

移动应用安全

API 文档

产品文档



腾讯云

【版权声明】

©2013-2020 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

API 文档

- 更新历史

- 简介

- API 概览

- 调用方式

 - 请求结构

 - 公共参数

 - 签名方法 v3

 - 签名方法

 - 返回结果

- 应用加固相关接口

 - 查询加固结果

 - 用户查询提交过的app列表

 - 批量删除提交过的app信息

 - 提交加固基础数据

 - 创建资源

 - 查询加固策略

 - 新增加固策略

- 应用安全扫描相关接口

 - 批量查询扫描结果

 - 查询扫描列表

 - 批量删除提交过的app扫描信息

 - 批量提交安全扫描

- 用户服务信息相关接口

 - 获取用户的所有资源信息

 - 将应用绑定到资源

- 其他接口

 - 获取用户基础信息

 - 获取云COS临时密钥

- 数据结构

- 错误码

API 文档

更新历史

最近更新时间：2019-06-21 16:11:39

第 7 次发布

发布时间：2019-06-21 16:08:39

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [ScanPermissionInfo](#)
- [ScanPermissionList](#)
- [ScanSensitiveInfo](#)
- [ScanSensitiveList](#)

修改数据结构：

- [ScanSetInfo](#)
 - 新增成员：PermissionInfo, SensitiveInfo

第 6 次发布

发布时间：2018-12-20 19:36:19

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeUserBaseInfoInstance](#)

第 5 次发布

发布时间：2018-11-22 20:56:36

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateCosSecKeyInstance](#)

修改接口：

- [DescribeShieldResult](#)
 - 新增出参：StatusDesc, StatusRef

修改数据结构：

- [ScanSetInfo](#)
 - 新增成员：StatusCode, StatusDesc, StatusRef

第 4 次发布

发布时间：2018-09-13 19:58:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateResourceInstances](#)

第 3 次发布

发布时间：2018-09-07 11:09:23

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [PlanInfo](#)
 - 修改成员：SoType

第 2 次发布

发布时间：2018-07-26 15:10:56

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateBindInstance](#)
- [CreateShieldPlanInstance](#)
- [DescribeResourceInstances](#)
- [DescribeShieldPlanInstance](#)

修改接口：

- [DescribeScanInstances](#)
 - 修改入参：Filters
- [DescribeShieldInstances](#)
 - 修改入参：Filters

新增数据结构：

- [BindInfo](#)
- [Filter](#)
- [PlanDetailInfo](#)
- [PlanInfo](#)
- [ResourceInfo](#)
- [ResourceServiceInfo](#)
- [ShieldPlanInfo](#)
- [SoInfo](#)

删除数据结构：

- Filters

修改数据结构：

- [ServiceInfo](#)
 - 新增成员：PlanId
- [ShieldInfo](#)
 - 新增成员：ServiceEdition

第 1 次发布

发布时间：2018-06-07 15:01:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateScanInstances](#)
- [CreateShieldInstance](#)
- [DeleteScanInstances](#)
- [DeleteShieldInstances](#)
- [DescribeScanInstances](#)
- [DescribeScanResults](#)
- [DescribeShieldInstances](#)
- [DescribeShieldResult](#)

新增数据结构：

- [AdInfo](#)
- [AppDetailInfo](#)
- [AppInfo](#)
- [AppScanSet](#)
- [AppSetInfo](#)
- [Filters](#)
- [PluginInfo](#)
- [ScanInfo](#)
- [ScanSetInfo](#)
- [ServiceInfo](#)
- [ShieldInfo](#)
- [VirusInfo](#)
- [VulInfo](#)
- [VulList](#)

简介

最近更新时间：2019-06-21 16:11:38

应用安全 API 升级到 **3.0 版本**。全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验。支持全地域就近接入让您更快连接腾讯云产品。

欢迎使用应用安全服务（Mobile Security，简称MS）。

应用安全（MS）为用户提供移动应用（APP）全生命周期的一站式安全解决方案，成长于 12 亿终端的多年实践，已服务于金融、互联网、车联网、物联网，运营商，以及政务等多个行业。稳定、简单、有效，让移动安全建设不再是一种负担。

目前应用安全（MS）以API形式为用户提供安全扫描和应用加固两项服务，用户可以使用本文档介绍的API，对移动应用进行全方位的安全检测，或进行安全加密、加固服务，从而确保常见逆向工具无法正常运行、无法还原应用源码，以保护移动应用安全。

API 概览

最近更新时间：2018-12-20 21:12:20

应用加固相关接口

接口名称	接口功能
CreateResourceInstances	创建资源
CreateShieldInstance	提交加固基础数据
CreateShieldPlanInstance	新增加固策略
DeleteShieldInstances	批量删除提交过的app信息
DescribeShieldInstances	用户查询提交过的app列表
DescribeShieldPlanInstance	查询加固策略
DescribeShieldResult	查询加固结果

应用安全扫描相关接口

接口名称	接口功能
CreateScanInstances	批量提交安全扫描
DeleteScanInstances	批量删除提交过的app扫描信息
DescribeScanInstances	查询扫描列表
DescribeScanResults	批量查询扫描结果

用户服务信息相关接口

接口名称	接口功能
CreateBindInstance	将应用绑定到资源
DescribeResourceInstances	获取用户的所有资源信息

其他接口

接口名称	接口功能
CreateCosSecKeyInstance	获取云COS临时密钥
DescribeUserBaseInfoInstance	获取用户基础信息

调用方式

请求结构

最近更新时间：2020-05-11 10:12:07

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 `ms.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `ms.ap-guangzhou.tencentcloudapi.com`。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 `ms.ap-guangzhou.tencentcloudapi.com` 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	<code>ms.tencentcloudapi.com</code>
华南地区(广州)	<code>ms.ap-guangzhou.tencentcloudapi.com</code>
华东地区(上海)	<code>ms.ap-shanghai.tencentcloudapi.com</code>
华北地区(北京)	<code>ms.ap-beijing.tencentcloudapi.com</code>
西南地区(成都)	<code>ms.ap-chengdu.tencentcloudapi.com</code>
西南地区(重庆)	<code>ms.ap-chongqing.tencentcloudapi.com</code>
港澳台地区(中国香港)	<code>ms.ap-hongkong.tencentcloudapi.com</code>
亚太东南(新加坡)	<code>ms.ap-singapore.tencentcloudapi.com</code>
亚太东南(曼谷)	<code>ms.ap-bangkok.tencentcloudapi.com</code>
亚太南部(孟买)	<code>ms.ap-mumbai.tencentcloudapi.com</code>
亚太东北(首尔)	<code>ms.ap-seoul.tencentcloudapi.com</code>
亚太东北(东京)	<code>ms.ap-tokyo.tencentcloudapi.com</code>
美国东部(弗吉尼亚)	<code>ms.na-ashburn.tencentcloudapi.com</code>
美国西部(硅谷)	<code>ms.na-siliconvalley.tencentcloudapi.com</code>
北美地区(多伦多)	<code>ms.na-toronto.tencentcloudapi.com</code>
欧洲地区(法兰克福)	<code>ms.eu-frankfurt.tencentcloudapi.com</code>
欧洲地区(莫斯科)	<code>ms.eu-moscow.tencentcloudapi.com</code>

