Venustech WAF 虚拟化版本 Web API 接口

目录

VENUSTECH WAF 虚拟化版本 WEB API 接口 **1**

目录 **1**

1 接口约定 **6**

接口设置 **6**

2 认证流程 **6**

3 接口说明 **9**

3.1 获取系统信息 9
3.2 管理员用户 10

3.2.1 获取单个用户 *10*

3.2.2 添加用户 *11*

3.2.3 删除用户 *12*

3.2.4 修改用户密码 *13*

3.2.5 解锁用户 *14*

3.3 业务口 IP 配置 15

3.3.1 获取单个业务口 *15*

3.3.2 修改单个业务口 *16*

3.4 DNS 配置 17

3.4.1 获取 DNS 配置 *17*

3.4.2 修改 DNS 配置 *18*

3.5 NTP 服务器配置 19

3.5.1 获取 NTP 服务器配置 *19*

3.5.2 修改 NTP 配置 20
3.6 授权 21
3.6.1 导入授权 21
3.6.2 激活授权 22
3.6.3 查看授权 23
3.7 站点安全配置 25
3.7.1 获取所有防护策略 25
3.7.2 获取单个防护策略 29
3.7.3 添加防护策略 32
3.7.4 修改防护策略 35
3.7.5 删除防护策略 37
3.7.6 启用/禁用防护策略 38
3.7.7 新建站点安全 39
3.7.8 删除站点安全 39
3.7.9 查看站点安全或查看所有站点安全列表 40
3.7.10 查看暴力破解、扫描防护 43
3.7.11 修改暴力破解、扫描防护(有默认值) 45
3.7.12 查看慢速攻击 47
3.7.13 修改慢速攻击(有默认值) 48
3.7.14 查看 SQL 注入防护 50
3.7.15 修改 SQL 注入防护(有默认值) 51
3.7.16 查看 XSS 攻击防护 52

3.7.17 修改 XSS 攻击防护(有默认值) 53
3.7.18 查看敏感信息防护 55
3.7.19 修改敏感信息防护(有默认值) 56
3.7.19 修改事件引擎 58
3.8 SSL 证书配置 60
3.8.1 查看证书 60
3.8.2 删除证书 63
3.8.3 导入证书 64
3.9 事件库升级 65
3.9.1 事件库立即升级 65
3.9.2 获取事件库升级日志 66
3.10 安全事件统计 67
3.10.1 获取安全事件统计 67
3.11 FLOOD 防护 68
3.11.1 查看智能 TCP FLOOD 防御配置 68
3.11.2 修改智能 TCP FLOOD 防御配置 69
3.11.3 查看所有防护策略 70
3.11.4 查看单个防护策略 73
3.11.5 添加防护策略 76
3.11.6 修改防护策略 78
3.11.7 删除单个防护策略 80
3.11.8 查看所有 Flood 防护 预留需要平台修改地址对象 81
3.12 软件 BYPASS 82
3.12.1 修改软件 Bypass 82

3.13 黑/白名单配置 83
3.13.1 获取所有黑/白名单 83
3.13.2 添加/修改黑/白名单 85
3.13.3 删除黑/白名单 88
3.13.4 开启/关闭黑/白名单 88
3.14 网关配置 89
3.14.1 获取所有网关 89
3.14.2 添加网关 90
3.14.3 删除网关 91
3.15 静态路由配置 92
3.15.1 获取所有静态路由 92
3.15.2 添加静态路由 94
3.15.3 修改路由 95
3.15.4 删除路由 96
3.16 设备参数 97
3.16.1 设备重启 97
3.16.2 保存配置 98
3.17 地址对象 99
3.17.1 获取所有地址对象 99
3.17.2 获取单个地址对象 101
3.17.3 添加地址对象 103
3.17.4 修改地址对象 104
3.17.5 删除地址对象 105
3.18 虚拟服务 106
3.18.1 获取所有虚拟服务 106
3.18.2 获取单个虚拟服务 108
3.18.3 添加虚拟服务 110
3.18.4 修改虚拟服务 112
3.18.5 删除虚拟服务 114
3.18.6 开启/关闭虚拟服务 115

3.19 服务对象 116
3.19.1 获取所有服务对象 116
3.19.2 获取单个服务对象 118
3.19.3 添加服务对象 119
3.19.4 修改服务对象 120
3.19.5 删除服务对象 121
3.20 WEB 恶意扫描防护 122
3.20.1 web 恶意扫描防护修改 122
3.21 事件引擎 125
3.21.1 获取事件引擎一级二级 125
3.21.2 获取事件引擎三级 136
3.21.3 事件引擎保存 144
3.22 配置导入导出和一键回退 147
3.22.1 配置导出 147
3.22.2 配置导入 148
3.22.3 一键回退 149

1 接口约定

1. 接口URI：供第三方服务调用，用来获取WAF信息的访问地址

2. 接口遵循restful风格，略有调整

3. 允许使用http、https访问，受限于WAF设备本身开放的权限：如协议（HTTP、HTTPS）访问限制、IP访问限制

4. 每个接口的默认前缀为 **http(s)://ip:port/v2/**

5. 允许port为多个，具体设置需请咨询技术人员

6、POST参数为json格式，编码为utf-8编码

接口设置

webserver可以同时开启http和https，需要配置多个端口时，修改配置文件后，重启webserver即可

批注 [P1]: 增加“接口设置”，可通过 web 界面设置使用 http 还 https，可以设置端口

2 认证流程

1. 用户名为系统用户，密码为账户对应的密码。

2. 请求api接口时，需要携带两个参数：ts，表示13位unix时间戳（毫秒）；sign，表示签名（无需携带密码参数）

批注 [P2]: 10 位的时间戳只能用到 2033 年，这个最好改成 long 型，单位为毫秒

批注 [r3R2]: 修改

3. ts时间戳如果与服务器时间戳之差超过10分钟，则视为无效。

4. sign计算方式：sign = user + ':' + md5 (ts + api_uri + user +

md5(pass))，其中ts表示时间戳，api_uri表示api的

uri，user表示用户名，pass表示密码

举例： 请求/v2/sysinfo 这个接口：

api_uri = /v2/sysinfo,

ts = 1468395806542,

按照sign=user + ':' + md5 (ts + api_uri + user + md5(pass))计算，可得sign

值为：

user:64234aeb5392f125c811513e3b702ca7

所以最终的url为

http(s)://ip:port/v2/sysinfo?ts=1460362301129&sign=virtual.user:64234aeb5392f125c811513e3b702ca7

接口中的公共部分

返回 HTTP 200代表接口成功调用，否则说明接口调用有误，比如URL地址错误。

即使返回HTTP 200，也不能说明业务逻辑正常调用。

每个接口的返回内容中，用json字符串说明返回结果。其中包括reuslt字段，为数值类型，0表示执行成功，非0表示出错。

错误对照表

0	成功
1	失败，ts时间戳之差超过10分钟
2	失败，sign错误

3	其他失败原因，详细参加错误描述errmsg
4	

批注 [P4]: 用户名错误返回什么？

批注 [r5R4]: 返回 3，本版本接口其他类错误都是 3，详细错误信息需要看 errormsg

返回错误数据对应内容

返回数据	**数据**

ts时间戳之差超过10分钟	{
	"result" : 1,
	"errmsg" : "ts check error",
	"serv_time": 1460363462234 //表示服务器当前时间戳
	}

sign错误	{
	"result" : 2,
	"errmsg" : "sign error"
	}

其他错误	{
	"result" : 3,
	"errmsg" : "here is error message" //这里会返回相关错误信息
	}

3 接口说明

3.1 获取系统信息

请求接口	http(s)://ip:port/v2/sysinfo?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result" : 0, //参加错误码对照表
	"sysinfo" :
	{
	"cpu_usage" : "18", //cpu 利用率百分比
	"mem_usage" : "65", //内存利用率百分比
	"conn_num" : "1000", //连接数
	"disk_usage" : "51", //硬盘利用率百分比
	"sys_uptime" : "10000", //系统运行时间(秒)
	"halfopen_conn_num" : "20000"
	}

批注 [r6]: 第二阶段接口，加入系统运行时间

	} 错误数据返回请参见**返回错误数据对应内容**

3.2 管理员用户

3.2.1 获取单个用户

请求接口	http(s)://ip:port/v2/adminuser?action=show&username=xxx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 这里的username必填
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" ：0, //参见错误码对照表 "adminuser" : { "username": "user", //用户名，字符串，6-20长度 "password": "abcdef", //密码，字符串，6-20长度

批注 [P7]: 密码最好不是明文的，可以用 0x112233445566778899AABB 进行异或，异或后的结果用 base64 编码

批注 [r8R7]: 可以按照批注修改密码加解密

	"type": 0, //类型，0表示普通用户，1表示radius用户 "role": 2, //角色，只能为2，表示配置管理员 "priv": 2, //权限，1表示只读权限，2表示读写权限 "radius_server_name": "radiusServer", //radius服务器名 "email": "test@163.com", //邮箱名 "phone": 13311234312, //手机号 "desc": "描述" //描述信息 } } 错误数据返回请参见**返回错误数据对应内容**

3.2.2 添加用户

请求接口	http(s)://ip:port/v2/adminuser?action=add
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。只能单个添加 POST的参数： { "username": "user", //用户名，字符串，6-20长度 "password": "abcdef", //密码，字符串，6-20长度

	"type": 0, //类型，0表示普通用户，1表示radius用户 "role": 2, //角色，只能为2，表示配置管理员 "priv": 2, //权限，1表示只读权限，2表示读写权限 "radius_server_name": "radiusServer", //radius服务器名 "email": "test@163.com", //邮箱名 "phone": 13311234312, //手机号 "desc": "描述" //描述信息 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0 } //参见错误码对照表 错误数据返回请参见**返回错误数据对应内容**

3.2.3 删除用户

请求接口	http(s)://ip:port/v2/adminuser?action=delete&username=xxx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。username不能为空

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.2.4 修改用户密码

请求接口	http(s)://ip:port/v2/adminpassword?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST的参数：
	{
	"username": "user", //用户名，字符串，6-20长度
	" oldpasswd": "abcdef", //旧密码
	" newpasswd": "ZaSef", //新密码
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据：

