Cybersecurity

RESEARCH

Open Access

Time sensitive networking security: issues of precision time protocol and its implementation

Davide Berardi[1*], Nils O. Tippenhauer[2], Andrea Melis[1], Marco Prandini[1] and Franco Callegati[1]

Abstract

Time Sensitive Networking (TSN) will be an integral component of industrial networking. Time synchronization in TSN is provided by the IEEE-1588, Precision Time Protocol (PTP) protocol. The standard, dating back to 2008, marginally addresses security aspects, notably not encompassing the frames designed for management purposes (Type Length Values or TLVs). In this work we show that the TLVs can be abused by an attacker to reconfigure, manipulate, or shut down time synchronization. The effects of such an attack can be serious, ranging from interruption of operations to actual unintended behavior of industrial devices, possibly resulting in physical damages or even harm to operators. The paper analyzes the root causes of this vulnerability, and provides concrete examples of attacks leveraging it to de-synchronize the clocks, showing that they can succeed with limited resources, realistically available to a malicious actor.

Keywords Time synchronization, Time sensitive networking, Precision time protocol, Cybersecurity attacks

Introduction

The manufacturing industry is more and more relying on networking to empower machinery and processes. The rise of connected systems in the manufacturing Operational Technologies (OT) is the basis of the Industry 4.0, the industrial revolution of the XXI century. This evolution poses new challenges, because connecting OT networks to the Information Technology (IT) networks and also to the Internet opens the floor to cyber-security threats to the manufacturing environment. In the last few years, many examples of this have made the headlines of the newspapers (Hemsley et al. 2018; Miller et al. 2021), emphasising it is a key issue to be addressed. At the same time OT networks strongly rely on low latency and real time communications, therefore Time Sensitive Networking (TSN) is a key topic for industrial networks (Lo Bello and Steiner 2019; Fedullo et al. 2022; Raveling 2022; Electrics 2017; Siemens 2022).

In this manuscript we address the combination of these issues considering security threats to the Precision Time Protocol (PTP) which is a basic building block in TSNs. Existing research work on the topic is mainly focused on attacks to the synchronization mechanism, even if they are somewhat limited to adversarial settings in which a powerful attacker can perform continuous traffic manipulation (Moussa et al. 2016). A subject that has been essentially neglected in security analyses is the usage of Type Length Value (TLV) frames, a feature defined in the standard to manage the overall synchronization infrastructure. In this manuscript, after briefly reviewing the role of TLVs, we will show that some of them can be misused to successfully alter the clock synchronization. In particular they can be abused by an attacker to reconfigure, manipulate, or shut down time synchronization. To the best of our knowledge, the security threats posed by such manipulation of TLVs have not been addressed

*Correspondence:
Davide Berardi
davide.berardi@unibo.it
[1] Alma Mater Studiorum - Università degli studi di Bologna, Bologna, Italy
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

in the literature so far, in spite of the fact that it requires very little bandwidth and may allow the attacker to gain useful advantages to break time-related applications, such as Public Key Infrastructure services.

The contributions of this manuscripts are as follows.

- Show that the PTP standard fails to define appropriate security requirements for TLVs. As a result, implementations can be expected to have security issues related to TLV authentication.
- Analyse the semantic and use of the various TLVs to identify whether they may be exploitable for attacking the PTP infrastructure, and classify them in a cyber-security related taxonomy, which is here presented in Appendix 8.
- Demonstrate and prve the effecrtiveness of different attacks that leverage this lack of security in TLVs, and are aimed at desynchronizing clocks and introducing clock drifts.
- Present and discuss possible contermeasures to this kind of attacks in terms of Network segmentation and communication encryption.

This work was accomplished exploiting a PTP testbed implemented using LinuxPTP (Cochran et al. 2015) and PTPD (Owczarek et al. 2015) that allows to reproduce the complex PTP setup in a virtual environment.

A summary of PTP and of its security

TSN and IEEE-1588 (PTP)

Every system which operates according to a specific time frame, used to coordinate remote actors, suffers from the clock drift phenomenon, due to the fact that different time sources will have different time progressions. Synchronizing the clocks and leaving them go independently will inevitably result in a clock difference after some time.

Specific synchronization protocols were designed to cope with this problem and guarantee synchronization between remote devices. The *Precision Time Protocol* (PTP), standardized as IEEE-1588 is one of the most significant examples. There are three major releases of PTP: IEEE-1588 2002, v1.0, which was the first version, currently deprecated; IEEE-1588 2008, v2.0, which is, at the time of this writing, the main version of the protocol; IEEE-1588 2019, v2.1, which is a new version, typically found in the industrial devices now in production.

To achieve high resolution and precision, a hardware device with packet-time-stamping capabilities can be used. It will process network communications using a high precision clock, without passing the packets through any further stack layer that can alter the effective time and reduce the determinism of the process.

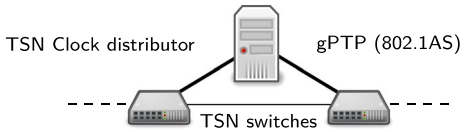


Fig. 1 A simple example of TSN network using the specialized PTP protocol (IEEE 1588), with a specialized profile called gPTP (802.1AS)

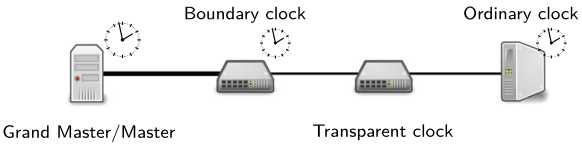


Fig. 2 PTP hierarchy. The clock symbol specifies which nodes are synchronized each other using PTP

Currently, TSN standard exploits a specific profile of PTP called Generalized Precision Time Protocol (gPTP). A device compliant with this profile must communicate with PTP using Ethernet packets; this constraint ensures low jitter and bounded delay between two synchronized nodes.

Figure 1 describes the protocol placement of IEEE-1588 in a TSN with two enabled switches. The switches' clocks use a single logical time source—pictured as the TSN clock distributor—which grants them synchronization references using gPTP.

More generally PTP exploits a hierarchical structure, as described in Fig. 2. Here the Grand Master is a logical node in charge of synchronizing the entire hierarchy.

While a least one Grand Master is always required, the presence of additional Master nodes is optional. Additional Master nodes are in charge of the distribution of the Grand Masters' clocks to the lower layer of the hierarchy.

