

Information Security

CS 3002

Dr. Haroon Mahmood
Assistant Professor
NUCES Lahore

Preface

- **Use Case: IoT security**
- **Components of an IoT system**
- **Vulnerability assessment**
- **Security Audit of IoT system**
- **Static and Dynamic Analysis**
- **Network security Assessment**
- **Hardware Security**

Internet of Things

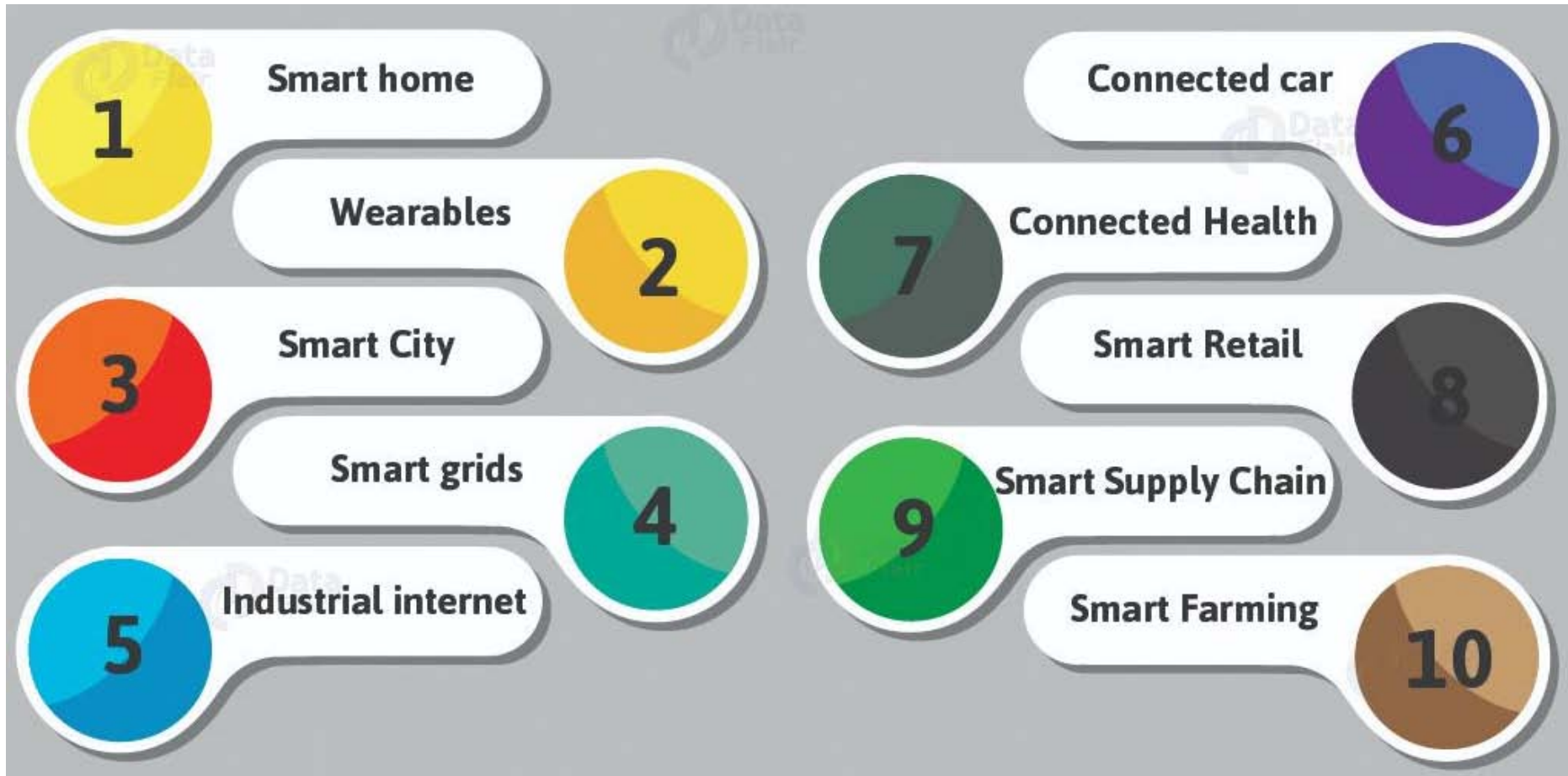
CONNECTED DEVICES ?

SMART DEVICES?

SMART CONNECTED DEVICES ?

- **Internet of things** is an ecosystem that is evolving by each passing day with
- **powerful devices**
- having **processing capability** (embedded, local server, cloud)
- connected with **internet** and accessible from anywhere
- making **Machine to Machine communication**
- with or without human intervention

IoT Applications



Source: data-flair

Security of IoT systems?

