Alghamdi and Schukat *Cybersecurity* (2021) 4:12
https://doi.org/10.1186/s42400-021-00080-y

Cybersecurity

RESEARCH Open Access

Precision time protocol attack strategies and their resistance to existing security extensions

Waleed Alghamdi* and Michael Schukat

Abstract

The IEEE 1588 precision time protocol (PTP) is very important for many industrial sectors and applications that require time synchronization accuracy between computers down to microsecond and even nanosecond levels. Nevertheless, PTP and its underlying network infrastructure are vulnerable to cyber-attacks, which can stealthily reduce the time synchronization accuracy to unacceptable and even damage-causing levels for individual clocks or an entire network, leading to financial loss or even physical destruction. Existing security protocol extensions only partially address this problem. This paper provides a comprehensive analysis of strategies for advanced persistent threats to PTP infrastructure, possible attacker locations, and the impact on clock and network synchronization in the presence of security protocol extensions, infrastructure redundancy, and protocol redundancy. It distinguishes between attack strategies and attacker types as described in RFC7384, but further distinguishes between the spoofing and time source attack, the simple internal attack, and the advanced internal attack. Some experiments were conducted to demonstrate the impact of PTP attacks. Our analysis shows that a sophisticated attacker has a range of methodologies to compromise a PTP network. Moreover, all PTP infrastructure components can host an attacker, making the comprehensive protection of a PTP network against a malware infiltration, as for example exercised by Stuxnet, a very tedious task.

Keywords: APT, Cyber-attacks, IEEE 1588, PTP, Security, Time synchronization protocols

Introduction

The recent decade has been marked by significant security problems, and the emergence of complex cyber-attacks called Advanced Persistent Threats (APTs). Such attacks often begin by targeting a small number of power users within the target organization with malicious software, for example, malware on secondary memory devices (i.e., USB sticks) or phishing emails. They then propagate themselves across the organization by exploiting software flaws. Several technology providers, including RSA and Google, fell victim to APTs and made it public. The emergence of APTs has demonstrated the limitations of network-centric perimeter security that has been exercised for many years, where a firewall isolates and protects infrastructure and information from unreliable networks, e.g., the Internet. With APTs, all networks are deemed unreliable, and the security perimeter has to be user-centric (Baize 2012).

APTs, by their nature, are very difficult to detect and typically incorporate either a static (and therefore less effective) signature-based malicious code detection or a behavior-based detection (Cho and Nam 2019) using correlation analysis, for example, of network traffic patterns. Here a recent trend to use machine learning methodologies can be observed (Quintero-Bonilla and Martín del Rey 2020).

Stuxnet and the subsequent attack on the Natanz Uranium enrichment facility in 2010 is an example of an advanced attack on critical infrastructure. It started with

* Correspondence: w.alghamdi1@nuigalway.ie
School of Computer Science, National University of Ireland Galway Ireland

an infected USB stick, which was unknowingly brought into this high-security facility by an employee. Stuxnet subsequently spread across the isolated Natanz network infrastructure and took over control of PLCs and SCADA systems responsible for the centrifuges used in the Uranium enrichment process, which were subsequently damaged via subtle changes to their operating parameters over many months (Chen and Abu-Nimeh 2011; Langner 2011). Stuxnet would have been very hard to detect even with today's advances in machine learning techniques, as its operation caused no apparent changes in network traffic patterns or PLC behavior.

Another important example of critical infrastructure relates to the exact clock synchronization between computer systems, as required by many sectors such as telecommunications or financial services, where local time sources (e.g., quartz-based real-time clocks) alone are not sufficient because of stability and accuracy problems, resulting in local clock derivations in the order of milliseconds per day (Shannon et al. 2012). Over packet-switched networks, such time synchronization can be provided by two protocols, the Network Time Protocol (NTP) and the Precision Time Protocol (PTP). These protocols are the base for time-sensitive systems, especially distributed network systems, as they manage how a host clock is adhering to a time-scale reference. Both protocols are based on a clock synchronization technique that specifies the order and sequence of message transmissions between a host and reference clock, the message structure as well as the required time synchronization processes (Shannon 2013).

Time synchronization protocols are typically designed for particular types of networks: The Network Time Protocol (NTP) is suitable for large and dynamic latency packet-switched networks (PSNs), using complex statistical techniques that effectively reduce the inherent synchronization errors in such networks (Mills 1991). It fulfills the requirements of distributed systems that need accuracy in the order of a few milliseconds over wide area networks. On the other hand, the Precision Time Protocol (PTP) (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2008) is designed for infrastructure networks, i.e., well-managed PSNs, that often use specialized (PTP-aware) hardware, providing clock synchronization accuracy down to microsecond and even nanosecond level (Alghamdi and Schukat 2017).

Many financial markets and leading exchanges such as IMC, Eurex, and NYSE allow PTP time synchronization from their systems with market client/participants, so that they can synchronize their clocks with the exchange (Estrela et al. 2014). Here PTP failure can lead to devastating consequences. For example, Eurex uses a very sophisticated PTP time synchronization network to timestamp financial and stock transactions of their clients, including high-frequency trading. The synchronicity and accuracy of these timestamps are very important to the exchange and its customers. However, on 26th August 2013, a PTP infrastructure glitch occurred that, even though it was detected in time, forced Eurex to postpone its market opening. It later turned out that an incorrect leap second calculation caused an erroneous synchronization of their critical systems (Estrela et al. 2014).

While this example demonstrates the impact of network time synchronization problems, it raises the more general question of the vulnerability of PTP- and NTP-based time synchronization packet-switched networks to APTs, which subsequently pose a high risk to many time-sensitive application areas (Mizrahi 2011). Previous attempts at security protocol extensions, such as the (OSI layer 7) IEEE 1588 Annex K for PTP, and Autokey for NTP, are insufficient to deter cyber-attacks (Alghamdi and Schukat 2017). Moreover, state-of-the-art network layer 2 and layer 3 protocols (i.e., MACsec and IPsec) can only deter a subset of possible attack strategies, namely external attacks (Mizrahi 2011).

