SnapShotter: Lightweight Intrusion Detection and Prevention System for Industrial Control Systems

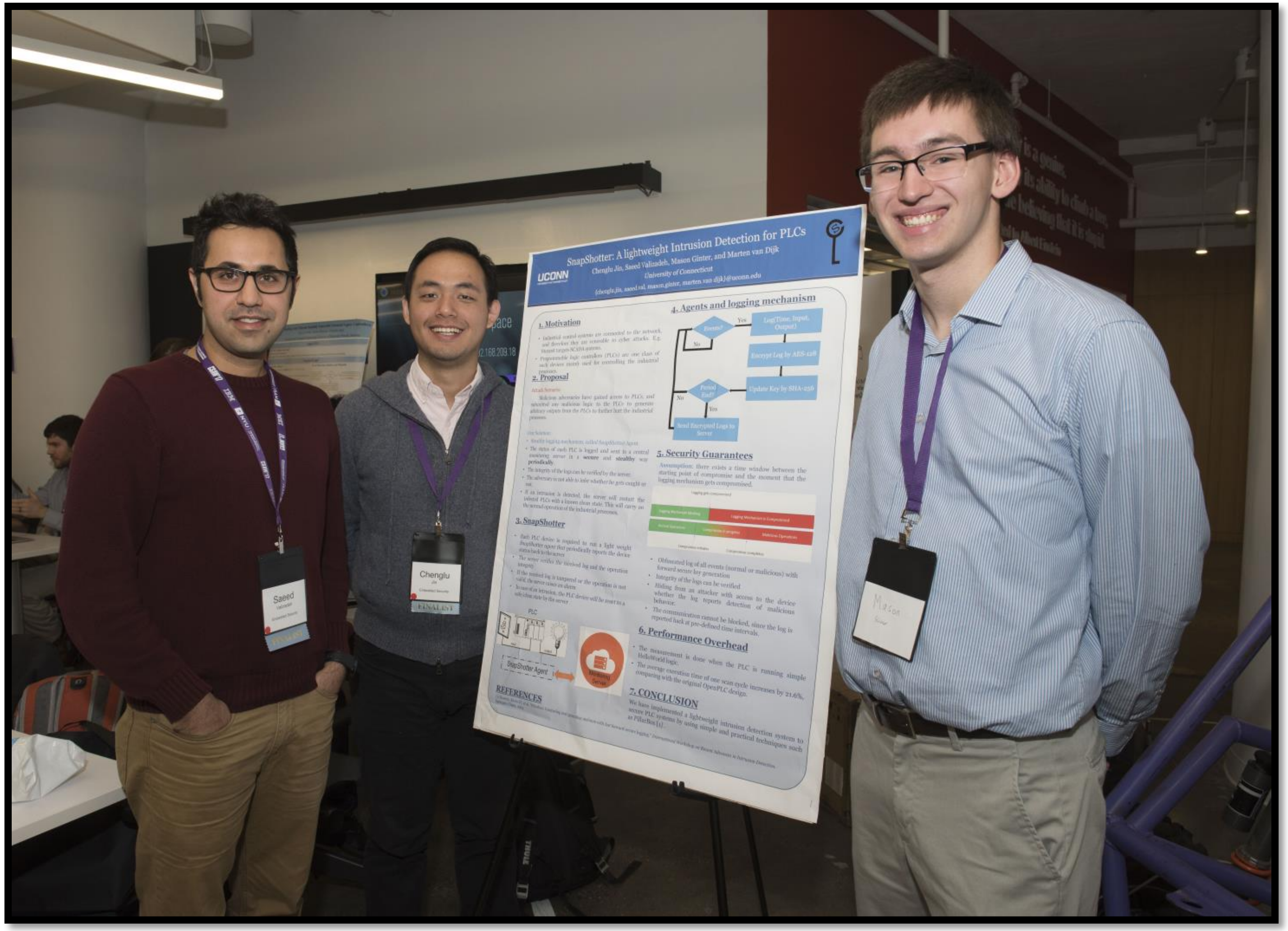
Chenglu Jin, Saeed Valizadeh, and Marten van Dijk

Secure Computation Laboratory
University of Connecticut

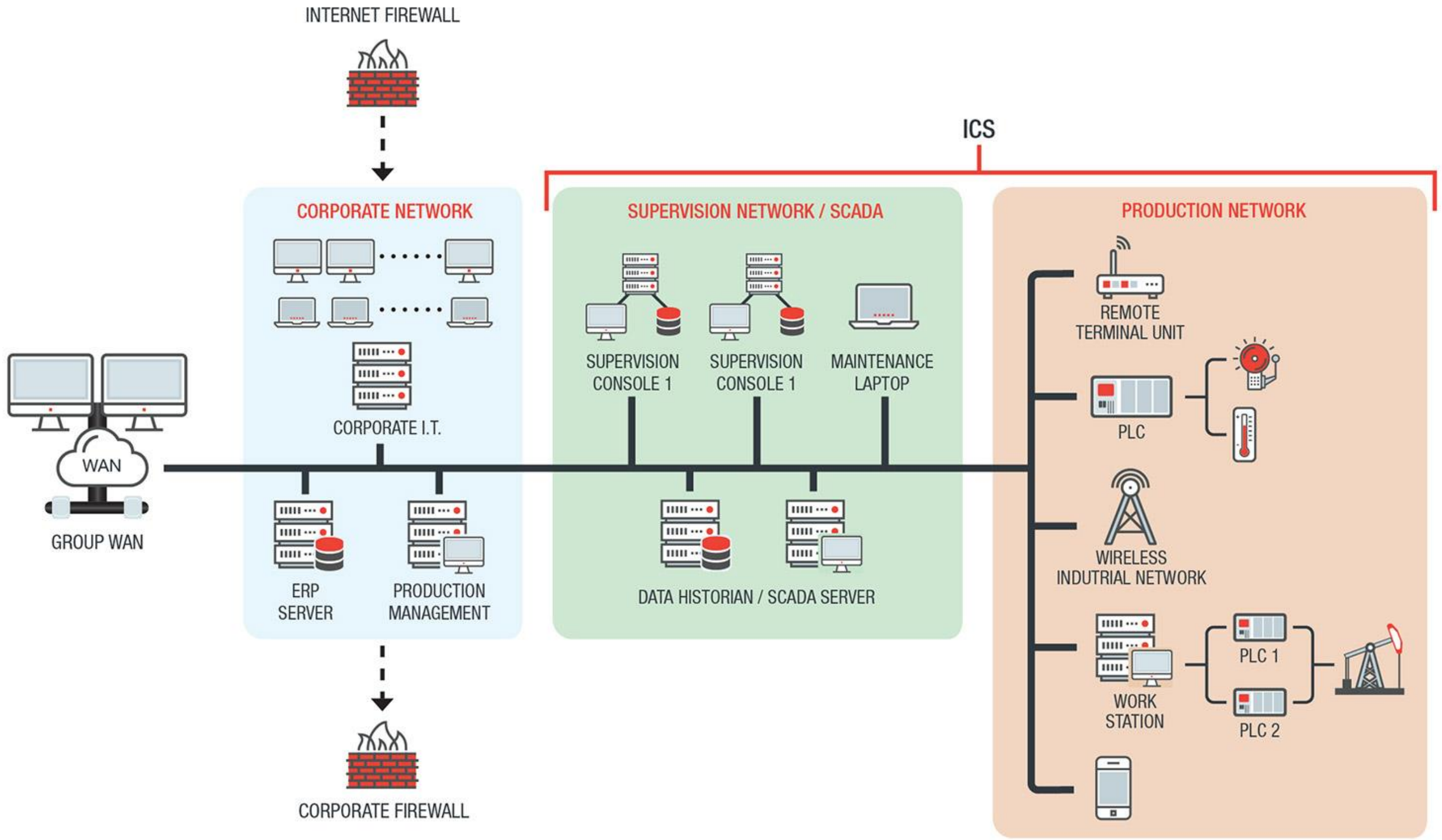
Email: {chenglu.jin, saeed.val, marten.van_dijk}@uconn.edu