	{"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.2.5 解锁用户

请求接口	http(s)://ip:port/v2/adminUnblock?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST的参数：
	{
	"ipaddr": "192.12.32.12", //要解锁的IP，IPv4
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.3 业务口 IP 配置

3.3.1 获取单个业务口

请求接口	http(s)://ip:port/v2/interface?action=show&name=xxx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。获取接口name必填，如gev0/1
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: { "result" : 0, "interface" : { "type": 0, //类型必填，0为静态IP，1为DHCP "name": "GEV0/1", //接口名称必填 "desc": "描述", "ipaddr": "192.168.1.2", //ipv4地址必填 "ipaddr_v6": "2001:da8:8000:d010::1", // ipv6地址选填 "ping": 0, //是否允许ping，0不允许，1允许

批注 [P9]: Name 怎么填？ 是 eth0 还是什么？

批注 [r10R9]: 修改

批注 [P11]: 是否区分大小写？

批注 [r12R11]: 区分大小写

批注 [P13]: 要给出掩码和默认网关

批注 [t14R13]: 192.168.1.2/24

	"mtu": 1500 //mtu } } 错误数据返回请参见**返回错误数据对应内容**

3.3.2 修改单个业务口

请求接口	http(s)://ip:port/v2/interfacesub?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST的参数:
	{
	"type": 0, //类型必填，0为静态IP，1为DHCP
	"name": "GEV0/1", //接口名称必填
	"desc": "描述",
	"ipaddr": "192.168.31.1/24", //ipv4地址/掩码 必填
	"ipaddr_v6": "2001:da8:8000:d010::1", // ipv6地址选填
	"ping": 0, //是否允许ping，0不允许，1允许
	"mtu": 1500 //mtu
	}

批注 [P15]: 掩码和网关在哪里配？

批注 [r16R15]: 这里的 ipaddr 填写地址/掩码

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0 } 错误数据返回请参见**返回错误数据对应内容**

3.4 DNS 配置

3.4.1 获取 DNS 配置

请求接口	http(s)://ip:port/v2/dns?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0, "dns_ipv4" : {

	"first_dns": "114.114.114.114", //首选
	"second_dns": "8.8.8.8" //备选
	},
	"dns_ipv6" :
	{
	"first_dns": "", //首选
	"second_dns": "" //备选
	}
	}
	错误数据返回请参见**返回错误数据对应内容**

3.4.2 修改 DNS 配置

请求接口	http(s)://ip:port/v2/dns?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "dns_ipv4" : { "first_dns": "114.114.114.114", //首选 "second_dns": "8.8.8.8" //备选

	},
	"dns_ipv6" :
	{
	"first_dns": "", //首选
	"second_dns": "" //备选
	}
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.5 NTP 服务器配置

3.5.1 获取 NTP 服务器配置

请求接口	http(s)://ip:port/v2/ntp?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200

返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result" : 0,
	"ntp" :
	{
	"ntp_server_ip ": "pool.ntp.org", //服务器是网址或 ip
	"ntp_interval ": "5" //ntp服务器同步时间间隔，单位分钟
	}
	}
	错误数据返回请参见**返回错误数据对应内容**

3.5.2 修改 NTP 配置

请求接口	http(s)://ip:port/v2/ntp?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 { "ntp_server_ip ": "pool.ntp.org", //服务器是网址或 ip

	"ntp_interval ": "5" //ntp服务器同步时间间隔，单位分钟 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0 } 错误数据返回请参见**返回错误数据对应内容**

3.6 授权

3.6.1 导入授权

请求接口	http(s)://ip:port/v2/license?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 { "license": "QvjfpVWt9NpxNUU", "generated_tmp_license": 1, //表示是否产生过临时授权，0 没有，1 有

	}
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.6.2 激活授权

请求接口	http(s)://ip:port/v2/license?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	分别有临时授权和激活授权，通过add_action区分
	POST参数
	{
	"add_action": 1, //1-临时授权，2-激活
	"generated_tmp_license": 1 //表示是否产生过临时授权，0 没有，1 有
	}

返回码	200
返回格式	application/json;charset=utf-8

批注 [P17]: 激活用在什么场景?

批注 [t18R17]: 导入正式授权后激活

返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.6.3 查看授权

请求接口	http(s)://ip:port/v2/license?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result" : 0,
	"license" :
	{//以下功能项中 enable 的含义:0-未授权，1-试用授权，2-正式授权
	"ips_lincense_enable": 2, // 特征库授权状态
	"ips_lincense_time": "188267379", //授权剩余有效期，单位毫秒，以下有效期单位一致

	"ips_alert": 0,//授权提示，授权时间小于30天时为1 "ssl_lincense_enable": 2, // HTTPS 应用防护授权状态 "ssl _lincense_time": "188267379", //授权剩余有效期 "ssl_alert": 0,//授权提示，授权时间小于30天时为1 "accel_lincense_enable": 1, // 缓存加速授权状态 "accel_lincense_time": "188267379", //授权剩余有效期 "accel_alert": 0,//授权提示，授权时间小于 30 天时为 1 "ipcheck_lincense_enable": 1, //源区域访问控制授权状态 "ipcheck_lincense_time": "188267379", //授权剩余有效期 " ipcheck_alert": 0,//授权提示，授权时间小于 30 天时为 1 "webtakeover_lincense_enable": 1, //网站锁授权状态 "webtakeover_lincense_time": "188267379", //授权剩余有效期 "webtakeover_alert": 0,//授权提示，授权时间小于 30 天时为 1 "portnum": 23,//授权端口总数

	"generated_tmp_license": 1,//是否产生过临时授权，1-产生过，0-没有
	"license": 0,//标志位，始终为 0
	"if_list": [
	//当前授权的端口
	{"ifname": "ge0/4"},
	{"ifname": "ge0/3"},
	{"ifname": "ge0/2"},
	{"ifname": "ge0/1"},
	{"ifname": "eth1"}
	]
	}
	}
	错误数据返回请参见**返回错误数据对应内容**

3.7 站点安全配置

3.7.1 获取所有防护策略

请求接口	http(s)://ip:port/v2/protectedPolicy?action=show
请求方式	GET

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result" : 0
	"protect_policy " : [
	{
	//桥模式、路由模式
	"name": "default", //站点防护策略名称，最大长度 64 字节
	"mode": 0, //0-桥模式，1-代理模式，2-单臂模式，3-路由模式
	"protocal": http, //应用类型
	"port": 80, //代理端口（代理模式和单臂模式填充）
	"ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启
	"local_cert": "", //本地证书，长度 64 字节
	"ssl_version": "SSL3.0", //SSL 版本，10 字节

批注 [P19]: 是证书的文件名还是什么？

批注 [t20R19]: 去掉后缀的证书名称

	"cipher_suit": "", //算法套件，1024 字节 "real_server" : [{ "type" : 0, //真实服务器类型，0-主机，1-子网，2-范围，10-ipv6 主机，11-ipv6 子网，12-ipv6 范围 "host" : "192.168.53.12", //type 为主机，64 字节，格式如：192.168.54.191 "net" : "192.168.54.191/32", //type 为子网，64 字节，格式：192.168.54.191/32 "range1" : "", //type 范围，64 字节，起始 ip 地址 "range2" : "", //type 范围，64 字节，结束 ip 地址 "port": 80, //真实服务器端口 "protocal": http, //协议，8 字节 "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "ssl_version" : "SSL3.0" //SSL 版本，10 字节 }], "website_safe": "high_level", //站点安全名称

	"doname": "" //域名，字符串
	},
	{
	//代理模式、单臂模式
	"name": "porxy", //站点防护策略名称，最大长度 64 字节
	"mode": 1, //0-桥模式，1-代理模式，2-单臂模式，3-路由模式
	"ip": "192.168.21.11", //代理 ip（代理模式和单臂模式填充），最大长度 64 字节
	"protocal": http, //应用类型
	"port": 8080, //代理端口（代理模式和单臂模式填充）
	"ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启
	"local_cert": "", //本地证书，长度 64 字节
	"ssl_version": "", //SSL 版本，10 字节
	"cipher_suit": "", //算法套件，1024 字节
	"ip_ver": 4, //代理模式和单臂模式时，填写这个字段，4 是 ipv4，6 是 ipv6
	"real_server" : {

	"type" : 0, //真实服务器类型，0-主机，10-ipv6 主机 "host" : "192.168.53.12", //type 为主机，64 字节，格式如：192.168.54.191 "port": 80, //真实服务器端口 "protocal": http, //协议，8 字节 "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "ssl_version" : "" //SSL 版本，10 字节 }, "website_safe": "mid_level", //站点安全名称 "doname": "" //域名，字符串 }] } 错误数据返回请参见**返回错误数据对应内容**

3.7.2 获取单个防护策略

请求接口	http(s)://ip:port/v2/protectedPolicy?action=show&name=xxx
请求方式	GET

批注 [P21]: Name 怎么填写？

批注 [r22R21]: 如果获取 default 就写 default，取决你添加的 name 是什么

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。名字如default
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0, "protect_policy " : [{ "name": "default", //站点防护策略名称，最大长度 64 字节 "mode": 0, //0-桥模式，1-代理模式，2-单臂模式，3-路由模式 "ip": "", //代理 ip（代理模式和单臂模式填充），最大长度 64 字节 "protocal": http, //应用类型 "port": 80, //代理端口（代理模式和单臂模式填充） "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "local_cert": "", //本地证书，长度 64 字节

	"ssl_version": "SSL3.0", //SSL 版本，10 字节 "cipher_suit": "", //算法套件，1024 字节 "real_server" : [{ "type" : 0, //真实服务器类型，0-主机，1-子网，2-范围，10-ipv6 主机，11-ipv6 子网，12-ipv6 范围 "host" : "192.168.53.12", //type 为主机，64 字节，格式如：192.168.54.191 "net" : "192.168.54.191/32", //type 为子网，64 字节，格式：192.168.54.191/32 "range1" : "", //type 范围，64 字节，起始 ip 地址 "range2" : "", //type 范围，64 字节，结束 ip 地址 "port": 80, //真实服务器端口 "protocal": http, //协议，8 字节 "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "ssl_version" : "SSL3.0" //SSL 版本，10 字节

