

IoT Pentesting 101

Puliya Session
Bridging gaps in IoT Pentesting

Mohammed Saqeeb Shariff

- Student of Mr. IoT
- IoT and Hardware Security Lead at SISA Information Security
- I like to break IoT gadgets and torture them with glitching assaults.
- Social Media



Mr-IoT

Veerababu Penugonda / Mr-IoT

Designation: IoT Pentester

Project: <https://github.com/IoT-PTv>

Work: v33raiot@hotmail.com

Twitter: <https://twitter.com/v33riot>

GitHub: <https://github.com/v33ru>

LinkedIn: <https://linkedin.com/in/veeraiot>

Web: <https://iotpentest.com>

Community: <https://iotsecurity101.org>

BugCrowd: https://bugcrowd.com/V33RU_Mr-IoT

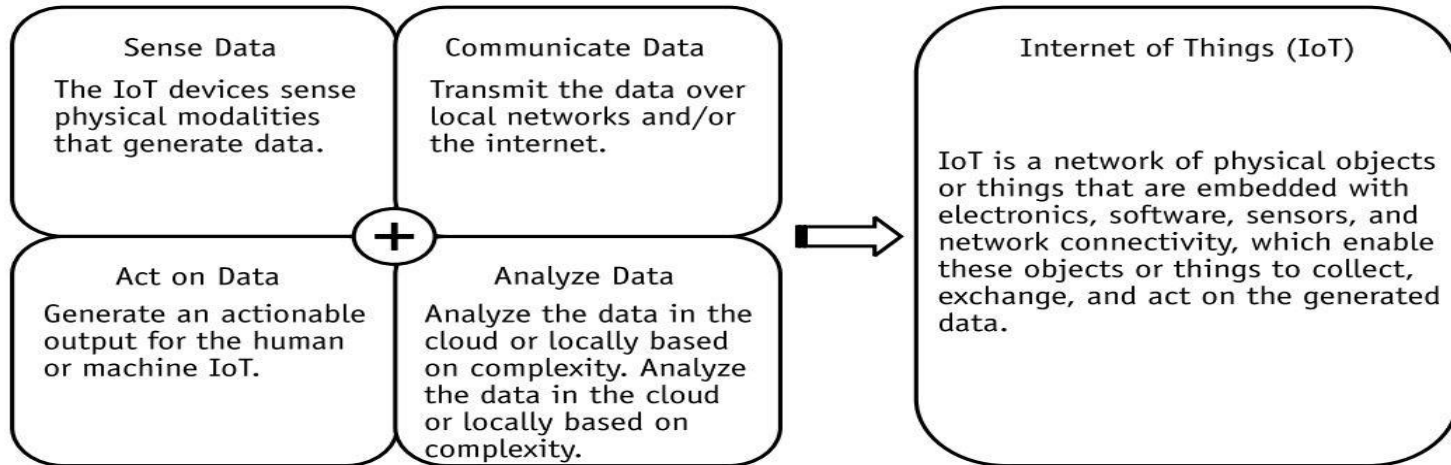
The Internet of Things is no longer a luxury item, but rather a necessity in today's environment.

Introduction to IoT and its security

1. Definition of IoT
2. IoT Architecture
3. IoT Key models of connectivity
4. IoT Attack surfaces
5. Why IoT Security is important?

Definition of IoT

- The definition we all acknowledge with IoT is any device which has the capability to route through the internet
- Below is the complete definition of IoT

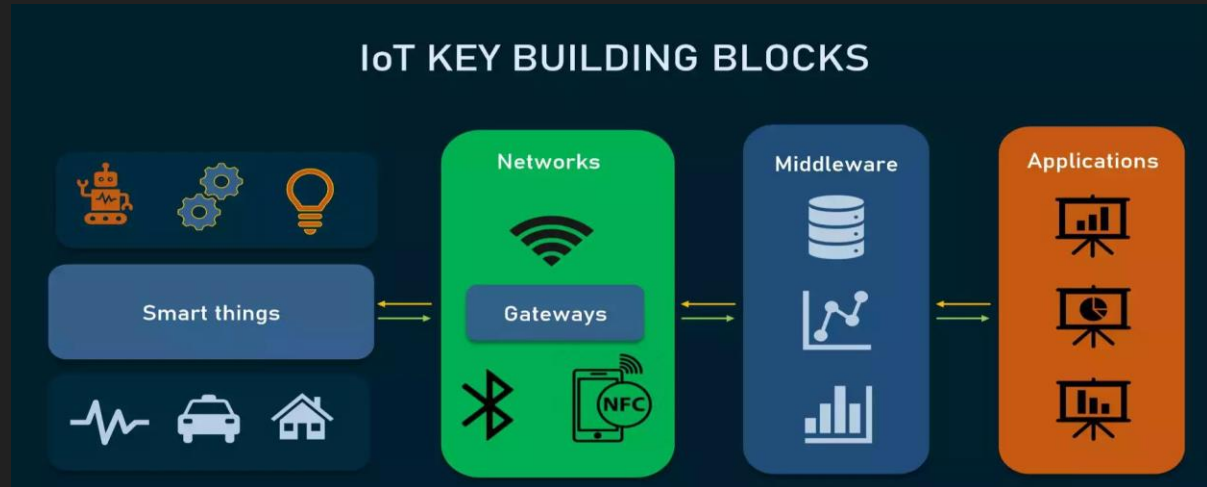


IoT Architecture

- IoT architecture refers to the tangle of components that comprise IoT networking systems such as sensors, actuators, cloud services, Protocols, and layers.
- The architecture of IoT is a four-step process through which data flows from devices connected to sensors, through a network, and then through the cloud for processing, analysis, and storage.
- However, there is no standard defined architecture of work that is strictly adhered to across the board. The complexity and number of architectural layers vary according to the specific business task at hand.

