Requirements and Security Challenges for Resource-Constrained IoT End-Devices Baseband Processor

International Winter School on Microarchitectural Security

Paris, France, December 6, 2022

Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOJNIAT

Lab-STICC, Team ARCAD, Université Bretagne Sud

[firstname].[lastname]@univ-ubs.fr
Introduction: IoT devices (1/2)

- Number of Internet of Things (IoT) devices expanding exponentially (+10 Billions, in 2021; [Jovanović and Vojinovic, 2021])
- A wide range of applications and use-cases (ex: healthcare, industry and agriculture)
- Multiple constraints on resources are related to IoT devices (energy, communication range, data rate, flexibility, life-cycle, . . . )
Figure: Global number of connected IoT devices [IoT, 2020]
• Explosion of number of IoT device connections (+20 Billions in 2019) [IoT, 2020])

• Emergence of a large number of IoT standards and protocols

• Development of Low Data Rate and Low Power protocols to match the challenges of the IoT environment (LoRa, BLuetooth/BLE, NB-IoT, Zigbee, SigFox, …)
Figure: Total number of device connections (incl. Non-IoT) [IoT, 2020]
Motivation and Identified challenges

- Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).

- Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021]).

- Network systems are considered to be one of the most important potential entry points for attacks. DoS, DDoS, Jamming, MITM, ...

- Physical layers are implemented with a dedicated hardware architecture.

- New approaches to implement the physical layer using Software Defined Radio (SDR) architecture are proposed to reach flexibility and multi-protocol operations.

- The implementation of wireless connectivity using (SDR) could expand the attack surface for traditional security exploits (ROP, Overflow, ...).

- Various requirements and challenges have to be considered in the design of IoT devices: Security, Flexibility and Power Consumption.
Motivation and Identified challenges

- Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).

- Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021]).
Motivation and Identified challenges

- Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).
- Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021])
- Network systems are considered to be one of the most important potential entry points for attacks. DoS, DDoS, Jamming, MITM, …
Motivation and Identified challenges

- Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).

- Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021]).

- Network systems are considered to be one of the most important potential entry points for attacks. DoS, DDoS, Jamming, MITM, …

- Physical layers are implemented with a dedicated hardware architecture.
Motivation and Identified challenges

- Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).

- Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021])

- Network systems are considered to be one of the most important potential entry points for attacks. DoS, DDoS, Jamming, MITM, …

- Physical layers are implemented with a dedicated hardware architecture

- New approaches to implement the physical layer using Software Defined Radio (SDR) architecture are proposed to reach flexibility and multi-protocol operations.
Motivation and Identified challenges

- Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).
- Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021])
- Network systems are considered to be one of the most important potential entry points for attacks. DoS, DDoS, Jamming, MITM, ...
- Physical layers are implemented with a dedicated hardware architecture
- New approaches to implement the physical layer using Software Defined Radio (SDR) architecture are proposed to reach flexibility and multi-protocol operations.
- The implementation of wireless connectivity using (SDR) could expand the attack surface for traditional security exploits (ROP, Overflow, ...).
Motivation and Identified challenges

• Several challenges resulting from the evolution of IoT infrastructures (number of devices, waveforms and communication protocols).

• Appearance of attacks and vulnerabilities affecting the IoT devices (1.5 Billion attacks in 2021 [Price, 2021])

• Network systems are considered to be one of the most important potential entry points for attacks. DoS, DDoS, Jamming, MITM, …

• Physical layers are implemented with a dedicated hardware architecture

• New approaches to implement the physical layer using Software Defined Radio (SDR) architecture are proposed to reach flexibility and multi-protocol operations.

• The implementation of wireless connectivity using (SDR) could expand the attack surface for traditional security exploits (ROP, Overflow, …).

• Various requirements and challenges have to be considered in the design of IoT devices: Security, Flexibility and Power Consumption.
wrap-up

Security of embedded systems?