	}], "website_safe": "high_level", //站点安全名称 "doname": "" //域名，字符串 }] } 错误数据返回请参见**返回错误数据对应内容**

3.7.3 添加防护策略

请求接口	http(s)://ip:port/v2/protectedPolicy?action=add
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 此处以桥模式为例 POST参数 { "name": "default", //站点防护策略名称，最大长度 64 字节

批注 [P23]: 名称应该不能重复吧？重复了会怎样？

批注 [r24R23]: 不能重复，重复添加会失败

	"mode": 0, //0-桥模式，1-代理模式，2-单臂模式，3-路由模式 "ip": "", //代理 ip（代理模式和单臂模式填充），最大长度 64 字节 "protocal": http, //应用类型 "port": 80, //代理端口（代理模式和单臂模式填充） "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "local_cert": "", //本地证书，长度 64 字节 "ssl_version": "SSL3.0", //SSL 版本，10 字节 "cipher_suit": "", //算法套件，1024 字节 "real_server" : [{ "type" : 0, //真实服务器类型，0-主机，1-子网，2-范围，10-ipv6 主机，11-ipv6 子网，12-ipv6 范围 "host" : "192.168.53.12", //type 为主机，64 字节，格式如：192.168.54.191 "net" : "192.168.54.191/32", //type 为子网，64 字节，格式：192.168.54.191/32

批注 [P25]: 代理模式和单臂模式下，就是业务口的 IP 吧?

批注 [t26R25]: 可以是业务口 ip 也可以是单独 ip

批注 [P27]: 这个字段怎么填?

批注 [r28R27]: ssl 证书名称，不带后缀

批注 [P29]: 给个例子

批注 [t30R29]: ALL:eNULL
或者
ALL:!ADH:@STRENGTH
跟 openssl 规则格式一样

	"range1" : "", //type 范围，64 字节，起始 ip 地址 "range2" : "", //type 范围，64 字节，结束 ip 地址 "port": 80, //真实服务器端口 "protocal": http, //协议，8 字节 "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "ssl_version" : "SSL3.0" //SSL 版本，10 字节 }], "website_safe": "high_level", //站点安全名称 "doname": "" //域名，字符串 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

批注 [P31]: 这个是什么含义？

批注 [t32R31]: 就是我们之前说的高中低内置 安全

3.7.4 修改防护策略

请求接口	http(s)://ip:port/v2/protectedPolicy?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 修改防护策略与添加防护策略结构体一致 此处以桥模式为例 POST参数 { "name": "default", //站点防护策略名称，最大长度 64 字节 "mode": 0, //0-桥模式，1-代理模式，2-单臂模式，3-路由模式 "ip": "", //代理 ip（代理模式和单臂模式填充），最大长度 64 字节 "protocal": http, //应用类型 "port": 80, //代理端口（代理模式和单臂模式填充） "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "local_cert": "", //本地证书，长度 64 字节

批注 [P33]: 修改时的主键是什么？ 是名称吗？

批注 [t34R33]: 是名称

	"ssl_version": "SSL3.0", //SSL 版本，10 字节 "cipher_suit": "", //算法套件，1024 字节 "real_server" : [{ "type" : 0, //真实服务器类型，0-主机，1-子网，2-范围，10-ipv6 主机，11-ipv6 子网，12-ipv6 范围 "host" : "192.168.53.12", //type 为主机，64 字节，格式如：192.168.54.191 "net" : "192.168.54.191/32", //type 为子网，64 字节，格式：192.168.54.191/32 "range1" : "", //type 范围，64 字节，起始 ip 地址 "range2" : "", //type 范围，64 字节，结束 ip 地址 "port": 80, //真实服务器端口 "protocal": http, //协议，8 字节 "ssl_enable": 0, //是否开启 ssl 配置，0-不开启，1-开启 "ssl_version" : "SSL3.0" //SSL 版本，10 字节 }],

	"website_safe": "high_level", //站点安全名称 "doname": "" //域名，字符串 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.5 删除防护策略

请求接口	http(s)://ip:port/v2/protectedPolicy?action=del&name=xxx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 name必须填写
返回码	200
返回格式	application/json;charset=utf-8

批注 [P35]: 按 name 删除一个防护站点后，还能再添加同样名称的站点吗？

批注 [r36R35]: 可以，name 是主键，不能重复，删除后可以添加同样名称的站点

返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.7.6 启用/禁用防护策略

请求接口	http(s)://ip:port/v2/protectedPolicyState?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"name": "porxy", // 启用/禁用的防护策略名字
	"disable": 1, //1-不启用，0-启用
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.7.7 新建站点安全

批注 [r37]: 新接口

请求接口	http(s)://ip:port/v2/siteSecurity?action=add
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"name": "123", // 站点名称
	"domain": "", //站点域名
	"file_path": "", //填写空
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.8 删除站点安全

请求接口	http(s)://ip:port/v2/siteSecurity?action=del&name=xxx
请求方式	GET

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 name为站点名称
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.7.9 查看站点安全或查看所有站点安全列表

请求接口	http(s)://ip:port/v2/siteSecurity?action=show&name=xxx http(s)://ip:port/v2/siteSecurity?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 name为站点名称
返回码	200
返回格式	application/json;charset=utf-8
返回数据	url中有name时正确数据：

	{
	"result" : 0,
	"site_security" : {
	"name": "default", // 站点名称
	"domain": "", //站点域名
	"req_param": "", //占位符，无意义
	//以下功能项 0-不启用，1-启用
	"site_taking": "0", //网站锁，
	"protect_param": "1", //HTTP 协议合规锁
	"slow_attack": "1", //慢攻击
	"anti_escape": "1", //防逃逸
	"cookie_protect": "0", //cookie 防护
	"appd_ppr": "0", //网页防篡改
	"cc_protect": "0", //HTTP Flood(cc)防护
	"csrf_protect": "1", //CSRF 防护
	"sql_inject_protect": "1", //SQL 注入防护
	"xss_protect": "1", //XSS 攻击防护
	"hot_link_protect": "1", //盗链防护
	"access_ctrl ": "0", //URL 访问控制
	"url_flow_limit": "0", //URL 流量控制
	"trojan_protect": "1", //网页挂马防护

	"violence_scan": "1", //暴力破解防护
	"spider_protect": "1", //WEB 恶意扫描防护
	"file_filter": "1", //文件上传/下载过滤
	"cashespeed": "1", //缓存加速
	"keywordfilter": "1", //WEB 表单关键字过滤
	"cid_check_url": "1", //客户端访问控制
	" ip_check ": "1", //源区域访问控制
	"xml_dos": "1", //XML Dos 防护
	"weak_pwd": "1", //弱口令防护
	"apt_detect": "1", //恶意样本检测
	"appd_psm ": "0", //敏感信息防护
	"virtual_patch": "0", //虚拟补丁
	"business_check": "0", //业务合规
	"ref_flag": "1" //引用次数，大于 0 不允许删
	}
	
	}
	url 中没有 name 时
	{
	"result": "0",
	"site_security_list": [{

	"name": "default"
	}, {
	"name": "lr"
	}]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.7.10 查看暴力破解、扫描防护

请求接口	http(s)://ip:port/v2/scanProtect?action=show&profile=xx&module=1
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 暴力破解和扫描防护共用GET请求 profile为对应的站点安全名 module=1为暴力破解，module=0为扫描防护
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {

	"result" : 0,
	"profile": "default", // 站点名称
	"sub_mod": {
	"type": 1, //类型，1-爬虫，2-CGI 扫描，3-漏洞扫描，4-暴力破解
	"action": 0, //执行动作，0-通过，1-阻断
	"enable": 1, //开关，0-关闭，1-开启
	
	//以下为默认值
	"log_enable": 1, //日志开关，0-关闭，1-开启
	"log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警
	"acl_block_time": 0, //阻断时间(分钟)
	"detect_level ": 0, //检测等级，0-低，1-中，2-高，3-最高。(只有扫描防护有此节点，暴力破解不需要这个节点)
	"resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无
	"ipex_enable": 0, //IP 例外开关，0-关闭，1-开启
	"ipex_addr_obj_name":"any" , //IP 例外对象名，只有当 ipex_enable 开启时可配置(地址对象名)
	},

	"protect_url": "" // 被保护的 url(暴力破解才有此节点) } 错误数据返回请参见**返回错误数据对应内容**

3.7.11 修改暴力破解、扫描防护(有默认值)

请求接口	http(s)://ip:port/v2/scanProtect?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	暴力破解和扫描防护共用一个xml
	POST参数
	{
	"profile": "default", // 站点名称
	"sub_mod": {
	"type": 1, //类型，1-爬虫，2-CGI 扫描，3-漏洞扫描，4-暴力破解
	"action": 0, //执行动作，0-通过，1-阻断
	"enable": 1, //开关，0-关闭，1-开启

	//以下为默认值
	"log_enable": 1, //日志开关，0-关闭，1-开启
	"log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警
	"acl_block_time": 0, //阻断时间(分钟)
	"detect_level ": 1, //检测等级，0-低，1-中，2-高，3-最高。(只有扫描防护有此节点，暴力破解不需要这个节点)
	"resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无
	"ipex_enable": 0, //IP 例外开关，0-关闭，1-开启
	"ipex_addr_obj_name":"" , //IP 例外对象名，只有当 ipex_enable 开启时可配置(地址对象名)
	},
	"protect_url": [// 被保护的 url(暴力破解才有此节点)
	//此字段填空
	{
	"url": "/abc", //被保护 url
	"usr": "test", //用户参数
	"pwd": "test", //密码参数，明文
	}
	]

	}
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.12 查看慢速攻击

请求接口	http(s)://ip:port/v2/slowAttack?action=show&profile=x x
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 profile为对应的站点安全名
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0, "profile": "default", // 站点名称

	"enable": 1, //开关，0-关闭，1-开启 "action": 0, //动作，0-通过，1-丢弃 } 错误数据返回请参见**返回错误数据对应内容**

