Attacking Maritime Control Systems Through Process Mining[1]

(Discussion Paper)

Giacomo Longo[1], Francesco Lupia[2,*], Andrea Pugliese[2] and Enrico Russo[1]

[1]University of Genoa, Italy
[2]University of Calabria, Italy

Abstract

The evolution of technology in the maritime sector has markedly enhanced both operational efficiency and safety by incorporating Industrial Control Systems. Nonetheless, such advancements have concurrently unveiled cybersecurity vulnerabilities, introducing novel risks to maritime operations. In this paper, we explore the standard architecture of integrated maritime systems, with a specific emphasis on the Steering Gear subsystem, and present an adversary model tailored for the technological and operational context of the maritime sector. We introduce a methodology embedded in a custom malware that employs a process mining method, aimed at conducting physics-aware, targeted attacks. The results show that such a malware can delivery attacks that have disruptive consequences on the ships.

Keywords

Targeted Attacks, Process Mining, Maritime Industry, Industrial Control Systems, Physics-Awareness

1. Introduction

The technological advancement in the maritime sector has significantly improved operational efficiency and safety through the integration of Industrial Control Systems (ICS). However, this progress also brings to light cybersecurity vulnerabilities, introducing new risks to maritime operations. The increase in cyber-attacks targeting maritime systems emphasizes the urgent need for robust cybersecurity frameworks. This paper explores the development of process mining based attacks against maritime ICS, highlighting the vulnerabilities in this critical infrastructure.

Cybersecurity incidents in the maritime industry have historically leveraged generic vulnerabilities, but the most severe impacts have resulted from attacks tailored to exploit the unique operational contexts of maritime systems. As for notable cybersecurity breaches such as Stuxnet and Triton, our research suggests that maritime ICS are vulnerable to similarly sophisticated attacks. These incidents underline the potential for catastrophic outcomes from attacks that

[1]Extended abstract of [1]

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy

*Corresponding author.

giacomo.longo@dibris.unige.it (G. Longo); francesco.lupia@unical.it (F. Lupia); andrea.pugliese@unical.it (A. Pugliese); enrico.russo@unige.it (E. Russo)

0000-0003-0025-7191 (G. Longo); 0000-0003-0775-6890 (F. Lupia); 0000-0003-4385-958X (A. Pugliese); 0000-0002-1077-2771 (E. Russo)

exploit the specificities of ICS, underscoring the necessity of understanding these vulnerabilities in the maritime domain.

In this paper, we focus on the typical onboard ICS configurations from an adversarial standpoint. In more detail:

- We explore the standard architecture of integrated maritime systems, with a particular focus on the Steering Gear subsystem, and present an adversary model tailored for the technological and operational context of the maritime sector.
- We propose a methodology based on process mining, embedded in a novel malware designed to execute physics-aware targeted attacks through reconnaissance, weaponization, and delivery phases. We show how such a malware can deliver attacks that have disruptive consequences on the ships.

2. Integrated Maritime Systems and Adversary Model

2.1. Common Architecture

Modern vessels are complex Information Technology/Operational Technology systems that seamlessly integrate to ensure efficient coordination and control of diverse ship functions. The Integrated Bridge System (IBS) [2] and the Integrated Platform Management System (IPMS) [3] are the key elements in this integration. IBS oversees navigation from the bridge, while IPMS is responsible for managing and monitoring the ship's automation. Their functions operate within a common architecture in modern vessels, illustrated in Figure 1.

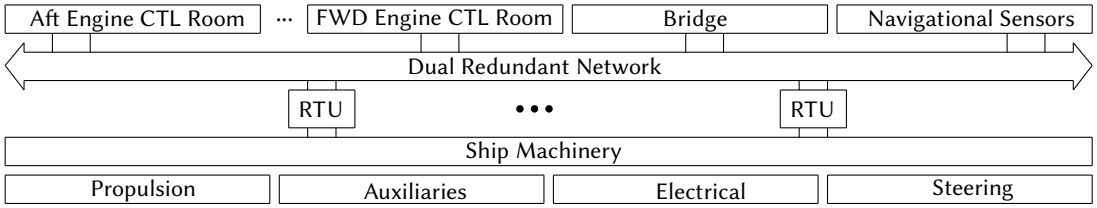

Figure 1: Common architecture for integrated maritime systems.

