Wireless Networks
https://doi.org/10.1007/s11276-022-03212-5

Low delay network attributes randomization to proactively mitigate reconnaissance attacks in industrial control systems

Xabier Etxezarreta[1] · Iñaki Garitano[1] · Mikel Iturbe[1] · Urko Zurutuza[1]

Accepted: 13 December 2022

Abstract
Industrial Control Systems are used in a wide variety of industrial facilities, including critical infrastructures, becoming the main target of multiple security attacks. A malicious and successful attack against these infrastructures could cause serious economic and environmental consequences, including the loss of human lives. Static networks configurations and topologies, which characterize Industrial Control Systems, represent an advantage for attackers, allowing them to scan for vulnerable devices or services before carrying out the attack. Identifying active devices and services is often the first step for many attacks. This paper presents a proactive network reconnaissance defense mechanism based on the temporal randomization of network IP addresses, MAC addresses and port numbers. The obtained information distortion minimizes the knowledge acquired by the attackers, hindering any attack that relies on network addressing. The temporal randomization of network attributes is performed in an adaptive way, minimizing the overhead introduced in the network and avoiding any error and latency in communications. The implementation as well as the tests have been carried out in a laboratory with real industrial equipment, demonstrating the effectiveness of the presented solution.

Keywords Industrial control systems · Moving target defense · Software defined networking · Industrial network security · Proactive intrusion response

1 Introduction

Industrial Control System (ICS) is a general term that covers several specialized elements used for monitoring and controlling industrial processes [1]. They are composed of diverse elements such as sensors, actuators, Programmable Logic Controllers (PLC) or Supervisory Control and Data Acquisition Systems (SCADA). They can be found in all types of industries, including Critical Infrastructures (CIs), becoming essential elements for the welfare and economic development of society. Examples of CI include nuclear power plants, transportation systems, power grids, hydroelectric dams and critical manufacturing plants.

Traditionally, ICSs have been deployed in isolated environments, using proprietary hardware and communication protocols. Isolation and obscurity have been the pillars in ensuring the security of industrial operations, but the integration of newer technology such as Information Technology (IT) systems and increasingly interconnected ICSs, has exposed the originally isolated ICSs to corporate networks, including the Internet. This change resulted in traditional isolation and security through obscurity techniques no longer being effective in ICS environments. Less isolation implies a greater need to secure these systems against attacks. The nature of ICS makes it difficult for IT security solutions to meet the requirements of these systems. According to NIST SP 800-82 Rev 2 [1], ICS differ from IT systems in the following aspects:

- ICSs are intended to control and monitor physical devices and processes.

✉ Xabier Etxezarreta
xetxezarreta@mondragon.edu

Iñaki Garitano
igaritano@mondragon.edu

Mikel Iturbe
miturbe@mondragon.edu

Urko Zurutuza
uzurutuza@mondragon.edu

[1] Electronics and Computing Department, Mondragon Unibertsitatea, Goiru 2, 20500 Arrasate-Mondragón, Spain

Published online: 06 January 2023

- Availability is a priority in ICSs. An unexpected system outage is not acceptable.
- ICSs are time-critical and must be within an acceptable delay.
- The replacement or upgrade period of ICS devices is very long compared to IT systems.
- The application of security patches in ICS is often postponed due to availability and reliability requirements.
- ICS devices are designed for industrial processes control and in many cases do not have the capability to integrate security mechanisms.

Compared to IT networks, industrial network topologies are generally static, and the control traffic is by nature repetitive and predictable, since most of the traffic is generated by automated processes [2]. These static characteristics of industrial networks is an advantageous scenario for the attacker, allowing vulnerabilities to be explored before the attack is launched. Due to this problem, a trend of proactive security solutions were developed in response to these static systems under the name of Moving Target Defense (MTD). MTD can be defined as a constantly changing system that shifts or reduces the attack surface, making it difficult for an attacker to easily explore and perform attacks.

As presented in [3], MTD techniques have been studied under various classifications criteria in the literature. On the one side, timeliness-based classification separates MTD approaches based on when MTD actions are triggered. Within this classification criteria, there are three groups: time-based MTD (MTD is applied in certain time intervals), event-based MTD (MTD is applied when an event occurs) and hybrid MTD (combination of time and event-based MTD). On the other side, MTD approaches can also be classified according to the strategy they follow, also known as operation-based MTD. Within this classification criteria, strategies such as shuffling-based MTD (randomization of system configuration), diversity-based MTD (same functionality but different deployments), redundancy-based MTD (replication of resources) and hybrid MTD (combination of shuffling, diversity and redundancy) can be found.

Software-Defined Networking (SDN) has become a promising technology for ICS security, both for developing MTD techniques [4] and for developing intrusion detection and response techniques in general [5]. RFC 7140 [6] defines SDN as a set of techniques used to facilitate the design, delivery, and operation of network services in a deterministic, dynamic, and scalable manner. For this, the control plane is separated and centralized, while the data plane remains in the network devices, focusing its functionality on packet forwarding. Figure 1 depicts the three

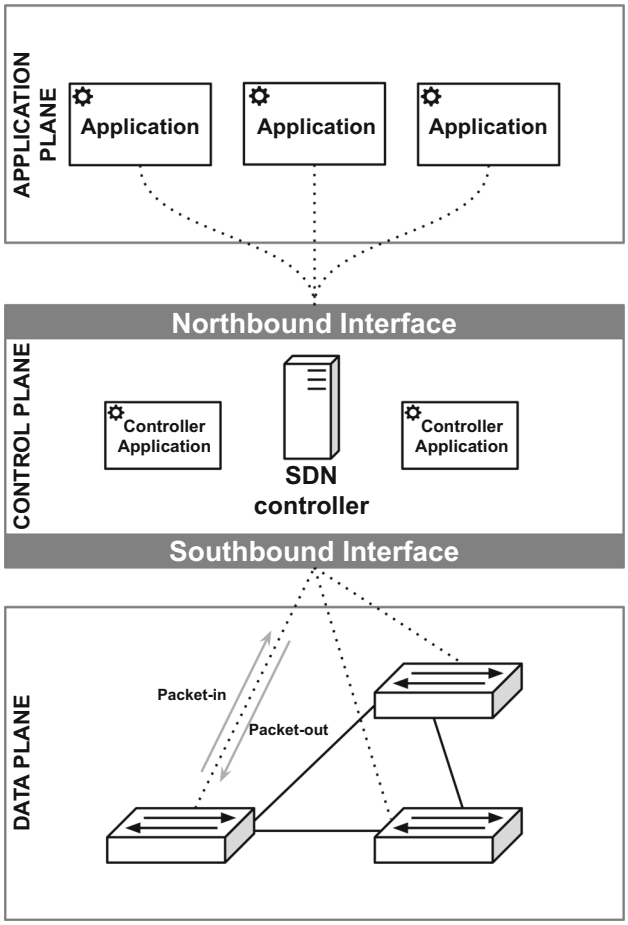


Fig. 1 SDN architecture

main planes of the SDN architecture. Firstly, the data plane is where forwarding decisions are implemented and traffic is processed. Secondly, the control plane is used to provide logic to the data plane through the use of SDN controllers. Communication between the data and control plane is done through the southbound interface with specific protocols in which OpenFlow is widely used, although there are some others (e.g. NETCONF). Finally, the application plane is where user-developed applications interact with the network through the northbound interface.

