Alibaba Cloud Web应用防火墙

Protection Settings

Issue: 20200704

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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	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.
	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.
	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.
	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]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	switch {active|stand}

Contents

1 Website protection

1.1 Best practices for website protection

This topic describes how to select protection modules and configure protection policies of Web Application Firewall (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

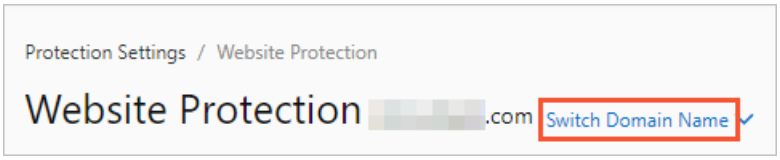
Your website configurations are added to WAF. For more information, see #unique_5.

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. Follow these steps 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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the 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 unfamiliar with website security and do not have any special requirements

- I am an O&M engineer. I want to ensure business stability and quickly troubleshoot issues
- 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 unfamiliar with website security and do not have any special requirements

You may have purchased WAF based on a need for classified protection or the intention to improve the security level of your enterprise. In either case, you need to add your website configurations to WAF so that you can 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:

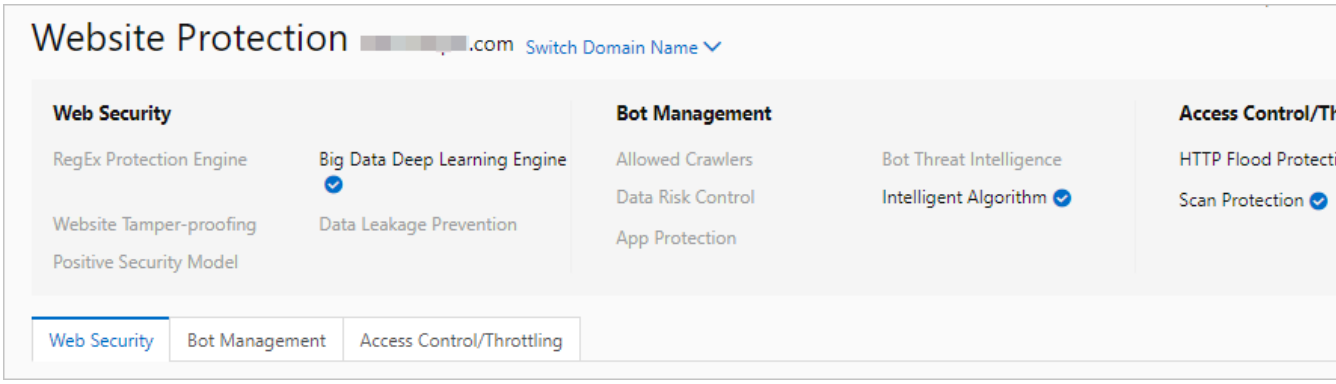
- #unique_6
- #unique_7

I am an O&M engineer. I want to ensure business stability and quickly troubleshoot issues

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.

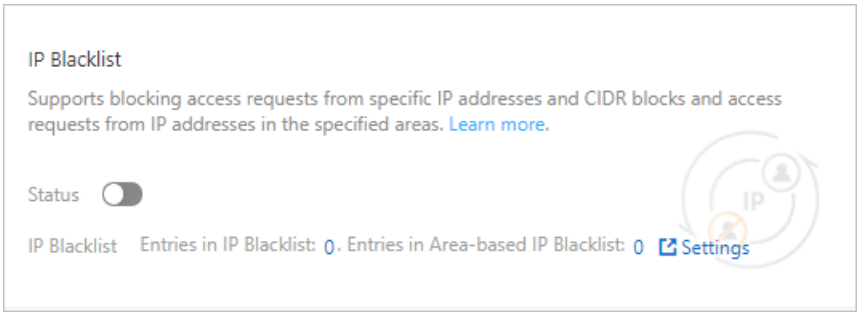
  Operations: On the **Website Protection** page, click **Website Whitelisting** in the upper-right corner. On the Website Whitelisting page, create a whitelist. For more information, see Configure the website whitelist.



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

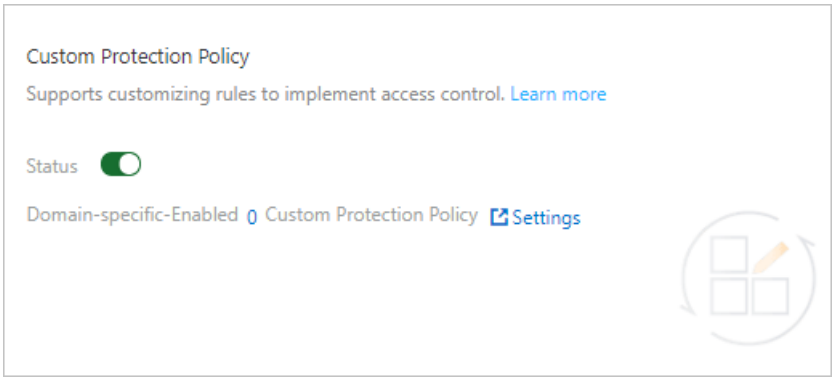
  - Configure the web intrusion prevention whitelist
  - Configure the access control and throttling whitelist
  - Configure the data security whitelist
  - Configure the bot management whitelist

- **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.



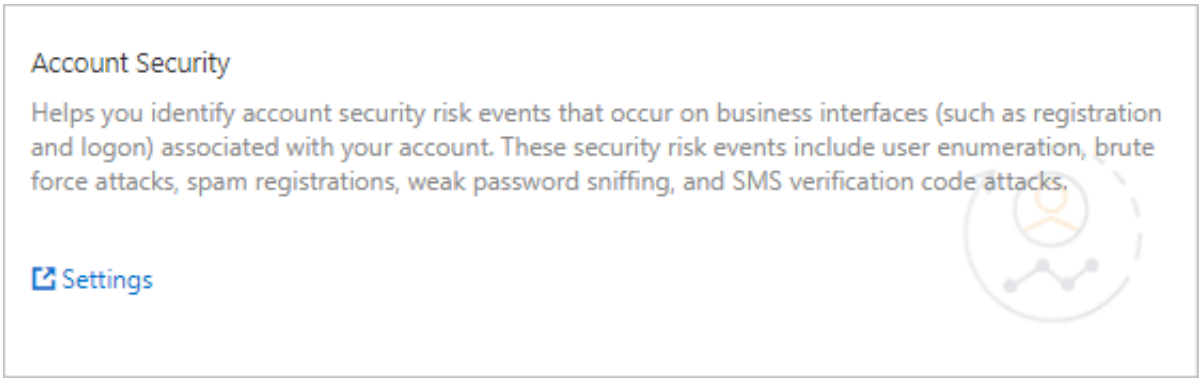
  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **IP Blacklist** card and configure the required parameters. For more information, see Configure the IP blacklist.

- **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.



  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 endpoints, such as the endpoints 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.

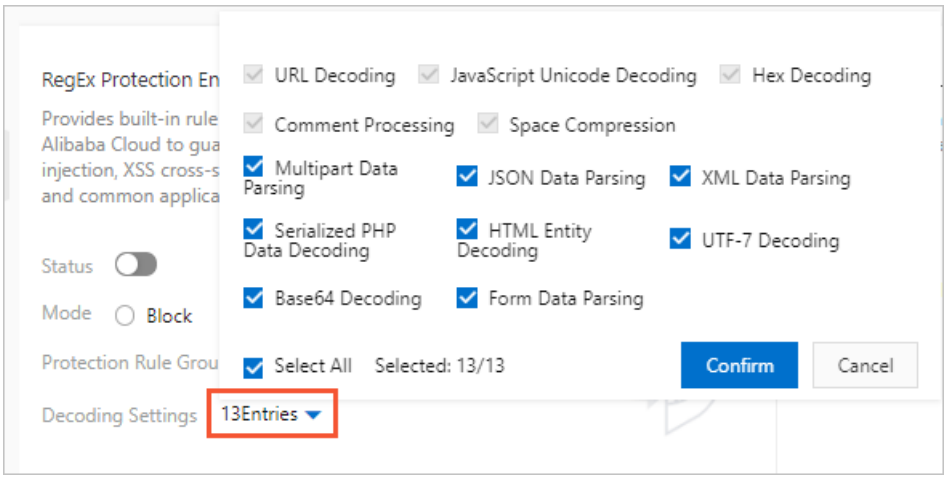


  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Account Security**. In the Account Security card, click **Settings** and configure the required parameters. For more information, see Configure account security.

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.

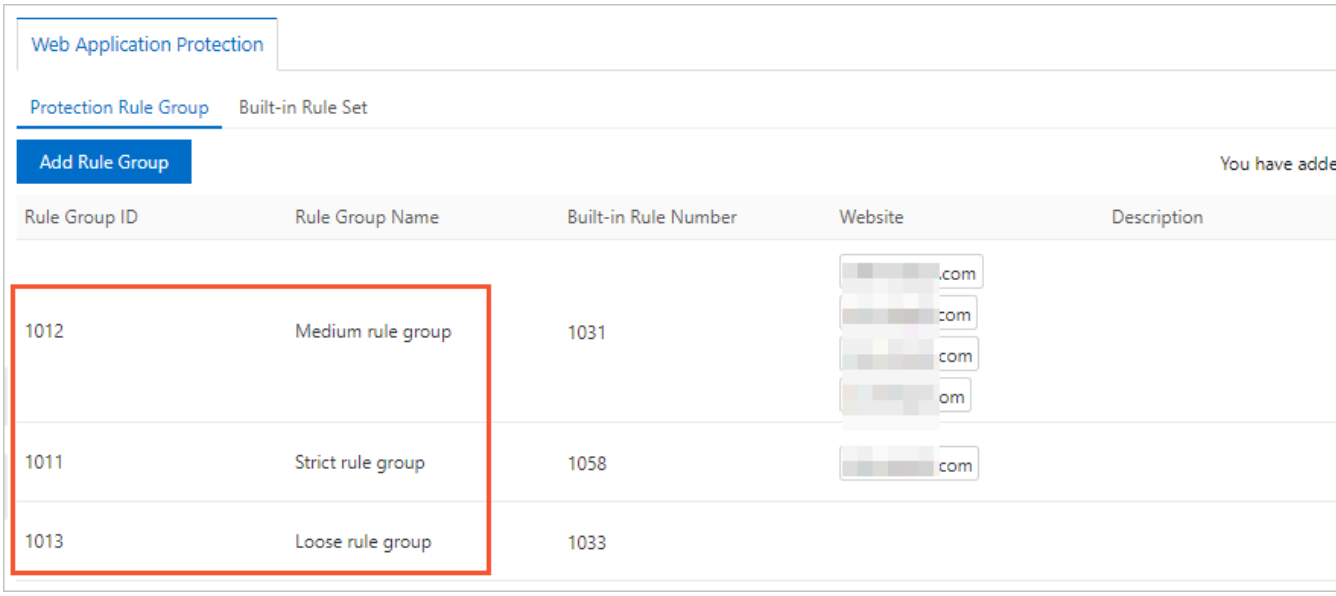


Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Web Intrusion Prevention** section, find **RegEx Protection Engine**. In the RegEx Protection Engine card, specify **Decoding Settings**. For more information, see Configure the RegEx Protection Engine.

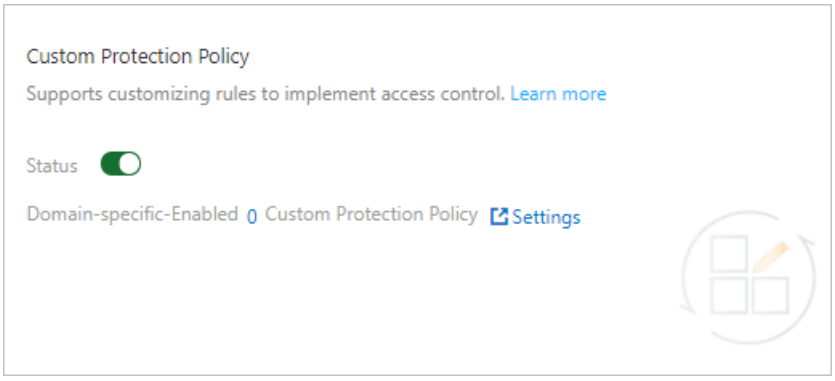
- **Protection Rule Group**: This feature allows you to select protection rules from a built-in protection rule set based on the form, framework, and middleware of your business system. You can use these rules to customize a rule group to prevent web attacks and apply the rule group to your website. 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.



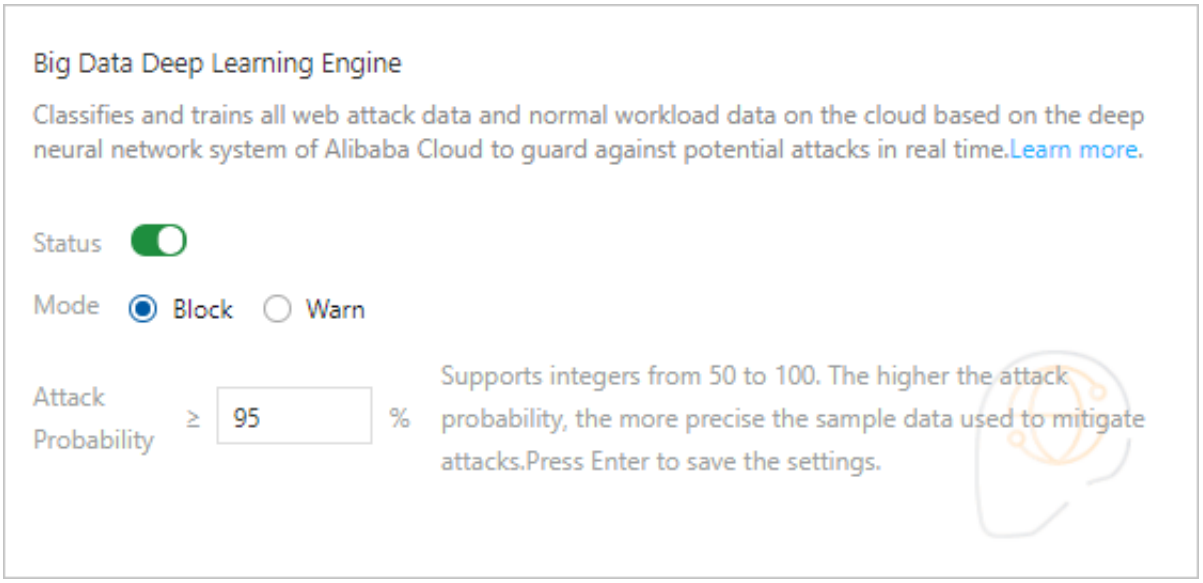
- **Custom Protection Policy**: This feature allows you to customize access control lists (ACLs) or throttling policies. For example, you can allow access to an API only from specific IP addresses or user agents and configure an upper limit for specific types of requests. You can also use this feature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



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

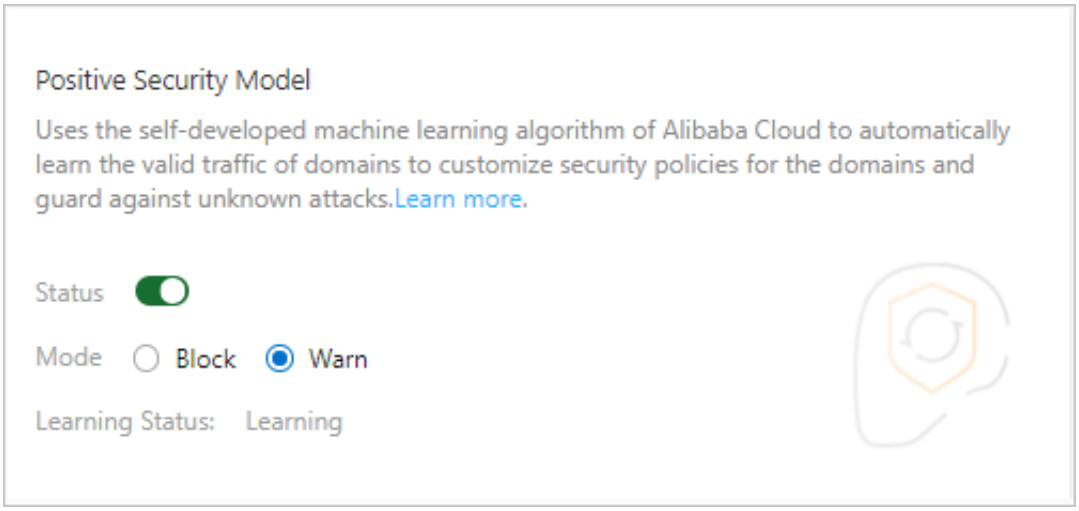
anomalies that are detected by the engine over a period of one to two weeks. If the engine works properly, switch to the **Block** mode.



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 the **Status** switch and set **Mode** to **Warn**. For more information, see Configure the big data deep learning engine.

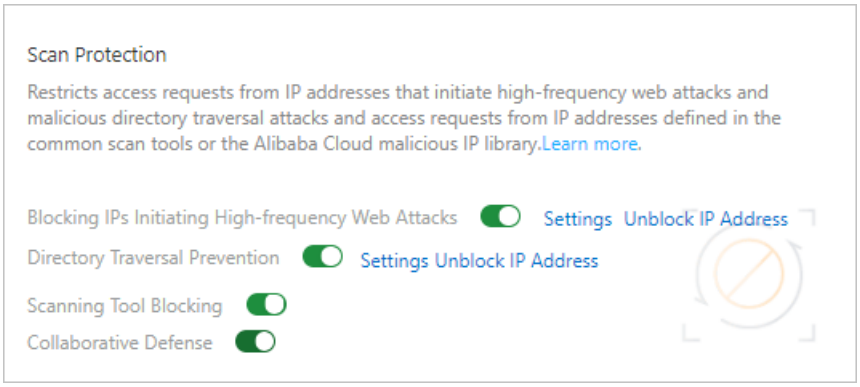
- **Positive Security Model** (**Warn** mode): The positive security model is built based on the learning of the traffic in the current domain. The model specifies the types and lengths of request parameters and whether the parameters are mandatory. After the model is built, if a request does not match 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.



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 the **Status** switch 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.

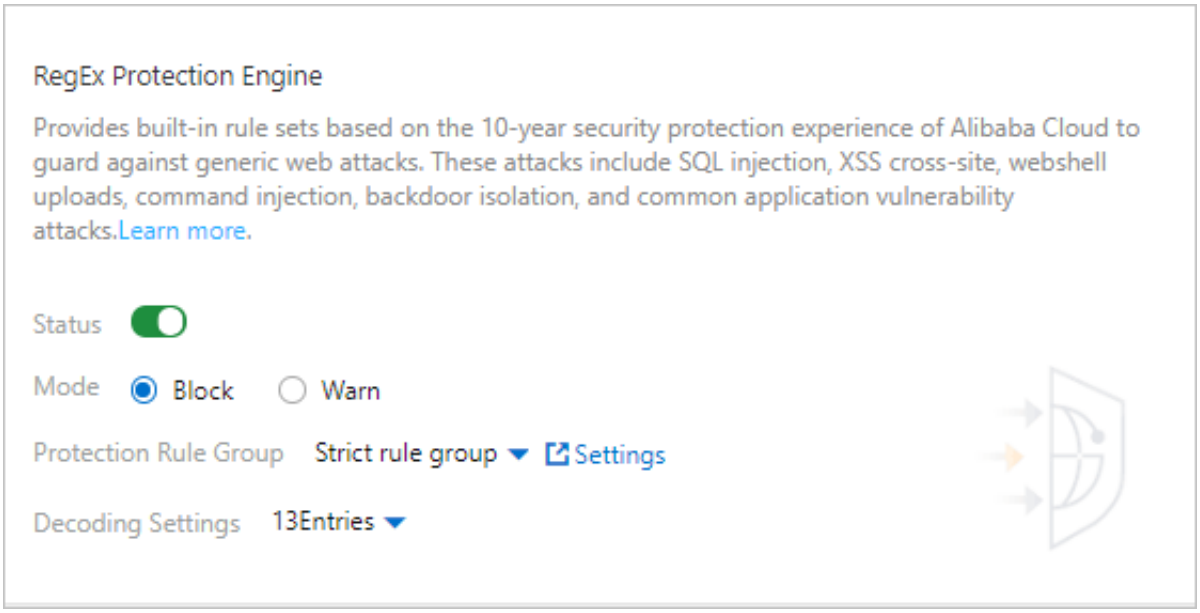


Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. In the **Scan Protection** card, enable all functions and specify appropriate thresholds. For more information, see Configure scan protection.

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:

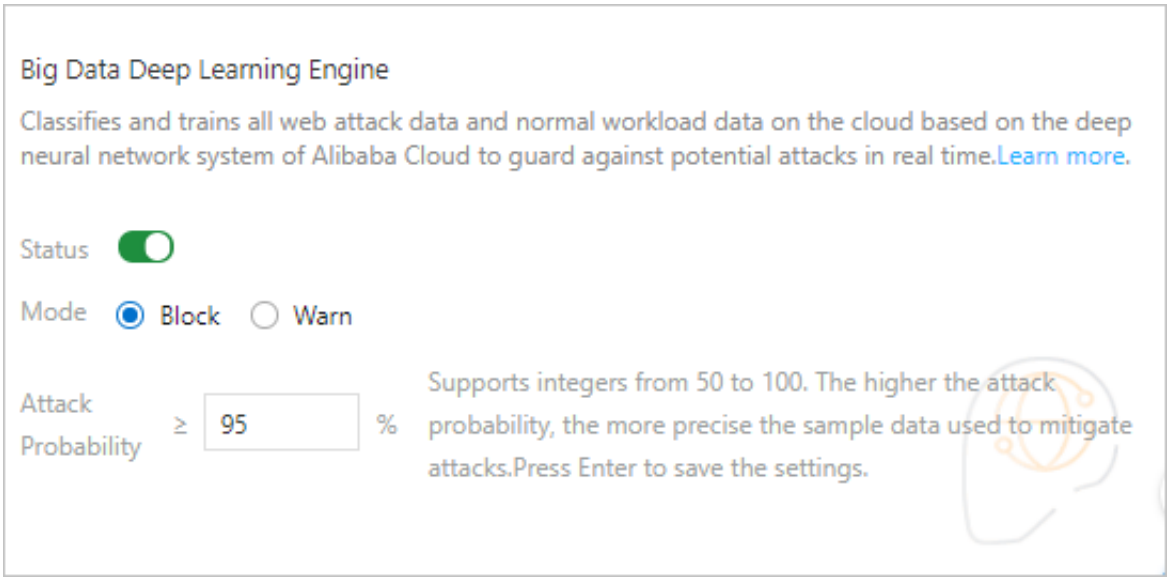
- **RegEx Protection Engine** (**Strict rule group**)



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,

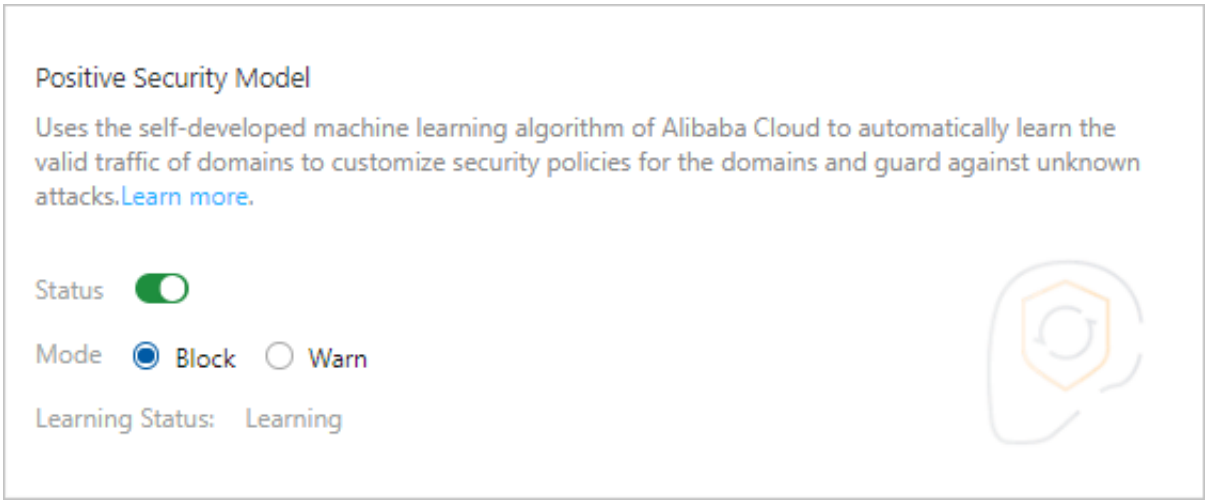
especially in terms of defense against deformed or unknown attacks. To achieve the strongest protection, we recommend that you enable the **Block** mode.



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 the **Status** switch and set **Mode** to **Block**. For more information, see Configure the big data deep learning engine.

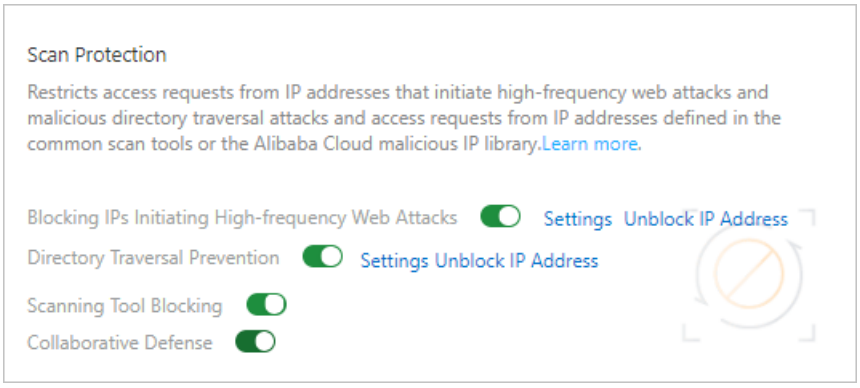
- **Positive Security Model** (**Block** mode): The positive security model is built based on the learning of the traffic in the current domain. The model specifies the types and lengths of request parameters and whether the parameters are mandatory. After the model is built, if a request does not match the characteristics described in the model, an alert

is generated. To achieve the strongest protection, we recommend that you enable the **Block** mode.



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 the **Status** switch and set **Mode** to **Block**. For more information, see Configure the positive security model.

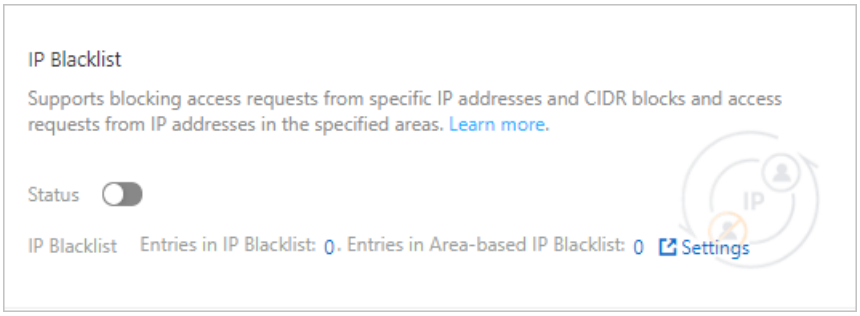
- **Scan Protection** (**Blocking IPs Initiating High-frequency Web Attacks**, **Directory Traversal Prevention**, **Scanning Tool Blocking**, and **Collaborative Defense**): This feature helps reduce the threats generated by your scanner from multiple dimensions, such as intelligence, scanner features,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



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.



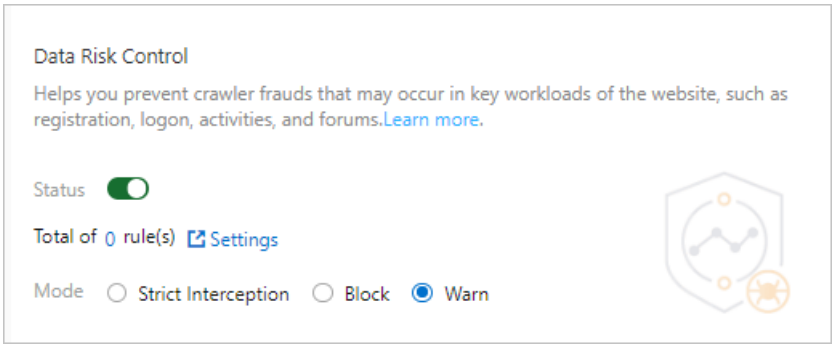
Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **IP Blacklist** card and configure the required parameters. For more information, see Configure the IP blacklist.

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:

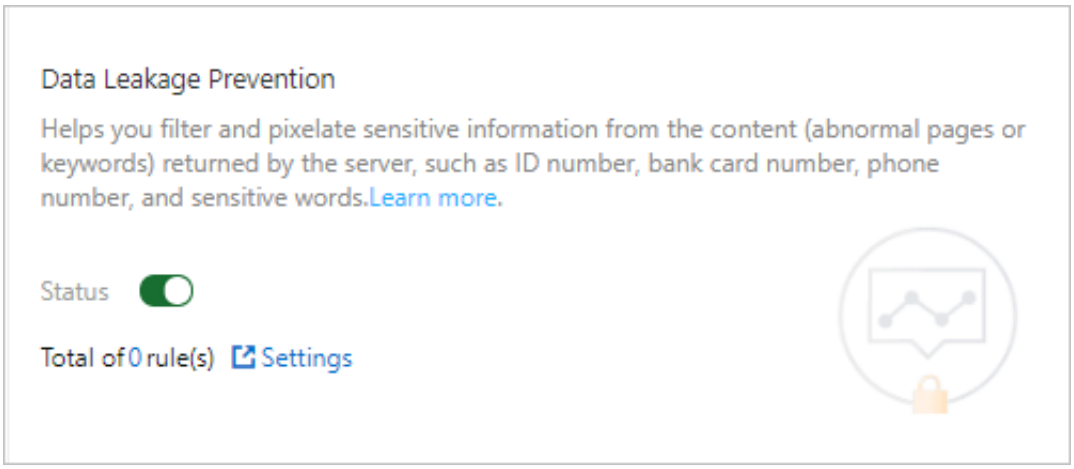
- **Data Risk Control**: This feature is best suited to defend against bot traffic that is generated by scripts or automated tools and destined for specific APIs for logon, registration, and order placing.

> **Note:**
> Data risk control depends on JavaScript injection 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 the DingTalk.



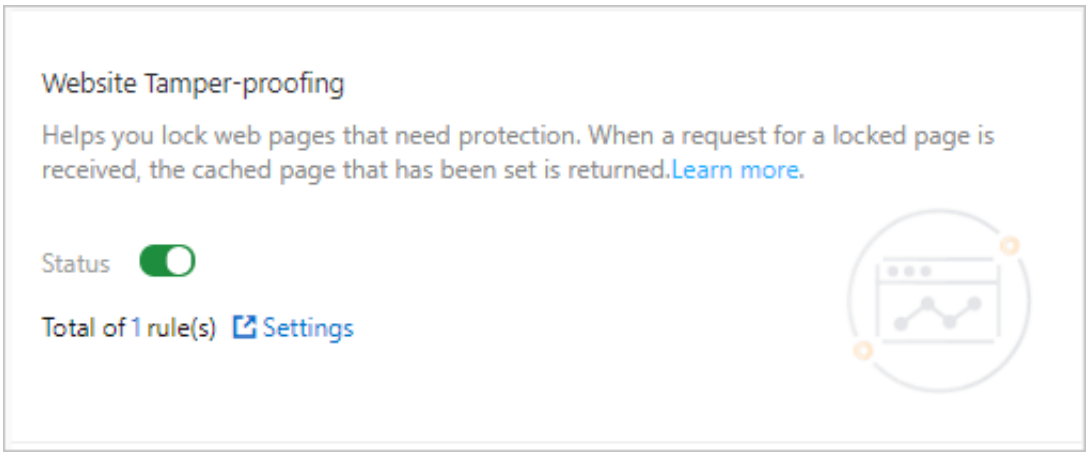
Operations: On the **Website Protection** page, click the **Bot Management** tab. In the **Data Risk Control** card, configure the required parameters. For more information, see Configure data risk control.

- **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.



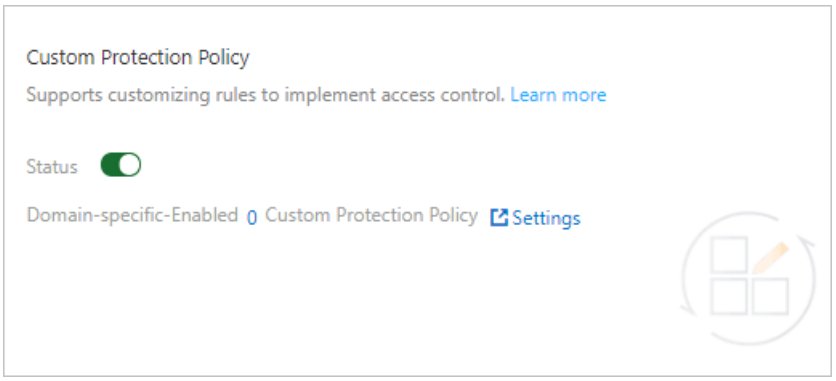
Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Data Leakage Prevention**. In the Data Leakage Prevention card, configure the required parameters. For more information, see Configure data leakage prevention.