- Physical Access
- Cryptography Implementation
- ...
- Network Entry Point

Figure: IoT architecture
Objectives

- Focus on wireless connectivity of resource-constraints IoT devices
Objectives

- Focus on wireless connectivity of resource-constrained IoT devices
- Development of secure, flexible processor for wireless connectivity
Objectives

- Focus on wireless connectivity of resource-constraints IoT devices
- Development of secure, flexible processor for wireless connectivity
- Target Low data rate and low power protocols and waveforms
  BLE, LoRa/LoRaWAN, Zigbee and 6LoWPAN

© Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT
Objectives

- Focus on wireless connectivity of resource-constrained IoT devices
- Development of secure, flexible processor for wireless connectivity
- Target Low data rate and low power protocols and waveforms BLE, LoRa/LoRaWAN, Zigbee and 6LoWPAN
- Achieve the integrity of IoT devices and network availability
Objectives

- Focus on wireless connectivity of resource-constrained IoT devices
- Development of secure, flexible processor for wireless connectivity
- Target Low data rate and low power protocols and waveforms: BLE, LoRa/LoRaWAN, Zigbee and 6LoWPAN
- Achieve the integrity of IoT devices and network availability
- Focus on RISC-V open source ISA for BaseBand/Network CPU
Main CPU for application user
SoC for IoT overview

- Main CPU for application user
- Peripherals and connectivity

Figure: SoC IoT overview
SoC for IoT overview

- Main CPU for application user
- Peripherals and connectivity
- Integration of protection mechanisms

Figure: SoC IoT overview
SoC for IoT overview

- Main CPU for application user
- Peripherals and connectivity
- Integration of protection mechanisms
- Isolation between Radio and user application

Figure: SoC IoT overview

Don’t forget that SoC are integrating a wireless connectivity unit!
Baseband architecture: Dedicated Hardware

- ESP32-C3 from Espressif
- Dedicated hardware (Baseband part) for each waveform / Protocol
- Lack of flexibility

Figure: Wireless Connectivity of ESP32-C3
Baseband architecture: CPU + DSP

- Generic CPU based architecture (Without ISA extension)
- Integration of a DSP for the radio part

Figure: wireless connectivity of CC1352R
Baseband Architecture: Hybrid FPGA

- Hybrid FPGA (Zynq) or FPGA + MCU

Figure: SoC TinySDR [Hessar et al., 2020]
Baseband Architecture: CPU with SIMD

- CPU Dedicated architecture

Figure: SMID based CPU Dedicated Architecture
[Chen et al., 2016]
Baseband Architecture: Generic CPU

- CPU with ISA extension (ARM, RISC-V)

Figure: Architecture ARM
[Xhonneux et al., 2021]

Figure: Architecture RISC-V
[Amor et al., 2019, Belhadj Amor et al., 2021]
Examples of SoC in industry

- Texas instruments
- ST Microelectronics
- NXP
- Espressif
- ...

Figure: CC1352R SoC Texas instruments
Examples of SoC in industry

- Texas instruments
- ST Microelectronics
- NXP
- Espressif
- ...

Several SoCs in industry include a core dedicated to wireless connectivity

Figure: CC1352R SoC Texas instruments
A software-defined baseband radio processor using a generic CPU architecture with an instruction set extension is more interesting.

The constraints of limited resources and consumption of connected objects must be taken into account.

Other challenges associated with the software radio must also be taken into account: security, programmability

<table>
<thead>
<tr>
<th></th>
<th>Baseband</th>
<th>Dedicated Hardware</th>
<th>Hybrid FPGA</th>
<th>CPU (dedicated)</th>
<th>CPU (Generic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Protocol</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Programmability</td>
<td>✗</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+++</td>
</tr>
<tr>
<td>Security Mechanism</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Flexibility</td>
<td>✗</td>
<td>+++</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Dynamic power</td>
<td>~ 100mW</td>
<td>~ 100mW</td>
<td>~ 10mW</td>
<td>~ 10µW</td>
<td></td>
</tr>
</tbody>
</table>

Table: A comparison of IoT SDR baseband processor architectures and their features
Target: Remote Attacks