Traditional networks are not optimized to meet the current and future needs of ICS, such as network manageability due to the growing number of networked devices or the collaboration between different ICS ecosystems, which demand more flexibility and heterogeneity without compromising quality of service [7]. From an intrusion detection and response techniques development point of view, SDNs offer advantages over traditional networks in the following aspects: (1) provides network-wide visibility, (2) increases network programmability by integrating user-developed applications and (3) allows a dynamic and centralized network flows management.

In this paper, the concepts of SDN and MTD are combined to develop a proactive defense mechanism to mitigate reconnaissance attacks in industrial control networks. The main contributions of this work can be summarized in the following points:

1. An MTD mechanism that randomizes the IP and MAC addresses and port numbers of network packets in real time, distorting the information obtained by an attacker during the reconnaissance phase and preventing direct access to devices by using real information. The randomization is performed at the data plane and is completely transparent to the end devices in the network.
2. A methodology to initialize flow rules that randomize the network attributes and forward the traffic to its destination by using a user defined allowlist. This allowlist defines which devices are authorized to communicate with each other. Based on this information, the flow rules are installed on the switches.
3. An adaptive network attributes randomization strategy in order to minimize the latency and comply with ICS real-time requirements. This is achieved by using backup flow rules and the *priority* field of the OpenFlow protocol.

2 Related work

This section presents a brief discussion of related work. On the one hand, information about the different MTD techniques in the literature is provided. On the other hand, the applicability of MTD in ICS is discussed.

2.1 Moving target defense techniques

MTD techniques aim to change the static nature of the networks by modifying the attack surface dynamically. These techniques can be classified into the following four operational groups [3]:

Shuffling-based MTD. These techniques dynamically randomize the network configuration confusing the attacker in the reconnaissance stage, reducing the attack surface and making it less predictable. In this group we can find techniques that randomize IP addresses [8, 9], ports [10], routes through which the network traffic flows [11] or the header of network packets [12, 13]. There are also proposals that combine several shuffling-based MTD techniques [14, 15]. Authors in [16] combine IP randomization, packet payload randomization and TCP port scan response adaptation to mitigate reconnaissance attacks.

Diversity-based MTD. Consists of providing equivalent services but with different implementations. Code diversity aims to divide a program into components that can be implemented in different execution environments [17]. Software diversity techniques deploy equivalent variants of web servers, applications or virtual servers to improve network resilience [18]. Finally, diversity techniques in programming languages allow mitigating code or SQL injection attacks [19].

Redundancy-based MTD. Consists of deploying replicas that offer the same functionality. There are techniques that provide redundancy in network sessions [20] or that deploy replicas of servers with the same functionality [21].

Hybrid MTD. These techniques combine MTD based on Shuffling, Diversity and Redundancy [22, 23].

2.2 MTD in industrial control systems

MTD techniques have been introduced in ICS to reverse the static and predictable nature of these systems. Research has focused on developing and adapting MTD techniques based on shuffling, especially techniques that randomize IP addresses and flow routes.

The IP address randomization techniques for ICS in the literature leverage IP packet filter rules by using the Netfilter [24] framework provided by the Linux kernel to perform IP address translation operations. Several works [25, 26] show that as the randomization interval decreases, the Round Trip Time (RTT) increases considerably, becoming a problem for systems that require low latency communications.

Germano da Silva et al. [27] propose a flow routes randomization technique to prevent all traffic from going down the same path by dispersing traffic over multiple routes to defend against unauthorized traffic interceptions. Since the transition to alternative routes is not performed adaptively, authors in [28] propose to use the *hard-timeout* field of the OpenFlow protocol to define several flow rules at once and minimize the latency introduced in the SDN-based industrial network.

Chavez et al. [14] propose an MTD approach combining multiple shuffling MTD strategies in a single solution. On the one hand, SDN is used to install flow rules to randomize IP addresses and forwarding paths with a configurable randomization period. On the other hand, port randomization is implemented in each host by using Netfilter kernel module. Results show that port randomization is the technique with the least amount of performance impact, but scalability could be an issue as requires manual configuration in each end-device. Similarly, Chavez [29] extended the IP and port randomization for the protection of power systems by increasing attackers uncertainty.

Unlike existing publications, this work combines SDN technology with IP, MAC and port randomization to develop a proactive defense mechanism that can be implemented in time-sensitive environments such as ICSs. The randomization process is implemented in the forwarding devices being a transparent process for end-devices, and the transition to new random IPs, MACs and ports is performed adaptively by using backup flow rules and the *priority* field of the OpenFlow protocol. An extensive evaluation is conducted in a testbed with real ICS equipment, demonstrating that the proposed solution is able to mitigate reconnaissance attacks with minimal performance impact.

3 Framework overview

This section presents the proactive defense mechanism against reconnaissance that provides IP, MAC and port randomization for ICS. Firstly, the architecture and the parts of which it is composed are detailed. Secondly, how the IP, MAC and port ranges are allocated is defined. Thirdly, the randomization procedure is detailed. Finally, we define how the transition to new random network attributes is implemented adaptively.

3.1 Architecture

The architecture is designed to be integrated and used in an SDN-based industrial environment. Figure 2 depicts the overview of the network attributes randomization architecture. The main components of this architecture are the following:

- *End devices/hosts:* it is composed of a wide variety of devices used in industrial environments such as SCADA servers, PLCs or workstations.
- *OpenFlow switches:* the function of these forwarding devices is to route traffic to its destination, based on predefined flow rules. In this case, these devices process network packets and randomize IP, MAC and Ports.
- *SDN controller:* it is responsible for the communication between the data plane and the application plane. The controller gets the requests from the MTD module and transmits them to the switches using the OpenFlow protocol.
- *MTD module:* it is an application deployed in the application plane. Its function is to initialize and update the flow rules in the OpenFlow switches that perform the translations between real and random network attributes. This is done using the SDN controller's northbound interface, in this case a REST API.

3.2 Network attributes range allocation

The generation and allocation of random IP and MAC addresses, and TCP/UDP port numbers to hosts in the industrial network must comply with the following constraints in order to avoid collisions during the allocation and maximize the unpredictability of the industrial network.

- *Constraint 1:* the same random IP and MAC address must not be assigned to more than one host within an MTD interval.
- *Constraint 2:* the same random port number must not be assigned simultaneously to more than one TCP or UDP service in a host within an MTD interval.
- *Constraint 3:* the same random IP, MAC or port must not be assigned to a host more than once in N consecutive MTD intervals in order to maximize uncertainty. This is achieved by keeping a track of allocated random attributes in the last N intervals.

Random IP address generation. Random IP addresses are generated taking into account the available address space of the industrial network. At each interval change, a historical record is kept of the previously assigned IP addresses to the devices. This prevents the same IP address from being repeatedly assigned to the same device over several consecutive intervals.

