浅谈信息安全主战场的转移

LinE@CloverSec Labs

1. • Whoami
2. •物联网的套路
3. •方法论
4. •举个栗子
5. •技能点/兵器库

我是谁

[LinE@CloverSec]:~# whoami
ID： LinE，Real name： YuanWei，Age：23 years old
[LinE@CloverSec]:~# groupinfo
LinE in the cloversec group
[LinE@CloverSec]:~# cat LinE_Info.txt

- 小伙伴们喜欢叫我懒妞 or 烂。。。T_T
- 打小就想知道那绿油油的电路板到底能做什么，对那神秘的世界心怀敬畏，然后不知道拆坏了多少玩意（学习成本嘛）
- 家里大大小小的电器基本上都遭到过我的摧残，从而锻炼了自己良好的维修能力（呃，专业修电脑）
- 从初三开始接触黑客，玩鸽子，搞入侵，探内网，看代码，写程序，到后来的搞电工，做网络，玩无线，改路由，做四轴，一发不可收拾，然后入了物联网的坑 233333
- 个人惯用 Debian，字符界面的那种，因为指法娴熟，手速惊人，遂荣幸的成为了，呃，网管（明明是OP好么！！！）
- 身高185，体重114，喜欢玩，喜欢结交各类好汉（不是.py的那种），24K纯直男不搞基谢谢
- 曾参加某次众测关于智能路由漏洞的挖掘，SSCTF线下出了一道NFC破解的题，嵌入式漏洞挖掘挑战赛挖掘到5个高危
- 一次机缘巧合结识了坤哥，有幸加入了四叶草，成为了CloverSec Lab的IoT方向负责人，欢迎各种技术非技术骚扰~

1. • Whoami
2. •物联网的套路
3. •方法论
4. •举个栗子
5. •技能点/兵器库

物联网的套路


智能设备中的问题
传统安全
注入
XSS
命令执行
.....
工控安全
震网病毒
APT
接口安全
.....
硬件安全
NFC
复制
重放
篡改
BadUSB
物理攻击
.....
远程管理
弱口令
客户信息泄露
接口安全
.....
协议安全
AirPlay
心脏出血
.....
物理安全
电器空转
供电过高
设备失控
.....

物联网的套路


智能设备开发中的问题
硬件本身的安全
调试触点/针脚
硬件改造
满载/高温
硬件中的系统安全
固件安全
后门
远程调试
硬件与服务器交互安全
接口安全
身份验证
封包截取
远程控制安全
后端系统
客户信息
越权
传统安全
注入
XSS
物理安全
物理层攻击
嗅探节点
无线
社会工程
物理链路通信
协议安全
无线电安全
中间人
信号重放

1. • Whoami
2. •物联网的套路
3. •方法论
4. •举个栗子
5. •技能点/兵器库

How To Do It

让我们聊聊方法论

如何从一无所知到开启上帝模式

黑盒阶段

1. 了解功能，使用范围，使用方法，以及能做什么
2. 拆机看PCB，从各种组件上了解其架构，寻找调试接口（TTL/JTAG）
3. 加电，进行常规性检测（扫端口，看服务等）
4. 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
5. 弄到固件，拆包分析，对其中的关键程序进行逆向
6. 重点关注Ping/Telnet等功能，尝试命令执行，进入白盒阶段
7. 自制添加了后门的固件，尝试刷入，进入白盒阶段
8. 其他脑洞大开的想法，做法

