Wireless security and hardware assisted Intrusion Detection System

CYBERUS SUMMER SCHOOL, France, July 03-07, 2023

Mohamed EL BOUAZZATI Tianxu LI Philippe TANGUY Guy GOGNIAT

Univ. Bretagne-Sud, Lab-STICC, Lorient, France

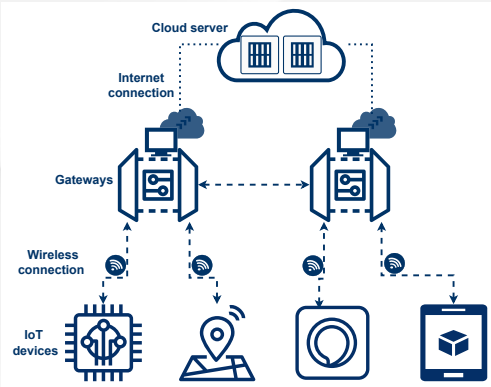
Outline

1. Research context
2. Threat model
3. Vulnerabilities in IoT
4. Proposed security mechanism: hardware based HIDS
5. Test-bed & evaluation

Outline

Security of embedded systems?

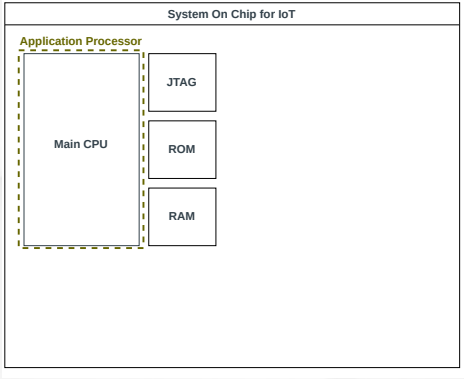
- Physical access protection
- Cryptography implementation
- …
- Network (wireless connection) entry point



Internet of things architecture

SoC for IoT overview

- Main CPU for application
- Peripherals and connectivity
- Integration of protection mechanism
- Isolation between wireless connectivity unit and application processor

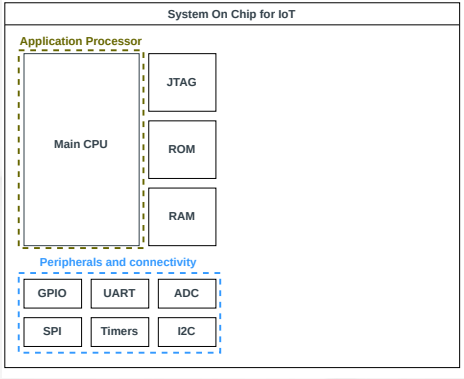


System-on-Chip overview

SoC has a built-in wireless connectivity unit!

SoC for IoT overview

- Main CPU for application
- Peripherals and connectivity
- Integration of protection mechanism
- Isolation between wireless connectivity unit and application processor

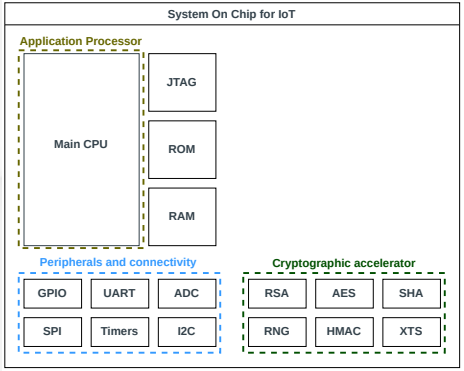


System-on-Chip overview

SoC has a built-in wireless connectivity unit!

SoC for IoT overview

- Main CPU for application
- Peripherals and connectivity
- Integration of protection mechanism
- Isolation between wireless connectivity unit and application processor

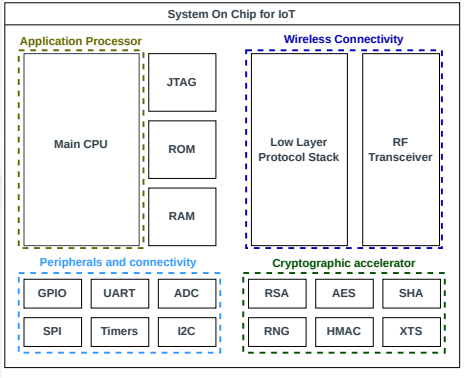


System-on-Chip overview

SoC has a built-in wireless connectivity unit!

SoC for IoT overview

- Main CPU for application
- Peripherals and connectivity
- Integration of protection mechanism
- Isolation between wireless connectivity unit and application processor

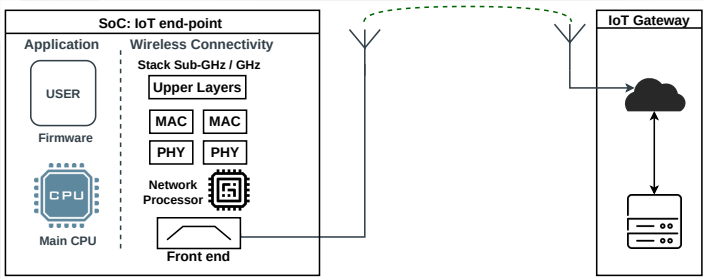


System-on-Chip overview

SoC has a built-in wireless connectivity unit!