This paper focuses on a much more devious attack type on PTP networks, the internal attacks, which are much harder to detect, as they allow an attacker to compromise PTP infrastructure components, similar to the way Stuxnet compromised industrial control infrastructure in its final stage of operation (Knapp and Langill 2015; Vacca 2017). As with Stuxnet, internal PTP attacks do not cause obvious behavioral changes in PTP devices or unusual network traffic patterns. Therefore this paper does not analyze the effectiveness of APT mitigation and detection strategies, but focuses on viable internal attack strategies and therefore making the following contributions: (1) analyze and classify all possible PTP attacks, thereby dividing the internal attack into two types, namely simple and advanced internal attacks, as well as dividing the spoofing attack into master and slave spoofing attacks, (2) outline possible implementations in detail, (3) demonstrate the vulnerabilities of existing security measures to prevent the internal attacks, (4) show the effects of the attacks on clock synchronization. It is structured as follows: Section II provides a brief overview of PTP as well as details on the attack and attacker types, followed by a description of existing security mechanisms (i.e., protocol extensions and infrastructure enhancements) and the security enhancements stipulated by the emerging PTP standard 2.1 (IEEE 1588–2019) (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2020) in section III. Section IV investigates potential attack strategies, their subsequent impact on slave clock synchronization, and the

robustness of the aforementioned security mechanisms. Section V provides experimental validation of some attacks as well as a summary of findings. A security extension is proposed in Section VI. Finally, an outlook of future work is given in Section VII.

PTP architecture and attack(er) types

PTP overview

In 2002, IEEE introduced the IEEE 1588 standard (PTP) to provide a synchronization protocol for time synchronization of distributed devices with microsecond-level accuracy (Shannon 2013), using GPS receivers or atomic clocks as a time reference. In 2008, the second version of PTP was published (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2008), while version 2.1 (IEEE 1588–2019) has been recently released (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2020).

PTP distinguishes between different clock types. Ordinary clocks (OC) consist of one PTP port. One of these clocks is the grandmaster clock (GM), which provides the time reference for the network. The other ordinary clocks are called slaves, and they exchange time synchronization messages with the master. PTP uses the Best Master Clock (BMC) algorithm to determine which ordinary clock will be the grandmaster. Boundary clocks (BC) and transparent clocks (TC) are similar to switches and placed between the grandmaster and slave clocks. A boundary clock has multiple PTP ports; one of them (the slave port) is synced to the grandmaster, and the other ports become masters to downstream slaves. A transparent clock determines the residence time of each time synchronization packet passing it and updates the packet's time correction field accordingly, therefore achieving a better slave time synchronization. The exchanged PTP messages differ depending on whether the End-to-End (E2E) or peer-to-peer (P2P) delay measurement mechanism is used (Garner 2008).

To achieve optimal clock synchronization, PTP assumes that the path delays between a slave and the time reference master are symmetric, i.e., uplink and downlink latencies are similar; otherwise, network delays are not properly computed, and the slave clock synchronization accuracy will be reduced (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2008).

Attack types

RFC7384 (Mizrahi 2014) entails cyber-attack threats to time synchronization protocols. It distinguishes between internal and external attacks by either a man-in-the-middle or an injector attacker (Alghamdi and Schukat 2017). Figure 1 illustrates such attacks, the attackers, and their locations within a PTP network model that incorporates the various PTP infrastructure components previously mentioned. The diagram distinguishes between two trusted networks (1 and 2), which are interconnected via an untrusted network. Both trusted networks are fully managed and have (potentially) implemented the same L2, L3, or L7 (i.e., Data Link, Network, or Application Layer) security mechanisms. Similarly, (Itkin and Wool 2020) distinguishes between insider and outsider adversaries. The outside adversary can only see multicast messages, while the inside adversary can see all protocol messages.

Internal attack

Here the attacker has access to a trusted component of the network and may have access to the security (i.e., authentication/encryption) keys used. An internal attacker can manipulate maliciously legitimate network traffic or create new packets that appear legal to the manipulated nodes (Mizrahi 2014). The internal attack will be further classified into two sub-categories, as follows:

a) *Advanced Internal Attack*

Here the attacker gains full access to a device, including access to the encryption/authentication keys used, by means of a malware infection or a manipulated firmware upgrade. Subsequently, the attacker takes control of the device behavior or configuration, for example, by changing its clock properties to fool the BMC algorithm. This kind of attack can also change packet content in transit or generate new legitimate-looking packets. In Fig. 1, Router1, Router2 GM, BC, TC, OC1, OC2, and OC5 are points from which to launch such an advanced internal attack.

b) *Simple Internal Attack*

Here an attacker resides within a trusted network, either on a secretly-added untrusted device, or on a legitimate trusted device, but without having access to cryptographic keys. This kind of attacker has limited capabilities that may include packet removal, packet delays, or traffic generation to perform a denial of service (DoS) attack. In Fig. 1, the switch and OC3 are points from which to launch a simple internal attack.

External attack

Here the attacker does not have possession of secret network encryption or authentication keys and resides outside the trusted network. In Fig. 1, Router3, and OC4 are possible external attack points.

Attacker types

Man-in-the-middle attacker

A man-in-the-middle attacker is located in a position where it can intercept and modify protocol packets in-flight. It has physical access to a node of the PTP network or has gained full control of one device in the network (Mizrahi 2014). For example, in Fig. 1, Router1, TC and Switch are possible internal man-in-the-middle attackers that reside in a trusted network (i.e., Trusted Network 1), while Router2 is another example of an internal man-in-the-middle attacker, who has access to an intermediate node with the cryptographic keys in another trusted network segment (Trusted Network 2). Please note that while BC is an intermediate node, it acts as an endpoint between uplink and downlink and does not forward any event messages between the grandmaster and the other slaves. In contrast, Router3 is an example of an external man-in-the-middle attacker, who can prevent some or all protocol messages from arriving at their destinations.

