ALIBABA CLOUD

Alibaba Cloud

Anti-Bot Service
User Guide

Document Version: 20210416

Alibaba Cloud

Legal disclaimer

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

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

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

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

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

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

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

Document conventions

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

Table of Contents

1.Access configuration 05
1.1. Deploy both Anti-Bot and Anti-DDoS Pro 05
1.2. Deploy both Anti-Bot and CDN 06
1.3. Retrieve the actual IP addresses of access users 07
1.4. Configure protection for the origin server 12
2.Protection Settings 15
2.1. Overview 15
2.2. Blacklist and whitelist 16
2.3. Access control list 17
2.4. Rate limiting 21
2.5. Bot intelligence 25
3.App protection 31
3.1. Solution overview 31
3.2. iOS SDK integration guide 32
3.3. Android SDK integration guide 37
3.4. Configure SDK protection 45
4.Real-time log query and analysis 50
4.1. Enable Log Service for Anti-Bot 50
4.2. Common statements used for log query and analytics 51
4.3. Log field description 53

1.Access configuration

1.1. Deploy both Anti-Bot and Anti-DDoS Pro

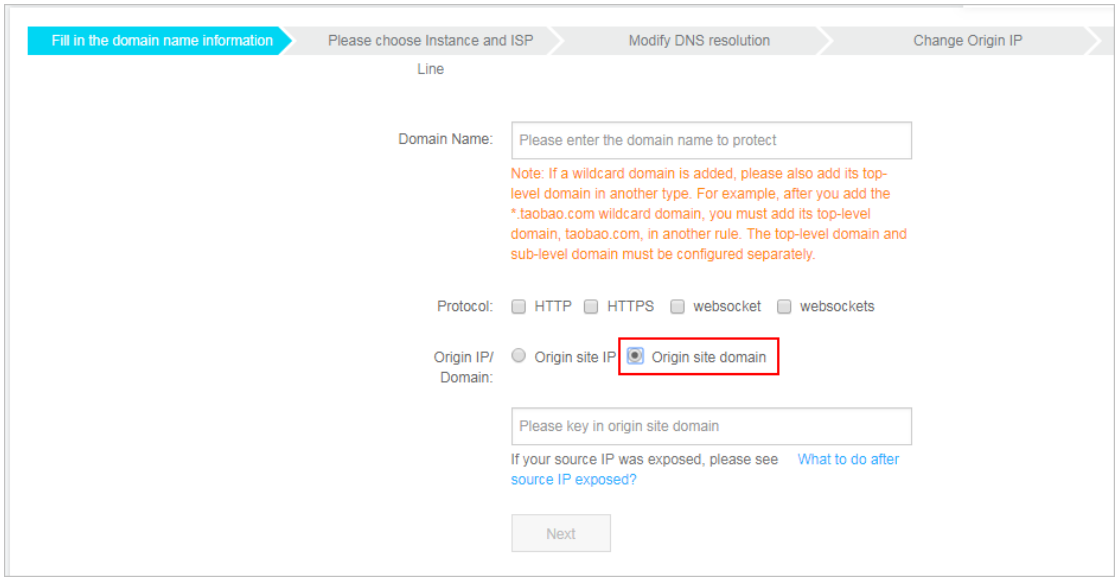
Anti-Bot Service (Anti-Bot) is fully compatible with Anti-DDoS Pro. You can deploy both Anti-Bot and Anti-DDoS Pro for your origin server in this schema: Anti-DDoS Pro (ingress to DDoS protection) > Anti-Bot (intermediate layer for application-layer protection) > origin server.

Procedure

1. Add website configuration in the Anti-Bot console.
   - **Server Address**: Select **IP** and enter the public IP address of an SLB instance, the public IP address of an ECS instance, or the IP address of a server in an IDC.
   - **Layer-7 gateways are in use, such as Alibaba Cloud Anti-DDoS Pro or CDN**: Select **Yes**.

   For more information, see Add domain name configuration.
2. Add website configuration in the Anti-DDoS Pro console. Procedure:
   i. Select **Access > Website**. Click **Add Domain**.
   ii. In the **Specify Domain Information** task, configure the following settings:
   - **Domain**: Enter the domain name of the website you want to protect.
   - **Protocol**: Select the protocol supported by the origin server.
   - **Origin Site IP/Domain**: Select **Origin Site Domain**, and enter the CNAME address generated by Anti-Bot.

   > **Note** For how to check the CNAME address generated by Anti-Bot, see Check the CNAME address allocated by Anti-Bot.



   iii. Click **Next**.

iv. Select an **instance** for the task.

3. Modify the DNS resolution of the domain name. Log on to the DNS system. Add a CNAME record to direct the resolved address of the website domain name to the CNAME address generated by Anti-DDoS Pro.

   For more information, see Configure a CNAME in Anti-DDoS Pro.

Result

After the preceding configuration, website traffic passes through Anti-DDoS Pro and then is forwarded to Anti-Bot for protection.

1.2. Deploy both Anti-Bot and CDN

Anti-Bot Service (Anti-Bot) can be deployed with CDN (such as Wangsu, Jiasule, Qiniu, Youpai, and Alibaba Cloud CDN) to protect CDN-enabled content against malicious bot traffic. You can deploy both Anti-Bot and CDN for your origin server in this schema: CDN (ingress to content acceleration) > Anti-Bot (intermediate layer for application-layer protection) > origin server.

Use Alibaba Cloud CDN

Here, Alibaba Cloud CDN is used as an example. Perform the following steps to deploy both Anti-Bot and CDN for your website:

1. Configure CDN for the (accelerating) domain name to be protected based on CDN quick start.
2. Create website configuration in the Anti-Bot console.
   - **Domain**: Enter the domain name that you want to protect.
   - **Server Address**: Enter the public IP address of an SLB instance, the public IP address of an ECS instance, or the IP address of a server in an IDC.
   - **Layer-7 gateways are in use, such as Alibaba Cloud Anti-DDoS Pro or CDN**: Select **Yes**.

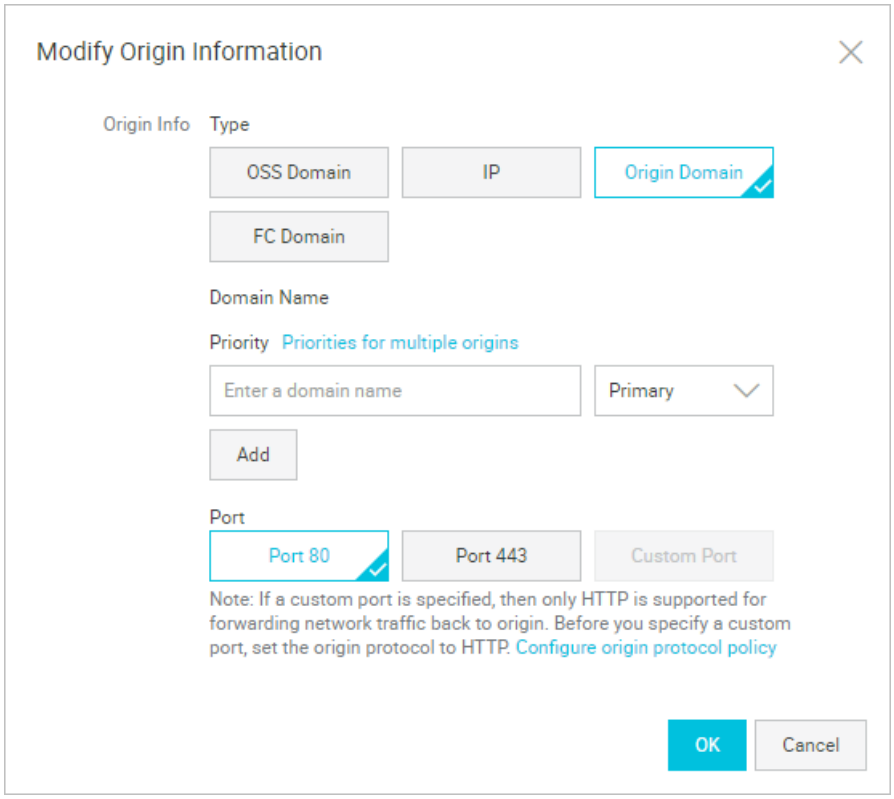
   For more information, see Add domain name configuration.
3. After the website configuration is created, Anti-Bot generates a dedicated CNAME address for the domain name.

   > **Note** For how to check the CNAME address generated by Anti-Bot, see Check the CNAME address allocated by Anti-Bot.

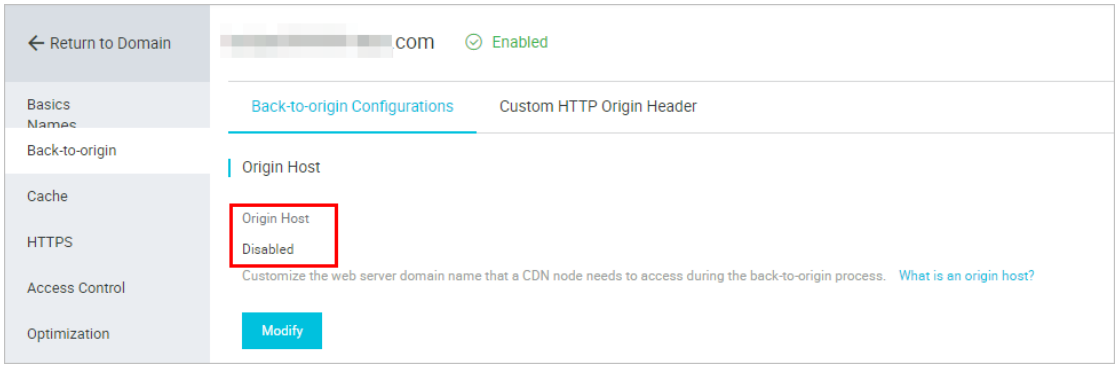
4. Perform the following step to change the origin server address in CDN configuration to the CNAME address allocated by Anti-Bot:
   i. Log on to the Alibaba Cloud CDN console.
   ii. On the **Domain Configuration** page, select the target domain name and click **Configure**.
   iii. Click **Edit** Origin IP under **Origin Edit**.

iv. Modify the origin server information.

- **Type**: Select **Origin Site Domain**.
- **Origin Site Domain**: Enter the CNAME address generated by Anti-Bot.
- **Use the same protocol as the back-to-origin protocol**: Enable this option.



v. Confirm that **Back-to-Origin Host** is disabled under **Back-to-Origin Configuration**.



After the preceding configuration, traffic passes through CDN, and dynamic content is detected and protected by Anti-Bot.

1.3. Retrieve the actual IP addresses of access users

In most service scenarios, access requests to a website are not directly sent from access users to the website's origin server. Instead, the access requests may pass through intermediate proxy servers, such as CDN, Anti-DDoS Service Pro, WAF, and Anti-Bot. For example, a website may be deployed in this schema: user > CDN, Anti-DDoS Service Pro, or Anti-Bot > origin server. After an access request is forwarded through multiple layers of acceleration or proxy, how does the origin server retrieve the actual client IP address that initiates the request?

In normal cases, before forwarding a user's access request to the next-hop server, the transparent proxy server adds an **X-Forwarded-For** record to the HTTP request header to record the user's actual IP address. The record format is `X-Forwarded-For:user IP address`. If the access request passes through multiple intermediate proxy servers, **X-Forwarded-For** records the user's actual IP address and the intermediate proxy servers' IP addresses in the following format: `X-Forwarded-For:user's IP address, proxy server 1-IP address, proxy server 2-IP address, proxy server 3-IP address, ···`.

> **Note** Anti-Bot and WAF adopt the same forwarding configuration and device. If your website domain name is configured with WAF and Anti-Bot to implement application-layer protection and intercept bot traffic, **X-Forwarded-For** records the IP addresses of only one proxy server.

Therefore, common application servers can **use X-Forwarded-For to retrieve access users' actual IP addresses.**

You can select a suitable X-Forwarded-For configuration scheme to retrieve access users' actual IP addresses based on your application server.

> **Notice** Before configuration, be sure to back up the existing environment, including the ECS instance snapshot and the configuration file of the web application server.

Nginx configuration scheme

1. Ensure that the http_realip_module module is installed.

To implement load balancing, Nginx uses **http_realip_module** for retrieving actual IP addresses.

Run `# nginx -V | grep http_realip_module` to check whether the module is installed. If the module is not installed, recompile Nginx and load the module.

> **Note** In normal cases, the module is not installed by default if Nginx was installed by using a one-click installation package.

Install **http_realip_module** by using the following method:

```
wget http://nginx.org/download/nginx-1.12.2.tar.gz
tar zxvf nginx-1.12.2.tar.gz
cd nginx-1.12.2
./configure --user=www --group=www --prefix=/alidata/server/nginx --with-http_stub_status_module --wit
hout-http-cache --with-http_ssl_module --with-http_realip_module
make
make install
kill -USR2 `cat /alidata/server/nginx/logs/nginx.pid`
kill -QUIT `cat /alidata/server/nginx/logs/ nginx.pid.oldbin`
```

2. Modify the configuration of the server for Nginx.

Open the `default.conf` configuration file and add the following content in `location /{}` :

> **Note**
>
> `ip_range1, 2, ..., x` indicates the origin CIDR block of Anti-Bot. The IP addresses must be added one by one.

```
set_real_ip_from ip_range1;
set_real_ip_from ip_range2;
...
set_real_ip_from ip_rangex;
real_ip_header   X-Forwarded-For;
```

3. Modify the log format (log_format).

log_format is typically located in the HTTP configuration of the `nginx.conf` configuration file. In *log_format*, replace the *remote-address* field with the *x-forwarded-for* field. That is, modify *log_format* as follows:

```
log_format main '$http_x_forwarded_for - $remote_user [$time_local] "$request" ' '$status $body_bytes_
sent "$http_referer" ' '"$http_user_agent" ';
```

After the preceding operation, run `nginx -s reload` to restart Nginx. After the configuration takes effect, the Nginx server can record access users' actual IP addresses by using X-Forwarded-For.