Outline

Threat model

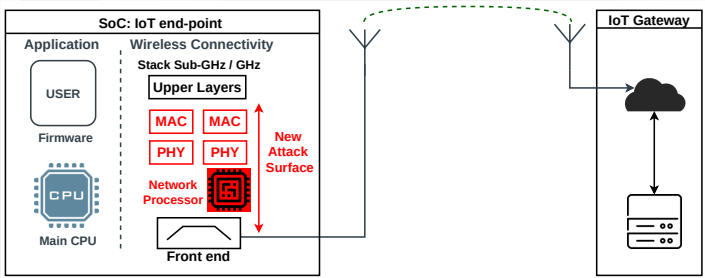


Potential threat model

Target: remote attacks

- Memory corruption attacks: packet injection, . . .
- Possible exploits: denial of service, man in the middle, remote code execution and privilege escalation,. . .

Threat model

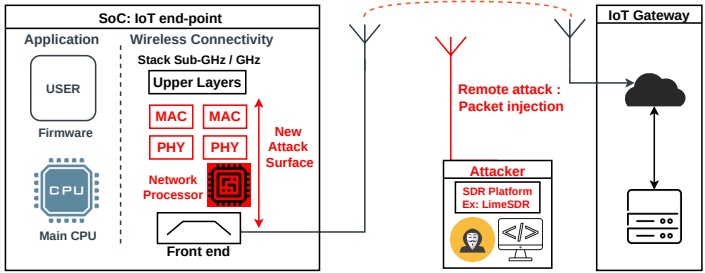


Potential threat model

Target: remote attacks

- Memory corruption attacks: packet injection, …
- Possible exploits: denial of service, man in the middle, remote code execution and privilege escalation,…

Threat model



Potential threat model

Target: remote attacks

- Memory corruption attacks: packet injection, …
- Possible exploits: denial of service, man in the middle, remote code execution and privilege escalation,…

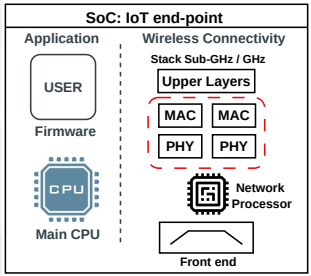
Outline

1. Research context
2. Threat model
3. **Vulnerabilities in IoT**
   - Attacks in IoT
   - Attacks targeting LoRaWAN: examples
   - Attacks targeting IEEE802.15.4: examples
   - Security mechanisms & mitigation
4. Proposed security mechanism: hardware based HIDS
5. Test-bed & evaluation

Vulnerabilities in IoT

Vulnerabilities

- A group of CVE found in IoT stacks
  - BLEEDINGBIT in Bluetooth/BLE
  - LoRaDawn in LoRa/LoRaWAN
  - AMNESIA33 in TCP/IP
  - Several CVE in Zigbee stacks (e.g. Philips HUE CVE-2020-6007)
- Reasons :
  - Poor software development
  - Encryption weakness
  - Pairing process bypass
  - ...



SoC for IoT with wireless connectivity

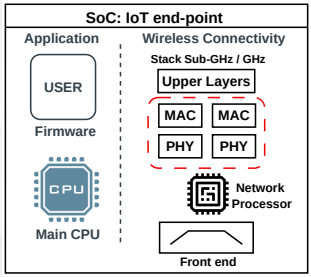
IoT Stack

- IoT stack implementation and standards are not secured
- Physical and MAC layers are vulnerable in various IoT stacks

Vulnerabilities in IoT

Vulnerabilities

- A group of CVE found in IoT stacks
  - BLEEDINGBIT in Bluetooth/BLE
  - LoRaDawn in LoRa/LoRaWAN
  - AMNESIA33 in TCP/IP
  - Several CVE in Zigbee stacks (e.g. Philips HUE CVE-2020-6007)
- Reasons :
  - Poor software development
  - Encryption weakness
  - Pairing process bypass
  - …



SoC for IoT with wireless connectivity

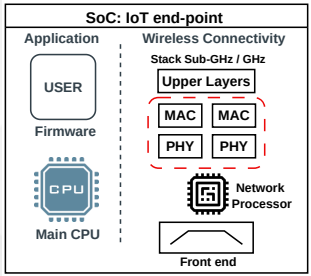
IoT Stack

- IoT stack implementation and standards are not secured
- Physical and MAC layers are vulnerable in various IoT stacks

Vulnerabilities in IoT

Vulnerabilities

- A group of CVE found in IoT stacks
  - BLEEDINGBIT in Bluetooth/BLE
  - LoRaDawn in LoRa/LoRaWAN
  - AMNESIA33 in TCP/IP
  - Several CVE in Zigbee stacks (e.g. Philips HUE CVE-2020-6007)
- Reasons :
  - Poor software development
  - Encryption weakness
  - Pairing process bypass
  - …



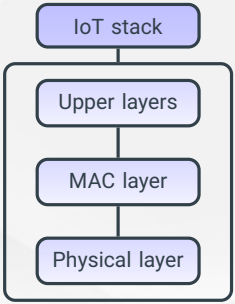
SoC for IoT with wireless connectivity

IoT Stack

- IoT stack implementation and standards are not secured
- Physical and MAC layers are vulnerable in various IoT stacks

Attacks in IoT

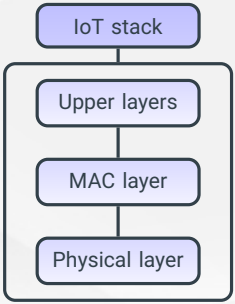
- Taking advantages of existing vulnerabilities in lower layers
- Targeting upper layers in IoT stack
- Possible exploits:
  - Taking control of IoT device
  - Denial of service
  - Stealing data
  - …



IoT stack layers

Attacks in IoT

- Taking advantages of existing vulnerabilities in lower layers
- Targeting upper layers in IoT stack
- Possible exploits:
  - Taking control of IoT device
  - Denial of service
  - Stealing data
  - ...