Figure: Potential Threat Model
Target : Remote Attacks
**Threat Model**

**Figure: Potential Threat Model**

**Target: Remote Attacks**

- Jamming Attack
- Logical Attacks: Packet Injection, ...
Vulnerabilities in IoT

<table>
<thead>
<tr>
<th>Vulnerability</th>
<th>AMNESIA33</th>
<th>BLEEDINGBIT</th>
<th>LoRaDawn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CVEs</td>
<td>33 [Labs, 2020]</td>
<td>2 [Seri, Benn (ARMIS et al., 2019]</td>
<td>2 [ten, 2020]</td>
</tr>
<tr>
<td>Where?</td>
<td>Poor Software Development</td>
<td>Masking Error, OAD</td>
<td>OTAA Process, 32bit Gateway</td>
</tr>
<tr>
<td>Target Device</td>
<td>uIP, FNET, picoTCP, NuTNet</td>
<td>AP with TI BLE</td>
<td>LoRaMac-node, LoRa Basics Station</td>
</tr>
<tr>
<td>Stack Layer</td>
<td>Physical /MAC</td>
<td>MAC</td>
<td>MAC</td>
</tr>
<tr>
<td>Stack / protocol</td>
<td>TCP/IP / IEEE 802.15.4</td>
<td>BLE</td>
<td>LoRaWAN</td>
</tr>
<tr>
<td>Exploit</td>
<td>RCE, DoS, Steal Data</td>
<td>Packet injection, RCE</td>
<td>DoS, RCE, Heap UAF</td>
</tr>
</tbody>
</table>

Table: A set of three Groups of vulnerabilities in IoT and their features

Figure: SoC for IoT with wireless connectivity
Example of Exploit: InjectBLE [Cayre et al., 2021]

- Vulnerabilities: Long synchronization time between Slave and Master BLE in connection step
- Exploit: Packet injection (Hijacking slave and master, MITM)
- InjectBLE Firmware
- Mirage framework
- Used BLE module: nRF52840-dongle

Figure: nRF52840-dongle: https://www.nordicsemi.com/
Example of Exploit: Main in the middle (MITM) attack

We reproduce the MITM attack using two modules from mirage framework in order to sniff packets between master and slave: *(ble_hijack and ble_master)*

- **ble_master**: Mobile App
- **ble_slave**: Led strip
- **Attacker**: Laptop with nRF52840-dongle

Figure: Sniffing packet exploit
Example of Exploit: Packet Injection

After hijacking the BLE Master we perform a packet injection exploit

Figure: Packet Injection exploit
Attacks in IoT

SoC: IoT end-point

Application

USER

Firmware

Main CPU

Wireless Connectivity

Stack Sub-GHz / GHz

Upper Layers

MAC

PHY

Network Processor

Front end

Figure: SoC for IoT

IoT Protocol Stack

Upper Layers

MAC Layer

Physical Layer

Table: Security SoA IoT Low Data rates protocols (Sub-GHz, Zigbee, BLE)

© Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT 22
Attacks in IoT

Figure: SoC for IoT

Figure: IoT protocol stack layers

<table>
<thead>
<tr>
<th>Ref</th>
<th>Protocol</th>
<th>Attack</th>
<th>PHY</th>
<th>MAC</th>
<th>Upper</th>
<th>Exploit</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Cayre et al., 2021]</td>
<td>Zigbee</td>
<td>Wazabee</td>
<td>E/T</td>
<td>E/T</td>
<td>T</td>
<td>DoS, packet injection</td>
</tr>
<tr>
<td>[Aras et al., ]</td>
<td>LoRaWAN</td>
<td>Selective Jamming</td>
<td>E/T</td>
<td>E/T</td>
<td>T</td>
<td>DoS, Wormhole</td>
</tr>
<tr>
<td>[Hessel et al., ]</td>
<td>LoRaWAN</td>
<td>Spoofing</td>
<td>E/T</td>
<td>E/T</td>
<td>T</td>
<td>DoS</td>
</tr>
<tr>
<td>[Avoine and Ferreira, 2018]</td>
<td>LoRaWAN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>T</td>
<td>replay, decrypt, DoS</td>
</tr>
<tr>
<td>[Cayre et al., 2021]</td>
<td>BLE</td>
<td>InjectBLE</td>
<td>E/T</td>
<td>E/T</td>
<td>T</td>
<td>MITM, Sniffing</td>
</tr>
<tr>
<td>[Zhang et al., 2020]</td>
<td>BLE</td>
<td>Downgrade</td>
<td>-</td>
<td>-</td>
<td>T</td>
<td>DoS, MITM</td>
</tr>
<tr>
<td>[Santos et al., 2019]</td>
<td>BLE</td>
<td>Injection-free</td>
<td>-</td>
<td>-</td>
<td>E/T</td>
<td>DoS, MITM</td>
</tr>
<tr>
<td>[Antonioli et al., 2020]</td>
<td>BT/BLE</td>
<td>Key.nego downgrade</td>
<td>E/T</td>
<td>E/T</td>
<td>T</td>
<td>Decrypt packet, MITM</td>
</tr>
</tbody>
</table>

