No Need to Marry to Change Your Name! Attacking Profinet IO Automation Networks Using DCP

Stefan Mehner[1] and Hartmut König[1]

Brandenburg University of Technology Cottbus - Senftenberg
Computer Networks and Communication Systems Group
{stefan.mehner,hartmut.koenig}@b-tu.de

Abstract. Current developments in digitization and industry 4.0 bear new challenges for automation systems. In order to enable interoperability and vertical integration of corporate management systems, these networks have evolved from formerly proprietary solutions to the application of Ethernet-based communication and internet standards. This development is accompanied by an increase in the number of threats. Although the most critical IT protection objective for automation systems is availability, usually no security mechanisms have been integrated into automation protocols. Also Ethernet offers no protection by design for these protocols. One of the most popular real-time protocols for industrial applications is Profinet IO. In this paper, we describe a Denial-of-Service attack on Profinet IO that exploits a vulnerability in the Discovery and Basic Configuration Protocol (DCP) which interrupts the Application Relationship between an IO Controller and an IO Device, and thus prevents the system from being repaired by the operator. The attack combines port stealing with the sending of forged DCP packets and causes a system downtime, which in affected production networks probably lead to a serious financial damage and, in case of critical infrastructures, even represents a high risk for the supply of society. We demonstrate the practical feasibility of the attack using realistic hardware and scenarios and discuss its significance for also other setups.

Keywords: Industrial control systems · Profinet IO attacks · Profinet IO vulnerabilities

1 Motivation

The ongoing digitization of industrial controls systems (ICS) calls for appropriate security measures to guarantee previous reliability and security. While these systems formerly used to operate in isolated environments within a distinct hierarchy based on mainly proprietary equipment, recent developments have led to an erosion of these natural protective barriers. Here, the benefits of standardization and the integration into corporate networks, such as better interoperability, maintenance, and control, lead to serious security issues.

In 2004, several real-time capable Ethernet-based fieldbus protocols were introduced by the IEC[1] [3] that can coexist with non-real-time Ethernet communications and facilitate the *vertical* integration by interoperability with legacy equipment at fieldbus level. In addition, the *horizontal* integration from the automation level up to the corporate network enables access for production process optimization, e. g., enterprise resource planning (ERP) systems or predictive maintenance solutions.

The downside of this development is the increasing threat of a broader range of attack vectors because new exploitable ways to access the systems have been created. Moreover, known attacks on common information and communication technology (ICT) can now also be adapted to Industrial Ethernet environments. Additionally, Ethernet inherently offers no security features, such as encryption or authentication. A setup error is sufficient to make the configuration interface (built-in web server) of a programmable logic controller (PLC) accessible via public networks. The tremendous number of PLCs on the Internet identified by specialized search engines, such as Shodan [1], demonstrate the drawback of standardization and the lack of integrated security. Since ICSs also represent a core element of many critical infrastructure environments, e. g., in case of power and water supply, or public train service, a successful attack can have a tremendous impact on larger parts of society. Recent examples are the attacks on power plants in Japan and South Korea in 2014 [18] and in the Ukraine in 2015 [10]. Countermeasures to prevent such threats are very limited because security is not included in the concepts of Industrial Ethernet or in fieldbus standards, and the introduction of additional security measures on embedded devices, whose resources are tailored and limited to fulfill only the specified automation tasks, is not feasible. For this reason, only general security measures, such as perimeter protection, network separation, or access control, are applied. However, these conventional measures are undermined by the named horizontal and vertical integration trends.

With 11% of the overall market share in industrial networks, an amount of 24% considering Ethernet-based field protocols and the application to about 20 million installed devices [8], Profinet IO is currently one of the most used automation protocols. As already mentioned, Profinet IO devices have no security functions in the sense of endpoint security [19]. Apart from some basic principles for communication control, such as the use of frame IDs for communication relations identification or the cycle counter for monitoring the IO data exchange, there are no barriers to disturb Profinet IO communication and devices. The only prerequisite for a targeted attack is an attacker eager to gain sufficient knowledge about the ongoing physical process in order to conduct comparatively primitive attacks that lead to an enduring failure state.

[1] International Electrotechnical Commission

In this paper, we present a recently identified attack with these characteristics. The attack causes the respective Profinet IO devices to enter a failure state that cannot be reset even by a complete device reboot. Hence, it is an example of an attack that can lead to a persistent breakdown of the related process control network with potentially severe consequences in case of a harmed critical infrastructure. We reported that case to a CERT and now started a disclosure process for this vulnerability. Furthermore, we are in discussions with the vendors and the Profinet organization. The main contributions of this paper are:

- the implementation and evaluation of attacks on known but not yet practically exploited vulnerabilities of Profinet IO [14],
- the presentation of a novel attack with long-term effects, and
- a comprehensive evaluation of the applicability of the attack with different hardware and topologies.

The following Section 2 introduces some basics of Profinet IO necessary to understand the attack that is presented in Section 3. The experimental setup and the results obtained performing the attack are explained in Section 4. After that, in Section 5 the presented work is put in the context of releated research concerning Profinet IO based attacks. This is concluded in the last section by a summary and an outlook on further research for preventing the identified attack.

2 Profinet IO Essentials

Profinet IO is an Ethernet-based fieldbus protocol with real-time capability specified in IEC 61784-2 [3]. In Real-Time (RT) mode, sending cycles of up to 1 ms are specified. This is achieved by precise timing and direct communication on the MAC layer. If lower cycles are required, e. g., in motion control systems, the Isochronous Real-Time (IRT) mode can be applied, which, however, requires special hardware due to the use of an adapted MAC layer, here. Furthermore, there is also a Non-Real-Time (NRT) mode that is based on UDP/IP. It is used for non-time-critical communication, such as diagnostics and configuration. A minimal Profinet IO system consists at least of one PLC and one or more devices as peripheral equipment connected over Ethernet. The standard supports star, tree, and ring topologies as well as a line topology implemented by the integrated switch functionality in the Profinet IO devices [16].