3.7.13 修改慢速攻击(有默认值)

请求接口	http(s)://ip:port/v2/slowAttack?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"profile": "default", // 站点名称
	"enable": 1, //开关，0-关闭，1-开启
	"action": 0, //动作，0-通过，1-丢弃
	
	//以下为默认值
	"log_enable": 1, //日志开关，0-关闭，1-开启
	"log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警

	"detect_sensitive_level ": 0, //检测敏感度，0-低，1-中，2-高，3-自定义 "detect_type ": 0, // "detect_cycle ": 0, //异常报文间隔（只能是 1-120 单位秒），只有检测敏感度是 3 时，才填写此字段 "packet_count ": 0, //整型，异常报文个数(只能是 1-30，单位个），只有检测敏感度是 3 时，才填写此字段 "resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 "ipex_enable": 0, //IP 例外开关，0-关闭，1-开启 "ipex_param":"" //IP 例外对象名，只有当 ipex_enable 开启时可配置(地址对象名) }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.14 查看 SQL 注入防护

请求接口	http(s)://ip:port/v2/sqlProtect?action=show&profile=xx&type=SQL
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 profile为对应的站点安全名 type为SQL
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result" : 0,
	"profile": "default", // 站点名称
	"sign_enable": 1, //开关，0-关闭，1-开启
	"sign_action": 0, //动作，0-通过，1-丢弃，4-返回错误页面 6-返回重定向 URL
	}
	错误数据返回请参见**返回错误数据对应内容**

3.7.15 修改 SQL 注入防护(有默认值)

请求接口	http(s)://ip:port/v2/sqlProtect?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"profile": "default"， // 站点名称
	"sign_enable": 1， //开关，0-关闭，1-开启
	"sign_action": 1， //动作，0-通过，1-丢弃，4-返回错误页面 6-返回重定向 URL
	
	//以下为默认值
	"log_enable": 1, //日志开关，0-关闭，1-开启
	"log_level": 1, //日志级别
	"raw_pkt_enable": 0, //是否提取原始报文
	"raw_pkt_time": 60, // 提取最长时间
	"raw_pkt_size": 1024, // 提取文件大小
	"record_header_enable": 0, //是否提取请求头信息
	"database": 31, //
	"acl_block_time": 0, //

	"resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 "ipex_enable": 0, //IP 例外开关，0-关闭，1-开启 "ipex_param":"" //IP 例外对象名，只有当 ipex_enable 开启时可配置(地址对象名) "exception_url": ""// }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.16 查看 XSS 攻击防护

请求接口	http(s)://ip:port/v2/xssProtect?action=show&profile=xx&type=XSS
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

	profile为对应的站点安全名 type为XSS
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result" : 0,
	"profile": "default", // 站点名称
	"sign_enable": 1, //开关，0-关闭，1-开启
	"sign_action": 1, //动作，0-通过，1-丢弃，4-返回错误页面 6-返回重定向 URL
	}
	错误数据返回请参见**返回错误数据对应内容**

3.7.17 修改 XSS 攻击防护(有默认值)

请求接口	http(s)://ip:port/v2/xssProtect?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 {

	"profile": "default", // 站点名称
	"sign_enable": 1, //开关，0-关闭，1-开启
	"sign_action": 0, //动作，0-通过，1-丢弃，4-返回错误页面 6-返回重定向 URL
	
	//以下为默认值
	"log_enable": 1, //日志开关，0-关闭，1-开启
	"log_level": 1, //日志级别
	"raw_pkt_enable": 0, //是否提取原始报文
	"raw_pkt_time": 60, // 提取最长时间
	"raw_pkt_size": 1024, // 提取文件大小
	"record_header_enable": 0, //是否提取请求头信息
	"database": 65535, //
	"acl_block_time": 0, //
	"resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无
	"ipex_enable": 0, //IP 例外开关，0-关闭，1-开启
	"ipex_param":"" //IP 例外对象名，只有当 ipex_enable 开启时可配置(地址对象名)
	"exception_url": ""//
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.18 查看敏感信息防护

请求接口	http(s)://ip:port/v2/appdPsm?action=show&profile=xx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 profile为对应的站点安全名
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0, "profile": "default", // 站点名称

	"enable": 1, //总开关，0-关闭，1-开启 "os_hiden": 1, //操作系统信息防护开关，0-关闭，1-开启 "web_server_hiden": 1, //服务器信息防护开关，0-关闭，1-开启 "web_error_hiden": 0, //错误页面信息防护开关，0-关闭，1-开启 "errorcode_customize": 503, //错误码，只能填写200,403,404,503 "card_hiden": 0, //银行卡信息防护开关，0-关闭，1-开启 "czn_hiden": 0, //身份证信息防护开关，0-关闭，1-开启 } 错误数据返回请参见**返回错误数据对应内容**

3.7.19 修改敏感信息防护(有默认值)

请求接口	http(s)://ip:port/v2/appdPsm?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

	POST参数
	{
	"profile": "default", // 站点名称
	"enable": 1, //总开关，0-关闭，1-开启
	"os_hiden": 1, //操作系统信息防护开关，0-关闭，1-开启
	"web_server_hiden": 1, //服务器信息防护开关，0-关闭，1-开启
	"web_error_hiden": 0, //错误页面信息防护开关，0-关闭，1-开启
	"errorcode_customize": 503, //错误码，只能填写200,403,404,503
	"card_hiden": 0, //银行卡信息防护开关，0-关闭，1-开启
	"czn_hiden": 0, //身份证信息防护开关，0-关闭，1-开启
	
	//以下为默认值
	"resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无
	"ftp_enable_flag": 0,

	"psm_log": 1, "psm_level": 6, "psm_replace_word": "x", "reg_pro": 0, "ftpserverinfo_hide": 0, "wag_page_flag": 0, // "reg_list":{} }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result"　: 0} 错误数据返回请参见**返回错误数据对应内容**

3.7.19 修改事件引擎

请求接口	http(s)://ip:port/v2/detectEvt?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

	POST参数
	{
	"set_name": "default", // 站点名称，必填
	"first_class_id": [
	//需要开启的一级事件集 一共 1-8 个一级事件分别对应拒绝服务、木马蠕虫、爬虫扫描、通用攻击、信息泄露、溢出攻击、注入攻击、其他类型。如想开启哪个事件，请将 level_id 按照如下形式填写。不填写的即为不启用。如下，开启 1，5，6，8
	{"level_id": 1},
	{"level_id": 5},
	{"level_id": 6},
	{"level_id": 8}
	],
	//以下字段无需关注，不需要传输
	"enable": 1, //总开关，0-关闭，1-开启
	"domainname": "", //填写空即可
	"mode": 3, //模式，下发默认值
	"search_str": "", //需要搜索的事件名称，下发默认值
	"action": "", //需要搜索的动作，下发默认值
	"sig_level": , //事件级别，下发默认值

	"sec_class_id": , //下发默认值 "sig_evt_id": , //下发默认值 "unchange_list": //下发默认值 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.8 SSL 证书配置

3.8.1 查看证书

请求接口	http(s)://ip:port/v2/certInfo?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result" : 0
	"certs" : [
	{
	// 证书 1
	"local_cert_name": "no1", //证书名称
	"local_cert_state": 1, //证书状态，1-正常，0-未确定
	"local_cert_locate": 0, //位置， 1-USB key，0-本地文件
	"issuer": "C=cn, ST=Some-State, O=Internet Widgits Pty Ltd, CN=ca", //证书发行者
	"subject": "C=cn, ST=Some-State, O=Internet Widgits Pty Ltd, CN=no1", //主题
	"start_time": "Dec 15 16:33:27 2014 GMT", //开始时间
	"end_time": "Mar 3 16:33:27 2023 GMT", //结束时间
	"version ": "3", //版本
	"serial_num": "2102112123", //序列号
	"ext_info": "X509v3 Subject Key Identifier:

%9", //序列号

	"ref": "1", //证书引用次数，如果大于 0 则改证书允许删除
	},
	{
	// 证书 2
	"local_cert_name": "no2", //证书名称
	"local_cert_state": 1, //证书状态，1-正常，0-未确定
	"local_cert_locate": 0, //位置， 1-USB key，0-本地文件
	"issuer": "C=cn, ST=Some-State, O=Internet Widgits Pty Ltd, CN=ca_user", //证书发行者
	"subject": "C=cn, ST=Some-State, O=Internet Widgits Pty Ltd, CN=no2", //主题
	"start_time": "Dec 15 16:33:27 2014 GMT", //开始时间
	"end_time": "Mar 3 16:33:27 2023 GMT", //结束时间
	"version ": "3", //版本
	"serial_num": "2102112126", //序列号
	"ext_info": "X509v3 Subject Key Identifier:

%9", //序列号

	"ref": "1", //证书引用次数，如果大于 0 则改证书允许删除 }] } 错误数据返回请参见**返回错误数据对应内容**

3.8.2 删除证书

请求接口	http(s)://ip:port/v2/certInfo?action=del&name=xxx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 name为证书名称
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.8.3 导入证书

请求接口	http(s)://ip:port/v2/certInfo?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"cert_file_name": "no1.crt", //证书文件名
	"key_file_name": "no1.key", //密钥文件名
	"type": 3, //证书类型，2-PKCS12，3-证书密钥分离
	"password": "xxxx"，//证书密码
	"cert_file_content": "xxxx"，//证书内容，格式为读取文件后 base64 编码后的内容
	"key_file_content": "xxxx"，//密钥内容，同上
	"upload_location": 0，//上传路径，0-本地，1-USB key
	}

返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.9 事件库升级

3.9.1 事件库立即升级

请求接口	http(s)://ip:port/v2/eventbaseupdate?action=add
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"server_style": 0,//服务器类型，0-默认升级服务器，1-指定升级服务器
	"server_addr": "", //升级地址，如果 style 为 0，此字段为空
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据：

	{"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.9.2 获取事件库升级日志

请求接口	http(s)://ip:port/v2/eventbaselogs?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result" : 0,
	"logs" : [
	{
	"version": " V0700R0703B20170309" , //更新版本
	"up_time":"2017-03-09 19:00:15", //升级时间
	"up_name":0 //0-系统软件，1-预定义事件库升级，2-恶意 URL 云库升级
	}