Table: Security SoA IoT Low Data rates protocols (Sub-GHz, Zigbee, BLE)
### Security mechanisms & mitigation

#### Features

<table>
<thead>
<tr>
<th>Features</th>
<th>CC1356</th>
<th>CC1352R1</th>
<th>STM32WL54CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec. Boot (protection)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cryptography (protection)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OTA (Update)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Heap ASLR (protection)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Monitoring (detection)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>DIFT (hard. monitor)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Code instrumentation (protection)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Anomaly/Intrusion detection</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Table: Platform security features comparison

#### Security Mechanisms

- Confidentiality, Integrity and availability
- Protection mechanisms
- Update & Over the air Mechanisms
- **Monitoring & Detection Mechanisms**

Figure: CC1352R1: SoC for IoT
Motivation and contribution

Motivation

- Remote attacks detection on wireless connectivity of IoT SoC
- The necessity of a monitoring detection mechanism that captures system behavior and identifies attacks.
Motivation and contribution

Motivation

- Remote attacks detection on wireless connectivity of IoT SoC
- The necessity of a monitoring detection mechanism that captures system behavior and identifies attacks.

Contribution: Intrusion Detection System (IDS)

- Acquisition, Analyze and Identification, warn or block attacks
IDS taxonomy

Figure: IDS taxonomy for IoT environment
IDS taxonomy

Figure: IDS taxonomy for IoT environment
IDS taxonomy

IDS for IoT

Placement
- Gateway
- End-Device
- Hybrid
- IDS device

Detection
- Signature
- Anomaly
- Rule
- Hybrid

Metrics layer

Figure: IDS taxonomy for IoT environment
IDS taxonomy

Figure: IDS taxonomy for IoT environment
What are the accurate metrics to be recorded for an HIDS?
What are the accurate metrics to be recorded for an HIDS?

<table>
<thead>
<tr>
<th>Ref</th>
<th>PHY</th>
<th>MAC</th>
<th>UL</th>
<th>μProc</th>
<th>RT</th>
<th>Target</th>
<th>PS</th>
<th>DM</th>
<th>Place</th>
</tr>
</thead>
</table>

- **MAC** (Mac layer): **TS** (Time series), **P** (Packet Header)
- **UP** (Upper layers): **TS** (Time series)
- **HW** (Hardware/processor): **IMA** (Illegal memory access), **HPC** (Hardware Performance counter)
- **SW** (Software/runtime): **SC** (Syscalls)
- **Target attacks**: **Spoof** (Spoofing), **Jamm** (Jamming), **Pinject** (Packet Injection), **Rout** (Rooting), **Snik** (Sinkhole)
- **PS** (Proposed Solution): **LKM** (Loadable kernel module), **min.FW** (mini firewall), **ML** (Machine Learning)
- **DM** (Detection Methodology): **B** (Behavior), **S** (signature)
- **Place** (Placement Strategy): **RC** (Resource constraint), **G** : (Gateway), **D** (Device), **H** (Hybrid)
Host based IDS in state of the art

What are the accurate metrics to be recorded for an HIDS?

<table>
<thead>
<tr>
<th>Ref</th>
<th>PHY</th>
<th>MAC</th>
<th>UL</th>
<th>µProc</th>
<th>RT</th>
<th>Target</th>
<th>PS</th>
<th>DM</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Yan et al., 2020]</td>
<td>RSSI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Spoof</td>
<td>Model legit.RSSI</td>
<td>B</td>
<td>G / RC</td>
</tr>
<tr>
<td>[Zhang et al., 2013]</td>
<td>RSSI</td>
<td>TS</td>
<td>TS</td>
<td>-</td>
<td>-</td>
<td>integrity</td>
<td>SDR</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>[Sousa et al., 2017]</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DoS</td>
<td>Analyze &amp; store</td>
<td>S</td>
<td>RC</td>
</tr>
<tr>
<td>[Kasinathan et al., 2013]</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DoS, Jamm</td>
<td>SURICATA</td>
<td>S</td>
<td>D</td>
</tr>
<tr>
<td>[Eskandari et al., 2020]</td>
<td>Trafic</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>P.inject</td>
<td>GUI LINUX</td>
<td>S</td>
<td>G</td>
</tr>
<tr>
<td>[Raza et al., 2013]</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Rout, Snik</td>
<td>IDS + min.FW</td>
<td>B+S</td>
<td>H</td>
</tr>
</tbody>
</table>