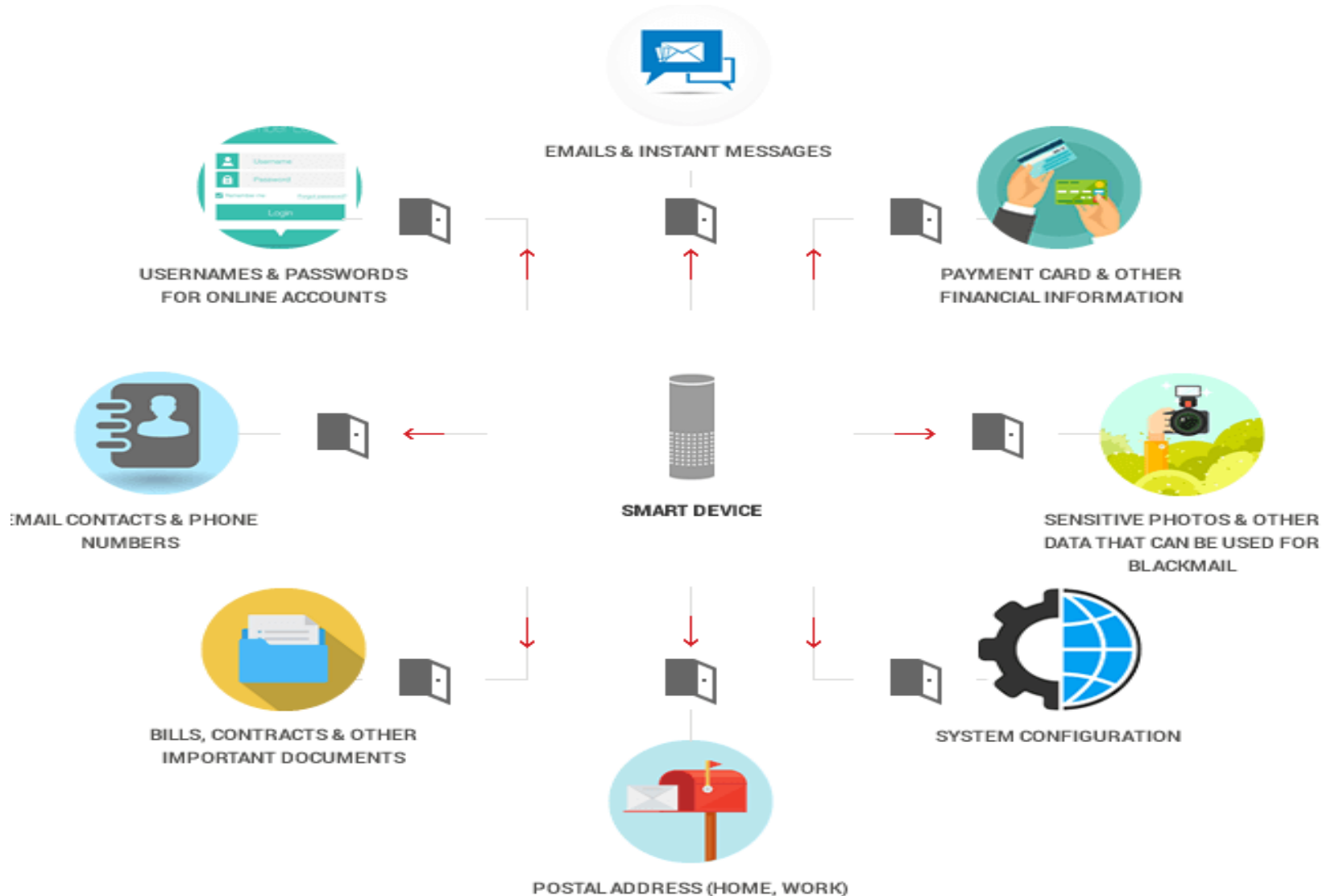
- A tech analyst company named IDC predicted that there will be 41.6 billion connected IoT devices by 2025 [1].
- However, a survey conducted by EIU (Economics Intelligence Unit) shows that about 74% of IoT consumers worry about losing their civil rights [2].
- Gap between people's privacy expectations and the posture of IoT security needs to be filled to achieve full swing in terms of growth.
- The most critical and vital part of an IoT system that cannot be overlooked at any cost is its security.



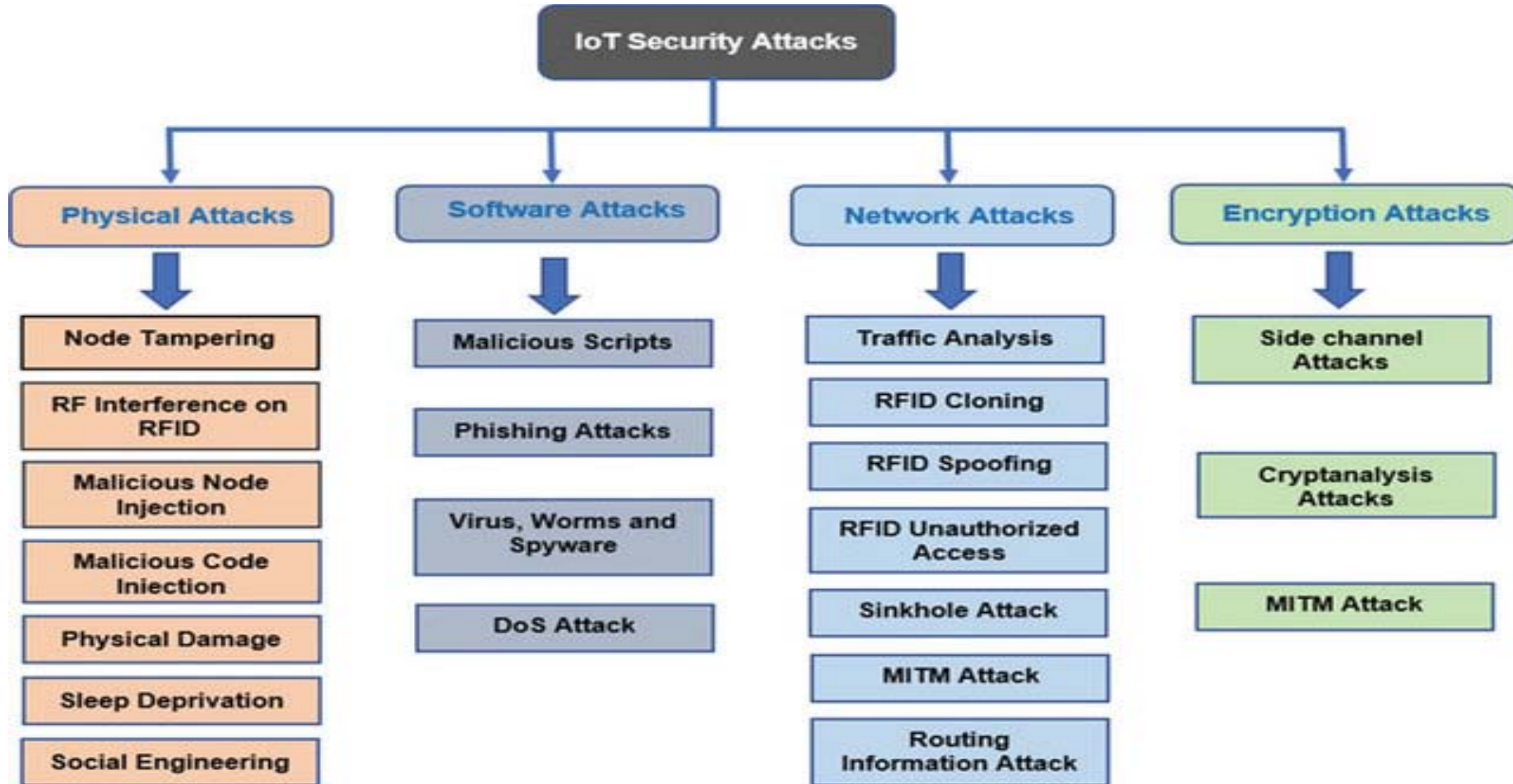
[1] IDC, "IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC." <https://www.idc.com/getdoc.jsp?containerId=prAP46737220> (available online)