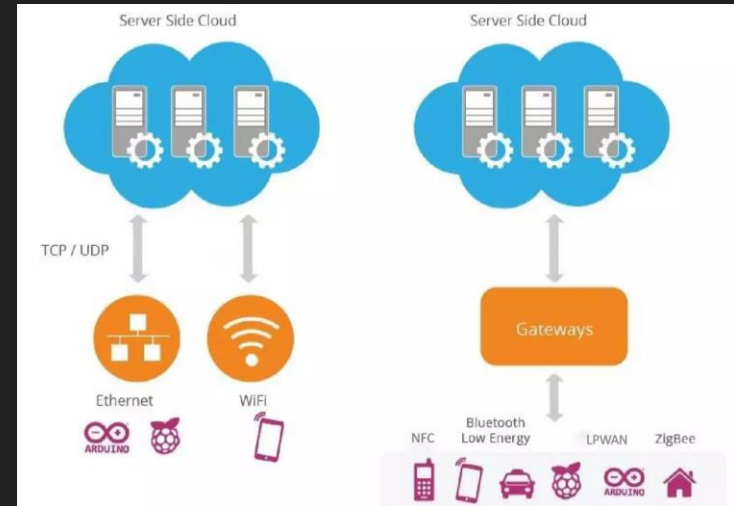
IoT Architecture - cntd.

- Sensing Layer - It acts as a link between the physical and digital worlds.
- Network Layer - Basically helps this device or the data sensed to route through the internet and communicate within the network.
- Middleware - The brain of the IoT ecosystem, where data is examined, pre-processed, and stored
- Applications - Remote user interaction



IoT Key models of connectivity

- Directly, using TCP or UDP/IP stack;
- Via gateways – hardware or software modules performing translation between different protocols as well as encryption and decryption of IoT data.



IoT Key models of connectivity - cntd.

NETWORKING TECHNOLOGIES USED in IoT				
Network	Connectivity	Pros and Cons	Popular use cases	
Ethernet	Wired, short-range	<ul style="list-style-type: none">⊕ High speed⊕ Security⊖ Range limited to wire length⊖ Limited mobility	Stationary IoT: video cameras, game consoles, fixed equipment	
WiFi	Wireless, short-range	<ul style="list-style-type: none">⊕ High speed⊕ Great compatibility⊖ Limited range⊖ High power consumption	Smart home, devices that can be easily recharged	
NFC	Wireless, ultra-short-range	<ul style="list-style-type: none">⊕ Reliability⊕ Low power consumption⊖ Limited range⊖ Lack of availability	Payment systems, smart home	
Bluetooth Low-Energy	Wireless, short-range	<ul style="list-style-type: none">⊕ High speed⊕ Low power consumption⊖ Limited range⊖ Low bandwidth	Small home devices, wearables, beacons	
LPWAN	Wireless, long-range	<ul style="list-style-type: none">⊕ Long range⊕ Low power consumption⊖ Low bandwidth⊖ High latency	Smart home, smart city, smart agriculture (field monitoring)	
ZigBee	Wireless, short-range	<ul style="list-style-type: none">⊕ Low power consumption⊕ Scalability⊖ Limited range⊖ Compliance issues	Home automation, healthcare and industrial sites	
Cellular networks	Wireless, long-range	<ul style="list-style-type: none">⊕ Nearly global coverage⊕ High speed⊕ Reliability⊖ High cost⊖ High power consumption	Drones sending video and images	

IoT Attack Surfaces

- **Network:** IoT devices can be attacked through the network, such as through man-in-the-middle attacks, SSLstripping or denial-of-service attacks. Example: Mirai Botnet, St. Jude Medical pacemaker hack
- **Devices:** The devices themselves can be vulnerable to physical tampering or hacking, including attacks on firmware and hardware. Example: Realtek Vulnerability Under Attack: Over 134 Million Attempts to Hack IoT Devices
- **Web interfaces:** IoT devices often have web-based interfaces that can be targeted by attackers, such as through cross-site scripting or SQL injection attacks. Example: Ring camera - cross-site scripting (XSS) flaw that it said could be weaponized as part of an attack chain to trick victims into installing a malicious app(2022)

IoT Attack Surfaces -cntd

- **Cloud services:** IoT devices often rely on cloud services for data storage and processing, making these services a potential target for attackers. Example: My personal attack on one of the device - bluetooth device
- **APIs:** IoT devices often have APIs that can be used by third-party applications, making them a potential attack surface if they are not properly secured.
- **User credentials:** Weak or easily guessable user credentials can provide a way for attackers to gain access to IoT devices.

Why IoT Security is important?

- **Privacy:** IoT devices often collect and store sensitive information, such as personal data or health information, which must be protected from unauthorized access and misuse.
- **Safety:** Some IoT devices, such as medical devices or industrial control systems, can have a direct impact on physical safety if they are compromised.
- **Business risks:** IoT security breaches can result in loss of revenue, damage to brand reputation, and increased costs for businesses.
- **National security:** IoT devices are increasingly being used in critical infrastructure and military applications, and a compromise of these systems could have serious consequences for national security.
- **Consumer trust:** Consumers must trust that their IoT devices are secure in order to fully embrace and utilize the technology.