The Ordinary clocks are leaf nodes with no responsibility on redistributing the clock to other part of the network. On the other hand, boundary clocks are bridge devices which interconnect parts of the network and keep their internal clock synchronized with the upper layers of the hierarchy.

The *transparent clock* shown in the figure is a passive bridge of the network which is limited to the transmission of network packets. This kind of 'clocks' do not take part in synchronization and do not have internal references.

To define the hierarchy and elect the Grand Master, an algorithm called Best Master Clock Selection (*BMC*) is employed. This algorithm uses parameters configurable in the PTP devices of the infrastructure which will take part to the selection.

PTP hierarchy management

Being a protocol with an hierarchy and involving elements with different roles management is an issue, which is address in the protocol by defining several sets of management messages called Type Length Value (TLV). They are the management messages used to configure the protocol settings. For instance, TLVs can be used to set or retrieve the name of the node set by the network administrator or get the PTP version implemented by the device.

Moreover TLVs can be used to set critical parts of the systems, such as the state of the port (e.g. by disabling it) or the position of the node in the hierarchy. A detailed list of TLVs usage can be found in the standards [19] and their security classification can be found in Appendix 8.

Related works on PTP Security

The PTP protocol is not completely devoid of security countermeasures. However, there are examples in the literature showing realistic adversarial models that allow to overcome them. This has been extensively explored by several works: starting from Ullmann (Ullmann and Vögeler 2009) who explores this protocol compared to Network Time Protocol (NTP) and culminating in recent works on the security of the PTPv2.1 standard (Han and Crossley 2019; Shereen et al. 2019).

As these works state, the clocks alterations that can be introduced controlling the infrastructure will still be present in the last PTP standard. Several options that can improve PTP security could be found in works by Moussa et al. (2019) and Neyer et al. (2019). These papers address the field of time synchronization and encryption, a theme which will return in the recently emerging TSN infrastructures. As stated by previous works (Nasrallah et al. 2018), this standard do not specifies any security profile and, like the PTP protocol, relies on specialized standards such as MACSec (Mizrahi 2011).

From an high level perspective, the PTP protocol implements security mechanism which rely on security service provided by other protocols. For instance it can implement a security layer to sign packets with a Message Authentication Code (MAC),[1] or it can be integrated within a secure tunnel like IPSec. These solutions are far to be optimal for a clock distribution service as stated in Ullmann and Vögeler (2009), since by controlling the network infrastructure hosting the PTP communication, attackers can control the speed of PTP Protocol Data Units (PDUs). Therefore they can delay or accelerate the clock synchronization packets,[2] even acting on protocols at different layers. Exploiting this kind of delays will therefore control the time reference of the victims.

In Ullmann and Vögeler (2009) is also presented a list of vulnerabilities that affects PTP networks, that can be classified as:

Byzantine masters By spoofing a Master node, the dependant clocks can be induced to a time drift or sensible clock skews. The effectiveness of this depends on the system configuration, in particular on the max applicable correction to the clocks, ϵ. This values determines the magnitude of the clock applicable by the master, if it is large enough an attacker can declare an arbitrary clock offset to the clients, changing all the hierarchy's references.

Boundary clock alteration An interposed device (boundary or transparent clock) can alter the timing between the packets. This operation can be made at the data-link or at the network layer, using network routes or dropping packets at a certain rate. These manipulations can effectively change the heuristics used in the Best Master Clock (BMC) selection algorithm or the synchronization between the clocks. Encrypted connection do not mitigate the possibility of this attack (Annessi et al. 2018).

The works above, to the best of our knowledge, are mostly focused on attack scenarios on the synchronization part of the protocol, as well as the countermeasures that were proposed (Shereen et al. 2019).

Management frames of PTP

The PTP standard also includes an application level, built using specific messages (called protocol Data Units or PDUs in the following) strictly related to time synchronization. These PDUS can be grouped in PDUs used to manage and set-up the hierarchy; PDUs that can be used to configure the parameters of the PTP nodes. In more detail:

- *Sync*, *Follow_Up*, *Delay_Req* and *Delay_Resp*: PDUs used to synchronize the clocks in a configuration referenced by the standard as "Request-Response" (E2E);
- *Pdelay_Req*, *Pdelay_Resp* *Pdelay_Resp_Follow_Up*: synchronization messages used in Peer to Peer (P2P) configuration.
- *Announce*: messages used in the hierarchy selection algorithm (BMC).

[1] Standardized as an option in the Annex-K of the IEEE-1588 2008 document [IEEE Std 1588-2008 (2008)].

[2] While delaying packets is trivial, Ullmann et al. in Ullmann and Vögeler (2009) describe a method to accelerate packets by exploiting changes on the infrastructure itself by using faster routes.

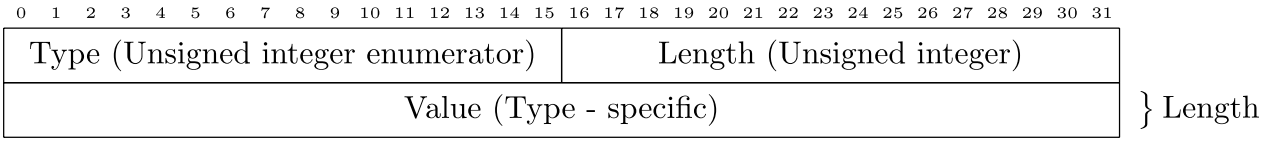


Fig. 3 TLV packet formats. The mapping between the type name and 16-bit unsigned integer value can be found in the IEEE standard. Type and Length are represented by two unsigned 16-bit integers. Value interpretation is dependent from the Type field and its length is based on the homonym field, interpreted as the number of octects

- *Signaling*: frames used by clocks to transport a sequence of management-related PDUs. This kind of messages are required by protocol extensions such as Unicast negotiation or the security part of the IEEE-1588 standard (Annex-K);
- *Management*: frames which use a format called Type Length Value (TLV) due to its packet structure (Fig. 3).

In this work we focus on the (in)security of the Management family of messages, demonstrating how they can be exploited and which risks can be lead. The TLVs are used to configure the protocol settings. For instance, they can be used to manage basic properties such as the name of the node set by the network administrator, or to get the PTP version implemented by a device, but also to set critical configuration values of the systems, such as the state of the port (e.g. by disabling it) or the position of the node in the hierarchy.

TLV work in a very straightforward way: they are basic fire-and-forget messages that, when received and accepted by the target, cause a change in its configuration, without the need to enact a complex, stateful protocol. Thus, the challenge to ensure proper management of the protocol is to guarantee that received TLVs are authentic and not malicious. A detailed list of TLVs usage can be found in the standards [IEEE Std 1588-2008 (2008)].

