Towards Low-Power and Low Data-Rate Software-Defined Radio Baseband with RISC-V Processor for Flexibility and Security

18th CryptArchi Workshop - Porquerolles 2022

Porquerolles, France May 29- June 1, 2022

Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT

Lab-STICC, Team ARCAD, Université Bretagne Sud, Lorient, France

[firstname].[lastname]@univ-ubs.fr

Context

Security of embedded systems?

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

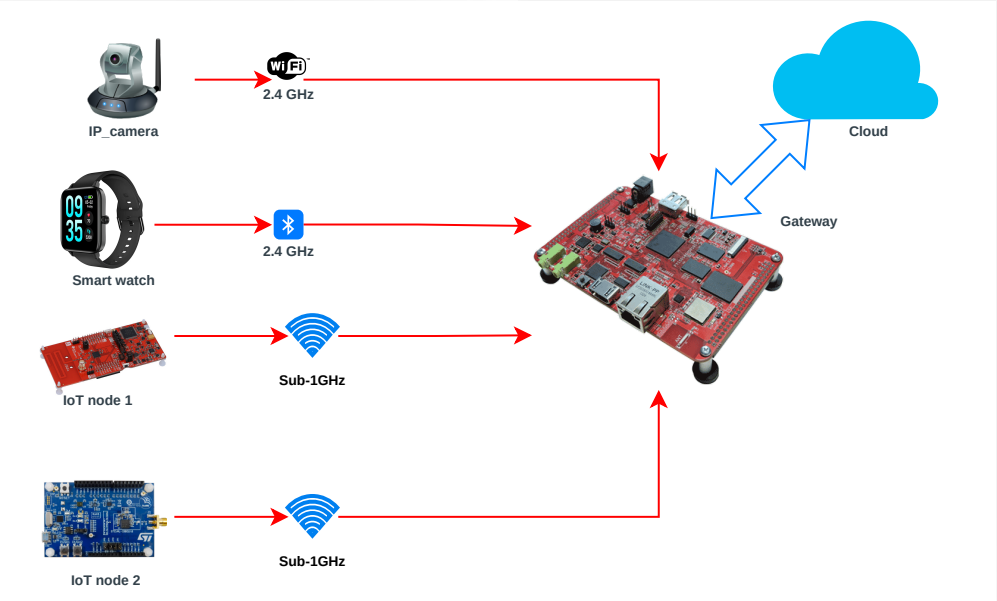


Figure: IoT architecture

SoC for IoT overview

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

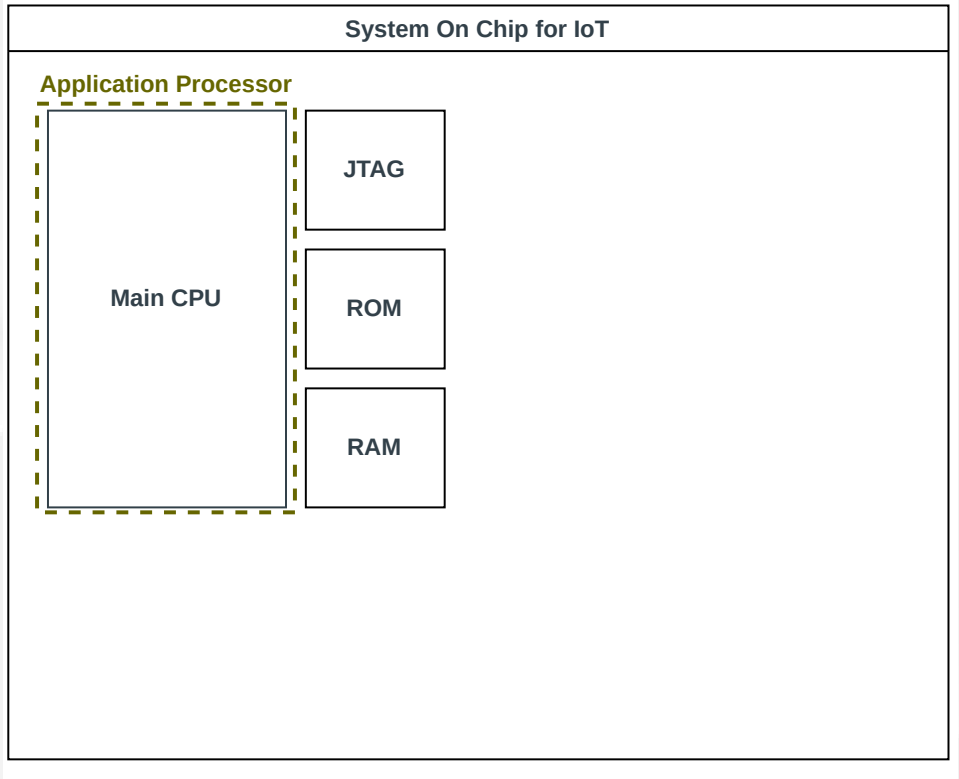


Figure: SoC IoT overview

Don’t forget that SoC are integrating a wireless connectivity unit!

SoC for IoT overview

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

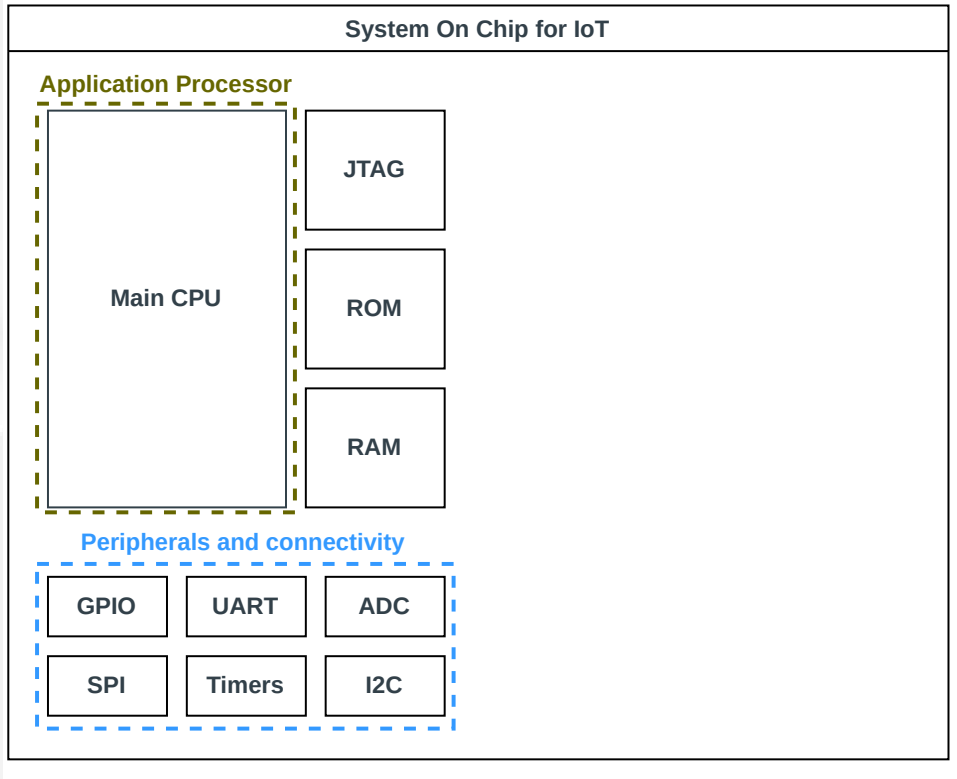


Figure: SoC IoT overview

Don’t forget that SoC are integrating a wireless connectivity unit!

SoC for IoT overview

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

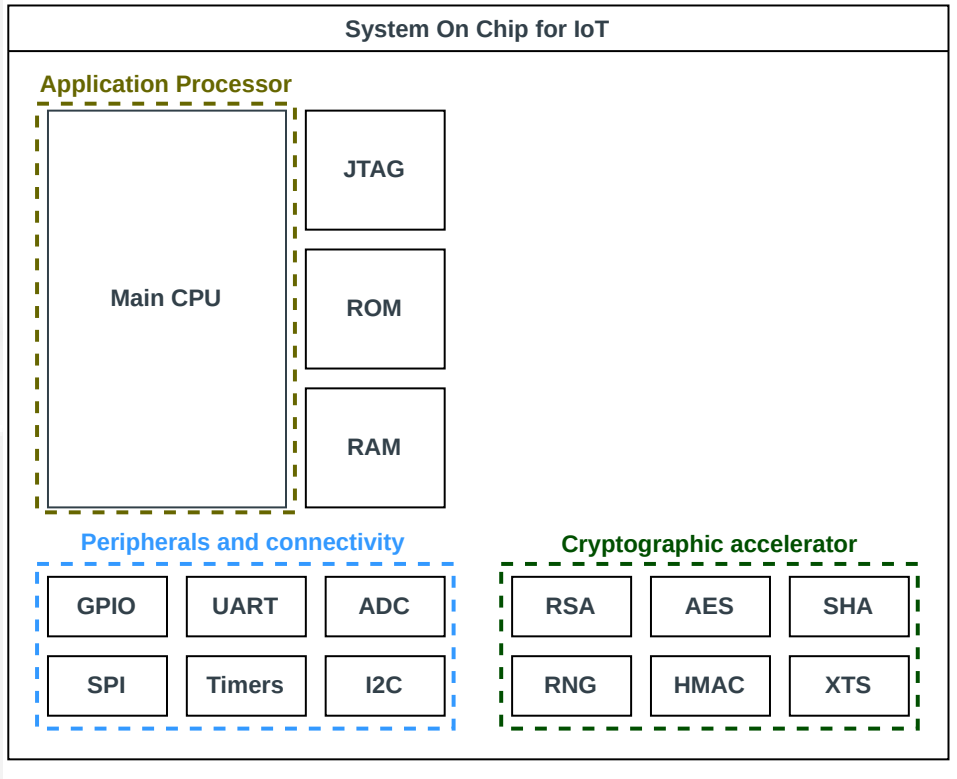


Figure: SoC IoT overview

Don’t forget that SoC are integrating a wireless connectivity unit!

SoC for IoT overview

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

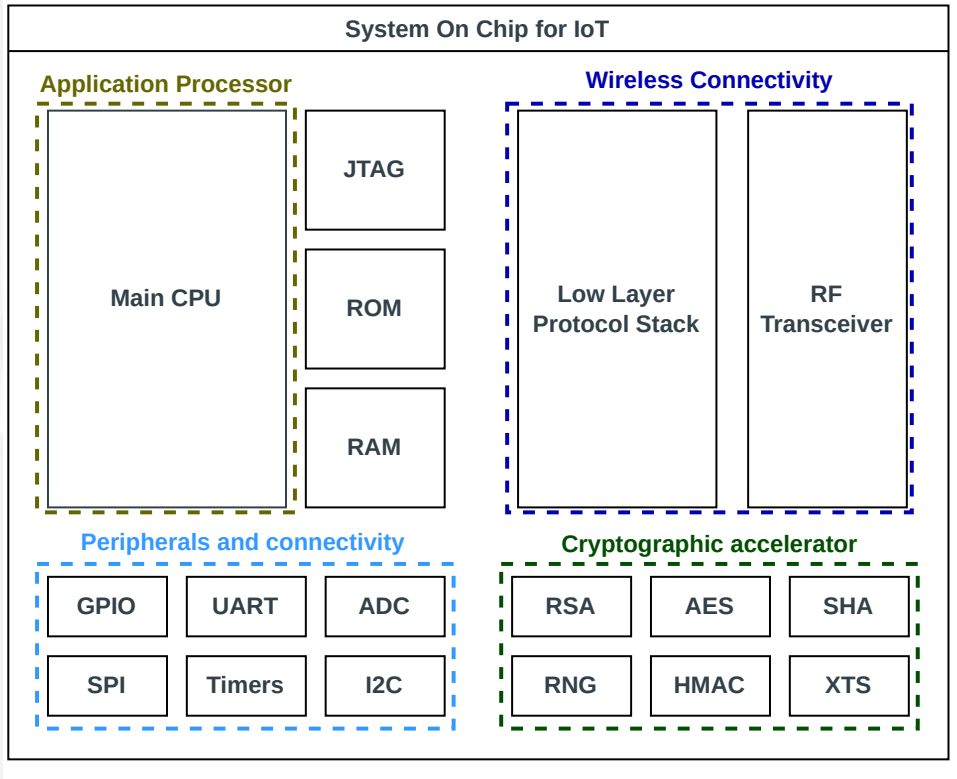


Figure: SoC IoT overview

Don’t forget that SoC are integrating a wireless connectivity unit!

