

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
| 青藤万相 |  |
| 安装指导书 |  |
| v3.4.0.10LTS |  |
|  |  |
| 让主机稳定又安全  2022.04 |  |

|  |
| --- |
| ■ 版权声明 |
| 本文中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除另有特别注明，版权均属**青藤云安全**所有，受到有关产权及版权法保护。任何个人、机构未经**青藤云安全**的书面授权许可，不得以任何方式复制或引用本文的任何片断。 |
| ■ 商标声明 |
| 和其他**青藤**商标均为**青藤云安全**的商标。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。 |
| ■ 适用性声明 |
| 本文档为北京升鑫网络有限公司（以下简称“青藤云安全”）的产品功能介绍，适用于用户使用及参考。 |

|  |  |  |
| --- | --- | --- |
| 时间 | 版本 | 说明 |
| 2022.04 | v3.4.0.10LTS | 第一次发布 |
| - | - | - |

**目录**

[1 一级标题（F1） 1](#_Toc99806772)

[1.1 二级标题（F2） 1](#_Toc99806773)

[1.1.1 三级标题（F3） 1](#_Toc99806774)

[1.1.2 三级标题2 2](#_Toc99806775)

[2 一级标题（F1） 4](#_Toc99806776)

[2.1 二级标题（F2） 4](#_Toc99806777)

[2.1.1 三级标题（F3） 4](#_Toc99806778)

# 概述

青藤 API提供了青藤产品中部分功能的API获取接口，您可以利用该接口，灵活的获取各维度的信息，当前API提供如下查询 。

| 功能名称 | 接口作用 | 接口名 |
| --- | --- | --- |
| 资产清点 - 主机信息 | 主机扫描及查询 | /external/api/assets/host |
| 资产清点 - 进程信息 | 进程扫描及查询 | /external/api/assets/process |
| 资产清点 - 端口信息 | 端口扫描及查询 | /external/api/assets/port |
| 资产清点 - 帐号信息 | 帐号扫描及查询 | /external/api/assets/account |
| 资产清点 - 用户组信息 | 用户组扫描及查询 | /external/api/assets/accountgroup |
| 资产清点 - Web站点信息 | web站点扫描及查询 | /external/api/assets/website |
| 资产清点 - Web 应用信息 | web应用扫描及查询 | /external/api/assets/webapp |
| 资产清点 - 软件应用 | 软件应用扫描及查询 | /external/api/assets/app |
| 资产清点 - Web应用框架 | Web应用框架扫描及查询 | /external/api/assets/webframe |
| 资产清点 - 数据库 | 数据库扫描及查询 | /external/api/assets/dbinfo |
| 资产清点 - 启动项 | 启动项扫描及查询 | /external/api/assets/service |
| 资产清点 - 计划任务 | 计划任务扫描及查询 | /external/api/assets/task |
| 资产清点 - 环境变量 | 环境变量扫描及查询 | /external/api/assets/env |
| 资产清点 - 内核模块 | 内核模块扫描及查询 | /external/api/assets/kernelmodule |
| 资产清点 - 安装包 | 安装包扫描及查询 | /external/api/assets/pkg |
| 资产清点 - Jar包 | Jar包扫描及查询 | /external/api/assets/jar\_pkg |
| 合规基线 | 基线扫描及查询 | /external/api/baseline |
| 风险发现 - Web风险文件 | web风险文件扫描及查询 | /external/api/websecurity/weakfile |
| 风险发现 - 应用风险 | 应用风险扫描及查询 | /external/api/vul/app |
| 风险发现 - 系统风险 | 系统风险扫描及查询 | /external/api/vul/system |
| 风险发现 - 对外访问性 | 对外访问性扫描及查询 | /external/api/vul/access |
| 风险发现 - 账号风险 | 账号风险扫描及查询 | /external/api/vul/account |
| 风险发现 - Linux弱密码 | Linux弱密码扫描及查询 | /external/api/vul/weakpwd/linux |
| 风险发现 - Windows弱密码 | Windows弱密码扫描及查询 | /external/api/vul/weakpwd/win |
| 风险发现 - 安全补丁 | 补丁扫描及查询 | /external/api/vul/patch |
| 风险发现 - Linux漏洞检测 | Linux漏洞扫描及查询 | /external/api/vul/poc/linux |
| 风险发现 - 漏洞检测作业管理 | Linux漏洞扫描作业管理及查询 | /external/api/vul/poc/job |
| 入侵检测 - 暴力破解 | 暴力破解封停/解封及数据查询 | /external/api/detect/brutecrack |
| 入侵检测 - 异常登录 | 异常登录规则设置及数据查询 | /external/api/detect/abnormallogin |
| 入侵检测 - 反弹shell | 反弹shell数据查询 | /external/api/detect/bounceshell |
| 入侵检测 - 本地/容器内提权 | 本地提权数据查询 | /external/api/detect/localrights |
| 入侵检测 - 可疑操作 | 可疑操作数据查询 | /external/api/detect/shelllog |
| 入侵检测 - 网络蜜罐 | 网络蜜罐规则设置及数据查询 | /external/api/detect/honeypot |
| 入侵检测 - Web后门 | web后门扫描及数据查询 | /external/api/websecurity/webshell |
| 入侵检测 - Linux后门检测 | Linux后门检测扫描及数据查询 | /external/api/detect/backdoor/linux |
| 入侵检测 - Windows后门检测 | Windows后门检测扫描及数据查询 | /external/api/detect/backdoor/win |
| 系统审计日志查询 | 系统审计日志数据查询 | /external/api/system/audit |
| 快速任务 | 任务的建立、扫描及数据查询 | /external/api/fastjob |
| 微隔离 - 主机一键隔离 | 查询一键隔离列表 | GET /external/api/ms-srv/api/segmentation/list |
| 微隔离 - 主机一键隔离 | 查询隔离详情 | GET /external/api/ms-srv/api/segmentation/detail |
| 微隔离 - 主机一键隔离 | 创建隔离接口 | POST /external/api/ms-srv/api/segmentation/create |
| 微隔离 - 主机一键隔离 | 修改隔离接口 | POST /external/api/ms-srv/api/segmentation/edit |
| 微隔离 - 主机一键隔离 | 解除隔离接口 | DELETE /external/api/ms-srv/api/segmentation/del |
| 微隔离 - 主机一键隔离 | 删除隔离接口 | DELETE /external/api/ms-srv/api/segmentation/realDel |
| 微隔离 - 主机一键隔离 | 重试隔离接口 | POST /external/api/ms-srv/api/segmentation/retry |
| 微隔离 - 主机管理 | 查询主机列表 | GET /external/api/ms-srv/api/hosts/list |
| 微隔离 - 主机管理 | 修改主机开关 | POST /external/api/ms-srv/api/hosts/ms-enable |
| 微隔离 - 拦截黑名单 | 查询黑名单策略列表 | GET /external/api/ms-srv/api/black-strategy/list |
| 微隔离 - 拦截黑名单 | 创建黑名单策略 | POST /external/api/ms-srv/api/black-strategy/create |
| 微隔离 - 拦截黑名单 | 修改黑名单策略 | POST /external/api/ms-srv/api/black-strategy/update |
| 微隔离 - 拦截黑名单 | 修改黑名单策略开关 | POST /external/api/ms-srv/api/black-strategy/update-switch |
| 微隔离 - 拦截黑名单 | 删除黑名单策略接口 | DELETE /external/api/ms-srv/api/black-strategy/delete |
| 微隔离 - 拦截黑名单 | 查看策略主机范围 | GET /external/api/ms-srv/api/black-strategy/strategy-host-list |
| 微隔离 - 拦截黑名单 | 查看策略详情 | GET /external/api/ms-srv/api/black-strategy/detail |

# 请求方法

## 基本请求方法

所有api都按照restful请求方式，返回结果都是json数据格式。说明：

* url为请求地址，对应的服务器地址为php服务器地址，端口为80，如http://${server}:80/v1/api/auth；
* 请求业务api之前，需要经过认证请求；业务api请求均需要经过参数签名，详细见下文的认证及签名说明；
* 对于GET方式的api，参数统一放在url中进行传递，参数使用page=1&size=50&sorts=id,-name&key1=value1&key2=value2的类似form表单的方式提交。注：sorts字段的约定不写表示升序，"-"表示降序，key1,key2为搜索的参数；
* 对于POST,PUT,DELETE的api，参数统一放在body中进行传递，必须设置Header：ContentType:application/json，返回数据为json格式，各接口的返回详见下文各接口说明；

## 身份认证

在请求业务API之前，需要经过身份验证，获取相关参数，通过后才能请求业务api。

**调用接口：**POST /v1/api/auth  
**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| username | String | 用户名，为青藤console系统中的用户 |
| password | String | 密码 |

**请求示例：**

{  
 "username": "zhangsan@qingteng.cn",  
 "password": "test1234"  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": null,  
 "errorDesc": null,  
 "data": {  
 "comId": "59080851823593e1a80b",  
 "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzaWduS2V5IjoiZTAwMzU2NGUzMzliNGEwZjk5ODIxODJlNTg2NWUzYTciLCJjb21JZCI6IjU5MDgwODUxODIzNTkzZTFhODBiIiwidXVpZCI6IjU5MDgwODUxODIzNTkzZTFhODBiIiwiaWF0IjoxNTIyNjQzMjI4OTYyfQ.4koI1tlBZU5\_ycpWENH3JyflDRYyUxeb70GdHjrjpYU",  
 "signKey": "e003564e339b4a0f9982182e5865e3a7"  
 }  
}

**返回部分说明：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| success | bool | 认证是否成功，true：成功，false：失败 |
| errorCode | String | 错误码，认证失败时可用 |
| errorDesc | String | 错误描述，认证失败时可用 |
| comId | String | 公司id |
| jwt | String | jwt认证后的token串 |
| signKey | String | 签名key，参数签名中使用的签名key |

**认证失败返回的错误码：**

| 错误码 | 说明 |
| --- | --- |
| LOGIN\_ERROR | 登录失败 |
| HEADER\_NULL\_COM\_ID | comId请求头不能为空 |
| HEADER\_NULL\_TIMESTAMP | timestamp请求头不能为空 |
| HEADER\_NULL\_SIGN | sign请求头不能为空 |
| AUTH\_SIGN\_FAILED | 验证签名失败 |
| AUTH\_VERIFY\_JWT\_FAILED | 验证jwt失败 |
| INVALID\_JWT | 非法的jwt串 |
| HEADER\_NULL\_AUTHORIZATION | authorization请求头不能为空 |
| AUTH\_ERROR | 认证发生错误 |

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "127.0.0.1"  
port = 80  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@xx.com", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)

## 请求签名

请求业务api，对请求参数有对应的参数签名验证，签名步骤如下：

1. 通过登录认证步骤获得jwt、comId、signKey；
2. 生成请求的timestamp；
3. 对于get请求，将请求参数按照参数名排序(自然升序)，将排序后的请求参数及值，和comId、timestamp、signKey按照以下形式拼接，得到string-to-sign  
   格式：{comId}{key1value1key2value2}{timestamp}{signKey}；
4. 对于put/post/delete请求，将body中的json作为参数，按照以下形式拼接，得到string-to-sign  
   格式：{comId}{body}{timestamp}{signKey}；
5. 对string-to-sign进行Hash，得到sign字符串。Hash算法使用SHA1。

组装请求头，通过http header传递以下参数：

| key | 说明 | 示例 |
| --- | --- | --- |
| comId | 公司id，认证返回 | 59080851823593e1a80b |
| timestamp | 时间戳 | 1522037099 |
| sign | 签名，计算的签名串 | 450af724f3e1c6bf8c8f41f715a1b2daf8e2f105 |
| Authorization | 认证信息，包含认证的jwt | Bearer ${jwt} |

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "127.0.0.1"  
port = 80  
  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 print info  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 print to\_sign  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
  
# get请求示例->获取linux主机列表  
def get\_linux\_host():  
 url = "http://%s:%s/external/api/assets/host/linux?page=0&size=50" % (host, port)  
 # 即是get请求后面的url参数键值对  
 data = {'page': 0, 'size': 50}  
 res = send\_request("GET", url, data)  
 print res  
  
  
# post请求示例->执行linux基线扫描任务  
def post\_execute\_linux\_baseline\_job():  
 data = {'specId': '5b83764c7d761b72e19e38b6'}  
 url = "http://%s:%s/external/api/baseline/job/linux/execute" % (host, port)  
 res = send\_request("POST", url, data)  
 print res

## 通用Get请求参数类型及说明

1. DateRange: 表示请求时间范围，yyyy-MM-dd HH:mm:ss - yyyy-MM-dd HH:mm:ss 前后用“ - ”(注意前后续的空格)分割为start和end，end部分可以省略，省略后表示到现在
2. Date：时间，yyyy-MM-dd HH:mm:ss
3. 多值参数表示：通过逗号分割，如<http://www.example.org/group=1,2,3>
4. 请求参数分页：page页码，从0开始；size每页条数；sorts:”+field1,-field2”,正序省略，倒序使用-号前缀，如<http://example.org/hosts?page=0&size=50&sorts=ip,-desc>

## 数据api请求结果

所有接口通过http status=200来判断接口调用是否成功，如果http status != 200 会有如下输出，errorMessage为错误详情。  
接口异常返回数据格式如下：

{  
 "errorCode": 401, // 错误码，含义与标准httpresponse的status一致  
 "errorMessage": "", // 错误描述  
 "detail":null  
}

# 资产清点

## 资产通用查询扫描状态

用于查询资产信息的扫描状态。

**调用接口：**

GET /external/api/assets/refreshjob/{jobId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 进程信息扫描接口返回的扫描任务ID |

**请求示例：**

{  
 id: "some job id"  
}

**返回示例：**

{  
id: "some job id",  
status: "Running|Success|Failed"  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | String | Varchar(24) | 扫描任务ID |
| status | String | Varchar(7) | 扫描的状态（进行中/成功/失败） |

如果请求的任务ID 不存在，则返回404，结果如下

{  
 "errorCode": 404, // 错误码  
 "errorMessage": "数据不存在", // 错误描述  
 "detail":null  
}

**代码示例(python)：**资产信息的扫描状态

#linux启动项信息扫描更新数据的代码示例(python)  
def refresh():  
 url = "http://%s:%s/external/api/assets/service/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res  
  
#资产信息的扫描状态接口调用  
def refresh\_job():  
 result = refresh()  
 job\_id = json.loads(result).get('id')  
 url = "http://%s:%s/external/api/assets/refreshjob/%s" % (host, port, job\_id)  
 data = {}  
 res = send\_request("GET", url, data)  
 print "result: ",res

## 主机信息

**功能描述**

该功能API用于查询资产中的主机信息，获取查询的结果并展示；您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

主机信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 主机信息查询结果
2. 主机信息扫描
3. 主机信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用主机信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 主机信息查询结果

该接口用于查询主机信息，按每台主机进行展示；

**调用接口：**

GET /external/api/assets/host/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| platform | String | 否 | 操作系统 |
| lastOnlineTime | DateRange | 否 | 最后一次在线时间，yyyy-MM-dd HH:mm:ss - yyyy-MM-dd HH:mm:ss |
| agentStatus | Integer数组 | 否 | 主机状态，0-在线 1-离线 2-停用 |
| serialNumber | String | 否 | 序列号 |

**请求示例：**

/external/api/assets/host/linux?page=0&size=50&groups=1,2&lastOnlineTime=2018-04-01 11:00:00 – 2018-04-01 16:00:00&sorts=+ip,-hostname&serialNumber=VMware-56 4d

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId":"70db8ef89e9ae79a",  
 "displayIp":"172.16.2.231",  
 "connectionIp":"172.16.2.231",  
 "externalIp":null,  
 "internalIp":null,  
 "bizGroupId":null,  
 "bizGroup":"被删除分组",  
 "remark":null,  
 "hostTagList":[  
 "test-test000",  
 "tmp"  
 ],  
 "hostname":"qingteng",  
 "proxyIp":null,  
 "platform":"Ubuntu 16.04.2 LTS",  
 "kernelVersion":"4.13.0-26-generic",  
 "cpu":{  
 "core":4,  
 "producer":"GenuineIntel",  
 "brand":"Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz",  
 "description":"GenuineIntel 4 Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz",  
 "loadAvgFifteen": 0.35  
 "cpuNum" : NumberInt(2),   
 "cpuArch" : "x86\_64",   
 "clockSpeed" : 2095.082,   
 "physicalCoreNum" : 8,   
 "logicalCoreNum" : 8  
 },  
 "memoryUsage":null,  
 "memorySize":null,  
 "onlineStatus":1,  
 "agentStatus":0,  
 "lastOnlineTime":null,  
 "installTime":null,  
 "agentVersion":"3.0.7-3.34.0-RC-Debug-2018-03-30\_18-50-07-436",  
 "bashVersion":null,  
 "bashPluginInstalled":null,  
 "offlineDays": 0,  
 "chargeName": "qingteng",  
 "hostLocation": "wuhan",  
 "systemLoad": 1,  
 "memories": [  
 {  
 "type": null,  
 "producer": "Not Specified",  
 "size": 980,  
 "speed": 0,  
 "description": "0.96GB(Not Specified null 0MHZ)",  
 "producerSize": 2048,  
 "bank": "Not Specified 2.0GB 0MHZ"  
 }  
 ],  
 "manufacturer": "",  
 "productName": "",  
 "serialNumber": "",  
 "networkCards": [  
 {  
 "name": "eth0",  
 "mac": "00:50:56:36:35:bb",  
 "ipv4": "192.168.78.131",  
 "ipv6": "fe80::250:56ff:fe36:35bb",  
 "gateway": "192.168.78.2",  
 "dnsServer": [  
 "yes"  
 ],  
 "netmask": "255.255.255.0",  
 "broadcast": "192.168.78.255",  
 "status": "1",  
 "speed": 1000  
 },  
 {  
 "name": "lo",  
 "mac": "00:00:00:00:00:00",  
 "ipv4": "127.0.0.1",  
 "ipv6": "::1",  
 "gateway": "",  
 "dnsServer": [  
 "192.168.78.2"  
 ],  
 "netmask": "255.255.255.0",  
 "broadcast": "192.168.78.255",  
 "status": "1",  
 "speed": 1000  
 }  
 ],  
 "diskSize": 51200,  
 "diskUsage": 0.6096,  
 "diskCount": 4  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| proxyIp | String | varchar(15) | 代理ip，仅linux可用 |
| platform | String | varchar(512) | 操作系统信息 |
| kernelVersion | String | varchar(512) | 内核版本 |
| cpu | Object | core：tinyint(4) producer：varchar(512) brand：varchar(512) description：varchar(1024) loadAvgFifteen:bigint(10) | cpu信息 core：处理器个数 producer：厂商 brand：品牌 description：描述 loadAvgFifteen: cpu十五分钟系统平均负载 cpuNum:处理器颗数 cpuArch:处理器架构 clockSpeed:处理器主频，单位：MHz physicalCoreNum:物理核心总数 logicalCoreNum:逻辑核心总数 |
| diskCount | Integer | tinyint(4) | 硬盘个数 |
| diskSize | double | double | 硬盘大小，单位MB |
| diskUsage | Integer | float | 硬盘使用率 |
| memoryUsage | double | double | 内存使用率 |
| memorySize | int | int(10) | 内存大小，单位MB |
| onlineStatus | int | tinyint(4) | 通信状态 1 – 在线 0 - 离线 |
| agentStatus | Integer数组 | tinyint(4) | Agent状态，0-在线 1-离线 2-停用 3-删除中 |
| lastOnlineTime | Date | date | 最后在线时间，时间戳秒 |
| installTime | Date | date | 安装时间，时间戳秒 |
| agentVersion | String | varchar(512) | agent版本 |
| bashVersion | String | varchar(512) | bash版本，仅linux可用 |
| bashPluginInstalled | Boolean | tinyint(1) | bash是否安装 true-安装 false-未安装 仅linux可用 |
| systemLoad | Integer | tinyint(4) | 系统负载0 – 未知 1 – 低2 – 中 3 – 高 |
| offlineDays | long | bigint (20) | 不在线天数=当前时间-lastOnlineTime |
| hostLocation | String | varchar(512) | 机房位置 |
| chargeName | String | varchar(128) | 负责人 |
| memories | Object数组 | type：varchar(512) producer：varchar(512) size：int(10) speed：tinyint(4) description：varchar(1024) producerSize:bigint(20) bank:varchar(1024) | 硬件配置信息 type：生产商 producer：生产厂商 size：单条内存大小,(系统实际大小)单位MB speed：内存主频率，单位MHz description：描述 producerSize:出厂内存条大小 bank：内存条信息 |
| manufacturer | String | varchar(512) | 生产商 |
| productName | String | varchar(512) | 设备型号 |
| serialNumber | String | varchar(512) | 序列号 |
| networkCards | Object数组 | name：varchar(512) mac：varchar(512) ipv4：varchar(15) ipv6：varchar(32) gateway：varchar(1024) dnsServer：varchar(512) netmask：varchar(15) broadcast：varchar(15) status：varchar(2) speed：long | 网卡信息 name：网卡名 mac：mac地址 ipv4：ipv4网卡地址 ipv6：ipv6网卡地址 gateway：网关 dnsServer：dnsserver netmask：子网掩码 broadcast：广播地址 status：网卡状态，空字符串：未知，"1"：已连接，"0"：未连接 speed：网卡速度，单位：Mb/s，0表示未知 |

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "127.0.0.1"  
port = 80  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@xx.com", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 print info  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 print to\_sign  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
def host\_linux():  
 url = "http://%s:%s/external/api/assets/host/linux?agentId=9d64f73542a1c78f&hostname=slave&ip=192.168.78.131&agentStatus=1" % (host, port)  
 data = {'agentId': '9d64f73542a1c78f', 'hostname': 'slave', 'ip': '192.168.78.131', 'agentStatus': 1}  
 res = send\_request("GET", url, data)  
 print "result: ",res  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 host\_linux()

### 主机信息扫描

该接口用于请求主机信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/host/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/host/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 主机信息查询状态扫描

用于查询主机信息的扫描状态；具体请求方式参照 3.1。

### 主机信息同步接口

该接口用于把外部的主机信息，同步到主机表中。

**调用接口：**

POST /external/api/assets/host/hostInfoSync

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| hostIp | String | 否 | 主机Ip |
| hostname | String | 否 | 主机名 |
| assetLevel | String | 否 | 资产等级（普通资产，重要资产，核心资产） |
| chargeName | String | 否 | 负责人 |
| chargeEmail | String | 否 | 负责人邮箱 |
| hostLocation | String | 否 | 机房位置 |
| assetNumber | String | 否 | 固定资产编号 |
| hostTagName | String | 否 | 标签名 |
| remark | String | 否 | 备注 |
| type | String | 是 | 导入的基准（１：hostname为基准， ２：hostIp为基准），只能是１或者２ |
| osType | String | 是 | 系统类型（１：liunx , 2:windows）, 只能是１或者２ |
| comid | String | 是 | 账号唯一标识 |

**返回示例：**

{  
 "code": 200,  
 "message": null,  
 "data": "success"  
}

**如果osType或type错误返回：**

{  
 "code": 100,  
 "message": "osType/type　error",  
 "data": ”fail“  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| code | Integer | tinyint(4) | 状态码，成功200， 失败100 |
| message | string | varchar(512) | 失败原因描述，code=200时，为null |
| data | string | varchar(24) | 字符“success”或"fail" |

**代码示例(python)：**主机信息同步

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
"""  
 该脚本实现的功能：把.csv文件读取数据，并拼接为insert\_data，把insert\_data,通过调用外部接口external/api/assets/host/hostInfoSync，把数据同步到mongo数据库中  
 脚本使用的方法：  
 　　　１、在当前目录放.csv文件，csv文件的列为  
 　　　２、安装相关依赖包，执行python endscripts.py即可  
"""  
import sys  
import csv  
import os  
import glob  
import requests  
import json  
import httplib  
import time  
import hashlib  
  
defaultencoding = 'utf-8'  
if sys.getdefaultencoding() != defaultencoding:  
 reload(sys)  
 sys.setdefaultencoding(defaultencoding)  
  
host = "127.0.0.1"  
port = 80  
comId = "xxx"  
osType = 1  
choiceType = 1  
  
# 数据集绝对路径以及文件名称  
csvx\_list = glob.glob('\*.csv')  
host\_file = sys.path[0] + '/'+ csvx\_list[0]  
url = 'http://%s:%s/external/api/assets/host/hostInfoSync'%(host, port)  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@xx.com", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)  
  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
def import\_csv():  
 if not os.path.exists(host\_file):  
 return  
  
 insert\_data = {}  
  
 #从csv中获取待插入mongodb的数据insert\_data  
 with open(host\_file,'rb') as csvfile:  
 csv\_reader = csv.reader(csvfile)  
 host\_header = next(csv\_reader)  
  
 for row in csv\_reader:  
 insert\_data = {"hostIp":row[0],"hostname":row[1],"assetLevel":row[2], "chargeName":row[3], "chargeEmail":row[4],  
 "hostLocation":row[5],"assetNumber":row[6], "hostTagName":row[7],"remark":row[8], "osType":osType, "type":choiceType, "comId":comId}  
  
 res = send\_request("POST", url, insert\_data)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print "start"  
 import\_csv()  
 print "done"

## 进程信息

**功能描述**

该功能API用于查询资产中的进程信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

进程信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 进程信息查询结果
2. 进程信息扫描
3. 进程信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用进程信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 进程信息查询结果

该接口用于查询进程信息，按每台主机每个进程进行展示；

**调用接口：**

GET /external/api/assets/process/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| startTime | Date | 否 | 进程启动时间 |
| versions | String数组 | 否 | 版本，仅windows可用 |
| root | Boolean | 否 | 是否root权限运行，仅linux可用 |
| packageName | String | 否 | 包名,仅linux可用 |
| packageVersions | String数组 | 否 | 包版本列表,仅linux可用 |
| installedByPm | Boolean | 否 | 是否包安装进程 ,仅linux可用 |
| pids | Integer数组 | 否 | 进程id |
| state | String | 否 | 进程状态,仅linux可用 |
| path | String | 否 | 进程路径（模糊查询） |
| uname | String | 否 | 用户名（Linux模糊查询） |
| gname | String | 否 | 用户组名（模糊查询）,仅linux可用 |
| name | String | 否 | 进程名（模糊查询) |
| startArgs | String | 否 | 进程启动参数（模糊查询） |
| tty | String | 否 | 进程启动的TTY（模糊查询），仅linux可用 |
| description | String | 否 | 进程描述（模糊查询），仅windows可用 |
| types | Integer数组 | 否 | 进程类型查询(其中：1-表示应用程序 2-表示后台程序 3-表示windows进程)），仅windows可用 |

**请求示例：**

/external/api/assets/process/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId":"70db8ef89e9ae79a",  
 "displayIp":"172.16.2.231",  
 "connectionIp":"172.16.2.231",  
 "externalIp":null,  
 "internalIp":"172.16.2.231",  
 "bizGroupId":39,  
 "bizGroup":"qingteng",  
 "remark":null,  
 "hostTagList":[  
 "test-test000",  
 "tmp"  
 ],  
 "hostname":"qingteng",  
 "name":"systemd-journal",  
 "pid":246,  
 "path":"/lib/systemd/systemd-journald",  
 "startArgs":"/lib/systemd/systemd-journald",  
 "state":"S",  
 "uname":"root",  
 "uid":0,  
 "gname":"root",  
 "tty":"?",  
 "startTime":"2018-03-30 13:55:45",  
 "portCount":0,  
 "md5":"6d8a3cd92c02ba9ce4a1e677dbf23462",  
 "installByPm":true,  
 "packageVersion":"229-4ubuntu17",  
 "packageName":"systemd",  
 "gid":0,  
 "root":true,  
 "dependencies":[  
 ],  
 "description":null,  
 "version":null,  
 "groups":null,  
 "type":null,  
 "sessionId":null,  
 "sessionName":null,  
 "size":null,  
 "ppid":1  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 显示IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| startTime | Date | date | 进程启动时间 |
| version | String | varchar(512) | 进程版本，仅windows可用 |
| root | Boolean | tinyint(1) | 是否root权限启动，仅linux可用 |
| prtCount | Integer | tinyint(4) | 进程端口数 |
| Md5 | String | varchar(32) | 可执行文件md5 |
| packageName | String | varchar(512) | 进程对应软件包名称，仅linux可用 |
| packageVersion | String | varchar(512) | 进程对应软件包版本，仅linux可用 |
| installByPm | Boolean | tinyint(1) | 是否包管理器安装，Windows为空 |
| pid | Integer | int(10) | 进程ID |
| ppid | Integer | int(10) | 父进程ID |
| path | String | varchar(512) | 进程路径 |
| startArgs | String | varchar(2048) | 进程启动参数 |
| state | String | varchar(2) | 进程状态，仅linux可用 |
| uname | String | varchar(128) | 用户名 |
| uid | Integer | bigint(20) | 用户id |
| gname | String | varchar(128) | 用户组名 |
| gid | Integer | bigint (20) | 用户组id，仅linux可用 |
| tty | String | varchar(512) | 进程启动的TTY，仅linux可用 |
| name | String | varchar(128) | 进程名 |
| sessionId | Integer | int(10) | 会话id, 仅windows可用 |
| sessionName | String | varchar(128) | 会话名，仅windows可用 |
| type | Integer | tinyint(4) | 进程类型，1-应用程序 2-后台程序 3-windows进程，仅windows可用 |
| description | String | varchar(512) | 进程描述，仅windows可用 |
| groups | String数组 | varchar(128) | 进程用户组，仅windows可用 |
| size | Integer | int(10) | 进程可执行文件大小，仅windows可用 |

### 进程信息扫描

该接口用于请求进程信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/process/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/process/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

* 返回部分

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 进程信息查询扫描状态

用于查询进程信息的扫描状态；具体请求方式参考[资产通用查询扫描状态](#资产通用查询扫描状态)。

## 系统账号信息

**功能描述**

该功能API用于查询资产中的系统账号信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

系统账号信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 系统账号信息查询结果
2. 系统账号信息扫描
3. 系统账号信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用系统账号信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 系统账号信息查询结果

该接口用于查询系统账号信息，按每台主机每个账号进行展示。

**调用接口：**

GET /external/api/assets/account/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| status | Integer数组 | 否 | 帐号状态 linux 账号状态，1:启用，0:禁用 ；windows 账号状态，0:启用，2:禁用 |
| name | String | 否 | 账号名 |
| home | String | 否 | home目录（模糊查询） |
| lastLoginTime | DateRange | 否 | 最后一次登录时间 |
| gid | Integer | 否 | 用户组id |
| uid | Integer | 否 | 用户id |

**请求示例：**

/external/api/assets/account/linux?agentId=70db8ef89e9ae79a

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId":"70db8ef89e9ae79a",  
 "displayIp":"172.16.2.231",  
 "connectionIp":"172.16.2.231",  
 "externalIp":null,  
 "internalIp":"172.16.2.231",  
 "bizGroupId":39,  
 "bizGroup":"qingteng",  
 "remark":null,  
 "hostTagList":[  
 "test-test000",  
 "tmp"  
 ],  
 "hostname":"qingteng",  
 "uid":116,  
 "gid":65534,  
 "groups":[  
 "nogroup"  
 ],  
 "name":"kernoops",  
 "home":"/",  
 "shell":"/bin/false",  
　　　　"root": false,  
 "status":0,  
 "lastLoginTime":"1970-01-01 08:00:00",  
 "pwdMaxDays":99999,  
 "pwdMinDays":-1,  
 "pwdWarnDays":7,  
 "loginStatus":0,  
 "sshAcl":"",  
　　　　"sudoAccesses": [],  
 "comment":"Kernel Oops Tracking Daemon,,,",  
 "lastLoginTty":"",  
 "lastLoginIp":"",  
 "expireTime":"1969-12-31 08:00:00",  
 "expired":false,  
 "fullName":null,  
 "description":null,  
 "lastChangPwdTime": "2017-09-04 08:00:00",  
 "accountLoginType": 3,  
 "interactiveLoginType": 2,  
 "passwordInactiveDays": null,  
 "sudo": true,  
 "authorizedKeys": [  
 {  
 "encryptType": "ssh-dss",  
 "comment": "root@centos-master",  
 "value": "AAAAB3Nz...+fgkXA==",  
 "md5": "36e4313ab5b1aae15ff0f1948a4d73ea"  
 }  
 ]  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 显示IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| uid | Integer | bigint(20) | 账号uid |
| gid | Integer | bigint(20) | 用户组id |
| groups | String | varchar(128) | 账户组 |
| name | String | varchar(128) | 账号名称 |
| status | Integer数组 | tinyint(4) | 账号状态 linux 账号状态，1:启用，0:禁用 ；windows 账号状态，0:启用，1：锁定，2:禁用 |
| home | String | varchar(512) | home目录 |
| shell | String | varchar(512) | 用户shell，仅linux可用 |
| loginStatus | Integer | tinyint(4) | 登录状态，0不可登入 1不可交互登入 2可交互登入，3 key&pwd登陆 ，仅linux可用 |
| lastLoginTime | Date | date | 最后登录时间 |
| pwdMaxDays | Integer | int(10) | 密码到期天数， null为不限 |
| pwdMinDays | Integer | int(10) | 密码多少天后可修改，nul为不限 |
| pwdWarnDays | Integer | int(10) | 密码到期告警天数，null为不限 |
| sshAcl | String | varchar(3) | ~./ssh访问权限, 如”777”, “666”，仅linux可用 |
| comment | String | varchar(1024) | 帐号备注，仅linux可用 |
| lastLoginTty | String | varchar(1024) | 最后登录终端，仅linux可用 |
| lastLoginIp | String | varchar(15) | 最后登录ip，仅linux可用 |
| expireTime | Date | date | 帐号到期时间，仅linux可用 |
| expired | Boolean | int(10) | 是否过期 |
| fullName | String | varchar(128) | 用户全名，仅Windows可用 |
| sudoAccesses | List | shell：varchar(128) user：varchar(128) | sudo权限 shell:权限 user:用户权限 |
| root | Boolean | tinyint(1) | 是否是root，仅linux可用 |
| description | String | varchar(512) | 用户描述，仅Windows可用 |
| type | Integer | tinyint(4) | 账号类型 仅windows可用 0：访客用户 1：标准用户 2：管理员用户 |
| lastChangPwdTime | Date | date | 密码最后修改时间 |
| accountLoginType | Integer | tinyint(4) | 账户登录方式，仅linux可用 0 不可登陆 1 key登陆 2 pwd登陆 3 key&pwd登陆 |
| interactiveLoginType | Integer | tinyint(4) | 交互登录方式，仅linux可用 0 不可登录 1 不可交互登录 2 可交互登录 |
| passwordInactiveDays | Integer | int(10) | 密码过期后变成无效的天数，-1为无限 |
| sudo | Boolean | tinyint(1) | 是否sudo权限，仅linux可用 |
| authorizedKeys | Object数组 | encryptType：varchar(512) comment：varchar(512) value：varchar(512) MD5：varchar(32) | 账号公钥信息，仅linux可用 encryptType：加密类型 comment：备注信息 value：公钥的值 MD5：MD5 |

### 系统账号信息扫描

该接口用于请求系统账号信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/account/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/account/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 系统账号信息查询扫描状态

用于查询系统账号信息的扫描状态；具体请求方式参照3.1。

## 用户组信息

用于查询系统用户组，将按每台主机每个用户组展示；每个用户组均提供其包含的用户，但仅包含用户名；每条数据均包含主机信息，您可以灵活的使用该接口组合出任何需要的显示结果。

**功能描述**

该功能API用于查询资产中的用户组信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

用户组信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 用户组信息查询结果
2. 用户组信息扫描
3. 用户组信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用用户组信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 用户组信息查询结果

该接口用于查询用户组信息，按每台主机每个用户组进行展示。

**调用接口：**

GET /external/api/assets/accountgroup/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | 用户组名（模糊查询） |
| gid | long | 否 | 用户组id（仅支持linux） |

**请求示例：**

/external/api/assets/accountgroup/linux?agentId=70db8ef89e9ae79a

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId":"70db8ef89e9ae79a",  
 "displayIp":"172.16.2.231",  
 "connectionIp":"172.16.2.231",  
 "externalIp":null,  
 "internalIp":"172.16.2.231",  
 "bizGroupId":39,  
 "bizGroup":"qingteng",  
 "remark":null,  
 "hostTagList":[  
 "test-test000",  
 "tmp"  
 ],  
 "hostname":"qingteng",  
 "gid":4,  
 "name":"adm",  
 "description":null,  
 "members":[  
 {  
 "name":"qt",  
 "type":null  
 },  
 {  
 "name":"syslog",  
 "type":null  
 }  
 ]  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid,16位 |
| displayIp | String | varchar(15) | 显示IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(128) | 用户组名 |
| gid | Integer | bigint(20) | 用户组id（仅支持linux） |
| members | Object | name ：varchar(128) type：tinyint(4) | 组成员，仅windows name: 用户名 type: 用户类型 |
| description | String | varchar(1024) | 用户组描述，仅windows |

### 用户组信息扫描

该接口用于请求系统账号信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/accountgroup/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/accountgroup/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 用户组信息查询扫描状态

用于查询系统用户组信息的扫描状态；具体请求方式参照3.1。

## 端口信息

用于查询系统端口，将按每台主机每个端口展示；每条数据均包含主机信息，您可以灵活的使用该接口组合出任何需要的显示结果。

**功能描述**

该功能API用于查询资产中的端口信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

端口信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 端口信息查询结果
2. 端口信息扫描
3. 端口信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用端口信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 端口信息查询结果

该接口用于查询端口信息，按每台主机每个端口进行展示。

**调用接口：**

GET /external/api/assets/port/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| proto | String数组 | 否 | 协议 |
| port | Integer | 否 | 端口 |
| bindIp | String | 否 | 绑定ip |
| processName | String | 否 | 进程名（模糊查询） |

**请求参数：**

/external/api/assets/port/linux?agentId=70db8ef89e9ae79a

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId":"70db8ef89e9ae79a",  
 "displayIp":"172.16.2.231",  
 "connectionIp":"172.16.2.231",  
 "externalIp":null,  
 "internalIp":"172.16.2.231",  
 "bizGroupId":39,  
 "bizGroup":"qingteng",  
 "remark":null,  
 "hostTagList":[  
 "test-test000",  
 "tmp"  
 ],  
 "hostname":"qingteng",  
 "proto":"tcp",  
 "port":41935,  
 "pid":4906,  
 "processName":"java(com.intellij.idea.Main)",  
 "bindIp":"127.0.0.1",  
 "status":null  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 显示IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| proto | String | varchar(512) | 协议 |
| port | Integer | int(10) | 端口号 |
| pid | Integer | int(10) | 进程id |
| processName | String | varchar(128) | 进程名 |
| bindIp | String | varchar(15) | 绑定ip |
| status | Integer | tinyint(4) | 端口状态：-1 - 端口状态未知；0 – 仅内网可访问；1 - 外网可访问,null不存在该字段 |

### 端口信息扫描

该接口用于请求端口信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/port/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/port/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 端口信息查询扫描状态

用于查询系统端口信息的扫描状态；具体请求方式参照3.1。

## Web应用信息

用于查询主机上web应用信息；每条数据均包含主机信息，您可以灵活的使用该接口组合出任何需要的显示结果。

**功能描述**

该功能API用于查询资产中的web应用信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

web应用信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. web应用信息查询结果
2. web应用信息扫描
3. web应用信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用web应用信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### web应用信息查询结果

该接口用于查询web应用信息，按每台主机每个用户组进行展示。

**调用接口：**

GET /external/api/assets/webapp/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| version | String数组 | 否 | 应用版本 |
| appName | String | 否 | 应用名 |
| rootPath | String | 否 | 根路径 |
| webRoot | String | 否 | 站点根路径 |
| serverName | String数组 | 否 | 服务类型，如nginx、apache、tomcat |
| domainName | String | 否 | 域名 |