白盒阶段

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

1. • Whoami
2. •物联网的套路
3. •方法论
4. •举个栗子
5. •技能点/兵器库

搞硬件
不仅要能硬起来，也要能软下去

举个栗子

现在，有这么一个路由器，官方未给出固件，也没有在线更新，怎么搞到固件？

固件存放在flash中，要拿固件，搞他硬盘（Flash）

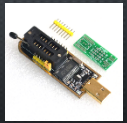
硬搞：拆机，取芯片，扔编程器读取，拿到整个flash的内容

软搞：拆机，连调试，从U-boot这里断下启动过程，利用U-Boot的查看内存功能搞定

所谓硬起来

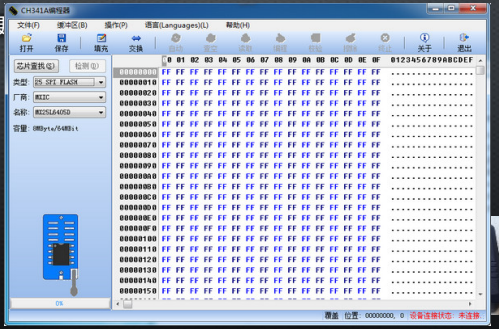
硬搞：拆机，取芯片，扔编程器读取，就像dd一样
拿到整个flash的内容

NOR Flash



Nand Flash

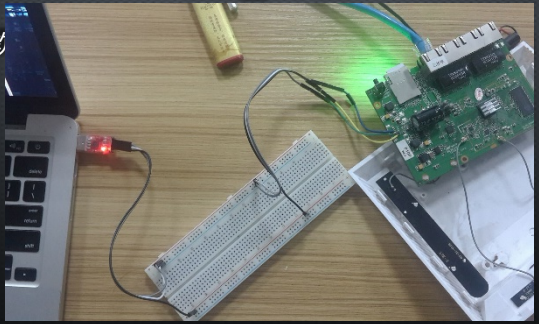
所谓硬起来


CH341A编程器
文件(F)
缓冲区(B)
操作(P)
语言(Languages)(L)
帮助(H)
打开
保存
填充
交换
关于
退出
芯片查找(Q)
检测(D)
类型:
25 SPI FLASH
厂商:
MXIC
名称:
MX25L4005D
容量:
0 0 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
00000000
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100
00000110
00000120
00000130
00000140
00000150
0%
覆盖
位置: 00000000, 0
设备连接状态: 未连接

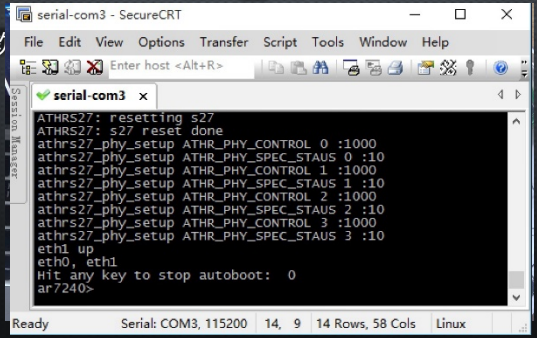
然后软下去

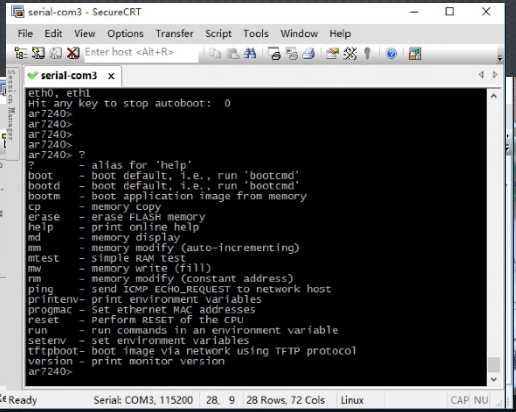
软搞：拆机，连调试，从U-boot这里断下启动过程
利用U-boot的查看内存功能搞定