2.1 Profinet IO Device Classes

Profinet IO defines three device roles. The *IO Supervisor* is an engineering device used for project engineering, diagnostics, and troubleshooting. It usually is a PC, a Human Machine Interface (HMI) or a programming device. The automation routine is executed in the *IO Controller*, which is typically a PLC. An *IO Device* is a distributed field device that exchanges data (e. g., sensor values) with one or more IO Controller. Every Profinet IO setup contains at least one IO Controller and one IO Device.

2.2 Configuration

Figure 1 depicts the eight steps from the configuration to the operational stage. At first, (1) the system is planned with the help of the IO Supervisor. In detail, an engineering software is used to model the desired topology as well as the automation process. Thereafter, (2) the IO Supervisor sets the IP address of the IO Controller and then (3) the device name. Next, the engineered project setup from (1) is then transferred to the IO Controller. After that, the work of the IO Supervisor is finished. The IO Controller (5) checks the name of the device and (6) assigns the configured IP address. This process is explained in detail in the following section. Before any process data can be exchanged, (7) a logical channel called Application Relationship (AR) has to be established between the IO Controller and the IO Device. Within an AR, further Communication Relationships (CRs) are set up, as shown in Figure 2. For the acyclic transmission of records (e. g., configuration parameters, diagnostics), a Record Data CR is used over the non-real-time channel, whereas cyclic data exchange and alarms are sent over the real-time channel. The connection is established and (8) the real-time data exchange starts. The details of this operational stage are not discussed further in this paper, as they are not relevant for the understanding of the attack.

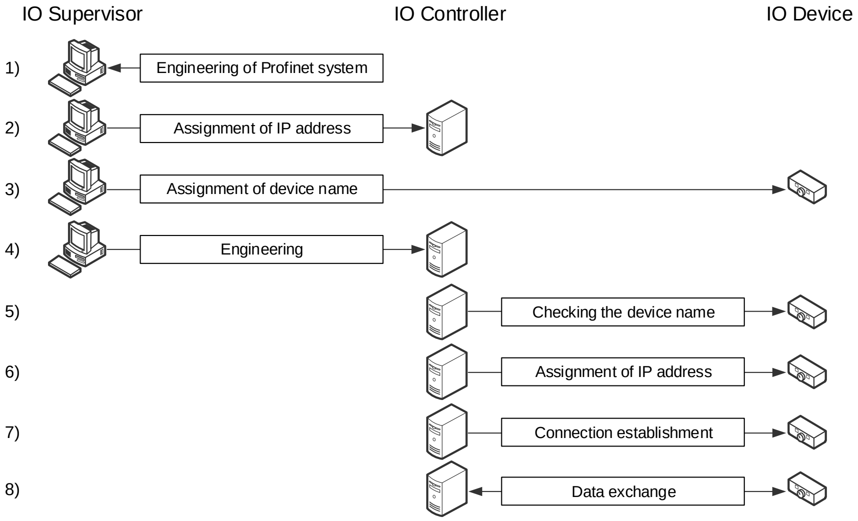


Fig. 1. Steps from project engineering to operational stage

2.3 Name and IP Assignment Using Profinet DCP

Before setting up a connection, the IO Supervisor assigns names to the IO Devices using the Discovery and basic Configuration Protocol (DCP). The name must be unique for every device of the Ethernet subnet and complies with the DNS conventions [17]. An example setup is illustrated in Figure 3a. Here, the

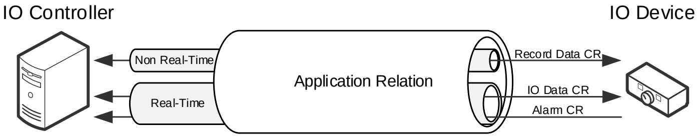
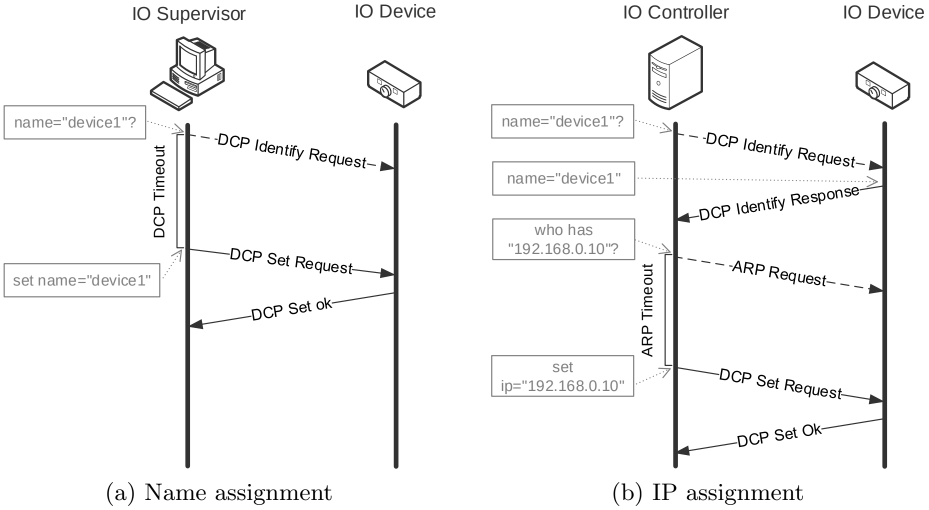


Fig. 2. Data exchange within an Application Relationship

name "device1" is assgned to the IO device. First, a *DCP Identify request* with the desired name is sent by the IO Supervisor to the Profinet IO multicast address. If a device has already assigned this name, it sends a *DCP Identify response* immediately. If no response is received within a timeout time (DCP Timeout) the supervisor assumes that the name is not already set. In this case, a *DCP Set request* is sent to the MAC address of the IO Device to set the desired name "device1". When the process is successful, it is concluded with a *DCP Set response* to the supervisor.

