FingerCI: Generating Specifications for Critical Infrastructures

Filipe Apolinário*, Nelson Escravana*, Éric Hervé§, Miguel L. Pardal¶, Miguel Correia¶

*INOV-INESC INOVAÇÃO §Alstef Group ¶INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
R. Alves Redol 9, 1000-029, Lisbon, Portugal*¶, 104 Bd de la Salle, 45760 Boigny-sur-Bionne, France§
filipe.apolinario@tecnico.ulisboa.pt,nelson.escravana@inov.pt,eric.herve@alstefgroup.com
miguel.pardal@tecnico.ulisboa.pt,miguel.p.correia@tecnico.ulisboa.pt

ABSTRACT

Cyber-physical attacks on critical infrastructures (CI) or industrial control systems (ICS) can compromise the integrity and operability of physical systems, potentially damaging critical facilities. Specification-based Intrusion Detection Systems (IDSs) can detect those attacks but often require an accurate specification of the monitored ICS, which is often a deterrent to their usage. This paper presents FingerCI, a solution to automatically generate a model of an ICS, which we name a fingerprint, based on network traffic inspection, business process discovery, and physical behaviour analysis. An airport baggage handling system testbed shows that the fingerprints can be used to configure specification-based IDS with high accuracy results, reducing the amount of effort required to use that detection approach.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**;

ACM Reference Format:
Filipe Apolinário, Nelson Escravana, Éric Hervé, Miguel L. Pardal, Miguel Correia. 2022. FingerCI: Generating Specifications for Critical Infrastructures. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event, .* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3477314.3507323

1 INTRODUCTION

The use of Information Technology (IT) in critical infrastructures (CI), or industrial control systems (ICS), has increased quality of service and reduced operational costs, but also exposed them to *cyber-physical attacks* [10]. Although *airport cybersecurity* has been able to withstand the increase of cyberattacks, airport systems are considered at high risk of being attacked. *Baggage handling systems* (BHS) ensure that bags left at check-in counters are properly screened and routed to the correct destinations. On a recent report [2], ENISA stated that BHSs are highly critical in the context of airport security.

Conventional *intrusion detection* fails to detect these cyberthreats, missing the opportunity to trigger a prompt reaction. This happens mainly because attacks on ICS devices often abuse *legitimate actions* to lead the ICS into an invalid physical state [1]. Examples include continuously turning off physical equipment to cause denial-of-service, or altering configurations, like equipment operation speed, to compromise quality of service and safety. These actions in the right context are legitimate, so many intrusion detectors are not able to flag them as malicious [3–8, 11, 12, 14–16, 21, 22]. *Specification-based IDSs* (SBIDSs) detect deviations from a specification of the behavior of the system or protocol, so their alarms are interpretable and accurate. However, these specifications are created by humans, which is deterrent to their use because of their high cost.

A recent approach to deal with these limitations of SBIDSs is to use *business processes* (BP) as the specifications [9, 14, 17]. These IDSs validate if the actions being executed are legitimate by checking if they conform to the BP. This approach reduces the amount of information the human expert needs to introduce into the system, and offers good readability of the resulting specification. However, they still require knowledge about the specification of the infrastructure being assessed. Namely, they require knowledge about network infrastructure (hosts addresses and communication protocols) and BPMN (Business Process Modelling Notation)[1] specifications of the processes that must be protected, which are often difficult to obtain automatically by inspecting the monitored environment.

This paper presents FingerCI, a solution to automatically construct a *fingerprint* specification of an ICS based on network traffic inspection. In other words, a model of the normal behaviour of the ICS system. This model can be used as configuration for a SBIDS while maintaining the readability of BP approaches to still allow human experts to interpret and perform corrections on the resulting specification. FingerCI performs *network reconnaissance* based on protocol dissection to interpret ICS network protocols and extract information about the ICS infrastructure, including devices and communications. The information gathered is further analysed using *business process model discovery* techniques [23] to build a business process representing all the activities in the ICS network. FingerCI also constructs a *behaviour model* that infers the possible conditions and number of times activities should occur.

An experimental evaluation on the proposed solution was conducted using a high-fidelity BHS simulation platform, commonly used by BHS providers to test their systems on contractual operating conditions. Within the scope of the experimental evaluation, FingerCI was used to generate a fingerprint of the accepted network behaviour of the BHS equipment and validate the correctness of the BHS sortation and screening services. FingerCI was integrated with BP-IDS, a BP-based SBIDS [14, 15, 20]. A FingerCI fingerprint was used to configure BP-IDS, that was installed on the BHS network to detect anomalous activity. The integration between FingerCI and BP-IDS allowed this SBIDS to detect anomalous bags circulating on the BHS with 100% accuracy.