**请求示例：**

/external/api/assets/webapp/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId":"fe107d9c77f4c77a",  
 "displayIp":"192.168.78.131",  
 "connectionIp":"172.16.2.244",  
 "externalIp":null,  
 "internalIp":"192.168.78.131",  
 "bizGroupId":41,  
 "bizGroup":"qingteng",  
 "remark":"12345",  
 "hostTagList":[  
 "联系",  
 "标签a"  
 ],  
 "hostname":"xdu-vm-cent51",  
 "version":"",  
 "webRoot":"/usr/share/nginx/html/",  
 "serverName":"nginx",  
 "domainName":"192.168.78.131",  
 "pluginCount":0,  
 "appName":"phpMyAdmin",  
 "description":"phpMyAdmin 是一个以PHP为基础，以Web-Base方式架构在网站主机上的MySQL的数据库管理工具，让管理者可用Web接口管理MySQL数据库。",  
 "rootPath":"/usr/share/nginx/html/phpMyAdmin",  
 "plugins":null  
 },  
 {  
 "agentId":"dbcf7ac1b5537764",  
 "displayIp":"192.168.126.203",  
 "connectionIp":"172.16.2.138",  
 "externalIp":null,  
 "internalIp":"192.168.126.203",  
 "bizGroupId":41,  
 "bizGroup":"qingteng",  
 "remark":"安装了mysql或者DNSmasq",  
 "hostTagList":[  
 "标签a",  
 "标签b"  
 ],  
 "hostname":"oldtcm",  
 "version":"4.5.3",  
 "webRoot":"/var/www/html/",  
 "serverName":"httpd",  
 "domainName":"www.qingtengdemoapache.com",  
 "pluginCount":4,  
 "appName":"WordPress",  
 "description":"WordPress是一种使用PHP语言开发的博客平台，用户可以在支持PHP和MySQL 数据库的服务器上架设自己的网志。",  
 "rootPath":"/var/www/html/wordpress",  
 "plugins":[  
 {  
 "pluginName":"JQuery Html5 File Upload",  
 "pluginUri":"http://wordpress.org/extend/plugins/jquery-html5-file-upload/",  
 "description":"This plugin adds a file upload functionality to the front-end screen. It allows multiple file upload asynchronously along with upload status bar.",  
 "author":"sinashshajahan",  
 "authorUri":" ",  
 "version":"3.0"  
 },  
 {  
 "pluginName":"Roomcloud",  
 "pluginUri":"http://www.roomcloud.net",  
 "description":"A Plugin to add roomcloud booking form to hotel website using [roomcloud] shortcode",  
 "author":"Raffaello Bindi",  
 "authorUri":"http://www.roomcloud.net",  
 "version":"1.3"  
 },  
 {  
 "pluginName":"Hello Dolly",  
 "pluginUri":"http://wordpress.org/plugins/hello-dolly/",  
 "description":"This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from <cite>Hello, Dolly</cite> in the upper right of your admin screen on every page.",  
 "author":"Matt Mullenweg",  
 "authorUri":"http://ma.tt/",  
 "version":"1.6"  
 },  
 {  
 "pluginName":"Akismet",  
 "pluginUri":"https://akismet.com/",  
 "description":"Used by millions, Akismet is quite possibly the best way in the world to <strong>protect your blog from spam</strong>. It keeps your site protected even while you sleep. To get started: 1) Click the \"Activate\" link to the left of this description, 2) <a href=\"https://akismet.com/get/\">Sign up for an Akismet plan</a> to get an API key, and 3) Go to your Akismet configuration page, and save your API key.",  
 "author":"Automattic",  
 "authorUri":"https://automattic.com/wordpress-plugins/",  
 "version":"3.1.11"  
 }  
 ]  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 显示IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| version | String | varchar(512) | 应用版本 |
| webRoot | String | varchar(1024) | 站点根路径 |
| serverName | String | varchar(128) | 站点类型 |
| domainName | String | varchar(512) | 域名 |
| pluginCount | Integer | int(10) | 插件数 |
| appName | String | varchar(512) | 应用名 |
| description | String | varchar(512) | 描述 |
| rootPath | String | varchar(512) | 根路径 |
| plugins | Object | pluginName：varchar(512) pluginUri：varchar(1024) description：varchar(1024) author：varchar(512) authorUri：varchar(1024) version：varchar(512) | 插件信息列表 pluginName：插件名 pluginUri：插件官网链接 description：插件描述 author：作者 authorUri：作者地址 version：版本 |

### web应用信息扫描

该接口用于请求web应用信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/webapp/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/webapp/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 |
| --- | --- | --- |
| id | string | varchar(24) |

### web应用信息查询扫描状态

用于查询web应用信息的扫描状态；具体请求方式参照3.1。

## Web站点信息

用于查询web站点信息；每条数据均包含主机信息，您可以灵活的使用该接口组合出任何需要的显示结果。

**功能描述**

该功能API用于查询资产中的web站点信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

web站点信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. web站点信息查询结果
2. web站点信息扫描
3. web站点信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用web站点信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### web站点信息查询结果

该接口用于查询web站点信息，按每台主机每个用户组进行展示。

**调用接口：**

GET /external/api/assets/website/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| port | Integer | 否 | 站点端口 |
| proto | String | 否 | 站点协议，精确匹配 |
| type | String数组 | 否 | 服务类型，精确匹配，如iis,nginx等 |
| rootPath | String | 否 | 站点路径，模糊匹配 |

**请求示例：**

/external/api/assets/website/linux?agentId=70db8ef89e9ae79a

**返回示例：**

{  
"total":1,  
 "rows": [  
 {  
 "agentId": "c0bb1116dfa69772",  
 "displayIp": "192.168.202.147",  
 "connectionIp": "192.168.199.16",  
 "externalIp": null,  
 "internalIp": "192.168.202.147",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": "测试接口哦哦11哦",  
 "hostTagList": [],  
 "hostname": "localhost.localdomain",  
 "type": "httpd",  
 "port": 80,  
 "portStatus": null,  
 "securityEnabled": false,  
 "allow": "all",  
 "deny": "",  
 "proto": "http",  
 "domains": [  
 {  
 "name": "\*",  
 "title": "",  
 "ip": ""  
 }  
 ],  
 "virtualDir": [  
 {  
 "path": "",  
 "physicalPath": "/var/www/html",  
 "owner": "root",  
 "group": "root",  
 "permission": "rwxr-xr-x",  
 "ownerType": 0,  
 "acls": null,  
 "appPath": null,  
 "appPool": null,  
 "root": true  
 },  
 {  
 "path": "/icons/",  
 "physicalPath": "/var/www/icons/",  
 "owner": "root",  
 "group": "root",  
 "permission": "rwxr-xr-x",  
 "ownerType": 0,  
 "acls": null,  
 "appPath": null,  
 "appPool": null,  
 "root": false  
 },  
 {  
 "path": "/cgi-bin/",  
 "physicalPath": "/var/www/cgi-bin/",  
 "owner": "root",  
 "group": "root",  
 "permission": "rwxr-xr-x",  
 "ownerType": 0,  
 "acls": null,  
 "appPath": null,  
 "appPool": null,  
 "root": false  
 },  
 {  
 "path": "/error/",  
 "physicalPath": "/var/www/error/",  
 "owner": "root",  
 "group": "root",  
 "permission": "rwxr-xr-x",  
 "ownerType": 0,  
 "acls": null,  
 "appPath": null,  
 "appPool": null,  
 "root": false  
 }  
 ],  
 "root": {  
 "path": "",  
 "physicalPath": "/var/www/html",  
 "owner": "root",  
 "group": "root",  
 "permission": "rwxr-xr-x",  
 "ownerType": 0,  
 "acls": null,  
 "appPath": null,  
 "appPool": null,  
 "root": true  
 },  
 "virtualDirCount": 4,  
 "pid": 3013,  
 "user": "apache",  
 "name": "httpd",  
 "cmd": "/usr/sbin/httpd ",  
 "configName": null,  
 "state": 0,  
 "bindingCount": 0,  
 "deployPath": "/root/apache-tomcat-8.0.38/webapps"  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 显示IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| pid | Integer | int(10) | 进程id |
| allow | String | varchar(1024) | 允许地址，仅Linux及 |
| deny | String | varchar(1024) | 拒绝地址，仅Linux及Windows-nginx |
| cmd | String | varchar(512) | 进程启动命令行参数 |
| domains | List | name: varchar(128) title：varchar(512) ip：varchar(15) | 域名信息列表 name：域名名称 title：标题 ip：绑定ip |
| user | String | varchar(512) | 启动服务用户 |
| type | String | varchar(10) | 站点类型，如nginx，http |
| port | Integer | int(10) | 端口 |
| proto | String | varchar(10) | 协议 |
| portStatus | Integer | tinyint(4) | 端口状态:-1 - 端口状态未知; 0 – 仅内网可访问;1 - 外网可访问 |
| securityEnabled | Boolean | tinyint(1) | 是否开启安全模块 false- 未开启 true-开启 （仅Linux及Windows-nginx） |
| virtualDir | List | path：varchar(1024) physicalPath：varchar(1024) root：tinyint(1) owner：varchar(512) group：bigint(20) permission: varchar(7) acls aceType：tinyint(4) user: varchar(512) userType：tinyint(4) accessMask:bigint(20) appPath: varchar(1024) appPool name: varchar(512) identityType: tinyint(4) user: varchar(512) | 虚拟目录信息 path：虚拟地址 physicalPath：物理地址 root：是否主目录 owner：目录所有者 group：目录所属用户组 仅linux permission：目录权限 仅linux acls：仅windows aceType ace类型 user 用户名 userType 用户类型 accessMask 访问控制掩码数组 appPath: 应用程序路径 （仅Windows-IIS） appPool: 程序池信息（仅Windows-IIS） name 程序池名称 identityType 运行账户标识 user 运行账户名 |
| root | Object | 参考虚拟目录 | 主目录信息 path：虚拟地址 physicalPath：物理地址 root：是否主目录 owner：目录所有者 group：目录所属用户组 仅linux permission：目录权限 仅linux acls：仅windows aceType ace类型 user 用户名 userType 用户类型 accessMask 访问控制掩码数组 appPath: 应用程序路径 （仅Windows-IIS） appPool: 程序池信息（仅Windows-IIS） name 程序池名称 identityType 运行账户标识 user 运行账户名 |
| virtualDirCount | Integer | int(10) | 虚拟路径数 |
| bindingCount | Integer | int(10) | 绑定地址数 |
| deployPath | String | varchar(4096) | War包部署总目录（仅Linux-Tomcat/Weblogic/JBoss/Wildfly/Jetty) |
| configName | String | varchar(4096) | 仅IIS可用，站点别名 |
| state | Integer | tinyint(4) | 仅IIS可用，站点状态 |
| path | String | varchar(4096) | 仅WINDOWS的weblogic\webshpere\jetty\wildfly使用，站点物理地址 |

### web站点信息扫描

该接口用于请求web站点信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/website/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/website/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### web站点信息查询扫描状态

用于查询web站点信息的扫描状态；具体请求方式参照3.1。

## 软件应用信息

**功能描述**

该功能API用于查询资产中的软件应用信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

软件应用信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 软件应用信息查询结果；
2. 软件应用信息扫描；
3. 软件应用信息查询扫描状态。

使用方法如下：

* 直接查询当前信息则直接调用软件应用信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示：

### 软件应用信息查询结果

该接口用于查询软件应用信息，按每台主机每个软件应用进行展示。

**调用接口：**

GET /external/api/assets/app/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | 软件应用名称（模糊查询） |
| version | String数组 | 否 | 软件应用版本 |
| binPath | String | 否 | linux为二进制路径，windows 为安装路径（模糊查询） |
| configPath | String | 否 | 配置文件路径（模糊查询） |

**请求示例：**

/external/api/assets/app/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "qingteng",  
 "remark": "12345",  
 "hostTagList": [  
 "联系",  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "name": "gam\_server",  
 "version": "0.1.7",  
 "uname": "root",  
 "binPath": "",  
 "configPath": "",  
 "processes": [  
 {  
 "name": "gam\_server",  
 "pid": 5571,  
 "uname": "root"  
 }  
 ]  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(512) | 软件应用名 |
| version | String | varchar(512) | 版本号 |
| uname | String | varchar(128) | 启动用户 |
| binPath | String | varchar(1024) | windows为安装路径，Linux为二进制路径 |
| configPath | String | varchar(1024) | 配置文件路径 |
| processes | List | pid：int(10) name：varchar(512) uname:varchar(512) | 关联进程列表 pid:进程id name:进程名 uname:进程启动用户 |

### 软件应用信息扫描

该接口用于请求软件应用信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/app/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/app/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 软件应用信息查询扫描状态

用于查询软件应用信息的扫描状态；具体请求方式参照3.1。

## Web应用框架信息

**功能描述**

该功能API用于查询资产中的Web应用框架信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

Web应用框架信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. .Web应用框架信息查询结果
2. Web应用框架信息扫描
3. Web应用框架信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用Web应用框架信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### Web应用框架信息查询结果

该接口用于查询Web应用框架信息，按每台主机每个Web应用框架进行展示。

**调用接口：**

GET /external/api/assets/webframe/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | web应用框架名称（模糊查询） |
| version | String | 否 | web应用框架版本 |
| type | String数组 | 否 | 框架语言 |
| serverName | String数组 | 否 | 服务类型 |

**请求示例：**

/external/api/assets/webframe/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "name": "Apache Commons CLI\r (wisteria.jar)",  
 "version": "1.4",  
 "type": "java",  
 "serverName": "Spring Boot",  
 "domainName": "",  
 "webAppDir": "/data/app/titan-wisteria/upload/wisteria-3.0.4-oldtcm\_180320145655/wisteria.jar",  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(512) | web应用框架名称 |
| version | String | varchar(512) | 框架版本号 |
| type | String | varchar(128) | 框架语言 |
| serverName | String | varchar(128) | 服务类型 |
| domainName | String | varchar(128) | 站点域名 |
| webAppDir | String | varchar(1024) | 框架绝对路径 |
| jarCount | String | varchar(1024) | 关联jar包数 |
| jarList | List | version:varchar(128), absDir:varchar(128),jarName:varchar(128) | 关联jar包详情 |
| webRoot | String | varchar(1024) | 根路径（php、django框架字段） |
| workDir | String | varchar(1024) | 应用路径php、django框架字段） |

### Web应用框架信息扫描

该接口用于请求Web应用框架信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/webframe/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/webframe/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### Web应用框架信息查询扫描状态

用于查询web应用框架信息的扫描状态；具体请求方式参照3.1。

## 数据库信息

**功能描述**

该功能API用于查询资产中的数据库信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

数据库信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 数据库信息查询结果
2. 数据库信息扫描
3. 数据库信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用数据库信息查询结果接口（即1-7号接口）；
* 获取更新数据则先调用8号接口进行更新再调用9号接口查看是否更新完成，最后再调用1-7号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 数据库信息查询结果

该接口用于通用数据库信息，按每台主机每个数据库进行展示。

**调用接口：**

GET /external/api/assets/dbinfo/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | 数据库类型 |
| versions | String数组 | 否 | 数据库版本 |
| port | Integer | 否 | 监听端口 |
| confPath | String | 否 | 配置文件路径（模糊查询） |
| logPath | String | 否 | 日志文件路径（模糊查询） |
| dataDir | String | 否 | 数据路径（模糊查询） |

**请求示例：**

/external/api/assets/dbinfo/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "a4a6b8074259377e",  
 "displayIp": "192.168.199.200",  
 "connectionIp": "192.168.199.200",  
 "externalIp": null,  
 "internalIp": "192.168.199.200",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "ubuntu",  
 "name": "mysql",  
 "version": "5.7.19-0ubuntu0.16.04.1",  
 "port": 3306,  
 "protoType": "tcp",  
 "user": "mysql",  
 "bindIp": "127.0.0.1",  
 "confPath": "/etc/mysql/my.cnf",  
 "logPath": "/var/lib/mysql/ubuntu.log",  
 "dataDir": "/var/lib/mysql/",  
 "pluginDir": "/usr/lib/mysql/plugin/",  
 "rest": "false",  
 "auth": "disabled",  
 "web": "false",  
 "webPort": 60010,  
 "webAddress": "0.0.0.0",  
 "regionServer": ["localhost"],  
 "dbName": "MSSQLSERVER\_1",  
 "loginModel": 1,  
 "auditLevel": 2,  
 "sysLogPath": "C:\Program Files\Microsoft SQL Server\MSSQL\data\mastlog.ldf",  
 "mainDbPath": "C:\Program Files\Microsoft SQL Server\MSSQL\data\master.mdf"  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid,16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(512) | 数据库类型 |
| version | String | varchar(512) | 数据库版本 |
| port | Integer | int(10) | 监听端口 |
| protoType | String | varchar(128) | 协议 |
| user | String | varchar(128) | 运行用户 |
| bindIp | String | varchar(1024) | 绑定IP |
| confPath | String | varchar(1024) | 配置文件路径 |
| logPath | String | varchar(1024) | 日志文件路径 |
| dataDir | String | varchar(1024) | 数据路径 |
| pluginDir | String | varchar(1024) | 插件目录,仅Linux MySQL |
| rest | String | varchar(5) | 是否开放rest:'true';'false',仅Linux MongoDB |
| auth | String | varchar(8) | 是否开启安全认证:'enabled'; 'disabled',仅Linux MongoDB |
| web | String | varchar(5) | 是否开启web接口:'true'; 'false',仅Linux MongoDB |
| webPort | Integer | int(10) | web界面端口,仅Linux HBase |
| webAddress | String | varchar(1024) | web界面地址,仅Oracle以及Linux HBase |
| regionServer | List | varchar(128) | region server列表,仅Linux HBase |
| dbName | String | varchar(128) | 数据库示例名,仅windows |
| loginModel | Integer | tinyint(4) | 身份验证,仅windows SQL Server |
| auditLevel | Integer | tinyint(4) | 审核级别,仅windows SQL Server |
| sysLogPath | String | varchar(1024) | 系统日志路径,仅windows SQL Server |
| mainDbPath | String | varchar(1024) | 主数据库路径,仅windows SQL Server |

### 数据库信息扫描

该接口用于请求数据库信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/dbinfo/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/dbinfo/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 数据库信息查询扫描状态

用于查询数据库信息的扫描状态；具体请求方式参照3.1。

## 启动项信息

**功能描述**

该功能API用于查询资产中的启动项信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

启动项信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 启动项信息查询结果
2. 启动项信息扫描
3. 启动项信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用启动项信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 启动项信息查询结果

该接口用于查询启动项信息，按每台主机每个启动项进行展示。

**调用接口：**

GET /external/api/assets/service/{linux/win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | 启动项名（模糊查询）（仅linux启动项名） |
| initLevel | Interger数组 | 否 | 默认启动模式(仅linux ） |
| defaultOpen | 布尔数组 | 否 | 默认模式启用状态（仅linux） |
| isXinetd | 布尔数组 | 否 | 启动方式 （仅linux） |
| showName | String | 否 | 启动项名（仅windows） |
| user | String | 否 | 服务启动用户windows |
| enable | 布尔 | 否 | 服务的状态windows |
| startType | Integer数组 | 否 | 启动类型windows |
| publisher | String | 否 | 发布者windows |

**请求示例：**

/external/api/assets/service/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "initLevel": 5,  
 "name": "NetworkManager",  
 "defaultOpen": true,  
 "rc0": 0,  
 "rc1": 0,  
 "rc2": 1,  
 "rc3": 1,  
 "rc4": 1,  
 "rc5": 1,  
 "rc6": 0,  
 "rc7": 0,  
 "xinetd": false,  
 "type": null,  
 "user": null,  
 "enable": null,  
 "startType": null,  
 "publisher": null  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(512) | 启动项名 |
| defaultOpen | 布尔 | tinyint(2) | 默认模式启用状态 |
| rc0 | Integer | bigint(16) | 停机(rc0) |
| rc1 | Integer | bigint(16) | 单用户模式(rc1) |
| rc2 | Integer | bigint(16) | 多用户无NFS模式(rc2) |
| rc3 | Integer | bigint(16) | 完全多用户模式(rc3) |
| rc4 | Integer | bigint(16) | 预留模式(rc4) |
| rc5 | Integer | bigint(16) | 桌面模式(rc5) |
| rc6 | Integer | bigint(16) | 重新启动(rc6) |
| rc7 | Integer | bigint(16) | 单用户自启动(rcs) |
| initLevel | Interger | bigint(16) | 默认启动模式(仅linux） |
| xinetd | 布尔 | tinyint(2) | 启动方式 |
| user | String | varchar(128) | 服务启动用户windows |
| enable | 布尔 | tinyint(2) | 服务的状态windows |
| startType | Integer | bigint(16) | 启动类型windows |
| publisher | String | varchar(64) | 发布者windows |
| showName | String | varchar(128) | 启动项名（仅windows） |

**windows启动项代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "127.0.0.1"  
port = 80  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@xx.com", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 print info  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 print to\_sign  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
def service\_win():  
 url = "http://%s:%s/external/api/assets/service/win?showName=Google%%20Chrome&enable=true&startType=1,2&publisher=Google&user=""" % (host, port)  
 data = {'showName': 'Google Chrome', 'enable': 'true', 'startType': '1,2', 'publisher': 'Google', 'user': ''}  
 res = send\_request("GET", url, data)  
 print "result: ",res  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 service\_win()

**代码示例(python)：**linux启动项

def service\_linux():  
 url = "http://%s:%s/external/api/assets/service/linux?initLevel=2&name=bootmisc.sh&defaultOpen=true,false&isXinetd=false,true" % (host, port)  
 data = {'initLevel': 2, 'name': 'bootmisc.sh', 'defaultOpen': 'true,false', 'isXinetd': 'false,true'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)。

### 启动项信息扫描

该接口用于请求启动项信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/service/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/service/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**代码示例(python)：**linux启动项信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/service/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

**代码示例(python)：**win启动项信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/service/win/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

### 启动项信息查询扫描状态

用于查询启动项信息的扫描状态；具体请求方式参照3.1。

## 计划任务信息

**功能描述**

该功能API用于查询资产中的计划任务信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

计划任务信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 计划任务信息查询结果
2. 计划任务信息扫描
3. 计划任务信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用计划任务信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 计划任务信息查询结果

该接口用于查询计划任务信息，按每台主机每个计划任务进行展示。

**调用接口：**

GET /external/api/assets/task/{linux}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| user | String数组 | 否 | 执行用户 |
| execPath | String | 否 | 执行命令或脚本 |
| conf | String | 否 | 配置文件 |
| taskTime | DateRange | 否 | 执行时间 |
| taskType | String | 否 | 任务类型（仅CRONTAB/AT/BATCH三类） |

**请求示例：**

/external/api/assets/task/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "user": "root",  
 "execTime": "47 6\t\* \* 7\t",  
 "execPath": "test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )",  
 "conf": "/etc/crontab",  
 "taskTime": "2016-04-06 05:59:09",  
 "taskId": 0,  
 "crondOpen": true  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| user | String | varchar(512) | 执行用户 |
| execTime | String | varchar(64) | 执行周期 |
| execPath | String | varchar(512) | 执行命令或脚本 |
| conf | String | varchar(512) | 配置文件 |
| taskTime | DateRange | bigint(10) | 执行时间 |
| taskId | Integer | bigint(20) | 任务Id |
| taskType | String | varchar(15) | 任务类型 |
| crondOpen | 布尔 | tinyint(2) | 启用状态 |

**代码示例(python)：**linux计划任务

def task\_linux():  
 url = "http://%s:%s/external/api/assets/task/linux?user=root&execPath=/usr/sbin/raid-check&conf=/etc/cron.d/raid-check&taskTime=2017-01-25%%2000:00:00%%20-%%202017-01-27%%2000:00:00&taskType=CRONTAB" % (host, port)  
 data = {'user': 'root', 'execPath': '/usr/sbin/raid-check', 'conf': '/etc/cron.d/raid-check', 'taskTime': '2017-01-25 00:00:00 - 2017-01-27 00:00:00', 'taskType': 'CRONTAB'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)，对于完整的模拟请求参考[启动项信息查询结果](#启动项信息查询结果)的windows代码示例(python)

### 计划任务信息扫描

该接口用于请求计划任务信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/task/{linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/task/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent",// 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**代码示例(python)：**linux计划任务信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/task/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

### 计划任务信息查询扫描状态

用于查询计划任务信息的扫描状态；具体请求方式参照3.1。

## 环境变量信息

**功能描述**

该功能API用于查询资产中的环境变量信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

环境变量信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 环境变量信息查询结果
2. 环境变量信息扫描
3. 环境变量信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用环境变量信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 环境变量信息查询结果

该接口用于查询环境变量信息，按每台主机每个环境变量进行展示。

**调用接口：**

GET /external/api/assets/env/{linux}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| key | String | 否 | 环境变量名 |
| value | String | 否 | 环境变量值 |
| user | String | 否 | 用户 |
| sysEnv | 布尔数组 | 否 | 环境变量类型 |

**请求示例：**

/external/api/assets/env/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "key": "BASH\_CMDS",  
 "value": "()",  
 "user": "weblogic",  
 "sysEnv": false  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| key | String | varchar(128) | 环境变量名 |
| value | String | varchar(128) | 环境变量值 |
| user | String | varchar(128) | 用户 |
| sysEnv | 布尔数组 | tinyint(2) | 环境变量类型 |

**代码示例(python)：**linux环境变量

def env\_linux():  
 url = "http://%s:%s/external/api/assets/env/linux?key=HOME&value=/home/yangwu&user=yangwu&sysEnv=false,true" % (host, port)  
 data = {'key': 'HOME', 'value': '/home/qingteng', 'user': 'qingteng', 'sysEnv': 'false,true'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)，对于完整的模拟请求参考[启动项信息查询结果](#启动项信息查询结果)的windows代码示例(python)

### 环境变量信息扫描

该接口用于请求环境变量信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/env/linux/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/env/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**代码示例(python)：**linux环境变量信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/env/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

### 环境变量信息查询扫描状态

用于查询环境变量信息的扫描状态；具体请求方式参照3.1。

## 内核模块信息

**功能描述**

该功能API用于查询资产中的内核模块信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

内核模块信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. 内核模块信息查询结果
2. 内核模块信息扫描
3. 内核模块信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用内核模块信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 内核模块信息查询结果

该接口用于查询内核模块信息，按每台主机每个内核模块进行展示。

**调用接口：**

GET /external/api/assets/kernelmodule/linux

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| moduleName | String | 否 | 模块名称 |
| path | String | 否 | 模块路径 |
| version | String数组 | 否 | 模块版本 |

**请求示例：**

/external/api/assets/kernelmodule/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "moduleName": "ablk\_helper",  
 "description": "",  
 "path": "/lib/modules/4.4.0-21-generic/kernel/crypto/ablk\_helper.ko",  
 "version": "",  
 "size": 16384,  
 "depends": [  
 "cryptd"  
 ],  
 "holders": [  
 "aesni\_intel"  
 ]  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| moduleName | String | varchar(512) | 模块名称 |
| description | String | varchar(512) | 模块描述 |
| path | String | varchar(128) | 模块路径 |
| version | String | varchar(128) | 模块版本 |
| size | String | varchar(128) | 模块大小 |
| depends | String数组 | varchar(1024) | 其依赖的模块进程 |
| holders | String数组 | varchar(1024) | 依赖其的内核模块 |

**代码示例(python)：**linux内核模块

def kernelmodule\_linux():  
 url = "http://%s:%s/external/api/assets/kernelmodule/linux?moduleName=ahci&path=/lib/modules/4.8.0-36-generic/kernel/drivers/ata/ahci.ko&version=3.0,2.21,1.4" % (host, port)  
 data = {'moduleName': 'ahci', 'path': '/lib/modules/4.8.0-36-generic/kernel/drivers/ata/ahci.ko', 'version': '3.0,2.21,1.4'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)，对于完整的模拟请求参考[启动项信息查询结果](#启动项信息查询结果)的windows代码示例(python)

### 内核模块信息扫描

该接口用于请求内核模块信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/kernelmodule/linux/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/kernelmodule/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**代码示例(python)：**linux内核模块信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/kernelmodule/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

### 内核模块信息查询扫描状态

用于查询内核模块信息的扫描状态；具体请求方式参照3.1。

## 安装包信息

**功能描述**

该功能API用于查询资产中的安装包信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

安装包信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1.安装包信息查询结果  
2.安装包信息扫描  
3.安装包信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用安装包信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### 安装包信息查询结果

该接口用于查询安装包信息，按每台主机每个安装包进行展示。

**调用接口：**

GET /external/api/assets/pkg/{win|linux}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | 安装包名 |
| type | String数组 | 否 | 安装类型（仅Linux） |
| installTime | DateRange | 否 | 安装时间 |
| publisher | String | 否 | 软件发布者(仅windows) |
| path | String | 否 | 安装路径(仅windows) |
| productVersion | String | 否 | 安装版本(仅windows) |
| psize | Ingeter | 否 | psize=1,pkgSize=[0,1MB];psize=2,pkgSize=[1,10MB]; psize=3,pkgSize=[10,100MB]; psize=4,pkgSize=[100,1024MB];psize=5,pkgSize=[1024,MB]; （仅windows） |

**请求示例：**

/external/api/assets/pkg/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "name": "im-chooser",  
 "summary": "Desktop Input Method configuration tool ",  
 "version": "1.3.1",  
 "installTime": "2017-10-31 21:39:39",  
 "type": "rpm",  
 "publisher": null,  
 "pkgSize": null,  
 "path": null,  
 "productVersion": null  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(512) | 安装包名 |
| version | String | varchar(512) | 版本 |
| type | String | varchar(128) | 安装类型（仅Linux） |
| installTime | Date | date | 安装时间 |
| summary | String | varchar(1024) | 总述(仅linux) |
| publisher | String | varchar(128) | 软件发布者(仅windows) |
| path | String | varchar(128) | 安装路径(仅windows) |
| productVersion | String | varchar(128) | 安装版本(仅windows) |
| pkgSize | Long | bigint(10) | 软件大小(仅windows) |

**代码示例(python)：**linux安装包

def pkg\_linux():  
 url = "http://%s:%s/external/api/assets/pkg/linux?name=bash-completion&version=1:2.1-4.2ubuntu1&installTime=2018-06-27%%2000:00:00%%20-%%202018-06-29%%2000:00:00&type=dpkg" % (host, port)  
 data = {'name': 'bash-completion', 'version': '1:2.1-4.2ubuntu1', 'installTime': '2018-06-27 00:00:00 - 2018-06-29 00:00:00', 'type': 'dpkg'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)，对于完整的模拟请求参考[启动项信息查询结果](#启动项信息查询结果)的windows代码示例(python)

**代码示例(python)：**win安装包

def pkg\_win():  
 url = "http://%s:%s/external/api/assets/pkg/win?name=Google%%20Chrome%%20Canary&publisher=Google%%20Inc.&installTime=2018-05-29%%2000:00:00%%20-%%202018-06-01%%2000:00:00&size=1&productVersion=69.0.3444.1&path=C:%%5cUsers%%5cadmin.ZHONGYANGLI%%5cAppData%%5cLocal%%5cGoogle%%5cChrome%%20SxS%%5cApplication" % (host, port)  
 data = {'name': 'Google Chrome Canary', 'publisher': 'Google Inc.', 'installTime': '2018-05-29 00:00:00 - 2018-06-01 00:00:00', 'size': 1, 'productVersion': '69.0.3444.1','path': 'C:\Users\\admin.QINGTENG\AppData\Local\Google\Chrome SxS\Application'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)，对于完整的模拟请求参考[启动项信息查询结果](#启动项信息查询结果)的windows代码示例(python)

### 安装包信息扫描

该接口用于请求安装包信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/pkg/{win|linux}/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/pkg/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**代码示例(python)：**linux安装包信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/pkg/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

**代码示例：**win安装包信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/pkg/win/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

### 安装包信息查询扫描状态

用于查询安装包信息的扫描状态；具体请求方式参照3.1。

## jar包信息

**功能描述**

该功能API用于查询资产中的jar包信息，获取查询的结果并展示；每条数据均包含主机信息，您可以灵活的使用该API组合出任何需要的显示结果。

**功能使用**

jar包信息查询有两种方式：

1. 直接获取当前信息；
2. 更新数据后，再获取最新信息。

为了满足上述功能，我们提供了三个接口：

1. jar包信息查询结果
2. jar包信息扫描
3. jar包信息查询扫描状态

使用方法如下：

* 直接查询当前信息则直接调用jar包信息查询结果接口（即1号接口）；
* 获取更新数据则先调用2号接口进行更新再调用3号接口查看是否更新完成，最后再调用1号接口查询最新信息。

接口的调用方法、参数和返回信息等具体如下所示。

### jar包信息查询结果

该接口用于查询jar包信息，按每台主机每个jar包进行展示。

**调用接口：**

GET /external/api/assets/jar\_pkg/linux

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 唯一标识Agent的ID |
| groups | Integer数组 | 否 | 业务组ID |
| hostname | String | 否 | 主机名（模糊查询） |
| ip | String | 否 | 主机IP（模糊查询） |
| name | String | 否 | 包名 |
| version | String数组 | 否 | 版本 |
| type | Integer数组 | 否 | 类型 （1：应用程序，2：系统类库，3:web服务自带库，8：其他依赖包） |
| executable | 布尔数组 | 否 | 是否可执行 |
| path | String | 否 | 绝对路径 |

**请求示例：**

/external/api/assets/jar\_pkg/linux

**返回示例：**

{  
 "total":1,  
 "rows":[  
 {  
 "agentId": "fe107d9c77f4c77a",  
 "displayIp": "192.168.78.131",  
 "connectionIp": "172.16.2.244",  
 "externalIp": null,  
 "internalIp": "192.168.78.131",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "12345",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "xdu-vm-cent51",  
 "name": "rt.jar",  
 "path": "/opt/IBM/WebSphere/AppServer/java/jre/lib/rt.jar",  
 "executable": false,  
 "version": "1.8.0",  
 "type": 2  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | agent uuid |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | String数组 | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| name | String | varchar(512) | 包名 |
| version | String | varchar(512) | 版本 |
| type | String | varchar(128) | 类型 |
| executable | 布尔 | tinyint(2) | 是否可执行 |
| path | String | varchar(128) | 绝对路径 |

**代码示例(python)：**linux-jar安装包

def jar\_pkg\_linux():  
 url = "http://%s:%s/external/api/assets/jar\_pkg/linux?name=ibmsaslprovider.jar&executable=false&version=8.0&type=2&path=/opt/IBM/WebSphere/AppServer/java/jre/lib/ext/ibmsaslprovider.jar" % (host, port)  
 data = {'name': 'ibmsaslprovider.jar', 'executable': 'false', 'version': '8.0', 'type': 2, 'path': '/opt/IBM/WebSphere/AppServer/java/jre/lib/ext/ibmsaslprovider.jar'}  
 res = send\_request("GET", url, data)  
 print "result: ",res

对于签名的部分，有疑问可以参考[身份认证](#身份认证)与[请求签名](#请求签名)，对于完整的模拟请求参考[启动项信息查询结果](#启动项信息查询结果)的windows代码示例(python)

### jar包信息扫描

该接口用于请求jar包信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/jar\_pkg/linux/refresh

**请求参数：**

无

**请求示例：**

/external/api/assets/jar\_pkg/linux/refresh

**返回示例：**

{  
 id: "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**代码示例(python)：**linux-jar包信息扫描更新数据

def refresh():  
 url = "http://%s:%s/external/api/assets/jar\_pkg/linux/refresh" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 return res

### jar包信息查询扫描状态

用于查询jar包信息的扫描状态；具体请求方式参照3.1。

## 批量卸载Agent

**功能描述**

该功能API用于批量卸载指定agent或者批量卸载离线超过七天的agent；您可以灵活的使用该API进行Agent卸载。

**功能使用**

卸载主机有两种方式：

1. 指定AgentId卸载；
2. 批量卸载离线超过７天的主机。

为了满足上述功能，我们提供了1个接口，接口的调用方法、参数和返回信息等具体如下所示。

### 卸载主机

该接口用于查询主机信息，按每台主机进行展示；

**调用接口：**

POST /external/api/assets/hostoperation/deletehost/{linux|win|unix}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | String数组 | 否 | 唯一标识Agent的ID |
| allOffline | Boolean | 否 | 是否批量删除离线超过7天主机，如果该参数为true，表示要批量删除离线超过七天的主机，忽略 agentIds参数 |
| password | String | 是 | 用户密码 |

注意：如果agentIds不为空且allOffline值为true，会按照指定agent的场景删除对应的主机。

**请求参数：**批量卸载离线超过7天主机

{  
 "allOffline": true,  
 "password":"123456"  
 }

**请求参数：**批量卸载指定主机

{  
 "agentIds": ["1222222222"],  
 "password":"123456"  
 }

**请求示例：**

/external/api/assets/hostoperation/deletehost/linux  
 {  
 "agentIds": ["1222222222"],  
 "password":"123456"  
 }

**返回示例：**

{  
 "code": 200,  
 "message": null,  
 "data": null  
}

**返回rows部分说明：**

如果密码错误，则返回结果如下

{  
 "code": 100,  
 "message": "password error",  
 "data": null  
}

如果不是主帐号进行删除操作，则返回403，结果如下

{  
 "errorCode": 403,  
 "errorMessage": "用户无权限",  
 "data": null  
}

如果待批量删除超过7天离线主机不存在，则返回结果如下

{  
 "errorCode": 500,  
 "errorMessage": "no agent offline over 7 days",  
 "detail": null  
}

如果批量删除指定agent时agentIds参数为空，则返回结果如下

{  
 "errorCode": ４00,  
 "errorMessage": "参数agentIds未填写",  
 "detail": null  
}

如果批量删除指定agent时有agentId不存在的情况，则返回结果如下

{  
 "errorCode": 400,  
 "errorMessage": "参数错误",  
 "detail": null  
}

## 批量创建业务组

**功能描述**

该功能API用于批量创建业务组。用户通过调用该接口，传入表示业务组层级结构的json串，创建json串中的业务组。如果对于同一个用户，同一个平台（linux/win）,同一级业务组已经存在，就会跳过不创建。否则创建为新业务组。

**功能使用**

为了满足功能，提供了四个接口：

1. 批量创建linux业务组接口。
2. 批量创建windows业务组接口。
3. 查询所有linux业务组。
4. 查询所有win业务组。

调用接口的方法、参数和返回信息具体如下：

### 批量创建linux|windows业务组

该接口用于批量创建linux（或者windows）业务组；

**调用接口：**

POST /external/api/assets/group/batch\_create\_group/{linux|win}

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| name | String | 是 | 要创建的业务组名称（字符最大长度128） |
| description | String | 否 | 要创建的业务组描述，可以为空，也可以不传入 |
| sonGroup | String数组 | 否 | name业务组的子业务组，业务组最多四层 |

请求参数如下：

{  
 "name":"第一级业务组a",  
 "description":"第一级业务组a的描述",  
 "sonGroup":[  
 {  
 "name":"第二级业务组b1",  
 "description":"第二级业务组b1的描述",  
 "sonGroup":[  
 {  
 "name":"第三级业务组c1",  
 "description":"第三级业务组c1的描述",  
 "sonGroup":[  
 {  
 "name":"第四级业务组d1",  
 "description":"第四级业务组d1的描述",  
 "sonGroup":[  
  
 ]  
 },  
 {  
 "name":"第四级业务组d2",  
 "description":"第四级业务组d2的描述",  
 "sonGroup":[  
  
 ]  
 }  
  
 ]  
 },  
 {  
 "name":"第三级业务组c2",  
 "description":"第三级业务组c2的描述",  
 "sonGroup":[  
 {  
 "name":"第四级业务组d3",  
 "description":"第四级业务组d3的描述",  
 "sonGroup":[  
  
 ]  
 },  
 {  
 "name":"第四级业务组d4",  
 "description":"第四级业务组d4的描述",  
 "sonGroup":[  
  
 ]  
 }  
  
 ]  
 }  
 ]  
 },  
 {  
 "name":"第二级业务组b2",  
 "description":"第二级业务组b2的描述",  
 "sonGroup":[  
 {  
 "name":"第三级业务组c3",  
 "description":"第三级业务组c3的描述",  
 "sonGroup":[  
 {  
 "name":"第四级业务组d5",  
 "description":"第四级业务组d5的描述",  
 "sonGroup":[  
  
 ]  
 }  
 ]  
 }  
 ]  
 }  
 ]  
}

**请求示例：**

external/api/assets/group/batch\_create\_group/linux  
 {  
 "name":"第一级业务组a",  
 "description":"第一级业务组a的描述",  
 "sonGroup":[  
 {  
 "name":"第二级业务组b1",  
 "description":"第二级业务组b1的描述",  
 "sonGroup":[  
 {  
 "name":"第三级业务组c1",  
 "description":"第三级业务组c1的描述",  
 "sonGroup":[  
 {  
 "name":"第四级业务组d1",  
 "description":"第四级业务组d1的描述",  
 "sonGroup":[  
  
 ]  
 },  
 {  
 "name":"第四级业务组d2",  
 "description":"第四级业务组d2的描述",  
 "sonGroup":[  
  
 ]  
 }  
  
 ]  
 },  
 {  
 "name":"第三级业务组c2",  
 "description":"第三级业务组c2的描述",  
 "sonGroup":[  
 {  
 "name":"第四级业务组d3",  
 "description":"第四级业务组d3的描述",  
 "sonGroup":[  
  
 ]  
 },  
 {  
 "name":"第四级业务组d4",  
 "description":"第四级业务组d4的描述",  
 "sonGroup":[  
  
 ]  
 }  
  
 ]  
 }  
 ]  
 },  
 {  
 "name":"第二级业务组b2",  
 "description":"第二级业务组b2的描述",  
 "sonGroup":[  
 {  
 "name":"第三级业务组c3",  
 "description":"第三级业务组c3的描述",  
 "sonGroup":[  
 {  
 "name":"第四级业务组d5",  
 "description":"第四级业务组d5的描述",  
 "sonGroup":[  
  
 ]  
 }  
 ]  
 }  
 ]  
 }  
 ]  
}

**返回示例：**

{  
 "code": 200,  
 "message": null,  
 "data": "success"  
}

如果有错误，则返回结果如下

{  
 "code": 100,  
 "message": "数据结构不正确，存在name为空问题！[{\"sonGroup\":[{\"sonGroup\":[{\"name\":\"第四级业务组d1\",\"description\":\"第四级业务组d1的描述\"},{\"name\":\"第四级业务组d2\",\"description\":\"第四级业务组d1的描述\"}],\"name\":\"第三级业务组c3\",\"description\":\"第四级业务组d1\"},{\"sonGroup\":[{\"name\":\"第四级业务组d4\",\"description\":\"第四级业务组d4\"},{\"name\":\"第四级业务组d3\",\"description\":\"第四级业务组d3的描述\"}],\"name\":\"第三级业务组c1\",\"description\":\"第三级业务组c1的描述\"}],\"name\":\"\",\"description\":\"第二级业务组b2的描述\"},{\"sonGroup\":[{\"$ref\":\"$[0]\"},{\"sonGroup\":[{\"sonGroup\":[{\"name\":\"eeeee45\",\"description\":\"第四级目录eeeee45\"}],\"name\":\"eeeee33\",\"description\":\"第三级目录eeeee33\"}],\"name\":\"第二级业务组b1\",\"description\":\"第二级业务组b1的描述\"}],\"name\":\"\",\"description\":\"第一级业务组a的描述\"}]",  
 "data": "fail"  
}