Table: Host based IDS for IoT

- **MAC** (Mac layer): **TS** (Time series), **P** (Packet Header)
- **UP** (Upper layers): **TS** (Time series)
- **HW** (Hardware/processor): **IMA** (Illegal memory access), **HPC** (Hardware Performance counter)
- **SW** (Software/runtime): **SC** (Syscalls)
- **Target attacks**: **Spoof** (Spoofing), **Jamm** (Jamming), **P.inject** (Packet Injection), **Rout** (Rooting), **Snik** (Sinkhole)
- **PS** (Proposed Solution): **LKM** (Loadable kernel module), **min.FW** (mini firewall), **ML** (Machine Learning)
- **DM** (Detection Methodology): **B** (Behavior), **S** (signature)
- **Place** (Placement Strategy): **RC** (Resource constraint), **G** : (Gateway), **D** (Device), **H** (Hybrid)
What are the accurate metrics to be recorded for an HIDS?

<table>
<thead>
<tr>
<th>Ref</th>
<th>PHY</th>
<th>MAC</th>
<th>UL</th>
<th>µProc</th>
<th>RT</th>
<th>Target</th>
<th>PS</th>
<th>DM</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yan et al., 2020</td>
<td>RSSI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Spoof</td>
<td>Model legiti.RSSI</td>
<td>B</td>
<td>G / RC</td>
</tr>
<tr>
<td>Zhang et al., 2013</td>
<td>RSSI</td>
<td>TS</td>
<td>TS</td>
<td>-</td>
<td>-</td>
<td>integrity</td>
<td>SDR</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>Sousa et al., 2017</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DoS</td>
<td>Analyze &amp; store</td>
<td>S</td>
<td>RC</td>
</tr>
<tr>
<td>Kasinathan et al., 2013</td>
<td>Trafic</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DoS, Jamm</td>
<td>SURICATA</td>
<td>S</td>
<td>D</td>
</tr>
<tr>
<td>Eskandari et al., 2020</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>P.inject</td>
<td>GUI LINUX</td>
<td>S</td>
<td>G</td>
</tr>
<tr>
<td>Raza et al., 2013</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Rout, Snik</td>
<td>IDS + min.FW</td>
<td>B+S</td>
<td>H</td>
</tr>
<tr>
<td>Saeed et al., 2016</td>
<td>-</td>
<td>-</td>
<td>Sensor</td>
<td>IMA</td>
<td>-</td>
<td>P.inject, DoS</td>
<td>C.Instru + ML</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>Gassais et al., 2020</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DD/DoS</td>
<td>Tracing + ML</td>
<td>S</td>
<td>H</td>
</tr>
<tr>
<td>Bourdon et al., 2021</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>P.inject</td>
<td>Tracing + ML</td>
<td>B</td>
<td>H</td>
</tr>
</tbody>
</table>

Table: Host based IDS for IoT

- **MAC** (Mac layer): **TS** (Time series), **P** (Packet Header)
- **UP** (Upper layers): **TS** (Time series)
- **HW** (Hardware/processor): **IMA** (Illegal memory access), **HPC** (Hardware Performance counter)
- **SW** (Software/runtime): **SC** (Syscalls)
- **Target attacks**: **Spoof** (Spoofing), **Jamm** (Jamming), **P.inject** (Packet Injection), **Rout** (Rooting), **Snik** (Sinkhole)
- **PS** (Proposed Solution): **LKM** (Loadable kernel module), **min.FW** (mini firewall), **ML** (Machine Learning)
- **DM** (Detection Methodology): **B** (Behavior), **S** (signature)
- **Place** (Placement Strategy): **RC** (Resource constraint), **G** (Gateway), **D** (Device), **H** (Hybrid)
Host based IDS in state of the art

What are the accurate metrics to be recorded for an HIDS?