	] } 错误数据返回请参见**返回错误数据对应内容**

3.10 安全事件统计

3.10.1 获取安全事件统计

请求接口	http(s)://ip:port/v2/securityEventStat?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: { "result" : 0 "attack" : [//按事件名称统计top10 { "HTTP_SQL注入攻击 " : "180", //事件数 "HTTP业务合规" : "65", //事件数

	"HTTP_url躲避"：10"//事件数 }] } 错误数据返回请参见**返回错误数据对应内容**

3.11 Flood 防护

3.11.1 查看智能 TCP FLOOD 防御配置

请求接口	http(s)://ip:port/v2/antiAttack?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result" : 0,
	"anti_attack": {

	"sys_cookie_enable": 1, //智能 tcp 防御功能开关，0-关闭，1-开启 "tcp_half": 2000, //TCP Flood 识别阈值(10-10000)，单位秒 "log_enable": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 } } 错误数据返回请参见**返回错误数据对应内容**

3.11.2 修改智能 TCP FLOOD 防御配置

请求接口	http(s)://ip:port/v2/antiAttack?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 {

	"sys_cookie_enable": 1, //智能 tcp 防御功能开关，0-关闭，1-开启 "tcp_half": 2000, //TCP Flood 识别阈值(10-10000)，单位秒 "log_enable": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.11.3 查看所有防护策略

请求接口	http(s)://ip:port/v2/shieldEntry?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0, "shield_entry": [{ "name":"flood1", //防护策略名称 "desc":"test", //描述 "tcp_src_con_enable": 1, //TCP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "tcp_src_con_limit": 1000, //TCP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "tcp_dst_con_enable": 1, //TCP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "tcp_dst_con_limit": 1000, //TCP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_src_con_enable": 1, //UDP Flood 防护对源主机进行流限制开关，0-关闭，1-开启

	"udp_src_con_limit": 1000, //UDP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_dst_con_enable": 1, //UDP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "udp_dst_con_limit": 1000, //UDP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_src_con_enable": 1, //ICMP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 " icmp_src_con_limit": 1000, //ICMP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_dst_con_enable": 1, //ICMP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "icmp_dst_con_limit": 1000, //ICMP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "action": 0, //响应动作，0-通过，2-丢弃/阻断 "block_time": 30, //阻断时间（1-60 分钟），当 action 是 2 时才生效 "log_enable": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警

	"resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 "ref": 1 //引用次数，大于 0 时，不允许删除 }] } 错误数据返回请参见**返回错误数据对应内容**

3.11.4 查看单个防护策略

请求接口	http(s)://ip:port/v2/shieldEntry?action=show&name=xx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 name为Flood防护策略的名称
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0, "shield_entry": [

	{ "name":"flood1", //防护策略名称 "desc":"test", //描述 "tcp_src_con_enable": 1, //TCP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "tcp_src_con_limit": 1000, //TCP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "tcp_dst_con_enable": 1, //TCP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "tcp_dst_con_limit": 1000, //TCP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_src_con_enable": 1, //UDP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "udp_src_con_limit": 1000, //UDP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_dst_con_enable": 1, //UDP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "udp_dst_con_limit": 1000, //UDP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_src_con_enable": 1, //ICMP Flood 防护对源主机进行流限制开关，0-关闭，1-开启

	" icmp_src_con_limit": 1000, //ICMP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_dst_con_enable": 1, //ICMP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "icmp_dst_con_limit": 1000, //ICMP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "action": 0, //响应动作，0-通过，2-丢弃/阻断 "block_time": 30, //阻断时间（1-60 分钟），当 action 是 2 时才生效 "log_enable": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 "ref": 1 //引用次数，大于 0 时，不允许删除 }] } 错误数据返回请参见**返回错误数据对应内容**

3.11.5 添加防护策略

请求接口	http(s)://ip:port/v2/shieldEntry?action=add
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 { "name":"flood1", //防护策略名称 "desc":"test", //描述 "tcp_src_con_enable": 1, //TCP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "tcp_src_con_limit": 1000, //TCP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "tcp_dst_con_enable": 1, //TCP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "tcp_dst_con_limit": 1000, //TCP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_src_con_enable": 1, //UDP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "udp_src_con_limit": 1000, //UDP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒)

	"udp_dst_con_enable": 1, //UDP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "udp_dst_con_limit": 1000, //UDP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_src_con_enable": 1, //ICMP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 " icmp_src_con_limit": 1000, //ICMP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_dst_con_enable": 1, //ICMP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "icmp_dst_con_limit": 1000, //ICMP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "action": 0, //响应动作，0-通过，2-丢弃/阻断 "block_time": 30, //阻断时间（1-60 分钟），当 action 是 2 时才生效 "log_enable": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 }

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.11.6 修改防护策略

请求接口	http(s)://ip:port/v2/shieldEntry?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 { "name":"flood1", //防护策略名称 "desc":"test", //描述 "tcp_src_con_enable": 1, //TCP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "tcp_src_con_limit": 1000, //TCP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒)

	"tcp_dst_con_enable": 1, //TCP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "tcp_dst_con_limit": 1000, //TCP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_src_con_enable": 1, //UDP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 "udp_src_con_limit": 1000, //UDP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "udp_dst_con_enable": 1, //UDP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "udp_dst_con_limit": 1000, //UDP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_src_con_enable": 1, //ICMP Flood 防护对源主机进行流限制开关，0-关闭，1-开启 " icmp_src_con_limit": 1000, //ICMP 每台源主机进行限制流大小(只能是 1-10000 连接数/秒) "icmp_dst_con_enable": 1, //ICMP Flood 防护对目的主机进行流限制开关，0-关闭，1-开启 "icmp_dst_con_limit": 1000, //ICMP 每台目的主机进行限制流大小(只能是 1-10000 连接数/秒) "action": 0, //响应动作，0-通过，2-丢弃/阻断

	"block_time": 30, //阻断时间（1-60 分钟），当 action 是 2 时才生效 "log_enable": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "resp": 4224, //响应方式，4224-邮件和短信，4096-短信，128-邮件，0-无 "ref": 1 //引用次数，大于 0 时，不允许删除 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.11.7 删除单个防护策略

请求接口	http(s)://ip:port/v2/shieldEntry?action=del&name=xx
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建

	议放到url里）。 name为Flood防护策略的名称
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.11.8 查看所有 Flood 防护 预留需要平台修改地址对象

请求接口	http(s)://ip:port/v2/antifloodEntry?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" : 0

	} 错误数据返回请参见**返回错误数据对应内容**

3.12 软件 Bypass

3.12.1 修改软件 Bypass

请求接口	http(s)://ip:port/v2/vsSoftBypass?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	POST参数
	{
	"bypass":1 //开关，0-关闭，1-开启
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.13 黑/白名单配置

3.13.1 获取所有黑/白名单

请求接口	http(s)://ip:port/v2/fwBlackList?action=show(黑名单) http(s)://ip:port/v2/fwWhiteList?action=show(白名单)
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: { "result" : 0, "fw_list" : [{ "name":"123", //名称 "if_in":"any", //作用域接口

	"src_addr_type": 0, //源地址类型，0-主机，1-网段，2-地址范围，10-ipv6 主机，11-ipv6 网段，12-ipv6 地址范围 "src_addr_host": "10.10.12.32", //源地址主机地址 "src_addr_net": "10.10.12.1/24", //源地址网段地址 "src_addr_range1": "11.24.53.1", //源地址范围开始地址 "src_addr_range2": "11.24.53.254", //源地址范围结束地址 "dst_addr_type": 0, //目的地址类型，0-主机，1-网段，2-地址范围，10-ipv6 主机，11-ipv6 网段，12-ipv6 地址范围 "dst_addr_host": "10.10.12.32", //目的地址主机地址 "dst_addr_net": "10.10.12.1/24", //目的地址网段地址 "dst_addr_range1": "11.24.53.1", //目的地址范围开始地址 "dst_addr_range2": "11.24.53.254", //目的地址范围结束地址 "dst_port": 80, //目的端口 "set_periodic": 0, //时间设置开关，0-关闭，1-开启

	"week_day": 1, //1、2、3、4、5、6、7 分别代表周一到周日 "day_enable_time": "1-5", //时间段，表示几点到几点(24 小时制) "log": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "enable": 1, //开关，0-关闭，1-开启 }] } 错误数据返回请参见**返回错误数据对应内容**

3.13.2 添加/修改黑/白名单

请求接口	http(s)://ip:port/v2/fwBlackList?action=add(黑名单) http(s)://ip:port/v2/fwBlackList?action=mod(黑名单) http(s)://ip:port/v2/fwWhiteList?action=add(白名单) http(s)://ip:port/v2/fwWhiteList?action=mod(白名单)
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建

	议放到url里）。 { "name":"123", //名称 "if_in":"any", //作用域接口 "src_addr_type": 0, //源地址类型，0-主机，1-网段，2-地址范围，10-ipv6 主机，11-ipv6 网段，12-ipv6 地址范围 "src_addr_host": "10.10.12.32", //源地址主机地址 "src_addr_net": "10.10.12.1/24", //源地址网段地址 "src_addr_range1": "11.24.53.1", //源地址范围开始地址 "src_addr_range2": "11.24.53.254", //源地址范围结束地址 "dst_addr_type": 0, //目的地址类型，0-主机，1-网段，2-地址范围，10-ipv6 主机，11-ipv6 网段，12-ipv6 地址范围 "dst_addr_host": "10.10.12.32", //目的地址主机地址 "dst_addr_net": "10.10.12.1/24", //目的地址网段地址 "dst_addr_range1": "11.24.53.1", //目的地址范围开始地址