Packet injector attacker

A traffic injector attacker is located in a position that allows it to generate network traffic. In Fig. 1, an internal injector attacker can reside and inject traffic within the main network (Router1, GM, TC, Switch, OC1, OC2, and OC3), or has access to a node in another trusted network (Router2, BC and OC5). Router3 and OC4 are external injectors with limited attack capabilities (Mizrahi 2014).

Scope of analysis

This research will mainly focus on internal attacks via packet injectors or man-in-the-middle, since a PTP network is typically a tightly managed and therefore trusted (and potentially even isolated) infrastructure that is confined within an organization. External attacks are only considered in the context of Fig. 1, with Router3 and OC4 being potential entry points.

The assumption is that an attacker gains access to one or more PTP infrastructure components via a malware infection (for example by means of phishing or USB exploits as documented with Stuxnet) and, once established, launches an ATP with the aim of compromising synchronization of PTP clocks stealthily, potentially over an extended period of time, in order to cause infrastructure failure or degraded service.

PTP safeguards

Cryptographic protocol security

IEEE 1588 Annex K

IEEE 1588 version 2 defines an experimental L7 security extension to PTP called Annex K. It provides group source authentication, message integrity, and replay protection security using symmetric keys. It creates a trust relationship utilizing a challenge-response three-way handshake mechanism based on pre-defined keys that are reached by subsets or the entire PTP domain (Itkin and Wool 2020). Since its release in 2008, various flaws have been discovered (Itkin and Wool 2020; Önal and Kirrmann 2012; Pathan et al. 2014), which resulted in various suggestions for protocol improvements, including the use of public-key encryption (Itkin and Wool 2020) and an improved handshake and replay counter (Alghamdi and Schukat 2017). Today Annex K is deemed to be obsolete and has been dropped in favor of conventional L2/L3 security extensions.

E2E protocol security (IPSec)

IPsec provides L3 security protocols for IP networks by authenticating and encrypting IP packet payloads or by

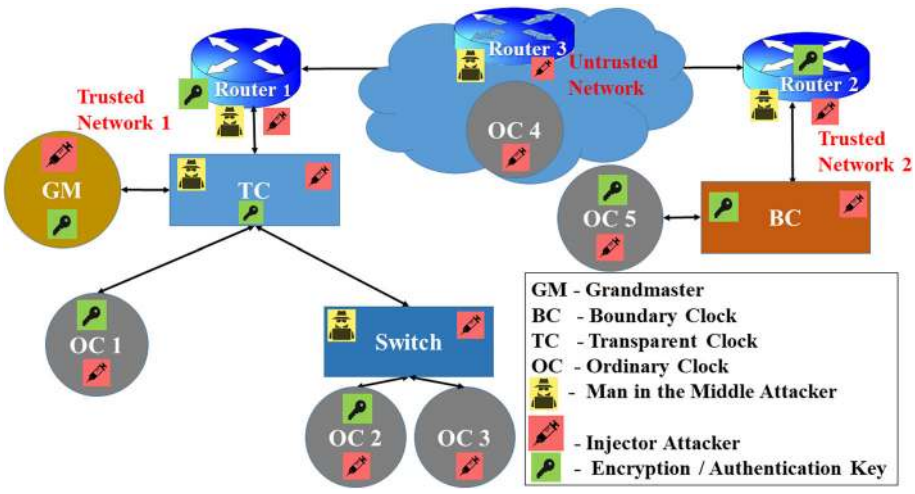

Fig. 1 PTP Network with Attack and Attacker Positions

authenticating the non-modifiable sections of the IP header, hereby supporting both a transport mode between endpoints and a tunnel mode between security gateways. IPsec is designed to deter some external attacks, such as eavesdropping, replay attacks, and packet modification. However, it is not designed to work in tandem with PTP. For example, as a L3 protocol, it does not allow for PHY layer hardware timestamping or the easy integration of intermediate TCs to deliver the best possible slave clock synchronization (Alghamdi and Schukat 2017; Mizrahi 2011). Also, tunnel mode IPsec does not support the integration of on-path intermediate BCs, while its cryptographic engine causes extra latency/jitter that negatively impacts on the synchronization performance (Chen 2013).

P2P protocol security (MACSec)

MACsec is an L2 security protocol that relies on IEEE 802.1X (for key management and session initiation) and IEEE 802.1AE (which specifies the authentication and encryption protocol). As an L2 protocol, MACsec provides a hop-by-hop authentication and encryption mechanism, therefore supporting hardware timestamping and the full integration of BCs and TCs (Alghamdi and Schukat 2017). MACsec protects the connection between trusted segments of the network infrastructure, but cannot prevent attacks that are launched from these trusted segments. It is complementary to end-to-end security protocols, as it can protect application data independently of network operations, but cannot necessarily protect the operation of network segments (IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security 2006).

Infrastructure enhancements

Multiple paths

The variability of network latencies presents a challenge, as the accuracy of clock synchronization relies on the symmetry and steadiness of propagation delays in the uplink and downlink direction between the master clock and the slave clock. A computer network is prone to path asymmetry and variable network latency, depending on the nature of the underlying network. Multiple network paths can improve fault-tolerance and PTP performance by providing multiple PTP message paths between a master and its slaves. Such means also improve security, as it complicates man-in-the-middle attacks (Shpiner et al. 2013). Multiple paths can be achieved via VLAN (Shpiner et al. 2013), or via High-availability Seamless Redundancy (HSR) in combination with the Parallel Redundancy Protocol (PRP) (Koskiahde and Kujala 2016).

