ALIBABA CLOUD

阿里云

爬虫风险管理

用户指南

文档版本：20210129

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.接入配置

1.1. 同时部署Anti-Bot和DDoS高防

爬虫风险管理（Anti-Bot Service，简称Anti-Bot）与DDoS高防IP服务完全兼容。您可以参照以下架构为源站同时部署Anti-Bot和DDoS高防IP：DDoS高防IP（入口层，实现DDoS防护）> Anti-Bot（中间层，实现应用层爬虫风险管理防护）> 源站。

操作步骤

1. 在爬虫风险管理控制台中添加网站配置。
   - **服务器地址**：勾选**IP**并填写ECS公网IP、SLB公网IP，或云外机房服务器的IP。
   - **Anti-Bot前是否有七层代理（高防/CDN等）**：勾选**是**。

   具体操作请参考添加域名配置。
2. 在高防IP管理控制台中添加网站配置。操作步骤如下：
   i. 在**接入 > 网站**页面，单击**添加域名**。
   ii. 在**填写域名信息**任务中， 完成以下配置：
   - **防护网站**：填写被防护网站的域名。
   - **协议类型**：勾选源站支持的协议类型。
   - **源站IP/域名**：勾选**源站域名**并填写爬虫风险管理生成的CNAME地址。

   > **说明** 关于如何查看爬虫风险管理生成的CNAME地址，请参考查看爬虫风险管理分配的CNAME地址。

   iii. 单击**下一步**。
   iv. 完成任务**选择实例与线路**。

3. 变更域名的DNS解析。登录域名的DNS系统，添加一条CNAME记录，将网站域名的解析地址指向DDoS高防生成的CNAME地址。

   具体操作请参考DDoS高防CNAME接入流程。

执行结果

完成上述配置后，网站流量先经过DDoS高防，再转发到爬虫风险管理实现防护。

1.2. 同时部署Anti-Bot和CDN

爬虫风险管理（Anti-Bot Service，简称Anti-Bot）可以与CDN（如网宿、加速乐、七牛、又拍、阿里云CDN等）结合使用，为开启内容加速的域名提供恶意爬虫防御功能。您可以参照以下架构为源站同时部署Anti-Bot和CDN：CDN（入口层，内容加速）> Anti-Bot（中间层，实现应用层爬虫风险管理防护）> 源站。

使用阿里云CDN

以阿里云CDN为例。参照以下步骤，为您的网站同时部署爬虫风险管理和CDN：

1. 参考CDN快速入门，将要防护的域名（即加速域名）接入CDN。
2. 在爬虫风险管理控制台中创建网站配置。
   - **域名**：填写要防护的域名。
   - **服务器地址**：填写SLB公网IP、ECS公网IP，或云外机房服务器的IP。
   - **Anti-Bot前是否有七层代理（高防/CDN等）**：勾选**是**。

   具体操作请参考添加域名配置。
3. 成功创建网站配置后，爬虫风险管理为该域名生成一个专用的CNAME地址。

   > **说明** 关于如何查看Anti-Bot生成的CNAME地址，请参考查看爬虫风险管理分配的CNAME地址。

4. 将CDN配置中的源站修改为爬虫风险管理分配的CNAME地址，操作步骤如下：
   i. 登录阿里云CDN控制台。
   ii. 在**域名管理**页面，选择要操作的域名，单击**配置**。
   iii. 在**源站设置**下，单击**修改配置**。

iv. 修改源站信息。

- **类型**：选择**源站域名**。
- **源站地址 域名**：填写Anti-Bot生成的CNAME地址。
- **协议跟随回源**：开启。

v. 在**回源设置**下，确认**回源host**未开启。

完成上述配置后，流量经过CDN，其中动态内容将继续通过爬虫风险管理进行安全检测防护。

使用非阿里云CDN

1. 配置CDN，将域名接入CDN。
2. 在爬虫风险管理控制台中创建网站配置。具体请参考添加域名配置。
3. 查看爬虫风险管理分配的CNAME地址。
4. 将CDN配置的源站改为Anti-Bot所分配的CNAME地址。

1.3. 获取访问者真实IP

在大部分实际业务场景中，网站访问请求并不是简单地从用户（访问者）的浏览器直达网站的源站服务器，中间可能经过所部署的CDN、高防IP、WAF、Anti-Bot等代理服务器。例如，网站可能采用这样的部署架构：用户 > CDN/高防IP/Anti-Bot > 源站服务器。这种情况下，访问请求在经过多层加速或代理转发后，源站服务器该如何获取发起请求的真实客户端IP？

一般情况下，透明的代理服务器在将用户的访问请求转发到下一环节的服务器时，会在HTTP的请求头中添加一条**X-Forwarded-For**记录，用于记录用户的真实IP，其记录格式为 `X-Forwarded-For:用户IP` 。如果期间经历多个代理服务器，则**X-Forwarded-For**将以该格式记录用户真实IP和所经过的代理服务器IP： `X-Forwarded-For:用户IP, 代理服务器1-IP, 代理服务器2-IP, 代理服务器3-IP, ……` 。

> **说明** 由于Anti-Bot与WAF采用同样的转发配置和设备，如果您的网站域名同时接入WAF和Anti-Bot实现应用层防护和恶意Bot流量拦截，X-Forwarded-For字段中将只记录一组代理服务器IP。

因此，常见的应用服务器可以**通过X-Forwarded-For的方式获取访问者真实IP**。

您可以根据您的应用服务器，选择相应的X-Forwarded-For配置方案来获取访问者真实IP。

> **注意** 开始配置前，请务必对现有环境进行备份，包括ECS快照备份和Web应用服务器配置文件备份。

1.4. 配置源站保护

正确配置源站ECS的安全组和SLB的白名单，可以防止黑客绕过Anti-Bot直接攻击您的源站IP。本文介绍了源站服务器保护的相关配置方法。

背景信息

> **说明** 源站保护不是必须的。没有配置源站保护并不会影响正常业务转发，但可能导致攻击者在您源站IP暴露的情况下，绕过爬虫风险管理直接攻击您的源站。

如何确认源站泄露

您可以在非阿里云环境直接使用Telnet工具连接源站公网IP地址的业务端口，观察是否建立连接成功。如果可以连通，表示源站存在泄露风险，一旦黑客获取到源站公网IP就可以绕过Anti-Bot直接访问；如果无法连通，则表示当前不存在源站泄露风险。

例如，测试已接入Anti-Bot防护的源站IP 80端口和8080端口是否能成功建立连接，测试结果显示端口可连通，则说明存在源站泄露风险。

注意事项

配置安全组存在一定风险。在配置源站保护前，请注意以下事项：

- 请确保该ECS或SLB实例上的所有网站域名都已经接入爬虫风险管理。
- 当爬虫风险管理集群出现故障时，可能会将域名访问请求旁路回源至源站，确保网站正常访问。这种情况下，如果源站已配置安全组防护，则可能会导致源站无法从公网访问。
- 当爬虫风险管理集群扩容新的回源网段时，如果源站已配置安全组防护，可能会导致频繁出现5xx错误。

操作步骤

1. 登录爬虫风险管理控制台，选择您的Anti-Bot实例所在的地区。

2. 定位到**域名接入**页面，单击**爬虫风险管理回源IP网段列表**，查看爬虫风险管理的所有回源IP段。

> **说明** Anti-Bot回源IP网段会定期更新，请关注定期变更通知。及时将更新后的回源IP网段添加至相应的安全组规则中，避免出现误拦截。

3. 在**回源IP段**对话框，单击**复制**即可复制所有回源IP。
4. 配置源站只允许Anti-Bot回源IP进行访问。
   - **源站是ECS**
     a. 前往ECS 实例列表，定位到需要配置安全组的ECS实例，单击其操作列下的**管理**。
     b. 切换到**本实例安全组**页面。
     c. 选择目标安全组，并单击其操作列下的**配置规则**。
     d. 单击**添加安全组规则**，并配置如下安全组规则：

> **说明** 安全组规则授权对象支持输入“10.x.x.x/32”格式的IP网段，且支持添加多组授权对象（以“,”隔开），最多支持添加10组授权对象。

     - **网卡类型**：内网

> **说明** 如果ECS实例的网络类型为经典网络，则网卡类型需设置为公网。

     - **规则方向**：入方向
     - **授权策略**：允许
     - **协议类型**：TCP
     - **授权类型**：地址段访问
     - **端口范围**：80/443
     - **授权对象**：粘贴步骤3中复制的所有爬虫风险管理回源IP段
     - **优先级**：1

e. 为所有爬虫风险管理回源IP段添加安全组规则后，再添加如下安全组规则，拒绝公网入方向的所有IP段访问，优先级为100。

- **网卡类型**：内网

  > **说明** 如果ECS实例的网络类型为经典网络，则网卡类型需设置为公网。

- **规则方向**：入方向
- **授权策略**：拒绝
- **协议类型**：TCP
- **端口范围**：80/443
- **授权类型**：地址段访问
- **授权对象**：0.0.0.0/0
- **优先级**：100

> **说明** 如果本安全组防护的服务器还与其他的IP或应用存在交互，需要将这些交互的IP和端口通过安全组一并加白放行，或者在最后添加一条优先级最低的全端口放行策略。

- **源站是SLB**

  通过类似的方式，将爬虫风险管理的回源IP加入相应负载均衡实例的白名单，具体设置方法请参考开启访问控制。

  a. 登录负载均衡管理控制台，前往**访问控制**页面，单击**创建访问控制策略组**。

  b. 填写策略组名称，添加Anti-Bot回源IP网段，单击**确定**。

  c. 在**实例管理**页面，选择相应的负载均衡实例。

  d. 在**监听**页签中，选择端口监听记录，单击**更多 > 设置访问控制**。

  e. 启用访问控制，选择访问控制方式为白名单，并选择所创建的Anti-Bot回源IP网段的访问控制策略组，单击**确定**。

后续步骤

源站保护配置完成后，您可以通过测试已接入Anti-Bot防护的源站IP80端口和8080端口是否能成功建立连接验证配置是否生效。如果显示端口无法直接连通，但网站业务仍可正常访问，则表示源站保护配置成功。