Outline

1. Threat model and countermeasures
2. Proposed security mechanism: a multi-metrics HIDS
3. Test-bed & Evaluation

Outline

1. **Threat model and countermeasures**
   - Threat Model
   - Vulnerabilities in IoT
   - Attack examples
   - Attacks
   - Security mechanisms & mitigation
2. **Proposed security mechanism: a multi-metrics HIDS**
3. **Test-bed & Evaluation**

Threat Model

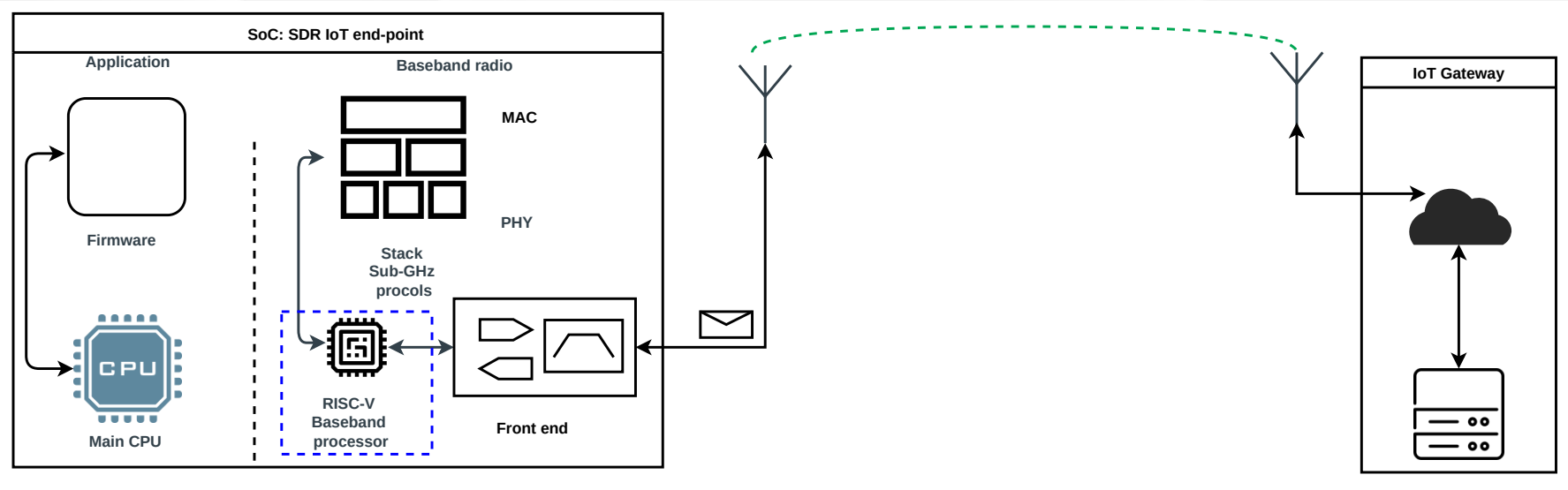


Figure: Potential Threat Model

Target : Remote Attacks

- Jamming Attack
- Logical Attacks: Packet Injection, . . .

Threat Model

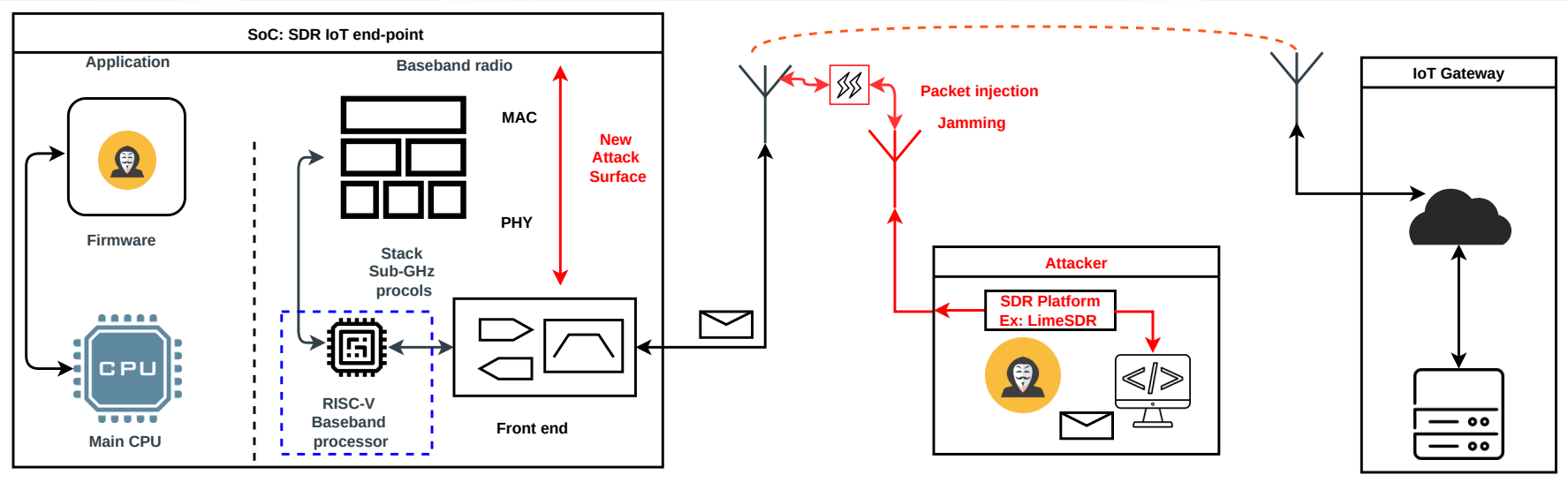


Figure: Potential Threat Model

Target : Remote Attacks

- Jamming Attack
- Logical Attacks: Packet Injection, . . .

Threat Model

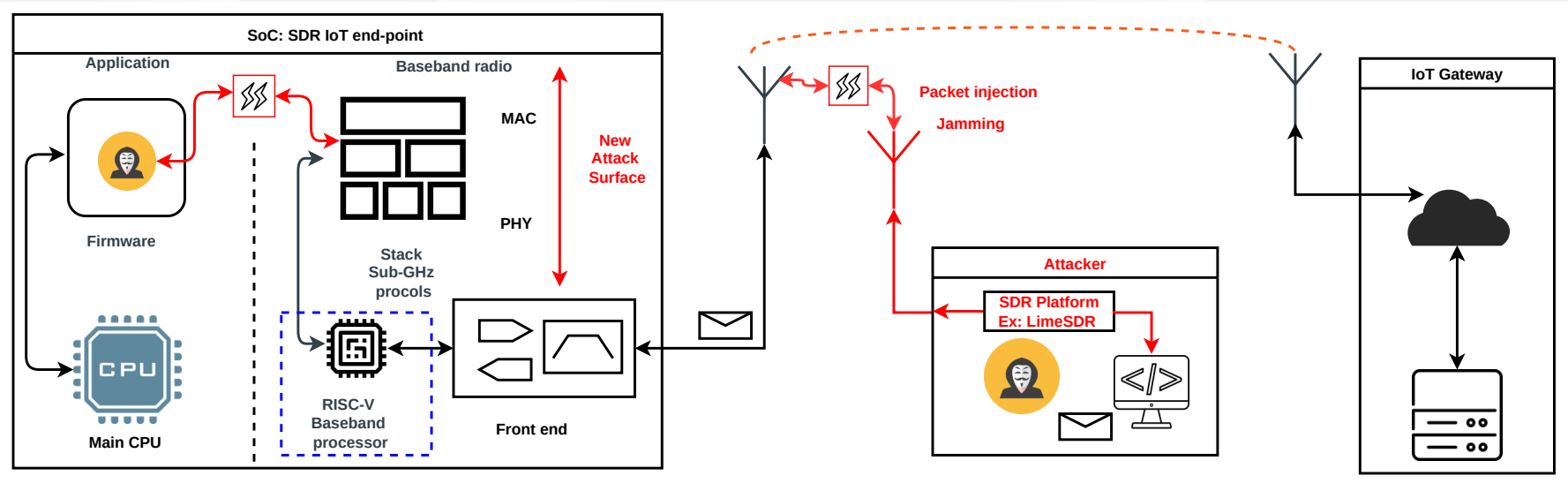


Figure: Potential Threat Model

Target : Remote Attacks

- Jamming Attack
- Logical Attacks: Packet Injection, . . .

Vulnerabilities in IoT

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

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

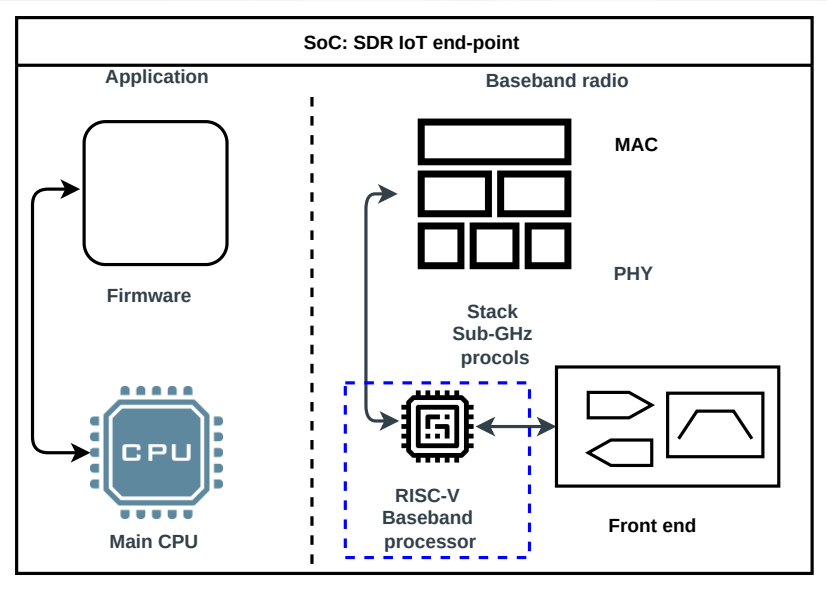


Figure: SoC for IoT with wireless connectivity

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

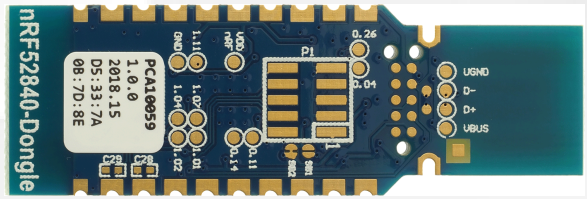


Figure: nRF52840-dongle : `https://www.nordicsemi.com/`

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_maste)