Up to PTPv2.0 no authentication for the TLV was recommended. In the latest version, PTPv2.1 (IEEE-1588-2019 (2020)), the TLVs are basically the same as in PTPv2.0 and two different ways to secure their use are introduced, called internal security and external security (Shereen et al. 2019), but TLVs authentication is still optional. Since it make the overall deployment more complex it is often disregarded in practice, because the focus is mostly placed on the the synchronization process. Furthermore, this remediation—as stated by the Annex P of the standard—is based on a centralized secrets manager (to manage a shared password), which introduces the problem of the management of the secrets, such as Insider Threats, Certificate life-flow management (release and revocation), etc.

Last, but not least, the security of the management part of the protocol is fully implementation-dependent; i.e. different implementation may have different vulnerability degrees (for instance, LinuxPTP implements stricter security methods than PTPd by exploiting UNIX sockets).

For this reason the attacks presented in this manuscript can be considered applicable both to PTPv2.0 and to PTPv2.1 as long as TLVs authentication is not implemented.

The approach to attack PTP

In this work we explore the PTP features that imply distributed communication and which are not directly connected to the synchronization. In this section we will present the main research question and an high-level description of the novel set of attacks derived from this investigation.

We can synthesize the main research question as: *Does the PTP standard define sufficient security features in its management application to ensure that standard-compliant implementations are immune against manipulation?*

Answering this question is not-trivial due to the complexity of the standard, and the need to check for vulnerabilities introduced in practical implementations. As outlined above we argue that TLVs have mostly been neglected when considering cybersecurity issues in PTP. This is understandable since in principle TLVs are not directly connected to the synchronization mechanisms and therefore may look innocuous. The purpose of TLVs is to enable massive, remote and automated management of a PTP network. While not usually required in small and closed contexts, these features are of paramount importance for the deployment of large scale implementations, as well as to more complex operations like topology discovery, etc (Arnold 2019). Thus, we assume that in the most common PTP deployment scenarios TLVs are kept enabled by default, and can be used for the aforementioned purposes. Furthermore, to the best of our knowledge, this common approach to keep TLVs as a default component of the deployment means also that they are not authenticated. We show in this work that, if they are not secured, they can easily be exploited creating tangible effects on the synchronization. Therefore

the answer to the research question is "no", unless proper attention is given to all the components of the protocol, included management messages.

Attacker model

Acting as described by the kill-chain model, the attacker will try to get information on the network topology and configuration; get an initial foothold in the network and alter parameters; then he/she will try to maintain the access and keep the network under control, cleaning up any trace of the attack. This model describes a very broad scenario, applicable to almost every attack; to narrow the goals of an attacker, we have identified two possible objectives.

Disruption of safety constraints Safety integrity levels (SIL) are requirements classes which specify threshold values to obtain context-specific safety. For instance, medical devices have specific parameters to ensure safety. This kind of requirements, can be disrupted if the time synchronization between the devices is altered. In general, devices implementing these requirements must have a procedure that handles failures. This kind of procedures could include the complete stop of the system.

Downgrade of endpoint confidentiality and authentication schemes Public key infrastructure (PKI) technologies are heavily dependent on the timing aspect of the various clocks. While the time requirements of these technologies is sensibly different from the application target of PTP, time de-synchronization can occur continuously to alter the wall-time clock of the endpoint. This will enable attacks on the validity of the certificate, enabling expired ones to be validated by the endpoint.

Other protocols rely on a more strict synchronization than the one used by PKI; a typical example is the Kerberos protocol. Since this protocol uses timestamps extensively, if the clock is not precisely synchronized between machines, it can be tricked into authenticating an entity using an old authenticator request.

Experimental set-up

The attacks described in the following were developed on an experimental set-up using two PTP enabled devices, interconnected by a single boundary device and using PTP version 2.1 (IEEE-1588-2019). While being a relatively new protocol, it shares most of his base with the previous version (IEEE-1588-2008), and the experiments are easily replicable using the old version.

PTP nodes can communicate using layer 4 (UDP) or layer 2 (Ethernet). In our experimental set-up it uses Ethernet at layer 2. This means that the attacker is connected to the LAN infrastructure to be attacked. Without lack of generality we assume that specific network segmentation strategies, such as VLAN, are not used or have been bypassed by the attacker.

Nonetheless the attacks proposed in this manuscript work in both cases, either with PTP over UDP and with PTP over Ethernet, because they rely on the application level logics of the PTP protocol and not on any specific vulnerability of the protocols used to transport the PTP information.

Attack complexity

To implement this kind of attacks, the attacker just needs to learn the network topology and send few management frames (as low as one frame for the Disable Port Attack). As a consequence, the attacker bandwidth requirements are close to zero, enabling this attack even from embedded network platforms such as low powered micro controllers.[3]

Compared to attack strategies such as, for instance (Ullmann and Vögeler 2009; Moussa et al. 2016), the family of attacks exploiting the TLVs obtain similar effects but use less bandwidth cost. As mentioned in the introduction, traditional attack to PTP synchronization require persistence in the attack phase, either based on denial of service for bandwidth exhaustion or on network addressing poisoning, that is not needed in the cases presented in this work.

Implementation

Given the preconditions illustrated so far, here we present the logics enabling two different ways to exploit the ensuing vulnerabilities, and describe the corresponding related TLVs turning them into practical attacks. We illustrate the PTP testbed we implemented, demonstrate the effectiveness of the proposed attacks and finally compare their efficiency with respect to prior work.

Exploiting TLVs in PTP implementations

The TLVs of PTP use three different verbs, `GET`, `SET` and `COMMAND`. While the first two are used to change or retrieve a value of the running daemon, through the third one it is possible to issue configuration commands (e.g. enabling or disabling a network port using the `DISABLE_PORT` command TLV). Our proposed attacks target those TLVs, as they are most relevant for the security of the network.

The three different attacks can be summarized as follows:

[3] TLV size is, on average, under a few hundreds of bytes. For instance, `DISABLE_PORT` TLVs have 44 bytes of PDU header, 4 Bytes of Type and Length fields and an empty Value.