2.防护配置

2.1. 防护总览

当您成功接入防护域名后，可以针对指定域名设置不同的防护配置，通过具体的防护规则过滤恶意爬虫流量。

操作步骤

1. 登录爬虫风险管理控制台，选择您的Anti-Bot实例所在的地区。
2. 定位到**防护配置 > 防护总览**页面，选择已接入防护的网站域名。

> **说明** 您也可以在**域名接入**页面，选择已接入防护的网站域名，单击**防护规则**跳转至**防护总览**页面。

3. 选择防护策略，单击**配置**，配置具体防护规则。

- **黑白名单**：根据指定IP名单放过或阻断来自该IP的Bot流量。具体规则配置说明，查看黑白名单。
- **精准访问控制**：通过对常见的HTTP字段（如IP、URL、Referer、UA、参数等）进行条件组合，配置适用于定制化业务场景的防护策略。具体规则配置说明，查看精准访问控制。
- **频次限制**：对特定URL路径上单个IP/Cookie/Header字段对服务器的访问频率进行限制，也可以基于响应码的比例和数量进行限制。具体规则配置说明，查看频次限制。
- **APP增强防护**：针对原生App端提供可信通信、防机器脚本滥刷等安全防护，有效识别代理、模拟器、非法签名的请求。配置APP增强防护需要在您的App端中集成爬虫风险管理的SDK，具体配置方案查看App增强防护方案。
- **合法爬虫**：通过提供合法搜索引擎爬虫白名单（例如Google、Bing、百度、搜狗、360、Yandex等），可在全域名或指定路径下放行相关搜索引擎的合法爬虫的访问请求。具体配置方案，查看爬虫情报。
- **威胁情报**：基于云平台强大的计算能力，提供拨号池IP、IDC机房IP、恶意扫描工具IP和通过云端实时模型生成的恶意爬虫库等多种纬度的威胁情报，可在全域名或定路径下根据威胁情报库阻断恶意爬虫的访问请求。具体配置方案，查看爬虫情报。

2.2. 黑白名单

通过为指定域名设置IP黑白名单，直接放行或阻断来自黑白名单中IP的爬虫流量。

背景信息

黑白名单策略在所有防护策略中拥有最高优先级，即来自黑白名单中IP的请求将被直接阻断或放行。其中，白名单策略的优先级高于黑名单，即如果一个IP同时配置在黑、白名单中，白名单策略生效，最终来自该IP的请求将被放行。

操作步骤

1. 登录爬虫风险管理控制台，选择您的Anti-Bot实例所在的地区。
2. 定位到**防护配置 > 黑白名单**页面，选择已接入防护的域名。
3. 单击**是否生效**开关

   ，开启黑白名单防护策略。
4. 在**IP黑名单**或**IP白名单**文本框中，添加您想要直接放行或阻断的IP或IP段，单击**保存**。

   > **说明** 支持输入IP或IP段（IP/掩码），添加多个IP时以英文逗号“,”分隔。

   保存成功后，所添加的IP黑白名单即生效。

2.3. 精准访问控制

通过精准访问控制策略，您可以根据您的业务场景定制访问控制规则。通过将常见的HTTP字段（如IP、URL、Referer、UA、参数等）进行组合形成匹配条件，筛选网站域名的访问请求，并对命中条件的访问请求配置观察、阻断、滑块或放行等处置动作。

背景信息

规则说明

精准访问控制规则由**规则条件**与**处置动作**构成。在创建规则时，您通过设置匹配字段、逻辑符和相应的匹配内容定义匹配条件，并针对符合匹配条件规则的访问请求定义相应的处置动作。

- **规则条件**

  规则条件包含匹配字段、逻辑符、匹配内容三个要素。

匹配字段	字段描述	适用逻辑符
URL	访问请求的URL地址。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于
IP	访问请求的来源IP。	○ 属于 ○ 不属于
Referer	访问请求的来源网址，即该访问请求是从哪个页面跳转产生的。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于 ○ 不存在
User-Agent	发起访问请求的客户端的浏览器标识、渲染引擎标识和版本信息等浏览器相关信息。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于
Params	访问请求的URL地址中的参数部分，通常指URL中”?”后面的部分。例如，`www.abc.com/index.html?action=login`中的`action=login`就是参数部分。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于

匹配字段	字段描述	适用逻辑符
Cookie	访问请求中的Cookie信息。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于 ○ 不存在
Content-Type	访问请求指定的响应HTTP内容类型，即MIME类型信息。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于
Content-Length	访问请求的响应内容所包含的字节数。	○ 值小于 ○ 值等于 ○ 值大于
X-Forwarded-For	访问请求的客户端真实IP。 **说明** X-Forwarded-For（XFF）用来识别通过HTTP代理或负载均衡方式转发的访问请求的客户端最原始的IP地址的HTTP请求头字段，只有通过HTTP代理或者负载均衡服务器转发的访问请求才会包含该项。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于 ○ 不存在
Post-Body	访问请求的响应内容信息。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于
Http-Method	访问请求的方法，如GET、POST等。	○ 等于 ○ 不等于

匹配字段	字段描述	适用逻辑符
Header	访问请求的头部信息，用于自定义HTTP头部字段。	○ 包含 ○ 不包含 ○ 等于 ○ 不等于 ○ 长度小于 ○ 长度等于 ○ 长度大于 ○ 不存在

- **处置动作**

  精准访问控制规则支持以下匹配动作：

  - **阻断**： 符合匹配条件的访问请求将被直接阻断。
  - **放行**： 符合匹配条件的访问请求将被放行。
  - **观察**： 符合匹配条件的访问请求将被放行，同时可以在数据报表中查看到匹配规则的请求，用于观察精准访问控制规则的效果。
  - **滑块**： 符合匹配条件的访问请求将给客户端发送滑块验证进行二次人机识别验证。访问者需要滑动滑块完成验证，通过验证方可继续进行业务操作，无法通过（不操作或者验证为非人为操作）则业务操作将被中断。

- **规则匹配顺序**

  如果您设置了多条精准访问控制规则，则规则间存在先后匹配顺序，即访问请求将根据您设定的精准访问控制规则顺序依次进行匹配，顺序靠前的精准访问控制规则优先进行匹配。当访问请求满足规则匹配条件后，将依此规则对访问请求执行处置动作，且不再继续匹配其他精准访问控制规则；若未满足匹配条件，则按顺序继续匹配下一条精准访问控制规则。

  您可以通过规则排序功能对所有精准访问控制规则进行排序，以获得最优的防护效果。

默认规则说明

精准访问控制规则中包含一条系统默认规则：

- **规则条件**：所有未命中以上规则的请求。即当您启用精准访问控制策略后，所有未命中您所设置的精准访问控制规则的访问请求都将执行该默认规则的处置动作。
- **处置动作**：默认处置动作为放行且后续继续执行其他安全策略，即所有未命中您所设置的精准访问控制规则的访问请求将被放行并继续执行其他防护策略（频次控制、APP端防护）。

> **说明** 默认规则不可删除，且其匹配条件不可修改。默认规则的排序不可调整，永远在最后执行匹配。

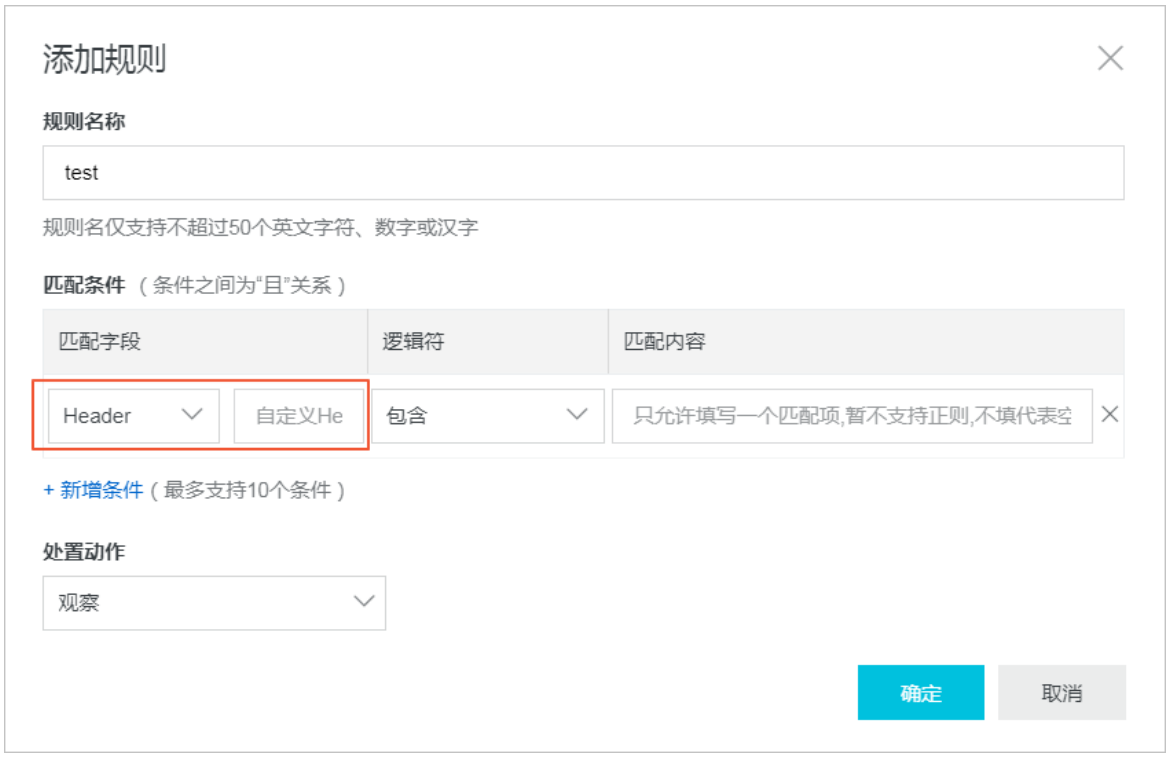
操作步骤

1. 登录爬虫风险管理控制台，选择您的Anti-Bot实例所在的地区。
2. 定位到**防护配置 > 精准访问控制**页面，选择已接入防护的域名。
3. 单击**生效状态**开关

，开启精准访问控制防护策略。

4. 单击**添加**，设置规则的匹配条件和相应的处置动作，完成后单击**确定**。

> **说明** 当选择Header作为匹配字段时，需要额外设置自定义Header中的key字段。例如，Header字段中带有业务含义的字段为 `userid=xxxxxx` ，则需要在自定义Header字段中填写 `userid` ，即可将userid字段的作为匹配条件。



规则添加成功后，您可以**编辑**其内容或将其**删除**。如果已添加多条规则，在**精准访问控制**页面单击**排序**，通过**上移**、**下移**、**置顶**、**置底**等操作可调整规则的匹配顺序，越靠上的规则越优先匹配。规则顺序调整完成后，单击**保存排序**即可生效。

2.4. 频次限制

频次限制防护策略通过限制特定请求对象对指定URL路径的访问频率实现对恶意爬虫流量的拦截。同时，支持在访问频率中叠加特定响应码的数量或比例来限制该请求对象的访问请求。

