cloud every day. This makes up for the weaknesses of the RegEx Protection Engine, especially in terms of defense against deformed or unknown attacks. To achieve the strongest protection, we recommend that you enable the **Block** mode.

  Big Data Deep Learning Engine - Block mode

Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Web Intrusion Prevention** section, find **Big Data Deep Learning Engine**. In the Big Data Deep Learning Engine card, turn on **Status** and set **Mode** to **Block**. For more information, see Configure the Big Data Deep Learning Engine.

- **Positive Security Model** (**Block** mode): The positive security model is built based on the learning of the traffic in the current domain name. The model specifies the types and lengths of request parameters and whether the parameters are required. After the model is built, if a request does not match the characteristics described in the model, an alert is generated. To achieve the strongest protection, we recommend that you enable the **Block** mode.

  Positive Security Model - Block mode

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Advanced Protection** section, find **Positive Security Model**. In the Positive Security Model card, turn on **Status** and set **Mode** to **Block**. For more information, see Configure the positive security model.

- **Scan Protection** (**Blocking IPs Initiating High-frequency Web Attacks**, **Directory Traversal Prevention**, **Scanning Tool Blocking**, and **Collaborative Defense**): This feature helps reduce the threats generated by your scanner from multiple dimensions, such as intelligence, scanner features, and scan behavior.

  Scan protection

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. In the **Scan Protection** card, enable all functions and specify appropriate thresholds. For more information, see Configure scan protection.

- **IP Blacklist**: This feature allows you to configure an IP address blacklist to block requests from IP addresses and CIDR blocks that are irrelevant to your business and from IP addresses in specific regions. For example, if a local government forum is accessed only by local IP addresses, you can add IP addresses from other regions to a regional blacklist. If your website does not have users outside China, you can add all the regions outside China to a regional blacklist.

  IP Blacklist

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **IP Blacklist** card and configure the required parameters. For more information, see Configure a blacklist.

My website is often crawled and is at risk of data breach or tampering

We recommend that you enable the following website protection features after you add your website configurations to WAF:

- **Data Risk Control**: This feature is best suited for defending against bot traffic that is generated by scripts or automated tools and destined for specific APIs for logon, registration, and order placing.

  > **Note** Data risk control depends on JavaScript plug-ins and is applicable only to web pages. Do not use this feature in applications. If you are not sure whether this feature is suitable for your API, submit a ticket or contact the technical support by using DingTalk.

  Data Risk Control

  Operations: On the **Website Protection** page, click the **Bot Management** tab. In the **Data Risk Control** card, configure the required parameters. For more information, see Configure data risk control.

- **Data Leakage Prevention**: This feature allows you to filter sensitive information in the returned content, such as abnormal pages and keywords, from the server. The sensitive information includes ID numbers, bank card numbers, telephone numbers, and sensitive words.

  Data Leakage Prevention

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Data Leakage Prevention**. In the Data Leakage Prevention card, configure the required parameters. For more information, see Configure data leakage prevention.

- **Website Tamper-proofing**: This feature allows you to lock specified web pages to avoid content tampering. When a locked web page receives a request, a cached page you have preconfigured is returned.

  Website Tamper-proofing

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Website Tamper-proofing**. In the Website Tamper-proofing card, configure the required parameters. For more information, see Configure website tamper-proofing.

- **Custom Protection Policy**: You can enable JavaScript verification for frequently crawled static web pages at one click to block most scripts and automated programs. You can also use fine-grained frequency control to enable slider verification for sessions from which access requests are initiated at an abnormally high frequency.

  Custom Protection Policy

  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **Custom Protection Policy** card and configure the required parameters. For more information, see Create a custom protection policy.

- **Account Security**: This feature allows you to monitor user authentication-related interfaces, such as the interfaces used for registration and logon, to detect events that may pose a threat to user credentials. These threats include credential stuffing, brute-force attacks, account registrations launched by bots, weak password sniffing, and SMS interface abuse.