Random MAC address generation. For each random IP address a random MAC address is generated. RFC 7042 [30] has been taken into account for the generation of MAC addresses to avoid generating addresses specifically reserved for different purposes such as IPv4/IPv6 multicast, Point to Point Protocol (PPP) or IPv4/IPv6 Virtual Router Redundancy Protocol (VRRP). In order to generate

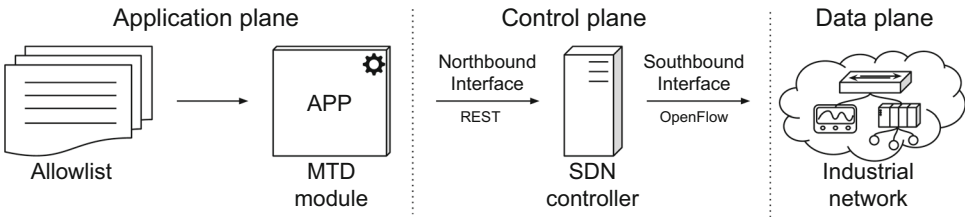


Fig. 2 Overview of the network attributes randomization architecture

more realistic MAC addresses, prefixes of MAC addresses from popular industrial devices vendors have been used.

Random port generation. For the generation of random ports, the entire available range has been used (0, 65535].

3.2.1 Limitations from an external observer perspective

From an external observer perspective, some concerns and limitations arise in the allocation and randomization of IP addresses, MAC addresses and port numbers.

IP randomization for an external observer must be implemented taking into account the available public IP addresses pool. If each end-device exposed to the Internet contains a unique public IP address or there are multiple public IP addresses available for use, IP address randomization is implementable. However, if industrial devices are exposed to the Internet via NAT (with a single public IP), the implementation of IP addresses randomization for an external observer would not be possible due to limited public IP address availability.

MAC randomization for an external observer implementation would not be effective since MAC addresses are not shared between different networks. This randomization technique is useful for adversaries aiming to discover end-devices physical addresses within the same network.

Port randomization for an external observer would be the most cost-effective randomization technique to provide false, temporal and changing exposed open ports in the reconnaissance stage of an attack. Unlike public IP addresses that are allocated by the ISPs and the number of available addresses is usually limited, users are free to use the full range of usable TCP/UDP ports to expose their services.

3.3 Real-time network attributes randomization

In the initialization phase, the initial flow rules that are installed on the switches are based on a user-defined allowlist. Taking advantage of the static nature of industrial networks and the repetitive and predictable nature of control traffic communications, this allowlist collects authorized communications and protocols between network devices. If a communication is authorized, the devices will be able to communicate with each other using real IPs, MACs and Ports. Conversely, in unauthorized communications, a device will only be able to communicate with another device using the random IP, MAC and port assigned to the target device.

The first step consists of generating and assigning a random IP/MAC address to each device on the network and a random port for each active service available in each device. These randomly assigned IP addresses, MAC addresses and listening service TCP/UDP protocol port numbers will only be valid for one MTD interval and will be replaced by other random values in the next interval. The MTD interval is defined by the user and has to be adapted for each use case. When random IPs, MACs and ports are generated and assigned to the devices, the system checks that two identical IP, MAC and port values have not been generated, thus avoiding collisions and possible errors in the operation of the system. The network configuration of the end devices is not changed, translations are performed on the OpenFlow switches and the process is completely transparent to the end devices.

Once the random network attributes are generated and assigned to each end device, flow rules are installed on the OpenFlow switches so that only authorized devices can communicate using real IPs, MACs and Ports. There are two approaches to changing or randomizing the network attributes of packets: (1) to send all packets to the SDN controller for centralized processing or (2) to process packets directly on the OpenFlow switches. In our approach, in order to minimize the delay introduced in a communication, the packets are processed directly in the OpenFlow switches, preventing each packet from being sent to the SDN controller for further processing.

For a communication between two end devices, the number of flow rules required on each OpenFlow switch depends on the number of protocols used in the communication. As detailed in the OpenFlow protocol specification [31], the Ethernet frame type must be defined in the match column in order to process packets using the header values of protocols encapsulated in the Ethernet frame. Similarly, to process packets using header values of protocols encapsulated in IP packets (e.g. TCP or UDP header fields), the protocol type must be defined in the match column of a flow rule. In our approach, ARP, IP and TCP/UDP packets are processed independently using two flow rules for each packet type, one flow rule to translate from real to random values and another flow rule to translate from random to real values.

Once the flow rules are installed on the switches, the process followed by every packet going from a source to a destination is defined in Algorithm 1. Consider an authorized communication between two end devices h_1 and h_2. When a packet arrives at the source OpenFlow switch, i.e. the switch to which h_1 is connected, the real source and destination IP (*rIP*) and MAC (*rMAC*) addresses are changed to random IP (*vIP*) and MAC (*vMAC*) addresses. In the case of the ports, if h_1 is making a request to a service of h_2, only the destination port will be changed. Otherwise, if h_1 is responding to a request of h_2, the source port will only be changed. After this translation process at the source switch, the traffic is forwarded to its destination until it reaches the destination switch, i.e. the switch to

which h_2 is connected. When the traffic reaches the destination OpenFlow switch, the random IPs *vIP*, MACs *vMAC* and Ports *vPort* are translated into real values. This way a direct and uninterrupted communication between h_1 and h_2 is guaranteed.

In the case of an unauthorized communication between two end devices, the randomization process changes slightly. In an unauthorized communication between h_1 and h_2, the destination device will only be reachable using the *vIP*, *vMAC*, and *vPort* assigned in that MTD interval to the destination device. If h_1 makes a request to a service of h_2 and the destination IP, MAC and port used by h_1 does not match the random IP, MAC and ports assigned to h_2 in that MTD interval, the packet is discarded at the source switch, preventing the packet from reaching its destination. Otherwise, if the destination IP, MAC and port used by h_1 match with those assigned in that MTD interval to h_2, the traffic is forwarded to its destination by randomizing only the source IP, MAC and port of h_1. When the network traffic reaches the destination switch, the destination *vIP* address, *vMAC* address, and *vPort* are replaced by *rIP*, *rMAC*, and *rPort* of h_2.

Algorithm 1 The process followed by each packet to reach its destination.

for all packets pkt from h_1 to h_2 **do**
  if h_1 is authorized to h_2 **then** ▷ *communication via real attributes*
    at src switch do
      set $pkt.src = \{vIP(h_1), vMAC(h_1)\}$
      set $pkt.dst = \{vIP(h_2), vMAC(h_2)\}$
    at dst switch do
      set $pkt.src = \{rIP(h_1), rMAC(h_1)\}$
      set $pkt.dst = \{rIP(h_2), rMAC(h_2)\}$
    if pkt is request **then**
      at src switch do
        set $pkt.dst = vPort(h_2)$
      at dst switch do
        set $pkt.dst = rPort(h_2)$
    else
      at src switch do
        set $pkt.src = vPort(h_1)$
      at dst switch do
        set $pkt.src = rPort(h_1)$
    end if
  else if $pkt.dst = \{vIP(h_2), vMAC(h_2)\}$ **then** ▷ *communication via random attributes*
    at src switch do
      set $pkt.src = \{vIP(h_1), vMAC(h_1)\}$
    at dst switch do
      set $pkt.dst = \{rIP(h_2), rMAC(h_2)\}$
    if pkt is request **and** $pkt.dst = vPort(h_2)$ **then**
      at dst switch do
        set $pkt.dst = rPort(h_2)$
    else if pkt is response **and** $pkt.src = rPort(h_1)$ **then**
      at src switch do
        set $pkt.src = vPort(h_1)$
    else
      drop pkt
    end if
  else ▷ *drop everything else*
    drop pkt
  end if
end for

3.4 Adaptive flow rules update process

The random IP addresses, MAC addresses and ports assigned to the devices are only valid for a user-defined time interval. At the end of each interval, these network attributes change to new randomly generated ones and are assigned to each end device, routing traffic using the newly generated network attributes. If the flow rules in use are directly deleted or updated, communications that are still using the random IPs, MACs and ports from the previous interval may be routed incorrectly, causing an interruption in the network. In order to prevent this problem and avoid the introduction of delays and interruptions in the network, an adaptive method of updating flow rules has been designed that leverages backup flow rules and the *priority* field of the OpenFlow protocol. This way, an adaptive transition to new random IP addresses, MAC addresses and ports is achieved. Fig. 3 shows the process followed at the end of each MTD interval and consists of four phases. In these examples, the instructions of the flow rules only represent the translation of IP addresses, but the instructions for translating MAC addresses and port numbers would be the same.

