智能IoT设备安全问题及研究方向思考

南京邮电大学网络空间安全学院

吴礼发　教授

（wulifa@njupt.edu.cn）

内容

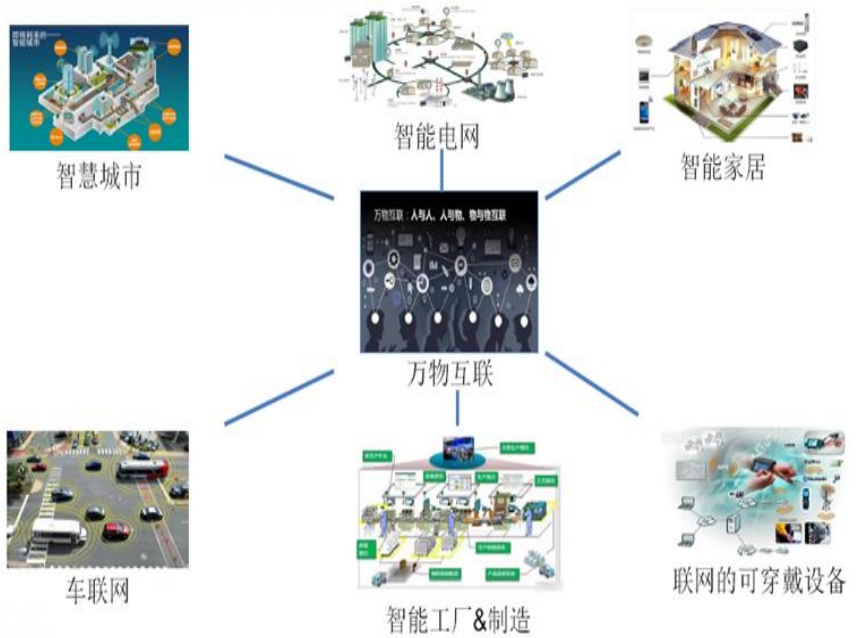
一、物联网（IoT）及其安全问题

二、智能IoT设备安全研究

三、有关研究方法的几点思考

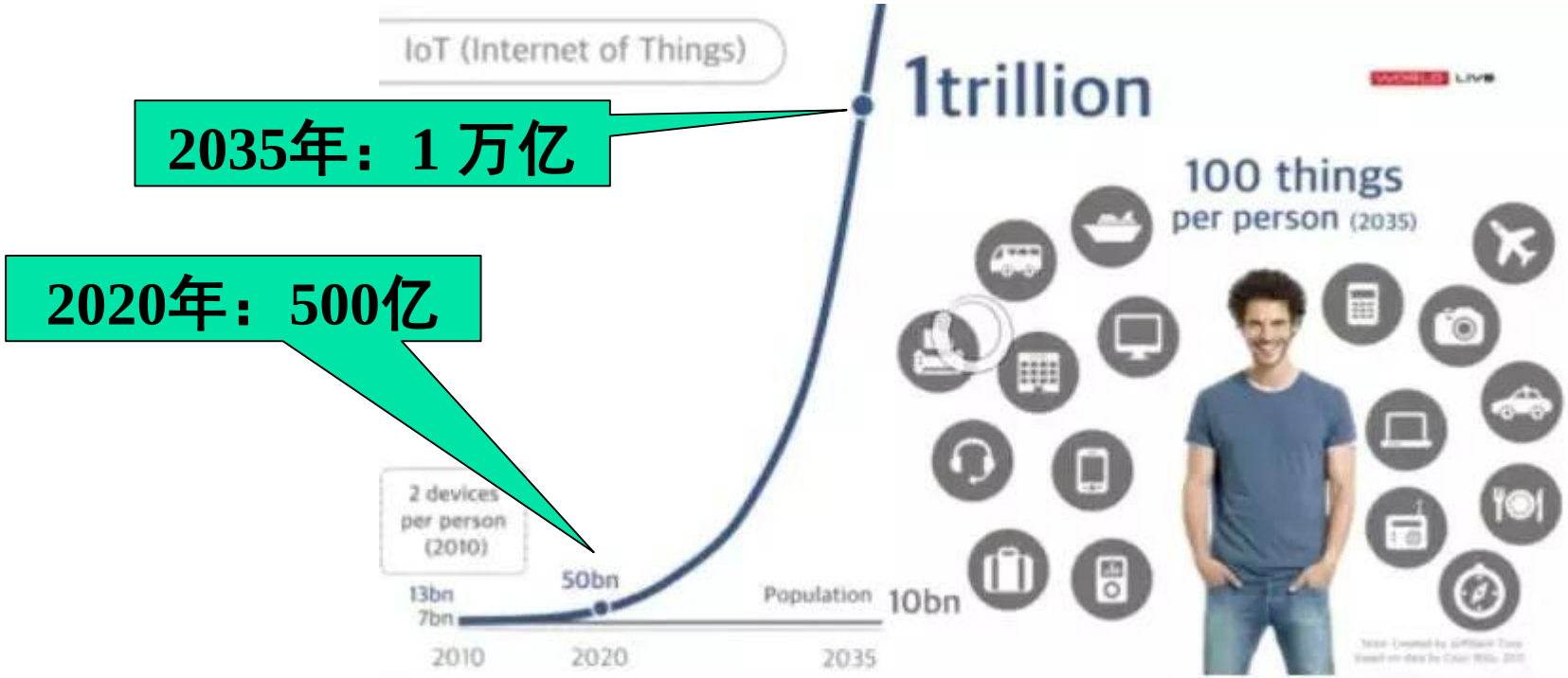
物联网（IoT）

- 物联网（Internet of Things）由ITU于2005年11月在《ITU Internet reports 2005 the Internet of Things》报告中正式提出：万物互联。

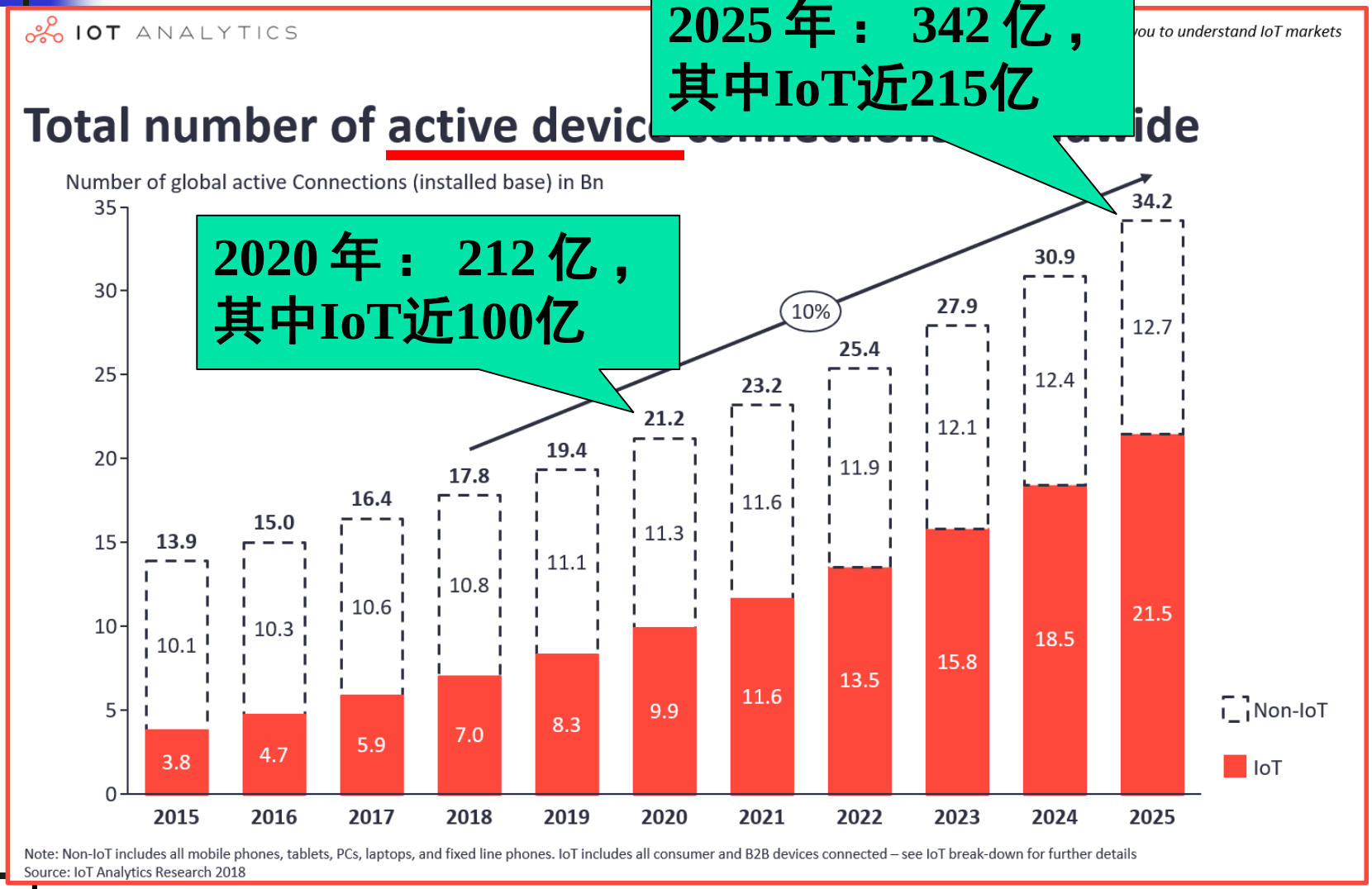


IoT设备数量爆发性增长

- 2018.11孙正义：未来30年的人工智能和物联网



IoT设备数量爆发性增长



IoT设备数量爆发性增长

- 2021.7：中国互联网发展报告（2021）

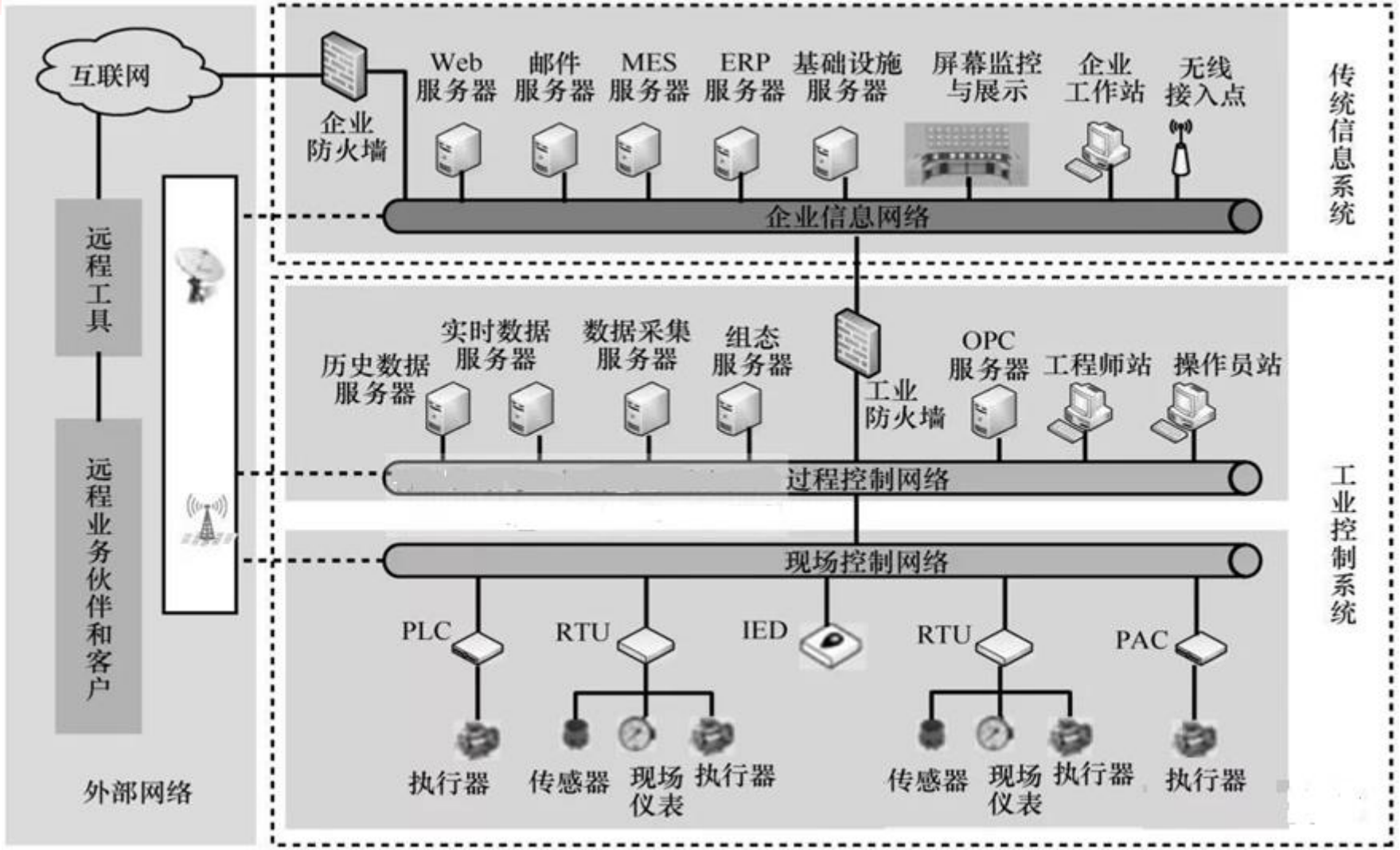
IoT设备数量爆发性增长

- 2021.7：中国互联网发展报告（2021）

物联网（IoT）

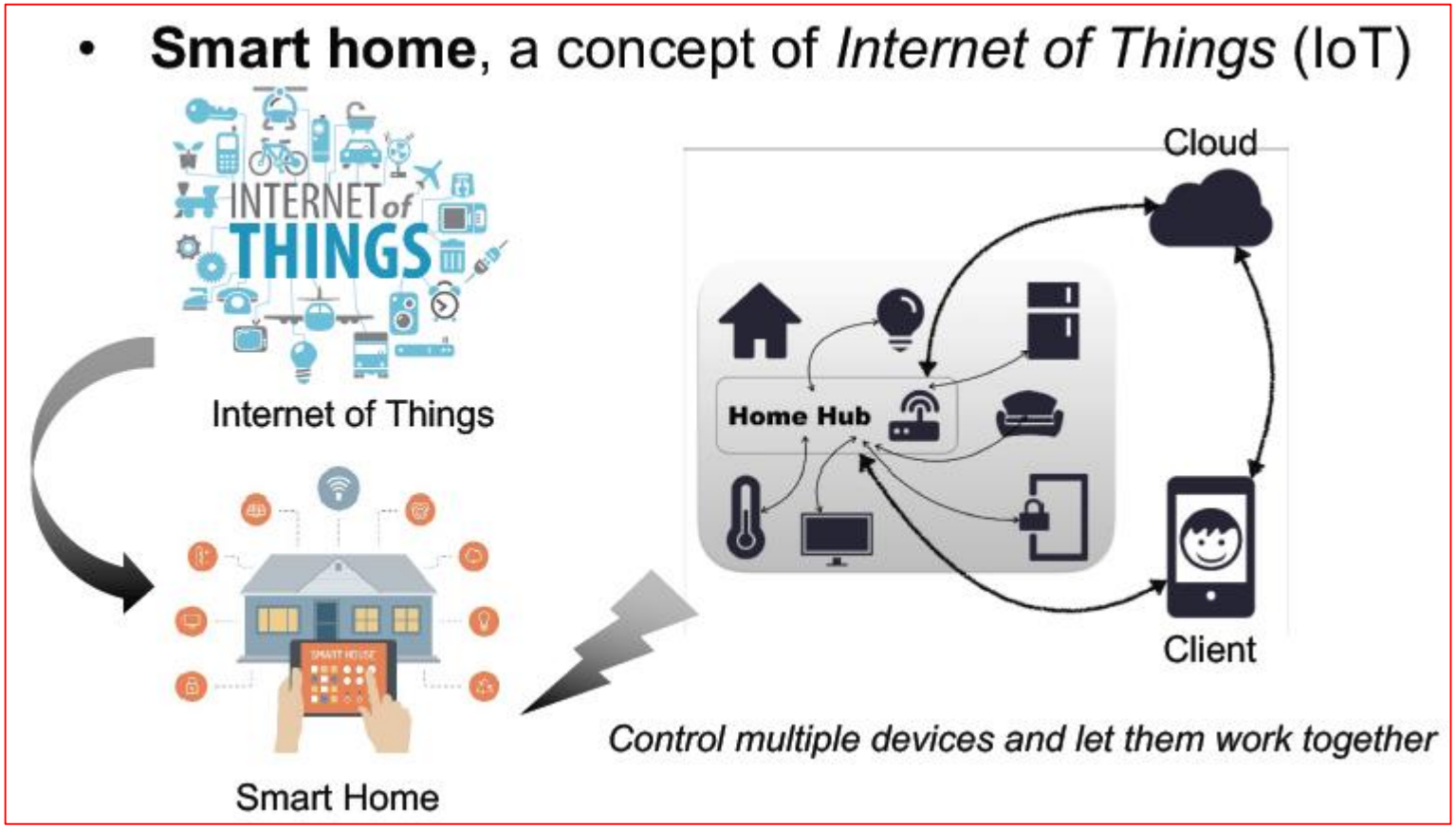
- 根据应用领域，物联网主要分为两大类：
  - 产业类IoT（工业控制、智能制造、智慧农业、供电供水等）
  - 消费类IoT（智能家居、智能监控、智慧医疗、车联网等）：增长最快！

产业类物联网


互联网
企业防火墙
Web服务器
邮件服务器
MES服务器
ERP服务器
基础设施服务器
屏幕监控与展示
企业工作站
无线接入点
传统信息系统
企业信息网络
远程工具
远程业务伙伴和客户
外部网络
实时数据服务器
历史数据服务器
数据采集服务器
组态服务器
工业防火墙
OPC服务器
工程师站
操作员站
过程控制网络
现场控制网络
PLC
RTU
IED
RTU
PAC
执行器
传感器
现场仪表
执行器
传感器
现场仪表
执行器
执行器
工业控制系统

消费类物联网

- 典型代表：智能家居

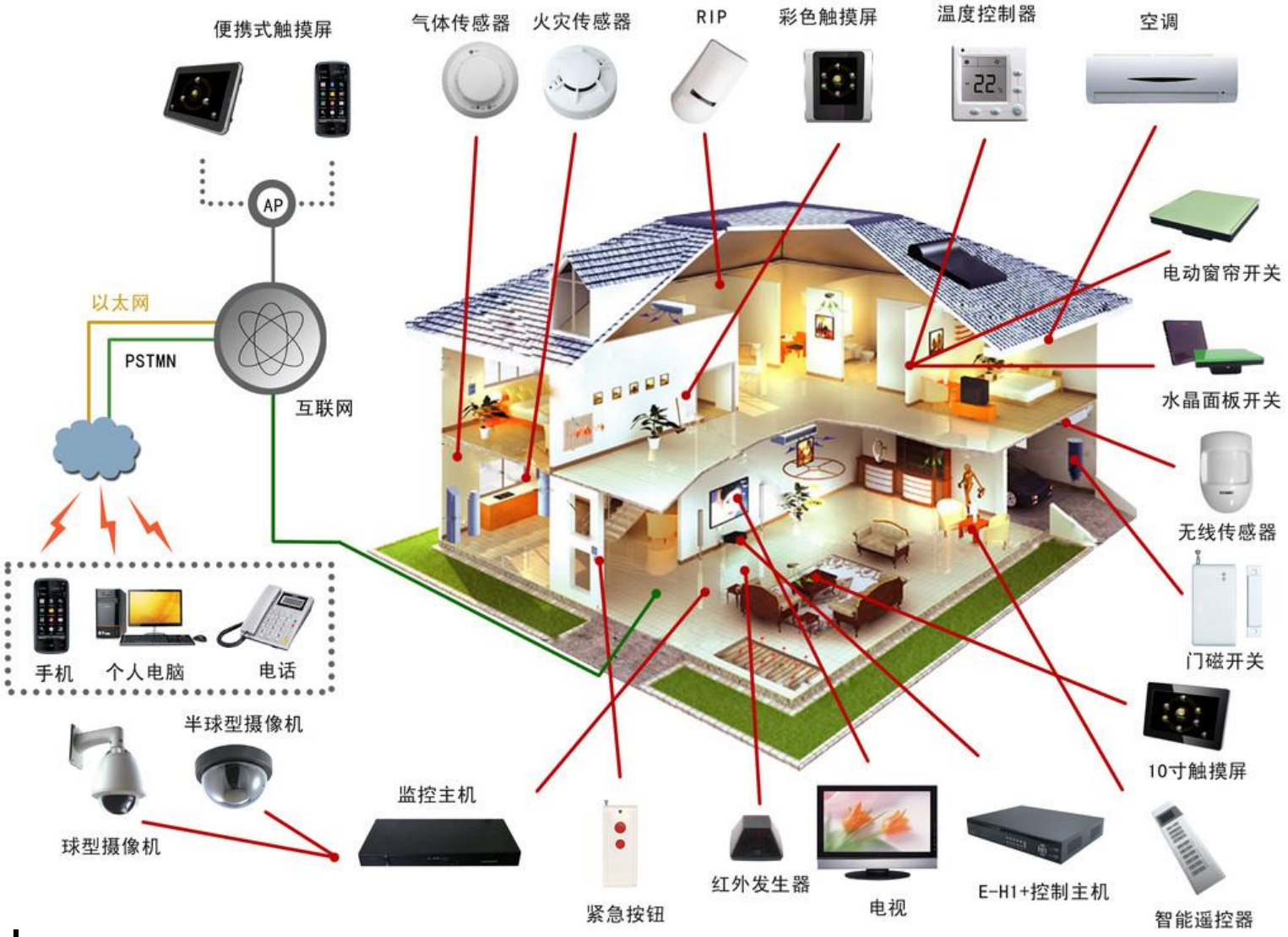


消费类物联网

- **2020.12：在“手机×AIoT”战略驱动下，小米智能物联网设备迅速发展，联网设备目前超2.89亿，拥有5件及以上IoT设备用户数达510万。**

来源：小米《消费级物联网安全基线》，2021.1.4

消费类物联网

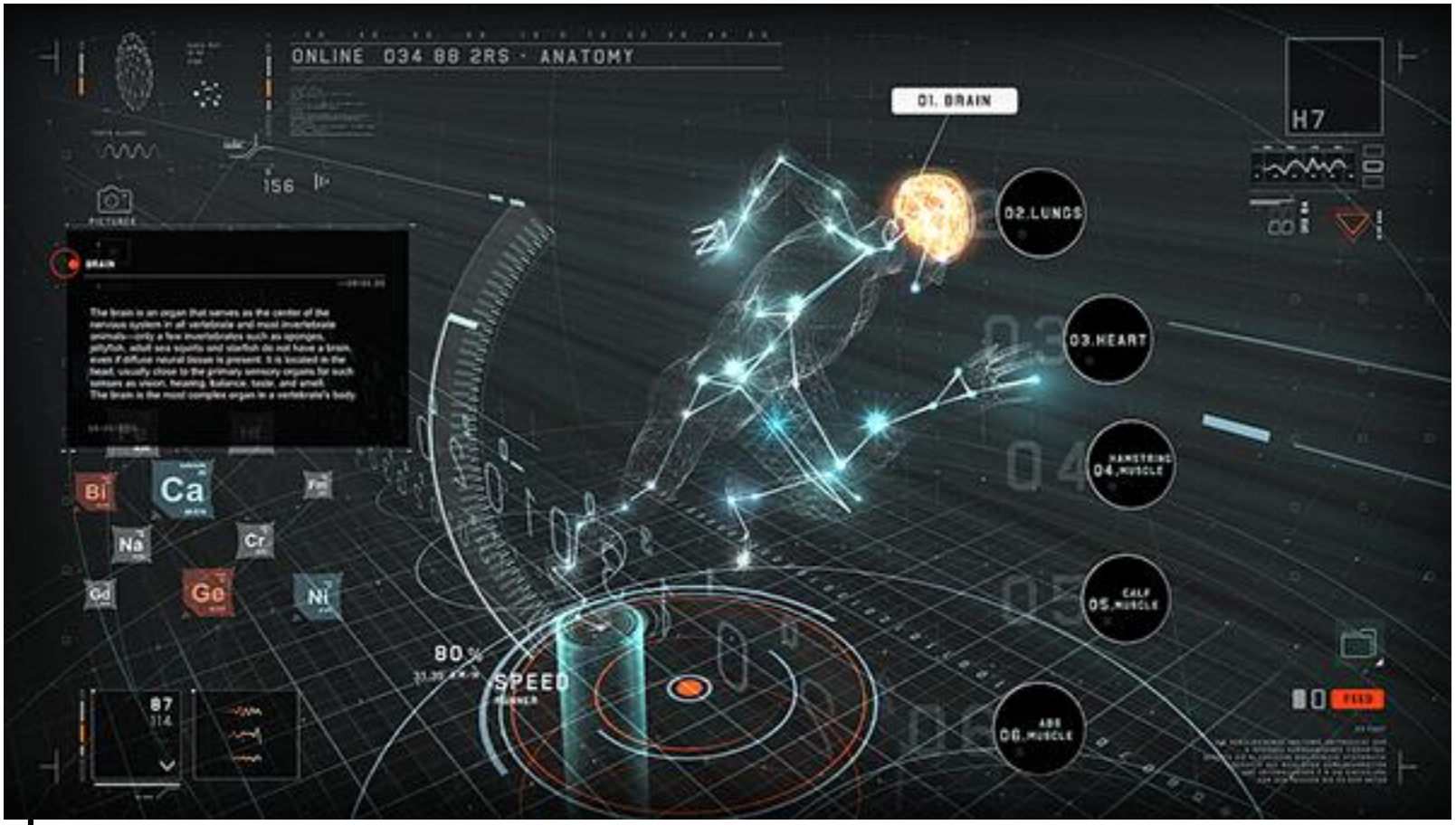

便携式触摸屏
气体传感器
火灾传感器
RIP
彩色触摸屏
温度控制器
空调
AP
以太网
PSTMN
互联网
电动窗帘开关
水晶面板开关
无线传感器
门磁开关
10寸触摸屏
手机
个人电脑
电话
半球型摄像机
球型摄像机
监控主机
紧急按钮
红外发生器
电视
E-H1+控制主机
智能遥控器