Intro to IoT Security Standard (NIST-800)

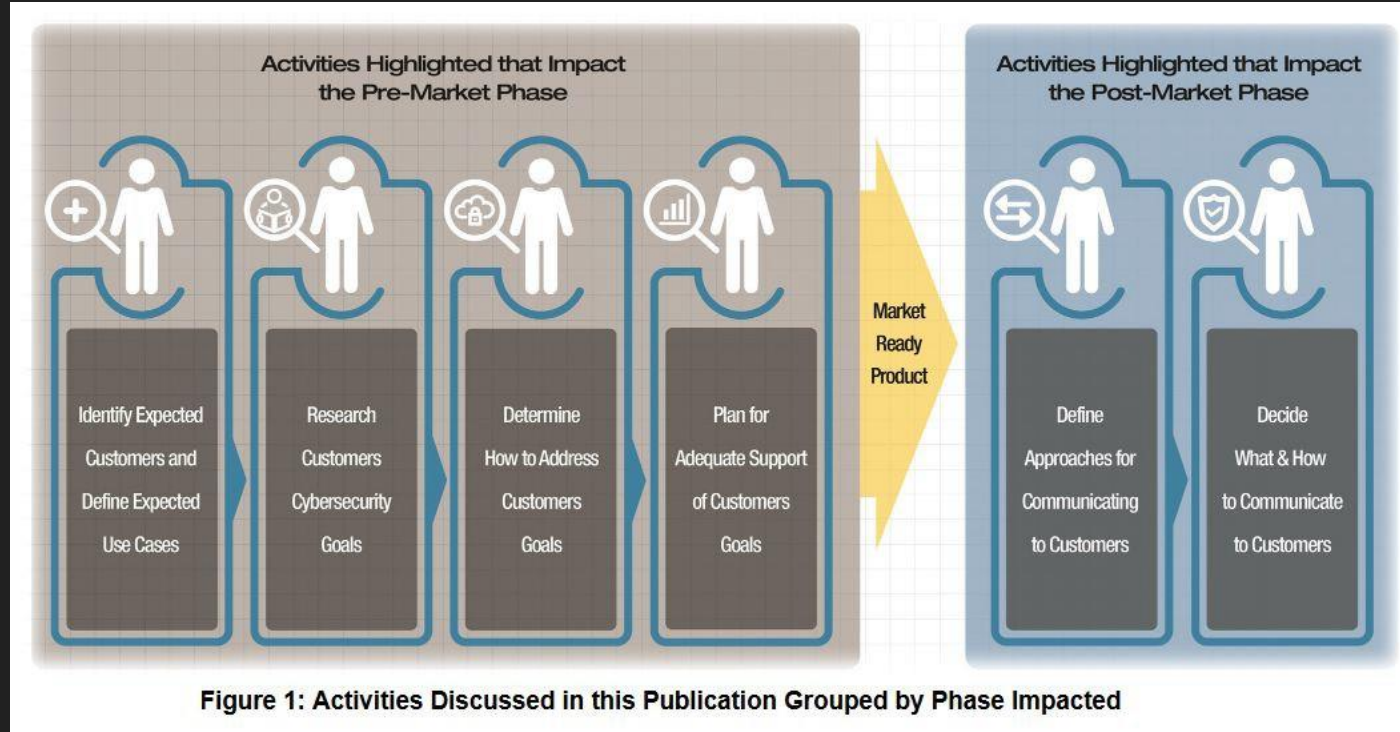
- NIST-8259 Explanation and other IoT Security Security standards
- Threat Modelling - STRIDE
- Example of STRIDE on IoT Device

NIST

The purpose of this publication is to give manufacturers recommendations for improving how securable the Internet of Things (IoT) devices they make are. We basically have three series in here

- NISTIR 8259 - Foundational Cybersecurity Activities for IoT Device Manufacturers
- NISTIR 8259A - IoT Device Cybersecurity Capability Core Baseline

NISTIR 8259



NISTIR 8259 - cntd.

Activity 1: Identify Expected Customers and Define Expected Use Cases

- Identifying the expected customers for an IoT device early in its design is vital for determining which device cybersecurity capabilities the device should implement and how it should implement them. For example, a large company might need a device to integrate with its log management servers, but a typical home customer would not

Activity 2: Research Customer Cybersecurity Needs and Goals

- Manufacturers cannot completely understand all of their customers' risks because every customer, system, and IoT device faces unique risks based on many factors. However, manufacturers can consider the expected use cases for their IoT devices, then make their devices at least minimally securable by customers who acquire and use them consistent with those use cases.

NISTIR 8259 - cntd.

Activity 3: Determine How to Address Customer Needs and Goals

- After researching the cybersecurity needs and goals for the IoT device's expected customers and use cases, manufacturers can determine how to address those needs and goals in order to help customers mitigate cybersecurity risks. For each cybersecurity need or goal, the manufacturer can answer this question: which one or more of the following is a suitable means (or combination of means) to achieve the need or goal?

Activity 4: Plan for Adequate Support of Customer Needs and Goals

- It is important for manufacturers to consider how to support their identified customers' needs and goals beyond the selection of specific device cybersecurity capabilities and their high-level implementations. This includes considering how to provision computing resources to support device cybersecurity capabilities and actions external to the device that may be required to continue to support cybersecurity needs and goals. For example, software-based encryption is processing-intensive, and a device with limited processing and no hardware-based encryption might not be able to provide what customers need

NISTIR 8259 - cntd.

Activity 5: Define Approaches for Communicating to Customers

- Clearly communicating cybersecurity information may necessitate different communication approaches for different kinds of customers based on their expectations and resources. For example, a home user will likely have less technical knowledge than points of contact at a large business (e.g., system administrators)







Activity 6: Decide What to Communicate to Customers and How to Communicate It

- To understand how their risks might differ from the manufacturer's expectations, some customers may benefit by knowing the cybersecurity-related assumptions the manufacturer made when designing and developing the device. For example, some IoT devices have specific intended purposes in systems, which may drive cybersecurity considerations for customers e.g., a device requires a monitoring system to be able to connect to it for cybersecurity purposes

NISTIR 8259A

- This publication defines a baseline set of device cybersecurity capabilities that organizations should consider when confronting the challenge of the Internet of Things (IoT). Device cybersecurity capabilities are cybersecurity features or functions that computing devices provide through their own technical means (i.e., device hardware and software). The IoT device cybersecurity capability core baseline (core baseline) defined in this publication is a set of device capabilities generally needed to support commonly used cybersecurity controls that protect devices as well as device data, systems, and ecosystems.

NISTIR 8259A - cntd.