The first phase, represented in the Fig. 3(a), represents the initial state of the flow rules at the end of each MTD interval. In this state, the flow rules make translations between real and random IPs, MACs and ports (and vice versa) assigned in that active interval.

When an MTD interval comes to its end, new random IPs, MACs and ports are generated for the next interval. If the active flow rules are updated or deleted, traffic that is still using random IPs, MACs and ports from the previous interval may be discarded or forced to pass through the SDN controller. This is because that traffic does not have any flow rules to match. In time sensitive ICSs, interrupting or introducing delay in traffic is not acceptable. To avoid these problems, at the end of each interval, a backup rule is generated for each active flow rule on each switch. As depicted in the Fig. 3(b), the backup rule is a copy of the active flow rule but with lower priority.

With the backup rules installed, the active rules are updated by assigning the new random IP, MAC and ports generated for the new interval. This phase is represented in the Fig. 3(c). In this state, the new traffic will start using the flow rules with the highest priority, while the traffic generated in the previous interval will use the backup flow rules. With this method, an adaptive transition between intervals is achieved, without generating delays or packet loss.

Finally, as shown in Fig. 3(d), the backup rules are removed from the flow tables, returning to the initial state. This process is applied iteratively at the end of each MTD interval, regardless of the user-defined time.

4 Experimental setup

This section describes the deployed testbed to validate the proactive network attributes randomization framework. On the one hand, we describe in detail the devices and communication protocols used in the test environment. On the other hand, we present the threat model considered for this experimental phase.

Fig. 3 Process of updating flow rules at the end of an MTD interval

Flow Table		
Priority	**Match**	**Instruction**
10	$H_1 \rightarrow H_2$	src = IP_1, dst = IP_2

(a) Flow rules initial state.

Flow Table		
Priority	**Match**	**Instruction**
10	$H_1 \rightarrow H_2$	src = IP_1, dst = IP_2
9	$H_1 \rightarrow H_2$	src = IP_1, dst = IP_2

(b) A backup flow rule is added with the same instruction but lower priority.

Flow Table		
Priority	**Match**	**Instruction**
10	$H_1 \rightarrow H_2$	src = IP_3, dst = IP_4
9	$H_1 \rightarrow H_2$	src = IP_1, dst = IP_2

(c) The instruction of the flow rule with the highest priority is updated with new network attributes.

Flow Table		
Priority	**Match**	**Instruction**
10	$H_1 \rightarrow H_2$	src = IP_3, dst = IP_4

(d) The backup flow rule is removed from the flow table

4.1 Experimental topology

The experimental topology, represented in Fig. 4, is composed of two independent production lines and end devices from different vendors. The devices have been deployed in two levels. In the first level, those devices that are closer to the physical industrial process such as PLCs or HMIs are deployed. These types of devices are usually designed to operate in hostile environments with high temperatures, vibration or electrical noise, featuring high reliability. On the other hand, in the second level, devices that are not normally used in such hostile environments, such as SCADA servers or MQTT brokers, have been deployed.

Regarding the deployed devices, we have used real industrial equipment, which make up the two production lines previously mentioned. On the one hand, we have a Siemens 1515C PLC, a Siemens SIMATIC IoT2040 that works as a gateway for the PLC and a Siemens HMI KTP600 used to visually monitor the industrial process. The HMI and the IoT devices pull data from the PLC using OPC UA protocol. The HMI displays the process values while the IoT device publishes the results to a message broker using the MQTT protocol. The data published by the IoT devices is consumed by the SCADABR SCADA server by subscribing to the MQTT broker queue. On the other side, we deployed another production line using an ABB PLC AC800M and ABB Compact working as an SCADA server. The ABB SCADA server is configured to pull data from the PLC every second, using the Manufacturing Message Specification (MMS) protocol.

In order to find out the open ports of the devices, we scanned the entire TCP protocol port range on each device without the MTD framework active. The results can be found in Table 1.

The network forwarding decisions are taken in a centralized manner using Ryu [32] SDN controller and 3 OpenVSwitch switches that connect the different devices in the network. Our application that manages the flow rules that perform IP, MAC and port randomization, interacts with the SDN controller using the northbound interface, a REST API. The SDN controller transfers these requests to the switches using the OpenFlow protocol and the forwarding devices process the packets directly on the switches.

4.2 Threat model

The industrial environment in which exists different types of industrial and forwarding devices is SDN-based. We assume that the SDN controller and the applications that interact with the controller through the northbound interface are trusted, while the attacker is in the internal network. To simulate a potential attacker and to test the full potential of our proposed proactive defense system, we have placed a computer performing a Man-in-the-Middle (MITM) attack on a link between two OpenFlow switches of the testbed. The attacker is able to recollect and inject network packets from the compromised link. Note that if an end device becomes compromised and reconnaissance attacks are launched from there, the attacker would be able

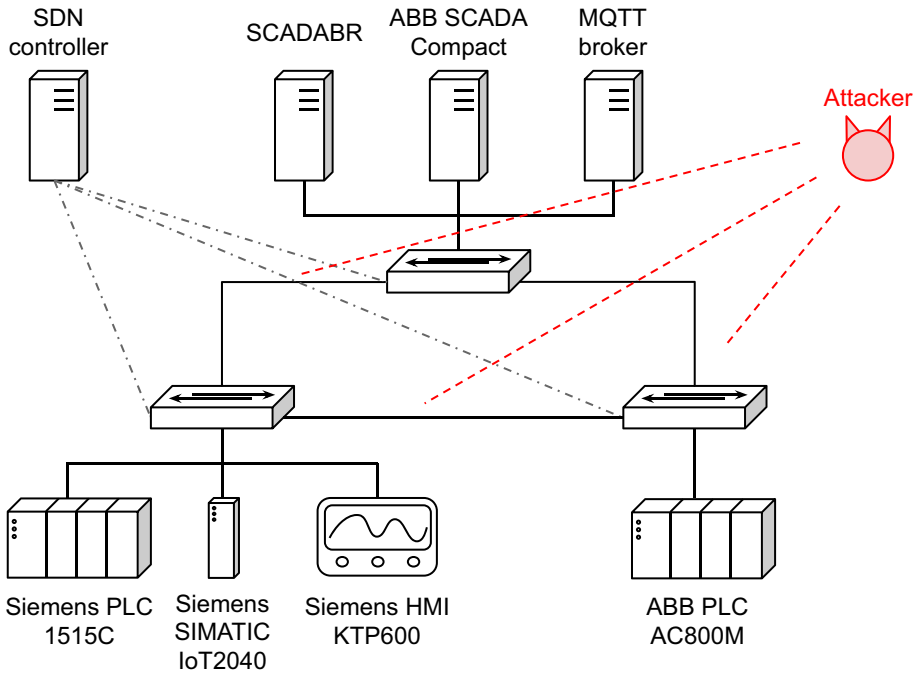


Fig. 4 The deployed experimental setup composed of different industrial control systems, forwarding devices and SDN controller