- **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.



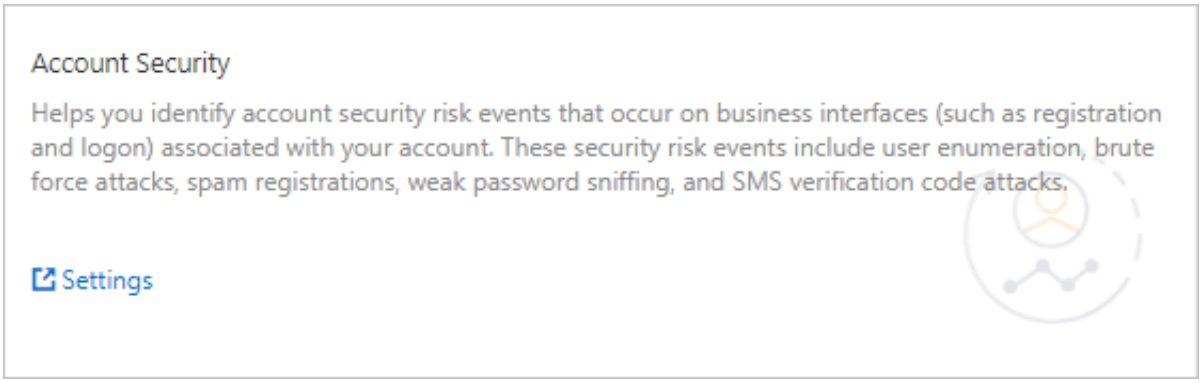
Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Website Tamper-proofing**. In the Website Tamper-proofing card, configure the required parameters. For more information, see Configure tamper-proofing.

- **Custom Protection Policy**: You can one-click enable JavaScript verification for frequently crawled static web pages 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.



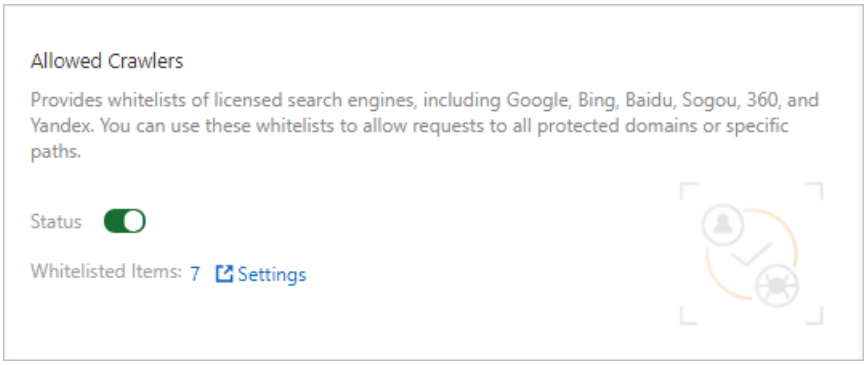
  Operations: On the **Website Protection** page, click the **Access Control/Throttling** tab. Find the **Custom Protection Policy** card and configure the required parameters. For more information, see Create a custom protection policy.

- **Account Security**: This feature allows you to monitor user authentication-related endpoints, such as the endpoints 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.



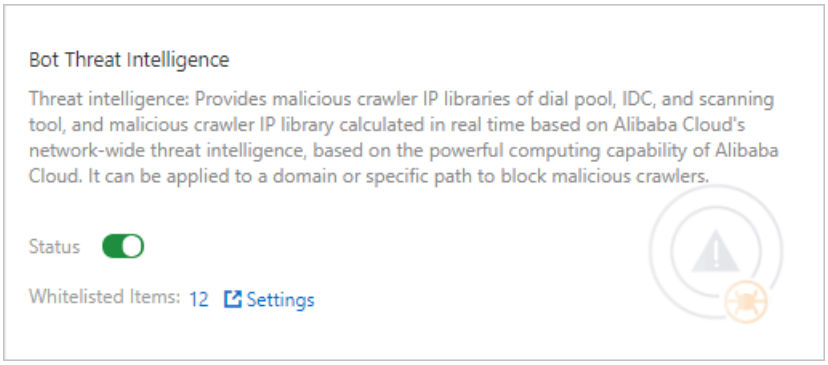
  Operations: On the **Website Protection** page, click the **Web Security** tab. In the **Data Security** section, find **Account Security**. In the Account Security card, click **Settings** and configure the required parameters. For more information, see Configure account security.

- **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 domains.



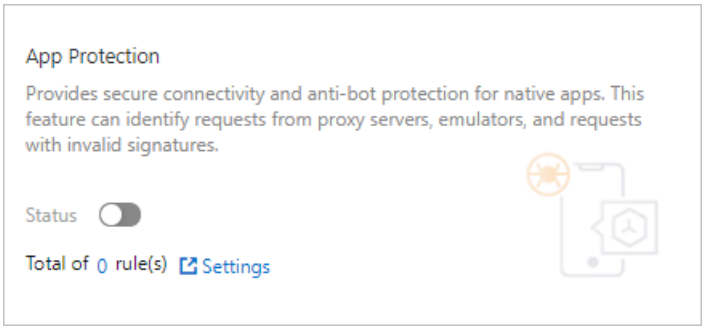
  Operations: On the **Website Protection** page, click the **Bot Management** tab. In the **Allowed Crawlers** card, configure the required parameters. For more information, see Set a threat intelligence rule to allow requests from specific crawlers.

- **Bot Threat Intelligence**: This feature provides information about suspicious IP addresses used by harassing phone calls, Internet data centers (IDCs), and malicious scanners. This feature also maintains an IP address library of malicious crawlers to prevent crawlers from accessing your website or specific directories.



  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 applications and can identify proxies, emulators, and requests with invalid signatures.



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.

1.2 Configure the website whitelist

After you add a website to WAF, its website protection policies filter all the access requests by default. The website whitelist allows access requests that match specified conditions. These access requests are directly returned to the origin site instead of being filtered by the WAF website protection policies.

Notice:
This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before this date, you cannot use the website whitelist.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

Background information

The WAF website protection policies include modules such as web intrusion prevention, access control and throttling, data security, advanced protection, and bot management. Access requests that match specified conditions in the whitelist skip all detection modules. The website whitelist is used to allow trusted access requests, such as access requests from trusted vulnerability scan tools and trusted authenticated third-party system endpoints.

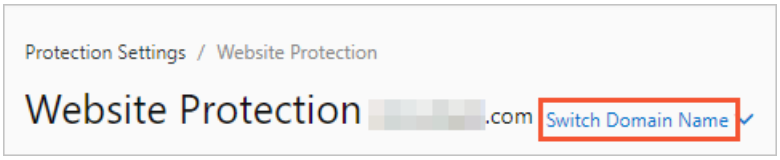
You can also create a whitelist for each specified detection module. Access requests that match specified conditions only skip the corresponding detection module. For more information, see:

- Configure the web intrusion prevention whitelist
- Configure the access control and throttling whitelist
- Configure the data security whitelist
- Configure the bot management whitelist

Note:
We recommend that you create a whitelist for a specified detection module as needed. A whitelist with more precise rules provides better security protection. A detection module whitelist provides better security protection than the 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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.

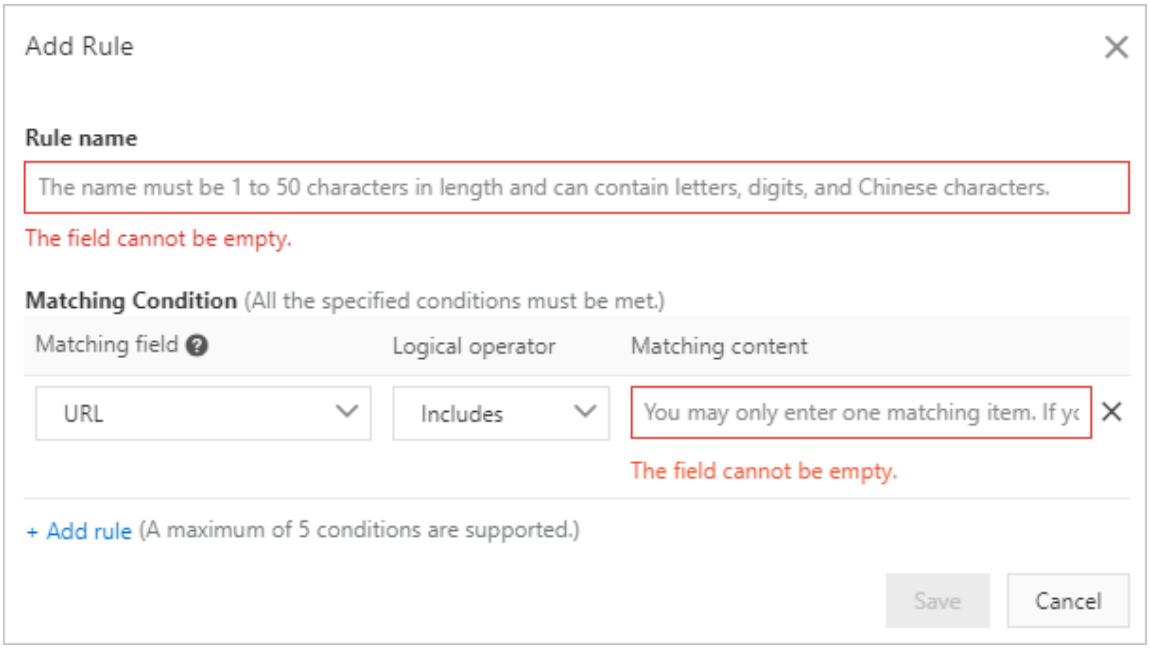


5. Click **Website Whitelisting** in the upper-right corner.

6. Create a website whitelist.

a) On the **Website Whitelisting** page, click **Create Rule**.

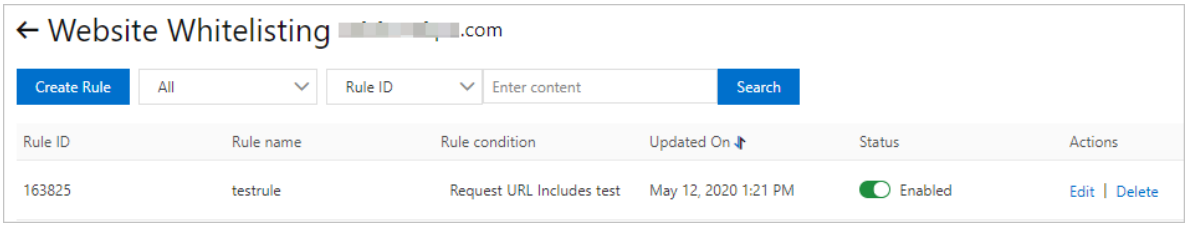
b) In the **Add Rule** dialogue box that appears, set the following parameters.



Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify the match conditions of the whitelist rule. Click **Add rule** to add more conditions. You can specify a maximum of five conditions. If you have set multiple conditions, the rule is matched only after all of them are met. For more information about match conditions, see Fields of match conditions.

c) Click **Save**.

After you create rules for the website whitelist, they are automatically enabled. You can view newly created rules in the rule list and disable, edit, or delete rules as needed.



1.3 Web intrusion prevention

1.3.1 Configure the web intrusion prevention whitelist

For websites added to Web Application Firewall (WAF), web intrusion prevention responds quickly to common web attacks and zero-day vulnerabilities to secure your websites. Web intrusion prevention supports the RegEx Protection engine and the big data deep learning engine. You can configure the web intrusion prevention whitelist. Requests that match specific conditions in the whitelist can skip specified detection modules.

> **Notice:**
> This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before this date, you cannot to use the web intrusion prevention whitelist.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

Background information

The web intrusion prevention whitelist is generally used to allow access requests that are mistakenly blocked. We recommend that you set the match conditions as precisely as possible to ensure that only the specific access requests are allowed.

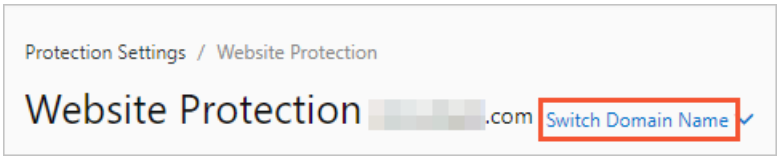
For more information about supported detection modules of web intrusion prevention, see:

- Configure the RegEx Protection Engine
- Configure the big data deep learning engine

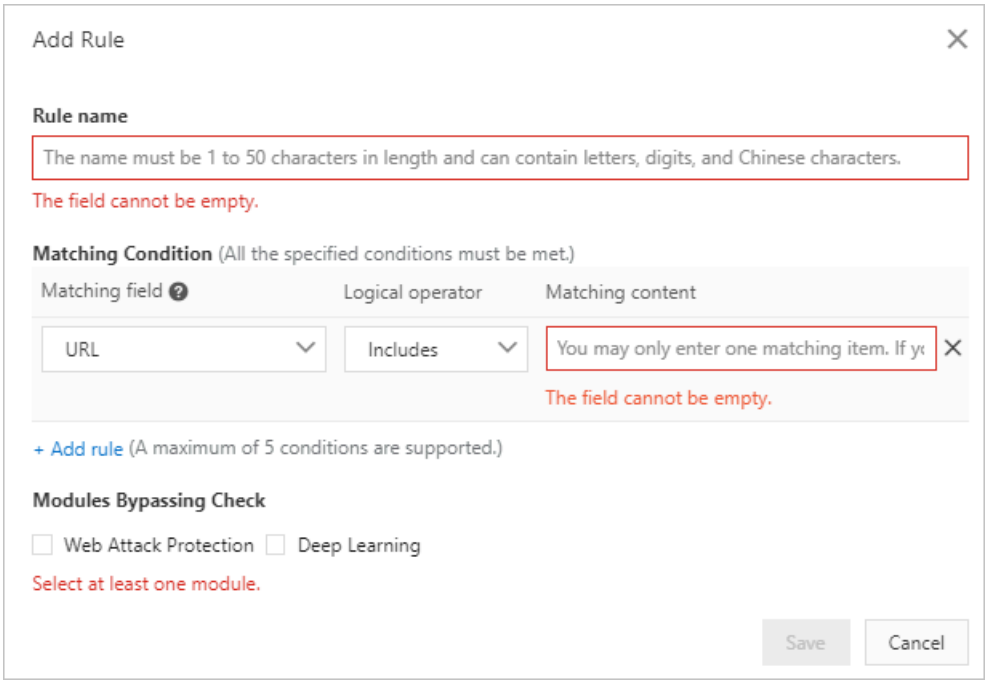
Procedure

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .

4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Web Security** tab, find the section, and then click .
6. Create the web intrusion prevention whitelist.
   a) On the **Web Intrusion Prevention - Whitelisting** page, click **Create Rule**.
   b) In the **Add Rule** dialogue box, set the following parameters.

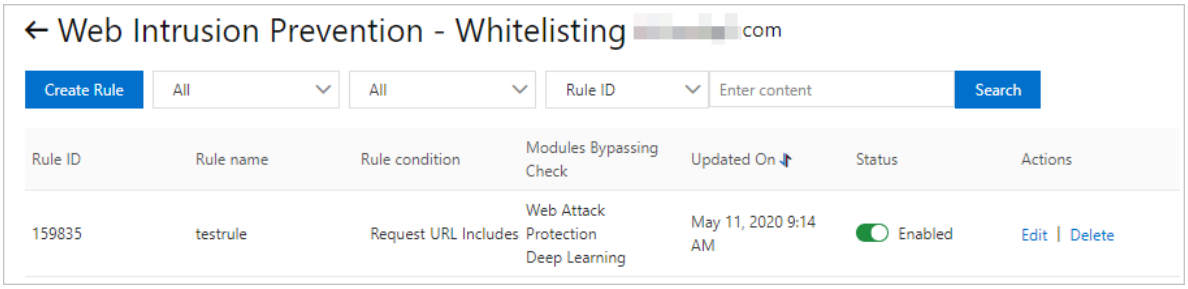


Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify the match conditions of the whitelist rule. Click **Add rule** to add more conditions. You can specify a maximum of five conditions. If you have set multiple conditions, the rule is matched only after all of them are met. For more information about match conditions, see Fields of match conditions.

Parameter	Description
Modules Bypassing Check	Specify the detection modules to be ignored after the match conditions of the rule are matched. Supported modules include: • **Web Attack Protection** • **Deep Learning**

c) Click **Save**.

After you create rules for the web intrusion prevention whitelist, they are enabled automatically. You can view newly created rules in the rule list and disable, edit, or delete rules as needed.



1.3.2 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, and common web attacks. You can adjust the protection policies of the RegEx Protection Engine as needed.

Notice:
This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before this date, see #unique_30.

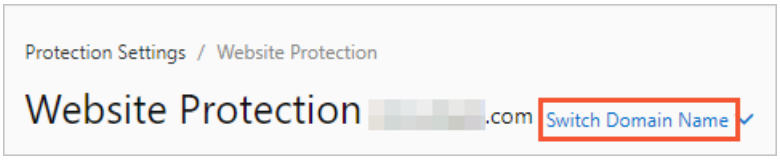
Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

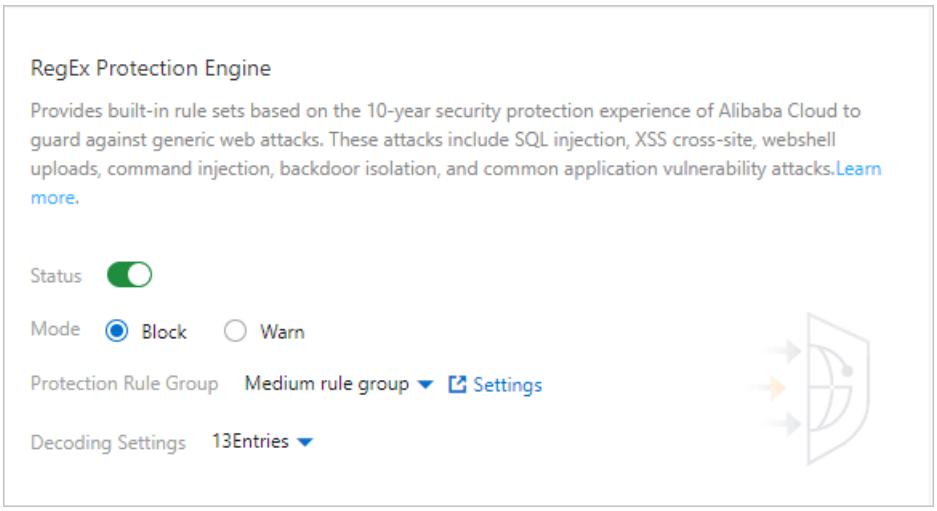
Procedure

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.

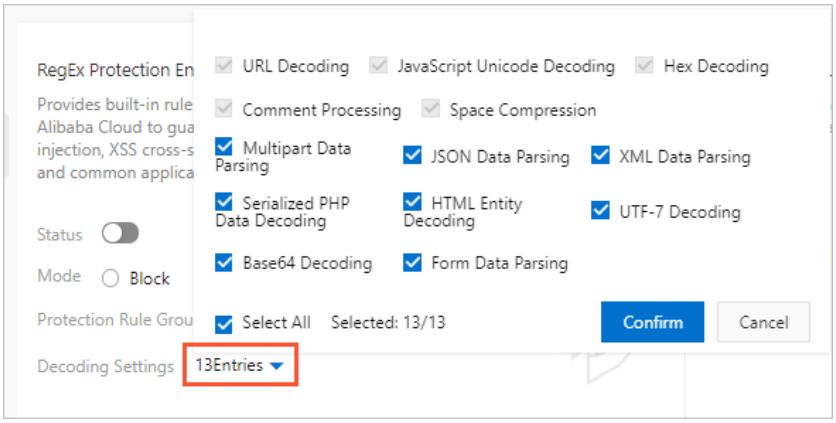


5. Click the **Web Security** tab, and find **RegEx Protection Engine** in the **Web Intrusion Prevention** module to set the following parameters.



Parameter	Description
Status	Enable or disable the RegEx Protection Engine.
Mode	Specify the action that is taken on attack requests when they are detected. Supported modes include: • **Block**: This mode blocks the attack requests. • **Warn** : This mode only triggers alerts without blocking the requests.

Parameter	Description
Protection Rule Group	Specify the detection rule group to be applied. Built-in rule groups and custom rule groups are supported. Built-in rule groups include the medium rule group, strict rule group, and loose rule group. • **Medium rule group**: This rule group detects common web application attacks and default applications in a standard way. • **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 #unique_31.

Parameter	Description
Decoding Settings	Specify 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 contents of all formats by default. If the RegEx Protection Engine blocks normal requests that contain contents of specified formats, you can cancel decoding the contents of the corresponding formats to reduce the false positive rate. Procedure **a.** Unfold the configuration menu.  **b.** Select or clear the target format to be decoded. • The following formats must be decoded: URL decoding , JavaScript unicode decoding, hexadecimal decoding, comment processing, and space compression. • The following formats are optional: 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. **c.** Click **Comfirm**.

1.3.3 Configure the big data deep learning engine

After you add a website to WAF, you can enable the big data deep learning engine for the 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. In this way, potential attacks can be blocked in real time. You can adjust the protection policies of the big data deep learning engine as required.

Notice:
This topic describes the big data deep learning engine in the WAF console released in January 2020. If your WAF instance was created before this date, see #unique_32.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.
- If the billing method of the instance is subscription, the edition of the instance must be Business or Enterprise.

Background information

Web attack methods keep evolving as the Internet develops rapidly. Traditional single-method protection can no longer meet the security requirements of complex Internet services. Collaborative protection powered by multiple detection engines offers stronger protection.

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 data is forwarded to the RegEx Protection Engine. Then, the traffic data is forwarded to the big data deep learning engine. The two engines complement each other.

Scenarios

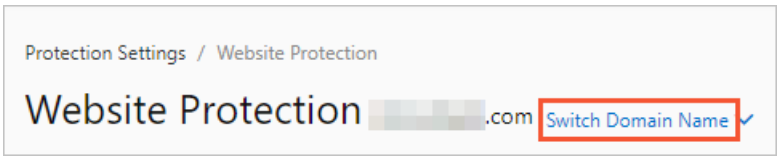
The big data deep learning engine mainly targets web attack requests with weak characteristics rather than HTTP flood attacks. If you have high requirements on web attack prevention, we recommend that you enable the big data deep learning engine.

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

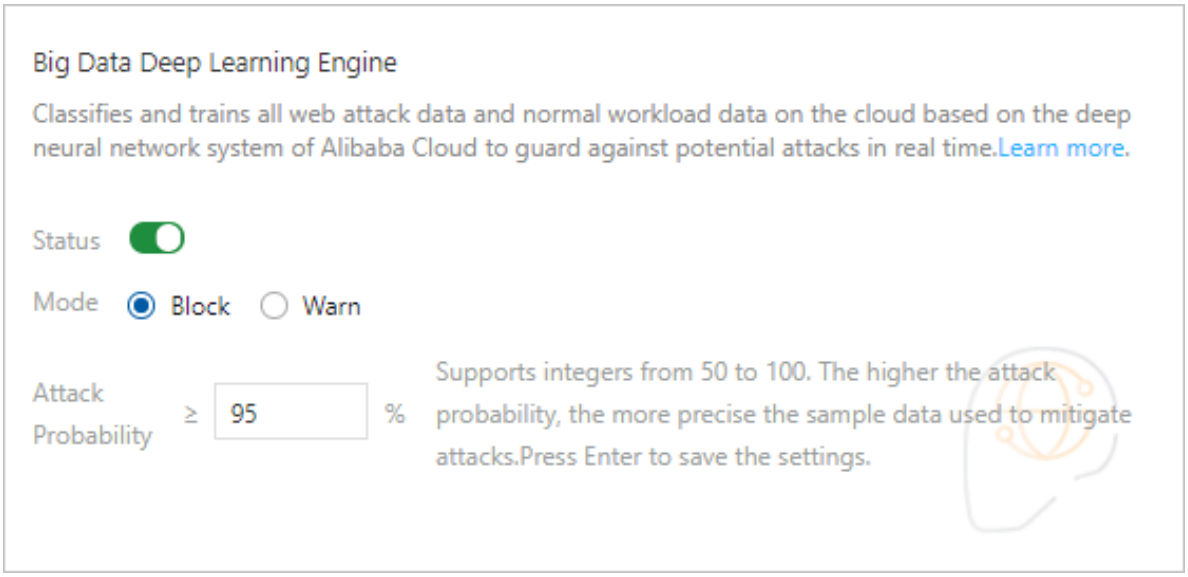
mode. In this case, you can enable the big data deep learning engine to identify and block requests with weak attack characteristics that cannot be identified by strict rules of the RegEx Protection Engine.

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. On the **Web Security** tab, set the following parameters in **Big Data Deep Learning Engine** of the **Web Intrusion Prevention** section.



Parameter	Description
Status	Enables or disables the big data deep learning engine.
Mode	Specifies the action that is taken on attack requests when they are detected. Valid values: • **Block**: Block the attack requests. • **Warn**: Trigger only alerts without blocking the attack requests.

Parameter	Description
Attack Probability	Sets the 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, this 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, this engine may also block some normal requests.

1.4 Access control and throttling

1.4.1 Configure the access control and throttling whitelist

The access control and throttling whitelist provides access control and throttling policies for websites that are added to Web Application Firewall (WAF) based on the application layer. It also ensures the accessibility of the website. The access control and throttling whitelist supports HTTP flood protection, IP blacklist, scan protection, and custom protection polices. You can configure the access control and throttling whitelist. Requests that match specific conditions in the whitelist can skip specified detection modules.

Notice:
This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before this date, you cannot use the access control and the throttling whitelist.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

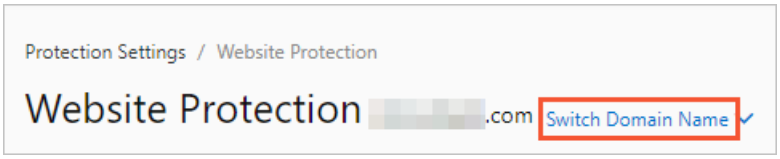
Background information

For more information about detection modules supported by access control and throttling, see the following topics:

- Configure HTTP flood protection
- Configure the IP blacklist
- Configure scan protection
- 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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.

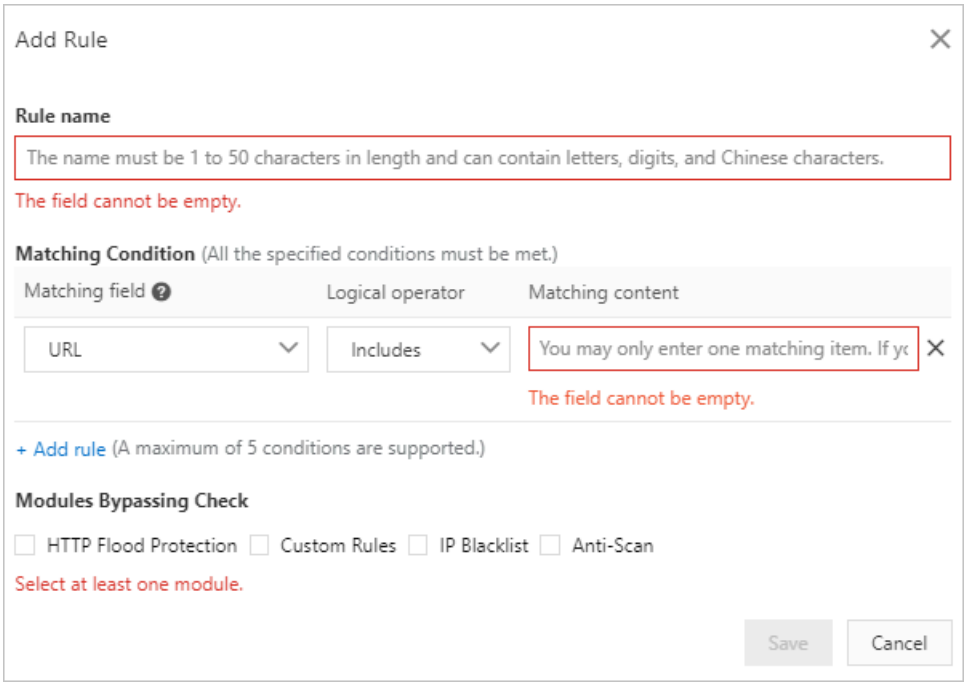


5. Click the **Access Control/Throttling** tab, find the section, and then click .

6. Create the access control and throttling whitelist.

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

   b) In the **Add Rule** dialogue box, set the following parameters.

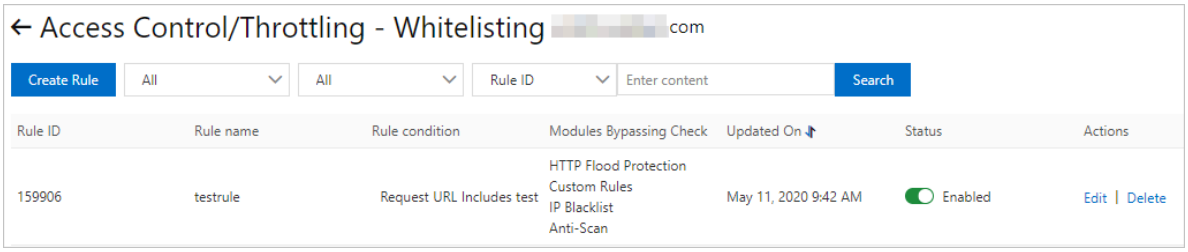


Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify the conditions that a whitelist request must match. Click **Add rule** to add more conditions. You can specify a maximum of five conditions. If you have set multiple conditions, the rule is matched only after all of them are met. For more information about match conditions, see Fields of match conditions.

Parameter	Description
Modules Bypassing Check	Specify the detection modules to be ignored after the match conditions of the rule have been matched. Detection modules include: • **HTTP Flood Protection** • **Custom Rules** • **IP Blacklist** • **Anti-Scan**

c) Click **Save**.

After you create rules for the access control and throttling whitelist, they are enabled automatically. You can view newly created rules in the rule list and disable, edit, or delete rules as needed.



1.4.2 Configure HTTP flood protection

After you add a website to Web Application Firewall (WAF), HTTP flood protection targeting web pages is enabled by default. HTTP flood protection terminates connections to block HTTP flood attacks. You can adjust the protection policies of HTTP flood protection as needed.

Notice:

This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before January 2020, see #unique_35.

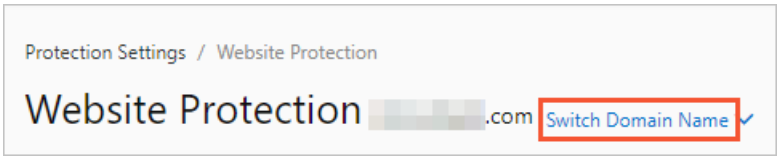
Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

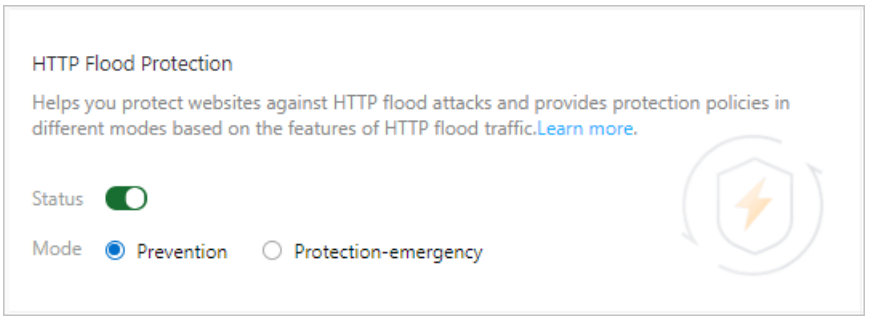
Precedure

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Access Control/Throttling** tab, and find **HTTP Flood Protection** in the **Access Control/Throttling** module to set the following parameters.



Parameter	Description
Status	Enable or disable HTTP flood protection.
Mode	Specify the protection mode. Supported modes: • **Prevention**: This mode only blocks 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 blocks a large number of requests and maintains a high false positive rate. You can apply this mode if the Protection mode fails to block HTTP flood attacks or if the website responds slowly and indicators such as traffic, CPU, and memory are abnormal. **Note:** You can only use the Protection-emergency mode to protect 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 or Native App scenarios. For more information, see Create a custom protection policy.

Related operations

- If the Protection-emergency mode causes a high false negative rate, we recommend that you check whether the attacks come from WAF back-to-origin IP addresses. If attacks are directly launched on the origin server, you can change the settings to only allow requests from WAF back-to-origin IP addresses. For more information, see #unique_36.
- If you need to reinforce protection and maintain a low false positive rate, you can create multiple custom protection policies. For more information, see Create a custom protection policy.