The situation is similar for the assignment of the IP address (see Figure 3b). Initially, a *DCP Identify request* is sent by the IO Controller to the multicast address to ask if the name "device1" is already assigned. The IO Device answers with a *DCP Identify response* directly to indicate that the name is assigned for this device. In the next step, an ARP request is broadcasted to determine if the desired IP address "192.168.0.10" is already assigned to another device. When no ARP reply is received within a certain time, it is assumed that the address is still available and a *DCP Set request* is sent to the IO Device containing the desired IP address. If this was successful, the device sends back a *DCP Set response* to the controller. Another possibility is to set the IP address via DHCP.



(a) Name assignment (b) IP assignment

Fig. 3. Name and IP assignment process in Profinet IO using DCP protocol

3 The Attack

The attack presented here is a combination of four consecutive steps (see Figure 4). In the first step a preliminary exploration of the topology is required. Thereafter, a port stealing attack is launched to interrupt existing Application Relationships between the IO Controller and all IO Devices. Subsequently, a reconfiguration attack is triggered by means of a *DCP Set request*. To get the system up and running again the operator would need to reinstate the old system configuration. However, the fourth attack step prevents this by exploiting the DCP protocol behavior (see Figure 5). As the result, the affected IO Devices stop their operation. In the following we consider the attack steps more in detail.

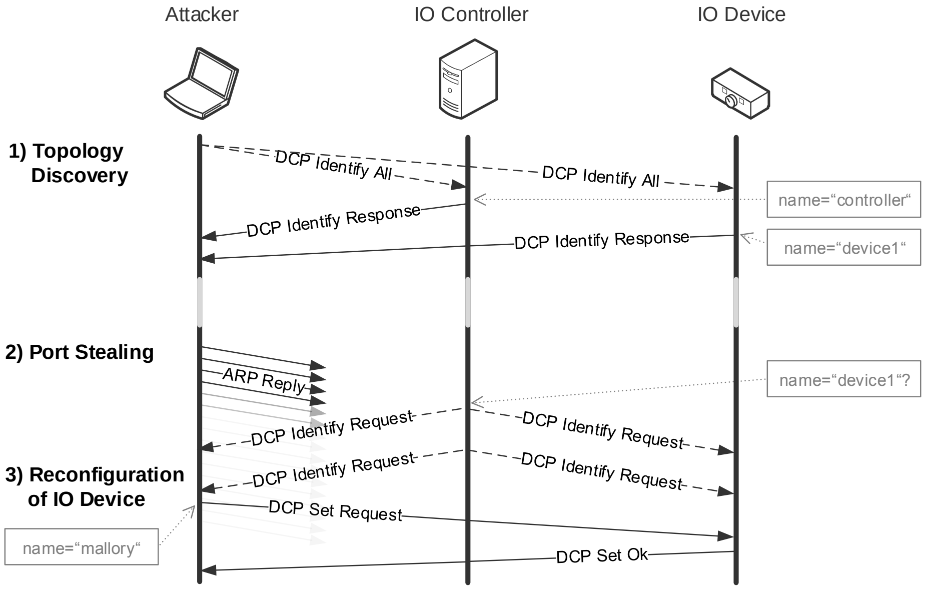


Fig. 4. Attack steps

3.1 Step 1: Topology Discovery

In order to successfully implement an attack, the attacker needs comprehensive knowledge about the components to be targeted. One of the design goals of Profinet IO was a simple configuration and engineering of the hardware. For this reason, each Profinet IO device must support the Link Layer Discovery Protocol (LLDP), Simple Network Management Protocol (SNMP) and the already introduced DCP protocol to provide functions for automatic addressing, acquisition of topology information, and network diagnostics. In the context of the described attack, DCP is used to obtain the required information in our attack. Therefore, in this preliminary step, the attacker sends a *DCP Identify request* via multicast to the network. Every Profinet-enabled device return its identifying parameters, such as name, network configuration, vendor, and model (see Table 1).

Table 1. An exemplary selection of information captured with topology exploration

	Device 1	Device 2
MAC	ac:64:17:01:05:09	ac:64:17:20:07:16
Name	cpu1500	et200sp
Device role	IO Controller	IO Device
Vendor	Siemens AG	Siemens AG
Model	S7-1500	ET200SP
IP Address	192.168.1.1	192.168.1.14

3.2 Step 2: Port Stealing

Once all devices in the subnet and their roles have been identified, the goal of the first attack step is to interrupt the Application Relationship (AR) between the IO Controller and the IO Devices. An efficient way to achieve this is a Denial-of-Service (DoS) attack based on port stealing. This method is well known and widely used to perform a Man-in-the-Middle (MitM) attack in traditional switched networks. Network switches manage the binding of a MAC address to a connected switch port in a forwarding table. If the MAC address at a port changes because a new device has been connected, the address in the forwarding table is updated and the old entry is removed. Port stealing exactly exploits this functionality. An attacker floods the switch with forged gratuitous *ARP replies* with the source MAC address of the target host and destination MAC address of the attacker. The switch assumes that the target host is now using the other switch port and forwards the packets to the new port. Since the target host continues to send packets during this time, the switch constantly changes the binding of the port to the MAC address back and forth. This effect no longer occurs when the attacker sends packets at a much higher frequency. If the packets of two communicating devices are redirected through port stealing to the attacker, the attacker only has to forward the packets accordingly (or manipulate them beforehand) to carry out a complete MitM attack. For this scenario, we only steal the port of the IO Controller to terminate the AR with all related IO Devices. As a result, the IO Controller starts multicasting *DCP Identify requests* for the currently not reachable IO Devices.

