Attack Scenarios and Embedded Intrusion Detection for Space Systems

Louis Lolive, Guillaume Auriol, Vincent Nicomette,
Florent Galtier, Simone Urbano, Jacques Girard

IRT Saint Exupéry CSS Project
Cybersecurity for Space Systems

LAAS-CNRS
/ Laboratoire d'analyse et d'architecture des systèmes du CNRS

Laboratoire conventionné avec
Université de Toulouse

CSS Project

> CSS : Cybersecurity for Space Systems, IRT Saint Exupéry project
> Aim : simulate and detect cyberattacks on a satellite constellation

> Partners :

GATEWATCHER

ThalesAlenia Space
a Thales / Leonardo company

STARION

IRT SAINT EXUPÉRY

ÉCOLE DE L'AIR & DE L'ESPACE
SALON-DE-PROVENCE

cnes

LAAS CNRS

Image: https://www.atout-france.fr/en/mice/us/toulouse

Context

- Satellites
  - Strategic assets
  - Target of cyberattacks (*ex: VIASAT attack, OPS-SAT hack, Hack-a-Sat…)*
  - No or very poor embedded security mechanisms
- Strategies to improve security
  - Security by-design
  - **Intrusion Detection**
- Space context challenges
  - Limited Resources
  - Critical Systems
  - Periods of autonomy
  - Specific protocols

Embedded Intrusion Detection

State of the art

Intrusion detection in embedded systems

- AI-based methods (often Anomaly Detection)
- Non-AI methods
  - State automata
  - Heuristics
  - Histogram-based detection
  - …

In satellites context

- Existing works are theoretical
- Or based on non-specific hardware / attacks (e.g. CAN-BUS)

Constraints

- Datasets
  - Existing datasets: only ground data
  - No datasets with internal bus data

→ *Incompatible with on-bus detection approaches*

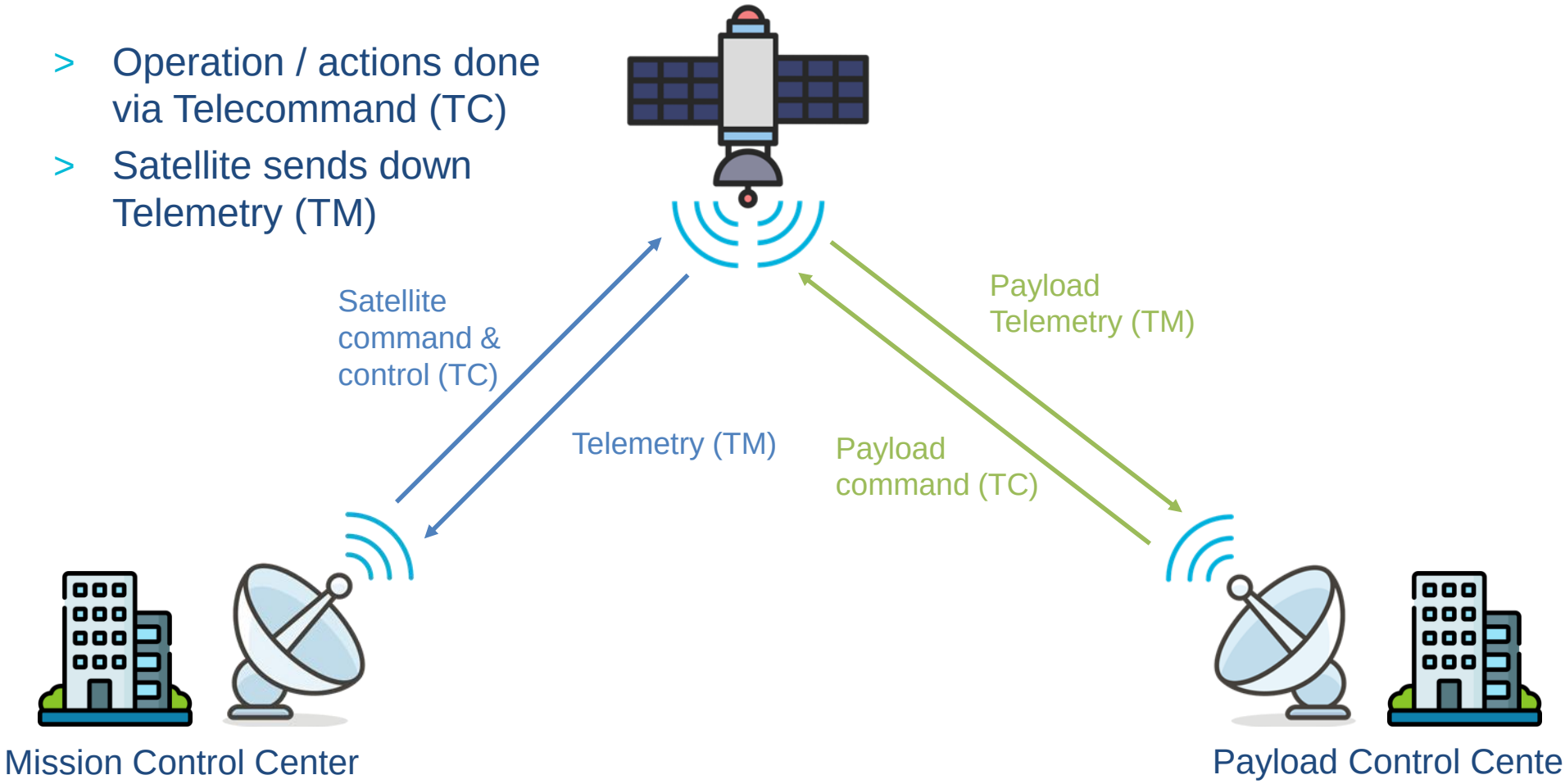
- Computing
  - Limited CPU
  - Limited RAM
- No human onboard intervention