背景信息

频次限制规则

频次限制规则由以下设置参数构成：

参数	描述
规则名称	指定该规则的名称，为了便于快速理解规则内容，建议设置有意义的规则名称。
URL	指定应用该规则的URL地址，例如 `/login.com` 。 **说明** 该字段不能为空。 URL地址设置支持以下匹配规则： • **精确匹配**：客户端所请求的URL地址必须与所设置的URL地址完全一致才会被该规则统计，所设置的URL地址必须以 `/` 开头。 • **前缀匹配**：即左包含匹配，只要客户端所请求的URL地址是以规则中所设置的URL地址开头即会被该规则统计，所设置的URL地址必须以 `/` 开头。例如，设置URL为 `/login` ，则请求 `/login.html` 地址的访问被该规则统计。 • **正则匹配**：以正则表达式进行匹配。在URL文本框中填写完整的正则表达式，客户端所请求的URL地址符合该正则表达式的访问将被该规则统计。 **说明** 支持填写包含参数的URL地址，例如 `/user?action=login` 。
统计对象	被该规则统计的主体。支持设置IP、默认Cookie、或者其它自定义Header、参数、cookie中的某个字段作为统计对象。 **说明** • 其中，默认Cookie是指正常的用户请求到达引擎时，系统自动添加的标记cookie，通常以 `acw_tc` 开头。 • **自定义字段的统计示例**：例如，业务在HTTP Header中以 `token: 123456` 标记用户身份，则可以将该自定义Header/token设置为统计对象进行频率统计。
统计时间	指定该规则统计请求次数的周期。

参数	描述
请求次数	指定在所设置的统计时间内，单个请求对象累计的请求次数上限。 **说明** 支持在请求次数限制的基础上叠加**响应码**条件（例如，“响应码503累计次数达到300次”，“响应码503所占的比例达到70%”）。即在所统计的请求对象的请求次数超过上限时，需要满足所设置的响应码条件才会触发处置动作。
处置动作	指定满足规则条件后触发的操作： • **观察**： 不会对请求触发任何动作，仅将该统计结果记录到匹配该规则的数据报表中，用于观察规则实际效果。 • **阻断**： 直接断开请求对象的连接。 • **JavaScript校验**： 使用重定向的方式向客户端发送验证要求，通过验证后才可继续业务操作。 • **滑块**： 通过向客户端发送滑块验证进行二次人机验证。访问者需要滑动滑块完成验证，通过验证方可继续进行业务操作，无法通过（不操作或者验证为非人为操作）则业务操作将被中断。 **说明** • 选择阻断处置动作时，可对请求对象设置阻断持续时长（即**拉黑时长**）。例如，设置拉黑时长为30min，则符合规则条件的IP请求访问在30分钟内将全部被阻断。 • 处置动作可选择对本域名全局请求生效或仅对符合当前规则URL的请求生效。

说明 由于需要将集群中的多台服务器的数据进行汇总来统计，统计过程中可能存在一定延时，因此频次限制规则的实际生效时间可能稍有滞后。

视频介绍

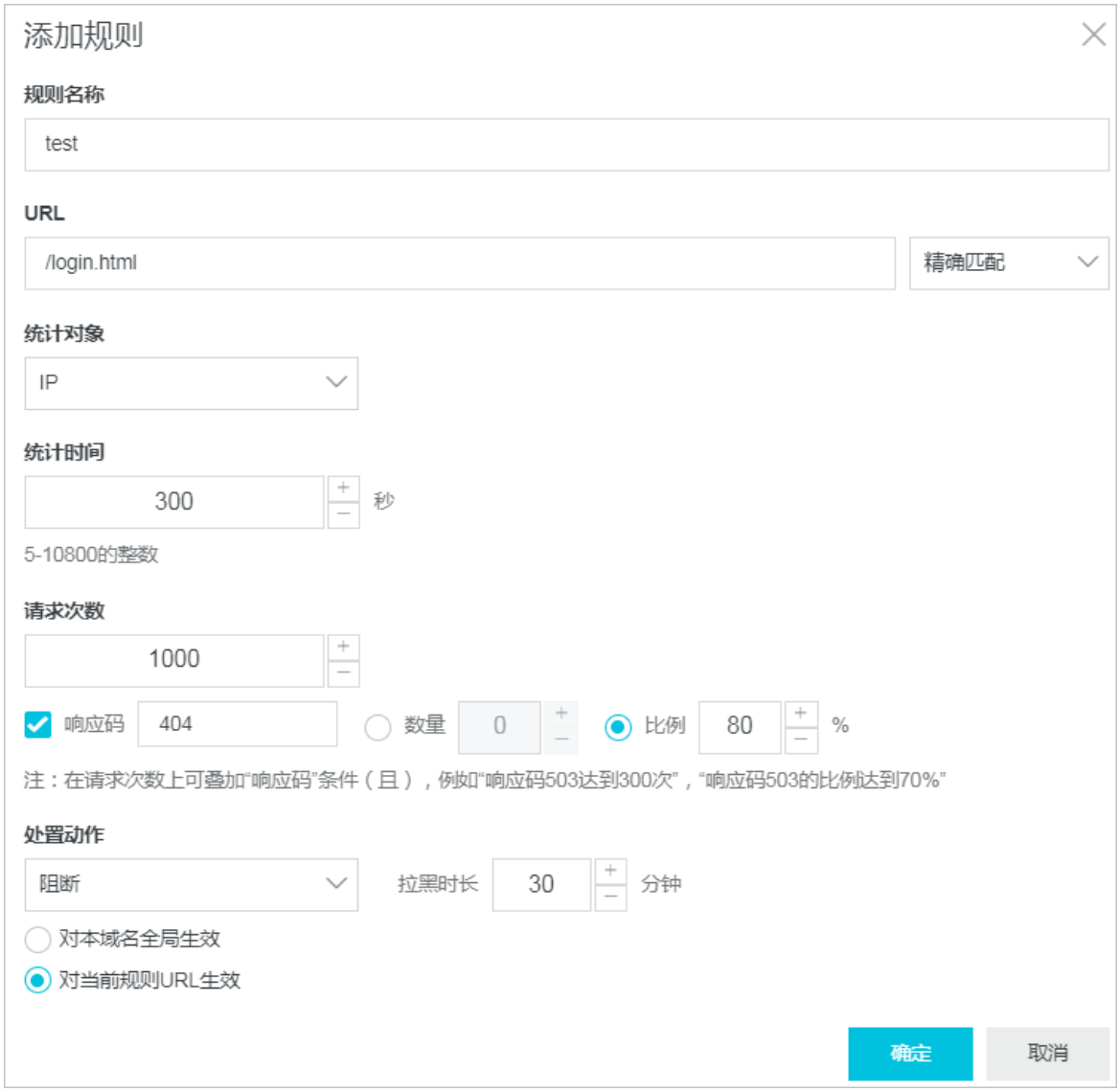
操作步骤

1. 登录爬虫风险管理控制台，选择您的Anti-Bot实例所在的地区。
2. 定位到**防护配置 > 频次限制**页面，选择已接入防护的网站域名。
3. 单击**是否生效**开关

，开启频次限制防护策略。

4. 单击**添加**，设置频次限制规则，完成后单击**确定**。例如，您可以配置当单个源IP在5分钟（300秒）内访

问 1.test.com/login.html 超过1000次，且响应码404的累计次数达到80%以上时进行阻断，持续封禁该IP 30分钟，仅对当前规则URL（1.test.com/login.html）生效。



2.5. 爬虫情报

爬虫情报规则依据阿里云爬虫情报库，帮助您直接放行合法爬虫请求并对来自威胁情报库的请求设防。

背景信息

阿里云爬虫情报库基于阿里云全网流量计算得出并可实时更新，涵盖以下访问请求来源的特征信息：

- 合法爬虫：主流搜索引擎的爬虫IP信息，可动态更新，目前包含Google、百度、搜狗、360、Bing、Yandex。

  启用合法爬虫规则后，来自相关搜索引擎的合法爬虫IP将被直接放行；此时您还可以使用黑白名单或精准访问控制规则进一步拦截来自于这些白名单IP的请求。

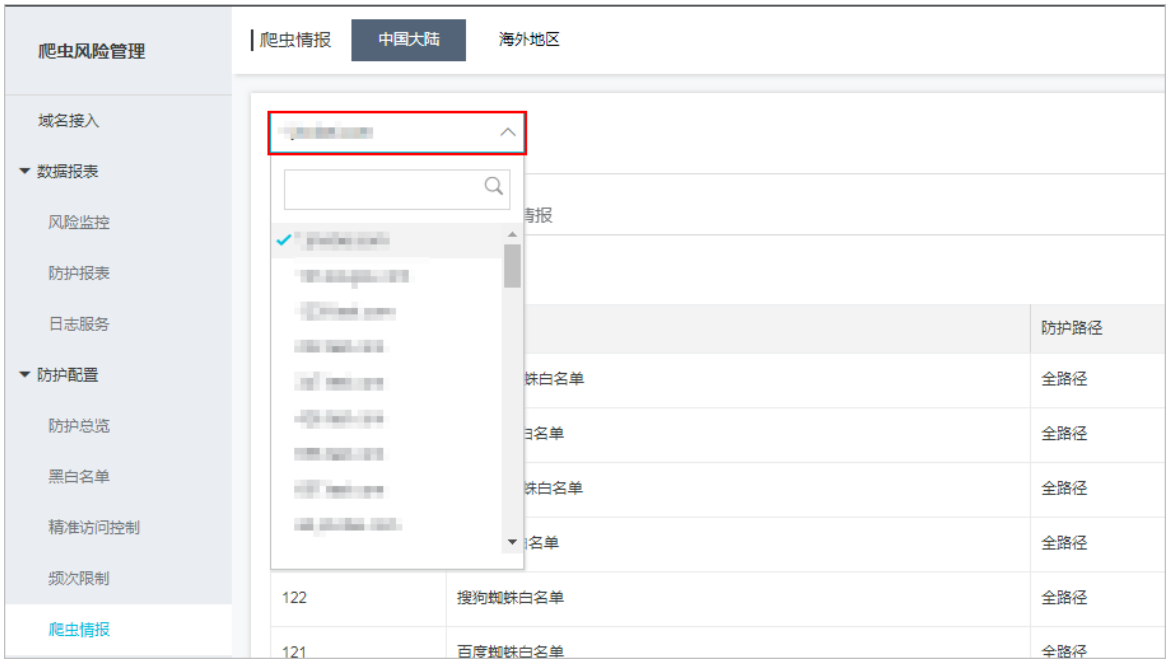
- 威胁情报：基于阿里云全网威胁情报实时计算得出的恶意爬虫IP情报库，以及动态更新的各大公有云/IDC IP库（很多爬虫程序会选择部署在云服务器上，而正常用户则很少通过公有云/IDC的源IP来访问您的业务）。