1.4.3 Configure the IP blacklist

After you add a website to WAF, you can enable the IP blacklist feature. The IP blacklist blocks the access requests from the specified IP addresses and CIDR blocks. It also blocks the access requests from IP addresses in specified regions. You can specify the IP addresses, CIDR blocks, and regions as needed.

Notice:

This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before January 2020, see #unique_37 and #unique_38.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.
- If the billing method of the instance is subscription, the edition of the instance must be Business or Enterprise.

Background information

The IP blacklist includes the common IP blacklist and the area-based IP blacklist.

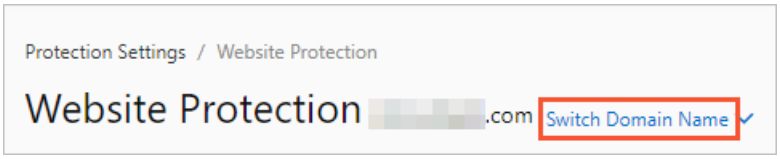
- The common IP blacklist: Blocks access requests from specified IP addresses and CIDR blocks.
- The area-based IP blacklist: Blocks the access requests of which the source IP addresses are from specified regions. You can specify 247 countries and regions as blocked

regions, including Hong Kong (China), Macau (China), Taiwan (China), and provinces in mainland China.

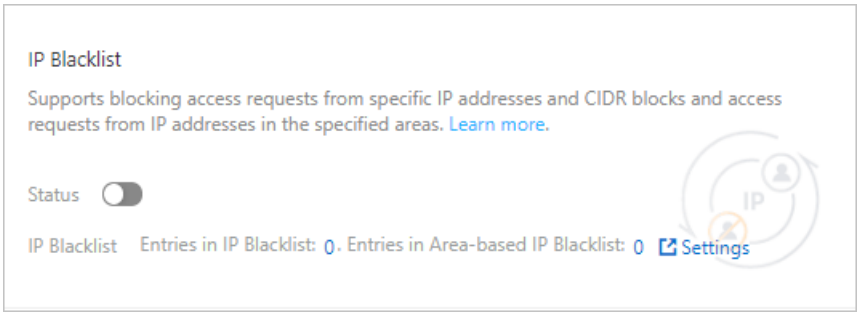
For more information about the source regions of IP addresses, see Taobao IP address library.

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.

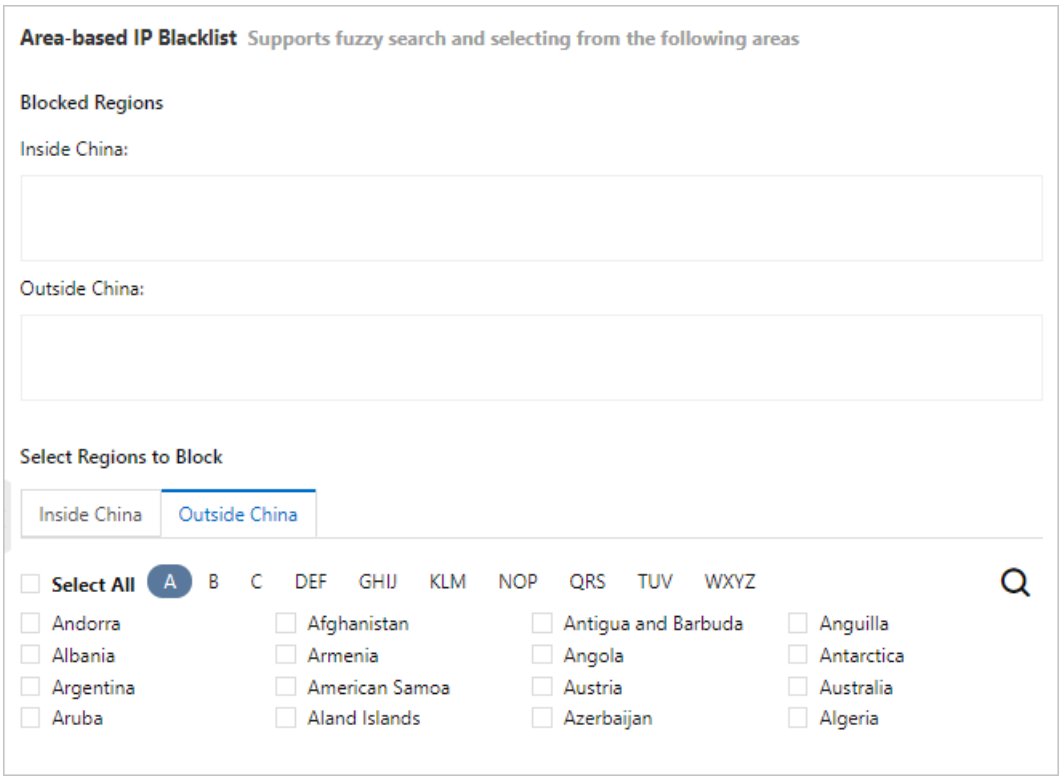


5. Click the **Access control/Throttling** tab and find the **IP Blacklist** section in the **Access Control/Throttling** module. Turn on the **Status** switch and click **Settings**.



6. In the **IP Blacklist** section, configure the **IP Blacklist** and the **Area-based IP Blacklist**.
   - **IP Blacklist**: Enter the IP addresses that you want to block and click **Save** at the bottom. Separate multiple IP addresses with a comma (,). You can add a maximum of 200 IP addresses.

   - **Area-based IP Blacklist**: Select the regions that you want to block from **Inside China** or **Outside China** and click **Save** at the bottom.



   After you turn on the status switch, all the access requests from the IP addresses in the blacklist are automatically blocked.

See also

- If you want more precise access control based on the IP blacklist, we recommend that you create custom protection policies. For more information, see Create a custom protection policy.
- If you want to limit the access traffic of a specified IP address, we recommend that you configure the access control or throttling whitelist. For more information, see Configure the access control and throttling whitelist.

1.4.4 Configure scan protection

After you add a website to Web Application Firewall (WAF), you can enable the scan protection feature for your website. The scan protection feature automatically blocks access requests that have specific characteristics. For example, if the source IP address of the 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 library that records malicious IP addresses. You can customize the policies of scan protection as needed.

Notice:
This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before January 2020, see #unique_39, #unique_40 and #unique_41.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.
- To customize protection policies of high-frequency web attack blocking and directory traversal protection, the billing method of your WAF instance must be either a monthly or annual subscription. The WAF instance must use the Enterprise edition, Ultimate edition or Exclusive edition. The Advanced edition only supports scan protection with the default protection policy.

Background information

The scan protection feature provides high-frequency web attack blocking, directory traversal protection, scan tool blocking, and collaborative protection.

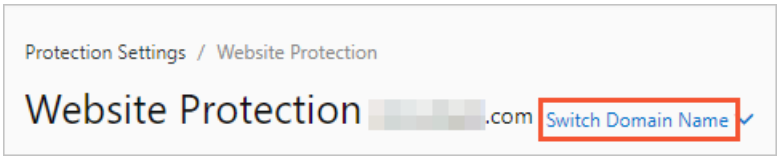
- High-frequency web attack blocking: Automatically blocks IP addresses that initiate multiple web attacks in a short period of time. If the number of web attacks initiated by a client IP address exceeds a certain number, the access requests from this IP address are blocked for a certain time period. You can customize the protection policies of high-frequency web attack blocking. You can manually unblock a blocked IP address.
- Directory traversal protection: Automatically blocks client IP addresses that initiate multiple directory traversal attacks in a short period of time. If the total number of requests initiated by a client IP address exceeds a certain number and the proportion of the 404 HTTP status code exceeds a certain proportion, the access requests from this IP

address are blocked for a certain time period. You can customize the protection policies of directory traversal protection. You can manually unblock a blocked IP address.

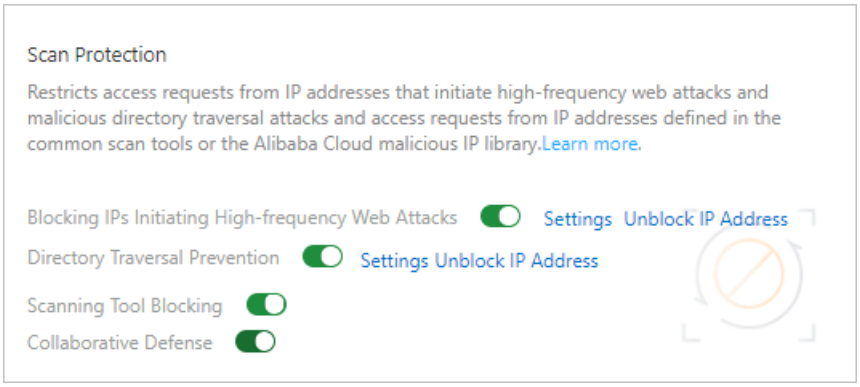
- Scan tool blocking: Automatically blocks access requests from IP addresses of common scan tools. Blocked scan tools include: Sqlmap, AWVS, Nessus, Appscan, Webinspect, Netsparker, and NiktoRsas.
- Collaborative protection: Automatically blocks access requests from IP addresses in the Alibaba Cloud library that records malicious IP addresses.

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Access Control/Throttling** tab, and find **Scan Protection** in the **Access Control/Throttling** module to complete the following settings.



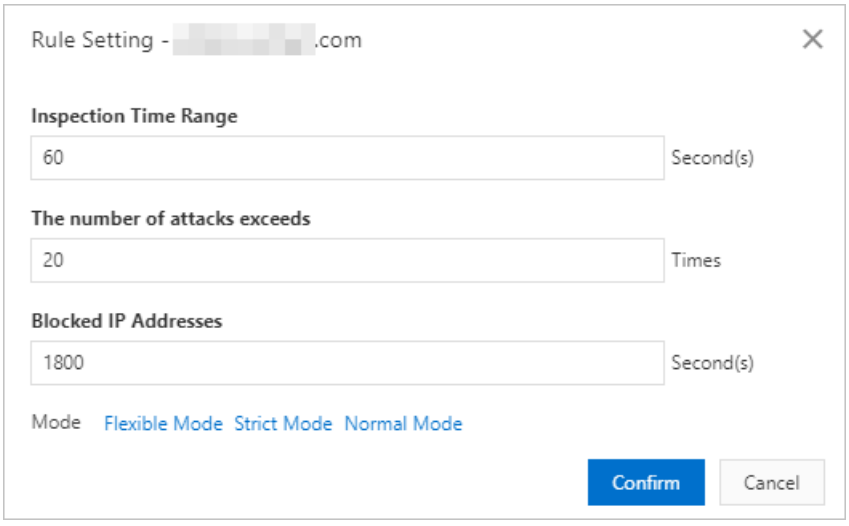
- **Blocking IPs Initiating High-frequency Web Attacks**: Enable or disable high-frequency web attack blocking.

  To configure the protection policies of high-frequency web attack blocking, follow these steps:

  a. Enable high-frequency web attack blocking.

  b. Click **Settings**.

  c. In the **Rule Setting** dialog box, set the following parameters: **Inspection Time Range** (seconds), **The number of attacks exceeds** (times), **Blocked IP Addresses** (seconds).



  Rule definition: If the number of web attacks initiated by a client IP address in the specified **Inspection Time Range** exceeds the specified number (**The number**

of attacks exceeds), the access requests from this IP address are blocked for the specified time period (**Blocked IP Addresses**).

Note:

We recommend that you use **Mode** and choose a built-in configuration mode from **Flexible Mode**, **Strict Mode**, and **Normal Mode**. You can adjust the parameters as needed.

d. Click **Confirm**.

Unblock a blocked IP address: Click **Unblock IP Address** to unblock the target IP address.

- **Directory traversal protection**: Enable or disable directory traversal protection.

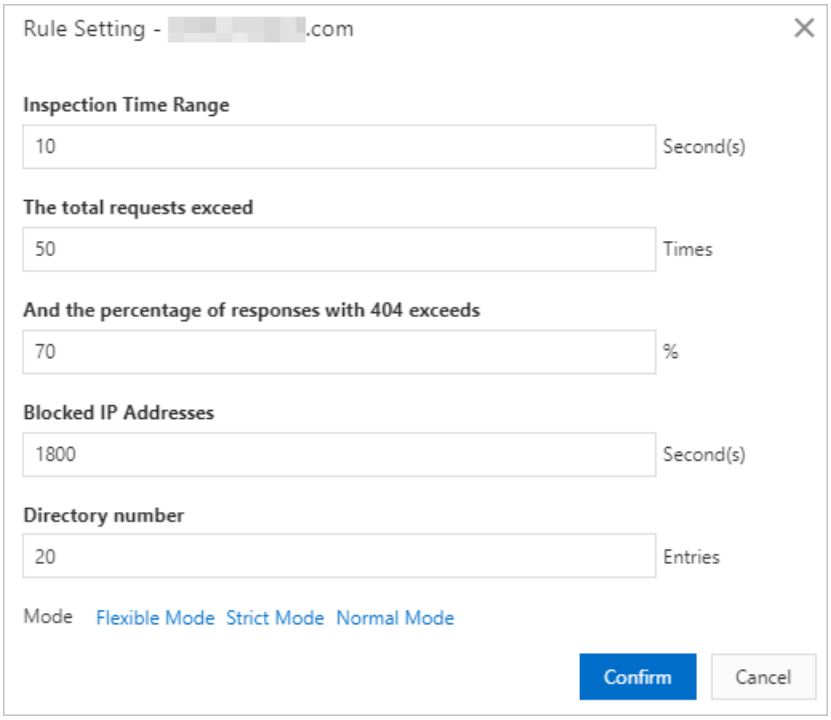
  To configure the protection policies of directory traversal protection, follow these steps:

  a. Enable directory traversal protection.

  b. Click **Settings**.

  c. In the **Rule Setting**dialog box, set the following parameters: **Inspection Time Range** (seconds), **The total requests exceed** (times), **And the percentage of**

responses with 404 exceeds (%), **Blocked IP Addresses** (seconds), and**Directory number**.



Rule definition: If within the specified **Inspection Time Range**, the total requests initiated by a client IP address exceeds the specified number (**The total requests exceed**) and **the proportion of the 404 HTTP** status code exceeds the specified proportion, the access requests from this IP address are blocked for the specified time period (**Blocked IP Addresses**). If the requested **Directory number** exceeds the specified number, the requests are also blocked for the specified time period (**Blocked IP Addresses**).

Note:

We recommend that you use **Mode** and choose a built-in configuration mode from **Flexible Mode**, **Strict Mode**, and **Normal Mode**. You can adjust the parameters as needed.

d. Click **Confirm**.

Unblock a blocked IP address: Click **Unblock IP Address** to unblock the target IP address.

- **Scanning Tool Blocking**: Enable or disable scan tool blocking.

  After you enable scan tool blocking, common scan tool behaviors are identified. If the behaviors of an access request match scan behaviors, this access request is always blocked. If you disable scan tool blocking, scan activities are no longer blocked.

- **Collaborative Defense**: Enable or disable collaborative protection.

  After you enable collaborative protection, the access requests are blocked if they are initiated by the IP addresses from the Alibaba Cloud library that records malicious IP addresses.

1.4.5 Create a custom protection policy

After you set up Web Application Firewall (WAF) for a website, you can create custom protection policies to protect the website. Custom protection policies allow you to customize ACL rules based on precise match conditions and specify the maximum request rate. Custom protection policies can be tailored for different scenarios, such as hotlinking protection and website backend protection. You can customize protection rules as needed.

Notice:
This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, see #unique_42.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

Background information

Custom protection policies are defined by custom rules. Custom rules include ACL rules and anti-HTTP flood rules.

- An ACL rule filters requests based on the client IP address, request URL, and precise match conditions that use common request headers.
- An anti-HTTP flood rule filters requests based on the precise match conditions and request rate that you have set.

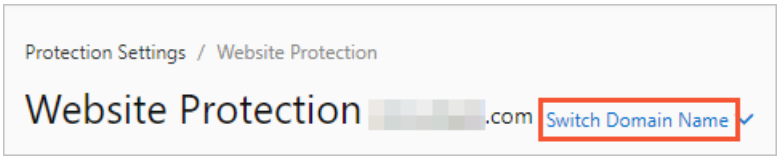
Limits

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

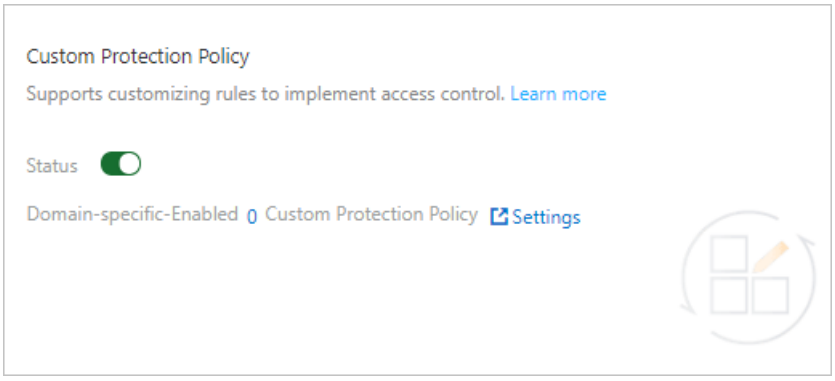
Specification	Description	Enterprise	Business	Pro
Number of custom rules	The maximum number of custom rules that you can create.	200	100	100
Advanced match fields	The advanced match fields other than IP addresses and URLs that you can specify in custom rules.	Supported	Supported	Not supported
Rate limiting	Custom anti-HTTP flood rules.	Supported	Supported	Not supported
Custom statistical objects	The custom statistical objects other than IP addresses and sessions that can be used to control the request rate.	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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



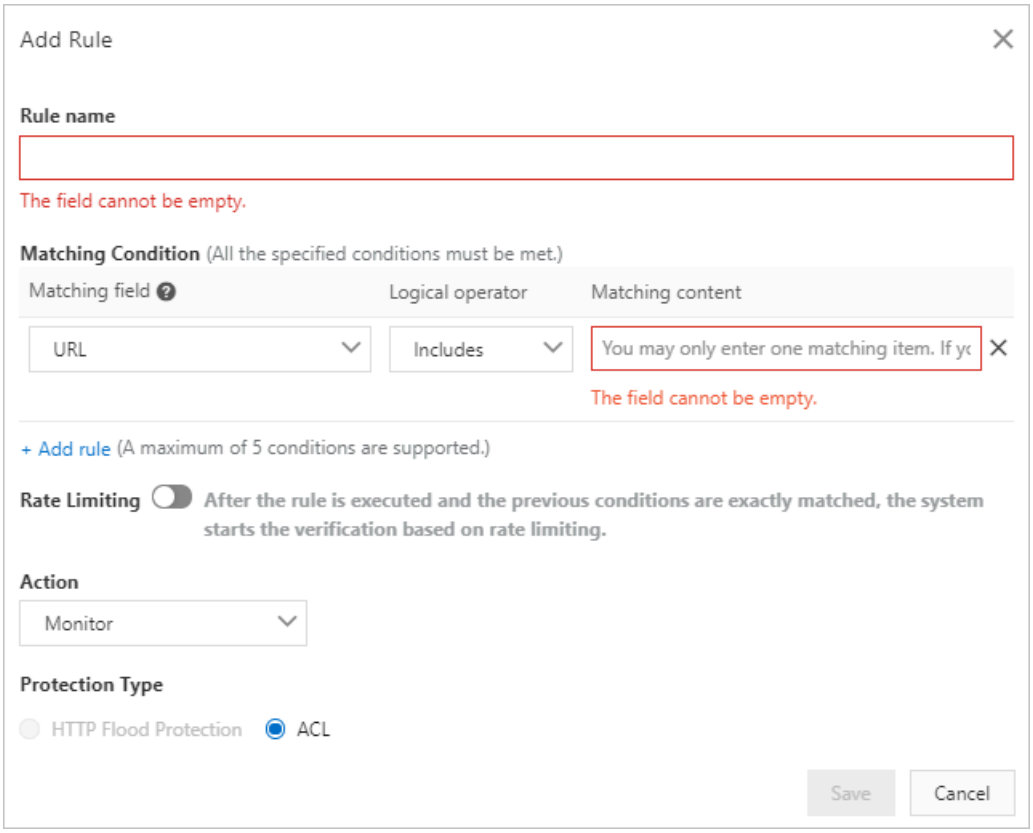
5. Click the **Access Control/Throttling** tab and find the **Custom Protection Policy** section. Turn on the **Status** switch and click **Settings**.



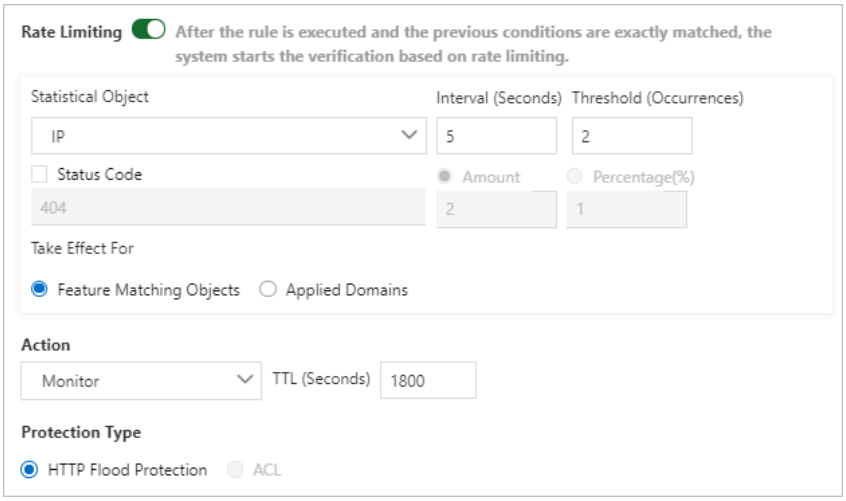
6. Create a custom rule.

   a) On the **Custom Protection Policy** page, click **Create Custom Protection Policy**.

   b) In the dialog box that appears, set the following parameters.



Parameter	Description
Rule name	Specify a name for the rule.

Parameter	Description
Matching Condition	Specify the detection logic of the rule. The rule is triggered only after the specified conditions are met. Click **Add rule** to add more conditions. You can specify a maximum of five conditions. If you have specified multiple conditions, the rule is hit only after all the conditions are met. For more information about match conditions, see Fields of match conditions.
Rate Limiting	Enable or disable rate limiting. WAF starts calculating the request rate only after the specified match conditions are met. Before you enable rate limiting, set the parameters to specify the object to be calculated.  For more information about rate limiting parameters, see Rate limiting parameters.
Action	Specify the action to be performed after the rule is triggered. Supported actions include: • **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)**, which is the effective time period of the action.

Parameter	Description
Protection Type	Specify the type of the rule. This parameter is automatically set based on the status of . • 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	Specify the object whose request rate is calculated. Valid value: • **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 requests that are allowed from the object during the specified time period. If this limit is exceeded , rate limiting is triggered.
Status Code	After the specified match conditions are met, 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 specified status code.

Parameter	Description
Take Effect For	Specify the objects to which rate limiting is applied. • **Feature Matching Objects** • **Applied Domains**

c) Click **Save**.

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

1.5 Bot management

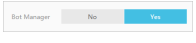
1.5.1 Configure the bot management whitelist

Bot management protects web applications, native applications, and APIs from malicious crawlers for protected websites. Bot management allows requests from specific crawlers, and supports bot threat intelligence rules, data risk control, and application protection. You can configure a bot management whitelist to allow specific requests to skip specified detection modules.

Notice:
This topic uses the new version of the Web Application Firewall (WAF) console released in January 2020. If your WAF instance was created before January 2020, bot management whitelists are not supported.

Prerequisites

- A Web Application Firewall instance that is deployed in a region inside mainland China and the **Bot Manager** feature are available.



- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

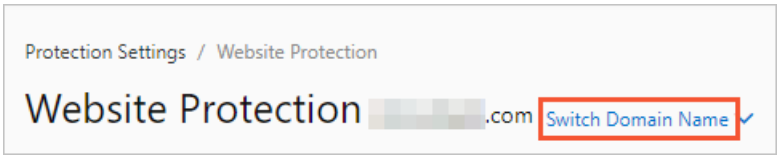
Background information

For more information about detection modules supported by bot management, see the following topics:

- Set a threat intelligence rule to allow requests from specific crawlers
- Set a bot threat intelligence rule
- Configure data risk control
- 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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.

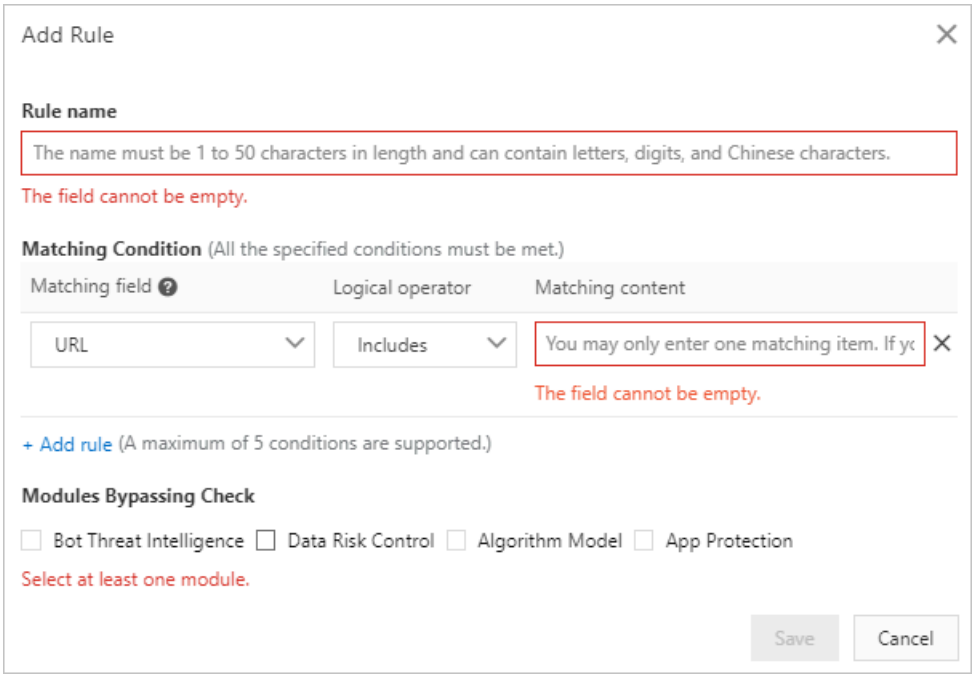


5. Click the **Bot Management** tab, find the **Bot Management** section, and then click .

6. Create a bot management whitelist.

   a) On the **Bot Management - Whitelist** page, click **Create Rule**.

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

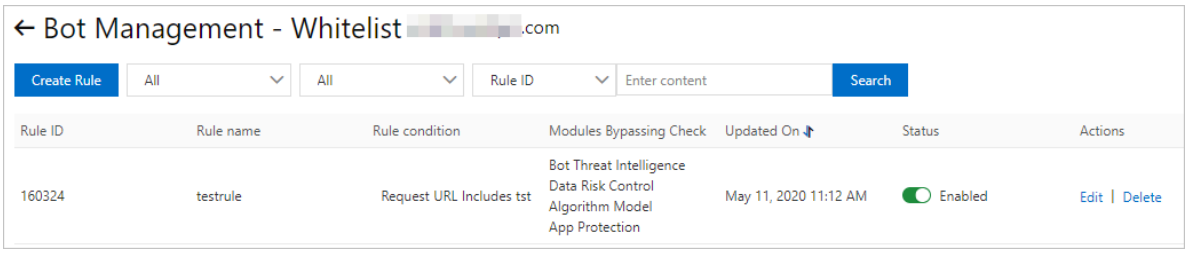


Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify the match conditions. Click **Add rule** to add more conditions. You can add a maximum of five conditions. If you specify multiple conditions, the rule is hit only after all the specified conditions are met. For more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

Parameter	Description
Modules Bypassing Check	The detection modules that can be skipped after the rule is hit. Detection modules include: • **Bot Threat Intelligence** • **Data Risk Control** • **Algorithm Model** • **App Protection**

c) Click **Save**.

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



1.5.2 Set a threat intelligence rule to allow requests from specific crawlers

This feature maintains a whitelist of authorized search engines, such as Google, Bing, Baidu, Sogou, 360, and Yandex, to facilitate the management of crawler requests that are forwarded to the target domain.

Notice:
This topic uses the new version of the Web Application Firewall (WAF) console released in January 2020. If your WAF instance was created before this date, you cannot use this feature.

Prerequisites

- A Web Application Firewall instance that is deployed in a region inside mainland China and the **Bot Manager** feature are available.

- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

Background information

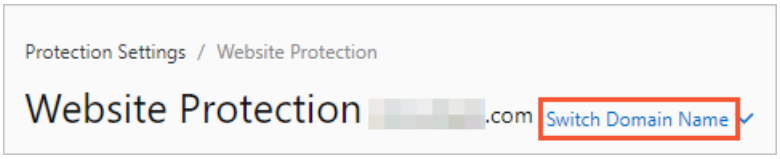
Rules described in this topic allow requests from specific crawlers to the target domain based on the Alibaba Cloud crawler library. The Alibaba Cloud crawler library is updated in real time based on the analysis of network traffic that flows through Alibaba Cloud, and captures the characteristics of 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 a rule that allows requests from specific crawlers to the target domain, requests initiated from the crawler IP addresses of the authorized search engines are directly sent to the target domains. The bot management module no longer detects these requests.

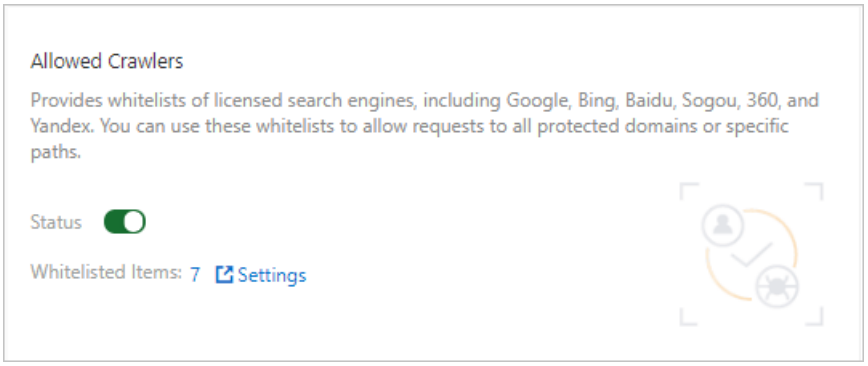
Note:
Alternatively, you can use other protection features, such as ACL rules and traffic throttling rules, to filter requests from IP addresses of crawlers that are in the whitelist.

Procedure

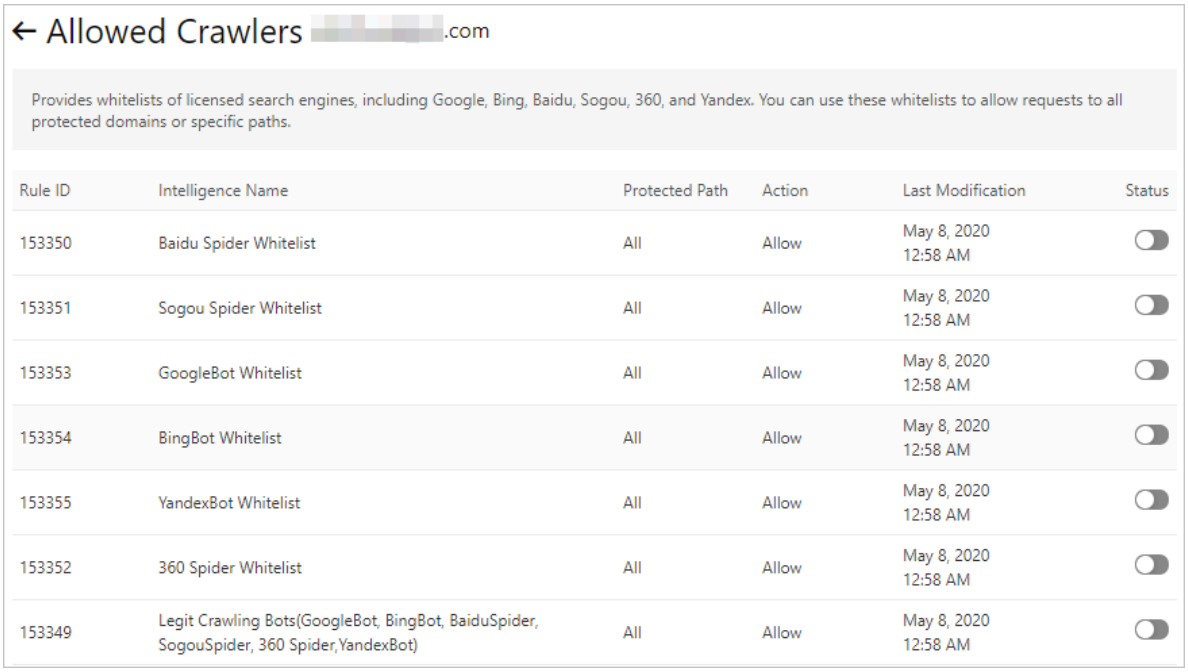
1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the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



5. Click the **Bot Management** tab and find the **Allowed Crawlers** section. Turn on the **Status** switch and click **Settings**.