↓

Signature and heuristic-based detection

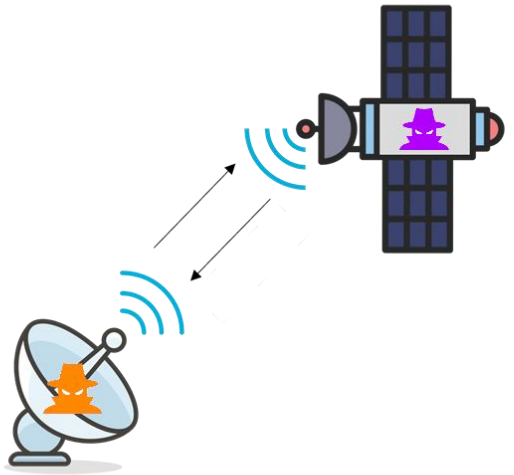
Satellite operations

- Operation / actions done via Telecommand (TC)
- Satellite sends down Telemetry (TM)



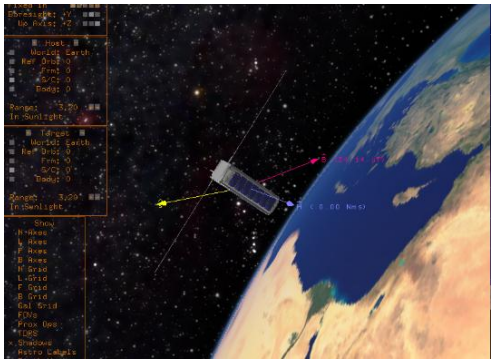
Threat model

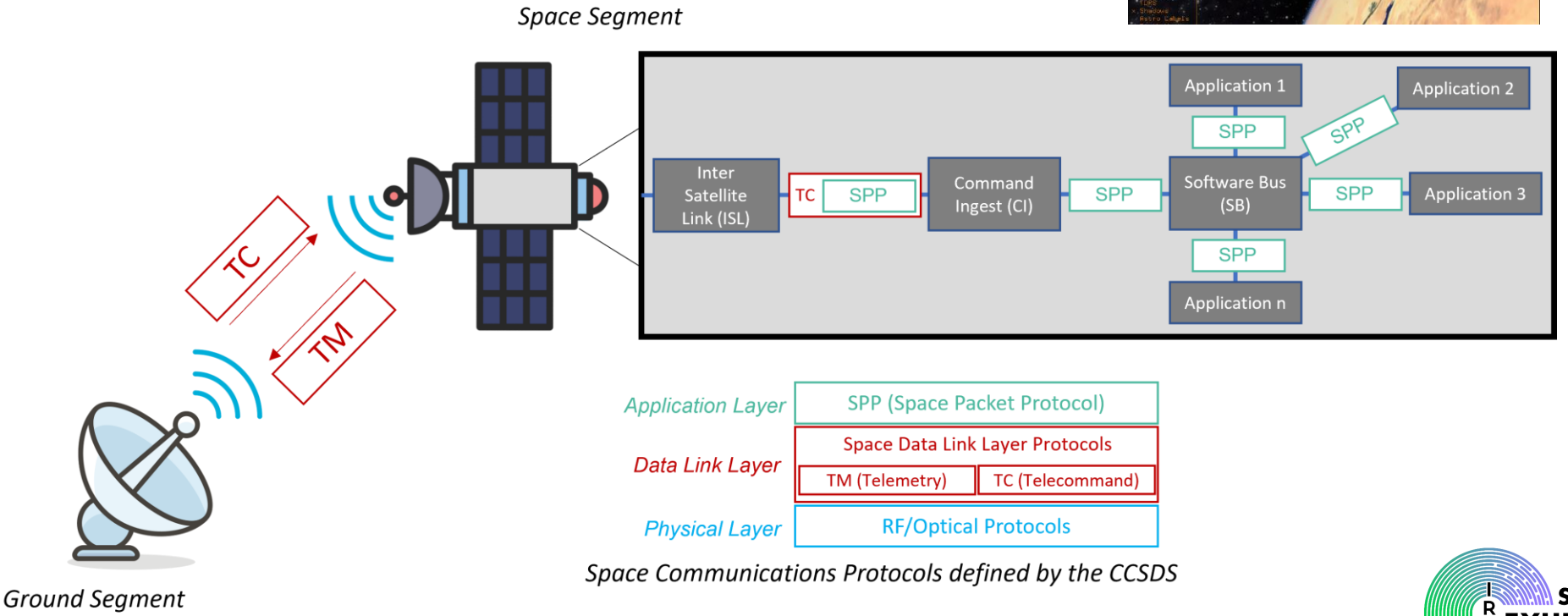
- Focus: attacks impacting the satellite (detectable on-board)
- Attacks can come from 2 main threat vectors
  - Ground station compromise
  - Satellite payload compromise (i.e. platform is assumed safe)
- Attacks can use one or more Telecommands (TC)
- Multiple attacks considered
  - TC flooding
  - System crash
  - System files deletion
  - …
- Detection has to adapt to the diversity of attacks !



Simulation environment

- NASA NOS3 CubeSat simulator (CSS Project fork)
- Well-known CCSDS communication protocols (i.e. SPP & TC)
- Software Bus architecture





Application Layer	SPP (Space Packet Protocol)
Data Link Layer	Space Data Link Layer Protocols: TM (Telemetry) / TC (Telecommand)
Physical Layer	RF/Optical Protocols