3.3 Step 3: Reconfiguration of the IO Device

The establishment of the AR is primarily based on the Profinet IO name. If it matches, the vendor and device ID of the IO Device is compared with the configured state in the next step [16]. While the DoS attack based on port stealing from the previous step is still active and therefore no active AR exists between the peers, a *DCP Set Name request* is send by the attacker to the IO Device which contains an arbitrary name. The IO Device answers with a *DCP Set Ok response*. Thereafter, the IO Controller periodically sends out *DCP Identify Name requests*, but no device reacts due to the wrong name. As consequence, no AR can be set up any more and the DoS attack can be stopped. The only way to get the hardware functional again is to restore the correct name. However, to do so the operator needs to detect and diagnose the cause of the problem. At worst, a corresponding attack on a power plant could halt the electricity production for hours and thus, destabilize the overall power supply. In real-life production environments, for instance, an insidious attack could be as follows: Based on the heuristics that typically multiple identical devices are installed within a automation system, an attacker swaps the names of two devices. It is conceivable that the connection can be re-established because the vendor ID and device ID will match in this case. Since the devices were probably configured differently, this will either lead to an unpredictable behavior of the automation process or errors will occur that are difficult to detect for the operator.

3.4 Step 4: Preventing Re-establishment of the Application Relationship

After successful completion of the attack, the Profinet IO system is in the setup state. The operator can now use the engineering software to reset the name or engineer the system anew. In order to be able to change the name, a *DCP Identify Name request* is send out to check whether the name already exists, as described in Section 2.3, since only unique names are allowed in the network. After setting the name, an *ARP request* is sent analogously to check whether the IP address already exists in the network. If the attacker now responds to every *DCP Identify request* with the corresponding response which contains the requested name and to every *ARP request* with a corresponding *ARP response* indicating that the IP address has already been assigned, the operator has no more possibility to reset the automation system as long as the attacker is in the network or has control over a malicious device in the network. This idea was presented conceptually in Paul et al. [14] and implemented in this work.

3.5 Implementation of the Attack

We have implemented the attack using the popular tool *Scapy* [7], which allows parsing and crafting of network packets with little effort. The bases for this are so-called layers that implement different protocols. We have written such a layer for Profinet DCP, which will soon be made available to the community. It should

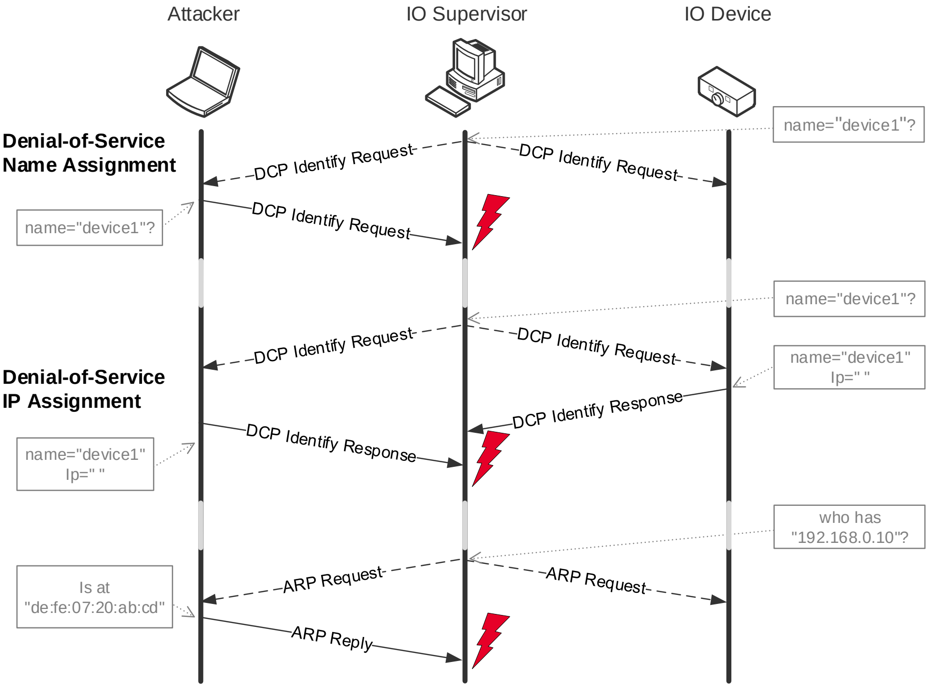


Fig. 5. Prevent re-establishment of the Application Relationship by exploiting DCP protocol behavior

be mentioned that *Scapy* can only send 24 packets per second by default on our attacker hardware which is much too little for a ProfinetIO send cycle of 1 ms. To increase performance *Scapy* offers the possibility to reuse the L2 socket. Using this option the transmission frequency increased significantly to approximately 5000 packets per second. With the resulting send cycle of 0.2 ms, the port stealing attack could be carried out successfully. Analysis and debugging were performed on a separate PC with Wireshark on a mirror port of the network switch.

4 Evaluation

In order to assess the general applicability of the attack, we performed the related steps systematically on several realistic hardware setups. These configurations have been chosen so that both star and line topologies were covered to determine their influence on the applicability of the attack. Moreover, we have also examined the effects on the attack's success if the operator configures the switch to block DCP packets. In the following, the available configurations are elaborated and, subsequently, the results of the evaluation are presented and discussed.