注意：由于金融区和非金融区是隔离不互通的，因此当访问金融区服务时（公共参数 `Region` 为金融区地域），需要同时指定带金融区地域的域名，最好和 `Region` 的地域保持一致。

金融区接入地域	金融区域名
华东地区(上海金融)	<code>ms.ap-shanghai-fsi.tencentcloudapi.com</code>
华南地区(深圳金融)	<code>ms.ap-shenzhen-fsi.tencentcloudapi.com</code>

2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型 :

- application/json (推荐) , 必须使用签名方法 v3 (TC3-HMAC-SHA256) 。
- application/x-www-form-urlencoded , 必须使用签名方法 v1 (HmacSHA1 或 HmacSHA256) 。
- multipart/form-data (仅部分接口支持) , 必须使用签名方法 v3 (TC3-HMAC-SHA256) 。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1 (HmacSHA1、HmacSHA256) 时不得超过1MB。POST 请求使用签名方法 v3 (TC3-HMAC-SHA256) 时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2020-05-11 10:12:08

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

签名方法 v3

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见[文档](#)。

注意：接口文档中的示例由于目的是展示接口参数用法，简化起见，使用的是签名方法 v1 GET 请求，如果依旧想使用签名方法 v1 请参考下文章节。

使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，通常为域名前缀，例如域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 ms； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
X-TC-Token	String	否	临时证书所用的 Token，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

假设用户想要查询广州地域的云服务器实例列表，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST（application/json）请求结构示例：

```
https://cvm.tencentcloudapi.com/
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
```

```
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
{"Offset":0,"Limit":10}
```

HTTP POST (multipart/form-data) 请求结构示例 (仅特定的接口支持) :

```
https://cvm.tencentcloudapi.com/
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b59
24a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
--58731222010402
Content-Disposition: form-data; name="Offset"
```

```
0
--58731222010402
Content-Disposition: form-data; name="Limit"
```

```
10
--58731222010402--
```

签名方法 v1

使用签名方法 v1 (有时会称作 HmacSHA256 和 HmacSHA1), 公共参数需要统一放到请求串中, 完整介绍详见[文档](#)

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口, 取值为 DescribeInstances。
Region	String	-	地域参数, 用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意: 某些接口不需要传递该参数, 接口文档中会对此特别说明, 此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳, 可记录发起 API 请求的时间。例如1529223702, 如果与当前时间相差过大, 会引起签名过期错误。
Nonce	Integer	是	随机正整数, 与 Timestamp 联合起来, 用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId, 一个 SecretId 对应唯一的 SecretKey, 而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名, 用来验证此次请求的合法性, 需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式, 目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时, 才使用 HmacSHA256 算法验证签名, 其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token, 需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

假设用户想要查询广州地域的云服务器实例列表, 其请求结构按照请求 URL、请求头部、请求体示例如下:

HTTP GET 请求结构示例:

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

```
Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/
```

```
Host: cvm.tencentcloudapi.com
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

签名方法 v3

最近更新时间：2020-05-11 10:12:08

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 json 格式，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)。

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共请求参数中指定该签名结果并以指定的方式和格式发送请求。

申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- 用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄漏，请立刻禁用该安全凭证。**

申请安全凭证的具体步骤如下：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云API密钥](#) 的控制台页面。
- 在 [云API密钥](#) 页面，单击【新建密钥】即可以创建一对密钥。

签名过程

云 API 支持 GET 和 POST 请求。对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于 POST 方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式绝大多数接口均支持，multipart 格式只有特定接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

- 云服务器默认已开通，该接口很常用；
- 该接口是只读的，不会改变现有资源的状态；
- 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE。用户想查看广州云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
```

```
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠（/）。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode，字符集 UTF8，推荐使用编程语言标准库，所有特殊字符均需编码，大写形式。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接； 2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号（;）分隔。 此示例为 content-type;host
HashedRequestPayload	请求正文（payload，即 body，此示例为 {"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com

content-type;host
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + '\n' +
RequestTimestamp + '\n' +
CredentialScope + '\n' +
HashedCanonicalRequest
```

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。

字段名称	解释
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。 Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致 ；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031。

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败，返回签名过期错误。

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，此处不展示中间结果。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，此处的伪代码密钥参数在后，请以实际编程语言为准。通常标准库函数会提供二进制格式的计算值，也即此处使用的，也会提供打印友好的十六进制格式的计算值，将在下面计算签名结果时使用。

字段名称	解释
SecretKey	原始的 SecretKey，即 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名，伪代码如下：

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

此示例计算结果是 72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168。

4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
Algorithm + ' ' +
'Credential=' + SecretId + '/' + CredentialScope + ', ' +
'SignedHeaders=' + SignedHeaders + ', ' +
'Signature=' + Signature
```

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。

字段名称	解释
SecretId	密钥对中的 SecretId，即 AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host。
Signature	签名值。此示例计算结果是 72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}
```

5. 签名演示

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_JSON = "application/json; charset=utf-8";

    public static byte[] hmac256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(UTF8));
    }

    public static String sha256Hex(String s) throws Exception {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] d = md.digest(s.getBytes(UTF8));
        return DatatypeConverter.printHexBinary(d).toLowerCase();
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.tencentcloudapi.com";
        String region = "ap-guangzhou";
```

```
String action = "DescribeInstances";
String version = "2017-03-12";
String algorithm = "TC3-HMAC-SHA256";
String timestamp = "1551113065";
//String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
// 注意时区，否则容易出错
sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

// ***** 步骤 1：拼接规范请求串 *****
String httpRequestMethod = "POST";
String canonicalUri = "/";
String canonicalQueryString = "";
String canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n";
String signedHeaders = "content-type;host";

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
String hashedRequestPayload = sha256Hex(payload);
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ***** 步骤 2：拼接待签名字符串 *****
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = sha256Hex(canonicalRequest);
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ***** 步骤 3：计算签名 *****
byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
byte[] secretService = hmac256(secretDate, service);
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ***** 步骤 4：拼接 Authorization *****
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\")")
.append(" -H \"Host: ").append(host).append("\")")
.append(" -H \"X-TC-Action: ").append(action).append("\")")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\")")
.append(" -H \"X-TC-Version: ").append(version).append("\")")
.append(" -H \"X-TC-Region: ").append(region).append("\")")
.append(" -d '").append(payload).append("'");
System.out.println(sb.toString());
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": [u"未命名"]}]}

# ***** 步骤 1 : 拼接规范请求串 *****
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** 步骤 2 : 拼接待签名字符串 *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

# ***** 步骤 3 : 计算签名 *****
# 计算签名摘要函数
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** 步骤 4 : 拼接 Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"
+ ' -H "Host: ' + host + '"')
```

```
+ ' -H "X-TC-Action: ' + action + "'
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + "'
+ ' -H "X-TC-Version: ' + version + "'
+ ' -H "X-TC-Region: ' + region + "'
+ " -d '" + payload + "'")
```

go

```
package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "fmt"
    "time"
)

func sha256hex(s string) string {
    b := sha256.Sum256([]byte(s))
    return hex.EncodeToString(b[:])
}

func hmacsha256(s, key string) string {
    hashed := hmac.New(sha256.New, []byte(key))
    hashed.Write([]byte(s))
    return string(hashed.Sum(nil))
}

func main() {
    secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
    canonicalQueryString := ""
    canonicalHeaders := "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n"
    signedHeaders := "content-type;host"
    payload := `{"Limit": 1, "Filters": [{"Values": [{"u672a\u547d\u540d"}, {"Name": "instance-name"}]}`
    hashedRequestPayload := sha256hex(payload)
    canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
        httpRequestMethod,
        canonicalURI,
        canonicalQueryString,
        canonicalHeaders,
        signedHeaders,
        hashedRequestPayload)
    fmt.Println(canonicalRequest)