### 查询所有linux|windows业务组

该接口用于获取已经存在的linux（或者windows）业务组；

**调用接口：**

GET external/api/assets/group/{linux|win}

**请求参数：**

无 |

**请求示例：**

external/api/assets/group/linux

**返回示例：**

[  
 {  
 "id": 254,  
 "parent": 0,  
 "name": "eeee1",  
 "relationshipName": "eeee1",  
 "owner": 1,  
 "platform": 1,  
 "description": "第一级目录"  
 },  
 {  
 "id": 255,  
 "parent": 254,  
 "name": "eeeee2",  
 "relationshipName": "eeee1/eeeee2",  
 "owner": 1,  
 "platform": 1,  
 "description": "第二级目录zi"  
 },  
 {  
 "id": 256,  
 "parent": 255,  
 "name": "eeeee3",  
 "relationshipName": "eeee1/eeeee2/eeeee3",  
 "owner": 1,  
 "platform": 1,  
 "description": "第三级zison"  
 },  
 {  
 "id": 257,  
 "parent": 256,  
 "name": "eeeee44",  
 "relationshipName": "eeee1/eeeee2/eeeee3/eeeee44",  
 "owner": 1,  
 "platform": 1,  
 "description": "第四级目录test"  
 }  
]

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | Long | bigint(20) | 业务组id |
| parent | Long | bigint(20) | 业务组父id，parent=0,该业务组是第一层业务组，无父业务组 |
| name | string | varchar(128) | 业务组名（最多128个字符长度） |
| relationshipName | string | varchar(512) | 业务组关系名（即从顶层业务组到当前业务组的全路径名） |
| owner | Long | bigint(20) | 用户id |
| platform | Byte | tinyint(4) | 平台（１:linux， ２:windows） |
| description | string | varchar(512) | 业务组描述 |

## 全部资产批量更新扫描

该接口用于批量请求资产信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/assets/{win|linux}/refresh

**请求参数：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| type | Integer | 范围类型（0：全部主机 1：agent 2:group） |
| group | Integer数组 | 标识业务组的ID |
| agentIds | String数组 | 唯一标识Agent的ID |

**请求示例：**

/external/api/assets/linux/refresh  
  
{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 id: "some job id"  
}

**情景说明：**

一个账号同时只能一个任务运行，因此调用这个接口前，应该先调用3.1查询状态接口，如果satus非Running，则可以创建任务，如果创建成功，则正常返回， 如果创建失败， 则抛出异常。

**如果有作业在运行返回**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

## 移动主机业务组

该接口用于批量修改主机的业务组。由于修改业务组信息对服务性能会产生较大的消耗，每30分钟最多修改5000主机的业务组。

**调用接口：**

POST /external/api/assets/host/batch\_move\_host

**请求参数：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| osType | Integer | 操作系统（1：linux 2:windows） |
| groupId | Integer | 标识业务组的ID |
| agentIds | String数组 | 唯一标识Agent的ID |

**请求示例：**

{  
 "osType": 1,  
 "groupId": 2,  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "errorCode": 200,  
 "errorMessage": null,  
 "detail": null  
}

**情景说明：**

如果单次修改agent数超过5000

{  
 "errorCode": 500,  
 "errorMessage": "the number of agents is too large, please keep it within 5000",  
 "detail": null  
}

如果30分钟内多次修改agent数量总和超过5000

{  
 "errorCode": 500,  
 "errorMessage": "the maximum number of executions in 30 minutes is 5000",  
 "detail":null  
}

# 风险发现

## 安全补丁

该接口提供所有的linux和windows补丁的检测和查询。

### 补丁扫描

该接口用于请求主机信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/vul/patch/{linux|win}/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回：**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**情景说明：**

一个账号同时只能一个任务运行，因此调用这个接口前，应该先调用4.1.2查询状态接口，如果satus非Running，则可以创建任务，如果创建成功，则正常返回， 如果创建失败， 则抛出异常。

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### 查询补丁扫描执行状态

**调用接口：**

GET /external/api/vul/patch/{linux|win}/check/status

**请求参数：**

无

**返回示例：**

{  
 "id":""//当前job的jobId, 如果为null, 则当前没有job执行  
 "status":"Running|Success|Failed"  
}

**情景说明：**

如果id为null，则说明此前从未有过job执行，status为Running。如果id不为null， 则id为最近一次job执行的jobId, status为最近这次job执行的状态。

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID，如果为null，则当前没有job执行 |
| status | string | varchar(7) | 执行中；成功；失败 |

**代码示例(python)：**linux补丁检测和状态

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "127.0.0.1"  
port = 80  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@xx.com", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)  
  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 print info  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 print to\_sign  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
  
def service\_patch\_check\_status():  
 url = "http://%s:%s/external/api/vul/patch/linux/check/status" % (host, port)  
 data = {}  
 res = send\_request("GET", url, data)  
 return res  
  
  
def is\_patch\_check\_running():  
 res = service\_patch\_check\_status()  
 data = json.loads(res)  
 status = data.get('status')  
 return status == 'Running'  
  
  
def service\_patch\_check():  
 if not is\_patch\_check\_running():  
 start\_patch\_check()  
  
  
def start\_patch\_check():  
 url = "http://%s:%s/external/api/vul/patch/linux/check" % (host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 print "result: ", res  
  
  
def get\_linux\_patch():  
 url = "http://%s:%s/external/api/vul/patch/linux/list?page=0&size=50" % (host, port)  
 data = {'page':0, 'size': 50}  
 res = send\_request("GET", url, data)  
 print "result: ", res  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 service\_patch\_check()

### 查询补丁检测结果

**调用接口：**

GET /external/api/vul/patch/{linux|win}/list

**请求参数：**

公共参数（详情参见[基本请求方法](#基本请求方法)）

**请求示例：**

/external/api/vul/patch/win/list

**返回示例：**

{  
 "total":2,  
 "rows":[  
 {  
 "id":"5ab243407d761b1147bfb826",  
 "agentId":"a3ec5aca181b179a",  
 "displayIp":"172.16.6.124",  
 "internalIp":"172.16.6.124",  
 "externalIp":null,  
 "hostname":"servera.qingteng.cn",  
 "group":1,  
 "remark":null,  
 "hostTags":[  
 "test1",  
 "test2"  
 ],  
 "patchId":"QT012016000970",  
 "whiteRuleEffect":false,  
 "businessImpact":0  
 },  
 {  
 "id":"5ab243407d761b1147bfb827",  
 "agentId":"a3ec5aca181b179a",  
 "displayIp":"172.16.6.124",  
 "internalIp":"172.16.6.124",  
 "externalIp":null,  
 "hostname":"servera.qingteng.cn",  
 "group":1,  
 "remark":null,  
 "hostTags":[  
 "test2"  
 ],  
 "patchId":"QT012015000173",  
 "whiteRuleEffect":false,  
 "businessImpact":0  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 这条扫描结果的唯一标识 |
| agentId | string | varchar(16) | 主机id |
| displayIp | string | varchar(15) | 主机显示ip |
| internalIp | string | varchar(15) | 内网ip |
| externalIp | string | varchar(15) | 外网ip |
| hostname | string | varchar(512) | 主机名 |
| group | int | bigint(20) | 业务组id |
| remark | string | varchar(1024) | 主机备注 |
| hostTags | List | varchar(1024) | 主机标签 |
| patchId | string | varchar(64) | 补丁id |
| whiteRuleEffect | boolean | varchar(1) | 是否匹配白名单 |
| businessImpact | int | tinyint(4) | 是否存在进程影响 0/1 无进程影响/有进程影响(仅linux, 并且当前进程影响没有未知状态, 当主机没有进行进程影响扫描的时候, 都是无进程影响, 进程影响每天有定时执行, 所以安装主机后如果要立马知道是否补丁有进程影响, 需要先执行一次进程影响扫描, 或者等第二天定时执行后可以获得补丁是否有进程影响) |

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "127.0.0.1"  
port = 80  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@xx.com", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)  
  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 print info  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 print to\_sign  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
def get\_linux\_patch():  
 url = "http://%s:%s/external/api/vul/patch/linux/list?page=0&size=50" % (host, port)  
 data = {'page':0, 'size': 50}  
 res = send\_request("GET", url, data)  
 print "result: ", res  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 get\_linux\_patch()

### 查询补丁详情

**调用接口：**

GET /external/api/vul/patch/{linux|win}/{id}

**请求参数：**

无

**请求示例：**

/external/api/vul/patch/linux/5ab243407d761b1147bfb827

**返回示例：**

{  
 "patchId": "QT012016000970", //补丁id  
 "kbNum": null, //kb号，只有windows有  
 "bulletId": null, //公告号，只有windows有  
 "patchName": "CentOS 6 : kernel (CESA-2016:0855)", //补丁名  
 "severity": 2, //危险程度  
 "publishTime": "2016-05-16 00:00:00", //补丁发布时间  
 "apps": [ //影响的app  
 "其他"  
 ],  
 "desc": "An update for kernel is now available for Red Hat Enterprise Linux 6. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The kernel packages contain the Linux kernel, the core of any Linux operating system. Security Fix(es) : \* It was found that reporting emulation failures to user space could lead to either a local (CVE-2014-7842) or a L2->L1 (CVE-2010-5313) denial of service. In the case of a local denial of service, an attacker must have access to the MMIO area or be able to access an I/O port. Please note that on certain systems, HPET is mapped to userspace as part of vdso (vvar) and thus an unprivileged user may generate MMIO transactions (and enter the emulator) this way. (CVE-2010-5313, CVE-2014-7842, Moderate) \* It was found that the Linux kernel did not properly account file descriptors passed over the unix socket against the process limit. A local user could use this flaw to exhaust all available memory on the system. (CVE-2013-4312, Moderate) \* A buffer overflow flaw was found in the way the Linux kernel's virtio-net subsystem handled certain fraglists when the GRO (Generic Receive Offload) functionality was enabled in a bridged network configuration. An attacker on the local network could potentially use this flaw to crash the system, or, although unlikely, elevate their privileges on the system. (CVE-2015-5156, Moderate) \* It was found that the Linux kernel's IPv6 network stack did not properly validate the value of the MTU variable when it was set. A remote attacker could potentially use this flaw to disrupt a target system's networking (packet loss) by setting an invalid MTU value, for example, via a NetworkManager daemon that is processing router advertisement packets running on the target system. (CVE-2015-8215, Moderate) \* A NULL pointer dereference flaw was found in the way the Linux kernel's network subsystem handled socket creation with an invalid protocol identifier. A local user could use this flaw to crash the system. (CVE-2015-8543, Moderate) \* It was found that the espfix functionality does not work for 32-bit KVM paravirtualized guests. A local, unprivileged guest user could potentially use this flaw to leak kernel stack addresses. (CVE-2014-8134, Low) \* A flaw was found in the way the Linux kernel's ext4 file system driver handled non-journal file systems with an orphan list. An attacker with physical access to the system could use this flaw to crash the system or, although unlikely, escalate their privileges on the system. (CVE-2015-7509, Low) \* A NULL pointer dereference flaw was found in the way the Linux kernel's ext4 file system driver handled certain corrupted file system images. An attacker with physical access to the system could use this flaw to crash the system. (CVE-2015-8324, Low) Red Hat would like to thank Nadav Amit for reporting CVE-2010-5313 and CVE-2014-7842, Andy Lutomirski for reporting CVE-2014-8134, and Dmitriy Monakhov (OpenVZ) for reporting CVE-2015-8324. The CVE-2015-5156 issue was discovered by Jason Wang (Red Hat). Additional Changes : \* Refer to Red Hat Enterprise Linux 6.8 Release Notes for information on new kernel features and known issues, and Red Hat Enterprise Linux Technical Notes for information on device driver updates, important changes to external kernel parameters, notable bug fixes, and technology previews. Both of these documents are linked to in the References section.", //补丁描述  
 "check": { //校验信息  
 "details": [  
 {  
 "os": "centos-6",  
 "name": "kernel",  
 "fixVersion": "2.6.32-642.el6",  
 "version": "2.6.32-431.el6"  
 }  
],  
 "remedCmd": "sudo yum update -y kernel"  
 },  
 "remedDesc": "这个补丁修复了Red Hat Enterprise Linux 6 kernel的多个漏洞，本地攻击者可利用该漏洞耗尽系统所有可用内存、使系统崩溃甚至提升权限。漏洞修复后需要系统重启。建议业务不繁忙时修复。", //修复建议  
 "remedCmd": "sudo yum update -y kernel", //修复命令 linux独有  
 "installPackage": null, //安装包地址, windows有  
 "cvssScore": "7",  
 "cvssDetail": "AV:L/AC:M/Au:N/C:C/I:C/A:C",  
 "ref": "CVE-2010-5313,CVE-2013-4312,CVE-2014-7842,CVE-2014-8134,CVE-2015-5156,CVE-2015-7509,CVE-2015-8215,CVE-2015-8324,CVE-2015-8543",  
 "restartOpts": 3,  
 "hasExp": true,  
 "isKernel": true,  
 "isLocalEscalation": false,  
 "isRemote": false,  
 "hasPoc": false,  
 "pocRefs": "",  
 "impacts": null,  
 "firstCheckTime":"2016-05-16 00:00:00",  
 "businessImpact":0  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| patchId | String | varchar(64) | 补丁id |
| kbNum | String | varchar(32) | kb号 （仅windows） |
| bulletId | String | char(32) | 公告号（仅windows） |
| patchName | String | varchar(1024) | 补丁名 |
| severity | Integer | tinyint(4) | 危险程度 |
| publishTime | date | date | 补丁发布时间 |
| apps | List | varchar(1024) | 影响的app |
| desc | String | varchar(2048) | 补丁描述 |
| check | String | text | 补丁校验信息（仅linux） |
| details | List | text | 校验信息详情(仅linux) |
| remedCmd | String | varchar(2048) | 补丁修复命令（仅linux） |
| remedDesc | String | text | 修复描述（仅linux） |
| installPackage | String | varchar(1024) | 安装包地址（仅windows） |
| cvssScore | String | float | Cvss分 |
| cvssDetail | String | varchar(64) | Cvss详情 |
| ref | String | mediumblob | Cve链接 |
| restartOpts | Integer | tinyint(4) | 修复影响:0-未知;1-不需要重启;2-服务重启;3-系统重启 |
| hasExp | Boolean | tinyint(4) | 是否存在exp |
| isKernel | Boolean | tinyint(4) | 内核漏洞（仅linux） |
| isLocalEscalation | Boolean | tinyint(4) | 本地提权（仅linux） |
| isRemote | Boolean | tinyint(4) | 远程执行（仅linux） |
| hasPoc | Boolean | tinyint(4) | 存在poc（仅linux） |
| pocRefs | String | varchar(1024) | Poc参考链接（仅linux） |
| impacts | List | tinyint(4) | 补丁特征（仅windows）：1-纵深防御；2-拒绝服务；3-权限提升；4-信息泄露；5-远程代码执行；6-安全特征绕过；7-电子欺骗；8-数据篡改 |
| firstCheckTime | date | date | 第一次被检测出该补丁的时间(仅linux) |
| businessImpact | int | tinyint(4) | 是否存在进程影响 0/1 无进程影响/有进程影响(仅linux, 并且当前进程影响没有未知状态, 当主机没有进行进程影响扫描的时候, 都是无进程影响, 进程影响每天有定时执行, 所以安装主机后如果要立马知道是否补丁有进程影响, 需要先执行一次进程影响扫描, 或者等第二天定时执行后可以获得补丁是否有进程影响) |

### 创建linux补丁进程影响任务

**调用接口：**

POST /external/api/vul/patch/linux/business\_impact/check

**请求参数：**

无

**请求示例：**

/external/api/vul/patch/linux/business\_impact/check

**返回示例：**

{  
 id: "some job id"  
}

**情景说明：**

一个账号同时只能一个任务运行，因此调用这个接口前，应该先调用4.1.6查询状态接口，如果status非Running，则可以创建任务，如果创建成功，则正常返回， 如果创建失败， 则抛出异常。

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID，如果为null，则创建失败 |

### 查询linux补丁进程影响任务执行状态

**调用接口：**

GET /external/api/vul/patch/linux/business\_impact/check/status

**请求参数：**

无

**请求示例：**

/external/api/vul/patch/linux/business\_impact/check/status

**返回示例：**

{  
 "id":""//当前job的jobId, 如果为null, 则当前没有job执行  
 "status":"running|success|failed"  
}

**情景说明：**

如果id为null，则说明此前从未有过job执行，status为Running。如果id不为null， 则id为最近一次job执行的jobId, status为最近这次job执行的状态。

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID， 如果为null， 则当前没有job执行 |
| status | string | varchar(7) | running;success;failed |

### 查询某个补丁的进程影响

**调用接口：**

GET /external/api/vul/patch/linux/business\_impact/{patchId}

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | string | varchar(24) | 主机id |

**请求示例：**

/external/api/vul/patch/linux/business\_impact/QT012014001323?agentId=e9ee72243b0d879c

**返回示例：**

[  
 {  
 "agentId": "e9ee72243b0d879c",  
 "patchId": "QT012014001323",  
 "pid": 15669,  
 "processName": "bash",  
 "startArgs": "/bin/bash -l",  
 "pkg": {  
 "Ver": "5.2",  
 "Name": "libreadline5",  
 "Release": "147.17.30"  
 },  
 "ports": []  
 },  
 {  
 "agentId": "e9ee72243b0d879c",  
 "patchId": "QT012014001323",  
 "pid": 15816,  
 "processName": "bash",  
 "startArgs": "-bash",  
 "pkg": {  
 "Ver": "3.2",  
 "Name": "bash",  
 "Release": "147.17.30"  
 },  
 "ports": []  
 },  
 {  
 "agentId": "e9ee72243b0d879c",  
 "patchId": "QT012014001323",  
 "pid": 1392,  
 "processName": "qpidd",  
 "startArgs": "/usr/sbin/qpidd --data-dir /var/lib/qpidd --daemon",  
 "pkg": {  
 "Ver": "2.12-1.47.el6",  
 "Name": "qpidd",  
 "Release": null  
 },  
 "ports": [  
 {  
 "bindIp": "0.0.0.0",  
 "port": 5672,  
 "proto": "tcp"  
 }  
 ]  
 }  
]

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | string | varchar(16) | 主机id |
| patchId | string | varchar(14) | 补丁id |
| pid | Integer | int(10) | 进程id |
| processName | String | varchar(1024) | 进程名 |
| startArgs | String | varchar(2048) | 启动参数 |
| pkg | Object | text | 安装包详情：Ver-安装包版本；Name-安装包名；Release-安装包发性版本号 |
| ports | List | text | 影响进程关联的端口号信息列表 |

**Port返回信息说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| bindIp | string | varchar(64) | 端口绑定的Ip信息 |
| port | Integer | int(10) | 端口号 |
| proto | string | varchar(14) | 协议：udp/tcp等 |

## Linux应用风险/系统风险/对外访问性/账号风险

该接口提供所有的Linux应用风险，系统风险，对外访问性，账号风险的检测和查询。

### 风险扫描

该接口用于请求主机信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/vul/{app|system|access|account}/linux/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回：**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**场景说明：**

这个api调用前需要先调用4.2.2查询风险扫描执行状态接口，如果该接口返回状态非Running， 则可以调用该风险扫描接口创建job。

**异常说明：**

如果任务创建失败， 抛出异常。

### 查询风险扫描执行状态

**调用接口：**

GET /external/api/vul/{app|system|access|account}/linux/check/status

**请求参数：**

无

**请求示例：**

/external/api/vul/app/linux/check/status

**返回示例：**

{  
 "status":"Running|Success|Failed"  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| status | string | varchar(7) | "执行中;成功;失败"，最近一次job执行状态，如果从未执行过job，状态为Success |

### 查询风险检测结果

**调用接口：**

GET /external/api/vul/ {app|system|access|account}/linux/list

**请求参数：**

公共参数（详情参见[基本请求方法](#基本请求方法)）

**请求示例：**

/external/api/vul/app/linux/list

**返回示例：**

{  
  
 "rows":[  
 {  
 "agentId": "526fb2d7dd5ec7e9",  
 "displayIp": "172.16.5.118",  
 "connectionIp": "172.16.5.118",  
 "externalIp": null,  
 "internalIp": "172.16.5.118",  
 "bizGroupId": 4,  
 "bizGroup": "qingtenggroup",  
 "remark": null,  
 "hostTagList": [  
 "所有ip范围"  
 ],  
 "hostname": "qingteng.qingteng.cn",  
 "id": "5b3a8b567d761b22befc8e16",  
 "vulId": "QT032016001411",  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 这条扫描结果的唯一标识 |
| agentId | String | varchar(16) | agent uuid,16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectinIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | int | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| vulId | string | varchar(14) | 风险id |
| whiteRuleEffect | boolean | tinyint(1) | 是否匹配白名单(true/false) |

### 查询风险详情

**调用接口：**

GET /external/api/vul/ {app|system|access|account}/linux/{id}

**请求参数：**

无

**请求示例：**

/external/api/vul/app/linux/5b3a8b567d761b22befc8e16

**返回示例：**

{  
 "vulId": "QT032016001411",  
 "vulName": "SSH服务用户home目录中存在未加密私钥",  
 "severity": 3,  
 "apps": [  
 "SSH"  
 ],  
 "desc": "默认情况下在用户的home目录中存储的未加密私钥，一旦该服务器被入侵，则可能导致该私钥泄露。私钥是登录服务器的重要凭证，如果私钥被攻击者利用，可通过私钥进行ssh登录认证获取系统权限，带来不可预知的高危风险。",  
 "restartOpts": 2,  
 "publicDate": "2016-12-02 00:00:00",  
 "checkInfo": "检查所有用户home目录下是否存在未加密私钥",  
 "checkResult": "",  
 "remedDesc": "删除对应的私钥，若已经使用过该私钥对应的公钥进行登录，则应注意更换所有使用该公钥的主机的密钥",  
 "refs": ""  
 "firstCheckTime":"2016-12-02 00:00:00"  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | String | varchar(14) | 风险id |
| vulName | String | varchar(512) | 风险名 |
| severity | Integer | tinyint(4) | 危险程度：0-信息；1-低危；2-中危；3-高危；4-危急 |
| apps | List | varchar(1024) | 影响的应用 |
| desc | String | text | 风险描述 |
| restartOpts | Integer | tinyint(4) | 重启选项：0-未知；1-无需重启；；2-服务重启；3-系统重启 |
| publicDate | Date | date | 发布时间 |
| checkInfo | String | text | 验证信息 |
| checkResult | String | text | 验证结果 |
| remedDesc | String | text | 修复建议 |
| refs | String | varchar(2048) | 引用信息 |
| firstCheckTime | Date | Date | 首次检测出该风险的时间 |

## Linux/Windows弱密码

该接口提供Linux和Windows弱密码的检测和查询。

### 风险扫描

该接口用于请求主机信息扫描，更新数据查询的结果。

**调用接口：**

POST /external/api/vul/ weakpwd/{linux|win}/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回：**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**场景说明：**

这个api调用前需要先调用4.3.2查询风险扫描执行状态接口，如果该接口返回状态非Running， 则可以调用该风险扫描接口创建job。

**异常说明：**

如果任务创建失败， 抛出异常。

### 查询风险扫描执行状态

**调用接口：**

GET /external/api/vul/weakpwd/{linux|win}/check/status

**请求参数：**

无

**请求示例：**

/external/api/vul/weakpwd/linux/check/status

**返回示例：**

{  
 "status":"Running|Success|Failed"  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| status | string | varchar(7) | "执行中;成功;失败"，最近一次job执行状态，如果从未执行过job，状态为Success |

### 查询风险检测结果

**调用接口：**

GET /external/api/vul/weakpwd/{linux|win}/list

**请求参数：**

公共参数（详情参见[基本请求方法](#基本请求方法)）

**请求示例：**

/external/api/vul/weakpwd/linux/list

**返回示例：**

{  
 "rows": [  
 {  
 "id": "5b1edbed7d761b59ad1e7787",  
 "agentId": "c7b5cd9f4bd927b7",  
 "displayIp": "172.16.2.226",  
 "internalIp": "172.16.2.226",  
 "externalIp": null,  
 "hostname": "qt-System-Product-Name",  
 "group": 4,  
 "remark": "d",  
 "hostTags": [  
 "11111111111111111111111111111111",  
 "hello world"  
 ],  
 "vulId": "QT052016001395",  
 "whiteRuleEffect": false  
 }  
 ],  
"total": 1  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 这条扫描结果的唯一标识 |
| agentId | String | varchar(16) | agent uuid,16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectinIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | int | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| vulId | string | varchar(14) | 风险id |
| whiteRuleEffect | boolean | tinyint(1) | 是否匹配白名单 |

### 查询风险详情

**调用接口：**

GET /external/api/vul/weakpwd/{linux|win}/{id}

**请求参数：**

无

**请求示例：**

/external/api/vul/weakpwd/linux/5b1edbed7d761b59ad1e7787

**返回示例：**

"vulId": "QT052016001395",  
 "vulName": "Redis服务存在弱密码",  
 "apps": [  
 "Redis"  
 ],  
 "desc": "Redis配置文件中requirepass配置项为访问口令，如果配置弱口令，攻击者可通过字典快速破解该口令。攻击者连接到Redis可获取敏感信息，并根据Redis权限对服务器进行操作。如果为弱口令，攻击者可快速获取Redis访问权限，并且管理员有一定几率不会发现该爆破过程。",  
 "restartOpts": 2,  
 "publicDate": null,  
 "app": "Redis",  
 "uname": "",  
 "accountStatus": 1,  
 "weakType": 1,  
 "version": "3.0.6",  
 "binPath": "/usr/bin/redis-server",  
 "password": "",  
 "unChangePwdDays": null,  
 "accountLoginType":  
 null,  
 "interactiveLoginType": null,  
 "pwdStatus": null,  
 "firstCheckTime":"2016-12-02 00:00:00",  
 "bindIp":"127.0.0.1",  
 "port":6382,  
 "pid":111,  
 "remote":false,  
 "root":false,  
 "loginShell":1  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | string | varchar(14) | 风险id（仅linux） |
| vulName | string | varchar(512) | 风险名（仅linux） |
| apps | List | varchar(1024) | 影响的应用（仅linux） |
| restartOpts | Integer | tinyint(4) | 重启选项：0-未知；1-无需重启；2-服务重启；3系统重启 （仅linux） |
| publicDate | Date | date | 发布时间（仅linux） |
| app | String | varchar(128) | 弱密码应用（仅linux） |
| uname | String | varchar(256) | 用户名 |
| accountStatus | Integer | tinyint(4) | 账号状态：对于windows：0-正常；1-锁定；2-禁用；对于linux主机：0-禁用；1-正常） |
| weakType | Integer | tinyint(4) | 弱密码类型（1/2/3/4 1-空密码； 2-默认弱密码；3-跟用户名相同；4-常见弱密码） |
| version | string | varchar(64) | 应用版本号（仅linux） |
| binPath | string | varchar(512) | 应用路径（仅linux） |
| password | string | varchar(32) | 密码 |
| unChangePwdDays | Integer | int(10) | 密码未修改天数（仅windows） |
| accountLoginType | Integer | tinyint(4) | ssh账号登录方式：0-不可登录；1-key登录；2-pwd登录；3-key&pwd登录（仅linux， ssh） |
| interactiveLoginType | Integer | tinyint(4) | ssh账号交互登录方：0/1/2 0-不可登录；1-不可交互登录；2-可交互登录（仅linux， ssh） |
| pwdStatus | Integer | tinyint(4) | 密码状态：1/2/3/4 1-正常；2-将要失效；3-已经失效；4-已锁定（仅linux， ssh） |
| firstCheckTime | Date | Date | 第一次发现该弱密码的时间(仅linux) |
| bindIp | String | varchar(15) | 绑定ip(仅linux) |
| port | String | varchar(10) | 绑定端口(仅linux) |
| pid | String | varchar(10) | 进程id(仅linux) |
| root | boolean | tinyint(1) | 是否root权限运行, true/false 是/否(仅linux) |
| remote | boolean | tinyint(1) | 是否对外访问,true/false 是/否(仅linux) |
| loginShell | Integer | tinyint(4) | shell登录性：0/1 0-非登录shell；1-登录shell |

## Web风险文件

该接口提供web风险文件检测和查询。

### web风险文件查询结果

该接口提供web风险文件数据查询。

**调用接口：**

GET /external/api/websecurity/weakfile/{linux|win}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 主机ID |
| hostname | String | 否 | 主机名 |
| groups | int数组 | 否 | 主机组 |
| filePath | String | 否 | 文件名 |
| serverName | String | 否 | 域名 |
| serverNameEq | String | 否 | 域名，精确查找 |
| ip | String | 否 | 主机IP |
| regexTypes | int数组 | 否 | 风险文件类型 |
| port | int | 否 | 主机端口 |
| portStatus | int数组 | 否 | 端口访问性：-1-未知；0-关闭；1-打开 |
| translate | int数组 | 否 | 解析方式：0-host解析；1-dns解析 |

**请求示例：**

/external/api/websecurity/weakfile/linux?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "70db8ef89e9ae79a",  
 "displayIp": "172.16.159.1",  
 "connectionIp": "172.16.2.231",  
 "externalIp": null,  
 "internalIp": "172.16.159.1",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "hostname",  
  
 "hostname": "nan.zhang",  
  
 "id": "5a5d08417d761b67a5bb712b",  
 "port": 80,  
 "group": 1,  
 "portStatus": -1,  
 "translate": 1,  
 "serverName": "www.zhn.com",  
 "filePath": "/home/qt/nan/test.doc",  
 "regexLevel": 3,  
 "regexType": 8,  
 "regexTypeDesc": "Office文档",  
 "checkUrl": "http://www.zhn.com:80/test.doc",  
 "modifyTime": 1504234128,  
 "download": false,  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| id | String | varchar(24) | ID |
| port | Integer | int(10) | 端口 |
| portStatus | Integer | tinyint(2) | 端口访问性：-1-未知；0-对内；1-对外 |
| translate | Integer | tinyint(2) | 解析方式：0-host解析；1-dns解析 |
| serverName | String | varchar(128) | 域名 |
| filePath | String | varchar(128) | 文件路径 |
| regexLevel | Integer | int(6) | 危险等级：1-低危；2-中危；3-高危 |
| regexType | Integer | int(6) | 类型：'1'-'临时文件'; '2'-'压缩文件';'3'-'备份文件泄露';'4'-'数据文件泄露'；'5'-'配置文件泄露';'6'-'日志泄露';'7'-'脚本泄露';'8'-'Office文档';'9'-'源代码泄露';'10'-'系统文件';'11'-'phpinfo文件' |
| regexTypeDesc | String | varchar(128) | 类型描述 |
| checkUrl | String | varchar(128) | 验证路径 |
| modifyTime | Integer | int(10) | 最后修改时间：时间戳，精确到秒 |
| download | Boolean | tinyint(1) | 是否可下载 |
| whiteRuleEffect | Boolean | tinyint(1) | 是否命中白名单 |

### Web风险文件扫描

该接口提供web风险文件执行扫描任务。

**调用接口：**

POST /external/api/websecurity/weakfile/{linux|win}/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "flag": true  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### Web风险文件查询扫描状态

该接口提供web风险文件扫描任务的状态查询。

**调用接口：**

GET /external/api/websecurity/weakfile/{linux|win}/check/status

**请求参数：**

无

**请求示例：**

/external/api/websecurity/weakfile/linux/check/status

**返回示例：**

{  
 "retcode": 3  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| retcode | Integer | tinyint(1) | -1-无任务；0-未开始执行；1-排队执行；2-正在执行；3-执行成功；4-执行失败 |

### 生成下载文件

该接口提供web风险文件的生成。

**调用接口：**

POST /external/api/websecurity/weakfile/{linux|win}/download

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | int | 是 | 数据id |

**请求示例：**

/external/api/websecurity/weakfile/{linux|win}/download

**返回示例：**

{  
 "code": 200,  
 "message": null,  
 "data": {  
 "ready": 1,  
 "token": "558a45bedc41046dd1c4342d0e75e269"  
 }  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| ready | Integer | tinyint(1) | 0-文件没有准备好，应该轮巡继续调用该接口；1-文件已经准备好；-1-文件下载失败，应提示失败并停止轮询 |
| token | sting | varchar(128) | 下载的令牌 |

### Web风险文件下载

该接口提供web风险文件下载。

**调用接口：**

GET /external/api/websecurity/weakfile/{linux|win}/download/{id}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | int | 是 | 数据id |

**请求示例：**

/external/api/websecurity/weakfile/linux/download/5a790f0b7d761b6bc78e1116?token=558a45bedc41046dd1c4342d0e75e269

**返回示例：**

无

**返回rows部分说明：**

文件内容，文件名在header头中。

## Linux/Windows漏洞检测

该接口提供Linux/Windows漏洞检测和查询，Linux/Windows漏洞检测作业管理。

### Linux漏洞检测扫描

该接口提供Linux漏洞检测开始执行扫描任务功能。

**调用接口：**

POST /external/api/vul/poc/{linux|win}/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回：**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

**场景说明：**

这个api调用前需要先调用4.5.2查询风险扫描执行状态接口，如果该接口返回状态非Running， 则可以调用该漏洞检测扫描接口创建漏洞检测扫描job。

**异常说明：**

如果任务创建失败， 抛出异常。

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
from \_\_future\_\_ import absolute\_import  
  
import httplib  
import json  
import time  
import hashlib  
  
host = "tests.qingteng.cn"  
port = 80  
  
  
# 登录请求调用示例  
def login():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/v1/api/auth" % (host, port)  
 header = {"Content-Type": "application/json"}  
 body = {"username": "dev@qingteng.cn", "password": "abc@123"}  
 json\_body = json.dumps(body)  
 conn.request(method="POST", url=url, body=json\_body, headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return json.loads(res)  
  
  
# 发送请求  
def send\_request(method, url, data):  
 # 参看登录认证里面的登录方法代码示例  
 login\_result = login()  
 sign\_key = login\_result.get("data").get("signKey")  
 jwt = login\_result.get("data").get("jwt")  
 comid = login\_result.get("data").get("comId")  
  
 # 当前时间戳  
 ts = int(time.time())  
  
 if data is not None:  
 info = ""  
 if method == "GET":  
 # 对参数key进行字典排序  
 keys = sorted(data.keys())  
 for key in keys:  
 info = info + key + str(data.get(key))  
 print info  
 elif method == "POST" or method == "PUT" or method == "DELETE":  
 info = json.dumps(data)  
 # 拼接待签名字符串  
 to\_sign = comid + info + str(ts) + sign\_key  
 else:  
 # 拼接待签名字符串  
 to\_sign = comid + str(ts) + sign\_key  
  
 print to\_sign  
 # 对待签名字符串进行sha1得到签名字符串  
 sign = hashlib.sha1(to\_sign).hexdigest()  
  
 # 组装http请求头参数  
 header = {"Content-Type": "application/json", "comId": comid, "timestamp": ts,  
 "sign": sign, "Authorization": "Bearer " + jwt}  
  
 conn = httplib.HTTPConnection(host, port)  
 conn.request(method=method, url=url, body=json.dumps(data), headers=header)  
 response = conn.getresponse()  
 res = response.read()  
 return res  
  
# linux启动漏洞扫描的代码示例(python)  
def start\_poc\_check():  
 url = "http://%s:%s/external/api/vul/poc/linux/check" % (  
 host, port)  
 data = {}  
 res = send\_request("POST", url, data)  
 print "result: ", res  
  
# linux漏洞检测，检查运行应用  
def check\_app\_running():  
 url = "http://%s:%s/external/api/vul/poc/check\_app\_running" % (  
 host, port)  
 data = {"vulId":"QT042017002027"}  
 res = send\_request("POST", url, data)  
 print "result: ", res  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 start\_poc\_check()  
 check\_app\_running()

### 查询漏洞检测扫描执行状态

**调用接口：**

GET /external/api/vul/poc/{linux|win}/check/status

**请求参数：**

无

**请求示例：**

/external/api/vul/poc/linux/check/status

**返回示例：**

{  
 "status":"Running|Success|Failed"  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| status | string | varchar(7) | "执行中;成功;失败"，最近一次漏洞检测扫描job执行状态，如果从未执行过漏洞检测扫描job，状态为Success |

### 查询漏洞检测结果

**调用接口：**

GET /external/api/vul/poc/linux/list

**请求参数：**

无

**请求示例：**

/external/api/vul/poc/{linux|win}/list

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "d27260c9bb52d82c",  
 "displayIp": "172.16.2.184",  
 "connectionIp": "172.16.2.184",  
 "externalIp": null,  
 "internalIp": "172.16.2.184",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": null,  
 "hostTagList": null,  
 "hostname": "yongliang",  
 "id": "5c3d9930871e1b31a1c5c2c1",  
 "vulId": "QT042014001964",  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1,  
 "charts": {}  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 这条扫描结果的唯一标识 |
| agentId | String | varchar(16) | agent uuid,16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectinIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | int | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| vulId | string | varchar(14) | 风险id |
| whiteRuleEffect | boolean | tinyint(1) | 是否匹配白名单 |

### 查询漏洞风险详情

**调用接口：**

GET /external/api/vul/poc/{linux|win}/{recordId}

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| recordId | string | varchar(24) | 扫描结果的唯一标识 |

**请求示例：**

/external/api/vul/poc/linux/5c3d9930871e1b31a1c5c2c1

**返回示例：**

{  
 "vulId": "QT042014001966",  
 "vulName": "Bash环境变量远程命令执行漏洞(CVE-2014-7169)",  
 "apps": [  
 "Bash"  
 ],  
 "desc": "GNU Bash 存在安全漏洞，该漏洞源于GNU Bash 允许在环境变量的值中的函数定义，及在函数定义后加入额外的字符串，攻击者可利用此特性在远程写入文件或执行其他可以影响到系统的操作。如涉及ForceCommand功能的向量所示OpenSSH sshd，Apache HTTP Server中的mod\_cgi和mod\_cgid模块，由未指定的DHCP客户端执行的脚本，以及在Bash执行的特权边界内设置环境的其他情况。由于CVE-2014-6271的修复程序不完整，因此存在此漏洞。",  
 "restartOpts": 3,  
 "publicDate": "2014-09-24 00:00:00",  
 "firstCheckTime": "2019-05-20 01:30:23",  
 "family": 4,  
 "severity": 4,  
 "checkInfo": "版本比对检测原理：检查当前系统中Bash版本是否在受影响版本内|版本比对检测结果：- bash\r\n 当前安装版本：3.2-24.el5\r\n 漏洞修复版本：3.2-33.el5\_11.4\r\n该主机存在此漏洞|POC检测原理：检查当前系统中Bash版本是否在受影响版本内|POC检测结果：当前主机存在漏洞，在/dev/shm目录中发现Poc生成的文件/dev/shm/QT\_Test\_1558287004",  
 "pocCheckInfo": "bash官方在修复CVE-2014-6271漏洞时，由于考虑情况不全，导致漏洞修复可以被绕过，从而致使新的漏洞出现。",  
 "checkResult": null,  
 "data": "{\"code\": 0, \"execute\_type\": 3, \"cve\_id\": \"CVE-2014-7169\", \"poc\_check\_result\": {\"vuln\": 1, \"code\": 0, \"result\": {\"vuls\": [{\"msg\": \"\\u5b58\\u5728\\u6f0f\\u6d1e\", \"vuln\": true, \"info\": \"\\u5728/dev/shm\\u76ee\\u5f55\\u4e2d\\u53d1\\u73b0Poc\\u751f\\u6210\\u7684\\u6587\\u4ef6/dev/shm/QT\_Test\_1558287004\"}]}, \"error\": \"\"}, \"host\_id\": \"24b7d45d8095b846\", \"ver\_check\_result\": {\"vuln\": 1, \"code\": 0, \"result\": {\"vuls\": [{\"code\": 1, \"version\": \"3.2-24.el5\", \"fix\_version\": \"3.2-33.el5\_11.4\", \"name\": \"bash\"}]}, \"error\": \"\"}}",  
 "remedDescription": "升级bash到最新版本，漏洞修复后需要重启bash服务，建议业务不繁忙时修复。使用以下命令升级： \r\nRHEL/CentOS : sudo yum update -y bash \r\nUbuntu : sudo apt-get update && sudo apt-get install --only-upgrade -y bash",  
 "kernel": false,  
 "remote": true,  
 "localEscalation": false,  
 "hasExp": true,  
 "hasPoc": false,  
 "cvss": "AV:N/AC:L/Au:N/C:C/I:C/A:C",  
 "cvssScore": 10,  
 "refs": "CNVD-2014-06435,BID-70137",  
 "expRefs": "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169",  
 "pocRefs": "https://www.exploit-db.com/exploits/34879/\r\nhttps://www.exploit-db.com/exploits/36503/\r\nhttps://www.exploit-db.com/exploits/36504/\r\nhttps://www.exploit-db.com/exploits/35115/\r\nhttps://www.exploit-db.com/exploits/34766/\r\nhttps://www.exploit-db.com/exploits/36609/\r\nhttps://www.exploit-db.com/exploits/35146/\r\nhttps://www.exploit-db.com/exploits/34765/",  
 "cves": [  
 "CVE-2014-7169"  
 ],  
 "category": [  
 8  
 ],  
 "condition": "",  
 "appVersion": "bash＜4.3\r\n (不同操作系统影响的应用版本不同，具体以检测结果为准)",  
 "pocCheckResults": [  
 {  
 "executeType": 1,  
 "checkResult": "- bash\r\n 当前安装版本：3.2-24.el5\r\n 漏洞修复版本：3.2-33.el5\_11.4\r\n该主机存在此漏洞"  
 },  
 {  
 "executeType": 2,  
 "checkResult": "当前主机存在漏洞，在/dev/shm目录中发现Poc生成的文件/dev/shm/QT\_Test\_1558287004"  
 }  
 ]  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | string | varchar(14) | 风险id（仅linux） |
| vulName | string | varchar(512) | 风险名（仅linux） |
| apps | List | varchar(1024) | 影响的应用（仅linux） |
| desc | String | varchar(4096) | 风险描述 |
| restartOpts | Integer | tinyint(4) | 修复影响：0-未知；1-无需重启；2-服务重启；3系统重启 （仅linux） |
| publicDate | Date | date | 发布时间（仅linux） |
| firstCheckTime | String | date | 首次检测时间 |
| family | Integer | tinyint(4) | 风险类型，对于漏洞检测固定为4 |
| severity | Integer | tinyint(4) | 危险程度：0-信息；1-低危；2-中危；3-高危；4-危急 ） |
| checkInfo | String | text | 验证信息 |
| pocCheckInfo | String | text | POC检测信息 |
| checkResult | String | text | 验证结果 |
| data | String | text | 详情，见[附录3](#附录3-linux漏洞检测-查询结果-data数据结构) |
| remedDescription | string | text | 修复建议 |
| hasExp | Boolean | tinyint(4) | 是否存在exp |
| kernel | Boolean | tinyint(4) | 是否内核漏洞（仅linux） |
| localEscalation | Boolean | tinyint(4) | 是否本地提权（仅linux） |
| remote | Boolean | tinyint(4) | 是否远程执行（仅linux） |
| hasPoc | Boolean | tinyint(4) | 存在poc（仅linux） |
| cvssScore | String | float | Cvss分 |
| cvss | String | varchar(64) | Cvss详情 |
| refs | String | mediumblob | Cve引用 |
| expRefs | String | varchar(1024) | 参考链接 |
| pocRefs | String | varchar(1024) | Poc参考链接（仅linux） |
| cves | List | varchar(512) | CVE编号 |
| category | Integer数组 | Integer数组 | 漏洞类型:0 SQL注入,1 未授权访问,2 敏感信息泄露,3 XML外部实体注入,4 跨站脚本攻击,5 不安全的反序列,6 客户端请求伪造,7 服务端请求伪造,8 命令执行,9 代码执行,10 任意文件上传,11 任意文件读取,12 拒绝服务攻击,13 目录遍历,14 恶意后门,15 本地提权,16 注入漏洞 |
| condition | String | tinytext | 利用条件 |
| appVersion | Integer | varchar(255) | 受影响应用版本 |
| pocCheckResults | List |  | POC检测结果 |