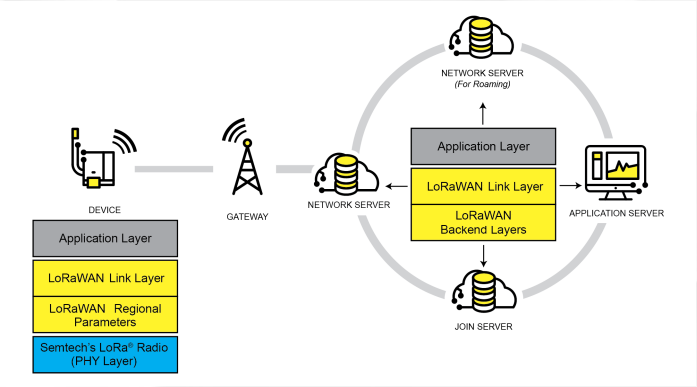
IoT stack layers

Example: LoRaWAN

LoRa (from "long range") is a physical proprietary radio communication technique. It provides long-range connectivity by using the chirp spread spectrum technique.

LoRaWAN (Wide Area Network) defines the communication protocol and system architecture.

LoRa + LoRaWAN –> Low Power, Wide Area (LPWA) networking protocol for IoTs



LoRaWAN network architecture [1]

[1] LoRa Alliance Certification Committee, "Test tool simplifies and automates LoRaWAN certification," 5G Technology World, Apr. 28, 2022. https://www.5gtechnologyworld.com/test-tool-simplifies-and-automates-lorawan-certification/ (accessed Jun. 30, 2023).

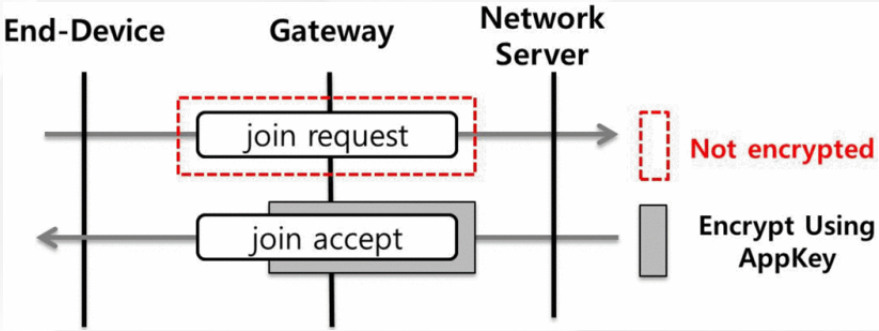
Example: LoRaWAN - attacks

Attack type	Summary
Replay attack	The attacker listens to the message, intercepts it and resends it if necessary to mislead the recipient. [2]
DDoS/DOS	The attacker floods the target servers with a large number of unwanted requests. This incapacitates the target server, thereby disrupting services to genuine users. [3]
Jamming Attack	LoRa devices which send data simultaneously using certain frequencies and parameters can corrupt each other's signal. By abusing this vulnerability, it is possible to jam LoRa messages maliciously. [4]
Buffer overflow	Due to a lack of buffer size checks, attackers can overflow a buffer by sending a message longer than expected to corrupt an unlicensed memory range. [5]

Different types of attacks against LoRaWAN (Non-exhaustive)

[2] S. Na, D. Hwang, W. Shin, and K.-H. Kim, "Scenario and countermeasure for replay attack using join request messages in LoRaWAN," in 2017 International Conference on Information Networking (ICOIN), Jan. 2017, pp. 718–720. doi: 10.1109/ICOIN.2017.7899580.
[3] O. Jullian, B. Otero, E. Rodriguez, N. Gutierrez, H. Antona, and R. Canal, "Deep-Learning Based Detection for Cyber-Attacks in IoT Networks: A Distributed Attack Detection Framework," J Netw Syst Manage, vol. 31, no. 2, p. 33, Feb. 2023, doi: 10.1007/s10922-023-09722-7.
[4] C.-Y. Huang, C.-W. Lin, R.-G. Cheng, S. J. Yang, and S.-T. Sheu, "Experimental Evaluation of Jamming Threat in LoRaWAN," in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Apr. 2019, pp. 1–6. doi: 10.1109/VTCSpring.2019.8746374.
[5] M. E. Bouazzati, R. Tessier, P. Tanguy, and G. Gogniat, "A Lightweight Intrusion Detection System against IoT Memory Corruption Attacks," in 2023 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), May 2023, pp. 118–123. doi: 10.1109/DDECS57882.2023.10139718.

- **Over-the-Air Activation(OTTA) in Lorawan**
  - End devices participate in the network after exchanging the information necessary for data transmission through the OTAA procedure. In the OTAA procedure, messages that exchanged between the end device and the network server consist of join request and join accept. [6]



OTAA message flow in LoRaWAN 1.0 [2]

[6] "End Device Activation," The Things Network. https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/ (accessed Jun. 30, 2023).