UCONN | UNIVERSITY OF CONNECTICUT

CSAW'17 Embedded Security Challenge



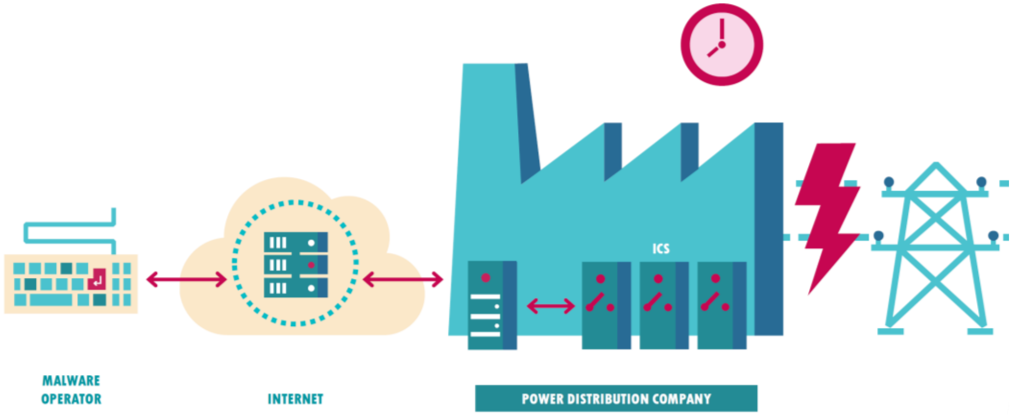
Overview of an Industrial Control System

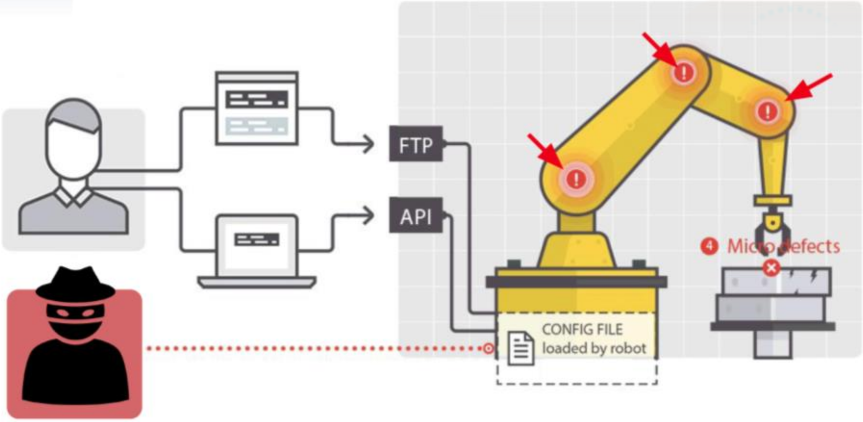


Holy grail of cyberwar?

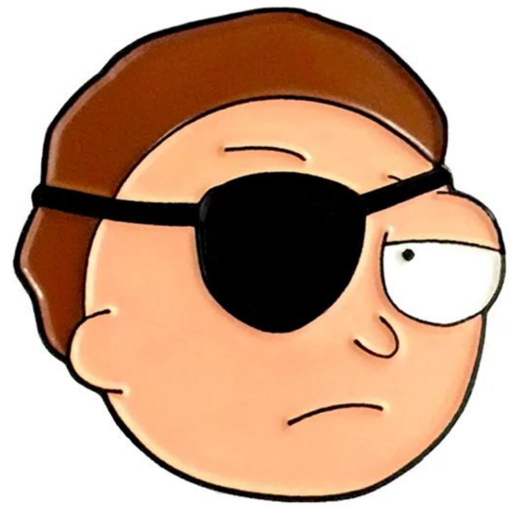
- 2010: STUXNET
  - Advanced malware (worm), Targeting SCADA systems
  - Causing substantial damage to nuclear plants (specially designed to sabotage the Iranian nuclear project)
- 2014: HAVEX
  - Semi-Stuxnet worm, Targeting ICS and SCADA systems
  - Impacted as many as 2,000 infrastructure sites, a majority of which were located in Europe and the United States
  - Capable of possibly disabling hydroelectric dams, overload nuclear power plants, and even can shut down a country's power grid with a single keystroke.
- 2015: BlackEnergy
  - A Trojan that is used to conduct DDoS attacks, cyber espionage and information destruction attacks
  - Mostly targeted ICS, energy, government and media in Ukraine
- 2016: Industroyer
  - A modular malware, capable of gaining direct control of switches and circuit breakers at an electricity distribution substation.
  - Attack on Ukraine's power grid that deprived part of its capital, Kiev, of power for an hour

And still, more attacks are on the way!