**返回部分说明：**rows中pocCheckResults

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| executeType | Integer | tinyint(2) | 检测方式，1：版本比对 2：POC检测 |
| checkResult | String | varchar(1024) | 检测结果描述 |

### 查询所有可用检测项

该接口提供Linux/Windows漏洞作业检测项查询。

**调用接口：**

GET /external/api/vul/poc/job/rule\_list

GET /external/api/vul/poc/job/win/rule\_list

**请求参数：**

无

**请求示例：**

/external/api/vul/poc/job/rule\_list

**返回示例：**

{  
 "rows": [  
 {  
 "vulId": "QT042008001658",  
 "title": "Struts2 S2-002 跨站脚本漏洞",  
 "description": "由于 Struts 2 在处理<s：url>和<s：a>标签时，对于用于构造<s：a>标记输出的参数值，双引号会被转义，对于用于构造<s：a>和<s：url>标记的输出的参数值，<script>标记会被递归地转义，导致存在 XSS 注入漏洞。",  
 "category": [  
 4  
 ],  
 "remedDescription": "请到官网下载升级到Struts 2.0.12以上版本修复该漏洞，漏洞修复后需要服务重启，建议业务不繁忙时修复。\r\n下载更新：https://struts.apache.org/download",  
 "permitExecuteTypes": [  
 1  
 ],  
 "executeSeverity": 0,  
 "executeSeverityTip": "",  
 "updateTime": 1540971348,  
 "inputSchemas": []  
 },  
 {  
 "vulId": "QT042012001685",  
 "title": "Mysql\_Mariadb 认证绕过漏洞(CVE-2012-2122)",  
 "description": "用户连接到MariaDB/MySQL后，应用会计算和比较令牌值，由于错误的转换，即使memcmp()返回非零值，也可能出现错误的比较，造成MySQL/MariaDB误认为密码是正确的，因为协议使用的是随机字符串，该Bug发生的几率为1/256。MySQL的版本是否受影响取决于程序的编译方式，很多版本（包括官方提供的二进制文件）并不受此漏洞的影响。",  
 "category": [  
 1  
 ],  
 "remedDescription": "升级MySQL到最新版本修复该漏洞，攻击者可利用该漏洞绕过某些安全限制，也可能导致攻击者无需知道正确口令就能登录到MySQL服务器。漏洞修复后需要服务重启。建议业务不繁忙时修复。使用以下命令升级：\r\nRHEL : sudo zypper update -y mysql\r\nCentOS : sudo yum update -y mysql\r\nUbuntu : sudo apt-get update && sudo apt-get install mysql",  
 "permitExecuteTypes": [  
 1  
 ],  
 "executeSeverity": 0,  
 "executeSeverityTip": "",  
 "updateTime": 1540971384,  
 "inputSchemas": []  
 }  
 {......}  
 ],  
 "total": 104,  
 "charts": {}  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | String | varchar(14) | 风险ID |
| title | String | varchar(512) | 风险名 |
| description | text | text | 风险描述 |
| category | String | varchar(15) | 类别 |
| remedDescription | text | text | 修复建议 |
| permitExecuteTypes | Integer | bigint(20) | 允许的检测方式，1：版本对比 2：POC检测 |
| executeSeverity | String | varchar(128) | 检测脚本执行风险，0：无风险，1：低风险，2：中风险，3：高风险 |
| executeSeverityTip | String | varchar(1024) | 执行风险提示 |
| updateTime | Integer | int(10) | 更新时间 |
| inputSchemas | List |  | 输入参数定义列表 |

**返回部分说明：**rows中inputSchemas

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | Integer | int(10) | 参数编号 |
| vulId | String | varchar(14) | 风险ID |
| field | String | varchar(64) | 参数名 |
| displayName | String | varchar(255) | 参数中文名 |
| caption | String | varchar(1024) | 参数说明 |
| required | Boolean | tinyint(1) | 是否必填 |
| defaultValue | String | varchar(1024) | 参数默认值 |

### 新建漏洞检测作业请求

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/add

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| name | string | varchar(255) | 作业名称,必传 |
| cronEnabled | Boolean | Boolean | 是否定时, 不传时为不定时 |
| cron | string | varchar(128) | cronEnabled=true时必填，cron任务表达式 |
| status | Boolean | Boolean | 是否启用,必传 |
| realm | Object | 执行范围 | 执行范围,必传 |
| realmName | string | varchar(128) | 执行范围名称，可选 |
| remark | string | varchar(256) | 备注,可选 |
| pocTaskCreateParams | List | List | 检测项列表,必传 |

**参数说明：**realm

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| type | Integer | tinyint(1) | 范围，0:全部主机,type为0时，agents和groups不传; 1:主机范围，如果type=1, 则agents必传; 2:业务组范围，如果type=2,则groups必传 不传时为0全部主机 |
| groups | Integer数组 | List | Linux业务组ID |
| agents | String数组 | List | Linux的主机agentId |

**参数说明：**pocTaskCreateParams

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | string | varchar(14) | 风险检查项Id，见4.5.6接口 ,必传 |
| executeTypes | List | Integer数组 | 检测方式,必传 |
| params | Map<String, Object> | Map | 输入参数 ,可选 |

**请求示例：**

POST /external/api/vul/poc/job/linux/add  
Content-Type: application/json  
  
{  
 "name": "liangyong\_test",  
 "cronEnabled": false,  
 "status": true,  
 "realm": {  
 "agents": [  
 "ff0878fac4add80c"  
 ],  
 "type": 1  
 },  
 "realmName": "全部主机222",  
 "remark": "ceshi",  
 "pocTaskCreateParams": [  
 {  
 "vulId": "QT042016002534",  
 "executeTypes": [  
 1  
 ],  
 "params": {}  
 }  
 ]  
}

**返回示例：**

{  
 "id": "5c417ad07d761b3224c4280a",  
 "comId": "59080851823593e1a80b",  
 "uuid": "59080851823593e1a80b",  
 "name": "liangyong\_test1",  
 "status": true,  
 "remark": "ceshi",  
 "createTime": "2019-01-18 15:05:52",  
 "updateTime": null,  
 "cronEnabled": null,  
 "cron": null,  
 "cronTaskId": null,  
 "realm": {  
 "type": 1,  
 "groups": null,  
 "agents": [  
 "ff0878fac4add80c"  
 ],  
 "osTags": null  
 },  
 "realmName": "全部主机222",  
 "lastExecuteTime": null  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(32) | 作业id |
| comId | string | varchar(32) | 公司id |
| uuid | string | varchar(32) | uuid |
| name | string | varchar(255) | 作业名称 |
| status | Boolean | Boolean | 是否启用 |
| createTime | Date | Date | 创建时间 |
| updateTime | Date | Date | 更新时间 |
| cronEnabled | Boolean | Boolean | 是否定时 |
| cron | string | varchar(128) | cron任务表达式 |
| cronTaskId | string | varchar(128) | cronTaskId |
| remark | string | varchar(256) | 备注 |
| realm | Object | 执行范围 | 执行范围 |
| realmName | string | varchar(128) | 执行范围名称 |
| lastExecuteTime | Date | Date | 最后执行时间 |

### 编辑漏洞检测作业请求

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/fix

与新增作业相比，jobId必传。

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| jobId | string | varchar(32) | 作业id，必传 |
| name | string | varchar(255) | 作业名称，必传 |
| cronEnabled | Boolean | Boolean | 是否定时, 不传时为不定时 |
| cron | string | varchar(128) | cronEnabled=true时必填，cron任务表达式 |
| status | Boolean | Boolean | 是否启用，必传 |
| realm | Object | 执行范围 | 执行范围，必传 |
| realmName | string | varchar(128) | 执行范围名称，可选 |
| remark | string | varchar(256) | 备注，可选 |
| pocTaskCreateParams | List | List | 检测项列表，必传 |

**参数说明：**realm

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| type | Integer | tinyint(1) | 范围，0:全部主机,type为0时，agents和groups不传; 1:主机范围，如果type=1, 则agents必传; 2:业务组范围，如果type=2,则groups必传 不传时为0全部主机 |
| groups | Integer数组 | List | Linux业务组ID |
| agents | String数组 | List | Linux的主机agentId |

**参数说明：**pocTaskCreateParams

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | string | varchar(14) | 风险检查项Id，见4.5.6接口，必传 |
| executeTypes | List | Integer数组 | 检测方式，必传 |
| params | Map<String, Object> | Map | 输入参数，可选 |

**请求示例：**

POST /external/api/vul/poc/job/linux/fix  
Content-Type: application/json  
  
{  
 "name": "liangyong\_test",  
 "cronEnabled": false,  
 "status": true,  
 "realm": {  
 "agents": [  
 "ff0878fac4add80c"  
 ],  
 "type": 1  
 },  
 "realmName": "全部主机222",  
 "remark": "ceshi",  
 "pocTaskCreateParams": [  
 {  
 "vulId": "QT042016002534",  
 "executeTypes": [  
 1  
 ],  
 "params": {}  
 }  
 ]  
}

**返回示例：**

{  
 "id": "5c417ad07d761b3224c4280a",  
 "comId": "59080851823593e1a80b",  
 "uuid": "59080851823593e1a80b",  
 "name": "liangyong\_test1",  
 "status": true,  
 "remark": "ceshi",  
 "createTime": "2019-01-18 15:05:52",  
 "updateTime": null,  
 "cronEnabled": null,  
 "cron": null,  
 "cronTaskId": null,  
 "realm": {  
 "type": 1,  
 "groups": null,  
 "agents": [  
 "ff0878fac4add80c"  
 ],  
 "osTags": null  
 },  
 "realmName": "全部主机222",  
 "lastExecuteTime": null  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(32) | 作业id |
| comId | string | varchar(32) | 公司id |
| uuid | string | varchar(32) | uuid |
| name | string | varchar(255) | 作业名称 |
| status | Boolean | Boolean | 是否启用 |
| createTime | Date | Date | 创建时间 |
| updateTime | Date | Date | 更新时间 |
| cronEnabled | Boolean | Boolean | 是否定时 |
| cron | string | varchar(128) | cron任务表达式 |
| cronTaskId | string | varchar(128) | cronTaskId |
| remark | string | varchar(256) | 备注 |
| realm | Object | 执行范围 | 执行范围 |
| realmName | string | varchar(128) | 执行范围名称 |
| lastExecuteTime | Date | Date | 最后执行时间 |

### 漏洞检测作业删除

**调用接口：**

DELETE /external/api/vul/poc/job/{linux|win}/{{jobId}}

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| jobId | string | varchar(32) | 作业id |

**请求示例：**

DELETE /external/api/vul/poc/job/linux/5c3fe2c77d761b1e0a8b5c2a

**返回示例：**

{  
 "flag": true  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | Boolean | 是否删除成功 |

### 开始作业执行

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/execute

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| jobId | string | varchar(32) | 作业id，必传 |
| jobType | Integer | Integer | 作业类型：1：全局作业 2：用户作业。必传 |

**请求示例：**

POST /external/api/vul/poc/job/linux/execute  
{"jobId":"5c3d7c7a7d761b6b1c9048fd","jobType":2}

**返回示例：**

{  
 "flag": true  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | Boolean | 是否开始执行成功 |

### 查询漏洞检测作业执行状态

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/status

**请求参数：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| jobId | string | varchar(32) | 作业id，必传 |
| jobType | Integer | Integer | 作业类型：1：全局作业 2：用户作业。必传 |

**请求示例：**

POST /external/api/vul/poc/job/linux/status  
{"jobId":"5c3d7c7a7d761b6b1c9048fd","jobType":2}

**返回示例：**

{  
 "id": "5c4182e87d761b3224c4293e",  
 "status": "Running"  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | String | Varchar(32) | 扫描任务ID |
| status | String | Varchar(7) | 扫描的状态（进行中/成功/失败） |

### 查询作业执行失败主机

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/error\_host

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 作业id，必传 |
| jobType | Integer | 作业类型：1：全局作业 2：用户作业。必传 |
| size | Integer | 每页数量：默认为50，可选 |
| page | Integer | 当前页，从0开始，默认为0，可选 |

**请求示例：**

POST /external/api/vul/poc/job/linux/error\_host  
{"jobId":"5bcef3c67d761b31e34ec895","jobType":"2","size":2,"page":3}

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "27b1a655e425a833",  
 "vulId": "QT042018000035",  
 "title": "Struts2 S2-056拒绝服务漏洞(CVE-2018-1327)",  
 "displayIp": "192.168.77.199",  
 "internalIp": "192.168.77.199",  
 "externalIp": null,  
 "connectionIp": "58.49.121.58",  
 "hostname": "www.mycluster.com",  
 "group": 866,  
 "remark": null,  
 "hostTagList": [],  
 "code": 1,  
 "description": "主机不在线"  
 },  
 {  
 "agentId": "27b1a655e425a833",  
 "vulId": "QT042017002041",  
 "title": "Struts2 S2-053远程代码执行漏洞(CVE-2017-12611)",  
 "displayIp": "192.168.77.199",  
 "internalIp": "192.168.77.199",  
 "externalIp": null,  
 "connectionIp": "58.49.121.58",  
 "hostname": "www.mycluster.com",  
 "group": 866,  
 "remark": null,  
 "hostTagList": [],  
 "code": 1,  
 "description": "主机不在线"  
 },  
 {  
 "agentId": "27b1a655e425a833",  
 "vulId": "QT042017002040",  
 "title": "Struts2 S2-052远程代码执行漏洞(CVE-2017-9805)",  
 "displayIp": "192.168.77.199",  
 "internalIp": "192.168.77.199",  
 "externalIp": null,  
 "connectionIp": "58.49.121.58",  
 "hostname": "www.mycluster.com",  
 "group": 866,  
 "remark": null,  
 "hostTagList": [],  
 "code": 1,  
 "description": "主机不在线"  
 },  
 {  
 "agentId": "27b1a655e425a833",  
 "vulId": "QT042013001856",  
 "title": "Struts2 S2-016远程命令执行漏洞(CVE-2013-2251)",  
 "displayIp": "192.168.77.199",  
 "internalIp": "192.168.77.199",  
 "externalIp": null,  
 "connectionIp": "58.49.121.58",  
 "hostname": "www.mycluster.com",  
 "group": 866,  
 "remark": null,  
 "hostTagList": [],  
 "code": 1,  
 "description": "主机不在线"  
 }  
 ],  
 "total": 11,  
 "charts": {}  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| agentId | String | agentID |
| vulId | String | 风险Id |
| title | String | 漏洞检测项名称 |
| displayIp | String | 主机IP |
| internalIp | String | 内网IP |
| externalIp | String | 外网IP |
| hostname | String | 主机名 |
| group | int | 业务组 |
| remark | String | 备注 |
| hostTagList | List | 标签 |
| code | int | 失败码 |
| description | String | 描述 |

### 用户所有漏洞作业查询

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/list

jobType 必传，因全局作业与用户作业分属不同库，不传无法分页处理。另外全局作业暂不支持分页查询。

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | jobid，可选 |
| jobName | String | jobName，可选 |
| jobType | Integer | 作业类型：1：全局作业 2：用户作业，必传 |
| size | Integer | 每页数量：默认为50，可选 |
| page | Integer | 当前页，从0开始，默认为0，可选 |

**请求示例：**

POST /external/api/vul/poc/job/linux/list  
Content-Type: application/json  
  
{"jobType":2,"page":0,"size":30}

**返回示例：**

{  
 "rows": [  
 {  
 "jobId": "5c3d7c7a7d761b6b1c9048fd",  
 "jobType": 2,  
 "name": "ceshi",  
 "status": true,  
 "realmName": "全部主机",  
 "lastExecuteTime": 1547797225,  
 "taskCount": 1,  
 "createTime": 1547533434,  
 "cronEnabled": null,  
 "cron": null,  
 "executeStatus": 1  
 }  
 ],  
 "total": 1,  
 "charts": {}  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| jobId | string | varchar(32) | 作业id |
| jobType | string | varchar(32) | 作业类型 |
| name | string | varchar(255) | 作业名称 |
| status | Boolean | Boolean | 是否启用 |
| realmName | string | varchar(128) | 执行范围名称 |
| lastExecuteTime | Date | Date | 最后执行时间 |
| taskCount | Integer | Integer | 作业下检测项数量（极少数情况下自定义作业可能包含已删除的无效检测项，真实数量以接口4.5.14为准 ） |
| createTime | Date | Date | 创建时间 |
| cronEnabled | Boolean | Boolean | 是否定时 |
| cron | string | varchar(128) | cron任务表达式 |
| executeStatus | Integer | varchar(128) | 执行状态; 0：未开始执行，1：排队执行，2：正在执行，3：执行成功，4：执行失败 |

### 作业下检测项信息查询

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/tasks

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 必传 |
| jobType | Integer | 必传，作业类型：1：全局作业 2：用户作业 |

**请求示例：**

POST /external/api/vul/poc/job/linux/tasks  
  
{"jobId":"4d6d38c4b39831f3c74e","jobType":1}

**返回示例：**

{  
 "rows": [  
 {  
 "id": "1756c4dc8f66365e3cb6",  
 "vulId": "QT042018000053",  
 "params": {},  
 "executeTypes": [  
 1  
 ],  
 "permitExecuteTypes": [  
 1  
 ],  
 "title": "timlong's test notice22",  
 "category": [  
 0,  
 3  
 ],  
 "executeSeverityTip": "timlong's poc脚本风险描述",  
 "executeSeverity": 2,  
 "updateTime": 1545184060,  
 "inputSchemas": [  
 {  
 "field": "qq",  
 "displayName": "请求",  
 "caption": "wqw",  
 "required": true,  
 "defaultValue": ""  
 }  
 ]  
 }  
 ],  
 "total": 1,  
 "charts": {}  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(32) | task id |
| vulId | string | varchar(32) | 风险id |
| params | Map<String,Object> | varchar(32) | 输入参数 |
| executeTypes | Integer数组 | Integer数组 | 检测方式，1：版本对比 2：POC检测 |
| permitExecuteTypes | Integer数组 | BoInteger数组olean | 允许的检测方式，1：版本对比 2：POC检测 |
| title | string | varchar(512) | 检测项名称 |
| category | Integer数组 | Integer数组 | 检测项类别 |
| executeSeverityTip | text | varchar(1024) | 检测脚本执行风险描述 |
| executeSeverity | Integer | Integer | 检测脚本执行风险，0：无风险，1：低风险，2：中风险，3：高风险 |
| updateTime | Date | Date | 更新时间 |
| inputSchemas | List | List | 输入参数定义 |

* 返回rows中inputSchemas部分说明

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| field | String | varchar(64) | 参数名 |
| displayName | String | varchar(255) | 参数中文名 |
| caption | String | varchar(1024) | 参数说明 |
| required | Boolean | tinyint(1) | 是否必填 |
| defaultValue | String | varchar(1024) | 参数默认值 |

### 作业执行基本信息查询

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/stats

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 必传 |
| jobType | Integer | 必传，作业类型：1：全局作业 2：用户作业 |

**请求示例：**

POST /external/api/vul/poc/job/linux/stats  
  
{"jobId":"5b8dfe107d761b471404112d","jobType":2}

**返回示例：**

{  
 "jobId": "5b8dfe107d761b471404112d",  
 "jobName": "testt",  
 "createTime": "2018-09-04 11:37:52",  
 "executeJobId": "5b9b327c1522e85a802faa34",  
 "executeStatus": 3,  
 "lastExecuteTime": "2018-09-14 12:01:00",  
 "duration": 0,  
 "vulCount": 0,  
 "hostCount": 0,  
 "taskCount": 1,  
 "successHostCount": 0,  
 "failedHostCount": 0  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 作业id |
| jobName | String | 作业名称 |
| createTime | date | 创建时间 |
| executeJobId | String | 执行jobId |
| executeStatus | String | 执行状态; 0：未开始执行，1：排队执行，2：正在执行，3：执行成功，4：执行失败 |
| lastExecuteTime | date | 最后执行时间 |
| duration | Long | 执行耗时秒数 |
| vulCount | long | 漏洞数量 |
| hostCount | long | 主机数 |
| taskCount | long | 检测项数 |
| failedHostCount | long | 失败主机数 |
| successHostCount | long | 成功主机数 |

### 作业执行漏洞结果查询

**调用接口：**

POST /external/api/vul/poc/job/{linux|win}/result\_detail

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 必传 |
| jobType | Integer | 必传，作业类型：1：全局作业 2：用户作业 |

**请求示例：**

POST /external/api/vul/poc/job/linux/result\_detail  
  
{"jobId":"5b8dfe107d761b471404112d","jobType":2}

**返回示例：**

{  
 "rows": [  
 {  
 "vulId": "QT042014001964",  
 "vulName": "Bash远程代码执行漏洞(CVE-2014-6271)",  
 "apps": [  
 "bash"  
 ],  
 "desc": "Shellshock，又称Bashdoor，是在Unix中广泛使用的Bash shell中的一个安全漏洞，首次于2014年9月24日公开。许多互联网守护进程，如网页服务器，使用bash来处理某些命令，从而允许攻击者在易受攻击的Bash版本上执行任意代码。这可使攻击者在未授权的情况下访问计算机系统。该漏洞会影响目前主流的Linux和Mac OSX操作系统平台，包括但不限于Redhat、CentOS、Ubuntu、Debian、Fedora、Amazon Linux、OS X 10.10等平台。",  
 "restartOpts": 3,  
 "publicDate": "2014-09-24 00:00:00",  
 "firstCheckTime": "2018-08-02 15:30:31",  
 "family": 4,  
 "severity": 4,  
 "checkInfo": "版本比对检测原理：检查当前系统中Bash版本是否在受影响版本内|版本比对检测结果：|POC检测原理：检查当前系统中Bash版本是否在受影响版本内|POC检测结果：",  
 "pocCheckInfo": "在Bash中，以\"(){\"开头定义的环境变量在命令env中解析成函数后，Bash执行并未退出，而是继续解析并执行shell命令，从而导致漏洞产生。",  
 "checkResult": null,  
 "data": "{\"code\": 0, \"execute\_type\": 3, \"cve\_id\": \"CVE-2014-0160\", \"poc\_check\_result\": {\"vuln\": 1, \"code\": 0, \"result\": {\"vuls\": [{\"vuln\": 1, \"key\": \"fake value\"}]}, \"error\": null}, \"host\_id\": \"86e354eb10f147f2\", \"ver\_check\_result\": {\"vuln\": 1, \"code\": 0, \"result\": {\"vuls\": [{\"vuln\": 1, \"version\": \"1.0.1e-fips\", \"home\": \"/etc/pki/tls\"}]}, \"error\": null}}",  
 "remedDescription": "升级bash到最新版本，漏洞修复后需要重启bash服务，建议业务不繁忙时修复。使用以下命令升级： \r\nRHEL/CentOS : sudo yum update -y bash \r\nUbuntu : sudo apt-get update && sudo apt-get install --only-upgrade -y bash",  
 "kernel": false,  
 "remote": false,  
 "localEscalation": false,  
 "hasExp": false,  
 "hasPoc": false,  
 "cvss": "AV:N/AC:L/Au:N/C:C/I:C/A:C",  
 "cvssScore": 10,  
 "refs": "CNVD-2014-06345,BID-70103",  
 "expRefs": "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271",  
 "pocRefs": "https://www.exploit-db.com/exploits/34879/ \r\nhttps://www.exploit-db.com/exploits/37816/ \r\nhttps://www.exploit-db.com/exploits/38849/ \r\nhttps://www.exploit-db.com/exploits/39918/ \r\nhttps://www.exploit-db.com/exploits/40619/ \r\nhttps://www.exploit-db.com/exploits/40938/ \r\nhttps://www.exploit-db.com/exploits/42938/\r\nhttps://www.exploit-db.com/exploits/36609/\r\nhttps://www.exploit-db.com/exploits/36503/\r\nhttps://www.exploit-db.com/exploits/36504/\r\nhttps://www.exploit-db.com/exploits/35146/\r\nhttps://www.exploit-db.com/exploits/35115/\r\nhttps://www.exploit-db.com/exploits/34766/\r\nhttps://www.exploit-db.com/exploits/34765/",  
 "cves": [  
 "CVE-2014-6271"  
 ],  
 "category": [  
 2  
 ],  
 "condition": "",  
 "appVersion": "bash＜4.3",  
 "pocCheckResults": [  
 {  
 "executeType": 1,  
 "checkResult": ""  
 },  
 {  
 "executeType": 2,  
 "checkResult": ""  
 }  
 ],  
 "agentId": "d27260c9bb52d82c",  
 "displayIp": "172.16.2.184",  
 "connectionIp": "172.16.2.184",  
 "externalIp": null,  
 "internalIp": "172.16.2.184",  
 "bizGroupId": 1,  
 "bizGroup": null,  
 "remark": null,  
 "hostTagList": null,  
 "hostname": "yongliang"  
 }  
 ],  
 "total": 1,  
 "charts": {}  
}

**返回rows部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vulId | string | varchar(14) | 风险id（仅linux） |
| vulName | string | varchar(512) | 风险名（仅linux） |
| apps | List | varchar(1024) | 影响的应用（仅linux） |
| desc | String | varchar(4096) | 风险描述 |
| restartOpts | Integer | tinyint(4) | 修复影响：0-未知；1-无需重启；2-服务重启；3系统重启 （仅linux） |
| publicDate | Date | date | 发布时间（仅linux） |
| firstCheckTime | String | date | 首次检测时间 |
| family | Integer | tinyint(4) | 风险类型，对于漏洞检测固定为4 |
| severity | Integer | tinyint(4) | 危险程度：0-信息；1-低危；2-中危；3-高危；4-危急 ） |
| checkInfo | String | text | 验证信息 |
| pocCheckInfo | String | text | POC检测信息 |
| checkResult | String | text | 验证结果 |
| data | String | text | 详情，见[附录2](#附录2-windows后门检测-查询结果-data数据结构) |
| remedDescription | string | text | 修复建议 |
| hasExp | Boolean | tinyint(4) | 是否存在exp |
| kernel | Boolean | tinyint(4) | 是否内核漏洞（仅linux） |
| localEscalation | Boolean | tinyint(4) | 是否本地提权（仅linux） |
| remote | Boolean | tinyint(4) | 是否远程执行（仅linux） |
| hasPoc | Boolean | tinyint(4) | 存在poc（仅linux） |
| cvssScore | String | float | Cvss分 |
| cvss | String | varchar(64) | Cvss详情 |
| refs | String | mediumblob | Cve引用 |
| expRefs | String | varchar(1024) | 参考连接 |
| cves | List | varchar(512) | CVE编号 |
| category | Integer数组 | Integer数组 | 漏洞类型:0 SQL注入,1 未授权访问,2 敏感信息泄露,3 XML外部实体注入,4 跨站脚本攻击,5 不安全的反序列,6 客户端请求伪造,7 服务端请求伪造,8 命令执行,9 代码执行,10 任意文件上传,11 任意文件读取,12 拒绝服务攻击,13 目录遍历,14 恶意后门,15 本地提权,16 注入漏洞 |
| condition | String | tinytext | 利用条件 |
| appVersion | Integer | varchar(255) | 受影响应用版本 |
| pocCheckResults | Object |  | POC检测结果 |
| agentId | String | varchar(16) | agentID |
| displayIp | String | varchar(15) | 主机IP |
| internalIp | String | varchar(15) | 内网IP |
| externalIp | String | varchar(15) | 外网IP |
| hostname | String | varchar(128) | 主机名 |
| bizGroupId | int | bigint(20) | 业务组Id |
| bizGroup | String | varchar(128) | 业务组名称 |
| remark | String | varchar(256) | 备注 |
| hostTagList | List |  | 标签 |
| code | int |  | 失败码 |
| description | String | varchar(1024) | 描述 |

## linux全部风险扫描

### 开始linux全部风险扫描(除风险文件外)

该接口提供开始linux全部风险扫描的功能

**调用接口：**

POST /external/api/vul/linux/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回**：

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### linux全部风险扫描状态查询(除风险文件外)

该接口提供linux全部风险扫描状态查询的功能

**调用接口：**

GET /external/api/vul/linux/check/status

**请求参数：**

无

**返回示例：**

{  
 "id":""//当前job的jobId, 如果为null, 则当前没有job执行  
 "status":"Running|Success|Failed"  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID，如果为null，则当前没有job执行 |
| status | string | varchar(7) | 执行中；成功；失败 |

### windows补丁扫描增加范围筛选

提供windows补丁扫描指定扫描主机范围的功能

**调用接口：**

POST /external/api/vul/patch/win/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回：**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

### window弱密码扫描增加扫描范围筛选

提供windows弱密码扫描指定扫描主机范围的功能

**调用接口：**

POST /external/api/vul/weakpwd/win/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | tinyint(4) | 是 | 0/1/2 全部主机/agent范围/业务组范围 |
| agentIds | string数组 | 如果type为1 必填 | 主机agentId数组 |
| group | int数组 | 如果type为2 必填 | 业务组id范围 |

**请求示例：**

{  
 "type":1, // 0 全部主机 1 agent 2 group  
 "group":[],  
 "agentIds":["94e4c7a4f5dec750"]  
}

**返回示例：**

{  
 "id": "some job id"  
}

**如果无在线agent返回：**

{  
 "errorCode": 500,  
 "errorMessage": "no online agent", // 错误描述  
 "detail":null  
}

**如果有作业在运行返回：**

{  
 "errorCode": 400,  
 "errorMessage": "已有任务正在执行",  
 "detail": null  
}

**返回部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| id | string | varchar(24) | 扫描任务ID |

# 入侵检测

## 可疑操作

### 查询结果

该接口提供Linux可疑操作上报事件的数据查询。

**调用接口：**

GET /external/api/detect/shelllog/linux

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| logTime | DateRange | 否 | 上报时间，格式参考[通用Get请求参数类型及说明](#通用get请求参数类型及说明) |
| auditStatus | int数组 | 否 | 审核状态：1-未审核；2-审核通过；3-审核未通过 |
| status | int数组 | 否 | 可疑操作类型，见[可疑操作类型](#可疑操作类型) |
| cmd | String | 否 | 命令内容 |
| loginIp | String | 否 | 登录主机IP |
| loginUser | String | 否 | 登录用户 |
| hostname | String | 否 | 主机名 |
| ip | String | 否 | 操作主机IP |

**可疑操作类型字典表：**

| 值 | 说明 |
| --- | --- |
| 0 | 未命中规则 |
| 1 | bash危险命令执行 |
| 2 | Wget下载黑客工具 |
| 3 | curl下载黑客工具 |
| 4 | rcp下载黑客工具 |
| 5 | scp下载黑客工具 |
| 6 | rsync下载黑客工具 |
| 7 | MYSQL明文密码显示 |
| 8 | Mongo明文密码显示 |
| 9 | scp外部下载或上传 |
| 10 | rcp外部下载或上传 |
| 11 | rsync外部下载或上传 |
| 12 | 赋给目录或文件危险权限 |

**请求示例：**

/external/api/detect/shelllog/linux?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "b470f51011e3b7a9",  
 "displayIp": "192.168.100.26",  
 "connectionIp": "192.168.100.26",  
 "externalIp": null,  
 "internalIp": "192.168.100.26",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": null,  
 "hostTagList": [],  
 "hostname": "localhost.localdomain",  
 "logTime": 1521686548,  
 "cmd": "cd /titan/agent",  
 "hitRuleName": [  
 "a"  
 ],  
 "status": [  
 1  
 ],  
 "loginUser": "root",  
 "loginIp": "192.168.199.109",  
 "auditStatus": 3  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| logTime | Integer | int(10) | 操作时间，时间戳，精确到秒 |
| cmd | String | varchar(1024) | 操作内容（带tag） |
| cmdNoHightLight | String | varchar(1024) | 操作内容（不带tag） |
| hitRuleName | List | varchar(50) | 命中规则 |
| status | List | tinyint(2) | 可疑操作类型，见[可疑操作类型](#可疑操作类型) |
| loginUser | String | varchar(50) | 登录用户 |
| loginIp | String | varchar(15) | 登录IP |
| auditStatus | List | tinyint(2) | 审核状态，1:未审核 2:审核通过 3:审核未通过 |

**代码示例：**

#入侵检测-可疑操作-查询结果接口调用的代码示例(python)  
def shelllog\_list():  
 url = "http://%s:%s/external/api/detect/shelllog/linux?page=0&size=50&groups=1,2" % (host, port)  
 data = {'page': 0, 'size': 50, 'groups': '1,2'}  
 res = send\_request("GET", url, data)  
 print res

## 暴力破解

### 查询结果

**调用接口：**

GET /external/api/detect/brutecrack/{linux,win}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| loginTime | DateRange | 否 | 攻击时间，格式参考[通用Get请求参数类型及说明](#通用get请求参数类型及说明) |
| serviceType | int数组 | 否 | 服务类型， Linux: 1:SSHD 2:VSFTPD Windows: 100:RDP 101:WINRM 102:SSH |
| block | int数组 | 否 | 封停状态，见[暴力破解封停状态](#暴力破解封停状态) |
| clientIp | String | 否 | 攻击来源 |
| ip | String | 否 | 攻击目标（模糊查询） |
| hostname | String | 否 | 主机名（模糊查询） |

封停状态字典表：

| 值 | 说明 |
| --- | --- |
| 0 | 未处理 |
| 1 | 自动封停 |
| 2 | 自动解封 |
| 3 | 手动封停 |
| 4 | 手动解封 |
| 5 | 解封中 |
| 6 | 封停中 |
| -1 | 封停失败 |
| -2 | 解封失败 |

**请求示例：**

/external/api/detect/brutecrack/linux?page=0&size=50&group=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "979f3c4d44b2e765",  
 "displayIp": "172.16.2.122",  
 "connectionIp": "172.16.2.122",  
 "externalIp": null,  
 "internalIp": "172.16.2.122",  
 "bizGroupId": 37,  
 "bizGroup": "test",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "tianxianhu",  
 "id": "tFU7aUo3o5w7GDPlJqQfMA==",  
 "serviceType": 1,  
 "loginTime": 1523412532,  
 "clientIp": "106.75.67.64",  
 "block": -1,  
 "serverName": "sshd"  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | String | varchar(50) | 记录ID |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| hostname | String | varchar(128) | 主机名 |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| loginTime | Integer | int(10) | 攻击时间，时间戳，精确到秒 |
| serviceType | Integer | tinyint(1) | 服务类型， Linux: 1:SSHD 2:VSFTPD Windows: 100:RDP 101:WINRM 102:SSH |
| clientIp | String | varchar(15) | 攻击来源IP |
| block | Integer | tinyint(1) | 封停状态，见[封停状态](#暴力破解封停状态) |
| serverName | String | varchar(12) | 服务类型名 |

### 查询记录

该接口提供Linux/Windows暴力破解上报事件记录的数据查询。

**调用接口：**

GET /external/api/detect/brutecrack/{linux,win}/log

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| crackId | String | 是 | 攻击ID（查询结果接口返回的id字段） |

**请求示例：**

/external/api/detect/brutecrack/linux/log?page=0&size=50&crackId=tFU7aUo3o5w7GDPlJqQfMA==

**返回示例：**

{  
 "rows": [  
 {  
 "loginTime": 1531392352,  
 "reason": 2,  
 "uname": [  
 "root"  
 ],  
 "status": -10,  
 "block": 0,  
 "blockDuration": 0  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| loginTime | Integer | int(10) | 攻击时间，时间戳，精确到秒 |
| reason | Integer | int(6) | 登录失败原因，1：用户名不存在 |
| uname | List | varchar(128) | 用户名 |
| status | Integer | int(6) | 状态 |
| block | Integer | tinyint(1) | 封停状态，见《封停状态字典表》 |
| blockDuration | Integer | int(10) | 封停时间，单位秒 |

### 封停/解封操作

该接口提供Linux/Windows暴力破解封停/解封操作。

**调用接口：**

POST /external/api/detect/brutecrack/{linux,win}/block

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 记录ID |
| block | int | 是 | 操作类型:0-解封;1-封停 |

**请求示例：**

{  
 "id": "tFU7aUo3o5w7GDPlJqQfMA==",  
 "block": 1  
}

**返回示例：**

{  
 "id": "tFU7aUo3o5w7GDPlJqQfMA==",  
 "block": 6,  
}

**返回部分说明：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 记录ID |
| block | Integer | 封停状态，见[**封停状态**](#暴力破解封停状态) |

**代码示例：**

# 入侵检测-暴力破解-封停/解封操作接口调用的代码示例(python)  
def brutecrack\_block():  
 url = "http://%s:%s/external/api/detect/brutecrack/linux/block" % (host, port)  
 data = {'id': 'tFU7aUo3o5w7GDPlJqQfMA==', 'block': 1}  
 res = send\_request("POST", url, data)  
 print res

**异常说明：**

1. 当agent不在线的时候，接口返回500， agent不在线。
2. 处于封停/解封中的项不能再进行封停/解封操作，否则会抛出异常及提示。

## 异常登录

### 查询结果

该接口提供Linux/Windows异常登录上报事件的数据查询。

**调用接口：**

GET /external/api/detect/abnormallogin/{linux,win}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| loginTime | DateRange | 否 | 登录时间，格式参考[通用Get请求参数类型及说明](#通用get请求参数类型及说明) |
| flag | int数组 | 否 | 异常登录类型:2-异常IP;3-异常区域;4-异常时间 |
| ip | String | 否 | 主机IP |
| uname | String | 否 | 用户名 |
| clientIp | String | 否 | 来源IP |
| hostname | String | 否 | 主机名（windows无该参数） |

**请求示例：**

/external/api/detect/abnormallogin/linux?page=0&size=50&group=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "dbcf7ac1b5537764",  
 "displayIp": "172.16.12.1",  
 "connectionIp": "172.16.2.138",  
 "externalIp": null,  
 "internalIp": "172.16.12.1",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [  
 "标签a",  
 "标签b"  
 ],  
 "hostname": "oldtcm",  
 "loginTime": 1523262390,  
 "uname": "qingteng",  
 "clientIp": "58.49.121.58",  
 "clientLocation": "湖北武汉",  
 "evilIpDb": 0,  
 "crackSuccess": 0,  
 "abnormalIp": 1,  
 "abnormalLocation": 1,  
 "abnormalTime": 1,  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| loginTime | Integer | int(10) | 登录时间，时间戳，精确到秒 |
| uname | String | varchar(128) | 用户名 |
| clientIp | String | varchar(15) | 来源IP |
| clientLocation | String | varchar(128) | 登录区域 |
| evilIpDb | Integer | tinyint(1) | 是否恶意IP库；0-否,1-是 |
| crackSuccess | Integer | tinyint(1) | 暴力破解成功的登录：0-否；1-是 |
| abnormalIp | Integer | tinyint(1) | 不在常用ip段登录，0:否,1：是 |
| abnormalLocation | Integer | tinyint(1) | 不在常用区域的登录，0-否；1-是 |
| abnormalTime | Integer | tinyint(1) | 不在常用时间的登录，0:否,1：是 |
| loginRuleId | Set | varchar(24) | 正常登录规则ID |

### 设置规则

该接口提供Linux/Windows异常登录设置正常登录规则功能。

**调用接口：**

POST /external/api/detect/abnormallogin/{linux,win}/rule

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| realmType | int | 是 | 主机范围，1：自定义，包括业务组、主机； 0：全部主机 |
| groups | int数组 | 否 | 业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| agentIds | String数组 | 否 | 主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| loginIpInfo | Object | 否 | 登录IP |
| loginTimeInfo | Object | 否 | 登录时间 |
| loginLocationInfo Object | 否 | 登录区域 |  |
| hostname | String | 否 | 主机名（windows无该参数） |

loginIpInfo数据结构：

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| type | int | 是 | IP类型，1：普通IP/CIDR 2：IP段 |
| ip | String | 否 | IP |
| startStr | String | 否 | IP段起始 |
| endStr | String | 否 | IP段结束 |

loginTimeInfo数据结构：

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| start | String | 是 | 开始时间，如14：00 |
| end | String | 是 | 结束时间，如18：20 |
| weeks | int数组 | 是 | 周期(周一到周日)，如[1,2,3,4,5,6,0] |

loginLocationInfo数据结构：

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| country | string | 是 | 国家 |
| province | string | 否 | 省（当国家为中国以外时，该参数无需传入） |
| city | string | 否 | 市（当国家为中国以外时，该参数无需传入） |