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

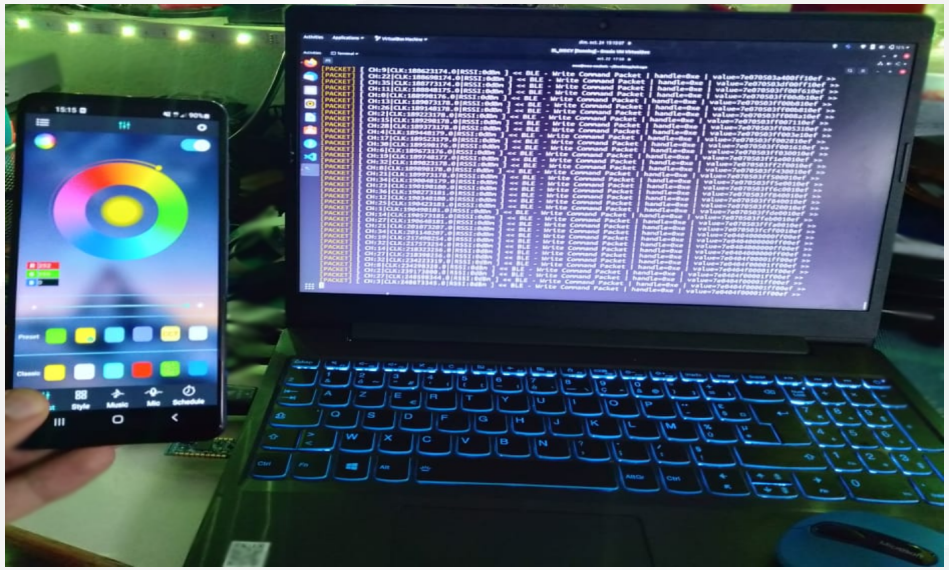


Figure: Sniffing packet exploit

Example of Exploit: Packet Injection

After hijacking the BLE Master we perform a packet injection exploit

```
[PACKET] [ CH:34|CLK:225120999.0|RSSI:0dBm ] << BLE - Find Information Request Packet >>
[PACKET] [ CH:2|CLK:225128729.0|RSSI:0dBm ] << BLE - Find Information Response Packet | format=0x1 | data=0b000229 >>
[PACKET] [ CH:7|CLK:225135999.0|RSSI:0dBm ] << BLE - Read By Type Request >>
[PACKET] [ CH:12|CLK:225143729.0|RSSI:0dBm ] << BLE - Error Response Packet | req=0x8 | handle=0xc | ecode=0xa >>
[PACKET] [ CH:17|CLK:225150999.0|RSSI:0dBm ] << BLE - Read By Type Request >>
[PACKET] [ CH:22|CLK:225158729.0|RSSI:0dBm ] << BLE - Read By Type Response | data=070d00060e00f3ff0f00101000f4ff >>
[PACKET] [ CH:27|CLK:225165999.0|RSSI:0dBm ] << BLE - Read By Type Request >>
[PACKET] [ CH:5|CLK:225188729.0|RSSI:0dBm ] << BLE - Error Response Packet | req=0x8 | handle=0x10 | ecode=0xa >>
[PACKET] [ CH:10|CLK:225195999.0|RSSI:0dBm ] << BLE - Find Information Request Packet >>
[PACKET] [ CH:20|CLK:225211229.0|RSSI:0dBm ] << BLE - Error Response Packet | req=0x4 | handle=0x11 | ecode=0xa >>
[PACKET] [ CH:25|CLK:225218499.0|RSSI:0dBm ] << BLE - Control PDU Packet | type=LL_CONNECTION_UPDATE_REQ | data=01000024000000f4012f00 >>
[PACKET] [ CH:23|CLK:225308499.0|RSSI:0dBm ] << BLE - Read Request Packet | handle=0xe >>
[PACKET] [ CH:1|CLK:225443731.0|RSSI:0dBm ] << BLE - Read Response Packet | value=454c4b39364b335232345632363400000000000 >>
[PACKET] [ CH:6|CLK:225488500.0|RSSI:0dBm ] << BLE - Write Command Packet | handle=0xe | value=7e07830f321800ffef >>
[PACKET] [ CH:4|CLK:227468745.0|RSSI:0dBm ] << BLE - Connection Parameter Update Request Packet | slaveLatency=1 | timeoutMult=100 | minInterval=40 | maxInterval=6
0 >>
[PACKET] [ CH:9|CLK:227513514.0|RSSI:0dBm ] << BLE - Control PDU Packet | type=LL_CONNECTION_UPDATE_REQ | data=0100003c00010064006700 >>
[INFO] Starting Master Hijacking attack: injecting LL_CONNECTION_UPDATE_REQ...
[SUCCESS] Injection successful after 2 attempts !
[INFO] Waiting for connection update instant...
[SUCCESS] Attack successful !
[INFO] SubInterface available: butterfly0:sub0 (master)
[INFO] Instantiating subdevice :butterfly0:sub0
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503ff00c110ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503ff00c110ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503ff00c110ee
[SUCCESS] Write Command : handle = 0xe / value = 7e070503ff00c110ee
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503ff00c110ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503ff00c110ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503002aff10ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503002aff10ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503ff00c110ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503ff00c110ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503ff00c110ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503ff00c110ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503002aff10ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503002aff10ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503002aff10ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503002aff10ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503002aff10ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503002aff10ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503ff00c110ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503ff00c110ef
[MASTER|0xe4773f5d]: write_cmd 0xe 7e070503002aff10ef
[SUCCESS] Write Command : handle = 0xe / value = 7e070503002aff10ef
[MASTER|0xe4773f5d]:
```

Figure: Packet Injection exploit

Attacks in IoT

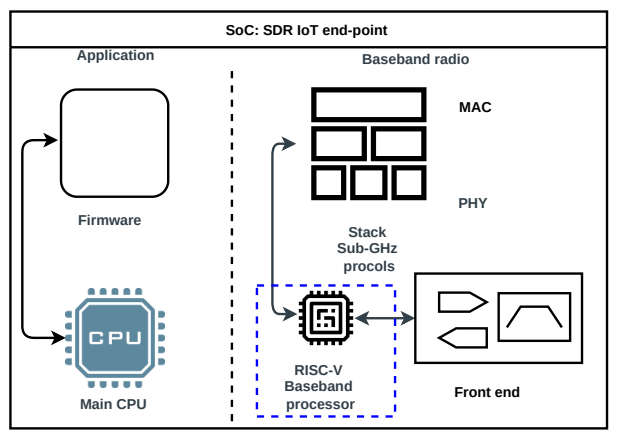


Figure: SoC for IoT

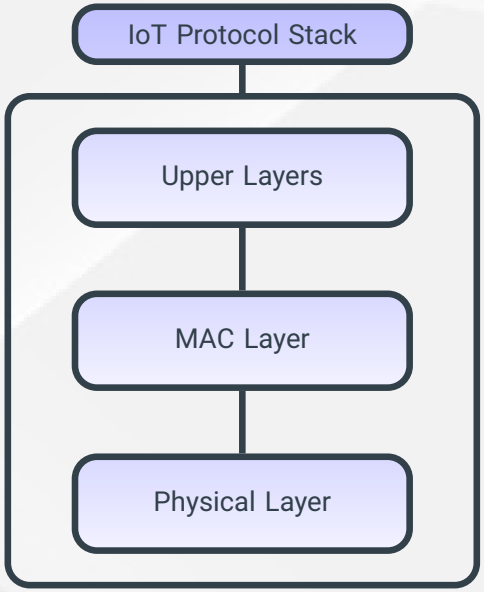


Figure: IoT protocol stack layers

E (Exploited Layer) **T** (Targeted Layer)

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

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

Attacks in IoT

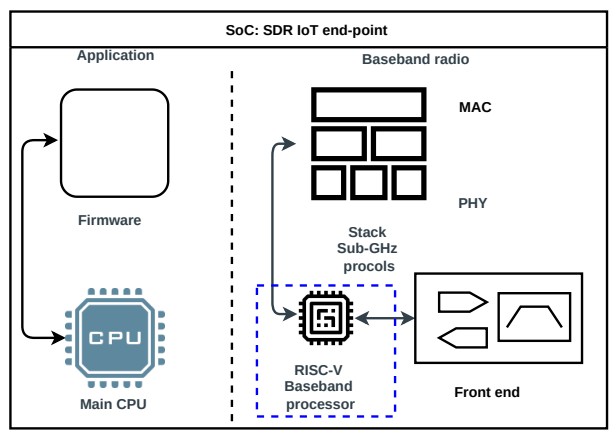


Figure: SoC for IoT

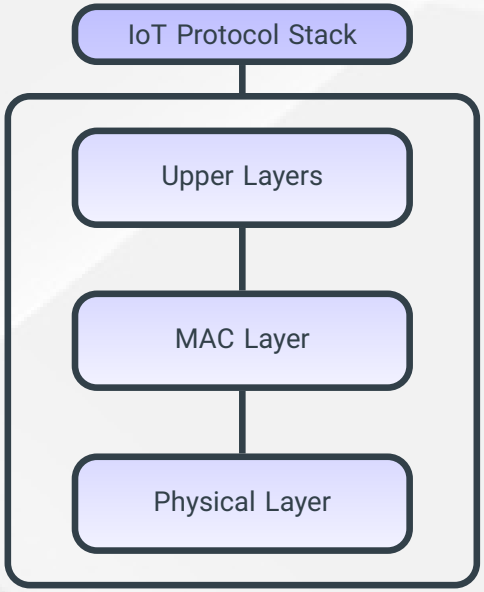


Figure: IoT protocol stack layers

E (Exploited Layer) **T** (Targeted Layer)

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

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

Security mechanisms & mitigation

Features	CC1356	CC1352R1	STM32WL54CC
Sec. Boot (protection)	✓	✓	✓
Cryptography (protection)	✓	✓	✓
OTA (Update)	✓	✓	✓
Heap ASLR (protection)	✗	✗	✗
Monitoring (detection)	✗	✗	✗
DIFT (hard. monitor)	✗	✗	✗
Code instrumentation (protection)	✗	✗	✗
Anomaly/Intrusion detection	✗	✗	✗

Table: Platform security features comparison

Security Mechanisms

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

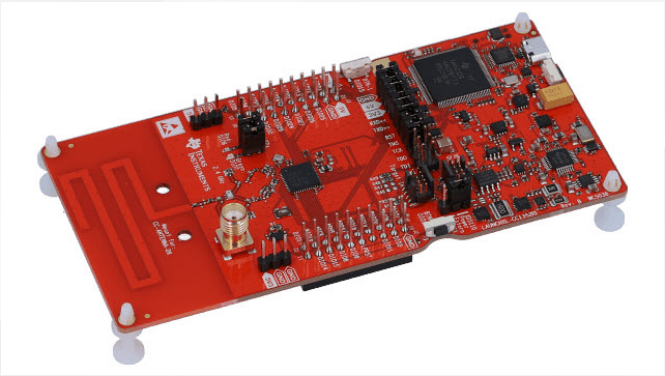


Figure: CC1352R1 : SoC for IoT

Outline

1. Threat model and countermeasures
2. **Proposed security mechanism: a multi-metrics HIDS**
   - Motivation and contribution
   - Intrusion Detection System Taxonomy
   - Host based IDS in state of the art
   - Towards a multi-level metrics HIDS
3. Test-bed & Evaluation

