Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021
(NetSys 2021)

Demonstration: A cloud-control system equipped with intrusion detection and mitigation

Fatemeh Akbarian, William Tärneberg, Emma Fitzgerald, and Maria Kihl

4 pages

Guest Editors: Andreas Blenk, Mathias Fischer, Stefan Fischer, Horst Hellbrück, Oliver Hohlfeld, Andreas Kassler, Koojana Kuladinithi, Winfried Lamersdorf, Olaf Landsiedel, Andreas Timm-Giel, Alexey Vinel
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Demonstration: A cloud-control system equipped with intrusion detection and mitigation

Fatemeh Akbarian[1], **William Tärneberg**[1], **Emma Fitzgerald**[2 1], **and Maria Kihl**[1]

[1] Dept. of Electrical and Information Technology at Lund University, Lund, Sweden
[2] Institute of Telecommunications, Warsaw University of Technology, Poland

Abstract: The cloud control systems (CCs) are inseparable parts of industry 4.0. The cloud, by providing storage and computing resources, allows the controllers to evaluate complex problems that are too computationally demanding to perform locally. However, connecting physical systems to the cloud through the network can provide an entry point for attackers to infiltrate the system and cause damage with potentially catastrophic consequences. Hence, in this paper, we present a demo of our proposed security framework for CCs and demonstrate how it can detect attacks on this system quickly and mitigate them.

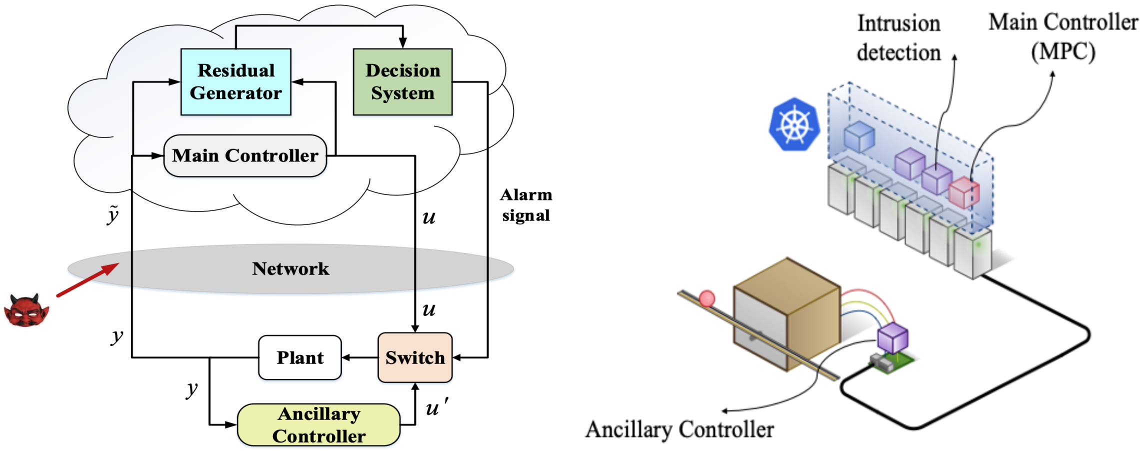
Keywords: Cloud control systems, Intrusion detection, Attack mitigation, Test-bed, Cyber security

1 Introduction

By adoption of new technologies, we have had several revolutions in industry and now some modern technologies like IoT, cloud computing, etc are paving the way for smart factory that will realise industry 4.0. Industrial control systems (ICs) as part of industry 4.0 are becoming more efficient and smarter. However, these systems are also becoming more and more connected and part of a network systems and this communication link between different components of ICs can provide an access point for attackers to intrude into the system and manipulate the signals that are sent through the network. For example, the attacker by manipulating measurement signals can deceive the controller to generate a wrong control signal that can make our system unstable and lead to catastrophic consequences like what we had in recent years. In recent years, we have had several attacks in different parts of industry that demonstrate ICs are still prone to cyber attacks and highlight the necessity for an appropriate security measure to protect these systems.

The cloud provides seemingly endless computing and storage resources that can be used to execute more advanced control strategies in ICs. However, in cloud control systems (CCs), there is a network between the plant and the cloud that the measurement and control signals are sent through this network and this can make these systems vulnerable to cyber attacks.

In this paper, we present a demo of our proposed security framework that is applied on CCs and include intrusion detection and mitigation. In this system, we have a ball and beam process as our plant, a Kubernetes (K8S)-cluster that hosts an intrusion detection and the main controller, and a local controller that is part of our mitigation method and implemented using Python code beside the plant.



(a) Security framework overview.

(b) Test-bed overview.

Figure 1: An overview of the demo system.

2 Secure cloud control systems demo

In this section, we present our demo and explain how different parts of the test-bed are implemented. Figure 1 shows an overview of our demo system. Figure 1a shows our proposed security framework, and Figure 1b shows the test-bed implementation. The system's components are detailed below.

2.1 Plant

We use a ball and beam process as our plant. The ball and beam system consists of a long beam which can be tilted by an electric motor together with a ball rolling back and forth on top of the beam. This system is open-loop unstable and without a controller, it will swing to one side or the other, and the ball will fall off the end of the beam. Our motivation for choosing the ball and beam system is that it is an intrinsically unstable and time-critical system such that any attack on this system can make it unstable quickly. So, evaluating our proposed security framework on this system can prove its effectiveness.