**请求示例：**

{  
 "realmType": 0,  
 "groups": [22,33],  
"agentIds": [“4735a8101f2847c6”,”b439d8a7f96e37d7”],  
"loginIpInfo": [  
 {  
 "type": 2,  
 "ip": null,  
 "startStr": "192.168.0.1",  
 "endStr": "192.168.0.10"  
 }  
 ],  
 "loginTimeInfo": [  
 {  
 "start": "09:00",  
 "end": "21:00",  
 "weeks": [1, 2, 3, 4, 5]  
 }  
 ],  
 "loginLocationInfo": [  
 {  
 "country": "中国",  
 "province": "湖北省",  
 "city": "武汉市"  
 }  
 ]  
}

**返回示例：**

{  
 "flag": true  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### 查询单条规则

该接口提供Linux/Windows异常登录单条规则的数据查询。

**调用接口：**

GET /external/api/detect/abnormallogin/{linux,win}/rule/{id}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 规则ID |

**请求示例：**

/external/api/detect/abnormallogin/linux/rule/5a5dceb83171a7202baafae4

**返回示例：**

{  
 "id": "5a9d026a7d761b3ccfcf2c3d",  
 "osType": 1,  
 "realmKind": 2,  
 "realmGroup": 2,  
 "realmAgentId": null,  
 "realmAgentGroup": null,  
 "condition": {  
 "loginIps": null,  
 "loginTimes": [  
 {  
 "start": "08:00",  
 "end": "10:00",  
 "startMinutes": 480,  
 "endMinutes": 600,  
 "weeks": [1, 2, 3 ]  
 }  
 ],  
 "loginLocations": [  
 {  
 "country": "中国",  
 "province": "北京",  
 "city": ""  
 }  
 ],  
 "combineType": 1,  
 "combineTypeAll": true  
 }  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | String | varchar(24) | 规则ID |
| osType | Integer | tinyint(1) | 系统类型，1:linux 2:windows |
| realmKind | Integer | tinyint(1) | 主机范围:0-全部主机;1-主机 2-业务组 |
| realmGroup | Integer | bigint(20) | 业务组，realmKind=2, 该字段有值 |
| realmAgentId | String | varchar(16) | 主机，realmKind=1, 该字段有值 |
| realmAgentGroup | Integer | bigint(20) | 主机所在业务组ID |
| condition | Object |  | 规则详情，参见[condition数据结构](#condition数据结构) |

#### condition数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| loginIps | Object |  | 登录IP，参见[loginIps数据结构](#loginips数据结构) |
| loginTimes | Object |  | 登录时间，[loginTimes数据结构](#logintimes数据结构) |
| loginLocations | Object |  | 登录区域，[loginLocations数据结构](#loginlocations数据结构) |
| combineType | Integer | tinyint(1) | 1:所有条件均满足 2:任一条件满足 |
| combineTypeAll | Boolean | tinyint(1) | 条件组合方式 true：所有条件满足(默认) |

#### loginIps数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| type | Integer | tinyint（1） | IP类型，1：普通IP/CIDR 2：IP段 |
| ip | String | varchar(15) | IP |
| startStr | String | varchar(25) | IP段起始 |
| endStr | String | varchar(25) | IP段结束 |
| start | Integer | int(15) | IP段起始 |
| end | Integer | int(15) | IP段结束 |

#### loginTimes数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| start | String | varchar(6) | 开始时间，如14：00 |
| end | String | varchar(6) | 结束时间，如18：20 |
| startMinutes | Integer | int(6) | 开始时间，如840(例：14×60=840) |
| endMinutes | Integer | int(6) | 结束时间，如1100(例：18×60+20=1100) |
| weeks | List | tinyint(1) | 周期(周一到周日)，如[1,2,3,4,5,6,0] |

#### loginLocations数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| country | String | varchar(128) | 国家 |
| province | String | varchar(128) | 省 |
| city | String | varchar(128) | 市 |

**异常说明：**

1. 规则不存在，则会抛出异常及相关提示
2. 帐号没有权限，会抛出异常及相关提示

### 查询多条规则

该接口提供Linux/Windows异常登录规则的数据查询。

**调用接口：**

GET /external/api/detect/abnormallogin/{linux,win}/rule

**请求示例：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| loginIp | String | 否 | 登录IP |
| loginLocation | String | 否 | 登录区域 |

**请求示例：**

/external/api/detect/abnormallogin/linux/rule

**返回示例：**

{  
 "rows": [  
 {  
 "id": "5a9d026a7d761b3ccfcf2c3d",  
 "osType": 1,  
 "realmKind": 2,  
 "realmGroup": 2,  
 "realmAgentId": null,  
 "realmAgentGroup": null,  
 "condition": {  
 "loginIps": null,  
 "loginTimes": [  
 {  
 "start": "08:00",  
 "end": "10:00",  
 "startMinutes": 480,  
 "endMinutes": 600,  
 "weeks": [  
 1,  
 2,  
 3  
 ]  
 }  
 ],  
 "loginLocations": [  
 {  
 "country": "中国",  
 "province": "北京",  
 "city": ""  
 }  
 ],  
 "combineType": 1,  
 "combineTypeAll": true  
 }  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| id | String | varchar(24) | 规则ID |
| osType | Integer | tinyint(1) | 系统类型：1-linux；2-windows |
| realmKind | Integer | tinyint(1) | 主机范围：0-全部主机；1-主机；2-业务组 |
| realmGroup | Integer | bigint(20) | 业务组，realmKind=2, 该字段有值 |
| realmAgentId | String | varchar(16) | 主机，realmKind=1, 该字段有值 |
| realmAgentGroup | Integer | bigint(20) | 主机所在业务组ID |
| condition | Object |  | 规则详情，见[condition数据结构](#condition数据结构) |

### 删除规则

该接口提供Linux/Windows异常登录删除规则功能。

**调用接口：**

DELETE /external/api/detect/abnormallogin/{linux,win}/rule/{id}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 规则ID |

**请求示例：**

/external/api/detect/abnormallogin/linux/rule/5a9d026a7d761b3ccfcf2

**返回示例：**

{  
 "flag": true  
}

**异常说明：**

1. 规则不存在，则会抛出异常及相关提示
2. 帐号没有权限，会抛出异常及相关提示

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

## Web后门

### Web后门查询结果

该接口提供web后门扫描结果数据查询。

**调用接口：**

GET /external/api/websecurity/webshell/{linux|win}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 否 | 主机ID |
| hostname | String | 否 | 主机名 |
| groups | int数组 | 否 | 业务组 |
| types | int数组 | 否 | 类型 |
| ip | String | 否 | 主机IP |
| serverName | String | 否 | 域名 |
| filePath | String | 否 | 文件路径 |

**请求示例：**

/external/api/websecurity/webshell/linux?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "70db8ef89e9ae79a",  
 "displayIp": "172.16.159.1",  
 "connectionIp": "172.16.2.231",  
 "externalIp": null,  
 "internalIp": "172.16.159.1",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "hostname",  
 "id": "5a5dd69c7d761b24f3ea04c1",  
 "type": 5,  
 "typeDesc": "已知后门",  
 "serverName": "www.zhn.com",  
 "filePath": "/home/qt/nan/ii.php",  
 "fileMd5": "112378feba9a9ba493a1d0bd0acfb180",  
 "fileSize": 16,  
 "regexDesc": "PHP WebShell特征码",  
 "matchCount": 1,  
 "modifyTime": 1505122390,  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| id | String | varchar(24) | ID |
| type | Integer | int(10) | 类型 |
| typeDesc | String | varchar(128) | 类型描述 |
| serverName | String | varchar(128) | 域名 |
| filePath | String | varchar(128) | 文件路径 |
| fileMd5 | String | varchar(50) | 文件MD5值 |
| fileSize | Integer | int(15） | 文件大小 |
| regexDesc | String | varchar(128) | 说明 |
| matchCount | Integer | int(10) | 命中规则数 |
| modifyTime | Integer | int(10) | 最后修改时间，时间戳，精确到秒 |
| whiteRuleEffect | Boolean | tinyint(1) | 是否命中白名单 |

### Web后门扫描

该接口提供web后门执行扫描任务。

**调用接口：**

POST /external/api/websecurity/webshell/{linux|win}/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| realmType | int | 是 | 范围，0:当前账号管理下的全部主机； 1:自定义主机； 2:自定义业务组； |
| agentIds | String数组 | 否 | 主机列表，如果realmType=1, 则agentIds不为空 |
| groups | int数组 | 否 | 业务组列表，如果realmType=2, 则groups不为空 |

**请求示例：**

// 当选全部主机  
{  
 "realmType": 0  
}  
// 当选自定义主机  
{  
 "realmType": 1  
 "agentIds": ["979f3c4d44b2e765", "979f3c4d44b2e766"]  
}  
// 当选自定义业务组  
{  
 "realmType": 2  
 "groups": [1, 2, 3]  
}

**返回示例：**

{  
 "flag": true  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### Web后门查询扫描状态

该接口提供web后门扫描任务的状态查询。

**调用接口：**

GET /external/api/websecurity/webshell/{linux|win}/check/status

**请求参数：**

无

**请求示例：**

/external/api/websecurity/webshell/linux/check/status

**返回示例：**

{  
 "retcode": 3  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| retcode | Integer | tinyint(1) | -1:无任务0:未开始执行 1:排队执行 2:正在执行 3:执行成功 4:执行失败 |

### Web后门-文件下载

该接口提供web后门文件下载。

**调用接口：**

GET /external/api/websecurity/webshell/{linux|win}/download/{id}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | int | 是 | 数据id |

**请求示例：**

/external/api/websecurity/webshell/{linux|win}/download/5a790f0b7d761b6bc78e

**返回示例：**

{  
 "fileName": "php.php",  
 "content": "\ufeff<?php @eval($\_POST['c']);?>\n"  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| fileName | String | varchar(128) | 文件名 |
| content | String | varchar(1024) | 文件内容 |

## 反弹shell(Windows&Linux)

### 查询结果

该接口提供Windows和Linux反弹shell上报事件的数据查询。

**调用接口：**

//windows  
GET /external/api/detect/bounceshell/win  
//linux  
GET /external/api/detect/bounceshell/linux

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| createTime | DateRange | 否 | 发现时间，格式参考[通用Get请求参数类型及说明](#通用get请求参数类型及说明) |
| processName | String | 否 | 连接进程 |
| ip | String | 否 | 主机IP（模糊查询） |
| targetIpLike | String | 否 | 目标主机(模糊查询) |
| targetPort | int | 否 | 端口 |
| hostname | String | 否 | 主机名（模糊查询） |

**请求示例：**

/external/api/detect/bounceshell/linux?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "dbcf7ac1b5537764",  
 "displayIp": "172.16.12.1",  
 "connectionIp": "172.16.2.138",  
 "externalIp": null,  
 "internalIp": "172.16.12.1",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [  
 "标签a",  
 "标签b"  
 ],  
 "hostname": "oldtcm",  
 "createTime": 1525935820,  
 "pid": 21794,  
 "processName": "bash",  
 "targetIp": "192.168.98.136",  
 "targetPort": 4444,  
 "userName": "root",  
 "currentPath": "/root",  
 "groupName": "业务组a",  
 "parentPid": 44382,  
 "parentName": "bash",  
 "parentPath": "/bin/bash",  
 "stdInfo": "stdin,stderr,stdout",  
 "processTree": [  
 {  
 "userName": "root",  
 "uid": 0,  
 "ppid": 0,  
 "pid": 1,  
 "path": "/sbin/init",  
 "name": "init",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "/sbin/init"  
 },  
 {  
 "userName": "root",  
 "uid": 0,  
 "ppid": 1,  
 "pid": 1097,  
 "path": "/usr/sbin/sshd",  
 "name": "sshd",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "/usr/sbin/sshd -D "  
 },  
 {  
 "userName": "root",  
 "uid": 0,  
 "ppid": 1097,  
 "pid": 44340,  
 "path": "/usr/sbin/sshd",  
 "name": "sshd",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "sshd: root@pts/19 "  
 },  
 {  
 "userName": "root",  
 "uid": 0,  
 "ppid": 44340,  
 "pid": 44382,  
 "path": "/bin/bash",  
 "name": "bash",  
 "groupName": "",  
 "fileMode": "100755",  
 "euidUserName": "",  
 "cmd": "-bash "  
 },  
 {  
 "userName": "root",  
 "uid": 0,  
 "ppid": 44382,  
 "pid": 21794,  
 "path": "/bin/bash",  
 "name": "bash",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "bash -i "  
 }  
 ],  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| createTime | Integer | int(10) | 发现时间，时间戳，精确到秒 |
| pid | Integer | int(10) | 连接进程ID |
| processName | String | varchar(20) | 连接进程 |
| targetIp | String | varchar(15) | 目标主机 |
| targetPort | Integer | int(10) | 目标端口 |
| userName | String | varchar(20) | 进程启动用户 |
| currentPath | String | varchar(128) | 进程路径 |
| groupName | String | varchar(20) | 用户所属组 |
| parentPid | Integer | int(10) | 父进程ID |
| parentName | String | varchar(20) | 父进程 |
| parentPath | String varchar(128) | 父进程路径 |  |
| stfInfo | String | varchar(50) | 标准I/O信息 |
| processTree | List |  | 进程树,按顺序为父进程至子进程 |
| whiteRuleEffect | Boolean | tinyint(1) | 是否匹配白名单 |

ProcessTree数据结构：

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| userName | String | varchar(50) | 进程启动用户 |
| uid | Integer | int(10) | 进程启动用户ID |
| pid | Integer | int(10) | 连接进程ID |
| ppid | Integer | int(10) | 连接进程父进程ID |
| path | String | varchar(256) | 进程路径 |
| name | String | varchar(20) | 连接进程 |
| groupName | String | varchar(50) | 用户所属组 |
| fileMode | String | varchar(256) | 完整文件权限 |
| euidUserName | String | varchar(50) | 进程euid用户 |
| cmd | String | varchar(256) | 命令行参数 |

注意： windows返回信息中没有linux中特有的euidUserName、fileMode、fileMode等字段。

## 本地提权

### Linux本地提权查询结果

该接口提供Linux本地提权上报事件的数据查询。

**调用接口：**

GET /external/api/detect/localrights/linux

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| time | DateRange | 否 | 发现时间，格式参考[通用Get请求参数类型及说明](#通用get请求参数类型及说明) |
| ip | String | 否 | 主机IP（模糊查询） |
| procName | String | 否 | 提权进程 |
| procUserName | String | 否 | 提权用户 |

**请求示例：**

/external/api/detect/localrights/linux?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "979f3c4d44b2e765",  
 "displayIp": "172.16.2.122",  
 "connectionIp": "172.16.2.122",  
 "externalIp": null,  
 "internalIp": "172.16.2.122",  
 "bizGroupId": 37,  
 "bizGroup": "test",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "tianxianhu",  
 "time": 1529984917,  
 "procPid": 67537,  
 "procName": "su",  
 "procPath": "/bin/su",  
 "procUserName": "test",  
 "procGroupName": "test",  
 "procFileMode": "104755",  
 "parentProcPid": 67533,  
 "parentProcName": "dcow\_(1)",  
 "parentProcPath": "/home/test/dcow\_(1)",  
 "parentProcUserName": "test",  
 "parentProcGroupName": "test",  
 "processTreeList": [  
 {  
 "pid": 1,  
 "ppid": 0,  
 "name": "init",  
 "path": "/sbin/init",  
 "uid": 0,  
 "userName": "root",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "/sbin/init"  
 },  
 {  
 "pid": 1008,  
 "ppid": 1,  
 "name": "sshd",  
 "path": "/usr/sbin/sshd",  
 "uid": 0,  
 "userName": "root",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "/usr/sbin/sshd -D "  
 },  
 {  
 "pid": 67477,  
 "ppid": 1008,  
 "name": "sshd",  
 "path": "/usr/sbin/sshd",  
 "uid": 0,  
 "userName": "root",  
 "groupName": "root",  
 "fileMode": "100755",  
 "euidUserName": "root",  
 "cmd": "sshd: test [priv] "  
 },  
 {  
 "pid": 67514,  
 "ppid": 67477,  
 "name": "sshd",  
 "path": "/usr/sbin/sshd",  
 "uid": 1000,  
 "userName": "test",  
 "groupName": "test",  
 "fileMode": "100755",  
 "euidUserName": "test",  
 "cmd": "sshd: test@pts/0 "  
 },  
 {  
 "pid": 67515,  
 "ppid": 67514,  
 "name": "bash",  
 "path": "/bin/bash",  
 "uid": 1000,  
 "userName": "test",  
 "groupName": "test",  
 "fileMode": "100755",  
 "euidUserName": "test",  
 "cmd": "-bash "  
 },  
 {  
 "pid": 67533,  
 "ppid": 67515,  
 "name": "dcow\_(1)",  
 "path": "/home/test/dcow\_(1)",  
 "uid": 1000,  
 "userName": "test",  
 "groupName": "test",  
 "fileMode": "100744",  
 "euidUserName": "test",  
 "cmd": "./dcow\_(1) -s "  
 },  
 {  
 "pid": 67537,  
 "ppid": 67533,  
 "name": "su",  
 "path": "/bin/su",  
 "uid": 1000,  
 "userName": "test",  
 "groupName": "test",  
 "fileMode": "104755",  
 "euidUserName": "root",  
 "cmd": "su - "  
 }  
 ],  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| time | Integer | int(10) | 发现时间，时间戳，精确到秒 |
| procPid | Integer | int(10) | 提权进程ID |
| procName | String | varchar(20) | 提权进程 |
| procPath | String | varchar(128) | 提权进程路径 |
| procUserName | String | varchar(20) | 提权用户 |
| procGroupName | String | varchar(20) | 用户所属组 |
| procFileMode | String | varchar(128) | 完整文件权限 |
| parentProcPid | Integer | int(10) | 父进程id |
| parentProcName | String | varchar(20) | 父进程 |
| parentProcPath | String | varchar(128) | 父进程路径 |
| parentProcUserName | String | varchar(20) | 父进程启动用户 |
| parentProcGroupName | String | varchar(20) | 父进程用户所属组 |
| processTreeList | List |  | 进程树,按顺序为父进程至子进程 |
| whiteRuleEffect | Boolean | tinyint(1) | 是否匹配白名单 |

## Linux后门检测

### 查询结果

该接口提供Linux后门检测扫描结果的数据查询。

**调用接口：**

GET /external/api/detect/backdoor/linux

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| ip | String | 否 | 主机IP（模糊查询） |
| hostname | String | 否 | 主机名（模糊查询） |
| backDoorTypes | String数组 | 否 | 后门类型，包含： 应用后门，Bootkit，Rootkit；恶意进程会返回对应检测库中的后门类型 |
| backDoorCheckNames | String数组 | 否 | 检查功能，见[Linux后门检查功能](#linux后门检查功能) |
| backDoorName | String | 否 | 检查项 |

Linux检查功能字典表：

| 值 | 说明 |
| --- | --- |
| RPM-based应用后门检查 | 无 |
| DPKG-based应用后门检查 | 无 |
| 磁盘MBR检查 | 无 |
| 计划任务检查 | 无 |
| 动态链接库检查 | 无 |
| 基本命令检查 | 无 |
| 已知rootkit检查 | 无 |
| 系统内核模块检查 | 无 |
| 网络状态检查 | 无 |
| 用户状态检查 | 无 |
| 系统文件状态检查 | 无 |
| 进程状态检查 | 无 |

**请求示例：**

/external/api/detect/backdoor/linux?page=0&size=50&group=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "dbcf7ac1b5537764",  
 "displayIp": "172.16.12.1",  
 "connectionIp": "172.16.2.138",  
 "externalIp": null,  
 "internalIp": "172.16.12.1",  
 "bizGroupId": 41,  
 "bizGroup": "业务组a",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [  
 "标签a"  
 ],  
 "hostname": "oldtcm",  
 "createTime": 1528398278,  
 "backDoorType": "Rootkit",  
 "backDoorName": "检查\"/bin/tar\"命令",  
 "backDoorCheckName": "基本命令检查",  
 "description": "通过检测strings命令成功后，利用strings命令打印命令的段信息，然后查找这些命令中是否有特征值，如果有那么表示该命令有问题",  
 "detail": "{\"cmd\":\"/bin/tar\",\"match\_rules\":{\"137\":\"^...s\"}}",  
 "instruct": "替换相应命令，并且重新检查",  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| createTime | Integer | int(10) | 发现时间，时间戳，精确到秒 |
| backDoorType | String | varchar(10) | 后门类型，包含：应用后门，Bootkit，Rootkit |
| backDoorName | String | varchar(50) | 检查项 |
| backDoorCheckName | String | varchar(50) | 检查功能，见**《Linux检查功能字典表》** |
| description | String | varchar(256) | 检查说明，恶意进程中为空 |
| detail | String | varchar(1024) | 检查结果，见**《附录1》** |
| instruct | String | varchar(256) | 修复方法 |
| whiteRuleEffect | Boolean | tinyint(1) | 是否匹配白名单 |

### Linux后门检测-开始扫描

该接口提供Linux后门检测执行扫描任务的功能。

**调用接口：**

POST /external/api/detect/backdoor/linux/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| realmType | int | 是 | 范围，0:当前账号管理下的全部主机； 1:自定义主机； 2:自定义业务组； |
| agentIds | String数组 | 否 | 主机列表，如果realmType=1, 则agentIds不为空 |
| groups | int数组 | 否 | 业务组列表，如果realmType=2, 则groups不为空 |

**请求示例：**

// 当选全部主机  
{  
 "realmType": 0  
}  
// 当选自定义主机  
{  
 "realmType": 1  
 "agentIds": ["979f3c4d44b2e765", "979f3c4d44b2e766"]  
}  
// 当选自定义业务组  
{  
 "realmType": 2  
 "groups": [1, 2, 3]  
}

**返回示例：**

{  
 "flag": true  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

**异常说明：**

1、如果功能开关未打开，则会抛出异常及相关提示  
2、如果上次扫描任务未执行完毕，则会抛出异常及相关提示

### Linux后门检测-查询扫描

该接口提供Linux后门检测当前正在执行的扫描任务的状态查询。

**调用接口：**

GET /external/api/detect/backdoor/linux/check/status

**请求参数：**

无

**请求示例：**

/external/api/detect/backdoor/linux/check/status

**返回示例：**

{  
 "retcode": 3,  
 "status": "success",  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| retcode | Integer | tinyint(1) | 2:正在执行 3:执行成功 4:执行失败 |
| status | String | varchar(10) | 2:running 3:success 4: failed |

## Windows后门检测

### 查询结果

该接口提供Windows后门检测扫描结果的数据查询。

**调用接口：**

GET /external/api/detect/backdoor/win

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| ip | String | 否 | 主机IP（模糊查询） |
| hostname | String | 否 | 主机名（模糊查询） |
| backDoorTypeIds | int数组 | 否 | 后门类型，见[Windows后门类型](#windows后门类型) |
| name | String | 否 | 问题摘要 |

Windows后门类型字典表：

| 值 | 说明 |
| --- | --- |
| 0 | 未知类型 |
| 1 | 可疑进程 |
| 2 | 可疑线程 |
| 3 | 可疑模块 |
| 4 | 可疑启动项 |
| 5 | 映像劫持 |
| 1000 | 恶意进程 |

**请求示例：**

/external/api/detect/backdoor/win?page=0&size=50&group=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "b6f30c43688e4798",  
 "displayIp": "192.168.199.191",  
 "connectionIp": "192.168.199.191",  
 "externalIp": null,  
 "internalIp": "192.168.199.191",  
 "bizGroupId": 2,  
 "bizGroup": "未分组主机（Windows）",  
 "remark": null,  
 "hostTagList": [],  
 "hostname": "ADMIN-PC",  
 "createTime": 0,  
 "backDoorTypeId": 1,  
 "name": "进程svchost.exe加载异常网络模块",  
 "reason": "进程svchost.exe存在被动态加载的网络模块，如果不是正常的挂载，那么极有可能被窃取信息",  
 "data": {  
 "item": {  
 "parent\_process\_id": 1712,  
 "b\_wow64": true,  
 "process\_id": 1480,  
 "b\_64bit": false,  
 "create\_time": 1512045079,  
 "process\_name": "svchost.exe",  
 "parent\_process\_name": "",  
 "image\_file\_name": "C:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\svchost.exe"  
 },  
 "ret\_msg": "dynamic load network module process",  
 "ret\_code": "0x00000005"  
 },  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| createTime | Integer | int(10) | 发现时间，时间戳，精确到秒 |
| name | String | varchar(256) | 问题摘要 |
| backDoorTypeId | Integer | tinyint(2) | 后门类型，见**《Windows后门类型字典表》** |
| reason | String | varchar(256) | 异常原因，恶意进程中为空 |
| data | String | varchar(1024) | 详情，见**《附录2》** |
| whiteRuleEffect | Boolean | tinyint(1) | 是否匹配白名单 |

### Windows后门检测-开始扫描

该接口提供Windows后门检测执行扫描任务的功能。

**调用接口：**

POST /external/api/detect/backdoor/win/check

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| realmType | int | 是 | 范围，0:当前账号管理下的全部主机； 1:自定义主机； 2:自定义业务组； |
| agentIds | String数组 | 否 | 主机列表，如果realmType=1, 则agentIds不为空 |
| groups | int数组 | 否 | 业务组列表，如果realmType=2, 则groups不为空 |

**请求示例：**

// 当选全部主机  
{  
 "realmType": 0  
}  
// 当选自定义主机  
{  
 "realmType": 1  
 "agentIds": ["979f3c4d44b2e765", "979f3c4d44b2e766"]  
}  
// 当选自定义业务组  
{  
 "realmType": 2  
 "groups": [1, 2, 3]  
}

**返回示例：**

{  
 "flag": true  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

**异常说明：**

1、如果功能开关未打开，则会抛出异常及相关提示  
2、如果上次扫描任务未执行完毕，则会抛出异常及相关提示

### Windows后门检测-查询扫描

该接口提供Windows后门检测当前正在执行的扫描任务的状态查询。

**调用接口：**

GET /external/api/detect/backdoor/win/check/status

**请求参数：**

无

**请求示例：**

/external/api/detect/backdoor/win/check/status

**返回示例：**

{  
 "retcode": 3,  
 "status": "success",  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| retcode | Integer | tinyint(1) | 2:正在执行 3:执行成功 4:执行失败 |
| status | String | varchar(10) | 2:running 3:success 4: failed |

## 网络蜜罐

### 查询结果

该接口提供Linux/Windows网络蜜罐上报事件的数据查询

**调用接口：**

GET /external/api/detect/honeypot/{linux,win}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| time | DateRange | 否 | 发现时间，格式参考[通用Get请求参数类型及说明](#通用get请求参数类型及说明) |
| clientIp | String | 否 | 源IP |
| ip | String | 否 | 被攻击IP（模糊查询） |
| port | int | 否 | 主机端口 |

**请求示例：**

/external/api/detect/honeypot/linux?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "70db8ef89e9ae79a",  
 "displayIp": "172.16.159.1",  
 "connectionIp": "172.16.2.231",  
 "externalIp": null,  
 "internalIp": "172.16.159.1",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "hostname",  
 "time": 1528979258,  
 "clientIp": "172.168.1.190",  
 "port": 11,  
 "whiteRuleEffect": false  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| time | Integer | int(10) | 发现时间，时间戳，精确到秒 |
| clientIp | String | varchar(15) | 源IP |
| port | Integer | int(8) | 主机端口 |
| whiteRuleEffect | Boolean | tinyint(1) | 是否匹配白名单 |

### 创建单条规则

该接口提供Linux/Windows网络蜜罐设置主机规则功能。

**调用接口：**

POST /external/api/detect/honeypot/{linux,win}/rule

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 是 | 主机ID |
| ports | int数组 | 是 | 端口，限制不能超过5个 |

**请求示例：**

{  
 "agentId": "979f3c4d44b2e765",  
 "ports": [8060,10020]  
}

**返回示例：**

{  
 "flag": true  
}

**异常说明：**

1. 规则主机不存在，则会抛出异常及相关提示
2. 帐号没有权限，会抛出异常及相关提示
3. 主机规则已经存在，则会抛出异常及相关提示
4. 规则端口为空，则会抛出异常及相关提示
5. 规则端口个数大于5个，则会抛出异常及相关提示

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### 批量创建规则

该接口提供Linux/Windows网络蜜罐批量创建主机规则功能

**调用接口：**

POST /external/api/detect/honeypot/{linux,win}/rules

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| realmType | int | 是 | 范围，0:全部主机 1:自定义，包括业务组和主机 |
| groups | int数组 | 否 | 业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| agentIds | String数组 | 否 | 主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| excludeGroups | int数组 | 否 | 排除业务组列表 |
| excludeAgentIds | String数组 | 否 | 排除主机列表 |
| ports | int数组 | 是 | 端口列表（端口个数不能超过5个） |

**请求示例：**

/external/api/detect/honeypot/linux/rules

**返回示例：**

{  
 "flag": true  
}

**异常说明：**

1. 范围参数不合法，则会抛出异常及相关提示
2. 范围为自定义范围时，业务组和主机同时为空，则会抛出异常及相关提示
3. 端口为空，或端口数量大于5，会抛出异常及相关提示

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### 删除单条规则

该接口提供Linux/Windows网络蜜罐删除主机规则功能。

**调用接口：**

DELETE /external/api/detect/honeypot/{linux,win}/rule/{id}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 主机ID |

**请求示例：**

/external/api/detect/honeypot/linux/rule/70db8ef89e9ae79a

**返回示例：**

{  
 "flag": true  
}

**异常说明：**

1. 规则主机不存在，则会抛出异常及相关提示
2. 规则不存在，则会抛出异常及相关提示
3. 帐号没有权限，会抛出异常及相关提示
4. 规则同步中不能删除，会抛出异常及相关提示
5. 规则启用状态不能删除，会抛出异常及相关提示

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### 批量删除规则

该接口提供Linux/Windows网络蜜罐批量删除主机规则功能。

**调用接口：**

DELETE /external/api/detect/honeypot/{linux,win}/rules

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| realmType | int | 是 | 范围，0:全部主机 1:自定义，包括业务组和主机 |
| groups | int数组 | 否 | 业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| agentIds | String数组 | 否 | 主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| ports | int数组 | 是 | 端口列表（端口个数不能超过5个） |

**请求示例：**

/external/api/detect/honeypot/linux/rules

**返回示例：**

{  
 "flag": true  
}

**异常说明：**

1. 范围参数不合法，则会抛出异常及相关提示
2. 范围为自定义范围时，业务组和主机同时为空，则会抛出异常及相关提示
3. 端口为空，或端口数量大于5，会抛出异常及相关提示

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

### 查询规则

该接口提供Linux/Windows网络蜜罐规则的数据查询。

**调用接口：**

GET /external/api/detect/honeypot/{linux,win}/rule

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| groups | int数组 | 否 | 业务组 |
| ip | String | 否 | 主机IP |

**请求示例：**

/external/api/detect/honeypot/linux/rule?page=0&size=50&groups=1,2

**返回示例：**

{  
 "rows": [  
 {  
 "agentId": "70db8ef89e9ae79a",  
 "displayIp": "172.16.159.1",  
 "connectionIp": "172.16.2.231",  
 "externalIp": null,  
 "internalIp": "172.16.159.1",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "hostname",  
 "status": 0,  
 "verifyStatus": 1,  
 "ports": [  
 {  
 "port": 80,  
 "status": 2  
 },  
 {  
 "port": 8000,  
 "status": 2  
 },  
 {  
 "port": 8080,  
 "status": 2  
 }  
 ]  
 }  
 ],  
 "total": 1  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| status | Integer | tinyint(1) | 规则状态，0:规则启用 1:规则禁用 |
| verifyStatus | Integer | tinyint(1) | 同步状态，0:规则同步中 1:同步完毕 |
| ports | List |  | 规则端口 |

**Port数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| port | Integer | int(8) | 端口 |
| status | Integer | int(4) | 端口状态，含义如下：-1:同步行中 1:同步成功 0:同步失败 |

### 查询单条规则

该接口提供Linux/Windows网络蜜罐单条规则的数据查询。

**调用接口：**

GET /external/api/detect/honeypot/{linux,win}/rule/{id}

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 主机ID |

**请求示例：**

/external/api/detect/honeypot/linux/rule/70db8ef89e9ae79a

**返回示例：**

{  
 "agentId": "70db8ef89e9ae79a",  
 "displayIp": "172.16.159.1",  
 "connectionIp": "172.16.2.231",  
 "externalIp": null,  
 "internalIp": "172.16.159.1",  
 "bizGroupId": 1,  
 "bizGroup": "未分组主机（Linux）",  
 "remark": "安装了mysql或者DNSmasq",  
 "hostTagList": [],  
 "hostname": "hostname",  
 "status": 0,  
 "verifyStatus": 1,  
 "ports": [  
 {  
 "port": 1112,  
 "status": 2  
 },  
 {  
 "port": 80,  
 "status": 2  
 },  
 {  
 "port": 8000,  
 "status": 2  
 },  
 {  
 "port": 8080,  
 "status": 2  
 }  
 ]  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| agentId | String | varchar(16) | 主机ID，16位 |
| displayIp | String | varchar(15) | 主机IP |
| connectionIp | String | varchar(15) | 连接IP |
| externalIp | String | varchar(15) | 外网IP |
| internalIp | String | varchar(15) | 内网IP |
| bizGroupId | Integer | bigint(20) | 业务组ID |
| bizGroup | String | varchar(128) | 业务组名 |
| remark | String | varchar(1024) | 备注 |
| hostTagList | List | varchar(1024) | 标签 |
| hostname | String | varchar(512) | 主机名 |
| status | Integer | tinyint(1) | 规则状态，0:规则启用 1:规则禁用 |
| verifyStatus | Integer | tinyint(1) | 同步状态，0:规则同步中 1:同步完毕 |
| ports | List | 见详细 | 规则端口 |

**Port数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| port | Integer | int(8) | 端口 |
| status | Integer | int(4) | 端口状态，含义如下：-1:同步行中 1:同步成功 0:同步失败 |

**异常说明：**

1. 规则不存在，则会抛出异常及相关提示
2. 帐号没有权限，会抛出异常及相关提示

### 更新单条规则

该接口提供Linux/Windows网络蜜罐更新主机规则功能。

**调用接口：**

PUT /external/api/detect/honeypot/{linux,win}/rule

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 是 | 主机ID，16位 |
| ports | int数组 | 是 | 端口列表（端口个数不能超过5个） |

**请求示例：**

/external/api/detect/honeypot/linux/rule

**返回示例：**

{  
 "flag": true  
}

**返回部分说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| flag | Boolean | tinyint(1) | true:成功 false:失败 |

**异常说明：**

1. 规则主机不存在，则会抛出异常及相关提示
2. 规则不存在，则会抛出异常及相关提示
3. 帐号没有权限，会抛出异常及相关提示
4. 规则同步中不能更新，会抛出异常及相关提示
5. 规则禁用状态不能更新，会抛出异常及相关提示
6. 端口参数为空，会抛出异常及相关提示
7. 端口数量大于5，会抛出异常及相关提示

### 开关单条规则

该接口提供Linux/Windows网络蜜罐开启/关闭主机规则功能。

**调用接口：**

PUT /external/api/detect/honeypot/{linux,win}/rule/enable

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 是 | 主机ID，16位 |
| status | int | 是 | 规则状态，0:开启 1:关闭 |

**请求示例：**

/external/api/detect/honeypot/linux/rule/enable

**返回示例：**

{  
 "flag": true  
}

**异常说明：**

1. 规则主机不存在，则会抛出异常及相关提示
2. 规则不存在，则会抛出异常及相关提示
3. 帐号没有权限，会抛出异常及相关提示
4. 状态参数不合法，会抛出异常及相关提示

# 合规基线

基线检查提供基线检查项在所有主机中的检查结果，即可查询某一个检查项，也可查询某个主机IP的结果，请根据需要选择。

## 新建任务

**调用接口：**

POST /external/api/baseline/job/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| name | String | 任务名，必填 |
| kind | int | 必填，范围类型 0-主机 1-业务组 2-全部 |
| values | String数组 | kind=0时必填，所选主机列表 |
| groups | int数组 | kind=1时必填，所选业务组列表 |
| ruleId | String | 必填，基线规则id，规则平台上查询id |
| cronEnabled | 布尔 | 必填，是否定时任务 true-定时任务 false-普通任务 |
| cron | String | cronEnabled=true必填，cron任务表达式 |
| description | String | 任务描述 |

**请求示例：**

{  
 "name": "test\_001", //必填，任务名  
 "kind": 1, //必填，范围类型 0-主机 1-业务组 2-全部  
 "values": ["407c1dd87b1d7771"], //kind=0时必填，所选主机列表  
 "groups": [1, 3, 446], //kind=1时必填，所选业务组列表  
 "ruleId": "1b8094cf2cacf0e42acf", //必填，基线规则id  
 "cronEnabled": false, //必填，是否定时任务 true-定时任务 false-普通任务  
 "cron": "0 17 \* \* \*", //cronEnabled=true必填，cron任务表达式  
 "description": "xxx" //任务描述  
}

**返回示例：**

{  
 "status":"success",//任务创建状态  
 "id": "xxxx" //任务specId  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| status | String | 任务创建状态 |
| id | String | 任务specId |

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
import httplib  
import json  
  
host = "127.0.0.1"  
port = 80  
  
# 创建基线任务示例  
def create\_baseline\_spec():  
 conn = httplib.HTTPConnection(host, port)  
 url = "http://%s:%s/external/api/baseline/job/linux" % (host, port)  
  
 body = {"name": "test\_spec","kind": 2,"values":[],"groups": [],"ruleId": "1b8094cf2cacf0e42acf","cronEnabled": false,"cron": "","description": "123"}  
 payload = json.dumps(body)  
 headers = {'Content-Type': "application/json"}  
  
 conn.request("POST", url, data=payload, headers=headers)  
 response = conn.getresponse()  
  
 print(response.text)

## 删除任务

**调用接口：**

DELETE /external/api/baseline/job/{linux，win}/{specId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| specId | String | 任务specId |

**请求示例：**

{  
 "specId": "xxxx" //任务specId  
}

**返回示例：**

{  
 "status":"success",//删除结果  
 "id": "xxxx" //任务specId  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| status | String | 删除结果 |
| id | String | 任务specId |

## 修改任务请求

**调用接口：**

PUT /external/api/baseline/job/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| name | String | 任务名，必填 |
| specId | String | 任务specId，必填 |
| cronEnabled | 布尔 | 必填，是否定时任务 true-定时任务 false-普通任务 |
| cron | String | cronEnabled=true必填，cron任务表达式 |
| description | String | 任务描述 |
| kind | int | 必填，范围类型：0-主机；1-业务组；2-全部 |
| values | String数组 | kind=0时必填，所选主机列表 |
| groups | int数组 | kind=1时必填，所选业务组列表 |
| ruleId | String | 必填，基线规则id |

**请求示例：**

{  
 "specId" : "5a71940b7d761b15dbce17ef",//必填  
 "name": "test\_001", //必填，任务名  
 "kind": 1, //必填，范围类型 0-主机 1-业务组 2-全部  
 "values": ["407c1dd87b1d7771"], //kind=0时必填，所选主机列表  
 "groups": [1, 3, 446], //kind=1时必填，所选业务组列表  
 "ruleId": "1b8094cf2cacf0e42acf", //必填，基线规则id  
 "cronEnabled": false, //必填，是否定时任务 true-定时任务 false-普通任务  
 "cron": "0 17 \* \* \*", //cronEnabled=true必填，cron任务表达式  
 "description": "xxx" //任务描述  
}

**返回示例：**

{  
 "status":"success",//任务修改状态  
 "id": "xxxx" //任务specId  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| status | String | 删除结果 |
| id | String | 任务specId |

## 查询任务

**调用接口：**

GET /external/api/baseline/job/{linux，win}

ps：暂不支持分页

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| osType | int | 操作系统，必填：1-linux；2-windows |
| specName | String | 任务名 |
| ruleIds | String数组 | 规则id列表 |
| groups | int数组 | 业务组列表 |
| ip | String | 主机ip |
| hostName | String | 主机名称 |
| platforms | String数组 | 操作系统列表 |

**请求示例：**

{  
 "osType": 1, //操作系统，必填 1-linux 2-windows  
 "specName": "test", //任务名  
 "ruleIds": ["1b8094cf2cacf0e42acf"], //规则id列表  
 "groups": [1, 446], //业务组列表  
 "ip": "xxx", //主机ip  
 "hostName": "xxx", //主机名  
 "platforms": ["centos-6"], //操作系统列表  
}

**返回示例：**

{  
 "specId": "591e9185629f9f196bba7a2b", // 任务规格Id  
 "jobId": "asdf", // 最后一次任务ID  
 "specName": "Test1", // 任务规格名  
 "descroption": "xxx", //任务描述  
 "ruleId": ""//规则ID  
 "ruleName": "规则２", // 基线规则名  
 "checkTime": "2017-03-01", // 检查时间  
 "duration": 0, //持续时间  
 "hostCount": 12043, // 主机数  
 "failedHostCount": 0, // 失败主机数  
 "successHostCount": 0, // 成功主机数  
 "progress": 0.2, // 进度  
 "passRate": 0.5, // 通过率  
 "status": 0, // 状态 0-Pending，1-Running，2-Done  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| specId | String | 任务Id |
| jobId | String | jobid |
| specName | String | 任务规格名 |
| descroption | String | 描述 |
| ruleId | String | 基线规则id |
| ruleName | String | 基线规则名 |
| checkTime | date | 检查时间 |
| duration | long | 持续时间 |
| hostCount | long | 主机数 |
| failedHostCount | long | 失败主机数 |
| successHostCount | long | 成功主机数 |
| progress | float | 进度 |
| passRate | float | 通过率 |
| status | int | 状态：0-Pending；1-Running；2-Done |

