WHAT IS IOT?

Bunch of devices hooked up to the internet, and these don’t usually don’t have security features and they have huge supply chains, with multiple product makers and people creating the thing.

The kinds of IOT

-> stuff that gets a lot of upgrades for a short while like your phone or teslas
-> stuff that once made barely get upgrades (old cars, thermostats, etc)

Monthly upgrades are great because you can roll out new updates instead of recalling all the cars to the factory as that would cost a lot more, a few billions. The issue being that the car needs to get upgrade patches all the time, not for a few years like phones but for longer times like decades. And these cars are made from smaller components, so now each component maker needs to maintain the products and upgrade, and then who assembles it, who pays for it, who is liable if something goes wrong. Android OEMs barely get upgraded, and so the court but a car is now something that can cause people to die, and the law states that there has to be someone liable should something go wrong.

TALK ABOUT SLIDE

Some stuff

WHAT MAKES IOT DEVICES MORE VULNERABLE?

PCs have security stuff like firewall and antivirus, but in IOT they are neglected as the companies want to make products cheap and fast and don't think security is really important, cuz what is a hacker gonna do with a hacked light bulb, flicker it on and off in the night to irritate the user?

Types of IOT Attacks

Communication Vulnerabilities

Attackers can try multiple means to intercept, spoof or disrupt messages sent from devices back to the server. Best-practice cryptographic defences must match the increasing value data being communicated.

Physical Vulnerabilities

Silicon attacks are often split into two categories: non-invasive and invasive. Non-invasive (side-channel) use different ways to try to observe the chip to gain information. These include perturbation techniques–altering the power supply voltage or interfering with electromagnetic signatures. Invasive techniques involve opening the chip to probe or modify part of the passivation layer.

Lifecycle Vulnerabilities

Devices change hands many times—from factory to user, to maintenance and to end-of-life. The integrity of the device must be protected at each step: who is repairing it, how is confidential data handled, are firmware upgrades legitimate. Unplanned or forbidden paths, such a theft, overages, or Wi-Fi changes are all vulnerabilities to consider.

Software Vulnerabilities

These are the most common attacks where someone finds a way of using existing cost to get access to restricted resources. It could be due to a software bug or to unexpected call sequences that are open to whole classes of exploits.

Software

Knowledge required is the same as an ES engineer
There's binary reversing, firmware analysis, bug finder (DB of all issues and spam at it to check if it can break) , if it’s running a web server then you can SQL injection and XSS and everything else in between

Hardware

Non-invasive:

Just the signals and stuff, here you can get the bootloader and login as root and now through the UART you can read and write. TIming attacks like secret data check, cache miss and hit time diff.

Side channels:

HW glitching by altering the voltage, power analysis here the device takes a diff power to compute a diff value like a string compare. EM radiation slike infrared caused by maybe infrared so now you know weak transistors and can pop out some gate and photons and now you know attack locations

Fully invasive attacks

Micro probing paths by changing the material on the chip, this takes time and money, only way to work around is by having hardware security when it was being built, software can't help here.

You can buy pre-made hardware attack tools like oscilloscopes, facedancer, hackRF, chipwhisperer.

Real world attacks

UART, it doesn't need to be populated for someone

And because they all run linux, you can boot ion with init=/bin/sh and you boot into as root and boom, or the NAND pins can shorted(pin 29 and 30) and it loads a corrupt bootloader in root mode

Bruteforcing is easy and data stored as base64 encoding which isn't an encryption.

Shell injections by going with ;reboot; in system("ls %s")

A few real world examples could be

Electronic safe lock, here you just need to follow the eeprom voltage drop and now you know the pwd. What to do? Don't store data on an eeprom or store the hash of it, as this doesn’t tell them how it is calculated.

Hardware

UART attacks

1. Talk to debug ports:

This thing’s been there since forever and are very simple with 4 straight pins of Tx Rx Gnd and something else and then you can oscilloscope it

2. Printer by epson has linux and the hardware has an open UART and now on connecting you get a menu driven with stuff like reboot or reset or run a shell command so well yeah

3. Belkin wemo - This was rampaged by people and since it’s a wall socket this guy can crash the power lines.

4. U Boot with no settings changed so you can change the bootloader and change the command kernel line, here you can do stuff like change the first program that’s run so we tell it to start a root shell before anything kicks in and now what use is all the security smh.

5. Timing based attacks:

Well UART “did patch” except they didn’t and you have 500 microseconds Time frame with in computing power in a lot and it does nothing

EMMC

1. Connect and modify storage and access OS
2. This is an SD card on a chip as SD cards have stuff like bit checking and parity checks this makes life easy.
3. Amazon fire TV - UART and EMMC
4. Mount the factory setting and then chmod 4755 and then you install like supersu or something and you're good.
5. LG smart fridge - well there's EMMC so you can install a stock android launcher and then gg

sys.fs is file containing shell codes that are randomly executed by the shell

SHELL INJECTIONS

("ls %s") to display some directory bu you can go ;root; and ls ;root; turn out that it executes the next command so you wanna patch these things up

1. Smart TV - smash the bash commands in to the wifi pwd
2. Blu-ray players - LG, Sony, and Panasonic they are made by the same company and sold to them, so now all these devices have the same issue, now you can telnet into them and whatnot.