消费类物联网

- 典型代表：智能家居

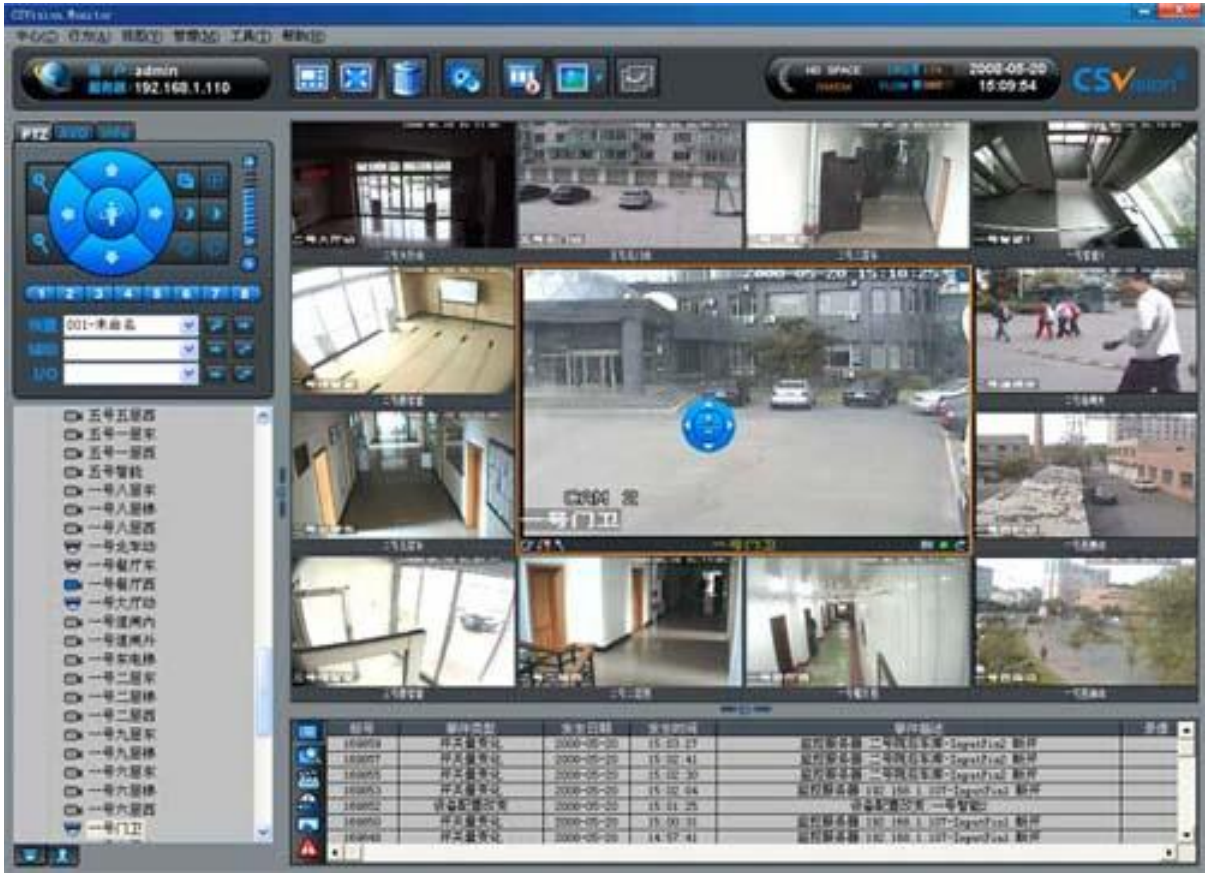
消费类物联网

- 典型代表：智慧医疗

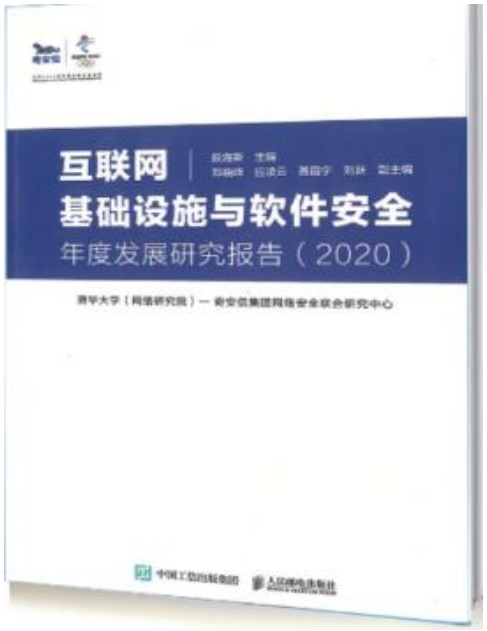


消费类物联网

- 典型代表：智能视频监控



IoT组成



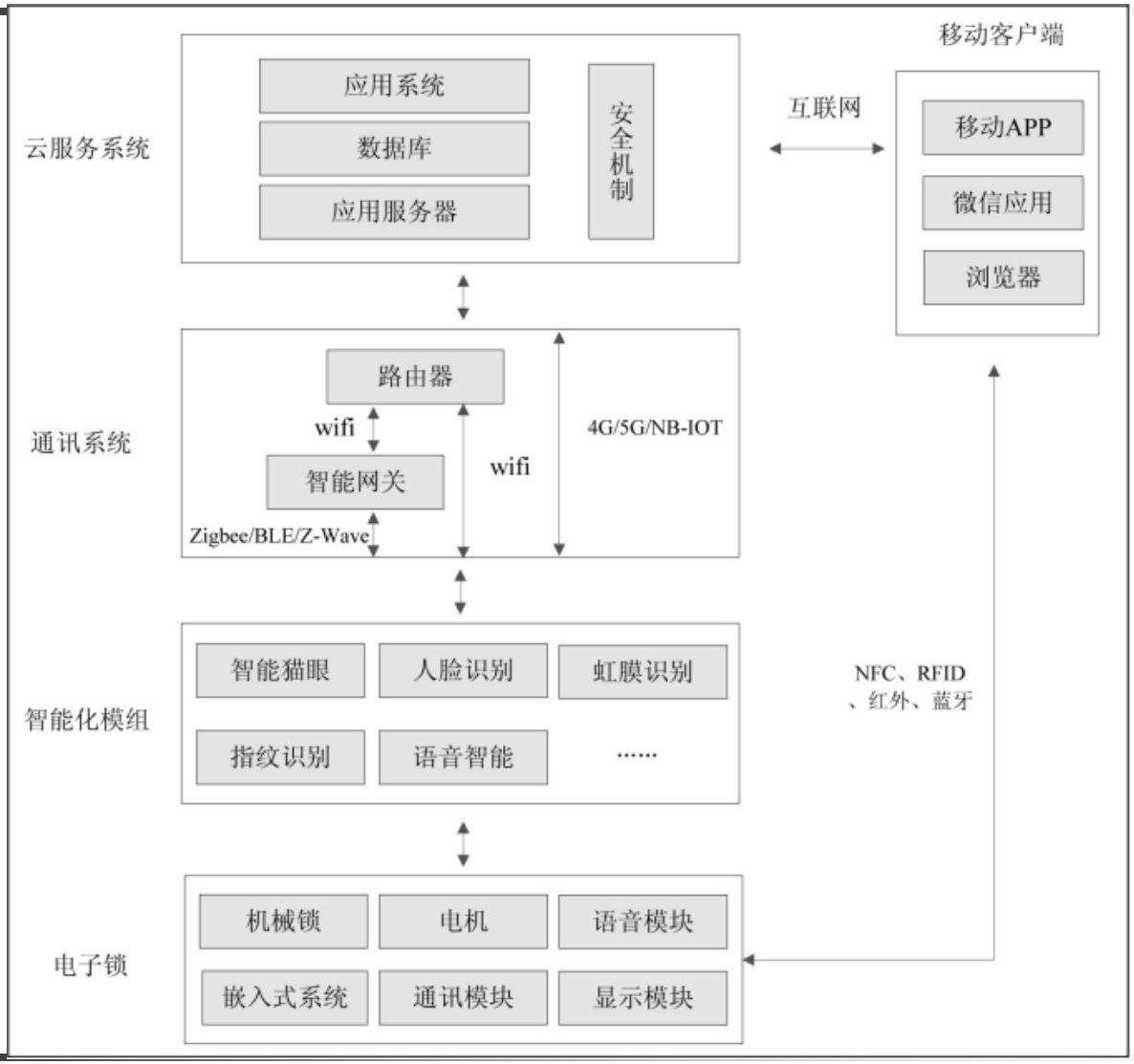
随着物联网技术的不断演进与发展，当前物联网常见基础架构（见图 2-7）大致包括“云、管、端、移”4 个角色。

- 云：通常指物联网云，物联网当前的计算资源部署模式正逐步向云上迁移，更多重要的业务功能由物联网云实现。
- 管：这里指网络接入方式，具体为物联网的各类通信协议及网络通信基础设施。物联网通信的大多数场景为机器和机器之间（M2M）的通信模式。
- 端：这里指物联网终端设备，如智能硬件等。但在不同的物联网场景下，物联网终端的表现各不相同。如在车联网中，终端除了 IVI（车载娱乐系统）、TBOX（车联网控制单元），还包括 CAN 总线连接的各类 ECU；在 LoRaWAN 及 NB-IoT 中，物联网终端通常由智能网关、各类传感器、执行器组成。
- 移：这里指移动端应用程序，通常指 App 程序，广义上指应用侧的所有系统，包括数据分析、行为监控、业务告警等。

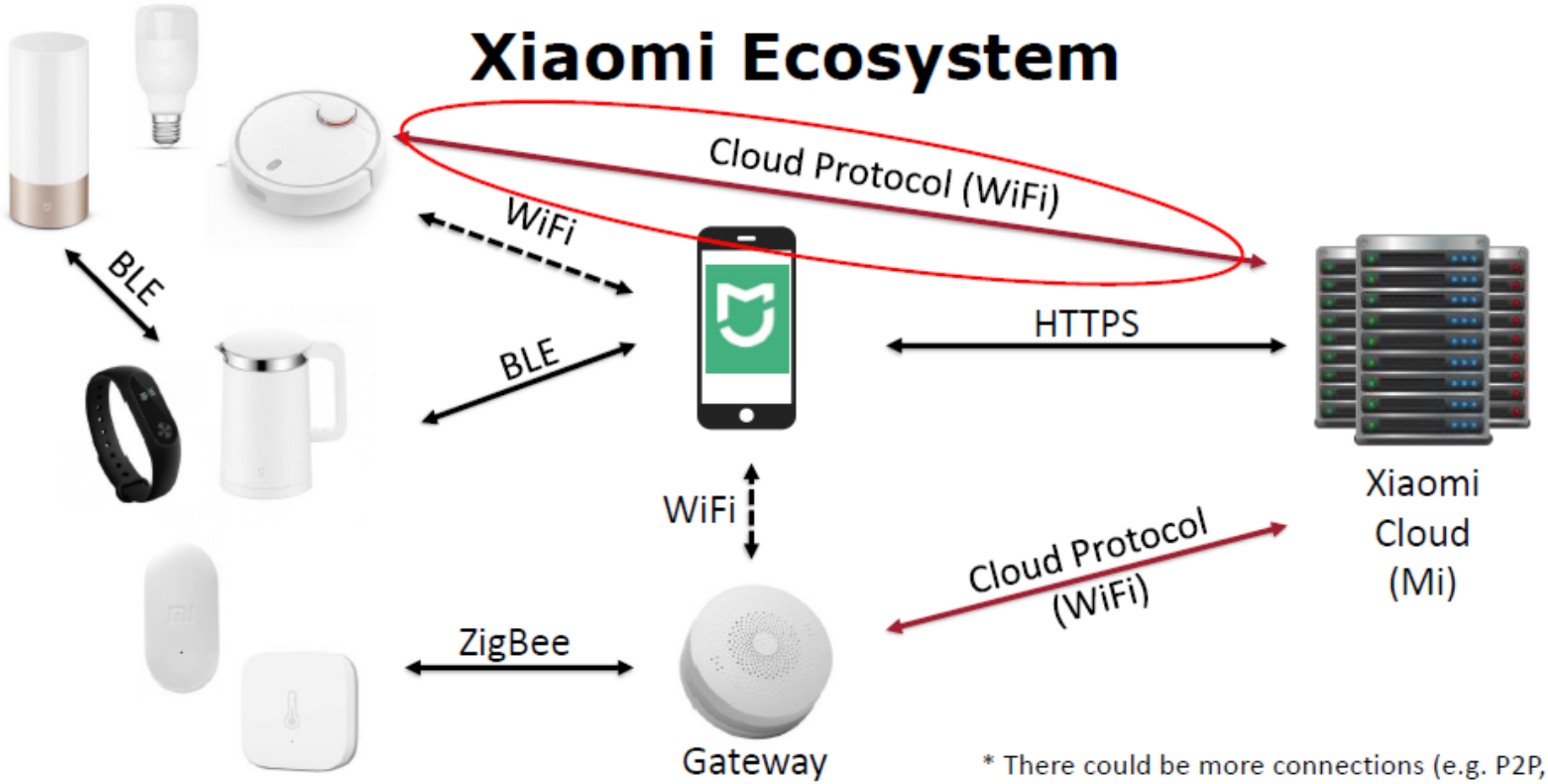
图 2-7　物联网常见架构

IoT组成

- 以


移动客户端
应用系统
数据库
应用服务器
安全机制
云服务系统
互联网
移动APP
微信应用
浏览器
路由器
wifi
智能网关
Zigbee/BLE/Z-Wave
wifi
4G/5G/NB-IOT
通讯系统
智能猫眼
人脸识别
虹膜识别
指纹识别
语音智能
.....
智能化模组
NFC、RFID
、红外、蓝牙
机械锁
电机
语音模块
嵌入式系统
通讯模块
显示模块
电子锁

IoT组成---通信接口


Xiaomi Ecosystem
Cloud Protocol (WiFi)
WiFi
BLE
BLE
HTTPS
WiFi
Xiaomi
Cloud
(Mi)
Cloud Protocol
(WiFi)
ZigBee
Gateway
* There could be more connections (e.g. P2P, FDS)

IoT组成 - 通信接口

- 智能设备的常见通信接口

物理接触使用	正常接触，如碰触智能设备的键盘，或其他输入模块等。
	破坏性接触，如用工具拆设备，接通电路板上的串口等。
通过手机使用	http 通信，如路由器 web 管理等。
	tcp 通信，如 telnet 协议，以及其他私有 tcp 通信协议等。
	udp 通信，如 upnp 协议中的 ssdp 协议部分等。
	其他更底层私有协议，如海康威视某协议（被 wireshark 识别为 0x8033）
	bluetooth 通信，如常见的 BLE 通信。
	NFC 通信，如常见的水卡，饭卡，公交卡等。
	红外通信，如万能遥控器等。
通过云端使用	http 通信，如某些共享单车。
	tcp 通信，如 mqtt 协议，以及其他私有 tcp 通信协议。
	udp 通信，常见于智能摄像头推送的视频流。
通过其他设备使用	wifi 组网通信，通信内容多采用 tcp/udp 方式。
	bluetooth 组网通信，如 Amazon Alexa 通过 BLE 控制其他智能设备。
	zigbee 组网通信，如小米智能家居套装。
	2G/4G 组网通信，通信内容多采用 tcp/udp 方式。
	其他 unlicense 频段组网通信，如某些智能门锁采用 433MHz 通信。
	其他组网通信方式，如 LoRa 组网、NB-IoT 组网等。

IoT安全问题

Hyppönen's law, and IoT safety and security

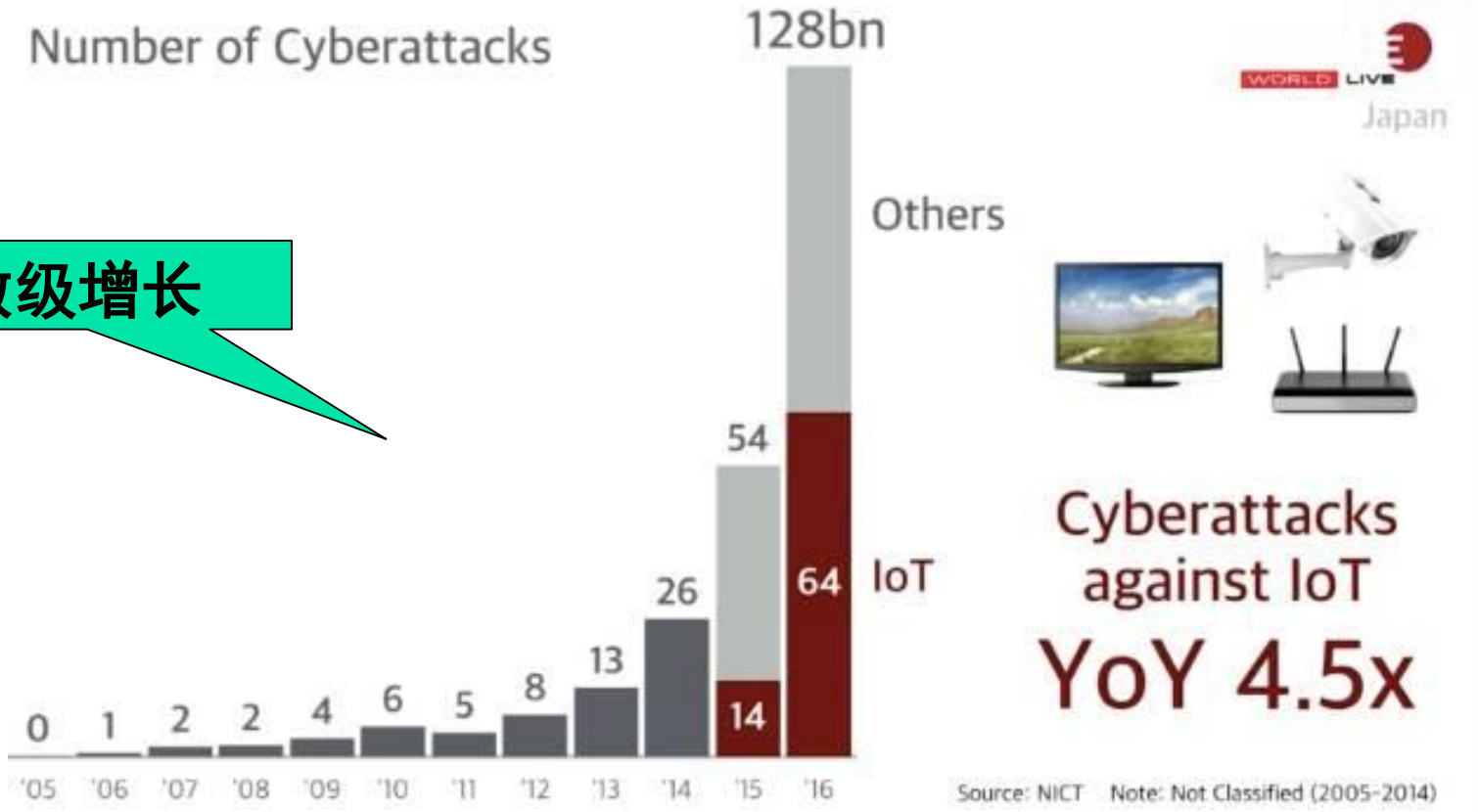
Security expert Mikko Hyppönen posited that ...

"Whenever an appliance is described as being "smart", it's vulnerable."

As everything that can be smart will be smart, and interact with each other, they will become targets of adversaries.