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

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

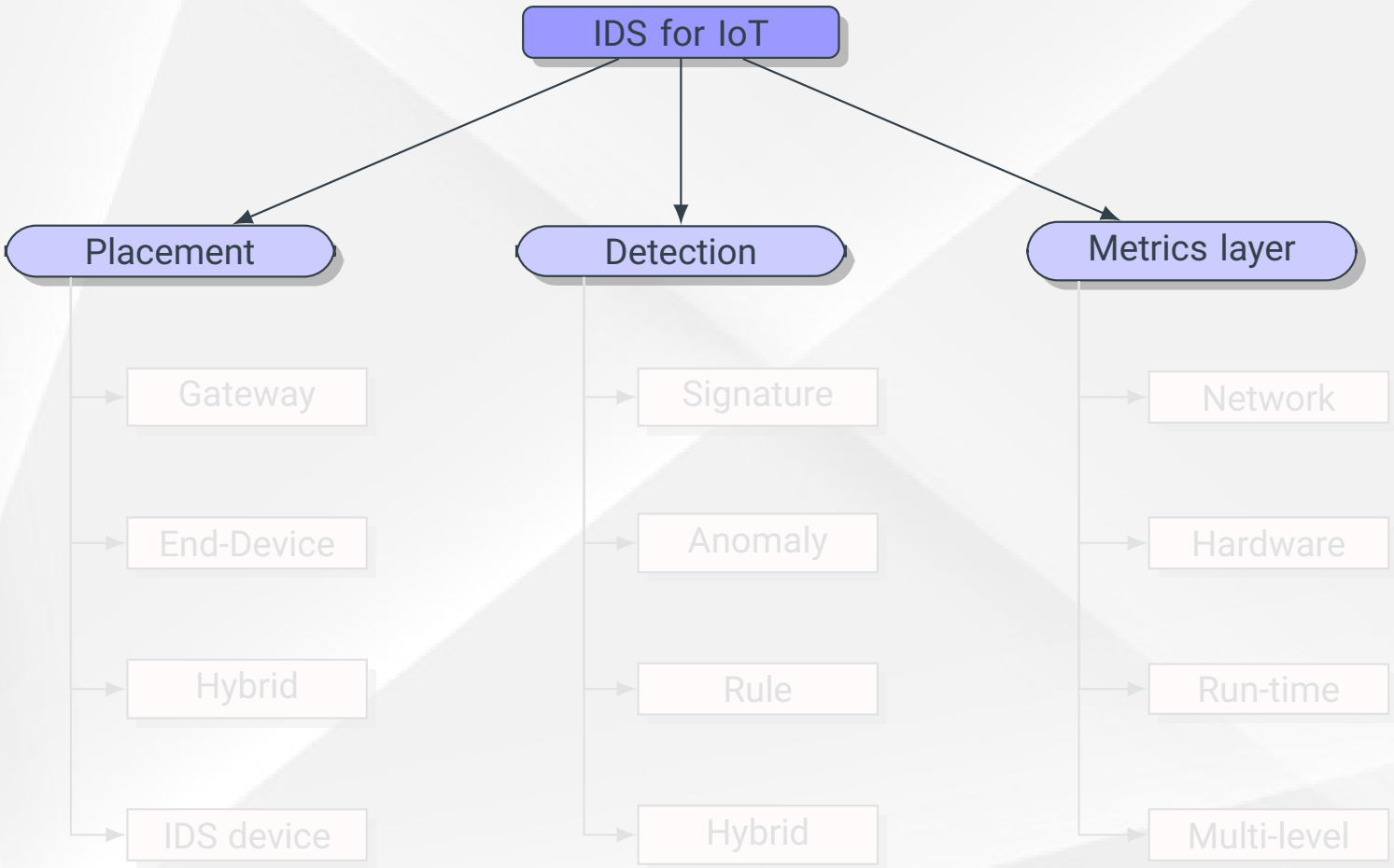


Figure: IDS taxonomy for IoT environment

IDS taxonomy

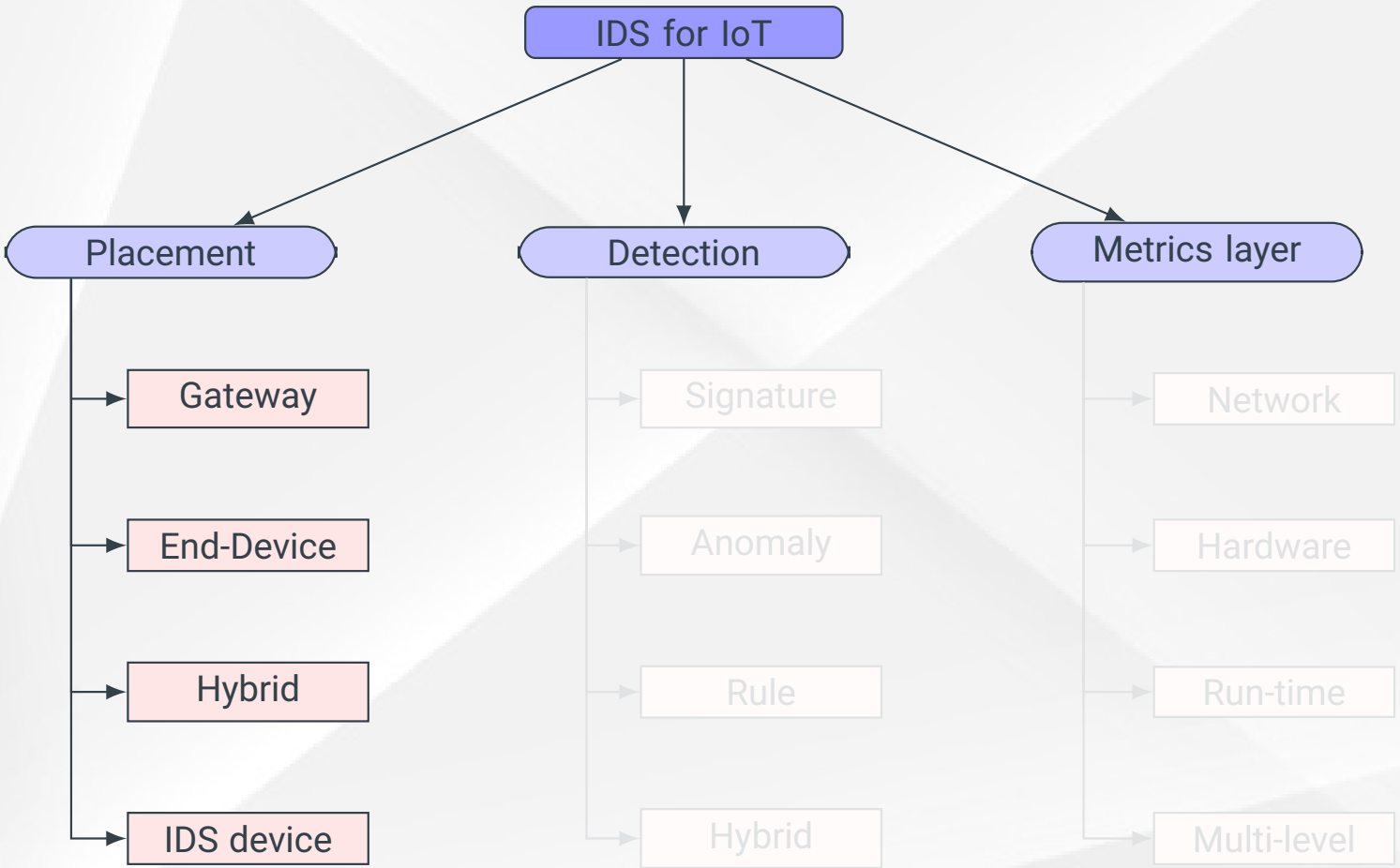


Figure: IDS taxonomy for IoT environment

IDS taxonomy

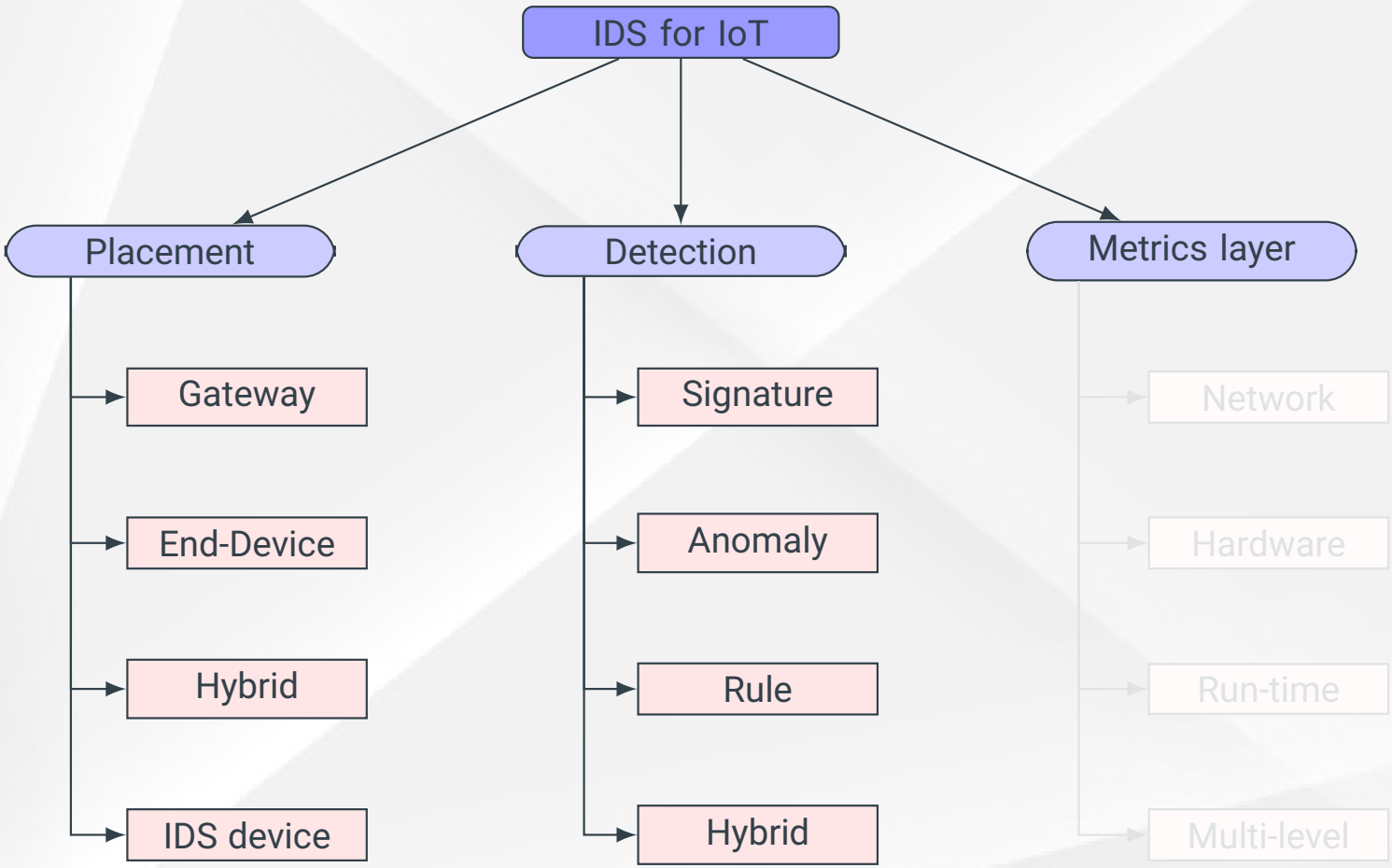


Figure: IDS taxonomy for IoT environment

IDS taxonomy

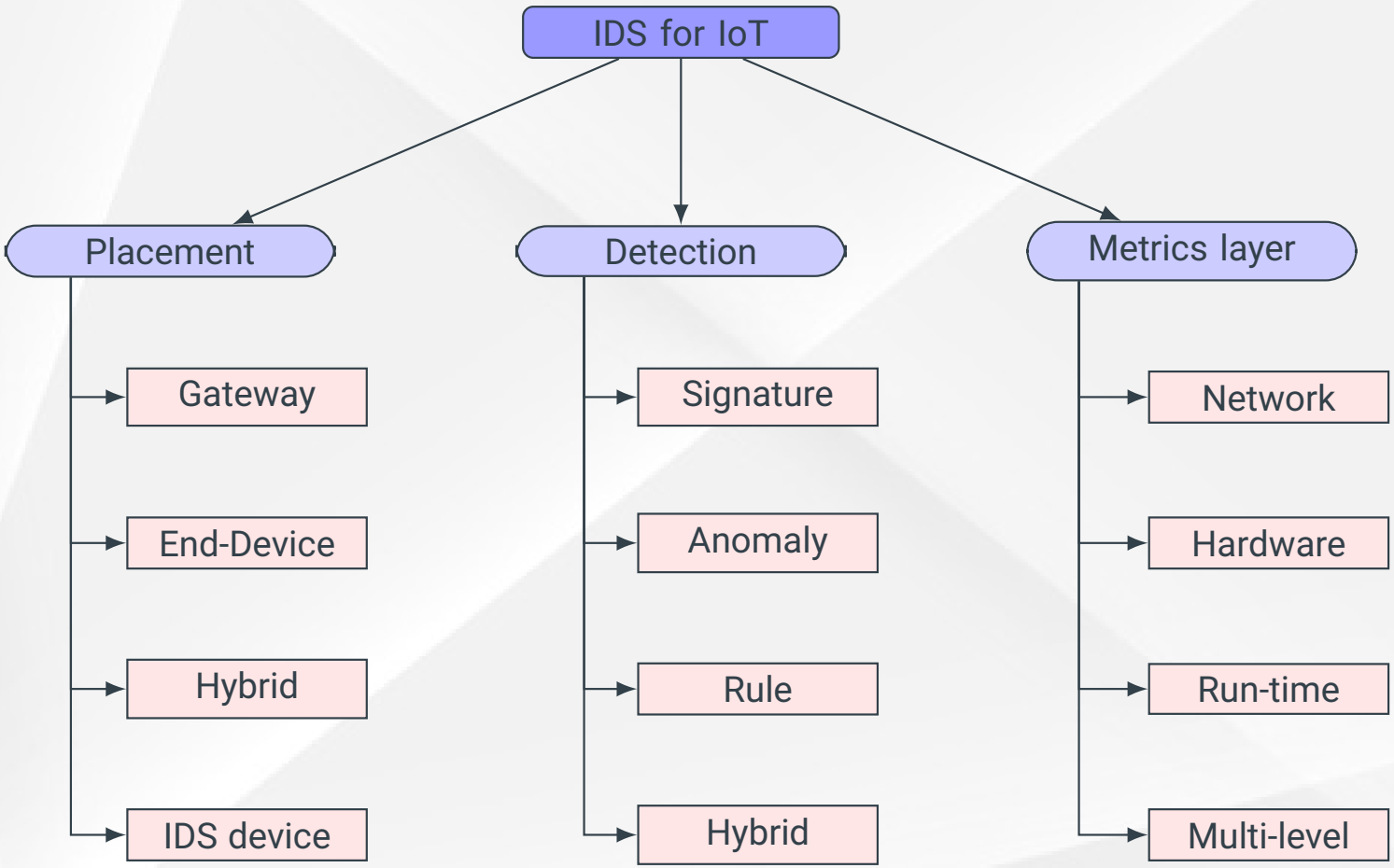


Figure: IDS taxonomy for IoT environment

Host based IDS in state of the art

What are the accurate metrics to record for an HIDS?

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

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

Host based IDS in state of the art

What are the accurate metrics to record for an HIDS?

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

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

Host based IDS in state of the art