<table>
<thead>
<tr>
<th>Ref</th>
<th>PHY</th>
<th>MAC</th>
<th>UL</th>
<th>µProc</th>
<th>RT</th>
<th>Target</th>
<th>PS</th>
<th>DM</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Yan et al., 2020]</td>
<td></td>
<td>RSSI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Spoof</td>
<td>Model legit.RSSI</td>
<td>B</td>
<td>G / RC</td>
</tr>
<tr>
<td>[Zhang et al., 2013]</td>
<td></td>
<td>RSSI</td>
<td>TS</td>
<td>TS</td>
<td>-</td>
<td>integrity</td>
<td>SDR</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>[Sousa et al., 2017]</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DoS</td>
<td>Analyze &amp; store</td>
<td>S</td>
<td>RC</td>
</tr>
<tr>
<td>[Kasinathan et al., 2013]</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>DoS, Jamm</td>
<td>SURICATA</td>
<td>S</td>
<td>D</td>
</tr>
<tr>
<td>[Eskandari et al., 2020]</td>
<td>Trafic</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>P.Inject</td>
<td>GUI LINUX</td>
<td>S</td>
<td>G</td>
</tr>
<tr>
<td>[Raza et al., 2013]</td>
<td>-</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ROut, Snik</td>
<td>IDS + min.FW</td>
<td>B+S</td>
<td>H</td>
</tr>
<tr>
<td>[Saeed et al., 2016]</td>
<td>-</td>
<td>-</td>
<td>Sensor</td>
<td>IMA</td>
<td>-</td>
<td>P.Inject, DoS</td>
<td>C.Instru + ML</td>
<td>B</td>
<td>G</td>
</tr>
<tr>
<td>[Gassais et al., 2020]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>CTF</td>
<td>-</td>
<td>DD/DoS</td>
<td>Tracing + ML</td>
<td>S</td>
<td>H</td>
</tr>
<tr>
<td>[Bourdon et al., 2021]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>HPC</td>
<td>-</td>
<td>P.Inject</td>
<td>Tracing + ML</td>
<td>B</td>
<td>H</td>
</tr>
<tr>
<td>[Breitenbacher et al., 2019]</td>
<td>-</td>
<td>N/A</td>
<td>-</td>
<td>SC</td>
<td>0-day, DoS</td>
<td>LKM + Whitelist</td>
<td>B</td>
<td>RC</td>
<td></td>
</tr>
</tbody>
</table>

Table: Host based IDS for IoT

- **MAC** (Mac layer): **TS** (Time series), **P** (Packet Header)
- **UP** (Upper layers): **TS** (Time series)
- **HW** (Hardware/processor): **IMA** (Illegal memory access), **HPC** (Hardware Performance counter)
- **SW** (Software/runtime): **SC** (Syscalls)
- **Target attacks**: **Spoof** (Spoofing), **Jamm** (Jamming), **P.Inject** (Packet Injection), **Rout** (Rooting), **Snik** (Sinkhole)
- **PS** (Proposed Solution): **LKM** (Loadable kernel module), **min.FW** (mini firewall), **ML** (Machine Learning)
- **DM** (Detection Methodology): **B** (Behavior), **S** (signature)
- **Place** (Placement Strategy): **RC** (Resource constraint), **G** (Gateway), **D** (Device), **H** (Hybrid)

The multi-level approach is not yet addressed in the state of the art
Towards a multi-level metrics HIDS

Wireless connectivity block diagram with IDS

© Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT
Towards a multi-level metrics HIDS

Wireless connectivity block diagram with IDS
Towards a multi-level metrics HIDS

Wireless connectivity block diagram with IDS
Objective

- **Proposed Hardware:**
  - CV32E41P RISC-V Processor for handling the wireless connectivity

Figure: CV32E41P/40P block diagram

Figure: Testbed block diagram
Objective

- **Proposed Hardware:**
  - CV32E41P RISC-V Processor for handling the wireless connectivity
  - Record Hardware Performance Counters (HPC) from CV32E41P by HPMtracer (Hardware block)

Figure: CV32E41P/40P block diagram

Figure: Testbed block diagram
Objective

- **Proposed Hardware:**
  - CV32E41P RISC-V Processor for handling the wireless connectivity
  - Record Hardware Performance Counters (HPC) from CV32E41P by HPMtracer (Hardware block)

- **Scenario**
  - Reproduction of simple buffer overflow exploit on stack and heap on software running on wireless connectivity part

Figure: CV32E41P/40P block diagram

Figure: Testbed block diagram
Objective

- **Proposed Hardware:**
  - CV32E41P RISC-V Processor for handling the wireless connectivity
  - Record Hardware Performance Counters (HPC) from CV32E41P by HPMtracer (Hardware block)