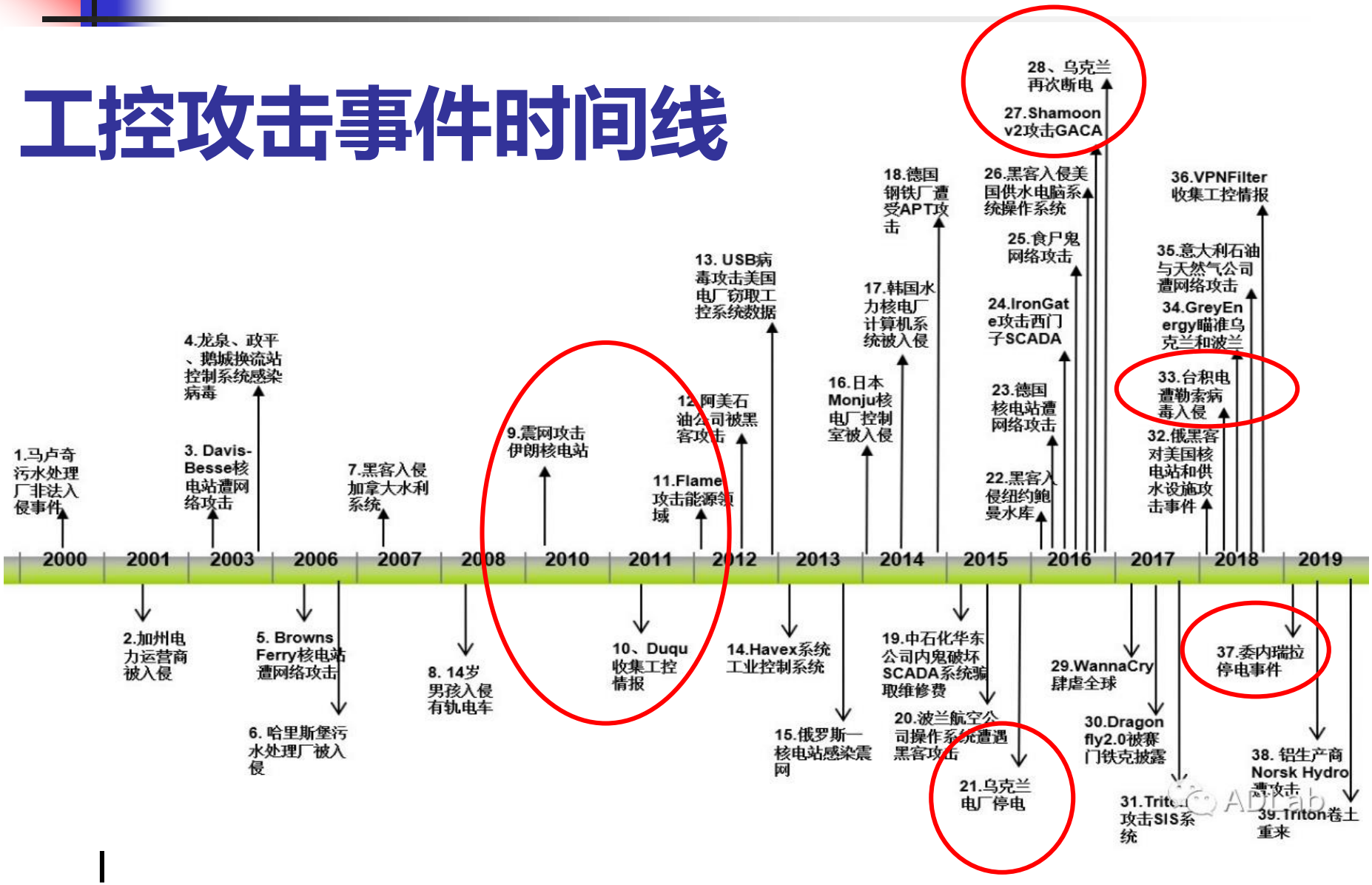
IoT攻击事件快速增长

- 2018.11孙正义：未来30年的人工智能和物联网



产业类IoT-安全事件

工控攻击事件时间线



产业类IoT-安全事件

2010 ,震网，伊朗核电站 ,8000台离心机损坏

2007

2017

乌克兰停电事件带来的未来网络战启示-搜狐

2016年3月11日 - 乌克兰停电事件的具体实现手段反而不太重要了:网络军力的持续提升已经成为各个国家的关注重点,但其难于溯源的属性又可令各国政府在攻击完成后推诿称自...

mt.sohu.com/20160311/n... - 百度快照 - 1522条评价

工业互联网APT

产业类IoT-安全事件

- 2018.8.3晚间，全球最大的代工芯片制造商台积电遭受勒索病毒入侵，引发生产线停摆，仅三天损失高达11.5亿元人民币

台積公司公布電腦病毒感染事件影響

發佈單位：台灣積體電路製造股份有限公司
發佈日期：2018/08/05

台灣積體電路製造股份有限公司今（5）日針對電腦病毒感染事件提供進一步說明，台積公司於8月3日傍晚受到電腦病毒感染，影響台灣廠區部分電腦系統及廠房機台，受病毒感染的程度因工廠而異，台積公司已經控制此病毒感染範圍，同時找到解決方案，至台灣時間下午兩點為止，約80%受影響的機台已經恢復正常，台積公司預計在8月6日前，所有受影響機台皆能夠恢復正常。

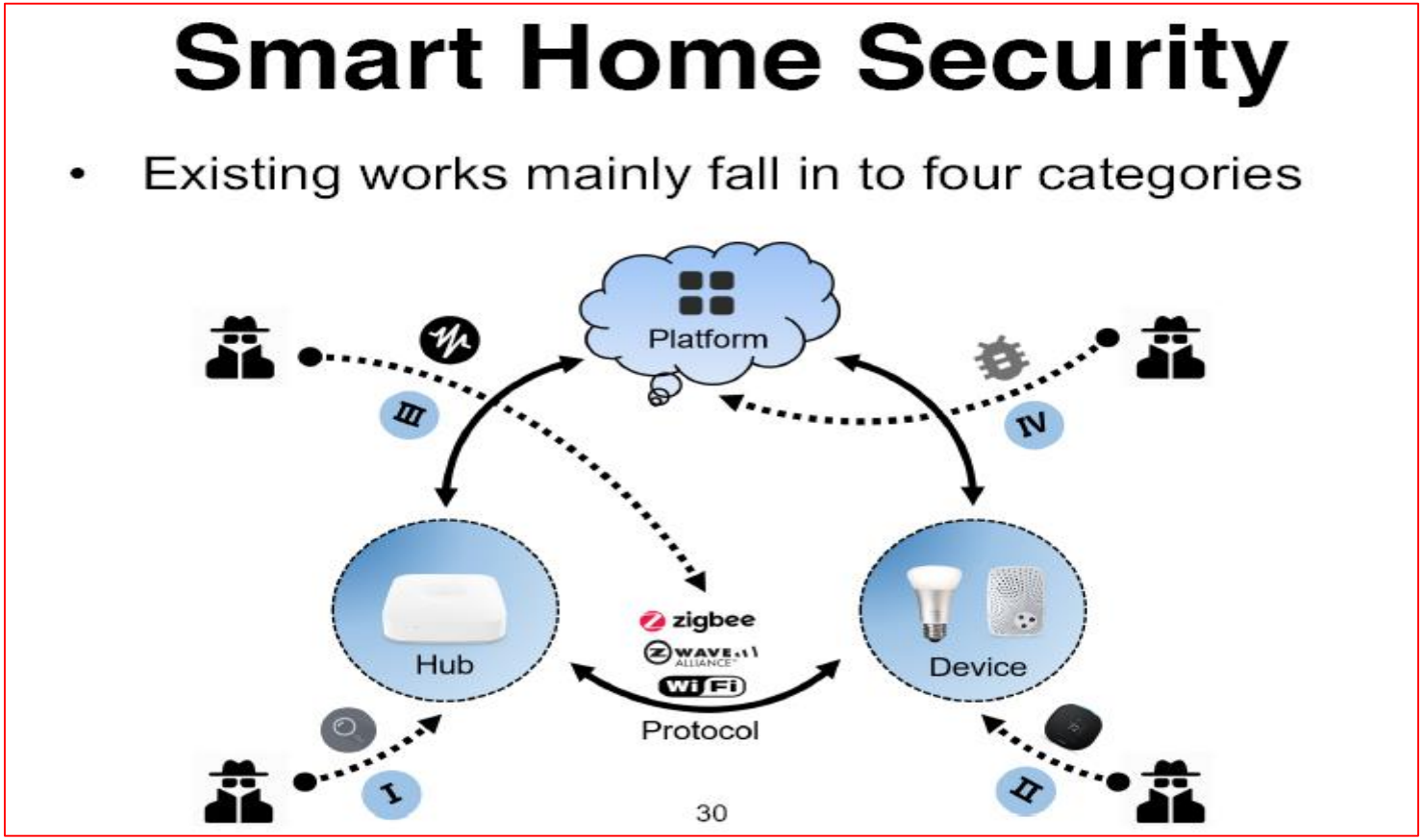
台積公司預估此次病毒感染事件將導致晶圓出貨延遲以及成本增加，對台積公司第三季的營收影響約為百分之三，毛利率的影響約為一個百分點。台積公司有信心第三季晶圓出貨延遲數量將於第四季補回，全年業績展望以美元計仍將維持7月19日所說的高個位數成長。

台積公司多數客戶皆已收到相關事件的通知，我們也正與客戶緊密合作，溝通其晶圓交貨時程，台積公司將在未來幾天內與個別客戶溝通細部資訊。

此次病毒感染的原因為新機台在安裝軟體的過程中操作失誤，因此病毒在新機台連接到公司內部電腦網路時發生病毒擴散的情況。惟台積公司資料的完整性和機密資訊皆未受到影響，台積公司已採取措施彌補此安全問題，同時將進一步加強資訊安全措施。

消费类IoT安全问题

- 安全威胁



消费类IoT安全-案例（Mirai）

- 2016年出现的IoT恶意代码Mirai：以IoT设备来目标（大多是数字录像机、监控摄像头和路由器）。Mirai僵尸网络攻击美国知名DNS服务提供商Dyn导致北美众多知名网站大断网

消费类IoT安全-案例（智能门铃）

Domestic IoT Nightmares: Smart Doorbells

智能门铃-物联网的噩梦

0431实验室 ChaMd5安全团队 2020-12-29

Dale Pavey

5G Security & Smart Environments, Public interest technology, Reverse Engineering, Tutorial/S

Vulnerability

December 18, 2020 17 Minutes

nccgroup

Victure：VD300
无品牌：高清 Wi-Fi 视频门铃 V5
无品牌：智能 WiFi 门铃 （YinXn）
奇　虎：360 D819智能视频门铃
Accfly：智能视频门铃 V5
无品牌：智能无线上网门铃 -XF-IP007H

应用程序通信未加密

在很多设备上，HTTPS 未强制执行，或者甚至作为通信方法存在于一系列移动应用程序上，例如 Victure 移动应用程序，这些应用程序被发现通过 HTTP 请求根证书。

服务未经身份验证

我们测试的一些设备显示，门铃和移动应用程序用于通信的后端服务访问控制的很差。特别是无品牌的门铃（HD Wi-Fi 视频门铃 V5）在处理特权 API 请求时，没有任何身份验证就可以修改设置。

消费类IoT安全-案例（智能门铃）

{"userid":1000396085,"m
[{"eventid":"030101","t
{"eventid":"020101","ti
{"selectdevicetype":"2"
{"eventid":"020201","ti
{"eventid
{"eventid
<redacted
{"eventid
<redacted
<redacted
west.mear
{"eventid
<redacted
<redacted>","resultCode
west.meari.com.cn\/app\
<redacted>","time":"202
<redacted>","time":"202
<-- Removed for bevity

Wi-Fi SSID 和密码被发送到位于美国的服务器，该服务器由中国的云提供商运营。从其他门铃的研究可以发现这种行为是不寻常的，因为其他移动应用程序不可以这样做。另外，还提出了更多的问题，为什么这些细节需要导出？这些数据可以很容易地与开源工具结合使用，以映射上传到这些服务器的 Wi-Fi 网络的物理位置，以创建活动设备的地图，以便使用 https://wiggle.net 等资源进行进一步分析。

域日志发送到： **https://app-logs.meari.com.cn**

我们使用 Frida 绕过证书防护获得后端执行数据或信息。在这些调查期间，可以看到移动应用程序定向和通信的域名，这是一个由**maerai.com.cn**重定向到 **https://www.mearitek.com/**。

Mearitek 是 Victure 的一家独立公司，似乎是 Victure 门铃利用的"CloudEdge"应用程序的创建者。Victure还 没有为自己的设备创建自己的应用程序，并且正在使用另一家公司的基础设施和软件来托管他们的设备，这也是不寻常的行为。

Wigle Wi-Fi OSINT website

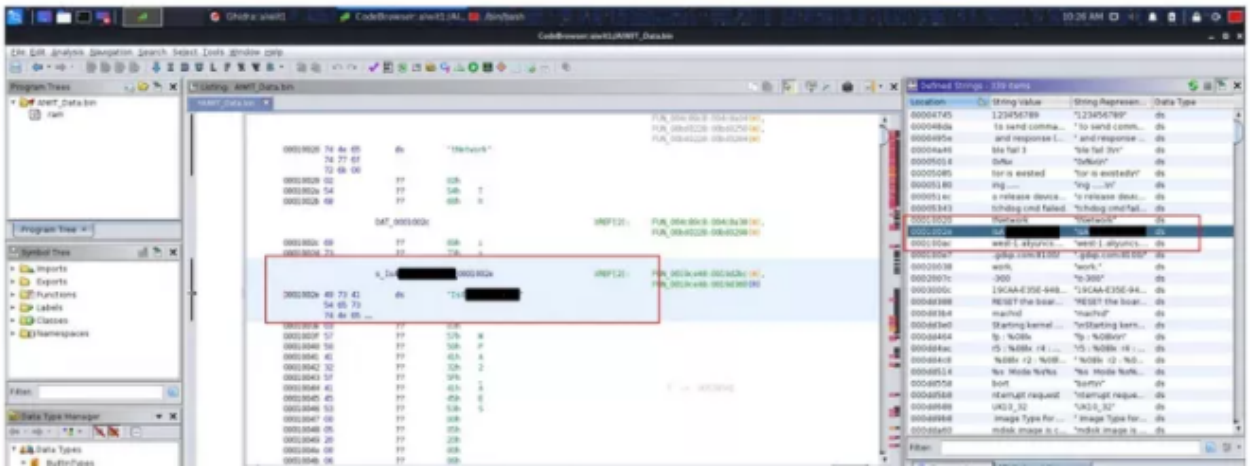
消费类IoT安全-案例（智能门铃）

不安全的固件更新过程

一旦设备暴露在公网，制造商可以部署软件更新到移动应用程序和设备固件。通常移动应用程序更新由相应的应用程序平台（Google Play、iOS 应用商店等）处理，但设备固件升级却很难处理。固件可能是 IoT 设备制造商可以开发的最敏感信息，因为这些信息控制硬件（摄像机、物理 Wi-Fi 模块等）需要保护。

在研究移动应用程序和远程固件更新功能时，发现调用特定 URL 也会下载更新固件所需的二进制文件。在调查的一些案例中，这些 URL 仅使用 HTTP 并且未经身份验证，允许任何了解该 URL 的人下载固件进行分析。

获得固件后，可以使用一系列二进制分析工具（Binwalk、Ghidra，甚至像字符串一样简单的 Linux 工具）进行分析，以分解固件结构，发现固件中包含的敏感信息，包括硬编码凭据、IP 地址和固件，以了解固件及其潜在的弱点。



消费类IoT安全-案例（智能门铃）

未记录的功能

我们发现许多未记录的功能对设备的运行也会造成伤害。例如，Victure 门铃能够通过登录到设备来管理和控制设备。此攻击中使用的凭据是从闪存获取的，闪存指出运行此固件的所有设备都使用相同的硬编码值。更严重的是，网上还可以查找到许多设备的登录界面。

在奇虎360设备上存在DNS服务，

在Victure门铃端口 80 上运行的 HTTP 服务可以通过用户网络访问

消费类IoT安全-案例（九安）

【漏洞预警】九安视频监控设备疑似存在“后门”等多个漏洞

liudao 922天前

近日，白帽汇安全研究院发现九安（闭路电视监控系统）视频监控设备的一个0day漏洞。利用该漏洞可以在未登录情况下查看实时监控截图。并且该设备存在一个后门，并被黑产非法利用，在网络上进行售卖。该后门可直接获得系统root权限，利用比较简单。危害等级较高。通过该设备可观看监控录像。目前发现在一些论坛上已经有了批量利用的自动化工具。该工具可批量检测漏洞，并下载执行远程服务器上的可执行文件。该设备还存在一个登录绕过漏洞，可以绕过用户名密码验证并查看监控视频。

安全公告

当前位置： 首页 >> 网络安全 >> 安全公告 >>正文

关于九安品牌视频监控系统漏洞的安全公告

2020-02-15 （点击： 900）

大工网安告[2020]006号

一、情况分析

九安品牌视频监控存在系统漏洞。攻击者可利用漏洞在未登录情况下查看实时监控截图，并可绕过用户名密码验证并查看监控视频。

二、处置建议

关于九安品牌视频监控系统漏洞的情况，广东网警已责令相关厂商使用在线固件升级等方式修复漏洞，用户也可自行到 http://help.dvr163.com/index.php/%E8%AE%BE%E5%A4%87%E5%9B%BA%E4%BB%B6 下载升级包，或致电 厂商电话 400-8752999 。

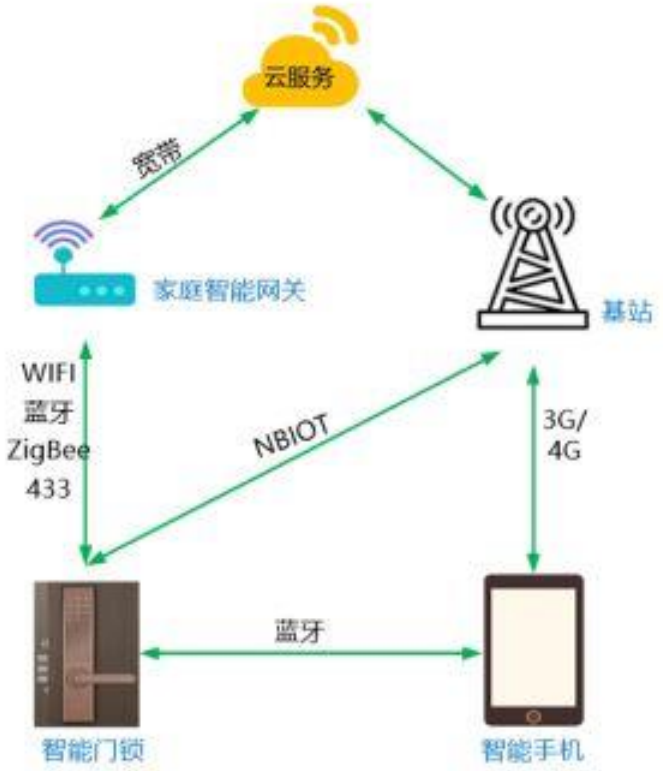
消费类IoT安全-案例（智能音箱）

- CNCERT：2019智能音箱隐私与网络安全分析报告

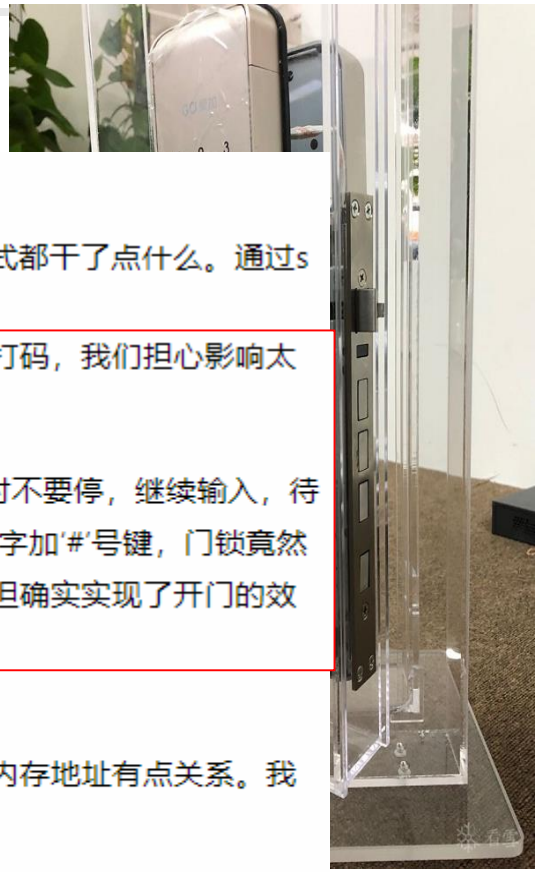
“无声”攻击

消费类IoT安全-案例（智能门锁）

- CNCERT：智能门锁网络安全分析报告（2018）



消费类IoT安全-案例（智能门锁）



看雪论坛 > 智能设备

5. 后门密码分析

在表4-1中，我们可以看到有一个“进入测试模式”的语音输出。顺其自然的，我们想看看测试模式都干了点什么。通过sub_800D40C(0x3)函数，我们可以定位到原固件中进入测试模式的代码，如下图所示：

上图中的内容，我们打了码，不过还是可以看出来，这些内容是智能门锁的键盘内容。如果不打码，我们担心影响太糟糕。

我们试着在门锁关闭的状态下输入这一串字符，当输入前两位时，门锁提示说“密码错误”，此时不要停，继续输入，待全部输入完毕之后，门锁果然提示说“进入测试模式”。然后，我们随便在门锁键盘上按了2个数字加‘#’号键，门锁竟然开启了。就这样，我们有了一个惊天发现：后门密码。看起来这并不是故意留下的后门密码，但确实实现了开门的效果。

图5-1 进入测试模式的相关代码

上图中，可以看到一个关键的判断是v84的返回值。观察图中的函数参数，应该是和0x80126EE内存地址有点关系。我们跳转到0x80126EE函数的位置，可以看到如下内容：

```
ROM:080126EE a1[redacted]      DCB "*#[redacted]",0
ROM:080126FC a[redacted]       DCB "[redacted]",0
ROM:08012700 a[redacted]       DCB "[redacted]",0
ROM:0801270B a[redacted]       DCB "[redacted]",0
ROM:08012712 a[redacted]       DCB "[redacted]",0
ROM:0801271C a[redacted]       DCB "[redacted]",0
```

图5-2 偏移为0x80126EE位置的内存内容

消费类IoT安全-案例（智能门锁）

- 某品牌锁：otas_init对OTA服务（利用该服务可以直接使用手机蓝牙对设备固件进行更新）进行初始化，其第3个参数为加密密钥，也就是说该智能门锁厂商将OTA升级的加密密钥硬编码在固件中！

```
int prf_init()
{
  int result; // r0

  qpps_init();
  if ( otas_init(0x12000u, 0, (unsigned int)"[redacted]") )
    assert_err("0", "..\\..\\src\\profiles\\prf_utils.c", 1508);
  result = 0;
  LOBYTE(word_1000DE9A) = 0;
  return result;
}
```

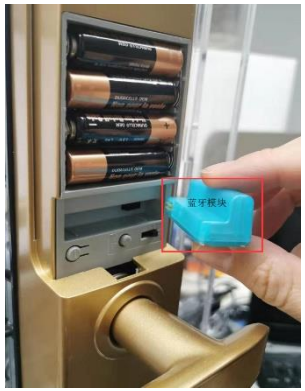
消费类IoT安全-案例（智能门锁）

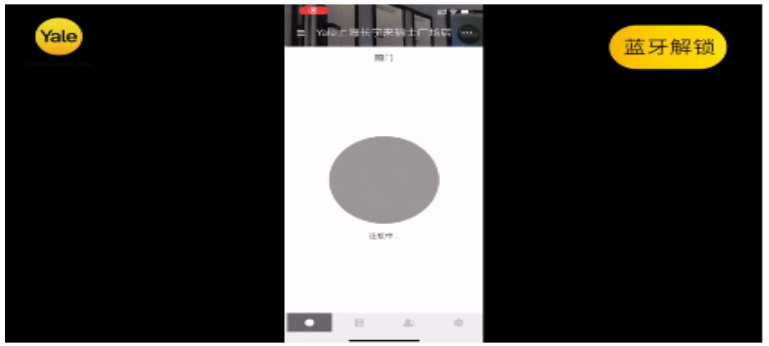
- 德国Yale智能门锁：门锁和手机的BLE通信未加密，嗅探productInfo，在非绑定手机上实现未授权开锁

【胖猴小玩闹】智能门锁与BLE设备安全Part 3： 耶鲁智能门锁的简单测试（上）

罗小黑　技术　2020-03-25 09:38:05　收藏

导语：本篇中，我们选择Yale品牌的一款智能门锁进行安全测试，一起来研究一下这款锁是否存在安全问题。





消费类IoT安全-案例（扫地机器人）

- SenSys'20

Spying with Your Robot Vacuum Cleaner: Eavesdropping via Lidar Sensors

Sriram Sami
National University of Singapore
srirams@comp.nus.edu.sg

Yimin Dai
National University of Singapore
e0505408@u.nus.edu

Sean Rui Xiang Tan
National University of Singapore
seantanr@comp.nus.edu.sg

Nirupam Roy
University of Maryland, College Park
nirupam@cs.umd.edu

Jun Han
National University of Singapore
junhan@comp.nus.edu.sg

ABSTRACT

Eavesdropping on private conversations is one of the most common yet detrimental threats to privacy. A number of recent works have explored side-channels on smart devices for recording sounds without permission. This paper presents *LidarPhone*, a novel acoustic side-channel attack through the lidar sensors equipped in popular commodity robot vacuum cleaners. The core idea is to repurpose the lidar to a laser-based microphone that can sense sounds from subtle vibrations induced on nearby objects. *LidarPhone* carefully processes and extracts traces of sound signals from inherently noisy laser reflections to capture privacy sensitive information (such as *speech* emitted by a victim's computer speaker as the victim is engaged in a teleconferencing meeting; or known music clips from television shows emitted by a victim's TV set, potentially leaking the victim's political orientation or viewing preferences). We implement *LidarPhone* on a Xiaomi Roborock vacuum cleaning robot and evaluate the feasibility of the attack through comprehensive real-world experiments. We use the prototype to collect both spoken

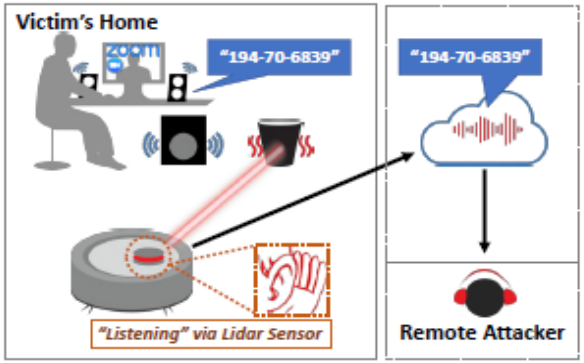


Figure 1: Figure depicts the *LidarPhone* attack, where the adversary remotely exploits the lidar sensor equipped on a victim's robot vacuum cleaner to capture parts of privacy sensitive conversation (e.g., credit card, bank account, and/or social security numbers) emitted through a computer speaker as the victim engages in a teleconference meeting.