Capability		Capability Description
	Device Identification	The IoT device can be uniquely identified logically and physically
	Device Configuration	The configuration of the IoT device's software can be changed, and such changes can be performed by authorized entities only
	Data Protection	The IoT device can protect the data it stores and transmits from unauthorized access and modification.
	Logical Access to Interfaces	The IoT device can restrict logical access to its local and network interfaces, and the protocols and services used by those interfaces, to authorized entities only.
	Software Update	The IoT device's software can be updated by authorized entities only using a secure and configurable mechanism.
	Cybersecurity State Awareness	The IoT device can report on its cybersecurity state and make that information accessible to authorized entities only.

Other Security Standards and Regulations

1. ISO/IEC 27402
2. Singapore CLS Scheme
3. California Internet of Things (IoT) Cybersecurity Improvement Act
4. UAE IoT Policy - Regulation

Threat Modelling 101

- Attack surfaces(entry points) refer to the many ways in which a device can be compromised via a source of input(hardware,software, wireless).
- Each attack surface discovered will have an **associated risk, likelihood, and impact**
- Attack surfaces are threats which have the potential to negatively affect a device to perform unintended actions.
- In order to discover each attack surface, **theoretical use cases will need to be thought of** before testing has taken place
- It's a process between design phase and prior to deployment
- Identified threats have to be rated based on risk - **CVSS** and **DREAD**

DREAD 101

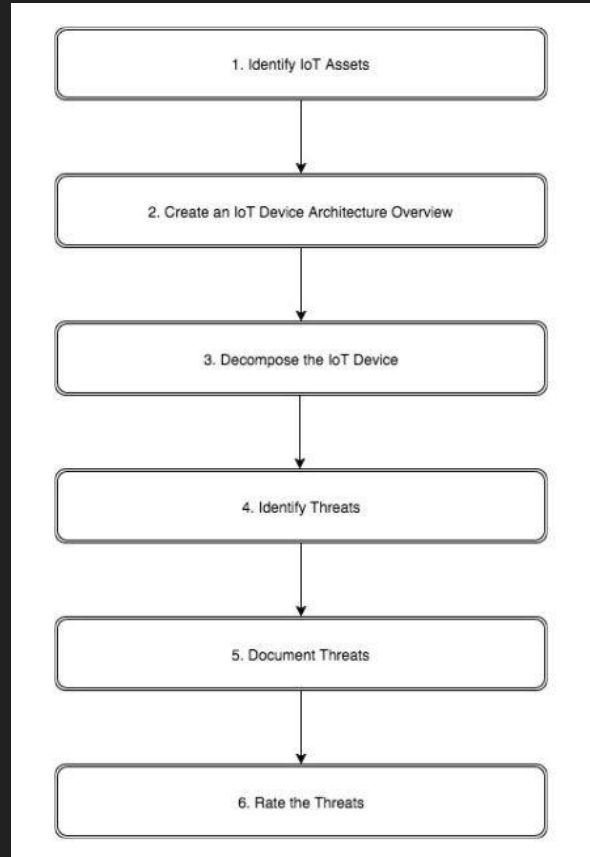
- Damage potential: How great is the damage if exploited?
- Reproducibility: How easy is it to reproduce the attack?
- Exploitability: How easy is it to attack?
- Affected users: Roughly how many users are affected?
- Discoverability: How easy is it to find the vulnerability?
- DREAD has a risk rating system ranging from 1-3.
 - 1 is low risk
 - 2 is medium risk
 - 3 is high risk.

Risk rating	Result
High	12-15
Medium	8-11
Low	5-7

Stride Methodology

Domain	My actions are in this domain if I can...
Spoofing	Pose as someone/something that I am not.
Tampering	Change something I am not supposed to.
Repudiation	Do something without the owner being able to prove I did it.
Information disclosure	Get access to data or information while I am not supposed to.
Denial of service	Keep it from working (via continuous interaction or a one-shot operation).
Escalation of privileges	Do things I am not supposed to without the privileges they normally require.

Steps Process



Steps Involved

Step 1: Identify the IoT assets

- This could be various different components that will be involved like as we discussed in our attack surfaces

Step 2 - creating an IoT device architecture overview

- Document the functionalities and features
- Create an architectural diagram
- Identify the tech used in the IoT Device

Steps Involved

Step 3 - decomposing the IoT device

- Analyze the application and protocol data flows through the device environment to locate vulnerable entry points into the device.
- We will look for locations that may have higher privilege access and document each possible entry point

Step 4 - identifying threats

- We now have to identify the **risks** of each entry point as it relates to the user, the network, and the application as well, as the vendors who wrote the application code
- To help with identifying threats and categorizing them, let's apply the STRIDE model t

Steps Involved

Step 5 - documenting threats

- we will document a few of the threat use cases we have identified in step 4 with a description, threat target, attack technique(s), and any countermeasures that may be in place

Step 6 - rating the threats

- Rate the threats with their likelihood as well as their possible impact

Step 1: Creating an architecture overview

- UART
- JTAG
- Flash Chips
- USB ports
- Power ports
- Bluetooth/BLE
- HTTP/HTTPS
- NFC
- Actuator
- Ethernet port

Step 2: Identifying threats

- Gain access to the consoles via UART
- Debug, manipulate the register value and Dump/reupload Firmware using jtag
- Dump secrets with flash
- Exploit a flaw in the USB stack
- Bypassing security checks, running unverified code through glitching - cve-2019-15894
- Performing un-authenticated and un-authorized attacks on the ble based devices by writing to its characteristics
- MiTM transmitted data

Step 3: Document Threats

Threat Description	Attacker could gain access to the consoles via UART
Threat target	UART
Attack Techniques	Attack UART headers on the PCB board of the device.
Countermeasures	UART access is password protected, UART access is blocked by another chip, baudrate is anonymize.
Threat Description	Debug, manipulate the register value and Dump/reupload Firmware using jtag
Threat target	JTAG
Attack Techniques	Attack JTAG pins on the hardware
Countermeasures	Jtaglock, JTAG pins should be removed