Redundant grandmaster

Multiple redundant grandmasters can be utilized to compensate for byzantine failures, where the master clock provides an incorrect time reference (Dalmas et al. 2015). Here the redundant grandmasters compare the active master's time with their own time. If the computed difference exceeds a particular value, one of the passive grandmasters becomes the main grandmaster.

Protocol redundancy

Multi-time protocol synchronization of PTP slaves provides another mechanism to prevent byzantine failures (Estrela et al. 2014). Here a slave uses NTP in parallel with PTP and determines offsets from multiple stratum time sources. Their median value is compared against the measured PTP offset, and the former is used to correct the local clock if the difference between the two values is larger than a threshold.

De-militarized zone

The De-Militarized Zone (DMZ) is a method of creating a semi-secure network that works as the first line of defense to secure the internal infrastructure of an organization from external attackers (Dadheech et al. 2018). DMZ is useful for networks that need to share devices or endpoints (e.g., web servers) publicly. As such, it does not protect against an internal attacker who is already inside a trusted network.

PTP V2.1 (IEEE 1588–2019) Annex P

PTP v2.1 (IEEE 1588–2019) introduced a new security extension called Annex P, which retains backward compatibility to previous PTP versions. It addresses security in four prongs as follows (Neyer et al. 2019; Shereen et al. 2019):

1) Prong A (PTP Integrated Security Mechanism) describes a type-length-value (TLV) extension for message authentication using symmetric encryption. There are two different operating modes supported; (1) immediate security processing that relies on a shared group key (2) delayed security processing that is supported by the Timed Efficient Stream Loss-Tolerant Authentication (TESLA) protocol. This prong can be classified as cryptographic protocol security as described above.
2) Prong B (PTP External Transport Security Mechanisms) describes existing external security mechanisms including IPsec and MACsec.
3) Prong C (Architecture Guidance) describes an overview of architectural security measurements, namely redundancy. With redundancy, an attacker must compromise multiple points to manipulate the time synchronization. IEEE 1588 defined three

types of redundancy: redundant time system, redundant grandmaster, and redundant paths (Donoghue et al. 2017). This prong is similar to infrastructure enhancements as already described in this section.

4) Prong D (Monitoring and Management Guidance) describes a monitoring mechanism to observe the PTP behavior to detect (rather than deflect) a potential attack such as a DoS attack, via monitoring slave clock parameters including offset and delay measurements. Please note that this research focuses on prevention means rather than detection.

Attack strategies and implementations

Previous research has shown that MACsec, IPsec, and Annex K only protect against certain external attacks, but not against internal attacks, as any trust token that either identifies the origin or guarantees the security/integrity of PTP messages may be compromised (Mizrahi 2011). Consequently, Prong A and B of PTP 2.1 (IEEE 1588–2019) Annex P do not provide protection against internal attacks.

This section will extend these results by providing a similar vulnerability analysis of infrastructure enhancements including the Annex P Prong C in the presence of an internal attacker. This is complemented by an assessment of possible internal attack implementations and their impact/severity in the presence of cryptographic protocols or infrastructure enhancements.

Table 1 summarizes different PTP attack strategies as outlined in (Mizrahi 2011) and further distinguishes between master spoof attacks and slave spoof attacks, as further described in this section.

Figure 2 shows a PTP network model that incorporates all relevant network elements (i.e., routers, secured and unsecured network segments) and PTP hardware elements (GM, TC, BC, and OC). For each element, it shows what attacker type, as described in Section II, can use what attack strategy as listed in Table 1. Here, yellow and red stars denote if the strategy can or cannot be averted by at least one of the PTP security safeguards as listed in Section III. Note that infrastructure enhancements are not explicitly integrated into this diagram, instead multi-paths redundancy is referred to (Shpiner et al. 2013) and (Koskiahde and Kujala 2016), while protocol redundancy is referred to (Estrela et al. 2014).

Table 1 PTP Attack Strategies

NO	Attack Strategy
1	Packet Content Manipulation Attack
2	Packet Removal Attack
3	Packet Delay Manipulation Attack
4	Time Source Degradation Attack
5	Master Spoofing Attack
6	Slave Spoofing Attack
7	Replay Attack
8	BMCA Attack
9	Denial of Service Attack

All considered attacks must be persistent (i.e., continuously manipulate PTP traffic for the duration of the attack) in order to have the desired effect. Once an attack is terminated, normal PTP operation will resume, and affected slave clocks will slowly resynchronize again.

Packet content manipulation attack

Attack overview

In a packet content manipulation attack, a *man-in-the-middle attacker* manipulates suitable fields of time protocol packets in transit, hereby manipulating the clock synchronization of all clocks downstream, or making them go into free-running mode (Mizrahi 2014).

Attack implementation

In Fig. 2, Router1, Router2, Router3, Router4, TC1, TC2, and TC3 are all suitable points from which to launch a packet content manipulation attack (red and yellow stars number 1); for example:

1) TC1 has access to the network security key(s). So, an attacker who resides on TC1 can launch an advanced internal man-in-the-middle attack on OC2 and OC3 by intercepting and changing *all* Sync/Follow_Up messages as follows:
   a) For every Sync/Follow_Up message, add a fixed or an incremental error value to the *originTimestamp / preciseOriginTimestamp* or *correctionField* fields. Since PTP clients disregard clock offset calculations beyond a certain threshold and go into free-running mode instead, such values must be carefully selected. Likewise, the sudden termination of such an attack would cause the slave to detect the cumulated error.
   b) Change the *versionPTP* value from (version) 2 to (version) 1; OC2 and OC3 won't support the obsolete older version of PTP and will eventually go into free-running mode.
   c) PTP clocks can only communicate with each other, if they share the same *domainNumber* value. Changing this parameter will cause them to discard all synchronization messages they receive, and to eventually go into free-running mode.
2) TC2 is a suitable (simple internal man-in-the-middle) attack point from which to perform a packet