您可以设置威胁情报规则，针对不同的类型的黑名单IP选择不同的处置动作（如直接拦截、进行JavaScript校验、弹出滑块验证或观察）；也可以为某些关键接口配置针对特定类型黑名单IP的防护，以避免其他业务逻辑受到影响。

视频介绍

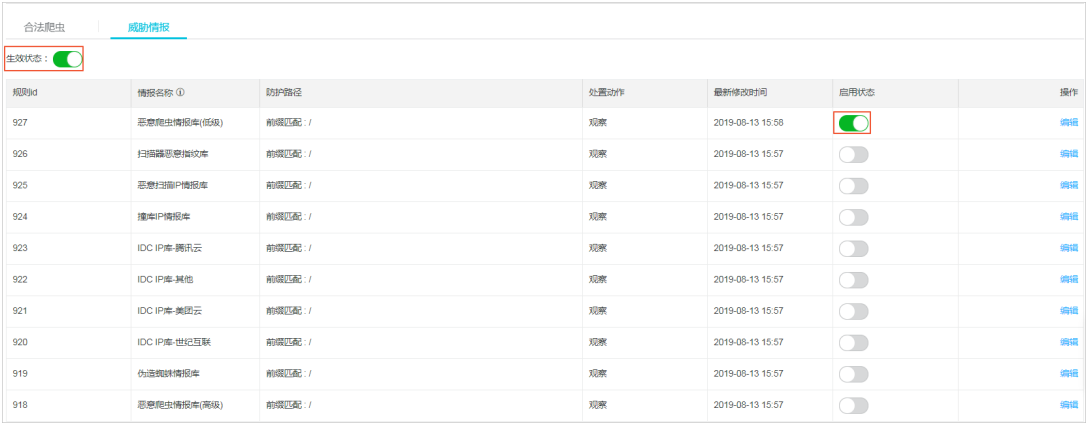
操作步骤

1. 登录爬虫风险管理控制台。
2. 在左侧导航栏，选择**防护配置 > 爬虫情报**。
3. 在域名下拉框中选择要配置的域名。



4. 分别在**合法爬虫**和**威胁情报**页签下完成相关配置。
   - 放行合法爬虫
     a. 在**合法爬虫**页签下，打开**生效状态**开关。

> **说明** 若您不再需要使用此功能，请在此页面关闭**生效状态**开关。



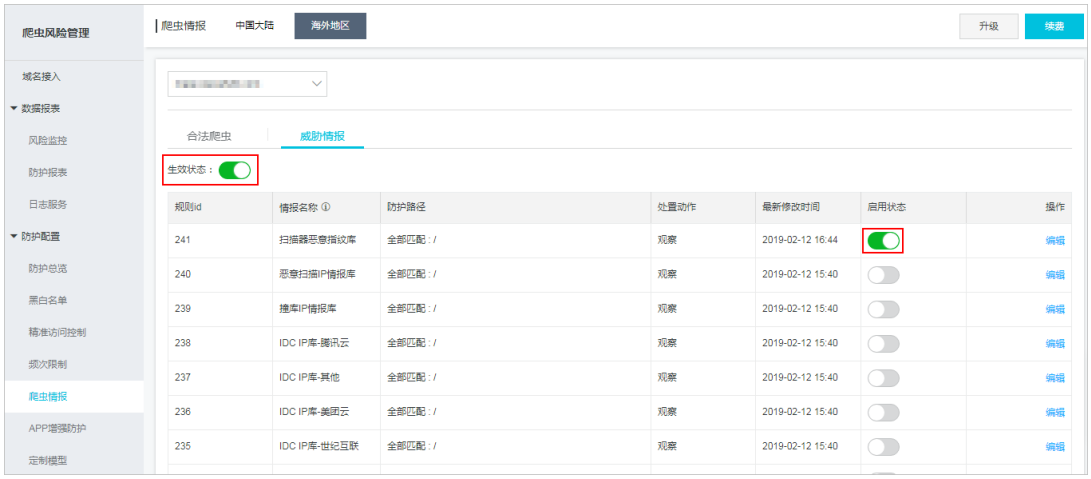
b. 在规则列表中，根据**情报名称**选择要放行的合法爬虫，打开对应的**启用状态**开关。默认规则支持放行来自以下搜索引擎的爬虫请求：Google、Bing、百度、搜狗、360、Yandex。

> **说明** 您也可以只打开规则106（**合法搜索引擎白名单**），放行所有支持的搜索引擎白名单。

- 添加威胁情报规则

a. 在**威胁情报**页签下，打开**生效状态**开关。

> **说明** 若您不再需要使用此功能，请在此页面关闭**生效状态**开关。



b. 在规则列表中，根据**情报名称**选择要设防的黑名单IP情报库，打开对应的**启用状态**开关。支持的情报库包括：

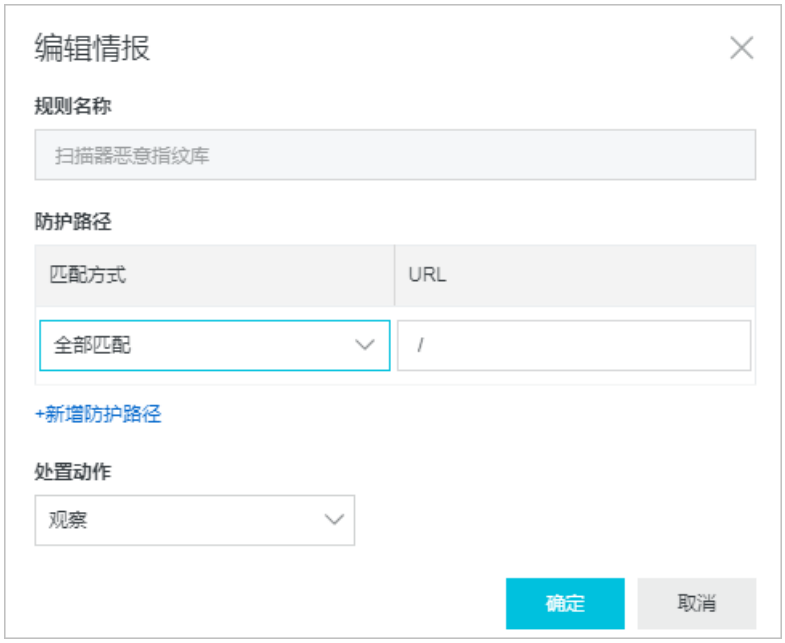
情报库	描述
扫描器恶意指纹库	常见扫描器的特征库。
恶意扫描IP情报库	基于阿里云全网实时检测到的恶意扫描行为攻击源IP进行分析，得到的动态IP情报库。
撞库IP情报库	基于阿里云全网实时检测到的撞库、暴力破解行为攻击源IP进行分析，得到的动态IP情报库。
伪造蜘蛛情报库	识别爬虫程序伪造合法搜索引擎的user-agent（如BaiduSpider）来逃避检测的行为。 **注意** 开启该名单之前，请确保已经放行合法爬虫白名单，否则可能导致误拦截。
恶意爬虫情报库	基于阿里云全网实时检测到的爬虫行为攻击源IP进行分析，得出的动态IP情报库。 该名单库有低级、中级、高级三个等级。级别越高，对应的情报库内IP数量越多，相应的误判概率更大。 建议您对高级情报库规则设置二次校验（使用滑块验证、JS校验等处置动作）；对不适用于二次校验的场景（如API等）配置低级别的情报库规则。
IDC情报库	相关公有云或IDC机房的IP库，包括：阿里云、腾讯云、美团云、世纪互联、其他。这些IP段往往被爬虫用于部署爬虫程序或用作代理，而不会被正常用户使用。

开启默认规则后，当目标情报库内的来源IP向域名下任意路径发起访问请求时，一律触发观察操作（即放行请求同时进行记录）。

如果您希望进一步调整默认规则（如指定要防护的关键路径或者修改处置动作），请参照后续步骤自定义威胁情报规则。

c. （可选）（可选）选择要调整的默认规则，单击**编辑**。

d. （可选）（可选）在**编辑情报**对话框，完成以下配置：



配置	描述
防护路径	填写要防护的具体URL（如"/abc"、"/login/abc"、"/"表示所有路径），并选择与该防护路径的**匹配方式**： ■ **全部匹配**：被访问地址与防护路径完全匹配时，记作命中。 ■ **前缀匹配**：被访问地址与防护路径的前缀相同时，记作命中。 ■ **正则匹配**：被访问地址满足防护路径的正则表达式时，记作命中。 **说明** 单击**新增防护路径**可以添加最多10个路径。
处置动作	指定命中规则后的操作： ■ **观察**：放行请求并进行记录。 ■ **阻断**：直接阻断访问请求。 ■ **JavaScript校验**：通过JavaScript校验请求数据，验证通过后放行请求。 ■ **滑块**：在客户端跳出滑块验证页面，客户端完成验证后放行请求。 **说明** 滑块验证仅支持同步请求，如有异步请求（如AJAX）防护需求请联系阿里云安全团队。如果不确定您防护的接口能否正常使用滑块验证，建议您先在精准访问控制中配置针对测试IP和URL的规则来验证和调试。

自定义威胁情报规则示例

- 规则说明：防护当前域名下“/login.do”开头的URL，当请求源IP来自撞库IP情报库时，则弹出滑块验证。

  规则配置：

- 规则说明：防护当前域名下“/houselist”开头的URL，当请求源IP来自恶意爬虫情报库（高级）时，则进行JavaScript校验。

  规则配置：

e. （可选）（可选）单击**确定**完成编辑。

3.App增强防护SDK

3.1. 方案概述

爬虫风险管理（Anti-Bot Service，简称Anti-Bot）针对原生App端提供安全SDK解决方案。为您的App提供可信通信、防机器脚本滥刷等安全防护，有效识别高风险手机、猫池、牧场等特征。

App端安全SDK方案集成了阿里巴巴集团多年来对抗黑灰产、羊毛党的经验和技术积累。将您的App集成Anti-Bot安全SDK后，您的App将获得与天猫、淘宝、支付宝等App端相同的可信通信技术能力，并可共享阿里巴巴集团多年对抗黑灰产、羊毛党所积累的恶意设备指纹库，从根本上帮助您解决App端的安全问题。

Anti-Bot提供的App端安全SDK方案帮助您解决以下**原生App端**的安全问题：

- 恶意注册、撞库、暴力破解
- 针对App的大流量CC攻击
- 短信/验证码接口被刷
- 薅羊毛、抢红包
- 恶意秒杀限时限购商品
- 恶意查票、刷票（例如，机票、酒店等场景）
- 价值资讯爬取（例如，价格、征信、融资、小说等内容）
- 机器批量投票
- 灌水、恶意评论