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



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.

1.5.3 Set a bot threat intelligence rule

Threat intelligence 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 site or specific directories.

Notice:

This topic uses the new version of the Web Application Firewall (WAF) console released in January 2020. If your WAF instance was created before January 2020, you cannot set bot threat intelligence rules.

Prerequisites

- A Web Application Firewall instance that is deployed in a region inside mainland China and the **Bot Manager** feature are available.

- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

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, and captures the characteristics of requests that are initiated from crawlers. The Alibaba Cloud crawler library contains IP addresses of crawlers, public clouds, and on-premises data centers. The IP addresses are dynamically calculated and updated based on the threat intelligence collected from network traffic that flows through Alibaba Cloud.

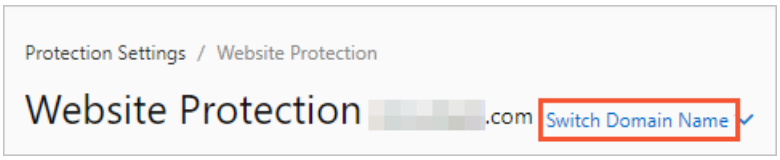
Note:
IP addresses of public clouds and on-premises data centers are also contained in the crawler library because a large number of crawlers are deployed on cloud servers. However, general users rarely access your workloads through the source IP address of a public cloud or on-premises data center.

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 service logic.

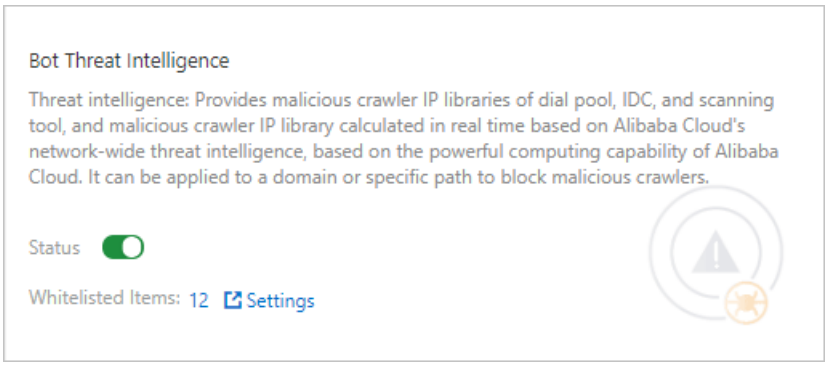
Procedure

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 , in which the instance is deployed.

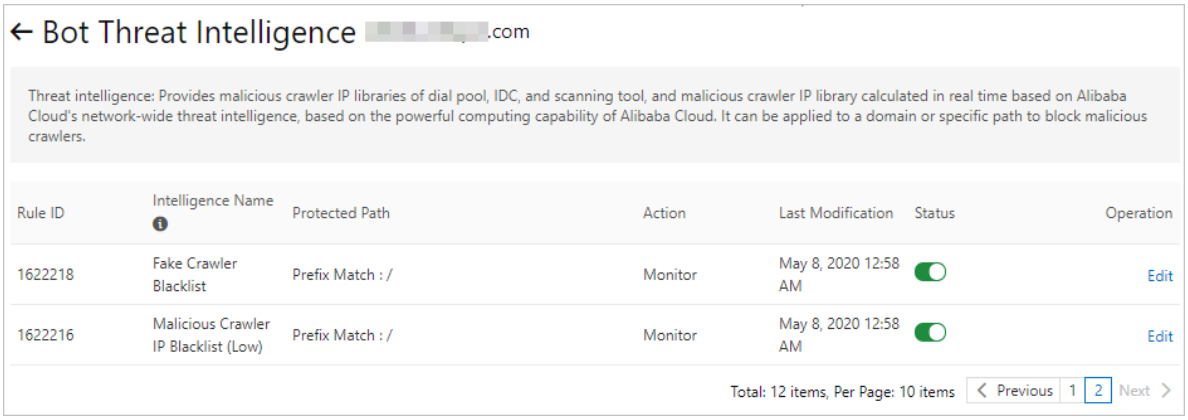
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Bot Management** tab and find the **Bot Threat Intelligence** section. Turn on the **Status** switch and click **Settings**.



6. In the **Bot Threat Intelligence** rule list, find the target threat intelligence library by **Intelligence Name**, and turn on the **Status** switch.



The following table lists the bot threat intelligence libraries that are supported by WAF.

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 dynamicall y updated based on the source IP addresses of scan attacks detected on Alibaba Cloud.

Intelligence library	Description
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 the whitelist of crawlers. Otherwise, false positives may occur. For more information, see Set a threat intelligence rule to allow requests from specific crawlers.
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, such as API calls, we recommend that you set threat intelligence rules based on the low-severity library.
IDC IP List	This library contains IP addresses of public clouds and on-premises data centers, including Alibaba Cloud, Tencent Cloud , Meituan Open Services, 21Vianet, and other public clouds . Attackers typically use CIDR blocks of public clouds or on-premises data centers to deploy crawlers or as proxies to access sites. General users rarely access sites in this way.

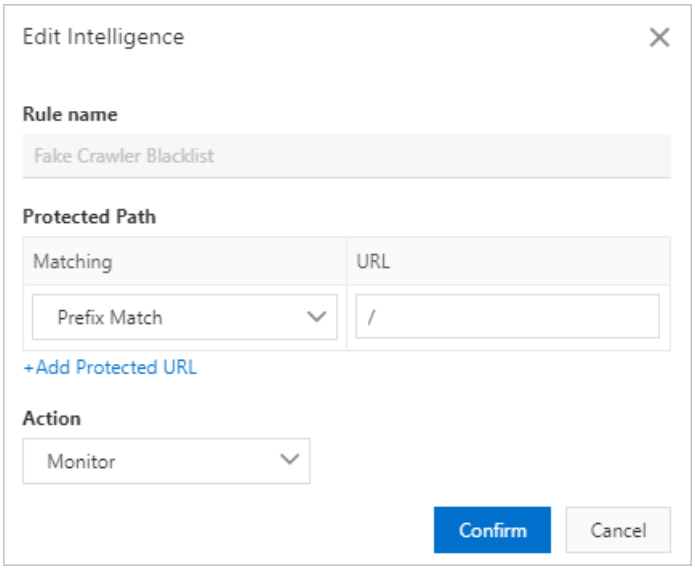
After you enable the default rule, requests initiated from IP addresses in the threat intelligence library to any directory of the protected domain trigger the **Monitor** action. This action allows the requests to the destination directories and records the events.

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

7. Optional: Customize a threat intelligence rule.

   a) Find the target rule, and click **Edit** in the Actions column.

   b) In the **Edit Rule** dialog box that appears, set the following parameters.



Parameter	Description
Protected Path	• **URL**: specifies the URL that you want to protect, such as /abc and /login/abc. You can also enter a single forward slash (/) to include all directories. • **Matching**: specifies a condition for matching the URL. - **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. - **Regular Expression 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	Specifies the action to be performed after the match conditions of the rule are met. Supported actions include: • **Monitor**: allows the request to the destination directory and records the event. • **Block**: blocks the request. • **JavaScript Validation**: requires JavaScript verification. Request 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. • **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.

c) Click **Confirm**.

1.5.4 Configure data risk control

After you set up Web Application Firewall (WAF) for a website, you can enable data risk control to protect the website. Data risk control helps you protect crucial website services, such as registrations, logons, activities, and forums, against fraud. You can customize data risk control rules based on your actual requirements.

Notice:

This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, see #unique_44.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.

- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.
- If the billing method of the instance is subscription, the edition of the instance must be Business or Enterprise.

Background information

Data risk control is based on Alibaba Cloud big data. It uses industry-leading engines for risk decision-making and integrates with human-machine identification technologies to protect crucial services in different scenarios against fraud. To use data risk control, set up WAF for your website. No further configurations are required on the server or client side.

Data risk control is suitable in a wide array of scenarios, including but not limited to: malicious registrations, 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 spam.

For details about scenarios and protection effects of data risk control, see Examples.

Compatibility

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

- Static web pages that can be directly accessed through its URL by visitors, such as HTML details page, shared pages, website homepages, and documents. Web pages that have adopted redirection methods such as direct modifications of `location.href`, and uses of `window.open` and anchor tags.
- 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.
- Requests in the service code contain hooks.

Solutions

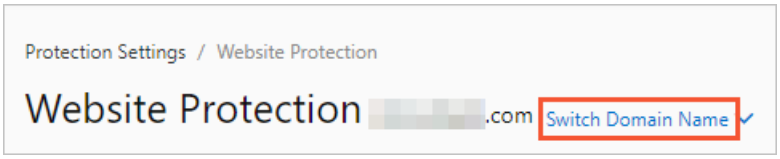
We recommend that after you enable data risk control, choose the detection mode and use data risk control together with real-time log analysis to run a compatibility test. For more information, see #unique_45.

If incompatibility occurs, use Alibaba Cloud Human-Machine Validation together with WAF to protect your websites.

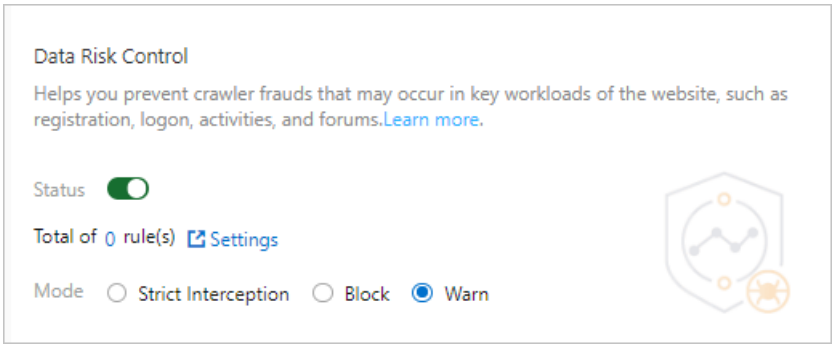
To protect native applications, we recommend that you use the Anti-Bot SDK. For more information, see Configure application protection.

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Bot Management** tab and find the **Data Risk Control** section. Set the following parameters and click **Settings**.



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 all pages of the website. Web pages are returned to visitors as compressed data in formats other than GZIP. If your website uses non-standard ports, no further configurations are required. **Note:** You must enable data risk control before you can set the mode and protection rules.

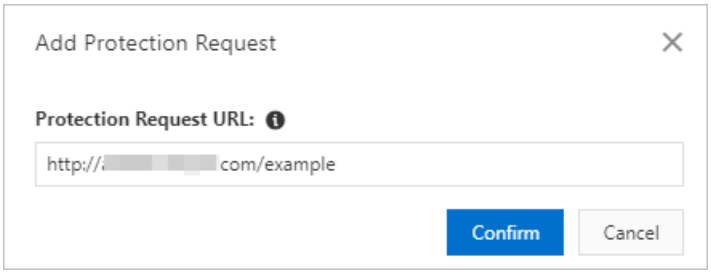
Parameter	Description
Mode	Specify a protection mode for data risk control. Supported modes: • **Strict Interception**: If WAF determines that the workloads are under attack, visitors are required to pass strict two-factor authentication. • **Block**: If WAF determines that the workloads are under attack, visitors are required to pass two-factor authentication. • **Warn**: If WAF determines that the workloads are under attack, requests are forwarded to your website but relevant events are recorded. You can view detailed information in risk reports.  **Note:** The warning mode is selected by default. In this mode, data risk control does not block requests, but inserts a JavaScript plug-in into static web pages to analyze client behaviors.

6. Add a data risk control request.

   a) On the **Data Risk Control** page, click the **Protection Request** tab, and then click **Add Protection Request**.

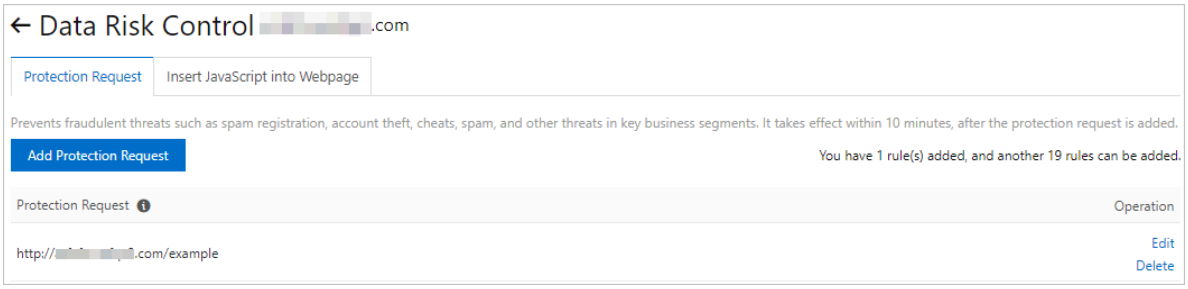
   b) In the **Add Protection Request** dialog box, enter the URL that you want to protect in the **Protection Request URL** field.

   For more information, see What is a protected URL in a protection request.



   c) Click **Confirm**.

   A newly added request takes effect after 10 minutes. You can view newly added requests in the request list, and modify or delete requests.



7. Optional: Specify the web page into which you want to insert the JavaScript plug-in.

   Some web page code may be incompatible with JavaScript. In this case, we recommend that you insert JavaScript into specific pages that are compatible with JavaScript.

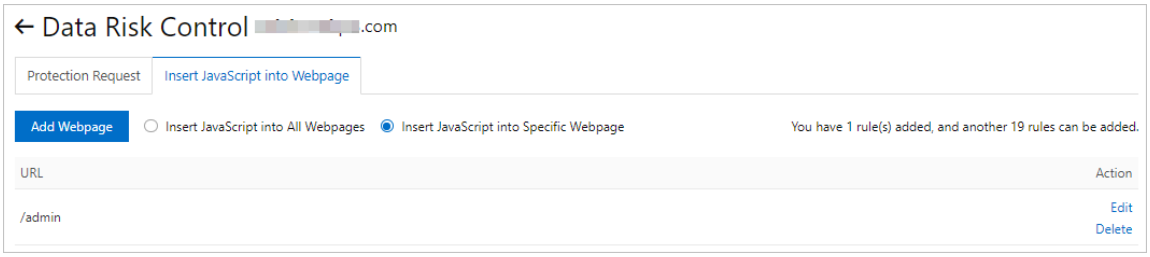
Note:

When JavaScript is inserted into specific pages, data risk control may fail to obtain all visitor behaviors and reduce protection capabilities.

Limits: You can insert JavaScript into a maximum of 20 URLs.

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

b) Select **Insert JavaScript into Specific Webpage** and click **Add Webpage**.



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

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

After data risk control is enabled, you can use the logs feature provided by WAF to monitor the protection status. For more information, see View protection results.

What is 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 need to add a protection request to protect the endpoints www.abc.com/getsmscode and www.abc.com/register.do, respectively. WAF protects these URLs from SMS message abuse and malicious registrations. If you set the protected URL to www.abc.com/new_user, general visitors are also required to pass CAPTCHA verification, which may adversely affect the user experience.

Notes on protected URLs

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

  For example, if you set the protected URL to www.test.com/test, data risk control only filters request sent to this URL. Data risk control does not filter requests sent to the sub-directories of this URL.
- Data risk control only protects website directories.

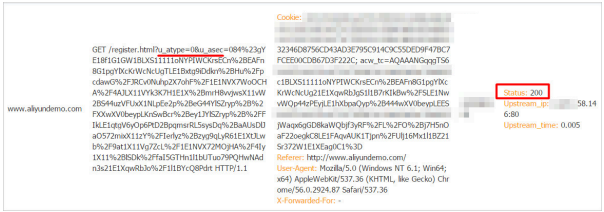
  For example, if you set the protected URL to www.abc.com/book/*, data risk control filters requests sent to all pages under www.abc.com/book. We recommend that you do not set data risk control to monitor the entire website. If you set the protected URL to www.abc.com/*, general visitors need to pass CAPTCHA verification to visit the website homepage. This adversely affects the user experience.
- Requests sent to a protected URL always trigger CAPTCHA verification. Make sure that the protected URL is not directly requested by general visitors in normal cases. Otherwise, general visitors must pass multiple layers of verification before they can visit the URL.
- Data risk control does not support API calls. API calls are directly initiated machine actions and cannot pass the CAPTCHA verification of data risk control. However, if an API operation is called by general visitors clicking a button on a page, you can implement data risk control.

View protection results

You can use the logs feature provided by WAF to monitor the protection status and view blocked requests.

- Allowed requests

  The following figure shows a request that has passed data risk control verification. The URL of a request from a general visitor that has passed data risk control verification contains a parameter that starts with u_a. This 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 has been 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 User-Agent parameter. This type of request is blocked by WAF and the origin server does return any response.

After you enable the logs feature, you can navigate to **Advanced Search** > **URL Key Words** and set the endpoints to be protected by data risk control. This feature helps you monitor the status of data risk control and records blocked requests. For more information, see #unique_46.

Examples

User Tom has a website and the website domain is www.abc.com. General visitors can register as website members at www.abc.com/register.html. User Tom noticed that attackers can use malicious scripts to submit registration requests. 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. Compared to traditional HTTP flood attacks, this type of malicious request is difficult to identify.

Sample configurations

User Tom set up WAF for the website and enabled data risk control for the domain www.abc.com. The URL of the most crucial registration service is www.abc.com/register.html. Therefore, User Tom set 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. 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 data provided by 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 visitor visiting the website to submitting the registration request. For example, if a visitor directly submits a registration request without performing other operations first, the request is potentially malicious.

- If data risk control determines that the request is from a general visitor based on previous visitor behaviors, the visitor can register accounts without verification.
- If data risk 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 general 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 every page of the website to analyze visitor behaviors. However, protection and verification are only enabled for the URL www.abc.com/register.html where visitors submit registration requests. Data risk control is triggered only after a registration request is submitted.

1.6 Integrated App protection

1.6.1 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.

1.6.2 Integrate the Anti-Bot SDK into iOS applications

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

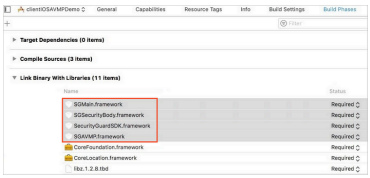
SDK files for iOS applications

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

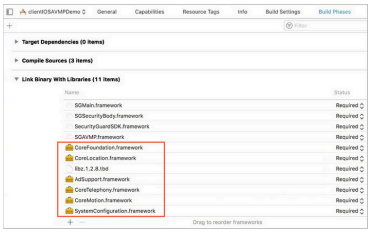
Configure an iOS project

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.

Parameter	Type	Required	Description
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:NSURLRequestReloadIg
noringCacheData timeoutInterval: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 startImmediately: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.
1904	Upgrade the image version. The AVMP signature function only supports v5 images.

Error code	Description
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.

1.6.3 Integrate the Anti-Bot SDK into Android applications

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

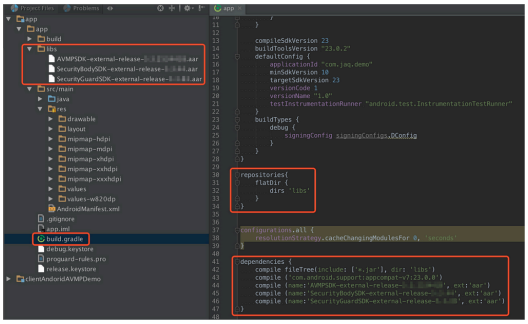
SDK files for Android applications

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.

File name	Description
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.

a. 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.

b. 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.

```
buildTypes {
    release {
        minifyEnabled true
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
```

   - 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()).getInterface(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`.

Parameter	Type	Required	Description
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
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.

Error code	Description
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.
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.

1.6.4 Configure application protection

Application protection provides secure connections and anti-bot protection for native applications. This feature 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.

Notice:
This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, you cannot use the application protection feature.

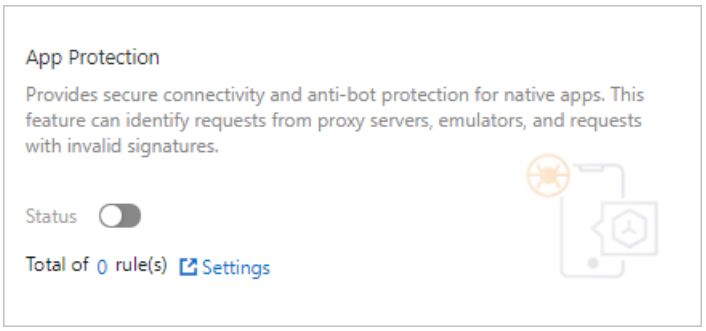
Prerequisites

- You have activated a Web Application Firewall instance, and have purchased the **Mobile App Protection** module.

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

Procedure

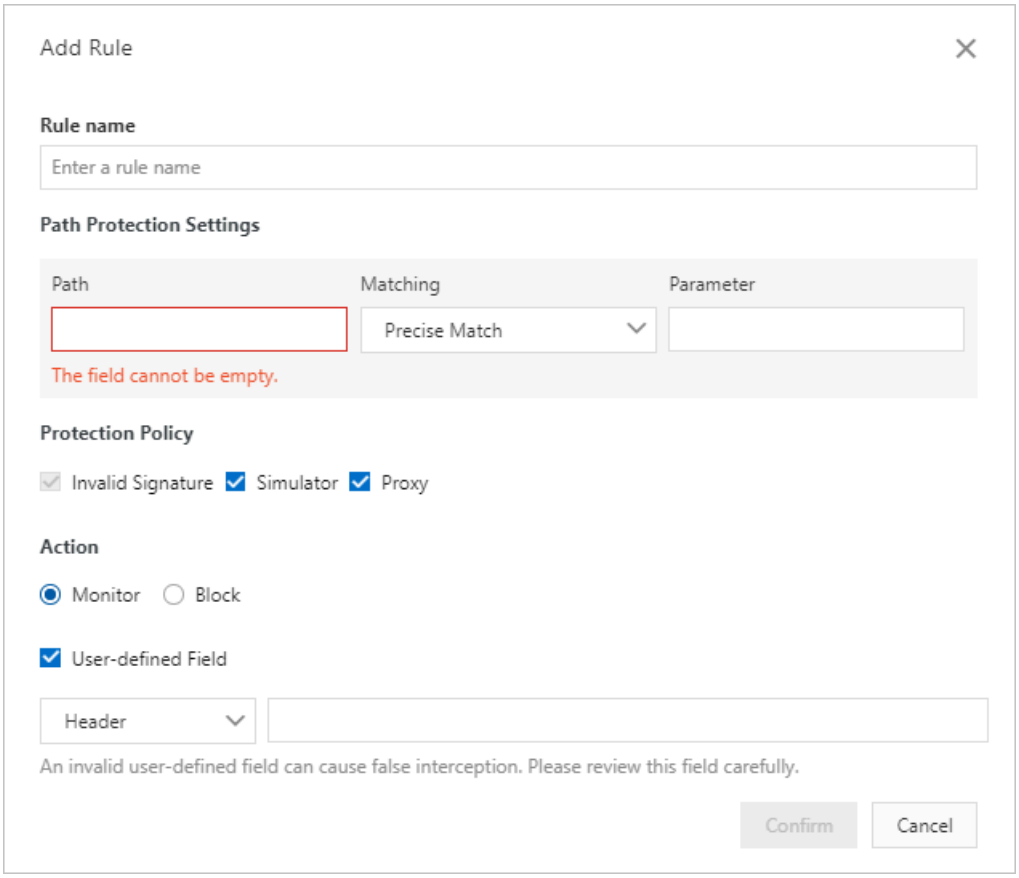
1. Log on to the Web Application Firewall console.
2. In the top navigation bar, choose a resource group and a region (**Mainland China** or **International**).
3. In the left-side navigation pane, choose **Protection Settings** > **Website Protection**.
4. Click the **Bot Management** tab, find **App Protection** in the **Bot Management** section, and click **Settings**.



5. Create a path protection rule.

   a) On the **App Protection** page, find the **Interface Protection** section, and click **Add Rule**.

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



Note:

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

Parameter	Description
Rule Name	Specify a name for the rule.

Parameter	Description
Path Protection Settings	Specify the path that you need to protect. The following parameters are required: • **Path**: Required. 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 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.

Parameter	Description
Protection Policy	Select a protection policy. • **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 the 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. We recommend that you select this policy. 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.
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 domains 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.

Parameter	Description
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. After you have selected the User-defined Field check box, 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.

c) Click **Confirm**.

6. Enable version protection.

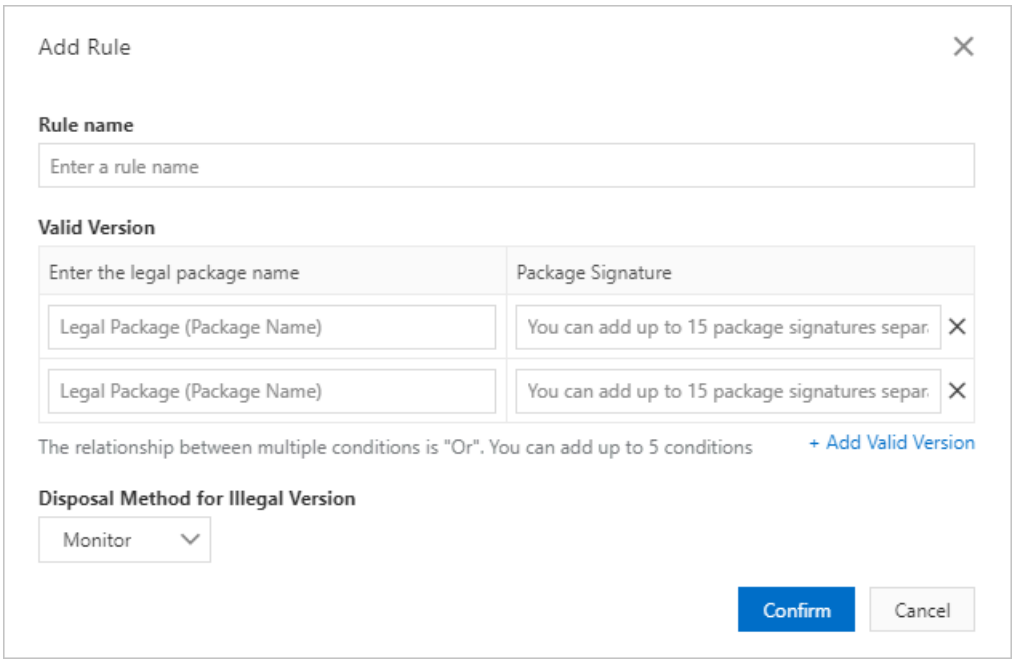
   You can configure version protection to block requests from non-official applications. You can also use this feature to verify the validity of an application.

   Note:

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

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

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



Parameter	Description
Rule Name	Specify a name for the rule.

Parameter	Description
Valid Version	Specify the valid versions of an application. • **Enter the legal package name**: Specify 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, only the package name will be verified. **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.

c) Click **Confirm**.

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

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.

1.7 Data security

1.7.1 Configure the data security whitelist

The data security feature prevents the content of web pages protected by Web Application Firewall (WAF) from being leaked or modified. This feature helps you maintain data integrity and confidentiality. Data security supports website tamper-proofing and data

leakage prevention. You can configure the data security whitelist to have specific requests skip specified detection modules.

Notice:
This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, you cannot use the data security whitelist.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

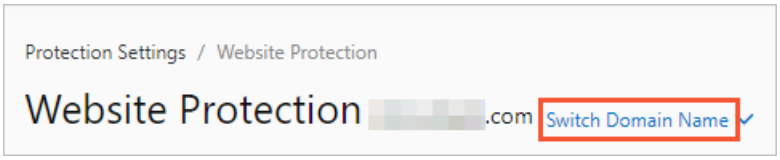
Background information

For more information about detection modules supported by data security, see the following topics:

- Configure tamper-proofing
- Configure data leakage prevention

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.

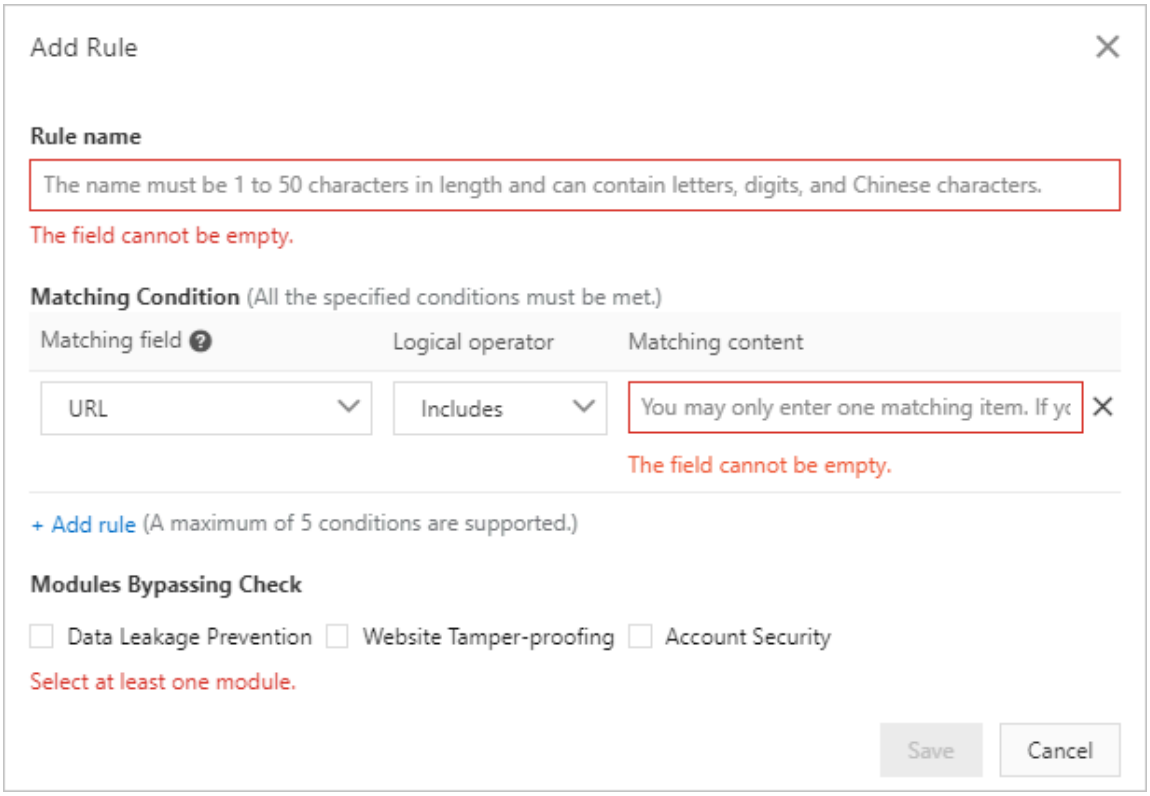


5. Click the **Web Security** tab, find the section, and then click .

6. Create a data security whitelist.

   a) On the **Data Risk Control - Whitelisting** page, click **Create Rule**.

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

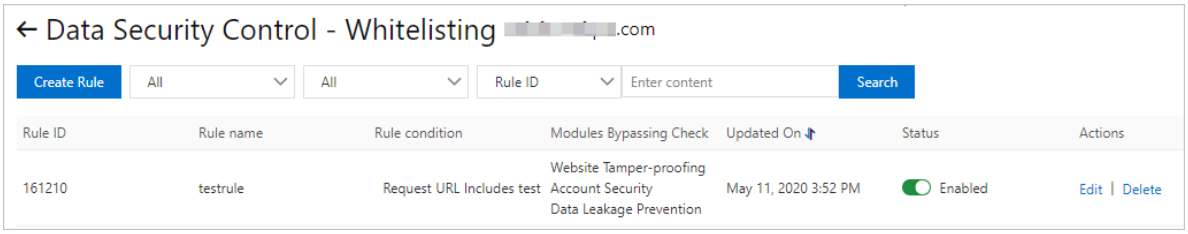


Parameter	Description
Rule name	Specify a name for the rule.
Matching Condition	Specify the match conditions of the whitelist rule. Click **Add rule** to add more conditions. You can specify a maximum of five conditions. If you have specified multiple conditions, the rule is matched only after all the specified conditions are met. For more information about match conditions, see Fields of match conditions.

Parameter	Description
Modules Bypassing Check	Valid values: • **Data Leakage Prevention** • **Website Tamper-proofing** • **Account Security**

c) Click **Save**.

After a whitelist rule is created, it is automatically enabled. You can view newly created rules, and disable, modify, or delete rules as needed.



1.7.2 Configure tamper-proofing

After you set up Web Application Firewall (WAF) for a website, you can enable the tamper-proofing feature to protect the website from website defacement. 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 tamper-proofing rules as needed.

Notice:

This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, see #unique_53.