消费类IoT安全-案例（智能电视）

- 智能电视越来越普及，功能也越来越多，不仅可以看电视，有的机型还配备了摄像头，实现视频通话和家庭监控等功能，但是随之而来的还有安全问题：监听、监视、肉鸡

IoT僵尸网络Ares利用开放ADB端口感染安卓机顶盒

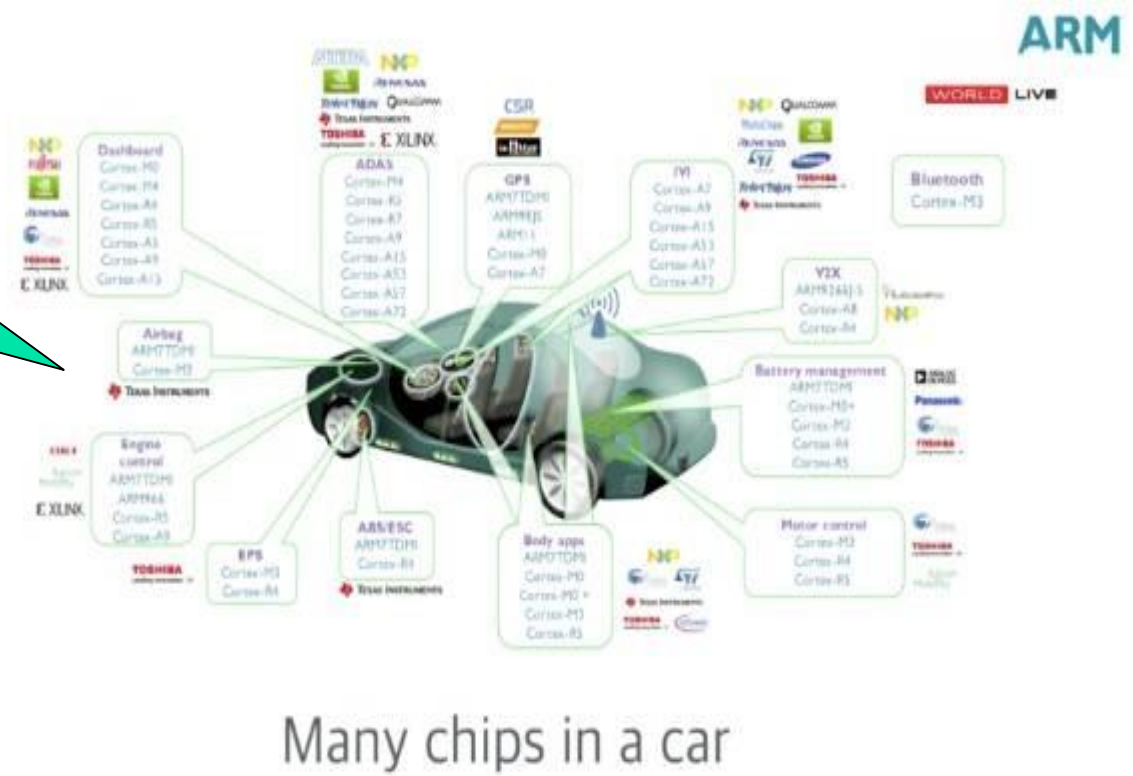
网络攻击 · 代码卫士 · 2019-08-29

摘要：安全公司WootCloud指出，该僵尸网络最常见的受害者是由 HiSilicon、Cubetek 和 QezyMedia 公司制造的安卓机顶盒。

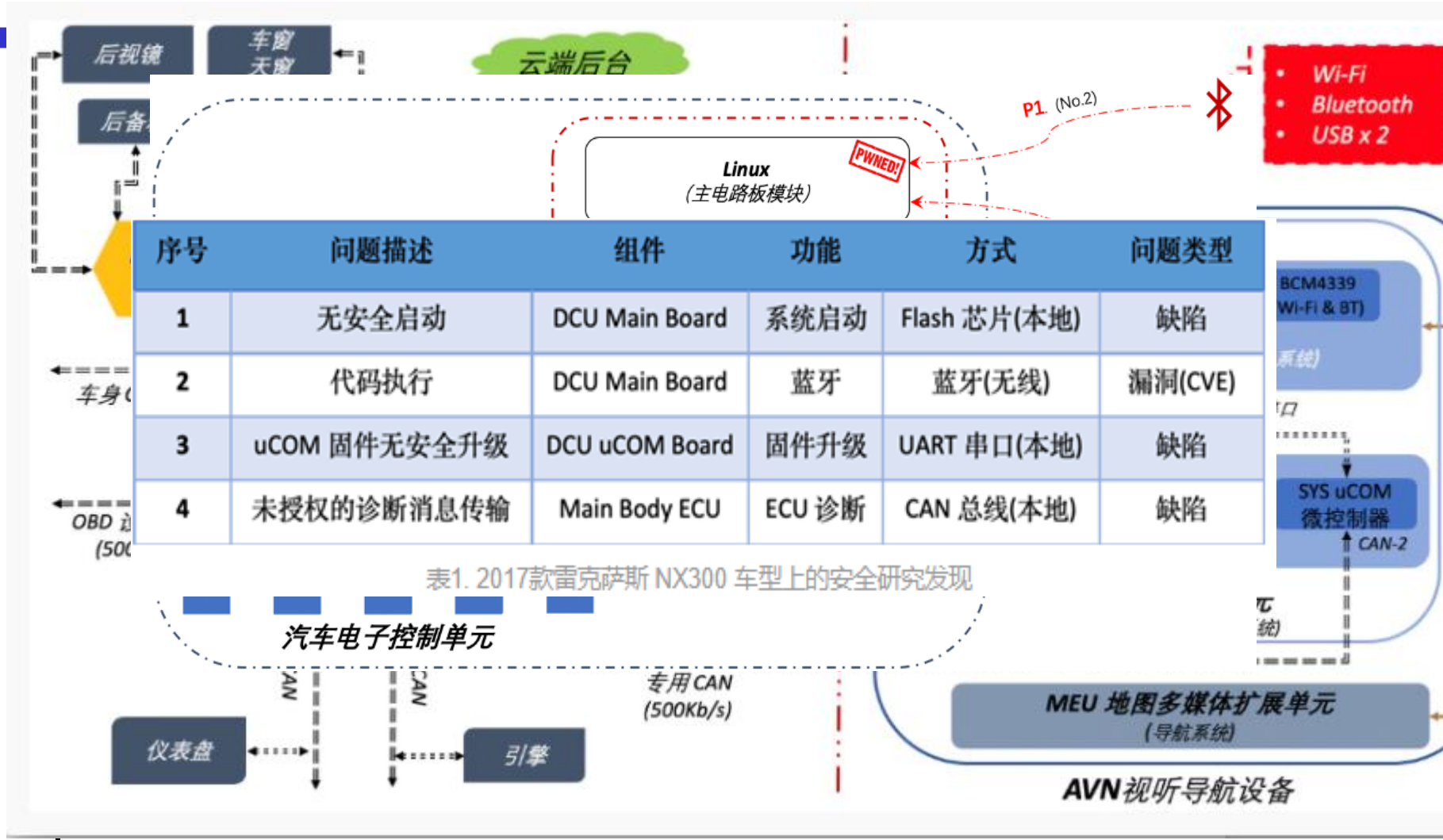
消费类IoT安全-案例（智能汽车）

- 2018.11孙正义：未来30年的人工智能和物联网

2018：一辆汽车约有500 块ARM芯片



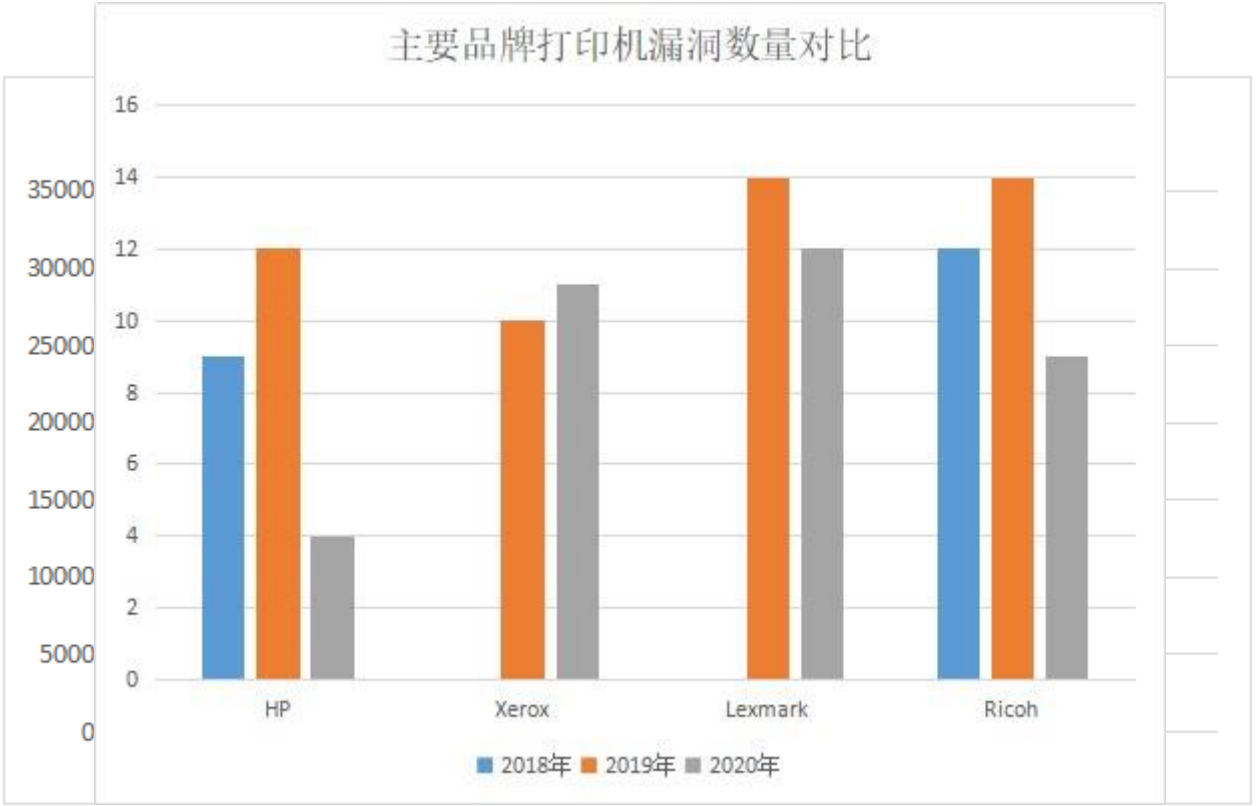
消费类IoT安全-案例（智能汽车）



序号	问题描述	组件	功能	方式	问题类型
1	无安全启动	DCU Main Board	系统启动	Flash 芯片(本地)	缺陷
2	代码执行	DCU Main Board	蓝牙	蓝牙(无线)	漏洞(CVE)
3	uCOM 固件无安全升级	DCU uCOM Board	固件升级	UART 串口(本地)	缺陷
4	未授权的诊断消息传输	Main Body ECU	ECU 诊断	CAN 总线(本地)	缺陷

表1. 2017款雷克萨斯 NX300 车型上的安全研究发现

消费类IoT安全-案例（打印机）



天融信云服务威胁感知平台（2020.08）

消费类IoT安全问题

消费级物联网安全基线
小米AIoT安全实验室

《消费级物联网安全基线》正式发布

小米安全中心 1月4日

13个安全控制域

77项基线要求点

- 硬件安全
- 通信安全
- 以太网安全
- 蓝牙安全
- Zigbee安全
- 射频安全
- 系统安全
- Linux嵌入式
- 应用安全
- 编码安全
- 逻辑安全
- 数据安全

消费类IoT安全问题

消费级物联网安全基线

小 米 A I o T 安 全 实 验 室

目 录

Contents

前言 4

范围 5

规范性引用文件 5

第一章 硬件安全

1.1 物理调试接口 7
1.2 本地数据存储 7
1.3 通信链路数据传输 8
1.4 安全启动 8
1.5 启动异常 8
1.6 MCU IAP 更新机制 9
1.7 防强电磁攻击 9
1.8 防物理拆除 9
1.9 智能门锁开锁 9

第二章 通用通信安全

2.1 特权功能接口 12
2.2 密钥硬编码 12
2.3 通信信道加密 12
2.4 通信鉴权 12
2.5 防重放 13

第三章 以太网通信安全

3.1 数据传输加密 15
3.2 HTTPS 证书校验 15
3.3 设备服务端口 16
3.4 WiFi 接入点口令 16

第四章 低功耗蓝牙 (BLE) 通信安全

4.1 蓝牙 SoC SDK 版本
4.2 蓝牙配对
4.3 控制指令合法性校验
4.4 传感器设备蓝牙广播
4.5 蓝牙协议版本
4.6 蓝牙控制指令鉴权
4.7 蓝牙广播防追踪机制
4.8 蓝牙敏感信息通信

第五章 设备 Zigbee 通信安全

5.1 默认 TCLK
5.2 防重放
5.3 Rejoin 功能

第六章 设备射频通信安全

6.1 防重放
6.2 射频通信数据包序列号
6.3 通信秘钥硬编码
6.4 通信频率

第七章 设备通用系统安全

7.1 固件升级包完整性与合法性
7.2 固件降级
7.3 高风险网络服务
7.4 OTA 升级指令

第八章 嵌入式设备 Linux 系统安全

8.1 地址空间布局随机化
8.2 Bootloader 启动
8.3 串行端口
8.4 系统默认用户密码
8.5 基础文件系统权限
8.6 外部存储的程序和脚本

第九章 Android 应用安全

9.1 Socket 端口请求 32
9.2 私有目录权限 32
9.3 本地信息存储 32
9.4 外部可执行文件 32
9.5 解压文件 33
9.6 XML 配置 33
9.7 防逆向工程 33
9.8 应用完整性 34
9.9 安装包签名 34
9.10 安卓组件 34
9.11 防屏幕录像 37
9.12 内存数据保护 37

第十章 通用编码安全

10.1 第三方软件 / 库 39
10.2 随机数生成函数 39
10.3 字符串或内存操作函数 39
10.4 格式化字符串函数参数 40
10.5 代码库管理 40

第十一章 Linux 应用编码安全

11.1 栈 Cookie 防溢出 42
11.2 栈不可执行保护 42
11.3 基址随机加载保护 42
11.4 Linux 程序代码编译 42
11.5 系统调用函数参数 43

第十二章 业务逻辑安全

12.1 设备可绑定状态 45
12.2 绑定确认 45
12.3 防重复绑定 45
12.4 强绑定关系 45
12.5 恢复出厂设置 46
12.6 Wi-Fi 接入点用途 46
12.7 地数据存储 46

第十三章 数据安全

13.1 加密算法
13.2 哈希算法
13.3 多重密钥
13.4 日志上报
13.5 跨境网络请求

术语和定义 50

缩略语 55

附录 56

参考资料 58

《消费级物联网安全基线》正式发布

小米安全中心 1月4日

内容

一、物联网（IoT）及其安全问题

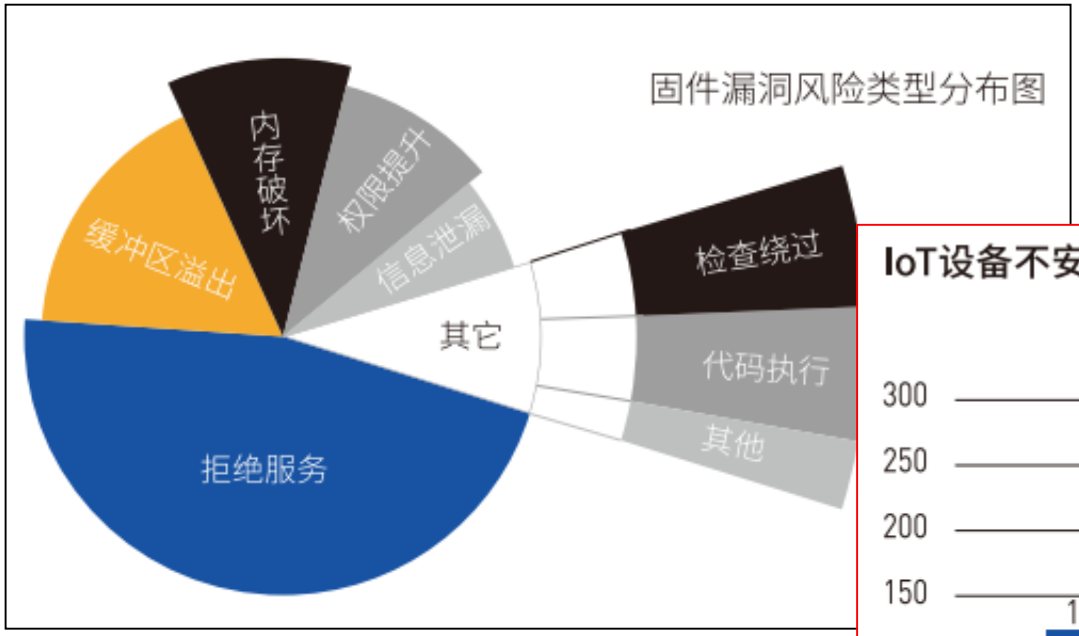
二、智能IoT设备安全研究

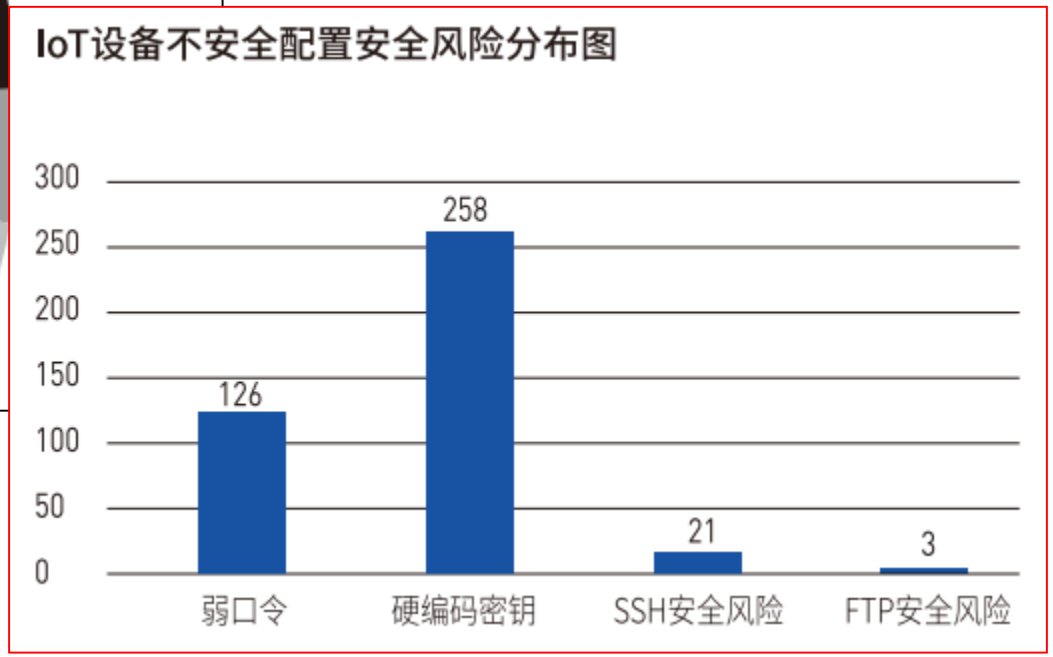
三、有关研究方法的几点思考

IoT组成

IoT设备安全问题

- 腾讯科恩实验室《2018年IoT安全白皮书》





一、固件安全

IoT 固件

- 固件就是运行在设备端的程序和数据，一方面固件和底层MCU等硬件直接打交道；另一方面固件还要完成智能设备的各种逻辑功能。

IoT 固件

- 固件大致可以分为 3 种：
  - 使用完整操作系统（大多基于Linux/BSD）和文件系统的固件
  - 使用功能较弱的实时操作系统（RTOS，如VxWorks、FreeRTOS等）且可能没有文件系统的固件
  - 没有操作系统和文件系统的固件

