2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy. October 6-9, 2019

MimePot: a Model-based Honeypot for Industrial Control Networks

Giuseppe Bernieri[1], Mauro Conti[1], and Federica Pascucci[2]

***Abstract*— Complex and heterogeneous systems characterize the *Industry 4.0*. Due to the Information Technology (IT) convergence towards the Operational Technology (OT), the development of innovative cyber-physical security tools represents a milestone for the Industrial Control Systems (ICSs) protection. In this context, honeypots are systems used as decoys to detect and analyze malicious actions. However, industrial networks require specific honeypot development capabilities.**

In this work, we present *MimePot*, a cyber-physical honeypot conceived for industrial control networks. Compared to classic honeypots, *MimePot* offers a model-based approach: it is able to simulate physical processes to lure skilled attackers targeting industrial plants. Moreover, *MimePot* uses the Software Defined Networking (SDN) technology to provide a consistent future proof security approach. We demonstrate the usefulness of *MimePot* performing data integrity attacks against a water distribution system in a simulated environment.

I. Introduction

Nowadays, ICSs represent the backbone of Critical Infrastructures (CIs), where the control processes characterize safety-critical applications for the main national assets (e.g., electric power distribution, water distribution systems). For these complex systems, disruptions can have a significant impact on public health and the environment, leading to disasters and large economic losses. Thus, security is crucial to guarantee safe operations: the research community acknowledged the importance of addressing the challenge of designing secure ICSs [1]. Due to the integration of networks within the physical processes' control loops, we assisted to the evolution of industrial systems from a "monolithic" design to interconnected Cyber-Physical Systems (CPSs). These systems are useful for the management of complex scenarios, such as CIs, but vulnerable to cyber-physical threats. Skilled attackers exploit control theory knowledge as well as cyber hacking to perform cyber-physical attacks, specific cyber attacks with physical consequences on the processes.

The most famous threat targeting CIs was Stuxnet [2]. This worm disclosed to the world that there are groups with the motivation, skills, and resources to set up sophisticated computer-based attacks against CIs. The ultimate goal of Stuxnet was to sabotage the industrial control assets by reconfiguring Programmable Logic Controllers (PLCs) to operate out of their normal procedures. HAVEX [3], Industroyer [4], and more recently, TRITON/TRISIS/HatMan [5] represent other dangerous ICS-tailored malware. The latter shows the existence of attacks that threaten control devices used as last line of industrial safety defense, such as the industrial safety systems.

In this context, the detection of complex cyber-physical threats in industrial networks represents a challenge for security vendors and researchers. Advanced attacks have the ability of remain dormant without rising any security alerts until problems on plant processes occur. It is therefore necessary to be able to exploit every possible tool that allows to increase the possibilities of plant safeguarding. In the computer science field, there are several and well established detection and countermeasure methodologies to defend against cyber-attacks. On the other hand, in industrial scenarios, there are approaches to detect anomalies related to malfunctions or failures at the plant level but specific cyber-physical security methodologies are still missing.

Honeypots are formally defined as monitored computing resources that can be probed, attacked, or compromised by malicious cyber actors [6]. These systems are by definition more vulnerable with respect to real systems, their goal is to attract possible cyber-attacks replacing and safeguarding the real devices.

In this work, we present a cyber-physical honeypot, namely *MimePot*, that can be connected directly to the plant network with the capabilities of simulating the physical processes operations. With *MimePot* it is possible to mislead adversarial actions and, at the same time, study the behavior of skilled malicious actors: *the more interaction you allow, the more you can learn*. Our final aim is to develop a framework able to capture complex cyber-physical attacks behaviors, such as Zero Dynamics Attacks [7]. For this class of threats, the attacker performs cyber-physical attacks verifying the consistency of the physical behavior of the system under attack.

The main contributions of this paper are summarized as follows:

- we draw the attention to the industrial honeypots development that will be of considerable importance for the *Industry 4.0*;
- we illustrate the importance of model-based approaches for the cyber-physical security of CI control networks;
- we present *MimePot*, a model-based honeypot for CI scenarios conceived to lure the attackers using the SDN technology;
- we validate *MimePot* by analyzing experimental results on a proof-of-concept simulated system under adversarial actions.