Table 1 Exploited TLVs, details on attacks possible with these TLVs are presented in Sect. 5

TLV	Security problem
DISABLE_PORT	Exploitation: Denial of service
DELAY_MECHANISM	Exploitation: Denial of service
CLOCK_DESCRIPTION	Reconnaissance: Topology map
USER_DESCRIPTION	Reconnaissance: Topology map
PRIORITY1	Exploitation: BMC tampering
PRIORITY2	Exploitation: BMC tampering
CLOCK_ACCURACY	Exploitation: BMC tampering

- *Reconnaissance Attacks* leverage commands such as CLOCK_DESCRIPTION and USER_DESCRIPTION to map the topology of the PTP-enabled network. The CLOCK_DESCRIPTION TLV can be the first reconnaissance tool available to the attacker, if not correctly limited to specific users (this safer default tactic is used by LinuxPTP) or not correctly confined within the network;
- *Clock Disable Port Attacks* disrupt synchronization between nodes using enable or disable network ports with the eponymous ENABLE_PORT and DISABLE_PORT commands;
- *Clock Accuracy Attacks* change the parameters used by the Best Master Clock Selection algorithm using PRIORITY related TLVs.

Regarding the latter attack, it is interesting to note that the relevance and the possible risks of a misuse of the relate TLVs is not a new issue, since it is also highlighted in the notes of the PTP Standard documents. For instance, an extract from the IEEE-1588-2008 document [IEEE Std 1588-2008 (2008)] specifies a note on the CLOCK_ACCURACY TLV:

> *The accuracy and the time in the grandmaster clock is normally determined by interacting with a primary or application-specific time source, e.g., GPS, by means outside the scope of this standard. If the time is set in the grandmaster by means of the TIME TLV, then the accuracy should also be set. Since the clockAccuracy attribute is considered in the operation of the BMC algorithm, the setting of the clockAccuracy attribute in any clock by means of this TLV can result in a change of grandmaster the next time the BMC algorithm is performed.*

Nonetheless this critical role of such TLVs is not directly associated with a security threat, in spite of the fact that the quoted TLV can be used to control the BCM algorithm outcome, resulting in election of different grandmaster and master clocks. The specific commands/TLVs exploited for attacking are summarized in Table 1. Before delving into the implementation details and describing the corresponding test results in Sect. 6, we need to provide some background on the purposedly developed tools and the experimental setup.

Virtualized testbed design and implementation

To practically support this work we designed and implemented a testbed which we used to test the attacks outlined above, but that can also be used to reproduce the attacks from prior works, such as byzantine master. We plan to release the testbed framework as open source for others to replicate our results and further elaborate on them.

The testbed leverages virtualization, allowing us to run actual PTP programs on guest machines connected by virtual networks. This approach results in a portable and self-contained specification of attacks, and allows to easily replicate results.

The architecture of our virtualized testbed can be seen in action in two of the performed tests, as shown in Figs. 5 and 7. The configuration relies on LinuxPTP, PTPD, and PTPv2 software time-stamping. The testbed can show the results of clock alteration on the target system, accelerating or slowing down the clock difference between nodes. We configured the PTP hosts to enable configuration via TLVs, to replicate a setting in which TLVs are in general used for remote PTP management. Initially, TLVs are correctly configured (e.g., priority1 and priority2 in LinuxPTP), but can be reconfigured remotely.

The testbed is deployed using Infrastructure-as-Code (IaC) tools, namely Vagrant[4] and Ansible[5] to automate the tests. All the nodes of the infrastructures run the same operating system: Ubuntu-18.04.

The nodes are implemented as VirtualBox guests equipped with E1000 GigE virtual NICs. In our setup, the system was tested using an Intel® Core^{TM} i7-8700 CPU, with a nominal clock frequency of 3.20GHz. The Hypervisor does not offer support for PTP or TSN-enabled virtual NICs. To overcome this problem we configured software time stamping in the PTP daemons. Therefore, we connected the virtual machines with the default hypervisor network support (NatNetwork, in the terminology of VirtualBox). We implemented PTP synchronization in virtual testbed with software timestamps

[4] A tool to build portable development environment in form of virtual machines or containers: https://vagrantup.com.

[5] A configuration toolkit to build and configure environments using SSH: https://ansible.com.

using UDP, therefore it is completely transparent to layer 2 topology and protocols employed.

The workflow we enacted using the virtualized testbed is as follows:

1. we defined a topology using configuration files, in this case a topology with two legitimate entities, a network switch and an attacker;
2. we chose the nodes on which to instantiate a PTP clock, in this case a PTPd daemon in the two servers, which became Ordinary Clocks,
3. the other entities were left as passive elements and the switch became a Transparent Clock and not a Boundary Clock;
4. the attacker started issuing the TLVs to execute the attack;
5. the various nodes were observed to highlight clocks behavior.

Software tools

To support the attack operations, different software tools were developed, composing a modular toolkit we named PTP Exploitation Framework (PEF).

The main modules of PEF are:

1. *TLV Forging Module*, the baseline to forge and send TLV frames;
2. *Reconnaissance Module*, used to investigate the topology and active functionality of the PTP network;
3. *Indirect Attack Module*, used to modify the PTP network behavior through indirect misconfiguration of nodes, i.e. a node (the attacker) will modify the behavior of a victim node by sending a specific TLV to a third node;
4. *Direct Attack Module*, used to modify the PTP network behavior by directly sending a target node specific TLVs from the attacker's node.

The PEF tools was implemented for GNU/Linux systems in Python 3 and interfacing it with C APIs and reference tools such as `ptp4l`.

Experimental results

In this section we discuss the implementation of the attacks outlined in Sect. 4, and demonstrate their effectiveness.

Reconnaissance

The *Reconnaissance attack* is implemented as an exploitation module of PEF. The goal is to analyse the network and identify the hosts that are running PTP daemons, leveraging TLV-based queries.

```
[root@haigha pef]# ./pef.py
 pef> interface enp0s25 type 2
 pef> recon
 Found Host:  BridgeNode#1
 clockType 32768
 physicalLayerProtocol IEEE802.3
 physicalAddress 6c:b3:11:1c:92:a9
 protocolAddress 3 6c:b3:11:1c:92:a9
 manufacturerId 00:00:00
 productDescription PEFLinuxPTP;2.0;1
 revisionData 2.0;2.0;2.0
 userDescription BridgeNode#1
 profileId 00:1b:19:00:01:00

 Found Host:  LeafNode#1
 clockType 32768
 physicalLayerProtocol IEEE802.3
 physicalAddress 6c:b3:11:1d:b9:23
 protocolAddress 3 6c:b3:11:1d:b9:23
 manufacturerId 00:00:00
 productDescription PEFLinuxPTP;2.0;1
 revisionData 2.0;2.0;2.0
 userDescription LeafNode#1
 profileId 00:1b:19:00:01:00
```