Space Communications Protocols defined by the CCSDS

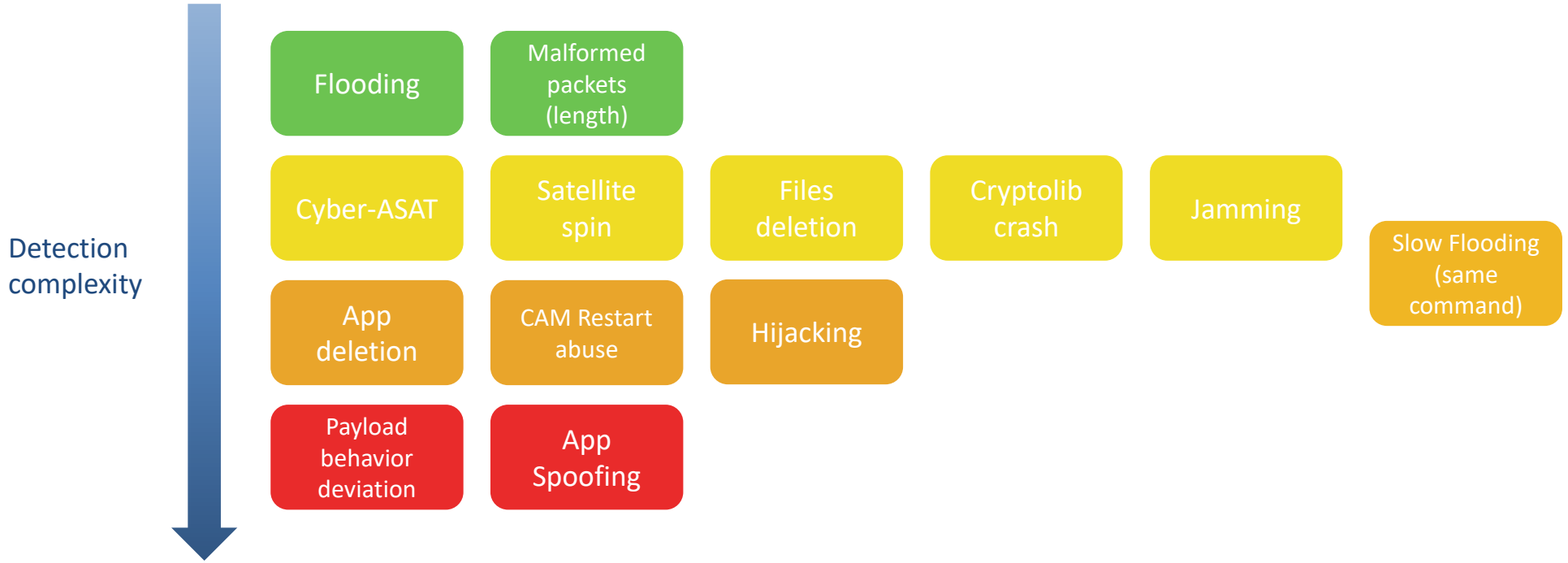
Attacks considered

> Ground attacks
- Tool developed (**Input Generator**) to send custom SPP commands
- Used for ground attacks but also for legitimate traffic

> Payload attacks
- Edited payload (camera) code
- Present before launch / loaded during updates

Attacks	Description	Threat Source
FA_1	Network Flooding	Ground
FA_2	Network Flooding	Payload
CA_1	Application Kill	Ground
CA_2	Application Kill	Payload
AD_1	App Erase	Ground
AD_2	App Erase	Payload
SW_1	Files Wipe	Ground
SW_2	Files Wipe	Payload
SP_1	Satellite Spin	Ground
SP_2	Satellite Spin	Payload
CR_1	System Crash	Ground
CR_2	System Crash	Payload
CR_3	System Crash	Ground
HJ_1	Satellite Hijacking	Ground