The rest of the paper is organized as follow. Section II discusses the related work. In Section III, we present *Mime-*

[1]Giuseppe Bernieri and Mauro Conti are with the Department of Mathematics, University of Padua, Padua, Italy. {bernieri, conti}@math.unipd.it.

[2]Federica Pascucci is with the Department of Engineering, Roma Tre University, Rome, Italy. federica.pascucci@uniroma3.it.

Pot. In Section IV, we evaluate our honeypot using a water distribution system case-study under cyber-physical attacks. Section V concludes the paper.

II. Related Work

Honeypots are systems usually more vulnerable than real infrastructures and are used to attract and lure attackers. Cyber-attacks against these systems do not affect the normal functioning of real systems. Honeypots can be designed with low-interaction, high-interaction, and hybrid-interaction capabilities, based on the development complexity level they offer. Low-interaction honeypots are easy to implement but with basic network services; high-interaction honeypots are more complex to implement with advanced network services coupled with real devices; hybrid-interaction honeypots are a combination of the first two.

The development of specific ICS honeypots represents an interesting topic for the security research community, but the importance of creating accurate plant physical models to effectively manage cyber-physical correlations is missing. The work in [8] provides a comprehensive analysis of the visibility of unprotected ICS traffic across network domains, highlighting the growing trend of not controlled interconnection of industrial network services. One of the first examples of specific ICS honeypots is presented in [9], where the developers investigate the feasibility of building a software-based framework to simulate a variety of industrial networks. Within the "Honeynet Project", a low interactive server side ICS honeypot, namely *Conpot*, has been designed to be easy to deploy and extend [10]. It is important to highlight that our *MimePot* can improve the *Conpot* deployment integrating physical processes simulation. *GasPot* [11] represents a functional gas-tank-monitoring systems honeypot that logs connections and attempts to compromise. A limitation of *GasPot* is the use of random values for the physical simulation considering plausible ranges. In [12], the authors present an ICS honeypot architecture showing the ability of attracting multiple potentially malicious connections. During the experiments, the honeypot was regularly probed by Shodan[1], a search engine for Internet-connected devices. In [13], the authors present *Honeyd+*, an evolution of the *Honeyd* honeypot framework presented in [14]. This work discusses pilot studies to determine the feasibility of using *Honeyd* as an industrial control system honeypot exploiting real PLC devices for the evaluation. In [15], the authors design and implement the *Crysys PLC Honeypot (CryPLH)* to detect targeted attacks against ICSs. *CryPLH* simulates the behavior of real PLCs without consistent physical processes simulation interaction capabilities. In [16], the authors propose the design of a realistic virtual ICS honeypot developed on top of *MiniCPS* [17], a toolkit for CPSs simulation. This ICS honeypot can be managed with SDN controllers. The fusion between *MimePot* and the honeypot they propose can be useful to improve the model-based simulation exploiting the *MiniCPS* toolkit capabilities. The physical system evolution for ICSs is treated with particular importance in [18]. In this work, the authors implement a nonlinear control system emulating a three-water tank system with the associated sensors, actuators, and control devices. While the approach is very interesting, the authors are not using SDN technology in their setup. In [19], the authors present the *Honeyphy* framework to develop a CPS honeypot. This work correctly identifies the importance of the data coming from the physical systems within the honeypot. However, this work does not consider a possible implementation with SDN technology. In [20], the authors present a SDN-based honeypot for Internet of Things. The honeypot is able to reroute the traffic coming from suspicious nodes and mislead the attackers using phantom nodes to simulate physical processes behaviors. Compared to this work, we investigate the effect of data modification attacks directly on the physical simulation in order to study how to lure attackers with advanced knowledge of physical processes evolution.