    // step 2: build string to sign
    date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
    credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
    hashedCanonicalRequest := sha256hex(canonicalRequest)
    string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
        algorithm,
        timestamp,
        credentialScope,
        hashedCanonicalRequest)
```

```
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = "content-type:application/json; charset=utf-8\n".host."\n";
$signedHeaders = "content-type;host";
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/".$service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
```

```
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=".$secretId."/". $credentialScope
.", SignedHeaders=content-type;host, Signature=".$signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"
.' -H "Content-Type: application/json; charset=utf-8"
.' -H "Host: '.$host.'"
.' -H "X-TC-Action: '.$action.'"
.' -H "X-TC-Timestamp: '.$timestamp.'"
.' -H "X-TC-Version: '.$version.'"
.' -H "X-TC-Region: '.$region.'"
.' -d "'.$payload.'"';
echo $curl.PHP_EOL;
```

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不相符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2020-05-11 10:12:09

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)。

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。

1. 申请安全凭证

在第一次使用云 API 之前，请前往 [云 API 密钥页面](#) 申请安全凭证。安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。**

申请安全凭证的具体步骤如下：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云 API 密钥](#) 的控制台页面
- 在 [云 API 密钥](#) 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。上述示例参

数的排序结果如下:

```
{
  'Action': 'DescribeInstances',
  'InstanceId.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
  'Timestamp': 1465185768,
  'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时, 可对上面示例中的参数进行排序, 得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。 将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式, 如对 Action 参数, 其参数名称为 "Action", 参数值为 "DescribeInstances", 因此格式化后就为 Action=DescribeInstances。 **注意: “参数值”为原始值而非 url 编码后的值。**

然后将格式化后的各个参数用 "&" 拼接在一起, 最终生成的请求字符串为:

```
Action=DescribeInstances&InstanceId.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原字符串

此步骤生成签名原字符串。 签名原字符串由以下几个参数构成:

1. 请求方法: 支持 POST 和 GET 方式, 这里使用 GET 请求, 注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为: cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同, 详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。
4. 请求字符串: 即上一步生成的请求字符串。

签名原串的拼接规则为: 请求方法 + 请求主机 + 请求路径 + ? + 请求字符串。

示例的拼接结果为:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。 首先使用 HMAC-SHA1 算法对上一步中获得的**签名原字符串**进行签名, 然后将生成的签名串使用 Base64 进行编码, 即可获得最终的签名串。

具体代码如下, 以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGel=
```

使用其它程序设计语言开发时, 可用上面示例中的原文进行签名验证, 得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数, 需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGel=, 最终得到的签名串请求参数 (Signature) 为: EliP9YW3pW28FpsEdkXt/+WcGel=, 它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要以 UTF-8 进行编码。

注意：有些编程语言的库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 SDK 中心。当前支持的编程语言有：

- Python
- Java
- PHP
- Go
- JavaScript
- .NET

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt/+WcGel=&Timestamp=1465185768&Version=2017-03-12。`

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";
```

```
public static String sign(String s, String key, String method) throws Exception {
    Mac mac = Mac.getInstance(method);
    SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
    mac.init(secretKeySpec);
    byte[] hash = mac.doFinal(s.getBytes(CHARSET));
    return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
    StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
    // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
    for (String k : params.keySet()) {
        s2s.append(k).append("=").append(params.get(k).toString()).append("&");
    }
    return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间，例如：params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceId.0", "ins-09dx96dg"); // 业务参数
    params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE", "HmacSHA1")); // 公共参数
    System.out.println(getUrl(params));
}
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包：`pip install requests`。

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)
```

```
if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

Go

```
package main

import (
    "bytes"
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "fmt"
    "sort"
)

func main() {
    secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": "1465185768",
        "Region": "ap-guangzhou",
        "SecretId": secretId,
        "Version": "2017-03-12",
        "Action": "DescribeInstances",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": "20",
        "Offset": "0",
    }

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("cvm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

    // sort keys by ascii asc order
    keys := make([]string, 0, len(params))
    for k, _ := range params {
        keys = append(keys, k)
    }
    sort.Strings(keys)

    for i := range keys {
        k := keys[i]
        buf.WriteString(k)
        buf.WriteString("=")
        buf.WriteString(params[k])
    }
}
```

```
buf.WriteString("&")
}
buf.Truncate(buf.Len() - 1)

hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceId.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $params as $key => $value ) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

返回结果

最近更新时间：2019-08-14 19:17:48

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	错误描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	HTTPS 请求方法错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

应用加固相关接口

查询加固结果

最近更新时间：2020-03-05 22:24:10

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

通过唯一标识获取加固的结果

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeShieldResult。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
ItemId	是	String	任务唯一标识

3. 输出参数

参数名称	类型	描述
TaskStatus	Integer	任务状态: 0-请返回,1-已完成,2-处理中,3-处理出错,4-处理超时
AppDetailInfo	AppDetailInfo	app加固前的详细信息
ShieldInfo	ShieldInfo	app加固后的详细信息
StatusDesc	String	状态描述
StatusRef	String	状态指引
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询加固结果

通过唯一标识获取加固的结果，唯一标识ItemId通过请求加固接口返回

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeShieldResult
&ItemId=1shi2e-2387hjgus
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "TaskStatus": 0,
```

```
"StatusDesc": "",
"StatusRef": "",
"AppDetailInfo": {
  "AppName": "微信",
  "AppPkgName": "com.tencent.mm",
  "AppVersion": "1.0",
  "AppSize": 1234565,
  "AppMd5": "dd5b29a800246d7089febf228286d901",
  "AppIconUrl": "https://example.com/1.png"
},
"ShieldInfo": {
  "ShieldCode": 1,
  "ShieldSize": 1024,
  "ShieldMd5": "25787jhids",
  "AppUrl": "https://example.com/1.apk",
  "TaskTime": 1245478744,
  "ItemId": "hji452-huhsxqwq1212"
}
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.ServerError	服务端无法响应。
MissingParameter.MissingItemId	缺少ItemId字段。
ResourceNotFound.ItemIdNotFound	ItemId不存在。

用户查询提交过的app列表

最近更新时间：2020-03-05 22:24:11

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

本接口用于查看app列表。可以通过指定任务唯一标识ItemId来查询指定app的详细信息，或通过设定过滤器来查询满足过滤条件的app的详细信息。指定偏移(Offset)和限制(Limit)来选择结果中的一部分，默认返回满足条件的前20个app信息。

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeShieldInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
Filters.N	否	Array of Filter	支持通过app名称，app包名，加固的服务版本，提交的渠道进行筛选。
Offset	否	Integer	偏移量，默认为0。
Limit	否	Integer	数量限制，默认为20，最大值为100。
ItemIds.N	否	Array of String	可以提供ItemId数组来查询一个或者多个结果。注意不可以同时指定ItemIds和Filters。
OrderField	否	String	按某个字段排序，目前仅支持TaskTime排序。
OrderDirection	否	String	升序（asc）还是降序（desc），默认：desc。

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合要求的app数量
AppSet	Array of AppSetInfo	一个关于app详细信息的结构体，主要包括app的基本信息和加固信息。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询加固实例列表

根据加固实例的包名和名称进行筛选，并返回最新的20条记录

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeShieldInstances
&Filters.0.Name=AppName
&Filters.0.Value=wechat
&Filters.1.Name=AppPkgName
&Filters.1.Value=com.tencent.mm
&Offset=0
&Limit=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "TotalCount": 10,
    "AppSet": [
      {
        "ItemId": "1234-xcse-ddw1",
        "AppName": "微信",
        "AppPkgName": "com.tencent.mm",
        "AppVersion": "6.5",
        "AppMd5": "der2331ds",
        "AppSize": 123454,
        "ServiceEdition": "basic",
        "ShieldCode": 0,
        "AppUrl": "https://www.example.com/a.apk",
        "AppIconUrl": "https://www.example.com/12334",
        "ShieldMd5": "ae5df985a27b07f56d8c670fef70d7c9",
        "ShieldSize": 1193311,
        "TaskStatus": 1,
        "TaskTime": 1524744997
      }
    ]
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.ServerError	服务端无法响应。
InvalidParameterValue.InvalidCoexistItemIdsFilters	不能同时指定ItemIds和Filters。
InvalidParameterValue.InvalidFilter	无效的过滤器。
InvalidParameterValue.InvalidItemIds	ItemIds不合法。

错误码	描述
InvalidParameterValue.InvalidLimit	Limit不是有效的数字。
InvalidParameterValue.InvalidOffset	Offset不是有效的数字。
InvalidParameterValue.InvalidOrderDirection	OrderDirection参数。
InvalidParameterValue.InvalidOrderField	OrderField取值不合法。

批量删除提交过的app信息

最近更新時間：2020-03-05 22:24:12

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

删除一个或者多个app加固信息

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DeleteShieldInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
ItemIds.N	是	Array of String	任务唯一标识ItemId的列表

3. 输出参数

参数名称	类型	描述
Progress	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 根据ItemId对加固实例进行删除

通过传入ItemId集合删除1个或者多个加固实例

输入示例