Step 3: Document Threats - cntd

Threat Description	Dump secrets with flash
Threat target	EEPROM
Attack Techniques	Attacker attaches an SOIC clip on top of the EEPROM to read its contents.
Countermeasures	Prevent storage of sensitive data within the EEPROM
Threat Description	Exploit a flaw in the USB stack
Threat target	USB
Attack Techniques	Attack USB port on the hardware
Countermeasures	USB port blocker, disconnecting the data lines

Step 3: Document Threats - cntd

Threat Description	Bypassing security checks, running unverified code
Threat target	Power ports
Attack Techniques	Attacker provides arbitrary voltage changes to the device to behave the normal cycle.
Countermeasures	Circuit protection measures such as voltage clamping and over-current protection, multiple checks in place
Threat Description	Writing to bluetooth characteristics
Threat target	Bluetooth/BLE
Attack Techniques	Writing data to GATT characteristics
Countermeasures	Implementing strong authentication mechanisms, such as a password or PIN, Encryption

Step 4: Rating the threats

Threat Description	Attacker could gain access to the consoles via UART
Damage Potential 3	3
Reproducibility	3
Exploitability	2
Affected Users 1	1
Risk rating score: High	11

Intro to Hardware hacking tools and related software

1. Fault Injection 101

2. Hardware Attacks of the past

- a. Secure boot bypassing with glitching
- b. Trezo wallet - retrieving 2 million dollars
- c. Breaking AES enc with glitching attacks -

3. Tools Repo link -

<https://github.com/IoTSecurity101/IoT-Lab-Setup-Guide>

Fault Injection 101

```
...  
if( key_is_correct ) <-- Glitch here!  
{  
    open_door();  
}  
else  
{  
    keep_door_closed();  
}  
...
```



clock



voltage



e-magnetic



laser

CVE-2019-15894

- Espressif ESP32: Bypassing Secure Boot
- An attacker can bypass the Secure Boot verification at the startup of the ESP32
- CPU via fault injection.
- Leads to the execution of unverified code from flash.
- It was identified that if the flash encryption is enabled, this attack could be mitigated. Hence the flash encryption was then later implemented.

CVE-2020-15048

- Espressif ESP32: Bypassing Flash Encryption
- An attacker can bypass the Secure Boot and Flash Encryption physical security features of the ESP32 CPU via fault injection.
- Leads to the execution of unverified code despite both the secure boot and flash encryption implementation.

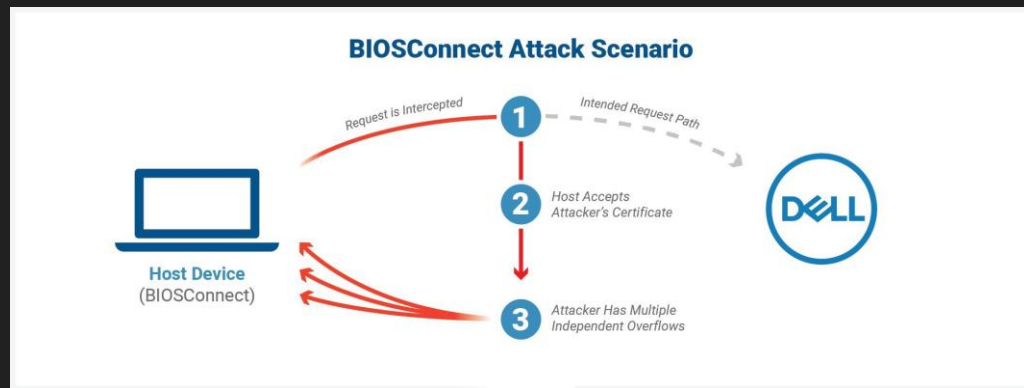
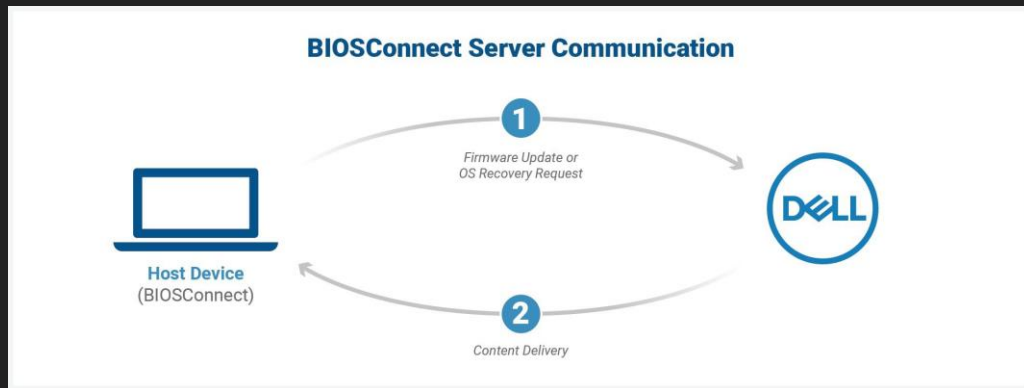
CVE-2020-15808

- STM32 USB Device Library: Buffer overflow vulnerability exploit
- The buffer overflow vulnerability in the CDC communication interface code exploited using fault injection.
- Leading to the access of sensitive information, keys, obtaining firmware, etc., with the properly crafted USB packets.