  Account Security tab

  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Account Security**. In the Account Security card, click **Settings** and configure the required parameters. For more information, see Configure account security.

- **Allowed Crawlers**: This feature maintains a whitelist of authorized search engines, such as Google, Bing, Baidu, Sogou, 360, and Yandex. The crawlers of these search engines are allowed to access the specified domain names.

  Allowed Crawlers

  Operations: On the **Website Protection** page, click the **Bot Management** tab. In the **Allowed Crawlers** card, configure the required parameters. For more information, see Configure the allowed crawlers function.

- **Bot Threat Intelligence**: This feature provides information about suspicious IP addresses used by dialers, data centers, and malicious scanners. This feature also maintains an IP address library of malicious crawlers and prevents crawlers from accessing your website or specific directories.

  Bot Threat Intelligence

  Operations: On the **Website Protection** page, click the **Bot Management** tab. In the **Bot Threat Intelligence** card, configure the required parameters. For more information, see Set a bot threat intelligence rule.

- **App Protection**: This feature provides secure connections and anti-bot protection for native apps and can identify proxies, emulators, and requests with invalid signatures.

  App Protection

  Operations: On the **Website Protection** page, click the **Bot Management** tab. In the **App Protection** card, configure the required parameters. For more information, see Configure application protection.

8.2. Best practices for using RegEx Protection Engine

This topic describes best practices for using RegEx Protection Engine provided by Web Application Firewall (WAF).

Scenario

WAF protects your website against web attacks, such as SQL injection, XSS attacks, remote code execution, and webshell attacks. For more information about web attacks, see Definitions of common web vulnerabilities.

> **Note** WAF cannot defend against server intrusion caused by host security issues, such as unauthorized access to ApsaraDB for Redis or ApsaraDB RDS for MySQL.

Protection policies

By default, **RegEx Protection Engine** is enabled and Protection Rule Group is set to Medium rule group after you add your website configurations to WAF. This blocks common attacks. To view the settings, go to the **Website Protection** page and view the **RegEx Protection Engine** settings. For more information about how to configure RegEx Protection Engine, see Configure RegEx Protection Engine.

Web application protection

Protection status description

- **Status**: Turn on or off the switch to enable or disable the RegEx Protection Engine function. This function is enabled by default.
- **Mode**: Specify the actions that you want WAF to take on attack requests when the attack requests are detected. Valid values:
  - Block: WAF automatically blocks attack requests and logs attacks in the backend.
  - Warn: WAF does not block attack requests but logs attacks in the backend.
- **Protection Rule Group**: Specify a set of protection rules that you can apply. Valid values:
  - **Medium rule group**: blocks common web application attacks by using a standard way. These attacks can bypass protection policies.
  - **Strict rule group**: blocks web application attacks by using a strict way. These attacks can bypass complex protection policies.
  - **Loose rule group**: blocks common web application attacks.

> **Note** These settings take effect only when you enable RegEx Protection Engine.

If you are using WAF Business or Enterprise in mainland China or WAF Enterprise in regions outside mainland China, you can customize protection rule groups. The custom rule groups combine all protection rules provided by WAF and provide specific protection policies for your website. For more information, see Customize protection rule groups.

Recommended configurations

- If you are not clear about the characteristics of your business traffic, we recommend that you set Mode to **Warn**. After one or two weeks, analyze the attack logs in this mode.
  - If the attack logs show that normal traffic is not blocked, you can set Mode to **Block**.
  - If the attack logs show that normal traffic is blocked, you can contact an Alibaba Cloud security expert to resolve the issue.
- If you add phpMyAdmin and development technology forums to WAF for protection, WAF may block normal requests. If this occurs, we recommend that you contact an Alibaba Cloud security expert to resolve this issue.
- You need to pay attention to the following issues:
  - Do not pass original SQL statements or JavaScript code in the HTTP requests of your normal business.
  - Do not use special keywords (such as UPDATE and SET) in the path for normal business URLs, such as `www.example.com/abc/update/mod.php?set=1`.
  - Do not upload files that exceed 50 MB by using a browser. We recommend that you upload the files by using OSS or other methods. For more information about how to use OSS, see Get started with Object Storage Service.