Table 1 Open ports per device

Device name	Open ports
Siemens PLC 1515C	80, 102, 443, 4840
Siemens SIMATIC IoT2040	22, 1534, 1880
Siemens HMI KTP600	102, 2308
ABB PLC AC800M	102
SCADABR	8080
ABB SCADA Compact	22, 80, 102, 135, 445, 1536, 1537, 1538, 1539, 1540, 1541, 1543, 5040, 17500
MQTT broker Mosquitto	1883

to obtain a mixture of real (in case of authorized communications) and random (in case of unauthorized communications) information about end devices and protocols. In our scenario, the attacker's main goal is to gather intelligence about active devices on the network and services/protocols used for device-to-device communication. We define the attacker's characteristics as follows:

Attack targets: In an industrial environment, any device or communication could be vulnerable to attacks. Most vulnerabilities in ICS occur due to the lack of authentication mechanisms, encryption and integrity checks in industrial control and monitoring protocols [33]. As most of the protocols were designed to operate in isolated environments, not much importance was given to security in their design [34]. Although some industrial manufacturers have integrated encryption, integrity checks and authentication support into protocols, legacy protocols without any security mechanisms are still widely used [35]. In our scenario, we assume that every device in the industrial network can be a potential attack target.

Attack procedure: Based on the Cyber Kill Chain (CKC) [36] attack model, we assume that the attacker starts with the first stage of the CKC. The attacker performs a reconnaissance attack in order to identify real assets on the network (IP and port scanning). The attacker will first attempt to identify the devices on the network by scanning the entire IP address space of the network. Once the devices have been identified, the attacker will run a port scan on each device. Due to the existence of the proactive network attributes randomization framework, the information that the attacker will receive will be random and will change over time.

5 Results and discussion

In this section, we investigate the impact of the presented work on the performance of the network. Then, we analyze the effectiveness of mitigating reconnaissance attacks in different randomization interval scenarios. Finally, we discuss several topics related to this approach.

5.1 Performance

To measure the impact on the performance of the presented work, we have taken into account the delay introduced in the network by using the Round Trip Time (RTT) metric. Also, we measured the flow tables length of the OpenFlow switches and compared the number of flow rules needed in a static and MTD network.

Round Trip Time: To see the overhead introduced in the network by the MTD defense system presented in this article, the measurement Round Trip Time (RTT) has been used. RTT measures the time it takes for a data packet to return to its sender after having passed through its destination. Measurements have been performed taking into account the longest path of the topology (two switch hops), in this case the communication between the ABB PLC AC800M and the ABB SCADA server. Six different scenarios have been considered. On the one hand, measurements were performed on a static network without randomization of IPs, MACs and ports. On the other hand, measurements on an MTD network with randomization intervals of 60, 30, 10, 5 and 1 second. For each scenario, 15 measurements of 150 seconds each were performed. The Table 2 represents the mean, standard deviation and minimum/maximum RTT values of the results of the measurements.

Flow table length: The number of flow rules that perform translations between IP addresses, MAC addresses and ports varies depending on the number of end devices and services available in the network. Let's consider a list of end devices $\delta \in \Delta$ in the network where each end device has a certain number of open TCP/UDP ports $p(\delta)$. N represent the total number of end devices in the network.

In a static network, using a simple switch S_w that only works on the second layer of the OSI reference model and the forwarding is performed using destination MAC addresses of network packets, a single rule for each end device available in the network would be enough. Equation 1 represents the number of required flow rules on a switch in a static network.

Table 2 RTT between the ABB SCADA and PLC in a static and MTD network

		MTD interval					
		Without MTD	60s	30s	10s	5s	1s
RTT (ms)	avg	6.316	6.376	6.383	6.414	6.43	6.505
	stdv	0.166	0.132	0.119	0.148	0.173	0.222
	min	5.952	6.08	6.115	6.138	6.084	6.102
	max	6.818	6.925	6.866	6.991	7.068	7.323

$$F^S(S_w) = N \tag{1}$$

In the case of the MTD framework presented in this article, in a communication between two devices, a total of 2 flow rules for ARP packets, 2 for IP packets and 2 flow rules for each TCP/UDP open port is needed in a switch S_w. Equation 2 represents the number of required flow rules on a switch in an MTD network.

$$F^M(S_w) = 1 + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(p(\delta_i) + p(\delta_j) + 2), \delta \in \Delta \tag{2}$$

Figure 5 compares the number of flow rules required in a static and MTD network for different number of end devices available in the network.

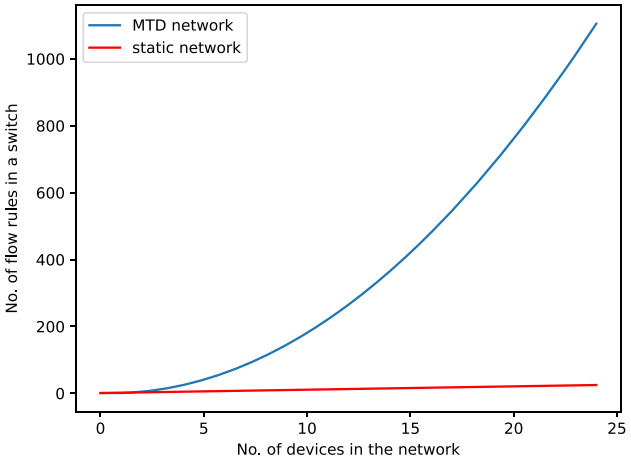


Fig. 5 Flow table length of a switch for different number of end devices available in the network