In this paper, we address the relations between cyber and physical approaches for ICS honeypots development, fundamental for the *Industry 4.0* paradigm. Our work extends capabilities of already deployed honeypots for ICSs. In fact, our main goal is to give consistency to the model-based physical simulation in order to lure skilled hackers able to perform attacks targeting the data integrity of physical values, such as Zero Dynamics Attacks [7].

III. MimePot

In this Section, we present *MimePot*, a model-based honeypot conceived for ICSs. In the following subsections, we describe the *MimePot* modules together with the physical and cyber simulation components, highlighting the main characteristics.

A. MimePot modules

MimePot needs to be a realistic and attractive target for potential malicious actors. To this end, physical processes and control routines are separated into two modules:

- *Mime Plant*: is the network node that implements physical plant processes simulations. It can be considered as a PLC interacting with physical plants. Model-based processes are simulated inside this module.
- *Mime E&C*: represents the module designed for the control routines computation, where E&C stands for Estimation & Control. It can be considered a Supervisory Control And Data Acquisition (SCADA) workstation. The controller manages and regulates the behavior of the plant processes according to the project specifications. The estimator updates the control vectors.

B. MimePot physical components

MimePot implements the plant simulation for a selected subset of physical processes, with the related estimation and control routines. Depending on the modeling aims, different approaches can be adopted to mathematically describe the physical plant. According to [21], the common methodology to represent a system is by using Linear Time Invariant

[1]https://www.shodan.io

(LTI) models. The simulated plant state-space equations for *MimePot* are:

$$x_{k+1}^M = A^M x_k^M + B^M u_k^M + w_k^M, \quad (1)$$

$$y_k^M = C^M x_k^M + v_k^M, \quad (2)$$

where $x^M \in \mathbb{R}^n$ is the mimed state vector, $u^M \in \mathbb{R}^p$ is the mimed control vector, and $y^M \in \mathbb{R}^q$ is the mimed output vector. Matrices A^M, B^M, C^M, are real constant matrices, and are defined as follows: $A^M \in \mathbb{R}^{n \times n}$ represents the state transition matrix, $B^M \in \mathbb{R}^{n \times p}$ is the input matrix, $C^M \in \mathbb{R}^{q \times n}$ is the output matrix. In *MimePot* the feedthrough matrix (i.e., $D^M \in \mathbb{R}^{q \times p}$) is not considered because in general cases physical systems are strictly causal. The vectors $w^M(\cdot) \sim N(0, Q)$ and $v^M(\cdot) \sim N(0, R)$ are identical independent Gaussian noises with zero mean and variance respectively Q and R.

For the *Industry 4.0* scenario, the discrete LTI system model formalized above changes, introducing the network communications. The use of networks on ICS scenarios increases the performances of the systems, reducing the implementation costs of new devices. On the other hand, the use of networks to decentralize traditional control systems increased the design complexity of ICSs and opened a wider cyber-attack surface.

According to this networked ICS scenario, the mimed state vector (1) becomes:

$$x_{k+1}^M = A^M x_k^M + B^M \tilde{u}_k^M + w_k^M, \quad (3)$$

where $\tilde{u}_k^M$ represents the control vector passing through the communication channel defined as follows:

$$\tilde{u}_k^M = u_k^M + u_k^c + u_k^a, \quad (4)$$

where u_k^c is the quantization error and u_k^a is the attack vector. Also the sensor readings pass through the network and the mimed output vector (2) becomes:

$$\tilde{y}_k^M = y_k^M + y_k^c + y_k^a, \quad (5)$$

where y_k^c is the quantization error and y_k^a is the attack vector.

C. MimePot cyber components

From the cyber point of view, *MimePot* can be implemented in virtualized environments where the *Mime Plant* and the *Mime E&C* communicate through channels that make use of industrial protocols. As in a normal client/server implementation, data is interchanged between *Mime Plant* and *Mime E&C* modules to replicate sensors and actuators behaviors managed through real industrial communication networks. These two modules need to be implemented on two different network nodes in order to be considered real plant devices from the attacker. The network traffic load coming from and to the *MimePot* modules is not critical because it replicates a small part of the physical processes characterizing the entire plant. The design of the plant model needs to be related to the real plant but it has to be different: network traffic has to be considered as normal industrial traffic by an attacker but the real topology, physical processes behaviors, and configurations are "mimed" (i.e., fake). For instance, in a *MimePot* implementation, the plant processes are simulated considering real physical laws but using imaginary systems and topologies.