Protection effects

After you enable RegEx Protection Engine, you can view its protection records. To view the records, click **Security report**. On the page that appears, click **Web Security** and view the report on the **Web Intrusion Prevention** tab. For more information, see View security reports.

Web Intrusion Prevention displays attack records in the last 30 days. The section below the report shows the attack records. You can select **Regular Protection**, find an attack record, and click **View details** to query the attack details. The following figure shows an SQL injection request that is blocked by WAF.

> **Note** If you find that WAF blocks normal traffic, we recommend that you use the **Whitelisting Rules** function to configure a whitelist for the blocked URLs and then contact an Alibaba Cloud security expert to find a solution. For more information about how to configure a whitelist, see Configure a whitelist for Web Intrusion Prevention.

Rule updates

WAF updates protection rules and releases protection bulletins in a timely manner to fix known and zero-day vulnerabilities. To query **Rule updates notice**, go to the **Product Information** page. For more information, see View product information.

Note Web attacks typically have more than one proof of concept (POC). Alibaba Cloud security experts conduct a thorough analysis of vulnerability principles to ensure that published web protection rules cover all disclosed and undisclosed vulnerabilities.

8.3. Best practices for preventing HTTP flood attacks

This topic describes common types of HTTP flood attacks and how to defend against them by using protection policies offered by WAF.

Overview

You can determine which protection policies to use based on the attack type.

- Volumetric and high-rate HTTP flood attacks
- Attacks from regions outside China and public clouds
- Malformed packets
- API abuse
- Malicious scans
- App attacks
- Malicious crawlers

Volumetric and high-rate HTTP flood attacks

In volumetric HTTP flood attacks, a zombie server sends requests at a higher frequency than a normal server does. To prevent such attacks, the most effective measure is to limit the request rate of request sources. WAF provides the **Rate Limiting** function for this purpose. You can configure this function from the **Custom Protection Policy** page. For more information, see Create a custom protection policy.

You can configure a rule, as shown in the following figure. The rule blocks all IP addresses that initiate more than 1,000 requests in a 30 second interval to any path under the domain name. The blocking period lasts for 10 hours. This rule is used to protect small and medium-sized websites.

Rate limiting

You can modify the protected path, adjust the threshold, and select the optimal action to best suit your protection requirements. For example, to prevent credential stuffing on logon endpoints, you can set **Matching field** to URL and **Matching content** to `/login.php`, and block IP addresses that send more than 20 requests to access the path within 60 seconds.

Example rate limiting

Note the following points when you configure HTTP flood protection policies:

- **Captcha** and **Strict Captcha** in the **Action** drop-down list aim to verify whether requests originate from a human or an automation script. You can use these two actions to protect common and HTML5 web pages, but not native apps or APIs. To protect the native apps and APIs, set **Action** to **block**.
- You can configure whitelist policies for APIs or IP addresses that may be mistakenly blocked by HTTP flood protection on the **Access Control/Throttling** tab. For more information, see Configure a whitelist for Access Control/Throttling.
- Do not select the **Protection-emergency** mode for native apps or APIs in the **HTTP Flood**

Protection section.

If you have purchased an instance of the WAF Enterprise edition, you can configure rate limiting by using custom statistical objects, IP addresses, and sessions. Blocking IP addresses may affect NAT. You can use cookies or parameters that identify users as statistical objects. In the following example, the request rate is calculated based on the cookie that is used to identify the user, and Captcha is used to verify the requests. Assume that the cookie format is as follows: `uid=12345`.

Cookie

Attacks from regions outside China and public clouds

A large portion of HTTP flood attacks originate from regions outside China, on-premises data centers, and public clouds.