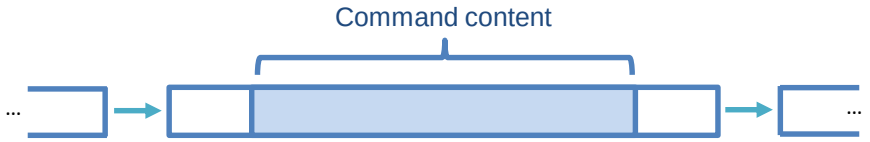
Attacks and detection



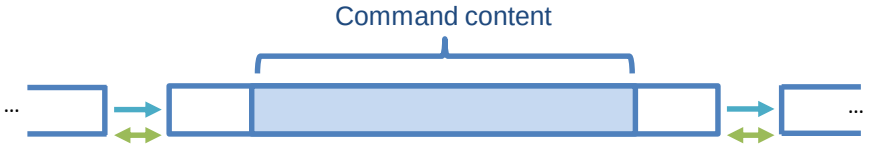
Detection strategies

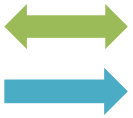
- On packet arrival (encrypted traffic)
  - Inter-packet times
  - CRC Verification
  - TC header correctness
  - Packet size

- After decryption (access to TC content)
  - Analysis of the dangerousness of the commands
  - Whitelist of commands usually sent from ground



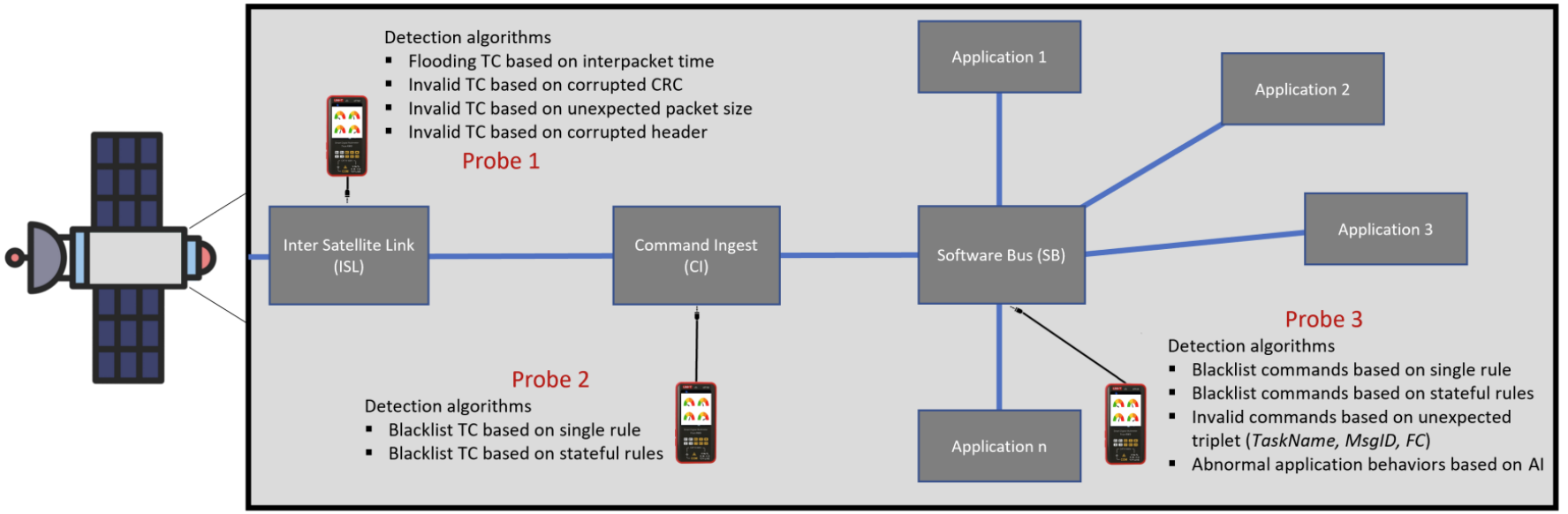
- On communications bus (internal packets)
  - Analysis of the dangerousness of the internal commands
  - Whitelist of commands per application
  - AI to detect payload deviations