In order to divert the malicious traffic to *MimePot*, the methodology conceived includes the SDN paradigm. The SDN [22] represents a network architectural approach consisting in the *control plane* separation from the *data plane*: the first one manages the network traffic forwarding, while the second one effectively forwards data to connected nodes. With respect to conventional network traffic filtering mechanisms, the SDN technique allows granular control of the traffic flows inside specific *OpenFlow* switches. The rules defined by the SDN Controller allow to carry out traffic forwarding operations with a high level of detail. In a possible threat scenario, the SDN Controller implements a white-list of all the licit devices network addresses present in the real plant network in order to perform two defensive actions:

- redirection of the malicious traffic to *MimePot*;
- camouflaging of the network addresses of the real devices to make the attacker believes that he/she is attacking the real plant.

According to these actions, the attacker continues to believe in a direct connection with the real plant, but on the contrary, he/she is connected to the *MimePot* due to the traffic redirection ability of the SDN switch. At the same time, the plausible physical behavior implemented within the *MimePot* permits to lure the control theory skills of the attacker.

Thanks to the SDN redirection capability, *MimePot* starts to silently study the attacker behavior without interrupting the malicious connection. In this way, *MimePot* permits to perform advanced threat intelligence and forensic analyses of the attacker. When facing a skilled attacker, it is possible to reconstruct the attack stages and the methodologies exploited in order to enhance malicious activities knowledge to better address cyber-physical security of ICSs in advanced zero-day attack situations. It is important to notice that legacy switches and routers are not compatible with SDN technology. For this reason, specific *OpenFlow* devices must be used to develop the *MimePot* implementation with SDN capabilities.

IV. Evaluation

In this Section, we present preliminary results for the proof-of-concept implementation of *MimePot*. We consider a water distribution system as target industrial scenario for our evaluation. In the following subsections, we describe the *MimePot* cyber-physical implementation.

A. MimePot: Cyber-Physical simulation

In order to perform preliminary tests for the *MimePot* development, we consider a small part of the water distribution industrial scenario. In particular, we define subprocesses implemented and we gather physical equations characterizing the scenario. In Fig. 1, we present the *MimePot* high-level functioning scheme. We consider a three tanks scenario for the physical process simulation implemented in the *Mime*