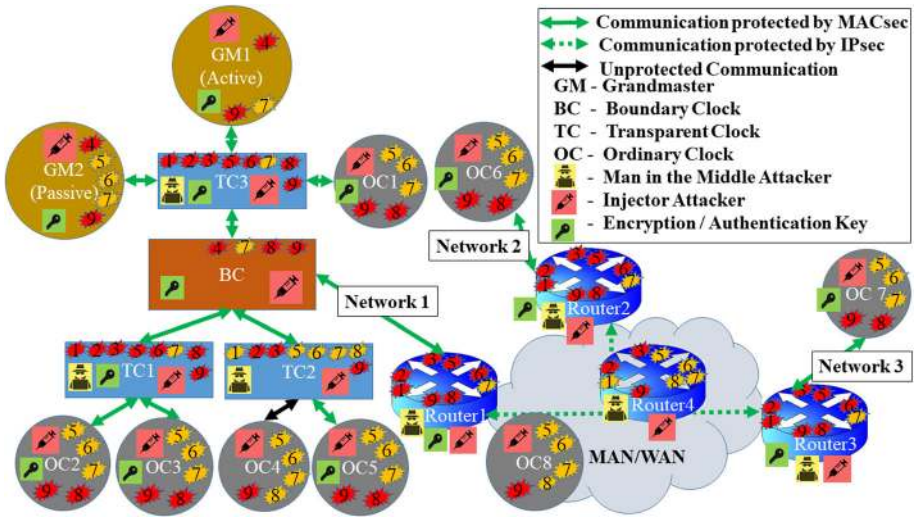

Fig. 2 Single-path PTP Network Model with Attacker Type as listed in Table 1

content manipulation attack in the absence of a cryptographic security protocol. As a result, OC4 and OC5 are compromised.

3) Router 4 is a suitable (external man-in-the-middle) attack point from which to perform a packet content manipulation attack in the absence of a cryptographic security protocol. As a result, OC6 and OC7 are compromised.

PTP safeguards have the following impact:

1) Cryptographic security protocols can deter the simple internal attacker (i.e., yellow stars number 1), but the advanced internal man-in-the-middle attacker (i.e., red stars number 1) has legitimate network access and can perform such an attack.
2) Multiple paths cannot deter such attacks, especially if the manipulated packet arrives faster than the others in the case of the HSR approach, or if all the intermediate nodes were attacked by a man-in-the-middle attacker.
3) A redundant GM cannot mitigate packet content manipulation, as a man-in-the-middle attacker can manipulate all packets regardless of the sender. In other words, if the passive GM2 (see Fig. 2) recognizes that it has better accuracy than the active GM1, it will become the active GM, but the attacker can manipulate its messages in the same manner.
4) With protocol redundancy, NTP messages are also vulnerable to a packet content manipulation attack by a man-in-the-middle attacker.

Packet removal attack

Attack overview

In a packet removal attack, protocol packets are intercepted and removed by a *man-in-the-middle attacker*, which again either leads to clock synchronization errors of all clocks downstream or makes them go into free-running mode. An internal (and an external) man-in-the-middle attacker can perform such an attack, as it only requires them to reside in an intermediate node, regardless of whether the attacker has access to the authentication/encryption keys (Mizrahi 2014).

Attack implementation

Most of the intermediate nodes (Router1, Router2, Router3, Router4, TC1, TC2, and TC3) are points from which to launch a packet removal attack (red stars number 2, as shown in Fig. 2); for example:

1) TC1 (advanced internal man-in-the-middle attack point) in Fig. 2 can selectively intercept and remove PTP messages (i.e., delay request messages only), causing degradation of OC2 and OC3 synchronization. TC1 also can remove *all* PTP messages, forcing OC2 and OC3 to go into free-running mode.
2) TC2 (simple internal man-in-the-middle attack point) in Fig. 2 can launch a similar attack to OC4 as in example 1, but since it cannot distinguish between encrypted PTP packets and other network traffic, it would randomly remove messages to/from OC5. TC2 would certainly not block all OC5 traffic, as this could be easily spotted by the slave. Instead, packets have to be removed more subtly, so that

the TCP retransmission mechanism compensates for packet loss of other affected network services.
3) Router4 (external man-in-the-middle attack point) in Fig. 2 can randomly or systematically drop PTP messages to/from OC6/OC7, causing either a slave clock synchronization degradation or a switch into free-running mode.

PTP safeguards have the following impact:

1) Cryptographic security protocols cannot protect against packet loss.
2) Multiple paths can mitigate such an attack unless all intermediate nodes are simultaneously manipulated by a man-in-the-middle attacker.
3) A redundant GM cannot mitigate such an attack, as it would be targeted as well, once it is active.
4) In protocol redundancy, NTP messages are also vulnerable to a packet removal attack by a man-in-the-middle attacker.

Packet delay manipulation attack

Attack overview

IEEE 1588 requires symmetric network delays between master and slave in order to achieve optimal clock synchronization (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2008). If the time propagation delays of a sync message and its corresponding delay request message are not equal, the slave clock will calculate an inaccurate offset. A packet delay manipulation occurs when the transmission of protocol packets is purposely delayed by a *man-in-the-middle attacker* (Mizrahi 2011). As a result, all clocks downstream from the attacker location will be manipulated. An internal (and even external) man-in-the-middle attacker can perform such an attack, as it only requires them to reside in an intermediate node without having access to the authentication/encryption keys used (Mizrahi 2014).

Attack implementation

A packet delay manipulation attacker can use an intermediate node to selectively hold PTP packets for a certain time before forwarding them to their destination. Such an attack must happen in one direction only (uplink or downlink) to produce an asymmetric delay between the master and slave.