2.2 Kubernetes cluster

The test-bed has been equipped with a six-node Kubernetes cluster as the edge cloud. Kubernetes (K8S) [1] is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation . The cluster has been equipped with an nginx ingress [2] and prometheus operator [3] . The nginx ingress is exposed

[1] https://kubernetes.io/

[2] https://github.com/kubernetes/ingress-nginx/

[3] https://github.com/coreos/prometheus-operator/

using the K8S NodePort paradigm. We use this K8S cluster to implement our main controller and intrusion detection algorithm.

2.3 Main controller

To stabilize the ball, we need a feedback controller that uses measurement signals to adjust the beam accordingly. A Model Predictive Control (MPC)-based controller is designed based on [STÅK20] as the main controller in Figure 1. This controller for execution needs some computation resources that cloud can provide it. Thus, we deploy this controller using Python and Container technology as a pod in the Kubernetes cluster.

2.4 Attack and intrusion detection

All communication between the plant and the cloud is over a Local Area Network (LAN) and uses a protocol defined in Protocol Buffers (Protobuf) [4] which is realized in gRPC [5]. Measurement signals includes position of the ball, angle of the beam, and speed of the ball are sent through this network to the main controller in the cloud. The main controller using these generates the control signal and send it back to the plant. This control signal by adjusting the beam's speed controls the position of the ball on the beam. We assume the attacker tries to manipulate the measured position signal, and we implement the attack by adding a ramp signal with small slope (between 0.001 to 0.05) to the position signal using python codes.

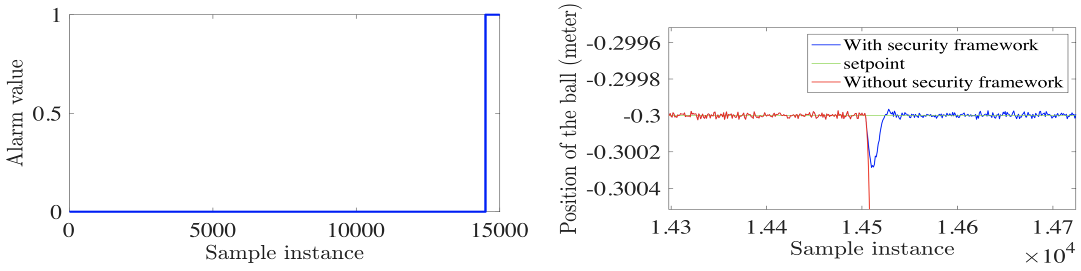
We deploy our proposed intrusion detection in [ATFK21] using python and container technology as a pod in the Kubernetes cluster. This intrusion detection consists of two parts: residual generator and decision system. Residual generator estimates the value of the measurement signal and then by comparing it with the real one generates a residual signal. In healthy condition during which there is no attack, the residual signal is close to zero. Thus, the decision system evaluates the residual signal and by comparing it with a certain threshold decides if there is any attack in the system or not. If it detects any attack in the system, it will trigger an alarm signal.

2.5 Attack mitigation method and ancillary controller

Our objective for mitigation is to keep the plant stable under the attack with an acceptable performance. As mitigation, we consider an ancillary controller that is placed close to the plant such that there is no public network between the plant and this controller so there is no possibility for the attacker to intrude into the system. Measurement signals from the plant are sent to both main and local controller, but our priority is to use the control signal generated by the main controller that is more advanced controller. Once the attack has been detected, and the alarm signal has beet triggered, we will switch to the ancillary controller. Ancillary controller is implemented as a Linear–quadratic regulator (LQR). We refer to [ATFK21] for a full detail of the mitigation algorithm. LQR is a simpler controller than MPC and it requires little computational capacity, allowing it to be implemented in the physical domain.

[4] https://developers.google.com/protocol-buffers/

[5] https://grpc.io/



(a) Security framework overview.

(b) Test-bed overview.

Figure 2: Results for ramp attack with slope=0.001.

3 Experiments

We design the attack based on section 2.4 and start applying it on the position signal at the 14500th sample. Figure 2 shows effects of this attack on the system in the presence and absence of our security framework. As it is seen in Figure 2a, Our intrusion detection can detect this attack really fast and based on this detection, our mitigation will be activated quickly. So, as the blue line in Figure 2b shows, the attack tries to deviate the ball from its setpoint, but by activating the mitigation part and switching to ancillary controller, we can move the ball back to its setpoint. Otherwise, in the absence of mitigation method, as the red line shows, the attack will move the ball to the end of the beam and finally cause the ball to fall off.

Acknowledgements: This paper was supported by the Celtic-Next project 5G PERFECTA funded by Vinnova, and the SSF project SEC4FACTORY under grant no. SSF RIT17-0032. The authors are part of the Excellence Center at Linköping-Lund on Information Technology (ELLIIT) strategic research area, and the Nordic University Hub on Industrial IoT (HI2OT) funded by NordForsk. Maria Kihl is partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The work of Emma Fitzgerald was supported by the National Science Centre, Poland, under the grant no. 2019/35/D/ST7/00350 “Quality of Service in IoT Applications Using Large Scale Antenna Systems”.

Bibliography

[ATFK21] F. Akbarian, W. Tärneberg, E. Fitzgerald, M. Kihl. A Security Framework in Digital Twins for Cloud-based Industrial Control Systems: Intrusion Detection and Mitigation. In *2021 IEEE 26th International Conference on Emerging Thecnologies and Factory Automation (ETFA), in press*. 2021.

[STÅK20] P. Skarin, W. Tärneberg, K.-E. Årzén, M. Kihl. Control-over-the-cloud: A performance study for cloud-native, critical control systems. In *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*. Pp. 57–66. 2020.