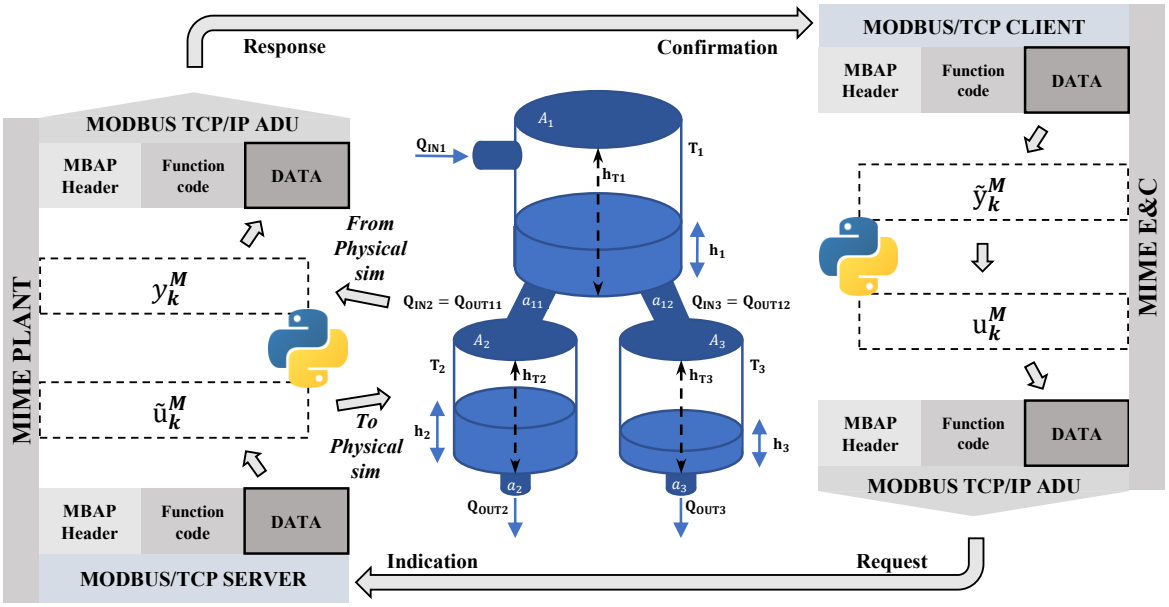

Fig. 1: *MimePot* high-level functioning with the water distribution system scenario.

Plant. One of the tanks ($T1$) is placed in a higher position with respect to the others (we consider the gravitational acceleration for the experiments) and it represents a reservoir. A water pump (P) refills the reservoir according to predefined rules. The remaining two tanks ($T2$ and $T3$) represent water consumers, having different consumption rates. Moreover, it is possible to model different proportional water flows from the reservoir to the consumers.

For the evaluation, we implement the *Mime Plant* and the *Mime E&C* using python and Mininet[2]. The Modbus/TCP [23] represents a largely used protocol in the real industrial scenarios and we use it for the simulation. It is important to highlight that it is possible to use different industrial protocols for the *MimePot* implementation. Once the TCP connection is established, *Mime Plant*, acting as Modbus/TCP Server, and *Mime E&C*, acting as Modbus/TCP Client, begin to communicate ready to lure attackers. The Client sends two kind of Modbus/TCP Request to the Server: *Read Input Register (Function Code: 04)* and *Write Single Coil (Function Code: 05)*. With the first one, the Client makes requests to the Server for the actual sensor status, while with the other, the Client sends acting commands (ON/OFF) to the Server in order to apply control operations. From the *Mime Plant* side, two Modbus/TCP responses are possible: sensor data, packed as two bytes per register, in the *Read Input Register* response message or an echo of the *Write Single Coil* request, returned after the coil status update. On both nodes, physical values, consistent with the simulation, are encapsulated inside the data fields of the Modbus TCP/IP Application Data Unit (ADU) using *Scapy*[3], a python-based interactive packet manipulation library. According to Fig. 1, the *Mime Plant* computes the simulated sensor values y_k^M and then encodes the data inside Modbus/TCP packets. The vector $\tilde{y}_k^M$ is than received by the *Mime E&C* and analyzed by the estimation and control routines implemented. Subsequently, the control vector u_k^M is encapsulated inside data field of Modbus/TCP Request packets and sent to the Modbus/TCP Server. The latter receives the packets and extracts $\tilde{u}_k^M$ vectors, in order to apply the necessary control operations to the simulated plant.

Fig. 2 depicts a normal situation scenario lasting $t = 100\ s$ of *MimePot*. For the three tanks, it is possible to observe the physical simulated status of the *Mime Plant*, the sensor values received, and the status of the actuators in the *Mime E&C*. The values in the graphs are extracted from the Modbus/TCP traffic. For the normal scenario conceived, the initial water levels are $h_{T1}(0) = 15000\ mm$, $h_{T2}(0) = 5000\ mm$, and $h_{T3}(0) = 3000\ mm$. The pump P remains always active and once the maximum water level set for $T2$ and $T3$ is reached (i.e., $10000\ mm$) at $t \approx 45\ s$, the *Mime E&C* sends the commands to the *Mime Plant* for closing the valves a_{11} and a_{12} that connect respectively $T1$ with $T2$ and $T1$ with $T3$. This simple scenario can be modified according to the *MimePot* implementation requirements.

B. Attack simulated scenario

We conceived *MimePot* to improve the security of modern industrial networks. In particular, our aim is to lure skilled attackers with control theory knowledge and the ability to verify the physical process model under attack.