CVE-2022-31224

- Dell BIOS versions contain an Improper Protection Against Voltage and Clock Glitches vulnerability.
- An attacker with physical access to the system could potentially exploit this vulnerability by triggering a fault condition in order to change the behavior of the system.
- There are neither technical details nor an exploit publicly available

Dell code execution at the BIOS/UEFI



Dell code execution at the BIOS/UEFI - cntd

- Insecure TLS Connection from BIOS to Dell - CVE-2021-21571. When attempting to connect to the backend Dell HTTP server, the TLS connection from BIOSConnect will accept any valid wildcard certificate. This allows an attacker with a privileged network position to impersonate Dell and deliver attacker-controlled content back to the victim device.
- Vulnerable HTTPS Boot configurations - Some HTTPS Boot configurations may also be exploitable due to using the same underlying verification code. When configuring HTTPS Boot, the user is required to provision a Certificate Authority (CA) certificate, which is intended to be used to verify connections to the remote boot server before allowing the HTTP Boot process to proceed. However, when this verification is performed, any valid certificate for any domain acquired from the same CA will be accepted, not just those for the configured remote boot server.

Bypassing Secure Boot Using Fault Injection

```
...  
if( key_is_correct ) <-- Glitch here!  
{  
    open_door();  
}  
else  
{  
    keep_door_closed();  
}  
...
```

Understanding Application level testing in IoT Devices

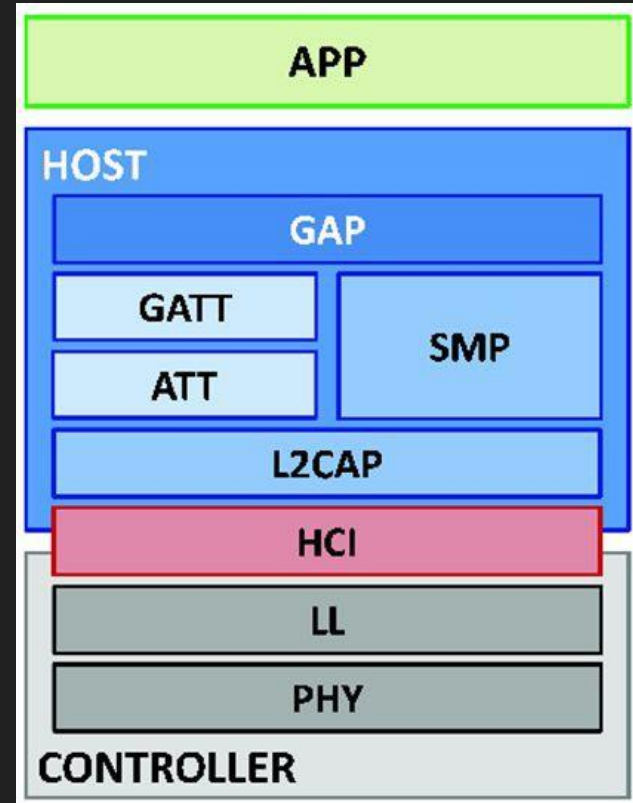
- Business logic functions are same
- Except major things what we need to find is SQL injection instead of command injection

Intro to various Radio Protocols and tools
used for attacking

- Various Radio protocols
- Mostly widely used protocols
 - Zigbee
 - Bluetooth

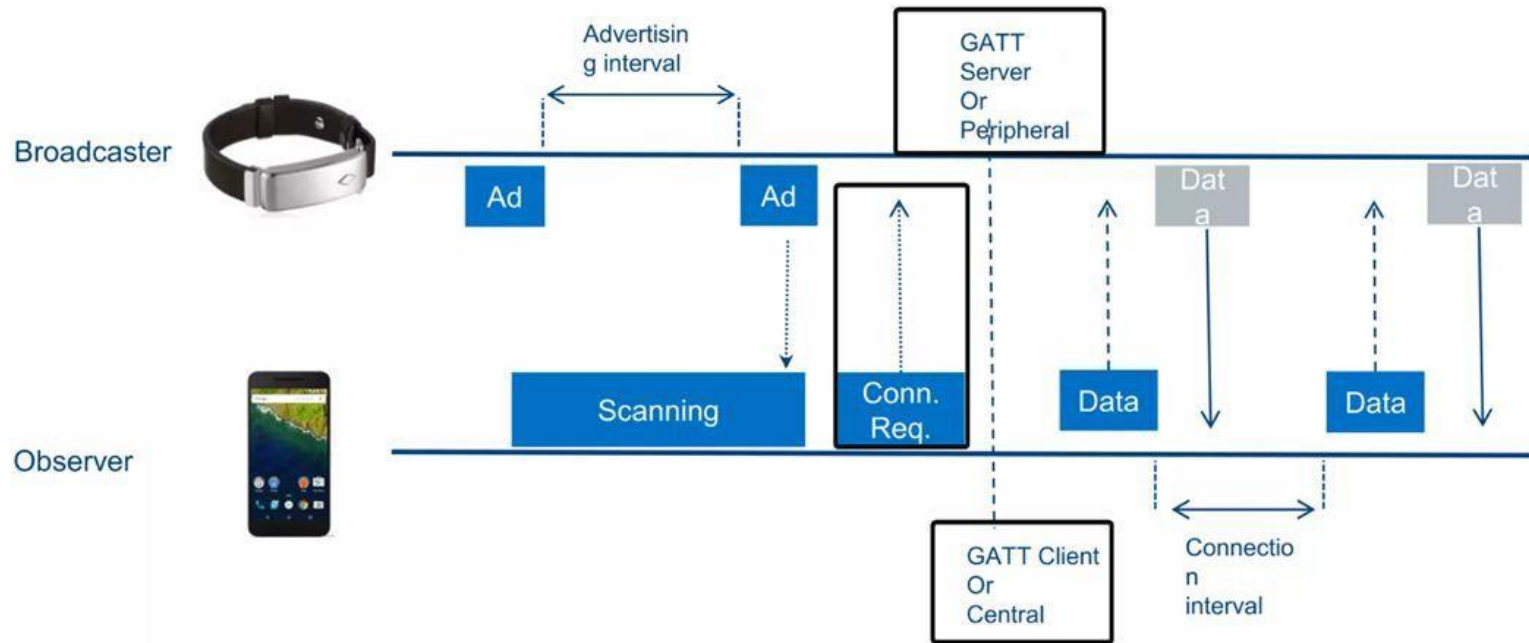
BLE Stack

- Generic Access Profile (GAP) – defines how devices discover, connect and create bonding between them.
- Generic Attribute Profile (GATT) – Describes characteristics, services and type of attributes (their usage)
- Security Manager Protocol (SMP) – protocol for pairing, key distribution and authenticating other devices.
- Host Controller Interface (HCI) – Software interface to the hardware