配置App端安全SDK方案

参考以下操作步骤，为您的App配置安全SDK解决方案：

> **说明** 配置App端安全SDK方案，您无需在服务器端进行任何改动。配置完成后，Anti-Bot将自动过滤恶意请求，将合法的请求转发给源站服务器。恶意请求产生的压力将全部由Anti-Bot承担，保障您的服务器端稳定运行。

1. 登录爬虫风险管理控制台，选择您的Anti-Bot实例所在的地区。
2. 定位到**域名接入**页面，单击**添加域名**，为您App端使用的域名添加域名接入配置。具体操作步骤，请参见添加域名配置。
3. 在您App使用的域名的DNS解析服务提供商处，添加Anti-Bot实例分配的CNAME记录，将App的域名解析指向Anti-Bot实例。具体操作步骤，请参见修改DNS解析。
4. 在您的App中集成Anti-Bot提供的SDK组件。

   > **说明** 集成SDK工作可能需要您投入1-2天时间的工作量。

   关于集成SDK的详细操作说明，请参见：

   - iOS SDK集成指南
   - Android SDK集成指南

   > **说明** 在集成SDK时，您可以自定义所需验证的URL。

5. 测试通过后，打包并发布已集成SDK的新App版本，即可享受App端安全SDK解决方案为您提供的安全

防护。

> 说明 集成SDK后，您可以通过SDK防护配置自定义所需防护的接口和防护策略。

3.2. iOS SDK集成指南

参考以下SDK集成说明将您的iOS App集成Anti-Bot SDK。

iOS SDK文件说明

联系爬虫风险管理技术支持人员获取对应的SDK包，解压至本地。

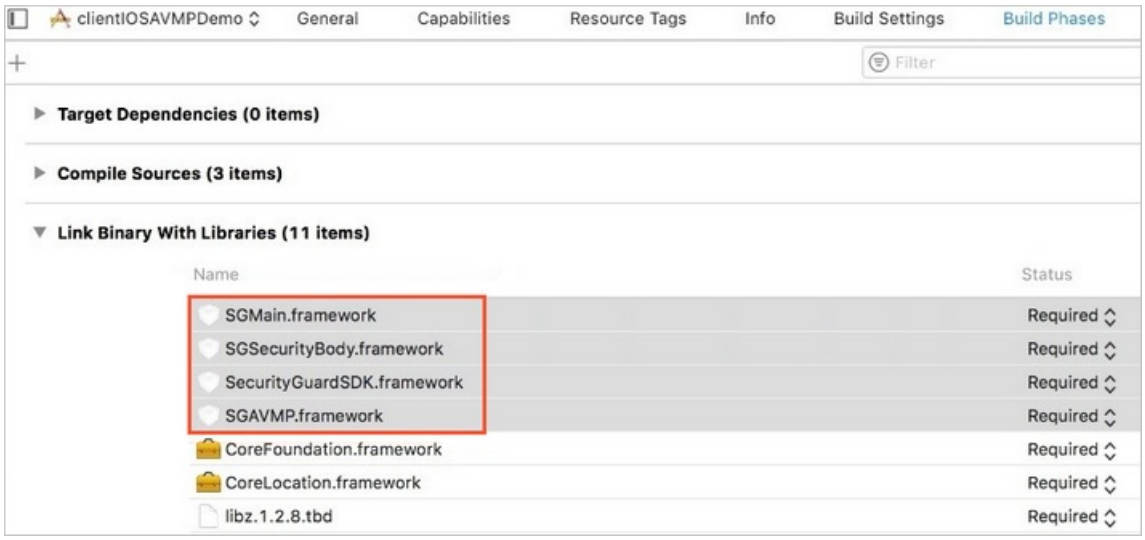
在sdk-iOS文件夹中，包含以下iOS SDK文件：

文件	说明
SGMain.framework	主框架SDK文件
SecurityGuardSDK.framework	基础安全插件
SGSecurityBody.framework	人机识别插件
SGAVMP.framework	虚拟机插件
yw_1222_0335_mwua.jpg	配置文件

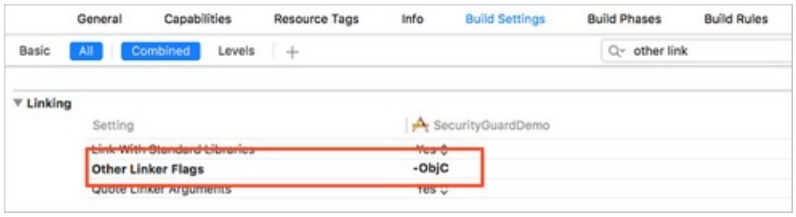
项目工程配置

参考以下步骤，为您的App项目工程配置：

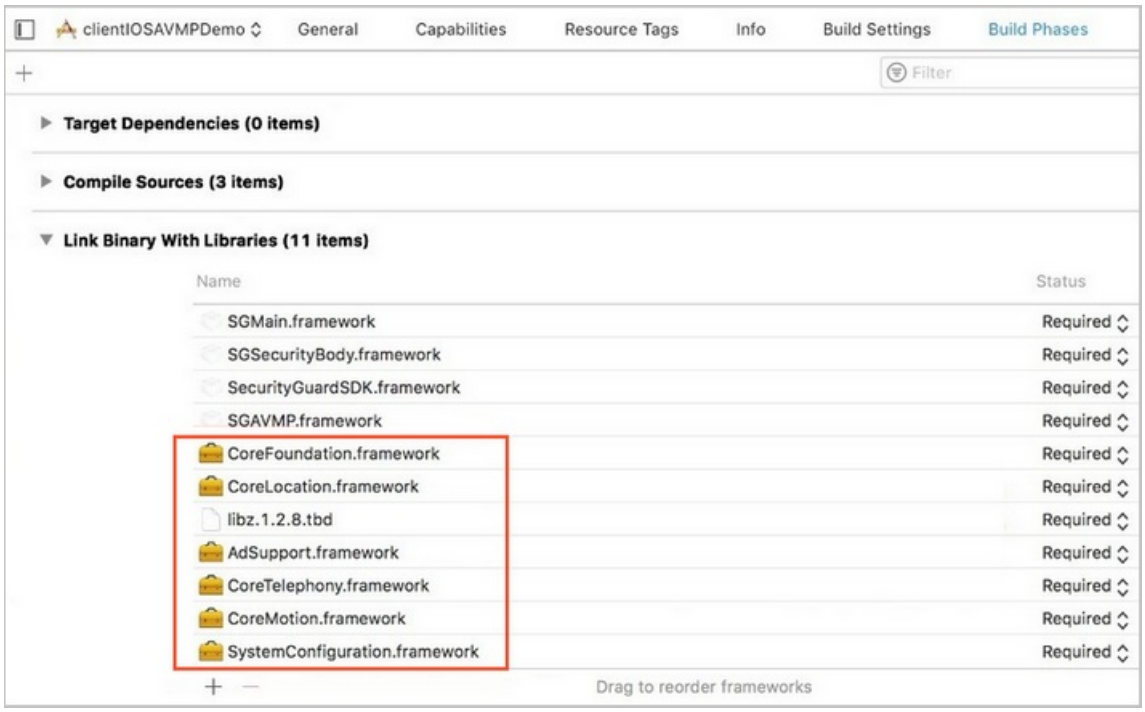
1. 添加Framework。将Anti-Bot SDK包中的四个 `.framework` 文件添加到iOS App工程的依赖库中。



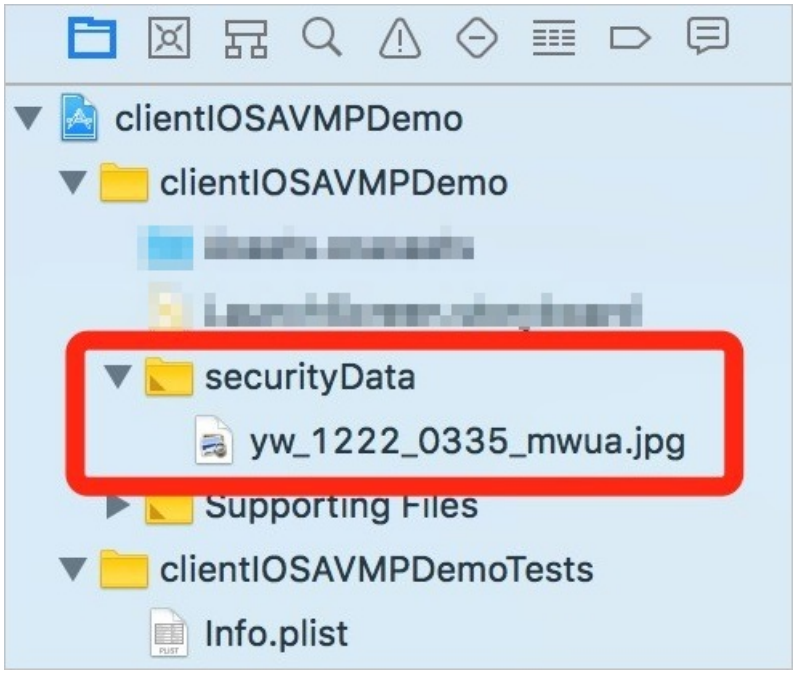
2. 添加链接选项。



3. 添加以下系统依赖库。



4. 引入配置文件。将SDK包中的 `yw_1222_0335_mwua.jpg` 配置文件加至 `mainbunle` 目录。



> **说明** 在应用集成多个target的情况下，确认将 `yw_1222_0335_mwua.jpg` 配置文件加入到正确的Target Membership中。

代码编写

1. 初始化SDK。
   - **接口定义：** `+ (BOOL) initialize;`
   - **接口描述：**
     - 功能：初始化SDK。
     - 参数：无。
     - 返回值：BOOL类型。初始化成功返回YES，失败返回NO。
   - **调用方式：** `[JAQAVMPSignature initialize];`
   - **代码示例：**

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

2. 签名请求数据。
   - **接口定义：** `+ (NSData*) avmpSign: (NSInteger) signType input: (NSData*) input;`
   - **接口描述：**
     - 功能：使用avmp技术对input的数据进行签名处理，并返回签名串。

> **警告** 被签名的请求体应该与客户端实际发送的请求体完全一致。完全一致的含义包括请求体中字符串的编码格式、空格、特殊字符以及参数的顺序等均一致，否则将导致签名验证失败。

- 参数：详见下表。

参数名	类型	是否必须	说明
signType	NSInteger	是	签名使用的算法。目前是固定值，填写 `3` 。
input	NSData*	否	待签名的数据，一般是整个请求体（request body）。 **说明** 如果请求体为空（例如POST请求的body为空、或者GET请求），则填写空对象null或空字符串的Bytes值。