Fig. 4 PTP Exploitation Framework demonstration; the reconnaissance module displays the information got from the running PTP daemons

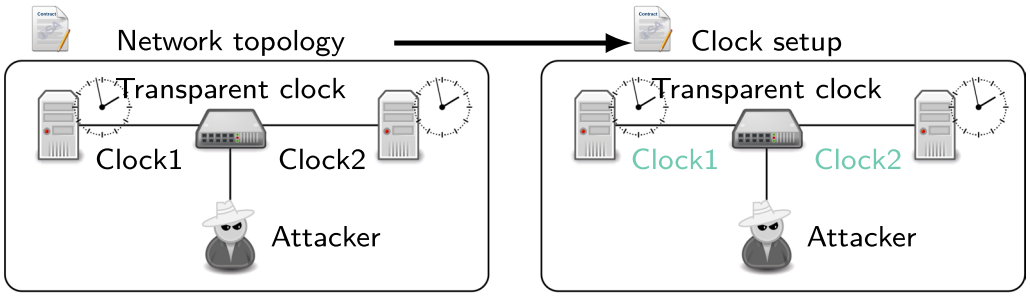
Implementation

The `CLOCK_DESCRIPTION` management TLV is used as follows:

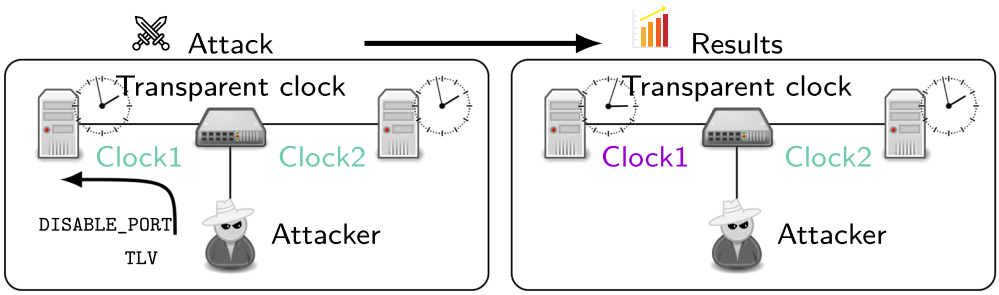
1. the TLV is sent to all network hosts (e.g. using a multicast or broadcast address) and will cause all TLV-capable hosts to reply to the scanning hosts;
2. from the replies a list of all PTP-capable hosts is compiled, providing meta-data for each host such as the name of the network clock, the specific PTP daemon used, its version number, BMC algorithm properties etc.

Results

An example of the results of this attack is shown in Fig. 4. Thanks to this analysis the list of running daemons is available. If a daemon is known to be vulnerable, it can be exploited, leading to the problems described in Sect. 4.



(a) Setup of the testbed. First, the topology is designed, then the clock are initialized.



(b) Attack usage of the testbed. First, the TLV is sent to the victim, then the result of the attack become measurable.

Fig. 5 Testbed design and usage for the Clock Disable Port Attack. Our modular framework allows to build a system with IaC technologies, which can be configured to include an arbitrary number of entities such as clocks, attackers, etc. connected according to different topologies. In black, entities that are not part of the time synchronization platform; in green, entities that are correctly and mutually synchronized; in red de-synchronized entities

Clock disable port attack

This attack exploits the Direct Attack Module of PEF. The attacker will misconfigure the target by sending IT a TLV that will then result in a synchronization failure.

Implementation

The attack scenario is described in Fig. 5.

- Fig. 5a show the start up of the scenario, with the two synchronized hosts and the attacker. The latter does not require to be synchronized with the hosts.
- Fig. 5b show the attack phase. A `DISABLE_PORT` TLV is sent by the attacker to the victim. As a result the PTP port is disabled for two minutes, stopping the PTP daemon synchronization capability.

Results

Figure 6 graphically show the result of this attack. The figure reports the drift of the clock of the host under attack with respect to the master. In the leftmost part of the Figure (green curve) it is shown that the drift becomes quickly negligible because of the synchronization provided by PTP. Then at time $t = 30$ min the attack takes place and the port is disabled for 2 min (the blank part of the figure between the two dashed vertical lines).

When the port of the attacked node is enabled again (sending a `ENABLE_PORT` TLV to the victim), the clock of the node start drifting away the reference master (purple line on the rightmost part of the figure). In other words by stopping the synchronization for a period of time large enough our attack successfully introduced a significant clock drift, which increases in time and eventually reaching a value of 8 ms after thirty minutes. As a typical PTP synchronization should exhibit an accuracy in the range between 10ns to 100ns [IEEE 1588 Precise Time Protocol (2017)], we argue that the drift introduced by our attack is significant.

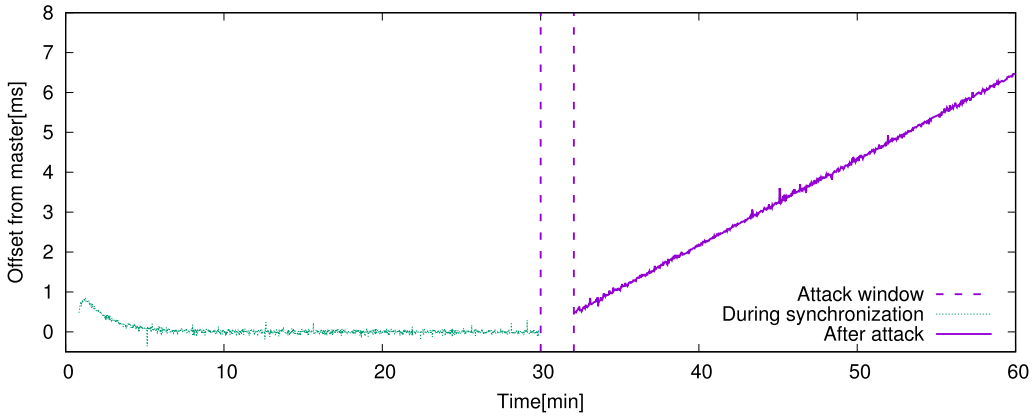


Fig. 6 Results of a clock de-synchronization attack in the virtualized testbed. In this attack we have exploited a `DISABLE_PORT` TLV. The clock then proceeds to drift from the reference master; during the attack we keep the port disabled and then proceed to re-enable it after one minute