	"dst_addr_range2": "11.24.53.254", //目的地址范围结束地址 "dst_port": 80, //目的端口 "set_periodic": 0, //时间设置开关，0-关闭，1-开启 "week_day": 1, //1、2、3、4、5、6、7 分别代表周一到周日 "day_enable_time": "1-5", //时间段，表示几点到几点(24 小时制) "log": 1, //日志开关，0-关闭，1-开启 "log_level": 6, //日志级别，6-信息，5-通知，4-警示，1-告警 "enable": 1, //开关，0-关闭，1-开启 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.13.3 删除黑/白名单

请求接口	http(s)://ip:port/v2/fwBlackList?action=del&name=xx(黑名单) http(s)://ip:port/v2/fwWhiteList?action= del&name=xx(白名单)
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 name为黑/白名单的名称
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.13.4 开启/关闭黑/白名单

请求接口	http(s)://ip:port/v2/fwListState?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

	{
	"result" ：0,
	"mode" ：2, //1-白名单，2-黑名单
	"name" ："test", //需要开启/关闭名单的名称，不能为空
	"enable" ：1， //开关，0-关闭，1-开启
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.14 网关配置

3.14.1 获取所有网关

请求接口	http(s)://ip:port/v2/gateList?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result" : 0,
	"gate_list" : [//可以是多个网关
	{
	"gate_ip":"1.1.1.1", //网关 IP 地址
	"oif ":"", //出接口，无
	"get_style": 0, //默认值
	"distance": 1, //管理距离(1-255)
	"weight":1, //权重(1-100)
	"ip_ver":4 //IP 类型，4-IPv4，6-IPv6
	}
	]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.14.2 添加网关

请求接口	http(s)://ip:port/v2/gateList?action=add
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"gate_ip":"1.1.1.1", //网关 IP 地址
	"distance": 1, //管理距离(1-255)
	"weight":1, //权重(1-100)
	"ip_ver":4 //IP 类型，4-IPv4，6-IPv6
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0,} 错误数据返回请参见**返回错误数据对应内容**

3.14.3 删除网关

请求接口	http(s)://ip:port/v2/gateList?action=del
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

	{
	"gate_ip":"1.1.1.1", //网关 IP 地址
	"oif ":"", //出接口，保留值，不填写值
	"ip_ver":4 //IP 类型，4-IPv4，6-IPv6
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0,} 错误数据返回请参见**返回错误数据对应内容**

3.15 静态路由配置

3.15.1 获取所有静态路由

请求接口	http(s)://ip:port/v2/staticRouteList?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result" : 0,
	"static_route_list" : [//可以是多个静态路由
	{
	"dst_ip":"0.0.0.0/0", //网段及掩码
	"nh_type ":0, //下一跳类型，0-下一跳，1-出接口
	"nh_ip": "192.168.1.2", //如果 nh_type=0，则表示下一跳 IP 地址；如果 nh_type=1，则表示出接口名称，比如 ETH1
	"distance": 1, //管理距离(1-255)
	"weight":1, //权重(1-100)
	}
	]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.15.2 添加静态路由

请求接口	http(s)://ip:port/v2/staticRouteList?action=add
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"dst_ip":"0.0.0.0/0", //网段及掩码
	"nh_type ":0, //下一跳类型，0-下一跳，1-出接口
	"nh_ip": "192.168.1.2", //如果 nh_type=0，则表示下一跳 IP 地址；如果 nh_type=1，则表示出接口名称，比如 ETH1
	"distance": 1, //管理距离(1-255)
	"weigh":1, //权重(1-100)
	"ip_ver":4 //IP 类型，4-IPv4，6-IPv6
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据：

	{"result"　: 0,} 错误数据返回请参见**返回错误数据对应内容**

3.15.3 修改路由

请求接口	http(s)://ip:port/v2/staticRouteList?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 只允许修改权重和管理距离，其他项必须在路由表里有 { "dst_ip":"0.0.0.0/0",　//网段及掩码(修改时不能改此项) "nh_type ":0,　//下一跳类型，0-下一跳，1-出接口(修改时不能改此项) "nh_ip": "192.168.1.2", //如果 nh_type=0，则表示下一跳 IP 地址；如果 nh_type=1，则表示出接口名称，比如 ETH1(修改时不能改此项) "distance": 1, //管理距离(1-255) "weigh":1, //权重(1-100)

	"ip_ver":4 //IP 类型，4-IPv4，6-IPv6(修改时不能改此项) }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0,} 错误数据返回请参见**返回错误数据对应内容**

3.15.4 删除路由

请求接口	http(s)://ip:port/v2/staticRouteList?action=del
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "dst_ip":"0.0.0.0/0", //网段及掩码 "nh_type ":0, //下一跳类型，0-下一跳，1-出接口

	"nh_ip": "192.168.1.2", //如果 nh_type=0，则表示下一跳 IP 地址；如果 nh_type=1，则表示出接口名称，比如 ETH1 "ip_ver":4 //IP 类型，4-IPv4，6-IPv6 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0,} 错误数据返回请参见**返回错误数据对应内容**

3.16 设备参数

3.16.1 设备重启

请求接口	http(s)://ip:port/v2/powerOff?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 POST参数 {

	"operation":0, //重启参数，必须是 0 }
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.16.2 保存配置

请求接口	http(s)://ip:port/v2/saveConfig?action=mod
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.17 地址对象

3.17.1 获取所有地址对象

请求接口	http(s)://ip:port/v2/addressObject?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result": "0",
	"data": [{
	"name": "any", //名称
	"desc": "", //描述
	"ref": "1", //引用次数
	"item": [{ //多个地址
	"type": "1", //ipv4 主机：0，子网：1，范围：2， ipv6 主机：10，子网：11，范围：12

```
                "host": "",
                "net": "0.0.0.0/0",
                "range1": "",
                "range2": ""
            }, {
                "type": "11",
                "host": "",
                "net": "::0",
                "range1": "",
                "range2": ""
            }]
        }, {
            "name": "zng",
            "desc": "666666",
            "ref": "0",
            "item": [{
                "type": "0",
                "host": "192.168.58.61",
                "net": "",
                "range1": "",
                "range2": ""
```

	}, {
	"type": "0",
	"host": "192.168.58.63",
	"net": "",
	"range1": "",
	"range2": ""
	}]
	}]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.17.2 获取单个地址对象

请求接口	http(s)://ip:port/v2/addressObject?action=show&name=zhangheng
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据：

	{
	"result": "0",
	"data": [{
	"name": "zng",
	"desc": "666666",
	"ref": "0",
	"item": [{
	"type": "0", //ipv4 主机：0，子网：1，范围：2， ipv6 主机：10，子网：11，范围：12
	"host": "192.168.58.61",
	"net": "",
	"range1": "",
	"range2": ""
	}, {
	"type": "0",
	"host": "192.168.58.63",
	"net": "",
	"range1": "",
	"range2": ""
	}]
	}]

	} 错误数据返回请参见**返回错误数据对应内容**

3.17.3 添加地址对象

请求接口	http(s)://ip:port/v2/addressObject?action=add
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"name": "zng",
	"desc": "fff",
	"item": [{
	"type": "0", //ipv4 主机：0，子网：1，范围：2，ipv6 主机：10，子网：11，范围：12
	"host": "192.168.58.61"
	}, {
	"type": "0",
	"host": "192.168.58.63"
	}]
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.17.4 修改地址对象

请求接口	http(s)://ip:port/v2/addressObject?action=mod
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"name": "zng",
	"desc": "fff",
	"item": [{
	"type": "0",
	"host": "192.168.58.61"
	}, {
	"type": "0",
	"host": "192.168.58.63"
	}]

	}
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.17.5 删除地址对象

请求接口	http(s)://ip:port/v2/addressObject?action=delete
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"name": "zng"
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.18 虚拟服务

3.18.1 获取所有虚拟服务

请求接口	http(s)://ip:port/v2/virtualService?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result": "0",
	"data": [{
	"name": "vs_default",
	"mode": "0",
	"ip_ver": "",
	"ip": "",
	"port": "",
	"protocal": "http",
	"disable": "0",

	"profile": "default",
	"website_safe": "default",
	"addr_obj": "any",
	"serv_obj": "all",
	"real_server_protocol": "http"
	}, {
	"name": "liuri",
	"mode": "0",
	"ip_ver": "",
	"ip": "",
	"port": "",
	"protocal": "http",
	"disable": "0",
	"profile": "default",
	"website_safe": "default",
	"addr_obj": "any",
	"serv_obj": "all",
	"real_server_protocol": "http"
	}]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.18.2 获取单个虚拟服务

请求接口	http(s)://ip:port/v2/virtualService?action=show&name=liuri
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据:
	{
	"result": "0",
	"data": [{
	"name": "liuri",
	"mode": "0",
	"ip_ver": "",
	"ip": "",
	"port": "",
	"protocal": "http",
	"health_enable": "",

	"profile": "default", "website_safe": "default", "doname": "", "ssl_enable": "0", "local_cert": "", "enc_cert": "", "sign_cert": "", "gm_enable": "0", "load_balance": "0", "cipher_suit": "", "real_server": [{ "addr_obj": "any", "serv_obj": "all", "protocal": "http", "weight": "0", "ssl_enable": "0", "enc_cert": "", "sign_cert": "", "gm_enable": "0", "local_cert": "", "cipher_suit": ""

	}], "ssl_protect": [{ "url": "" }] }] }错误数据返回请参见**返回错误数据对应内容**

3.18.3 添加虚拟服务

请求接口	http(s)://ip:port/v2/virtualService?action=add
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "name": "liuri", "mode": "0", "ip_ver": "", "ip": "", "port": "", "protocal": "http", "health_enable": "",

	"profile": "default", "website_safe": "default", "doname": "", "ssl_enable": "0", "local_cert": "", "enc_cert": "", "sign_cert": "", "gm_enable": "0", "load_balance": "1", "cipher_suit": "", "real_server": [{ "addr_obj": "any", "serv_obj": "all", "protocal": "http", "weight": "0", "ssl_enable": "0", "enc_cert": "", "sign_cert": "", "gm_enable": "0", "local_cert": "", "cipher_suit": ""

	}],
	"ssl_protect": [{
	"url": ""
	}]
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.18.4 修改虚拟服务

请求接口	http(s)://ip:port/v2/virtualService?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "name": "liuri", "mode": "0", "ip_ver": "", "ip": "",