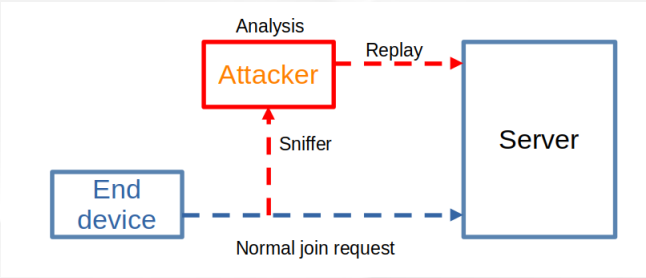
- **Architecture of a Join Request Message** [6]
  - AppEUI: a 64-bit globally unique application identifier in IEEE EUI64 address space that uniquely identifies the entity able to process the Join-req frame.
  - DevEUI: a 64-bit globally unique device identifier in IEEE EUI64 address space that uniquely identifies the end-device.
  - DevNonce: a unique, random, 2-byte value generated by the end device. The Network Server uses the DevNonce of each end-device to keep track of their join requests. If an end device sends a Join-request with a previously used DevNonce, the Network Server rejects the Join-request and does not allow that end device to register with the network.

Size(bytes)	8	8	2
Join Request	AppEUI	DevEUI	DevNonce

Join request message

- **Vulnerability of a Join Request Message**
  - We can check all of the contents in join request message including frequency and SF(Spread Factor) information without decryption process. The DevNonce in this message is a value required for replay attack, we can use it to let the network server discards the request message of other target end device.

- **Three steps of Replay attack Using Sniffed Join Request Messages**
  - **1.Information Gathering**: The attacker uses a sniffer devices to collect join requests messages. In this step attaker will try to collect messages as much as possible.
  - **2.Analysis of Data**: The attacker analyzes the period in which a particular end device generates join request messages.
  - **3.Attack**: The attacker sends the same message with the same period. At this point the web server will try to connect with the attacker, and subsequent messages sent by the end device will be ignored. Since it is considered a repeated join request, the end device will be disconnected at this time.

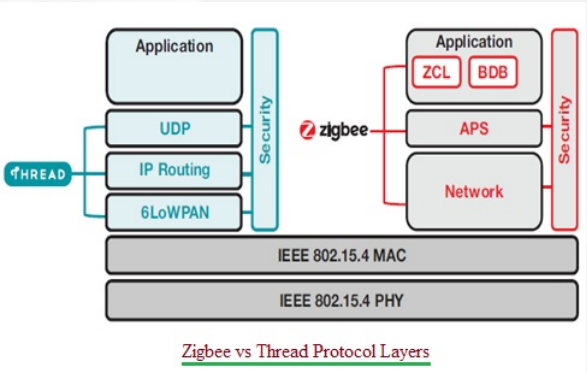


Schematic diagram of replay attack

Example: IEEE 802.15.4

- **What is IEEE 802.15.4**:
  - IEEE 802.15.4 is a technical standard which defines the operation of a low-rate wireless personal area network (LR-WPAN). It specifies the physical layer and media access control for LR-WPANs.
- **Zigbee & Thread**

- Both Zigbee and Thread build on the physical layer and media access control defined in IEEE standard 802.15.4

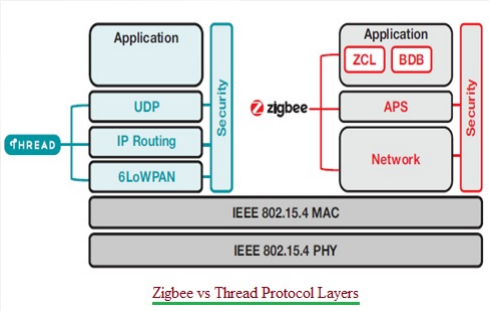


Zigbee and Thread protocol layering

Example: IEEE 802.15.4 and upper layers

Function	Zigbee	Thread
IPv6 support	No	Yes
Defin of App-lication layer	Yes	No
Authentication process	Via a trust center with proximity-based commissioning	Smartphone-based, QR code scanning
Security	Network-wide encryption and authentication through install code	Password-based authentication with Datagram Transport Layer Security (DTLS)

Table : Diffrences between Zigbee and Thread [7]



Zigbee and Thread protocol layering

[7] "Zigbee vs Thread | Difference between Zigbee and Thread."
https://www.rfwireless-world.com/Terminology/Difference-between-Zigbee-and-Thread.html (accessed Jun. 30, 2023).

Example: IEEE 802.15.4 - attacks

Attack type	Summary
Sybil Attack	A Sybil attack uses a single node to operate many active fake identities simultaneously, within a peer-to-peer network. [8]
Energy Depletion Attack	An attacker constructs bogus messages to lure a node to do superfluous security-related computations to intentionally deplete that node's energy. [9]
Jamming Attack	A malicious device emits high-power jamming signal to make all the IEEE802.15.4 devices in its proximity unable to communicate. [10]
Time Synchronization Attack	Attacker uses faking DIO packets to damage the structure of time synchronization tree. [11]

Different types of attacks against IEEE802.15.4 (Non-exhaustive)

[8] F. Amini, J. Misic, and H. Pourreza, "Detection of Sybil Attack in Beacon Enabled IEEE802.15.4 Networks," in 2008 International Wireless Communications and Mobile Computing Conference, Aug. 2008, pp. 1058–1063. doi: 10.1109/IWCMC.2008.184.
[9] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-ZigBee: Energy Depletion Attack on ZigBee-Based Wireless Networks," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 816–829, Oct. 2016, doi: 10.1109/JIOT.2016.2516102.
[10] H. Pirayesh, P. Kheirkhah Sangdeh, and H. Zeng, "Securing ZigBee Communications Against Constant Jamming Attack Using Neural Network," IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4957–4968, Mar. 2021, doi: 10.1109/JIOT.2020.3034128.
[11] W. Yang, Q. Wang, Y. Wan, and J. He, "Security Vulnerabilities and Countermeasures for Time Synchronization in IEEE802.15.4e Networks," in 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Jun. 2016, pp. 102–107. doi: 10.1109/CSCloud.2016.44.