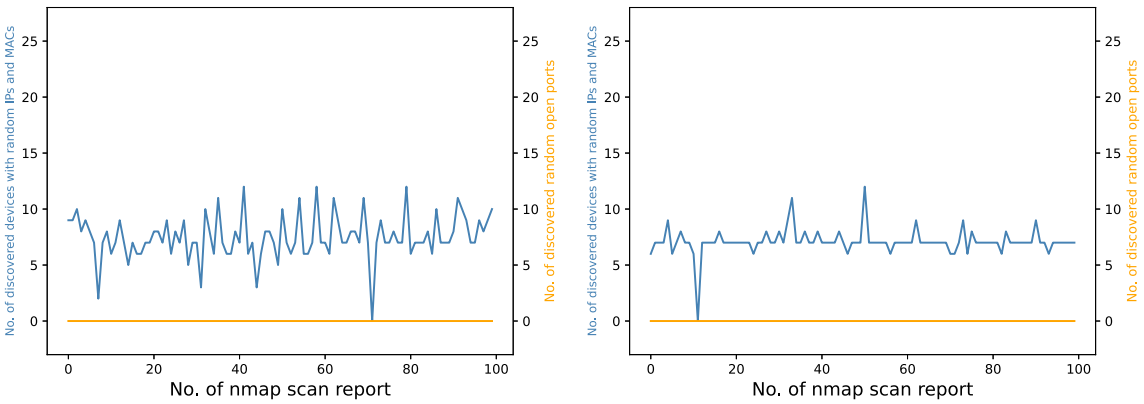
5.2 Reconnaissance attacks mitigation

To test the effectiveness of mitigating reconnaissance attacks, we performed 100 consecutive scans on our network with 24 bits of subnet mask. The *nmap* [37] tool has been used to perform scans looking for active devices and open ports on the network. The *nmap* tool has multiple methods for discovering devices and open ports on a network. By default, *nmap* first starts scanning the entire IP address space looking for devices. Once active devices are discovered, executes a port scan on each device discovered in the previous step. The following techniques have been tested to discover active devices and open ports in the network: *TCP SYN scan, TCP ACK scan, TCP connect scan, TCP FIN/NULL/XMAS scan, TCP window scan, TCP maimon scan, TCP idle scan, IP Protocol scan, ICMP ping and ARP Ping*.

On the one hand, scans have been performed on a traditional static network without the MTD proactive defense mechanism. By scanning the entire IP address and port range, the attacker is able to identify in each scan all active devices and services on the network. On the other hand, with the proactive MTD defense active, 100 consecutive scans were performed with different randomization intervals (1, 5, 10, 30 and 60 seconds). The different subfigures represented in the Figure 6 reflect the number of hosts with random IP/MAC addresses and open ports discovered in 100 consecutive scans of the entire IP address and port space.

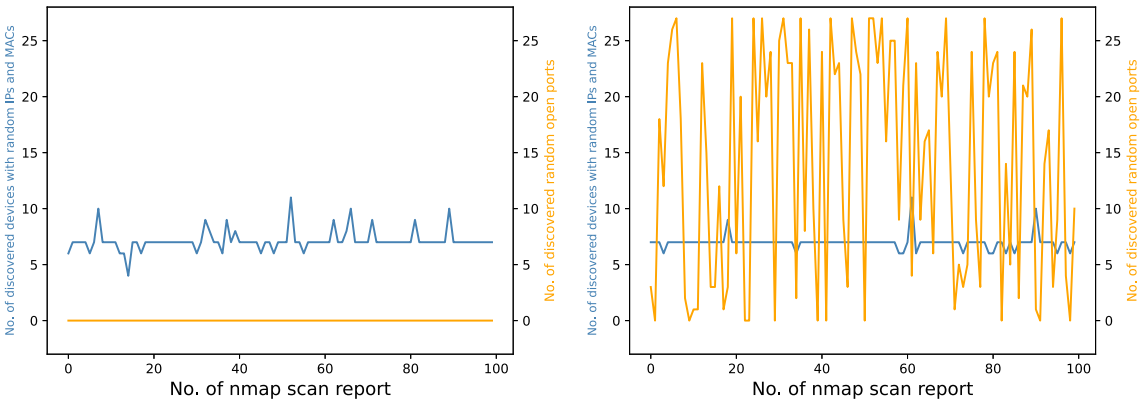
As reflected in the different subfigures, when the randomization interval decreases, the number of devices discovered with randomized IP and MAC addresses in different scans varies more than with longer randomization intervals. This is because with longer randomization intervals, the probability that a scan of the entire IP address space of the network will be performed within the limits of that interval increases. At shorter randomization intervals, the probability that a complete scan will be executed in more than one interval is higher, increasing the randomness of the results.

In the case of port scanning, only scans performed in randomization intervals of 60 and 30 seconds were able to discover random open ports. With shorter randomization intervals, information obtained in reconnaissance attacks is rapidly invalidated. When an attacker discovers an active host in the network and starts scanning its ports, if the randomization interval comes to its end and new random IP and MAC addresses are assigned to that host, the attacker loses access to the host by using the random IP and MAC addresses assigned to that host in the previous interval, interrupting the port scanning process. The attacker has to rescan the IP address space in order to discover active



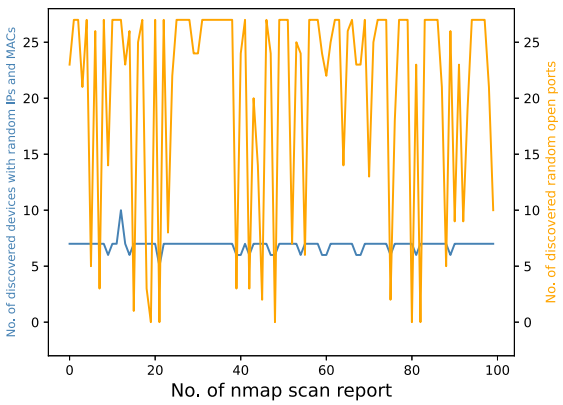
(a) 1 second MTD interval.

(b) 5 seconds MTD interval.



(c) 10 seconds MTD interval.

(d) 30 seconds MTD interval.



(e) 60 seconds MTD interval.

Fig. 6 Results obtained in the experimental setup with 100 consecutive nmap scans with different randomization intervals. The blue line indicates the number of hosts discovered with random IP and MAC addresses in each nmap scan. The orange line indicates the number of random open ports discovered in each nmap scan

devices in the industrial network. Due to the time required to scan the entire IP address space and port range, at 10, 5 and 1 second randomization intervals, the information obtained in the host discovery phase becomes irrelevant before the port scan is performed, making port scanning of the network devices impossible.

5.3 Discussion

The performance in terms of RTT is similar with different MTD intervals because the transition to new random IPs, MACs and ports is done gradually using backup flow rules. The optimal MTD interval will depend on the criticality of the system to be defended and should be adapted to each use case. With a shorter randomization interval, the information obtained by the attacker is more diffuse and the information obtained in previous reconnaissance becomes irrelevant more frequently. As opposed to longer intervals, the information obtained is more constant, even the same information can be obtained in several consecutive scans if they are performed in the same MTD interval. For this, the criticality of each device could be quantified, and more frequent randomization intervals could be defined for more critical communications while longer intervals could be defined for less critical communications.

The number of flow rules required in switches for network attributes randomization increases considerably depending on the number of devices and TCP/UDP open ports in the network. In these scenarios, when a flow table becomes large, network performance can be adversely affected, and poses higher requirements on hardware. This is something to be considered especially in time-sensitive environments such as ICSs. In order to solve this problem and optimize the randomization process in large networks where many flow rules are required, OpenFlow 1.1 and later versions support pipeline processing with multiple flow tables. When a packet arrives to a switch, matching process starts in the table with the lowest sequence number and continues to additional flow tables in the pipeline, sequentially or directly jumping to specific tables. Leveraging this, groups of flow rules could be created in different tables, preventing packets from being processed by all flow entries in a single table, reducing flow table length and improving the matching process efficiency.