Probes deployment

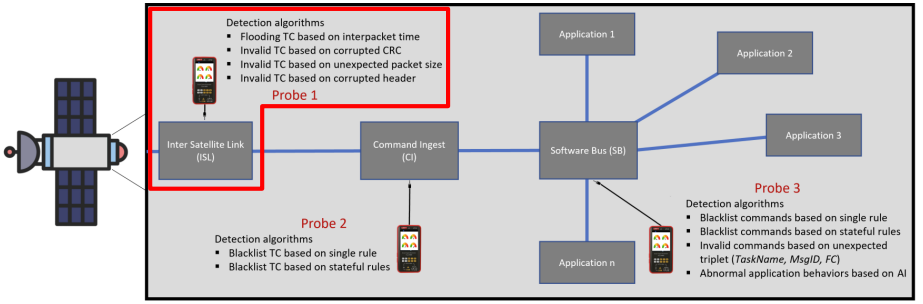
- Implementation on NOS3 platform
- Probes location matching detection strategies
- Probes data fed to detection algorithms



Overview of probes placement & detection algorithms

Probe 1

> Located « before decryption »
  - Logging capabilities
  - Detection modules



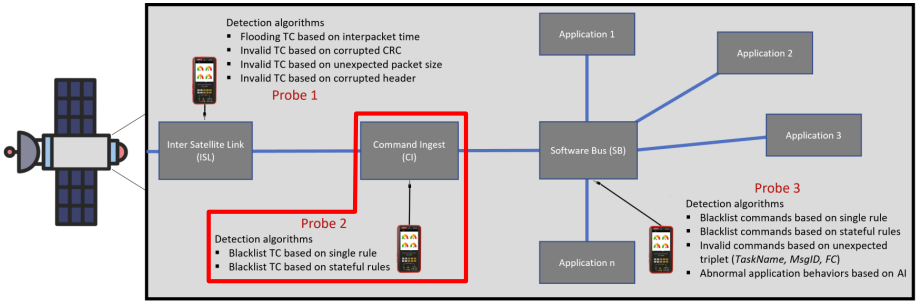
Detection algorithms

Malicious behavior	Detection method
TC Flooding from ground	Anti-flooding module
Jamming / malformed packets	CRC verification
Malformed packets with correct CRC	Packet size verification (if block cipher)
Incorrect TC header values (e.g. Cryptolib crash)	TC header verification

> Simple but efficient heuristics

Probe 2

> Located « after decryption »
  - Logging capabilities
  - Detection modules



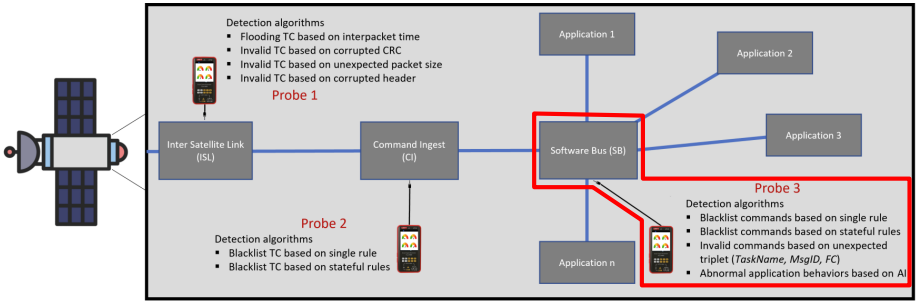
Detection algorithms

Malicious behavior	Detection method
Correctly formed but dangerous single commands	TC Analyzer (stateless mode)
Correctly formed but dangerous command sequences	TC Analyzer (state machine mode)
‘Slow’ flooding using the same command	Slow-Flooding module

> Access to packet contents
> Enables rule-based approach

Probe 3

- Located « on bus »
  - Logging capabilities
  - Detection modules



Detection algorithms

Malicious behavior	Detection method
Internal application sending unauthorized commands	Application profiling / TC analyzer
Dangerous commands / sequences received	TC analyzer
Payload deviating from usual behavior	Anomaly Detection