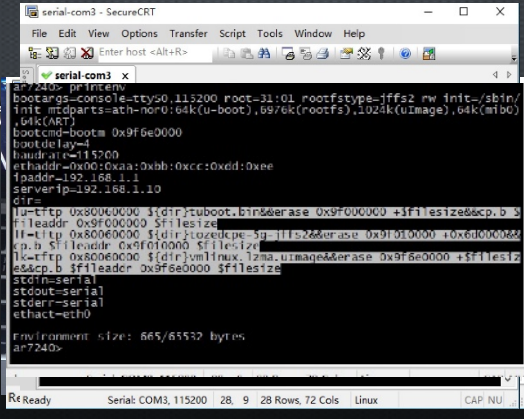
然后软下去



然后软下去

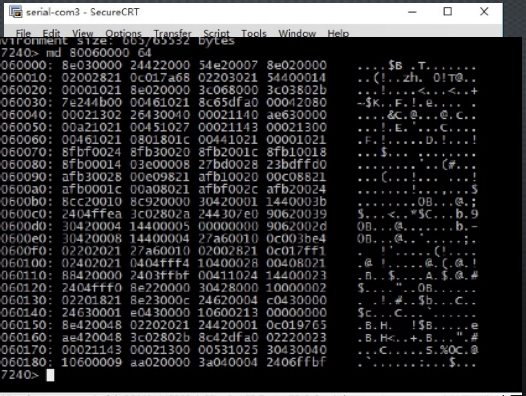

serial-com3 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Session Manager
serial-com3
ATHRS27: resetting s27
ATHRS27: s27 reset done
athrs27_phy_setup ATHR_PHY_CONTROL 0 :1000
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 0 :10
athrs27_phy_setup ATHR_PHY_CONTROL 1 :1000
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 1 :10
athrs27_phy_setup ATHR_PHY_CONTROL 2 :1000
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 2 :10
athrs27_phy_setup ATHR_PHY_CONTROL 3 :1000
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 3 :10
eth1 up
eth0, eth1
Hit any key to stop autoboot: 0
ar7240>
Ready
Serial: COM3, 115200
14, 9
14 Rows, 58 Cols
Linux


serial-com3 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3
eth0, eth1
Hit any key to stop autoboot: 0
ar7240>
ar7240>
ar7240>
ar7240>
ar7240> ?
? - alias for 'help'
boot - boot default, i.e., run 'bootcmd'
bootd - boot default, i.e., run 'bootcmd'
bootm - boot application image from memory
cp - memory copy
erase - erase FLASH memory
help - print online help
md - memory display
mm - memory modify (auto-incrementing)
mtest - simple RAM test
mw - memory write (fill)
nm - memory modify (constant address)
ping - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
progmac - Set ethernet MAC addresses
reset - Perform RESET of the CPU
run - run commands in an environment variable
setenv - set environment variables
tftpboot- boot image via network using TFTP protocol
version - print monitor version
ar7240>
Ready
Serial: COM3, 115200
28, 9
28 Rows, 72 Cols
Linux
CAP NU
Session Manager
Session Manager



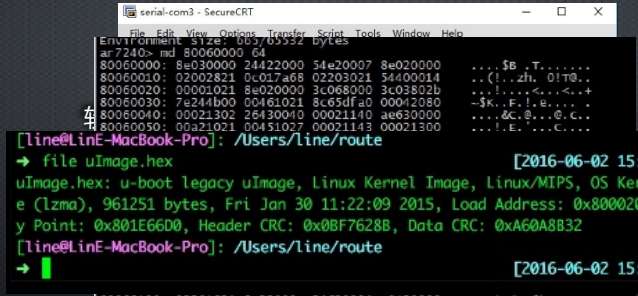
KCon West 2016

LinE@CloverSec Labs



KCon West 2016

LinE@CloverSec Labs



KCon West 2016

LinE@CloverSec Labs

再举个栗子

现在，有这么一个机器，通过无线连接到云端，接受云端的指令
然而机器自身没有发现问题，怎么知道机器与云端的通信内容？

通信必然要通过网络，要拿流量，直接抓

从硬件层面：使用Throwing Star LAN Tap等类似设备直接搭线攻击

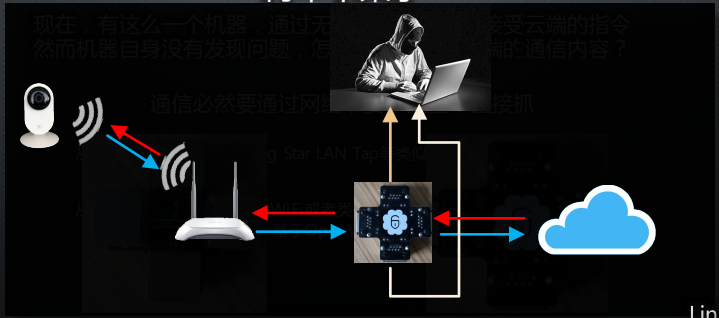
从软件层面：使用WooyunWiFi或者类似的拥有流量抓取功能的路由（基于OpenWRT）截取cap