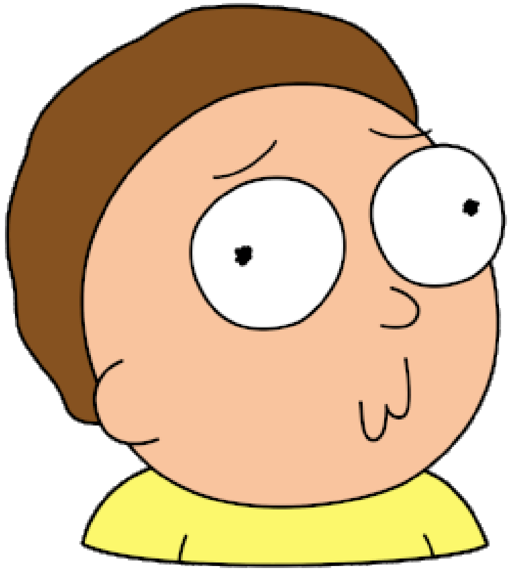
So, why do attackers target ICS?



- Easy targets!
- Big financial gains!
- Industrial espionage!
- Huge physical impact and damage!
- Many other malicious intents and/or maybe mental problems!

UCONN

So, what is the problem?



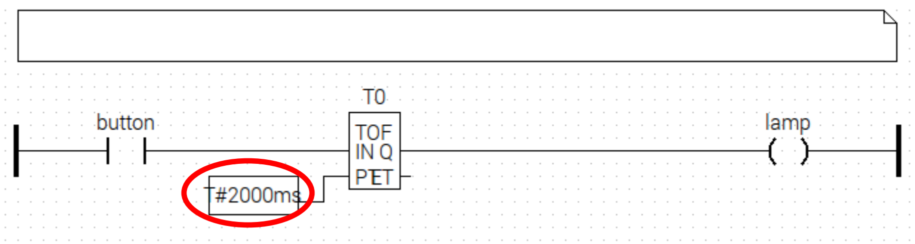
- Widespread applications in critical infrastructure
  - Transportation, Manufacturing, Power grids, Oil/gas processing, etc.
- Lack of security considerations in the design and lifecycle of traditional ICS
- Exposure to outside world (i.e., the Internet)
- Increased connectivity via embracing the new information technologies

UCONN

Adversarial Model

- Strong(est) Malicious adversaries
  - Are capable to get remote/physical access to Programmable Logic Controllers (PLCs)
  - Can submit any arbitrary (malicious) logic to the PLCs to generate arbitrary outputs from the PLCs to further hurt the industrial processes.
- What the attacker cannot do:
  - Physically tampering the PLC hardware

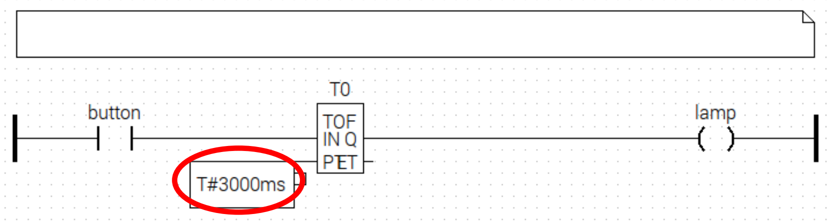
#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	button	Local	BOOL	%IX0.0			
2	lamp	Local	BOOL	%QX0.0			
3	T0	Local	TOF				



Expected logic

Description: Class Filter: All

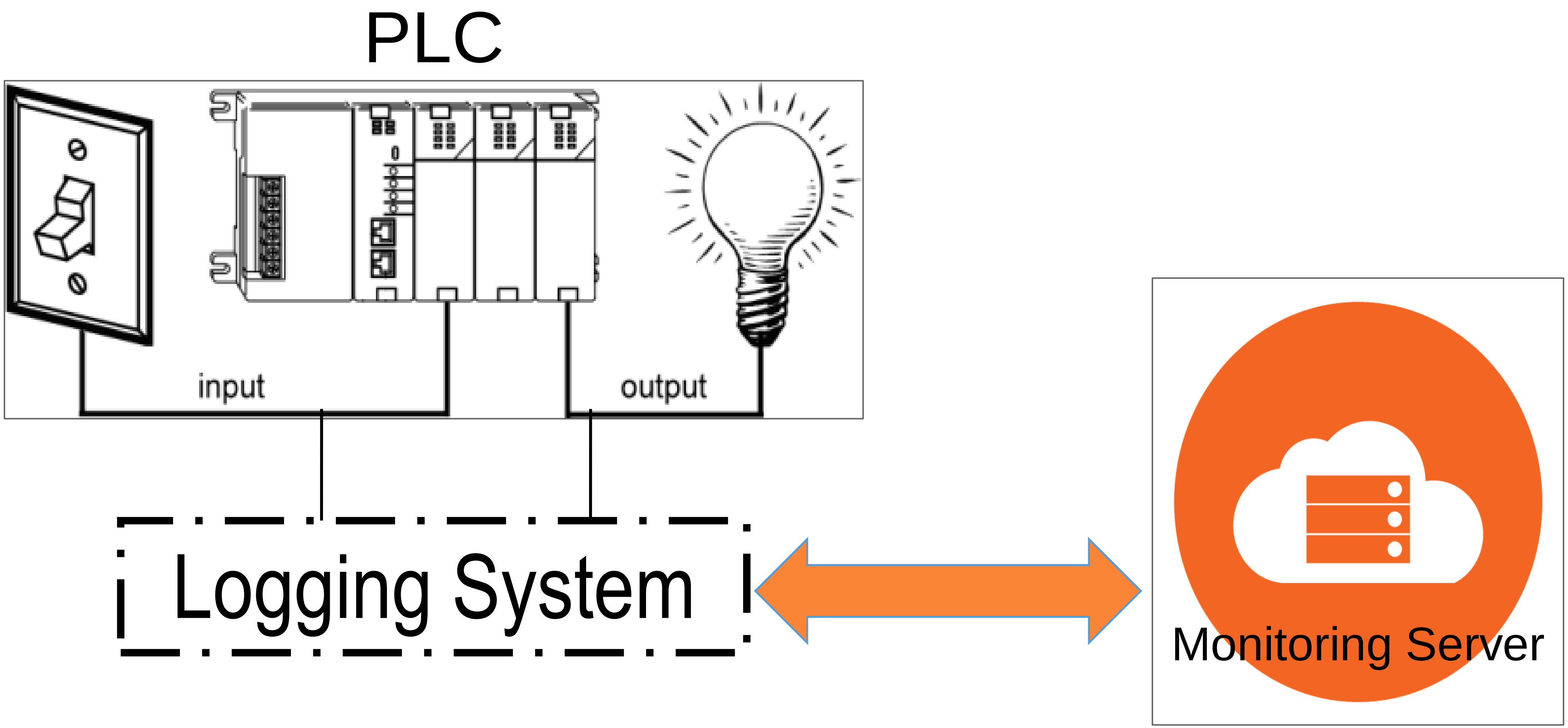
#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	button	Local	BOOL	%IX0.0			
2	lamp	Local	BOOL	%QX0.0			
3	T0	Local	TOF				