These also have web support pages and this is another place where you can do the injections.
Now you can bruteforce and whatnot to get the pwd

Man in the Middle

Something secure like netgear that has signed and encrypted updates.
You can man in the middle change the update and send it through it

Baby monitor
Has a hardcoded username and pwd

Samsung smart cam
You can reset the admin’s pwd without logging in

Command injection bugs
Really popular

CRAPPY SOFTWARE DESIGNS

-> A smart lock, here you needed your fingerprint to unlock the device or you could use a mobile app connected via bluetooth, upon reverse engineering the app it was discovered that a key was required to unlock the lock, turns out the key to unlock the lock was just the bluetooth MAC address of the lock which is the only bit of information that the device constantly emits.

-> Smart wifi kettle, telnet connection and the hardware component (UART module) ran the default username and pwd, so now any device can telnet to the kettle and check the stored data and now you got the home PC’s wifi password

-> swearing doll (Cayla), anyone can connect to the doll and the doll has a mic and a speaker, so a random person could connect and listen and talk to your kids.

-> ransomware on a thermostat, now you can hack it and then you can control the geysers and stuff, well you could maybe shut down an entire city electric grid by just turning all the devices on at the same time as the issue isn’t “your IOT but everyone’s IOT”. Also imagine if your car got ransomware in the middle of a drive, not sounding good

Also remember during the Pewdiepie vs T series fight there were cases where printers were hijacked and security cams were hijacked where the hackers told the people to subscribe to a channel and show it at the camera and they would leave.

Miirai botnet -> 60 diff login and pwd list took down twitter and FB and trip advisor because people did not change the default credentials, this is a lot of compute power for crypto mining, hackers, imagine a DDOS attack from all the IOT devices, the mirai was about 655 GBPS at the traffic point, which is half a TBPS. There are like a dozen billion IOT stuff, also on PCs you need to bother about CnCs and stuff but iot is like free real estate for hackers

PSA SECURITY STANDARD

Unique Identification
To interact with a particular device, a unique identity should be assigned to the device and this identity should be attestable. This identity facilitates trusted interaction with the device for example, exchanging data and managing the device.

Security lifecycle
Devices should support a security lifecycle that depends upon software versions, run-time status, hardware configuration, status of debug ports and the product lifecycle phase. Each security state of the security lifecycle should be attestable and may impact access to the device

Attestation
Attestation is the evidence of the device’s properties, including the identity and lifecycle security state of the device. The device identification and attestation data should be part of a device verification process using a trusted third party.

Secure boot

To ensure only authorized software can be executed on a device, secure boot and secure loading processes are required. Unauthorized boot code should be detected and prevented. If the software cannot compromise the device, unauthorized software may be allowed.

Secure update

Secure updates are required in order to provide security or feature updates to devices. Only authentic and legitimate firmware should be updated on the device. Authentication, at the time of download, may be performed however, the execution of the update must be authorized via secure boot.

Anti-rollback

Preventing rollback to previous software versions is essential to ensure that previous versions of the code can’t be reinstated. Rollback should be possible for recovery purposes only when authorized.

Isolation

Isolation aims to prevent one service from compromising other services. This is done by isolating trusted services from one another, from less trusted services and from un-trusted services.

Interaction

Devices should support interaction over isolation boundaries to enable the isolated services to be functional. The interfaces must not allow the system to be compromised. It may be required to keep the data confidential. Interaction should be considered both within the device and between the device and the outside world.

Secure storage

To prevent private data being cloned or revealed outside the trusted service or device, it must be uniquely bound to them. Confidentiality and integrity of private data is typically achieved using keys, which themselves need to be bound to the device and service.

Cryptographic/trusted services
A minimal set of trusted services and cryptographic operations should be implemented as the building blocks of a trusted device. These should support critical functions including security lifecycle, isolation, secure storage, attestation, secure boot, secure loading and binding of data.

PROTECTION

1. Use the safer functions like fgets instead of gets
2. Check buffer bounds
3. Make them run in constant time even if the secret keyt char is failed then run longer
4. Whitelist commands instead of blacklist
5. Don't let user use system commands
6. Don't have hardcoded secrets and don't let hackers rollback upgrades.
7. If it's a web server then check SQL and XSS
8. Use diff network for IOTs and use hard tokens on all devices so you don't get all hacked at the same time
9. Remove unused shell libraries and interpreters
10. Ancient protocols like FTP and telnet to be disabled.
11. Nowadays modems stop coming with the default pwd of like root and admin, instead they come with some random long looking 16 digit pwds that are unique to the device.
12. Backdoors are supposed to be closed properly,
13. Encryption is something that must be done.

Rn iot is in a state where it's like running your PC without any security, not just a light or something. The S in IOT stands for security.

Sources

Gtvhackers
Android authority
Computerphile
PSAcertified
ARM
TEDxTalks