If your website targets users inside China, you can block requests from regions outside China to mitigate this type of attack. WAF provides the **Area-based IP Blacklist** function for this purpose. For more information, see Configure a blacklist

封禁区域

If you need to block the crawler IP addresses of common IP libraries, such as the CIDR blocks of Alibaba Cloud, Tencent Cloud, and on-premises data centers, you can use the **Bot Threat Intelligence** function on the **Bot Management** tab.

> **Note** Many crawlers are deployed on ECS instances. Users do not access your services by using the source IP addresses of public clouds or on-premises data centers.

Example: You can use the following bot threat intelligence rule to block accesses from the crawler IP addresses of Tencent Cloud. For more information, see Set a bot threat intelligence rule.

Bot threat intelligence rule for Tencent Cloud

Malformed packets

Malicious requests in HTTP flood attacks are specially crafted and contain malformed packets. Malformed packets have the following features:

- Abnormal or malformed User-Agent string: has characteristics of automation tools (such as Python), is in an incorrect format (such as `Mozilla///`), or is impossible to be used in normal requests (such as `www.baidu.com`). If abnormal or malformed User-Agent strings are detected, block the requests.
- Unusual User-Agent string: Promotional HTML5 pages that target WeChat users are supposed to be accessed through WeChat. It is unusual if the User-Agent string indicates that the request is sent from a Windows desktop browser, such as Microsoft Internet Explorer 6.0. If unusual User-Agent strings are detected, block the requests.
- Abnormal referer field: If a request does not have a referer field or has a referer field that identifies the addresses of illegitimate websites, block the request. However, when a user visits your homepage or your website for the first time, the request may not contain the referer field. If a URL can only be accessed by using redirects, you can decide whether to block the URL based on the referer field.
- Abnormal cookie: Similar to the referer field, a normal request contains cookies that identify the requested websites, unless it is the first time for the user to visit your website. Malicious requests in HTTP flood attacks typically do not contain any cookie information. You can block access requests without cookies.
- Missing HTTP headers: Normal requests contain authorization headers while malicious requests do

not.

- Incorrect request methods: If an API has only received POST requests before but is now overwhelmed by GET requests, you can block these GET requests.

You can analyze the features of requests and set Protection Type to **ACL** from the **Custom Protection Policy** page to block malicious requests. For more information, see Create a custom protection policy.

Configuration examples:

- Example 1: Block requests that do not contain cookies.

  Block requests that do not contain cookies

- Example 2: Block requests that do not contain authorization headers.

  拦截不带authorization

API abuse

We recommend that you use the data risk control function to protect important APIs from attacks. These APIs include logon, registration, voting, and SMS verification APIs.

Data risk control injects a JavaScript snippet into your website and collects information about user behaviors and environment variables to determine whether requests originate from a human or an automation script. Data risk control makes decisions based on CAPTCHA rather than the request rate or the source IP address. The function mitigates low-frequency attacks very effectively.

> **Notice** Data risk control checks whether requests contain authentication parameters required by all normal requests to identify malicious requests. The function is not suitable for environments where JavaScript is not supported, such as APIs and native apps. To prevent false positives, we recommend that you test data risk control in the test environment before you enable it. Alternatively, you can use the observation mode and contact engineers before you enable the prevention mode.

For more information, see Configure data risk control.

Malicious scans

A large number of malicious scans pose a serious threat to the performance of your servers. Apart from rate limiting, you can also use the **Scan Protection** function to enhance security. Scan protection supports the following settings:

- **Blocking IPs Initiating High-frequency Web Attacks**: automatically blocks client IP addresses that initiate high-frequency web attacks.
- **Directory Traversal Prevention**: automatically blocks client IP addresses that initiate multiple directory traversal attacks in a short period of time.
- **Scanning Tool Blocking**: automatically blocks access requests from IP addresses defined in the common scan tools or the Alibaba Cloud malicious IP library.
- **Collaborative Defense**: automatically blocks access requests from IP addresses defined in the Alibaba Cloud malicious IP library.