The architecture includes multifunction control consoles, navigational sensors, and Remote Terminal Units (RTUs). On the Bridge, consoles provide operators access to essential navigation functions within the IBS, including RADAR and the Electronic Chart Display and Information System (ECDIS). Additionally, they act as the Human Machine Interface for the IPMS in control rooms like the Engine Control Room. Navigational sensors collect and transmit data regarding the ship's position, orientation, velocity, and surrounding environmental conditions. RTUs handle process-level data acquisition and control, interfacing with ship machinery actuators and sensors. They interact with propulsion systems, auxiliary pumps and compressors, electrical generators and switchboards, and steering mechanisms.

In essence, RTUs assume the role of specialized Programmable Logic Controllers (PLC) [4], which function as both an intermediary for communication and a solution for hiding the complexity of multiple other PLCs that interface with ship machinery. As generic PLCs, a CPU

executes user-defined logic, manages the operational cycle, and communicates with peripherals. This cycle involves scanning inputs, processing data, and updating outputs, known as the *scan cycle*.

The architecture follows an integration pattern that enables connected endpoints to use a dual redundant network for message transmission. Navigational sensors provide real-time data, while RTUs exchange setpoints and commands with ship machinery, and consoles process and display data. This setup enhances operators' situational awareness and decision-making by facilitating comprehensive data fusion. Additionally, it simplifies redundancy in control rooms and control station duplication. For example, an engineering station can be placed on the bridge for officers to monitor machinery subsystems.

Navigational sensors and maritime equipment adhere to the NMEA 0183 standard that specifies an electrical and data exchange format between maritime electronics. Devices utilize the IGMP protocol to interact with multicast flows, establishing the Lightweight Ethernet (LWE) configuration.

RTUs follow protocols commonly employed in Operational Technology setups. While no single standard prevails, onboard installations commonly prefer two widely adopted protocols: Modbus and OPC [5]. In this paper, our focus is on Modbus, as it remains the protocol of choice for most vendors [6]. Briefly, the Modbus protocol organizes PLC program memory into four main types of registers: discrete output coils (1-bit), discrete input contacts (1-bit), analog input registers or `IW` (16-bit), and analog output holding registers or `QW` (16-bit). Commands, known as function codes, manipulate these registers for read/write access. Each Modbus transaction comprises a single query, response, or broadcast frame containing the receiver's address, command to execute, and associated data.

Regarding cybersecurity, these protocols lack inherent security features like encryption or authentication, leaving transmitted data vulnerable to eavesdropping and false information injection. Attackers could exploit integrated configurations for lateral movements between systems. To address these risks, recent ship design or refit efforts align with the International Maritime Organization recommendations [7] by opting for segregated systems grouped by function, such as navigation and automation. However, some control consoles need relaxed restrictions for operational continuity. For example, the steering subsystem control station relies on real-time monitoring of the ship's behavior [8], and correlating data from navigational sensors and ship machinery requires it to have both navigation and automation functions.