Malicious logic

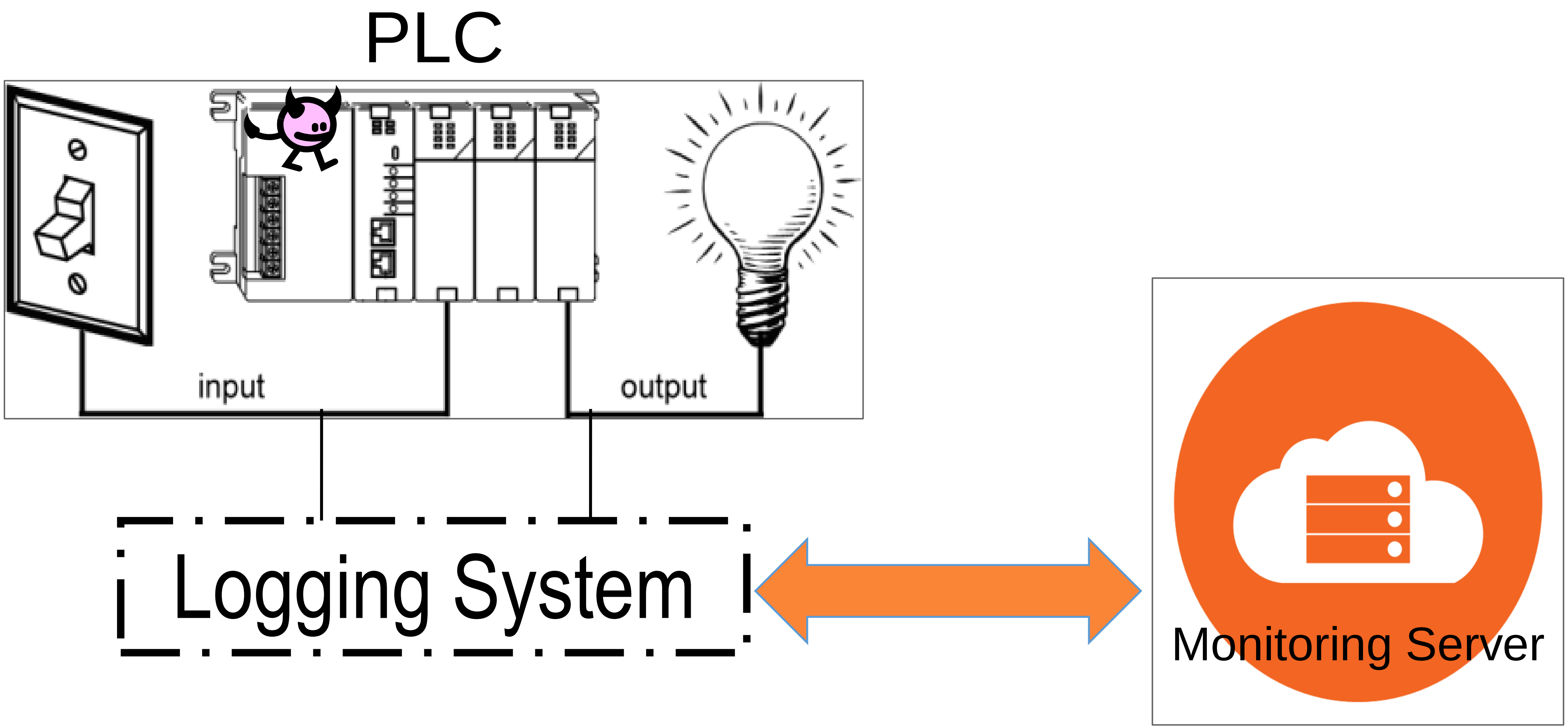
Simple Idea

- Hmm, we need a secure and trustable logging mechanism:



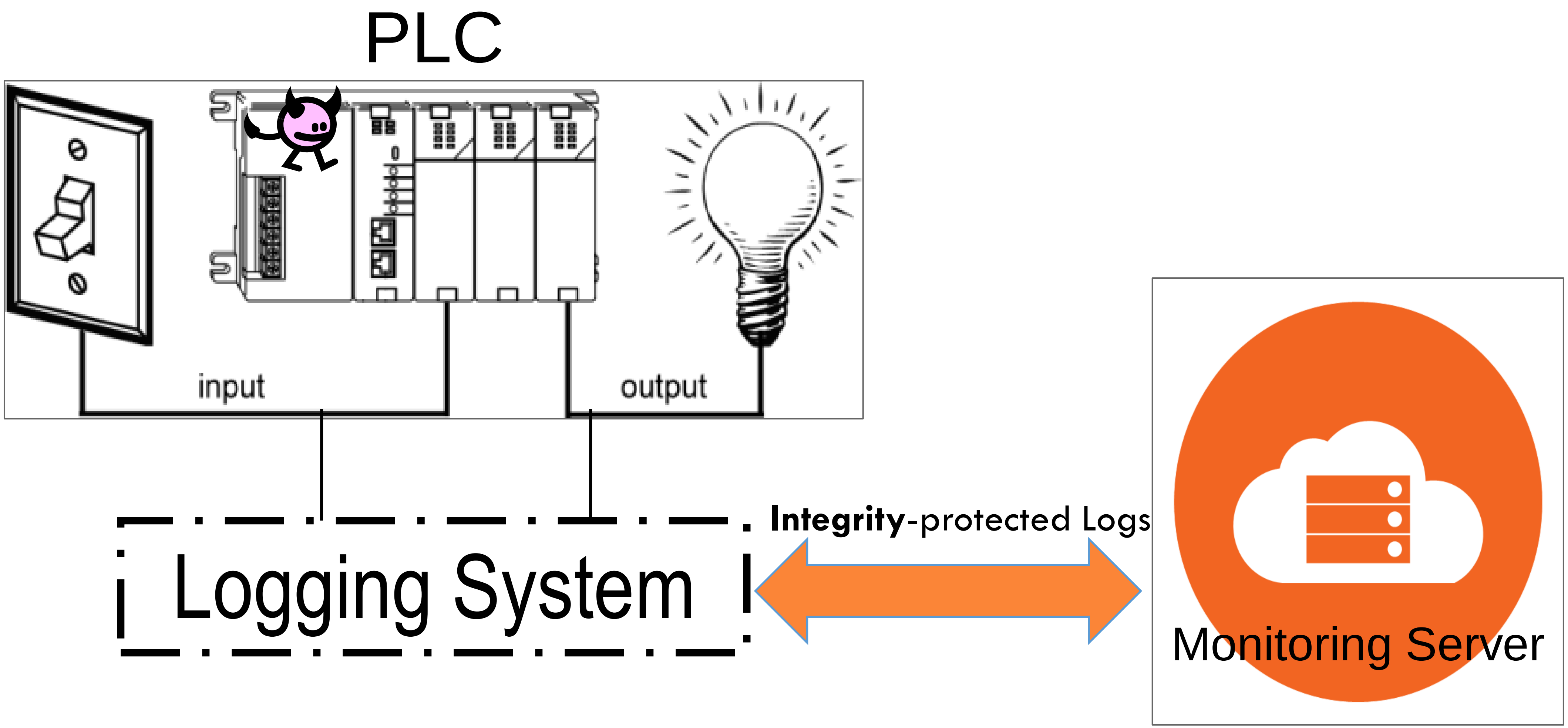
Simple Idea

- Hmm, we need a secure and trustable logging mechanism:



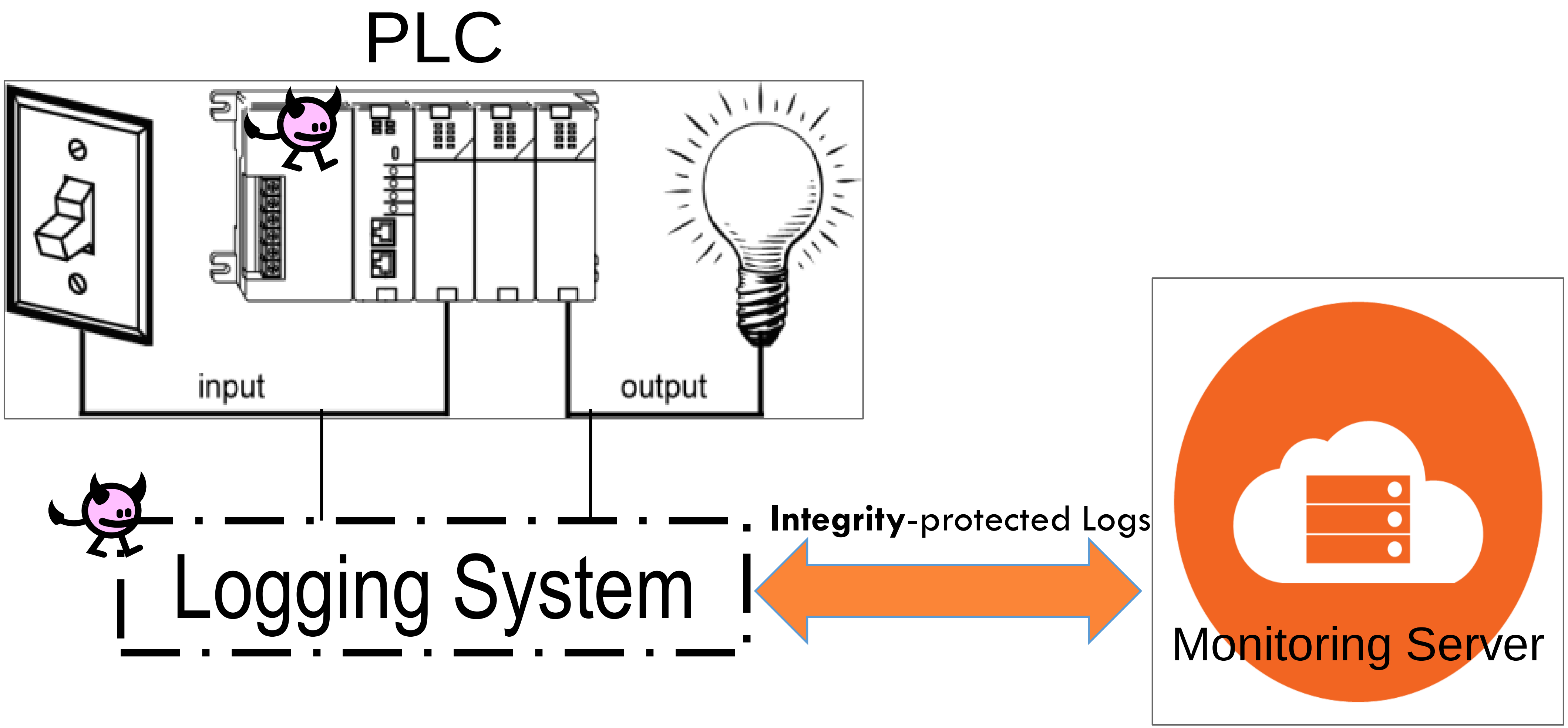
Simple Idea

- Hmm, we need a secure and trustable logging mechanism:



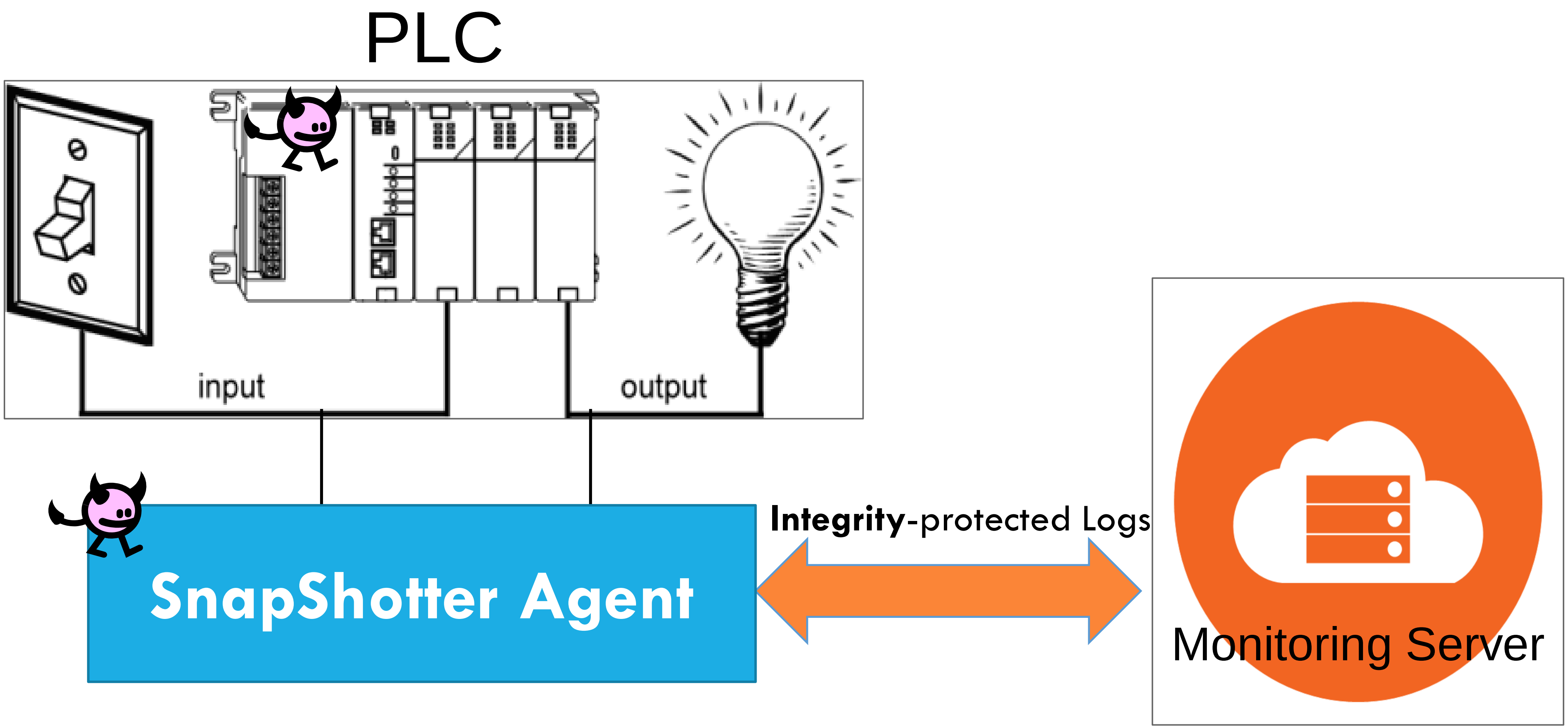
Simple Idea

- Hmm, we need a secure and trustable logging mechanism:

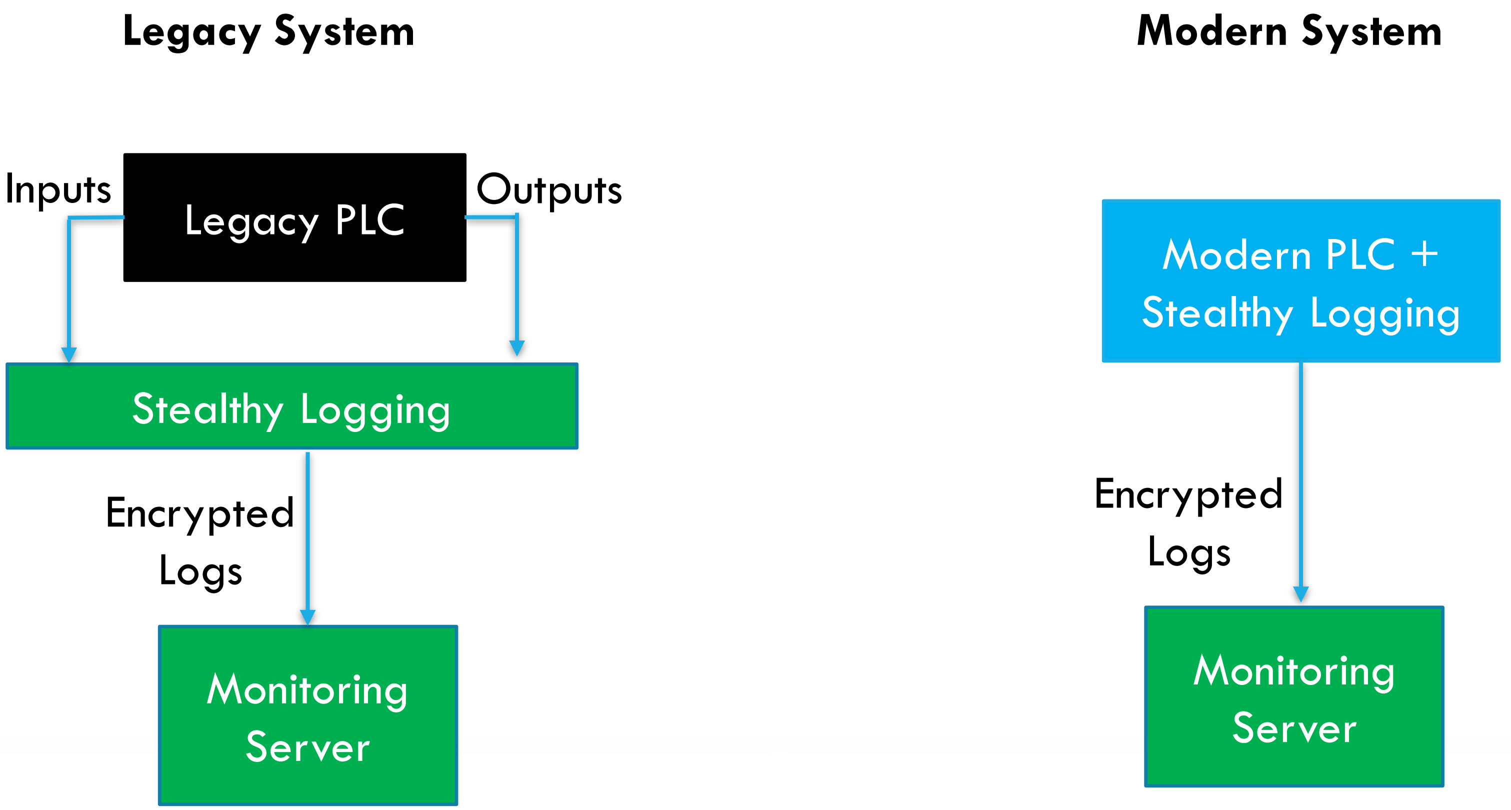


Simple Idea

- Hmm, we need a secure and trustable logging mechanism:



Modern vs Legacy Systems



Agent and Server tasks in a nutshell

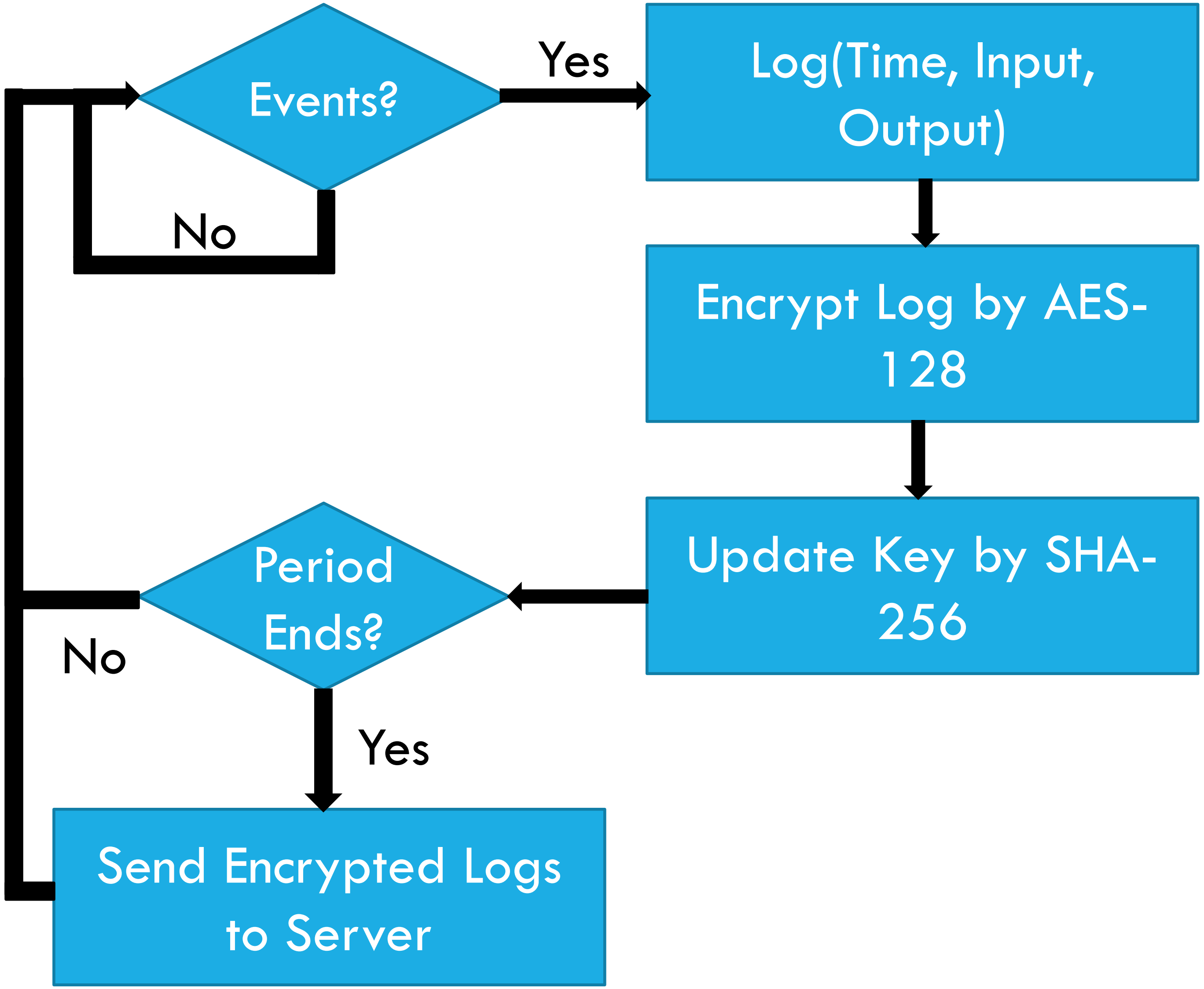
- Intrusion detection agent (i.e., the Snapshotter)
  - Security-related information gathering (e.g., integrity of the logic, paramount file accesses, I/O operations)
  - Checking the occurrence of events or state updates of the monitored device
  - Fast forward-secure logging
  - Transmitting the logs to the server

- The Trusted Server:
  - Logs integrity verification
    - Making sure logs are valid and not tampered by an adversary
  - Log analysis and incident identification
    - Tracing deviations from expected PLC profiles (Potentially stablished during system Installation)
    - Checking if the device is functioning properly and not compromised
    - Raising a flag, If log' integrity check fails or system state is recognized as compromised
  - Incident response
    - Further investigation of device status
    - Recovering the infected machine to a clean state
    - Activating a redundant (backup) PLC

SnapShotter Agent in more details

- **Secure and reliable logging mechanism with Forward Secure Key Management System.**
- The status of each PLC is logged and sent to a central monitoring server in a **secure** (and potentially **stealthy**) way **periodically**.
- The **integrity** of the logs can be verified by the server.
- The adversary is not able to infer whether he/she gets caught or not, even when he/she compromised the device completely, including the **logging mechanism** and **secret key**.
- If an intrusion is detected, the server can take effective actions, e.g., **restore** the infected PLCs to a known **clean state +** Activate a redundant PLC. This will carry on the normal operation of the industrial processes.