[2] V. Lara, "What the "Internet of Things" Means for consumer privacy," Gigaom Research, p. 14, 2018

Dangers of IoT devices

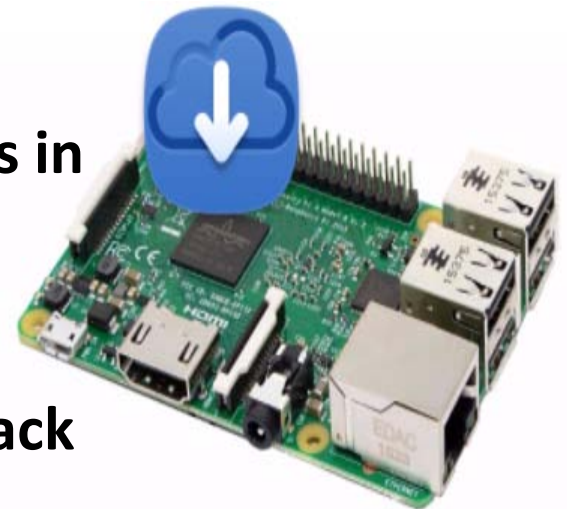


IoT security Attacks



IoT components

- **Three main components of an IoT device.**
 - Hardware
 - Firmware/Software and applications
 - Radio communication
- **All three can be used as attack surfaces in IoT devices**
- **Web interface (if present) is a vital attack surface.**



Common vulnerabilities of each component

- **Hardware**
 - Exposed serial ports
 - Firmware dumping over JTAG or via Flash chips
 - Side channel-based attacks
- **Firmware**
 - Backdoors
 - Sensitive data: passwords, keys, certificates, URLs
 - Coding errors (buffer overflow)
 - Default, weak, and hardcoded credentials
- **Communication**
 - Clear text protocols and unnecessary open ports
 - No/weak encryption
 - Default, weak, and hardcoded credentials

Acquiring data using hardware Exploitation

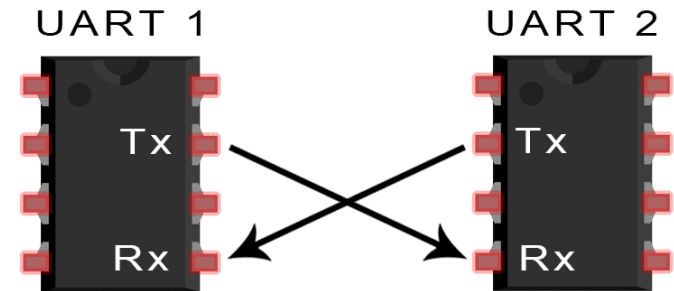
- **Identification of communication protocol/interfaces**
 - UART
 - JTAG
 - SPI
 - I2C
- **Use of specific tools to communicate with the target device**
- **Collection or modification of information**

IoT device hardware exploitation

- **UART Exploitation**
 - In first phase we perform UART pin identification
 - Then Identification of baud rate is performed for intercepting communication
 - Then we use TTL to USB converter to monitor traffic on UART interface
 - We can get access to the root of device and perform firmware exploitation

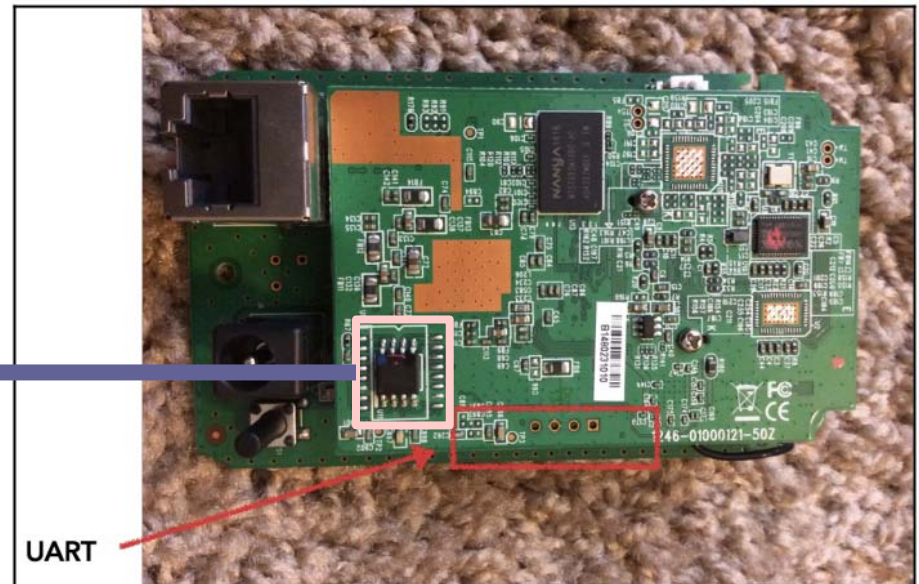
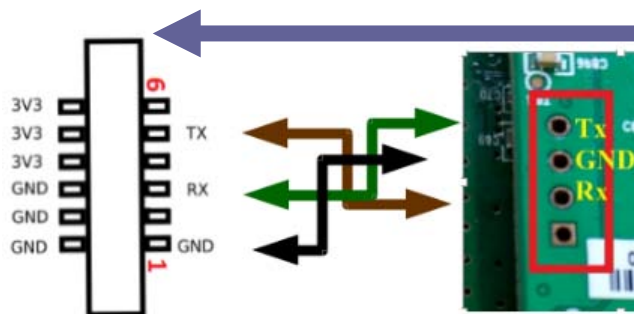
UART Exploitation