Security mechanisms & mitigation

Features	CC1356	CC1352R1	STM32WL54CC
Secure boot (protection)	✓	✓	✓
Cryptography (protection)	✓	✓	✓
Over the air programming (update)	✓	✓	✓
Memory protection	✗	✗	✗
Code instrumentation (protection)	✗	✗	✗
Information tracking (detection)	✗	✗	✗
Anomaly/intrusion detection	✗	✗	✗

Platform security features comparison

Security mechanisms

- Confidentiality, integrity and availability
- Protection mechanisms
- Update & over the air mechanisms
- Monitoring & detection mechanisms

Security mechanisms & mitigation

Features	CC1356	CC1352R1	STM32WL54CC
Secure boot (protection)	✓	✓	✓
Cryptography (protection)	✓	✓	✓
Over the air programming (update)	✓	✓	✓
Memory protection	✗	✗	✗
Code instrumentation (protection)	✗	✗	✗
Information tracking (detection)	✗	✗	✗
Anomaly/intrusion detection	✗	✗	✗

Platform security features comparison

Security mechanisms

- Confidentiality, integrity and availability
- Protection mechanisms
- Update & over the air mechanisms
- Monitoring & detection mechanisms

Outline

1. Research context
2. Threat model
3. Vulnerabilities in IoT
4. **Proposed security mechanism: hardware based HIDS**
   - Motivation and contribution
   - Proposed lightweight hardware based HIDS
5. Test-bed & evaluation

Motivation and contribution

Motivation

- Memory corruption attacks detection on wireless connectivity of IoT SoC
- Require a monitoring and detection capability in order to record system activity and identify potential attacks.

Contribution: Intrusion detection system (IDS)

- Acquisition, analyze and identification, warn or block attacks

Motivation and contribution

Motivation

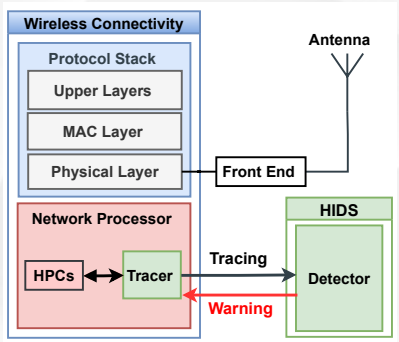
- Memory corruption attacks detection on wireless connectivity of IoT SoC
- Require a monitoring and detection capability in order to record system activity and identify potential attacks.

Contribution: Intrusion detection system (IDS)

- Acquisition, analyze and identification, warn or block attacks

Proposed lightweight hardware based HIDS

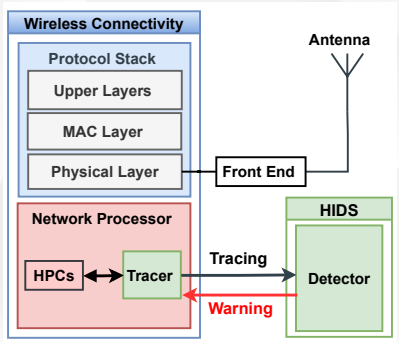
A hardware implementation of monitoring and detection modules on IoT device's wireless connectivity unit:



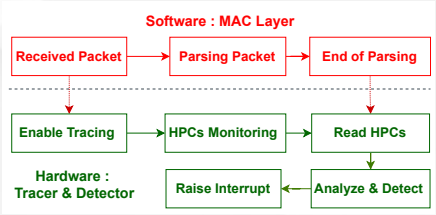
Wireless connectivity and HIDS (Host Intrusion Detection System) block diagram

Proposed lightweight hardware based HIDS

A hardware implementation of monitoring and detection modules on IoT device's wireless connectivity unit:



Wireless connectivity and HIDS (Host Intrusion Detection System) block diagram



Flow diagram of network packet processing, HPC monitoring and detection.

Outline

1. Research context
2. Threat model
3. Vulnerabilities in IoT
4. Proposed security mechanism: hardware based HIDS
5. **Test-bed & evaluation**
   - Objective
   - Simulation test-bed
   - LoRa stack test-bed with HIDS
   - Experimental results

Objective

Test-bed and scenarios

- Record by HPMtracer (hardware block) micro-architectural events using hardware performance counters (HPC) available on CV32E41P (32 bits RISC-V Processor)
- Reproduction of memory corruption attacks simple buffer overflow exploit
- Build large dataset of HPC values per each packet network for further analysis