```
https://ms.tencentcloudapi.com/?Action=DeleteShieldInstances
&ItemIds.0=hhusxu-hui2677-kk
&ItemIds.1=xyuu-csu-ee78236l
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "Progress": 1
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.ServerError	服务端无法响应。
InvalidParameterValue.InvalidItemIds	ItemIds不合法。
MissingParameter.MissingItemIds	缺少ItemIds字段。
ResourceNotFound.ItemIdNotFound	ItemId不存在。

提交加固基础数据

最近更新时间：2020-03-05 22:24:13

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

用户通过该接口提交应用进行应用加固，加固后需通过DescribeShieldResult接口查询加固结果

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreateShieldInstance。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
AppInfo	是	AppInfo	待加固的应用信息
ServiceInfo	是	ServiceInfo	加固服务信息

3. 输出参数

参数名称	类型	描述
Progress	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
ItemId	String	任务唯一标识
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 提交一个app进行加固

通过该接口提交要加固的app信息（示例中的AppUrl和CallbackUrl仅示例，实际无法使用，请用实际可用的url进行替换）

输入示例

```
https://ms.tencentcloudapi.com/?Action=CreateShieldInstance
&AppInfo.AppUrl=http://example.com/1.apk
&AppInfo.AppMd5=dd5b29a800246d7089feb228286d901
&AppInfo.AppSize=1024
&ServiceInfo.ServiceEdition=basic
&ServiceInfo.CallbackUrl=http://example.com/cb
&ServiceInfo.SubmitSource=MC
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "Progress": 1,
    "ItemId": "shgugu-hiw72-334kd"
  }
}
```

```
}  
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.ServerError	服务端无法响应。
InvalidParameter.MissingServiceInfo	ServiceInfo结构体参数缺失。
InvalidParameter.PlanIdNotFound	不能找到指定的加固策略。
MissingParameter.MissingAppInfo	AppInfo结构体参数缺失。
ResourceNotFound.PlanIdNotFound	无法找到指定的加固策略。
ResourceUnavailable	资源不可用。
ResourceUnavailable.NotFound	找不到该资源。
UnauthorizedOperation	未授权操作

创建资源

最近更新时间：2020-03-05 22:24:13

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

用户可以使用该接口自建资源，只支持白名单用户

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreateResourceInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
Pid	是	Integer	资源类型id。13624：加固专业版。
TimeUnit	是	String	时间单位，取值为d，m，y，分别表示天，月，年。
TimeSpan	是	Integer	时间数量。
ResourceNum	是	Integer	资源数量。

3. 输出参数

参数名称	类型	描述
ResourceSet	Array of String	新创建的资源列表。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建加固资源

白名单用户通过创建加固资源自己加固。

输入示例

```
https://ms.tencentcloudapi.com/?Action=CreateResourceInstances
&Pid=12750
&TimeUnit=m
&TimeSpan=6
&ResourceNum=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "2dfd3c41-5606-446f-a437-ca7bbbe07b26",
    "ResourceSet": [
      "mspro_20180913_6b2787b5_0",
```



```
"mspro_20180913_6b2787b5_1",
"mspro_20180913_6b2787b5_2",
"mspro_20180913_6b2787b5_3",
"mspro_20180913_6b2787b5_4"
]
}
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
InvalidParameter	参数错误
LimitExceeded	超过配额限制
ResourceUnavailable.NotFound	找不到该资源。
UnauthorizedOperation	未授权操作
UnauthorizedOperation.NotWhiteUser	不是白名单用户。

查询加固策略

最近更新时间：2020-03-05 22:24:11

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

查询加固策略

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeShieldPlanInstance。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
ResourceId	是	String	资源id
Pid	是	Integer	服务类别id

3. 输出参数

参数名称	类型	描述
BindInfo	BindInfo	绑定资源信息
ShieldPlanInfo	ShieldPlanInfo	加固策略信息
ResourceServiceInfo	ResourceServiceInfo	加固资源信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询加固策略信息

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeShieldPlanInstance
&Pids=12750
&ResourceId=jsihsih-xx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "BindInfo": {
      "AppIconUrl": "",
      "AppName": "微信",
      "AppPkgName": "com.tencent.mm"
    },
    "ShieldPlanInfo": {
```

```
"TotalCount": 5,
"PlanSet": [
{
  "IsDefault": 0,
  "PlanId": 1,
  "PlanName": "部分加固",
  "PlanInfo": {
    "ApkSizeopt": 1,
    "Dex": 1,
    "So": 1,
    "Bugly": 1,
    "AntiRepack": 1,
    "SeperateDex": 0,
    "Db": 0,
    "DexSig": 1,
    "AntiLogLeak": 0,
    "AntiVMP": 0,
    "AntiQemuRoot": 0,
    "AntiAssets": 0,
    "AntiScreenshot": 0,
    "AntiSSL": 0,
    "SoType": [
      "so_low_com_dump_huidu"
    ],
    "SoInfo": {
      "SoFileNames": [
        "1.so",
        "2.so"
      ]
    }
  }
}
],
},
"ResourceServiceInfo": {
  "CreateTime": "",
  "ExpireTime": ""
}
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
ResourceUnavailable.NotBind	资源未绑定应用包名。
ResourceUnavailable.NotFound	找不到该资源。