	"port": "", "protocal": "http", "health_enable": "", "profile": "default", "website_safe": "default", "doname": "", "ssl_enable": "0", "local_cert": "", "enc_cert": "", "sign_cert": "", "gm_enable": "0", "load_balance": "1", "cipher_suit": "", "real_server": [{ "addr_obj": "any", "serv_obj": "all", "protocal": "http", "weight": "0", "ssl_enable": "0", "enc_cert": "", "sign_cert": "",

	"gm_enable": "0",
	"local_cert": "",
	"cipher_suit": ""
	}],
	"ssl_protect": [{
	"url": ""
	}]
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.18.5 删除虚拟服务

请求接口	http(s)://ip:port/v2/virtualService?action=delete
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "name": "liuri"

	}
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.18.6 开启/关闭虚拟服务

请求接口	http(s)://ip:port/v2/virtualService?action=disable
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"name": "liuri",
	"disable": "0"
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0}

	错误数据返回请参见**返回错误数据对应内容**

3.19 服务对象

3.19.1 获取所有服务对象

请求接口	http(s)://ip:port/v2/serviceObject?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result": "0",
	"data": [{
	"name": "http_80",
	"desc": "",
	"ref": "0",
	"item": [{
	"sev_str": "TCP\/1-65535:80"

```
        }]
    }, {
        "name": "https_443",
        "desc": "",
        "ref": "0",
        "item": [{
            "sev_str": "TCP\/1-65535:443"
        }]
    }, {
        "name": "any",
        "desc": "",
        "ref": "0",
        "item": [{
            "sev_str": "IP\/256"
        }]
    }, {
        "name": "all",
        "desc": "",
        "ref": "2",
        "item": [{
            "sev_str": "TCP\/1-65535:1-65535"
```

	}]
	}, {
	"name": "liuri",
	"desc": "dsfsfsf",
	"ref": "0",
	"item": [{
	"sev_str": "TCP\/1-65535:80"
	}]
	}]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.19.2 获取单个服务对象

请求接口	http(s)://ip:port/v2/serviceObject?action=show&name=zhangheng
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result": "0",
	"data": [{
	"name": "liuri",
	"desc": "dsfsfsf",
	"ref": "0",
	"item": [{
	"sev_str": "TCP\/1-65535:80"
	}]
	}]
	}
	错误数据返回请参见**返回错误数据对应内容**

3.19.3 添加服务对象

请求接口	http(s)://ip:port/v2/serviceObject?action=add
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 {

	"name": "liuri",
	"desc": "dsfsfsf",
	"ref": "0",
	"item": [{
	"sev_str": "TCP\/1-65535:80"
	}]
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据: {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.19.4 修改服务对象

请求接口	http(s)://ip:port/v2/serviceObject?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "name": "liuri", "desc": "dsfsfsf",

	"ref": "0",
	"item": [{
	"sev_str": "TCP\/1-65535:80"
	}]
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.19.5 删除服务对象

请求接口	http(s)://ip:port/v2/serviceObject?action=delete
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"name": "zng"
	}

返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据： {"result" : 0} 错误数据返回请参见**返回错误数据对应内容**

3.20 web 恶意扫描防护

3.20.1 web 恶意扫描防护修改

请求接口	http(s)://ip:port/v2/webSanProtect?action=mod
请求方式	POST
请求参数	{ "profile": "default", //绑定的站点名称 "sub_mod": [//这里有三类：爬虫防护、CGI扫描防护、漏洞扫描防护，这三项是必有的 { "type": "1", //爬虫防护 "action": "0", // 响应动作 0：通过 ，2：阻断 "enable": "1", // 启用与否, 0：否，1：启用 "acl_block_time": "0", //阻断时间，如果action是2，则阻断时间显示，范围1-60 分钟 "log_enable": "1", //日志启用，0：否，1：启用 "log_level": "6", //日志级别，6：信息，5：通知，

	4：警示，1：告警 "detect_level": "5", // 检测敏感度，0：低，1：中，2：高，3：最高，5：自定义 "resp": "4224", //响应方式，邮件：128，短信：4096，邮件和短信：4224 "ipex_enable": "1", // IP列外是否启用， 0：否，1：启用 "ipex_addr_obj_name": "liuri", //IP例外启用后，显示该项，可以为空，此处数据来源为地址对象 "check_pack_num": "151" //可疑报文观察个数，如果检测敏感度选择自定义，则显示该项，范围100-220 }, { "type": "2", //CGI扫描防护 "action": "0", "enable": "1", "acl_block_time": "0", "log_enable": "1", "log_level": "6", "detect_level": "5", "resp": "4224",

	"ipex_enable": "0",
	"ipex_addr_obj_name": "",
	"check_pack_num": "150"
	},
	{
	"type": "3", //漏洞扫描防护
	"action": "0",
	"enable": "1",
	"acl_block_time": "0",
	"log_enable": "1",
	"log_level": "6",
	"detect_level": "5",
	"resp": "4224",
	"ipex_enable": "1",
	"ipex_addr_obj_name": "",
	"check_pack_num": "150"
	}
	]
	}
	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result"　: 0} 错误数据返回请参见**返回错误数据对应内容**

3.21 事件引擎

3.21.1 获取事件引擎一级二级

请求接口	http(s)://ip:port/v2/detectSignPredef?action=show&level=first
请求方式	POST

请求参数	{
	"set_name": "default",　　//站点名称
	"search_str": "",　　//事件名称
	"action": "",　　//响应动作　0：通过，1：丢弃，2：阻断，4：返回错误页面，6：返回重定向URL
	"sig_level": ""　　//事件级别　10：非攻击事件，20：低级事件，30：中级事件，40：高级事件
	}

	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result": "0",
	"data": [{
	"first_name": "\u901a\u7528\u653b\u51fb",
	"enable_num": "1",
	"num": "1249",
	"first_id": "4",
	"sec_level_class": [{
	"sec_name": "\u6d4f\u89c8\u5668\u653b\u51fb",
	"enable_num": "0",
	"num": "451",
	"sec_id": "43"
	}, {

```
                "sec_name":
"\u7f51\u5b89\u8bbe\u5907\u653b\u51fb",
                "enable_num": "0",
                "num": "31",
                "sec_id": "42"
            }, {
                "sec_name":
"\u7b2c\u4e09\u65b9\u5e94\u7528\u653b\u51fb",
                "enable_num": "1",
                "num": "240",
                "sec_id": "47"
            }, {
                "sec_name": "CMS\u653b\u51fb",
                "enable_num": "0",
                "num": "387",
                "sec_id": "45"
            }, {
                "sec_name":
"\u4e2d\u95f4\u4ef6\u653b\u51fb",
                "enable_num": "0",
                "num": "140",
```

```
                "sec_id": "46"
            }]
        }, {
            "first_name": "\u5176\u4ed6",
            "enable_num": "8",
            "num": "381",
            "first_id": "8",
            "sec_level_class": [{
                "sec_name": "\u5176\u4ed6\u4e8b\u4ef6",
                "enable_num": "8",
                "num": "355",
                "sec_id": "58"
            }, {
                "sec_name":
"\u5f31\u53e3\u4ee4\u4e8b\u4ef6",
                "enable_num": "0",
                "num": "2",
                "sec_id": "56"
            }, {
                "sec_name": "\u53cd\u5e8f\u5217\u5316",
                "enable_num": "0",
```

```
                "num": "23",
                "sec_id": "59"
            }, {
                "sec_name":
"\u81ea\u5b9a\u4e49\u4e8b\u4ef6",
                "enable_num": "0",
                "num": "1",
                "sec_id": "57"
            }]
        }, {
            "first_name": "\u6ce8\u5165\u653b\u51fb",
            "enable_num": "0",
            "num": "196",
            "first_id": "7",
            "sec_level_class": [{
                "sec_name": "\u547d\u4ee4\u6ce8\u5165",
                "enable_num": "0",
                "num": "105",
                "sec_id": "32"
            }, {
                "sec_name": "\u5176\u4ed6\u6ce8\u5165",
```

```
                "enable_num": "0",
                "num": "43",
                "sec_id": "35"
            }, {
                "sec_name": "SQL\u6ce8\u5165",
                "enable_num": "0",
                "num": "45",
                "sec_id": "34"
            }, {
                "sec_name": "XML\u6ce8\u5165",
                "enable_num": "0",
                "num": "3",
                "sec_id": "33"
            }]
        }, {
            "first_name": "\u6728\u9a6c\u8815\u866b",
            "enable_num": "0",
            "num": "1007",
            "first_id": "2",
            "sec_level_class": [{
                "sec_name": "\u6728\u9a6c\u653b\u51fb",
```

```
            "enable_num": "0",
            "num": "449",
            "sec_id": "37"
        }, {
            "sec_name": "Webshell\u653b\u51fb",
            "enable_num": "0",
            "num": "531",
            "sec_id": "36"
        }, {
            "sec_name":
"\u8815\u866b\u75c5\u6bd2\u653b\u51fb",
            "enable_num": "0",
            "num": "15",
            "sec_id": "38"
        }, {
            "sec_name":
"\u95f4\u8c0d\u8f6f\u4ef6\u653b\u51fb",
            "enable_num": "0",
            "num": "12",
            "sec_id": "39"
        }]
```

```
    }, {
        "first_name": "\u6ea2\u51fa\u653b\u51fb",
        "enable_num": "0",
        "num": "101",
        "first_id": "6",
        "sec_level_class": [{
            "sec_name":
"FTP\u670d\u52a1\u5668\u6ea2\u51fa\u653b\u51fb",
            "enable_num": "0",
            "num": "48",
            "sec_id": "41"
        }, {
            "sec_name":
"web\u670d\u52a1\u5668\u6ea2\u51fa\u653b\u51fb",
            "enable_num": "0",
            "num": "53",
            "sec_id": "40"
        }]
    }, {
        "first_name": "\u722c\u866b\u626b\u63cf",
        "enable_num": "1",
```

```
            "num": "806",
            "first_id": "3",
            "sec_level_class": [{
                "sec_name":
"\u654f\u611f\u6587\u4ef6\u8bbf\u95ee",
                "enable_num": "0",
                "num": "601",
                "sec_id": "48"
            }, {
                "sec_name": "\u7f51\u9875\u722c\u866b",
                "enable_num": "1",
                "num": "154",
                "sec_id": "49"
            }, {
                "sec_name": "\u7f51\u7edc\u626b\u63cf",
                "enable_num": "0",
                "num": "51",
                "sec_id": "50"
            }]
        }, {
            "first_name": "\u62d2\u7edd\u670d\u52a1",
```