Objective

Test-bed and scenarios

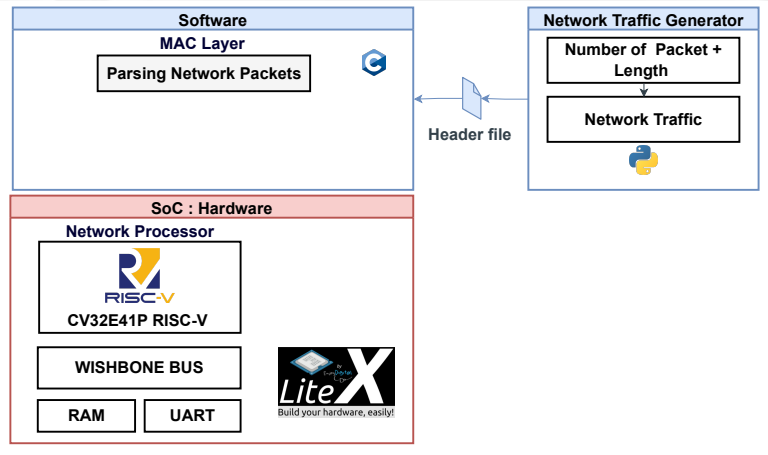
- Record by HPMtracer (hardware block) micro-architectural events using hardware performance counters (HPC) available on CV32E41P (32 bits RISC-V Processor)
- Reproduction of memory corruption attacks simple buffer overflow exploit
- Build large dataset of HPC values per each packet network for further analysis

Objective

Test-bed and scenarios

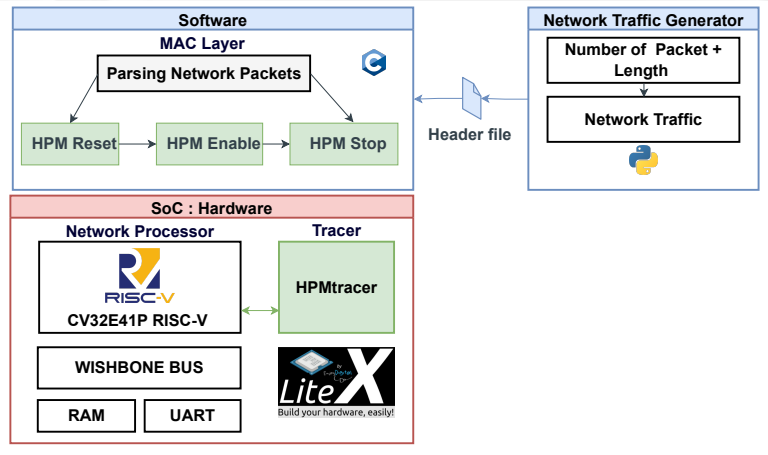
- Record by HPMtracer (hardware block) micro-architectural events using hardware performance counters (HPC) available on CV32E41P (32 bits RISC-V Processor)
- Reproduction of memory corruption attacks simple buffer overflow exploit
- Build large dataset of HPC values per each packet network for further analysis

Test-bed with HPC tracing from RISC-V CV32E41P



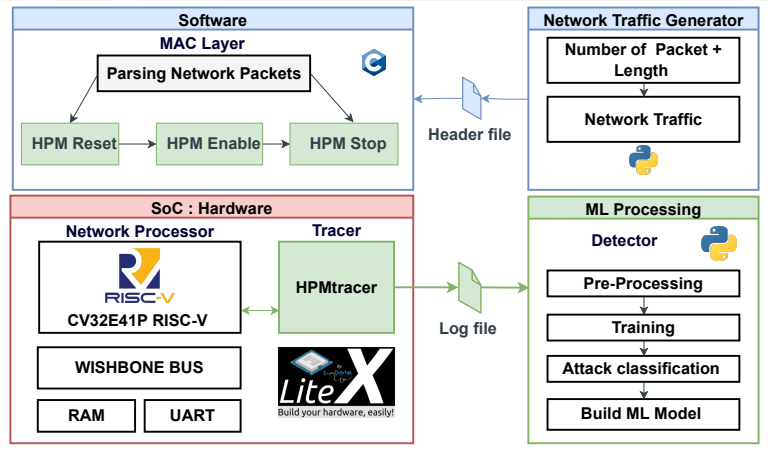
Test-bed block diagram

Test-bed with HPC tracing from RISC-V CV32E41P



Test-bed block diagram

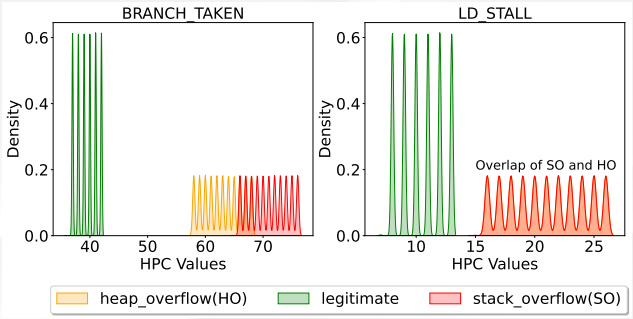
Test-bed with HPC tracing from RISC-V CV32E41P



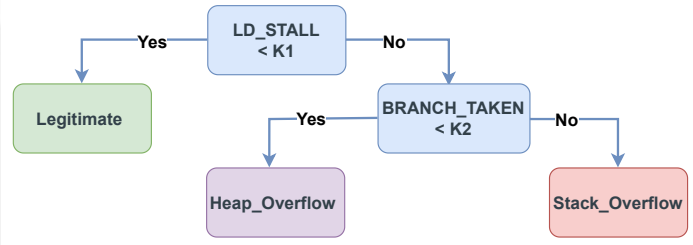
Test-bed block diagram

Attacks scenarios & generated dataset

Attack scenarios			Buffer size	
Dataset(packets)	**Packet type**	**Traffic size**	**Stack**	**Heap**
2,000,000	**Legitimate**	5 — 10 *bytes*	10 *bytes*	10 *bytes*
1,000,000	**S1: Stack overflow**	13 — 23 *bytes*	10 *bytes*	23 *bytes*
1,000,000	**S2: Heap overflow**	13 — 23 *bytes*	23 *bytes*	10 *bytes*