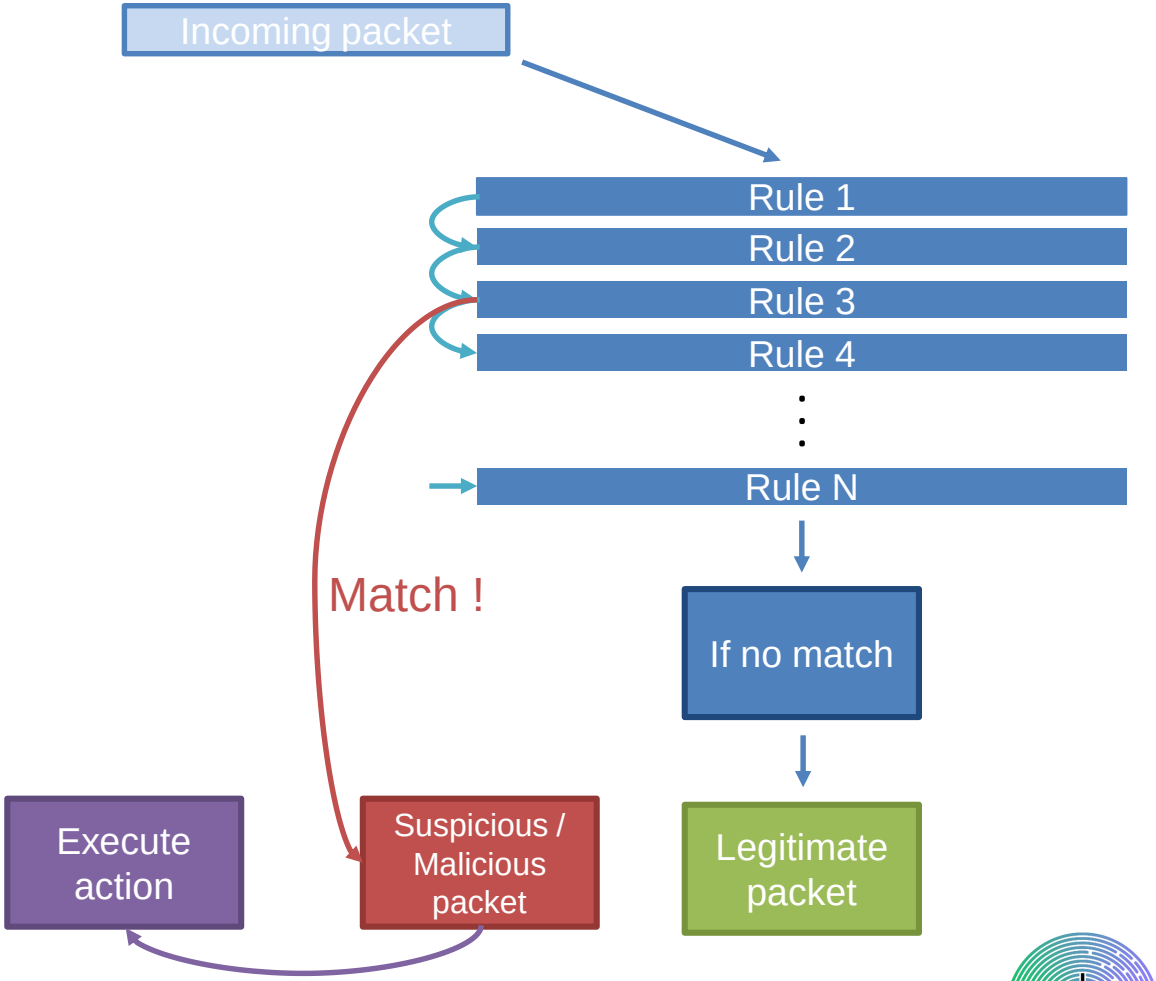
- Access to internal exchanges
- Least-privilege approach

TC Analyzer Logic

Global principles

- Similar to a firewall
- Top-down rule processing
- Internal state machines
- Drop packets / raise alerts
- Flex/Bison parsing