Large instantaneous delays cause large slave clock offset errors and are likely to be picked up by PTP slave daemons, so incremental delay over time must be used.

Most of the intermediate nodes (Router1, Router2, Router3, Router4, TC1, TC2, and TC3) are points from which to launch a packet delay manipulation attack (red stars number 3, as shown in Fig. 2), for example:

1) TC1 (advanced internal man-in-the-middle attack point) can repeatedly delay *all* Sync or Delay_Req messages, resulting in an asymmetric path delay between the master and its slaves. As a result, there is a degradation of the synchronization of both OC2 and OC3.
2) TC2 (simple internal man-in-the-middle attack point) can similarly attack OC4 and OC5 by delaying *all* packets that go towards or come from these endpoints.
3) Router4 (external man-in-the-middle attack point) can launch a packet delay manipulation attack on OC6 and OC7.

PTP safeguards have the following impact:

1) Cryptographic protocols do not guarantee that messages will be delivered to their destinations in a fixed or deterministic time.
2) Multiple paths can mitigate such an attack unless the intermediate nodes along all network paths delay PTP packets synchronously.
3) The same applies to protocol redundancy, where NTP packets (coming from multiple time servers) are synchronously delayed on their way to the host.
4) GM redundancy cannot address this problem.

Time source degradation attack

Attack overview

Time source attacks occur when an *internal injector attacker* compromises the precise time source of the master clock, i.e., GM or BC, as shown in Fig. 2. Subsequently, all clocks downstream are manipulated.

Attack implementation

GM1, GM2, and BC are targets of such an advanced internal injector attack, for example:

1) Since GPS is usually used as a network time reference, an attacker can jam or spoof the satellite signals, causing the grandmaster clock to become an incorrect reference time (Mizrahi 2014).
2) An attacker can target GM1 (the active GM in Fig. 2) by manipulating its firmware. Subsequently, GM1 provides inaccurate timestamps to all PTP nodes causing degradation of synchronization.
3) An attacker can manipulate the BC in the same manner as in example 2. As a result, all PTP slave clocks downstream will be manipulated.

PTP safeguards have the following impact:

1) Cryptographic security protocols cannot prevent the degradation of the time source.

2) Multiple paths do not provide a solution either, since the attack occurs at the endpoint of a network (the BC will act as an endpoint for all PTP messages).
3) All redundant active / passive GMs can be simultaneously compromised.
4) Protocol redundancy can mitigate such attacks unless NTP synchronization is interrupted or manipulated as well.

Master spoofing attack

Attack overview

In a master spoofing attack, an *injector attacker* is depicted as a legitimate master by generating and transmitting PTP packets (Mizrahi 2014). The attacker impersonates the master clock and distributes false synchronization messages, causing all clocks downstream to be compromised.

Attack implementation

All non-master PTP nodes are suitably located to launch a master spoofing attack (yellow and red stars number 5, as shown in Fig. 2); for example:

1) TC1 (advanced internal injector attack point) in Fig. 2 can masquerade as the master BC by using its IP address, continuously generate manipulated Sync/Follow_Up packets, and send them to OC2/OC3.
2) OC1 (advanced internal injector attack point) can similarly masquerade as an active GM (GM1) and send manipulated Sync/Follow_Up packets to BC. As a result, BC as well as all nodes downstream (OC2 to OC7) will be affected. Note that this attack can only occur if no cryptographic security (i.e., MACsec) is applied.
3) TC2 (simple internal injector attack point) can continuously send spoofed Announce/Sync messages to OC4 and OC5, if no cryptographic security protocol (i.e., MACsec) is used. As a result, OC4 and OC5 will be compromised (DeCusatis et al. 2019).
4) Router4 or OC8 (external injector attack points) can similarly attack networks 2 and 3, if no cryptographic security protocol (i.e., IPsec) is used. As a result, OC6 and OC7 will be manipulated.

Note that an attacker can send malicious messages from an active GM or BC as a time source degradation attack rather than a master spoofing attack.

PTP safeguards have the following impact:

1) Cryptographic security protocols cannot prevent such an attack, if the spoofed messages use the same security keys and come from a trusted intermediate node (red stars number 5, as shown in Fig. 2).
2) In the multiple paths approach, all intermediate nodes can be simultaneously attacked and send orchestrated spoofed master messages.
3) A redundant GM cannot mitigate such attacks, as the attacker can spoof any active GM.
4) With protocol redundancy, NTP can mitigate such attacks if it is not otherwise manipulated.

Slave spoofing attack

Attack overview

In a slave spoofing attack, an *injector attacker* masquerades as the target (a legitimate intermediate or a slave clock) and transmits delay request messages to the master sooner than the attacked node. The master responds to the spoofed node, which in turn calculates its delay using incorrect timestamps (Mizrahi 2014). Note that if the slave receives a spoofed delay response message with a sequence number that does not match its last delay request message, the response message will be discarded, and this attack attempt fails.

Attack implementation

All PTP nodes (except the active GM and the BCs) are suitably located to launch slave spoofing attacks (yellow and red stars number 6, as shown in Fig. 2); for example:

1) Router1 (an advanced internal injector attack point) can continuously create spoofed delay request packets using OC6's or OC7's IP address and their expected sequence numbers and send them to BC. As a result, OC6 and OC7 will be manipulated because of the asymmetric uplink/downlink path between the master and the slave.
2) TC2 or OC4 (simple internal injector attack points) can similarly attack OC5, but only if no cryptographic security protocol (i.e., MACsec) is used.
3) Likewise, Router4 or OC8 (external injector attack points) can attack OC6 and OC7, as long as no cryptographic security protocol (i.e., IPsec) is used.