IIS 6 configuration scheme

You can install the F5XForwardedFor.dll plug-in to retrieve access users' actual IP addresses from the access log recorded by the IIS 6 server.

1. Based on the operating system version of your server, copy the *F5XForwardedFor.dll* file from the *x86\Release* or *x64\Release* directory to the specified directory, such as *C:\ISAPIFilters*. Ensure that the IIS process has read and write permissions on the directory.
2. Open IIS Manager, locate the currently activated website, right-click it, and choose **Attributes**.
3. On the Attributes page, switch to **ISAPI Filters** and click **Add**.
4. In the Add window, set the following parameters and click **Add**.
   - **Filter Name**: F5XForwardedFor
   - **Executable**: Full path of F5XForwardedFor.dll, for example, *C:\ISAPIFilters\F5XForwardedFor.dll*
5. Restart the IIS server and wait for the configuration to take effect.

IIS 7 configuration scheme

You can install the F5XForwardedFor module to retrieve access users' actual IP addresses.

1. Based on the operating system version of the server, copy the *F5XFFHttpModule.dll* and *F5XFFHttpModule.ini* files from the *x86\Release* or *x64\Release* directory to the specified directory, such as *C:\x_forwarded_for\x86* or *C:\x_forwarded_for\x64*. Ensure that the IIS process has read and write permissions on the directory.
2. In **IIS Server**, double-click **Module**.

3. Click **Configure Local Module**.
4. In the **Configure Local Module** dialog box, click **Register** to register the downloaded DLL file.
   - Register the x_forwarded_for_x86 module
     - **Name**: x_forwarded_for_x86
     - **Path**: *C:\x_forwarded_for\x86\F5XFFHttpModule.dll*
   - Register the x_forwarded_for_x64 module
     - **Name**: x_forwarded_for_x64
     - **Path**: *C:\x_forwarded_for\x64\F5XFFHttpModule.dll*
5. After registration, select the newly registered modules x_forwarded_for_x86 and x_forwarded_for_x64, and click **OK**.
6. In **API and CGI Restrictions**, add the registered DLL file, and change **Restriction** to **Allow**.
7. Restart the IIS server and wait for the configuration to take effect.

Apache configuration scheme

For Windows operating systems

The installation packages of Apache 2.4 and later provide the remoteip_module module file (mod_remoteip.so). You can retrieve access users' actual IP addresses by using this module.

1. Create a configuration file named httpd-remoteip.conf in the extra configuration folder (*conf/extra/*) of Apache.

   > **Note** Load the related configuration by introducing the remoteip.conf configuration file. This reduces the number of times of direct modification of the httpd.conf file, and avoids service exceptions due to misoperation.

2. In the httpd-remoteip.conf configuration file, add the following rule of retrieving access users' actual IP addresses.

```
# Load the mod_remoteip.so module
LoadModule remoteip_module modules/mod_remoteip.so
# Set the RemoteIPHeader header
RemoteIPHeader X-Forwarded-For
# Set the origin CIDR block
RemoteIPInternalProxy 112.124.159.0/24 118.178.15.0/24 120.27.173.0/24 203.107.20.0/24 203.107.21.0/24 203.107.22.0/24 203.107.23.0/24 47.97.128.0/24 47.97.129.0/24 47.97.130.0/24 47.97.131.0/24
```

3. Modify the *conf/httpd.conf* configuration file and include the httpd-remoteip.conf configuration file.

```
Include conf/extra/httpd-remoteip.conf
```

4. Modify the log format in the httpd.conf configuration file.

```
LogFormat "%a %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%a %l %u %t \"%r\" %>s %b" common
```

5. Restart Apache to make the configuration effective.

For Linux operating systems

You can retrieve access users' actual IP addresses by installing the mod_rpaf third-party module of Apache.

1. Run the following commands to install the mod_rpaf module:

```
wget http://stderr.net/apache/rpaf/download/mod_rpaf-0.6.tar.gz
tar zxvf mod_rpaf-0.6.tar.gz
cd mod_rpaf-0.6
/alidata/server/httpd/bin/apxs -i -c -n mod_rpaf-2.0.so mod_rpaf-2.0.c
```

2. Modify the Apache configuration file */alidata/server/httpd/conf/httpd.conf* and append the following content to the end of the file:

> Note `RPAFproxy_ips IP address` is not the public IP address provided by SLB. For the specific IP addresses, see the Apache log. Typically, you can find two IP addresses.

```
LoadModule rpaf_module modules/mod_rpaf-2.0.so
RPAFenable On
RPAFsethostname On
RPAFproxy_ips IP address
RPAFheader X-Forwarded-For
```

3. After appending the preceding content, run the following command to restart Apache to make the configuration effective:

```
/alidata/server/httpd/bin/apachectl restart
```

mod_rpaf module configuration example

```
LoadModule rpaf_module modules/mod_rpaf-2.0.so
RPAFenable On
RPAFsethostname On
RPAFproxy_ips 10.242.230.65 10.242.230.131
RPAFheader X-Forwarded-For
```

Tomcat configuration scheme

Retrieve access users' actual IP addresses by enabling the X-Forwarded-For function of Tomcat.

Open the *tomcat/conf/server.xml* configuration file and modify the AccessLogValve log recording function as follows:

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt"
pattern="%{X-FORWARDED-FOR}i %l %u %t %r %s %b %D %q %{User-Agent}i %T" resolveHosts="false"/>
```

1.4. Configure protection for the origin server

If the IP address of your origin server is exposed, an attacker may bypass Anti-Bot Service and launch direct attacks on your origin server. This topic describes how to configure protection for the origin server.

Context

> **Note** Protection of the origin server is not required. Traffic forwarding is not affected when the protection is not configured. However, we recommend that you protect your origin server to eliminate risks arising from IP exposure.

Verify whether the origin server IP is exposed

Use Telnet to connect to the public IP of the origin server through its service port from a non-Alibaba Cloud host. If the connection is established, it indicates that the origin server is at risk. If an attacker obtains the public IP of the origin server, they can bypass Anti-Bot Service and access the origin server directly. If the connection fails, it indicates that the origin server is secure.

For example, test whether connections to the origin server IP through port 80 and 8080 can be established after you set up Anti-Bot Service for your domain. If connections are established, it indicates that your origin server is at risk.

Notes

Security groups can be difficult to use. Pay attention to the following points before configuring protection for the origin server.

- Make sure that you have configured Anti-Bot Service for all domains that are attached to the ECS or SLB instance where security groups are created.
- When failures occur in the Anti-Bot cluster, requests may be forwarded to the origin server to avoid service interruptions. In this situation, if you have configured protection for the origin server, users may not be able to access your origin server through the Internet.
- When the Anti-Bot cluster scales out, and more back-to-origin CIDR blocks are added, 5xx errors may be frequently returned if you have configured security groups to protect the origin server.

Procedure

1. Log on to the Anti-Bot Service console, and select the region where your Anti-Bot instance is located.
2. Choose **Domain Configuration** and select **Back-to-origin CIDR Blocks for Anti-Bot** to view the back-to-origin CIDR blocks used by Anti-Bot Service.

   > **Note** Back-to-origin CIDR blocks are periodically updated. We recommend that you pay attention to update notifications and add new back-to-origin CIDR blocks to corresponding security group rules in a timely manner to avoid false positives.

3. In the **Back-to-origin CIDR Blocks** dialog box, click **Copy** to copy all CIDR blocks.
4. Configure security groups to allow access from back-to-origin CIDR blocks only.
   - **If your origin server is an ECS instance**
     a. Go to the Instances page, select the ECS instance that you want to configure security groups for, and click **Manage** under the Actions column.
     b. In the left-side navigation pane, click **Security Groups**.
     c. Select the security group that you want to change, and click **Add Rules**.
     d. Click **Add Security Group Rule** and configure the security group rule as follows:

> **Note** The Authorization Objects field supports CIDR blocks in the following format: 10.x.x.x/32. You can enter up to 10 comma-separated CIDR blocks.

- **NIC**: Internal Network

> **Note** If your ECS instance is connected to a classic network, you need to set NIC to Public Network.

- **Rule Direction**: Ingress
- **Action**: Allow
- **Protocol Type**: Customized TCP
- **Authorization Type**: IPv4 CIDR Block
- **Port Range**: 80/443
- **Authorization Object**: Paste the back-to-origin CIDR blocks from step 3.
- **Priority**: 1

e. After you create this security group rule, add another security group rule as follows to block access from the Internet.

- **NIC**: Internal Network

> **Note** If your ECS instance is connected to a classic network, you need to set NIC to Public Network.

- **Rule Direction**: Ingress
- **Action**: Forbid
- **Protocol Type**: Customized TCP
- **Port Range**: 80/443
- **Authorization Type**: IPv4 CIDR Block
- **Authorization Object**: 0.0.0.0/0
- **Priority**: 100

> **Note** If your origin server needs to communicate with other IP addresses or applications, you must add another security group rule to allow access from them. Alternatively, you can add a security group rule to allow access from all ports and set it to the lowest priority.

- **If your origin server is an SLB instance**

In a similar manner, you need to add the back-to-origin CIDR blocks used by Anti-Bot Service to the whitelist of your SLB instance. For more information, see Enable access control.

a. Log on to the Server Load Balancer console, choose **Access Control**, and click **Create Access Control List**.

b. Specify an access control list name, add the back-to-origin CIDR blocks used by Anti-Bot Service, and click **OK**.

c. On the **Server Load Balancer** page, select your SLB instance.

d. In the **Listeners** tab, select the listener that you want to change and click **More > Manage Access Control**.

e. Create a whitelist, add the access control list that contains the back-to-origin CIDR blocks used by Anti-Bot Service to the whitelist, and click **OK**.

What's next

After you configure protection for the origin server, you can test whether the origin server is accessible through port 80 and 8080 to check whether the configuration takes effect. If the origin server cannot be connected through these ports and your service is running normally, it indicates that the protection configuration is in effect.

2.Protection Settings

2.1. Overview

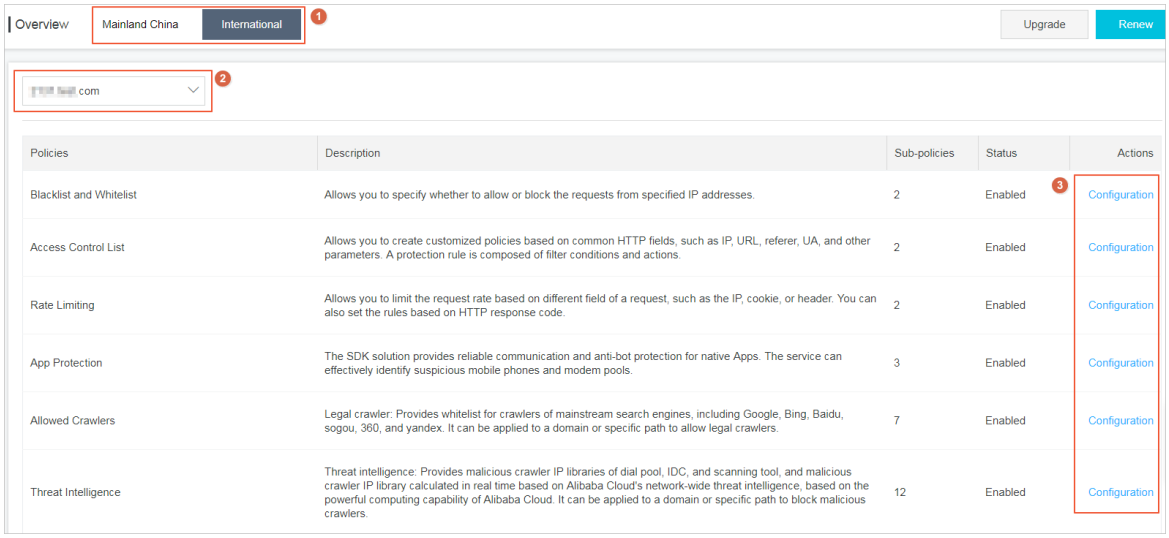
After you add a domain and set up DNS settings, you can configure custom protection policies to protect the domain from malicious bot traffic.

Procedure

1. Log on to the Anti-Bot Service console and select the region where your Anti-Bot instance is located.
2. Choose **Protection > Overview** and select the domain that you want to change settings for.

   > **Note** You can also choose **Domain Configuration**, select the domain in the domain list, and click **Policies** to open the **Overview** page.

3. Select a protection policy and click **Configure** to change configurations.



   - **Black and Whitelist**: You can allow or block bot traffic from specific IP addresses. For more information, see Blacklist and whitelist.
   - **Access Control List**: You can create custom protection policies based on common HTTP request fields, such as IP address, URLs, references, UA, and parameters to meet business needs. For more information, see Access control list.
   - **Rate Limiting**: You can limit the number of requests to specific URLs based on the IP address, cookie, or header fields of the request. You can also set limits based on response codes. For more information, see Rate limiting.
   - **App Protection**: Provides secure connectivity and anti-bot protection for native apps. This feature can accurately identify requests from proxy servers, emulators, and requests with invalid signatures. To enable App Protection, you must integrate the Anti-Bot SDK with your app. For more information, see App Protection.
   - **Allowed Crawlers**: Provides a whitelist of crawlers used by mainstream search engines, such as Google, Bing, Baidu, Sogou, 360, and Yandex. You can allow these search engine crawlers to access the entire site or specific directories. For more information, see Bot intelligence.

- **Threat Intelligence**: Provides information about suspicious IP addresses used by harassing phone calls, data centers, and malicious scanners based on Alibaba Cloud's powerful computing capability. This feature also maintains an IP library of malicious crawlers and can prevent crawlers from accessing your site or specific directories in real time. For more information, see Bot intelligence.

2.2. Blacklist and whitelist

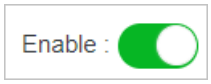
You can set an IP address blacklist and whitelist for the specified domain name to directly allow or block the bot traffic from the IP addresses in the blacklist or whitelist.

Context

Blacklist and whitelist policies take precedence over other protection policies. That is, requests from the IP addresses in the blacklist or whitelist are directly blocked or allowed. The whitelist policy takes precedence over the blacklist policy. That is, if an IP address is added in both the blacklist and whitelist, the whitelist policy for the IP address takes effect, allowing requests from the IP address.

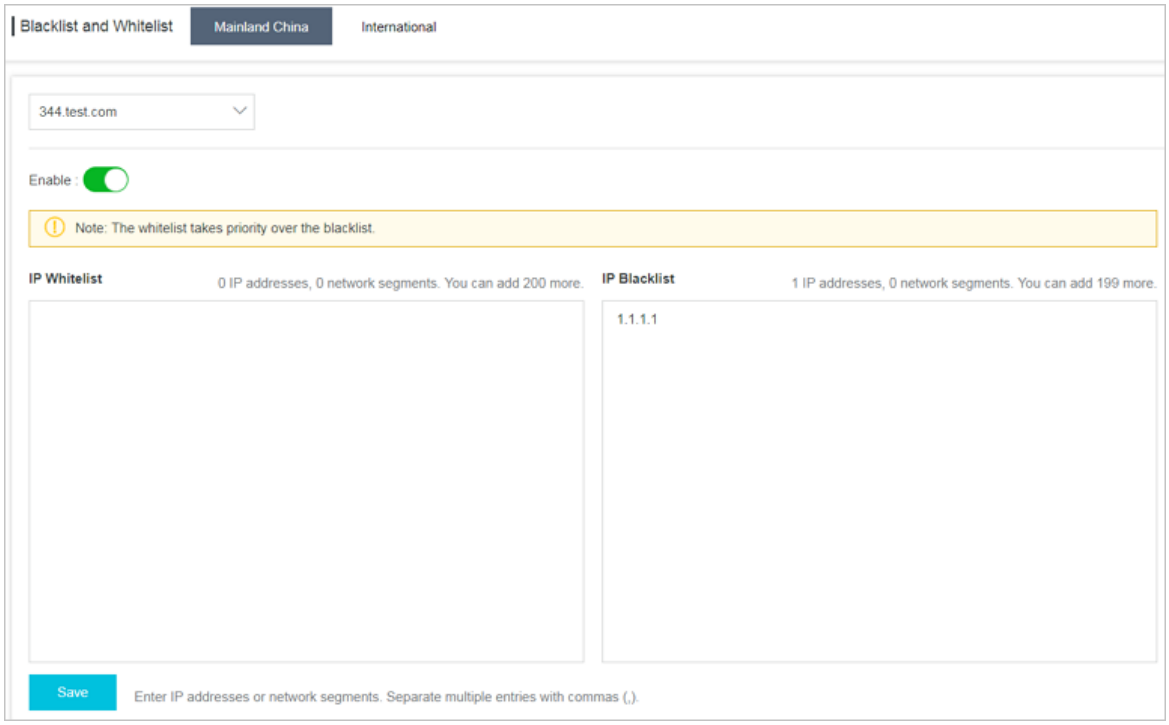
Procedure

1. Log on to the Anti-Bot console, and select the region where your Anti-Bot instance is located.
2. Choose **Protection > Blacklist and Whitelist**. Select the protected domain name.
3. Turn on **Enable**

   to enable the blacklist and whitelist policies.
4. In the **IP Blacklist** or **IP Whitelist** field, enter the IP address or CIDR block to be directly allowed or blocked, and click **Save**.

   > **Note** You can enter IP addresses or a CIDR block (IP address/mask). Separate multiple IP addresses with commas (,).



The IP address blacklist or whitelist takes effect once it is saved.

2.3. Access control list

Through access control list (ACL) policies, you can configure custom access control rules based on your service scenarios. Combine common HTTP fields (such as IP, URL, Referer, UA, and other parameters) to formulate filter conditions, filter the access requests initiated to the website domain name, and set Monitor, Block, Slider Captcha, or Allow for the access requests that match the filter conditions.

Context

Rule description

An ACL rule consists of **Rule Condition** and **Rule Action**. When creating a rule, define a filter condition by setting Filter Field, Operator, and Filter Pattern, and define Rule Action for the access requests that match the filter condition.

- **Rule Condition**

  Rule Condition consists of Filter Field, Operator, and Filter Pattern.

Filter Field	Field description	Applicable operator
URL	The URL in the access request.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal
IP	The source IP address of the access request.	○ Belongs To ○ Does Not Belong To

Filter Field	Field description	Applicable operator
Referer	The source URL of the access request, that is, the page from which the access request is redirected.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than ○ Is Not Part Of