What are the accurate metrics to record for an HIDS?

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

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

Host based IDS in state of the art

What are the accurate metrics to record for an HIDS?

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

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

Host based IDS in state of the art

What are the accurate metrics to record for an HIDS?

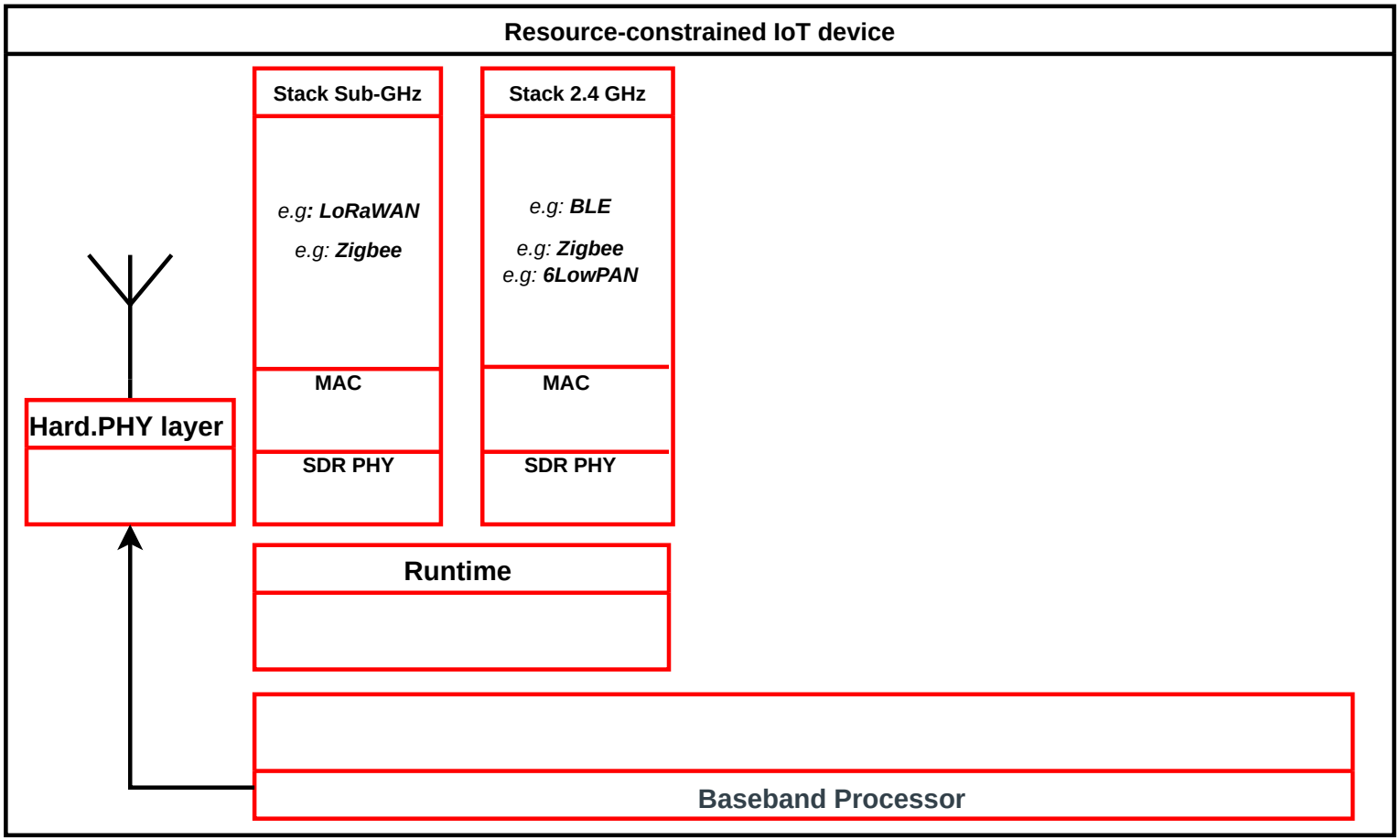
Ref	PHY	MAC	UL	μProc	RT	Target	PS	DM	Place
[Yan et al., 2020]	RSSI	-	-	-	-	Spoof	Model legiti.RSSI	B	G / RC
[Zhang et al., 2013]	RSSI	TS	TS	-	-	integrity	SDR	B	D
[Sousa et al., 2017]	-	P	-	-	-	DoS	Analyze & store	S	RC
[Kasinathan et al., 2013]	-	P	-	-	-	DoS, Jamm	SURICATA	S	D
[Eskandari et al., 2020]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

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)