新增加固策略

最近更新时间：2020-03-05 22:24:12

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

对资源进行策略新增

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreateShieldPlanInstance。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
ResourceId	是	String	资源id
PlanName	是	String	策略名称
PlanInfo	是	PlanInfo	策略具体信息

3. 输出参数

参数名称	类型	描述
PlanId	Integer	策略id
Progress	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 新增加固策略

输入示例

```
https://ms.tencentcloudapi.com/?Action=CreateShieldPlanInstance
&ResourceId=127500-shi
&PlanName=默认加固策略
&PlanInfo.ApkSizeOpt=1
&PlanInfo.Dex=1
&PlanInfo.So=1
&PlanInfo.Bugly=0
&PlanInfo.AntiRepack=1
&PlanInfo.Db=0
&PlanInfo.DexSig=1
&PlanInfo.SeperateDex=0
&PlanInfo.AntiLogLeak=0
&PlanInfo.AntiVMP=0
&PlanInfo.AntiQemuRoot=0
&PlanInfo.AntiAssets=0
&PlanInfo.AntiScreenshot=0
```

```
&PlanInfo.AntiSSL=0
&PlanInfo.SoType=so_low_com_dump_huidu
&PlanInfo.SolInfo.SoFileNames.0=1.so
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "Progress": 1,
    "PlanId": 1
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
ResourceUnavailable	资源不可用。
ResourceUnavailable.NotFound	找不到该资源。
UnauthorizedOperation	未授权操作

应用安全扫描相关接口

批量查询扫描结果

最近更新時間：2020-03-05 22:24:08

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

用户通过CreateScanInstances接口提交应用进行风险批量扫描后，用此接口批量获取风险详细信息,包含漏洞信息，广告信息，插件信息和病毒信息

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeScanResults。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
ItemId	是	String	任务唯一标识
AppMd5s.N	否	Array of String	批量查询一个或者多个app的扫描结果，如果不传表示查询该任务下所提交的所有app

3. 输出参数

参数名称	类型	描述
ScanSet	Array of ScanSetInfo	批量扫描的app结果集
TotalCount	Integer	批量扫描结果的个数
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询扫描结果

通过唯一标识获取加固的结果，唯一标识ItemId通过请求扫描接口返回

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeScanResults
&ItemId=1shi2e-2387hjgus&AppMd5s.0=dd5b29a800246d7089feb228286d901
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8fdbd2b2-2867-4758-b450-5bac2d37440f",
    "ScanSet": [
      {
        "AdInfo": {
          "Banners": [],
          "BoutiqueRecommends": [],
```

```
"FloatWindowses": [],
"IntegralWalls": [],
"NotifyBars": [],
"Spots": []
},
"AppDetailInfo": {
"AppIconUrl": "http://ms-shield-1251001047.cos.ap-guangzhou.myqcloud.com/legu_icon/1255750086/a9d5f689b518beccc54a6f4286d18bcd?sign=q-sign-algorithm=sha1&q-ak=AKIDmQtAxYTAB2iBS8s2DCzazCD2g7OUq4Zw&q-sign-time=1524646732;1524647092&q-key-time=1524646732;1524647092&q-h
eader-list=&q-url-param-list=&q-signature=a8553d5e3c61c2a44f16e9949b409b94d553d3e0",
"AppMd5": "bc80057cf4323db8579e011f8a5f1402",
"AppName": "",
"AppPkgName": "",
"AppSize": 484980,
"AppVersion": "",
"FileName": "testname"
},
"PermissionInfo": {
"PermissionList": [
{
"Permission": "android.permission.READ_EXTERNAL_STORAGE"
},
{
"Permission": "android.permission.WRITE_EXTERNAL_STORAGE"
},
{
"Permission": "android.permission.INTERNET"
}
],
"SensitiveInfo": {
"SensitiveList": [
{
"FilePath": "0000bbd2fdef20ad2fa7caef1a9f1e762532f78c/resources.arsc",
"FileSha": "3513b0e608e075f2e1beb49ba32d59b3e44d7622",
"WordList": [
"手槍"
]
}
]
},
"StatusCode": 0,
"StatusDesc": "",
"StatusRef": "",
"TaskStatus": 1,
"TaskTime": 1546511829,
"VirusInfo": {
"SafeType": 1,
"VirusDesc": "",
"VirusName": ""
},
"VulInfo": {
"VulFileScore": 0,
"VulList": [
{
"RiskLevel": 1,
"VulCode": "<com.unity3d.player.UnityPlayer: void <init>(android.content.ContextWrapper)>",
"VulDesc": "在android中使用Broadcast接收广播的过程中，当receiver没有进行合适的配置时，可能导致攻击者构造广播发送给被攻击receiver。导致被攻击应用接受到错误信息，从而导致进一步的危害。",
"VulFilepath": "classes.dex",
"VulId": "22",
"VulName": "Broadcast Receiver组件暴露风险（含动态注册）",
"VulSolution": "1.如果应用的Receiver组件不必要导出，或者组件配置了intent filter标签，建议在AndroidManifest.xml文件中设置组件的“android:exported”属性为false\n2.如果组件必须要提供给外部应用使用，建议对组件进行权限控制",
"VulSrcType": 1
},
{
"RiskLevel": 2,
"VulCode": "debuggable option do not disable in AndroidManifest",
```



```
"VulDesc": "在AndroidManifest.xml中配置debuggable=true导致程序可以被调试，有数据泄漏的风险",
"VulFilepath": "AndroidManifest.xml",
"VulId": "1",
"VulName": "调试标志未关闭漏洞",
"VulSolution": "在AndroidManifest.xml配置debuggable=false",
"VulSrcType": 0
},
{
  "RiskLevel": 2,
  "VulCode": "allowBackUp option do not disable in AndroidManifest",
  "VulDesc": "在AndroidManifest.xml若将allowBackUp属性设为true，用户即可通过adb backup命令备份文件数据并通过adb restore命令将备份文件导入",
  "VulFilepath": "AndroidManifest.xml",
  "VulId": "2",
  "VulName": "allowBackUp文件备份漏洞",
  "VulSolution": "在应用正式发布之前将AndroidManifest.xml文件中的allowBackUp属性值设置为false",
  "VulSrcType": 0
}
],
"TotalCount": 1
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
MissingParameter.MissingItemId	缺少ItemId字段。
ResourceNotFound.ItemIdNotFound	ItemId不存在。

查询扫描列表

最近更新时间：2020-03-05 22:24:09

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

本接口用于查看app列表。可以通过指定任务唯一标识ItemId来查询指定app的详细信息，或通过设定过滤器来查询满足过滤条件的app的详细信息。指定偏移(Offset)和限制(Limit)来选择结果中的一部分，默认返回满足条件的前20个app信息。

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeScanInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
Filters.N	否	Array of Filter	支持通过app名称，app包名进行筛选
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	数量限制，默认为20，最大值为100。
ItemIds.N	否	Array of String	可以提供ItemId数组来查询一个或者多个结果。注意不可以同时指定ItemIds和Filters。
OrderField	否	String	按某个字段排序，目前仅支持TaskTime排序。
OrderDirection	否	String	升序（asc）还是降序（desc），默认：desc。

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合要求的app数量
ScanSet	Array of AppScanSet	一个关于app详细信息的结构体，主要包括app的基本信息和扫描状态信息。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询扫描实例列表

根据提交实例的包名和名称进行筛选，并返回最新的20条记录

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeScanInstances
&Filters.0.Name=AppName
&Filters.0.Value=wechat
&Filters.1.Name=AppPkgName
&Filters.1.Value=com.tencent.mm
&Offset=0
&Limit=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "TotalCount": 1,
    "ScanSet": [
      {
        "ItemId": "1234-xcse-ddw1",
        "AppName": "微信",
        "AppPkgName": "com.tencent.mm",
        "AppVersion": "6.5",
        "AppMd5": "der2331ds",
        "AppSize": 123454,
        "TaskTime": 1524744997,
        "AppSid": "wiqhiqxsqx282787",
        "ScanCode": 0,
        "AppIconUrl": "https://www.example.com/12334",
        "TaskStatus": 1
      }
    ]
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
InvalidParameterValue.InvalidFilter	无效的过滤器。
InvalidParameterValue.InvalidItemIds	ItemIds不合法。
InvalidParameterValue.InvalidLimit	Limit不是有效的数字。
InvalidParameterValue.InvalidOffset	Offset不是有效的数字。

错误码	描述
InvalidParameterValue.InvalidOrderField	OrderField取值不合法。
LimitExceeded	超过配额限制

批量删除提交过的app扫描信息

最近更新时间：2020-03-05 22:24:09

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

删除一个或者多个app扫描信息

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DeleteScanInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
AppSids.N	是	Array of String	删除一个或多个扫描的app，最大支持20个

3. 输出参数

参数名称	类型	描述
Progress	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 根据AppSid对扫描实例进行删除

通过传入AppSid集合删除1个或者多个扫描实例

输入示例