- 返回值：NSData*类型，返回签名串。

- **调用方式**：`[JAQAVMPSignature avmpSign: 3 input: request_body];`
- **代码示例**：

说明 客户端向服务器端发送数据时，需要调用avmpSign接口对整个body数据进行签名处理，所得到的签名串就是wToken。

```
# 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;
  }
 }
}
```

说明 如果请求体为空，仍需要调用avmpSign接口生成wToken，第二个参数直接传入空值即可。

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

3. 将wToken放进协议头。

   代码示例

```
#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. 发送数据到服务器。将修改好协议头的数据发送到Anti-Bot，通过解析wToken进行风险识别、拦截恶意请求，然后将合法请求转发回源站。

错误码

上述initialize和avmpSign接口的调用过程中可能出现异常。如果生成签名串异常或失败，在console中搜索与 `SG Error` 相关的错误码信息。

常见错误代码及含义

错误代码	含义
1901	参数不正确，请检查输入的参数。
1902	图片文件错误。可能是由于BundleID不匹配导致。
1903	图片文件格式有问题。
1904	请升级新版本图片。AVMP签名功能仅支持v5图片。
1905	无法找到图片文件。请确保图片文件yw_1222_0335_mwua.jpg已正确添加在工程中。

错误代码	含义
1906	图片中缺少AVMP签名对应的byteCode。请检查使用的图片是否正确。
1907	初始化AVMP失败，请重试。
1910	非法的avmpInstance实例。可能由于以下原因导致： • AVMPInstance被destroy后，调用InvokeAVMP。 • 图片byteCode版本与SDK不匹配。
1911	加密图片的byteCode没有相应导出的函数。
1912	AVMP调用失败，请联系我们。
1913	AVMPInstance被destroy后，调用InvokeAVMP时出现该错误。
1915	AVMP调用内存不足，请重试。
1999	未知错误，请重试。

3.3. Android SDK集成指南

参考以下SDK集成说明将您的Android App集成Anti-Bot SDK。

Android SDK文件说明

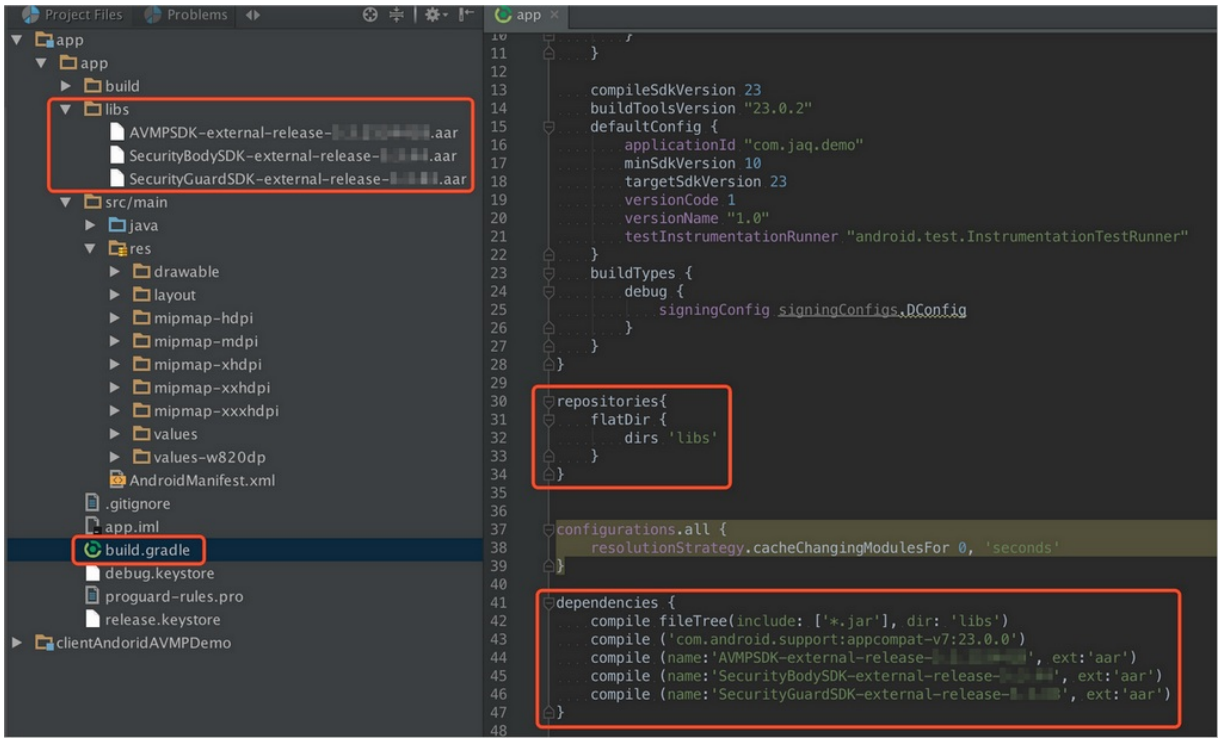
联系爬虫风险管理技术支持人员获取对应的SDK包，解压至本地。

在sdk-Android文件夹中，包含以下Android SDK文件：

文件	说明
SecurityGuardSDK-xxx.aar	主框架SDK文件
AVMPSDK-xxx.aar	虚拟机引擎插件
SecurityBodySDK-xxx.aar	人机识别插件
yw_1222_0335_mwua.jpg	虚拟机引擎配置文件

项目工程配置

参考以下步骤，完成项目工程配置：



1. 在Android Studio中导入Anti-Bot SDK的aar文件。将sdk-Android文件夹中所有aar文件复制到Android App工程项目的 `libs` 目录中。

   > **说明** 如果当前工程项目中不存在 `libs` 目录，在指定路径下手工创建 `libs` 文件夹。

2. 打开该Module的build.gradle文件，增加以下配置信息。
   - 将 `libs` 目录添加为查找依赖的源。

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

   - 添加编译依赖。

     > **说明** aar文件的版本号可能有所不同，以您下载解压得到的文件名为准。

```
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. 将Anti-Bot SDK的jpg配置文件导入 drawable 目录。将sdk-Android文件夹中的yw_1222_0335_mwua.jpg配置文件复制到Android App工程项目的 drawable 目录中。

> **说明** 如果当前工程项目中不存在 drawable 目录，在指定路径下手工创建 drawable 文件夹。

4. 过滤ABI（删除多余架构SO）。Anti-Bot SDK目前仅支持armeabi、armeabi-v7a、arm64-v8a架构的SO。因此，您需要对最终导出的ABI进行过滤。否则，可能导致App崩溃。
   i. 在Android App工程的lib目录中，删除除armeabi、armeabi-v7a、arm64-v8a文件夹外所有其他CPU架构的文件夹，包括x86、x86_64、mips、mips64等，只保留armeabi、armeabi-v7a、arm64-v8a文件夹。
   ii. 参考以下代码示例，在App工程的build.gradle配置文件中增加过滤规则，被abiFilters指定的架构将会被包含在APK文件中。

   > **说明** 本代码示例中仅指定了armeabi架构，您可以根据实际情况指定或兼容armeabi-v7a、arm64-v8a架构。

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

   > **说明** 只保留armeabi架构的SO，不会影响App的兼容性，还能大幅减小App的体积。

5. 添加App权限。
   - 如果是Android Studio项目，并且使用aar方式进行集成。由于在aar中已经声明了相关权限，因此不

需要在项目中额外配置权限。

- 如果是Eclipse项目，您需要在AndroidMenifest.xml文件中添加以下权限配置：

```
<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. 添加ProGuard配置。

> 说明 如果您使用了Proguard进行混淆，则需要添加ProGuard配置。ProGuard的配置根据集成方式的不同，分为Eclipse和Android Studio两种情况。

- Android Studio

  如果在build.gradle中配置了proguardFiles，并且开启了minifyEnabled，则表明使用了proguard-rules.pro配置文件进行混淆。

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

- Eclipse

  如果在project.properties中指定了proguard配置，例如在project.properties中存在 `proguard.config=proguard.cfg` 语句，则表明使用了proguard进行混淆。

> 说明 混淆配置在proguard.cfg 文件中。

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

添加keep规则

为了保证一些需要的类不被混淆，需要在proguard的配置文件中添加以下规则：

```
-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.**{*;}
```

代码编写

1. 导入包。

```
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. 初始化。
   - **接口定义**：`boolean initialize();`
   - **接口描述**：
     - 功能：初始化SDK。
     - 参数：无。
     - 返回值：Boolean类型。初始化成功返回true，失败返回false。
   - **示例代码**：

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

3. 签名请求数据。
   - **接口定义**：`byte[] avmpSign(int signType, byte[] input);`
   - **接口描述**：
     - 功能：使用avmp技术对input的数据进行签名处理，并且返回签名串。
     - 参数：详见下表。

参数名	类型	是否必须	说明
signType	int	是	签名使用的算法。目前是固定值，填写 `3` 。
input	byte[]	否	待签名的数据，一般是整个请求体（request body）。 **说明** 如果请求体为空（例如POST请求的body为空、或者GET请求），则填写空对象null或空字符串的Bytes值（例如，"".getBytes("UTF-8"))。

- 返回值：byte[]类型，返回签名串。
- **示例代码**：客户端向服务器端发送数据时，需要调用avmpSign接口对整个body数据进行签名处理，所得到的签名串就是wToken。

```
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. 将wToken放进协议头。在HttpURLConnection类的对象中添加wToken字段的内容。

   示例代码：

```
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. 发送数据到服务器。将修改好协议头的数据发送到App自有服务器，中间会由Anti-Bot截获，并通过解析wToken进行风险识别。

警告 被签名的请求体应该与客户端实际发送的请求体完全一致。完全一致的含义包括请求体中字符串的编码格式、空格、特殊字符以及参数的顺序等均一致，否则将导致签名验证失败。

错误码

上述initialize和avmpSign接口的调用过程中可能出现异常。如果生成签名串异常或失败，搜索Log中与 `SecException` 相关的信息。

常见错误码及含义

错误代码	含义
1901	参数不正确，请检查输入的参数。

错误代码	含义
1902	图片文件有问题。一般是获取图片文件时的APK签名和当前程序的APK签名不一致。请使用当前程序的APK重新生成图片。
1903	图片文件格式有问题。
1904	请升级新版本图片。AVMP签名功能仅支持V5图片。
1905	没有找到图片文件。请确保图片文件在res\drawable目录下，与AVMP相关的图片为yw_1222_0335_mwua.jpg。
1906	图片中缺少AVMP签名对应的byteCode。请检查使用的图片是否正确。
1907	初始化AVMP失败，请重试。
1910	非法的avmpInstance实例。可能由于以下原因导致： • AVMPInstance被destroy后，调用InvokeAVMP。 • 图片byteCode版本与SDK不匹配。
1911	加密图片的byteCode没有相应导出的函数。
1912	AVMP调用失败。请联系我们。
1913	AVMPInstance被destroy后，调用InvokeAVMP出现该错误。
1915	AVMP调用内存不足，请重试。
1999	未知错误，请重试。