We assume *MimePot* and the attacker nodes connected to a central switch on the plant network, in addition to

[2]http://mininet.org/

[3]https://github.com/secdev/scapy

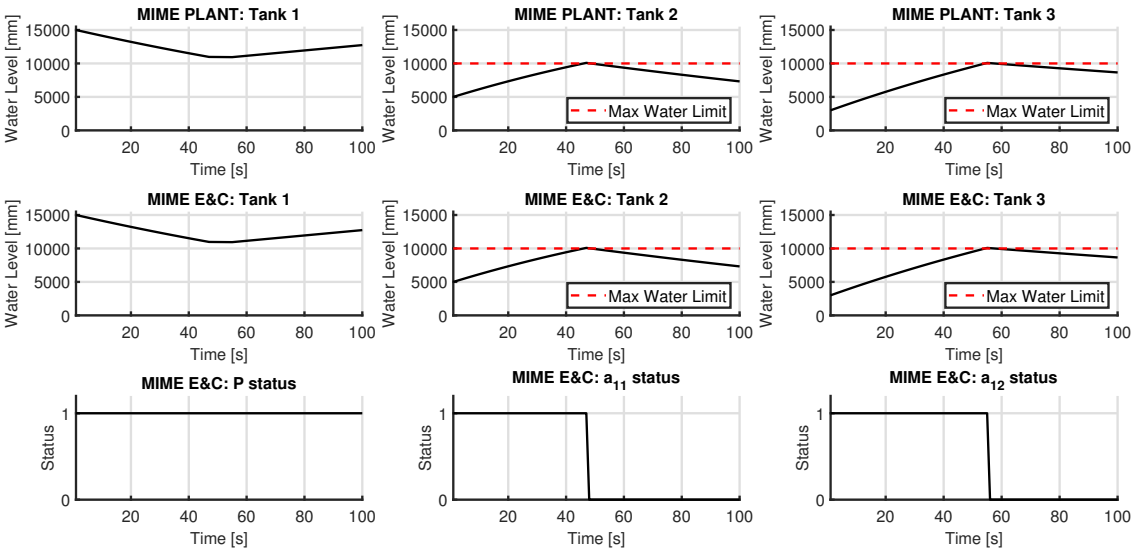

Fig. 2: *MimePot* under normal conditions.

the normal industrial nodes (e.g., PLC, Human Machine Interfaces, SCADA Workstation).

Fig. 3 describes the stages performed by the attacker for an advanced data integrity attack. In this configuration, the attacker exploits a Man In The Middle (MITM) attack (i.e., ARP spoofing) to intercept communication between *Mime Plant* and *Mime E&C*. The attack is composed of two stages: *Model Verification* and *Integrity Attack*. During the *Model Verification* stage, the attacker performs advanced reconnaissance by sniffing the control network traffic and evaluates the physical plausibility of data hijacked. Once verified that the control network traffic is pertinent to the plant processes, the attacker starts the *Integrity Attack* stage by modifying data directed to the actuators (i.e., $\tilde{u}^M$) and data coming from the sensors (i.e., $\tilde{y}^M$). The final aim of this kind of advanced threat is to change plant behavior without the security operators notice it.

In Fig. 4 we present the results of a data integrity attack against *MimePot*. The traffic generated by the attacker is redirected to the *MimePot* thanks to the SDN Controller. Once completed the *Model Verification* stage, the attacker starts to modify actuator commands $\tilde{u}^M$ directed to the *Mime Plant* with malicious commands u^a and at the same time he/she spoofs the sensor values $\tilde{y}^M$ directed to the *Mime E&C* with small modification y_a^M in order to evade detection techniques. In the *Mime Plant*, at $t \approx 45\ s$ and $t \approx 55\ s$, $T1$ and $T2$ overflow respectively, while the *Mime E&C* continues to perceive normal conditions. The attacker is lured because he/she sees what he would expect from a physical evolution point of view.