```
https://ms.tencentcloudapi.com/?Action=DeleteScanInstances
&AppSids.0=hhussxu-hui2677-kk
&AppSids.1=xyuu-csu-ee78236l
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "Progress": 1
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
LimitExceeded	超过配额限制

批量提交安全扫描

最近更新時間：2020-03-05 22:24:10

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

用户通过该接口批量提交应用进行应用扫描，扫描后需通过DescribeScanResults接口查询扫描结果

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreateScanInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
AppInfos.N	是	Array of AppInfo	待扫描的app信息列表，一次最多提交20个
ScanInfo	是	ScanInfo	扫描信息

3. 输出参数

参数名称	类型	描述
ItemId	String	任务唯一标识
Progress	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
AppMd5s	Array of String	提交成功的app的md5集合
LimitCount	Integer	剩余可用次数
LimitTime	Integer	到期时间
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 提交一个app进行扫描

通过该接口提交要扫描的app信息（示例中的AppUrl和CallbackUrl仅示例，实际无法使用，请用实际可用的url进行替换）

输入示例

```
https://ms.tencentcloudapi.com/?Action=CreateScanInstances
&AppInfos.0.AppUrl=http://example.com/1.apk
&AppInfos.0.AppMd5=dd5b29a800246d7089febf228286d901
&AppInfos.0.AppSize=1024
&ScanInfo.CallbackUrl=http://example.com/cb
&ScanInfo.ScanTypes.0=ADSCAN
&ScanInfo.ScanTypes.1=VULSCAN
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "Progress": 1,
    "ItemId": "shgugu-hiw72-334kd",
    "AppMd5s": [
      "sashiuashiah",
      "hugugusaw76712"
    ],
    "LimitTime": 0,
    "LimitCount": 0
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
LimitExceeded	超过配额限制
MissingParameter.MissingAppInfo	AppInfo结构体参数缺失。

用户服务信息相关接口

获取用户的所有资源信息

最近更新时间：2020-03-05 22:24:07

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

获取某个用户的所有资源信息

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeResourceInstances。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
Pids.N	是	Array of Integer	资源类别id数组，13624：加固专业版，12750：企业版。空数组表示返回全部资源。
Filters.N	否	Array of Filter	支持通过资源id，pid进行查询
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	数量限制，默认为20，最大值为100。
OrderField	否	String	按某个字段排序，目前支持CreateTime、ExpireTime其中的一个排序。
OrderDirection	否	String	升序（asc）还是降序（desc），默认：desc。

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合要求的资源数量
ResourceSet	Array of ResourceInfo	符合要求的资源数组
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取某个用户的加固资源信息

获取某个用户的加固资源信息

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeResourceInstances
&Pids.0=12750
&Offset=0
&Limit=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "TotalCount": 10,
    "ResourceSet": [
      {
        "ResourceId": "",
        "ResourceName": "应用加固",
        "Pid": "",
        "CreateTime": "",
        "ExpireTime": "",
        "IsBind": 1,
        "BindInfo": {
          "AppIcon": "",
          "AppName": "微信",
          "AppPkgName": "com.tencent.mm"
        }
      }
    ]
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.ServerError	服务端无法响应。
InvalidParameterValue.InvalidFilter	无效的过滤器。
InvalidParameterValue.InvalidLimit	Limit不是有效的数字。
InvalidParameterValue.InvalidOffset	Offset不是有效的数字。
LimitExceeded	超过配额限制
ResourceUnavailable	资源不可用。
ResourceUnavailable.NotFound	找不到该资源。

错误码	描述
UnauthorizedOperation	未授权操作

将应用绑定到资源

最近更新时间：2020-03-05 22:24:08

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

将应用和资源进行绑定

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreateBindInstance。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
ResourceId	是	String	资源id，全局唯一
AppIconUrl	是	String	app的icon的url
AppName	是	String	app的名称
AppPkgName	是	String	app的包名

3. 输出参数

参数名称	类型	描述
Progress	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 将应用绑定到资源

将应用绑定到资源

输入示例

```
https://ms.tencentcloudapi.com/?Action=CreateBindInstance
&ResourceId=hsih12923-ioio
&AppIconUrl=http://q.com/1.png
&AppName=微信
&AppPkgName=com.tencent.mm
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "5e93a212-ca01-0fdc-eedd-5a1fce5e83e6",
    "Progress": 1
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
ResourceUnavailable	资源不可用。
ResourceUnavailable.NotFound	找不到该资源。

其他接口

获取用户基础信息

最近更新时间：2020-03-05 22:24:14

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

获取用户基础信息

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeUserBaseInfoInstance。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
UserUin	Integer	用户uin信息
UserAppid	Integer	用户APPID信息
TimeStamp	Integer	系统时间戳
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取用户基础信息

输入示例

```
https://ms.tencentcloudapi.com/?Action=DescribeUserBaseInfoInstance
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "ce5e66e0-aab5-4d31-9b98-c52caf0fdfae",
    "TimeStamp": 1544607650,
    "UserAppid": 1251006373,
    "UserUin": 20548499
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

获取云COS临时密钥

最近更新時間：2020-03-05 22:24:15

1. 接口描述

接口请求域名：ms.tencentcloudapi.com。

获取云COS文件存储临时密钥，密钥仅限于临时上传文件，有访问限制和时效性。

默认接口请求频率限制：20次/秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreateCosSecKeyInstance。
Version	是	String	公共参数，本接口取值：2018-04-08。
Region	否	String	公共参数，本接口不需要传递此参数。
CosRegion	否	String	地域信息，例如广州：ap-guangzhou，上海：ap-shanghai，默认为广州。
Duration	否	Integer	密钥有效时间，默认为1小时。

3. 输出参数

参数名称	类型	描述
CosAppId	Integer	COS密钥对应的AppId
CosBucket	String	COS密钥对应的存储桶名
CosRegion	String	存储桶对应的地域
ExpireTime	Integer	密钥过期时间
CosId	String	密钥ID信息
CosKey	String	密钥KEY信息
CosToken	String	密钥TOKEN信息
CosPrefix	String	密钥可访问的文件前缀人。例如：CosPrefix=test/123/666，则该密钥只能操作test/123/666为前缀的文件，例如test/123/666/1.txt
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取COS临时密钥

输入示例