- **Scenario**
  - Reproduction of *simple buffer overflow exploit on stack and heap* on software running on wireless connectivity part
  - Build Dataset of HPC values per each packet network

Figure: CV32E41P/40P block diagram

Figure: Testbed block diagram
Test-bed with tracing metrics from RISC-V CV32E41P

Software: MAC Layer

- Parsing Network Packets

Hardware

- CV32E41P RISC-V
- WISHBONE BUS
- RAM
- UART
- SPI
- TIMER

Figure: Test-bed block diagram
Test-bed with tracing metrics from RISC-V CV32E41P

Software: MAC Layer

- Parsing Network Packets
- HPM Reset
- HPM Enable
- HPM Stop

Hardware

- RISC-V CV32E41P
- HPMtracer
- WISHBONE BUS
- RAM
- UART
- TIMER
- SPI

Figure: Test-bed block diagram
Test-bed with tracing metrics from RISC-V CV32E41P

Software: MAC Layer
- Parsing Network Packets
  - HPM Reset
  - HPM Enable
  - HPM Stop

Hardware
- RISC-V CV32E41P
- HPMtracer
- WISHBONE BUS
- RAM
- UART
- SPI
- TIMER

Network Traffic Generator
- Number of Packet + Length
  - Header file
  - Network Traffic

Figure: Test-bed block diagram
Test-bed with tracing metrics from RISC-V CV32E41P

Software: MAC Layer

- Parsing Network Packets
- HPM Reset
- HPM Enable
- HPM Stop

Hardware

- CV32E41P RISC-V
- WISHBONE BUS
- RAM
- UART
- SPI

Network Traffic Generator

- Number of Packet + Length
- Header file
- Network Traffic

Post-Processing

- Pre-Processing
- Training
- Attack classification
- Build ML Model

Log file

Figure: Test-bed block diagram
Flow diagram

Figure: Flow diagram of network packet processing, HPC monitoring and detection.
### Attack Scenarios

<table>
<thead>
<tr>
<th>Packet Type</th>
<th>Traffic Size</th>
<th>Stack</th>
<th>Heap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legitimate</td>
<td>5 – 10 bytes</td>
<td>10 bytes</td>
<td>10 bytes</td>
</tr>
<tr>
<td>S1: Stack Overflow</td>
<td>13 – 23 bytes</td>
<td>10 bytes</td>
<td>23 bytes</td>
</tr>
<tr>
<td>S2: Heap Overflow</td>
<td>13 – 23 bytes</td>
<td>23 bytes</td>
<td>10 bytes</td>
</tr>
</tbody>
</table>

Table: The physical buffer size is 10 or 23 bytes. Larger packets result in a buffer overflow.
## List of monitored hardware events

<table>
<thead>
<tr>
<th>Hardware Event</th>
<th>Description</th>
<th>Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYCLES</td>
<td>Number of cycles</td>
<td>0</td>
</tr>
<tr>
<td>INSTR</td>
<td>Number of instructions retired</td>
<td>2</td>
</tr>
<tr>
<td>LDSTALL</td>
<td>Number of load use hazards</td>
<td>3</td>
</tr>
<tr>
<td>JMPSTALL</td>
<td>Number of jump register hazards</td>
<td>4</td>
</tr>
<tr>
<td>IMISS</td>
<td>Cycles waiting for instruction fetches</td>
<td>5</td>
</tr>
<tr>
<td>LD</td>
<td>Number of load instructions</td>
<td>6</td>
</tr>
<tr>
<td>ST</td>
<td>Number of store instructions</td>
<td>7</td>
</tr>
<tr>
<td>JUMP</td>
<td>Number of jumps (unconditional)</td>
<td>8</td>
</tr>
<tr>
<td>BRANCH</td>
<td>Number of branches (conditional)</td>
<td>9</td>
</tr>
<tr>
<td>BRANCH_TAKEN</td>
<td>Number of branches taken (conditional)</td>
<td>10</td>
</tr>
<tr>
<td>COMP_INSTR</td>
<td>Number of compressed instructions retired</td>
<td>11</td>
</tr>
</tbody>
</table>

Table: List of hardware events monitored by the CV32E41P performance counters
Figure: Distribution of cumulative values of hardware events IMISS, Store and JMP_STALL in attack scenarios
Preliminary results

This histogram shows the evaluation results of the comparison of several classification algorithms.