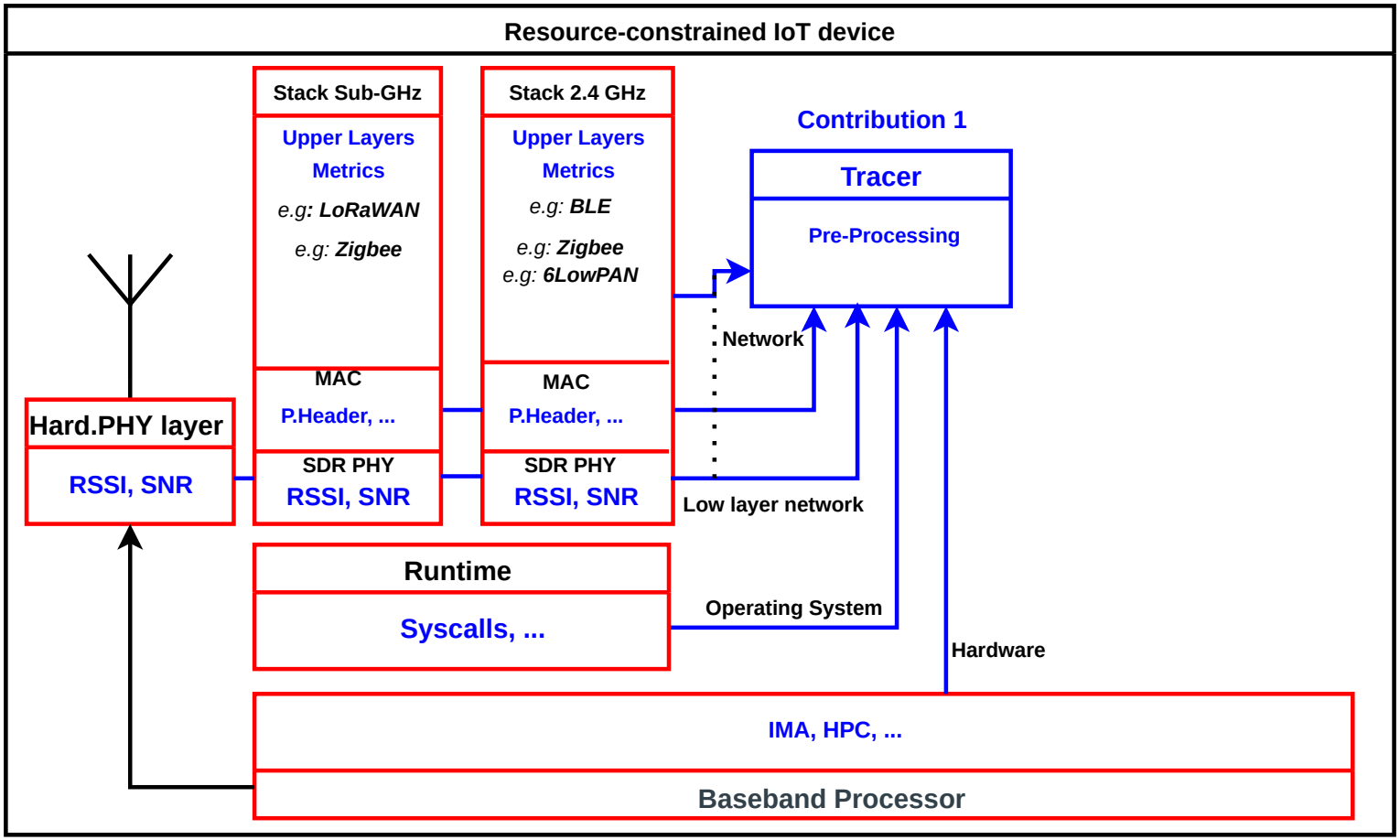
The multi-level approach is not yet addressed in the state of the art

Towards a multi-level metrics HIDS



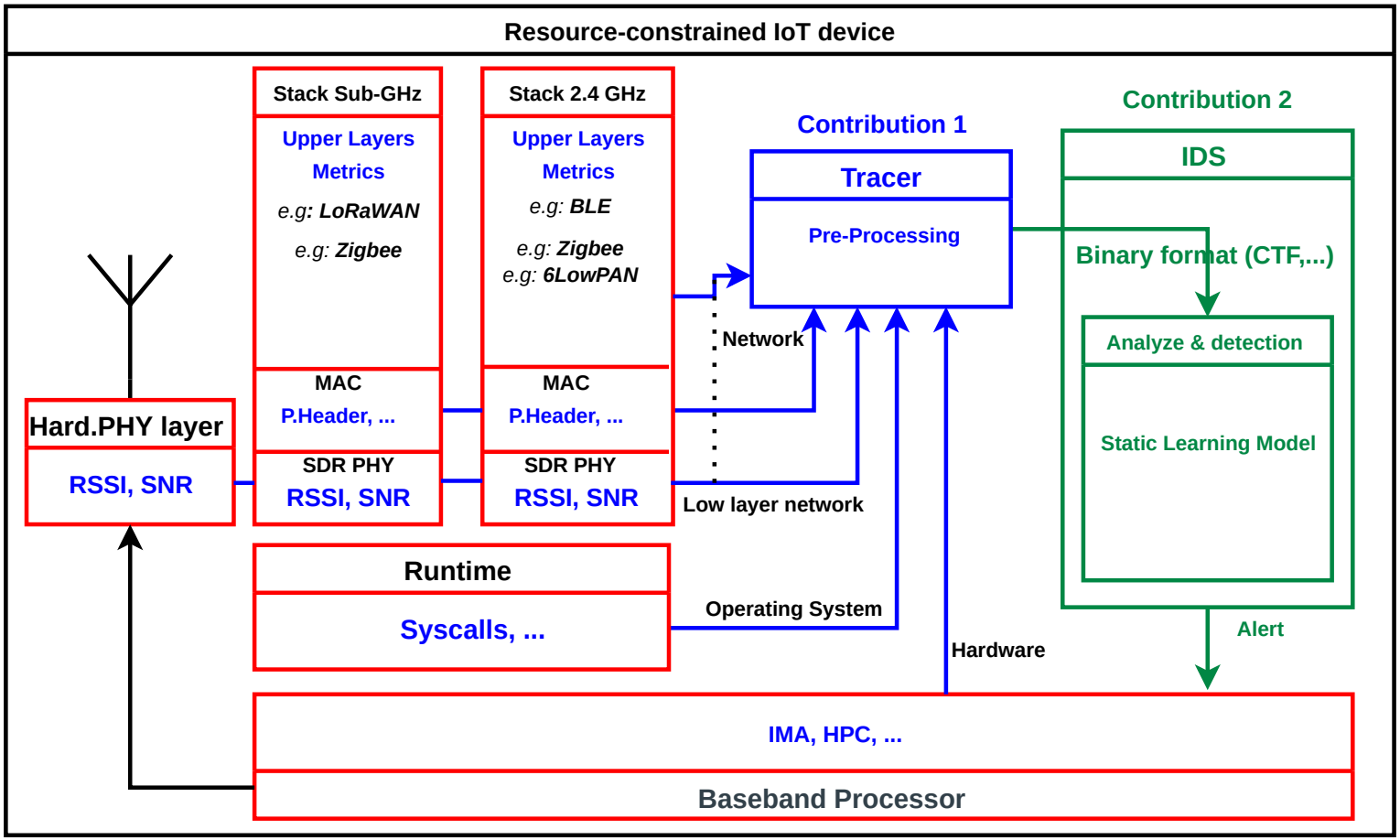
Wireless connectivity block diagram with IDS

Towards a multi-level metrics HIDS



Wireless connectivity block diagram with IDS

Towards a multi-level metrics HIDS



Wireless connectivity block diagram with IDS

Outline

1. Threat model and countermeasures
2. Proposed security mechanism: a multi-metrics HIDS
3. **Test-bed & Evaluation**
   - Objective
   - Test-bed
   - Preliminary results
   - Conclusion

Objective

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

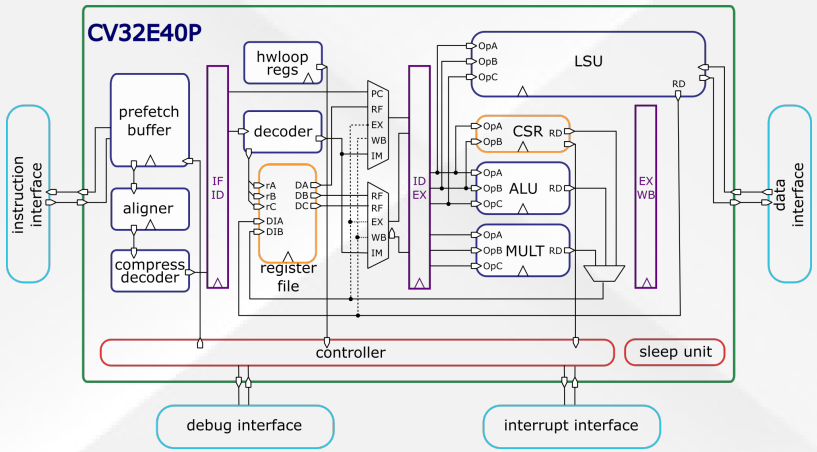


Figure: CV32E41P/40P block diagram

- **Scenario**
  - Reproduction of simple buffer overflow exploit on software running on wireless connectivity area
  - Build a dataset of HPC values per each packet network

Objective

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

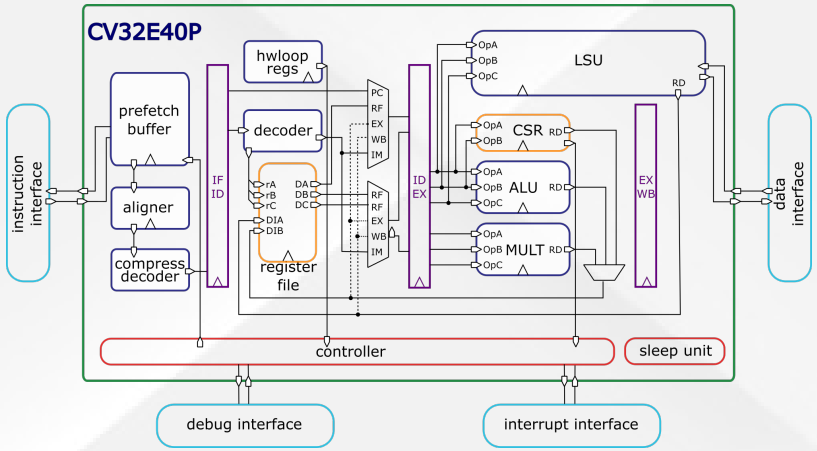


Figure: CV32E41P/40P block diagram

- **Scenario**
  - Reproduction of simple buffer overflow exploit on software running on wireless connectivity area
  - Build a dataset of HPC values per each packet network