```
https://ms.tencentcloudapi.com/?Action=CreateCosSecKeyInstance
&CosRegion=ap-guangzhou
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "CosAppid": 123456,
    "CosBucket": "ms-shield",
    "CosId": "AKIDzgG3O5Cm9ii31sTgph1XhFISnvKPw0Zi",
    "CosKey": "55VuqLWV4HKvuYom4tYkn6FdXpVoM7hz",
    "CosPrefix": "pctool/123456789/1542613158",
    "CosRegion": "ap-guangzhou",
    "CosToken": "13606435fd46b2765dd01aa4eaf356dfca88817030001",
    "ExpireTime": 3600,
    "RequestId": "ce5e66e0-aab5-4d31-9b98-c52caf0fdfae"
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InternalServerError.ServerError	服务端无法响应。
ResourceUnavailable	资源不可用。

数据结构

最近更新时间：2020-03-23 08:10:38

AdInfo

广告信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
Spots	Array of PluginInfo	插播广告列表
BoutiqueRecommends	Array of PluginInfo	精品推荐广告列表
FloatWindowses	Array of PluginInfo	悬浮窗广告列表
Banners	Array of PluginInfo	banner广告列表
IntegralWalls	Array of PluginInfo	积分墙广告列表
NotifyBars	Array of PluginInfo	通知栏广告列表

AppDetailInfo

app的详细基础信息

被如下接口引用：DescribeScanResults、DescribeShieldResult。

名称	类型	描述
AppName	String	app的名称
AppPkgName	String	app的包名
AppVersion	String	app的版本号
AppSize	Integer	app的大小
AppMd5	String	app的md5
AppIconUrl	String	app的图标url
FileName	String	app的文件名称

AppInfo

提交的app基本信息

被如下接口引用：CreateScanInstances、CreateShieldInstance。

名称	类型	必选	描述
AppUrl	String	是	app的url，必须保证不用权限校验就可以下载
AppMd5	String	是	app的md5，需要正确传递
AppSize	Integer	否	app的大小
FileName	String	否	app的文件名
AppPkgName	String	否	app的包名，需要正确的传递此字段
AppVersion	String	否	app的版本号

名称	类型	必选	描述
AppIconUrl	String	否	app的图标url
AppName	String	否	app的名称

AppScanSet

扫描后app的信息，包含基本信息和扫描状态信息

被如下接口引用：DescribeScanInstances。

名称	类型	描述
ItemId	String	任务唯一标识
AppName	String	app的名称
AppPkgName	String	app的包名
AppVersion	String	app的版本号
AppMd5	String	app的md5
AppSize	Integer	app的大小
ScanCode	Integer	扫描结果返回码
TaskStatus	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
TaskTime	Integer	提交扫描时间
AppIconUrl	String	app的图标url
AppSid	String	标识唯一该app，主要用于删除
SafeType	Integer	安全类型:1-安全软件，2-风险软件，3病毒软件
VulCount	Integer	漏洞个数

AppSetInfo

加固后app的信息，包含基本信息和加固信息

被如下接口引用：DescribeShieldInstances。

名称	类型	描述
ItemId	String	任务唯一标识
AppName	String	app的名称
AppPkgName	String	app的包名
AppVersion	String	app的版本号
AppMd5	String	app的md5
AppSize	Integer	app的大小
ServiceEdition	String	加固服务版本
ShieldCode	Integer	加固结果返回码
AppUrl	String	加固后的APP下载地址
TaskStatus	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
ClientIp	String	请求的客户端ip

名称	类型	描述
TaskTime	Integer	提交加固时间
AppIconUrl	String	app的图标url
ShieldMd5	String	加固后app的md5
ShieldSize	Integer	加固后app的大小

BindInfo

用户绑定app的基本信息

被如下接口引用：DescribeResourceInstances、DescribeShieldPlanInstance。

名称	类型	描述
AppIconUrl	String	app的icon的url
AppName	String	app的名称
AppPkgName	String	app的包名

Filter

筛选数据结构

被如下接口引用：DescribeResourceInstances、DescribeScanInstances、DescribeShieldInstances。

名称	类型	必选	描述
Name	String	是	需要过滤的字段
Value	String	否	需要过滤字段的值

PlanDetailInfo

加固策略具体信息

被如下接口引用：DescribeShieldPlanInstance。

名称	类型	描述
IsDefault	Integer	默认策略，1为默认，0为非默认
PlanId	Integer	策略id
PlanName	String	策略名称
PlanInfo	PlanInfo	策略信息

PlanInfo

加固策略信息

被如下接口引用：CreateShieldPlanInstance、DescribeShieldPlanInstance。

名称	类型	描述
ApkSizeOpt	Integer	apk大小优化，0关闭，1开启
Dex	Integer	Dex加固，0关闭，1开启

名称	类型	描述
So	Integer	So加固，0关闭，1开启
Bugly	Integer	数据收集，0关闭，1开启
AntiRepack	Integer	防止重打包，0关闭，1开启
SeperateDex	Integer	Dex分离，0关闭，1开启
Db	Integer	内存保护，0关闭，1开启
DexSig	Integer	Dex签名校验，0关闭，1开启
SoInfo	SoInfo	So文件信息
AntiVMP	Integer	vmp，0关闭，1开启
SoType	Array of String	保护so的强度，
AntiLogLeak	Integer	防日志泄漏，0关闭，1开启
AntiQemuRoot	Integer	root检测，0关闭，1开启
AntiAssets	Integer	资源防篡改，0关闭，1开启
AntiScreenshot	Integer	防止截屏，0关闭，1开启
AntiSSL	Integer	SSL证书防窃取，0关闭，1开启

PluginInfo

插件信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
PluginType	Integer	插件类型，分别为 1-通知栏广告，2-积分墙广告，3-banner广告，4- 悬浮窗图标广告，5-精品推荐列表广告, 6-插播广告
PluginName	String	插件名称
PluginDesc	String	插件描述

ResourceInfo

拉取某个用户的所有资源信息

被如下接口引用：DescribeResourceInstances。

名称	类型	描述
ResourceId	String	用户购买的资源id，全局唯一
Pid	Integer	资源的pid，MTP加固-12767，应用加固-12750 MTP反作弊-12766 源代码混淆-12736
CreateTime	Integer	购买时间戳
ExpireTime	Integer	到期时间戳
IsBind	Integer	0-未绑定，1-已绑定
BindInfo	BindInfo	用户绑定app的基本信息
ResourceName	String	资源名称，如应用加固，漏洞扫描