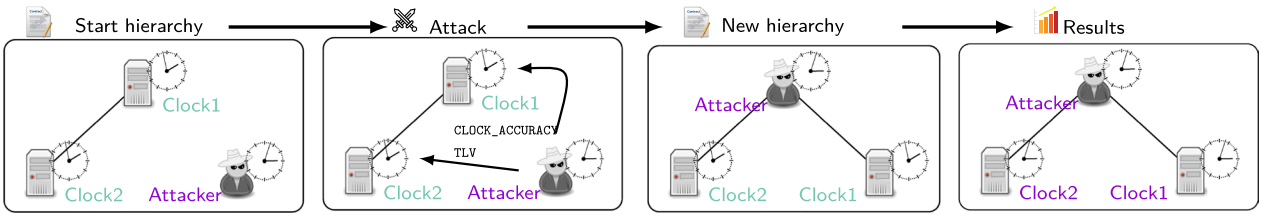


Fig. 7 Hierarchy-related attack: the attacker issue a CLOCK_ACCURACY TLV. This TLV will poison the hierarchy management algorithm; when an election occurs, the attacker will become the Master / Grand-Master of the system. The synchronized clocks not controlled by the attacker are represented in green; the ones controlled by the attacker at every step are drawn in red

Clock accuracy attack

This attack exploits the Direct Attack Module of PEF. In brief, the attacker will send a TLV that causes the target to choose a malicious reference clock and therefore lock on a wrong synchronization.

Implementation

The attack scenario is described in Fig. 7.

- In a first phase, referred as the *learning* phase, we synchronized the attacker node with the leaf nodes. This phase is required because the synchronization of PTP leaves can have some constraints like the maximum amount of μs which can be corrected. This is possible because, for the time being, there is no authorization mechanism over the synchronization of new leaves, this operation is completely transparent to the clock master.
- After the learning phase, the actual *spoofing* phase is started. The attacker takes the role of master-clock by sending a TLV announcing a clock with more priority by sending a `CLOCK_ACCURACY TLV`

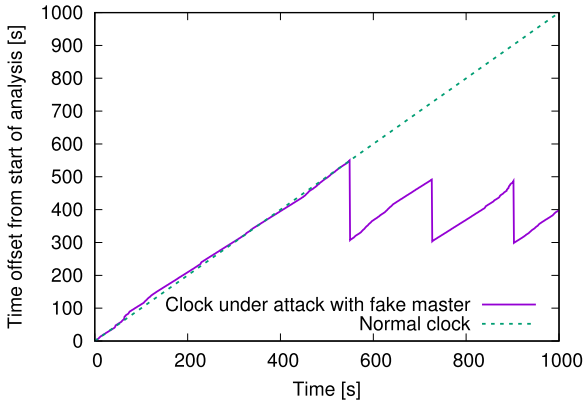


Fig. 8 When the attacker is elected as the master, the other clocks in the system are completely controllable. This graph shows a running query of the internal clock of a victim, compared with the internal clock value of a previously synchronized external observer

- The declared `ACCURACY` of the clock is used to alter the outcome of the next Best Master Clock Election (BMC), and will force the victims to choose the attacker as new reference.

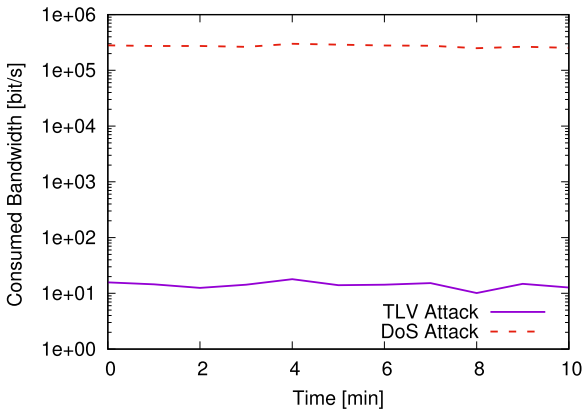


Fig. 9 Comparison between Bandwidth-requirements of a Denial of Service (DOS) attack and normal PTP operations (PTP). The data are a result of an average between five runs of 10 minutes and represented using logarithmic scale. The Denial of Service attack was done using hping3

Results

The final result of the attack is shown in Fig. 8. A normal clock should follow the values of the green dashed figure, but the clock of the attacked station is periodically altered on fake values by the attacker (that brings it back of a second every ten seconds. In practise Fig. 8 proves that the clocks of the victims are prevented from counting time in the right way.

Implementation complexity

As a final result, Fig. 9 supports what discussed in Sect. 4. Here are compared the amounts of bandwidth consumed by the messages that must be sent on the network to implement a Bandwidth Denial of Service and a TLV attack. This orders of magnitude larger for the DDoS. This means a larger effort for the attacker but also a larger probability of the attack being detected but some network monitoring tool.

Countermeasures

Now let us briefly discuss possible countermeasures to the attacks previously described.

As TLVs are enabling the attacks described, the most trivial countermeasure could be to disable the support of TLVs in the specific protocol deployment. Unfortunately this is a solution with many weaknesses. First of all, as already stated, TLVs are used to implement the configuration features of a deployment exploiting the PTP protocol. Disabling the TLVs will therefore result into a significant loss of functionality. This could be fine in a deployment which will never require any form of runtime configuration, but every system administrator knows that this is a very unrealistic case. Disabling the TLV while keeping some form of configuration capability would require a different configuration channel, implemented with another protocol such as SNMP or TELNET. But these protocols may also be subject to problems equivalent to the one described in this work for TLVs. In summary, disabling the TLVs would either cause a significant loss of functionality or require the introduction of additional protocols that may result in alternative attack vectors. Thus, we do not believe disabling the TLVs can be considered a reasonable countermeasure to the attacks described in this manuscript.

While disabling TLVs can generate problems, the access control can be delegated to the network infrastructure. This comes at the price of a more complicated management of the entire network. A common approach to achieve this kind of segmentation is the use of VLANs, usually implemented by IEEE-802.1Q. More sophisticated segmentation methods can be implemented using technologies such as SDN (Callegati et al. 2021). As stated, the network administrator is then responsible of the isolation of the visibility between the devices, with regard to TLV packets.