Objective

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

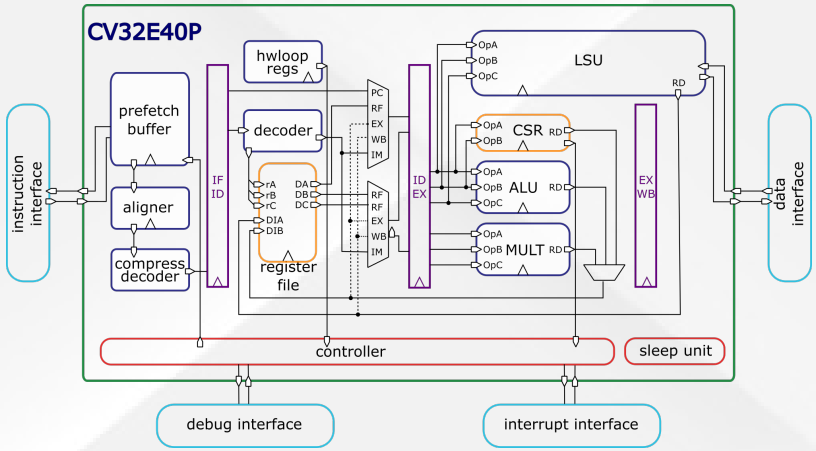


Figure: CV32E41P/40P block diagram

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

Objective

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

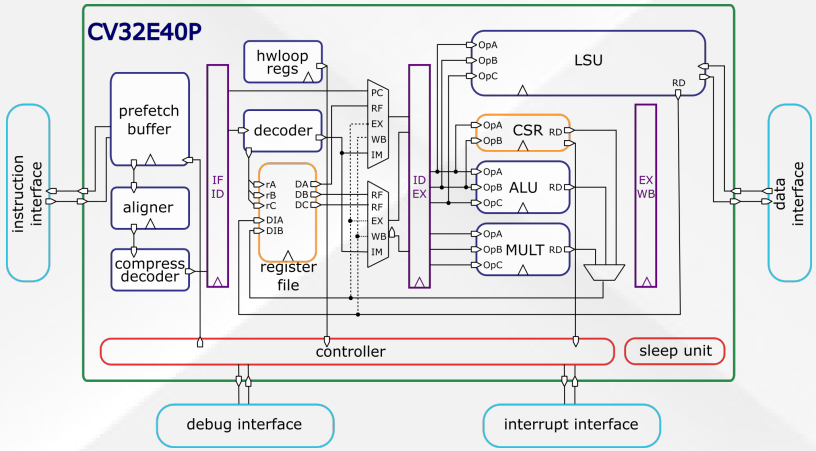


Figure: CV32E41P/40P block diagram

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

Test-bed with tracing metrics from RISC-V CV32E41P

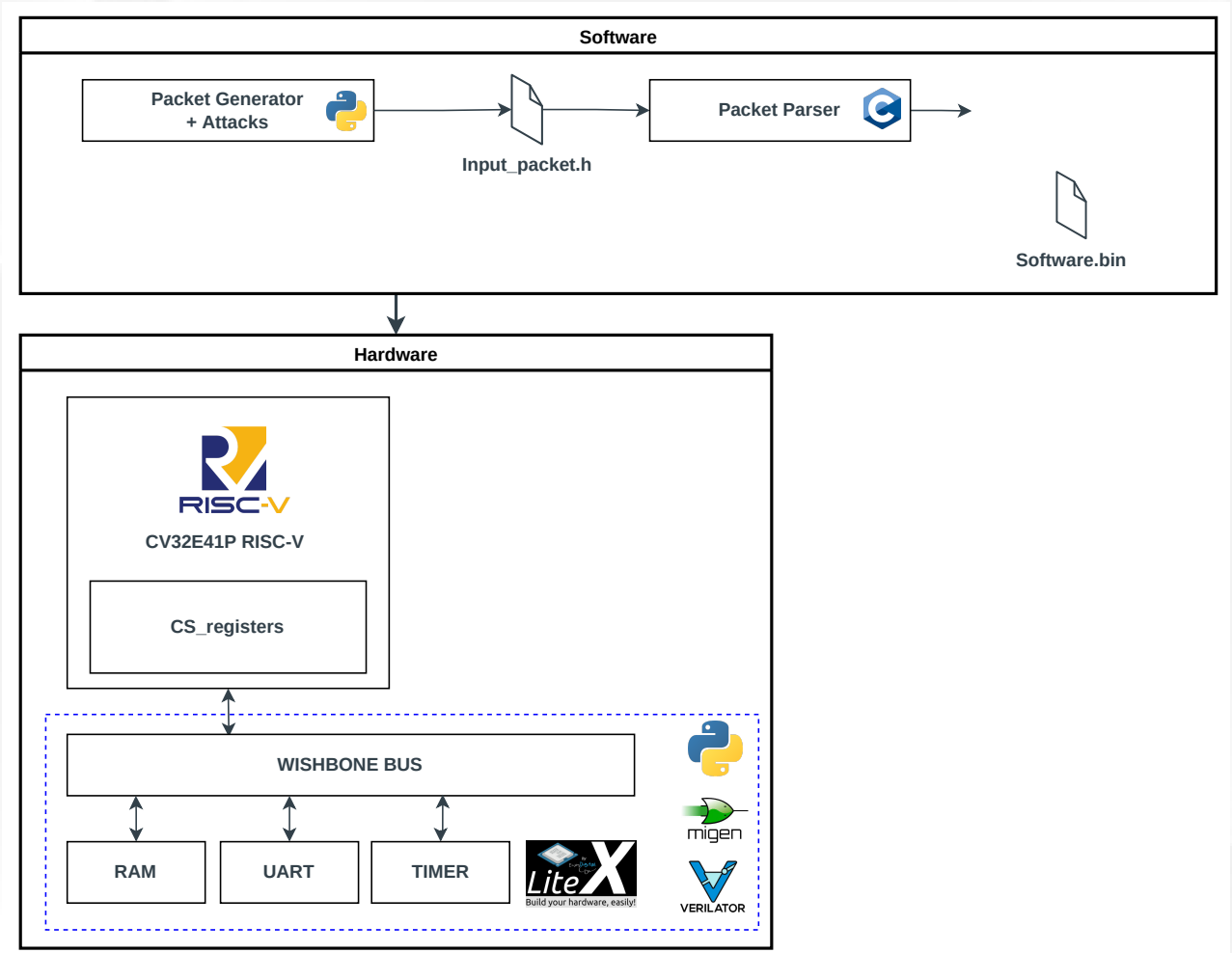


Figure: Test-bed block diagram

Test-bed with tracing metrics from RISC-V CV32E41P

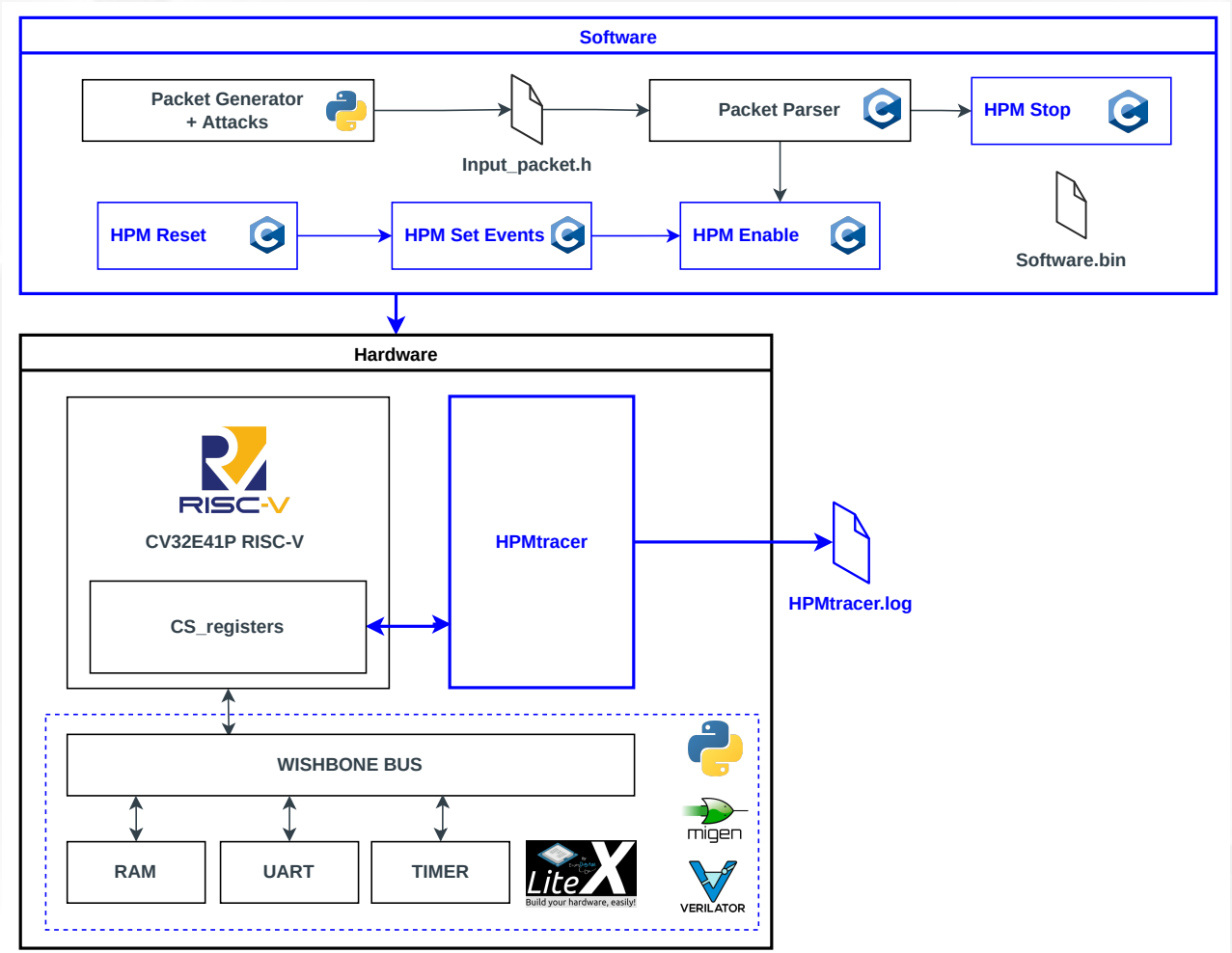


Figure: Test-bed block diagram

Test-bed with tracing metrics from RISC-V CV32E41P

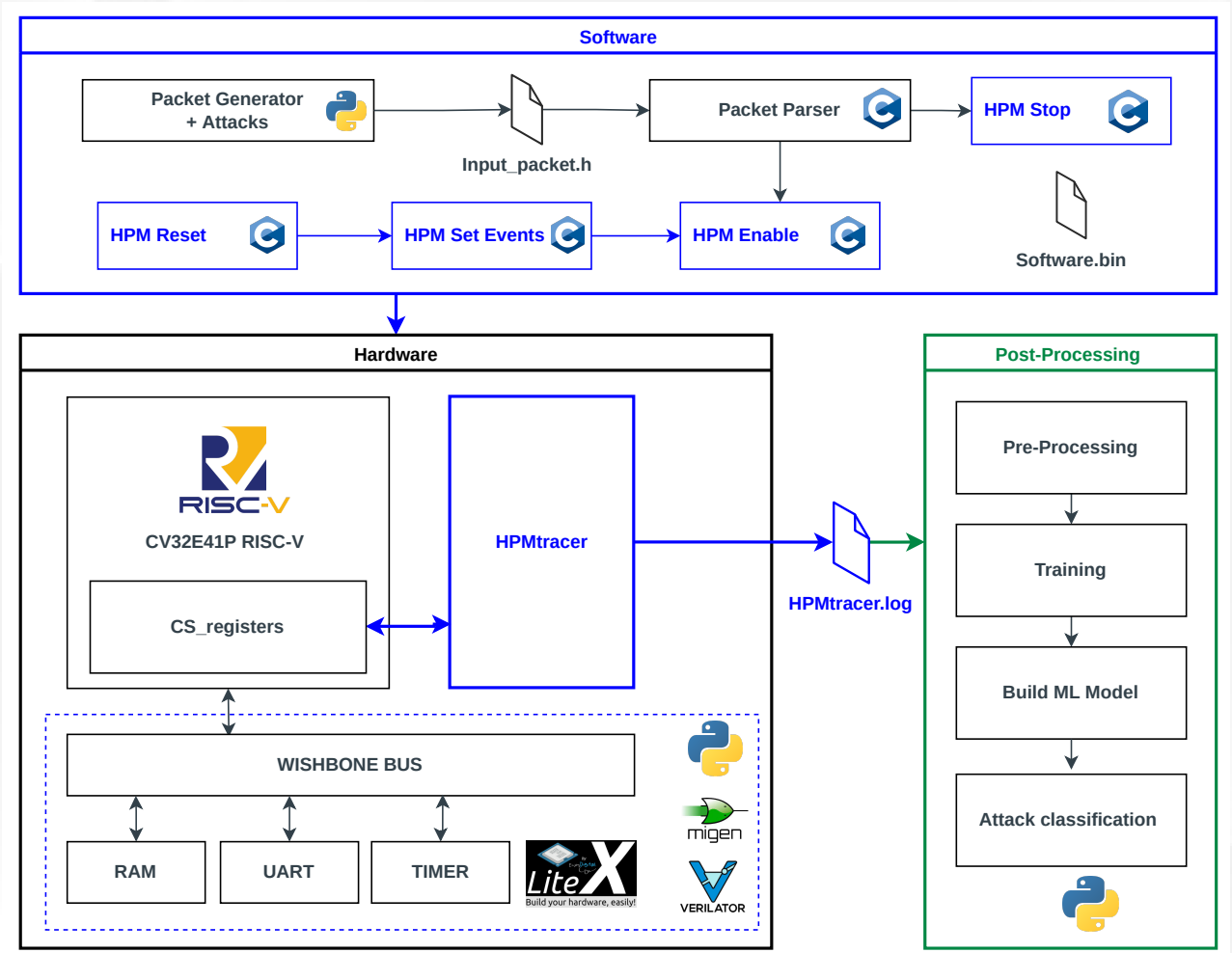


Figure: Test-bed block diagram

Dataset

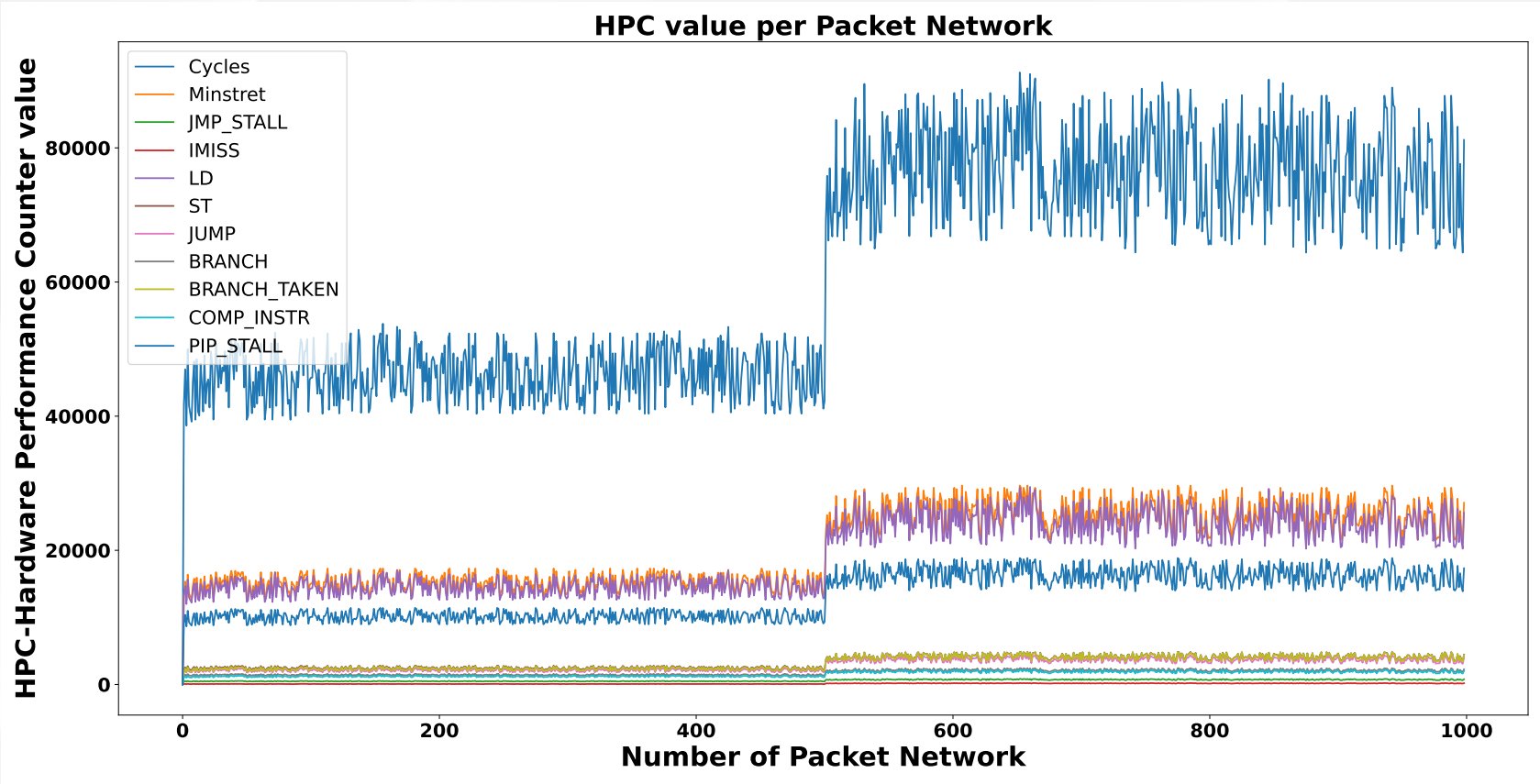


Figure: Dataset from HPC monitors

Preliminary results

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

Method	Accuracy	Precision	Recall	F1 score
Nearest Neighbors	0.998	0.995	1.00	0.998
Linear SVM	0.998	0.995	1.00	0.998
RBF SVM	0.765	1.000	0.550	0.710
Gaussian Process	0.887	1.000	0.785	0.879
Decision Tree	0.998	0.995	1.000	0.998
Random Forest	0.998	0.995	1.000	0.998
Neural Net	0.583	0.977	0.206	0.340
AdaBoost	0.998	0.995	1.000	0.998
Naive Bayes	0.995	0.995	0.995	0.995
QDA	0.995	0.995	0.995	0.995

- Interesting Results
- An in-depth study to follow

Conclusion

- **Ongoing work**
  - New approach for monitoring and detecting software attacks from a network entry point.
  - Test-bed to detect buffer overflow using hardware counters.
  - Promising results of machine learning classification algorithms.
- **Future work**
  - Tracer Implementation.
  - Lightweight IDS Detection Module on Co-processor.
  - Tracer & IDS Evaluation (Detection, Benchmarks, Overhead, Power consumption).

THANK YOU

Towards Low-Power and Low Data-Rate Software-Defined Radio Baseband with RISC-V Processor for Flexibility and Security

18th CryptArchi Workshop - Porquerolles 2022

Porquerolles, France May 29- June 1, 2022

Mohamed EL-BOUAZZATI, Philippe TANGUY, Guy GOGNIAT

Lab-STICC, Team ARCAD, Université Bretagne Sud, Lorient, France

[firstname].[lastname]@univ-ubs.fr

References (1/6)

[ten, 2020] (2020).
Loradawn - multiple lorawan security vulnerabilities.

[Antonioli et al., 2020] Antonioli, D., Tippenhauer, N. O., and Rasmussen, K. (2020).
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.

[Avoine and Ferreira, 2018] Avoine, G. and Ferreira, L. (2018).
Rescuing LoRaWAN 1.0.
In *Financial Cryptography and Data Security: 22nd International Conference, FC 2018*, Nieuwpoort, Curaçao.

References (2/6)

[Bourdon et al., 2021] Bourdon, M., Gimenez, P.-f., Alata, E., Kaâniche, M., Migliore, V., Nicomette, V., Laarouchi, Y., Bourdon, M., Gimenez, P.-f., Alata, E., Kaâniche, M., Migliore, V., Bourdon, M., and Edf, R. (2021).
Hardware-Performance-Counters-based anomaly detection in massively deployed smart industrial devices To cite this version : HAL Id : hal-03328251 Hardware-Performance-Counters-based anomaly detection in massively deployed smart industrial devices.

[Breitenbacher et al., 2019] Breitenbacher, D., Homoliak, I., Aung, Y. L., Tippenhauer, N. O., and Elovici, Y. (2019).
HADES-IoT: A practical host-based anomaly detection system for iot devices.
AsiaCCS 2019 - Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, pages 479–484.