For more information, see Configure scan protection.

Scan protection

App attacks

In addition to the preceding measures, you can also use SDK to enhance protection.

After you integrate the SDK with your app, all incoming requests are verified before they are sent to your server. The device information and request signature are combined to determine whether the requests are from legitimate apps. Requests that do not originate from official apps are automatically blocked. This ensures that only valid requests are served. You do not need to analyze the patterns of invalid requests.

To use the SDK, you must enable **App Protection**. For more information, see Configure application protection.

Malicious crawlers

For informational websites that offer services such as credit reports, apartment rentals, airline tickets, and e-book reading, malicious crawlers can significantly increase the bandwidth usage and server workload, and even cause data leaks. If the preceding measures cannot prevent against malicious crawlers, we recommend that you enable and use the **Bot Management** feature for more effective protection. For more information, see Configure a whitelist for Bot Management.

8.4. Best practices for blocking malicious crawlers

This topic describes the best practices for blocking malicious crawlers by using WAF.

Background information

Malicious crawlers come in various types. They constantly change their crawling methods to bypass anti-crawling policies configured by website administrators. Therefore, it is impossible to block all malicious crawlers by using fixed rules. To block the malicious crawlers, WAF provides a bot management feature. This feature has close relations to the characteristics of your business. However, this feature can deliver optimal protection only with the help of security experts.

If you need stronger protection against malicious crawlers or need help from security experts, we recommend that you use the **bot management** feature. The feature provides malicious crawler IP libraries and dynamically updates IP libraries of various public clouds and data centers based on network-wide threat intelligence of Alibaba Cloud in real time. This helps you block malicious requests from the addresses in the malicious crawler IP libraries. For more information, see Configure the bot management whitelist.

> **Note** Bot management is a value-added service that is separately enabled when you purchase or upgrade WAF.

In addition to the bot management feature, you can also use the **custom protection policy** and **IP blacklist** functions to configure specific crawler blocking policies based on the following characteristics of malicious crawlers.

Risks and characteristics of malicious crawlers

Normal crawler requests typically contain the `xxspider` keyword in the User-Agent field and have the following characteristics: lower request rate, scattered URLs, and wide time range. If you run a reverse `nslookup` or `tracert` command on a legitimate crawler, you can obtain the source IP address that initiates the crawler request. For example, if you run the reverse nslookup command with the IP address of the Baidu crawler, you can obtain the source IP address of the crawler.

View origin server information

Malicious crawlers may send a large number of requests to a specific URL or port of a domain name during a certain period of time, for example, HTTP flood attacks that are disguised as crawlers or requests that are disguised by third parties to crawl targeted sensitive information. A large number of malicious requests can cause a sharp rise in CPU utilization, website access failure, and service interruptions.

Create a custom protection policy

You can use the custom protection policy function to combine key fields such as User-Agent and URL to filter out malicious crawler requests. For more information, see Create a custom protection policy.

Sample configuration:

- Log on to the WAF console. On the **Custom Protection Policy** page, configure the following ACL rule to allow only Baidu spiders.

  New rule to allow Baidu spiders

- Log on to the WAF console. On the **Custom Protection Policy** page, configure the following ACL rule to prevent all crawlers from accessing the */userinfo* directory.

  Block crawlers

> **Note** The method used to restrict the User-Agent field is ineffective for specially crafted crawler attacks. For example, an attacker may include a baidu character in the User-Agent field of the malicious crawler request to disguise the malicious crawler as a Baidu crawler. This way, the ACL rule does not block the malicious crawler request. In addition, an attacker can hide the crawler identity by removing the spider character in the User-Agent field. This way, the ACL rule does not block the attack.

For high-frequency malicious crawler requests, you can configure **Rate Limiting** on the Custom Protection Policy page to block domain-specific IP addresses that send requests exceeding the threshold.

You can configure a rule, as shown in the following figure. If an IP address sends requests to any path under the domain more than 1,000 times in 30 seconds, the IP address is blocked for 10 hours.