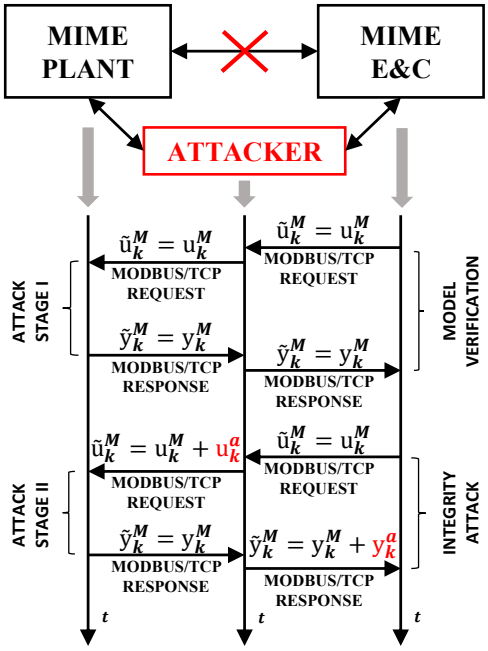

Fig. 3: *MimePot* under MITM attack scheme.

V. Conclusion

In this paper, we presented a model-based ICS honeypot able to lure skilled cyber-physical attackers. We used a water distribution system simulation as a proof-of-concept to present preliminary results. The integration of the SDN technology with *MimePot* adds value and degrees of freedom for future advanced cyber security implementations. With this work, we demonstrate the effective necessity of innovative security systems for the *Industry 4.0* paradigm.

In future work, we will consider hybrid interactions for the honeypot with real cyber-physical testbeds. Moreover, extensive implementation experiments of SDN with *MimePot* will be addressed using real SDN-ready switches to evaluate

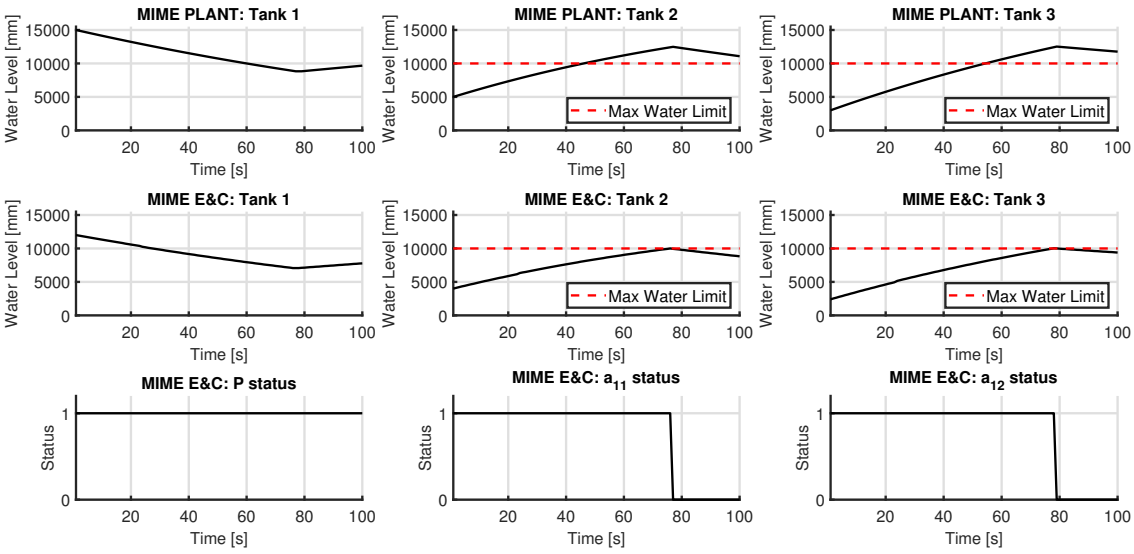

Fig. 4: *MimePot* under MITM attack.

the network traffic load coming from and to the *Mime Plant* and the *Mime E&C* modules. We will also implement the automatic generation of rules for intrusion detection systems, based on the threat intelligence activities performed by *MimePot*.

ACKNOWLEDGMENT

This work is supported by a grant of the Italian Presidency of the Council of Ministers.