- **UART or Universal Asynchronous Receiver Transmitter is a dedicated hardware associated with serial communication of the device.**
- **UART Pins Identification?**



UART Pin Identification

- UART Pins
 - Tx
 - Rx
 - Power
 - Ground
- Can be identified using Multimeter

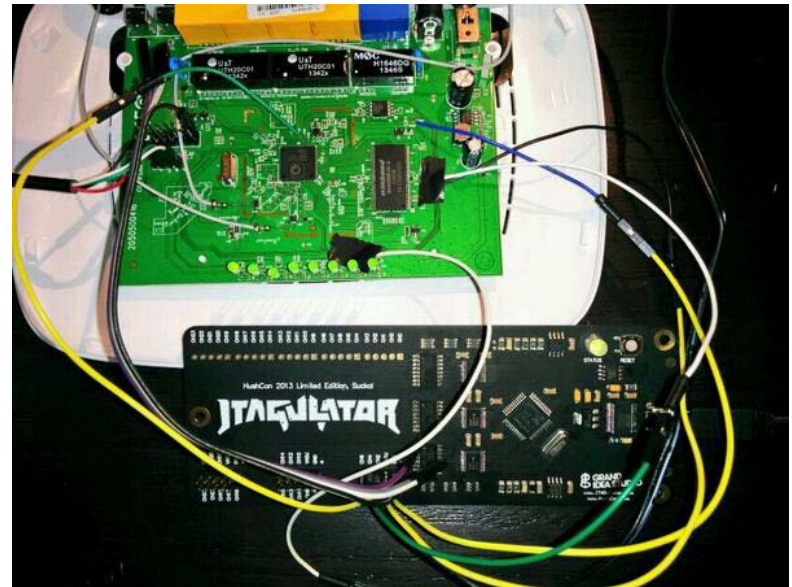


JTAG Exploitation

- **Used by device developers and testers to ensure that each of the pins on the chip are functional, interconnected, and operational as intended.**
- **For penetration testers, JTAG serves a number of purposes; ranging from giving the ability to read/write data, debug running processes and modifying the program execution flow.**
- **The four most important pins of JTAG are test data in (TDI), test data out (TDO), test clock (TCK), and test mode select (TMS).**

JTAG Pins Identification

- JTAG comes in multiple interface options such as 13 pins, 14 pins or 20 pins.
- Devices like JTAGulator can easily identify JTAG pinouts, and specific functionality or role of each pin.



IoT device hardware exploitation

- **SPI Exploitation**
 - This technique is useful in dumping firmware from SPI Flash memory of IoT device.
 - Locating SPI flash IC of an IoT device.
 - Attach CH341a programmer with SPI flash by using clip.
 - Functions we perform on a SPI Flash firmware
 - Read
 - Write
 - Delete
 - Upload
 - Save



IoT Firmware

- **Any operation performed by a computer requires code execution.**
 - **Printing “Hello world”**
 - **Peripheral checkup**
 - **Transfer of data to printer for printing**
- **Firmware is the software component of a computer system that talks to the underlying hardware.**
- **Bugs in the firmware can allow direct access to the hardware.**
- **Firmware in IoT versus that of a classical computer system differ greatly.**

Security Issues in IoT Firmware

- **Memory corruption**
- **Outdated (core) components**
- **Hard-coded credentials**
- **Faulty update mechanisms**
- **Authentication issues**
- **Non-compliance**
- **Vulnerable interfaces**
- **Vulnerable network communications**
- **Misconfigurations**

IoT Firmware Vulnerabilities

Ibrahim Nadir, Haroon Mahmood, Ghalib Asadullah, “**A taxonomy of IoT firmware security and principal firmware analysis techniques**”, International Journal of Critical Infrastructure Protection, Volume 38, 2022,