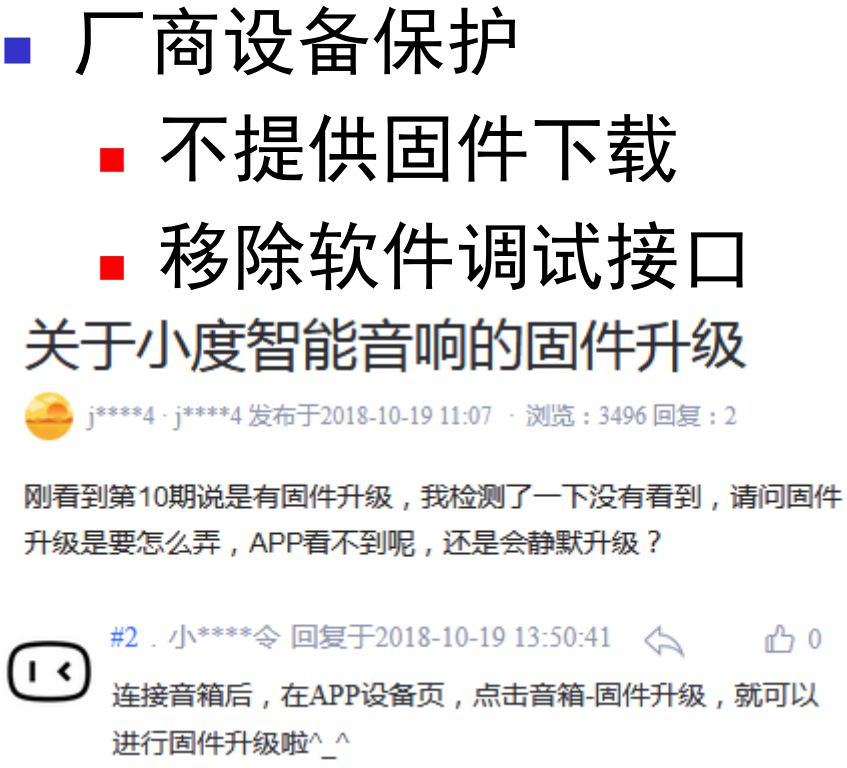
IoT固件平台

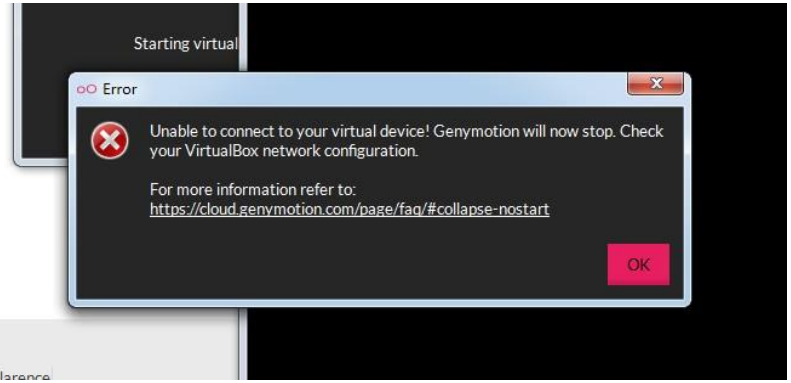
操作系统及硬件	RTOS实时、精简内核的Linux，ESP乐鑫、Arduion片上系统、AVR、STM32系列，SPI Flash存储	Openwrt、精简内核的Linux，ARM、Mips处理器，NAND/SPI Flash存储	完整的Android、Linux发行版，ARMv5/6/7/x86处理器EMMC/EMCP/NAND存储存储
应用场景	智能门锁、智能电饭煲、智能插座、智能灯、智能手环	路由器、mini版智能音箱、智能摄像头	智能音箱、智能手表、自动售货机、电视盒子、智能电视、智能广告牌、车机
特点	功能单一简单但大多有通过网络进行简单控制，模拟电路无法实现	单一但高级功能、无需屏幕展示内容or小尺寸屏幕	较多功能、较大的存储、易于开发APP的载体、大多有大屏幕

固件安全分析难点

- IOT设备多样化
  - 不仅仅是路由器
  - 单片机和嵌入式

- 厂商设备保护
  - 不提供固件下载
  - 移除软件调试接口





（一）固件逆向分析

固件逆向分析-固件提取

- 固件提取

MIDC·2020
小米AIoT安全峰会

打开IoT设备分析的第一扇门

胡一米、郭韬 未来安全胖猴实验室安全研究员

固件逆向分析-固件提取

- 硬件工具

精密螺丝刀

陶瓷螺丝刀

Dennis Giese（DEFCON 26上做报告，主搞小米）**的工作间**

固件逆向分析-固件提取

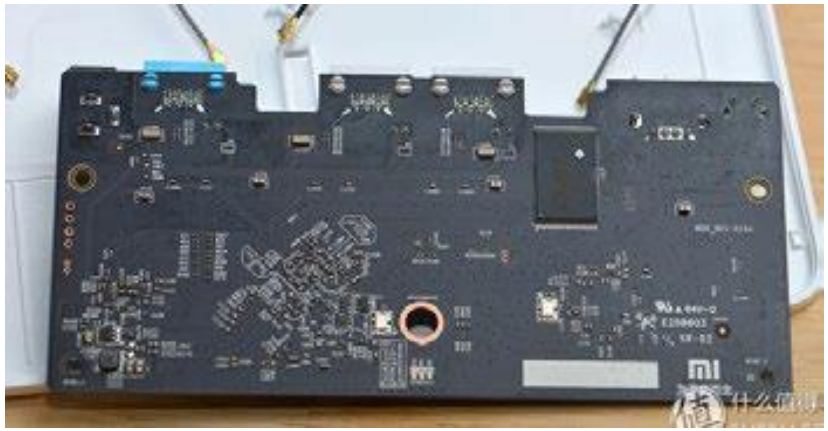
- 硬件工具

Georgia Institute of Technology的Omar Alrawi**的工作间**

固件逆向分析-固件提取

- 固件存储位置：固件一般会保存在2个位置：外置的Flash存储器，MCU内置Flash存储器
- 集中式存储(MCU)
  - AVR、PIC
  - ARM M系列
- 分离式存储(MPU)
  - MIPS
  - ARM A系列

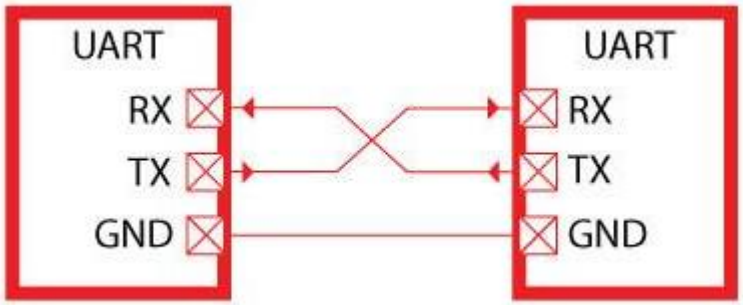
单片机



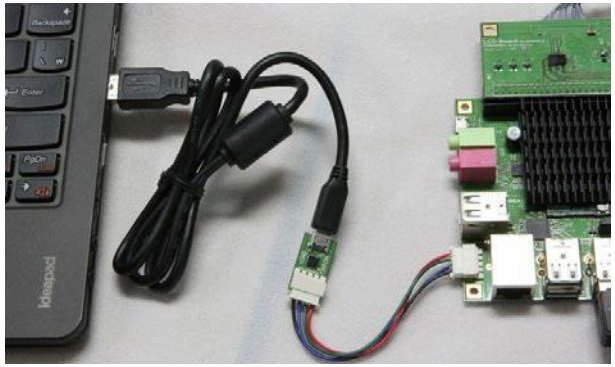
嵌入式系统

固件逆向分析-固件提取

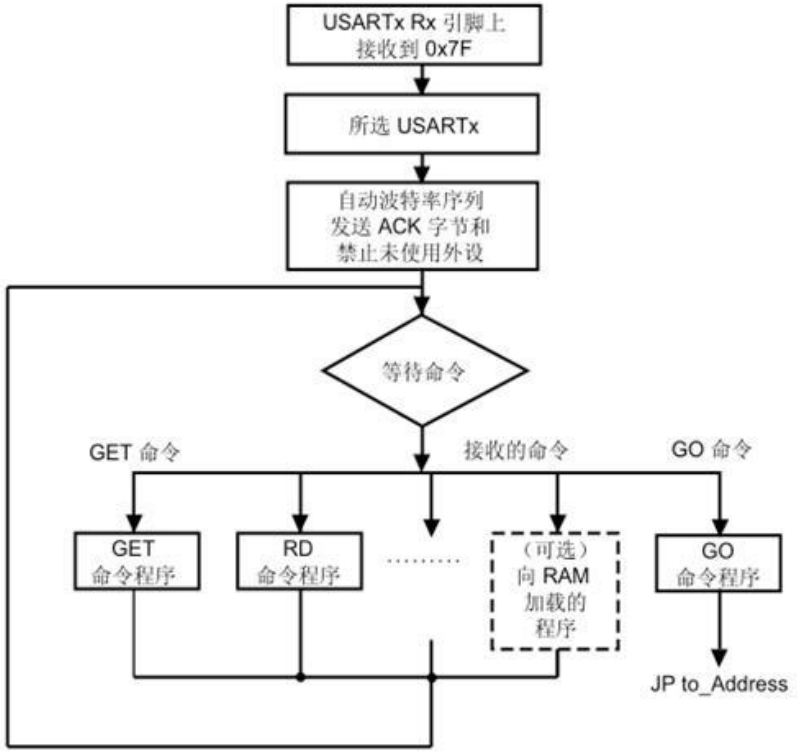
- 集中式存储-ISP(UART)接口提取



UART连接方式



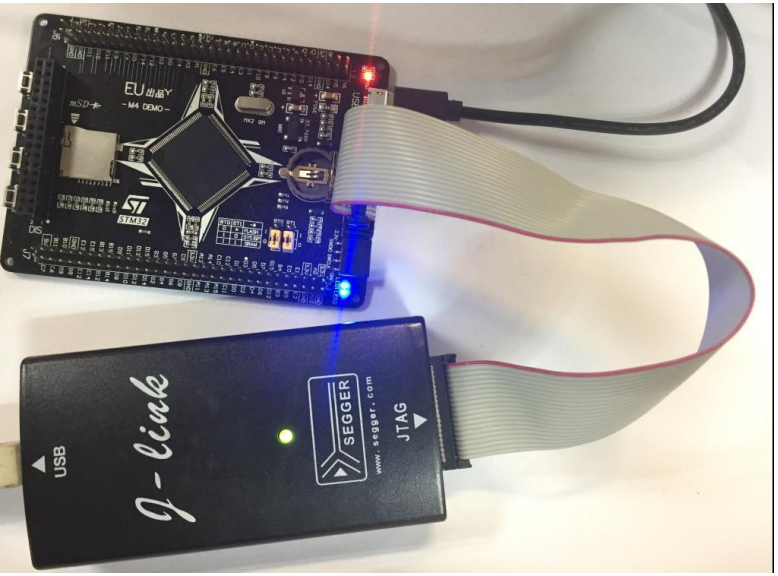
上下位机连接



ISP工作流程

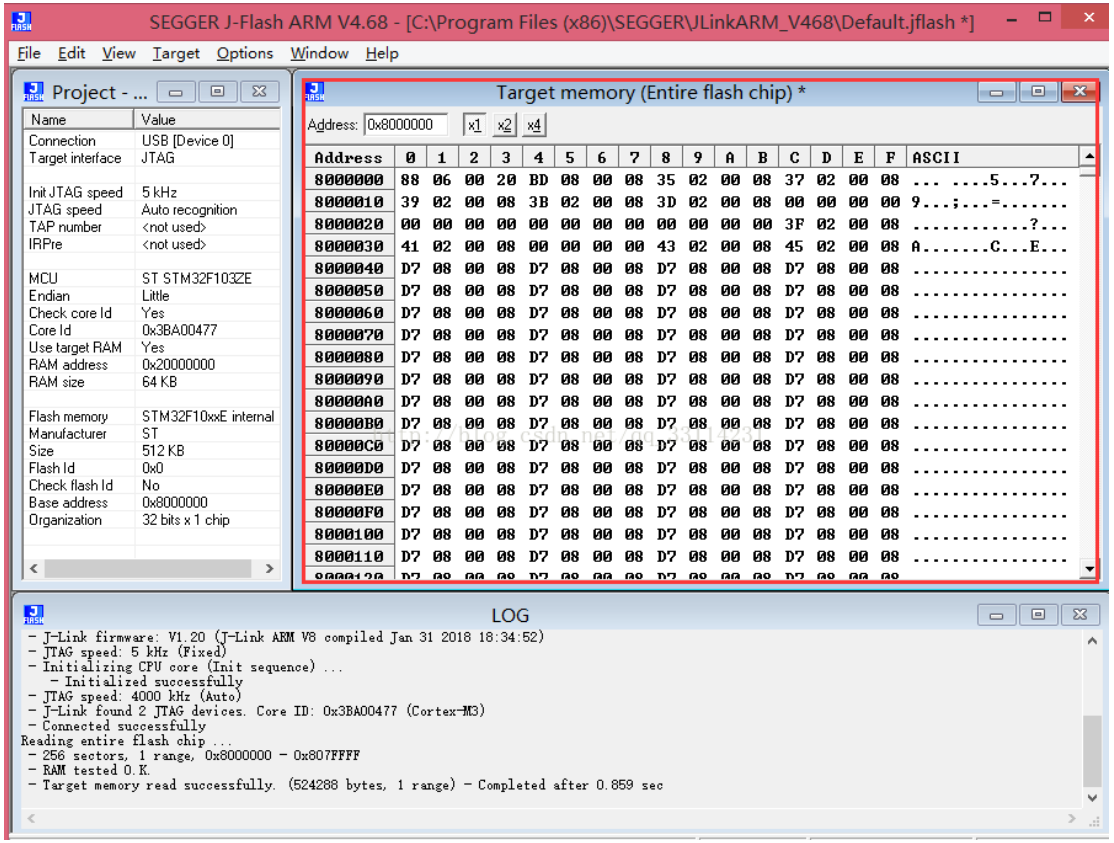
固件逆向分析-固件提取

- 集中式存储-Debug(JTAG)接口提取



J-Link调试器

其它调试器



调试器提取固件

固件逆向分析-固件提取

- 集中式存储-单片机破解提取

Crack MCU Brand List

Actel	Altera	AMD
Atmel		
ChipON	Coreriver	Cygnal
Cypress		
Dallas		
Elan		
Feeling	Freescale	Fujitsu
Gould		
Holtek	Hynix	Hyundai
Hitachi		
ICT	ICSI	Infineon
Intel	ISSI	

固件逆向分析-固件提取

- 集中式存储-单片机破解提取

"单片机解密"是什么?

张巧龙 大鱼机器人 6天前

单片机（MCU）一般都有内部程序区和数据区（或者其一）供用户存放程序和工作数据（或者其一）。为了防止未经授访问或拷贝单片机的机内程序，大部分单片机都带有加密锁定位或者加密字节，以保护片内程序。

如果在编程时**加密锁定位**被使能（锁定），就无法用普通编程器直接读取单片机内的程序，这就叫单片机加密。**（BugPS:单片机程序基本上都存在于Flash中，大部分能够读取或者识别Flash上的数据就能够获得Firmware文件，从而给复制产品带来了机会）**

单片机攻击者借助专用设备或者自制设备，利用单片机芯片设计上的漏洞或软件缺陷，通过多种技术手段，就可以从芯片中提取关键信息，获取单片机内程序这就叫单片机解密。

1 什么是单片机解密?