Rule limiting

If you have purchased a WAF Enterprise instance, you can use custom statistical objects in addition to IP addresses and sessions during the rate limiting configuration. Blocking IP addresses may affect NAT. You can use cookies or default parameters that identify users as statistical objects. In the following example, select Cookie for Statistical Object and Captcha for Action. Assume that the cookie format is as follows: `uid=12345`.

Cookie

Configure an IP address blacklist

If a large number of malicious crawler requests are from the same region and normal requests are not from this region, you can enable **IP Blacklist** to block all access requests from this specific region. For more information, see Configure a blacklist.

Configuration example: You can log on to the WAF console, go to the **IP Blacklist** page, and then configure the following rule to block access requests from IP addresses outside China.

8.5. Account security best practices

Web Application Firewall (WAF) provides an account security feature that helps you identify account risks. This topic describes how to protect interfaces in different scenarios. You can follow the instructions in this topic to better protect interfaces on which user authentication is performed.

Context

WAF supports the account security feature that detects account risks. This feature monitors interfaces related to user authentication, such as registration and logon interfaces, and detects risks on these interfaces. These risks include credential stuffing, brute-force attacks, spam registration, weak password sniffing, and SMS interface abuse. After interfaces are added to WAF, you can view detection results in WAF security reports. For more information, see Configure account security.

Use verification services to protect common and HTML5 web pages

Verification services are the easiest and most effective approaches to protect interfaces. The integration of verification services into your business typically requires minor code changes. It may take one or two business days to modify the code.

Common verification methods can block direct calls launched from simple tools or scripts. However, due to the adaptation of attack methods and tools, the common verification methods can be easily bypassed. We recommend that you use professional verification services to better protect interfaces against attacks.

Use SDK signatures to protect native apps

Verification services may be unsuitable for native apps. Alibaba Cloud provides an SDK solution for native apps. The solution collects the information about the hardware and environment of a mobile device, calculates signatures, and verifies signatures of requests. This ensures that only requests from verified apps are directed to the origin server. Requests sent from scripts, automated programs, simulators, and other unverified sources are blocked.

> **Note** To use the SDK solution, you must enable **App Protection** in the WAF console. For more information, see App protection overview.

Configure frequency control to block attack sources

Frequency control helps you identify requests that contain a common field among a large number of requests. You can specify the maximum occurrences of the common field. The source of the requests is blocked when the maximum occurrences are exceeded. Traditional protection methods typically block malicious IP addresses. Malicious requests sent from proxies or rotating IP addresses may contain the same token, for example, the same UID, in their cookies. In this case, you can configure the maximum occurrences based on the cookies to block malicious accounts.

WAF provides **Rate Limiting** for this purpose. You can configure rate limiting on the **Custom Protection Policy** page, as shown in the following figure. For more information, see Create a custom protection policy.

> **Note** All WAF editions allow you to use IP addresses and sessions as statistical objects. WAF Enterprise allows you to use more objects, such as custom cookies, custom headers, and custom parameters.

Analyze suspicious requests

Malicious requests have certain common characteristics. The following examples describe common characteristics among malicious requests.

- Incomplete HTTP headers. Malicious requests may exclude certain fields, such as Referer, Cookie, or Content-Type.
- Abnormal User-Agent values. User-Agent headers used in requests that target Java or Python-based websites are found in requests sent to common websites. User-Agent headers used in requests initiated from desktop browsers are found in requests sent to WeChat mini programs. In these cases, requests that contain abnormal User-Agent headers may be malicious.
- Missing cookies. Typically, multiple cookies are used in an application. Common cookies include SessionID, userid, deviceid, and lastvisit. However, crawlers may include only one or two cookies that are required for retrieving information and exclude other cookies that identify users.
- Abnormal parameters. Similar to missing cookies, some parameters are not required for crawlers to retrieve information. Crawlers may exclude or repeatedly submitted these parameters in requests.
- Suspicious fields. Suspicious fields may be contained in email addresses, phone numbers, and account information.