Classification model with decision tree



Generated decision tree classifier model

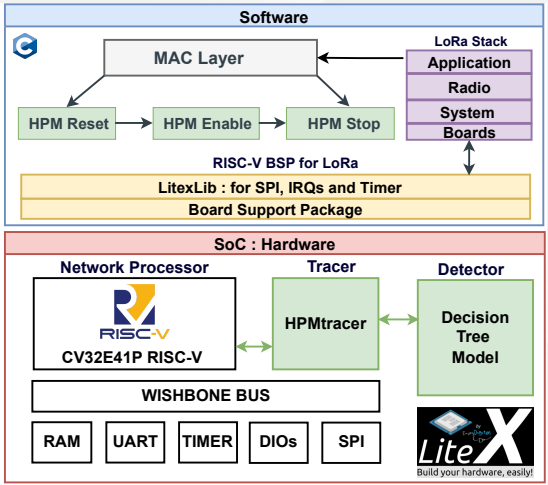
ML classification:

- Decision tree classifier: limited overhead and suitable classification speed in hardware
- BRANCH_TAKEN and LD_STALL selected from 11 other micro-architectural events by Decision Tree

LoRa test-bed over FPGA

FPGA setup

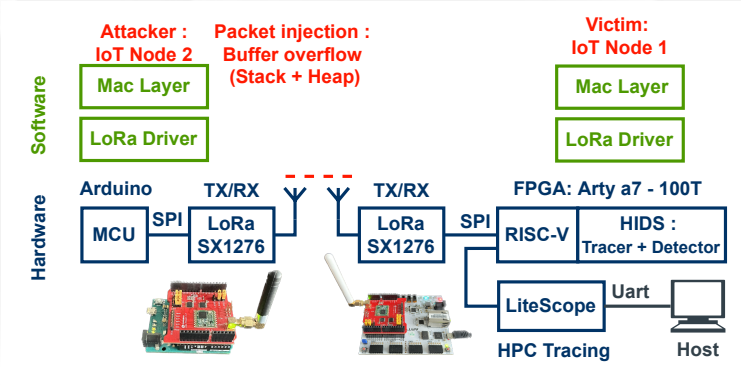
- Arty A7-100T: Artix-7 FPGA
- LoRaWAN end-device stack by Semtech
- HIDS elements : HPMtracer + decision tree model



SoC architecture with LoRaMACnode stack

Attack scenarios with LoRa test-bed

Test-bed used for attack reproduction and security evaluation of HIDS components:



LoRa test-bed with HIDS on Arty A7 FPGA board

Detection rate evaluation

True Positives	False Positives	True Neg.	False Neg.
195, 704	13	193, 327	53

False Negative Rate (FNR):	0.027%
False Positive Rate (FPR):	0.013%
Detection Accuracy :	99.98%

Hardware decision tree implementation evaluation metrics

Detection rate: LoRa test-bed

- High detection rates +99.98%
- Few malicious network packets detected as legitimate -0.030%

FPGA implementation cost

	HIDS elements			Overhead		Freq
	HPC (nb)	Tracer	Detector	*LUT*	*FF*	*MHz*
V1	✓ (1)	-	-	4636 (+00%)	1237 (+00%)	65.86 (+00%)
V2	✓ (2)	-	-	4802 (+3.58%)	1318 (+6.54%)	65.35 (−0.77%)
V3	✓ (2)	✓	✓	4932 (+6.38%)	1318 (+6.54%)	65.47 (−0.59%)

Implementation resource utilization and power consumption

Resource utilization: Arty-A7 35T FPGA

- 6.4%, 6.5% of LUTs/FFs area overhead
- 0.6% No impact on the design's performance (65*MHz*)

Conclusion

Addressing IoT device security issues at network entry point:

Current work

- New approach for monitoring and detecting memory corruption attacks against IoT devices
- Simulation test-bed generates large dataset of micro-architectural events using hardware counters
- Evaluation using real prototype test-bed with LoRa protocol
- Achievement of high detection rates +99.98% with an FPGA area overhead of less than 6.5% and without impact of maximum clock frequency 65 MHz.

Future work

- Include new metrics (SNR, RSSI, IAT,…) + new attacks (jamming, …)
- Implementation with embedded operating system (FreeRTOS, Zephyr, …)
- HIDS security and resources evaluation (comparison with software version, power consumption).

Conclusion

Addressing IoT device security issues at network entry point:

Current work

- New approach for monitoring and detecting memory corruption attacks against IoT devices
- Simulation test-bed generates large dataset of micro-architectural events using hardware counters
- Evaluation using real prototype test-bed with LoRa protocol
- Achievement of high detection rates +99.98% with an FPGA area overhead of less than 6.5% and without impact of maximum clock frequency 65 MHz.

Future work

- Include new metrics (SNR, RSSI, IAT,...) + new attacks (jamming, ...)
- Implementation with embedded operating system (FreeRTOS, Zephyr, ...)
- HIDS security and resources evaluation (comparison with software version, power consumption).