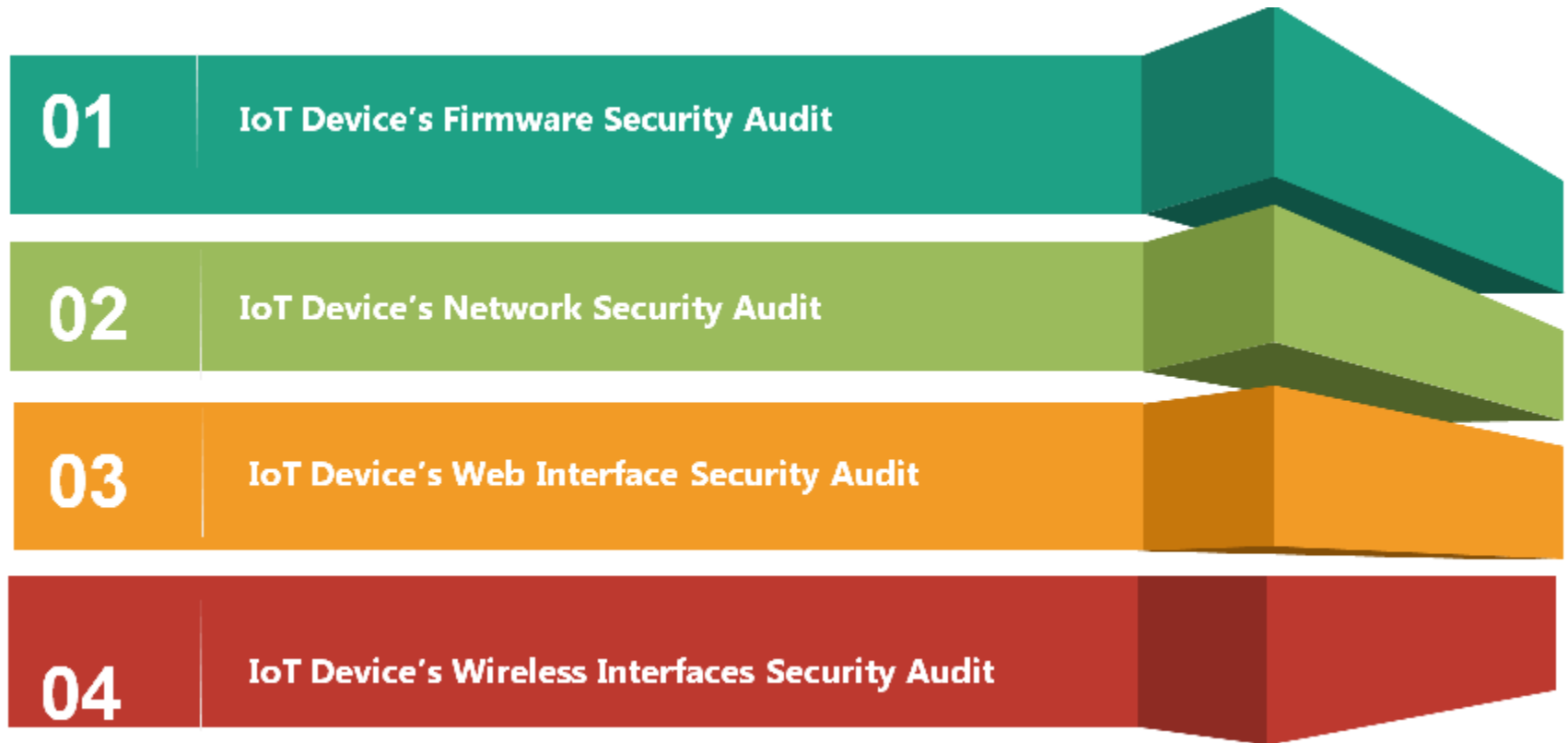
Existing Solutions

- **Static Analysis**
- **Dynamic Analysis**
- **Fuzzing**
- **Hybrid solutions**

Issues with existing solutions

- **Based on either static or dynamic analysis but a complete framework is missing**
- **Not expandable, can not be integrated with other security solutions**
- **Lesser support for multiple platforms and architectures**

Auditing of an IoT system



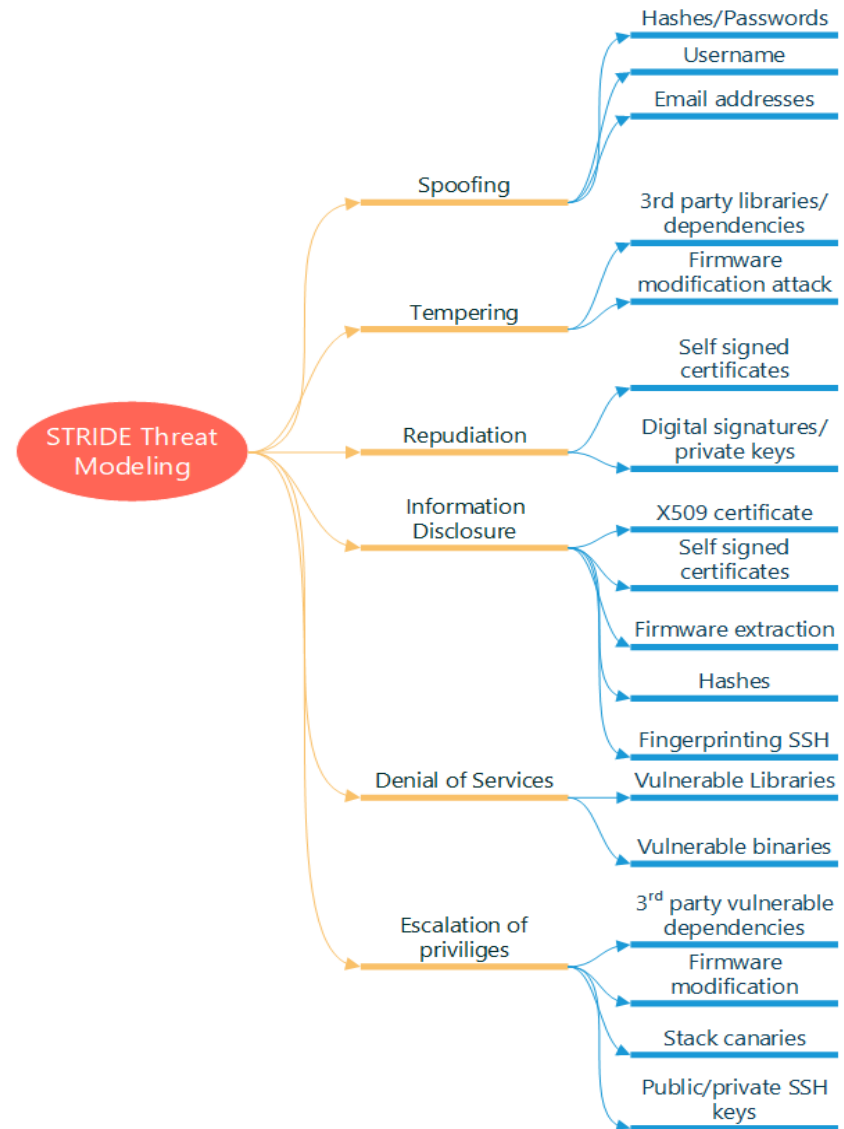
STRIDE Threat Model

- Developed by Microsoft corporation to identify the threats associated with each component of a system that needs to be developed
- Normally used at design time even before a single line of code is being written
- It requires realization of assets and associated vulnerabilities