Figure: Comparison of ML Classifiers Models

- Interesting Results
- An in-depth study to follow: Data-set, Scenarios, Detection, Cost?
Generated decision tree classifier model

Figure: Generated decision tree classifier model
Table: Implementation resource utilization and power consumption

<table>
<thead>
<tr>
<th>HIDS elements</th>
<th>Overhead</th>
<th>Freq</th>
<th>Average Total Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPM (nb)</td>
<td>LUT</td>
<td>FF</td>
<td>MHz</td>
</tr>
<tr>
<td>Tracer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V1 ✓ (1)</td>
<td>4636 (+00%)</td>
<td>1237 (+00%)</td>
<td>65.86 (+00%)</td>
</tr>
<tr>
<td>V2 ✓ (2)</td>
<td>4802 (+3.58%)</td>
<td>1318 (+6.54%)</td>
<td>65.35 (−0.77%)</td>
</tr>
<tr>
<td>V3 ✓ (2) ✓ ✓</td>
<td>4932 (+6.38%)</td>
<td>1318 (+6.54%)</td>
<td>65.47 (−0.59%)</td>
</tr>
</tbody>
</table>

Resource Utilization: Arty-A7 35T FPGA

- 6.4%/6.5% of LUTs/FFs Area overhead
- 7.61% Total Power (around 7mW)
- 0.6% No impact on the design’s performance (65MHz)
Prototype with LoRaWAN

Software: LoRaMaCNode

- LoRa Driver + LoRaMac stack
- HPM Reset
- HPM Enable
- HPM Stop

Hardware:

- CV32E41P RISC-V
- HPMtracer
- WISHBONE BUS
- RAM
- UART
- SPI
- TIMER
- DIOs
- RESET
- Decision Tree Detector
  - Traces Analysis
  - Attack classification
  - Raising alerts

Figure: SoC architecture with LoRaMACnode stack

Figure: Arty-a7 100T FPGA with SX1276 based LoRa shield

© Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT
Conclusion

- **Ongoing work**
  - New approach for monitoring and detecting remote attacks against IoT devices
  - Simulation Test-bed to detect buffer overflow using hardware counters.
  - Promising results of machine learning classification algorithms.
  - Prototype Testbed with LoRa & LoRaWAN Protocol

- **Future work**
  - Include new features (SNR, RSSI, IAT,…) + new attacks (Jamming, …)
THANK YOU

Q & A
Requirements and Security Challenges for Resource-Constrained IoT End-Devices Baseband Processor

International Winter School on Microarchitectural Security

Paris, France, December 6, 2022

Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT

Lab-STICC, Team ARCAD, Université Bretagne Sud

[firstname].[lastname]@univ-ubs.fr
Loradawn - multiple lorawan security vulnerabilities.

Number of connected iot devices //iot-analytics.com/.

Baseband Processor for IoT To cite this version : HAL Id : cea-01936120
Software-Hardware Co-Design of Multi-Standard Digital Baseband Processor for IoT.

Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy.
ACM Transactions on Privacy and Security, 23(3).

[Aras et al., ] Aras, E., Small, N., Ramachandran, G. S., Delbruel, S., Joosen, W., and Hughes, D.
Selective jamming of LoRaWAN using commodity hardware.


Références (5/8)

TinySDR : Low-Power SDR Platform for Over-the-Air Programmable IoT Testbeds This paper is included in the Proceedings of the TinySDR : Low-Power SDR Platform for. Proc. of NSDI.

[Hessel et al., ] Hessel, F., Almon, L., and Álvarez, F.

45 fascinating iot statistics for 2021: The state of the industry.


Amnesia:33, how tcp/ip stacks breed critical vulnerabilities in iot, ot and it devices.
Iot cyber attacks double to 1.5 billion in first half of 2021.

SVELTE: Real-time intrusion detection in the Internet of Things. 

Intelligent intrusion detection in low-power IoTs. 
ACM Transactions on Internet Technology, 16(4).

BLE injection-free attack: a novel attack on bluetooth low energy devices. 
Journal of Ambient Intelligence and Humanized Computing, (0123456789).

[Seri, Benn (ARMIS et al., 2019] Seri, Benn (ARMIS, I., Zusman, Dor (ARMIS, I., and Vishnepolsky, Gregory (ARMIS, I. (2019). 
BLEEDINGBIT : The hidden attack surface within BLE chips.