THANK YOU

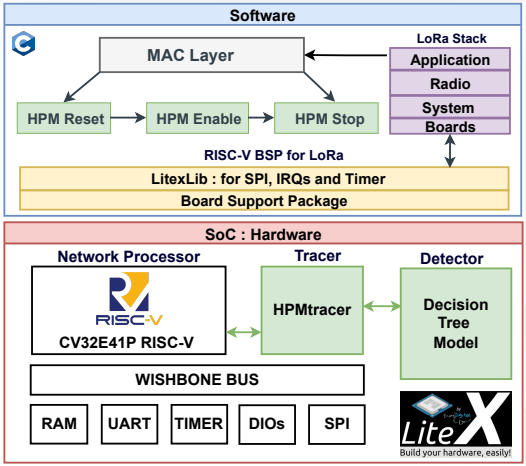
Wireless security and hardware assisted Intrusion Detection System

CYBERUS SUMMER SCHOOL, France, July 03-07, 2023

Mohamed EL BOUAZZATI Tianxu LI Philippe TANGUY Guy GOGNIAT

Univ. Bretagne-Sud, Lab-STICC, Lorient, France

References (1/1)

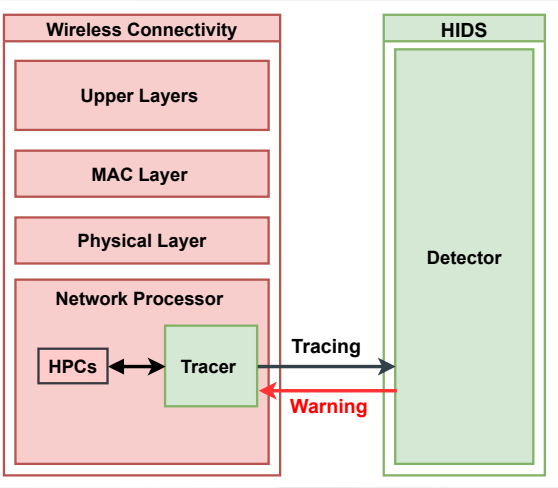


SoC architecture with LoRaMACnode stack

Appendix (2/8)

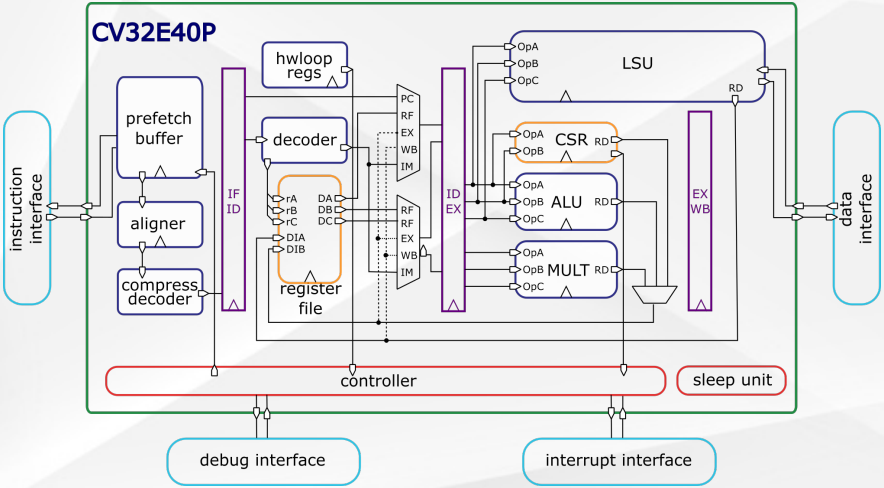
Hardware Event	Description	Counter
CYCLES	Number of cycles	0
INSTR	Number of instructions retired	2
LD_STALL	Number of load use hazards	3
JMP_STALL	Number of jump register hazards	4
IMISS	Cycles waiting for instruction fetches	5
LD	Number of load instructions	6
ST	Number of store instructions	7
JUMP	Number of jumps (unconditional)	8
BRANCH	Number of branches (conditional)	9
BRANCH_TAKEN	Number of branches taken (conditional)	10
COMP_INSTR	Number of compressed instructions retired	11

List of hardware events monitored by the CV32E41P performance counters



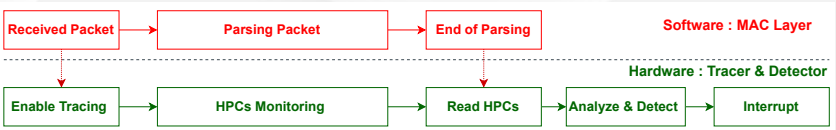
Wireless connectivity and HIDS (Host Intrusion Detection System) block diagram

Appendix (5/8)

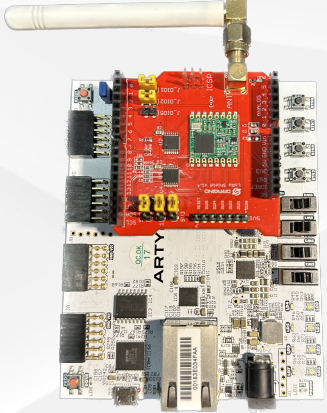


CV32E41P/40P block diagram

Appendix (7/8)



Flow diagram of network packet processing, HPC monitoring and detection.



Arty-a7 100T FPGA with SX1276 based LoRa shield