[1] http://www.bpmn.org/

SAC '22, April 25–29, 2022, Virtual Event,

ACM ISBN 978-1-4503-8713-2/22/04.
https://doi.org/10.1145/3477314.3507323

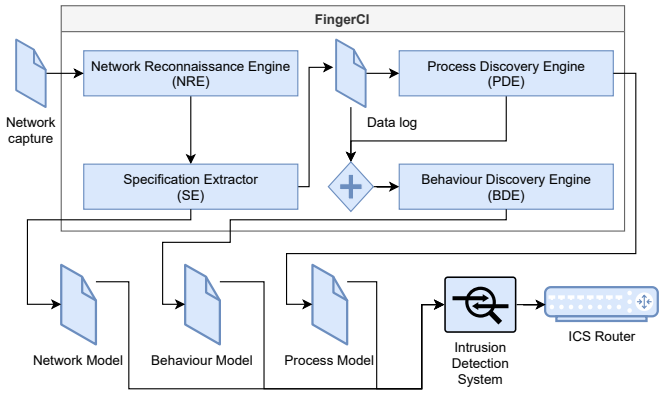

Figure 1: FingerCI analysing an ICS network.

2 FINGERCI

This work aims at creating a fingerprint representation of processes and devices present on an ICS infrastructure. To do so, the proposed solution considers an ICS architecture and, by inspecting router traffic, creates fingerprints of the interactions between supervisory and control layer devices.

As illustrated in Figure 1, the solution is composed of four components that work in sequence to produce a specification. The first component, *Network Reconnaissance Engine* (NRE), receives captures of communications between the IT components of the monitored infrastructure, parses those captures using protocol dissection techniques, and extracts information about the network and physical components of the infrastructure. The second component, *Specification Extractor* (SE), uses the output of the NRE to build a network topology and a process specification based on feature extraction methods. The third component, *Process Discovery Engine* (PDE), uses the features selected by previous component and produces business processes, using process discovery techniques. The fourth component, *Behaviour Discovery Engine* (BDE), analyses the process model according to features extracted by SE and produces a behaviour model, that reflects the correct conditions of ICS physical devices before and after activities are executed. The solution involves doing deep packet inspection of the ICS communication protocols, and produces valid process and behaviour models representing a fingerprint of the whole ICS.

The **NRE component** is designed as a network analyser tool that *dissects ICS communications* present in packet captures, and extracts information about network and industrial control devices based on the inspected packets. The output of this component is a log of all packets captured and their attributes.

Dissection of ICS communication can be seen as two related tasks: to identify ICS network device interaction packets based on the protocol stack; and to extract profiling information about the interaction taking place, based on the information present in the network packet. The solution addresses both tasks by using a set of dissection rules based on network *protocol specification standards*, usually published as RFCs (request for comments) in case of IT network protocols and as International Air Transport Association Resolution Manual in case of airport specific network protocols. The *dissection rules*, classify network messages being exchanged according to protocol identifiers.

Once dissection rules have properly identified all communication protocols enclosed in the packet, the NRE component analyses the packets that use airport specific network protocols. The component resorts to deep packet inspection for attribute extraction to properly identify industrial control devices and profile the operations present in each network packet. In this case, it considers the protocol structure described by the dissection rules of the packet and extracts the protocol attributes. The component extracts network information to profile the devices, such as device identifiers (e.g., Internet Protocol (IP) and media access control addresses (MAC)), and information about the ICS cyber and physical devices from industrial protocols. Also, the NRE component pinpoints physical features about the ICS devices and operations issued present on network packets. Namely, for each extract, the operation (read/write), the ID (identifier) of the operation, the name(s) of the physical devices accessed and their state (based on variable(s) accessed and their values) are identified.

The **SE component** uses the packet log produced by NRE to build the network model and identify an activity log that contains all business processes and activities observed in the packets.

To build the network topology, SE traverses the packet log and creates a dependency graph. For each different network protocol used between two hosts, an association between those hosts is placed in the graph. For each ICS network packet describing operations performed over physical devices, an association between the executor IT device and the physical device is also placed in the graph. The resulting graph is the complete ICS infrastructure *network specification*.

The activity log for process discovery is built in two steps: business activity identification and process instance inference. In the first step, *business activity identification*, SE inspects each entry of the log (data packet) attributes and for each industrial control protocol creates a representation of an activity by concatenating the NRE attributes into an unique *activity identification pattern*. This identifier includes, network information (source/destination IP address and network protocol used) and information about the ICS action (type of operation performed and variable(s) accessed).