	"enable_num": "1",
	"num": "34",
	"first_id": "1",
	"sec_level_class": [{
	"sec_name":
	"HTTP\u62d2\u7edd\u670d\u52a1",
	"enable_num": "1",
	"num": "27",
	"sec_id": "54"
	}, {
	"sec_name":
	"\u7b2c\u4e09\u65b9\u5e94\u7528\u62d2\u7edd\u670d\u52a1",
	"enable_num": "0",
	"num": "7",
	"sec_id": "55"
	}]
	}, {
	"first_name": "\u4fe1\u606f\u6cc4\u9732",
	"enable_num": "0",
	"num": "26",

```
            "first_id": "5",
            "sec_level_class": [{
                "sec_name":
"\u670d\u52a1\u5668\u4fe1\u606f\u6cc4\u9732",
                "enable_num": "0",
                "num": "20",
                "sec_id": "51"
            }, {
                "sec_name": "\u6e90\u7801\u6cc4\u9732",
                "enable_num": "0",
                "num": "1",
                "sec_id": "53"
            }, {
                "sec_name":
"\u9519\u8bef\u4fe1\u606f\u6cc4\u9732",
                "enable_num": "0",
                "num": "5",
                "sec_id": "52"
            }]
        }]
}
```

	错误数据返回请参见**返回错误数据对应内容**

3.21.2 获取事件引擎三级

请求接口	http(s)://ip:port/v2/ detectSignPredef?action=show&level=second
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	"sec_id": "55", //二级事件id
	"set_name": "default", //站点名称
	"search_str": "", //事件名称
	"action": "", //响应动作 0：通过，1：丢弃，2：阻断，4：返回错误页面，6：返回重定向URL
	"sig_level": "" //事件级别 10：非攻击事件，20：低级事件，30：中级事件，40：高级事件
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据：

```
{
    "result": "0",
    "data": [{
        "sig_id": "152525648",
        "set_name": "",
        "sec_id": "55",
        "sig_name":
"HTTP_Nagios_Core_CGI_Process_cgivars_Off-By-
One_\u5b89\u5168\u6f0f\u6d1e[CVE-2013-7108]",
        "action": "1",
        "log_enable": "1",
        "enable": "0",
        "selected": "1",
        "log_level": "4",
        "acl_block_time": "0",
        "resp": "0",
        "raw_pkt_enable": "0",
        "raw_pkt_time": "60",
        "raw_pkt_size": "1024",
        "sig_level": "30",
        "event_type": "1",
```

```
            "url_white": "",
            "record_header_enable": "0"
        }, {
            "sig_id": "152452185",
            "set_name": "",
            "sec_id": "55",
            "sig_name":
"FTP_\u7578\u5f62\u6587\u4ef6\u540d\u62d2\u7edd\
u670d\u52a1\u653b\u51fb\u5c1d\u8bd5[CVE-2005-
0256]",
            "action": "1",
            "log_enable": "1",
            "enable": "0",
            "selected": "1",
            "log_level": "4",
            "acl_block_time": "0",
            "resp": "0",
            "raw_pkt_enable": "0",
            "raw_pkt_time": "0",
            "raw_pkt_size": "0",
            "sig_level": "30",
```

```
            "event_type": "1",
            "url_white": "",
            "record_header_enable": "0"
        }, {
            "sig_id": "152447560",
            "set_name": "",
            "sec_id": "55",
            "sig_name":
"FTP_WS_FTP\u670d\u52a1\u7a0b\u5e8f\u8fdc\u7a0b\
u62d2\u7edd\u670d\u52a1\u6f0f\u6d1e\u5229\u7528"
,
            "action": "1",
            "log_enable": "1",
            "enable": "0",
            "selected": "1",
            "log_level": "4",
            "acl_block_time": "0",
            "resp": "0",
            "raw_pkt_enable": "0",
            "raw_pkt_time": "0",
            "raw_pkt_size": "0",
```

```
            "sig_level": "30",
            "event_type": "1",
            "url_white": "",
            "record_header_enable": "0"
        }, {
            "sig_id": "152327315",
            "set_name": "",
            "sec_id": "55",
            "sig_name":
"SMB_\u62d2\u7edd\u670d\u52a1_Winnuke_\u653b\u
51fb[CVE-1999-0153]",
            "action": "1",
            "log_enable": "1",
            "enable": "0",
            "selected": "1",
            "log_level": "5",
            "acl_block_time": "0",
            "resp": "0",
            "raw_pkt_enable": "0",
            "raw_pkt_time": "0",
            "raw_pkt_size": "0",
```

```
            "sig_level": "20",
            "event_type": "1",
            "url_white": "",
            "record_header_enable": "0"
        }, {
            "sig_id": "152445619",
            "set_name": "",
            "sec_id": "55",
            "sig_name":
"FTP_Vsftpd_STAT\u62d2\u7edd\u670d\u52a1\u653b\u
51fb\u6f0f\u6d1e\u5229\u7528[CVE-2011-0762]",
            "action": "1",
            "log_enable": "1",
            "enable": "0",
            "selected": "1",
            "log_level": "4",
            "acl_block_time": "0",
            "resp": "0",
            "raw_pkt_enable": "0",
            "raw_pkt_time": "0",
            "raw_pkt_size": "0",
```

```
            "sig_level": "30",
            "event_type": "1",
            "url_white": "",
            "record_header_enable": "0"
        }, {
            "sig_id": "152452858",
            "set_name": "",
            "sec_id": "55",
            "sig_name": "FTP_Serv-
U_7.4.0.1_\u62d2\u7edd\u670d\u52a1\u6f0f\u6d1e\u5
229\u7528",
            "action": "0",
            "log_enable": "1",
            "enable": "0",
            "selected": "1",
            "log_level": "5",
            "acl_block_time": "0",
            "resp": "0",
            "raw_pkt_enable": "0",
            "raw_pkt_time": "0",
            "raw_pkt_size": "0",
```

```
            "sig_level": "20",
            "event_type": "1",
            "url_white": "",
            "record_header_enable": "0"
        }, {
            "sig_id": "152447494",
            "set_name": "",
            "sec_id": "55",
            "sig_name":
"FTP_ProFTPD_STAT\u62d2\u7edd\u670d\u52a1\u653b
\u51fb\u6f0f\u6d1e\u5229\u7528",
            "action": "1",
            "log_enable": "1",
            "enable": "0",
            "selected": "1",
            "log_level": "4",
            "acl_block_time": "0",
            "resp": "0",
            "raw_pkt_enable": "0",
            "raw_pkt_time": "0",
            "raw_pkt_size": "0",
```

	"sig_level": "30", "event_type": "1", "url_white": "", "record_header_enable": "0" }] }错误数据返回请参见**返回错误数据对应内容**

3.21.3 事件引擎保存

请求接口	http(s)://ip:port/v2/detectSignPredef?action=mod
请求方式	POST
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。 { "set_name": "default", //站点名称 "domainname": "", // "mode": 3, //固定3 "search_str": "", //事件名称 "action": "", //响应动作 0：通过，1：丢弃，2：阻断，4：返回错误页面，6：返回重定向URL "sig_level": "", //事件级别

	10：非攻击事件，20：低级事件，30：中级事件，40：高级事件
	"enable": 1, //固定1
	"first_class_id": [{ //拒绝服务(level_id = 1)、木马蠕虫(level_id = 2)、爬虫扫描(level_id = 3)、通用攻击(level_id = 4)、信息泄露(level_id = 5)、溢出攻击(level_id = 6)、注入攻击(level_id = 7)、其他类型(level_id = 8)；一级菜单下的子节点都选中level_id需要赋值
	"level_id": "1"
	}, {
	"level_id": "2"
	}, {
	"level_id": "3"
	}, {
	"level_id": "4"
	}, {
	"level_id": "6"
	}, {
	"level_id": "7"
	}],
	"sec_class_id": [{ //二级分类

	二级级菜单下的子节点都选中level_id需要赋值
	"level_id": "51"
	}, {
	"level_id": "53"
	}, {
	"level_id": "56"
	}, {
	"level_id": "59"
	}],
	"sig_evt_id": [{ //3级分类未全选时，填写选中的事件id
	"sig_id": "152526511"
	}, {
	"sig_id": "152521920"
	}, {
	"sig_id": "152521938"
	}],
	"unchange_list": [{ //只能时二级菜单，修改时未改动的二级菜单id
	"level_id": "58"
	}]

	}
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result" ：0} 错误数据返回请参见**返回错误数据对应内容**

3.22 配置导入导出和一键回退

3.22.1 配置导出

请求接口	http(s)://ip:port/v2/system_config?action=export
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： { "result" ：0,

	"data"： "c2Rmc2Rmc2Rmc2RmZHNmc2ZzZGZzZGZzZGZzZGY="　　//需要 base64 解码 } 错误数据返回请参见**返回错误数据对应内容**

3.22.2 配置导入

请求接口	http(s)://ip:port/v2/system_config?action=import
请求方式	POST

请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
	{
	'content':'',　　//配置文件流
	'type':'1'　　//是否重启设备
	}

返回码	200
返回格式	application/json;charset=utf-8
返回数据	正确数据： {"result"　: 0} 错误数据返回请参见**返回错误数据对应内容**

3.22.3 一键回退

请求接口	http(s)://ip:port/v2/onekeyRollback?action=show
请求方式	GET
请求参数	需携带ts、sign两个参数，具体见认证流程（这两个参数建议放到url里）。
返回码	200
返回格式	application/json;charset=utf-8

返回数据	正确数据：
	{
	"result" : 0,
	}
	错误数据返回请参见**返回错误数据对应内容**