ResourceServiceInfo

资源服务信息

被如下接口引用：DescribeShieldPlanInstance。

名称	类型	描述
CreateTime	Integer	创建时间戳
ExpireTime	Integer	到期时间戳
ResourceName	String	资源名称，如应用加固，源码混淆

ScanInfo

需要扫描的应用的服务信息

被如下接口引用：CreateScanInstances。

名称	类型	必选	描述
CallbackUrl	String	是	任务处理完成后的反向通知回调地址,批量提交app每扫描完成一个会通知一次,通知为POST请求，post信息{ItemId:
ScanTypes	Array of String	是	VULSCAN-漏洞扫描信息，VIRUSSCAN-返回病毒扫描信息，ADSCAN-广告扫描信息，PLUGINS SCAN-插件扫描信息，PERMISSION-系统权限信息，SENSITIVE-敏感词信息，可以自由组合

ScanPermissionInfo

安全扫描系统权限信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
Permission	String	系统权限

ScanPermissionList

安全扫描系统权限信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
PermissionList	Array of ScanPermissionInfo	系统权限信息

ScanSensitiveInfo

安全扫描敏感词

被如下接口引用：DescribeScanResults。

名称	类型	描述
WordList	Array of String	敏感词
FilePath	String	敏感词对应的文件信息
FileSha	String	文件sha1值

ScanSensitiveList

安全扫描敏感词列表

被如下接口引用：DescribeScanResults。

名称	类型	描述
SensitiveList	Array of ScanSensitiveInfo	敏感词列表

ScanSetInfo

app扫描结果集

被如下接口引用：DescribeScanResults。

名称	类型	描述
TaskStatus	Integer	任务状态: 1-已完成,2-处理中,3-处理出错,4-处理超时
AppDetailInfo	AppDetailInfo	app信息
VirusInfo	VirusInfo	病毒信息
VulInfo	VulInfo	漏洞信息
AdInfo	AdInfo	广告插件信息
TaskTime	Integer	提交扫描的时间
StatusCode	Integer	状态码，成功返回0，失败返回错误码
StatusDesc	String	状态描述
StatusRef	String	状态操作指引
PermissionInfo	ScanPermissionList	系统权限信息
SensitiveInfo	ScanSensitiveList	敏感词列表

ServiceInfo

提交app加固的服务信息

被如下接口引用：CreateShieldInstance。

名称	类型	必选	描述
ServiceEdition	String	是	服务版本，基础版basic，专业版professional，企业版enterprise。
CallbackUrl	String	是	任务处理完成后的反向通知回调地址，如果不需要通知请传递空字符串。通知为POST请求，post包体数据示例{"Response": {"ItemId": "4cdad8fb86f036b06bccb3f58971c306", "ShieldCode": 0, "ShieldMd5": "78701576793c4a5f04e1c9660de0aa0b", "SI调用方需要返回如下信息，{"Result": "ok", "Reason": "xxxxx"}，如果Result字段值不等于ok会继续回调。
SubmitSource	String	是	提交来源 YYB-应用宝 RDM-rdm MC-控制台 MAC_TOOL-mac工具 WIN_TOOL-window工具。
PlanId	Integer	否	加固策略编号，如果不传则使用系统默认加固策略。如果指定的plan不存在会返回错误。

ShieldInfo

加固后app的信息

被如下接口引用：DescribeShieldResult。

名称	类型	描述
ShieldCode	Integer	加固结果的返回码
ShieldSize	Integer	加固后app的大小

名称	类型	描述
ShieldMd5	String	加固后app的md5
AppUrl	String	加固后的APP下载地址，该地址有效期为20分钟，请及时下载
TaskTime	Integer	加固的提交时间
ItemId	String	任务唯一标识
ServiceEdition	String	加固版本，basic基础版，professional专业版，enterprise企业版

ShieldPlanInfo

加固策略信息

被如下接口引用：DescribeShieldPlanInstance。

名称	类型	描述
TotalCount	Integer	加固策略数量
PlanSet	Array of PlanDetailInfo	加固策略具体信息数组

SoInfo

so加固信息

被如下接口引用：CreateShieldPlanInstance、DescribeShieldPlanInstance。

名称	类型	描述
SoFileNames	Array of String	so文件列表

VirusInfo

病毒信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
SafeType	Integer	软件安全类型，分别为0-未知、1-安全软件、2-风险软件、3-病毒软件
VirusName	String	病毒名称，utf8编码，非病毒时值为空
VirusDesc	String	病毒描述，utf8编码，非病毒时值为空

VulInfo

漏洞信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
VulList	Array of VulList	漏洞列表
VulFileScore	Integer	漏洞文件评分

VulList

漏洞信息

被如下接口引用：DescribeScanResults。

名称	类型	描述
VulId	String	漏洞id
VulName	String	漏洞名称
VulCode	String	漏洞代码
VulDesc	String	漏洞描述
VulSolution	String	漏洞解决方案
VulSrcType	Integer	漏洞来源类别，0默认自身，1第三方插件
VulFilepath	String	漏洞位置
RiskLevel	Integer	风险级别：1 低风险；2中等风险；3 高风险

错误码

最近更新时间：2020-05-11 10:12:10

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
UnsupportedOperation	操作不支持。
ResourceInUse	资源被占用。
InternalError	内部错误。
RequestLimitExceeded	请求的次数超过了频率限制。
AuthFailure.SecretIdNotFound	密钥不存在。请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
LimitExceeded	超过配额限制。
NoSuchVersion	接口版本不存在。
ResourceNotFound	资源不存在。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
UnsupportedRegion	接口不支持所传地域。
UnauthorizedOperation	未授权操作。
InvalidParameter	参数错误。
ResourceUnavailable	资源不可用。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.TokenFailure	token 错误。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
UnknownParameter	未知参数错误。

错误码	说明
UnsupportedProtocol	HTTP(S)请求协议错误，只支持 GET 和 POST 请求。
InvalidParameterValue	参数取值错误。
InvalidAction	接口不存在。
MissingParameter	缺少参数错误。
ResourceInsufficient	资源不足。

业务错误码

错误码	说明
InternalServerError.ServerError	服务端无法响应。
InvalidParameter.MissingServiceInfo	ServiceInfo结构体参数缺失。
InvalidParameter.PlanIdNotFound	不能找到指定的加固策略。
InvalidParameterValue.InvalidCoexistItemIdsFilters	不能同时指定ItemIds和Filters。
InvalidParameterValue.InvalidFilter	无效的过滤器。
InvalidParameterValue.InvalidItemIds	ItemIds不合法。
InvalidParameterValue.InvalidLimit	Limit不是有效的数字。
InvalidParameterValue.InvalidOffset	Offset不是有效的数字。
InvalidParameterValue.InvalidOrderDirection	OrderDirection参数。
InvalidParameterValue.InvalidOrderField	OrderField取值不合法。
MissingParameter.MissingAppInfo	AppInfo结构体参数缺失。
MissingParameter.MissingItemId	缺少ItemId字段。
MissingParameter.MissingItemIds	缺少ItemIds字段。
ResourceNotFound.ItemIdNotFound	ItemId不存在。
ResourceNotFound.PlanIdNotFound	无法找到指定的加固策略。
ResourceUnavailable.NotBind	资源未绑定应用包名。
ResourceUnavailable.NotFound	找不到该资源。
UnauthorizedOperation.NotWhiteUser	不是白名单用户。