We recommend that you use the Log Service of WAF feature to query logs. This feature allows you to analyze request characteristics, such as top IP addresses and the proportion of requests with certain characteristics to total requests.

> **Note** To use the Log Service of WAF feature, you must enable **Log Service** from the WAF console. For more information, see Enable Log Service for WAF.

Enable credential stuffing and bot threat intelligence

WAF provides a **Bot Management** feature. This feature uses algorithms and identifies malicious IP addresses from credential stuffing attacks detected by Alibaba Cloud. A credential stuffing IP address blacklist is created and updated dynamically. You can use the **Bot Threat Intelligence** function from the Bot Management tab to set the credential stuffing IP address blacklist to the Monitor, Block, or Captcha mode. For more information, see Set a bot threat intelligence rule.

> **Note** You must enable the **Bot Management** feature before you can use the **Bot Threat Intelligence** function.

8.6. Best practices for using custom rule groups to provide enhanced protection

If you find that RegEx Protection Engine of WAF blocks normal requests to your website, you can customize protection rule groups to avoid this issue.

Prerequisites

- A WAF instance is purchased. The instance must meet the following requirements:
  - The instance is billed on a subscription basis.
  - If the instance is deployed in **mainland China**, the instance must be of the **Business** edition or higher.
  - If the instance is deployed **outside mainland China**, the instance must be of the **Enterprise** edition or higher.

  For more information, see Purchase a WAF instance.
- Your website is added to the WAF console. For more information, see Add websites.

Context

To address this issue, you must identify the protection rule that causes the issue, create a custom rule group for the affected domain name, and then remove the protection rule from the custom rule group.

Procedure

1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or **International**, in which the instance is deployed.
3. In the left-side navigation pane, click **Security report**.
4. Identify the ID of the protection rule that causes false positives.
   i. On the **Web Security** tab, click **Web Intrusion Prevention**, select the target domain name, and select **Regular Protection** in the lower part of the page to view attack records.

   The ID of the protection rule that causes false positives

   ii. In the attack record list, find the false positive record and record the rule ID. You can search for the record by using the attack IP address.
5. In the left-side navigation pane, choose **System Management > Protection Rule Group**.
6. Create a custom rule group and remove the protection rule from the rule group.
   i. In the rule group list on the **Web Application Protection** tab, find the rule group that applies to the affected domain name.

   > **Note** To find the rule group, search for the affected domain name in the **Website** column.

   Website

   ii. Click **Copy** in the Action column. Assume that the **Medium rule group** causes the issue.

iii. On the **Copy Rule Group** page, modify **Rule Group Name**, turn on **Automatic Update**, and click **Save**. You can change the rule group name to medium rule group-remove false positive rule.

Copy rule group

After you copy the rule group, you can view it in the rule group list.

Rule group that you copy

iv. Find the rule group that you copy and click **Edit** in the Action column.

v. On the **Edit Rule Group** page, search for the rule that causes false positives by using the **rule ID**, select the rule, and then click **Remove Selected Rules**.

> **Note** Before you remove a protection rule from a custom rule group, make sure that you select the exact rule that blocks normal requests.

Edit rule group

vi. Click **Save**.

7. Apply the custom rule group to your website.

i. Find the rule group that you copy and click **Apply to Website** in the Action column.

ii. On the **Apply to Website** page, add the affected domain name to the **Websites Added to WAF** section and click **Save**.

Apply to website

After you apply the custom rule group, you can go to the **Website Protection** page and view the **RegEx Protection Engine** settings. The **Protection Rule Group** changes to the custom rule group that you apply. For more information, see Configure the RegEx Protection Engine.

Custom rule groups

When the website receives the same access requests again, WAF does not block the requests.

> **Note** If the requests are still blocked, make sure you identify the correct ID of the protection rule that causes false positives and remove this rule from the custom rule group.