## 执行任务

**调用接口：**

POST /external/api/baseline/job/{linux，win}/execute

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| specId | String | 任务specId |

**请求示例：**

{  
 "specId": "xxxx" //任务specId  
}

**返回示例：**

{  
 "status": "success" //执行结果  
 "id": "xxxx" //jobId  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 任务id |
| status | String | 结果 |

## 查询任务执行任务

**调用接口：**参数

GET /external/api/baseline/job/{linux，win}/getStatus/{specId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| specId | String | 任务specId |

**请求示例：**

{  
 "specId": "xxxx" //任务specId  
}

**返回示例：**

{  
 "status": "success" //查询结果  
 "id": "xxxx" //jobId  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 任务id |
| status | String | 执行结果 |

## 查询基线规则

**调用接口：**

GET /external/api/baseline/spec/rule/{linux，win}

ps：检查项数据较多，最终汇总的数据量较大，接口性能受影响

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| osType | int | 操作系统，必填 1 linux 2 windows |
| family | inti | 基线类别，必填 1 系统基线 2 应用基线 |
| platform | String | 使用系统， 系统基线时过滤 |
| appId | String | 应用id， 应用基线时过滤 |

**请求示例：**

{  
 "osType": 1, //操作系统，必填 1 linux 2 windows  
 "family": 1, //基线类别，必填 1 系统基线 2 应用基线  
 "plaform": "ubuntu-16.04", //使用系统， 系统基线时过滤  
 "appId": "6871696328758afa" //应用id， 应用基线时过滤  
}

**返回示例：**

{  
 {  
 "id":"5fba7c27fc524b1a0831", // 规则id  
 "userRule":false,  
 "name":"CIS Centos 7 Level 2", // 规则名  
 "osTag":[  
 "centos-7"  
 ], // 适用平台  
 "appName":null, // 适用应用-应用名：应用基线规则  
 "appVersion":null, // 适用应用-应用版本集：应用基线规则  
 "family": 1, //基线类别，必填 1 系统基线 2 应用基线  
 "appRule":false, // 是否应用基线  
 "needAuthCount":0, // 应用基线需要授权的检查项  
 "checkTotal":10, // 基线规则包含的检查项条数  
 "rows":[  
 {  
 "id" : "0c1a041e6a24b10b3683", //检查项ID  
 "name" : "检查/etc/passwd 中的所有组在 /etc/group是否存在", //检查项名称  
 "category" : "用户和组设置", //类别  
 "description" : "随着时间的推移，系统管理错误和更改可能导致组在 /etc/passwd 中但不在 /etc/group 中定义。", //描述  
 "content" : "/etc/passwd 中的所有组在 /etc/group都应存在", //内容  
 "reference" : "", //引用信息  
 "needAuth" : false //是否需要授权  
 }  
 ]  
 }  
]

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 规则id |
| userRule | 布尔 | 是否是用户规则 |
| name | String | 规则名 |
| osTag | List | 适用平台 |
| appName | String | 适用应用-应用名：应用基线规则 |
| appVersion | List | 适用应用-应用版本集：应用基线规则 |
| family | int | 基线类别，必填 1 系统基线 2 应用基线 |
| appRule | 布尔 | 是否应用基线 |
| needAuthCount | long | 应用基线需要授权的检查项 |
| checkTotal | long | 基线规则包含的检查项条数 |
| rows |  | 检查项明细 |

## 查询检查结果

**调用接口：**

GET /external/api/baseline/spec/checkResult/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 本次job id，必填 |
| checkId | String | 检查项的checkId |
| group | inti | 业务组 |
| flag | inti | 检查状态 |
| ip | String | 主机ip |
| hostname | String | 主机名 |

**请求示例：**

{  
 "jobId"：本次job id，string 必填  
 "checkId"：检查项的checkId，string  
 "group"：业务组，int  
 "flag"：检查状态，0-未通过，1通过，-1失败，int  
 "ip"：主机ip，string  
 "hostname"：主机名，string  
}

**返回示例：**

{  
 "specId": xxx, //任务ID  
 "ruleId": xxx, //规则ID  
 "id": "检查结果的id",  
 "internalIp": "xxxx",  
 "exinternalIp": "xxxx",  
 "hostname": "xxxx",  
 "groupName": "xxxx",  
 "flag": -1失败, 0未通过,1通过  
 "code": 0,  
 "error": ""，  
 "title": "xxxx",  
 "category": "xxxx",  
 "content": "检查内容",  
 "result": "检查结果",  
 // 如果为应用基线为多实例返回，结果以列表形式填充在results中，result为空  
 "results" : [ {  
 "flag" : 0, //检查结果，0：不通过; 1：通过; -1: 检查失败  
 "version" : "3.4.10", //版本号  
 "pid" : 4222, //PID,进程id  
 "port" : 27017, //端口  
 "result" : "mongod进程不存在\r\n", //详细结果  
 "kb\_name" : "mongod", //应用名  
 "bin\_path" : "/usr/local/mongodb/bin/mongod", //运行文件  
 "conf\_path" : "/usr/local/mongodb/conf/mongodb.conf", //配置文件  
 "msg" : "错误信息", //错误信息，可能为空， 如"MySQLNotAuthed" -> "mysql缺少授权信息"  
 } ],  
 "description": "描述"，  
 "reference": "引用信息",  
 "level": 0, // 危险级别 Integer  
 "remedDes"："修复建议"  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| specId | String | 任务id |
| ruleId | String | 规则ID |
| id | String | 检查结果的id |
| internalIp | String | 内网IP |
| exinternalIp | String | 外网IP |
| hostname | String | 主机名 |
| groupName | String | 业务组 |
| hostTagList | List | 标签 |
| remark | String | 备注 |
| results |  | 检查明细 |
| category | String | 检查类型 |
| content | String | 检查内容 |
| description | String | 描述 |
| reference | String | 引用信息 |
| level | Integer | 风险级别 |
| remedDes | String | 修复建议 |

## 查询检失败主机

**调用接口：**

GET /external/api/baseline/spec/failedHost/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | jobid，必填 |
| ip | String | 主机ip |
| hostname | String | 主机名 |
| code | int数组 | 失败原因 |

**请求示例：**

{  
 "jobId": "59f6cddf7d761b2828add61b", //jobid，必填  
 "ip": "xxx", //主机ip  
 "hostname": "xxx", //主机名  
 "code": [1, 2] //失败原因  
}

**返回示例：**

{  
 "agentId" : "407c1dd87b1d7771", //agentID  
 "displayIp" : "192.168.50.131", //主机IP  
 "internalIp" : "192.168.50.131", //内网IP  
 "externalIp" : null, //外网IP  
 "hostname" : "ZYLI-WIN8SERVER", //主机名  
 "group":0,//业务组  
 "remark": "", //备注  
 "hostTagList": [], //标签  
 "code" : 1, //失败码  
 "description" : "主机不在线", //描述  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| agentId | String | agentID |
| displayIp | String | 主机IP |
| internalIp | String | 内网IP |
| externalIp | String | 外网IP |
| hostname | String | 主机名 |
| group | int | 业务组 |
| remark | String | 备注 |
| hostTagList | List | 标签 |
| code | int | 失败码 |
| description | String | 描述 |

## 登录请求

**调用接口：**

GET /external/api/baseline/auth/login/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| comId | String | 公司id |
| uuid | String | 用户id |
| password | String | 密码 |

**请求示例：**

{  
 //验证账号密码  
 "password": "xxx"  
}

**返回示例：**

{  
 "flag" : true,  
 "authToken" : "1da285b0062c4b0ca56f24dfce25b7f8"  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| flag | 布尔 | 状态 |
| authToken | String | 登陆产生的token,30min失效 |

## 新建授权请求

**调用接口：**

POST /external/api/baseline/auth/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| comId | String | 公司id |
| uuid | String | 用户id |
| osType | int | 必填，1-linux；2-windows |
| realmType | int | 主机范围：1-自定义，包括业务组、主机；0-全部主机，必填 |
| groups | int数组 | 业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| agentIds | String数组 | 主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| appName | String | 应用名, 必填,范围:MySQL,WebLogic |
| userName | String | 用户名, 必填 |
| password | String | 密码 |
| port | int | 端口 |
| params | MAP | 存放token |

**请求示例：**

{  
 "osType": 1, //1：linux； 2：windows， 必填  
 "realmType": 1, //主机范围，1：自定义，包括业务组、主机； 0：全部主机， 必填  
 "groups": [1], //业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空  
 "agentIds": ["xx1", "xx2"], //主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空  
 "appName": "MySQL", //应用名, 必填  
 "userName": "admin", //用户名, 必填  
 "password": "admin", //密码  
 "port": 3306, //端口  
 "params": {  
 "authToken": "1da285b0062c4b0ca56f24dfce25b7f8", //授权token，由前端传入，必填  
 }  
}

**返回示例：**

{  
 true/false  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| flag | 布尔 | 状态 |

## 更新授权结果

**调用接口：**

PUT /external/api/baseline/auth/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| comId | String | 公司id |
| uuid | String | 用户id |
| osType | int | 必填，1-linux；2-windows |
| realmType | int | 主机范围，1-自定义，包括业务组、主机；0-全部主机，必填 |
| groups | int数组 | 业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| agentIds | String数组 | 主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空 |
| appName | String | 应用名, 必填,范围:MySQL,WebLogic |
| userName | String | 用户名, 必填 |
| password | String | 密码 |
| id | String | 原授权id，必填 |
| port | int | 端口 |
| params | Map | 存放token |
| modifyPassword | 布尔值 | 当修改密码时，需指定modifyPassword＝true,其他情况可默认不填 |

**请求示例：**

{  
 "osType": 1, //1：linux； 2：windows， 必填  
 "realmType": 1, //主机范围，1：自定义，包括业务组、主机； 0：全部主机， 必填  
 "groups": [1], //业务组列表，如果realmType=1, 则agentIds和groups至少有一个不为空  
 "agentIds": ["xx1", "xx2"], //主机列表，如果realmType=1, 则agentIds和groups至少有一个不为空  
 "appName": "MySQL", //应用名, 必填  
 "userName": "admin", //用户名, 必填  
 "password": "admin", //密码  
 "id": "5ceb46b97d761b0bd9c44314", //原授权id，必填  
 "port": 3306, //端口  
 "params": { "authToken": "1da285b0062c4b0ca56f24dfce25b7f8", //授权token，由前端传入，必填  
 }，  
　　 "modifyPassword": true  
}

**返回示例：**

{  
 true/false  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| flag | 布尔 | 状态 |

## 删除授权请求

**调用接口：**

DELETE /external/api/baseline/auth/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| comId | String | 公司id |
| uuid | String | 用户id |
| osType | int | 1：linux； 2：windows， 必填 |
| ids | String数组 | id列表， 必填 |
| params | Map | 存放token |

**请求示例：**

{  
 "osType": 1, //1：linux； 2：windows， 必填  
 "ids": ["5a18eaab3171a748ec593837"], //id列表， 必填  
 "params": {  
 "authToken": "1da285b0062c4b0ca56f24dfce25b7f8", //授权token，由前端传入，必填  
 }  
}

**返回示例：**

{  
 true/false  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| flag | 布尔 | 状态 |

## 查询授权请求

**调用接口：**

GET /external/api/baseline/auth/list/{linux，win}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| comId | String | 公司id |
| uuid | String | 用户id |
| authToken | String | 授权token，由前端传入，必填 |
| osType | int | 1：linux； 2：windows |
| appName | String | 应用名,范围:MySQL,WebLogic |

**请求示例：**

{  
 "authToken": "1da285b0062c4b0ca56f24dfce25b7f8", //授权token，由前端传入，必填  
 "osType": 1, //1：linux； 2：windows  
 "appName": "MySQL" //应用名  
}

**返回示例：**

[  
 {  
 "id" : "5a18eaab3171a748ec593838",  
 "realmKind" : 1, //范围类型，1：主机维度；2：业务组；0：全部主机  
 "group" : null,  
 "groupName" : null,  
 "agentId" : "76dc78d64e7b379a",  
 "appName" : "MySQL",  
 "userName" : "admin",  
 "port" : 3306,  
 "internalIp" : "172.16.6.113",  
 "displayIp" : "172.16.6.113",  
 "connectionIp" : null,  
 "externalIp" : null,  
 "hostTagList":[  
 "id" : "",  
 "comId" : "",  
 "tagName" : "", //标签名  
 "tagColor" : "", //标签颜色  
 "tagDesc" :"" , //标签描述  
 "osType" : 1, //操作系统  
 "createTime" : "2017-03-01", //创建时间  
 "hostCount" : 0, //主机数量  
 "agentIds" : [] //id列表  
 ],  
 "remark": "",  
 "hostname": "localhost.localdomain"  
 }  
]

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | ObjectId |
| realmKind | int | 范围类型，1：主机维度；2：业务组；0：全部主机 |
| group | int | 业务组 |
| groupName | String | 业务组名称 |
| agentId | String | 主机id |
| appName | String | 应用名 |
| userName | String | 用户名 |
| port | int | 端口号 |
| displayIp | String | 主机IP |
| connectinIp | String | 连接IP |
| internalIp | String | 内网IP |
| externalIp | String | 外网IP |
| hostTagList | List | 标签 |
| remark | String | 备注 |
| hostname | String | 主机名称 |

**代码示例(python)：**

#! /usr/bin/env python  
# -\*- coding: utf-8 -\*-  
  
import requests  
  
host = "127.0.0.1"  
port = 80  
  
def app\_auth\_list():  
 url = "http://%s:%s/external/api/baseline/auth/list/linux?comId=1e46b07f93a6410172c4&uuid=1e46b07f93a6410172c4&authToken=557cc4d3cc45414d9438173a3cd1296c&osType=1&appName=WebLogic" % (host, port)  
  
 querystring = {"comId":"1e46b07f93a6410172c4","uuid":"1e46b07f93a6410172c4","authToken":"557cc4d3cc45414d9438173a3cd1296c","osType":"1","appName":"WebLogic"}  
  
 response = requests.request("GET", url, params=querystring)  
  
 print(response.text)

## 批量新建任务

**调用接口：**

POST /external/api/baseline/job/{multilinux，multiwin}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| name | String | 任务名，必填 |
| kind | int | 必填，范围类型 0-主机 1-业务组 2-全部 |
| values | String数组 | kind=0时必填，所选主机列表 |
| groups | int数组 | kind=1时必填，所选业务组列表 |
| ruleIds | String数组 | 必填，基线规则id列表 |
| cronEnabled | 布尔 | 必填，是否定时任务 true-定时任务 false-普通任务 |
| cron | String | cronEnabled=true必填，cron任务表达式 |
| description | String | 任务描述 |

**请求示例：**

{  
 "name": "test\_001", //必填，任务名  
 "kind": 1, //必填，范围类型 0-主机 1-业务组 2-全部  
 "values": ["407c1dd87b1d7771"], //kind=0时必填，所选主机列表  
 "groups": [1, 3, 446], //kind=1时必填，所选业务组列表  
 "ruleIds": ["1b8094cf2cacf0e42acf","1b8094cf2cacf0e42abc"] //必填，基线规则id列表  
 "cronEnabled": false, //必填，是否定时任务 true-定时任务 false-普通任务  
 "cron": "0 17 \* \* \*", //cronEnabled=true必填，cron任务表达式  
 "description": "xxx" //任务描述  
}

**返回示例：**

{  
 "status":"success",//任务创建状态  
 "ids": ["xxxx"] //成功创建的任务specId列表  
 "failedRuleIds": ["xxxx"] //创建失败的规则列表  
}

**异常情景**

ruleIds为空, 会抛出相应的异常及相关提示

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| status | String | 任务创建状态,全部创建失败时才算失败 |
| ids | String数组 | 任务specId列表 |
| failedRuleIds | String数组 | 创建失败的规则列表 |

# 系统审计日志查询

该接口用于查询系统的审计日志信息；

**调用接口：**

GET /external/api/system/audit

**请求参数：**

| 参数 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| eventName | String | 否 | 操作名称（模糊查询） |
| userName | String | 否 | 操作的帐号名（模糊查询） |

**请求示例：**

/external/api/system/audit?size=1

**返回示例：**

{  
 "rows": [  
 {  
 "eventId": "5bf4db32c0591c4ab22b6307",  
 "eventType": "新建",  
 "eventName": "Linux-资产清点-主机发现",  
 "eventSource": "Console",  
 "osType": 1,  
 "resourceFunction": "资产清点",  
 "resourceSubFunction": "主机资产",  
 "requestIp": "127.0.0.1",  
 "location": "本机地址",  
 "requestId": "362a33c0fc5a4d858ed22e1800ade08d",  
 "requestParam": "{\n \"comid\": \"59080851823593e1a80b\",\n \"uuid\": \"59080851823593e1a80b\",\n \"page\": 1,\n \"size\": 50,\n \"orders\": [\n {\n \"field\": \"agentStatus\",\n \"ascend\": true\n }\n ],\n \"filters\": [\n \"platform\",\n \"group\"\n ],\n \"charts\": [\n \"agentStatus\",\n \"platform\",\n \"group\"\n ]\n}",  
 "userName": "dev@xx.com",  
 "userType": 1,  
 "responseCode": 200,  
 "errorInfo": null,  
 "eventTime": 1542773554  
 }  
 ],  
 "total": 13,  
 "charts": {}  
}

**返回rows部分说明：**

| 字段 | 类型 | 长度 | 说明 |
| --- | --- | --- | --- |
| eventId | String | varchar(16) | 事件id |
| eventType | String | varchar(15) | 事件类型，包括新建、修改、删除、查看、认证、文件、执行 |
| eventName | String | varchar(15) | 操作名称 |
| eventSource | String | varchar(15) | 事件来源，包括Console、外部api、系统内部 |
| osType | Integer | tinyint(4) | 操作系统 1-linux 2-windows 0-通用 |
| resourceFunction | String | varchar(20) | 所属主功能 |
| resourceSubFunction | String | varchar(128) | 所属子功能 |
| requestIp | String | varchar(1024) | 请求ip |
| location | String | varchar(1024) | 请求来源区域 |
| requestId | String | varchar(512) | 请求id |
| requestParam | String | varchar(512) | 请求参数，为json字符串 |
| userName | String | varchar(512) | 操作帐号名 |
| userType | Integer | tinyint(4) | 操作帐号类型 1-主帐号 2-子帐号 |
| responseCode | Integer | tinyint(4) | 响应码 和http返回码相同 200-成功 |
| errorInfo | String | varchar(512) | 错误描述 |
| eventTime | Long | Long | 发生时间 为秒级时间戳格式 |

# 快速任务

## 获取快速任务检测项列表

**调用接口：**

GET /external/api/fastjob/task/list

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| osType | int | 1：linux； 2：windows (必填) |
| ids | List<String> | 任务id |
| name | String | 任务名 |
| categories | List<String> | 类别 |
| updatedTimeRange | Map<String, String> | 更新时间范围 |

**排序字段**

| 参数 | 说明 |
| --- | --- |
| name | 任务名 |
| category | 任务类别 |
| updatedTime | 更新时间 |

**请求示例：**

GET /external/api/fastjob/task/list

{  
 "osType": 1, //1：linux； 2：windows  
 "ids":["0478ee5024763edc6d3c"],  
 "name":"Weblogic",  
 "categories":["安全检测", "应急响应"],  
 "updatedTimeRange":{min: "2019-06-21 16:39:59", max: "2019-06-28 16:39:59"}  
}

**返回示例：**

{  
 "rows":[  
 {  
 "id": "0478ee5024763edc6d3c",  
 "name": "Weblogic < 10.3.6 'wls-wsat' XMLDecoder 反序列化漏洞(CVE-2017-10271)",  
"description": "漏洞描述： WebLogic是美国Oracle公司出品的一个application server，确切的说是一个基于JAVAEE架构的中间件，WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。 Weblogic的WLS Security组件对外提供webservice服务，其中使用了XMLDecoder来解析用户传入的XML数据，在解析的过程中出现反序列化漏洞，导致可执行任意命令。 检测过程： 1.首先获取weblogic的kb,进行遍历，获取其中的servers字段后，再进行一次遍历一遍获取服务器监听的ip和端口信息，将每一个server组合成一个新的数组 2.遍历每一个server，获取其中的proto(协议类型)、ip、port,其中先对ip地址进行处理(weblogic 10.3.4后为ipv6地址)，然后组合成url,并往url发送xml数据。根据当前时间戳生成一个文件名，xml数据的作用则是创建该文件。 3.遍历根据时间戳生成的文件名数组,判断其是否存在。如果文件存在，则上报存在漏洞。否则上报不存在漏洞 使用场景： 可通过该脚本进行快速识别哪些主机上存在Weblogic服务然后判断是否存在反序列化漏洞，并将存在漏洞的主机上报。",  
 "category": "安全检测",  
 "riskInfo": "注意： 漏洞检测结果基于已有Weblogic KB数据进行漏洞存在性判断，若您在是在刚刚修复完成漏洞后进行检测，则需要先更新数据依赖后再执行该检测脚本获取最新漏洞检测结果。",  
 "updatedTime": "2019-05-30 09:42:42"  
 }  
],  
 "total":1  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 快速任务id |
| name | String | 快速任务名称 |
| category | String | 快速任务类型 |
| description | String | 任务概述 |
| riskInfo | String | 风险提示 |
| updatedTime | date | 更新时间 |
| total | long | 总任务数 |

## 获取某一个任务的详细信息

GET /external/api/fastjob/task/{taskId}

**请求参数：**  
无

**请求示例：**

/external/api/fastjob/task/0478ee5024763edc6d3c

**返回示例：**

{  
 "id": "0478ee5024763edc6d3c",  
 "name": "Weblogic < 10.3.6 'wls-wsat' XMLDecoder 反序列化漏洞(CVE-2017-10271)",  
 "category": "安全检测",  
 "description": "漏洞描述：\r\nWebLogic是美国Oracle公司出品的一个application server，确切的说是一个基于JAVAEE架构的中间件，WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。\r\nWeblogic的WLS Security组件对外提供webservice服务，其中使用了XMLDecoder来解析用户传入的XML数据，在解析的过程中出现反序列化漏洞，导致可执行任意命令。\r\n\r\n检测过程：\r\n1.首先获取weblogic的kb,进行遍历，获取其中的servers字段后，再进行一次遍历一遍获取服务器监听的ip和端口信息，将每一个server组合成一个新的数组\r\n2.遍历每一个server，获取其中的proto(协议类型)、ip、port,其中先对ip地址进行处理(weblogic 10.3.4后为ipv6地址)，然后组合成url,并往url发送xml数据。根据当前时间戳生成一个文件名，xml数据的作用则是创建该文件。 \r\n3.遍历根据时间戳生成的文件名数组,判断其是否存在。如果文件存在，则上报存在漏洞。否则上报不存在漏洞 \r\n使用场景：\r\n可通过该脚本进行快速识别哪些主机上存在Weblogic服务然后判断是否存在反序列化漏洞，并将存在漏洞的主机上报。",  
 "riskInfo": "注意：\r\n漏洞检测结果基于已有Weblogic KB数据进行漏洞存在性判断，若您在是在刚刚修复完成漏洞后进行检测，则需要先更新数据依赖后再执行该检测脚本获取最新漏洞检测结果。",  
 "createdTime": "2019-05-30 09:42:42",  
 "updatedTime": "2019-05-30 09:42:42",  
 "inputSchemas": [],  
 "outputSchemas": []  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 快速任务id |
| name | String | 快速任务名称 |
| category | String | 快速任务类型 |
| description | String | 任务概述 |
| riskInfo | String | 风险提示 |
| createdTime | date | 创建时间 |
| updatedTime | date | 更新时间 |
| inputSchemas | Collection<TaskInputSchema> | 输入参数 |
| outputSchemas | Collection<TaskOutputSchema> | 输出参数 |

**TaskInputSchema说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| field | String | 参数字段名 |
| caption | String | 描述 |
| required | boolean | 是否必须 |
| defaultValue | String | 默认值 |
| name | String | 参数名称 |
| example | String | 参数示例 |

**TaskOutputSchema说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| field | String | 参数 |
| displayName | String | 参数显示名称 |
| displayOrder | int | 显示顺序 |
| caption | String | 描述 |
| example | String | 示例 |

## 创建一个快速作业

**调用接口：**

POST /external/api/fastjob/job

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| name | String | 作业名 |
| osType | Integer | 系统类型 1：linux; 2：windows (必填) |
| description | String | 作业描述 |
| realm | Realm | 作业范围 |
| realmName | String | 作业执行范围描述 |
| taskType | int | 任务类型 1. 系统内置任务, 2, 用户添加任务 |
| taskId | String | 任务id |
| taskParams | List<TaskParam> | 任务参数 |
| cron | String | cron表达式 |
| cronEnable | boolean | 是否开启定时执行 |

**Realm结构说明：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| type | tinyint | 范围，0:全部主机,type为0时，agents和groups不传; 1:主机范围，如果type=1, 则agents必传; 2:业务组范围，如果type=2,则groups必传 不传时为0全部主机 |
| agents | List<String> | 主机agentId数组 |
| groups | List<Integer> | 业务组数组 |

**TaskParam 说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| field | String | 参数名 |
| value | String | 参数值 |

**请求示例：**

POST /external/api/fastjob/job

{  
 "name":"testxxxx",  
 "osType":1,  
 "description":"description xxx",  
 "realm":{  
 "type":0  
 },  
 "realmName":"全部主机",  
 "taskType":1,  
 "taskId":"0478ee5024763edc6d3c",  
 "taskParams":[{}],  
 "cron":null,  
 "cronEnable":false  
}

**返回示例：**

{  
 "id": "5d1b568b67657c1b743cf33b",  
 "createdTime": "2019-07-02 21:05:11"  
}

**返回部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 作业id |
| createdTime | date | 创建时间 |

## 编辑作业

**调用接口：**

PUT /external/api/fastjob/job/{jobId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 作业id |
| name | String | 作业名 |
| osType | Integer | 系统类型(必填) |
| description | String | 作业描述 |
| realm | Realm | 作业范围 |
| realmName | String | 作业执行范围描述 |
| taskType | int | 任务类型 1. 系统内置任务, 2, 用户添加任务 |
| taskId | String | 任务id |
| taskParams | List<TaskParam> | 任务参数 |
| cron | String | cron表达式 |
| cronEnable | boolean | 是否开启定时执行 |

**请求示例：**

PUT /external/api/fastjob/job/5d1b568b67657c1b743cf33b

{  
 "name":"testxxxx",  
 "osType":1,  
 "description":"description xxx",  
 "realm":{  
 "type":0  
 },  
 "realmName":"全部主机",  
 "taskType":1,  
 "taskId":"0478ee5024763edc6d3c",  
 "taskParams":[],  
 "cron":null,  
 "cronEnable":false  
}

**返回示例：**

{  
 "id": "5d1b568b67657c1b743cf33b",  
 "updatedTime": "2019-07-03 11:04:41"  
}

**返回部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 作业id |
| updatedTime | date | 更新时间 |

## 删除作业

**调用接口：**

DELETE /external/api/fastjob/job/{jobId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| jobId | String | 作业id |

**请求示例：**

/external/api/fastjob/job/5d1b568b67657c1b743cf33b

**返回示例：**

{  
 "executeTime": "2019-07-03 11:46:21",  
 "flag": true  
}

**返回部分说明：**

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| executeTime | date | 执行时间 |
| flag | boolean | 是否执行成功 |

## 获取作业列表

**调用接口：**

GET /external/api/fastjob/job

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| osType | tinyint | 必填 |
| jobName | String | 作业名 |
| updateTimeRange | Map | 更新时间范围, 例如{min: "2019-06-21 16:39:59", max: "2019-06-28 16:39:59"} |
| cronEnables | List<Boolean> | 是否开启定时 |

**排序字段：**

| 参数 | 说明 |
| --- | --- |
| updatedTime | 更新时间 |
| name | 执行作业名 |
| cronEnable | 是否定时执行 |
| lastExecuteTime | 最后执行时间 |

**请求示例：**

/external/api/fastjob/job?osType=1

**返回示例：**

{  
 "rows": [  
 {  
 "id": "5d1c486167657c321a8c925f",  
 "name": "testxxxx",  
 "description": null,  
 "cronEnable": false,  
 "cron": null,  
 "lastExecuteTime": null,  
 "lastExecuteTaskRecordId": null,  
 "updatedTime": "2019-07-03 14:17:05",  
 "createdTime": "2019-07-03 14:17:05",  
 "realmName": null,  
 "realm": {  
 "type": 0,  
 "groups": null,  
 "agents": null,  
 "osTags": null  
 },  
 "task":{  
 "taskType": 1,  
 "taskId": "0478ee5024763edc6d3c",  
 "taskName": null,  
 "description": null,  
 "riskInfo": null,  
 "realm": null,  
 "params": [  
 ]  
 }  
 }  
 ],  
 "total": 1,  
 "charts": {}  
}

**返回部分说明：**

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| id | varchar(24) | 作业id |
| name | varchar(128) | 快速作业名 |
| description | varchar(256) | 作业描述 |
| cronEnable | boolean | 是否开启定时 |
| cron | varchar(128) | 定时表达式 |
| realm |  |  |
| updatedTime | date | 更新时间 |
| createdTime | date | 创建时间 |
| lastExecuteTime | date | 上次作业执行时间 |
| lastExecuteTaskRecordId | varchar(24) | 最近一次执行的task的执行记录id |
| task | Task | 快速作业对应的任务详情 |

**Task结构说明：**

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| taskType | tinyint | 任务类型 |
| taskId | String | 任务id |
| params | TaskParam | 与创建的时候传递的参数一致 |

## 立即执行某个作业

**调用接口：**

POST /external/api/fastjob/job/execute/{id}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 作业id |

**请求示例：**

POST /external/api/fastjob/job/execute/5d1c76e067657c0a04cfbf67

**返回示例：**

{  
 "id": "5d1d61c323e46813eb98f513"  
}

**返回部分说明：**

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 作业执行记录id |

## 查看作业执行列表

**调用接口：**

GET /external/api/fastjob/job/execute

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| osType | tinyint | 系统类别(必填) |
| jobName | String | 作业名 |
| startDateRange | map | 作业执行起始时间范围 |
| durations | List<Integer> | 执行时间范围 0/1/2/3/4/5, 1分钟内/1~3分钟内/3~5分钟内/5~10分钟内/10~30分钟内/超出30分钟 |
| statuses | List<Integer> | 扫描状态 1.准备执行, 2.正在执行, 3.执行成功, 4.执行失败 |

**排序字段：**

| 参数 | 说明 |
| --- | --- |
| startTime | 执行时间 |
| name | 执行名 |
| duration | 执行时间 |
| status | 执行结果 |

**请求示例：**

/external/api/fastjob/job/execute?osType=1

**返回示例：**

{  
 "rows": [  
 {  
 "id": "5d1da35425d89101991a6e32",  
 "jobId": "5d1da35425d89101991a6e31",  
 "name": "文件完整性校验-20190704145709",  
 "taskRecordId": "5d1da35425d89101991a6e33",  
 "realm": {  
 "type": 0,  
 "groups": [],  
 "agents": [],  
 "osTags": null  
 },  
 "cronEnable": false,  
 "cron": null,  
 "startTime": "2019-07-04 14:57:24",  
 "duration": 3,  
 "status": 3  
 },  
 {  
 "id": "5d136b6825d8915db28cabc0",  
 "jobId": null,  
 "name": "Weblogic反序列化漏洞(CVE-2018-2628)检测-20190625181834",  
 "taskRecordId": "5d136b6925d8915db28cabc1",  
 "realm": {  
 "type": 0,  
 "groups": null,  
 "agents": null,  
 "osTags": null  
 },  
 "cronEnable": false,  
 "cron": null,  
 "startTime": "2019-06-26 20:56:08",  
 "duration": 0,  
 "status": 1  
 },  
 {  
 "id": "5d11f51025d891243fb540cb",  
 "jobId": "5d11f50825d891243fb540c0",  
 "name": "Weblogic反序列化漏洞(CVE-2018-2628)检测-20190625181834",  
 "taskRecordId": "5d11f51025d891243fb540cc",  
 "realm": {  
 "type": 0,  
 "groups": [],  
 "agents": [],  
 "osTags": null  
 },  
 "cronEnable": false,  
 "cron": null,  
 "startTime": "2019-06-25 18:18:56",  
 "duration": 3,  
 "status": 3  
 },  
 {  
 "id": "5d11f50825d891243fb540c1",  
 "jobId": "5d11f50825d891243fb540c0",  
 "name": "Weblogic反序列化漏洞(CVE-2018-2628)检测-20190625181834",  
 "taskRecordId": "5d11f50825d891243fb540c2",  
 "realm": {  
 "type": 0,  
 "groups": [],  
 "agents": [],  
 "osTags": null  
 },  
 "cronEnable": false,  
 "cron": null,  
 "startTime": "2019-06-25 18:18:48",  
 "duration": 2,  
 "status": 3  
 },  
 {  
 "id": "5d11bbc325d891243fb4be21",  
 "jobId": "5d11bbc225d891243fb4be20",  
 "name": "rootkit快速任务检测-20190625141335",  
 "taskRecordId": "5d11bbc325d891243fb4be22",  
 "realm": {  
 "type": 1,  
 "groups": [],  
 "agents": [  
 "190a788a48150865",  
 "422a580c5470c865"  
 ],  
 "osTags": null  
 },  
 "cronEnable": false,  
 "cron": null,  
 "startTime": "2019-06-25 14:14:26",  
 "duration": 215,  
 "status": 3  
 },  
 {  
 "id": "5d11bb4325d891243fb4be06",  
 "jobId": "5d11bb4325d891243fb4be05",  
 "name": "Weblogic反序列化漏洞(CVE-2018-2628)检测-20190625141201",  
 "taskRecordId": "5d11bb4325d891243fb4be07",  
 "realm": {  
 "type": 0,  
 "groups": [],  
 "agents": [],  
 "osTags": null  
 },  
 "cronEnable": false,  
 "cron": null,  
 "startTime": "2019-06-25 14:12:19",  
 "duration": 4,  
 "status": 3  
 }  
 ],  
 "total": 6,  
 "charts": {}  
}

**返回部分说明：**

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| id | varchar(24) | 作业执行记录id |
| name | varchar(128) | 作业名 |
| taskRecordId | varchar(24) | 任务执行记录id |
| realm | Realm | 作业执行范围 |
| cronEnable | boolean | 是否开启定时 |
| cron | varchar(64) | cron定时表达式 |
| startTime | date | 作业开始时间 |
| duration | long | 作业扫描时间(单位s) |
| status | tinyint | 作业状态 1.准备执行, 2.正在执行, 3.执行成功, 4.执行失败 |

## 查看执行记录结果

**调用接口：**

GET /external/api/fastjob/job/task/result/{taskRecordId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| taskRecordId | String | 任务执行id |
| searchKeys | List<String> | searchKeys为field对应参数名素组,与searchValues顺序对应 |
| searchValues | List<String> | searchValues为数据值, 与searchKeys的顺序对应 |

**排序字段：**

无

**请求示例：**

/external/api/fastjob/job/task/result/5d1d61c423e46813eb98f514

**返回示例：**

{  
 "fields": [  
 {  
 "field": "host.displayIp",  
 "displayName": "主机IP"  
 },  
 {  
 "field": "host.agentId",  
 "displayName": "主机ID"  
 },  
 {  
 "field": "host.hostname",  
 "displayName": "主机名"  
 },  
 {  
 "field": "data.ip",  
 "displayName": "检测的目标主机ip"  
 },  
 {  
 "field": "data.port",  
 "displayName": "检测的目标主机端口（默认为7001）"  
 },  
 {  
 "field": "data.vuln",  
 "displayName": "检测结果"  
 }  
 ],  
 "rows": [  
 {  
 "data.vuln": "不存在weblogic CVE-2018-2628 反序列化漏洞",  
 "host.ip": "192.168.146.131",  
 "host.hostTagList": [  
 {  
 "id": "5d1ca89225d8915cedfee156",  
 "osType": 1,  
 "tagName": "server-4",  
 "tagColor": "#FF00FF"  
 }  
 ],  
 "host.assetLevel": 20,  
 "data.port": "7001",  
 "host.agentId": "f7869518b8e84865",  
 "host.hostname": "localhost.localdomain",  
 "host.hasDeleted": false,  
 "\_trimmed\_fields": [],  
 "host.agentStatus": 1,  
 "host.displayIp": "192.168.146.131",  
 "data.ip": "127.0.0.1"  
 },  
 {  
 "data.vuln": "不存在weblogic CVE-2018-2628 反序列化漏洞",  
 "host.ip": "192.168.167.132",  
 "host.hostTagList": [  
 {  
 "id": "5d1ca89225d8915cedfee156",  
 "osType": 1,  
 "tagName": "server-4",  
 "tagColor": "#FF00FF"  
 }  
 ],  
 "host.assetLevel": 20,  
 "data.port": "7001",  
 "host.agentId": "8ed75611c5014865",  
 "host.hostname": "localhost.localdomain",  
 "host.hasDeleted": false,  
 "\_trimmed\_fields": [],  
 "host.agentStatus": 1,  
 "host.displayIp": "192.168.167.132",  
 "data.ip": "127.0.0.1"  
 },  
 {  
 "data.vuln": "不存在weblogic CVE-2018-2628 反序列化漏洞",  
 "host.ip": "172.16.5.102",  
 "host.hostTagList": null,  
 "host.assetLevel": null,  
 "data.port": "7001",  
 "host.agentId": "3bf67e5c8d8e1865",  
 "host.hostname": "zbx.qingteng.cn",  
 "host.hasDeleted": true,  
 "\_trimmed\_fields": [],  
 "host.agentStatus": null,  
 "host.displayIp": "172.16.5.102",  
 "data.ip": "127.0.0.1"  
 },  
 {  
 "data.vuln": "不存在weblogic CVE-2018-2628 反序列化漏洞",  
 "host.ip": "192.168.146.136",  
 "host.hostTagList": null,  
 "host.assetLevel": null,  
 "data.port": "7001",  
 "host.agentId": "82f68fb82ce4d865",  
 "host.hostname": "localhost",  
 "host.hasDeleted": true,  
 "\_trimmed\_fields": [],  
 "host.agentStatus": null,  
 "host.displayIp": "192.168.146.136",  
 "data.ip": "127.0.0.1"  
 }  
 ],  
 "total": 4  
}

**rows字段说明：**

rows 为List<Map> 类型, 字段名可以根据fields中获取到

## 查看失败主机

**调用接口：**

GET /external/api/fastjob/job/task/error/{taskRecordId}

**请求参数：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| taskRecordId | String | 作业的task执行记录id |

**请求示例：**

POST /external/api/fastjob/job/task/error/5d1d61c423e46813eb98f514

**返回示例：**

{  
 "rows": [  
 {  
 "id": "5d11f50b25d891243fb540c8",  
 "agentId": "acd675759d37d865",  
 "displayIp": "192.168.135.132",  
 "internalIp": "192.168.135.132",  
 "externalIp": null,  
 "hostName": "localhost.localdomain",  
 "group": 1,  
 "groupName": "未分组主机",  
 "remark": "server组",  
 "hostTags": [  
 "server-4"  
 ],  
 "code": 1,  
 "error": "主机不在线"  
 },  
 {  
 "id": "5d11f50c25d891243fb540c9",  
 "agentId": "034a6e7b17558865",  
 "displayIp": "172.20.0.1",  
 "internalIp": "172.20.0.1",  
 "externalIp": null,  
 "hostName": "ubuntu",  
 "group": 3,  
 "groupName": "test",  
 "remark": "server组",  
 "hostTags": [  
 "server-6"  
 ],  
 "code": 1,  
 "error": "主机不在线"  
 }  
 ],  
 "total": 2,  
 "charts": {}  
}

**返回部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | varchar(24) | 记录id |
| agentId | varchar(24) | 主机id |
| displayIp | varchar(17) | 主机ip |
| internalIp | varchar(17) | 内网ip |
| externalIp | varchar(17) | 外网ip |
| hostName | varchar(64) | 主机名 |
| group | int(10) | 业务组id |
| groupName | varchar(64) | 业务组名 |
| remark | varchar(256) | 主机备注 |
| hostTags | List<String> | 主机标签 |
| code | tinyint | 错误码 |
| error | varchar(128) | 错误信息 |

# 主机发现

## 获取主机发现结果

**调用接口：**

GET /external/api/discoveredhost/list

**请求参数：**

无

**请求示例：**

GET /external/api/discoveredhost/list

**返回示例：**

{  
 "rows": [  
 {  
 "id": "5fa5219373f7a26f9cfaa5ba",  
 "macAddr": "00:50:56:E5:B5:D0",  
 "ipAddr": "172.16.5.102",  
 "deviceType": "host",  
 "osFamily": "Linux",  
 "agentId": "5f8fe67e3125215c",  
 "discoveryIp": "172.16.111.100",  
 "agentStatus": 0,  
 "scanMethods": [  
 "tcp\_syn",  
 "ping"  
 ],  
 "firstDiscoverTime": 1604657555,  
 "lastDiscoverTime": 1604657555,  
 "osType": 1  
 }  
 ],  
 "total": 19,  
 "charts": {}  
}

* 返回rows部分说明

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 被发现主机的id |
| macAddr | String | 被发现主机的mac地址 |
| ipAddr | String | 被发现主机的ip地址 |
| deviceType | String | 被发现主机的设备类型 |
| osFamily | String | 被发现主机的操作系统类型 |
| agentId | String | 执行发现任务的主机的agent id,16位 |
| discoveryIp | String | 执行发现任务的主机的ip地址 |
| agentStatus | Integer | 执行发现任务的主机的agent状态 0-在线；1-离线；2挂起；3-删除 |
| scanMethods | String | 发现该主机的扫描方法 |
| firstDiscoverTime | Integer | 第一次发现时间，时间戳，精确到秒 |
| lastDiscoverTime | Integer | 最近一次发现时间，时间戳，精确到秒 |
| osType | Integer | 执行发现任务的主机的操作系统 1-linux；2-windows |