Logging Mechanism

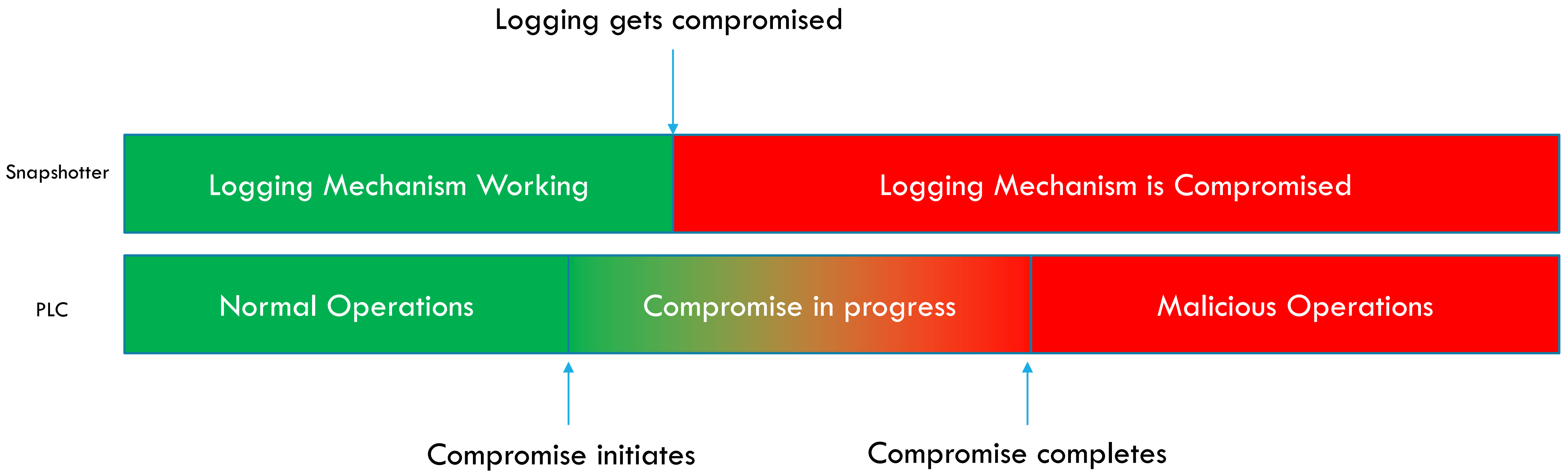


Log Data Format

#Byte	1 Byte	2 Bytes	2 Bytes	4 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte
	Start	Event ID	Device ID	Time	Digital Inputs	Digital Outputs	Analog Outputs	End
Example	0xFF	0x0002	0x1234	0x00000010	0xC000	0x8000	0x7832	0xFF

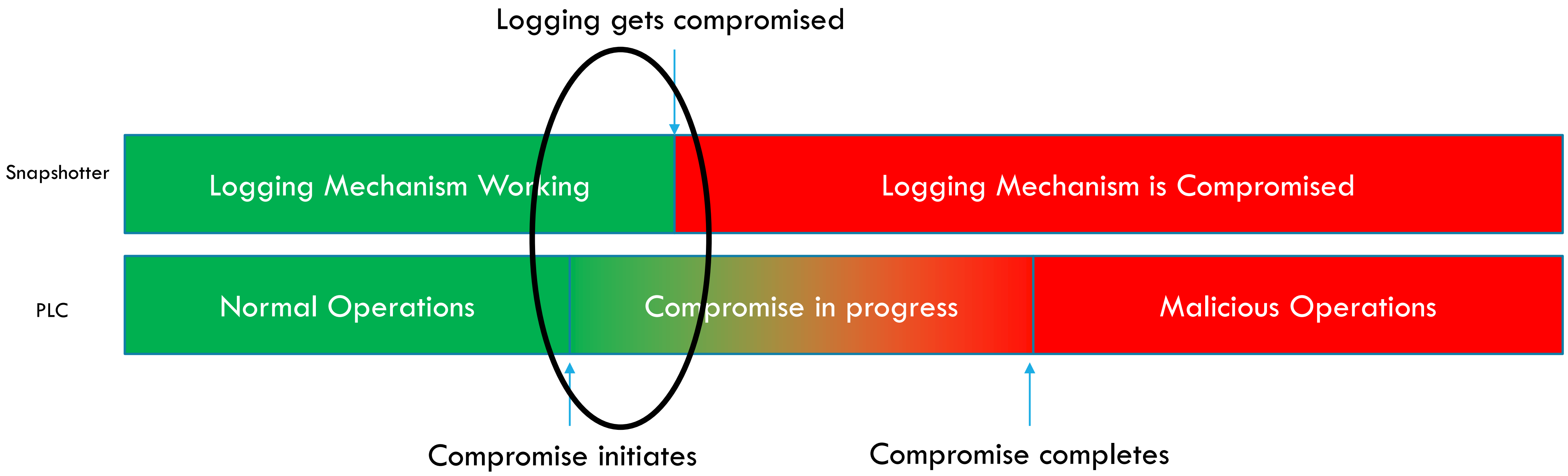
$\underbrace{\hspace{20em}}_{\text{16 Bytes in total}}$

Assumption



Assumption: Some logs are generated between the beginning of the attack and the moment that the logging system gets compromised.

Assumption



Assumption: Some logs are generated between the beginning of the attack and the moment that the logging system gets compromised.

So, What options does an attacker have?

So, What options does an attacker have?

- Do nothing!

So, What options does an attacker have?

- Do nothing!

So, What options does an attacker have?

- Do nothing!

- Try to decrypt the logs!

So, What options does an attacker have?

- Do nothing!

- Try to decrypt the logs!

So, What options does an attacker have?

- Do nothing!

- Try to decrypt the logs!

- Tamper with the encrypted logs!

So, What options does an attacker have?

- Do nothing!

- Try to decrypt the logs!

- Tamper with the encrypted logs!

So, What options does an attacker have?

- Do nothing!

- Try to decrypt the logs!

- Tamper with the encrypted logs!

- Packet dropping!

So, What options does an attacker have?

- Do nothing!

- Try to decrypt the logs!

- Tamper with the encrypted logs!

- Packet dropping!

Performance Overhead

- The performance overhead we measured on our platform is at most 54 μs per scan cycle comparing with the original OpenPLC design.
- We tested our implementation by uploading a malicious logic to the controller, the server was able to catch the intrusion immediately after receiving the logs from the agent

Conclusion

- We have implemented a lightweight intrusion detection system to secure PLC systems by using simple and practical techniques.

Conclusion

- We have implemented a lightweight intrusion detection system to secure PLC systems by using simple and practical techniques.

- Security Guarantees: Guaranteed reports of malicious behaviors **even when the logging mechanism and secret key are compromised.**

Conclusion

- We have implemented a lightweight intrusion detection system to secure PLC systems by using simple and practical techniques.

- Security Guarantees: Guaranteed reports of malicious behaviors **even when the logging mechanism and secret key are compromised.**

- Performance Overhead: 54 μs per scan cycle comparing with original OpenPLC

Questions?