In large network infrastructures, updating all flow rules at the same time could also generate network performance problems by introducing overhead spikes on network switches or the SDN controller. In our proposed network attributes randomization solution, all flow rules installed in the switches are updated at the end of each MTD interval, by assigning new random IP addresses, MAC addresses, and port numbers translations instructions to flow rules. A large amount of requests from the application plane to the control plane and from the control plane to the data plane could generate overhead problems on the SDN controller or network switches, requiring high computational resources to respond to all requests without affecting network performance. In the event that network performance is affected by this problem, SDN enables developing strategies for updating flow rules gradually. Instead of updating all flow rules at once, flow rules could be updated in blocks or individually in order to reduce the overhead in the network.

The integration of IDSs in an MTD network where network identifiers are randomized could be a concern due to the risk of false alarms generation. The development of IDSs on top of MTD networks is not new and several approaches have been proposed in the literature [38]. In the presented network attributes randomization framework, the MTD module, deployed in the application plane and which uses the SDN controller to install, update and remove flow rules from the data plane, is capable of operating over the entire network and is aware of the mappings between real and random network attributes. Leveraging the network-wide management and global view capabilities, SDN enables to dynamically manage network packets by translating from random to real network attributes before mirroring or forwarding to an IDS. Such flexibility allows the randomization framework presented in this manuscript to be interoperable with different types of IDSs.

Finally, security through obscurity is a technique that uses concealment to provide security. MTD can be considered a security through obscurity technique, especially Shuffling-based MTD techniques that base their security on preventing an attacker from discovering potential attack vectors by hiding the configuration, services, or devices available on the network. As cited in volume 2 of NIST SP 800-160 [39], security through obscurity cannot be the primary defense mechanism. These types of techniques can be used as a complementary layer of security in secure and resilient environments.

6 Conclusion

This paper presents a mechanism that randomizes IP addresses, MAC addresses and port numbers in industrial communications using the SDN paradigm. The main objective of this system is to proactively mitigate reconnaissance attacks and prevent an unauthorized device from communicating with its target. This is achieved by assigning random IP and MAC addresses, and port numbers to each end device in the network that are only valid for a limited time period. The results show that this system helps the information obtained through reconnaissance attacks to lose relevance because it is only valid temporarily. Also, because the transition to new random IP addresses, MAC addresses and port numbers is done adaptively, the delay introduced compared to a traditional static network is minimal, allowing the solution to be implemented in time-critical systems.

As future lines, we want to extend our approach by implementing the transition to new random IPs, MACs and port numbers independently for each network flow, so that it can be deployed in large networks without affecting network performance. We also want to explore the possibility of integrating industrial honeypots in an MTD network.

Acknowledgements This work has been developed by the intelligent systems for industrial systems group supported by the Department of Education, Language Policy and Culture of the Basque Government (IT1676-22). It has been partially funded by REMEDY project. This project has received funding from the Department of Economic Development and Infrastructures under the grant agreement KK-2021/00091. This work was partially funded by the GAITZERDI Project of the Science, Technology and Innovation Network of Gipuzkoa (2022-CIEN-000065-01).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., Hahn, A. (2015). Guide to industrial control systems (ics) security. https://doi.org/10.6028/NIST.SP.800-82r2
2. Iturbe, M., Garitano, I., Zurutuza, U., Uribeetxeberria, R. (2016). Visualizing network flows and related anomalies in industrial networks using chord diagrams and whitelisting. In: VISIGRAPP (2: IVAPP), pp. 101–108
3. Cho, J.-H., Sharma, D. P., Alavizadeh, H., Yoon, S., Ben-Asher, N., Moore, T. J., Kim, D. S., Lim, H., & Nelson, F. F. (2020). Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys Tutorials, 22*(1), 709–745. https://doi.org/10.1109/COMST.2019.2963791
4. Zheng, J., & Namin, A. S. (2019). A survey on the moving target defense strategies: An architectural perspective. *Journal of Computer Science and Technology, 34*(1), 207–233.
5. Sainz, M., Iturbe, M., Garitano, I., & Zurutuza, U. (2018). Software defined networking opportunities for intelligent security enhancement of industrial control systems. In H. Pérez García, J. Alfonso-Cendón, L. Sánchez González, H. Quintián, & E. Corchado (Eds.), *International joint conference SOCO'17-CISIS'17-ICEUTE'17 León, spain, september 6–8, 2017, proceeding* (pp. 577–586). Cham: Springer.
6. Boucadair, M., & Jacquenet, C. (2014). Software-defined networking: A perspective from within a service provider environment. RFC Editor. https://doi.org/10.17487/RFC7149. https://www.rfc-editor.org/info/rfc7149
7. Molina, E., & Jacob, E. (2018). Software-defined networking in cyber-physical systems: A survey. *Computers and Electrical Engineering, 66*, 407–419. https://doi.org/10.1016/j.compeleceng.2017.05.013
8. Jafarian, J.H., Al-Shaer, E., & Duan, Q. (2012). Openflow random host mutation: Transparent moving target defense using software defined networking. In: Proceedings of the first workshop on hot topics in software defined networks. HotSDN '12, pp. 127–132. Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2342441.2342467
9. Sharma, D.P., Kim, D.S., Yoon, S., Lim, H., Cho, J.-H., & Moore, T.J. (2018) Frvm: Flexible random virtual ip multiplexing in software-defined networks. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 579–587. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00088
10. Chowdhary, A., Alshamrani, A., Huang, D., & Liang, H. (2018). Mtd analysis and evaluation framework in software defined network (mason). In: Proceedings of the 2018 ACM international workshop on security in software defined networks & network function virtualization. SDN-NFV Sec'18, pp. 43–48. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3180465.3180473
11. Aydeger, A., Manshaei, M. H., Rahman, M. A., & Akkaya, K. (2021). Strategic defense against stealthy link flooding attacks: A signaling game approach. *IEEE Transactions on Network Science and Engineering, 8*(1), 751–764. https://doi.org/10.1109/TNSE.2021.3052090
12. Skowyra, R., Bauer, K., Dedhia, V., & Okhravi, H. (2016). Have no phear: Networks without identifiers. In: Proceedings of the 2016 ACM workshop on moving target defense, pp. 3–14
13. Wang, Y., Chen, Q., Yi, J., & Guo, J. (2017). U-tri: Unlinkability through random identifier for sdn network. In: Proceedings of the 2017 workshop on moving target defense. MTD '17, pp. 3–15. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3140549.3140554
14. Chavez, A.R., Stout, W.M., & Peisert, S. (2015) Techniques for the dynamic randomization of network attributes. In: 2015 international carnahan conference on security technology (ICCST), pp. 1–6. IEEE
15. Zhou, Y., Cheng, G., & Yu, S. (2021). An sdn-enabled proactive defense framework for ddos mitigation in iot networks. *IEEE Transactions on Information Forensics and Security, 16*, 5366–5380. https://doi.org/10.1109/TIFS.2021.3127009
16. Kampanakis, P., Perros, H., & Beyene, T. (2014). Sdn-based solutions for moving target defense network protection. In: Proceeding of IEEE international symposium on a world of wireless, mobile and multimedia networks 2014, pp. 1–6. IEEE
17. Koo, H., Chen, Y., Lu, L., Kemerlis, V.P., & Polychronakis, M. (2018). Compiler-assisted code randomization. In: 2018 IEEE symposium on security and privacy (SP), pp. 461–477. https://doi.org/10.1109/SP.2018.00029
18. Huang, Y., & Ghosh, A.K. (2011). Introducing diversity and uncertainty to create moving attack surfaces for web services. Springer New York, 131–151. https://doi.org/10.1007/978-1-4614-0977-9_8
19. Taguinod, M., Doupé, A., Zhao, Z., & Ahn, G.-J. (2015). Toward a moving target defense for web applications. In: 2015 IEEE international conference on information reuse and integration, pp. 510–517. https://doi.org/10.1109/IRI.2015.84