再举个栗子

现在，有这么一个机器，通过无 接受云端的指令
然而机器自身没有发现问题，怎 端的通信内容？

通信必然要通过网络，_接抓

再举个栗子



再举个栗子

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？

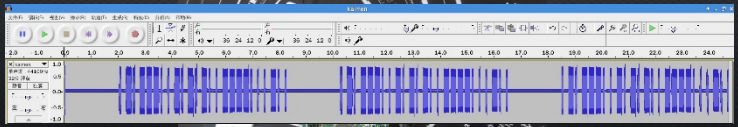
想要知道，直接抓~，用HackRF去捕捉信号

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

再举个栗子

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？

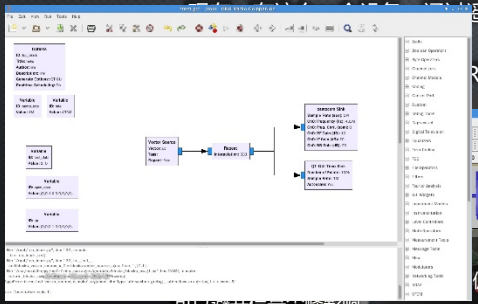
想要知道，直接抓~，用HackRF去捕捉信号

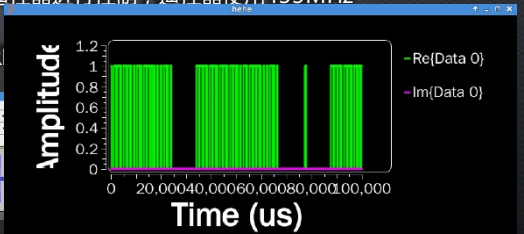


得到的原始信号长这样
这里有个小诀窍，在抓取的时候建议偏离中心信号一点，比如432.7MHz
可以避开信号尖峰影响

再举个栗子

遥控器进行控制，遥控器使用433MHz





偏离中心信号一点，比如432.7MHz

可以避开信号尖峰影响

1. • Whoami
2. •物联网的套路
3. •方法论
4. •举个栗子
5. •技能点/兵器库

技能加点

- Linux系统得会玩吧，这可是物联网的基石。
- 嵌入式/Arduinc开发板总要玩过吧
- 常见单片机（例如51，STM32）总得把玩一番
- 逆向工程好歹得会吧，不光是x86，还有ARM和MIPS架构的
- Android总得了解吧，调试应用啊，抓个包啊，逆个APK啊，ADB调试，RemoteADB
- 网络架构，通信方面总得入个门吧
- 常见的嵌入式解决方案得有个了解
  - 路由器解决方案（OpenWRT？DD-WRT?开源方案？自主研发？）
  - 摄像头解决方案（嵌入式Linux还是Android）
  - 手持终端解决方案（Android？Linux？自研？）
- 电子电路入个门，需要的时候焊个板子吧~
- 最后，对新鲜事物的接受能力、动手能力和学习能力！！！很重要！！！

我的兵器库
软件篇

虚拟机：VMware Workstation，Virtual Box，Parallels Desktop
终端软件：SecureCRT，SecureFX
逆向软件：OllyDbg，IDA
动态调试：QEMU虚拟机+逆向软件
固件分析：Binwalk， squashfs-tools， 7zip ，010editor，C32ASM
代码编辑：Sublime Text，NotePad++，Visual Studio Code
编程语言：Python，C，Java
流量分析：tcpdump，wireshark
无线射频：GNURadio，aircrack-ng，
Web调试：Burp Suite，Chrome开发者工具

我的兵器库
硬件篇

趁手的螺丝刀，我用的南旗的22in1，拆机片，撬棒，吹风机等等
趁手的电烙铁，我用的自己DIY的白菜白光，配刀头，尖头，马蹄头发热芯，焊锡，助焊剂
USB转TTL，USB转串口，USB转JTAG，USB转各类奇葩接口
Arduino学习套件，外围模块，面包板，洞洞板，杜邦线，各类电子相关的配件
HackRF One，UberTooth One，Proxmark3， Throwing Star LAN Tap，Teensy
刷了OpenWRT的路由器（WiFiPineApple），8187/3070网卡+各类天线，树莓派3
逻辑分析仪，芯片编程器等等
一双无所不能的手，一个停不下来的大脑，一颗永不言弃的心

Hacking For Fun And Enjoy
Just Do It

KCon West 2016

Designed By LinE @ CloverSec Labs

ThankS

Designed By LinE @ CloverSec Labs