The **PDE component** inspects business activities and corresponding process instances extracted from the previous components, and creates a *process model* [24] which is a generalized specification of a log that summarizes the order of which activities should executed to achieve process correctness (most common way to represent these models are BPMN, Petri nets or process trees). This generalization into a process model is achieved using inductive miner process discovery technique that studies the relationships between activities based on the order of occurrence, and compile them into a generic Petri Net or BPMN process representations.

The *inductive miner* [26] used by this component to construct process models follows a two-step procedure. The first step, involves classifying process instances according to activity patterns. The second step, involves identifying the conditions for those patterns to occur. The first step identifies the sequence of activities while the second step determines the conditions to which activities can occur on a given process instance.

The **BDE component** extracts behaviour rules for an activity to be considered as a *legitimate action* (e.g., bags can only be routed to the destination flight if they have clear screening result). The rules are extracted in two stages, the activity and gateway condition extraction

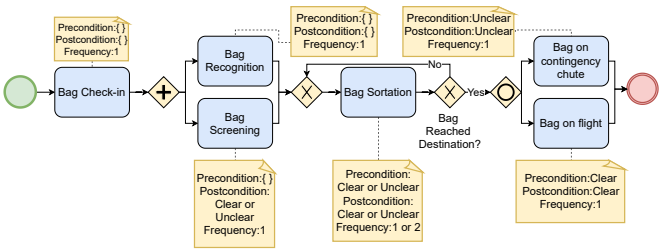

Figure 2: Example of a FingerCI specification for a BHS (process model in blue, and behaviour model in yellow).

stages. The resulting behaviour model contains the validation rules required for the IDS to validate the process activities.

At the first stage, activity extraction is performed by traversing the activity log obtained in the specification extractor component and registering as activity *preconditions* the physical state of the referenced device before the action took place, and registering as post-condition the device physical state after the action took place.

At the second stage, the gateway extractor stage, BDE specifies the necessary conditions for gateway validation, which include activity *frequency counting* to identify the number of times activities can occur on gateway loops, and branch precondition discovery by evaluating based on the activities preconditions present in the branch. The resulting process behaviour model is in Figure 2 for the example log of a baggage handling system (BHS). The model provides the validation rules for intrusion detection systems to detect non-legitimate actions performed on the ICS physical systems.

3 EVALUATION

The evaluation of FingerCI is based on the Airbus simulation platform[2]. This platform is a training environment that offers virtualized airport systems digital twins. The systems installed on the platform include an airport network infrastructure (with virtual machines connected on VLANs), with virtual machines mimicking airport cyber systems and physical hardware machines (e.g., PLCs with sensors and actuators) to reproduce the overall functioning of a real BHS. The FingerCI evaluation focused on three airport systems represented in the simulation platform (the FIMS, the AODB and BHS) manufactured by Alstef[3] [4]. The evaluation assessed the quality of FingerCI fingerprints for validating BHS service operations. To validate the accuracy of its fingerprints, FingerCI was integrated with Business Process Intrusion Detection System (BP-IDS). BP-IDS is a SBIDS that uses the network topology, business process specification and information about the conditions for critical operations to be executed, to model the accepted behaviour of an ICS infrastructure. BP-IDS detects in real-time machine abnormal operations on ICS services by comparing the behaviour observed by network sensors with its specification.

Under this setting, the experiments conducted to evaluate FingerCI used network packet captures (pcaps) recorded on the Airbus simulation platform of the baggage handling system operating continuously during 25 hours. This simulation included a virtualized airport database that provided to the BHS sortation unit identifiers

[2]https://airbus-cyber-security.com/products-and-services/prevent/cyberrange/
[3]https://www.alstef.com/Baggage-handling-and-screening
[4]https://alstefgroup.com/baggage-handling/software/simulation-and-emulation/

AS	Bags	Anomalies	FN	FP	Accuracy	FPR
Screening	1914	42	0	0	100%	0%
Sortation	1920	36	0	2	100%	6%
Both	1956	78	0	2	100%	3%

Table 1: SBIDS with FingerCI accuracy results

of fictitious bags and the corresponding fictitious flights assigned to physical locations of the BHS. The simulation used for this experiment, represented a high-fidelity representation of a real BHS available on airports, by using Emulate3D[5], a testing platform commonly used by BHS service providers to test their systems on contractual operating conditions before they are installed on airports.