STRIDE Model	Concerned Attributes
Spoofing	Authenticity
Tampering of data	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Denial of services	Availability
Escalation of privileges	Authorization

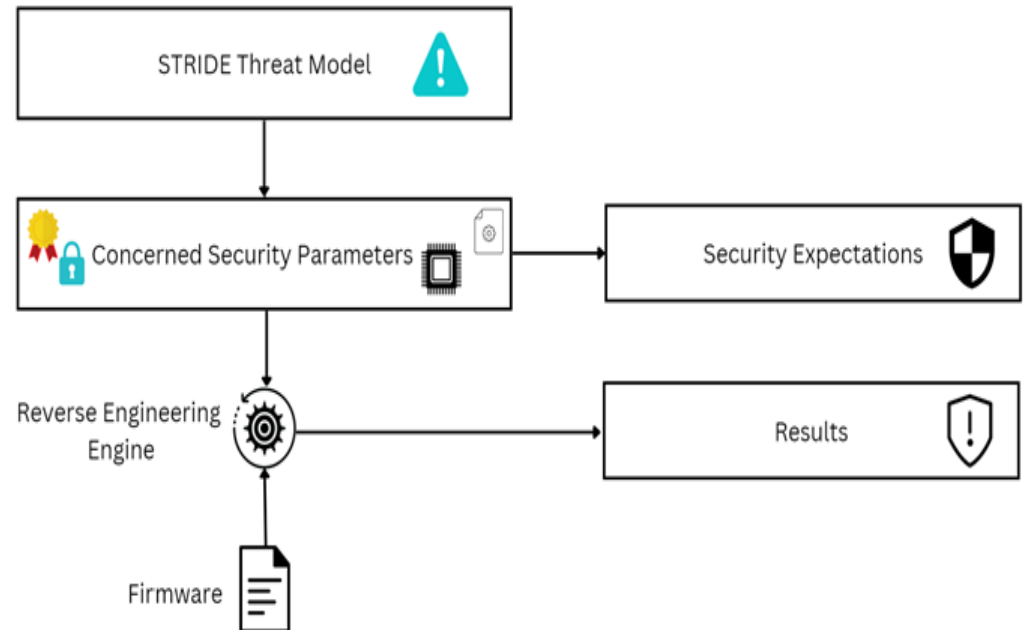
Identification of parameters using Stride

- In our methodology, we are proposing that scope of threat modelling can be increased to identify the critical vulnerabilities of already developed systems.
- Even if threat modelling has not been performed by developers, it is still important for the user to find the weaknesses of the system that could be exploited to launch different attacks.



Methodology – Phase 2

- Collection of a large firmware dataset using web crawlers and scrappers for different types of devices (a total of 4364 firmware images)
- Use of Reverse engineering to extract information from firmware and its components – development of an auditing engine ‘Thingzalyzer’ [3]
- Analysis based on the security parameters enumerated using STRIDE model



[3] “IoT cyber security solution providers | Vulnerability scan Tool,” 2021. [Online]. Available: <https://thingzeze.com/>

ThingzAnalyzer – An Auditing Engine

It provides the capability to Reverse engineer the IoT firmware and perform vulnerability assessment.

Some of its features include:

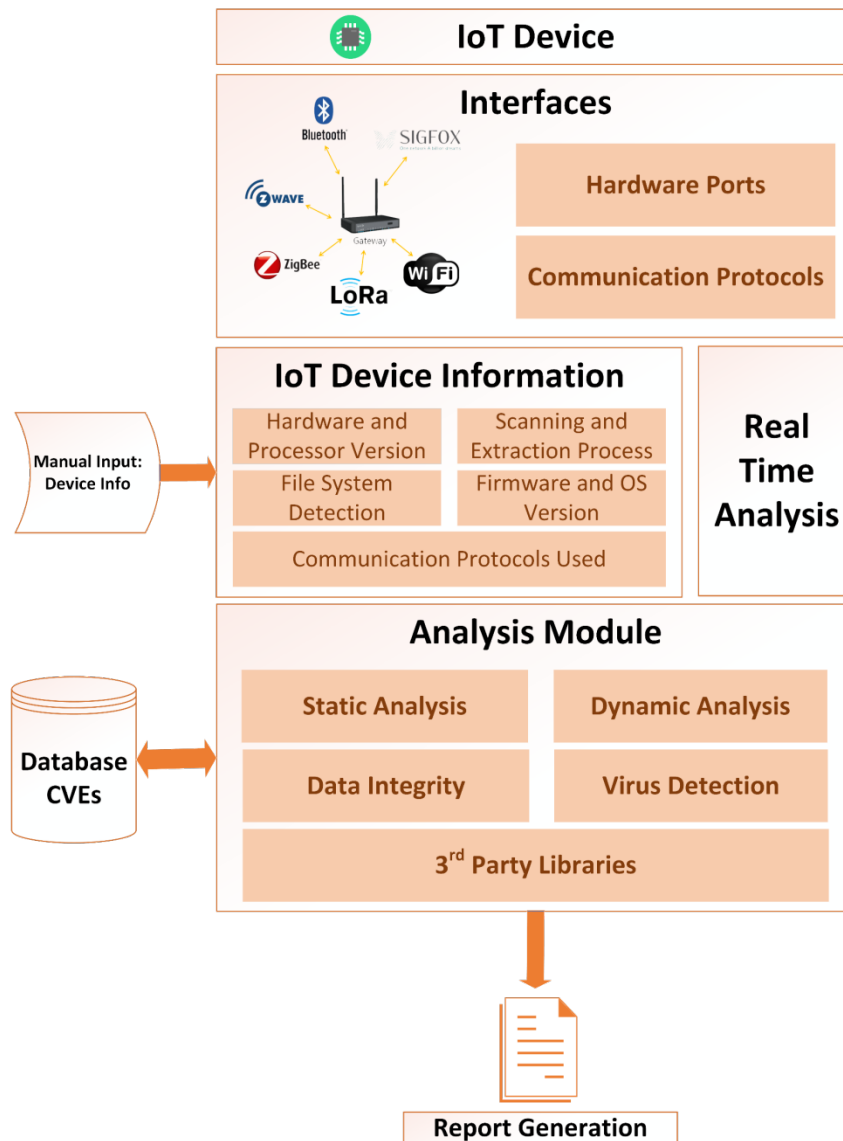
- Firmware extraction and information gathering
- Collection of various security relevant files using custom build parsers
- Cracking of passwords
- Analyse certificate files- Expiration date, signed status, presence of private keys
- Fingerprinting
- Binary files analysis