2 单片机解密方法

2.1 软件攻击
2.2 电子探测攻击
2.3 过错产生技术
2.4 探针技术

3 单片机解密分类

4 侵入式解密过程

5 单片机解密几点建议

固件逆向分析-固件提取

- 分离式存储

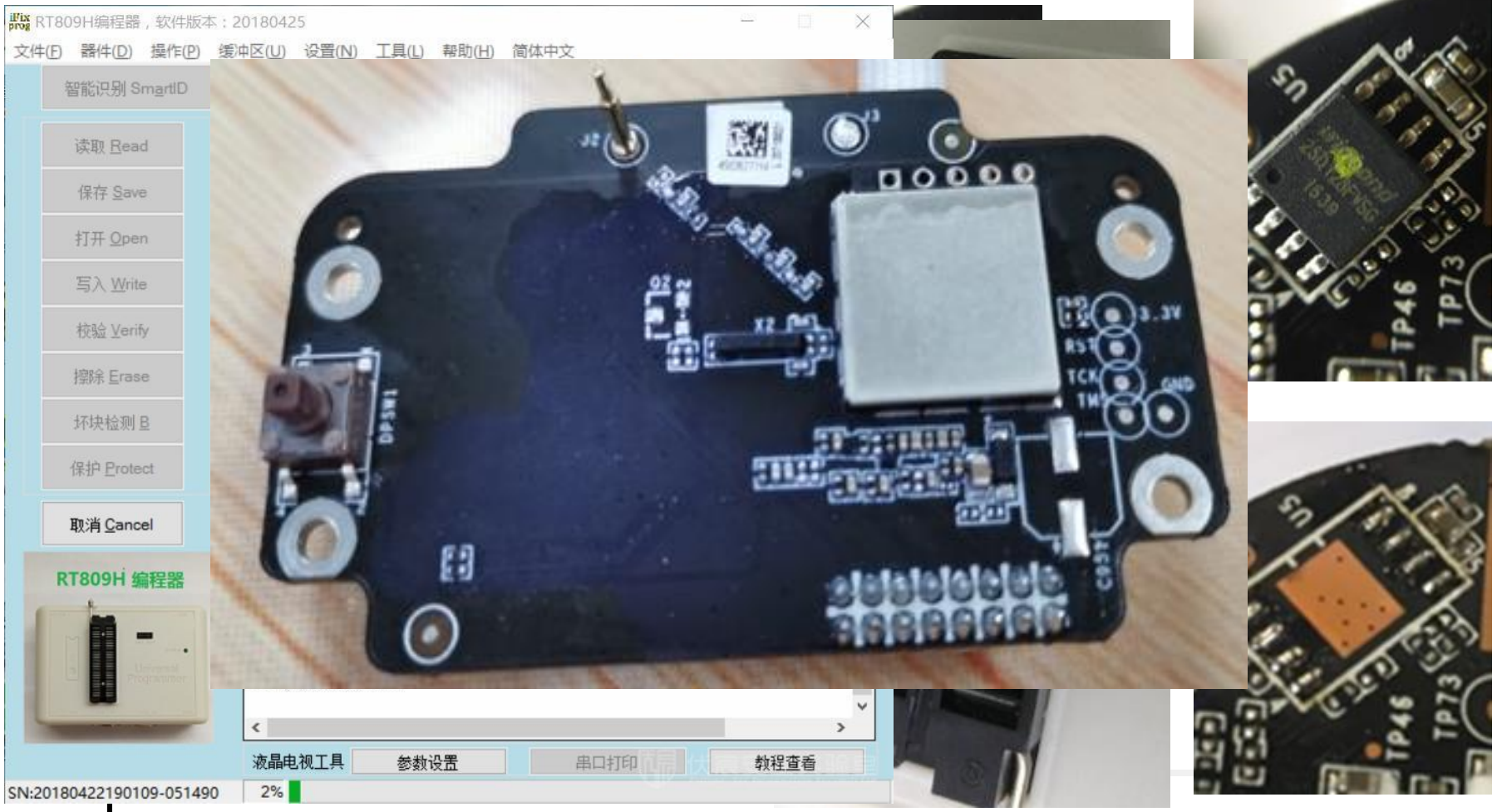
硬件方式

软件方式

运维方式

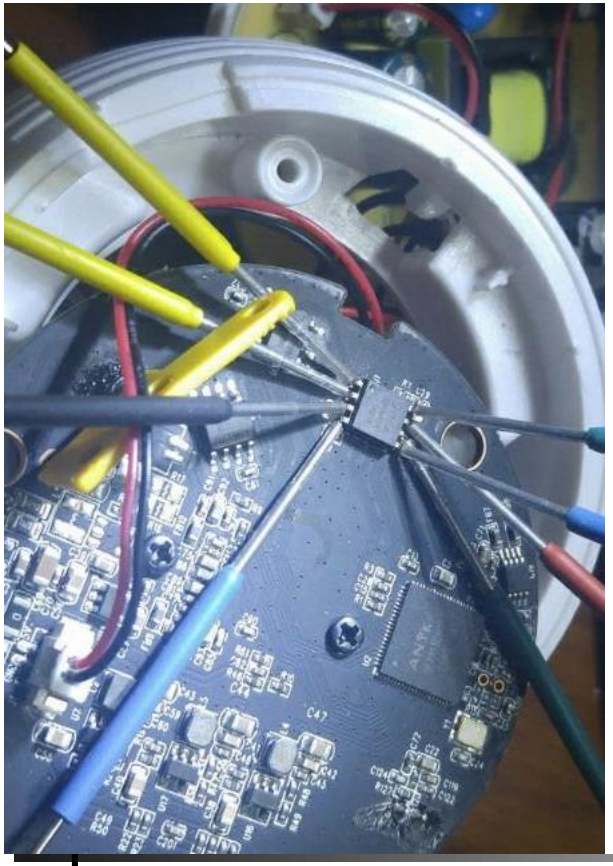
固件逆向分析-固件提取

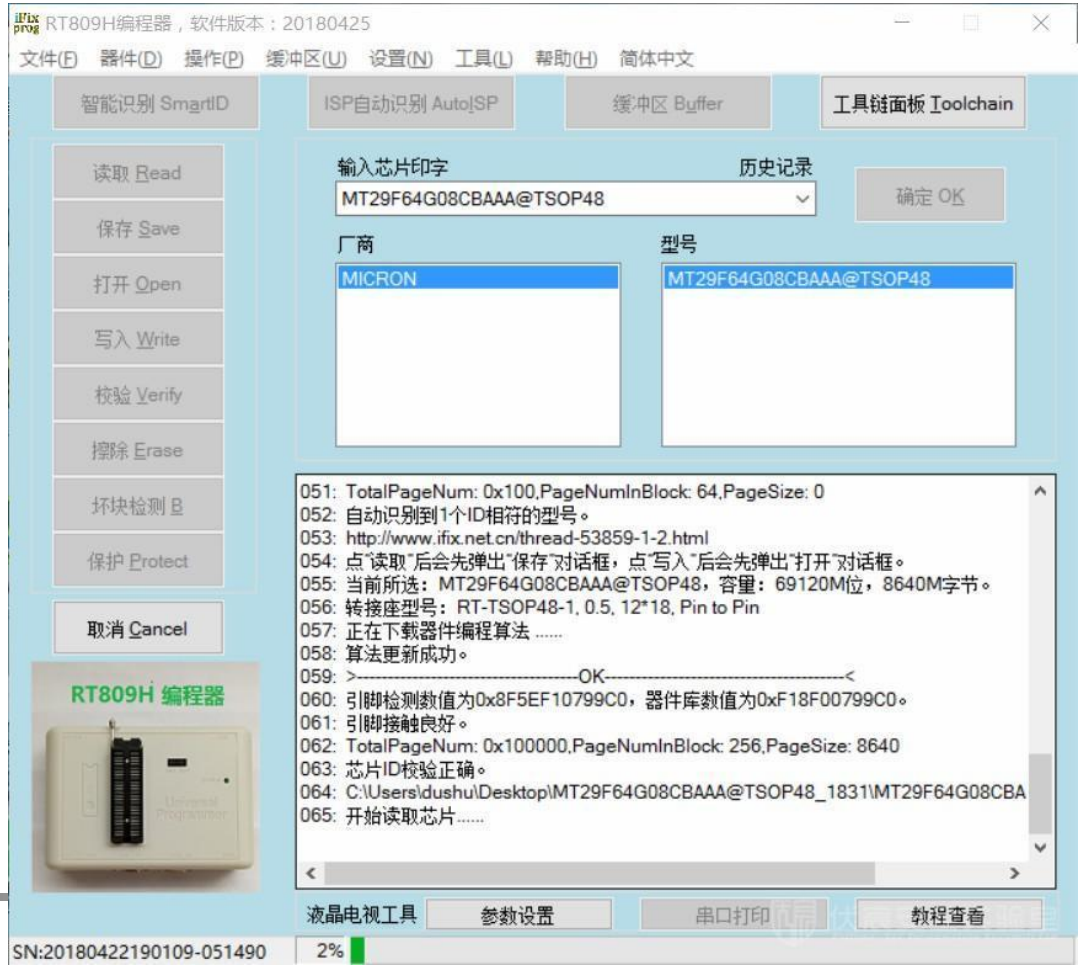
- 分离式存储-硬件方式-焊下读



固件逆向分析-固件提取

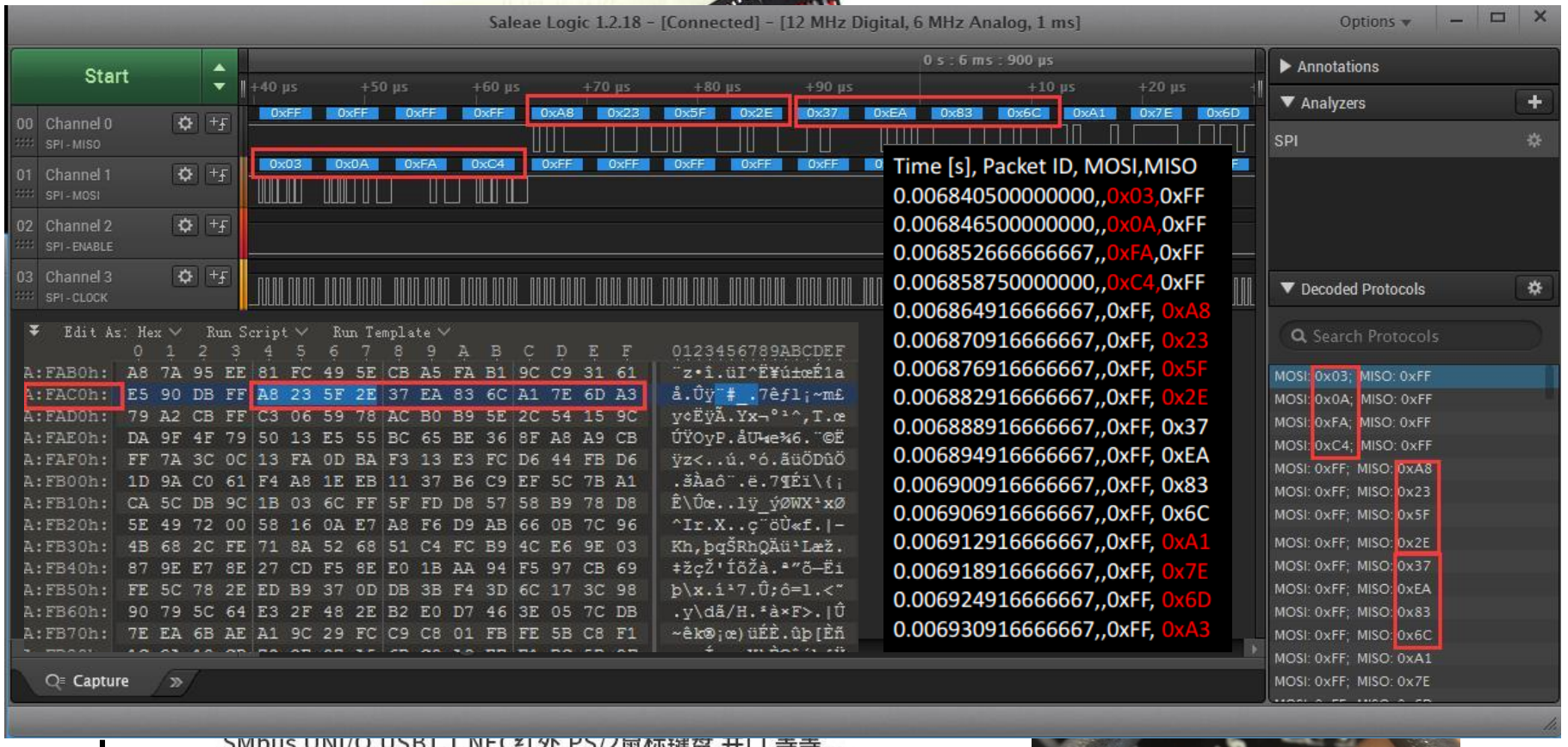
- 分离式存储-硬件方式-片上读





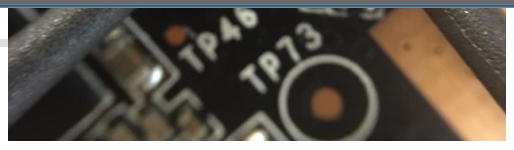
固件逆向分析-固件提取

- 分离式存储-硬件方式-片上听



SMBus,UNI/O,USB1.1,NEC红外,PS/2鼠标键盘,并口,等等...

- 4.10进制/16进制/2进制/ASCII码格式数据显示,方便观察分析
- 5.采样数据与解析数据均可导出,方便存档或其它分析软件使用



固件逆向分析-固件提取

- 软件方式

7、认证后的多个任意命令执行漏洞

第一个：设备通过CloudSetup.cgi支持Avtech云服务，在登录认证通过之后，由于没有对参数进行验证，可以通过exefile参数以root权限执行任意命令。

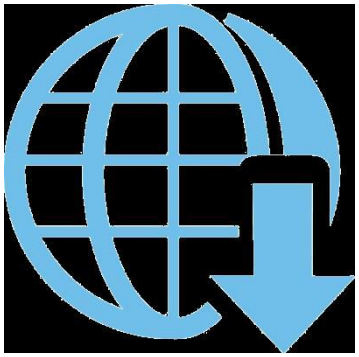
```
http://<device_ip>/cgi-bin/supervisor/CloudSetup.cgi?exefile=ps
```

第二个：部分设备支持ActionD命令，通过adcommand.cgi文件实现，新版本设备的ActionD提供了DoShellCmd功能，在认证通过之后，由于没有对参数进行验证，可以以root权限执行任意命令。此功能需要以post方式实现，其中cookie中的SSID为用户名和密码的base64值。

```
POST /cgi-bin/supervisor/adcommand.cgi HTTP/1.1

Host: <device_ip>

Content-Length: 23

Cookie: SSID=YWRtaW46YWRtaW4=

DoShellCmd "strCmd=ps&"
```

固件逆向分析-固件提取

- 运维方式（第三方获取）
  - 官网下载客服索取
  - 网络升级截获
  - 维修服务商



左搜 LEFTSO.COM 首页 搜索 编程技术 软件硬件 玩机篇 SEO 生活日常 更多

小米路由器3G刷固件Padavan_小米路由器3G刷Breed

首先来张小米路由器3G照片正楼，Padavan约等于华硕梅林远房亲戚也就是联发科版本首先来张小米路由器3G照片正楼，Padavan约等于华硕梅林远房亲戚也就是联发科版本。也可以说叫小米路由器刷“梅林”小米路由器3G刷机步骤下载所需工具

玩机篇 小米路由器3G padavan固件

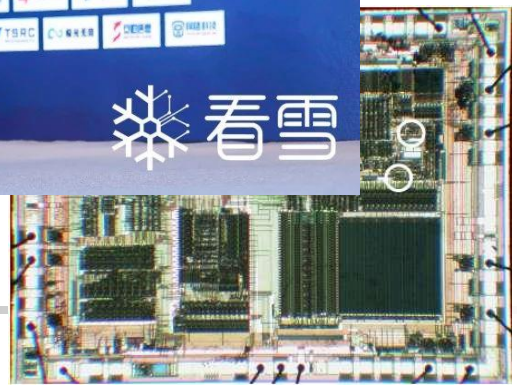
padavan 编译设置默认LAN地址等默认信息

由于最近刷机量多由于最近刷机量多。遇到一些子网络冲突得情况。很多设备都把默认得LAN设置为192.168.1.1，为了不冲突我决定，重新设置默认得padavan LAN ip信息。padavan 编译设置默认LAN地址等默认信息配置文件路径：/opt/rt-

玩机篇 padavan

固件逆向分析-固件提取

- 技术难点
  - 芯片保护
- 解决方案
  - 芯片解密
  - 逆向



固件逆向分析-固件内容分析

- 固件逆向分析方法是在不对嵌入式系统进行实际运行的情况下，通过对固件文件进行逆向解析，分析固件中各代码模块的调用关系及代码内容，从而发现嵌入式系统中可能存在的漏洞及后门。
- 涉及固件的识别和解压、固件的静态分析等技术。

固件逆向分析-固件内容分析

- 固件识别与解压

Device	Vendor	OS	Binwalk	BAT
Camera	STL	Linux	不支持	支持
Router	Bintec	-	不支持	不支持
ADSL Gateway	Zyxel	ZynOS	支持	支持
PLC	Siemens	-	支持	支持
DSLAM	-	-	支持	支持
PC	Intel	BIOS	支持	支持
ISDN Server	Planet	-	支持	支持
Voip	Asotel	Vxworks	支持	支持
Modem	-	-	不支持	不支持
Home Automation	Belkin	Linux	不支持	不支持

固件逆向分析-固件内容分析

- 固件识别与解压：固件解密
  - 如何判断固件加密

使用binwalk查看一下固件的信息，如果是未加密的固件，通常可以扫描出来使用了何种压缩算法。以常见的嵌入式文件系统squash-fs为例，比较常见的有LZMA、LZO、LAMA2这些。如下是使用binwalk分别查看一个未加密固件（netgear）和加密固件（DIR 3040）信息。

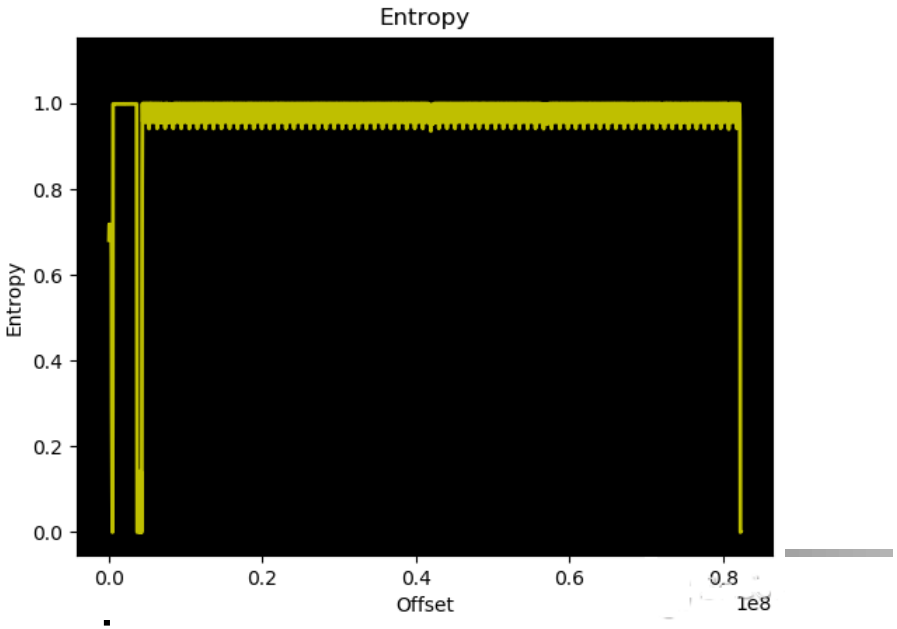
```
$ binwalk GS108Tv3_GS110TPv3_GS110TPP_V7.0.6.3.bix

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
64            0x40            LZMA compressed data, properties: 0x5D, dictionary size: 67
```

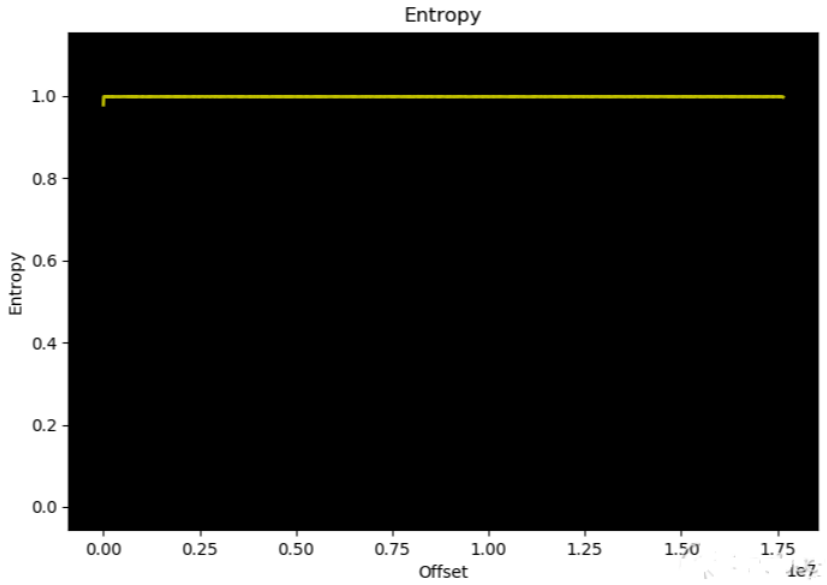
```
$ binwalk DIR3040A1_FW113B03.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
```

固件逆向分析-固件内容分析

- 固件识别与解压：固件解密
  - 如何判断固件加密

还有一种方式就是查看固件的熵值。熵值是用来衡量不确定性，熵值越大则说明固件越有可能被加密或者压缩了。这个地方说的是被加密或者压缩了，被压缩的情况也是会让熵值变高或者接近1的，如下是使用`binwalk -E`查看一个未加密固件（RAX200）和加密固件（DIR 3040）。可以看到，RAX200和DIR 3040相对比，不像后者那样直接全部是接近1了。



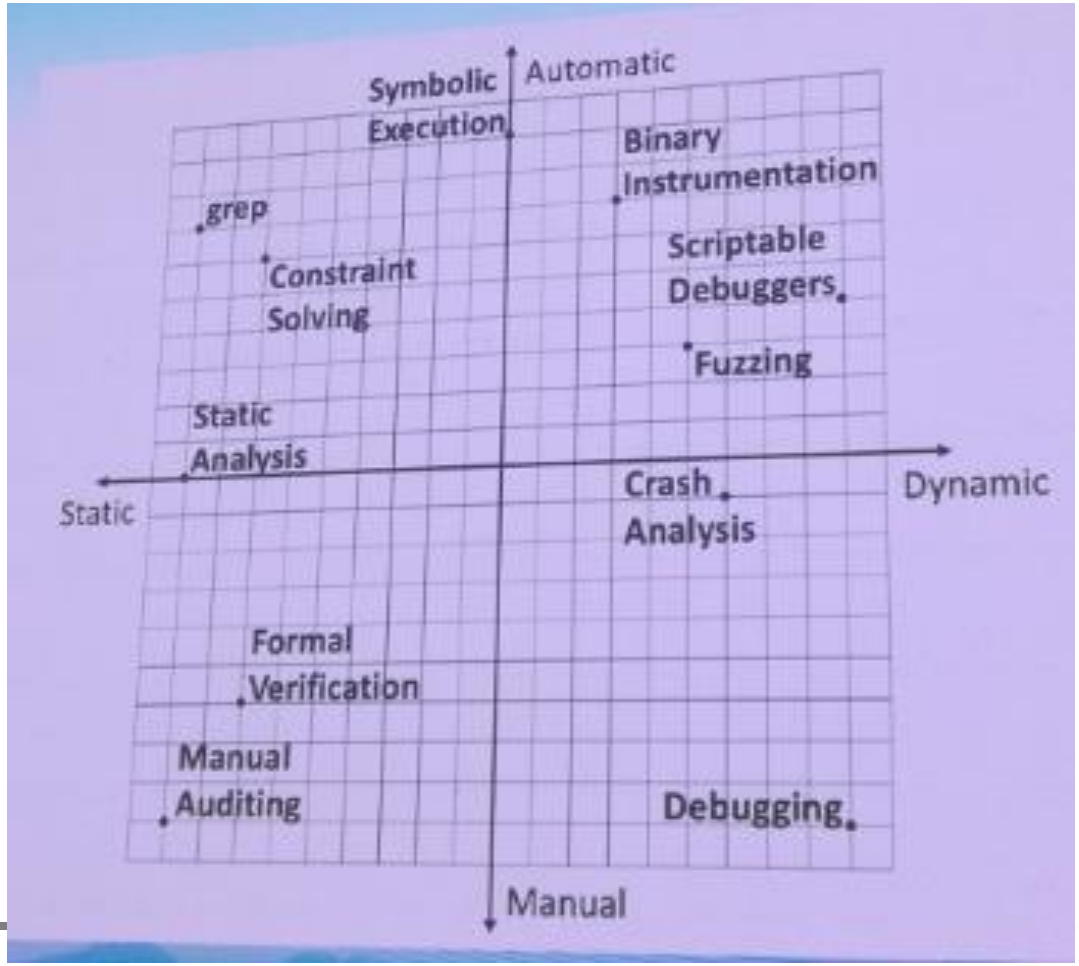


固件逆向分析-固件程序分析

- 固件程序分析：
  - 得到固件中的程序之后，就可以开始逆向分析了，常用的分析工具有IDA、Ghidra、Cutter、Radar2等。与Windows逆向、Android逆向不同的是，智能设备的MCU是多种多样的，这就意味着在进行逆向工作时会见到各种各样的指令集，在家用设备中最常见的有x86、ARM、MIPS、PowerPC等。

固件程序分析

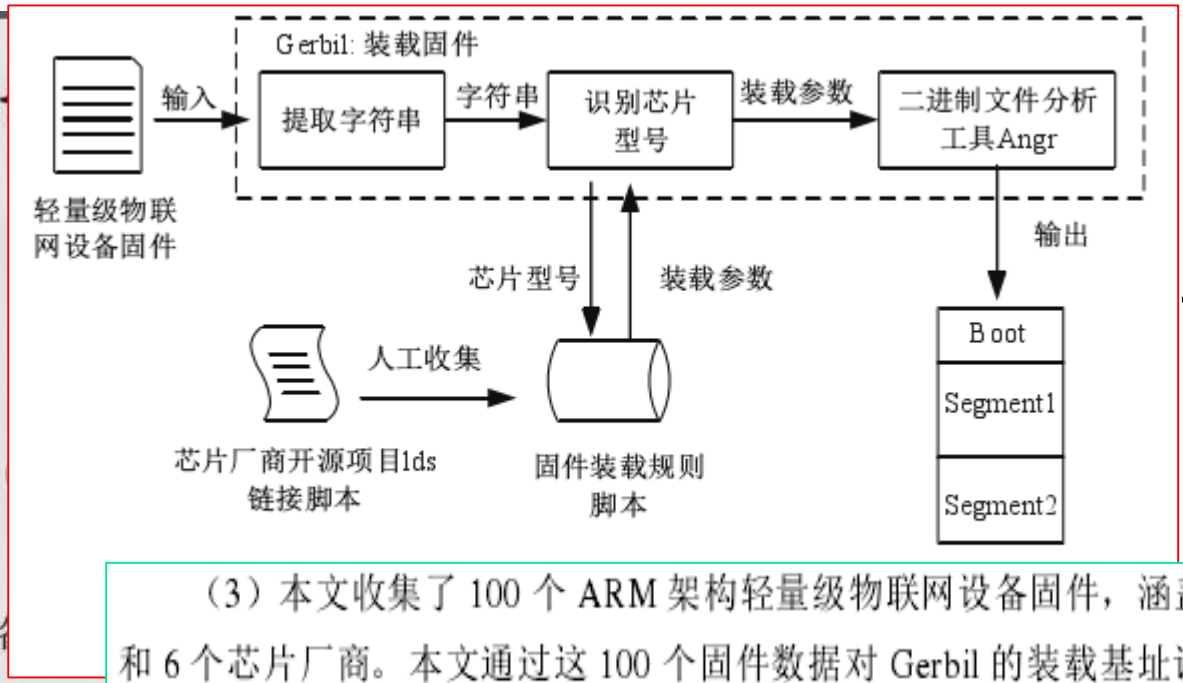
- 基本分析思路与关键技术
  - 静态分析
  - 动态分析
  - 动静混合



固件程序分析

- 现有分析平台主要基于Windows, Linux, 以X86架构为主

固件程序逆向分析



正确

正确

硕士

轻量级物联网设

作者姓名 姚

指导教师姓名、职称

申请学位类别 工

（3）本文收集了 100 个 ARM 架构轻量级物联网设备固件，涵盖 4 个设备厂商和 6 个芯片厂商。本文通过这 100 个固件数据对 Gerbil 的装载基址识别、控制流图恢复、检测时间以及检测结果等性能进行评估。实验结果表明，Gerbil 可以实现准确率为 100% 的固件装载基址识别。同时，与 Angr 的分析结果相比，Gerbil 对单个固件控制流图的恢复结果平均增加了 24,822 个节点和 58,625 条边。此外，Gerbil 分析单个固件的所有认证路径的平均时间仅为 527.41 秒，并检测出 37 个固件中存在指令混合型认证绕过漏洞。本文通过人工测试 3 个智能设备来验证检测到的漏洞，证明了 Gerbil 检测结果的正确性。测试结果表明，本文发现的指令混合型认证绕过漏洞允许攻击者实现设备劫持和设备拒绝服务攻击。

固件程序逆向分析

- 固件静态分析
  - 固
    - 识别
    - ）
    - 号表
    - 的
    - 数名

2.4.3 快速库函数识别技术 FLIRT

快速的库识别与鉴定技术 FLIRT[72] 是 IDA Pro 用于快速识别库函数的主要方法，能够极大的提高反汇编的可读性。FLIRT 技术为每个库文件生成一个签名文件，这个签名文件实际上是一个数据库，记录了这个库文件中每个库函数的模式。函数模式包括：函数的前 32 个字节、函数的 CRC16 以及它调用其他函数的路径。因为

17

同一个库函数在编译时部分字节会变化，因此函数的前 32 个字节中包含变体字节，用“.”表示。FLIRT 技术就是依据签名文件中的这三个模式对二进制字节进行匹配，从而识别出二进制文件中的库函数。下面介绍如何使用 FLIRT 技术生产库函数签名文件以及如何识别库函数。

固件程序逆向分析

☆ An Empirical Study on ARM Disassembly Tools

Muhui Jiang, Yajin Zhou, Xiapu Luo, Ruoyu Wang, Yang Liu, Kui Ren

With the increasing popularity of embedded and mobile devices, ARM is becoming the dominant architecture in them. Accordingly, there is a pressing need to perform security assessments to these devices. Due to the fragmentation, it is an ongoing research question to dynamically run the systems of these devices (or the firmware) in an emulated environment. Mainly due to this, the static analysis approach is still a commonly used way. In particular, existing works usually leverage off-the-shelf disassembly tools to disassemble stripped (ARM) binaries, and assume that reliably disassembling them and identifying functions are solved problems. However, whether this assumption holds for real world ARM binaries is unknown.

In this paper, we conduct a comprehensive study on ARM disassembly tools. Specifically, we build 1,896 ARM binaries (including 248 obfuscated ones) with different compilers, compiling options, and obfuscation methods. Using these binaries, we then evaluate eight state-of-the-art ARM disassembly tools (including both commercial and noncommercial ones) on their capabilities to locate instruction and function boundaries. These two primitives are fundamental ones and could be leveraged to build other primitives. Based on our evaluation, we present observations that were not systematically summarized and/or confirmed previously. For instance, we find that the existence of both the ARM and the Thumb instruction sets, and the reuse of the BL instruction for both direct function call and direct branch bring serious challenges to disassembly tools. Our evaluation sheds light on the limitations of the state-of-the-art disassembly tools and points out potential directions to improve them. To engage the community, we will publicly release the compiled ARM binaries, the retrieved ground truth, and the result.