2.2. Steering Gear Subsystem

In the maritime domain, the *steering gear system* comprises equipment essential for vessel navigation, including the rudder, mechanical linkages, actuators, and associated Steering Gear Control System (SGCS). The International Convention for the Safety of Life at Sea (SOLAS) and its amendments [9, 10, 11, 12] define mandatory requirements for steering gear systems on international voyaging ships, such as the need for a rudder angle indicator on the navigation bridge. Consequently, implementations often rely on established solutions. This paper focuses on electro-hydraulic SGCS, prevalent in large commercial vessels, with dual pumps, or *power units*, for redundancy. The ship's rudder is operated by hydraulic cylinders which counteract its lift. Valves near those cylinders allow the selection of the actuation direction, while relief

valves ensure system safety from overpressure. The entire system contains an ISO VG 100 oil as its working fluid. In total, our implementation of the SGCS exposes 312 registers. They hold actuator setpoints and sensor measurements via a dedicated RTU. A subset of these registers is of particular interest to the attackers. *IW101* contains the current rudder angle as measured by its sensor, while *QW100* stores the desired rudder angle set by the bridge. *QW104* and *QW105* contain the hydraulic cylinder valve command, which dictates whether the valve is open, closed, or in reverse flow. Lastly, *QW106* and *QW107* store the desired speed of the pump, also known as the governor setting.

2.3. Adversary Model and Assumptions

The focus of this study is on sophisticated threat actors [13] who are active in the maritime sector. These attackers possess the necessary skills and resources to develop custom malware and utilize various tactics for its deployment. Their techniques include exploiting maintenance operations, supply chain compromise, social engineering, and vulnerabilities in onboard workstations.

We assume the targeted vessel follows the NMEA 0183 standard using an LWE configuration and is equipped with a SOLAS-compliant single rudder electro-hydraulic steering gear. We presume that the attackers have successfully introduced malware into an onboard system capable of receiving NMEA data and interacting with the RTU of the steering subsystem via Modbus.

The objective is to manipulate the ship's course by gaining control over the steering system, thereby causing disruption or interfering with navigation. This action could result in substantial economic or reputational losses while jeopardizing safety.

The malware must operate as a standalone entity, capable of carrying out malicious activities without contacting a command and control infrastructure. It should maintain a covert operational approach by observing system behavior before making changes, rather than using trial-and-error methods. Additionally, it must be able to run on existing onboard systems, including legacy ones, while minimizing resource usage to avoid detection. Upon activation, the malware should quickly bring the ship to the desired state within a limited time frame, thereby reducing the crew's capacity to respond effectively.

3. Methodology

3.1. Attack Phases and Tasks of the Stealth Malware

Figure 2 sketches the different phases of the attack, along with the tasks the stealth malware carries out to conduct it. In the reconnaissance phase, the malware starts by identifying key

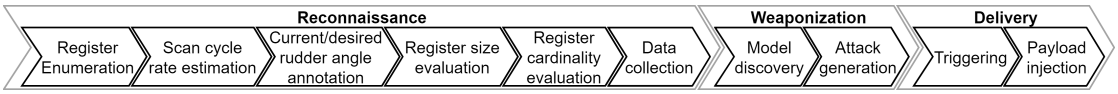

Figure 2: Phases and tasks of the stealth malware.

features of the onboard automation system. The *register enumeraton* task involves identifying the

list of registers utilized by the SGCS automation system among available ones, typically through enumeration techniques like scanning the Modbus address range. Then, attackers employ *scan cycle rate estimation* to estimate the fixed scan cycle rate (f_p) at which the PLCs operate. It involves sampling registers at a higher rate than f_p and monitoring the intervals between updates. The *current/desired rudder angle annotation* task identifies the registers for the current and desired rudder angles. Identification is possible by listening to the broadcasted NMEA and correlating the RSA (Rudder Sensor Angle) and ROR (Rudder Order Status) sentences with values returned by reading registries. In the register size and cardinality evaluation, attackers aim to understand the size, i.e., the range and representation format (e.g., integer, float32 or float64), and the cardinality of registers, i.e., the diversity and the type of data they can hold, within the SGCS system. Size evaluation involves referencing adjacent registers and determining if they can be decoded as a floating-point number. Cardinality evaluation includes querying values and applying analysis and statistical methods over multiple cycles. The reconnaissance phase concludes with the *data collection* task, collecting data from the SGCS system automation to facilitate the model discovery task. Each entry represents a register reading that includes a timestamp, the register ID, and the current value.