User-Agent	The web browser information about the client initiating the access request, including the web browser identifier, rendering engine identifier, and version.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than
Params	The parameter part of the URL in the access request, which is typically the part following the question mark (?) in the URL. For example, in `www.abc.com/index.html?action=login`, `action=login` is the parameter part.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than
Cookie	The cookie information in the access request.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than ○ Is Not Part Of
Content-Type	The HTTP content type of the response specified by the access request, that is, MIME type information.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than

Filter Field	Field description	Applicable operator
Content-Length	The number of bytes in the response to the access request.	○ Is Smaller Than ○ Has a Value Of ○ Is Larger Than
X-Forwarded-For	The actual client IP address in the access request. **Note** 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 HTTP proxy or load balancing. XFF is only included in the access requests that are forwarded by the HTTP proxy or SLB.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than ○ Is Not Part Of
Post-Body	The content of the response to the access request.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal
Http-Method	The method of the access request, such as GET and POST.	○ Equals ○ Does Not Equal

Filter Field	Field description	Applicable operator
Header	The header of the access request, which is used to customize the HTTP header field.	○ Is Part Of ○ Does Not Contain ○ Equals ○ Does Not Equal ○ Is Shorter Than ○ Has a Length Of ○ Is Longer Than ○ Is Not Part Of

- **Rule Action**

  An ACL rule supports the following rule actions:

  - **Block**: blocks the access requests that match the filter condition.
  - **Allow**: allows the access requests that match the filter condition.
  - **Monitor**: allows the access requests that match the filter condition. Meanwhile, you can view the requests that match the filter rule in a data report to check the effect of the access control list rule.
  - **Slider Captcha**: sends a slide-to-verify request to the client whose access request matches the filter condition to perform secondary bot recognition. The access user must drag the slider to complete verification in order to continue service operation. Service operation will be terminated when the verification fails (the access user does not drag the slider or the operation is not human-initiated).

- **Rule matching order**

  If you configure multiple ACL rules, the rules are matched in order. That is, access requests are matched with rules in the specified order. Rules on top are matched first. When an access request matches the filter condition of a rule, the request is processed based on the action specified by the rule, and matching stops.

  You can sort all the ACL rules by using the rule sorting function to obtain optimal protection.

Default rule description

An ACL contains a default rule.

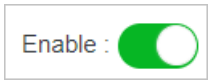
- **Rule Condition**: All the requests that do not match the preceding rules. After the ACL policy is enabled, all the access requests that do not match the configured ACL rules are processed based on the action specified by the default rule.
- **Rule Action**: The default action is that the system allows all the access requests that do not match the configured ACL rules and continues to execute other protection policies (rate limiting and app protection).

> **Note** The default rule cannot be deleted, and its filter condition cannot be modified. The default rule is always the last to be matched, and its order cannot be changed.

Procedure

1. Log on to the Anti-Bot console, and select the region where your Anti-Bot instance is located.

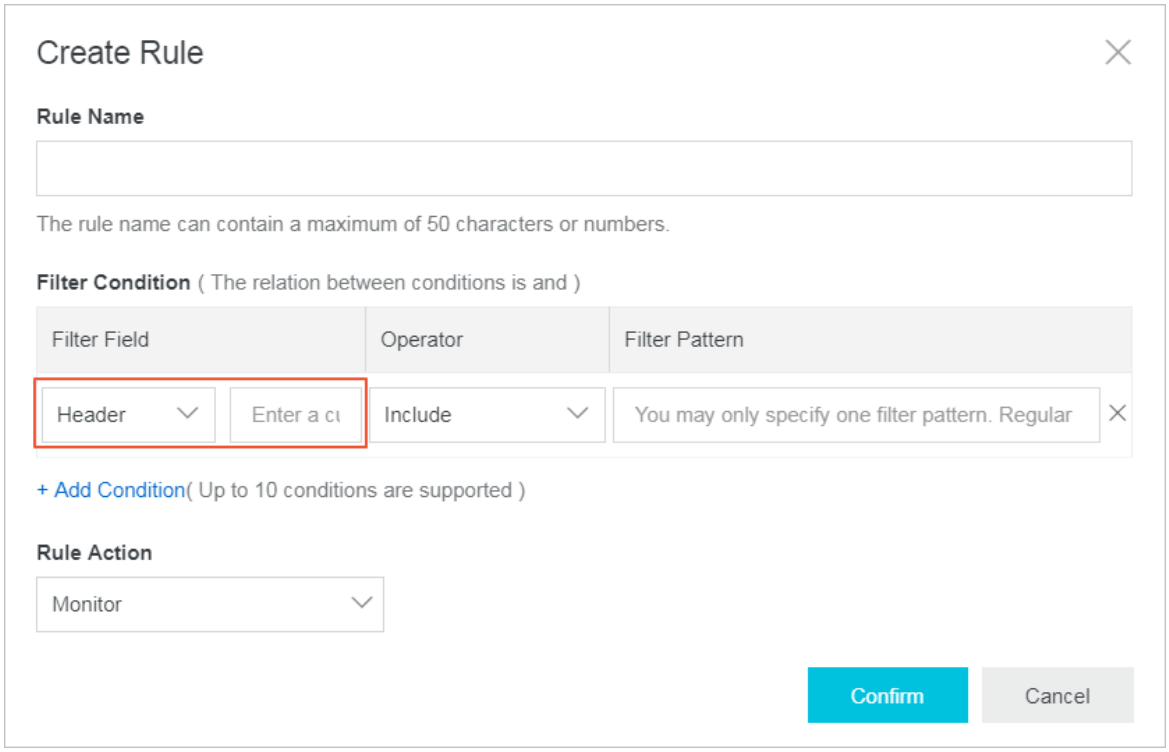
2. Choose **Protection > Access Control List**. Select the protected domain name.
3. Turn on **Enable**



to enable the ACL policy.

4. Click **Add**. Set the filter condition and action of the rule. Then, click **OK**.

> **Note** When you select Header as the filter field, you need to set the Key field of the customer header. For example, if the service-indicating field in the Header field is `userid=xxxxxx`, enter `userid` in the custom Header field to use the userid field as the filter condition.



When the rule is added, you can **modify** or **delete** it. If multiple rules are added, you can go to the **Access Control List** page and click **Sort**. Then, click **Move Up**, **Move Down**, **Stick to Top**, or **Stick to Bottom** to adjust the rule matching order. The rules on top are matched first. After adjusting the rule matching order, click **Save Order** to make the order effective.

2.4. Rate limiting

A rate limiting policy is used to limit the rate at which request objects access specified URLs to intercept malicious bot traffic. In addition to rate limiting, you can add other limitations, such as the number or proportion of specific response codes to limit the access requests of request objects.

Context

Rate limiting rule

A rate limiting rule consists of the following parameters:

Parameter	Description
Rule Name	The name of the rule. We recommend that you set a name that reflects the meaning of the rule.
URL	The URL to which the rule is applied, such as `/login.com`. **Note** This field cannot be empty. URL setting supports the following match rules: • **Full match**: is also referred to as exact match. The rule collects statistics only on those client-requested URLs that are exactly the same as the configured URL, which must start with a `slash (/)`. • **Prefix match**: is also referred to as left-include match. The rule collects statistics only on those rules that start with the configured URL prefix. The configured URL must start with a `slash (/)`. For example, if the configured URL is `/login`, then the requested URL `/login.html` is counted by the rule. • **Regular match**: Matching is based on a regular expression. Enter a complete regular expression in the URL text box. Then, the rule collects statistics on the client access with requested URLs that are compliant with the regular expression. **Note** You can enter a parameter-carrying URL, such as `/user? action=login`.

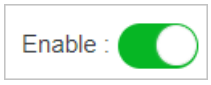
Parameter	Description
Request Object	The entity that is counted by the rule. The request object can be an IP address, default cookie, custom header, parameter, or a field in the cookie. **Note** • The default cookie is the automatically added cookie when a normal user request reaches the engine. It typically starts with `acw_tc`. • **Example of statistics based on a custom field**: A service identifies users by using `token: 123456` in the HTTP header. You can configure the custom header or token as a request object for rate statistics.
Duration	The period when the rule counts request times.
Requests	The maximum number of request times accumulated by a single request object during the configured statistical period. **Note** In addition to the request times limit, you can add a **response code** limit condition, such as a maximum of 300 accumulated request times with Response Code 503 and 70% of request times with Response Code 503. The action specified by the rule is triggered only when the counted request times exceed the maximum number and the number or proportion of request times meets the response code limit condition.

Parameter	Description
Rule Action	The action triggered when the rule condition is met. • **Monitor**: The system does not trigger any action on the request, but only records the statistical results in the rule-matched data report for the purpose of checking the actual effect of the rule. • **Block**: The system disconnects the request object. • **JavaScript**: The system sends a verification request message to the client through redirection. The client must pass verification before continuing service operation. • **Slider Captcha**: The system sends a slide captcha verification request message to the client for secondary bot recognition. The access user must drag the slider to complete verification in order to continue service operation. Service operation will be terminated when the verification fails (the access user does not drag the slider or the operation is not human-initiated). **Note** • When Rule Action is set to Block, you can set a block duration (**blacklisting duration**) for request objects. For example, if the block duration is set to 30 minutes, all the IP address access requests that meet the condition are blocked within 30 minutes. • You can configure Rule Action to take effect for the global requests of the domain name or only the requests that match the rule-specified URL.

Note The statistical process may encounter a delay because the data of multiple servers in the cluster must be summarized for statistics. Therefore, the actual effective time of the rate limiting condition may be delayed.

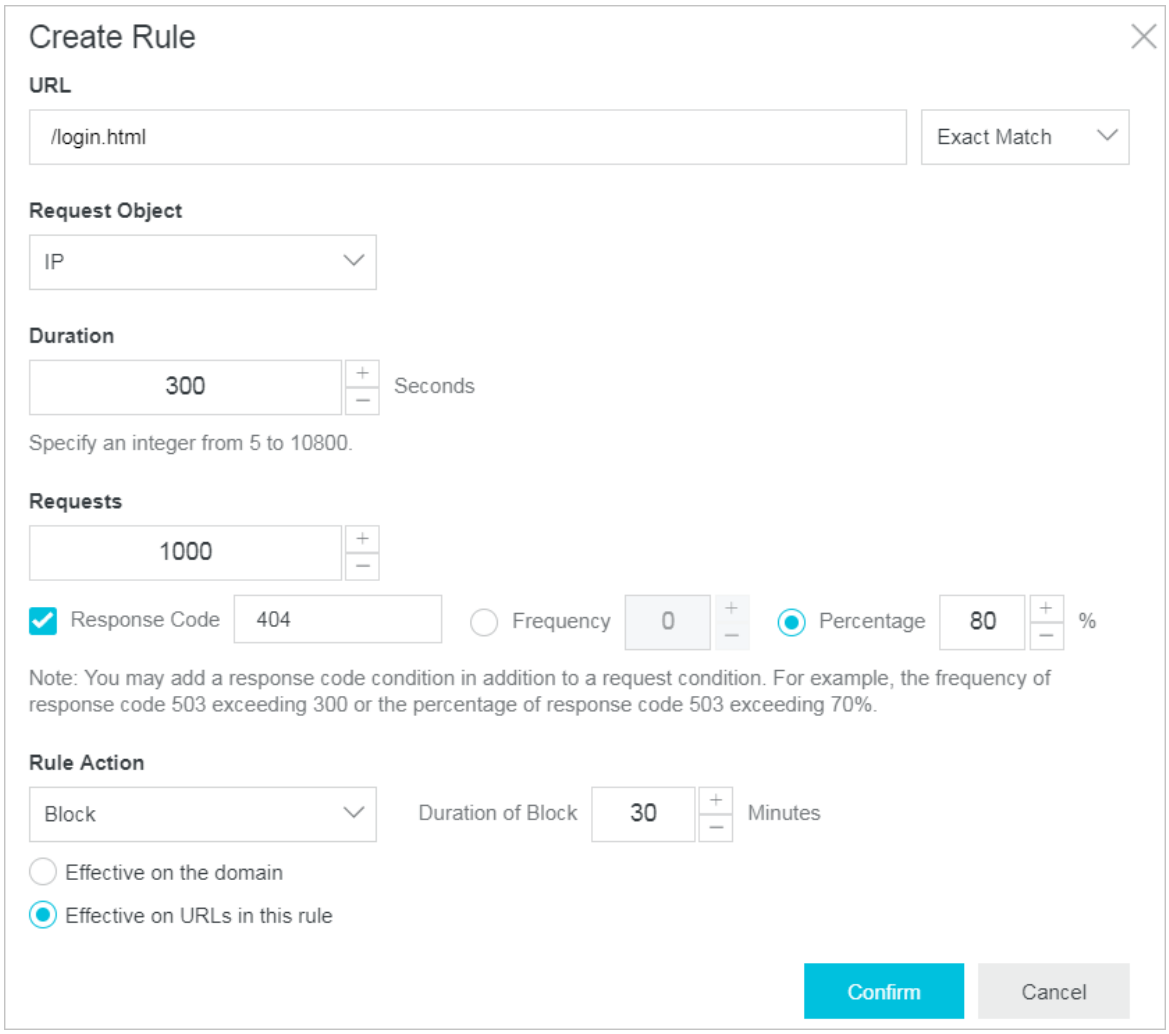
Procedure

1. Log on to the Anti-Bot console, and select the region where your Anti-Bot instance is located.
2. Choose **Protection > Rate Limiting**. Select the domain name of the protected website.
3. Turn on **Enable**

   to enable the rate limiting policy.

4. Click **Add** to configure a rate limiting rule. Then, click **OK**. For example, you can configure the Block action to be triggered when a single source IP address initiates more than 1,000 access requests to `1.test.com/login.html` within 5 minutes (300 seconds) and the proportion of accumulated requests with Response Code 404 exceeds 80%, and block the IP address for 30 minutes. This configuration only takes effect for the rule-specified URL 1.test.com/login.html.



2.5. Bot intelligence

The bot intelligence rule is backed with the Alibaba Cloud Bot Intelligence Library and helps you allow requests from valid crawlers and inspect suspicious requests from known threatening source IPs.

Context

The Alibaba Cloud Bot Intelligence Library is calculated based on Alibaba Cloud's network-wide traffic and is updated in real time. It covers the following source IP characteristics:

- Valid crawlers: Dynamically updated source IP library of crawlers of mainstream search engines, including Google, Bing, Baidu, sogou, 360, and yandex.

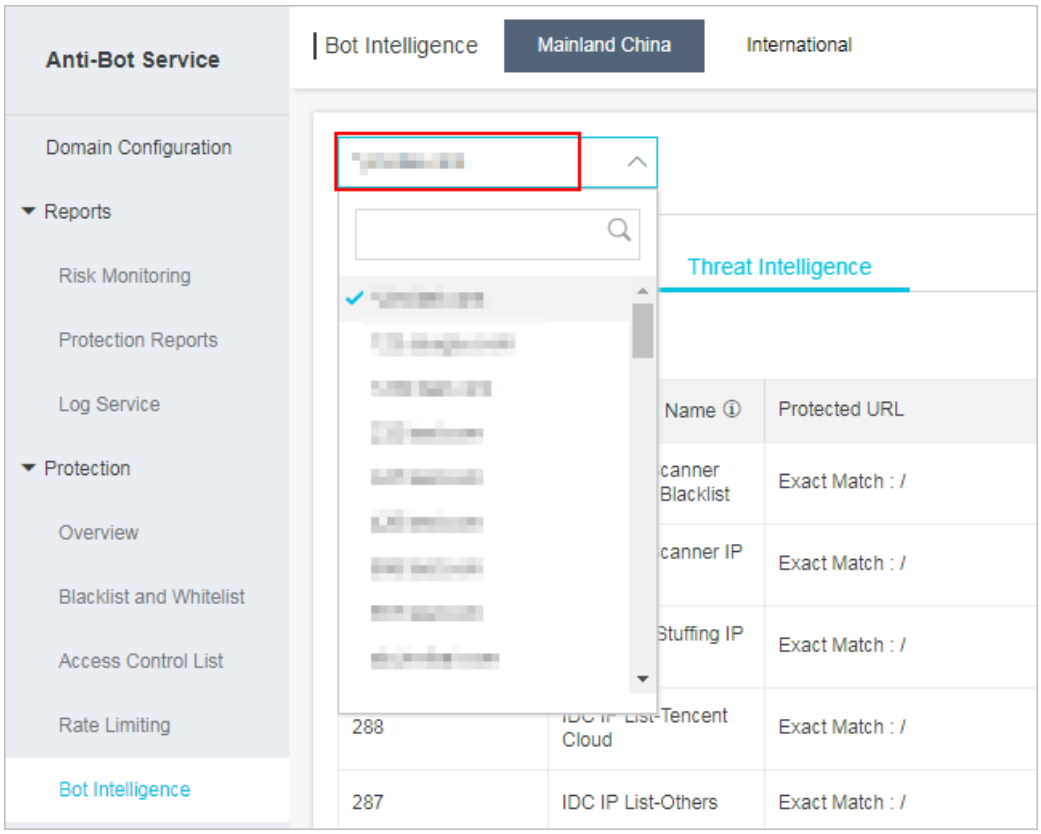
  You can enable the default valid crawler rule to allow requests from the specified search engines. The Blacklist and Whitelist, and Access Control List features still apply to the valid crawler requests.