- Probe 2 & Probe 3 instances with different rulesets



TC Analyzer Rules

Detection rule syntax:

	Action	Command Identification		Parameter Characteristics				
	(S)witch to State X							
[Current State] |	(A)lert	Msg Id	FC	OffSet	: Type	: Length	Parameter Value	
	(D)rop							

Stateless rules:

- > Packets inspected independently
- > One-shot attacks

- > System files wipe:
  - DELETE_ALL command (wipe a folder)
  - With parameter “/cf” (config files)
- > DoS Cyber-ASAT attack:
  - STOP_APP command (stop an app)
  - On “CI” app (Ingests commands from ground)
  - Isolates satellite from ground

DELETE_ALL

Rule 1: ‘D 0x188C 7 0:STRING:512 /cf’

STOP_APP

Rule 2: ‘D 0x1806 5 0:STRING:160 CI’

TC Analyzer Rules

Detection rule syntax:

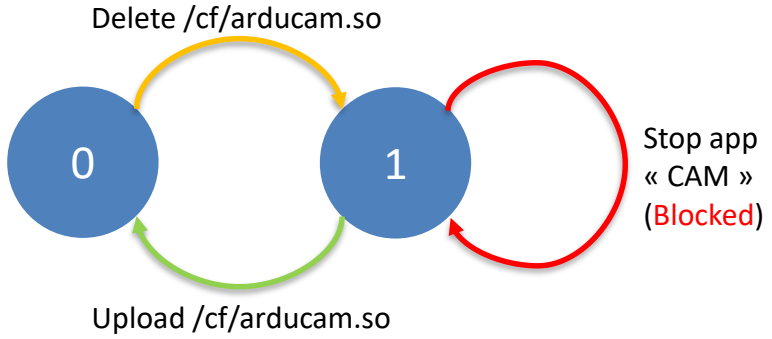
	Action	Command Identification		Parameter Characteristics				
	(S)witch to State X							
[Current State] |	(A)lert	Msg Id	FC	OffSet	: Type	: Length	Parameter Value	
	(D)rop							

Stateful rules:

- > Detect malicious sequence of commands
- > State machines fully customizable
- > Deterministic finite automaton

- > App Deletion attack
  1. Application file deletion
  2. Application stop
  3. If a file is received in between, update ! (legitimate)

State machine:



Rule:

‘S0|S1 0x188C 5 0:STRING:512 /cf/arducam.so;
S1|S0 0x18B3 2 672:STRING:512 /cf/arducam.so;
S1|D 0x1806 5 0:STRING:160 CAM’

IDS Evaluation

Detection evaluation

> No false positives*
> All attacks were detected
➡ Our approach is relevant

Performance evaluation

Two experimental setups:

> Current ruleset (20-ish rules)
> 1000 rules (for scalability)

	Metric	IDS On – Default	IDS On – 1000 rules
	CPU Usage	Non-significant impact	Non-significant impact
RAM	Memory usage (stack)	+ 374B	+ 374B
RAM	Memory usage (BSS)	+ 1070B	+ 132270B
	Latency	+ 21µs (8,7%)	+ 34µs (14%)

> Low impact overall
> With many rules impact on memory

* Except for Anti-flooding heuristic during file transfer

Future work

IDS improvements

- Correlation with
  - State variables of satellite
  - Ephemerides for detection
- Specify conditions or ranges in the detection rules (TC Analyzer)

Complementary approaches

- Anomaly detection on payload behavior / satellite bus
  - Dataset generation (with CSS project)
  - Simulated data
  - Resilient to unknown attacks
- Correlation with an on-ground probe
  - Transmit alerts to ground
  - Determine attack location
  - Detailed information for operators (diagnostic)
  - Could allow IDS configuration from ground

Thank you for your attention !

Louis Lolive
louis.lolive@laas.fr

Simulator & IDS code