4.1 Setup

It is assumed that the attacker has gained direct access to the network or that a component within the network is already under her control. The attacker has no knowledge about the automation process, and the motivation is to limit the availability of the automation system as long as possible. In the testbed, hardware of different vendors is present (see Table 2). The equipment from Siemens is frequently used in production and power plants, whereas the devices from Pepperl+Fuchs and ifm are usually installed in public trains. All devices were equipped with the latest firmware version. The network switches from Siemens are designed for industrial applications, and the switch from D-Link is commercially available off-the-shelf (COTS) equipment. The Scalance XC208 switch offers comprehensive control capabilities. One of these functions is the possibility to stop forwarding DCP packets. As it will be described later, we have also examined the feasibility of the attack under these conditions.

Table 2. Hardware used for the evaluation

Device Role	Vendor	Model	Firmware
IO Controller	Siemens	CPU1516-3 PN/DP	V 2.6.0
		CPU315-2 PN/DP	V 2.6.12
IO Device	Siemens	ET200SP	V 4.2.0
		ET200S	V 2.0
	Pepperl+Fuchs	ICE1-8IOL-G60L-V1D	-
	ifm	AL1301	V 2.2.18
Switch	Siemens	Scalance XC208	V 4.1
		Scalance X108	-
	D-Link	DGS-1100-08	1.10.011

All setups are configured with the common software TIA Portal V 15.0 from Siemens. Also, the project configuration of one setup (IO Controller: CPU315; IO Device: ET200S) was performed with Siemens STEP7 5.6 configuration software as well to determine if this affects the findings. Since this is not the case, it is not mentioned in the further consideration. In industrial setups, both star and line topologies are common. In the setups with star topology the IO Controller, IO Device, and the attacker were connected to the same network switch. Since the results were the same in all cases regardless of the switch used, no differentiated consideration of the different switches is made here. All devices except the ET200S are equipped with at least two network ports with an integrated switch functionality. The line topology is achieved by directly connecting the devices. The attacker was placed at the free switch port of the IO Controller as well as at the IO Device.

4.2 Results

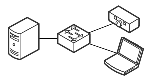
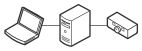
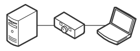
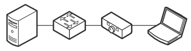
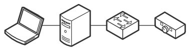
After describing our findings of the attack steps explained in Section 3 we go into more detail regarding the influence of blocking DCP packets on the attack.

Topology Discovery. In the first step, a preliminary exploration of the topology is performed. As expected, the detection of all devices in the network was successful regardless of the underlying topology.

Port Stealing. The port stealing attack aims to interrupt the existing AR. We have examined on which topologies this attack is successful. An overview of the results is given in Table 3. In a simple star topology, in which all devices are connected to the same switch, the attack was successful regardless of the switch used. This was obvious as this attack exploits the core functionality of switches. With a line topology, the attack fails if the attacker is connected to the free port of the IO device. However, if the attacker is on the free port of the IO Controller, the port can be successfully stolen if the attacker uses the MAC address of the IO device as the target instead of the MAC address of the IO Controller like described in Section 3.2. In mixed topologies, where both star and line topology are used, this attack step is successful in any case.

Reconfiguration Attack. The reconfiguration attack changes the name of the IO Device to prevent the re-establishment of the AR. We have evaluated to what extent the attack is successful both individually and in sequence with the port stealing attack. Table 4 contains the results of this attack step. One checkmark indicates that the attack is successful as long as it is active. This means, as soon as the attack is stopped, the AR is re-established and the devices continue to communicate like before. Two checkmarks indicate that the system requires operator intervention to continue working after the attack. After the attack the AR is broken and cannot be re-established, and therefore no further communication is possible. A cross symbol means that the attack was not successful. We have configured the different available IO Devices with the CPU1516 as IO Controller. Except for the ET200S, we were able to change the name of all IO Devices regardless of the topology. For investigating the fact why it was not possible to change the name of this device type, a further setup was created using the ET200S with the CPU315, which were both configured with TIA Portal and STEP7. In all configurations the result was the same, i. e., changing the device name failed. In the following Section 4.3 we discuss the reasons for the different behavior of the devices. However, in sequence with the port stealing attack, it led to success in a star topology as well as in line topology when the attacker

Table 3. Results of the port stealing attack in different topologies

Topology	Configuration	Successful
Star		✓
Line		✓
		x
Star + Line		✓
		✓

= IO Device = IO Controller
= Attacker = Switch

is connected to the free port of the IO Controller (see Table 3). Since both the CPU315 and ET200S only have a single network port, an attack in line topology configuration was not applicable.

Re-establishment Prevention of the AR. If the previous steps were successful, the AR is broken and must be repaired by restoring the originally configured name. The last step of the presented attack prevents these efforts as described in Section 3.4. If the operator attempts to restore the old name (or tries to assign a new one) with the help of the engineering software, he receives an error message claiming that the name cannot be assigned to this MAC address because it is supposed to already belong to another MAC address (see Figure 6). Since it is necessary to check whether the name and the IP have already been assigned in the startup phase (see Section 2.2), even devices that have not been attacked will not be able to establish an AR as long as this attack is active.

Blocking the Forwarding of DCP Packets. As mentioned before, the Scalance XC208 switch can be configured to block the forwarding of DCP packets. One may assume that the operator may simply activate this option in order to prevent the presented attack. Hence, we will evaluate the attack under these new conditions. The topology discovery failed, so the attacker needs to obtain the necessary device information in another way, e. g., by the use of SNMP or LLDP. The MAC address is also printed on the front of the devices so that the