A more interesting option is to make the exchange of the TLVs more secure. Annex-K of the 2008 standard version discusses the possibility of TLV encryption and the 2019 standard version introduces it. Supported encryption modes include both common approaches, i.e. symmetric encryption with shared keys, which however is prone to insider threats, dictionary attacks and so on, and asymmetric encryption enabled by a public key infrastructure. The latter is not vulnerable to dictionary or brute-force attacks, but may still be a source of challenges, since it requires a complex setup to ensure security of the entire system. First of all the PKI need to provide a fine handling of revocation processes, otherwise insider threats can pass unnoticed. Moreover, if not configured and designed correctly, this kind of systems can trick the user into man-in-the-middle scenarios or admit rogue machines (Prandini et al. 2010). In many application cases the complexity of setting up a security infrastructure of this kind will discourage its application, leading to the adoption of a simpler, yet vulnerable set-up.

The implementation of more usable, but still robust methods to configure authentication architectures based on asymmetric cryptography in industrial network is a topic which we are currently investigating, mainly referring to the OPC-UA protocol, and which could provide a more robust answer to the security problems presented in this work.

Conclusions

In this manuscript we have discussed the issue of cybersecurity attacks to time synchronization in industrial networks. We argue that, while attacking high level encrypted protocol can be cumbersome, under some preconditions attacking time synchronization protocols can be the best way for an attacker to exploit a system.

In particular we considered the Precision Time Protocol and have shown that Type Length Value (TLV) frames, used for management purposes, can be abused by an attacker to reconfigure, manipulate, or shut down time synchronization. In the manuscript we proposed and experiment attacks based on TLV exploitation, demonstrating that they effectively de-synchronize the clocks, achieving a drift which is well above the typical tolerance required in networks where PTP synchronization is deployed. Moreover these attacks require an amount of bandwidth and of packets dedicated to the attack that is very limited. This makes the attacks very difficult to detect by typical traffic monitoring and/or IDS systems.

Since the vulnerability stems from common configuration practices, we highlighted the need for them to address authentication of management messages, at the same time noting that the way PTP standards foresee to perform authentication is not necessarily easy to implement in an industrial network. In fact, a relevant contribution of our work is to highlight that PTP in any version is only as secure as the deployer decides to configure it. There is no mandatory security measure, only available tools that need to be implemented, bearing the associated complexity costs.

Appendix A: TLVs taxonomy

In this appendix we present a taxonomy of the most common TLVs according to how they can be exploited for attacks to the protocol. The score of the attack is calculated using the CVSS scoring system[6]. In this system we classify the different score based on the following table:

AC Complexity of the attack **H** → high, **M** → medium, **L** → low.
C Confidentiality is harmed: **N** → none, **P** → partial, **C** → complete.
I Integrity is harmed: **N** → none, **P** → partial, **C** → complete.
A Availability is harmed: **N** → none, **P** → partial, **C** → complete.

[6] https://nvd.nist.gov/vuln-metrics/cvss visited 2020-06-04.

Table 2 PTP IEEE-1588 2008 TLVs that can be used in the Reconnaissance phase

TLV name	Scope	Score
CLOCK_DESCRIPTION	Network	AC:L/C:N/I:N/A:N
USER_DESCRIPTION	Network	AC:L/C:N/I:N/A:N
FAULT_LOG	Log	AC:M/C:N/I:N/A:N
DEFAULT_DATA_SET	BMCS	AC:L/C:N/I:N/A:N
CURRENT_DATA_SET	BMCS	AC:L/C:N/I:N/A:N
PARENT_DATA_SET	BMCS	AC:L/C:N/I:N/A:N
PORT_DATA_SET	BMCS	AC:L/C:N/I:N/A:N
PATH_TRACE_LIST	Extension	AC:L/C:N/I:N/A:N
PATH_TRACE_ENABLE	Extension	AC:L/C:N/I:P/A:N

The TLVs not presented in these tables seems to be unexploitable.

The attacker model will follow different phases, nominally:

1 Reconnaissance;
2 Intrusion and delivery of the attack;
3 Exploitation of the vulnerability;
4 Persistence of the attack;
5 Cleanup of the traces.

Mimicking this model, we can separate the various unsecure TLVs in the following clusters. These TLVs can imply security in different scopes, these scopes include:

Network General PTP network domain;
Time Time syncronization;
BMCS Best Master Clock Selection algorithm;
Log Log management;
Extension PTP-2008 standard extensions.

TLVs that can be used for Reconnaissance

The reconnaissance phase is the first foothold in the network, it analyzes the network to search for the exploitable vulnerabilities.

In Table 2, is presented a list of TLVs that can be used to scan the network or gain sensible information that can be used in subsequent phases.

TLVs that can be used to exploit the network

In the exploitation phase, the attacker will exploit the discovered vulnerabilities and try to maximize the damage of his attack. The TLVs in Table 3 can be employed to do so:

TLVs that can be used to tamper the network

After the exploitation phase, the attacker can try to maintain the access to the system. To do so a set of TLVs, listed in Table 4, can be employed.

Table 3 PTP IEEE-1588 2008 TLVs that can be used to exploit the network

TLV name	Scope	Score
INITIALIZE	BMCS	AC:M/C:N/I:N/A:C
UTC_PROPERTIES	Network,Time	AC:L/C:N/I:N/A:C
VERSION_NUMBER	Network,Time	AC:L/C:N/I:N/A:C
LOG_SYNC_INTERVAL	Network,Time	AC:L/C:N/I:P/A:C
ANNOUNCE_RECEIPT_TIMEOUT	Network,Time	AC:L/C:N/I:N/A:P
LOG_ANNOUNCE_INTERVAL	Network,Time	AC:L/C:N/I:N/A:P
DOMAIN	Network,Time	AC:L/C:N/I:N/A:P
SLAVE_ONLY	Network,Time	AC:L/C:N/I:N/A:P
DISABLE_PORT	Network	AC:L/C:N/I:N/A:C
TIME	Time	AC:L/C:N/I:N/A:C
CLOCK_ACCURACY	BMCS	AC:L/C:N/I:P/A:P
PRIORITY1	BMCS	AC:L/C:N/I:P/A:P
PRIORITY2	BMCS	AC:L/C:N/I:P/A:P
DELAY_MECHANISM	Network	AC:L/C:N/I:P/A:C
TIMESCALE_PROPERTIES	Time	AC:L/C:N/I:P/A:P
UNICAST_NEGOTIATION_ENABLE	Network	AC:L/C:N/I:P/A:P
GRANDMASTER_CLUSTER_TABLE	Network	AC:L/C:N/I:C/A:P
UNICAST_MASTER_TABLE	Network	AC:L/C:N/I:C/A:P
ACCEPTABLE_MASTER_TABLE	BMCS	AC:L/C:N/I:P/A:P
ACCEPTABLE_MASTER_TABLE_ENABLE	BMCS	AC:L/C:N/I:P/A:P
ALTERNATE_MASTER	Network	AC:L/C:N/I:P/A:P