集成效果确认

参考以下步骤，确认您的App已正确集成Anti-Bot SDK：

1. 将打包生成的APK文件通过修改扩展名的方式转换成ZIP压缩文件，并将该压缩文件解压至本地。
2. 定位到工程的lib目录，确保文件夹中只存在armeabi、armeabi-v7a、arm64-v8a文件夹。

   > **说明** 如果存在其他架构的文件夹，参考项目工程配置移除其它架构的文件夹。

3. 定位到工程的res/drawable目录，确认存在yw_1222_0335_mwua.jpg文件，且文件大小不为0。
4. 通过打印日志，确保调用avmpSign接口后能生成正确的签名信息。

   > **说明** 如果签名信息未生成，参考错误码信息进行排查。

常见问题

指定shrinkResources后，密钥图片被错误地优化

在Android Studio中，如果指定shrinkResources为true，在工程编译时可能对未在代码中引用的资源文件进行优化。该操作可能导致Anti-Bot SDK中的jpg文件无法正常工作。如果打包后得到APK中，yw_1222_0335.jpg配置文件的大小为0KB，则表明该图片文件已被优化。

解决方法

1. 在工程的res目录中新建raw目录，并在raw目录中创建keep.xml文件。
2. 在keep.xml文件中，添加以下内容：

```
<?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. 添加完成后，重新编译工程APK即可。

3.4. SDK防护配置

在App中完成SDK接入后，您需要在爬虫风险管理控制台进行SDK防护配置，配置需要防护的路径和版本。

SDK的完整接入流程如下：

1. 在App上完成SDK集成操作，具体请参见iOS SDK集成指南和Android SDK集成指南。
2. 在爬虫风险管理控制台中配置防护路径。具体请参见配置路径防护。
3. 使用集成了SDK的App发送测试请求，并通过响应和日志分析调试错误和异常，直到确认正确集成。
4. 发布正确集成SDK的新版本App，并在爬虫风险管理控制台中开启防护。具体请参见开启APP增强防护。

> **说明** 发布新版本App时，建议您进行强制更新，否则老版本App依然存在安全风险。

配置路径防护

通过配置防护路径指定要防护的地址，并在其下产生防护规则。

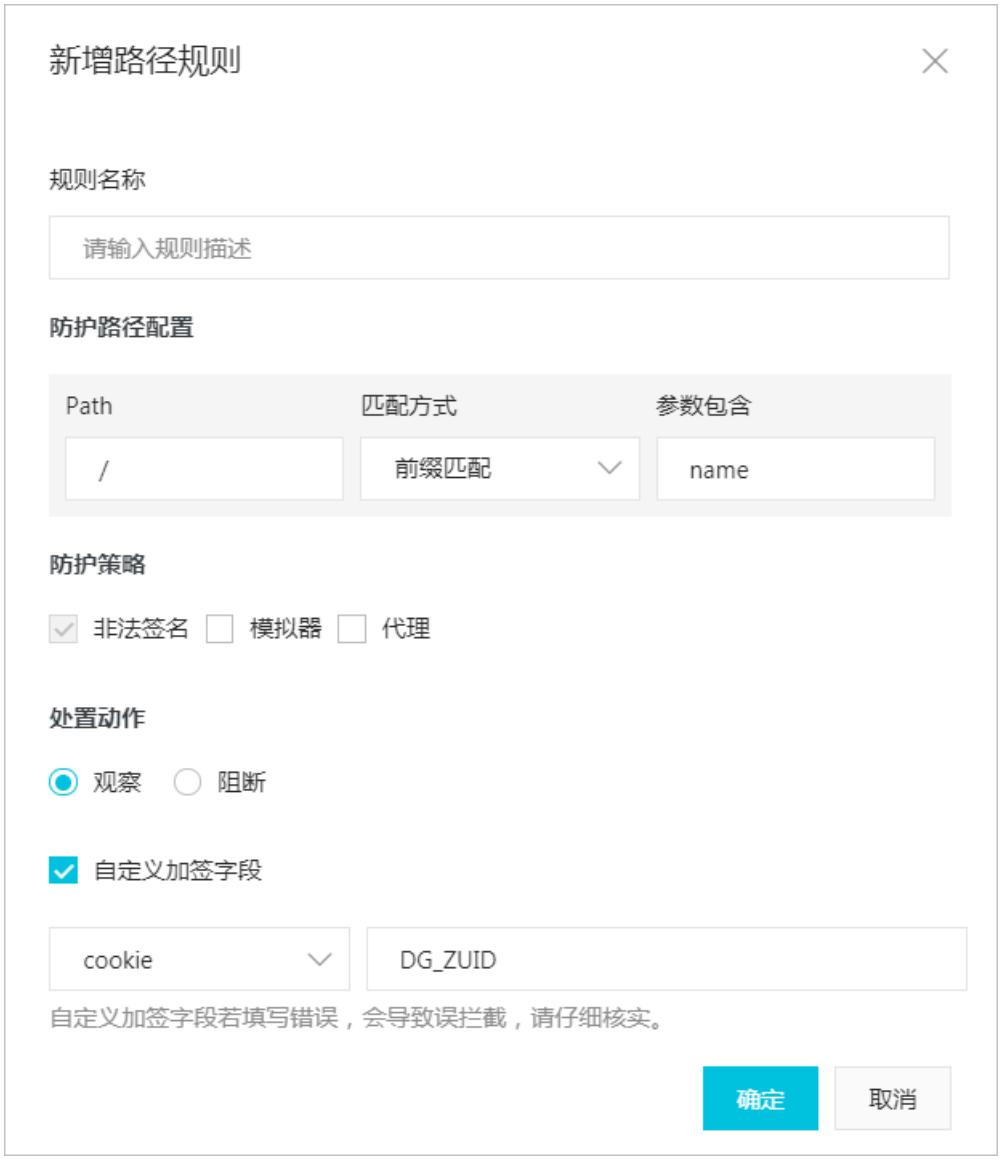
操作步骤

1. 登录爬虫风险管理控制台。
2. 在**防护配置 > APP增强防护**页面，选择要配置的域名。
3. 在**路径防护**下，单击**添加**。
4. 在**新增路径规则**对话框中，完成以下配置。

> **说明** 建议您在测试阶段设置全路径防护（使用前缀匹配“/”），并将处置动作设置为观察（如果是测试域名，可以设置为拦截）。这样可以在不影响线上业务的前提下进行调试。

配置项	说明
规则名称	必填，为该规则命名。

配置项	说明
防护路径配置	○ **Path**：必填，要防护的路径地址。使用“/”表示全路径。 **说明** POST请求的body长度超过8k的情况下，可能会造成验签失败。如果该类型接口没有防护必要（如上传大图片等），建议不要经过SDK防护；如确实有防护必要，请使用自定义加签字段。 ○ **匹配方式**：支持**前缀匹配**、**精确匹配**和**正则匹配**。 前缀匹配会匹配指定路径下的所有接口，精确匹配只匹配指定路径，正则匹配支持以正则表达式的方式描述指定路径。 ○ **参数包含**：要防护的路径下包含固定参数时，指定要匹配的参数内容，更准确地定位接口。参数内容指请求地址中问号后面的内容。 示例：假设要防护的URL包括 `域名/?action=login&name=test` 。您可以将**Path**设置为“/”，**匹配方式**设置为“前缀匹配”，并在**参数包含**中填写“name”或者“login”或者“name=test”或者“action=login”。
防护策略	○ **非法签名**：默认勾选（不支持取消），验证对指定路径的请求的签名是否正确。签名不正确则命中。 ○ **模拟器**：（建议勾选）勾选后，检测用户是否使用模拟器对指定路径发起请求。使用模拟器则命中。 ○ **代理**：（建议勾选）勾选后，检测用户是否使用代理工具对指定路径发起请求。使用代理工具则命中。
处置动作	对命中防护策略的用户请求执行的操作。 ○ **观察**：只记录日志，不阻断请求。 ○ **阻断**：阻断请求，返回405状态码。
自定义加签字段	启用自定义加签字段后，系统将根据所设置的需要加签的请求字段和对应的字段值进行加签验证，判断是否命中该防护策略。 系统默认情况对请求的body加签，如果body长度超过8k则可能导致验签失败。这种情况下，您可以启用自定义加签字段，使用您指定的请求字段来取代系统默认的加签字段。 启用自定义加签字段后，您可以选择请求Header、参数或Cookie类型，然后填写需要加签的字段即可。例如，您可以选择**Cookie**，填写*DG_ZUID*，请求Cookie中的DG_ZUID字段将取代系统默认的body字段作为加签字段。



5. 单击**确定**完成新增。

   已添加规则支持**编辑**和**删除**。

（可选）配置版本防护

通过配置版本防护可以拦截来自非官方App的请求。如果您需要验证App合法性，可通过配置该策略实现。

> **说明** 如果不需要进行App合法性验证，则可不配置版本防护策略。

操作步骤

1. 登录爬虫风险管理控制台。
2. 在**防护配置 > APP增强防护**页面，选择要配置的域名。
3. 在**版本防护**下，开启**仅允许指定版本通过**开关。

> **说明** 如需取消版本防护，在此处关闭**仅允许指定版本通过**开关。

4. 在**新增版本规则**对话框中，完成以下配置。

配置项	描述
规则名称	为该规则命名。
合法版本	○ **指定合法包名**：必填，指定合法的App包名称。例如，*com.aliyundemo.example* ○ **包签名**：联系阿里云相关安全技术人员获取。 **说明** 注意此处填写的不是App证书签名。 **说明** 如果无需验证对应的App包签名，则无需设置包签名项，将只验证所设定的合法App包名称。 单击**新增合法版本**可以添加最多5条版本记录，包名称不允许重复。目前不区分iOS和Android，合法的记录都可以填写进去以匹配多个包名。
非法版本的处置操作	○ **观察**：只记录日志，不阻断请求。 ○ **阻断**：阻断请求，返回405状态码。

5. 单击**确定**完成新增。

   已添加规则支持**编辑**。

开启App增强防护

通过调试，确定已在App中正确集成SDK并发布新版App后，您需要开启App增强防护，使防护配置生效。

1. 登录爬虫风险管理控制台。

2. 在**防护配置 > APP增强防护**页面，选择要配置的域名。
3. 开启**生效状态**开关。

> **注意** 未集成SDK或未调试完成前，请不要为生产环境中的域名开启阻断模式；否则可能会因为SDK没有正确集成导致合法请求被拦截。在测试接入阶段，可以开启观察模式，通过日志调试SDK集成。

更多帮助

通过钉钉软件扫描该二维码，加入技术支持群。您可以直接向安全专家咨询关于爬虫风险管理使用的任何技术问题或紧急问题。

> **说明** 请参考钉钉官网，下载并安装钉钉即时通信软件。

4.日志实时查询分析

4.1. 启用Anti-Bot日志服务

日志服务（Log Service）支持实时采集阿里云爬虫风险管理（Anti-Bot Service，简称Anti-Bot）已防护的网站访问日志以及防护日志，并支持对采集到的日志数据进行实时检索与分析。

您可以在爬虫风险管理控制台中基于采集到的网站日志对网站的访问和攻击行为进行即时分析研究、协助您的安全管理人员制定防护策略。

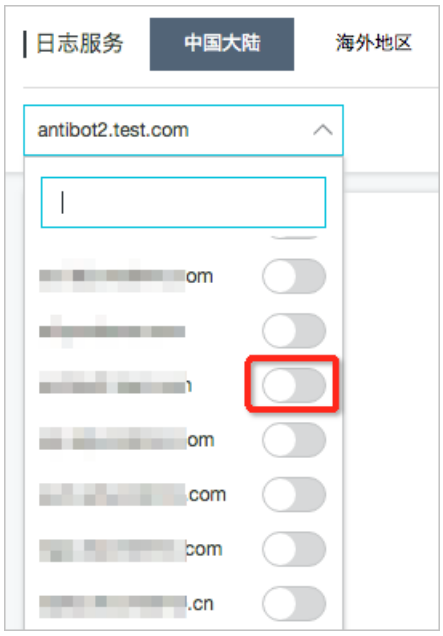
操作步骤

1. 登录爬虫风险管理控制台。
2. 定位到**数据报表 > 日志服务**页面，选择您的实例所在地域。

   > **说明** 如果您是第一次使用爬虫风险管理的日志服务，需要单击**授权**，并根据页面提示完成授权操作，授权爬虫风险管理产品将记录的所有日志存储到您专属的日志服务Logstore中。

3. 单击网站域名下拉框，选择需要启用日志服务的网站域名，单击启用开关。

   > **说明** 网站域名下拉列表中将展示所有您已接入爬虫风险管理进行防护的网站域名。



至此，您已成功为该网站域名开启日志服务。日志服务会在您的阿里云账号中自动创建一个专属日志库和专

属Logstore，爬虫风险管理自动将所有启用日志服务的网站域名的日志实时导入该专属日志库（antibot-logstore）。

然后，您就可以对启用日志服务的网站域名的访问日志进行检索和分析。

限制与说明

- 专属日志库不支持写入其他数据。

  > **说明** 爬虫风险管理记录的网站日志将被存储在您的专属日志库中，该日志库不支持通过包括API、SDK在内的任何方式写入其他数据。

- 暂不支持修改专属日志库的存储周期等基本设置。
- 切勿随意删除或修改日志服务为您创建的默认Project、Logstore、索引和仪表盘等设置。
- 日志服务将不定期更新、升级日志查询与分析功能，您专属日志库中的索引与默认报表也将自动更新。
- 如果子账号需要使用日志查询分析功能，您可以通过RAM为其授予日志服务的相关权限。

4.2. 常用日志查询分析语句

为已防护的域名启用Anti-Bot日志服务后，您可以通过编写日志查询分析语句查询该网站域名的详细访问、攻防日志信息。

您可以参考以下常用日志查询分析语句，根据业务需要编写查询语句获取网站域名的相关日志信息。

> **说明** 您可以根据实际情况调整查询分析语句中的 `limit` 值来返回需要的记录数。例如，`limit 10` 表示返回10条记录。未指定 `limit` 值时，则默认返回前100条记录。

查询网站域名业务访问相关信息

- 查询入方向带宽流量

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

- 查询出方向带宽流量