REFERENCES

[1] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems." in *HotSec*, 2008.
[2] N. Falliere, L. Murchu, and E. Chien, "W32. Stuxnet Dossier," Symantec, Tech. Rep. 1.4, February 2011.
[3] J. Rrushi, H. Farhangi, C. Howey, K. Carmichael, and J. Dabell, "A quantitative evaluation of the target selection of havex ics malware plugin," in *Industrial Control System Security (ICSS) Workshop*, 2015.
[4] A. Cherepanov, "Win32/industroyer: a new threat for industrial control systems," *White paper, ESET (June 2017)*, 2017.
[5] "TRISIS Malware - Analysis of Safety System Targeted Malware (https://dragos.com/blog/trisis/TRISIS-01.pdf)," Dragos Inc., Tech. Rep., 2017.
[6] N. Provos, "A virtual honeypot framework." in *USENIX Security Symposium*, vol. 173, 2004, pp. 1–14.
[7] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st international conference on High Confidence Networked Systems*. ACM, 2012, pp. 55–64.
[8] M. Nawrocki, T. C. Schmidt, and M. Wählisch, "Uncovering Vulnerable Industrial Control Systems from the Internet Core," Open Archive: arXiv.org, Technical Report arXiv:1901.04411, January 2019. [Online]. Available: https://arxiv.org/abs/1901.04411
[9] V. Pothamsetty and M. Franz, "Scada honeynet project: Building honeypots for industrial networks," *Cisco Systems, Inc.,[Online]. Available: http://scadahoneynet. sourceforge. net/.[Accessed 21 August 2016]*, 2005.
[10] "Conpot Honeypot, (https://github.com/mushorg/conpot)." [Online]. Available: https://github.com/mushorg/conpot
[11] K. Wilhoit and S. Hilt, "The gaspot experiment: Unexamined perils in using gas-tank-monitoring systems," *Trend Micro*, vol. 6, 2015.
[12] A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "A flexible architecture for industrial control system honeypots," in *e-Business and Telecommunications (ICETE), 2015 12th International Joint Conference on*, vol. 4. IEEE, 2015, pp. 16–26.
[13] M. Winn, M. Rice, S. Dunlap, J. Lopez, and B. Mullins, "Constructing cost-effective and targetable industrial control system honeypots for production networks," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 47–58, 2015.
[14] N. Provos, "Honeyd-a virtual honeypot daemon," in *10th DFN-CERT Workshop, Hamburg, Germany*, vol. 2, 2003, p. 4.
[15] D. I. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer, "Cryplh: Protecting smart energy systems from targeted attacks with a plc honeypot," in *International Workshop on Smart Grid Security*. Springer, 2014, pp. 181–192.
[16] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards high-interaction virtual ICS honeypots-in-a-box," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM, 2016.
[17] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," in *Proceedings of the First ACM workshop on cyber-physical systems-security and/or privacy*. ACM, 2015, pp. 91–100.
[18] A. F. Murillo, L. F. Cómbita, A. C. Gonzalez, S. Rueda, A. A. Cardenas, and N. Quijano, "A virtual environment for industrial control systems: A nonlinear use-case in attack detection, identification, and response," in *Proceedings of the 4th Annual Industrial Control System Security Workshop*. ACM, 2018, pp. 25–32.
[19] S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos, and R. Beyah, "Rethinking the honeypot for cyber-physical systems," *IEEE Internet Computing*, vol. 20, no. 5, pp. 9–17, 2016.
[20] H. Lin, "Sdn-based in-network honeypot: Preemptively disrupt and mislead attacks in iot networks," in *1st International Workshop on Security and Privacy for the Internet-of-Things (IoTSec)*. ACM, 2018.
[21] S. Ding, *Model-Based Fault Diagnosis Techniques. Design Schemes, Algorithms and Tools*, ser. Advances in Industrial Control. Springer, 2013.
[22] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
[23] I. Modbus, "Modbus messaging on TCP/IP implementation guide," *v1.0b*, 2004.