- Threat intelligence: Malicious crawler IP library calculated in real time based on Alibaba Cloud's network-wide threat intelligence, and dynamically updated public cloud/IDC IP libraries.

You can customize the threat intelligence rule to trigger different protection actions against requests from different types of blacklist IP addresses. Options are monitoring, interception, JavaScript verification, and slider verification. In addition, you can configure protection for some key interfaces against specific blacklisted IP addresses to avoid impact on other business logic.

Procedure

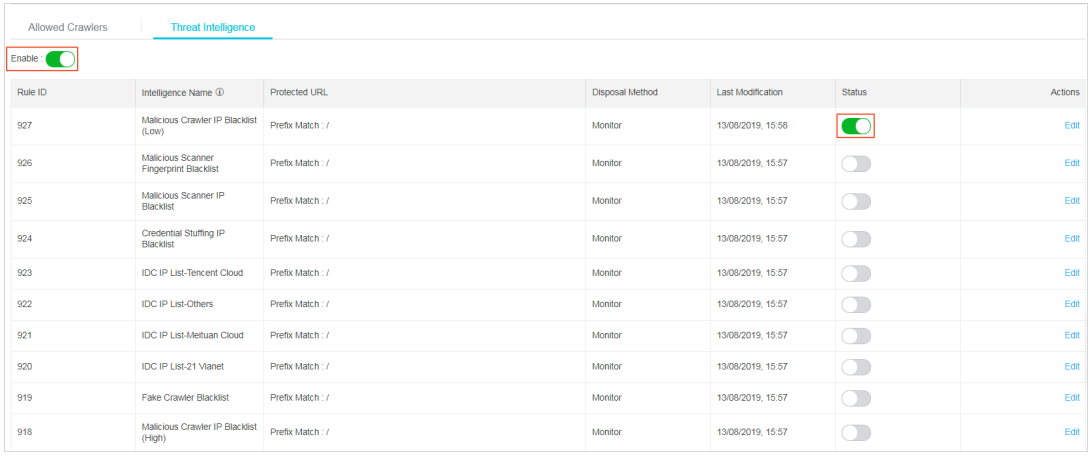
1. Log on to the Anti-Bot Service management console.
2. In the left-side navigation pane, select **Protection > Bot Intelligence**.
3. In the domain name drop-down box, select the domain name to be configured.



4. Complete the following configuration respectively on the **Allowed Crawlers** and **Threat Intelligence** tab pages.
   - Allow valid crawlers

a. On the **Allowed Crawler** tab page, turn on the **Enable** switch.

> **Note** If you no longer need this feature, turn off the **Enable** switch on this page.



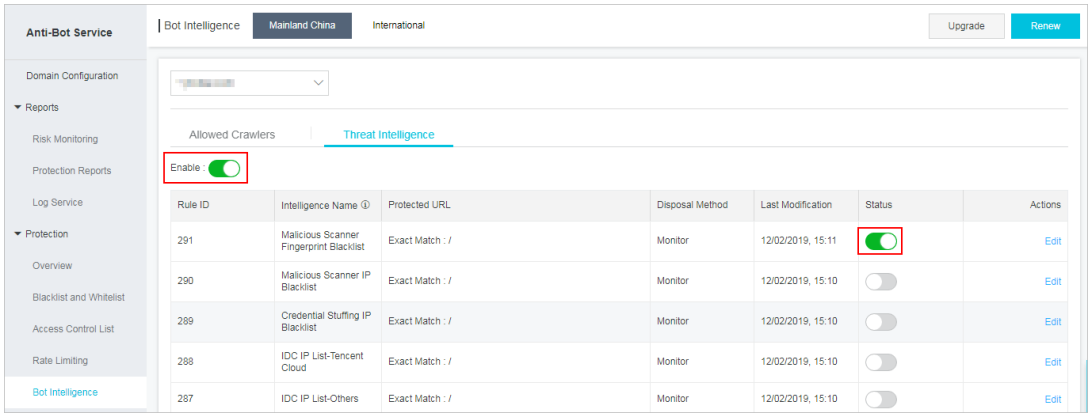
b. In the rule list, locate to a valid crawler according to the **Intelligence Name**, and turn on the corresponding **Status** switch to allow requests from it. Currently, crawling requests from the following search engines can be configured: GoogleBot, BingBot, BaiduSpider, SogouSpider, 360 Spider, and YandexBot.

> **Note** Alternatively, you can only enable rule 106 (**Legit Crawling Bots**) to allow all supported search engine bots.

- Add a threat intelligence rule

a. On the **Threat Intelligence** tab page, turn on the **Enable** switch.

> **Note** If you no longer need this feature, turn off the **Enable** switch on this page.



b. In the rule list, locate to an IP blacklist according to the **Intelligence Name**, and turn on the corresponding **Status** switch. The optional IP blacklists are as follows:

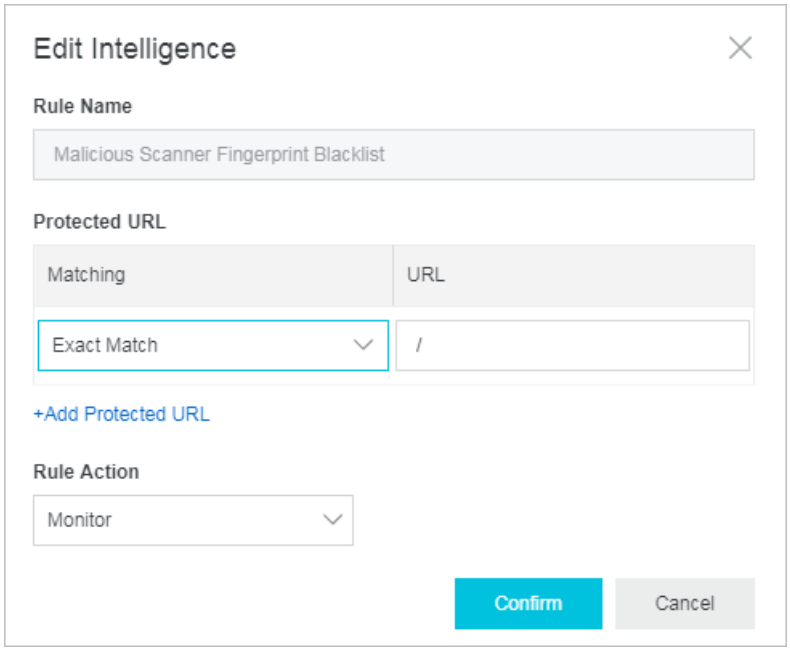
IP Blacklist	Description
Malicious Scanner Fingerprint Blacklist	The common scanners.
Malicious Scanner IP Blacklist	A dynamic IP library that is obtained by analyzing the source IP addresses of malicious scanning behaviors detected by Alibaba Cloud in real time.
Credential Stuffing IP Blacklist	A dynamic IP library that is obtained by analyzing the source IP addresses of credential stuffing and brute-force cracking behaviors detected by Alibaba Cloud in real time.
Fake Crawler Blacklist	Invalid crawlers that forge the user-agent of legitimate search engines (such as baiduspider) to avoid detection. **Notice** Before enabling this rule, make sure that you have allowed valid crawler requests. Otherwise, false positives may occur.
Malicious Crawler IP Blacklist	A dynamic IP library that is obtained by analyzing the source IP addresses of malicious crawling behaviors detected by Alibaba Cloud in real time. This IP blacklist has three levels: Low, Medium, and High. The higher the level, the more IP addresses are included and the higher the possibility of false positives. We recommend that you enable two-step verification for the high-level blacklist. Options are using slider verification or JS verification. For interfaces that do not apply to two-step verification (such as APIs), you can enable the low-level blacklist rule.
IDC IP List	The following IDC IP lists are covered: Alibaba Cloud, Tencent Cloud, Meituan Cloud, 21 Vianet, and Others. These IP segments are often used by crawlers to deploy crawling programs or act as proxies, and are seldom used by normal users.

After the default threat intelligence rule is enabled, when the source IP address in the specified blacklist initiates an access request to any path under the domain name, the monitoring operation is triggered. That is, the request is allowed and recorded.

If you want to modify the default rules (for example, specify the key interfaces to be protected or change the disposal action), follow these steps to customize threat intelligence rules.

c. (Optional)(Optional) Select the default rule to be configured and click **Edit**.

d. (Optional)(Optional) In the **Edit Intelligence** dialog box, complete the following configuration:

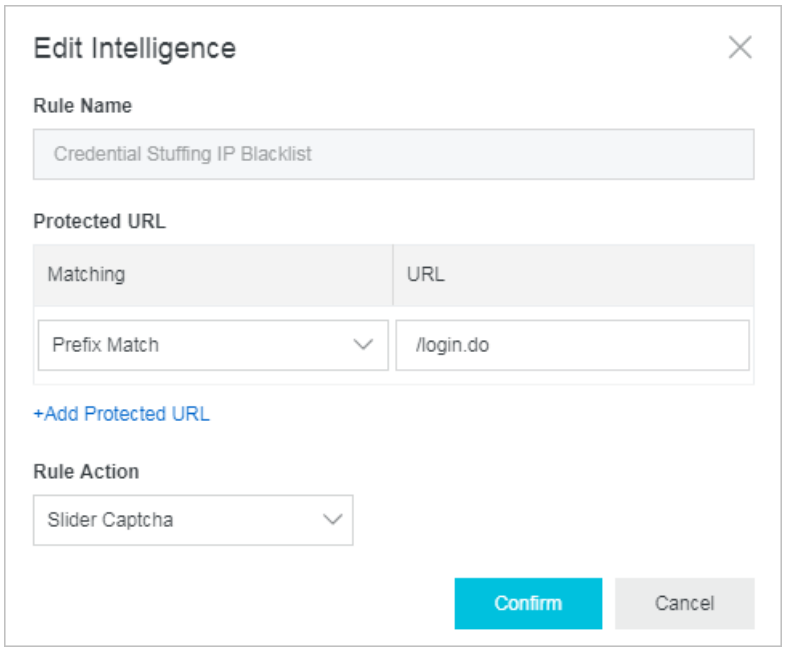


Configuration	Description
Protected URL	Enter the specific **URL** to inspect (for example, "/ABC", "/login/ABC". "/" indicates all paths) and select the **Matching** method: ■ **Exact Match**: Hit when the requested URL exactly matches the specified URL. ■ **Prefix Match**: Hit when the requested URL's prefix matches the specified URL. ■ **RegExp Match**: Hit when the requested URL meets the regular expression of the specified URL. **Note** Click **Add Protected URL** to add up to 10 URLs.
Disposal method	Specify the action to perform when the rule is triggered. ■ **Monitor**: Allow and record the request. ■ **Block**: Block the request. ■ **JavaScript Validation**: Verify the request data through JavaScript and allow the request after the verification is passed. ■ **Slider Captcha**: Start a slider verification page on the client and allow the request after the verification is completed. **Note** Slider Captcha is applicable to synchronous requests. For asynchronous requests (such as Ajax) protection, contact Alibaba Cloud security team. If you are not sure whether the interface you want to protect can use slider verification, we recommend that you use a test IP and URL to perform a test with the Access Control List rule.

Examples of custom threat intelligence rules

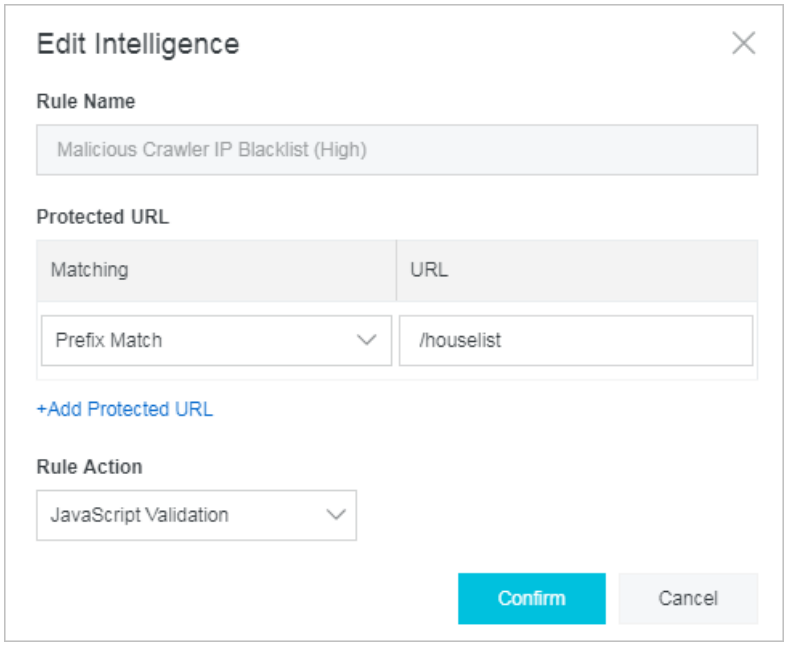
- Rule description: Use this rule to protect the URLs starting with "/login.do" under the current domain name. When the request source IP address matches the Credential Stuffing IP Blacklist, require the client to complete slider verification.

  Rule configuration:



- Rule description: Use this rule to protect the URLs starting with "/houselist" under the current domain name. When the request source IP address matches the Malicious Crawler IP Blacklist (High), perform JavaScript verification on the request.

  Rule configuration:



e. (Optional)(Optional) Click **OK** to finish the configuration.

3.App protection

3.1. Solution overview

Anti-Bot Service (Anti-Bot) provides a security solution, Anti-Bot SDK, for native apps. It provides your apps with enhanced protection against bot traffic, supports secure communication, and can accurately identify suspicious IP addresses and modem pools.

The Anti-Bot SDK was developed based on years of experience in protecting against frauds and bargain speculators in their online business. After your app is integrated with the Anti-Bot SDK, your app will gain the same trusted channel as Tmall, Taobao, Alipay, and other apps. It will have access to a library of malicious devices accumulated by Alibaba Group against frauds and bargain speculators in their online business, helping you to solve your app's security problems.

The Anti-Bot SDK helps you to solve the following security problems that can threaten **native apps**:

- Malicious registration, credential stuffing, and brute-force attacks
- HTTP flood attacks against apps
- Malicious attacks against SMS and CAPTCHA interfaces
- Bargain speculation and red envelope snatching
- Seckill and time-and-purchase-limited goods
- Malicious ticket checking and brushing (such as air tickets or hotel bookings)
- Valuable information crawling (such as price, credit information, financing, and fiction)
- Machine batch voting
- Spams and malicious comments

Configure the Anti-Bot SDK for your app

Perform the following steps to configure the Anti-Bot SDK for your app:

> **Note** You do not need to make any modifications on the server to configure the Anti-Bot SDK for your app. After the configuration is complete, Anti-Bot automatically filters out malicious traffic and forwards valid requests to the origin server. Anti-Bot handles all pressure from malicious traffic to ensure the stability of your server.

1. Log on to the Anti-Bot console, and select the region where your Anti-Bot instance is located.
2. Go to the **Domain Names** page and click **Add Domain** to configure domain access to the domain name used by your app. For more information, see Add domain name configuration.
3. At the DNS resolution service provider of the domain name used by your app, add the CNAME allocated by your Anti-Bot instance, and point the domain name resolution of your app to the Anti-Bot instance. For more information, see Update DNS settings.
4. Integrate the Anti-Bot SDK into your app.

   > **Note** This may take one to two days to complete.

   For more information about how to integrate the Anti-Bot SDK into your app, see the following documents:

   - iOS SDK integration guide

- Android SDK integration guide

5. After verifying that the configuration is successful, package and publish the new app version integrated with the Anti-Bot SDK for security protection.

3.2. iOS SDK integration guide

To integrate the Anti-Bot SDK into your iOS app, follow the instructions provided in this topic.

iOS SDK files

Contact the technical support team for Anti-Bot Service to obtain the correct SDK package. Decompress it on your local machine.

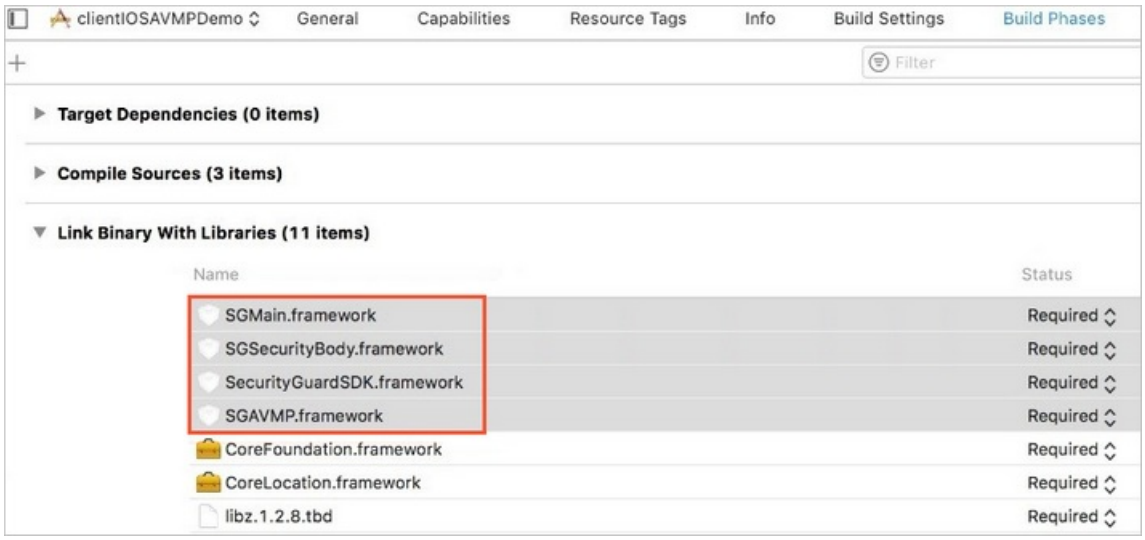
The sdk-iOS folder contains the following iOS SDK files.

File	Description
SGMain.framework	Main framework
SecurityGuardSDK.framework	Basic security plugin
SGSecurityBody.framework	Bot recognition plugin
SGAVMP.framework	VM plugin
yw_1222_0335_mwua.jpg	Configuration file

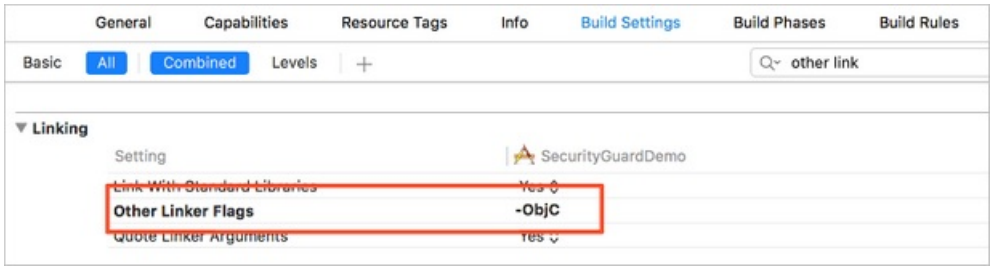
Configure a project

Perform the following steps to configure a project for your app:

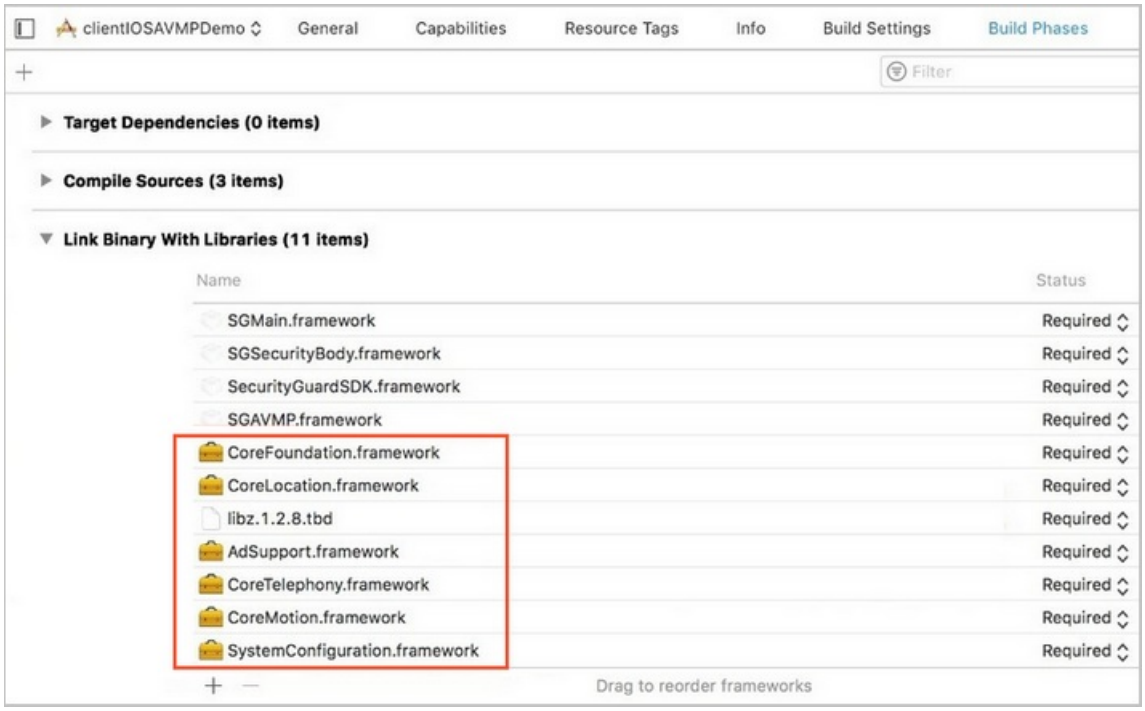
1. Add frameworks. Add the four `.framework` files in the Anti-Bot SDK package to the dependent library of the iOS app project.



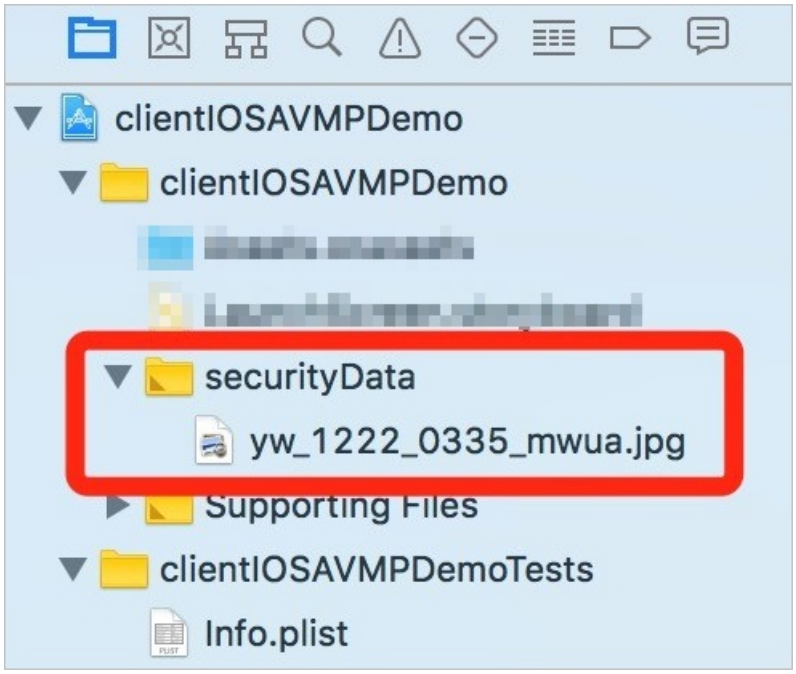
2. Add other linker flags.



3. Add the following system dependent libraries.



4. Import the configuration file. Add the `yw_1222_0335_mwua.jpg` configuration file in the SDK package to the `mainbunle` directory.



> **Note** When the app integrates multiple targets, make sure to add the `w_1222_0335_mwua.jpg` configuration file to the correct Target Membership.

Develop code

1. Initialize the SDK.
   - **Interface definition:** `+ (BOOL) initialize;`
   - **Interface description:**
     - Function: Initializes the SDK.
     - Parameter: None.
     - Return value: Boolean type. YES is returned if initialization is successful, whereas NO is returned if initialization fails.
   - **Call method:** `[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;
 }
}
```