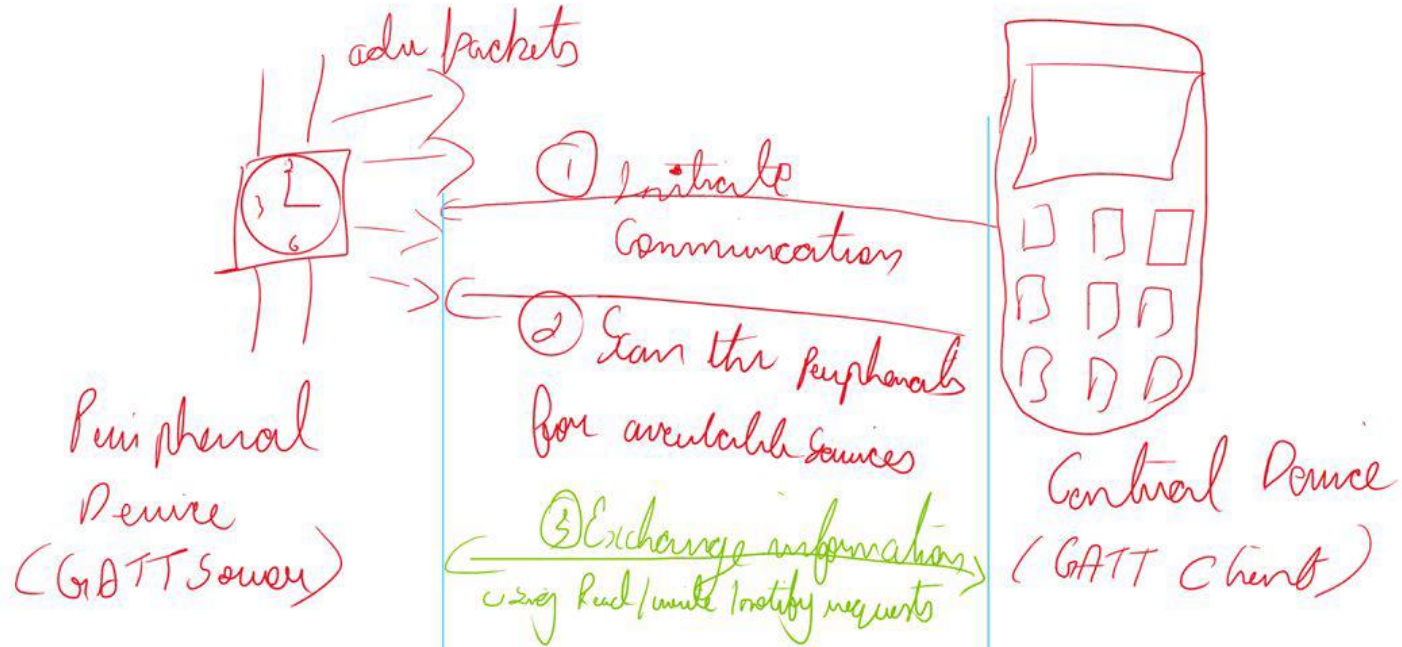


BLE Communication

How it works?

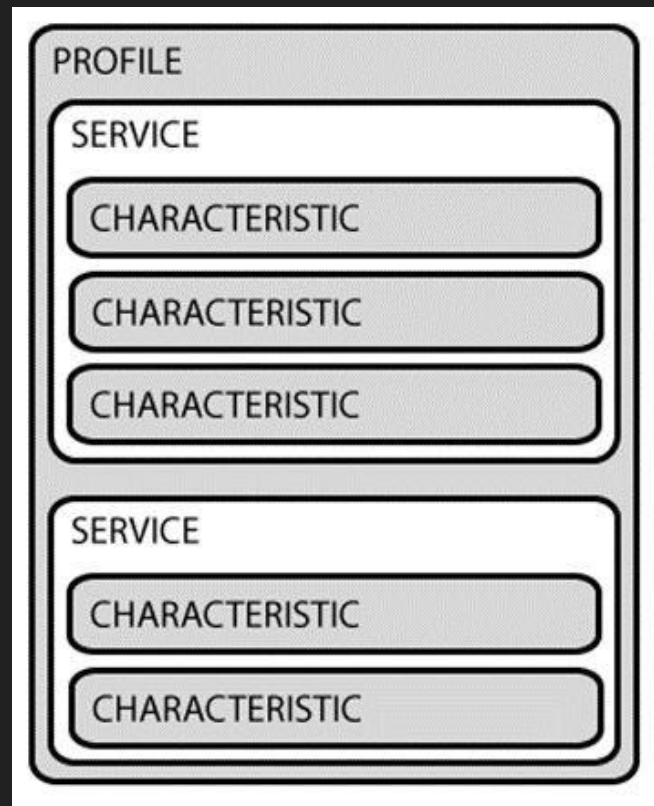


My Understanding



GATT

- GATT defines the way two BLE devices transfer the data back and forth using services and characteristics.
- Services are used to break data up into logical entities, and contain specific chunks of data called characteristics.
- Characteristics are the main point that we will interact with your BLE peripheral.
- Each characteristic is given a UUID that's generally 16 bit with reference to SIG
- Characteristics have three permissions that would be either READ, WRITE, NOTIFY.
- As an example, the Heart Rate Measurement characteristic is mandatory for the Heart Rate Service, and uses a UUID of 0x2A37.