In the *weaponization* phase, the malware utilizes a technique to reverse-engineer the operational procedure for controlling the rudder as utilized by the automation system. The involved tasks are detailed in Section 3.2.

Finally, in the *delivery* phase, the malware determines the optimal timing to initiate and execute the attack. The *triggering* task operates based on the ship's state inferred from NMEA traffic. The attack starts when the *payload injection* task transmits the output obtained from the weaponization phase to the SGCS RTU using the Modbus protocol.

3.2. Model Discovery and Attack Generation

Process mining is recognized as a powerful and effective approach for analyzing and understanding process models via event log analysis. The core of process mining is process discovery, a technique aimed at creating process models from event logs. These models, often represented as graphs of causal dependencies, describe the control flow among activities. Notably, in addition to the information gathered from the log, there are methods that can take advantage of background knowledge available to the domain expert in the form of precedence constraints over the topology of the dependency graphs [14] ultimately discovering meaningful models. In the context of ICS, process discovery algorithms have been successfully applied to generate process models that accurately reflect the expected behavior of the system, learning from the device logs generated by ICS devices like PLCs.

Expanding on these concepts, we now recall our proposed method (originally presented in [1]). The method is designed to support malicious reverse engineering focused on ship control systems. This requires the introduction of several foundational definitions and notations.

We start by introducing the concept of device status record which represents an individual entry that is generated by a SCADA device. Let T, VN, and RA denote sets of timestamps, variable names, and attribute names, respectively. In addition, let VV_v denote the set of possible values of the variable named $v \in VN$. A *device status record* r is a tuple of attribute name/value pairs. The value of attribute $a \in RA$ for device status record r is denoted by $a(r)$. Every device

status record r has at least the following attributes: (i) $time(r) \in T$ is the timestamp of the record; (ii) $vName(r) \in VN$ is the variable name in the record; (iii) $vVal(r) \in VV_{vName}$ is the value in the record. We denote the set of all possible device status records as R.

A *device log* $L \subseteq R$ is a set of device status records. A device log represents the sequential recording of system states and activities as derived from the readings of the various registers. Table 1 illustrates a portion of a device log, including the values recorded for different variables during two consecutive scan cycle.

Table 1
Excerpt of a raw device log.

time	vName	vVal
00:00:00	QW100	32768
00:00:00	IW101	32768
00:00:00	QW104	32768
00:00:00	QW105	32768
00:00:00	QW106	32768
00:00:00	QW107	32768

To facilitate the construction of a model for SGCS and the subsequent generation of attacks, our methodology proceeds through several stages, starting with preprocessing to convert raw data into a format suitable to process mining. Given a device status record with a variable name $v \in VN$ and its associated value $val \in VV_v$, the *activity name* is obtained as follows: (i) if v is a numerical variable (i.e., $|VV_v| > 3$), the activity name indicates the change relative to the last recorded value for v. Specifically, we denote an increase in value by "v_Increasing" and a decrease by "v_Decreasing"; (ii) if v is a boolean or ternary variable ($|VV_v| \leq 3$), the activity name captures the specific transition from the previous recorded value. Specifically, we use "v_(previousvalue)_to_(currentvalue)" as activity name. If a previous value is not in the log, the activity name for v is not specified. The set of all possible activity names is denoted as A. Observe that $|VV_v|$ corresponds to the cardinality of the registers (see Section 3.1).

Finally, given a device log L, a *case identifier* is a function $cid : L \to \mathbb{N}$. Consider the set $\{r_1, r_2, \ldots, r_k\} \subseteq L$ of device status records such that $\forall i \in [1, k],\ vName(r_i) = v_{sc}$. Then:

- $\forall r \in L$ s.t. $time(r) \leq time(r_1)$, $cid(r) = 0$.
- $\forall i \in [1, k], \forall r \in L$ s.t. $time(r_i) < time(r) \leq time(r_{i+1})$, $cid(r) = i$.