ISSTA 2020

固件程序逆向分析

- 使用机器学习的方法来实现二进制到源代码的转换

2020年CCF-腾讯犀牛鸟基金申请启动：“深度学习在软件安全领域的应用研究”课题

建议研究方向

- 计算机语言的表征和分类研究，例如识别二进制软件对应的编译器、编译优化选项、第三方库、开发作者等信息；
- 计算机语言的自动生成和翻译技术研究，例如自动生成用于编译器(解释器)模糊测试的符合语法结构的程序代码；
- 用机器翻译技术实现二进制和源代码之间的相互翻译工作；
- 面向复杂交互程序的智能分析方法研究，例如研究代码相似性分析、演化API的误用检测、基于程序内状态机的符号执行、用户界面和运行时事件等复杂输入驱动的软件测试方法。

固件程序逆向分析

- 使用机器学习的方法来实现二进制到源代码的转换

2021年CCF-腾讯犀牛鸟基金申请启动：“深度学习在软件安全领域的应用研究”课题

深度学习在软件安全领域的应用研究

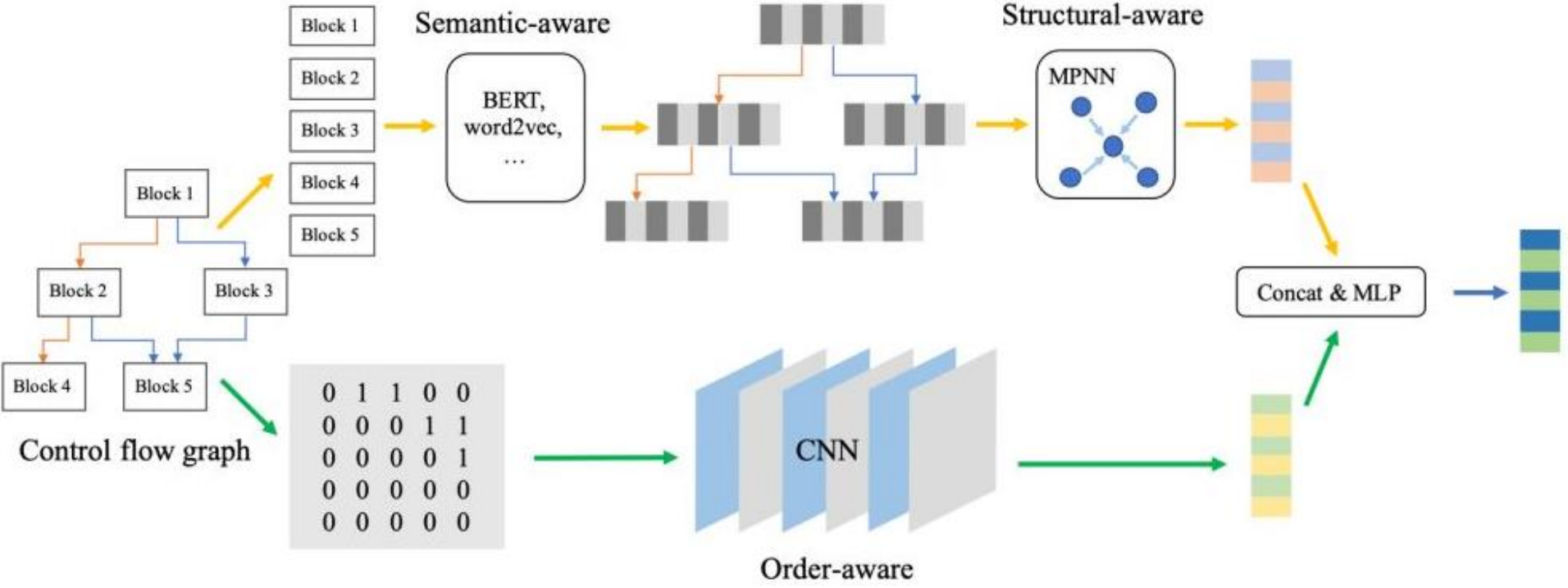
随着软件复杂度的不断提升，大规模源代码和二进制软件的漏洞挖掘工作面临新的机遇和挑战。本命题希望把深度学习相关技术（例如自然语言处理、图神经网络、深度强化学习等）应用于软件安全研究中，其成果可以对传统的逆向工程、模糊测试、漏洞挖掘等有较大促进。

建议研究方向

- 计算机语言的表征和分类研究，例如识别二进制软件对应的编译器、编译优化选项、第三方库、开发作者等信息。
- 计算机语言的自动生成和翻译技术研究，例如自动生成用于编译器(解释器)模糊测试的符合语法结构的程序代码；利用机器翻译技术实现二进制和源代码之间的相互翻译工作。
- 基于程序语义表征的安全属性分析研究，例如代码相似性分析、API 误用分析、已知/ 未知漏洞检索等。
- 二进制可执行文件的软件成分分析，如第三方库及其版本号等的分析与识别。

固件程序逆向分析

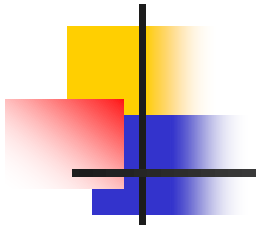
- 使用机器学习的方法来实现大规模二进制程序函数相似性分析，科恩实验室发表在AAAI2020上的论文“Order Matters: Semantic-Aware Neural Networks for Binary Code Similarity Detection”



固件程序逆向分析

Model	gcc-x64-O0		clang-arm-O3	
	Binary2Source	Source2Binary	Binary2Source	Source2Binary
BinPro	38.6 / 41.1	39.2 / 41.8	39.4 / 42.1	39.7 / 42.4
B2SFinder	34.1 / 39.6	34.4 / 39.8	33.5 / 39.2	34.2 / 39.5
TextCNN + HBMP	54.3 / 84.7	54.7 / 85.1	48.8 / 82.5	49.3 / 82.8
LSTM + HBMP	63.7 / 89.4	63.9 / 88.7	60.2 / 86.9	60.6 / 87.3
DPCNN + Word2vec	69.2 / 91.0	69.6 / 90.7	63.6 / 88.3	64.0 / 88.5
DPCNN + BERT	74.3 / 93.9	74.5 / 94.0	66.1 / 89.0	66.5 / 89.5
DPCNN + HBMP	80.8 / 96.4	81.2 / 96.6	72.9 / 91.2	73.2 / 92.1
Hungarian (Integer)	9.0 / 15.6	11.6 / 17.7	7.7 / 14.3	10.4 / 17.3
LSTM (Integer)	10.7 / 17.9	13.0 / 19.0	8.9 / 15.2	10.9 / 18.6
Integer-LSTM (Integer)	12.3 / 19.4	15.5 / 23.2	11.5 /17.1	12.2 / 20.7
Hungarian (String)	33.9 / 35.4	34.0 / 35.5	35.6 / 36.2	35.8 / 37.3
Hier-LSTM (String)	42.4 / 44.5	42.8 / 45.1	45.0 / 46.9	45.5 / 48.7
Random	81.9 / 97.3	82.3 / 98.0	74.2 / 92.0	74.8 / 92.6
Distance-Weight	86.2 / 97.4	86.5 / 97.8	77.4 / 94.3	78.2 / 94.7
Norm-Weight ($s = 0.5$)	85.3 / 97.2	85.4 / 97.5	76.9 / 93.4	77.5 / 93.6
Norm-Weight ($s = 2$)	89.0 / 97.9	89.1 / 98.2	81.2 / 95.1	82.5 / 95.4
Norm-Weight ($s = 5$)	**90.2 / 98.3**	**90.3 / 98.5**	**87.3 / 97.5**	**87.7 / 97.9**

{ravinyu, dodgetang, snie, shiwu}@tencent.com {zwx9810, yakkiwa}@gmail.com



（二）固件漏洞挖掘

IoT固件安全漏洞

- 逻辑漏洞：
  - 固件后门、认证绕过、加密不当等
- 代码漏洞
  - 栈溢出、整数溢出、堆溢出、格式化字符串、命令注入漏洞、UAF等

漏洞挖掘

- **Code Review** (10%?)
- Static analysis
- Dynamic Analysis
- Taint Analysis
- Symbolic Execution
- Model Checking
- **Fuzzing** (80?)

固件漏洞挖掘（Fuzzing）

- 要解决的问题
  - 让固件运行起来（动态分析平台）
  - 会话认证与通信加密问题
  - 故障定位
  - 高质量测试用例生成（提高路径覆盖率）

IoT固件漏洞研究

- **Avatar: A framework to support dynamic security analysis of embedded systems' firmwares, NDSS 2014**
  - 技术：半模拟，动态分析，符号执行，施加了强假设或依赖调试端口，白盒模糊测试
  - 点评：一个框架，通过将固件仿真与真实硬件一起编排，可以对嵌入式设备进行复杂的动态分析。指令在模拟器执行，外围设备IO被转发到真是设备，允许研究者应用高级动态分析技术，如跟踪，污染，符号执行等。

IoT固件漏洞研究

- **Firmalice-Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware, NDSS 2015**
  - 技术：静态分析，单独分析，黑盒，二进制，基于符号执行和程序切片
  - 点评：提供了一个框架，用于检测基于符号执行和程序切片的二进制固件中的身份验证绕过漏洞（后门）。先使用静态分析提取数据依赖图，然后提取从入口点到手动确定的特权操作位置的程序切片，应用符号执行引擎找到可能成功的路径。然而，它受到约束求解器的较大影响

IoT固件漏洞研究

- **Scalable graph-based bug search for firmware images, ACM CCS 2016**
  - 技术：基于代码分析；基于模式匹配（代码相似性）的静态分析方法
  - 点评：将CFG转换为高级数字特征向量，对跨架构的代码鲁棒性更强。

IoT固件漏洞研究

- **IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing, NDSS 2018**
  - 技术：动态分析，基于生成的模糊测试，只关注面向网络的二进制文件，黑盒fuzz，通过配套app指导fuzz
  - 点评：缺乏对生成输入质量的考虑，会造成资源浪费。分析 Android 应用程序以检测物联网设备中与内存相关的漏洞。 通过改变应用程序中的数据流，跳过了协议分析。

IoT固件漏洞研究

- **FIRM-AFL: High-Throughput Greybox Fuzzing of IoT Firmware via Augmented Process Emulation, USENIX Security 2019**
  - 技术：动态分析，基于变异的fuzz，结合AFL和Firmadyne
  - 点评：一种基于变异的物联网固件灰盒模糊测试平台。 通过在用户模式模拟器中运行目标程序并在目标程序调用具有特定硬件依赖性的系统调用时切换到全系统模拟器来实现高吞吐量模糊测试。这项工作解决了性能瓶颈。但是，Firm-AFL 侧重于单个程序的覆盖范围，难以触发程序间漏洞。

IoT固件漏洞研究

- **P2IM: Scalable and hardware-independent firmware testing via automatic peripheral interface modeling, USENIX Security 2020**
  - 技术：动态分析，固件模拟，模糊测试
  - 点评：实现独立于硬件和可扩展的固件测试。抽象了各种外设，并基于自动生成的模型动态处理固件 I/O。P2IM 无视外设设计和固件实现的通用性，因此适用于各种嵌入式设备。

IoT固件漏洞研究

- **Sharing More and Checking Less: Leveraging Common Input Keywords to Detect Bugs in Embedded Systems, USENIX Security 2021**
  - 技术：利用前后端共享关键字作为污点分析开始位置，降低符号执行复杂度
  - 点评：基于前后端共享关键字来指导漏洞挖掘。前端输入->后端逻辑->RCE漏洞。通过关键字串联起前端输入和后端处理逻辑，从而定位漏洞触发流程。

IoT固件漏洞研究

- IoT固件中密码学误用漏洞自动化分析与检测

CRYPTOREX: Large-scale Analysis of Cryptographic Misuse in IoT Devices

Li Zhang*, Jiongyi Chen†, Wenrui Diao‡§(✉), Shanqing Guo‡§, Jian Weng*, and Kehuan Zhang†
*Jinan University, {zhanglikernel, cryptjweng}@gmail.com
†The Chinese University of Hong Kong, {cj015, khzhang}@ie.cuhk.edu.hk
‡Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, {diaowenrui, guoshanqing}@sdu.edu.cn
§School of Cyber Science and Technology, Shandong University

固件漏洞挖掘（Fuzzing）

IoT fuzzing

- 2020 - ARM-AFL: Coverage-Guided Fuzzing Framework for ARM-Based IoT Devices
- 2020 - Bug detection in embedded environments by fuzzing and symbolic execution
- 2020 - FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware
- 2020 - EM-Fuzz: Augmented Firmware Fuzzing via Memory Checking
- 2020 - Verification of Embedded Binaries using Coverage-guided Fuzzing with System C-based Virtual Prototy
- 2020 - DICE: Automatic Emulation of DMA Input Channels for Dynamic Firmware Analysis
- 2020 - Fw-fuzz: A code coverage-guided fuzzing framework for network protocols on firmware
- 2020 - TAINT-DRIVEN FIRMWARE FUZZING OF EMBEDDED SYSTEMS THESIS
- 2020 - A Dynamic Instrumentation Technology for IoT Devices
- 2020 - Vulcan: a state-aware fuzzing tool for wear OS ecosystem
- 2020 - A Novel Concolic Execution Approach on Embedded Device
- 2020 - HFuzz: Towards automatic fuzzing testing of NB-IoT core network protocols implementations
- 2020 - FIRMCORN: Vulnerability-Oriented Fuzzing of IoT Firmware via Optimized Virtual Execution
- 2018 - IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing
- 2017 - Towards Automated Dynamic Analysis for Linux-based Embedded Firmware
- 2016 - Scalable Graph-based Bug Search for Firmware Images
- 2015 - SURROGATES: Enabling Near-Real-Time Dynamic Analyses of Embedded Systems
- 2015 - Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware
- 2014 - A Large-Scale Analysis of the Security of Embedded Firmwares
- 2013 - RPFuzzer: A Framework for Discovering Router Protocols Vulnerabilities Based on Fuzzing

https://github.com/0xricksanchez/paper_collection

IoT固件漏洞研究

- 综述论文
  - **A Survey of Security Vulnerability Analysis, Discovery, Detection, and Mitigation on IoT Devices, Future Internet, 2020**
  - **Automatic Vulnerability Detection in Embedded Devices and Firmware: Survey and Layered Taxonomies, ACM Computing Surveys, 2021**
  - 物联网固件安全缺陷检测研究进展, 信息安全学报, 2021
  - 嵌入式设备固件安全分析技术研究, 计算机学报, 2021
  - 物联网设备漏洞挖掘技术研究综述, 信息安全学报, 2021
  - 物联网设备软件安全综述, 广州大学学报, 2019

固件安全分析技术研究综述

计 算 机 学 报
2020 CHINESE JOURNAL OF COMPUTERS Online

嵌入式设备固件安全分析技术研究综述

于颖超[1)] 陈左宁[2)] 甘水滔[1)] 秦晓军[1)]

[1)](数学工程与先进计算国家重点实验室，江苏无锡 214083)

[2)](中国工程院，北京 100088)

摘 要 近年来，随着嵌入式设备的种类和数量的增加，设备之间日益增长的互联互通、制造商对安全的忽视、设备固件更新不及时或难以更新等，使其安全性成为了一个突出的问题，越来越多的设备漏洞被披露。虽然当前不断呈现出国内外安全专家和学者对嵌入式设备固件的安全分析和评测技术相关工作，但缺乏详细和全面介绍最新安全研究成果的论文，使研究人员难以系统了解嵌入式设备固件安全分析技术的研究进展。为解决此问题，本文围绕着当前嵌入式设备固件面临的安全风险，分析和总结了国内外最新的研究成果，并对相关安全技术进行了综合分析和评估。首先对嵌入式设备及其固件的表现形式、分类及获取方法、面临的安全攻击层面以及自动化解析情况进行了深入研究。然后，对嵌入式设备固件安全分析技术进行了细化分析，从静态分析、符号执行、二进制漏洞关联、动态分析平台和模糊测试等五个方面进行了详细分析和横向评估。最后对未来的研究方向进行了展望。

（三）固件动态分析平台

固件动态分析

- 与Windows上的应用程序不同，很多固件的运行依赖硬件平台。
- 动态分析固件之前，需要先把固件运行起来，但我们手头又没有路由器、摄像头之类的物联网硬件，该如何运行呢？

然而，嵌入式设备对分析人员来说属于“黑盒”状态，攻击载荷在嵌入式系统内部的执行过程难以分析。为了解决这个问题，研究人员提出固件动态仿真分析的方法，在仿真器上运行固件系统并动态分析载荷在固件系统中的执行状态，并试图克服物联网设备异构性、多样性等挑战。

动态分析平台

- 模拟与插桩：让固件运行起来，监控程序的运行过程，获得程序的执行轨迹
- 难点分析：固件类型的多样化

动态分析平台

- **ACM Computing Surveys, 54 (1): 1-36**

Challenges in Firmware Re-Hosting, Emulation, and Analysis

CHRISTOPHER WRIGHT, Purdue University
WILLIAM A. MOEGLEIN, Sandia National Laboratories
SAURABH BAGCHI and MILIND KULKARNI, Purdue University
ABRAHAM A. CLEMENTS, Sandia National Laboratories

System emulation and firmware re-hosting have become popular techniques to answer various security and performance related questions, such as determining whether a firmware contain security vulnerabilities or meet timing requirements when run on a specific hardware platform. While this motivation for emulation and binary analysis has previously been explored and reported, starting to either work or research in the field is difficult. To this end, we provide a comprehensive guide for the practitioner or system emulation researcher. We layout common challenges faced during firmware re-hosting, explaining successive steps and surveying common tools used to overcome these challenges. We provide classification techniques on five different axes, including emulator methods, system type, fidelity, emulator purpose, and control. These classifications and comparison criteria enable the practitioner to determine the appropriate tool for emulation. We use our classifications to categorize popular works in the field and present 28 common challenges faced when creating, emulating, and analyzing a system from obtaining firmwares to post emulation analysis.

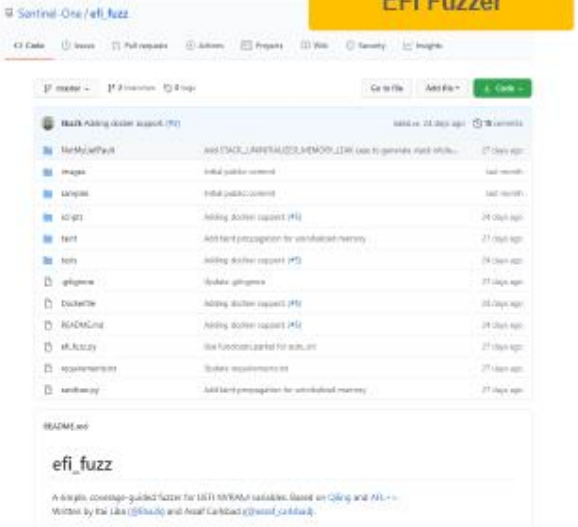
动态分析平台

麒麟框架

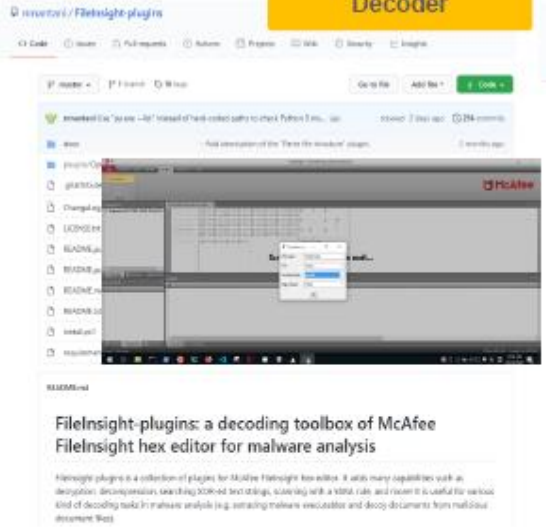
不是虚拟机的虚拟机

Framework, NOT Tools

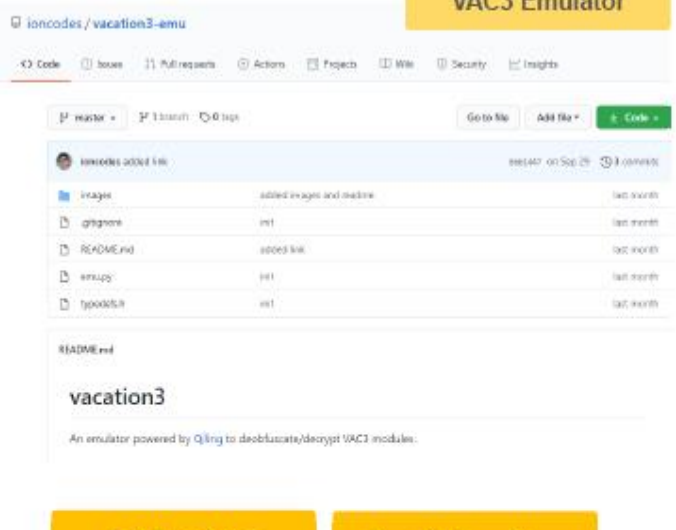
EFI Fuzzer



Decoder



VAC3 Emulator



IoT Emulator

MacOS Emulator

Binary Fuzzer

IoT Fuzzer

Malware Sandbox

CTF Solver

IOS Emulator

Binary Decrypt

Qiling Framework

CPU Architecture

Loader

OS

Debugger

Extensions

Instrumentation (Qiling's API)

动态分析平台

- 利用 QEMU 的方案为 IoT 设备提供虚拟化环境

动态分析平台

- *u*EMU

Automatic Firmware ...rence

Wei ...

[1]*National Computer Netw...*
[2]*D...*
[3]*College of Info...*
[5]*School of C...*

Abstract

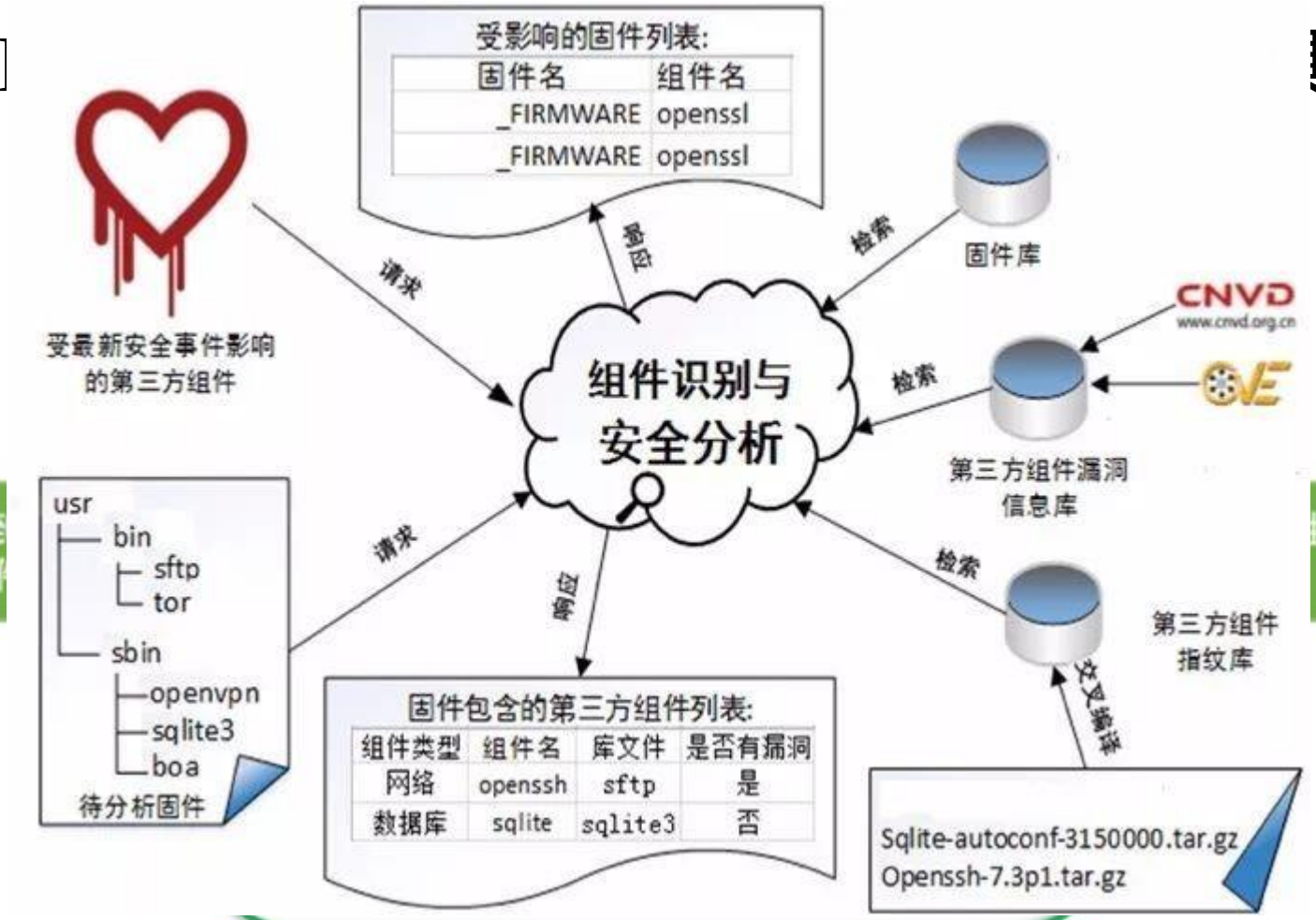
Emulating firmware for microcontrollers is challenging due to the tight coupling between the hardware and firmware. This has greatly impeded the application of dynamic analysis tools to firmware analysis. The state-of-the-art work automatically models unknown peripherals by observing their access patterns, and then leverages heuristics to calculate the appropriate responses when unknown peripheral registers are accessed. However, we empirically found that this approach and the corresponding heuristics are frequently insufficient to emulate firmware. In this work, we propose a new approach called *μEmu* to emulate firmware with unknown peripherals. Unlike existing work that attempts to build a general model for each peripheral, our approach learns how to correctly emulate firmware execution at individual peripheral access points. It takes the image as input and symbolically executes it by representing unknown peripheral registers as symbols. During symbolic execution, it infers the rules to respond to unknown peripheral accesses. These rules are stored in a knowledge base, which is referred to during the dynamic firmware analysis. *μEmu* achieved a passing rate of 93% in a set of unit tests for peripheral drivers without any manual assistance. We also evaluated *μEmu* with real-world firmware samples and new bugs were discovered.