## 扫描任务

### 创建扫描任务

**调用接口：**

POST /external/api/assetdiscovery/job/create

**请求参数：**

{  
 "name": "assetdiscovery\_job\_name", // 自定义扫描任务名  
 "kind": 0, // 任务发起主机类型 0：自定义主机；1：自定义业务组 2：全部主机（注意：只能选择 Linux 的主机作为扫描发起主机，建议选择两台及以上主机扫描整个网段）  
 "values": [ // 当kind=0时，valus传agentid；当kind=1时，valus传groupid  
 "5fcf47bf4891559a",  
 "5f8e5559b767de2f"  
 ],  
 "cronExpression": "0 15 \* \* \*", // 定时扫描cron表达式  
 "osDetection": true, // 是否获取操作系统 true/false  
 "ipList": [], // 扫描网段（选填）  
 "advanceConfigs": [ // 扫描方式设置，以下三种扫描方式至少选择一个  
 {  
 "scanType": 0 // ARP缓存方式扫描  
 },  
 {  
 "scanType": 1 // Ping方式扫描  
 },  
 {  
 "scanType": 2, // Nmap方式扫描，以下两种协议至少选择一个  
 "tcpPort": "80", // tcp协议扫描端口  
 "udpPort": "40125" //udp协议扫描端口  
 }  
 ],  
 "maxParallelism": 50, // 并发扫描最大数量（请填写整数）  
 "maxRate": 50, // 每秒最大发包数（请填写整数）  
 "taskInterval": 5.0 // 服务器下发任务间隔，不超过600（单位：秒，精确值小数点后一位）  
}

**返回示例：**

{  
 "id": "5fdacbf3edc90d7a292ae9a5" // 任务id  
}

### 删除扫描任务

**调用接口：**

POST /external/api/assetdiscovery/job/delete

**请求参数：**

{  
 "specId": "5fdacbf3edc90d7a292ae9a5" // 任务id  
}

**返回示例：**

{  
 "id": "5fdacbf3edc90d7a292ae9a5" // 被删除的任务id  
}

### 任务配置详情

**调用接口：**

POST /external/api/assetdiscovery/job/find

**请求参数：**

{  
 "specId": "5fdaca26398e682e95d9d3f4" // 任务id  
}

**返回示例：**

{  
 "id": "5fdaca26398e682e95d9d3f4",  
 "name": "my\_discovery\_job",  
 "kind": 2,  
 "values": [],  
 "osType": 1,  
 "cronExpression": "20 20 \* \* \*",  
 "osDetection": true,  
 "ipList": [],  
 "advanceConfigs": [  
 {  
 "scanType": 2,  
 "tcpPort": "80",  
 "udpPort": ""  
 },  
 {  
 "scanType": 1,  
 "tcpPort": null,  
 "udpPort": null  
 }  
 ],  
 "maxParallelism": null,  
 "maxRate": null,  
 "taskInterval": null,  
 "cronTaskId": 21  
}

### 修改扫描任务配置

**调用接口：**

POST /external/api/assetdiscovery/job/update

**请求参数：**

先通过 11.3 的 find 接口，查出任务的配置详情，再修改其中某些配置项后提交update

{  
 "specId": "5fdaca26398e682e95d9d3f4",  
 "name": "my\_discovery\_job\_update",  
 "kind": 2,  
 "values": [],  
 "osDetection": true,  
 "advanceConfigs": [  
 {  
 "scanType": 2,  
 "tcpPort": "",  
 "udpPort": ""  
 },  
 {  
 "scanType": 1  
 }  
 ],  
 "maxParallelism": 30,  
 "maxRate": null,  
 "taskInterval": 6.0,  
 "ipList": [],  
 "cronExpression": "10 00 \* \* \*"  
}

**返回示例：**

{  
 "id": "5fdaca26398e682e95d9d3f4" // 被修改的任务id  
}

### 查询任务列表

**调用接口：**

POST /external/api/assetdiscovery/job/list

**请求参数：**

筛选项都是选填字段，如果要列出所有的任务则以下字段传空值即可。

{  
 "name": "test\_7", // 任务名称，支持模糊查询  
 "scanType": [ // 扫描方式  
 1,  
 2  
 ]  
}

**返回示例：**

{  
 "rows": [  
 {  
 "specId": "5fcf34458780f401ba8d63e9",  
 "jobId": "5fd9fb70398e682e95d9ccde", // 执行任务的jobid  
 "name": "lix\_externalapi\_test\_7",  
 "progress": 1.0,  
 "kind": 2,  
 "values": [],  
 "scanType": [  
 2,  
 1  
 ],  
 "status": 2, // 任务状态 0：待执行 1：执行中 2：执行完成  
 "lastScanTime": 1608121200638 // 该任务上一次执行时间  
 }  
 ],  
 "total": 1, // 总条数  
 "charts": {}  
}

### 执行任务

**调用接口：**

POST /external/api/assetdiscovery/job/execute

**请求参数：**

{  
 "specId": "5fcf34458780f401ba8d63e9"  
}

**返回示例：**

{  
 "id": "5fd9fb70398e682e95d9ccde" // 返回执行任务的jobid  
}

# 微隔离一键隔离接口

## 查询一键隔离列表

**调用接口：**

GET /external/api/ms-srv/api/segmentation/list

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| ip | String | 否 | 主机ip |
| direction | List | 否 | 隔离方向,in:入站，out：出站，all：全部 |
| status | int数组 | 否 | 隔离状态,1：隔离中,2：隔离成功,3：隔离失败,4：修改中,5：修改失败,6：解除中,7：解除失败 |
| groups | int数组 | 否 | 业务组 |
| hostname | String | 否 | 主机名 |
| page | Integer | 否 | 分页的页数，默认0 |
| size | Integer | 否 | 分页的大小，默认50 |
| sorts | String | 否 | 排序字段,多个用逗号隔开,sorts字段的约定不写表示升序，"-"表示降序 |

**请求示例：**

GET /external/api/ms-srv/api/segmentation/list?page=0&size=50&groups=1,2&sorts=updatedTime,-hostname

**返回示例：**

{  
 "rows": [  
 {  
 "id": "61d7b2de27b7b26916736511",  
 "agentId": "5fa27259dae9af8a",  
 "updatedTime": 1641525982,  
 "remark": "123456",  
 "remoteLoginEnabled": null,  
 "displayIp": "192.168.253.128",  
 "displayIpBak": null,  
 "hostname": "localhost.localdomain",  
 "groupName": "未分组主机",  
 "onlineStatus": 0,  
 "ipList": [  
 "192.168.1.100",  
 "172.16.2.187"  
 ],  
 "portList": [  
 "80",  
 "81",  
 "82"  
 ],  
 "status": 3,  
 "operation": "micro\_segmentation\_add",  
 "error": "主机离线",  
 "direction": "out"  
 }  
 ],  
 "total": 1  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | id |
| agentId | String | 主机id |
| updatedTime | long | 修改时间 |
| remark | String | 备注 |
| remoteLoginEnabled | int | 开启远程登录 0-关闭，1-开启 |
| displayIp | String | 主机ip |
| displayIpBak | String | 根据ip搜索的匹配ip |
| hostname | String | 主机名 |
| groupName | String | 业务组名称 |
| onlineStatus | int | 在线状态，0：离线，1：在线 |
| ipList | List | 放行ip |
| portList | List | 放行端口 |
| status | int | 任务状态：1隔离中 2隔离成功 3隔离失败 4修改中 5修改失败 6解除中 7解除失败 |
| operation | String | 操作类型， micro\_segmentation\_add:新增， micro\_segmentation\_edit：修改 ，micro\_segmentation\_del：删除 |
| error | String | 错误描述 |
| direction | String | 隔离方向,in:入站，out：出站，all：全部 |

## 查询隔离详情

**调用接口：**

GET /external/api/ms-srv/api/segmentation/detail

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentId | String | 是 | 主机Id |

**请求示例：**

GET /external/api/ms-srv/api/segmentation/detail?agentId=5fa27259dae9af8a

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": {  
 "id": "61d7b2de27b7b26916736511",  
 "agentId": "5fa27259dae9af8a",  
 "comId": "513835545069574e666d",  
 "ipList": [  
 "192.168.1.100",  
 "172.16.2.187"  
 ],  
 "portList": [  
 "80",  
 "81",  
 "82"  
 ],  
 "protocol": "\*",  
 "updatedTime": 1641525982,  
 "remark": "123456",  
 "operation": "micro\_segmentation\_add",  
 "status": 3,  
 "error": "主机离线",  
 "direction": "out"  
 }  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |
| id | String | id |
| agentId | String | 主机id |
| comId | String | 公司id |
| ipList | List | 放行ip |
| portList | List | 放行端口 |
| protocol | String | 协议，\*表示tcp和udp都支持 |
| updatedTime | long | 修改时间 |
| remark | String | 备注 |
| status | int | 任务状态：1隔离中 2隔离成功 3隔离失败 4修改中 5修改失败 6解除中 7解除失败 |
| operation | String | 操作类型， micro\_segmentation\_add:新增， micro\_segmentation\_edit：修改 ，micro\_segmentation\_del：删除 |
| error | String | 错误描述 |
| direction | String | 隔离方向,in:入站，out：出站，all：全部 |

## 创建隔离接口

**调用接口：**

POST /external/api/ms-srv/api/segmentation/create

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 是 | 主机Id数组 |
| direction | String | 是 | 隔离方向,in:入站，out：出站，all：全部 |
| ipList | List | 是 | 放行ip |
| portList | List | 是 | 放行端口 |
| remark | String | 是 | 备注 |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
 "remark":"123456",  
 "direction":"out",  
 "ipList":["192.168.1.100","172.16.2.187"],  
 "portList":["80","81","82"]  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 修改隔离接口

**调用接口：**

POST /external/api/ms-srv/api/segmentation/edit

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 是 | 主机Id数组 |
| direction | String | 是 | 隔离方向,in:入站，out：出站，all：全部 |
| ipList | List | 是 | 放行ip |
| portList | List | 是 | 放行端口 |
| remark | String | 是 | 备注 |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
 "remark":"123456",  
 "direction":"out",  
 "ipList":["192.168.1.100","172.16.2.187"],  
 "portList":["80","81","82"]  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 解除隔离接口

**调用接口：**

DELETE /external/api/ms-srv/api/segmentation/del

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 是 | 主机Id数组 |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 删除隔离接口

**调用接口：**

DELETE /external/api/ms-srv/api/segmentation/realDel

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 是 | 主机Id数组 |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 重试隔离接口

**调用接口：**

POST /external/api/ms-srv/api/segmentation/retry

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 是 | 主机Id数组 |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

# 微隔离主机管理接口

## 查询主机列表

**调用接口：**

GET /external/api/ms-srv/api/hosts/list

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| ip | String | 否 | 主机ip |
| hostname | String | 否 | 主机名 |
| tags | List | 否 | 主机标签id |
| msEnabled | List | 否 | 微隔离插件开关：null全部 0关闭 1开启 |
| accessControlMode | List | 否 | 访问控制模式：null全部 0服务控制 1主机控制 |
| agentId | String | 否 | 主机id |
| agentIds | List | 否 | 主机id数组 |
| groupId | List | 否 | 业务组id |
| onlineStatus | List | 否 | 在线状态,0离线 1在线 |
| available | String | 否 | 是否可用: no 不可用 |
| runStatus | List | 否 | 运行状态，运行状态: warn告警 monitor监控 protect防护 |
| limitOut | Integer | 否 | 出站限制,0不允许出站 1允许出站 |
| platform | String | 否 | 操作系统 |
| runLevel | List | 否 | 运行级别：0正常 1降级 2停用 |
| page | Integer | 否 | 分页的页数，默认0 |
| size | Integer | 否 | 分页的大小，默认50 |
| sorts | String | 否 | 排序字段,多个用逗号隔开,sorts字段的约定不写表示升序，"-"表示降序 |
| blockingSwitch | Boolean | 否 | 拦截记录开关 |
| blockingState | Boolean | 否 | 拦截记录状态 |
| blackStrategyEnable | Boolean | 否 | 是否开启黑名单开关, true-是，false-否 |

**请求示例：**

GET /external/api/ms-srv/api/hosts/list?page=0&size=50&groups=1,2

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": {  
 "total": 1,  
 "rows": [{  
 "agentId": "63042b6fc7de70f2",  
 "onlineStatus": 1,  
 "displayIp": "10.0.2.15",  
 "displayIpBak": null,  
 "ips": ["192.168.56.105", "fe80::94c1:b476:6605:a0e6", "10.0.2.15", "fe80::4fd9:7a78:5b7c:e3ea"],  
 "hostname": "localhost.localdomain",  
 "group": 5,  
 "groupName": "licai",  
 "tags": null,  
 "msEnabled": 0,  
 "dcSyncNum": 0,  
 "dcNotSyncNum": 0,  
 "accessControlMode": null,  
 "platform": "CentOS Linux release 7.9.2009 (Core)",  
 "runStatus": null,  
 "protectService": null,  
 "limitOut": 0,  
 "blockingSwitch": false,  
 "checkStatus": "success",  
 "available": "yes",  
 "availableReason": null,  
 "osType": 1,  
 "runLevel": 0,  
 "updateUsername": null,  
 "updateTime": null,  
 "osPlatform": "linux",  
 "blackStrategyEnable": false  
 }],  
 "filters": {},  
 "remark": null  
 }  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| agentId | String | 主机id |
| onlineStatus | int | 在线状态,0离线 1在线 |
| displayIp | String | 主机IP |
| displayIpBak | String | 匹配ip |
| ips | List | 主机所有ip |
| hostname | String | 主机名 |
| group | int | 业务组id |
| groupName | String | 业务组名称 |
| tags | List | 主机标签 |
| msEnabled | int | 微隔离开关：0关闭 1开启 |
| dcSyncNum | long | 策略已同步数量 |
| dcNotSyncNum | long | 策略未同步状态 |
| accessControlMode | int | 访问控制模式: null未设置 0服务控制模式 1主机控制模式 |
| platform | String | 操作系统 |
| runStatus | String | 运行状态，运行状态: warn告警 monitor监控 protect防护 |
| protectService | List | 自定义防护服务 |
| limitOut | int | 出站限制,0不允许出站 1允许出站 |
| checkStatus | String | 检查状态 |
| availableReason | String | 不可用原因 |
| osType | String | 操作系统类型，1：linux， 2：windows |
| runLevel | int | 运行级别 ，0：整除，1：降级，2：停用，3：自动降级 |
| updateUsername | String | 操作人名称 |
| updateTime | long | 操作时间 |
| available | String | 是否是有效主机，yes：可用，no：不可用 |
| blockingSwitch | boolean | 拦截流量开关状态 |
| checkStatus | String | 拦截流量检查状态 |
| blackStrategyEnable | boolean | 黑名单策略开关状态 |

## 修改主机开关

**调用接口：**

POST /external/api/ms-srv/api/hosts/ms-enable

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 否 | 主机id数组 |
| enabled | Integer | 否 | 开启:1 关闭：0 |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
 "enabled":0  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 修改黑名单开关

**调用接口：**

POST /external/api/ms-srv/api/hosts/black-strategy-enable

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| agentIds | List | 否 | 主机id数组 |
| enabled | boolean | 否 | 开启:true 关闭：false |

**请求示例：**

{  
 "agentIds":["5fa27259dae9af8a"],  
 "enabled":true  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

# 微隔离黑名单策略接口

## 查询黑名单策略列表

**调用接口：**

GET /external/api/ms-srv/api/black-strategy/list

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 否 | 策略id |
| ids | String | 否 | 策略id集合 |
| strategyName | String | 否 | 策略名称 |
| dstGroupIds | List | 否 | 被访问业务组 |
| dstTagIds | List | 否 | 被访问主机标签 |
| dstIp | String | 否 | 被访问者ip |
| srcGroupIds | List | 否 | 访问业务组 |
| srcTagIds | List | 否 | 访问主机标签 |
| srcIp | String | 否 | 访问者ip |
| protos | List | 否 | 协议 |
| port | String | 否 | 端口 |
| switchStatuses | List | 否 | 开关状态，0：关闭，1：打开 |
| username | String | 否 | 操作人 |
| updateTime | TimeRange | 否 | 修改时间 |
| displayPort | String | 否 | 端口搜索字段 |
| page | Integer | 否 | 分页的页数，默认0 |
| size | Integer | 否 | 分页的大小，默认50 |
| sorts | String | 否 | 排序字段,多个用逗号隔开,sorts字段的约定不写表示升序，"-"表示降序 |

**TimeRange字段：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| start | String | 开始时间,例如：2022-06-01 15:47:52 |
| end | String | 结束时间 |

**请求示例：**

GET /external/api/ms-srv/api/black-strategy/list?page=0&size=50&groups=1,2

**返回示例：**

{  
 "data": {  
 "total": 1,  
 "rows": [{  
 "id": "629db182b4064413ce8b08bf",  
 "strategyName": "新建黑名单-20220606154903",  
 "dstRealmType": 1,  
 "dstGroups": [{  
 "groupId": 1,  
 "platform": 1,  
 "name": "未分组主机",  
 "description": "",  
 "parentId": 0  
 }],  
 "dstTags": [{  
 "id": null,  
 "tagName": "全部标签",  
 "tagColor": null  
 }],  
 "dstIpRanges": null,  
 "dstAgentSize": 2,  
 "dstAgentIds": null,  
 "dstIpName": "",  
 "srcRealmType": 1,  
 "srcGroups": [{  
 "groupId": 2,  
 "platform": 2,  
 "name": "未分组主机",  
 "description": "",  
 "parentId": 0  
 }],  
 "srcTags": [{  
 "id": null,  
 "tagName": "全部标签",  
 "tagColor": null  
 }],  
 "srcIpRanges": null,  
 "srcAgentSize": 0,  
 "srcAgentIds": null,  
 "srcIpName": "",  
 "ports": ["99"],  
 "protos": ["TCP"],  
 "status": 0,  
 "updateTime": 1654501762618,  
 "optUsername": "admin@sec.com",  
 "remark": "",  
 "displayDstIp": null,  
 "displaySrcIp": null,  
 "displayPort": "99",  
 "switchStatus": 0,  
 "operation": true  
 }],  
 "filters": {},  
 "remark": null  
 },  
 "success": false,  
 "errorCode": 0,  
 "errorDesc": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 策略id |
| strategyName | String | 策略名称 |
| dstRealmType | Integer | 被访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| dstGroups | List | 被访问主机业务组 |
| dstTags | List | 被访问主机标签 |
| dstIpRanges | List | 被访问者ip范围 |
| dstAgentSize | long | 被访问主机数量 |
| dstAgentIds | List | 被访问自定义主机范围 |
| dstIpName | String | 被访问主机ip |
| srcRealmType | Integer | 访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| srcGroups | List | 访问主机业务组 |
| srcTags | List | 访问主机标签 |
| srcIpRanges | List | 访问者ip范围 |
| srcAgentSize | long | 访问主机数量 |
| srcAgentIds | List | 访问自定义主机范围 |
| srcIpName | String | 访问主机ip |
| ports | List | 端口 |
| protos | List | 协议 |
| status | Integer | 策略状态,0：新建，1：待同步，2：已同步,3:删除 |
| updateTime | Long | 修改时间 |
| optUsername | String | 操作用户 |
| remark | String | 备注 |
| displayDstIp | String | 目标IP搜索字段 |
| displaySrcIp | String | 源IP搜索字段 |
| displayPort | String | 端口搜索字段 |
| switchStatus | Integer | 开关状态，0：关闭，1：打开 |
| operation | boolean | 是否有操作权限，true：有权限，false：无权限 |

**GroupInfo：**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| groupId | int | 业务组id |
| platform | int | 操作系统，1：linux，2：windows |
| name | String | 业务组名称 |
| description | String | 描述 |
| parentId | int | 父节点id |

**HostTagBO**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 标签id |
| tagName | String | 标签名称 |
| tagColor | String | 标签颜色 |

**IpRange**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| startIp | String | 开始ip |
| endIp | String | 结束ip |

## 创建黑名单策略

**调用接口：**

POST /external/api/ms-srv/api/black-strategy/create

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| strategyName | String | 是 | 策略名称 |
| dstRealmType | Integer | 是 | 被访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| dstGroups | List | 否 | 被访问主机业务组 |
| dstTags | List | 否 | 被访问主机标签 |
| dstIpRanges | List | 否 | 被访问者ip范围 |
| dstAgentIds | List | 否 | 被访问自定义主机范围 |
| srcRealmType | Integer | 是 | 访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| srcGroups | List | 否 | 访问主机业务组 |
| srcTags | List | 否 | 访问主机标签 |
| srcIpRanges | List | 否 | 访问者ip范围 |
| srcAgentIds | List | 否 | 访问自定义主机范围 |
| ports | List | 是 | 端口 |
| protos | List | 是 | 协议 |
| remark | String | 否 | 备注 |
| displayDstIp | String | 否 | 目标IP搜索字段 |
| displaySrcIp | String | 否 | 源IP搜索字段 |
| displayPort | String | 否 | 端口搜索字段 |
| switchStatus | Integer | 是 | 开关状态，0：关闭，1：打开 |

**IpRange**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| startIp | String | 开始ip |
| endIp | String | 结束ip |

**请求示例：**

{  
 "strategyName": "新建黑名单-20220606164812",  
 "remark": "",  
 "ports": ["7788"],  
 "protos": ["TCP"],  
 "displayPort": "7788",  
 "switchStatus": 0,  
 "dstTagIds": ["all"],  
 "dstGroupIds": [9],  
 "dstRealmType": 1,  
 "srcTagIds": ["629d90ceba7f06736d40f5a5", "629d90c6ba7f06736d40f5a4"],  
 "srcGroupIds": [2],  
 "srcRealmType": 1  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 修改策略

**调用接口：**

POST /external/api/ms-srv/api/black-strategy/update

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 策略id |
| strategyName | String | 是 | 策略名称 |
| dstRealmType | Integer | 是 | 被访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| dstGroups | List | 否 | 被访问主机业务组 |
| dstTags | List | 否 | 被访问主机标签 |
| dstIpRanges | List | 否 | 被访问者ip范围 |
| dstAgentIds | List | 否 | 被访问自定义主机范围 |
| srcRealmType | Integer | 是 | 访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| srcGroups | List | 否 | 访问主机业务组 |
| srcTags | List | 否 | 访问主机标签 |
| srcIpRanges | List | 否 | 访问者ip范围 |
| srcAgentIds | List | 否 | 访问自定义主机范围 |
| ports | List | 是 | 端口 |
| protos | List | 是 | 协议 |
| remark | String | 否 | 备注 |
| displayDstIp | String | 否 | 目标IP搜索字段 |
| displaySrcIp | String | 否 | 源IP搜索字段 |
| displayPort | String | 否 | 端口搜索字段 |
| switchStatus | Integer | 是 | 开关状态，0：关闭，1：打开 |

**IpRange**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| startIp | String | 开始ip |
| endIp | String | 结束ip |

**请求示例：**

{  
 "strategyName": "新建黑名单-20220606164812",  
 "remark": "",  
 "ports": ["7788"],  
 "protos": ["TCP"],  
 "displayPort": "7788",  
 "switchStatus": 0,  
 "dstTagIds": ["all"],  
 "dstGroupIds": [9],  
 "dstRealmType": 1,  
 "srcRealmType": 2,  
 "srcAgentIds": ["628f27265aa65cab", "628f272b5aa65cac"],  
 "id": "629dbf65c56dd50d10c9a984"  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 修改黑名单策略开关

**调用接口：**

POST /external/api/ms-srv/api/black-strategy/update-switch

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| ids | List | 是 | 策略id |
| switchStatus | Integer | 是 | 开启:1 关闭：0 |

**请求示例：**

{  
 "ids":["5fa27259dae9af8a"],  
 "switchStatus":0  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 删除策略接口

**调用接口：**

DELETE /external/api/ms-srv/api/black-strategy/delete

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| ids | List | 是 | 策略id |

**请求示例：**

{  
 "ids":["5fa27259dae9af8a"]  
}

**返回示例：**

{  
 "success": true,  
 "errorCode": 0,  
 "errorDesc": null,  
 "data": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| success | boolean | 是否成功，true：成功，false：失败 |
| errorCode | int | 状态码，0：成功，其他异常 |
| errorDesc | String | 异常描述 |

## 查看策略主机范围

**调用接口：**

GET /external/api/ms-srv/api/black-strategy/strategy-host-list

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| ip | String | 否 | 主机ip |
| msSwitchStatus | String | 否 | 微隔离插件状态，0-关闭 |
| strategyId | String | 是 | 策略id |
| strategyHostType | String | 是 | 查询数据类型，0:被访问者类型，1:访问者类型 |
| page | Integer | 否 | 分页的页数，默认0 |
| size | Integer | 否 | 分页的大小，默认50 |
| sorts | String | 否 | 排序字段,多个用逗号隔开,sorts字段的约定不写表示升序，"-"表示降序 |

**请求示例：**

{  
 "page": 1,  
 "size": 50,  
 "sorts":"ip",  
 "ip": "10",  
 "msSwitchStatus": 1,  
 "strategyId": "629dbf65c56dd50d10c9a984",  
 "strategyHostType": 0  
}

**返回示例：**

{  
 "data": {  
 "total": 1,  
 "rows": [{  
 "ip": "10.106.24.15",  
 "matchIp": null,  
 "hostname": "wgl15d",  
 "groupName": "l15a29a",  
 "tags": [{  
 "id": null,  
 "tagName": "a",  
 "tagColor": "#27C28E",  
 "osType": null,  
 "agentId": "628f270f5aa65ca9"  
 }],  
 "msSwitchStatus": 1,  
 "available": "yes",  
 "agentId": "628f270f5aa65ca9",  
 "status": null,  
 "osType": 1,  
 "onlineStatus": 1  
 }],  
 "filters": null,  
 "remark": false  
 },  
 "success": false,  
 "errorCode": 0,  
 "errorDesc": null  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| ip | String | 主机IP |
| matchIp | String | 匹配ip |
| hostname | String | 主机名 |
| groupName | String | 业务组名称 |
| tags | List | 主机标签 |
| agentId | String | 主机id |
| onlineStatus | int | 在线状态,0离线 1在线 |
| msSwitchStatus | int | 微隔离开关：0关闭 1开启 |
| available | String | 是否可用，yes：可用，no：不可用 |
| agentId | String | 主机id |
| status | Integer | 策略状态,0：新建，1：待同步，2：已同步,3:删除 |
| osType | String | 操作系统类型，1：linux， 2：windows |
| onlineStatus | int | 在线状态，0：离线，1：在线 |

**HostTagVO**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 标签id |
| tagName | String | 标签名称 |
| tagColor | String | 标签颜色 |
| osType | String | 操作系统类型，1：linux， 2：windows |
| agentId | String | 主机id |

## 查看策略详情

**调用接口：**

GET /external/api/ms-srv/api/black-strategy/detail

**请求参数：**

| 字段 | 类型 | 是否必填 | 说明 |
| --- | --- | --- | --- |
| id | String | 是 | 策略id |

**请求示例：**

/external/api/ms-srv/api/black-strategy/detail?id=111111

**返回示例：**

{  
 "id": "629dbf65c56dd50d10c9a984",  
 "strategyName": "新建黑名单-20220606164812",  
 "dstRealmType": 1,  
 "dstGroups": [{  
 "groupId": 9,  
 "platform": 1,  
 "name": "l15a29a",  
 "description": "l15a29a",  
 "parentId": 0  
 }],  
 "dstTags": [{  
 "id": null,  
 "tagName": "全部标签",  
 "tagColor": null  
 }],  
 "dstIpRanges": null,  
 "dstAgentSize": 2,  
 "dstAgentIds": null,  
 "dstIpName": "",  
 "srcRealmType": 2,  
 "srcGroups": [],  
 "srcTags": [],  
 "srcIpRanges": null,  
 "srcAgentSize": 2,  
 "srcAgentIds": ["628f27265aa65cab", "628f272b5aa65cac"],  
 "srcIpName": "10.106.24.32",  
 "ports": ["7788"],  
 "protos": ["TCP"],  
 "status": 1,  
 "updateTime": 1654505570177,  
 "optUsername": "jjw",  
 "remark": "",  
 "displayDstIp": null,  
 "displaySrcIp": null,  
 "displayPort": "7788",  
 "switchStatus": 0,  
 "operation": true  
}

**返回rows部分说明：**

| 参数 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 策略id |
| strategyName | String | 策略名称 |
| dstRealmType | Integer | 被访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| dstGroups | List | 被访问主机业务组 |
| dstTags | List | 被访问主机标签 |
| dstIpRanges | List | 被访问者ip范围 |
| dstAgentSize | long | 被访问主机数量 |
| dstAgentIds | List | 被访问自定义主机范围 |
| dstIpName | String | 被访问主机ip |
| srcRealmType | Integer | 访问者范围，1:业务组、标签，2:自定义主机，3：自定义ip |
| srcGroups | List | 访问主机业务组 |
| srcTags | List | 访问主机标签 |
| srcIpRanges | List | 访问者ip范围 |
| srcAgentSize | long | 访问主机数量 |
| srcAgentIds | List | 访问自定义主机范围 |
| srcIpName | String | 访问主机ip |
| ports | List | 端口 |
| protos | List | 协议 |
| status | Integer | 策略状态,0：新建，1：待同步，2：已同步,3:删除 |
| updateTime | Long | 修改时间 |
| optUsername | String | 操作用户 |
| remark | String | 备注 |
| displayDstIp | String | 目标IP搜索字段 |
| displaySrcIp | String | 源IP搜索字段 |
| displayPort | String | 端口搜索字段 |
| switchStatus | Integer | 开关状态，0：关闭，1：打开 |
| operation | boolean | 是否有操作权限，true：有权限，false：无权限 |

**GroupInfo**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| groupId | int | 业务组id |
| platform | int | 操作系统，1：linux，2：windows |
| name | String | 业务组名称 |
| description | String | 描述 |
| parentId | int | 父节点id |

**HostTagBO**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 标签id |
| tagName | String | 标签名称 |
| tagColor | String | 标签颜色 |

**IpRange**

| 字段 | 类型 | 说明 |
| --- | --- | --- |
| startIp | String | 开始ip |
| endIp | String | 结束ip |

# 附录1 Linux后门检测-查询结果-detail数据结构

**共同字段说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| access\_time | long |  | 文件的最近访问时间 |
| modify\_time | long |  | 文件内容的修改时间 |
| file\_md5 | String | varchar(50) | 文件md5 |
| file\_name | String | varchar(100) | 文件名称 |
| file\_size | long |  | 文件大小 |
| change\_time | long |  | 文件属性的修改时间 |
| file\_sha256 | String | varchar(50) | 文件sha256 |
| file\_sha1 | String | varchar(100) | 文件sha1 |
| file\_permission | String | varchar(50) | 文件访问权限 |
| file\_owner | String | varchar(50) | 文件所属用户 |
| file\_group | String | varchar(50) | 文件所属用户组 |

## 检查功能：DPKG-based应用后门检查、RPM-based应用后门检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| package\_name | String | varchar(50) | 包名，rpm：检查rpm包; dpkg：检查dpkg包 |
| package\_version | String | varchar(50) | 包版本 |
| package\_full\_name | String | varchar(50) | 包名全称 |
| package\_hash | String |  | 软件包备份的MD5 |
| file\_hash | String |  | 脚本扫描计算的MD5 |
| access\_time | long |  | 文件的最近访问时间 |
| modify\_time | long |  | 文件内容的修改时间 |
| file\_name | String | varchar(100) | 文件路径 |
| file\_size | long |  | 文件大小 |
| change\_time | long |  | 文件属性的修改时间 |
| file\_md5 | String | varchar(50) | 文件md5 |
| file\_sha256 | String | varchar(100) | 文件sha256 |
| file\_sha1 | String | varchar(100) | 文件sha1 |
| file\_permission | String | varchar(50) | 文件权限 |
| file\_owner | String | varchar(50) | 文件所属用户 |
| file\_group | String | varchar(50) | 文件所属组 |

**返回示例：**

{  
 "package\_full\_name": "procps\_1:3.3.9-1ubuntu2.3\_amd64",  
 "modify\_time": 1554124485,  
 "package\_name": "procps",  
 "package\_hash": "f482a1f4e8809875d0d1b93d5f02709e",  
 "change\_time": 1554124485,  
 "file\_sha256": "20c5ecccfd63cd5eeb701b34690ebadf1a532222105c572a2d38021fee8c1824",  
 "file\_sha1": "bf84d15e81b28cdc5c8227c8d261a43ccd5ad929",  
 "file\_name": "/usr/bin/top",  
 "access\_time": 1554083656,  
 "file\_md5": "f52296c7e3e42b2919e131f5af21e6e1",  
 "file\_permission": "rwxr-xr-x",  
 "package\_version": "1:3.3.9-1ubuntu2.3",  
 "file\_hash": "f52296c7e3e42b2919e131f5af21e6e1",  
 "file\_owner": "root",  
 "file\_size": 106832,  
 "file\_group": "root"  
}

## 检查功能：磁盘MBR检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| com\_id | String | varchar(50) | COM ID |
| pcid | String | varchar(50) | PC ID |
| dist\_ver | String | varchar(50) | linux发行版本 |
| grub\_ver | String | varchar(50) | grub版本 |
| ts | Integer | int(10) | 时间戳，精确到秒 |
| msg\_ver | String | varchar(50) | 上报消息版本 |
| boot\_mbr | String | varchar(300) | 上磁盘的MBR |
| grub\_mbr | String | varchar(300) | GRUB备份的MBR |
| single\_mbr | boolean |  | 是否磁盘有多份MBR |

**返回示例：**

{  
 com\_id": "a85acd72128679169a0e",  
 "ts": 1533631489,  
 "boot\_mbr": "",  
 "pcid": "1cc64150a44d57c4",  
 "dist\_ver": "CentOS release 6.9 (Final)",  
 "grub\_mbr": "",  
 "grub\_ver": "0.97",  
 "single\_mbr": true,  
 "msg\_ver": "bootkit 1.0.0"  
}

## 检查功能：动态链接库检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| cmd | String | varchar(50) | **命令** |
| before\_unset\_output | String | varchar(50) | 安装前路径 |
| after\_unset\_output | String | varchar(50) | 安装后路径 |
| before\_compare\_after | String | varchar(50) | 数据的对比 |

**返回示例：**

{  
"ld\_library\_path": {  
 "ret\_msg": [  
 {  
 "cmd": "/usr/bin/find",  
 "after\_unset\_output":{},  
 "before\_unset\_output":{},  
 "before\_compare\_after":{}  
 }  
 ],  
 "ret\_code": 0  
}  
}

## 检查功能：基本命令检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| match\_rules | String | varchar(50) | 命令命中的规则 |
| access\_time | long |  | 文件的最近访问时间 |
| modify\_time | long |  | 文件内容的修改时间 |
| file\_md5 | String | varchar(50) | 文件md5 |
| file\_name | String | varchar(100) | 命令 |
| file\_size | long |  | 文件大小 |
| change\_time | long |  | 文件属性的修改时间 |
| file\_sha256 | String | varchar(50) | 文件sha256 |
| file\_sha1 | String | varchar(100) | 文件sha1 |
| file\_permission | String | varchar(50) | 文件访问权限 |
| file\_owner | String | varchar(50) | 文件所属用户 |
| file\_group | String | varchar(50) | 文件所属用户组 |

**返回示例：**

{  
 "match\_rules":{  
 "175":".\*"  
 },  
 "file\_sha256":"e11183e4d33672e548804659841f3b013f9f366312af9fb85afbba458608e41c",  
 "file\_sha1":"0e25541ac9b5096df961e3532a84bddde2bd75c5",  
 "file\_name":"/bin/pwd",  
 "access\_time":1550569446,  
 "file\_md5":"1f30f796aa5218cdca9645990020752c",  
 "file\_permission":"rwxr-xr-x",  
 "modify\_time":1548210801,  
 "change\_time":1548210801,  
 "file\_owner":"root",  
 "file\_size":31472,  
 "file\_group":"root"  
}

## 检查功能：已知rootkit检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| rootkit\_name | String | varchar(50) | rootkit名称 |
| check\_type | String | varchar(20) | 检查类型,用来区分是文件、文件夹、内核符号 |
| extra\_info | String | varchar(20) | 仅当检查内核符号时可能有值,表示内核符号 |
| access\_time | long |  | 文件的最近访问时间 |
| modify\_time | long |  | 文件内容的修改时间 |
| file\_md5 | String | varchar(50) | 如果是文件，那么是整个文件内容计算MD5；如果是文件夹，是空字符串 |
| file\_name | String | varchar(100) | 文件路径 |
| file\_size | long |  | 文件大小 |
| change\_time | long |  | 文件属性的修改时间 |
| file\_owner | String | varchar(50) | 文件所属用户 |
| file\_group | String | varchar(50) | 文件所属用户组 |
| file\_permission | String | varchar(50) | 文件访问权限 |
| file\_sha1 | String | varchar(50) | 文件SHA1 |
| file\_sha256 | String | varchar(50) | 文件SHA256 |
| level | int |  | 1:中危 ,2:高危 |
| rule | String | varchar(1024) | 命中的crontab内容 |
| file | String | varchar(100) | 命中crontab内容的文件 |

**返回示例：**

"detail":{  
 "extra\_info":"",  
 "file\_sha256":"",  
 "modify\_time":1550564341,  
 "file\_md5":"",  
 "file\_sha1":"",  
 "check\_type":"dirs",  
 "access\_time":1550564341,  
 "rootkit\_name":"ddgs3011 rootkit",  
 "file\_permission":"rwx-wx---",  
 "change\_time":1550564341,  
 "file\_name":"/var/spool/cron/crontabs",  
 "file\_owner":"root",  
 "file\_size":4096,  
 "file\_group":"crontab"  
}  
已知rootkit二期  
"detail": "{\"rootkit\_name\":\"Fuckit Rootkit\",\"files\":[{\"extra\_info\":\"\",\"file\_sha256\":\"6ea43b2ef985f10fbbebf1aff9c302e1f448395bbf548aed1c1511d03aa5a72d\",\"file\_md5\":\"91f94b1ccffbd4381f86eaaf98fcb4b2\",\"file\_sha1\":\"d1e0248ea42947ae405a9613f51d30fd8dc22074\",\"check\_type\":\"core\",\"access\_time\":1554717866,\"modify\_time\":1554717797,\"file\_permission\":\"rw - r--r--\",\"change\_time\":1554717866,\"file\_name\":\"/etc/1\",\"file\_owner\":\"root\",\"file\_size\":5,\"file\_group\":\"root\"}],\"level\":3,\"crontab\":[{\"rule\":\"abcd\",\"file\":\"/etc/hosfilt\"}]}"

## 检查功能：系统内核模块检查

### 缺失模块检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| add | List | varchar(1024) | 多余模块(lsmod相对/proc/modules) |
| lack | List | varchar(1024) | 缺失模块(lsmod相对/proc/modules) |

**返回示例：**

{  
 "add": [],  
 "lack": [  
 "udp\_diag",  
 "binfmt\_misc",  
 "tcp\_diag",  
 "inet\_diag",  
 "bnep",  
 "bluetooth"  
 ]  
}

### 恶意内核模块

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| malicious\_name | String | varchar(64) | 恶意模块名称 |
| file\_sha256 | String | varchar(128) |  |
| file\_sha1 | String | varchar(128) |  |
| file\_name | String | varchar(128) | 文件路径 |
| access\_time | long |  | 最近访问时间 |
| file\_md5 | String | varchar(128) |  |
| file\_permission | String | varchar(128) |  |
| modify\_time | long |  | 文件内容修改时间 |
| change\_time | long | 文件属性修改时间 |  |
| file\_owner | String | varchar(32) |  |
| file\_size | long |  |  |
| file\_group | String | varchar(32) |  |

**返回示例：**

{  
 "malicious\_name":"vmci.ko",  
 "file\_sha256":"798bea1cc8a7241df97abb1b4e85ab94b98e4a3a100757498008bf24fd831c8b",  
 "file\_sha1":"f8bc6aa9a6409c67881c250057adc44e3246a52e",  
 "file\_name":"/lib/modules/2.6.32-754.3.5.el6.x86\_64/misc/vmci.ko",  
 "access\_time":1536084007,  
 "file\_md5":"2cc8051c7d9d2bc0ed5e3546f8701e47",  
 "file\_permission":"rw-r--r--",  
 "modify\_time":1535886506,  
 "change\_time":1535886506,  
 "file\_owner":"root",  
 "file\_size":131432,  
 "file\_group":"root"  
}

## 检查功能：计划任务检查 \ 网络状态检查\用户状态检查系统文件状态检查\进程状态检查

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| cmd | String | varchar(50) | 检查的命令 |
| count | Integer | int(6) | 被几种检查方式检查出问题 |
| match\_rules | String | varchar(50) | 匹配的规则，是字符串模式的结果 |
| local\_more | List | varchar(128) | 运行命令的diff |
| local\_less | List | varchar(128) | 运行命令的diff |
| pkg | Object |  | 包完整性结果 |

**pkg数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| package\_full\_name | String | varchar(128) | 完整包名 |
| package\_name | String | varchar(128) | 简单包名 |
| package\_version | String | varchar(50) | 包版本 |
| files | Object |  | 涉及的文件列表 |

**file数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| file\_name | String | varchar(128) | 文件名 |
| modify\_time | Integer | int(10) | 修改时间，时间戳，精确到秒 |
| permissions | String | varchar(50) | 权限 |
| package\_hash | String | varchar(50) | 包hash值 |
| file\_hash | String | varchar(50) | 文件hash值 |

**返回示例：**

{  
 "cmd": "ls",  
 "count": 2,  
 "match\_rules": {  
 "7": "uname -a"  
 },  
 "local\_more": [],  
 "local\_less": [],  
 "pkg": {  
 "package\_full\_name": "coreutils-8.4-46.el6.x86\_64",  
 "package\_name": " coreutils ",  
 "package\_version": "8.4",  
 "files": [  
 {  
 "file\_name": "/bin/ls",  
 "modify\_time": 1394048072,  
 "permissions": "rwxr-xr-x",  
 "package\_hash": "",  
 "file\_hash": ""  
 }  
 ]  
 }  
}