<https://thingzeeye.com/>

<https://analyzer.thingzeeye.com/>

An auditing framework for vulnerability analysis



Features



Static Vulnerability assessment of
firmware

01



Static Compliance testing of
firmware

02



Dynamic web application testing of
IoT device

03



Dynamic network services testing of IoT
device

04

Static Vulnerability Assessment



Firmware

Check firmware core components against NVD for vulnerabilities

- Kernel version and bootloader version



Binaries

Check binaries version against NVD database for vulnerabilities

- Busybox, miniupnpd, openssl, lighttpd etc



Libraries

Check Libraries version against NVD database for vulnerabilities

- libzebra, libthread, libsqlite etc

STATIC COMPLIANCE TESTING OF FIRMWARE

1



Check for x509 certificates

Encryption algorithm, expiry,
self signing of certificate,
Private keys

2



Check for encryption keys

Private Keys, public keys

3



Check for configuration files

Find hashes and its cracking

4



Check for Shadow files

Password hashes cracking

5



Check for email addresses

Email accounts which are
found in firmware

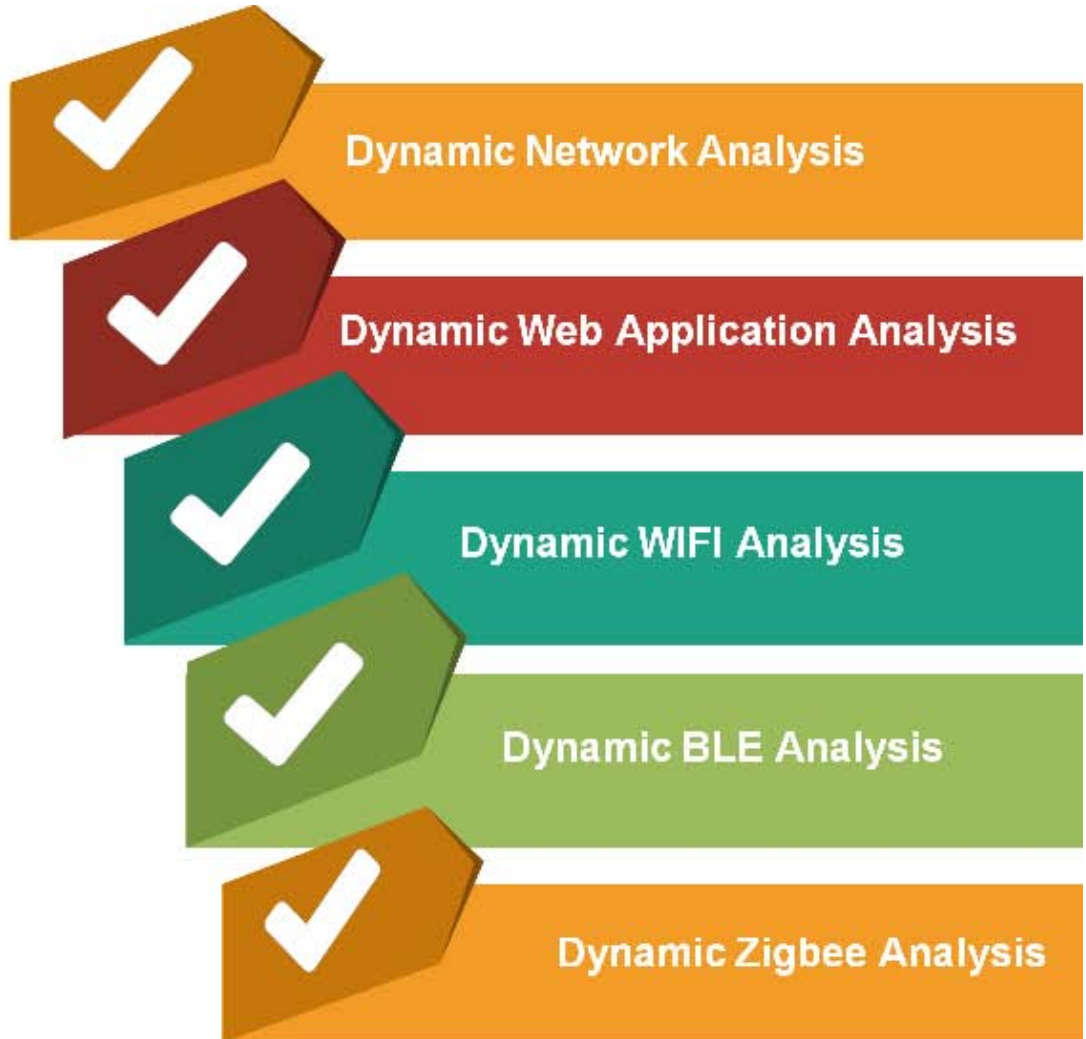
6



Check for ELF executable files

Check for RELRO support,
Stack canaries support, PIE
support etc.