2. Sign the request data.
   - **Interface definition:** `+ (NSData*) avmpSign: (NSInteger) signType input: (NSData*) input;`
   - **Interface description:**
     - Function: Signs the input data by using the AVMP technique, and returns the signature string.

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

- Parameters: See the following table.

Parameter	Type	Required	Description
signType	NSInteger	Yes	The algorithm used by the signature. Currently this parameter is fixed, and is set to `3`.
input	NSData*	No	The data to be signed, which is generally the entire request body. **Note** If the request body is empty (for example, the body of a POST or GET request is empty), enter null or the bytes value of an empty string.

- Return value: NSData* type. The signature string is returned.

- **Call method:** `[JAQAVMPSignature avmpSign: 3 input: request_body];`
- **Sample code:**

Note When the client sends data to the server, it must call the avmpSign interface to sign the entire request body. Then, the signature string wToken is obtained.

```
# 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 = nil;
  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;
  }
 }
}
```

Note Even if the request body is empty, the client still must call the avmpSign interface to generate the wToken. In this case, directly use null as the second parameter.

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

3. Insert the wToken in the protocol header.

   Sample code

```
#define VMP_SIGN_WITH_GENERAL_WUA2 (3)
-(void)setHeader
{ NSString* request_body = @"i am the request body, encrypted or not!" ;
 NSData* body_data = [request_body dataUsingEncoding:NSUTF8StringEncoding];
 NSString* wToken = nil;
 NSData* sign = [JAQAVMPSignature avmpSign: VMP_SIGN_WITH_GENERAL_WUA2 input:body_data];
 wToken = [[NSString alloc] initWithData:sign encoding: NSUTF8StringEncoding];
 NSString *strUrl = [NSString stringWithFormat:@"http://www.xxx.com/login"];
 NSURL *url = [NSURL URLWithString:strUrl];
 NSMutableURLRequest *request =
  [[NSMutableURLRequest alloc]initWithURL:url cachePolicy:NSURLRequestReloadIgnoringCacheData t
imeoutInterval: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 startIm
mediately:true];
 [mConn start];
 // ...
}
```

4. Send data to the server mapping the app. Send the data with the modified protocol header to

Anti-Bot Service, which parses the wToken for risk identification and malicious request interception, and then forwards legitimate requests to the mapping server.

Error codes

Calling of the initialize and avmpSign interfaces may encounter exceptions. If an exception or error occurs when generating the signature string, search `SG Error` for related information in the console.

Common error codes and definitions

Error code	Definition
1901	Incorrect parameter value. Enter the correct parameter value.
1902	Image file error. BundleID mismatch.
1903	Incorrect image file format.
1904	Upgrade to the latest image version. The AVMP signature function only supports v5 images.
1905	Unable to find the image file. Make sure that the yw_1222_0335_mwua.jpg image file has been correctly added in the project.
1906	byteCode corresponding to the AVMP signature is missing from the image. Check whether the image is correct.
1907	Failed to initialize AVMP. Try again later.
1910	Invalid avmpInstance instance. Possible causes are as follows: • InvokeAVMP is called after AVMPInstance is destroyed. • The byteCode version of the image does not match that of the SDK.
1911	byteCode of the encrypted image does not have the corresponding export function.
1912	The AVMP call failed. Submit a ticket for help.
1913	InvokeAVMP is called after AVMPInstance is destroyed.
1915	Insufficient memory during the AVMP call. Try again later.
1999	Unknown error. Try again later.

3.3. Android SDK integration guide

To integrate the Anti-Bot SDK into your Android app, follow the instructions provided in this topic.

Android SDK files

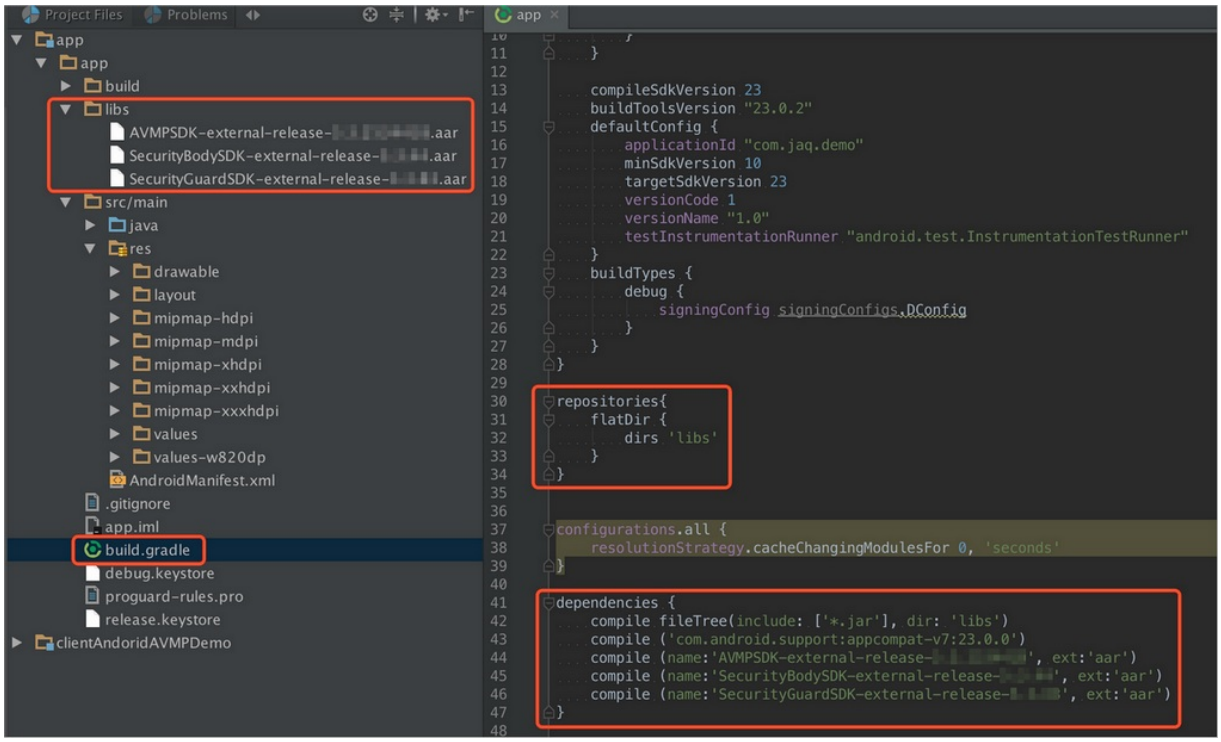
Contact Customer Services to obtain the correct SDK package. Decompress it on your local machine.

The sdk-Android folder contains the following Android SDK files.

File	Description
SecurityGuardSDK-xxx.aar	Main framework
AVMPSDK-xxx.aar	VM engine plugin
SecurityBodySDK-xxx.aar	Bot recognition plugin
yw_1222_0335_mwua.jpg	VM engine configuration file

Configure a project

Perform the following steps to configure a project:



1. Import the .aar files of the Anti-Bot SDK to Android Studio. Copy all the .aar files in the sdk-Android folder to the `libs` directory of the Android app project.

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

2. Open the build.gradle file of this project and add the following configuration.
   - Add the `libs` directory as the source for searching dependencies.

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

- Add compilation dependencies.

> **Note** The .aar file versions discussed in this topic 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. Import the .jpg configuration file of the Anti-Bot SDK 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 app project.

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

4. Add "abiFilters" to remove redundant .so files. Currently, the Anti-Bot SDK only provides .so files in the armeabi, armeabi-v7a, and arm64-v8a architectures. Therefore, you must filter the exported ABIs. Otherwise, the app may crash.
   i. In the libs directory of the Android app project, except for armeabi, armeabi-v7a, and arm64-v8a, delete folders for all the other CPU architectures, including x86, x86_64, mips, and mips64.

ii. Add a filter rule in the build.gradle configuration file of the app project. Architectures specified by abiFilters are included in the APK. See the following sample code.