Table 4. Results of the attacks to terminate the AR

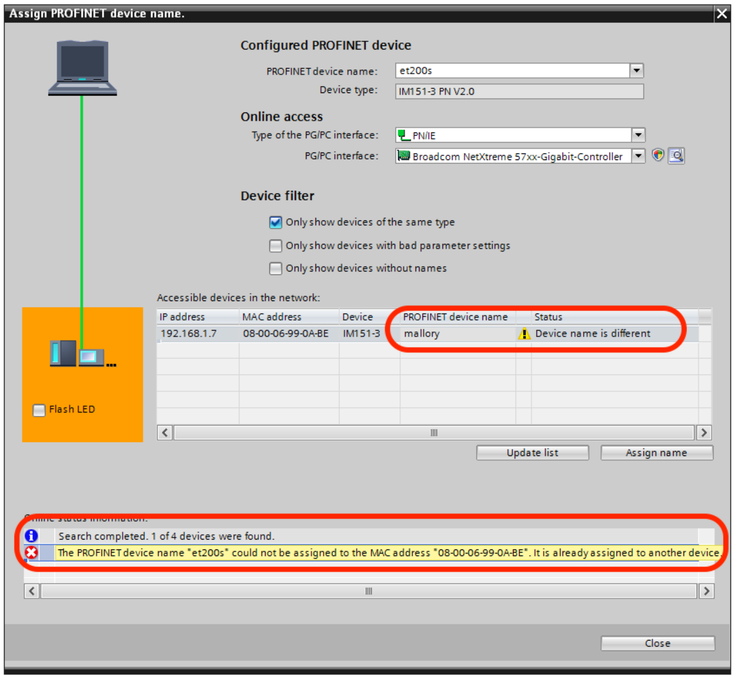
Topology	Controller	Device	Port Stealing (PS)	Reconfiguration (R)	Sequence PS + R
Star	CPU1516	ET200SP	✓	✓✓	✓✓
		ET200S	✓	x	✓✓
		Pepperl+Fuchs	✓	✓✓	✓✓
		ifm	✓	✓✓	✓✓
	CPU315	ET200S	✓	x	✓✓
Line	CPU1516	ET200SP	(✓)	✓✓	✓✓
		ET200S	(✓)	x	(✓✓)
		Pepperl+Fuchs	(✓)	✓✓	✓✓
		ifm	(✓)	✓✓	✓✓
	CPU315	ET200S	NA	NA	NA

x = not successful
✓ = successful; AR will be restored after attack
✓✓ = successful; AR permanently broken, needs to be repaired
(. . .) = attacker is connected to the CPU directly
NA = not applicable

attacker could read it directly on site. Once the attacker has determined the MAC address of one device, port stealing can be launched. As this attack is not based on DCP, it works as expected. Unlike in the previous setup, the AR does not re-establish automatically after stopping the attack, since DCP is necessary for establishing the connection. Also the reconfiguration attack could be carried out successfully with the known limitations from the previous setup because *DCP Set requests* are not blocked, contrary to our assumption. The last step of the attack that prevents the re-establishment of the AR partially succeeded. *DCP Identify requests* are blocked but the DoS for IP assignment is still possible because it is based on the ARP protocol. To sum up, if the attacker can obtain the MAC address of a single Profinet IO device in the network, the attack is applicable even if the forwarding of DCP packets is blocked.

4.3 Discussion

Our evaluation demonstrates the general practical feasibility of our attack in realistic environments. The fact that one of the devices deviates from the protocol behavior compared to the others is of particular interest. The relevant standard



Assign PROFINET device name.

Configured PROFINET device

PROFINET device name: et200s

Device type: IM151-3 PN V2.0

Online access

Type of the PG/PC interface: PN/IE

PG/PC interface: Broadcom NetXtreme 57xx-Gigabit-Controller

Device filter

☑ Only show devices of the same type

☐ Only show devices with bad parameter settings

☐ Only show devices without names

Accessible devices in the network:

IP address	MAC address	Device	PROFINET device name	Status
192.168.1.7	08-00-06-99-0A-BE	IM151-3	mallory	⚠ Device name is different

☐ Flash LED

Search completed. 1 of 4 devices were found.

The PROFINET device name "et200s" could not be assigned to the MAC address "08-00-06-99-0A-BE". It is already assigned to another device.

Fig. 6. Error message when trying to reset the name in TIA Portal V15.0

for this issue is IEC 61158-6 [2]. However, it does not explicitly specify the behavior for the devices when the AR is already established and unexpected requests are received, as in the case of the *DCP Set request*. When a valid *DCP Set request* is received, all ARs should be aborted if the device is in state *"W_Connect"*, which stands for "Wait for the AR establishment" (see IEC 61158-6-10:2014 p. 587 #38 [2]). Evidently, this is also applied by the vendors to the existing ARs.

As identified in the previous section, the attack cannot be prevented by the blocking of DCP packet forwarding. However, several security concepts in the literature may be used to detect and prevent this attack. In the approach of Paul et al. [14] an intrusion detection system (IDS) is suggested using anomaly detection. The IDS is connected to a switch mirror port in order to monitor all network packets from this switch. In a training phase the normal system behavior is learned and in the following detection phase deviations can be identified as anomalies. Although port stealing would be identified as such an anomaly, the detection of the reconfiguration attack depends on the traffic learned. The DoS attack results in characteristic patterns that are detected by this system. For example, while performing a DoS on IP assignment, there are two *DCP Identify responses* to every *DCP Identify request*, one from the IO Device that is to be configured and one from the attacker.

Pfrang et al. [15] introduce a signature-based IDS using Snort [20]. Rules describe unwanted communication and trigger an alarm if these packet sequences are detected. The network traffic is forwarded to a mirror port of the switch to perform the analysis. The authors present two such rules that will detect both port stealing and the reconfiguration attack. If the MAC address changes from one interface to another and a Profinet alarm frame is sent due to the loss of connection, this is detected as port stealing. The trigger for the reconfiguration attack is the typical packet sequence (*DCP Set request*, *DCP Set Ok* and a Profinet alarm frame followed by several *DCP Identify requests*). To detect the DoS attack, further rules would be necessary. Both solutions presented are designed so that an attack is detected after it was carried out successfully. An alarm event is triggered, but the IDS does not actively intervene in the network. Especially in industrial networks, this is best practice as a false positive event could disturb the automation network. Others [13] suggest the encryption of real-time traffic. Since the attack is not based on real-time packets, it should work in this case, nevertheless.