Utilizing domain expertise, we partition the log into subsets reflecting significant operational states—specifically, those ending with v_{sc}*_decreasing* or v_{sc}*_increasing*. This partitioning distils the log into more manageable portions that represent the SGCS's dynamics, facilitating the extraction of meaningful process models without the clutter of irrelevant data.

Employing the Heuristics Miner algorithm [15] (due to its computational efficiency, which is crucial in a setting with limited resources, and capacity to generate clear, interpretable models) we extract a process model from the event log. This model mirrors the SGCS's operational dynamics, forming the basis for the development of automated attacks through specific operational sequences of Modbus packets.

For instance, starting from the initial raw data of Table 1, by applying the definitions and procedure described above, we generate event sub-logs that capture the SGCS's behavior under standard operational conditions. One such sub-log, focusing on the decreasing behavior of the register *IW101*, is shown in Table 2.

Table 2
Excerpt of the final event sub-log.

time	activity	caseid
00:00:08	QW104_32768.0_to_65536.0	9
00:00:08	QW106_Increasing	9
00:00:09	IW101_Decreasing	9
00:00:53	QW106_Increasing	54
00:00:53	QW107_Increasing	54
00:00:54	IW101_Decreasing	54

The process model derived from it is depicted in Figure 3—it provides a visual and functional blueprint for constructing targeted attacks.

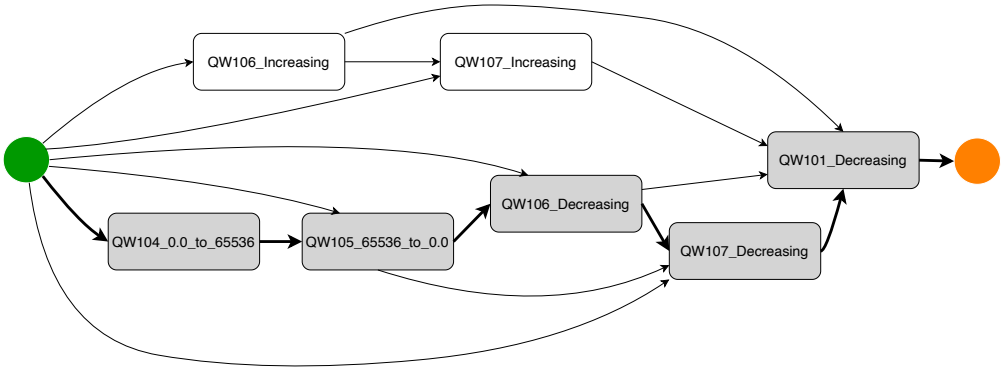

Figure 3: A process model for the final event sub-log of Table 2.

Table 3 reports an example targeted attack, showcasing the sequential execution leading to SGCS disruption. The attack is based on the model of Figure 3 and mirrors the operations observed along the most frequent path in the model—from the starting to the terminating activity (bold arrows in Figure 3)—to orchestrate specific Modbus packet injections aimed at controlling valve actuators and pump governors, consequently disrupting SGCS's normal operation. The attack leverages NMEA traffic analysis to identify optimal attack timings based on ship telemetry, proximity to other vessels, and environmental conditions. The malware processes the incoming attack file containing the payloads outlined in Table 3, then proceeds to inject these payloads to the SGCS RTU via Modbus, according to the operational flow depicted in the process model.

The example above shows how the application of process mining in cybersecurity enables the development of advanced malware which can precisely target and manipulate SGCS operations

Table 3
Example attack.