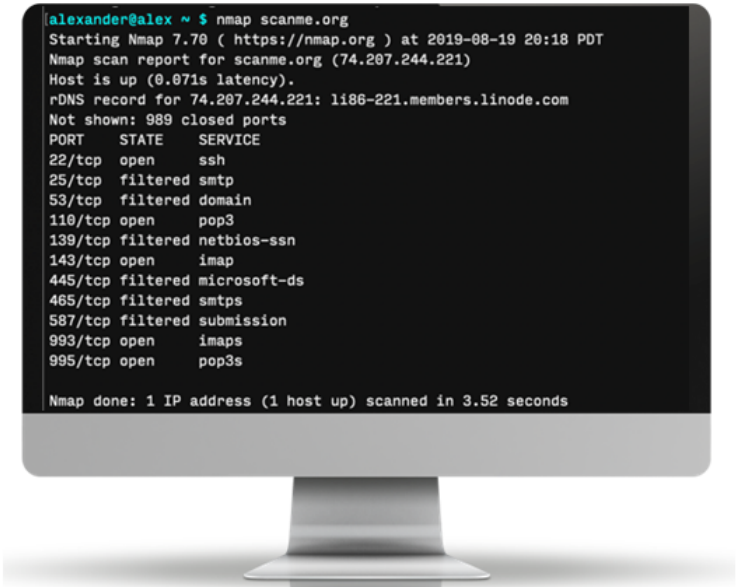
Security Audit Platform (SAP)



DYNAMIC NETWORK SERVICES TESTING

Check Network Services Versions

Scanning of open ports



```
alexander@alex ~ $ nmap scanme.org
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-19 20:18 PDT
Nmap scan report for scanme.org (74.207.244.221)
Host is up (0.071s latency).
rDNS record for 74.207.244.221: 1186-221.members.linode.com
Not shown: 989 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    filtered smtp
53/tcp    filtered domain
110/tcp    open  pop3
139/tcp    filtered netbios-ssn
143/tcp    open  imap
445/tcp    filtered microsoft-ds
465/tcp    filtered smtps
587/tcp    filtered submission
993/tcp    open  imaps
995/tcp    open  pop3s

Nmap done: 1 IP address (1 host up) scanned in 3.52 seconds
```

DYNAMIC WEB APPLICATION TESTING

HTTP

Checking for
allowed HTTP
methods



Hidden

Checking Hidden
files or
directories



HTTP

Checking HTTP
response
headers



Check

Check for
Robot.txt file



Metadata

Check for
metadata and
comments to find
an information
leakage



Cipher

Check for SSL
cipher suit
fingerprint



Policy File

Check for RIA
cross domain
policy file

Versions

Sever Version
and web
application
version

SAP features

Wifi	BLE	Zigbee	Web Application	Network Services
Authentication method check	Security manager protocol check	ID Encryption Check	Checking HTTP response headers	Check for open ports
802.11w Protocol Active Flag Check	Slave Bluetooth Address Check	Over-the-Air (OTA) PAN key transport Check	Checking Hidden files/directories	Check Network service version
Gather Information on hidden access points	Secure Conn flag check		Check for Robot.txt file	
Identify fake access points	MITM Flag Check		Check allowed HTTP methods	
Cracking AP Password via Customized Dictionary Attack	Link Key Check		Check SSL fingerprint	
	Signature Key Check		Check RIA cross domain policy file	
	Long Term Key Check		Check for metadata and comments	
	Cracking BLE Encryption		Sever Version and web app version	

Vulnerable devices

Device	Total No. of Firmware	Type No. of Firmware Posing Threat
Routers/AP	1247	662
Network/IP Cameras	1540	832
DVR/NVRs	414	298
Network Switches	60	48
Smart Doorbell	25	12
Smart TV	22	21
Smart Switches	122	96
Smart Wi-Fi systems	129	34
Air Quality Monitor	7	2
Misc.	798	750
Total	4364	2564

Most common Vulnerabilities

STRIDE Model	Concerned Security Parameters	No of Firmware Posing Threat
Spoofing	Username	1002
	Hashes	1002
	Decoded Passwords	585
	Email Addresses	1513
Tampering	3rd Parties Libraries/Dependencies	404
	Firmware Modification Attack	2622
Repudiation	Self Signed Certificates	862
	Digital Signatures/Private Keys	74
Information Disclosure	X509 Certificate	862
	Self-Signed Certificates	862
	Firmware Extraction	2622
	Hashes	1002
	Fingerprinting SSH	24
Denial of Service	Vulnerable Libraries	404
	Vulnerable Binaries	1411
Escalation of Privileges	3rd Party Vulnerable Dependencies	404
	Firmware Modification	2622
	Stack Canaries	0
	Public/Private SSH key	24

Open research areas

#	Issue	Domain
1	Unified firmware auditing framework	Standardization
2	Unified instruction set architecture	Standardization
3	Unified firmware stack	Standardization
4	De-compilers for all architectures	Technical
5	Automation of reverse engineering	Technical
6	Improved emulations	Technical
7	Secure firmware update	Technical
8	Emulation support for additional architectures	Design
9	Investigate non-Linux firmware	Design
10	IoT specific operating systems	Design