The experiment answered the following research question: *Are specifications generated by FingerCI reliable for ICS intrusion detection?*

To validate accuracy of an SBIDS configured with FingerCI, BP-IDS was configured with the fingerprint generated by FingerCI. The fingerprint was produced by inspecting FIMS, AODB, control unit, and sortation unit interactions during one hour. During this hour, FIMS sent flight information messages, bag check-in operations were inspected, with messages about the bags sortation and screening decisions. The rest of the network traffic (24 hours) was used for monitoring the BHS, summing on a total of 1956 bags monitored. During two hours, the simulation platform was used to force the BHS to two abnormal situations. The first one, "screening anomaly", where the EDS screening results of 42 bags were changed by the simulation platform to route unclear bags to flight instead of the contingency chute,and clear bags to the contingency chute. The second abnormal situation, "sortation anomaly", the simulation platform overwrote the BHS sortation messages to falsely route bags to different flight destinations.

As can be seen in Table 1, the fingerprints generated by FingerCI for profiling the BHS behaviour were very accurate for detecting the two abnormal situations (AS) with accuracy of 100%. This is due to the fact that, the combination of process mining with the behaviour model created by FingerCI makes the verification deterministic. This reasoning can be seen by the fact that BP-IDS did not have any false negatives (FN) regarding the sortation and screening anomalies, and displayed two false positives (FP) from the 1956 bags inspected on the 24 hours of BHS functioning. The FP in this case were due to the reintroduction of anomalous bags in the system[6]. In this case, BP-IDS had information cached of the previous inspections of the sortation anomaly and considered bags as anomalous when in fact no anomaly was taking place. Due to the false positives reported in the tests, when inspected on a isolated manner, BP-IDS achieved 0% false positive rate (FPR) on the screening anomaly, while the fingerprints obtained 6% FPR for sortation anomaly, resulting on total of 3% FPR for the whole abnormal period. These results show

[5]https://www.demo3d.com/Baggage-Handling/
[6]Reintroduction of bags can often happen on airports due to logistic reasons. Bags can be stored on early bag store (EBS) systems and reintroduced in the sorting system when required to be delivered to baggage handlers. In this evaluation, bags are introduced in random order to simulate the EBS reintroduction of bags. For more information about EBS, refer to the following video (not created by authors): https://www.youtube.com/watch?v=d6_OeC0qZPE

that using FingerCI process and behaviour models allow SBIDS to successfully identify abnormal situations that may affect ICSs.

Based on the results presented in this evaluation, it is possible to conclude that FingerCI integrated with a SBIDS automates the work required from experts on writing specifications for the ICS, with accurate detection results and low false positive rate.

4 RELATED WORK

There are a few previous process discovery techniques, but they require the logs of all machines involved, something that is not practical in ICSs [13, 17–19, 23, 24, 26]. The three major difficulties would be accessing control layer devices that are normally on a restricted network, making the data logs in an uniform format to be analysed by those process discovery methods, and the absence of logs on the several low computing resource devices present on the ICS control network that could lead the absence of critical data required for building a process model. FingerCI solves these challenges by relying only on packet captures and converting them into a uniform data log. Although some techniques have been tested to perform process discovery on specific network protocols (e.g., TCP [25]), to the best of our knowledge, the solution proposed is the first to accomplish process discovery based on network traffic inspection that supports multiple network protocols simultaneously. Furthermore, the FingerCI behaviour model is novel since previous works that employed process discovery techniques for anomaly detection, only evaluated the causal relation between activities based on the ordering of their execution. The behaviour model complements the process diagram by providing the semantic meaning to the diagram that cannot be interpreted otherwise, namely it represents the environment preconditions for activities to occur, (i.e., the state of variables, system device state, etc.), represents the effect activities can have on the system, and the frequency cyclic activities occur on normal conditions.

5 CONCLUSION

This paper presented FingerCI, a solution that automatically fingerprints an ICS infrastructure and collects the necessary network, process and behaviour configuration required to setup a SBIDS. To the best of our knowledge, the solution proposed is the first capable of automating SBIDS setup by performing process mining solely based on ICS network traffic, looking beyond the causal relations between business activities and providing detection rules based on behaviour analysis. Moreover, the evaluation conducted shows FingerCI that it produces very precise and accurate specification models that can be used by SBIDS to detect abnormal activity.

ACKNOWLEDGMENTS

The research work presented in this paper was undertaken in the scope of project ENSURESEC, which has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 883242, and supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID).