> **Note** Only the armeabi architecture is specified in the following sample code. You can also specify the armeabi-v7a or arm64-v8a architecture.

```
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 only the .so files in the armeabi architecture, you can remarkably reduce the size of the app without affecting its compatibility.

5. Configure app permissions.
   - Assume that the project is an Android Studio project and uses AAR for integration. The relevant permissions have been stated in AAR, so there is no need to configure additional permissions in the project.
   - For an Eclipse project, you must add the following permissions configuration 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 the ProGuard configuration.

> **Note** If you have used ProGuard for obfuscation, then you must add the ProGuard configuration. Based on different integration methods, the ProGuard configuration is divided into two types: Eclipse and Android Studio.

   - **Android Studio**

The proguard-rules.pro configuration file is used for obfuscation if proguardFiles is configured in build.gradle and minifyEnabled is enabled.

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

- **Eclipse**

  ProGuard is used for obfuscation if the ProGuard configuration is specified in project.properties (for example, if project.properties contains the `proguard.config=proguard.cfg` statement).

  > **Note** Obfuscation is configured in the proguard.cfg file.

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

Add keep rules

To ensure that certain classes are not obfuscated, you must add the following rules in 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. **{*;}
```

Develop code

1. Import the SDK package.

   ```
   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;
   ```

2. Initialize the SDK.

   - **Method definition**: `boolean initialize();`
   - **Method description**:
     - Function: initializes the SDK.
     - Parameter: None.
     - Return value: Boolean type. true is returned if initialization is successful, whereas false is returned if initialization fails.
   - **Sample code**:

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

3. Sign the request data.
   - **Method definition:** `byte[] avmpSign(int signType, byte[] input);`
   - **Method description:**
     - Function: signs the input data by using the AVMP technique, and returns the signature string.
     - Parameters: See the following table.

Parameter	Type	Required	Description
signType	Integer	Yes	The algorithm used by the signature. Currently this parameter is fixed, and is set to `3`.
input	byte[]	No	The data to be signed, which is generally the entire request body. **Note** If the request body is empty (for example, the body of a POST or GET request is empty), fill in null or the bytes value of an empty string, such as "".getBytes("UTF-8")).

     - Return value: byte[] type. The signature string is returned.

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

```
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);
```

4. Insert the wToken in the protocol header. Add the content of the wToken field to the HttpURLConnection class object.

   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();
```

5. Send data to the server. Send the data with the modified protocol header to the server mapping the app. Anti-Bot Service captures the data and parses the wToken for risk identification.

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

Error codes

If you call the initialize and avmpSigni methods, exceptions may occur. If an exception or error occurs when generating the signature string, search `SecException` for related information in the log.

Common error codes and definitions

Error code	Description
1901	Incorrect parameter value. Enter the correct parameter value.
1902	Image file error. The APK signature used to retrieve the image file is inconsistent with the current app's APK signature. Generate a new image file by using the current app's APK.
1903	Incorrect image file format.

Error code	Description
1904	Upgrade to the latest image version. The AVMP signature function only supports v5 images.
1905	Unable to find the image file. Make sure that the image file is in the res\drawable directory. The AVMP image is yw_1222_0335_mwua.jpg.
1906	byteCode corresponding to the AVMP signature is missing from the image. Check whether the image is correct.
1907	Failed to initialize AVMP. Try again later.
1910	Invalid avmpInstance instance. Possible causes are as follows: • InvokeAVMP is called after AVMPInstance is destroyed. • The byteCode version of the image does not match that of the SDK.
1911	byteCode of the encrypted image does not have the corresponding export function.
1912	AVMP call fails. Submit a ticket for further assistance.
1913	InvokeAVMP is called after AVMPInstance is destroyed.
1915	Insufficient AVMP memory. Try again later.
1999	Unknown error. Try again later.

Test and verify the effect of integration

Perform the following steps to verify that your app has been correctly integrated with the Anti-Bot SDK:

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 subfolders.

   > **Note** If you find folders for other architectures, delete them. For more information, see Delete folders for other architectures.

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

> **Note** If the signature information is not generated, see the error messages to resolve the problem.

FAQ

Why is the key image incorrectly optimized after shrinkResources is specified?

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

Solution

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 adding the content, re-compile the project APK.

3.4. Configure SDK protection

After the SDK is configured in your app, configure SDK protection in the Anti-Bot console to specify the path and version of the app that you want to protect.

Perform the following steps to integrate the SDK:

1. Integrate the SDK into the app. For more information, see the iOS SDK integration guide and the Android SDK integration guide.
2. Configure the path of the app to be protected in the Anti-Bot console. For more information, see Configure path protection.
3. Send a test request by using the SDK-integrated app, and analyze the debugging errors and exceptions based on the response and log until SDK integration is verified to be correct.
4. Publish the new version of the app integrated with the SDK, and enable protection in the Anti-Bot console. For more information, see Enable app protection.

> **Note** We recommend that you perform an upgrade when publishing the new app version. Otherwise, the previous app version still contains security threats.

Configure path protection

Configure path protection to specify the address that you want to protect, and generate protection rules for the address.

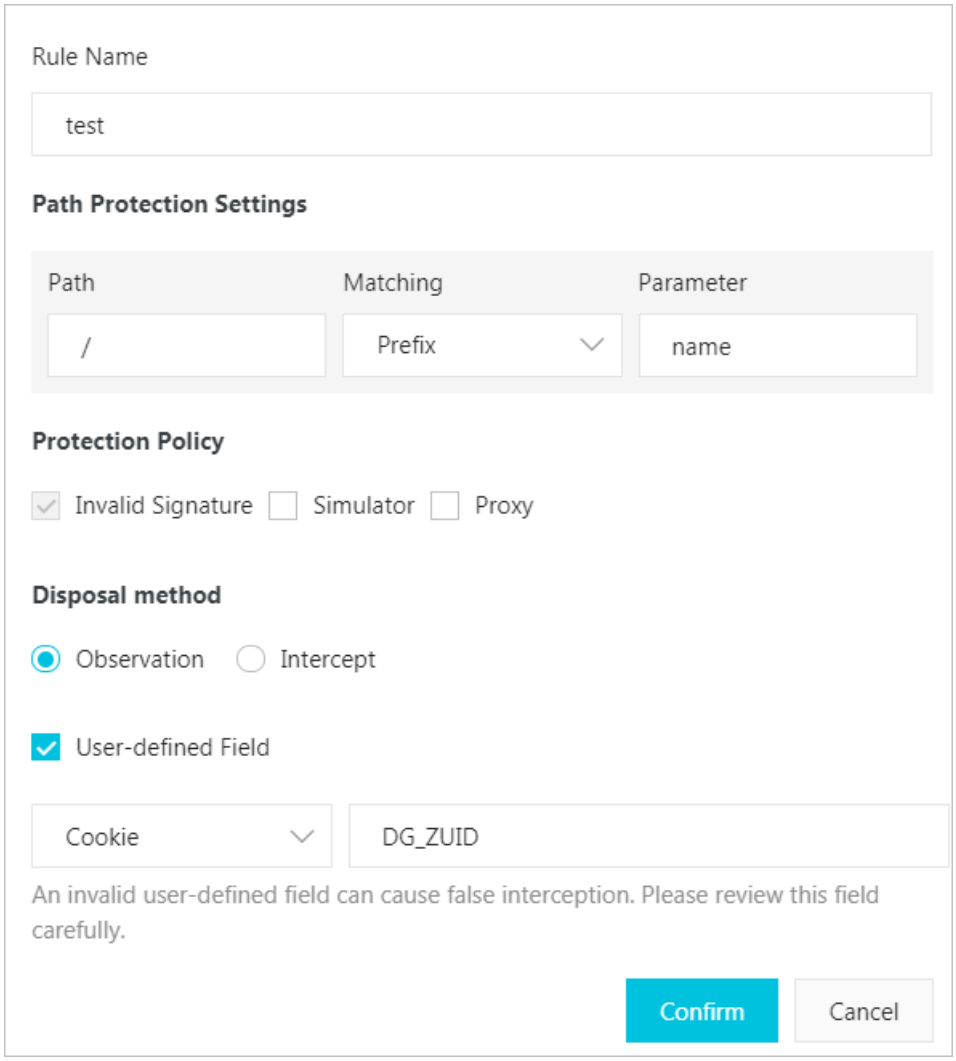
Procedure

1. Log on to the Anti-Bot console.
2. Choose **Protection > App Protection**. Select the domain name that you want to configure.

3. Click **Add** under **Interface Protection**.
4. In the **Add Path Rule** dialog box, configure the following settings.

> **Note** We recommend that you set full-path protection (prefix matching "/") and set Rule Action to Monitor (or set to Intercept during domain name testing). This allows debugging without affecting online services.

Configuration item	Description
Rule Name	(Required) The name of the rule.
Protected path configuration	◦ **Path**: (Required) The path that you want to protect. Use a slash (/) to indicate a full path. > **Note** Signature verification may fail when the body length of a POST request exceeds 8 KB. We recommend that you disable SDK protection for APIs that do not require protection (such as the APIs used to upload large images). If you do need to enable SDK protection, contact Alibaba Cloud engineers through DingTalk. ◦ **Matching**: The options **Prefix Match** and **Exact** are supported. In prefix match mode, all the APIs in the specified path are matched. In exact match mode, only the specified path is matched. ◦ **Parameter**: This specifies the parameter content to be matched when the protected path contains invariable parameters, to locate APIs more accurately. The parameter content follows the question mark in the requested address. Example: Assume that the protected URL contains `domain name/? action =login&name=test`. You can set **Path** to "/" and **Matching** to "Prefix Match", and enter "name", "login", "name=test", or "action=login" in **Parameter**.
Protection policy	◦ **Invalid Signature**: This is selected by default and cannot be cleared. The system checks whether the request signature of the request sent to the specified path is correct. This policy is matched if the signature is incorrect. ◦ **Simulator**: If this is selected, the system checks whether users initiate requests to the specified path by using a simulator. We recommend that you select this option. This policy is matched if a simulator is used. ◦ **Proxy**: If this is selected, the system checks whether users initiate requests to the specified path by using a proxy tool. We recommend that you select this option. This policy is matched if a proxy tool is used.
Rule Action	The action to be taken for users who hit the protection policy. ◦ **Monitor**: The system only records logs, but does not block requests. ◦ **Block**: The system blocks requests and returns Status Code 405.



5. Click **OK** to add the protection rule.

   You can **modify** and **delete** the added rules.

(Optional) Configure version protection

You can configure version protection to intercept requests from non-official apps. You can configure a version protection policy to verify app validity.

> **Note** A version protection policy is required only when you need to verify app validity.

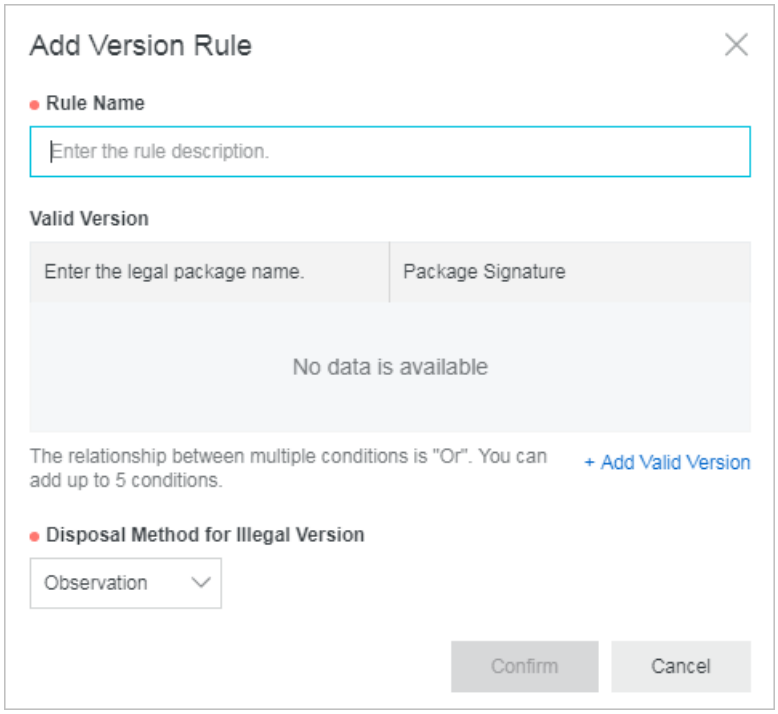
Procedure

1. Log on to the Anti-Bot console.
2. Choose **Protection > App Protection**. Select the domain name that you want to configure.
3. Select **Allow Specified Version Requests** under **Version Protection**.

   > **Note** To disable version protection, turn off **Allow Specified Version Requests**.

4. In the **Add Version Rule** dialog box, configure the following settings.

Configuration item	Description
Rule Name	The name of the rule.
Valid Version	◦ **Valid Package Name**: (Required) The valid app package name, such as *com.aliyundemo.example*. ◦ **Package Signature**: For this value, contact the related security technical engineer of Alibaba Cloud. **Note** Do not enter the app certificate signature here. **Note** Set Package Signature only when you need to verify the corresponding app package signature. In this case, the system only verifies the configured valid app package name. Click **Add Valid Version** to add up to five version records. The package names must be different. Currently, the system does not distinguish between iOS and Android. You can enter valid records to match multiple package names.
Invalid version processing	◦ **Monitor**: The system only records logs, but does not block requests. ◦ **Block**: The system blocks requests and returns Status Code 405.



5. Click **OK** to add the rule.

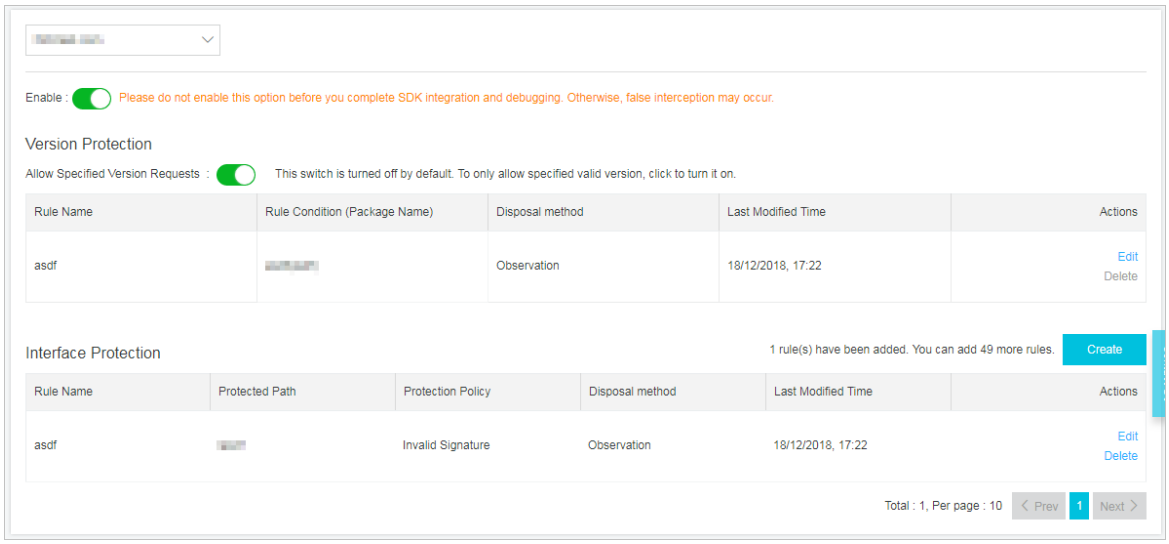
   You can **modify** the added rules.

Enable app protection

After verifying through debugging that the app is correctly integrated with the SDK and that the new app version is published, you need to enable app protection to put the protection configuration into effect.

1. Log on to the Anti-Bot console.
2. Choose **Protection > App Protection**. Select the domain name that you want to configure.
3. Turn on **Enable**.

> **Notice** Do not enable the Block mode for the domain names in the production environment before the SDK is integrated or debugging is completed. Otherwise, valid requests may be intercepted because the SDK is not correctly integrated. You can enable the Monitor mode during access testing to debug SDK integration based on logs.



More information

Scan the QR code through DingTalk to join the technical support group. Consult a security expert in the group if you have any technical or urgent problems when using Anti-Bot.

> **Note** Visit the DingTalk website, and download and install DingTalk.

4.Real-time log query and analysis

4.1. Enable Log Service for Anti-Bot

Log Service can collect access logs and protection logs from the websites protected by Anti-Bot in real time. Also, it can retrieve and analyze the collected log data in real time.

You can analyze the website access and attack behaviors based on the website logs collected in the Anti-Bot console in real time. This further allows you to assist your security management personnel in developing protection policies.

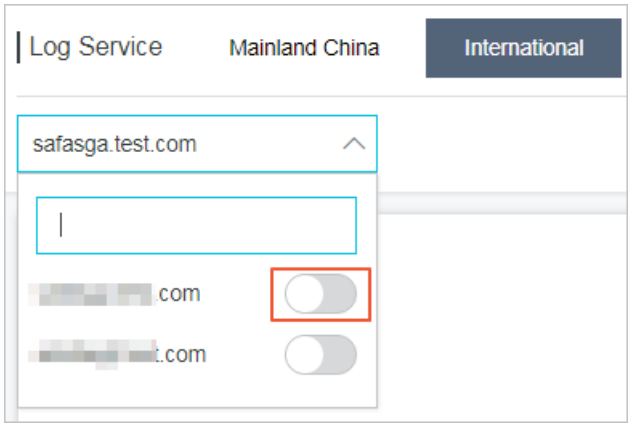
Procedure

1. Log on to the Anti-Bot console.
2. Choose **Reports > Log Service**. Select the region where your instance is located.

   > **Note** If you are using Log Service for Anti-Bot for the first time, click **Authorize** to authorize Anti-Bot to store all the recorded logs in your logstore as instructed.

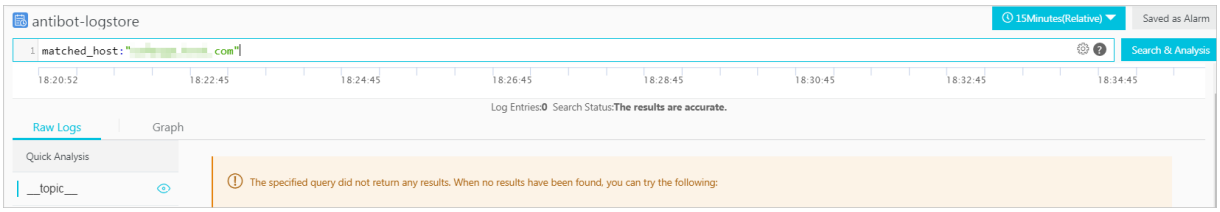
3. From the Website Domain drop-down list, select the website domain name for which you want to enable Log Service. Then, click Enable.

   > **Note** The Website Domain drop-down list displays all the website domain names configured with Anti-Bot.



Now, Log Service has been enabled for the website domain name. Log Service automatically creates a dedicated logstore for your Alibaba Cloud account. Anti-Bot automatically imports the logs of all the website domain names enabled with Log Service to the dedicated logstore in real time.

Then, you can retrieve and analyze the access logs of the website domain names enabled with Log Service.



Restrictions and instructions

- Other data cannot be written to the dedicated logstore.

  > **Note** The website logs recorded by Anti-Bot are stored in the dedicated logstore, where you cannot write other data through APIs and SDKs.

- Currently, the basic settings (such as the storage period) of the dedicated logstore cannot be modified.
- Do not delete or modify the default settings created by Log Service, such as the default project, logstore, index, and dashboard.
- Log Service updates and upgrades the log query analysis function from time to time. The indexes and default reports of the dedicated logstore will be automatically updated.
- If the RAM user requires the log query analysis function, grant the related Log Service permissions to the RAM user through RAM.

4.2. Common statements used for log query and analytics

After you enable Anti-Bot Service for a domain name, you can run statements to query and analyze requests sent to the domain, and logs about attack and defense events. This topic provides the commonly used statements for log query and analytics. You can write statements to query log data based on your actual needs.

> **Note** You can change the value of `limit` to specify the number of entries to be returned. For example, `limit 10` indicates that 10 log entries is returned. If you do not specify the value of `limit`, the system returns the first 100 entries.

Query information about access requests to a website

- Query inbound traffic