#	Event	Payload	Description
1	QW104_0.0_to_65536	07d4ffff	Inject *L Valve command register* with "write single discrete output coil" packet to set 65536 value
2	QW105_65536_to_0.0	07d5000	Inject *R Valve command register* with "write single discrete output coil" packet to set 0.0 value
3	h_1	07d60001	Send a "read discrete output coil" packet to *L Pump Governor*
4	h_2	07d70001	Send a "read discrete output coil" packet to *R Pump Governor*
5	QW106_Decreasing	07d65d7a	Send a "write single discrete output coil" packet to *L Pump Governor*
6	QW107_Decreasing	07d77de8	Send a "write single discrete output coil" packet to *R Pump Governor*

by sending Modbus packets to critical registers. The interested reader is referred to [1] for further empirical analyses.

4. Discussion

In this work, we shed light on the potential for attackers to exploit specialized malware designed to disrupt the physical operations of ships. The urgent need for research into countermeasures is evident, with potential solutions including early detection mechanisms [16] like physics-aware ICS honeypots [17, 18, 19, 20], and process mining for identifying physics-aware attacks based on game theory frameworks [21, 22, 23, 24]. Additionally, compact activity log representations suitable for both procedural and declarative process mining methods are also essential. The discovered models can be refined with explainable AI to create accurate, understandable models for defense [25, 26, 27]. Finally, the challenges related to anonymity in maritime communication networks also present a critical area for future exploration [28]. Indeed, the integration of privacy-preserving techniques, could further strengthen maritime systems against operational data interception and manipulation [29, 30].

Acknowledgements

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union — NextGenerationEU.

References

[1] G. Longo, F. Lupia, A. Pugliese, E. Russo, Physics-aware targeted attacks against maritime industrial control systems, Journal of Information Security and Applications 82 (2024) 103724. doi:10.1016/j.jisa.2024.103724.

[2] L. P. Perera, C. G. Soares, Collision risk detection and quantification in ship navigation with integrated bridge systems, Ocean Engineering 109 (2015) 344–354.

[3] R. Warżała, Modern Integrated Platform Management System Laboratory for Polish Naval Academy: Design and Implementation, Scientific Journal of Polish Naval Academy 220-221 (2020) 59–71.

[4] M. Tiegelkamp, K.-H. John, IEC 61131-3: Programming industrial automation systems, volume 166, Springer, 2010.

[5] X. Luo, Research on Communication Technology of Ship Integrated Monitoring System Based on OPC, in: International Conference on Intelligent Transportation, Big Data & Smart City, ICITBS, 2020.

[6] G. W. Adhane, D. Kim, Distributed control system for ship engines using dual fieldbus, Comput. Stand. Interfaces 50 (2017) 83–91.

[7] International Maritime Organization, Guidelines on Maritime Cyber Risk Management, 2022.

[8] A. Ariffin, J. Laurens, S. Mansor, Real-time evaluation of second generation intact stability criteria, Proceedings of the RINA, Royal Institution of Naval Architects— Smart Ship Technology (2016).

[9] International Maritime Organization, Recommendation Concerning Regulations for Machinery and Electrical Installations in Passenger and Cardo Ships, 1975. Resolution A.325(IX).

[10] International Maritime Organization, Improved Steering Gear Standards for Passenger and Cargo Ships, 1979. Resolution A.415(XI).

[11] United Nations, International Convention for the Safety of Life at Sea, 1974.

[12] United Nations, Protocol relating to the International Convention for the Safety of Life at Sea, 1974, 1978.

[13] G. Longo, E. Russo, A. Armando, A. Merlo, Attacking (and Defending) the Maritime Radar System, IEEE Transactions on Information Forensics and Security 18 (2023) 3575–3589. doi:10.1109/tifs.2023.3282132.

[14] G. Greco, A. Guzzo, F. Lupia, L. Pontieri, Process Discovery under Precedence Constraints, ACM Trans. Knowl. Discov. Data 9 (2015) 32:1–32:39.

[15] A. J. M. M. Weijters, J. T. S. Ribeiro, Flexible Heuristics Miner (FHM), in: IEEE Symposium on Computational Intelligence and Data Mining, CIDM, 2011.