REFERENCES

[1] Communication network dependencies for ICS/SCADA systems. *European Network and Information Security Agency (ENISA)*, 2016.
[2] Securing smart airport. *European Network and Information Security Agency (ENISA)*, 2016.
[3] C. M. Ahmed, J. Zhou, and A. P. Mathur. Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS. In *Proceedings of the 34th ACM Annual Computer Security Applications Conference*, pages 566–581, 2018.
[4] W. Aoudi, M. Iturbe, and M. Almgren. Truth will out: Departure-based process-level detection of stealthy attacks on control systems. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 817–831, 2018.
[5] C. Feng, V. R. Palleti, A. Mathur, and D. Chana. A systematic framework to generate invariants for anomaly detection in industrial control systems. In *Network and Distributed System Security Symposium (NDSS)*, 2019.
[6] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. A. Beyah. Who's in control of your control system? device fingerprinting for cyber-physical systems. In *Network and Distributed System Security Symposium (NDSS)*, 2016.
[7] N. Goldenberg and A. Wool. Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *International Journal of Critical Infrastructure Protection*, 6(2):63–75, 2013.
[8] V. Graveto, L. Rosa, T. Cruz, and P. Simões. A stealth monitoring mechanism for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 24:126–143, 2019.
[9] A. Hemmer, R. Badonnel, and I. Chrisment. A process mining approach for supporting iot predictive security. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2020.
[10] K. E. Hemsley, E. Fisher, et al. History of industrial control system cyber incidents. Technical report, Idaho National Lab (INL), 2018.
[11] A. Kleinmann and A. Wool. A statechart-based anomaly detection model for multi-threaded SCADA systems. In *International Conference on Critical Information Infrastructures Security*, pages 132–144. Springer, 2015.
[12] V. B. Krishna, K. Lee, G. A. Weaver, R. K. Iyer, and W. H. Sanders. F-DETA: A framework for detecting electricity theft attacks in smart grids. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 407–418, 2016.
[13] S. J. Leemans, D. Fahland, and W. M. van der Aalst. Discovering block-structured process models from event logs - a constructive approach. In *International conference on applications and theory of Petri nets and concurrency*, pages 311–329. Springer, 2013.
[14] J. Lima, F. Apolinário, N. Escravana, and C. Ribeiro. BP-IDS: Using business process specification to leverage intrusion detection in critical infrastructures. In *31st IEEE International Symposium on Software Reliability Engineering (ISSRE 2020)*, 2020.
[15] J. Lima, N. Escravana, and C. Ribeiro. BPIDS-using business model specification in intrusion detection. In *Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014*, volume 8688, page 479. Springer, 2014.
[16] H. Lin, A. Slagell, Z. T. Kalbarczyk, P. W. Sauer, and R. K. Iyer. Runtime semantic security analysis to detect and mitigate control-related attacks in power grids. *IEEE Transactions on Smart Grid*, 9(1):163–178, 2016.
[17] D. Myers. *Detecting cyber attacks on industrial control systems using process mining*. PhD thesis, Queensland University of Technology, 2019.
[18] D. Myers, K. Radke, S. Suriadi, and E. Foo. Process discovery for industrial control system cyber attack detection. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 61–75. Springer, 2017.
[19] D. Myers, S. Suriadi, K. Radke, and E. Foo. Anomaly detection for industrial control systems using process mining. *Computers & Security*, 78:103–125, 2018.
[20] F. Reuschling, N. Carstengerdes, T. H. Stelkens-Kobsch, K. Burke, T. Oudin, M. Schaper, I. P. Filipe Apolinário, and L. Perlepes. Toolkit to enhance cyber-physical security of critical infrastructures in air transport. *Cyber-Physical Threat Intelligence for Critical Infrastructures Security*, pages 254–287, 2021.
[21] R. Tan, H. H. Nguyen, E. Y. Foo, D. K. Yau, Z. Kalbarczyk, R. K. Iyer, and H. B. Gooi. Modeling and mitigating impact of false data injection attacks on automatic generation control. *IEEE Transactions on Information Forensics and Security*, 12(7):1609–1624, 2017.
[22] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105, 2016.
[23] W. van der Aalst. *Process mining: discovery, conformance and enhancement of business processes*, volume 2. Springer, Heidelberg, 2011.
[24] W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
[25] C. Wakup and J. Desel. Analyzing a TCP/IP-protocol with process mining techniques. In *International Conference on Business Process Management*, pages 353–364. Springer, 2014.
[26] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. A. de Medeiros. Process mining with the heuristics miner-algorithm. Technical report, Technische Universiteit Eindhoven, 2006.