Table 4 PTP IEEE-1588 2008 TLVs that can be used to tamper the network

TLV name	Scope	Score
SAVE_IN_NON_VOLATILE_STORAGE	BMCS,Time	AC:M/C:N/I:P/A:N
RESET_NON_VOLATILE_STORAGE	BMCS,Time	AC:M/C:N/I:P/A:N

Table 5 PTP IEEE-1588 2008 TLVs that can be used to cover tracks of a PTP attack

TLV name	Scope	Score
FAULT_LOG_RESET	Log	AC:L/C:N/I:P/A:N

TLVs that can be used to cleanup traces of attack

TLVs can also be employed to cleanup the trace of an attack, e.g. to hide the presence of a trojan horse in the network. This can be the first step in the track covering, making difficult for an incident response team to reconstruct the dynamics of the attack. In Table 5, is presented a set of TLVs of the PTP standard that can be used to do so.

Acknowledgements
Not applicable.

Author Contributions
DB: conceptualization, investigation, software, writing-original draft. NOT: validation, visualization. AM: investigation, software. MP: supervision, writing—review and editing. FCi: funding acquisition, writing—review and editing.

Funding
Not applicable. All authors reviewed the results and approved the final version of the manuscript.

Availability of data and materials
Not applicable.

Declarations

Competing interests
Not applicable.

Received: 23 August 2022 Accepted: 13 January 2023
Published online: 11 April 2023

References

Annessi R, Fabini J, Iglesias F, Zseby T (2018) "Encryption is futile: Delay attacks on high-precision clock synchronization," arXiv preprint arXiv:1811.08569,
Arnold D (2019) TLVs in PTP Messages. Accessed 22 June 2022 [Online]. Available: https://blog.meinbergglobal.com/2019/12/06/tlvs-in-ptp-messages/
Callegati F, Campi A, Contoli C, Di Santi S, Ghiselli N, Giannelli C, Pernafini A, Zamagna R (2021) Sdn-based differentiated traffic flow management for industrial internet of things environments in. IEEE Symposium on Computers and Communications (ISCC) 2021:1–6
Cochran R et al (2015) The linux ptp project. Accessed 7 July 2022 [Online]. Available https://sourceforge.net/projects/linuxptp/
Electrics S (2017) What is TSN? The backbone of future industrial ethernet networks. Accessed 13 July 2022 [Online]. Available https://blog.se.com/energy-management-energy-efficiency/2017/06/02/tsn-backbone-future-industrial-ethernet-networks/
Fedullo T, Morato A, Tramarin F, Rovati L, Vitturi S (2022) A comprehensive review on time sensitive networks with a special focus on its applicability to industrial smart and distributed measurement systems. Sensors 22(4):1638
Han M, Crossley P (2019) Vulnerability of ieee 1588 under time synchronization attacks. In: 2019 IEEE Power & Energy Society General Meeting (PESGM). IEEE, pp 1–5
Hemsley KE, Fisher E, *et al* (2018) History of industrial control system cyber incidents
Ieee standard for a precision clock synchronization protocol for networked measurement and control systems - redline. IEEE Std 1588-2008 (2008) (Revision of IEEE Std 1588-2002) - Redline, pp 1–300
"Ieee standard for a precision clock synchronization protocol for networked measurement and control systems," IEEE Std 1588-2019 (2020) (Revision of IEEE Std 1588-2008), pp. 1–499
Ieee 1588 precise time protocol: The new standard in time synchronization (2017). Accessed: 7 July 2022 [Online]. Available: https://www.microsemi.com/document-portal/doc_download/133186-ieee-1588-precise-time-protocol-the-new-standard-in-time-synchronization
Lo Bello L, Steiner W (2019) A perspective on ieee time-sensitive networking for industrial communication and automation systems. Proc IEEE 107(6):1094–1120
Miller T, Staves A, Maesschalck S, Sturdee M, Green B (2021) Looking back to look forward: Lessons learnt from cyber-attacks on industrial control systems. Int J Crit Infrastruct Prot 35:100464
Mizrahi T (2011) Time synchronization security using ipsec and macsec. In: 2011 IEEE international symposium on precision clock synchronization for measurement, control and communication. IEEE, pp 38–43
Moussa B, Debbabi M, Assi C (2016) A detection and mitigation model for ptp delay attack in an iec 61850 substation. IEEE Trans Smart Grid 9(5):3954–3965
Moussa B, Kassouf M, Hadjidj R, Debbabi M, Assi C (2019) An extension to the precision time protocol (ptp) to enable the detection of cyber attacks. IEEE Trans Ind Inform
Nasrallah A, Thyagaturu AS, Alharbi Z, Wang C, Shao X, Reisslein M, ElBakoury H (2018) Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research. IEEE Commun Surv Tutor 21(1):88–145
Neyer J, Gassner L, Marinescu C (2019) Redundant schemes or how to counter the delay attack on time synchronization protocols. In: 2019 IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS). IEEE, pp 1–6
Owczarek W et al (2015) The ptpd project. Accessed 7 July 2022 [Online]. Available https://github.com/ptpd/ptpd
Prandini M, Ramilli M, Cerroni W, Callegati F (2010) Splitting the https stream to attack secure web connections. IEEE Secur Privacy 8(6):80–84
Raveling A (2022) Time Sensitive Networking for industrial applications. Accessed 13 July 2022 [Online]. Available: https://www.controleng.com/articles/time-sensitive-networking-for-industrial-applications/
Shereen E, Bitard F, Dán G, Sel T, Fries S (2019) Next steps in security for time synchronization: Experiences from implementing ieee 1588 v2. 1. In: 2019 IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS). IEEE, pp 1–6
Siemens (2022) TSN—Time Sensitive Networking. Accessed 13 July 2022 [Online]. Available https://new.siemens.com/global/en/products/automation/industrial-communication/industrial-ethernet/tsn.html
Ullmann M, Vögeler M (2009) Delay attacks-implication on ntp and ptp time synchronization. In: International symposium on precision clock synchronization for measurement, control and communication. IEEE 2009, pp 1–6

Publisher's Note