PTP safeguards have the following impact:

1) Cryptographic security protocols cannot prevent such an attack, if the spoofed messages use the same security keys and come from a trusted intermediate node (red stars number 6, as shown in Fig. 2).
2) In the multiple paths approach, multiple intermediate nodes along all paths between the master and a slave can be simultaneously manipulated and send spoofed delay request

messages to the master in order to produce an asymmetric delay.
3) A redundant GM cannot mitigate such an attack as there is no reason for the passive GM to take action.
4) In protocol redundancy, NTP can mitigate such an attack as long as it is not separately manipulated.

Replay attack

Attack overview

In a replay attack, an *internal* (or even *external*) *injector/man-in-the-middle attacker* continuously records protocol packets and transmits them later without modification.

Attack implementation

All network nodes are suitably located to launch a replay attack (yellow stars number 7, as shown in Fig. 2); for example:

1) GM2 (advanced internal injector attack point) in Fig. 2 can replay multicast Sync/Follow_Up messages from GM1. As a result, all nodes downstream will be compromised.
2) OC4 (simple internal injector attack point) can replay multicast Sync/Follow_Up messages from BC and replay them later to OC5. As a result, OC5 will be manipulated.
3) Router4 or OC8 (external injector attack points) can similarly compromise OC6 and OC7.

PTP safeguards have the following impact:

1) All cryptographic security protocols have a replay protection mechanism (based on a sequence number field), protecting against such an attack (Mizrahi 2014; Stallings 2006).
2) With the multiple paths approach, intermediate nodes along all paths between the master and a slave can be simultaneously manipulated and record and resend later Sync/ Follow_Up messages to the slaves in order to manipulate the time synchronization. Moreover, the replay attack can also be performed by an injector attacker rather than a man-in-the-middle (i.e., a different slave), which cannot be avoided by the multiple paths approach.
3) A redundant GM cannot mitigate such attacks as the attacker can record and replay packets from any active GM.
4) In protocol redundancy, NTP is also vulnerable to the replay attack.

BMCA attack

Attack overview

In a BMCA attack, an *advanced internal attacker* guides other network clocks to elect it as the best master by tampering with the BMC algorithm. Here the BMCA attacker does not fake its identity but tampers with the master election process by advertising exaggerated and incorrect clock characteristics (Mizrahi 2014), and – once elected – manipulates the synchronization of all slave clocks.

Attack implementation

All PTP nodes are suitably located to host a BMCA attack (yellow and red stars number 8, as shown in Fig. 2), for example:

1) OC1 (advanced internal injector attack point) becomes a rogue master. It subsequently sends continuously crafted announce messages that carry the best clock attributes (i.e., *priority1, clockClass, clockAccuracy, offsetScaledLog-Variance, priority2,* and *clockIdentity*) of the entire network to tamper with the BMC algorithm, as explained in (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 2008). As a result, all nodes downstream (OC2 to OC7 and BC) will rely on this compromised time reference.
2) OC4 (simple internal injector attack point) can launch this attack if no cryptographic security protocol is present. As a result, all nodes downstream (BC and OC1 to OC7 excluding OC4) will rely on an inaccurate time source.
3) Router4 or OC8 (external injector attack points) can launch this attack if no cryptographic security protocol is present. As a result, OC1 to OC7 and BC will be manipulated.

PTP safeguards have the following impact:

1) Cryptographic security protocols can only stop an external or simple internal injector attacker. An advanced internal injector attacker can stealthily perform such an attack.
2) The multiple paths approach cannot prevent this manipulation as the attacker can infiltrate an endpoint to become the rogue grandmaster.
3) A redundant GM cannot mitigate such an attack, assuming that a rogue master *always* has better clock attributes than the other grandmasters.
4) In protocol redundancy, NTP can mitigate such an attack, unless it is separately compromised or disabled.

Denial of service attacks

Attack overview

A denial of service attack can be initiated by an *injector attacker*. There are many potential Layer 2 and Layer 3 DoS or DDoS attacks, such as MAC flooding, ARP spoofing, and IP spoofing, which compromise the target's availability and timely execution of the PTP protocol (Mizrahi 2014). In addition, an attacker can utilize cryptographic execution attacks by sending bogus IPsec or MACsec packets, which cause a high CPU load when the receiver's cryptographic engine tries to verify the validity of these packets. This attack can be launched by any internal (and even external) attacker (Mizrahi 2014), and forces all affected clocks to go into free-running mode.

Attack implementation

All PTP nodes are suitably located to launch a DoS attack (red stars number 9, as shown in Fig. 2; for example:

1) OC2 (advanced internal injector attack point) in Fig. 2 performs an ARP spoofing attack to bind its MAC address to OC3's IP address. As a result, OC3 cannot receive PTP messages and eventually goes into free-running clock mode.
2) OC4 (simple internal injector attack point) can launch a DoS attack by continuously transmitting protocol packets using a fake security key to OC5, which causes a high utilization of OC5's cryptographic engine. As a result, OC5 cannot process other PTP messages in time and goes into free-running clock mode.
3) Router4 or OC8 (external injector attack points) can launch a DoS attack as described in examples 1 and 2 in order to manipulate all slaves in networks 2 and 3. As a result, OC6 and OC7 will go into free-running mode.

PTP safeguards have the following impact:

1) Cryptographic security protocols cannot prevent, but may even support (D)DoS-style attacks, as shown in example 2.
2) The multiple paths approach fails if all interfaces of an endpoint are targeted.
3) A redundant GM cannot address this issue, as the attacker aims to compromise the slave availability rather than the existing GMs.
4) In protocol redundancy, NTP is also vulnerable to a DoS attack.