[16] C. Greco, M. Ianni, A. Guzzo, G. Fortino, Neural network based temporal point processes for attack detection in industrial control systems, in: 2022 IEEE International Conference on Cyber Security and Resilience (CSR), 2022, pp. 221–226. doi:10.1109/CSR54599.2022.9850333.

[17] M. Lucchese, F. Lupia, M. Merro, F. Paci, N. Zannone, A. Furfaro, HoneyICS: A High-interaction Physics-aware Honeynet for Industrial Control Systems, in: Proceedings of the 18th International Conference on Availability, Reliability and Security, ARES '23,

Association for Computing Machinery, New York, NY, USA, 2023. URL: https://doi.org/10.1145/3600160.3604984. doi:10.1145/3600160.3604984.

[18] F. Lupia, M. Lucchese, M. Merro, N. Zannone, ICS Honeypot Interactions: A Latitudinal Study, in: 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 3025–3034. doi:10.1109/BigData59044.2023.10386497.

[19] E. Russo, G. Costa, G. Longo, A. Armando, A. Merlo, Lidite: A full-fledged and feather-weight digital twin framework, IEEE Transactions on Dependable and Secure Computing 20 (2023) 4899–4912. doi:10.1109/TDSC.2023.3236798.

[20] G. Longo, A. Orlich, S. Musante, A. Merlo, E. Russo, Macyste: A virtual testbed for maritime cybersecurity, SoftwareX 23 (2023) 101426. doi:https://doi.org/10.1016/j.softx.2023.101426.

[21] G. Greco, F. Lupia, F. Scarcello, Coalitional games induced by matching problems: Complexity and islands of tractability for the Shapley value, Artif. Intell. 278 (2020).

[22] G. Greco, F. Lupia, F. Scarcello, The tractability of the shapley value over bounded treewidth matching games, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ijcai.org, 2017, pp. 1046–1052. URL: https://doi.org/10.24963/ijcai.2017/145. doi:10.24963/IJCAI.2017/145.

[23] G. Greco, F. Lupia, F. Scarcello, Structural tractability of shapley and banzhaf values in allocation games, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press, 2015, pp. 547–553.

[24] S. Saraeian, B. Shirazi, Process mining-based anomaly detection of additive manufacturing process activities using a game theory modeling approach, Computers & Industrial Engineering 146 (2020) 106584.

[25] M. Ianni, E. Masciari, Scout: Security by computing outliers on activity logs, Computers & Security 132 (2023) 103355. URL: https://www.sciencedirect.com/science/article/pii/S0167404823002651. doi:https://doi.org/10.1016/j.cose.2023.103355.

[26] M. L. Bernardi, M. Cimitile, F. M. Maggi, Data-aware process discovery for malware detection: an empirical study, Mach. Learn. 112 (2023) 1171–1199.

[27] F. Lupia, A. Mendicelli, A. Ribichini, F. Scarcello, M. Schaerf, Computing the Shapley value in allocation problems: approximations and bounds, with an application to the Italian VQR research assessment program, J. Exp. Theor. Artif. Intell. 30 (2018) 505–524.

[28] F. Buccafurri, V. De Angelis, M. F. Idone, C. Labrini, A protocol for anonymous short communications in social networks and its application to proximity-based services, Online Social Networks and Media 31 (2022) 100221. URL: https://www.sciencedirect.com/science/article/pii/S2468696422000258. doi:https://doi.org/10.1016/j.osnem.2022.100221.

[29] F. Buccafurri, V. De Angelis, M. F. Idone, C. Labrini, S. Lazzaro, Achieving sender anonymity in tor against the global passive adversary, Applied Sciences 12 (2022). URL: https://www.mdpi.com/2076-3417/12/1/137. doi:10.3390/app12010137.

[30] F. Buccafurri, V. de Angelis, S. Lazzaro, Mqtt-a: A broker-bridging p2p architecture to achieve anonymity in mqtt, IEEE Internet of Things Journal 10 (2023) 15443–15463. doi:10.1109/JIOT.2023.3264019.