（四）固件敏感信息搜索

敏感信息

- 问题分析
  - 用户名 / 口令、密钥硬编码
  - 证书 / 凭证文件
  - 关键系统信息（敏感IP/URL/DNS、未加密的数据文件/配置文件等、网络服务）

固件安全检测-CNCERT's FirmTool



固件安全检测-腾讯IoTSec

IoT 固件安全自动化检测平台

高准确率

基于常见攻击面、漏洞与安全风险模式建模信息，通过版本信息、数据流、控制流以及静态污点分析等方法匹配预设的漏洞模式，实现精准定位漏洞。

检测对象普适性

支持Linux、Android、VxWorks等各种物联网嵌入式操作系统，支持x86/x64、arm/arm64、mips、powerpc等主流cpu架构。

漏洞库完整性

漏洞库包含500余种第三方库超过1万个版本中的漏洞信息特征指纹，此外还有基于渗透测试经验抽象出的漏洞模式库，更完整的漏洞库信息有助于定位有效的安全风险问题。

提供漏洞的风险分析报告

系统自动生成漏洞报告，报告清晰展示漏洞名称、漏洞评分、漏洞描述以及相关匹配库信息，同时提供漏洞风险等级统计。

固件安全检测-UFO

- UFO: US BlackHat 2020

UFO: A Security Verification Tool for IoT Device Firmware

2018 IEEE International Symposium on Software Reliability Engineering Workshops

UFO - Hidden Backdoor Discovery and Security Verification in IoT Device Firmware

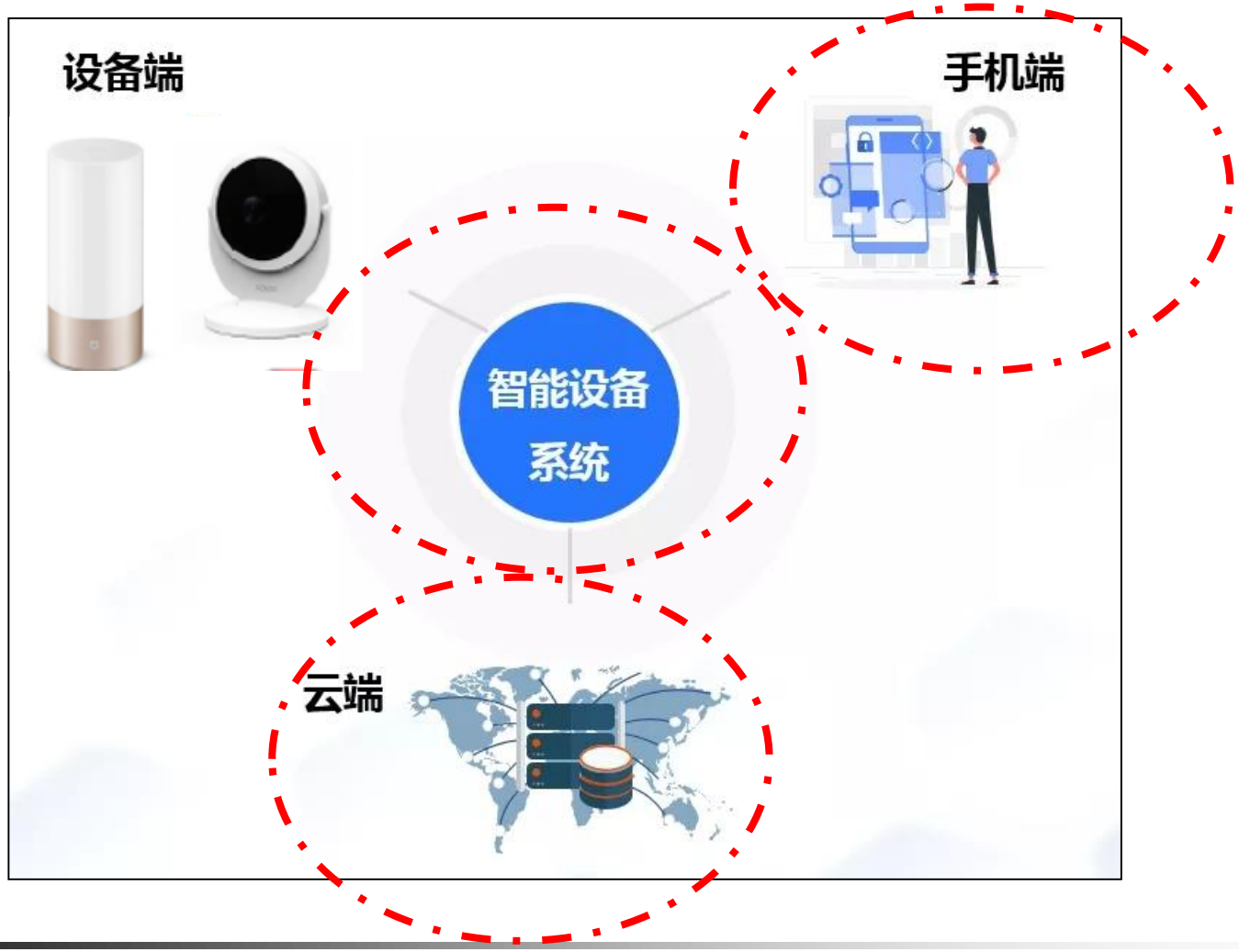
Chin-Wei Tien[1,2] Tsung-Ta Tsai[1] Ing-Yi Chen[3] Sy-Yen Kuo[2]

Cybersecurity Technology Institute, Institute for Information Industry[1]
Department of Electrical Engineering, National Taiwan University[2]
Department of Computer Science and Information Engineering, National Taipei University of Technology[3]
Taipei, Taiwan R.O.C.

- Cracked Passwords and Certificates Review: Check if your passwords or certificates are vulnerable.
- Shell Dependency Backdoor Paths: Produces a visual guide of backdoor paths.

二、其它研究方向

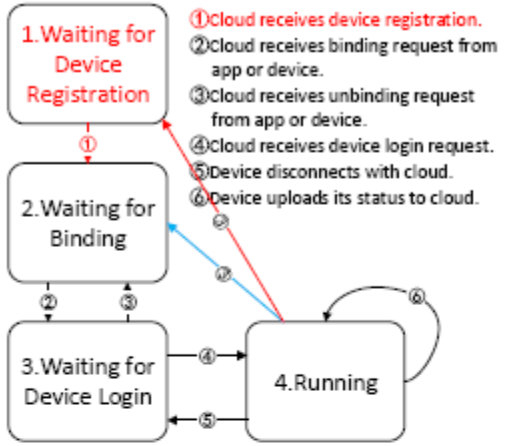
IoT组成



IoT安全评估

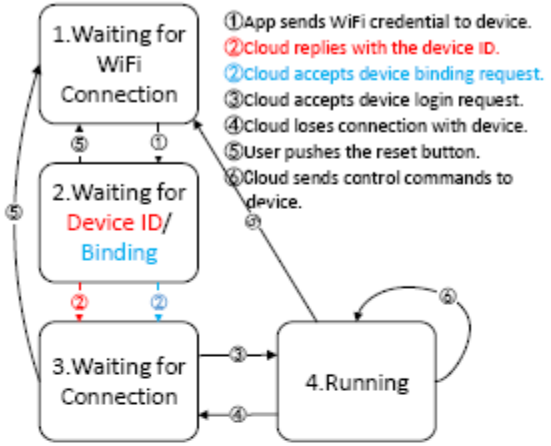
- 对某一类IoT设备从多个角度（固件、协议、App、软硬件开发平台、第三方组件的使用、API等）进行安全评估

IoT设备使用过程



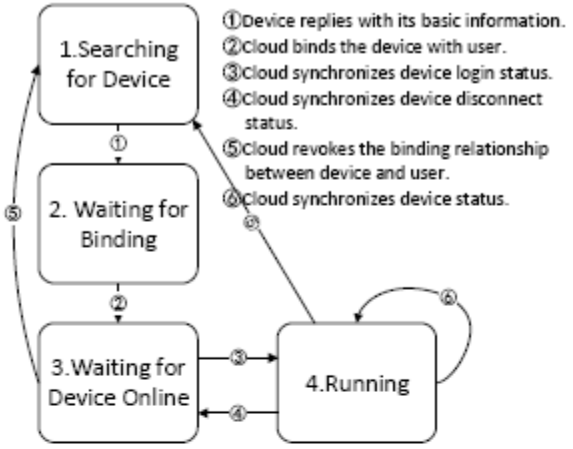
States	Description
Waiting for Device Registration	The cloud is in the initial state and waits for device registration request.
Waiting for Binding	The cloud waits for binding request from the mobile app or device.
Waiting for Device Login	After binding the device with user, the cloud waits for device login request.
Running	The cloud handles user's requests and syncs the device status.

(a) State Machine of an IoT Cloud



States	Description
Waiting for WiFi Connection	The device waits for discovery message and WiFi credential from app.
Waiting for Device ID/ Binding	The device sends register request/ binding request to the cloud, and waits for cloud response.
Waiting for Connection	Device keeps sending login request to get connection from cloud.
Running	Device keeps a connection with cloud and waits for new commands.

(b) State Machine of a Device



States	Description
Searching for Device	The mobile app keeps broadcasting discovery message to search for available device.
Waiting for Binding	The mobile app sends binding request to cloud or device, and waits for binding confirmation.
Waiting for Device Online	The mobile app waits for cloud to build connection with device.
Running	After device logs into the cloud, the mobile app receives user's instructions to the device via UI.

(c) State Machine of a Mobile App

Note: The states and transitions specific to *Type I* platforms are shown in red; the states and transitions specific to *Type II* platforms are shown in blue; and the shared states/transitions are in black.

Figure 2: High-Level State Machines of the Three Entities

IoT安全评估

- 从用户的视角，
  智能家居安全与

"It's the Company, the Government, You
Responsibility for Smart Home P

Julie Haney*, Yasemin Acar*†, and
*National Institute of Standards and Technology;
{julie.haney, susanne.furman}@nist.gov; a

Abstract

Smart home technology may expose adopters to increased risk to network security, information privacy, and physical safety. However, users may lack understanding of the privacy and security implications. Additionally, manufacturers often fail to provide transparency and configuration options, and few government-provided guidelines have yet to be widely adopted. This results in little meaningful mitigation action to protect users' security and privacy. But how can this situation be improved and by whom? It is currently unclear where *perceived responsibility* for smart home privacy and security lies. To address this gap, we conducted an in-depth interview study of 40 smart home adopters to explore where they assign responsibility and how their perceptions of responsibility relate to their concerns and mitigations. Results reveal that participants' perceptions of responsibility reflect an interdependent relationship between consumers, manufacturers, and third parties such as the government. However, perceived breakdowns and gaps in the relationship result in users being concerned about their security and privacy. Based on our results, we suggest ways in which these actors can address gaps and better support each other.

USENIX Security 2021

IoT 安全增强

- 从固件、协议、App等多个角度增强IoT设备的安全性

IoT 安全增强

How to increase security and reliability of IoT Apps and their interaction?

IoT Safety and Security

Soteria: Automated IoT Safety and Security Analysis [USENIX Annual Technical Conference, 2018] Z. Berkay Celik, Patrick McDaniel, and Gang Tan

IoTGuard: Dynamic Enforcement of Safety and Security Policy in Commodity IoT [NDSS, 2019] Z. Berkay Celik, Gang Tan, and Patrick McDaniel

IoT Privacy

Saint: Sensitive Information Tracking in Commodity IoT [USENIX Security, 2018] Z. Berkay Celik, Leo Babun, Amit Sikder, Hidayet Aksu, Gang Tan, Patrick McDaniel, and Selcuk Uluagac

IoT Fault Tolerance

IotRepair: Systematically Addressing Device Faults in Commodity IoT [Ongoing work] Michael Norris, Z. Berkay Celik, Prasanna Venkatesh, Shulin Zhao, Gang Tan, Patrick McDaniel, and Anand Sivasubramaniam

Surveys:

Program Analysis of IoT Applications for Security and Privacy: Challenges and Opportunities [ACM Computing Surveys, 2019] Z. Berkay Celik, Earlence Fernandes, Eric Pauley, Gang Tan, and Patrick McDaniel

Verifying Internet of Things Safety and Security in Physical Spaces [IEEE S&P magazine, 2019] Z. Berkay Celik, Patrick McDaniel, Gang Tan, Leo Babun, Salcuk Uluagac

IoT安全增强

- IoT

由此，作者设计了一个安全策略执行系统IoTSafe，如图3所示，该系统主要包含四个部分：

- App Analysis
  - 用于分析app中的静态控制流，收集app运行时用户配置
  - 利用代码分析和用户配置，生成包含触发条件和触发动作的细粒度静态交互流图。
- Real Physical Interaction
  - 用于识别真实设备之间的物理交互
  - 通过生成测试用例来执行动态测试，利用顺序测试和并行测试进一步生成包含设备状态和状态转换的有向的物理交互流图。
- Runtime Prediction
  - 用于维护物理交互模型，预测设备未来状态
  - 通过测试阶段收集到的数据来初始化物理模型，并根据新的条件设置或操作在线训练和更新交互流图；通过监视运行时事件，将当前状态与物理模型进行比较，以预测未来状态
- Policy Specification and Enforcement
  - 根据用户定义的策略检查违规操作
  - 能够从设备收集运行时信息并将其发送至服务器，服务器上的违规检测模块会将当前/预测情况与用户定义策略进行比较，以识别策略冲突

能

些方

交互

形

IoTS

with

IoT安全增强

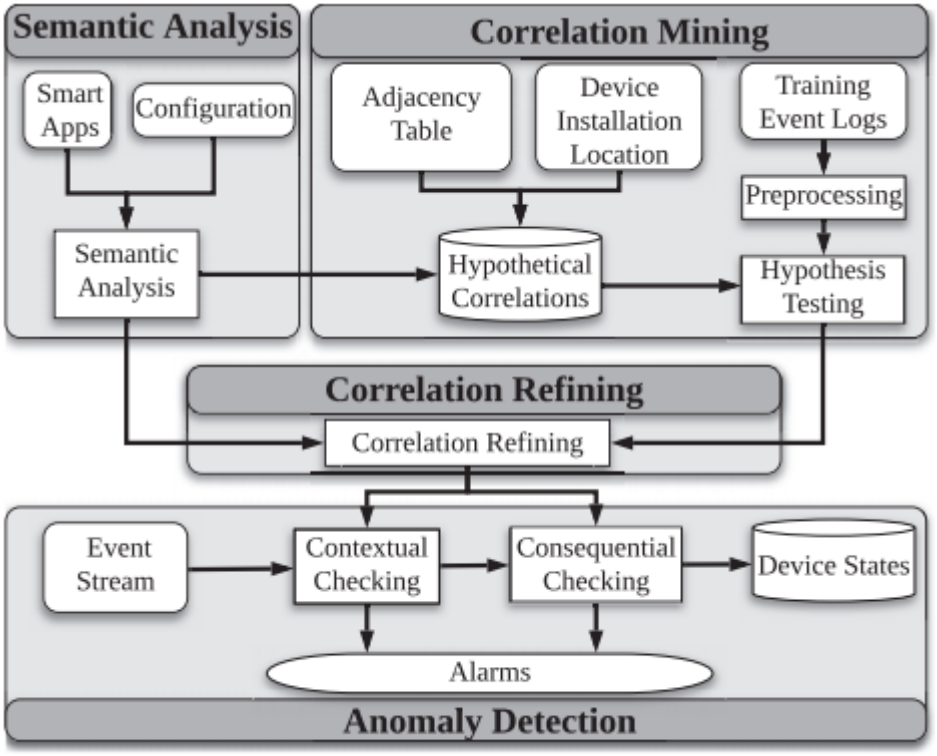


Figure 4: Architecture of HAWatcher.

Abstract

ces are integrated via automation and coupled
ıysical environment, anomalies in an appified
, whether due to attacks or device malfunctions,
severe consequences. Prior works that utilize data
ıniques to detect anomalies suffer from high false
and missing many real anomalies. Our observa-
lata mining-based approaches miss a large chunk
on about automation programs (also called *smart*
levices. We propose *Home Automation Watcher*
r), a semantics-aware anomaly detection system
smart homes. HAWatcher models a smart home's
aviors based on both event logs and semantics.
me, HAWatcher generates hypothetical correla-
ling to semantic information, such as apps, device
ons and installation locations, and verifies them
logs. The mined correlations are refined using
s extracted from the installed smart apps. The
elations are used by a *Shadow Execution* engine
to simulate the smart home's normal behaviors. During run-
time, inconsistencies between devices' real-world states and
simulated states are reported as anomalies. We evaluate our
prototype on the SmartThings platform in four real-world
testbeds and test it against totally 62 different anomaly cases.
The results show that HAWatcher achieves high accuracy,
significantly outperforming prior approaches.

USENIX Security 2021

IoT 协议逆向分析

- 需求分析
  - 模糊测试的需要
  - 远程控制的需要
- 研究方向
  - 针对IoT特点的协议格式与状态机逆向分析方法

IoT协议逆向分析

- NDSS 2021

Abstract—Network protocol reverse engineering is an impor-
curity applications. A popular kind
k message traces. These methods
lignment and/or tokenization. They
as difficulties of handling a large
ng with inherent uncertainty. In this
obabilistic method for network trace
ering. It first makes use of multiple
all messages and then reduces the
yword field from the set of aligned
ermines the type of a message. The
using random variables to indicate
(being the true keyword). A joint
among the random variables and
ages. Probabilistic inference is then
most likely keyword field, which
ly clustered by their true types and
age format and state machine. Our
ows that our technique substantially
art and our case studies show the
technique in IoT protocol reverse
engineering and malware analysis.

TABLE V: Evaluation of format inference

Protocol	Netzob		Discoverer		NETPLIER	
	Corr.	Perf.	Corr.	Perf.	Corr.	Perf.
DHCP	0.089	0.000	0.768	0.016	0.994	0.014
DNP3	0.702	0.099	0.486	0.018	0.752	0.183
FTP	1.000	1.000	1.000	1.000	1.000	1.000
ICMP	0.571	0.144	0.259	0.102	0.972	0.090
Modbus	0.587	0.084	0.344	0.049	0.698	0.049
NTP	0.830	0.000	0.661	0.000	0.851	0.000
SMB	0.660	0.152	0.608	0.207	0.964	0.237
SMB2	0.349	0.003	0.793	0.041	0.923	0.069
TFTP	0.666	0.454	0.147	0.000	0.986	0.009
ZeroAccess	N/A	N/A	0.155	0.000	0.980	0.000

$$perfection = \frac{\text{Number of accurate fields}}{\text{Number of total true fields}}$$

IoT App分析

- 需求分析
  - 分析App（快速提取与设备有关的ID号、用户名等隐私信息，与设备、云端的通信协议，采用的加密和认证方法）来实现对固件的安全分析

IoT设备侦察

- 通过（长时间）接收智能家居、安防设备（电子门帘、智能门锁、摄像头、猫眼等）发出的无线信号（蓝牙、Zigbee、Wi-Fi等），分析目标环境中的智能设备厂家、类型、型号、告警信息、位置（房间）、通信联络图等
  - 基于加密流量的设备指纹识别技术
  - 基于无线信号的定位技术
  - 根据MAC地址反查设备厂商、出厂时间等信息

内容

一、物联网（IoT）及其安全问题

二、智能IoT设备安全研究

三、有关研究方法的几点思考

研究方法的几点思考

- 从哪里了解最新的研究？
  - 4大安全顶会：ACM CCS, NDSS, IEEE S&P, USENIX Security（这些顶会文章特别关注解决实际问题，实验非常充分，一般在GitHub提供代码和样本）
  - 软件工程顶会：ACM SIGSOFT ISSTA
  - 网络领域顶会：IEEE/IFIP DSN
  - 其它：RAID，IoT S&P， ACM WiSec，AsiaCCS
  - 跟踪顶级团队的研究工作进展

研究方法的几点思考

- 从哪里了解最新的研究？
  - 重要外文期刊：IEEE Security & Privacy; IEEE Internet of Things; ELSEVIER Computer & Security
  - 重要中文期刊：计算机学报、软件学报、网络与信息安全学报
  - 重要公众号、博客（腾讯https://sec.today/pulses/，看雪https://bbs.pediy.com/，安全牛https://www.aqniu.com/）

研究方法的几点思考

- 从哪里了解最新的研究？
  - 国内著名IoT安全实验室：腾讯科恩实验室（https://keenlab.tencent.com/zh/）、京东安全牧者实验室（The Shepherd Lab，https://qiling.io/，https://security.jd.com/）、奇安信技术研究院（https://research.qianxin.com/）、嘶吼胖猴信息安全研究所（https://www.4hou.com/）、小米安全中心（https://sec.xiaomi.com/#/）

研究方法的几点思考

- 从哪里了解最新的研究？
  - https://github.com/V33RU/IoTSecurity101
  - https://github.com/fkie-cad/awesome-embedded-and-iot-security

研究方法的几点思考

- 工作要做得扎实、充分
  - 问题来自于实践
  - 实验要充分（全面、深入、样本量足够）
  - 对比分析要到位，贡献是什么？
    - 如果是漏洞挖掘，则比的是提交了多少个CVE、多少个别人没发现的未知漏洞；如果是分析平台，则比的是通用性、功能及性能；如果是漏洞检测则比的是漏报率、误报率、正确率等；如果是安全评估，比的是发现了多少问题、有没有普遍性

研究方法的几点思考

- 要注重编程实现能力！
  - 良好的程序架构设计：模块化，可扩展性
  - 熟悉反汇编工具（如IDA Pro, Capstone，Ghidra, Cutter等）的插件编程方法
- 要注重多种方法的综合应用
  - 动静结合
  - 机器学习方法在安全中的应用
  - ……

研究方法的几点思考

- 广泛、快速阅读，勤于、善于思考
- 学术素养的重要性！

欢迎加入我的团队

- 招收软件与网络安全方向的硕士生、博士生、博士后

我们有充足的经费和机会支持学生从事网络安全研究，热忱欢迎你的加入！

电话：13951702346
邮箱：wulifa@njupt.edu.cn

、物联网系统安全性评估

感谢聆听！