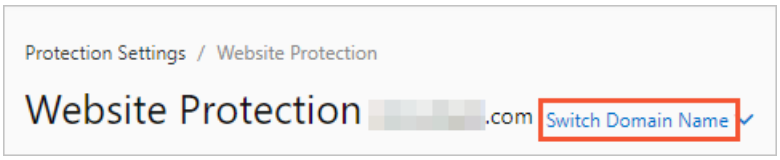
Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

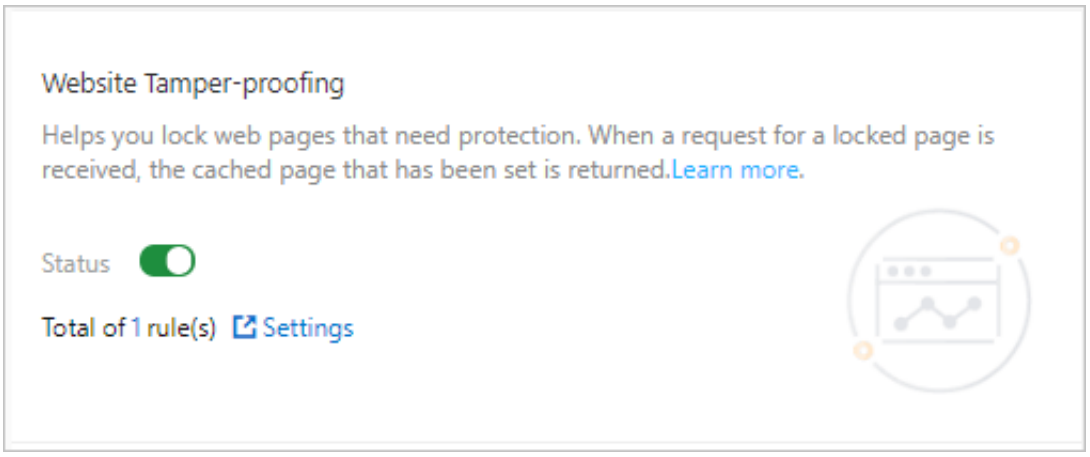
Procedure

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 , in which the instance is deployed.

3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Web Security** tab and find **Website Tamper-proofing** in the **Data Security** section. Turn on the **Status** switch and click **Settings**.



Note:
You must enable tamper-proofing before you can create protection rules.

6. Create a tamper-proofing rule.

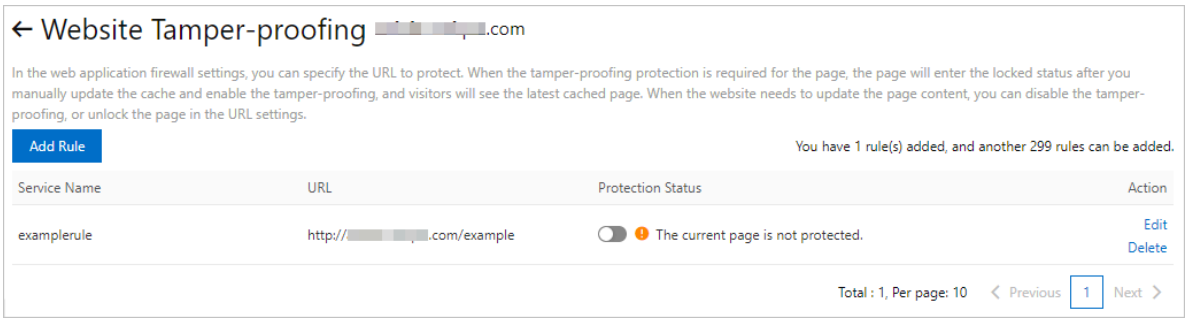
   a) On the **Website Tamper-proofing** page, click **Add Rule**.

   b) In the **Add Rule** dialog box that appears, specify the **Service Name** and **URL** of the web page that you need to protect.

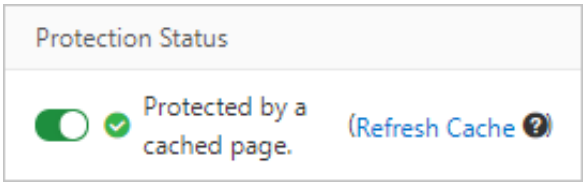
   - **Service Name**: Specify the name of the service that the web page provides.
   - **URL**: Enter the exact path. Wildcard characters such as `/*`, or parameters such as `/abc? xxx=` are not supported. Text data, HTML pages, and images under the specified path are protected.

   c) 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 the **Protection Status** of the rule is disabled.



7. Enable the rule. Find the target rule in the rule list, and turn on the **Protection Status** switch.



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

8. Optional: Update cached data. Find the target rule 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.

1.7.3 Configure data leakage prevention

After you set up Web Application Firewall (WAF) for a website, you can enable data leakage prevention for the website. Data leakage prevention helps websites filter 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 denying access to visitors. You can customize data leakage prevention rules as needed.

> **Notice:**
> This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, see #unique_54.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.

Background information

WAF provides the data leakage prevention feature to comply with the following regulation s 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 leakage, 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 leaks, 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.
- 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

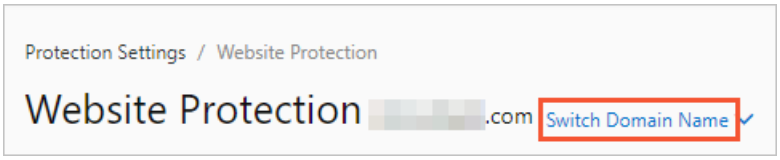
Based on the specified protection rules, data leakage prevention detects whether a web page contains sensitive information, such as identity card numbers, phone numbers, and band card numbers. If a rule is matched, WAF triggers alerts or masks the informatio n based on the action specified in the rule.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

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

Procedure

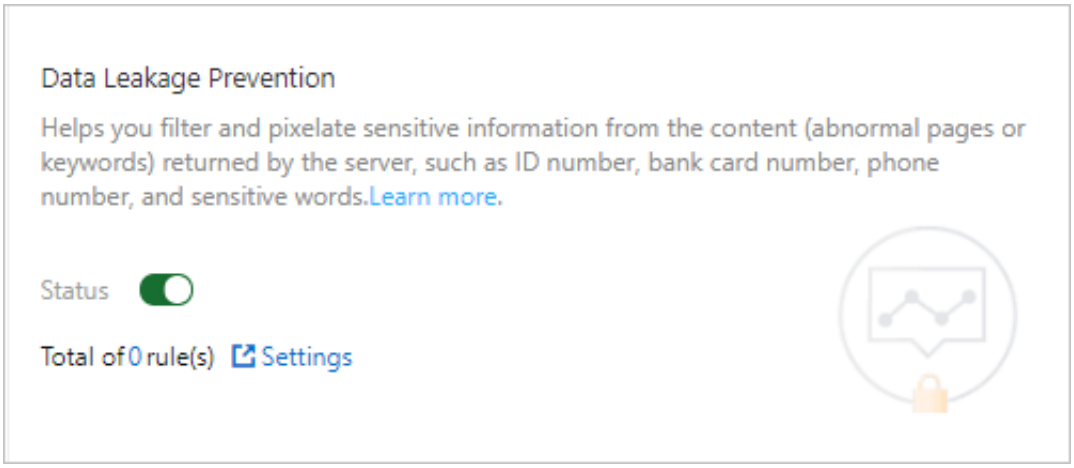
1. Log on to the Web Application Firewall console.

2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Web Security** tab and find **Data Leakage Prevention** in the **Data Security** section. Turn on the **Status** switch and click **Settings**.

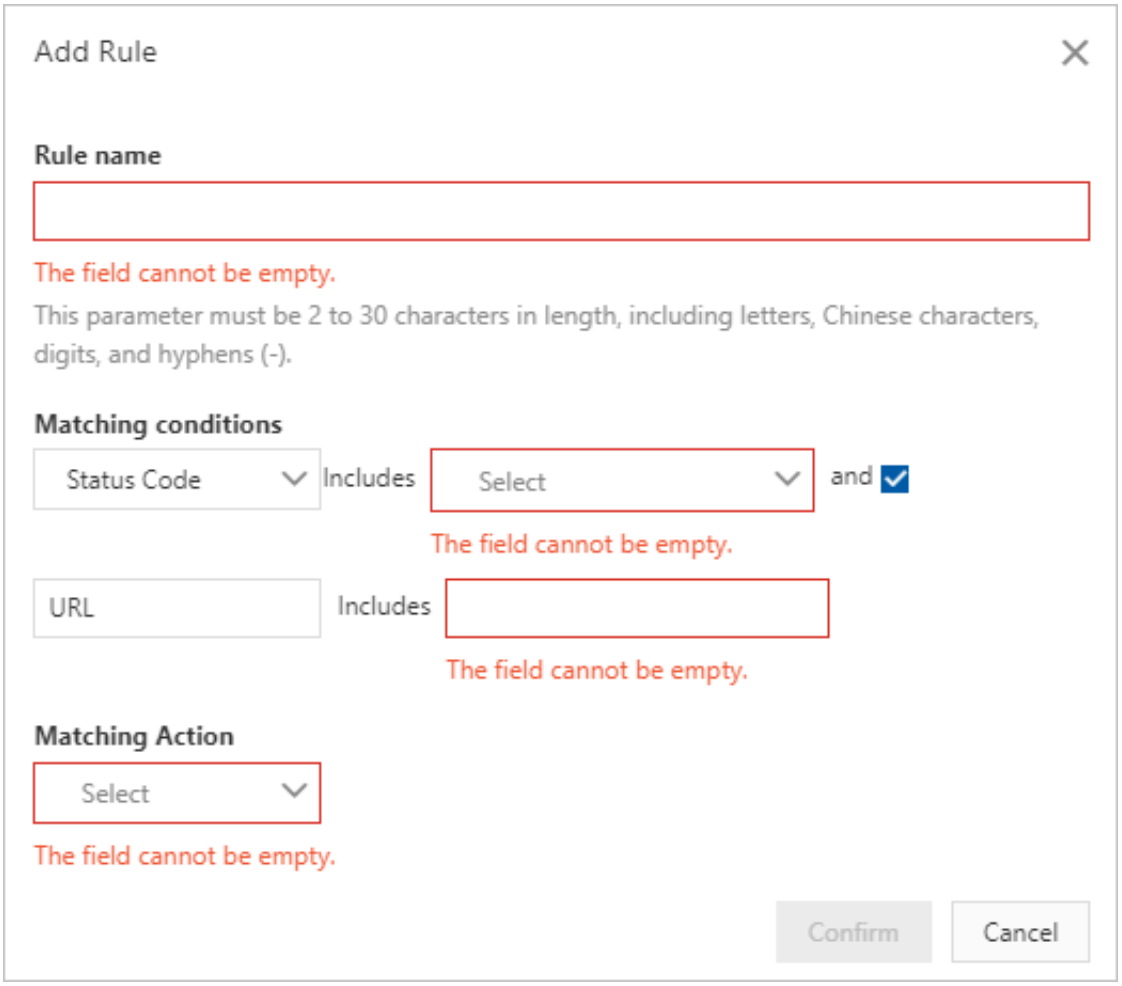
Note:
You must enable data risk prevention before you can set protection rules.



6. Create a data leakage prevention rule.

   a) On the **Data Leakage Prevention** page, click **Add Rule**.

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



Parameter	Description
Rule name	Specify a name for the rule.

Parameter	Description
Matching conditions	Specify 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**. 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 the specified page only.
Matching Action	Specify the action to be performed on detected sensitive information. • If you set the match condition to HTTP status codes, supported actions include: - **Warn**: triggers alerts upon sensitive information leaks. - **Block**: blocks requests and returns the default page denying access. • If you set the match condition to sensitive information, 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 you 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.

> **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 you how to create a rule that blocks the 404 HTTP code.
  - **Matching conditions**: **Status Code** of **404**
  - **Matching Action**: **Block**

  After this rule is applied, if a requested page does not exist, the specified page denying access is returned.
- Masks 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 you 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.

c) 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 #unique_55.

1.8 Advanced mitigation

1.8.1 Configure the positive security model

After you set up Web Application Firewall (WAF) for a website, you can enable the positive security model for the website. A positive security model is also known as a whitelist. The positive security model of WAF uses Alibaba Cloud machine learning algorithms to automatically study 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 as needed.

 **Notice:**

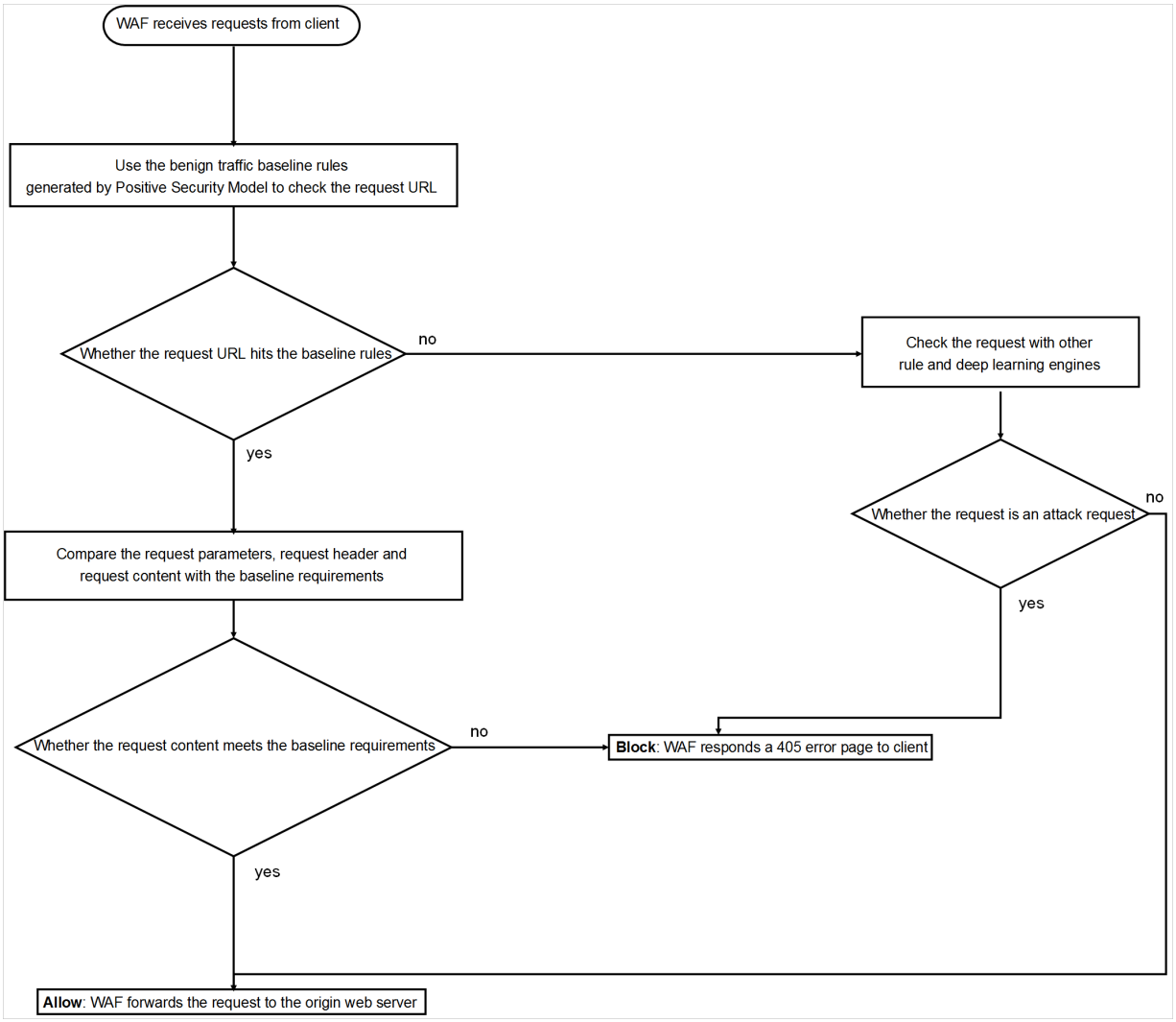
This topic uses the new version of the WAF console released in January 2020. If your WAF instance was created before January 2020, see #unique_57.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.
- Subscription-based WAF instances must use the Enterprise or Exclusive edition. For more information, see #unique_58.

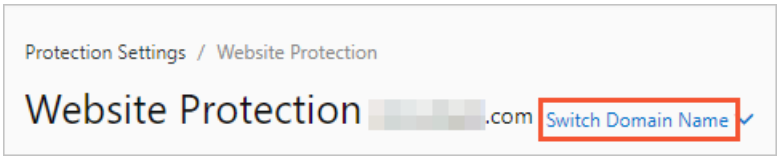
Background information

Traditional protection methods against web attacks are based on detection rules. The positive security model automatically studies the network traffic of a domain 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 domain and tailors security policies for the domain. The positive security model collaborates with other detection modules of WAF to detect attacks at different network layers.

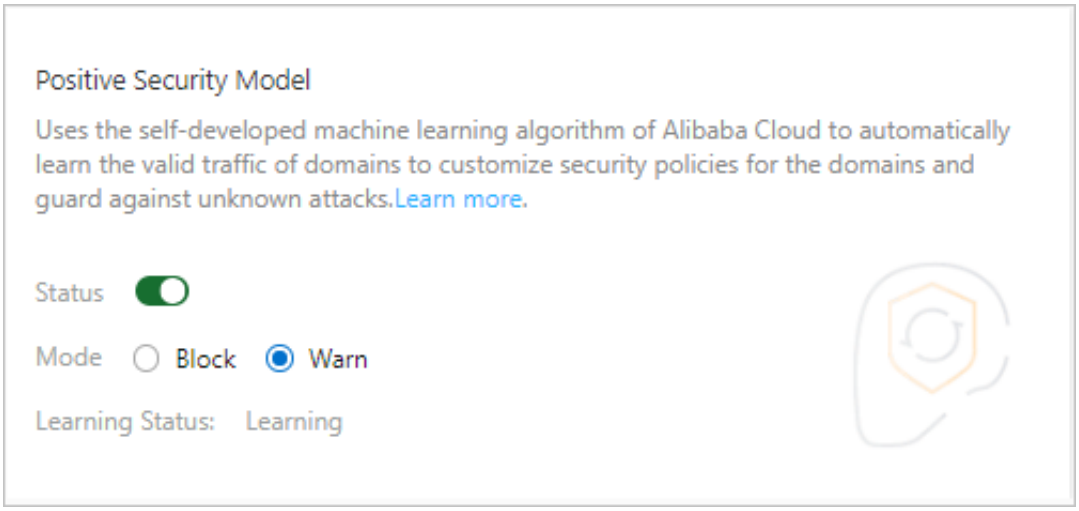


Procedure

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. In the upper part of the page, select the domain name for which you want to configure the whitelist.



5. Click the **Web Security** tab, and find **Positive Security Model** in the **Advanced protection** section.



Parameter	Description
Status	Enable or disable the positive security model.
Mode	Select a mode to manage attacks. Supported modes: • **Monitor**: triggers alerts but does not block requests. • **Block**: blocks requests. **Note:** By default, the positive security model is set to the monitor 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 domain, WAF automatically studies the network traffic history of the domain based on machine learning algorithms. WAF then tailors security rules to protect the domain. The initial machine learning process may take a long time to complete depending on the total amount of network traffic data. Typically, it takes about one hour for WAF to complete learning and generating security rules. After WAF completes the learning process, it notifies you through internal messages, SMS messages, and emails.

1.9 Protection lab

1.9.1 Configure account security

Web Application Firewall (WAF) supports the account security feature. This feature monitors the endpoints related to user authentication, such as registration and logon endpoints, and detects events that may threaten user credentials. Detectable risks include credential stuffing, brute-force attacks, account registration launched by bots, weak password sniffing, and SMS interface abuse. To use the account security feature, add endpoints that need to be monitored by WAF. You can view detection results in WAF security reports.

 **Notice:**

This topic uses the new version of the WAF console released in January 2020. If your WAF instances were purchased before January 2020, see #unique_60.

Prerequisites

- A Web Application Firewall instance is available. For more information, see #unique_27.
- The website is associated with the Web Application Firewall instance. For more information, see #unique_5.
- The billing method of your WAF instance must be a monthly or annual subscription. The WAF instance must use the Business, Enterprise, or Exclusive edition.

Background information

Before you enable account security, obtain the endpoint information that is required for configurations. For example, the domain name, the URL where visitors submit user credentials, and the parameters that specify the username and password. Each WAF instance allows you to enable account security simultaneously for up to three endpoints.

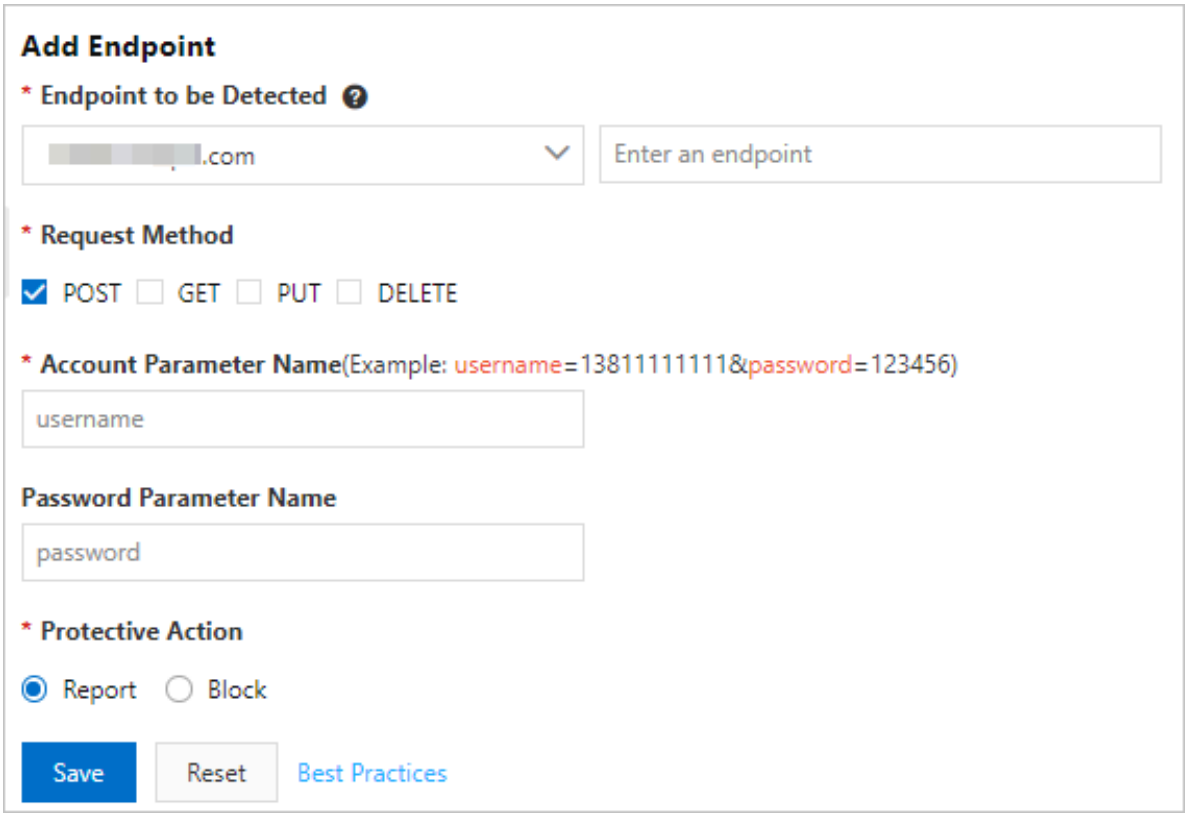
Add an endpoint

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Lab**.
4. On the **Account Security** page, click **Add Endpoint**.

 **Note:**

Each WAF instance allows you to enable account security simultaneously for up to three endpoints. If you have already added three endpoints, the**Add Endpoint** button is dimmed.

5. In the **Add Endpoint** dialog box, set the following parameters and click **Save**.



Parameter	Description
Endpoint to be Detected	Select the domain name that needs account security enabled. Then, enter the URL where user credentials are submitted. Do not enter the endpoint where users log on. For example, do not enter/login.html. Instead, enter the endpoint where visitors enter their usernames and passwords.
Request Method	Select the request method for the endpoint. Valid values: **POST**, **GET**, **PUT**, **DELETE**.
Account Parameter Name	Specify the username field.
Password Parameter Name	Set the parameter that specifies the password field. If passwords are not required to access the endpoint, do not set this parameter.

Parameter	Description
Protective Action	Select the action that manages requests that compromise account security. Valid values: • **Report** • **Block**

Sample configurations

- For example, the logon endpoint is /login.do, and the body of the POST request is username=Jammy&pwd=123456. In this case, set the value of **Account Parameter Name** to username and set the value of **Password Parameter Name** to pwd. You can set the parameters as shown in the screenshot.
- If the parameters that specify user credentials are included in the URL of a GET request, for example, /login.do? username=Jammy&pwd=123456, set the value of **Request Method** to **GET**. Keep other settings the same as those in the figure.
- If passwords are not required to access the endpoint, for example, a registration endpoint, set the **Account Parameter Name** parameter and do not set the **Password Parameter Name** parameter.
- If a phone number is required as a user credential to access the endpoint, then the phone number can be used as the account parameter. For example, the URL is /sendsms.do? mobile=13811111111. In this case, set the value of **Endpoint to be Detected** to /sendsms.do and set the value of **Account Parameter Name** to mobile and do not set **Password Parameter Name**.

After you add the endpoints, WAF automatically dispatches detection tasks. If the network traffic of the endpoint meets the detection conditions, account risks are reported within a few hours.

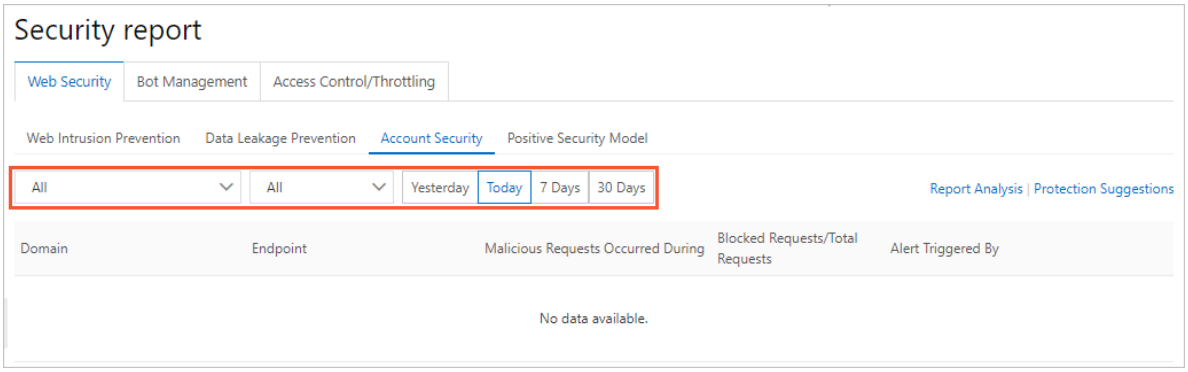
View account security reports

To view account security reports, find the target endpoint on the **Account Security** page, and click **View Report** in the Actions column. You can also view security reports on the **Security report** page in the WAF console.

The following procedure describes how to view security reports on the **Security report** 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 , in which the instance is deployed.
3. In the left-side navigation pane, click .
4. Click the **Web Security** tab, click **Account Security** and select the domain name, endpoint, data range (**Yesterday**, **Today**, **7 Days**, **30 Days**) to check the corresponding account security risks.



The following table lists the fields and descriptions in an account security report.

Field	Description
Endpoint	The URI where account risks are detected by WAF.
Domain	The domain name to which the endpoint belongs.
Malicious Requests Occurred During	The time period during which account risks are detected.
Blocked Requests	The number of requests blocked by WAF protection rules during the time period displayed in the **Malicious Requests Occurred During** column. WAF protection rules indicate those that are currently effective, including Web application protection rules, accurate access control, HTTP flood protection, and blocked regions. The proportion of the blocked requests indicates the account security status of the endpoint.
Total Requests	The total number of requests sent to the endpoint during the time period displayed in the **Malicious Requests Occurred During** column.

Field	Description
Alert Triggered By	The reason why the alert is triggered. Possible reasons include: • A request fits the behavior model of credential stuffing or brute-force attacks. • The traffic baseline of the endpoint is abnormal during the displayed time period. • A large number of requests sent to the endpoint fit the rules described in the threat intelligence library during the displayed time period. • Weak passwords are detected in a large number of requests sent to the endpoint during the displayed time period. In this case, credential stuffing and brute-force attacks may occur.

See also

The account security feature only detects account risks. Due to the variations of businesses and technologies, we recommend that you choose security services based on your actual business requirements to better safeguard your business. For more information, see Account security best practices.

1.9.2 API request security

You can use the API request security function to upload a custom API rule file to ensure only requests that comply with the rules are executed. This protects your website assets from threats such as tampering and replay attacks.

Prerequisites

WAF Business Edition and higher support API security. Ensure that WAF Business, Enterprise, or Exclusive Edition is activated.

Context

After you upload the API rule file, the API security function automatically parses the content of the file, and verifies access requests based on the rules. WAF either blocks API requests that do not comply with the rules or generates alerts for the requests.

Typically, API access requests that have inconsistent request paths or contain parameter values out of the valid range are identified as invalid.

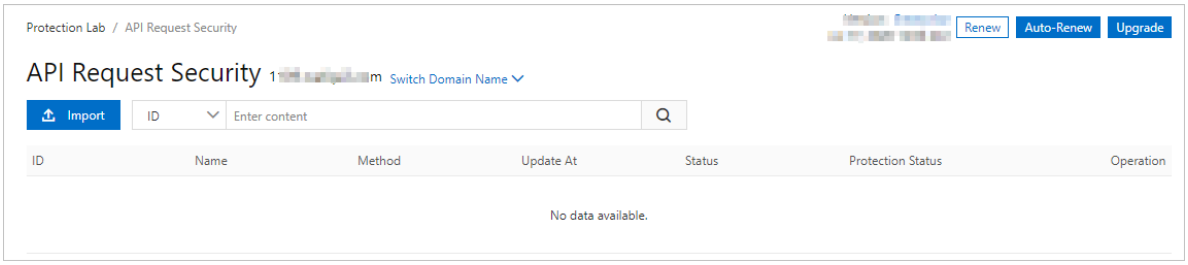
 **Note:**

The API security function is now under public preview and is provided free of charge.

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose **Protection Lab** > **API Request Security**.
4. On the **API Request Security** page, click **Import**.
5. In the dialog box that appears, select the API rule file to be uploaded and click **Open**.

   After the API rule file is imported, the file content is automatically parsed and displayed in the rule list on the **API Request Security** page.



Note:

The file has the following restrictions:

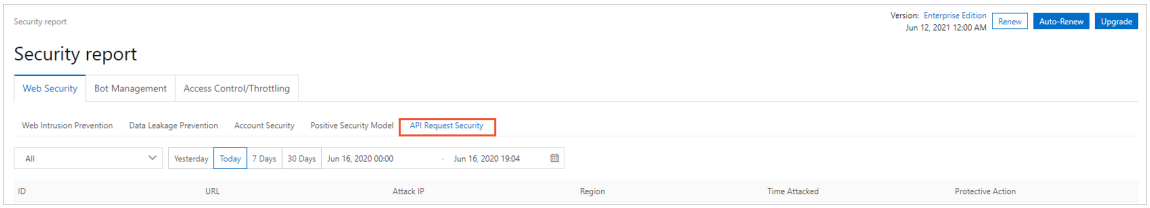
- The file size does not exceed 128 KB.
- The file must be in the Swagger 2.0-compliant XML or JSON format.

On the **API Request Security** page, you can:

- View the status of API security rules.

  After the file is imported, the status of the API rule is **Enabled** and the protection status is **Warn** by default. In this case, WAF generates an alert if an invalid request is

detected. You can view the alert information on the **API Request Security** tab on the **Security report** page.



- Modify the status.

  In the rule list, you can turn on or off the switch in the **Status** column to enable or disable the API rule. If you disable the API rule (**Disabled**), WAF no longer detects requests of this API or generates alerts.

- Modify the protection status.

  In the **Protection Status** column, you can click either **Warn** or **Block**. If you click **Block**, WAF blocks all invalid access requests to this API.

- View API information.

  In the **Operation** column, click **Details** to view WAF API information, including URL, request method, parameters, parameter values, description, and whether the parameters are required.

1.10 Fields of match conditions

You need to define the rule match conditions when you configure the whitelist and customize protection policies for Web Application Firewall (WAF). This topic describes the fields that can be used in rule match conditions and their definitions.

Notice:
This topic uses the new version of the WAF console released in January 2020. If the WAF instance was created before January 2020, see #unique_42.

What are match conditions

In the WAF console, you can customize whitelist rules, access control rules, and rate limiting policies. A custom rule consists of match conditions and actions. When you create a rule , you need to define match conditions by specifying the match fields, logical operators, and match content. You also need to select an action that will be triggered when a request matches the conditions.

Each match condition consists of a match field, logical operator, and match content. Currently, match content does not support regular expressions, but can be set to null. You can set a maximum of three match conditions in a custom rule and the logical relation between each condition must be AND. That is, only when the access request matches all the conditions at the same time, the corresponding action will be triggered.

Supported match fields

The following table lists the supported match fields in match conditions. **Advanced Field** indicates that the field is supported only by the **Business**, Enterprise, or Exclusive edition of WAF instances.