[Cayre et al.,] Cayre, R., Galtier, F., Auriol, G., Nicomette, V., Cayre, R., Galtier, F., Auriol, G., Nicomette, V., Kaâniche, M., Cayre, R., Galtier, F., Auriol, G., Nicomette, V., and Ka^, M.
WazaBee : attacking Zigbee networks by diverting Bluetooth Low Energy chips To cite this version : HAL Id : hal-03193299 WazaBee : attacking Zigbee networks by diverting Bluetooth Low Energy chips.

References (3/6)

[Cayre et al., 2021] Cayre, R., Galtier, F., Auriol, G., Nicomette, V., Kaaniche, M., and Marconato, G. (2021).
InjectaBLE: Injecting malicious traffic into established Bluetooth Low Energy connections.
Proceedings - 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2021, pages 388–399.

[Eskandari et al., 2020] Eskandari, M., Janjua, Z. H., Vecchio, M., and Antonelli, F. (2020).
Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices.
IEEE Internet of Things Journal, 7(8):6882–6897.

[Gassais et al., 2020] Gassais, R., Ezzati-Jivan, N., Fernandez, J. M., Aloise, D., and Dagenais, M. R. (2020).
Multi-level host-based intrusion detection system for Internet of things.
Journal of Cloud Computing, 9(1).

[Hessel et al.,] Hessel, F., Almon, L., and Álvarez, F.
ChirpOTLE: A framework for practical LoRaWAN security evaluation.
pages 306–316.

References (4/6)

[Kasinathan et al., 2013] Kasinathan, P., Costamagna, G., Khaleel, H., Pastrone, C., and Spirito, M. A. (2013).
Demo: An IDS framework for internet of things empowered by 6LoWPAN.
Proceedings of the ACM Conference on Computer and Communications Security, pages 1337–1339.

[Labs, 2020] Labs, F. R. (2020).
Amnesia:33, how tcp/ip stacks breed critical vulnerabilities in iot, ot and it devices.

[Raza et al., 2013] Raza, S., Wallgren, L., and Voigt, T. (2013).
SVELTE: Real-time intrusion detection in the Internet of Things.
Ad Hoc Networks, 11(8):2661–2674.

[Saeed et al., 2016] Saeed, A., Ahmadinia, A., Javed, A., and Larijani, H. (2016).
Intelligent intrusion detection in low-power IoTs.
ACM Transactions on Internet Technology, 16(4).

[Santos et al., 2019] Santos, A. C., Filho, J. L., Silva, Á. Í., Nigam, V., and Fonseca, I. E. (2019).
BLE injection-free attack: a novel attack on bluetooth low energy devices.
Journal of Ambient Intelligence and Humanized Computing, (0123456789).

References (5/6)

[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.

[Sousa et al., 2017] Sousa, B. F. L. M., Soeiro, N. C., Abdelouahab, Z., Ribeiro, W. F., and Ribeiro, D. C. P. (2017).
An intrusion detection system for denial of service attack detection in internet of things.
ACM International Conference Proceeding Series.

[Yan et al., 2020] Yan, W., Hylamia, S., Voigt, T., and Rohner, C. (2020).
PHY-IDS: A physical-layer spoofing attack detection system for wearable devices.
WearSys 2020 - Proceedings of the 6th ACM Workshop on Wearable Systems and Applications, Part of MobiSys 2020, pages 1–6.

[Zhang et al., 2013] Zhang, M., Raghunathan, A., and Jha, N. K. (2013).
MedMon: Securing medical devices through wireless monitoring and anomaly detection.
IEEE Transactions on Biomedical Circuits and Systems, 7(6):871–881.

References (6/6)

[Zhang et al., 2020] Zhang, Y., Weng, J., Dey, R., Jin, Y., Lin, Z., and Fu, X. (2020). **Breaking secure pairing of bluetooth low energy using downgrade attacks.** In *29th USENIX Security Symposium (USENIX Security 20)*, pages 37–54. USENIX Association.