5 Related Work

A broad overview of possible vulnerabilities in ICS networks is given in [11]. This includes the hardware and firmware of the PLCs, the software e.g., for engineering, the network part, and the ICS process itself. A systematic ICS vulnerability assessment approach and the use of testbeds are suggested to identify the existing vulnerabilities in a given environment. The authors identified attacks on PLCs and sensors as current threats in ICS networks besides the traditional vulnerabilities that are known from ICT. As emerging threats, the injection of false data, as well as the construction of payload to influence the behavior of the system, is considered.

One of the first papers that addressed the security of Profinet IO is that of Baud and Felser [6]. Based on possible errors in the establishment of the AR, different attack possibilities like duplicate name or IP assignment and the mix-up of MAC addresses are introduced. Moreover, the threat of MitM attacks in Profinet IO networks was presented. The implementation was done with the open source tool *ettercap*, but it was not successful because the minimum cycle time of 1 s was not sufficient for the attack. Akerberg [4] also performed a MitM attack both in a shared medium with a network hub and in a switched network. In the first case, the attacker waits for cyclic frames from the IO Controller and then sends altered frames with a correct frame cycle counter at the right time to the IO Device just before the correct frame arrives. The timing is essential in This scenario. In the second scenario, a network switch is used instead of a hub. To get knowledge about the topology a *DCP Identify All* request frame is sent. After that port stealing is performed using forged ARP packets. This attack is easier to perform because the attacker synchronizes it with the peers every cycle. There is no direct communication between them anymore. Furthermore, a security module as a software layer on top of Profinet IO is presented and discussed in detail in [5] which is intended to ensure the end-to-end authentication, integrity, and/or confidentiality of cyclic Profinet process data, but there is no protection against layer two DoS attacks.

The paper of [14] describes further attacks on Profinet IO on a conceptual level. In addition to the already introduced MitM attack during the operation stage, the possibility of such an attack during the set up phase is also described. As soon as the IO Controller requests the status of the existing IO devices in the network via *DCP Identify All request*, the port of the IO Controller is stolen by the attacker via the port stealing technique, so that the *DCP Identify response* can no longer reach the controller. Instead, the attacker sends a spoofed *DCP Identify response* that indicates that the IP address is already assigned. Now the attacker is able to set up an AR with the IO Controller and send faked input data. The possibility of DoS attacks in the setup phase is also described in this paper. Before a name can be assigned to an IO Device the IO Supervisor multicasts a *DCP Identify request* to check, whether the name has already been assigned to another device. An attacker simply responds to these requests with a spoofed *DCP Identify response* containing the requested name. Since the name has to be unique, the assignment cannot be finished in this case. Analogously to the previous attack, a fake *DCP Identify response* can also be used to disrupt the IP assignment process. The IO Controller makes sure that the device name is assigned before the assignment of the IP. If multiple responses arrive for the requested name it cannot be assumed anymore that this name is unique. Furthermore, it is possible to disturb the IP assignment using manipulated *ARP replies* for pretending that another device already has this address. Note that the presented attacks can only be exploited in the set up process and have no influence on the availability of the automation system during operation. Our last attack step is the implementation of Paul's approach [14]. To prevent such attacks the authors propose an intrusion detection system using anomaly detection.

In [15] a signature-based IDS for industrial control systems is presented. As motivation for this IDS, an attack case study is introduced for which an attacker is assumed that aims to disturb a stepping motor and with no knowledge about the industrial process. To achieve this goal the focus is on replaying sniffed network packets. Regarding the authors, two steps are necessary for that. In the first step, the attacker needs to take over the control of the motor using two kinds of replay attacks: port stealing or a reconfiguration attack with *DCP Set*. Since the AR between the PLC and the motor will be terminated, a new AR must be established between the attacker and the motor before the traffic can be replayed. According to our understanding, the attack scenario is not feasible as described here. Beside the fact, that the described attacks are not replay attacks but DoS attacks, it is not clear how the attacker manages to capture the packets between the motor and the PLC to replay the traffic in order to steer the motor in the second step. Since the Profinet RT communication is unicast, a MitM attack is needed instead. In our experimental setup, we reproduced both attacks from the scenario. After successfully stealing the port the attacker receives a few RT packets from the IO device before the AR terminates. To steer the motor the attacker needs RT packets from the PLC. The feasibility of the reconfiguration attack depends on the IO Device used (see Section 4 for further details). If it is successful, the AR is terminated immediately. Since RT packets are unicast, the attacker has to use a MitM attack. Port stealing can be used for this when it is performed as described in [4] or [14]. In our work we propose to perform port stealing in sequence with a reconfiguration attack. Thus, we make the attack universally applicable.

Mo [12] is investigating replay attacks in ICS. The attacker injects packets which are previously recorded to disrupt the operation of the control system while being undetected. The authors define the formal conditions for the feasibility of such attacks. The paper of Hui [9] examines possible attacks on the Siemens communication protocol S7Comm or in the newer version S7CommPlus. Apart from the S7Comm-specific attacks, the possibility of generating phantom PLCs with Profinet DCP is also described. The attacker responds to a *DCP Identify All request* with one or more fictional PLCs. This attack has no impact on the automation process, but leads the human user to confusion and opens the possibility for misconfiguration. Furthermore, it can be combined with other attacks.