Match field	Advanced field	Supported logical operator	Description
IP	No	Belongs to/Does not belong to	The source IP address of the access request. IP addresses or CIDR blocks are supported, for example, 1.1.1.1/24.  **Note:** You can enter up to 50 IP addresses or CIDR blocks. Separate multiple IP addresses and CIDR blocks with commas (,).
URL	No	• Includes/Does not include • Equals/Does not equal	The URL of the access request.
Referer	No	• Includes/Does not include • Equals/Does not equal • Length equals/Length greater than/Length less than • Does not exist	The URL of the source page from which the access request is redirected.

Match field	Advanced field	Supported logical operator	Description
User-Agent	No	• Includes/ Does not include • Equals/Does not equal • Length equals/ Length greater than /Length less than	The browser ID, rendering engine ID, version information, and other browser -related information of the client that initiates the access request.
Params	No	• Includes/ Does not include • Equals/Does not equal • Length equals/ Length greater than /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.
Cookie	Yes	• Includes/ Does not include • Equals/Does not equal • Length equals/ Length greater than /Length less than • Does not exist	The cookie information in the access request.

Match field	Advanced field	Supported logical operator	Description
Content-Type	Yes	• Includes/ Does not include • Equals/Does not equal • Length equals/ Length greater than /Length less than	The HTTP content type (MIME) specified in the response returned to the access request.
Content-Length	Yes	Value less than /Value equals/ Value greater than	The number of bytes in the response returned to the access request.
X-Forwarded-For	Yes	• Includes/ Does not include • Equals/Does not equal • Length equals/ Length greater than /Length less than • Does not exist	The client IP address of the access request . X-Forwarded-For (XFF) is used to identify the HTTP request header field of the initial IP address of the client initiating the access request that is forwarded 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 instances.
Post-Body	Yes	• Includes/ Does not include • Equals/Does not equal	The content of the response returned to the access request.
Http-Method	Yes	Equals/Does not equal	The request method, such as GET and POST.

Match field	Advanced field	Supported logical operator	Description
Header	Yes	• Includes/ Does not include • Equals/Does not equal • Length equals/ Length greater than /Length less than • Does not exist	The header information about the access request, which is used to customize the HTTP header fields.

Logical operator descriptions

Logical operator	Description
Belongs to/Does not belong to	Whether the match field belongs to the match content.
Includes/Does not include	Whether the match field includes the match content.
Equals/Does not equal	Whether the match field equals the match content.
Length equals/Length greater than/Length less than	Whether 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/Value greater than	The value of the match field is less than, equal to, or greater than that of the match content.

Related topics

- Configure the website whitelist
- Configure the web intrusion prevention whitelist
- Configure the access control and throttling whitelist
- Configure the bot management whitelist
- Configure the data security whitelist
- Create a custom protection policy

2 Customize protection rule groups

A protection rule group contains rules that are selected from the built-in protection rule set of Web Application Firewall (WAF) to implement a specific protection feature, such as web application protection, formally known as the 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 that uses the subscription billing method is activated. The instance is one of the following instances:

- A WAF instance of the Business or Enterprise Edition in mainland China
- A WAF instance of the Enterprise Edition outside China

Context

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

Use a custom rule group

Follow these steps to use a custom rule group:

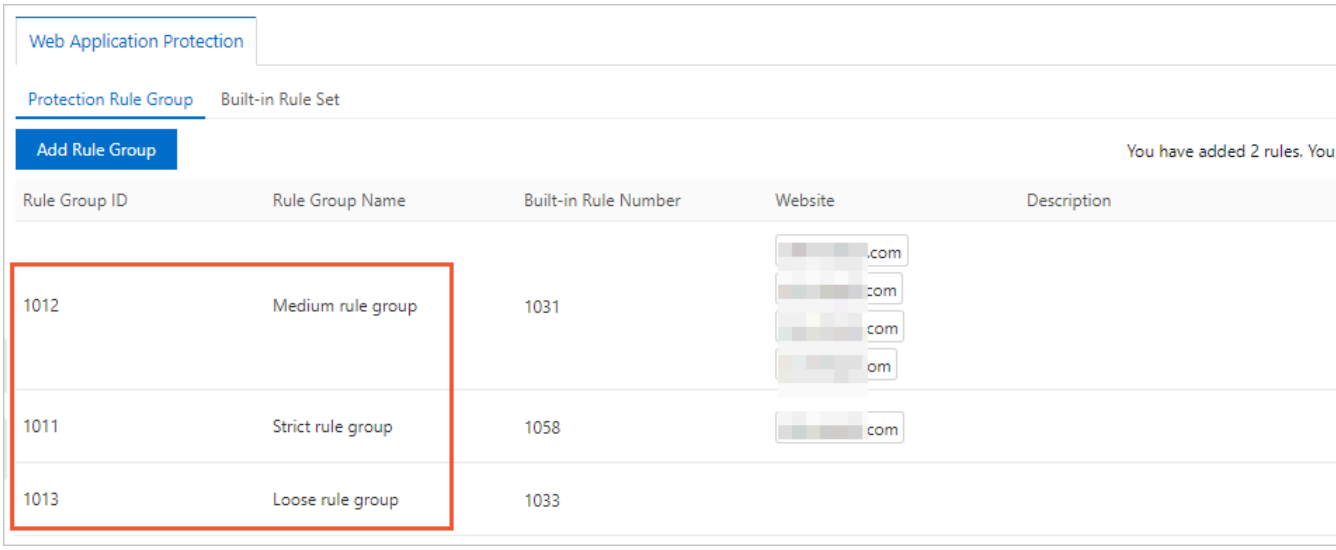
1. Create a rule group: Select rules from the built-in rule set of WAF to customize a custom rule group. The rule group provides protection policies for a specific protection feature.
2. Apply the rule group: Apply the created rule group to your website.

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 , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. Optional: On the **Protection Rule Group** page, click the tab that contains the target protection feature.

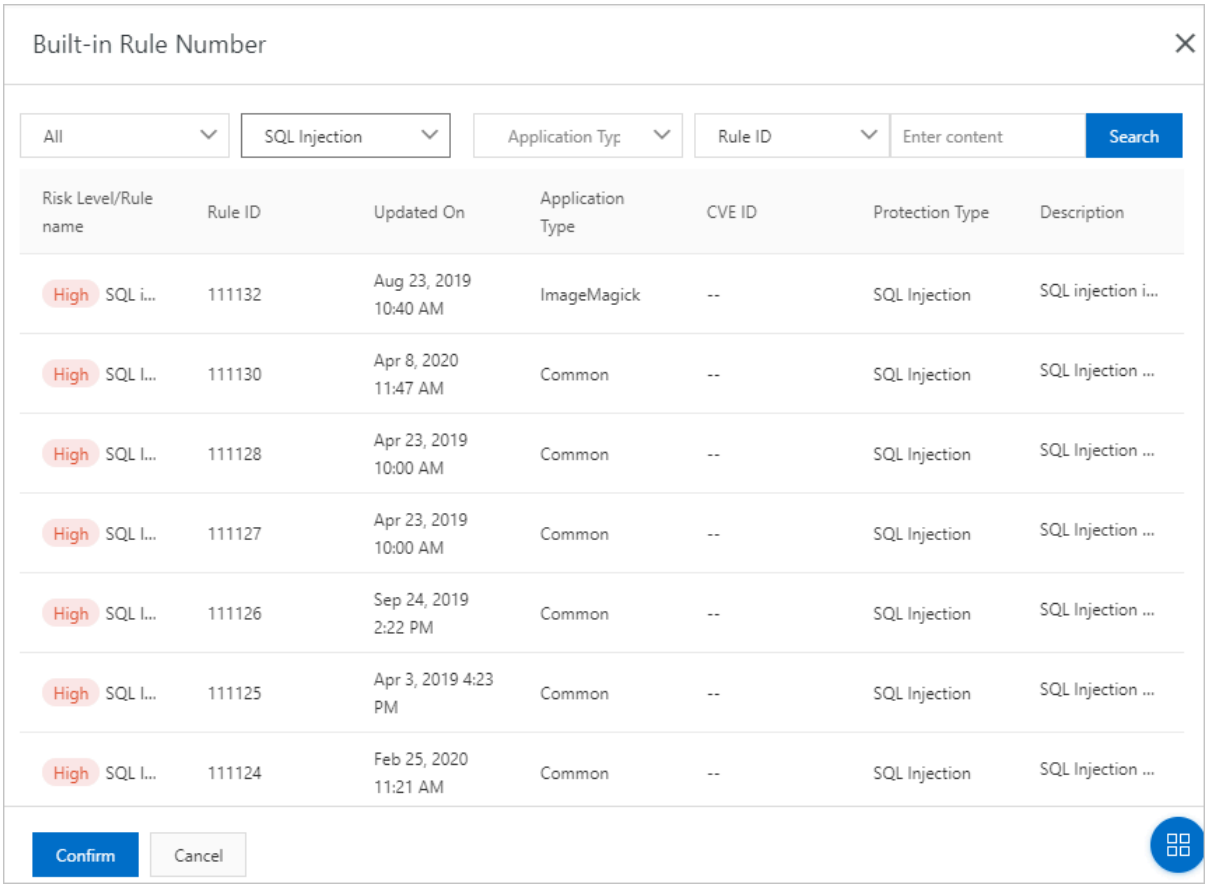
Note:

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



The **Web Application Protection** tab lists the default and custom rule groups. The rule groups **1011** (**Strict rule group**), **1012** (**Medium rule group**), and **1013** (**Loose rule group**) are default rule groups.

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

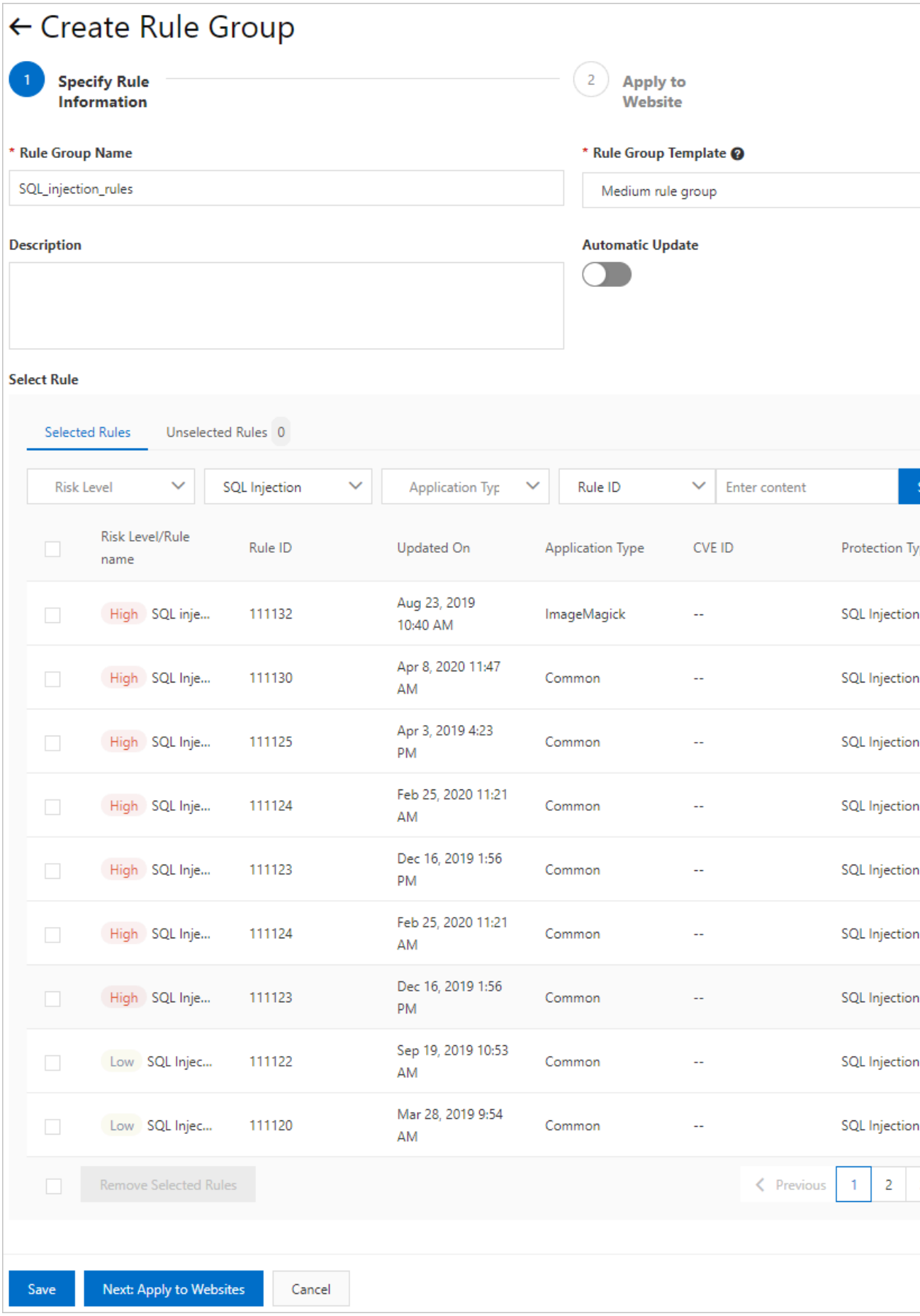


5. Click **Create Rule Group**.

> **Note:**
> You can create up to 10 rule groups for the web application protection feature.

6. Follow these steps to create a rule group:

   a) **Specify rule information**. Configure the following parameters and click **Next: Apply to Websites**.


Create Rule Group
1 Specify Rule Information
2 Apply to Website
* Rule Group Name
SQL_injection_rules
* Rule Group Template
Medium rule group
Description
Automatic Update
Select Rule
Selected Rules
Unselected Rules 0
Risk Level
SQL Injection
Application Typ
Rule ID
Enter content
Risk Level/Rule name
Rule ID
Updated On
Application Type
CVE ID
Protection Ty
High SQL inje... 111132 Aug 23, 2019 10:40 AM ImageMagick -- SQL Injection
High SQL Inje... 111130 Apr 8, 2020 11:47 AM Common -- SQL Injection
High SQL Inje... 111125 Apr 3, 2019 4:23 PM Common -- SQL Injection
High SQL Inje... 111124 Feb 25, 2020 11:21 AM Common -- SQL Injection
High SQL Inje... 111123 Dec 16, 2019 1:56 PM Common -- SQL Injection
High SQL Inje... 111124 Feb 25, 2020 11:21 AM Common -- SQL Injection
High SQL Inje... 111123 Dec 16, 2019 1:56 PM Common -- SQL Injection
Low SQL Injec... 111122 Sep 19, 2019 10:53 AM Common -- SQL Injection
Low SQL Injec... 111120 Mar 28, 2019 9:54 AM Common -- SQL Injection
Remove Selected Rules
Previous
1
2
Save
Next: Apply to Websites
Cancel

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 the rule group template from which you want to select rules for the rule group. Valid values: • **Strict rule group** • **Medium rule group** • **Loose rule group** Different rule group templates contain different rules. After you select the rule group template, you can select rules from the template.
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.

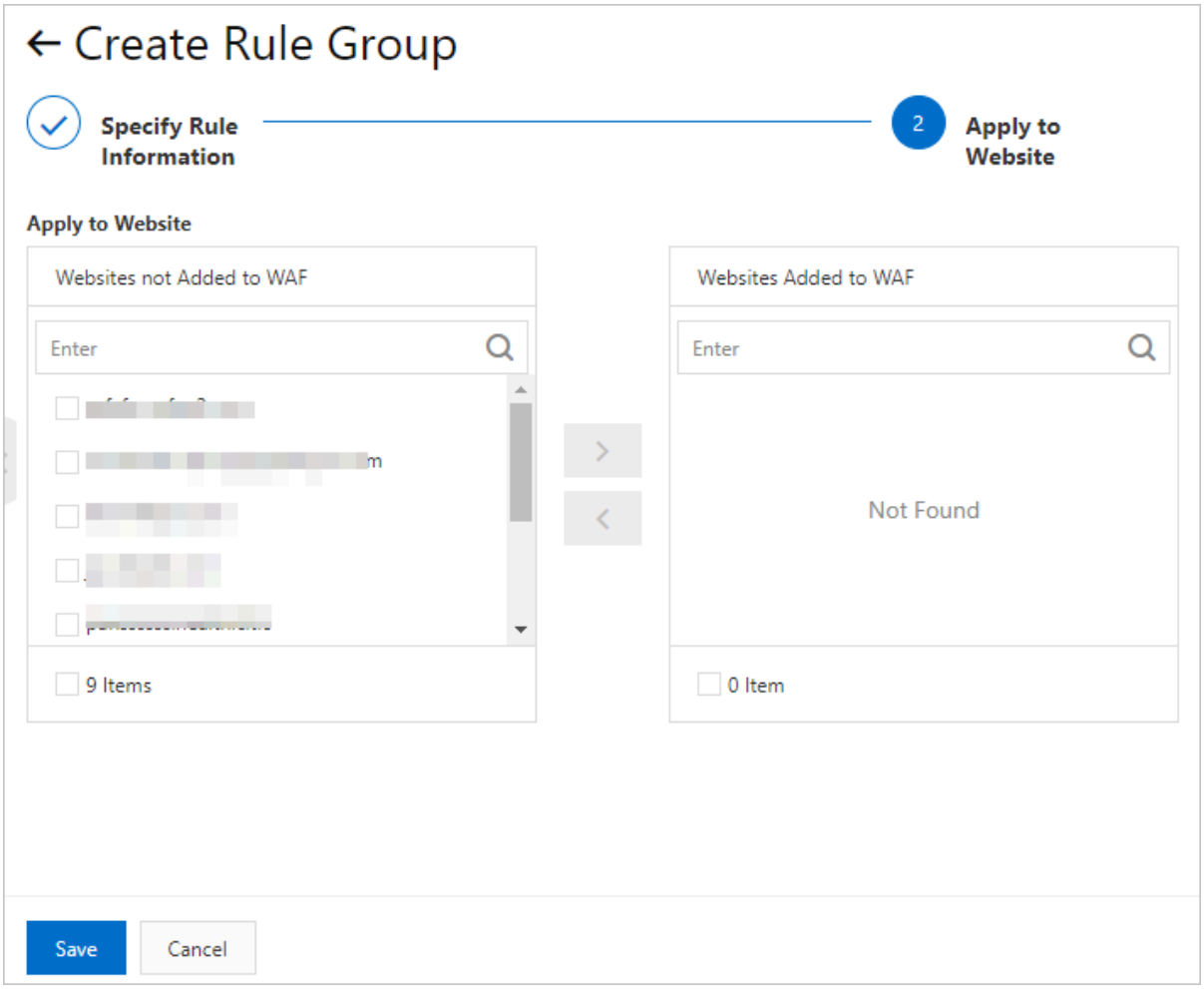
Parameter	Description
Select Rule	Select rules from the rule group template and add them to the current rule group. You can use the filter or search function to find target rules. You can filter rules by **Protection Type**, **Application Type**, or **Risk Level** or enter a rule name or ID to search for a rule. • **Risk Level**: indicates the risk level of web attacks that are defended against. Valid values: **High**, **Medium**, and **Low**. • **Protection Type**: indicates the web attack type. 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**.

b) Optional: **Apply to websites**. Select the websites to which you want to apply the new rule group from the **Websites not Added to WAF** pane and add them to the **Websites Added to WAF** pane.

Notice:

You must apply one rule group to each website.



c) Click **Save**.

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

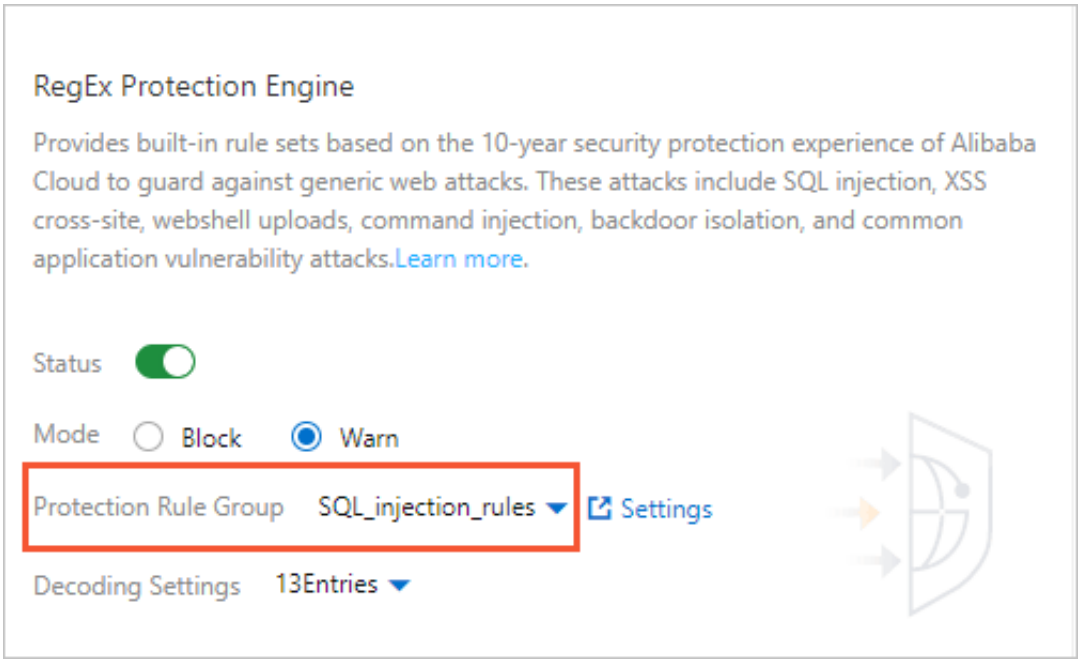
Apply the rule group

After you create a custom rule group, you can apply it in one of the following ways:

- On the **Protection Rule Group** page, apply the rule group to websites. The following steps are provided for this scenario.

- On the **Website Protection** page, select the custom rule group for a protection feature.

  For example, when you configure the web application protection feature, select the custom rule group from the **Protection Rule Group** drop-down list. For more information, see Configure the RegEx Protection Engine.

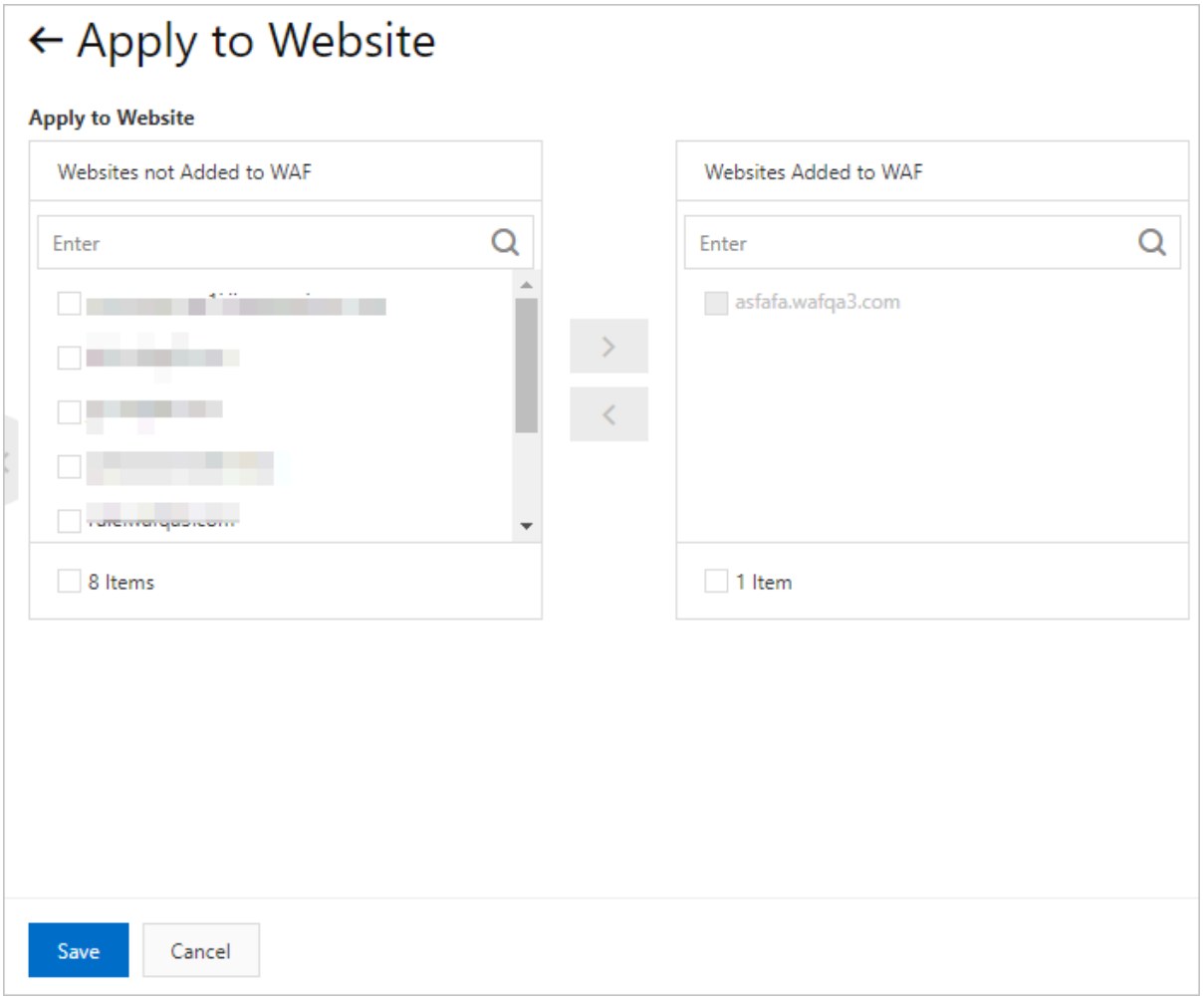


1. Log on to the Web Application Firewall console.
2. In the top navigation bar, select the resource group to which the instance belongs and the region, **Mainland China** or , in which the instance is deployed.
3. In the left-side navigation pane, choose .
4. Optional: On the **Protection Rule Group** page, click the tab that contains the target protection feature.

   Note:
   You can skip this step because only the **web application protection** feature supports custom rule groups. 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 websites to which you want to apply the rule group from the **Websites not Added to WAF** pane and add them to the **Websites Added to WAF** pane, and click **Save**.

   Notice:

You must apply one rule group to each website.



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

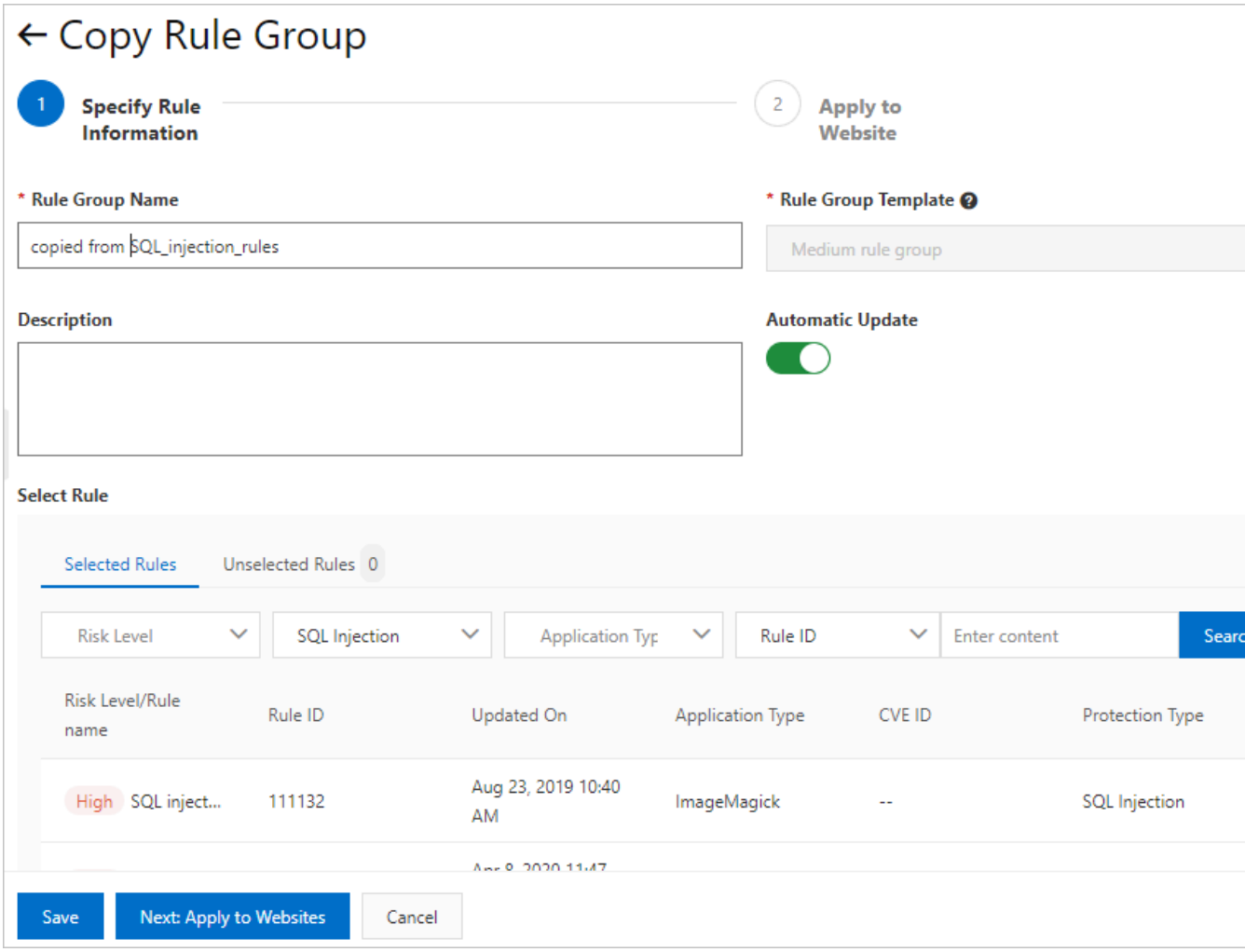
What to do next

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

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

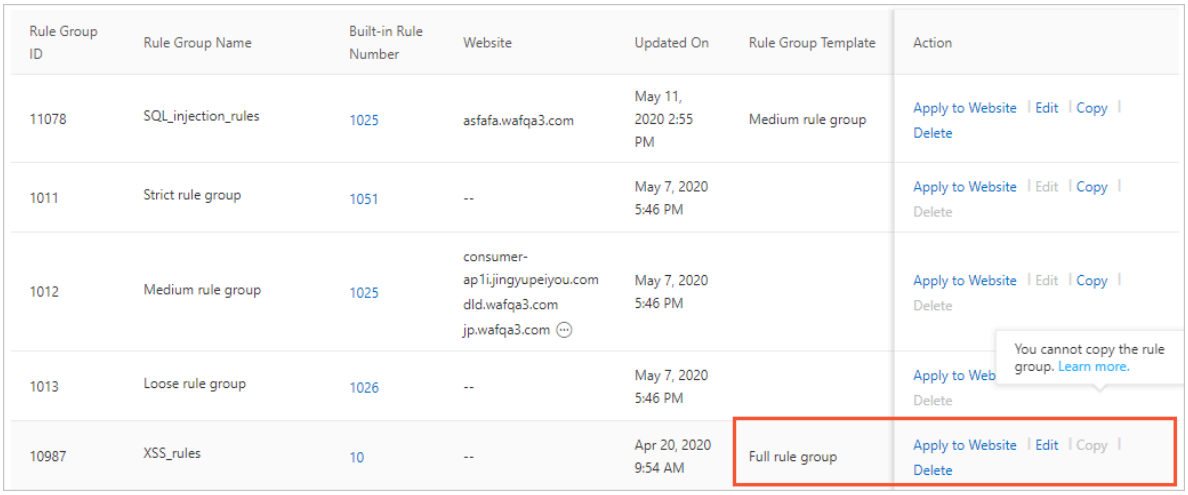
  The following figure shows the Copy Rule Group page. On this page, you can modify **Rule Group Name**, **Description**, and **Automatic Update**, but cannot modify **Rule Group**

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 of old versions cannot be copied because they do not support automatic rule update. 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.

  Note:

  Default rules cannot be edited.

- **Delete**: allows you to delete the rule group.

  Note:

  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 another rule group to the website before you delete the rule group.

3 Best practices for protection settings

3.1 Best practices for Web application protection

This topic describes the best practices for Web application protection based on WAF. The following aspects are covered: scenarios, protection policies, protection effects, and rule updates.