20. Li, Y., Dai, R., Zhang, J. (2014). Morphing communications of cyber-physical systems towards moving-target defense. In: 2014 IEEE international conference on communications (ICC), pp. 592–598. https://doi.org/10.1109/ICC.2014.6883383
21. Kanellopoulos, A., & Vamvoudakis, K. G. (2020). A moving target defense control framework for cyber-physical systems. *IEEE Transactions on Automatic Control, 65*(3), 1029–1043. https://doi.org/10.1109/TAC.2019.2915746
22. Alavizadeh, H., Hong, J.B., Jang-Jaccard, J., & Kim, D.S. (2018). Comprehensive security assessment of combined mtd techniques for the cloud. In: Proceedings of the 5th ACM workshop on moving target defense. MTD '18, pp. 11–20. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/3268966.3268967
23. Alavizadeh, H., Jang-Jaccard, J., & Kim, D.S. (2018). Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/ 12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), pp. 573–578 .https://doi.org/10.1109/TrustCom/BigDataSE.2018.00087
24. netfilter.org project, T.: Netfilter: Firewalling, NAT and Packet Mangling for Linux. https://www.netfilter.org/ Accessed 2022-07-27
25. Ulrich, J., Drahos, J., & Govindarasu, M. (2017). A symmetric address translation approach for a network layer moving target defense to secure power grid networks. In: 2017 Resilience week (RWS), pp. 163–169 (2017). https://doi.org/10.1109/RWEEK.2017.8088667
26. Pappa, A.C., Ashok, A & Govindarasu, M. (2017). Moving target defense for securing smart grid communications: Architecture, implementation amp; evaluation. In: 2017 IEEE power energy society innovative smart grid technologies conference (ISGT), pp. 1–5. https://doi.org/10.1109/ISGT.2017.8085954
27. Germano da Silva, E., Dias Knob, L.A., Wickboldt, J.A., Gaspary, L.P., Granville, L.Z., & Schaeffer-Filho, A. (2015). Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study. In: 2015 IFIP/IEEE international symposium on integrated network management (IM), pp. 165–173 (2015). https://doi.org/10.1109/INM.2015.7140289
28. Ndonda, G.K., & Sadre, R. (2017). A low-delay sdn-based countermeasure to eavesdropping attacks in industrial control systems. In: 2017 IEEE conference on network function virtualization and software defined networks (NFV-SDN), pp. 1–7. https://doi.org/10.1109/NFV-SDN.2017.8169840
29. Chavez, A.R. (2019). Moving target defense to improve industrial control system resiliency. In: industrial control systems security and resiliency, pp. 143–167. Springer
30. 3rd, D.E.E., & Abley, J. (2013). IANA considerations and IETF protocol and documentation usage for IEEE 802 parameters. RFC editor. https://doi.org/10.17487/RFC7042. https://www.rfc-editor.org/info/rfc7042
31. Foundation, O.N. OpenFlow switch specification, Version 1.3.5. https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.3.5.pdf Accessed 2022-07-12
32. Ryu SDN Framework. https://ryu-sdn.org/ Accessed 2022-07-27
33. Gómez, Á. L. P., Maimó, L. F., Celdran, A. H., Clemente, F. J. G., Sarmiento, C. C., Masa, C. J. D. C., & Nistal, R. M. (2019). On the generation of anomaly detection datasets in industrial control systems. *IEEE Access, 7*, 177460–177473. https://doi.org/10.1109/ACCESS.2019.2958284
34. Conti, M., Donadel, D., & Turrin, F. (2021). A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials, 23*(4), 2248–2294.
35. Barbieri, G., Conti, M., Tippenhauer, N.O., & Turrin, F. (2020). Sorry, shodan is not enough! assessing ICS security via IXP network traffic analysis. CoRR abs/2007.01114 2007.01114
36. Assante, M.J., & Lee, R.M. (2015). The industrial control system cyber kill chain. SANS Institute InfoSec Reading Room **1**
37. Nmap: the Network Mapper - Free Security Scanner. https://nmap.org/ Accessed 2022-07-17
38. Zhao, Z., Liu, F., & Gong, D. (2017). An sdn-based fingerprint hopping method to prevent fingerprinting attacks. Security and Communication Networks **2017**
39. Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. (2019). *Developing cyber resilient systems: A systems security engineering approach*. National Institute of Standards and Technology: Technical report.

Xabier Etxezarreta received the bachelor's degree in Computer Engineering in 2019 and the master's degree in Data Analytics, Cybersecurity and Cloud Computing in 2021, both at Mondragon Unibertsitatea, Arrasate-Mondragón, Spain. During the course of his studies, he has participated in numerous research projects related to artificial intelligence and multi-objective optimization. He is currently pursuing his doctoral studies, researching on intrusion detection and response techniques based on SDN for industrial control systems.

Iñaki Garitano is a researcher/lecturer in the Intelligent Systems for Industrial Systems research group, and member of the Data Analysis and Cybersecurity research area of Mondragon Unibertsitatea. He holds a PhD titled Behavioral Modeling for Anomaly Detection in Industrial Control Systems, and MSc in Telecommunication Engineering. Before joining Mondragon Unibertsitatea in 2015, he worked in Universitetssenteret påKjeller (UNIK) as postdoctoral researcher. In addition, he is the CTO of the Basic Internet Foundation. His main research interest is in the area of Cybersecurity applied to Industrial Automation and Control Systems and Information and Communication Networks including Internet of Things, and Application Container technologies. He currently participates in several European, national and regional level public funding projects.

Mikel Iturbe received the M.Sc. degree in ICT security from the Open University of Catalonia, Barcelona, Spain, in 2013, and the Ph.D. degree from Mondragon Unibertsitatea, Arrasate-Mondragón, Spain, in 2017, where he worked on data-driven intrusion detection in industrial networks. He is a Lecturer and a Researcher with Mondragon Unibertsitatea. He is currently part of the Data Analysis and Cybersecurity Research Group. His main research interest is related to cybersecurity, primarily in the industrial sector. The main lines he works on are industrial control system security, embedded security, and software security. He also works in exploring novel data-driven applications for cybersecurity.

Urko Zurutuza is the principal investigator of the Intelligent Systems for Industrial Systems research group, and coordinator of the Data Analysis and Cybersecurity research area. He obtained his PhD in January 2008 at Mondragon Unibertsitatea, in collaboration with the Zürich IBM Research Lab. His research interests revolve around applications of Machine Learning to real world problems, and specially Cybersecurity. He has published more than 20 articles in high impact journals, more than 55 publications in blind peer-reviewed conferences, edited 3 books (2 of them as conference proceedings), and coauthored 7 book chapters. He is member of the Board of Directors of RENIC (National Network of Excellence in Cybersecurity Research), and serves in Steering Boards of leading international conferences such as DIMVA or RAID.