```
host:example.com | SELECT
date_format(from_unixtime(__time__ - __time__% 600), '%H:%i') as dt,
round(sum(request_length)/1024.0/600, 2) as "inbound traffic (KB/s)", round(sum(if((block_action <> ''),
request_length, 0))/1024.0/600, 2) as "attack traffic (KB/s)"
group by __time__ - __time__% 600 order by dt limit 10000
```

- Query outbound traffic

```
host:example.com | SELECT
date_format(from_unixtime(__time__ - __time__% 600), '%H:%i') as dt,
round(sum(body_bytes_sent)/1024.0/600, 2) as "outbound traffic (KB/s)", round(sum(if((block_action <> '
'),
body_bytes_sent, 0))/1024.0/600, 2) as "attack traffic (KB/s)"
group by __time__ - __time__% 600 order by dt limit 10000
```

- Query peak request rate

```
host:example.com |SELECT COUNT(*) as c,date_trunc('second', __time__) as s GROUP by s
order by c  desc limit 1
```

- Query the number of requests per minute in the last 10 minutes (in descending order by time)

```
host:example.com |SELECT COUNT(*) as c,date_trunc('minute', __time__) as minute GROUP by s  order by
minute desc limit 10
```

- Query the top 10 client IP addresses that visit the website most frequently

```
host:example.com |SELECT real_client_ip,COUNT(*) as c group by real_client_ip order by c desc limit 10
```

- Query the top 10 most visited URLs

```
host:example.com |SELECT request_path,COUNT(*) as c group by request_path order by c desc limit 10
```

- Query HTTP status codes

> **Note** HTTP status codes help you determine whether your website services run properly.

```
host:example.com |SELECT status, upstream_status,COUNT(*) as c GROUP by status, upstream_status or
der by c desc limit 10
```

Query information about website protection

- Query the top 10 client IP addresses that visit a specified URL or endpoint most frequently

> **Note** Malicious IP addresses are typically highly ranked when attackers start attacks to your website from these IP addresses.

```
host:example.com and request_path:/login.php |SELECT real_client_ip,COUNT(*) as c group by real_client
_ip order by c desc
limit 10
```

- Query URLs visited by a specified IP address

> **Note** Malicious IP addresses that start HTTP flood attacks typically target certain URLs or endpoints.

```
host:example.com and real_client_ip:1.2.3.4 |SELECT request_path,COUNT(*) as c group by request_path
order by c desc limit
10
```

- Query IDs of protection policies that are hit by requests from specified client IP addresses

```
host:example.com and real_client_ip:1.2.3.4 |SELECT antibot,antibot_rule,COUNT(*) as c GROUP by antib
ot,antibot_rule  order by c desc limit 10
```

- Query whether specified protection policies are hit

> **Note** You can study the protection effects and hit rate of the protection policies based on the query results.

```
host:example.com and antibot_rule:1234 |SELECT real_client_ip,COUNT(*) as c GROUP by real_client_ip o
rder by c
desc limit 10
```

- Query the signature authentication status of the SDK that is used to enhance protection

```
host:taobao.com |SELECT wxbb_invalid_wua,COUNT(*) as c GROUP by wxbb_invalid_wua order by c desc
limit 10
```

4.3. Log field description

Log Service for Anti-Bot Service (Anti-Bot) records the access logs and attack and defense logs of protected website domain names in detail. A log contains dozens of fields. You can select specific fields for query analysis as needed.

Field	Description	Example
__topic__	The log topic. This field is invariably set to antibot_access_log.	antibot_access_log
antibot	The type of the triggered Anti-Bot protection policy, including: • ratelimit: rate limiting • sdk: app protection • algorithm: algorithm pattern • intelligence: bot intelligence • acl: access control list • blacklist: blacklist	ratelimit
antibot_action	The operation specified by the Anti-Bot protection policy, including: • *challenge*: Deliver a JavaScript script for verification • *drop*: Intercept • *captcha*: Verify by dragging a slider • *report*: Monitor only	drop
antibot_rule	The ID of the triggered Anti-Bot protection rule.	5472

Field	Description	Example
antibot_verify	The result of the verification performed by Anti-Bot. **Note** This value is recorded when the antibot_action field is set to challenge or captcha. • challenge_fail: JavaScript verification fails. • challenge_pass: JavaScript verification is passed. • captcha_fail: Slide captcha verification fails. • captcha_pass: Slide captcha verification is passed.	challenge_fail
block_action	The type of the bot protection that is triggered. The value is invariably set to *antibot*.	antibot
body_bytes_sent	The size of HTTP body (in byte) sent to the client.	2
content_type	The content type of the access request.	application/x-www-form-urlencoded
host	The source website.	api.aliyun.com
http_cookie	The cookie information about the access client, which is included in the access request header.	k1=v1;k2=v2
http_referer	The source URL of the access request, which is included in the access request header. `-` is displayed if no source URL is available.	http://xyz.com
http_user_agent	The User Agent field in the access request header, which typically includes the web browser identifier and operating system identifier of the source client.	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build/HUAWEIEDISON-AL10)
http_x_forwarded_for	The XFF header information in the access request header, which is used to identify the original IP addresses of the clients connected to a web server through the HTTP proxy or SLB.	-

Field	Description	Example
https	Whether the access request is an HTTPS request. Valid values: • true: The access request is an HTTPS request. • false: The access request is an HTTP request.	true
matched_host	The matched domain name configured with Anti-Bot, which may be a wildcard domain name. `-` is displayed if no related domain name configuration is matched.	*.aliyun.com
real_client_ip	The actual IP address of the access client. `-` is displayed if no actual IP address is retrieved.	1.2.3.4
region	The information about the region where the Anti-Bot instance is located.	cn
remote_addr	The IP address of the client that initiates the access request.	1.2.3.4
remote_port	The port of the client that initiates the access request.	23713
request_length	The length of the access request. Unit: bytes.	123
request_method	The HTTP request method of the access request.	GET
request_path	The relative path of the request (excluding the query string).	/news/search.php
request_time_msec	The duration of the access request. Unit: milliseconds.	44
request_traceid	The unique ID of the access request.	7837b11715410386943437009ea1f0
server_protocol	The protocol and version of the response returned by the origin server.	HTTP/1.1
status	The status of the HTTP response that Anti-Bot returns to the client.	200
time	The occurrence time of the access request.	2018-05-02T16:03:59+08:00
ua_browser	The information about the web browser that initiates the access request.	ie9

Field	Description	Example
ua_browser_family	The family of the web browser that initiates the access request.	internet explorer
ua_browser_type	The type of the web browser that initiates the access request.	web_browser
ua_browser_version	The version of the web browser that initiates the access request.	9.0
ua_device_type	The device type of the client that initiates the access request.	computer
ua_os	The operating system of the client that initiates the access request.	windows_7
ua_os_family	The operating system family of the client that initiates the access request.	windows
upstream_addr	The origin address list of Anti-Bot in the format of **IP address:Port**. Separate multiple IP addresses with commas (,).	1.2.3.4:443
upstream_ip	The origin IP address corresponding to the access request. For example, if Anti-Bot forwards the access request to an ECS instance, this parameter returns the IP address of the back-to-origin ECS instance.	1.2.3.4
upstream_response_time	The time for the origin server to respond to an Anti-Bot request. Unit: seconds. The response times out if "-" is returned.	0.044
upstream_status	The status of the response that the origin server returns to Anti-Bot. No response is available if "-" is returned. For example, the request is intercepted by Anti-Bot, or the response returned by the origin server times out.	200
user_id	AliUID of the Alibaba Cloud account.	12345678

Field	Description	Example
wxbb_action	If the protection type of Anti-Bot is app protection, the following actions are supported: • *close*: intercepts requests. That is, the antibot_action field is set to *drop*. • *test*: only monitors requests. That is, the antibot_action field is set to *report*. **Note** This field is set to - if SDK protection is not configured.	close
wxbb_invalid_wua	For more information about app protection, consult your technical engineer.	valid wua