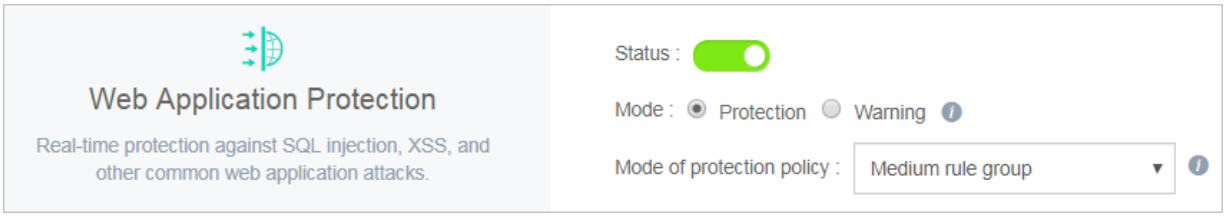
Scenarios

WAF provides protection against Web attacks, such as SQL injection, XSS, remote command execution, and webshell upload. For more information about Web attacks, see OWASP 2017 Top 10.

> **Note:**
> Server intrusions caused by security issues in host layers, such as unauthorized access to Redis and MySQL, are not covered by WAF.

Protection policies

After you add your domain to WAF, log on to the Web Application Firewall console. In the left-side navigation pane, choose **Management** > **Website Configuration**. Select your domain and click **Policies** to view the protection status of your website, as shown in the following figure:



By default, Web Application Protection is enabled and the normal mode protection is used. The parameters are as follows:

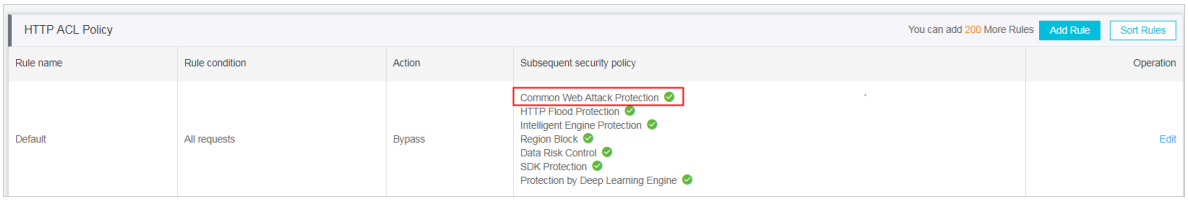
- **Status**
  - **Enabled** indicates that Web Application Protection is enabled.
  - **Disabled** indicates that Web Application Protection is disabled.

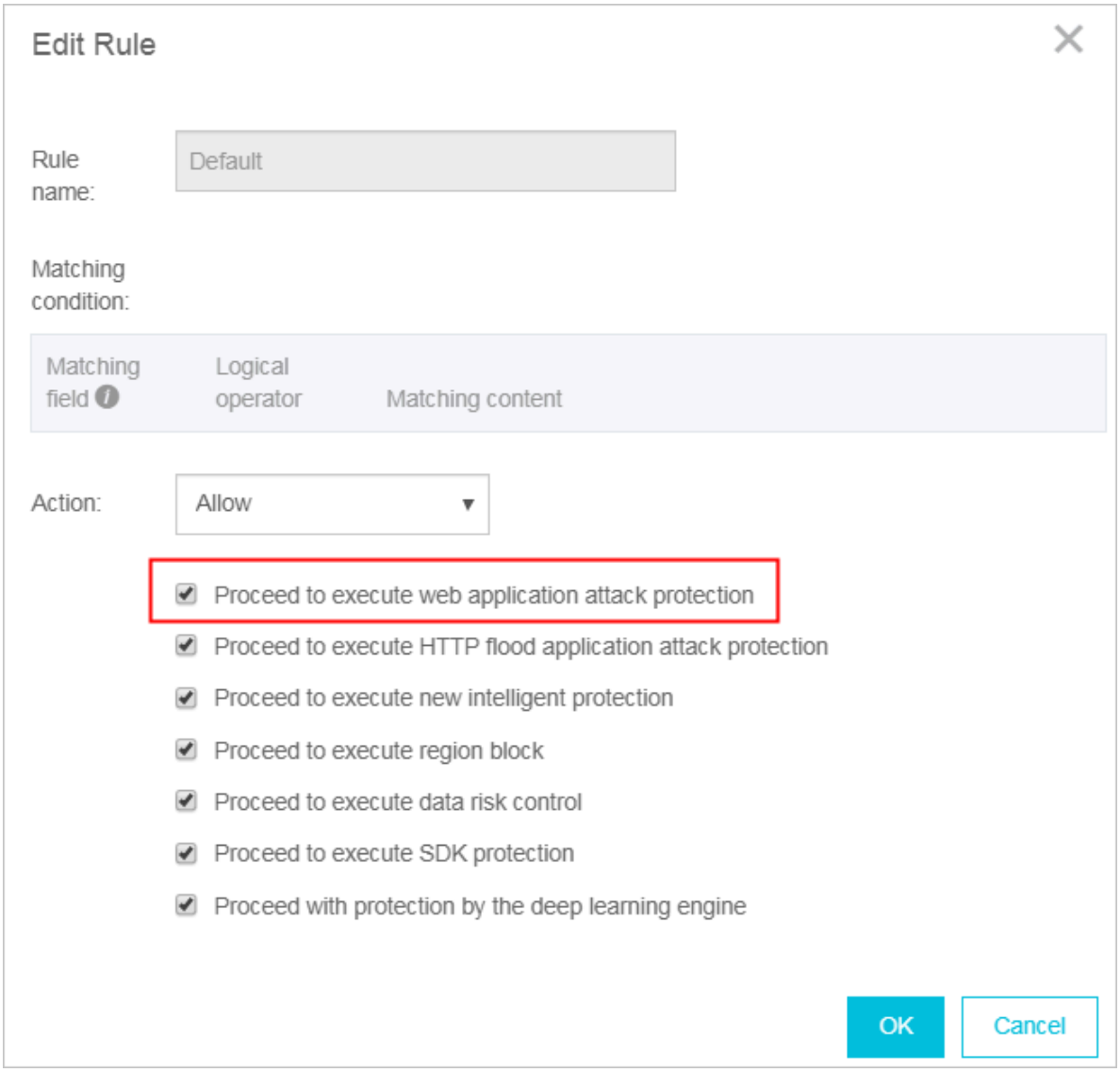
- **Mode**: Two modes are provided: Protection and Warning.
  - The **Protection** mode indicates that WAF automatically blocks malicious requests and logs attacks when the application is under attack.
  - The **Warning** mode indicates that WAF does not block malicious requests but logs attacks when the application is under attack.
- **Protection Policy**: Three protection policies are available when the Protection mode is selected: Loose, Normal, and Strict.
  - **Loose**: This policy only blocks requests that display typical attack patterns.
  - **Normal**: This policy blocks requests that display common attack patterns.
  - **Strict**: This policy blocks crafted requests that display specific types of attack patterns.

Protection tips:

- If you are not clear about your website's traffic patterns, we recommend that you use the Warning mode first. You can observe the traffic flow for one or two weeks and then analyze the attack log.
  - If you do not find any record indicating that normal requests are blocked, you can switch to the Protection mode to enable further protection.
  - If normal requests are found in the attack log, contact customer service to resolve the issue.
- If you add domains of PHPMyAdmin or tech forums to WAF, normal requests may be mistakenly blocked. We recommend that you contact customer service to resolve the issue.
- Note the following points in your operations:
  - Do not pass raw SQL statements or JavaScript code in HTTP requests.
  - Do not use special keywords, such as UPDATE and SET, to define the path in URLs, such as www.example.com/abc/update/mod.php? set=1.
  - If file uploads are required, restrict the maximum file size to 50 MB. We recommend that you use OSS or other methods to upload files exceeding the size limit.

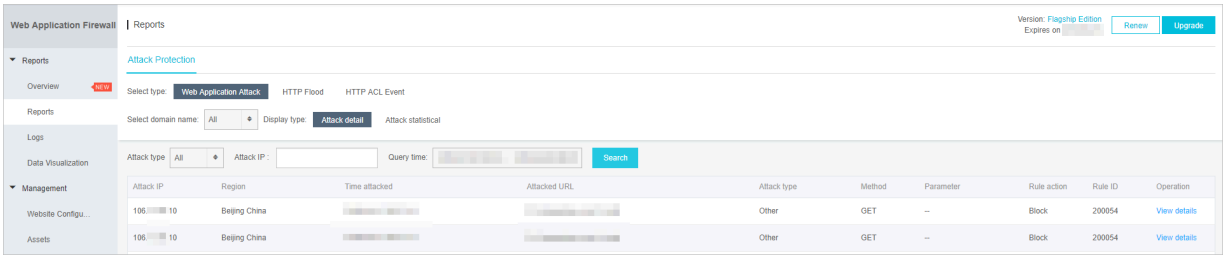
- After Web Application Protection is enabled, do not disable the All Requests option in the default rule of HTTP ACL Policy, as shown in the following figure:



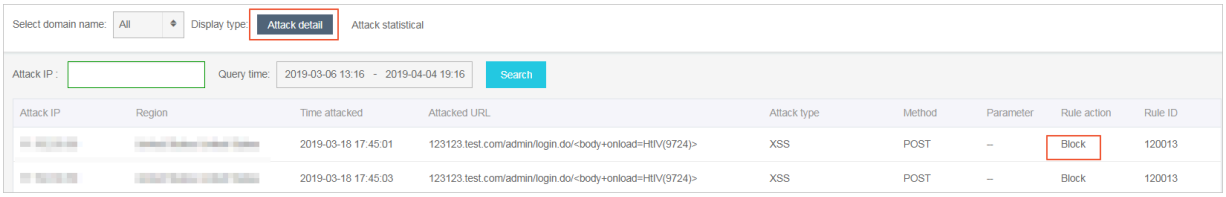


Protection effects

After Web Application Protection is enabled, you can choose **Reports** > **Reports** to view details about blocked attacks, as shown in the following figure:



On the **Reports** page, you can view attack details by time, such as yesterday, today, last seven days, or last month. You can click **View Attack Details** to view detailed attack information, as shown in the following figure:



The figure displays the details about a SQL injection attack that has been blocked by WAF.

Note:

If you find that normal requests are mistakenly blocked by WAF, we recommend that you whitelist the affected URLs in HTTP ACL policies and then contact customer service to resolve the issue.

Rule updates

When new vulnerabilities are discovered, WAF updates protection rules and releases security bulletins in a timely manner.

Log on to the Web Application Firewall console. In the left-side navigation pane, choose **Overview** > **Security** to view the latest security bulletins.

Note:

Web attacks usually have more than one proof of concept (POC). A thorough analysis is conducted to determine the cause of the vulnerability so that the protection rule can prevent all exploits of this vulnerability.

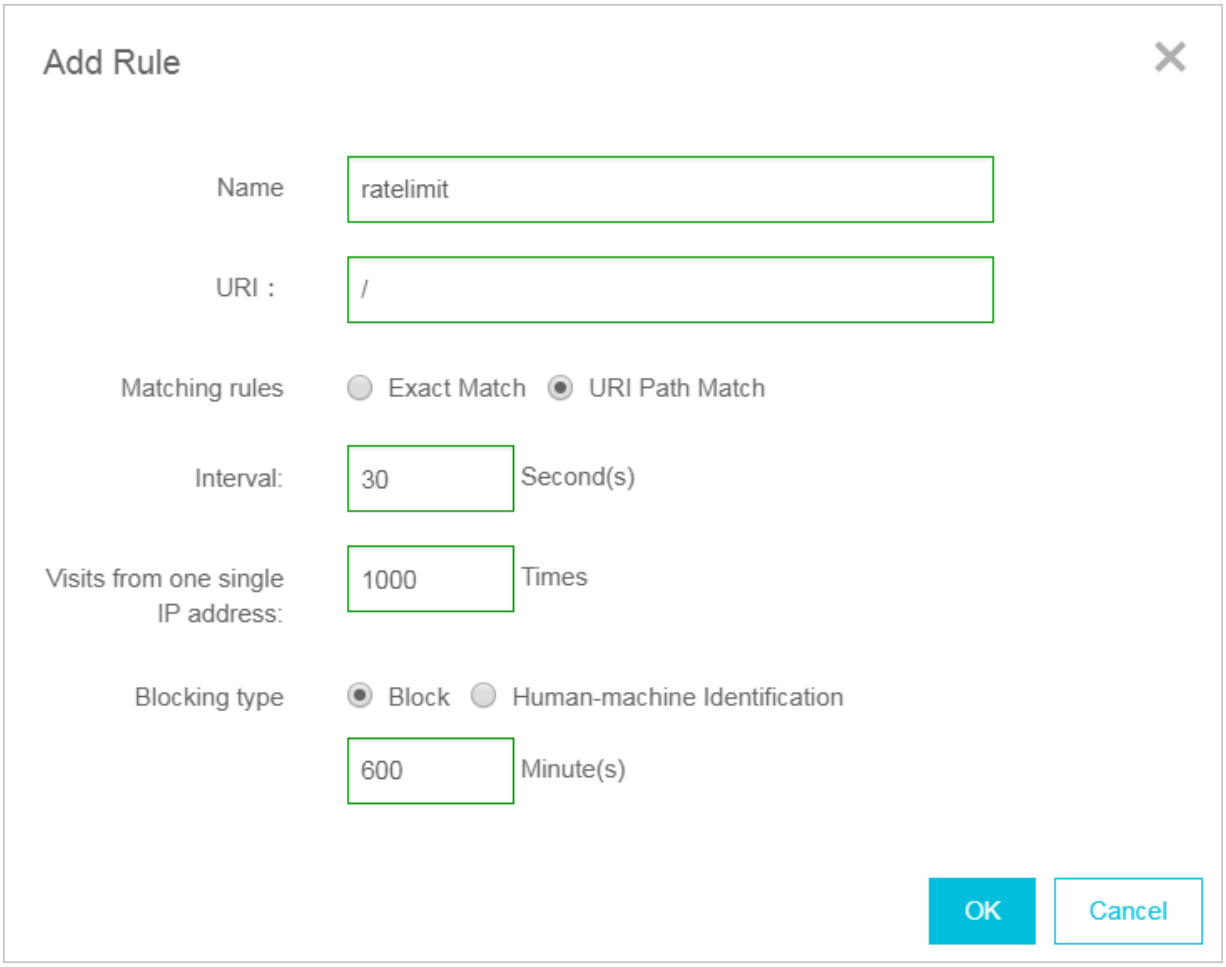
3.2 Best practices for HTTP flood protection

This topic describes common scenarios of HTTP flood attacks and introduces related protection strategies offered by Web Application Firewall (WAF). By using WAF, you can effectively protect your site from HTTP flood attacks.

Volumetric and high-frequency HTTP flood attacks

In a volumetric HTTP flood attack, a zombie server sends requests more frequently than a normal server does. In this case, the most effective measure is to restrict the request rate or forbid the requests from suspicious request sources.

You can create custom HTTP flood protection rules to restrict the request rate. The following is an example.



This rule sets **Matching rules** to URI Path Match and sets URI to a forward slash (/) to select all paths under the domain. If an IP address sends more than 1,000 requests to the domain within 30 seconds, WAF blocks this IP address for 10 hours. This rule can be used to protect small and medium-sized websites.

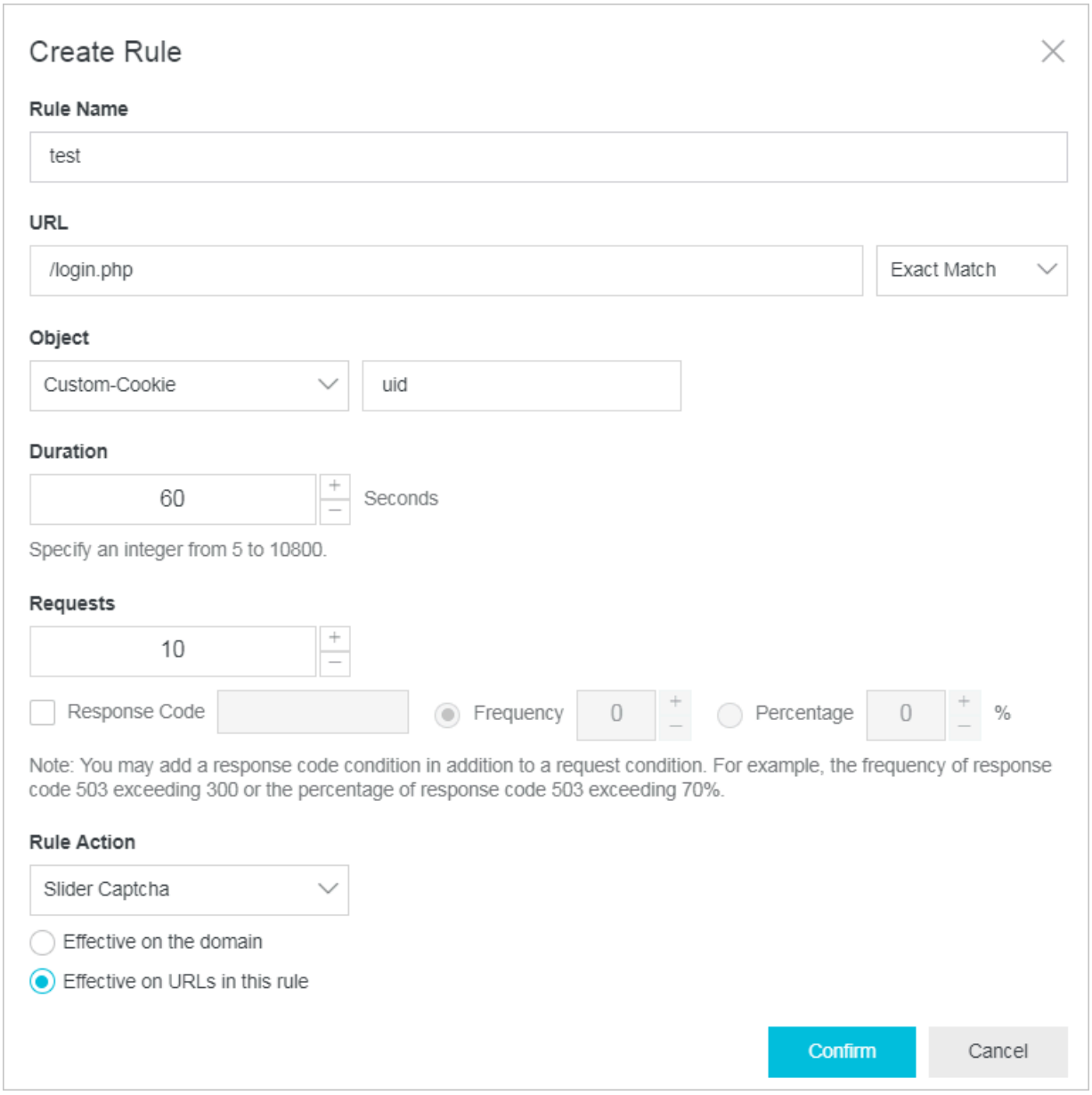
You can modify the protected paths, adjust the protection threshold, and change the blocking type based on your needs for better protection. For example, to prevent credential stuffing on the logon interface, you can set Matching rules to URI Path Match and set URI to /login.php, and block IP addresses that send more than 20 requests to access the path within 60 seconds.

Note the following points when you use HTTP flood protection:

- The **human-machine identificationblocking type** aims to verify whether requests are sent from Web browsers or automation scripts. You can use this blocking type to protect Web and HTML5 applications, but not native apps or APIs. To protect native apps and APIs, set **Blocking type** to **Block**.
- For APIs or IP addresses that may be mistakenly blocked by HTTP flood protection, you can use HTTP ACL policies to whitelist these request sources.
- Do not enable the emergency mode for native apps or APIs.

We recommend that you use **Anti-Bot Service** for more targeted protection and flexible handling methods.

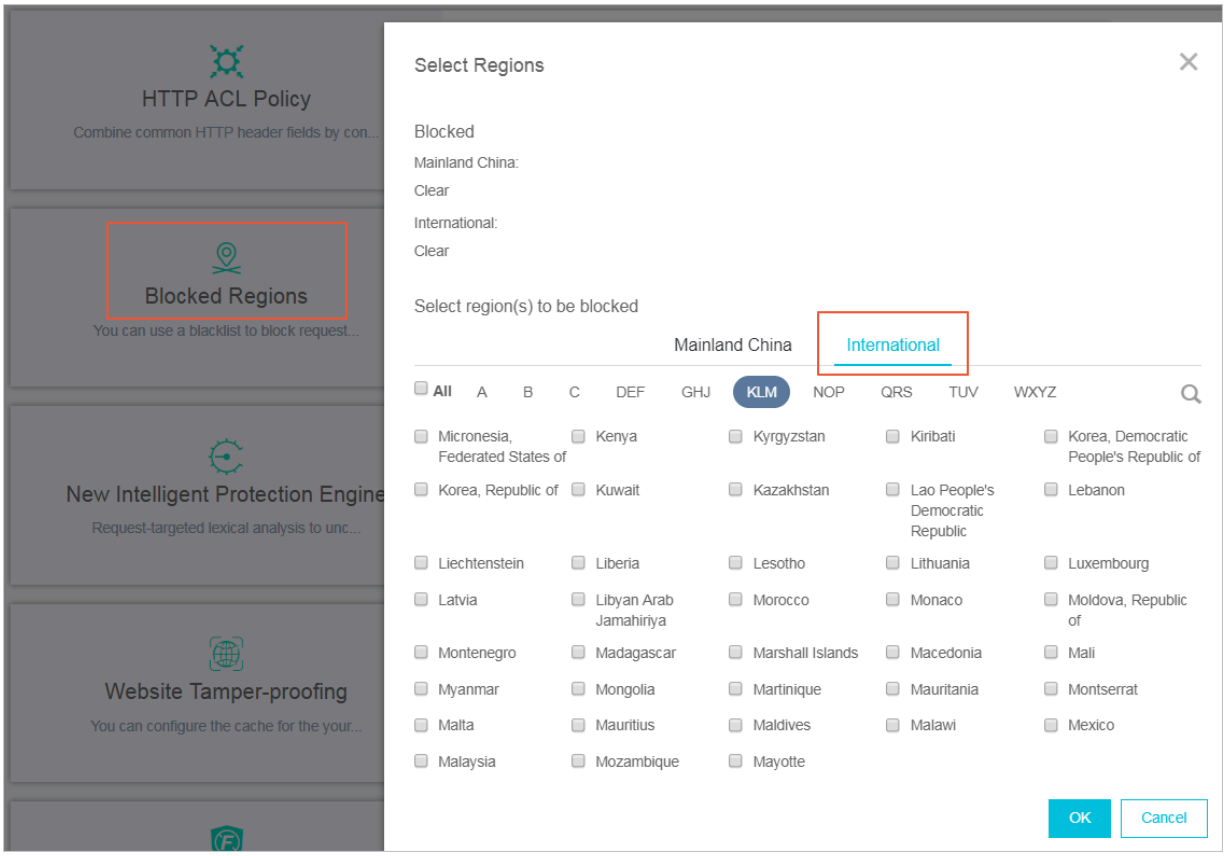
For example, blocking IP addresses may affect NAT. Anti-Bot Service allows you to use the parameters related to user information in the cookies or requests to calculate the request rate. You can also use slider CAPTCHA to verify the identity of the requester. This helps reduce false positives. In the following example, the request rate is calculated based on the cookie that is used to identify the user, and slider CAPTCHA is used to verify the user identity. Assume that the cookie format is as follows: uid=12345.



Attacks from regions outside mainland China and public clouds

A large portion of HTTP flood attacks originate from regions outside mainland China, data centers, and public clouds. If your website targets users in mainland China, you can block requests from regions outside mainland China to mitigate this attack.

WAF provides the blocked regions feature for this purpose.



If you need to block IP addresses from data centers or public clouds, such as Alibaba Cloud or Tencent Cloud, contact customer service through **DingTalk**.

Abnormal or unusual packets

Malicious requests in HTTP flood attacks are arbitrarily constructed and contain abnormal or unusual packets compared with normal requests. Most malicious requests have the following features:

- Abnormal or malformed user agent. For example, the user agent has characteristics of automation tools (such as Python), has an incorrect format (such as Mozilla///), or is impossible to be used in normal requests (such as www.baidu.com). Block the request if these features are detected.
- Unusual user agent. For example, promotional HTML5 pages targeting WeChat users are supposed to be accessed through WeChat. It is unusual if the user agent field indicates that the request is sent from a Windows desktop browser, such as MSIE 6.0. Block the request if these features are detected.
- Unusual referer. For example, the referer field does not exist or indicates the address of an illegitimate site. We recommend that you block this request. However, if the user is visiting your home page or visiting your website for the first time, the request may not

contain the referer field. If a URL can only be accessed through redirects, you can decide whether to block the URL based on the referer.

- Unusual cookies. Similar to the referer field, a normal request contains cookies that are related to the business of the requested website, unless it is the user's first visit to your site. In many situations, malicious requests in HTTP flood attacks do not contain any cookie information.
- Missing HTTP headers. For example, normal requests contain the authorization header while malicious requests do not.
- Incorrect request methods. For example, if an API has only received POST requests before but is now overwhelmed by GET requests, then you can block these GET requests.

You can analyze the features of requests and use HTTP ACL policies to block malicious requests.

Figure 3-1: Example 1: Block requests that do not contain cookies

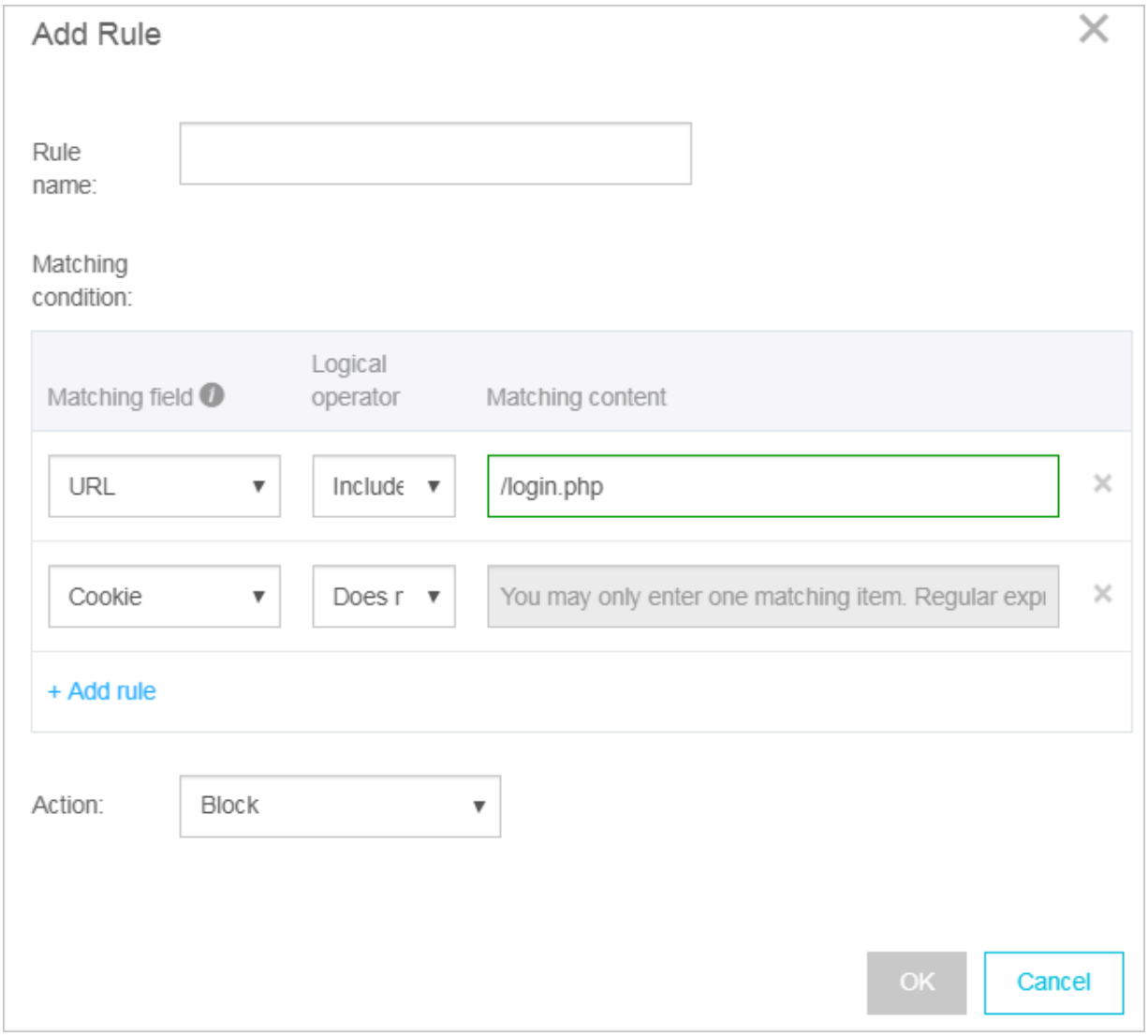
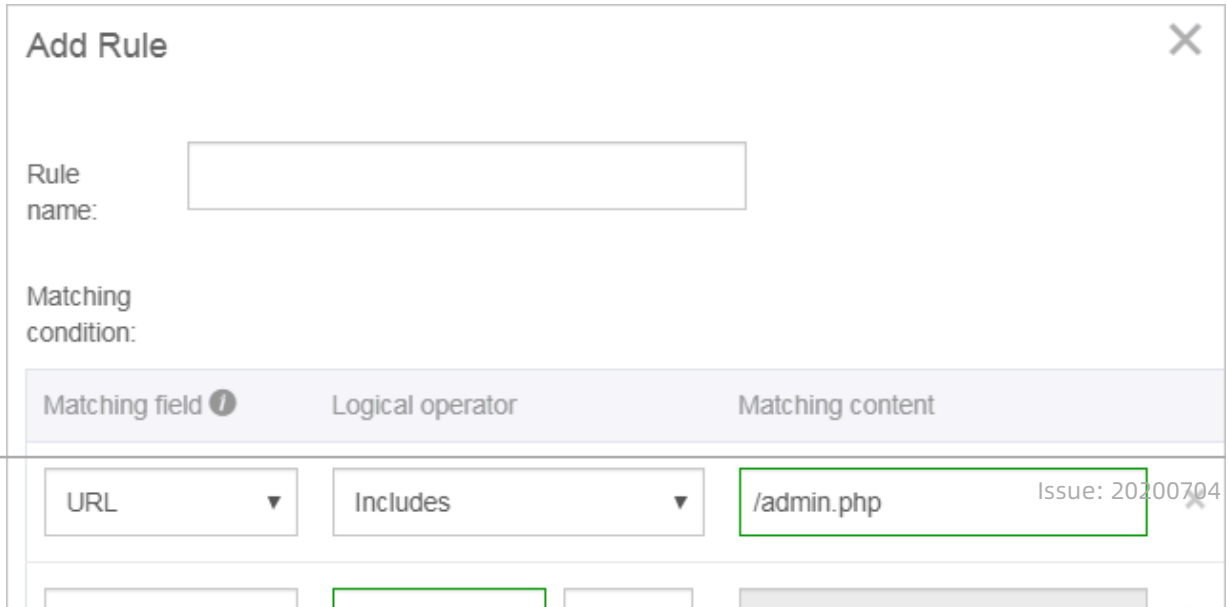


Figure 3-2: Example 2: Block requests that do not contain authorization headers



API abuse

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

Data risk control injects a JavaScript snippet into your webpage and collects informatio n about user behavior and environment variables to determine whether the request is sent from a real user or an automation script. Data risk control makes decisions based on CAPTCHA rather than the request rate or the source IP address. The feature is very effective in mitigating low-frequency attacks.

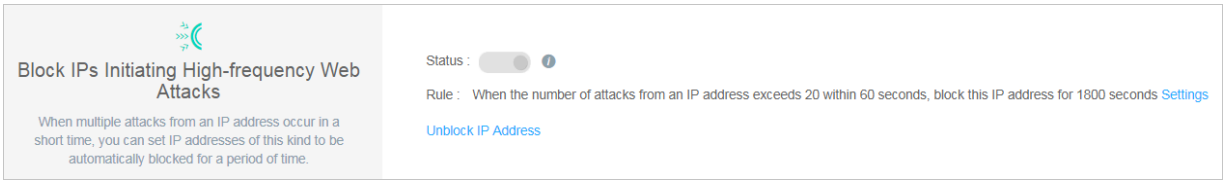
Note:

Data risk control identifies malicious requests by checking whether requests contain the authentication parameters that normal requests must contain. The service is not applicable to environments where JavaScript is not supported, such as APIs and native apps. To prevent false positives, we recommend that you test data risk control first before you enable it in the production environment. You can also use the detection mode first, and then contact customer services before you enable the prevention mode.

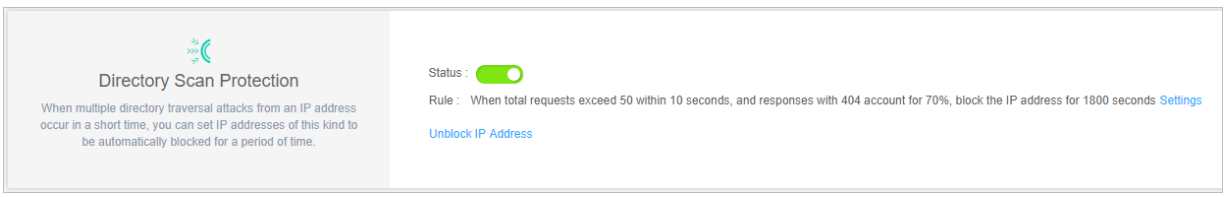
Malicious scans

A large number of malicious scans pose a serious threat to the performance of your servers. Apart from restricting scans based on frequency, you can also enhance security by using features such as IP blocking, directory scan protection, and threat intelligence.