```
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 "流出流量(KB/s)", round(sum(if((block_action <> ''),
body_bytes_sent, 0))/1024.0/600, 2) as "被攻击流量(KB/s)"
group by __time__ - __time__% 600 order by dt limit 10000
```

- 查询QPS峰值

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

- 查询近10分钟内每分钟的访问请求量（按时间降序排列）

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

- 查询TOP 10访问客户端IP

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

- 查询TOP 10被访问URL地址

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

- 查询HTTP状态码

> **说明** 通过观察异常状态码可确认业务是否正常。

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

查询网站域名安全防护相关信息

- 查询指定URL或接口地址的TOP 10访问客户端IP

> **说明** 遭受攻击时恶意攻击者的IP通常排名靠前。

```
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
```

- 查询指定IP访问的URL地址

> **说明** 一般遭受CC攻击时，被攻击的URL或接口地址比较集中。

```
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
```

- 查询来自指定访问客户端IP的请求命中的Anti-Bot防护策略ID

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

- 查询指定Anti-Bot防护策略ID的命中情况

> **说明** 基于查询结果可进一步观察防护策略实际效果和命中率。

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

- 查询Anti-Bot增强防护SDK的验签情况

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

4.3. 日志字段说明

爬虫风险管理（Anti-Bot Service，简称Anti-Bot）的日志服务功能详细记录网站域名的访问、攻防日志。日志中包含数十个字段，您可以根据不同需要选取特定的日志字段进行查询分析。

字段	说明	示例值
__topic__	日志主题（Topic），该字段值固定为antibot_access_log。	antibot_access_log
antibot	触发的爬虫风险管理防护策略类型，包括： • ratelimit：频次控制 • sdk：APP端增强防护 • algorithm：算法模型 • intelligence：爬虫情报 • acl：精准访问控制 • blacklist：黑名单	ratelimit
antibot_action	爬虫风险管理防护策略执行的操作，包括： • *challenge*：下发JavaScript脚本进行验证 • *drop*：拦截 • *captcha*：滑块验证 • *report*：仅观察记录	drop

字段	说明	示例值
antibot_rule	所触发的爬虫风险管理的规则ID。	5472
antibot_verify	爬虫风险管理采用的校验手段的验证结果。 **说明** 当antibot_action字段的值为challenge和captcha时将记录该值。 • challenge_fail：JS验证失败 • challenge_pass：JS验证通过 • captcha_fail：滑块验证失败 • captcha_pass：滑块验证通过	challenge_fail
block_action	触发防爬拦截的防护类型。该值固定为*antibot*。	antibot
body_bytes_sent	发送给客户端的HTTP Body的字节数。	2
content_type	访问请求内容类型。	application/x-www-form-urlencoded
host	源网站。	api.aliyun.com
http_cookie	访问请求头部中带有的访问来源客户端Cookie信息。	k1=v1;k2=v2
http_referer	访问请求头部中带有的访问请求的来源URL信息。若无来源URL信息，则显示`-`。	http://xyz.com
http_user_agent	访问请求头部中的User Agent字段，一般包含来源客户端浏览器标识、操作系统标识等信息。	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build/HUAWEIEDISON-AL10)
http_x_forwarded_for	访问请求头部中带有的XFF头信息，用于识别通过HTTP代理或负载均衡方式连接到Web服务器的客户端最原始的IP地址。	-
https	访问请求是否为HTTPS请求，其中： • true：HTTPS请求。 • false：HTTP请求。	true
matched_host	匹配到的已接入Anti-Bot防护配置的域名，可能是泛域名。若无法匹配到相关域名配置，则显示`-`。	*.aliyun.com
real_client_ip	访问的客户端的真实IP。若无法获取到，则显示`-`。	1.2.3.4
region	Anti-Bot实例地域信息。	cn

字段	说明	示例值
remote_addr	访问请求的客户端IP。	1.2.3.4
remote_port	访问请求的客户端端口。	23713
request_length	访问请求长度，单位字节。	123
request_method	访问请求的HTTP请求方法。	GET
request_path	请求的相对路径（不包含查询字符串）。	/news/search.php
request_time_msec	访问请求时间，单位为毫秒。	44
request_traceid	访问请求唯一ID标识。	7837b11715410386943437009ea1f0
server_protocol	源站服务器响应的协议及版本号。	HTTP/1.1
status	爬虫风险管理返回给客户端的HTTP响应状态信息。	200
time	访问请求的发生时间。	2018-05-02T16:03:59+08:00
ua_browser	访问请求来源的浏览器信息。	ie9
ua_browser_family	访问请求来源所属浏览器系列。	internet explorer
ua_browser_type	访问请求来源的浏览器类型。	web_browser
ua_browser_version	访问请求来源的浏览器版本。	9.0
ua_device_type	访问请求来源客户端的设备类型。	computer
ua_os	访问请求来源客户端的操作系统信息。	windows_7
ua_os_family	访问请求来源客户端所属操作系统系列。	windows
upstream_addr	Anti-Bot使用的回源地址列表，格式为 `IP:Port` ，多个地址用逗号分隔。	1.2.3.4:443
upstream_ip	访问请求所对应的源站IP。例如，Anti-Bot回源到ECS的情况，该参数即返回源站ECS的IP。	1.2.3.4
upstream_response_time	源站响应Anti-Bot请求的时间，单位秒。如果返回“-”，代表响应超时。	0.044
upstream_status	源站返回给Anti-Bot的响应状态。如果返回“-”，表示没有响应（例如该请求被Anti-Bot拦截或源站响应超时）。	200
user_id	阿里云账号AliUID。	12345678

字段	说明	示例值
wxbb_action	当爬虫风险管理防护类型为APP端增强防护时，执行的操作： • *close*：拦截，相当于antibot_action字段值为*drop*。 • *test*：仅观察记录，相当于antibot_action字段值为*report*。 **说明** 如果未接入SDK防护，该字段值为 -。	close
wxbb_invalid_wua	APP端增强防护策略类型。 • *valid request*：请求签名验证通过 • *valid wua*： 使用真机，验证通过 • *simulator*：使用模拟器，根据所设置的规则执行后续动作 • *proxy*：使用代理，根据所设置的规则执行后续动作 • *wToken header not found*：表示不存在wToken（未签名），请求将被阻断 • *vmp check failed*：vmp签名校验失败，请求将被阻断 • *InvalidVmp*：签名无效，验证失败，请求将被阻断 **说明** 其他的字段值对应一些具体的风险标签。由于该防护策略存在一定的误报率，如果没有出现大规模绕过的情况，一般不建议启用该防护策略。如果您需要针对特定类型进行拦截，请联系我们开启相应风险标签的拦截功能。	valid wua