6 Conclusion and Future Work

Due to the increasing use of Ethernet in automation technology, the number of IT security threats is also rising significantly. The reasons for this are, on the one hand, the easier access for a potential attacker compared to the previously established proprietary solutions and, on the other hand, the already known vulnerabilities of Ethernet which also be exploited here. In this work, we have presented an attack on the most common industrial communication protocol in Europe - Profinet IO - consisting of four steps. The attack has a severe impact on the affected automation system because it blocks the production process. It interrups the communication relationship between the IO controller and the IO device connected over Ethernet. The method of port stealing known from ICT is combined with spoofed Profinet DCP packets. As a result, the automation process is interrupted and cannot be reactivated as long as the attacker or the device controlled by the attacker is active in the network. Neither restarting the affected hardware nor re-engineering the automation process can solve the problem for the operator in this case. Since Profinet IO is used in applications, such as public train service, production or power generation, this may cause considerable financial damage for the operator, as the train does not run during this time or no products or electricity can be produced. The practicability was evaluated with realistic hardware setups and network topologies that are used in the field of public trains and the production or power plants. The results of our comprehensive evaluation showed that the attack can be successfully carried out in almost all setups. Only in a line topology, it was not possible to interrupt the communication of one of the devices, when the attacker is connected to network port of the IO Device.

All attack steps were implemented in Python. In parallel, we are working on a comprehensive framework for the implementation of network-enabled attacks on industrial networks. This framework will serve as a basis for the evaluation of further research in the field of security in industrial environments. In addition to Profinet IO, other protocols, like Modbus TCP, EtherNet/IP, and S7Comm will also be supported.

The hardware resources of the PLCs have been designed specifically for the defined requirements of the automation process. In our opinion, intelligent network management is necessary to not only detect but also prevent attacks like those presented here. Software-defined networking seems to us to be a promising concept to implement very fine-grained firewalls, which can act on a switch-port level and thus allow preventing these attacks. The SDN controller keeps a record of the existing devices and evaluates the statistics generated by the SDN switches. We are currently investigating such an approach.

References

1. Search engine Shodan, https://www.shodan.io. Last accessed 05 Feb 2019
2. IEC 61158-6-10:2014: Industrial communication networks - Fieldbus specifications - Part 6-10: Application layer protocol specification - Type 10 elements (10 2014)
3. IEC 61784-2:2014: Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3 (10 2014)
4. Akerberg, J., Bjorkman, M.: Exploring security in profinet io. In: 2009 33rd Annual IEEE International Computer Software and Applications Conference. vol. 1, pp. 406–412 (July 2009). https://doi.org/10.1109/COMPSAC.2009.61
5. Akerberg, J., Bjorkman, M.: Introducing security modules in profinet io (2009). https://doi.org/10.1109/ETFA.2009.5347205
6. Baud, M., Felser, M.: Profinet io-device emulator based on the man-in-the-middle attack. In: 2006 IEEE Conference on Emerging Technologies and Factory Automation. pp. 437–440 (Sep 2006). https://doi.org/10.1109/ETFA.2006.355228
7. Biondi, P.: Packet crafting for python2 and python3 (2018)
8. Dias, A.L., Sestito, G.S., Turcato, A.C., Brandao, D.: Panorama, challenges and opportunities in PROFINET protocol research. In: 2018 13th IEEE International Conference on Industry Applications (INDUSCON). IEEE (nov 2018). https://doi.org/10.1109/induscon.2018.8627173
9. Hui, H., McLaughlin, K.: Investigating current plc security issues regarding siemens s7 communications and tia portal. In: 5th International Symposium for ICS & SCADA Cyber Security Research 2018: Proceedings. pp. 67–73. BCS (8 2018). https://doi.org/10.14236/ewic/ICS2018.8
10. Liang, G., Weller, S.R., Zhao, J., Luo, F., Dong, Z.Y.: The 2015 ukraine blackout: Implications for false data injection attacks. IEEE Transactions on Power Systems **32**(4), 3317–3318 (July 2017). https://doi.org/10.1109/TPWRS.2016.2631891

11. McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A., Maniatakos, M., Karri, R.: The cybersecurity landscape in industrial control systems. Proceedings of the IEEE **104**(5), 1039–1057 (May 2016). https://doi.org/10.1109/JPROC.2015.2512235
12. Mo, Y., Sinopoli, B.: Secure control against replay attacks. In: 2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton). pp. 911–918 (Sep 2009). https://doi.org/10.1109/ALLERTON.2009.5394956
13. Mller, T., Doran, H.D.: Profinet real-time protection layer: Performance analysis of cryptographic and protocol processing overhead. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 258–265 (Sep 2018). https://doi.org/10.1109/ETFA.2018.8502670
14. Paul, A., Schuster, F., König, H.: Towards the protection of industrial control systems – conclusions of a vulnerability analysis of profinet io. In: Rieck, K., Stewin, P., Seifert, J.P. (eds.) Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 160–176. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
15. Pfrang, S., Meier, D.: Detecting and preventing replay attacks in industrial automation networks operated with profinet io. Journal of Computer Virology and Hacking Techniques **14**(4), 253–268 (Nov 2018). https://doi.org/10.1007/s11416-018-0315-0, https://doi.org/10.1007/s11416-018-0315-0
16. Pigan, R., Metter, M.: Automating with PROFINET: Industrial Communication Based on Industrial Ethernet. Wiley (2015)
17. Popp, M.: Industrial Communication with PROFINET. PROFIBUS Nutzerorganisation (2014)
18. Poresky, C., Andreades, C., Kendrick, J., Peterson, P.: Cyber security in nuclear power plants: Insights for advanced nuclear technologies. Department of Nuclear Engineering, University of California, Berkeley, Publication UCBTH-17-004 (2017)
19. PROFIBUS & PROFINET International: PROFINET Security Guideline (Nov 2013), https://www.profibus.com/download/profinet-security-guideline
20. Roesch, M.: Snort - lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX Conference on System Administration. pp. 229–238. LISA '99, USENIX Association, Berkeley, CA, USA (1999), http://dl.acm.org/citation.cfm?id=1039834.1039864