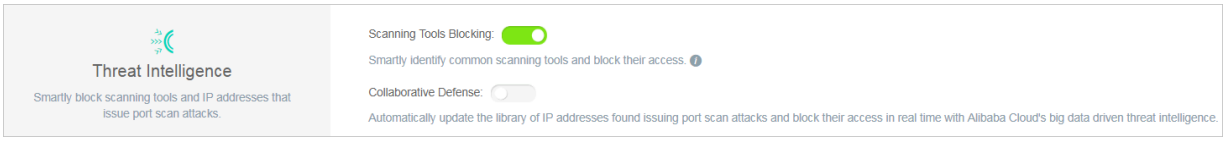
Scan requests with attack signatures are automatically blocked by WAF based on default protection rules. The malicious IP blocking feature directly blocks IP addresses that frequently trigger protection rules.



Directory scan protection automatically blocks client IP addresses that launch multiple directory traversal attacks within a short period of time.



Threat intelligence automatically blocks access requests from common vulnerability scanners or from IP addresses in the Alibaba Cloud library of identified port scan attackers.



Fake apps

To protect your business from fake apps, you can use features such as custom HTTP flood protection, region blocking, and HTTP ACL policies. You can also use the Alibaba Cloud Security SDK for enhanced 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 a request is sent from a legitimate app. Requests that do not originate from the official app are automatically blocked. This ensures that only requests from legitimate clients are served. You do not need to analyze the patterns of illegitimate requests.

To use the SDK, you must activate Anti-Bot Service. For more information, see SDK instructions.

Web crawlers

For informational websites offering services such as credit reports, apartment rentals, airline tickets, and e-book reading, Web crawlers can significantly increase the bandwidth usage and server load, and even cause data leakage. If the preceding approaches cannot prevent Web crawlers, we recommend that you use **Anti-Bot Service** for more effective protection.

3.3 Big data deep learning engine best practices

Alibaba Cloud web application firewall (WAF) provides comprehensive protection for your website by combining multiple web attack detection engines. WAF uses a combination of rule engine, semantic analysis engine, and deep learning engine to fully take the advantages of Alibaba Cloud's powerful intelligence, data analysis system, and expert vulnerability mining experience. Based on Alibaba Cloud's attack data analysis, 0day vulnerabilities are captured. Then, security experts actively mine and analyze the vulnerabilities, and finally summarize the vulnerabilities as protection rules and policies. The rules and policies of WAF are updated every week, exceeding the industry average

speed. WAF commits to providing users with the fastest and most comprehensive protection capabilities.

With the development of the Internet, web attack methods are constantly evolving. Traditional single-means protection methods cannot meet the security needs of complex Internet services. Only collaborative protection by multiple detection engines can achieve the best protection effect.

Alibaba Cloud WAF uses a combination of rule engine, semantic analysis engine, and deep learning engine to defend against web attacks.

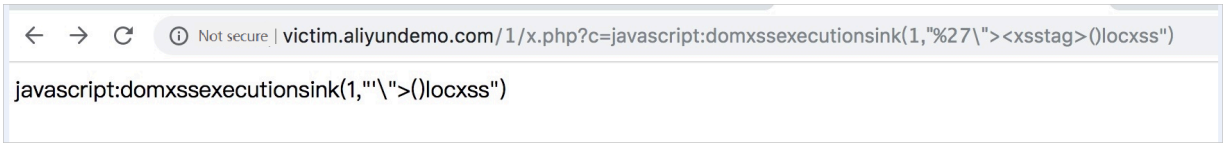
- **Rule engine**: Uses protection policy rules based on the expert experience accumulated by Alibaba Group over the years.
- **Semantic analysis engine**: Makes up the weak description of context-independent grammar features in the regular syntax field in traditional rule engine, reducing the security risks caused by false positives and vulnerabilities of rules.
- **Deep learning engine**: Conducts classification training for hundreds of millions of attack data on Alibaba Cloud every day through a neural network system built by Alibaba Cloud's powerful algorithm team through supervised learning. Finally, the model detects and intercepts unknown 0day vulnerabilities in real time, making up for other defense engines to detect unknown 0day vulnerabilities.

Deep learning engine protection practice

Generally, the rules engine uses strong descriptive rules. For requests with obvious attack features, the protection effect of regular rules is the best. However, in the face of attacks without obvious features (such as XSS feature requests), even if you enable the strict protection mode for web application attacks, it may still have potential security risks due to the inability to detect them.

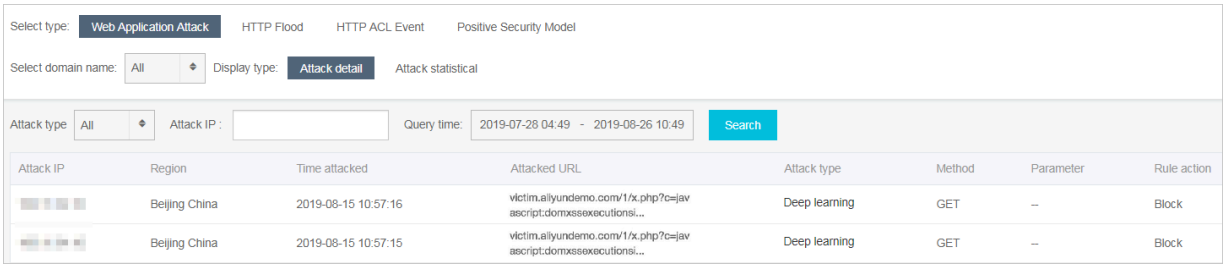
For example, you can enable the big data deep learning engine feature in WAF to identify and intercept attack requests without obvious features that cannot be identified by strict rules for web application attack protection.

In this case, the following XSS attack requests are not blocked by web application attack protection rules.



After enable the Big Data Deep Learning Engine feature, the XSS attack request that exceeds the regular rule engine's detection capability is successfully blocked.

Additionally, you can view detailed attack log information in the web application attack report. The attack type is**Deep learning**.



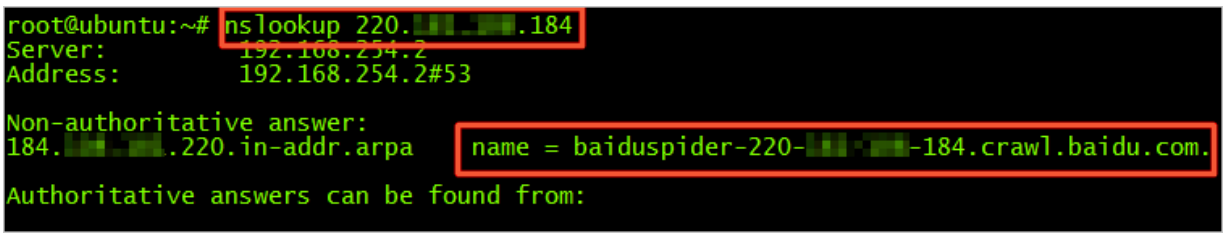
3.4 Intercept malicious crawlers

This topic explains the features of malicious crawlers and describes how to use WAF to block them.

It is noteworthy that, professional crawlers constantly change their crawling methods to bypass anti-crawling policies set by the website administrators. It is impossible to achieve perfect protection by applying fixed rules. In addition, anti-crawling has a strong associatio n with the characteristics of your own business. Therefore, you must regularly review and update the protection policies to achieve relatively ideal results.

Distinguish malicious crawlers

Normal crawlers are usually labeled with marks similar to xxspider's user-agent. They request in a regular manner, and the URLs and time are relatively scattered. If you perform an inverted nslookup or tracert on a legitimate crawler, you can always find the legitimate source address. For example, a Baidu crawler record is shown in the following figure.



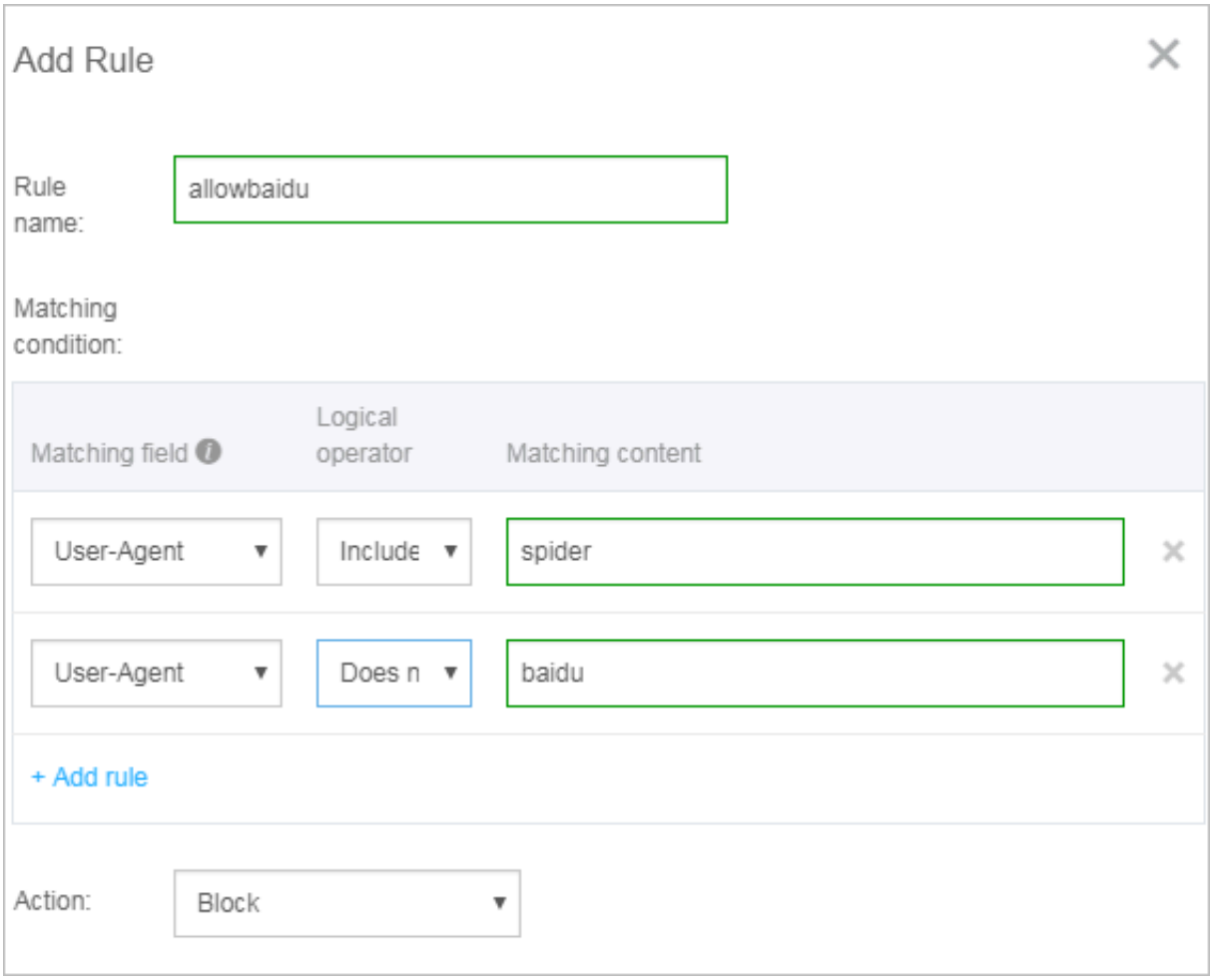
However, malicious crawlers may send a large number of requests to a specific URL/ interface of a domain name during a specific period of time. It may be an HTTP flood attack disguised as a crawler, or a crawler that crawls targeted sensitive information disguising as a third party. When the number of requests sent by a malicious crawler is large enough

, it can usually cause a sharp rise in CPU usage, failure to open the website, and service interruptions.

WAF performs Risk warning against malicious crawlers, and alerts you about yesterday's crawler requests. You can configure one or more of the following rules based on your actual business situation, to block the corresponding crawler requests.

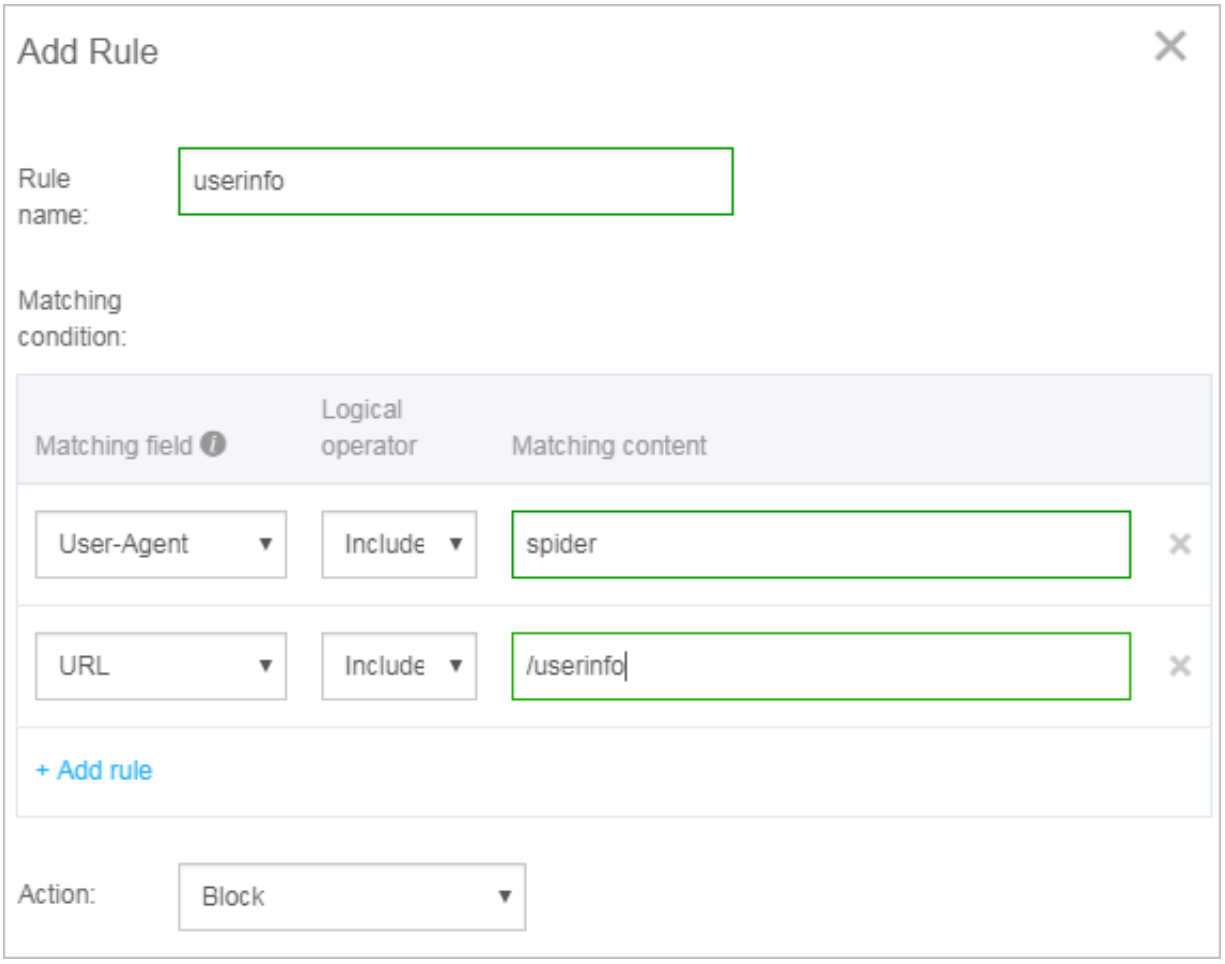
Configure HTTP ACL policy to block specific crawlers

You can configure the HTTP ACL policy to use user-agent, URL, and other keywords to filter out malicious crawler requests. For example, the following configuration only allows Baidu crawler, and filters out other crawlers (keywords are not case-sensitive).



Note:
Multiple conditions in a rule are connected by the "AND" logical relationship, that is, a request must satisfy all conditions of a rule for the rule to be effective.

You can use the following configurations to prevent all crawlers from accessing contents under the /userinfo directory.



Configure custom HTTP flood policies to block malicious requests

Using custom HTTP flood protection rules allows you to set a few specific URLs blocking rules under certain access frequency.

3.5 Account security best practices

Web Application Firewall (WAF) provides the account security feature to help you detect account risks. This topic provides suggestions on how to protect endpoints in different scenarios. You can follow the instructions in this topic to better protect endpoints where user authentication is performed.

Context

WAF supports the account security feature that detects account risks. This feature monitors endpoints related to user authentication, such as registration and logon endpoints, and detects events that may pose a threat to user credentials. Detectable risks include credential stuffing, brute-force attacks, account registration launched by bots, weak

password sniffing, and SMS interface abuse. After endpoints are added to WAF, you can view detection results in WAF security reports. For more information, see Account security.

Verification services for common and HTML5 web pages

Verification services, including SMS verification and CAPTCHA, are the easiest and most effective approaches to protect endpoints. Integrating 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, common verification methods can be easily bypassed. We recommend that you use professional verification services, for example, Alibaba Cloud CAPTCHA, to better protect endpoints against attacks.

Alibaba Cloud CAPTCHA is a suite of CAPTCHA and risk control systems that are developed based on years of defense experience of Alibaba Group. It provides a wide array of verification methods, for example, targeted verification, to verify suspicious requests and block malicious requests.

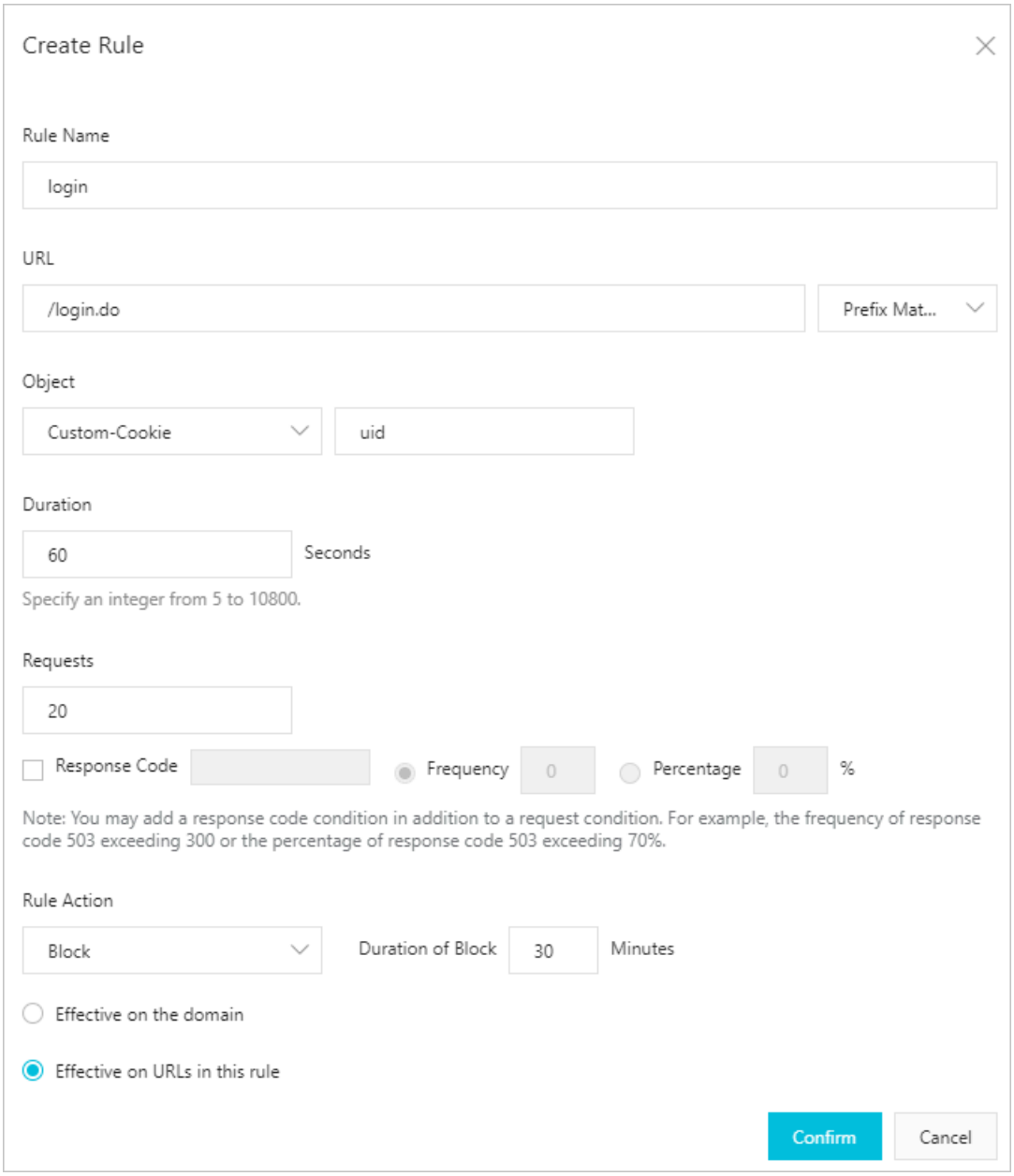
SDK signatures for native apps

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

Frequency control for blocking 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 set the maximum occurrences based on the cookies to block malicious accounts.

You can set frequency rules in Anti-Bot Service to block malicious requests. The following figure shows an example.



Analyze suspicious requests

Compared with normal requests, malicious requests typically have certain characteristics. The following examples describe common characteristics among malicious requests.

- Incomplete HTTP headers. Malicious requests may exclude certain fields, such as referer , cookie, and content-type.
- Abnormal User-Agent values. User-Agent headers typically used in requests targeting Java or Python-based websites are found in requests sent to common websites. User-Agent headers typically used in requests initiated from desktop browsers are found in requests sent to WeChat mini programs. In these cases, requests containing abnormal User-Agent headers may be malicious.

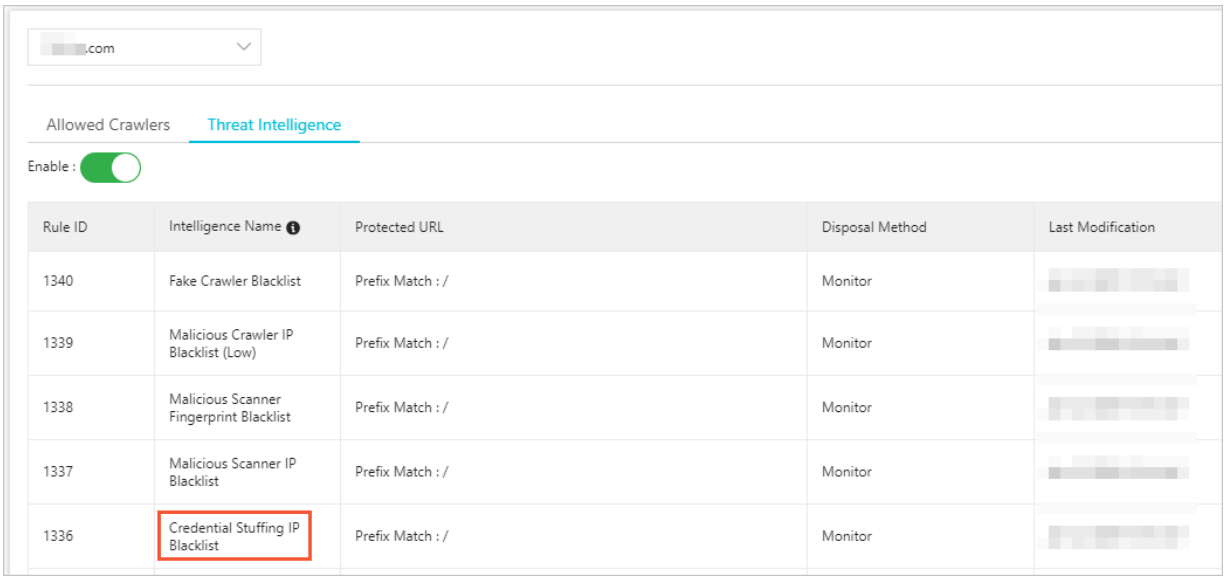
- Missing cookies. Typically, multiple cookies are used in an application. Common cookies include SessionID, userid, deviceid, and lastvisit. However, crawlers may include only one or two cookies that are required for retrieving information, and exclude other common cookies.
- Abnormal parameters. Similar 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.

Log Service integrated into both WAF and Anti-Bot Service can help you analyze request characteristics. For example, Log Service can sort out IP addresses based on the number of requests initiated from them, and calculate the proportion of requests with a certain characteristic.

For more information, see #unique_71.

Enable credential stuffing and crawler threat intelligence

Anti-Bot Service uses algorithms to identify malicious IP addresses from credential stuffing attacks detected by Alibaba Cloud. A credential stuffing IP blacklist is created and updated dynamically. You can log on to the Anti-Bot Service console and navigate to the Threat Intelligence tab to set the credential stuffing IP blacklist to the Monitor, Block, or Slider Captcha mode. For more information, see Bot intelligence.



Managed Security Service

If your business requires stronger protection, or you need help from security specialists, we recommend that you use Managed Security Service. Alibaba Cloud attack and defense specialists provide custom protection solutions based on your actual business scenarios and requirements. These solutions help you dynamically analyze, monitor, and block attacks to better safeguard your businesses.

3.6 Use custom rule groups to prevent false positives

When you find that normal requests to your site are mistakenly blocked by WAF, you can set custom rule groups to prevent this issue.

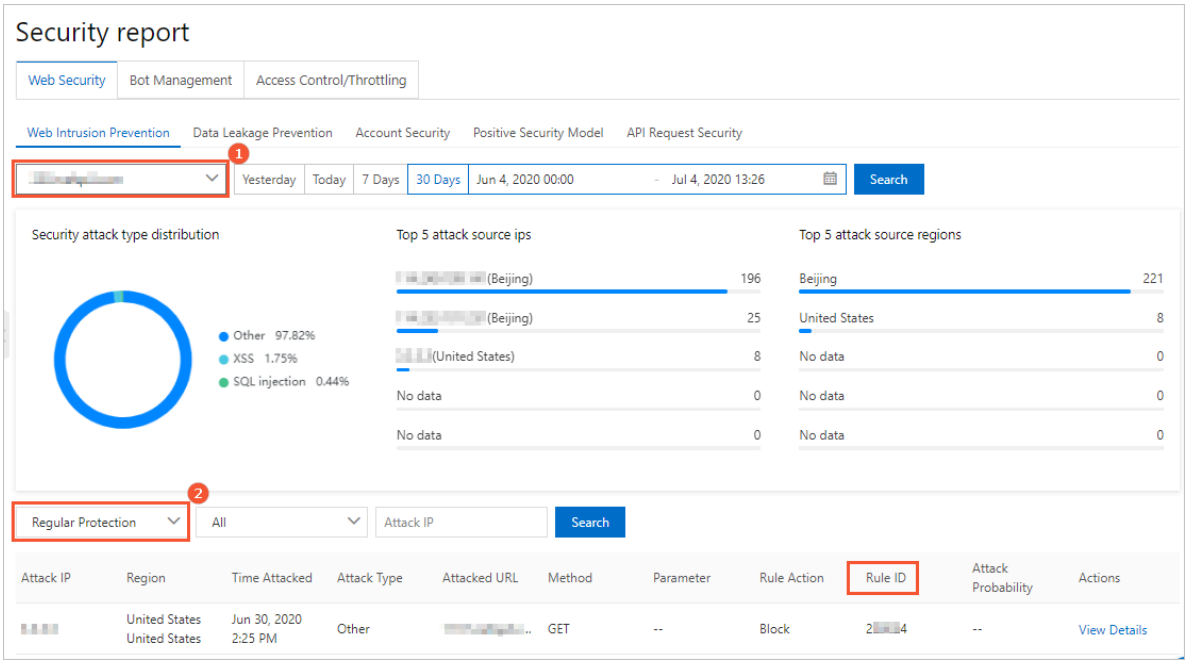
When a normal request to your site is blocked by WAF, you can identify the rule that causes the issue and create a custom rule group for the affected domain. You can then remove the specified rule to resolve the issue.

> **Note:**
> Before you remove a rule, make sure that the requests blocked according to this rule are normal requests.

Identify the ID of the protection rule that causes false positives

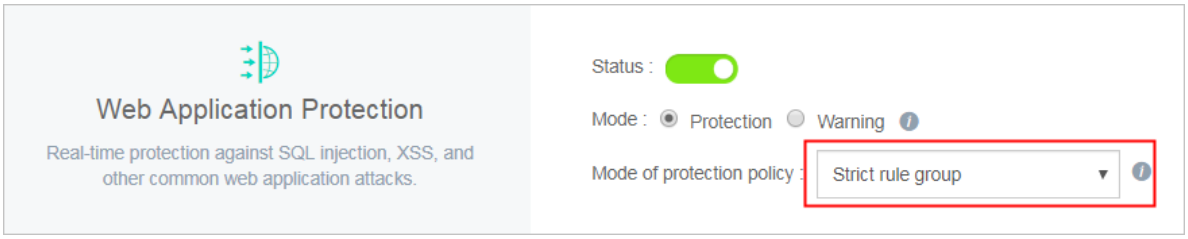
1. Log on to the Web Application Firewall console.
2. Select **Mainland China** or **International**.
3. In the left-side navigation pane, choose **Reports** > **Reports** and click **Attack Protection**.
4. Click **Web Application Attack** and select the affected domain from the drop-down list. Then click **Attack Details**.

5. You can select a time range or source IP to search for records you are interested in. The record contains the ID of the protection rule that causes false positives.



Create custom group rules for a domain

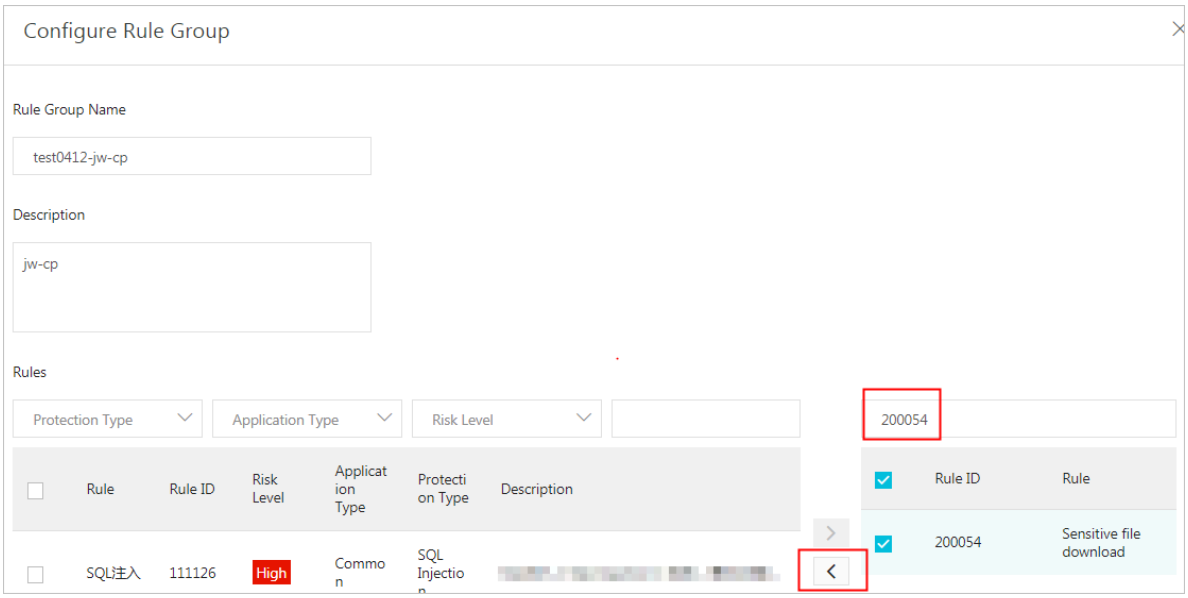
1. In the left-side navigation pane, choose **Management** > **Website Configuration**. Select the affected domain and click **Policies** to view the protection configuration of this domain.



2. In the left-side navigation pane, choose **Settings** > **Custom Rule Groups**. Select the rule group associated with the affected domain and click **Copy**.
3. Enter a rule group name and description. Click **OK** to create a custom rule group.
4. Select the newly created rule group and click **Edit**.
5. In the **Configure Rule Group** dialog box that appears, select the rule that has caused false positives from the rules list on the right side. Click  to remove this rule from the rule group and click **Confirm**.

Note:

All protection rules are listed on the left side, and rules in the specified custom rule group are listed on the right side.



6. On the **Custom Rule Groups** page, select the new rule group, click **Apply to Website**, and select the affected domain.

After the custom rule group is changed for the affected domain, the same requests sent to your domain are no longer blocked.

Note:

If requests are still blocked, make sure you have identified the right ID of the protection rule that causes false positives and remove this rule from the custom rule group.