## 恶意进程

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| virusName | String | varchar(50) | 病毒名称 |
| virusDescription | String | varchar(1024) | 说明 |
| libUpdateTime | Integer | int(10) | 更新时间，时间戳，精确到秒 |
| libName | String | varchar(50) | 检测库 |
| virusInstruct | String | varchar(1024) | 修复方法 |

**静态信息数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| path | String | varchar(1024) | 文件路径 |
| size | Long |  | 文件大小 |
| type | String | varchar(50) | 文件类型 |
| user | String | varchar(50) | 所属用户 |
| group | String | varchar(50) | 所属组 |
| permission | String | varchar(50) | 文件权限 |
| changeTime | Integer | int(10) | 文件change time |
| modifyTime | Integer | int(10) | 文件change time |
| accessTime | Integer | int(10) | 文件访问时间 |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |
| codeSectionMd5 | String | varchar(50) | 代码段md5 |
| codeSectionSha1 | String | varchar(50) | 代码段sha1 |
| codeSectionSha256 | String | varchar(50) | 代码段sha256 |

**进程信息数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| uname | String | varchar(50) | 进程用户 |
| processPath | String | varchar(1024) | 进程对应执行文件路径 |
| parentProcessName | String | varchar(50) | 父进程名 |
| parentPath | String | varchar(1024) | 父进程路径 |

**返回示例：**

data:  
{ \"uname\" : \"root\", \"processPath\" : \"/bin/ps\", \"parentProcessName\" : \"bash\", \"parentPath\" : \"/bin/bash\", \"path\" : \"/bin/ps\", \"size\" : 1223123, \"type\" : \"regular file\", \"user\" : \"root\", \"group\" : \"root\", \"permission\" : \"0755\", \"changeTime\" : 1545633340, \"modifyTime\" : 1545633340, \"accessTime\" : 1545806233, \"md5\" : \"1e6237ef30bc132155cdc3a39ad29d71\", \"sha1\" : \"9e0fc9af130255fdd42d70ca9aea4dfff8a3b46b\", \"sha256\" : \"d63afb1cdaa6b354c4dbafc75e1b146e1d79152f58d725b3208070282d4a70b8\", \"codeSectionMd5\" : null, \"codeSectionSha1\" : null, \"codeSectionSha256\" : null, \"ruleResults\" : [ { \"virusName\" : \"LINUX/Setag.ztrec\", \"virusDescription\" : \"Contains detection pattern of the Linux virus LINUX/Setag.ztrec\", \"libUpdateTime\" : 1534953600, \"libName\" : \"Avira\", \"virusInstruct\" : null }, { \"virusName\" : \"Unix.Malware.Agent-1407158\", \"virusDescription\" : null, \"libUpdateTime\" : 1537169743, \"libName\" : \"clamAV\", \"virusInstruct\" : null }, { \"virusName\" : \"tezheng\", \"virusDescription\" : \"\", \"libUpdateTime\" : 0, \"libName\" : \"青藤库\", \"virusInstruct\" : null } ]}"

# 附录2 Windows后门检测-查询结果-data数据结构

## 后门类型：可疑进程

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| ret\_code | String | varchar(20) | 返回code |
| ret\_msg | String | varchar(128) | 返回message |
| parent | Object |  | 父进程，见[item数据结构](#item数据结构) |
| item | Object |  | 当前进程，见[item数据结构](#item数据结构) |

### item数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| process\_name | String | varchar(50) | 进程名 |
| signature | String | varchar(128) | 签名 |
| parent\_process\_id | Integer | int(10) | 父进程ID |
| create\_time | Integer | int(10) | 创建时间，时间戳，精确到秒 |
| b\_wow64 | Boolean | tinyint(1) | 32位进程在64位系统标志 |
| process\_id | Integer | int(10) | 进程ID |
| b\_64bit | Boolean | tinyint(1) | 64进程标志 |
| image\_file\_name | String | varchar(128) | 进程映像文件的全路径 |
| parent\_process\_name | String | varchar(50) | 父进程名 |
| f\_size | Integer | int(10) | 文件大小 |
| f\_user | String | varchar(128) | 文件所有者 |
| f\_type | String | varchar(50) | 32/64/空 表示 32位文件，64位文件，空未知 |
| f\_ctime | Integer | int(10) | 文件create time |
| f\_mtime | Integer | int(10) | 文件modify time |
| f\_atime | Integer | int(10) | 文件access time |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |
| code\_section\_md5 | String | varchar(50) | 代码段MD5 |
| code\_section\_sha1 | String | varchar(50) | 代码段sha1 |
| code\_section\_sha256 | String | varchar(50) | 代码段sha256 |
| black\_id | Integer | int(10) | 黑名单id (默认0) |

**返回示例：**

{  
 "ret\_code": "0x00000006",  
 "ret\_msg": "bogus os process",  
 "parent": {  
 "process\_name": "explorer.exe",  
 "signature": "Microsoft Windows",  
 "parent\_process\_id": 1232,  
 "create\_time": 1527214398,  
 "b\_wow64": false,  
 "process\_id": 1828,  
 "b\_64bit": true,  
 "image\_file\_name": "C:\\Windows\\explorer.exe",  
 "parent\_process\_name": ""，f\_size=100,  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0,  
 },  
 "item": {  
 "process\_name": "iexplore.exe",  
 "signature": "Microsoft Corporation",  
 "parent\_process\_id": 1828,  
 "create\_time": 1527218500,  
 "b\_wow64": false,  
 "process\_id": 2168,  
 "b\_64bit": true,  
 "image\_file\_name": "C:\\Program Files\\Internet Explorer\\iexplore.exe",  
 "parent\_process\_name": "explorer.exe",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 }  
}

## 后门类型：可疑线程

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| ret\_code | String | varchar(50) | 返回code |
| thread\_info | Object |  | 线程信息，见[thread\_info数据结构](#threadinfo数据结构) |
| process\_info | Object |  | 进程信息，见[process\_info数据结构](#processinfo数据结构) |
| module\_info | Object |  | 模块信息，见[module\_info数据结构](#moduleinfo数据结构) |

### thread\_info数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| thread\_id | Integer | int(10) | 线程ID |
| create\_time | Integer | int(10) | 创建时间，时间戳，精确到秒 |
| wait\_reason | Integer | int(10) | 等待原因 |
| state | Integer | int(10) | 线程状态，0:初始化 1:就绪 2:运行 3:备用 4:终止 5:等待 6:转换 7:延迟就绪 其他值:未知状态 |
| start\_address | Integer | int(15) | 开始地址 |
| start\_address\_str | String | varchar(20) | 开始地址 |
| mem\_protect | List | varchar(20) | 内存保护 |
| mem\_type | String | varchar(50) | 内存类型 |

### process\_info数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| process\_id | Integer | int(10) | 进程ID |
| process\_name | String | varchar(50) | 进程名 |
| parent\_process\_id | Integer | int(10) | 父进程ID |
| parent\_process\_name | String | varchar(50) | 父进程名 |
| image\_file\_name | String | varchar(128) | 进程映像文件的全路径 |
| create\_time | Integer | int(10) | 创建时间，时间戳，精确到秒 |
| b\_64bit | Boolean | tinyint(1) | 64进程标志 |
| b\_wow64 | Boolean | tinyint(1) | 32位进程在64位系统标志 |
| signature | String | varchar(50) | 签名 |
| f\_size | Integer | int(10) | 文件大小 |
| f\_user | String | varchar(128) | 文件所有者 |
| f\_type | String | varchar(50) | 32/64/空 表示 32位文件，64位文件，空未知 |
| f\_ctime | Integer | int(10) | 文件create time |
| f\_mtime | Integer | int(10) | 文件modify time |
| f\_atime | Integer | int(10) | 文件access time |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |
| code\_section\_md5 | String | varchar(50) | 代码段MD5 |
| code\_section\_sha1 | String | varchar(50) | 代码段sha1 |
| code\_section\_sha256 | String | varchar(50) | 代码段sha256 |
| black\_id | Integer | int(10) | 黑名单id (默认0) |

### module\_info数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| module\_name | String | varchar(50) | 模块名 |
| module\_path | String | varchar(128) | 模块路径 |
| module\_size | Integer | int(15) | 模块大小 |
| module\_md5 | String | varchar(20) | 模块md5值 |
| module\_base | Integer | int(15) | 模块基址 |
| module\_base\_str | String | varchar(20) | 模块基址 |
| module\_type | Integer | tinyint(2) | 模块类型 0 普通模块 1 普通WOW64模块 2 内核模块 3 映像映射 4 文件映射 |
| load\_time | Integer | int(10) | 加载时间，时间戳，精确到秒（需要win8以上版本系统，且驱动模块无效） |
| load\_reason | Integer | tinyint(2) | 加载原因（需要win8以上版本系统，且驱动模块无效） 0 静态依赖 1 静态代理依赖 2 动态代理依赖 3 动态依赖 4 动态加载 5 作为映像加载 6 作为数据加载 其他值：未知原因 |
| signature | String | varchar(128) | 签名 |
| f\_size | Integer | int(10) | 文件大小 |
| f\_user | String | varchar(128) | 文件所有者 |
| f\_type | String | varchar(50) | 32/64/空 表示 32位文件，64位文件，空未知 |
| f\_ctime | Integer | int(10) | 文件create time |
| f\_mtime | Integer | int(10) | 文件modify time |
| f\_atime | Integer | int(10) | 文件access time |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |
| code\_section\_md5 | String | varchar(50) | 代码段MD5 |
| code\_section\_sha1 | String | varchar(50) | 代码段sha1 |
| code\_section\_sha256 | String | varchar(50) | 代码段sha256 |
| black\_id | Integer | int(10) | 黑名单id (默认0) |

**返回示例：**

{  
 "module\_info": {  
 "module\_path": "C:\\Program Files (x86)\\Internet Explorer\\iexplore.exe",  
 "signature": "Microsoft Corporation",  
 "module\_base\_str": "0x1150000",  
 "load\_time": 1527218501,  
 "module\_size": 819200,  
 "module\_type": 2,  
 "module\_md5": "5f1b1148c830c0f149a476a58ce0d09d",  
 "load\_reason": 0,  
 "module\_base": 18153472,  
 "module\_name": "iexplore.exe"  
 },  
 "thread\_info": {  
 "start\_address": 1962227952,  
 "create\_time": 1527218503,  
 "wait\_reason": 6,  
 "mem\_type": "MEM\_IMAGE",  
 "state": 5,  
 "thread\_id": 1860,  
 "mem\_protect": [  
 "EXECUTE\_READ"  
 ],  
 "start\_address\_str": "0x74F538F0",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "process\_info": {  
 "process\_name": "iexplore.exe",  
 "signature": "Microsoft Corporation",  
 "parent\_process\_id": 2168,  
 "create\_time": 1527218501,  
 "b\_wow64": true,  
 "process\_id": 2736,  
 "b\_64bit": false,  
 "image\_file\_name": "C:\\Program Files (x86)\\Internet Explorer\\iexplore.exe",  
 "parent\_process\_name": "iexplore.exe",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "ret\_code": "0x00000010"  
}

## 后门类型：可疑模块

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| ret\_code | String | varchar(50) | 返回code |
| extend\_ret\_code | String | varchar(50) | 返回扩展code |
| process\_info | Object |  | 进程信息，[process\_info数据结构](#processinfo数据结构) |
| module\_info | Object |  | 模块信息，[module\_info数据结构](#moduleinfo数据结构) |

**返回示例：**

{  
 "ret\_code": "0x00000100",  
 "module\_info": {  
 "module\_path": "C:\\Windows\\SysWOW64\\ieframe.dll",  
 "signature": "",  
 "module\_base\_str": "0x737D0000",  
 "load\_time": 1527218501,  
 "module\_size": 12840960,  
 "module\_type": 2,  
 "load\_reason": 4,  
 "module\_base": 1937571840,  
 "module\_name": "ieframe.dll",  
 "module\_md5"="",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "process\_info": {  
 "process\_name": "iexplore.exe",  
 "signature": "Microsoft Corporation",  
 "parent\_process\_id": 2168,  
 "create\_time": 1527218501,  
 "b\_wow64": true,  
 "process\_id": 2736,  
 "b\_64bit": false,  
 "image\_file\_name": "C:\\Program Files (x86)\\Internet Explorer\\iexplore.exe",  
 "parent\_process\_name": "iexplore.exe",  
 "module\_md5"="",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "extend\_ret\_code": "0x09000000"  
}

## 后门类型：可疑启动项

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| ret\_code | String | varchar(50) | 返回code |
| extend\_ret\_code | String | varchar(50) | 返回扩展code |
| process\_info | Object |  | 进程信息，[process\_info数据结构](#processinfo数据结构) |
| autorun\_info | Object |  | 启动项信息，[autorun\_info数据结构](#autoruninfo数据结构) |

### autorun\_info数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| autoruns\_type | Integer | int(5) | 自启动项的类型 1 用户登录启动 2 系统启动 4 计划任务 8 系统服务 16 系统驱动 32 Explorer 64 已知DLL 128 IE 插件 256 Security Providers 512 Network Provider 1024 Print Monitor 2048 视频解码器 4096 WMI对象 16384 移动到启动文件夹 32768 其他杂项 |
| reg\_path | String | varchar(256) | 自启动项目的注册表路径，可能为空 |
| reg\_value\_name | String | varchar(50) | 自启动项目的注册表值名称，可能为空 |
| file\_path | String | varchar(100) | 自启动项目最终运行的文件的全路径 |
| file\_md5 | String | varchar(20) | 自启动项目最终运行的文件的md5 |
| file\_size | Integer | int(10) | 自启动项目最终运行的文件的大小 |
| comand\_line | String | varchar(128) | 命令行参数 |
| attributes | Integer | int(10) | 文件属性，无效时为INVALID\_FILE\_ATTRIBUTES |
| name | String | varchar(100) | 自启动项程序名称 |
| f\_size | Integer | int(10) | 文件大小 |
| f\_user | String | varchar(128) | 文件所有者 |
| f\_type | String | varchar(50) | 32/64/空 表示 32位文件，64位文件，空未知 |
| f\_ctime | Integer | int(10) | 文件create time |
| f\_mtime | Integer | int(10) | 文件modify time |
| f\_atime | Integer | int(10) | 文件access time |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |
| code\_section\_md5 | String | varchar(50) | 代码段MD5 |
| code\_section\_sha1 | String | varchar(50) | 代码段sha1 |
| code\_section\_sha256 | String | varchar(50) | 代码段sha256 |
| black\_id | Integer | int(10) | 黑名单id (默认0) |

**返回示例：**

{  
 "autorun\_info": {  
 "reg\_path": "HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Windows\\CurrentVersion\\Run",  
 "file\_path": "E:\\TEst\\bin\\Win32Project2.exe",  
 "file\_md5": "b4884c9fe6c808cfcb48ca6c199815a9",  
 "comand\_line": "E:\\TEst\\bin\\Win32Project2.exe",  
 "attributes": 32,  
 "reg\_value\_name": "autorun",  
 "autoruns\_type": 1,  
 "file\_size": 88576,  
 "name": "Win32Project2.exe",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "ret\_code": "0x00002000",  
 "process\_info": {  
 "process\_name": "Win32Project2.exe",  
 "signature": "",  
 "parent\_process\_id": 2496,  
 "create\_time": 1528191066,  
 "b\_wow64": false,  
 "process\_id": 2636,  
 "b\_64bit": true,  
 "image\_file\_name": "E:\\TEst\\bin\\Win32Project2.exe",  
 "parent\_process\_name": "explorer.exe",  
 "module\_md5"="",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "extend\_ret\_code": "0x05000000"  
}

## 后门类型：映像劫持

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| ret\_code | String | varchar(20) | 返回code |
| process\_info | Object |  | 进程信息，[process\_info数据结构](#processinfo数据结构) |
| hijack\_image\_info | Object |  | 劫持的映像文件信息 [hijack\_image\_info数据结构](#hijackimageinfo数据结构) |
| hijacked\_image\_info | Object |  | 被劫持的映像文件信息 [hijack\_image\_info数据结构](#hijackimageinfo数据结构) |

### hijack\_image\_info数据结构

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| file\_name | String | varchar(100) | 文件名 |
| file\_path | String | varchar(100) | 文件所在目录的全路径 |
| file\_directory | String | varchar(100) | 文件所在目录 |
| file\_md5 | String | varchar(20) | 文件md5值 |
| file\_size | Integer | int(10) | 文件大小 |
| extension | String | varchar(10) | 扩展名，如”.exe” |
| attributes | Integer | int(10) | 文件属性，无效时为INVALID\_FILE\_ATTRIBUTES |
| b\_pe | Boolean | tinyint(1) | 是否是PE文件 |
| module\_base\_str | String | varchar(20) | 模块基址 |
| load\_reason | Integer | tinyint(2) | 加载原因 |
| module\_type | Integer | tinyint(2) | 模块类型 |
| load\_time | Integer | int(10) | 加载时间，时间戳，精确到秒 |
| signature | String | varchar(128) | 签名 |
| f\_size | Integer | int(10) | 文件大小 |
| f\_user | String | varchar(128) | 文件所有者 |
| f\_type | String | varchar(50) | 32/64/空 表示 32位文件，64位文件，空未知 |
| f\_ctime | Integer | int(10) | 文件create time |
| f\_mtime | Integer | int(10) | 文件modify time |
| f\_atime | Integer | int(10) | 文件access time |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |
| code\_section\_md5 | String | varchar(50) | 代码段MD5 |
| code\_section\_sha1 | String | varchar(50) | 代码段sha1 |
| code\_section\_sha256 | String | varchar(50) | 代码段sha256 |
| black\_id | Integer | int(10) | 黑名单id (默认0) |

**返回示例：**

{  
 "hijack\_image\_info": {  
 "extension": ".dll",  
 "file\_directory": "c:\\dll",  
 "file\_path": "c:\\dll\\dlltest.dll",  
 "module\_base\_str": "0x26150000",  
 "load\_reason": 4,  
 "file\_name": "dlltest.dll",  
 "file\_md5": "a3c033d7f7229e962dc25836f94868b1",  
 "module\_type": 1,  
 "attributes": 32,  
 "signature": "",  
 "load\_time": 1527222087,  
 "file\_size": 80896,  
 "b\_pe": true,  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "ret\_code": "0x00010000",  
 "process\_info": {  
 "process\_name": "Win32Project2.exe",  
 "signature": "",  
 "parent\_process\_id": 1828,  
 "create\_time": 1527222087,  
 "b\_wow64": false,  
 "process\_id": 1788,  
 "b\_64bit": true,  
 "image\_file\_name": "E:\\TEst\\bin\\Win32Project2.exe",  
 "parent\_process\_name": "explorer.exe",  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 },  
 "hijacked\_image\_info": {  
 "extension": ".dll",  
 "file\_directory": "e:\\test\\bin",  
 "module\_base\_str": "0x28850000",  
 "load\_reason": 4,  
 "file\_name": "dlltest.dll",  
 "module\_type": 1,  
 "attributes": 32,  
 "load\_time": 1527222087,  
 "file\_size": 80896,  
 "b\_pe": true,  
 "f\_size"=100,  
 "f\_user"="",  
 "f\_type"="32",  
 "f\_ctime"=100,  
 "f\_mtime"=100,  
 "f\_atime"=100,  
 "md5"="",  
 "code\_section\_md5"="",  
 "sha1" = "",  
 "sha256"="",  
 "code\_section\_sha1"="",  
 "code\_section\_sha256"="",  
 "black\_id" = 0  
 }  
}

## 后门类型：恶意进程

**ruleResults 结构信息**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| virusName | String | varchar(50) | 病毒名称 |
| virusDescription | String | varchar(1024) | 说明 |
| libUpdateTime | Integer | int(10) | 更新时间，时间戳，精确到秒 |
| libName | String | varchar(50) | 检测库 |
| virusInstruct | String | varchar(1024) | 修复方法 |

**静态信息数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| path | String | varchar(1024) | 文件路径 |
| size | Long |  | 文件大小 |
| type | String | varchar(50) | 文件类型 |
| user | String | varchar(50) | 所属用户 |
| changeTime | Integer | int(10) | 文件创建时间 |
| modifyTime | Integer | int(10) | 文件修改时间 |
| accessTime | Integer | int(10) | 文件访问时间 |
| md5 | String | varchar(50) | 全文md5 |
| sha1 | String | varchar(50) | 全文sha1 |
| sha256 | String | varchar(50) | 全文sha256 |

**进程信息数据结构：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| uname | String | varchar(50) | 进程用户 |
| processPath | String | varchar(1024) | 进程对应执行文件路径 |
| pid | Integer |  | 进程PID |
| processName | String | varchar(50) | 进程名 |
| cmdline | String | varchar(50) | 命令行参数 |
| parentProcessName | String | varchar(50) | 父进程名 |
| parentPid | Integer |  | 父进程PID |
| parentPath | String | varchar(1024) | 父进程路径 |
| parentUserName | String |  | 父进程所属用户 |

**winSignInfoDtos 结构签名信息：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| subjectUser | String | varchar(50) | 使用者 |
| issuer | String | varchar(1024) | 颁发者 |
| beginTime | Integer | int(10) | 开始时间 |
| endTime | Integer | int(10) | 结束时间 |

**返回示例：**

Data:  
 {\"uname\" : \"Administrator\",  
 \"pid\" : 2940,  
 \"processPath\" : \"C:\\\\Users\\\\Administrator\\\\AppData\\\\Roaming\\\\Microsoft\\\\Windows\\\\Start Menu\\\\Programs\\\\Startup\\\\svchost.exe\",  
 \"processName\" : \"svchost.exe\",  
 \"cmdline\" : \"\\\"C:\\\\Users\\\\Administrator\\\\AppData\\\\Roaming\\\\Microsoft\\\\Windows\\\\Start Menu\\\\Programs\\\\Startup\\\\svchost.exe\\\" \",  
 \"parentProcessName\" : \"Explorer.EXE\",  
 \"parentPid\" : 2180,  
 \"parentPath\" : \"C:\\\\Windows\\\\Explorer.EXE\",  
 \"parentUserName\" : \"Administrator\",  
 \"path\" : \"C:\\\\Users\\\\Administrator\\\\AppData\\\\Roaming\\\\Microsoft\\\\Windows\\\\Start Menu\\\\Programs\\\\Startup\\\\svchost.exe\",  
 \"size\" : 1662736,  
 \"type\" : \"PE32\",  
 \"user\" : \"Administrators\",  
 \"changeTime\" : 1546394239,  
 \"modifyTime\" : 1481516590,  
 \"accessTime\" : 1546394239,  
 \"md5\" : \"323443b3069a12499a0ca1d1d6ccd7ee\",  
 \"sha1\" : \"1c966941cee961286522f8cc1ac409d13c2d1b65\",  
 \"sha256\" : \"\",  
 \"winSignInfoDtos\":[{  
 \"subjectUser":"Beijing Sogou Technology Development Co., Ltd.",  
 \"issuer\":\"VeriSign Class 3 Code Signing 2010 CA\",  
 \"beginTime\":1569599999,  
 \"endTime\":1504713600  
 }]  
 \"ruleResults\" : [ {  
 \"virusName\" : \"BDS/Zegost.Gen\",  
 \"virusDescription\" : \"Contains a detection pattern of the (dangerous) backdoor program BDS/Zegost.Gen Backdoor server programs\",  
 \"libUpdateTime\" : 1534953600,  
 \"libName\" : \"Avira\",  
 \"virusInstruct\" : null } ]}

# 附录3 Linux漏洞检测-查询结果-data数据结构

**返回说明：**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| code | Integer | int(10) | 执行状态的状态码, 0表示成功 |
| execute\_type | Integer | int(10) | 检测方式，1 版本比对 2：POC检测 3：版本比对和POC检测 |
| cve\_id | String | varchar(256) | cve\_id |
| poc\_check\_result | Object |  | POC检测结果，见**poc\_check\_result数据结构** |
| host\_id | String | varchar(24) | 主机agentId |
| ver\_check\_result | Object |  | 版本比对检测结果，见 **ver\_check\_result数据结构** |

**poc\_check\_result数据结构**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vuln | Integer | tinyint(2) | 0 表示不受影响，1表示受影响 |
| code | Integer | int(10) | 执行状态的状态码, 0表示成功 |
| result | vuls |  | 结果详情数组，vuls下是一个Map组成的list，用于渲染检测结果模板，各个检测项返回的结果包含的字段不同 |
| error | Integer | varchar(1024) | 执行错误时错误描述 |

**poc\_check\_result下vuls数据结构--常见字段**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| checkid | Integer | int(10) | 该检查项在脚本中的排序id |
| vuln | Integer | tinyint(2) | 是否受漏洞影响，1或true表示受漏洞影响 |
| vulnmsg | String | varchar(1024) | 检测结果从内存得到的内容，msg中包含的木板变量 |
| msg | String | varchar(1024) | 返回的人类可读结构，可读的检测结果 |
| title | String | varchar(1024) | 检测点标题 |

**ver\_check\_result数据结构**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| vuln | Integer | tinyint(2) | 0 表示不受影响，1表示受影响 |
| code | Integer | int(10) | 执行状态的状态码, 0表示成功 |
| result | vuls |  | 结果详情，见 ver\_check\_result 下vuls数据结构 |
| error | String | varchar(1024) | 执行错误时错误描述 |

**ver\_check\_result下vuls数据结构**

| 字段 | 类型 | 建议长度 | 说明 |
| --- | --- | --- | --- |
| version | Integer | varchar(1024) | 软件版本 |
| fix\_version | Integer | varchar(1024) | 修复版本 |
| name | String | varchar(1024) | 软件名称 |

**返回示例：**

{  
 "code": 0,  
 "execute\_type": 1,  
 "cve\_id": "CVE-2014-6271",  
 "poc\_check\_result": {  
 "vuln": 0,  
 "code": 0,  
 "result": {  
 "vuls": []  
 },  
 "error": null  
 },  
 "host\_id": "e807ecef881e17f7",  
 "ver\_check\_result": {  
 "vuln": 1,  
 "code": 0,  
 "result": {  
 "vuls": [  
 {  
 "version": "4.3-6ubuntu1",  
 "fix\_version": "4.3-7ubuntu1.1",  
 "name": "bash"  
 }  
 ]  
 },  
 "error": ""  
 }  
}

{  
 "code": 0,  
 "execute\_type": 2,  
 "cve\_id": "CVE-2015-0235",  
 "poc\_check\_result": {  
 "vuln": 1,  
 "code": 0,  
 "result": {  
 "vuls": [  
 {  
 "checkid": 1,  
 "vuln": 1,  
 "vulnmsg": "0000000",  
 "msg": "if ghost glibc not exist return '1234567890123456', otherwise return %vulnmsg%",  
 "title": "ghost\_CVE-2015-0235\_poc\_check"  
 }  
 ]  
 },  
 "error": null  
 },  
 "host\_id": "ffd324ed2a80f7f9",  
 "ver\_check\_result": {  
 "vuln": 0,  
 "code": 0,  
 "result": {  
 "vuls": []  
 },  
 "error": null  
 }  
}

{  
 "code": 0,  
 "execute\_type": 3,  
 "cve\_id": "CVE-2014-7169",  
 "poc\_check\_result": {  
 "vuln": 1,  
 "code": 0,  
 "result": {  
 "vuls": [  
 {  
 "msg": "\\u5b58\\u5728\\u6f0f\\u6d1e",  
 "vuln": true,  
 "info": "\\u5728/dev/shm\\u76ee\\u5f55\\u4e2d\\u53d1\\u73b0Poc\\u751f\\u6210\\u7684\\u6587\\u4ef6/dev/shm/QT\_Test\_1536521467"  
 }  
 ]  
 },  
 "error": null  
 },  
 "host\_id": "308012a1e4f2e7fe",  
 "ver\_check\_result": {  
 "vuln": 1,  
 "code": 0,  
 "result": {  
 "vuls": [  
 {  
 "version": "4.1-2ubuntu3",  
 "fix\_version": "4.1-2ubuntu3.2",  
 "name": "bash"  
 }  
 ]  
 },  
 "error": ""  
 }  
}

# 附录4 数据字典

## 通信状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| onlineStatus | 0 | 离线 |
| onlineStatus | 1 | 在线 |

## Agent状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| agentStatus | 0 | 在线 |
| agentStatus | 1 | 离线 |
| agentStatus | 2 | 停用 |
| agentStatus | 3 | 删除中 |

## 系统负载

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| systemLoad | 0 | 未知 |
| systemLoad | 1 | 低 |
| systemLoad | 2 | 中 |
| systemLoad | 3 | 高 |

## windows进程类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| type | 1 | 应用程序 |
| type | 2 | 后台程序 |
| type | 3 | windows进程 |

## linux进程状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| state | R | 可执行状态&运行状态 |
| state | S | 可中断的睡眠状态, 可处理signal |
| state | D | 不可中断的睡眠状态, 可处理signal, 有延迟 |
| state | T | 暂停状态或跟踪状态 |
| state | Z | 退出状态，进程成为僵尸进程 |

## linux账号状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| accountStatus | 0 | 禁用 |
| accountStatus | 1 | 启用 |

## windows账号状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| accountStatus | 0 | 启用 |
| accountStatus | 1 | 锁定 |
| accountStatus | 2 | 禁用 |

## linux登录状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| loginStatus | 0 | 不可登入 |
| loginStatus | 1 | 不可交互登入 |
| loginStatus | 2 | 可交互登入 |
| loginStatus | 3 | key&pwd登陆 |

## windows账号类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| type | 0 | 访客用户 |
| type | 1 | 标准用户 |
| type | 2 | 管理员用户 |

## linux账户登录方式

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| accountLoginType | 0 | 不可登陆 |
| accountLoginType | 1 | key登陆 |
| accountLoginType | 2 | pwd登陆 |
| accountLoginType | 3 | key&pwd登陆 |

## linux交互登录方式

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| interactiveLoginType | 0 | 不可登录 |
| interactiveLoginType | 1 | 不可交互登录 |
| interactiveLoginType | 2 | 可交互登录 |

## jar包类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| type | 1 | 应用程序 |
| type | 2 | 系统类库 |
| type | 3 | web服务自带库 |
| type | 8 | 其他依赖包 |

## iis站点状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| state | 0 | starting |
| state | 1 | started |
| state | 2 | stopping |
| state | 3 | stopped |

## 平台类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| platform | 1 | linux |
| platform | 2 | windows |

## 系统类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| osType | 1 | linux |
| osType | 2 | windows |

## linux默认启动模式

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| initLevel | 0 | 关机。不能将系统缺省运行级别设置为0，否则无法启动。 |
| initLevel | 1 | 单用户模式，只允许root用户对系统进行维护。 |
| initLevel | 2 | 多用户模式，但不能使用NFS（相当于Windows下的网上邻居） |
| initLevel | 3 | 字符界面的多用户模式。 |
| initLevel | 4 | 未定义。 |
| initLevel | 5 | 图形界面的多用户模式。 |
| initLevel | 6 | 重启。不能将系统缺省运行级别设置为6，否则会一直重启。 |

## windows启动类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| startType | 1 | 注册表 |
| startType | 2 | 文件夹 |

## linux任务类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| taskType | CRONTAB | 定时任务 |
| taskType | AT | 单个任务 |
| taskType | BATCH | 批量任务 |

## linux安装包类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| type | rpm | rpm安装类型 |
| type | dpkg | dpkg安装类型 |
| type | java | java安装类型 |
| type | system | system安装类型 |

## windows SQL Server身份验证

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| loginModel | 1 | Windows模式 |
| loginModel | 2 | SQLServer和Windows模式 |

## windows SQL Server审核级别

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| auditLevel | 0 | 无 |
| auditLevel | 1 | 仅限成功的登录 |
| auditLevel | 2 | 仅限失败的登录 |
| auditLevel | 3 | 失败和成功的登录 |

## 风险类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| family | 1 | 补丁 |
| family | 2 | 对外访问性 |
| family | 3 | 安全配置 |
| family | 4 | POC |
| family | 5 | 弱密码 |
| family | 7 | 系统风险 |
| family | 8 | 账号风险 |

## 危险程度

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| severity | 0 | 信息 |
| severity | 1 | 低危 |
| severity | 2 | 中危 |
| severity | 3 | 高危 |
| severity | 4 | 危急 |

## 修复影响

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| restartOpts | 0 | 未知 |
| restartOpts | 1 | 不需要重启 |
| restartOpts | 2 | 服务重启 |
| restartOpts | 3 | 系统重启 |

## 是否存在进程影响

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| businessImpact | 0 | 无进程影响 |
| businessImpact | 1 | 有进程影响 |

## 弱密码类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| weakType | 1 | 空密码 |
| weakType | 2 | 默认弱密码 |
| weakType | 3 | 跟用户名相同 |
| weakType | 4 | 常见弱密码 |

## 密码状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| pwdStatus | 1 | 正常 |
| pwdStatus | 2 | 将要失效 |
| pwdStatus | 3 | 已经失效 |
| pwdStatus | 4 | 已锁定 |

## shell登录性

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| loginShell | 0 | 非登录shell |
| loginShell | 1 | 登录shell |

## 解析方式

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| translate | 0 | host解析 |
| translate | 1 | dns解析 |

## 风险文件危险等级

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| regexLevel | 1 | 低危 |
| regexLevel | 2 | 中危 |
| regexLevel | 3 | 高危 |

## 风险文件类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| regexType | 1 | 临时文件 |
| regexType | 2 | 压缩文件 |
| regexType | 3 | 备份文件泄露 |
| regexType | 4 | 数据文件泄露 |
| regexType | 5 | 配置文件泄露 |
| regexType | 6 | 日志泄露 |
| regexType | 7 | 脚本泄露 |
| regexType | 8 | Office文档 |
| regexType | 9 | 源代码泄露 |
| regexType | 10 | 系统文件 |
| regexType | 11 | phpinfo文件 |

## 漏洞类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| category | 0 | SQL注入 |
| category | 1 | 未授权访问 |
| category | 2 | 敏感信息泄露 |
| category | 3 | XML外部实体注入 |
| category | 4 | 跨站脚本攻击 |
| category | 5 | 不安全的反序列 |
| category | 6 | 客户端请求伪造 |
| category | 7 | 服务端请求伪造 |
| category | 8 | 命令执行 |
| category | 9 | 代码执行 |
| category | 10 | 任意文件上传 |
| category | 11 | 任意文件读取 |
| category | 12 | 拒绝服务攻击 |
| category | 13 | 目录遍历 |
| category | 14 | 恶意后门 |
| category | 15 | 本地提权 |
| category | 16 | 注入漏洞 |

## 漏洞允许的检测方式

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| permitExecuteTypes | 1 | 版本对比 |
| permitExecuteTypes | 2 | POC检测 |

## 漏洞检测脚本执行风险

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| executeSeverity | 1 | 无风险 |
| executeSeverity | 2 | 低风险 |
| executeSeverity | 3 | 中风险 |
| executeSeverity | 4 | 高风险 |

## 漏洞作业类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| jobType | 1 | 全局作业 |
| jobType | 2 | 用户作业 |

## 漏洞作业执行状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| executeStatus | 0 | 未开始执行 |
| executeStatus | 1 | 排队执行 |
| executeStatus | 2 | 正在执行 |
| executeStatus | 3 | 执行成功 |
| executeStatus | 4 | 执行失败 |

## 可疑操作审核状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| auditStatus | 1 | 未审核 |
| auditStatus | 2 | 审核通过 |
| auditStatus | 3 | 审核未通过 |

## 可疑操作类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| status | 0 | 未命中规则 |
| status | 1 | bash危险命令执行 |
| status | 2 | Wget下载黑客工具 |
| status | 3 | curl下载黑客工具 |
| status | 4 | rcp下载黑客工具 |
| status | 5 | scp下载黑客工具 |
| status | 6 | rsync下载黑客工具 |
| status | 7 | MYSQL明文密码显示 |
| status | 8 | Mongo明文密码显示 |
| status | 9 | scp外部下载或上传 |
| status | 10 | rcp外部下载或上传 |
| status | 11 | rsync外部下载或上传 |
| status | 12 | 赋给目录或文件危险权限 |

## 暴力破解封停状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| block | 0 | 未处理 |
| block | 1 | 自动封停 |
| block | 2 | 自动解封 |
| block | 3 | 手动封停 |
| block | 4 | 手动解封 |
| block | 5 | 解封中 |
| block | 6 | 封停中 |
| block | -1 | 封停失败 |
| block | -2 | 解封失败 |

## 暴力破解判定为爆破的原因（类型）

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| reason | 1 | 相同IP下同一用户名登录多次 |
| reason | 2 | 相同IP下多个不存在的用户名登录 |
| reason | 3 | 指定时间内重试达到指定次数 |

## 是否恶意IP库

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| evilIpDb | 0 | 否 |
| evilIpDb | 1 | 是 |

## 是否暴力破解成功的登录

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| crackSuccess | 0 | 否 |
| crackSuccess | 1 | 是 |

## 不在常用ip段登录

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| abnormalIp | 0 | 否 |
| abnormalIp | 1 | 是 |

## 不在常用区域的登录

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| abnormalLocation | 0 | 否 |
| abnormalLocation | 1 | 是 |

## 不在常用时间的登录

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| abnormalTime | 0 | 否 |
| abnormalTime | 1 | 是 |

## 主机范围

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| realmKind | 0 | 全部主机 |
| realmKind | 1 | 主机 |
| realmKind | 2 | 业务组 |

## web后门类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| type | 1 | 代码执行 |
| type | 2 | 系统执行 |
| type | 3 | 数据库后门 |
| type | 4 | 文件目录管理 |
| type | 5 | 特性后门 |
| type | 6 | 多功能木马 |
| type | 7 | 已知后门 |
| type | 8 | 云引擎检测 |
| type | 9 | Avira引擎 |
| type | 10 | 雷火引擎 |
| type | 11 | 已知恶意样本 |
| type | 301 | 远程下载 |
| type | 401 | 混淆加密 |

## 扫描状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| retcode | -1 | 无任务 |
| retcode | 0 | 未开始执行 |
| retcode | 1 | 排队执行 |
| retcode | 2 | 正在执行 |
| retcode | 3 | 执行成功 |
| retcode | 4 | 执行失败 |

## linux后门类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| backDoorType | 应用后门 | 应用后门 |
| backDoorType | Bootkit | Bootkit后门 |
| backDoorType | Rootkit | Rootkit后门 |

## linux后门检查功能

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| backDoorCheckName | RPM-based应用后门检查 | 无 |
| backDoorCheckName | DPKG-based应用后门检查 | 无 |
| backDoorCheckName | 磁盘MBR检查 | 无 |
| backDoorCheckName | 计划任务检查 | 无 |
| backDoorCheckName | 动态链接库检查 | 无 |
| backDoorCheckName | 基本命令检查 | 无 |
| backDoorCheckName | 已知rootkit检查 | 无 |
| backDoorCheckName | 系统内核模块检查 | 无 |
| backDoorCheckName | 网络状态检查 | 无 |
| backDoorCheckName | 用户状态检查 | 无 |
| backDoorCheckName | 系统文件状态检查 | 无 |
| backDoorCheckName | 进程状态检查 | 无 |

## windows后门类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| backDoorTypeId | 0 | 未知类型 |
| backDoorTypeId | 1 | 可疑进程 |
| backDoorTypeId | 2 | 可疑线程 |
| backDoorTypeId | 3 | 可疑模块 |
| backDoorTypeId | 4 | 可疑启动项 |
| backDoorTypeId | 5 | 映像劫持 |
| backDoorTypeId | 1000 | 恶意进程 |

## 网络蜜罐规则状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| status | 0 | 规则启用 |
| status | 1 | 规则禁用 |

## 网络蜜罐规则同步状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| verifyStatus | 0 | 规则同步中 |
| verifyStatus | 1 | 同步完毕 |

## 网络蜜罐规则端口状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| status | -2 | 端口删除规则进行中 |
| status | -1 | 端口新增规则同步行中 |
| status | 0 | 端口规则同步成功 |
| status | 1 | 端口同步规则失败,原因：agent离线 |
| status | 2 | 端口同步规则失败,原因：agent返回超时 |
| status | 3 | 端口同步规则失败,原因：agent停用 |
| status | 4 | 端口同步规则失败,原因：agent未返回对应端口信息 |
| status | 10 | 端口同步规则失败,原因：服务端参数错误 |
| status | 11 | 端口同步规则失败,原因：系统错误(线程创建失败，内存分配失败等) |
| status | 12 | 端口同步规则失败,原因：agent已降级功能不能使用 |
| status | 20 | 端口同步规则失败,原因：其他原因监听失败 |
| status | 21 | 端口同步规则失败,原因：其他原因监听失败 |

## 基线类别

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| family | 1 | 系统基线 |
| family | 2 | 应用基线 |

## 基线检查风险级别

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| level | 1 | 低风险 |
| level | 2 | 中风险 |
| level | 3 | 该风险 |

## pod状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| status | 1 | running |
| status | 2 | pending |
| status | 3 | succeed |
| status | 4 | failed |
| status | 5 | unknown |

## 快速任务类型

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| taskType | 1 | 系统内置任务 |
| taskType | 2 | 用户添加任务 |

## 快速任务执行时间范围

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| duration | 0 | 1分钟内 |
| duration | 1 | 1~3分钟内 |
| duration | 2 | 3~5分钟内 |
| duration | 3 | 5~10分钟内 |
| duration | 4 | 10~30分钟内 |
| duration | 5 | 超出30分钟 |

## 快速任务扫描状态

| 字段 | 字段内容 | 说明 |
| --- | --- | --- |
| status | 1 | 准备执行 |
| status | 2 | 正在执行 |
| status | 3 | 执行成功 |
| status | 4 | 执行失败 |