Surprise!

Firmware

Hands-on IoT Firmware analysis

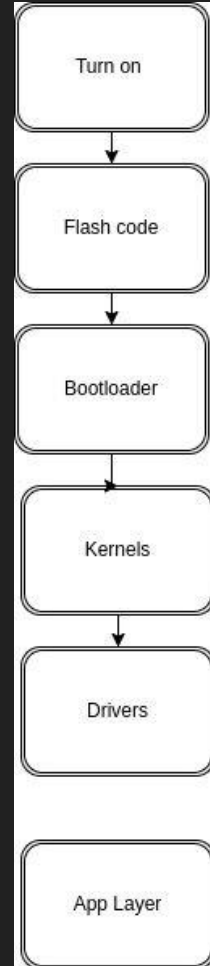
- Firmwares intro
- Firmware architectures
- Understanding firmware architectures
- Filesystems and internal architectures
- Analysis with Binwalk
- Emulation with FAT

Intro firmware's

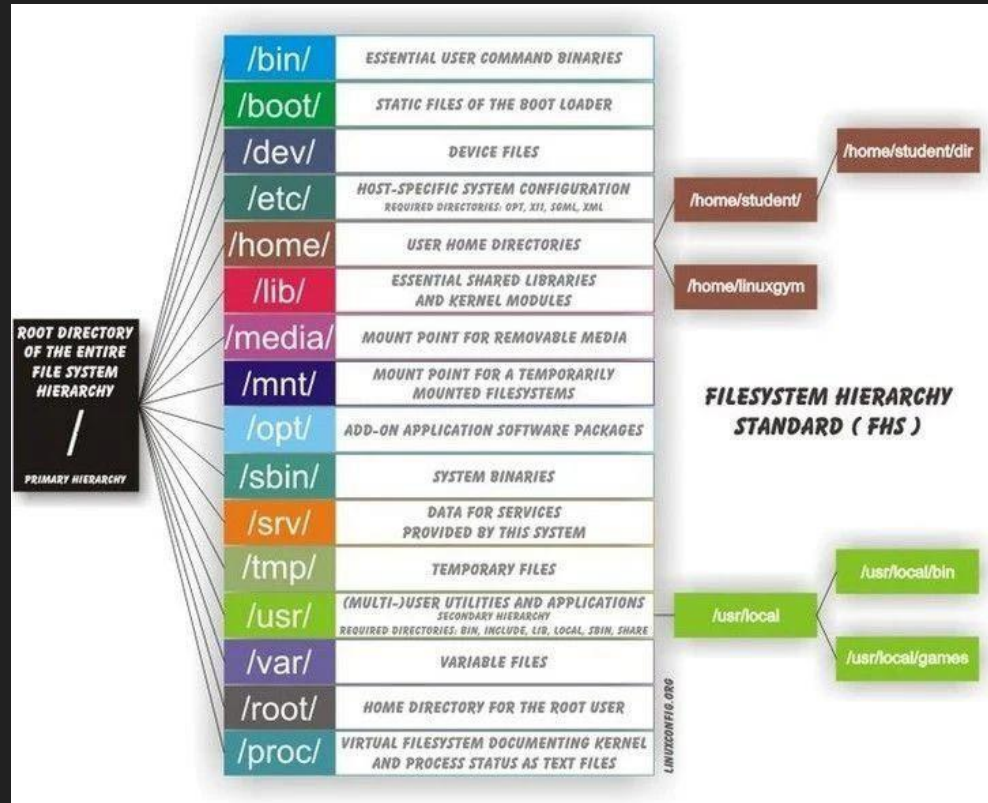
Basically it is OS of the device

- Firmware is a type of low-level software.
- Controls and operates the hardware of a device.
- Provides interface between hardware and operating system.
- Stored in non-volatile memory on the device.
- Responsible for basic functions like power management, booting and low-level device management tasks

Understanding firmware architecture



Filesystems and internal architectures



Types of Firmwares

1. Bare metal firmware - where directly interact with hardware mostly in less size files - not easy test them
2. File System / RTOS or custom firmware building
3. Fully Developed Firmware , mostly in our Big Machines

Firmware Vulnerabilities

1. Hardcoded data (password, hidden parameters, cloud , private keys)
2. Lack of encryption (Leads to data read in plaintext)
3. Non-signed firmwares (firmware recompiles , firmware backdoors)
4. Obfuscating code (use pseudo code readers)
5. Check for unsafe functions (malloc and calloc)
6. Check for DOS/CRASH/memory overflow attack based functions (scanf, alloca, strcpy, calloc, malloc)
7. Check for Command injection functions

Analysis with Binwalk

It's kind of cheat sheet

```
Sudo apt-get install  
binwalk
```

Dependencies

Vulnerability

Binwalk

-Y

Identifying
Architectures

1. Endian
2. Architecture

Binwalk

1. opcodes
2. Architecture

Binwalk

-B , --entropy

Binwalk

Extraction

E

Re

Mre

DD tool

Count , is
crazy is
enough play
with it

STRINGS

Use along with
GREP

Firmwalker

Firmwalker.sh

Automated
scanners
extracted
folders -
always best
one

Finishing subjects are coming

Build your own lab for practice..!

<https://github.com/IoTSecurity101/IoT-Lab-Setup-Guide>

<https://github.com/IoT-PTv/IoT-Security101-Kit-Roughly-costing-110USD>

<https://www.iotpentest.com/2021/02/how-i-purchase-costly-smart-devices-in.html>

Bug Bounty For IoT devices

Hackerone:

https://hackerone.com/directory/programs?asset_type=HARDWARE&order_direction=DESC&order_field=resolved_report_count

BugCrowd:

[https://bugcrowd.com/programs?sort\[\]=promoted-desc&target_category\[\]=iot](https://bugcrowd.com/programs?sort[]=promoted-desc&target_category[]=iot)