Experimental validation of attacks

Table 2 shows the potential impact of the various attack strategies outlined in the previous sections. From a slave clock perspective, the most effective attack that directly manipulates all clock synchronization downstream from the physical location of the attacker in the network is represented by the label "Clock Manipulation". In contrast, "Clock free-running" indicates that all downstream clocks go into free-running (non-PTP synchronized) mode, which is usually not picked up by a host operating system and causes a slow desynchronization over time as outlined in Section I. This table also shows the various attack strategies (i.e., simple/advanced attack, and man-in-the-middle /injector) that can be applied for each strategy, and their severity using the RAG rating: The red color indicates that the PTP safeguards, as listed in Section III do not provide protection, while the yellow color indicates that a given attack strategy can be detected. The green color indicates that the attack can be averted by the PTP safeguard

Continuing on from this work, a testbed was set up to simulate and experimentally validate some attack strategies (i.e., time source degradation, packet content manipulation, packet delay manipulation, replay, and DoS attack) that have a different impact on PTP slave(s). The testbed (see Fig. 3) consists of three slaves (OC - Raspberry Pi 3 model B), three transparent clocks (TC - Hirschmann RSP20), one grandmaster clock (GM - OMICRON OTMC 100), and one reference clock (OMICRON OTMC 100). The experiments were done using the PTP slave daemon PTPd. The reference clock provided an accurate time reference (similar to the grandmaster clock in normal operation - no attack) but it does not participate in the time synchronization process and it is assumed to be secure and outside the attack scope. It also collected timestamps from all other clocks in the network and subsequently computed the time drift of these clocks by calculating the difference between its timestamps and the timestamps received from the other slaves minus the time taken to transfer these timestamps from the slaves to the reference clock. All devices in the network are connected via CAT5e Ethernet cables with a data rate of 1000 Mbps.

In detail, an attacker has the following options:

1) Desynchronize all PTP clocks downstream via a BMCA- or time source degradation attack: These approaches exploit the grandmaster's role as a time source and propagate an inaccurate time reference to all other clocks in a network. Like all the other attacks presented, it has to be persistent to continuously manipulate PTP clocks. This attack was performed by attaching a new OC device to the network that advertises itself as the best clock. Figure 4 shows the possible impact of such an attack on PTP slaves. In this experiment, master timestamps are given an increasing negative offset of 100 μs per second, before being circulated to the

Table 2 PTP Attack Strategies and their Impacts

NO	Attack Strategy	Attack Impact	Impact Scope	Internal Attack Type (the RAG rating is used to highlight the severity)			
				Cryptographic Security	Multiple Paths	Redundant GM	Protocol Redundancy
1	Packet Content Manipulation	Clock Manipulation or Clock free-running	Based on attacker location All/subset/single slave(s)	Advanced man-in-the-middle	Simple man-in-the-middle	Simple man-in-the-middle	Simple man-in-the-middle
2	Packet Removal	Clock Manipulation or Clock free-running	Based on attacker location All/subset/single slave(s)	Simple man-in-the-middle	Simple man-in-the-middle	Simple man-in-the-middle	Simple man-in-the-middle
3	Packet Delay Manipulation	Clock Manipulation	Based on attacker location All/subset/single slave(s)	Simple man-in-the-middle	Simple man-in-the-middle	Simple man-in-the-middle	Simple man-in-the-middle
4	Time Source Degradation	Clock Manipulation	All slaves	Advanced Injector	Simple Injector	Simple Injector	Simple Injector
5	Master Spoofing	Clock Manipulation	Based on attacker location All/subset/single slave(s)	Advanced man-in-the-middle	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)
6	Slave Spoofing	Clock Manipulation	Single slave	Advanced man-in-the-middle	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)
7	Replay	Clock Manipulation	Based on attacker location All/subset/single slave(s)	N/A	Simple (Injector/ man-in-the-middle)	Simple (Injector / man-in-the-middle)	Simple (Injector/ man-in-the-middle)
8	BMCA	Clock Manipulation	All slaves	Advanced (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)
9	Denial of Service	Clock free-running	Based on attacker location All/subset/single slave(s)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)	Simple (Injector/ man-in-the-middle)

slaves via Sync messages. Similarly, the BMCA attack would have the same impact on the slave clock when the difference of the clocks frequencies (the new master and the reference clock) introduces a 100 μs time offset per second.

2) Manipulate a subset of PTP clocks by using packet content manipulation or packet delay manipulation strategies: Fig. 5 shows the impact of a packet content manipulation attack when an attacker intercepts either Sync or Follow_Up messages and

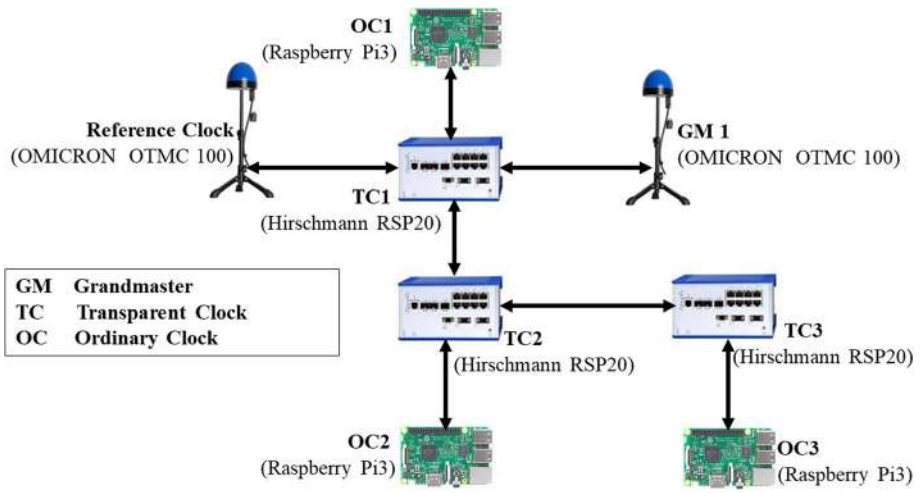

Fig. 3 Testbed

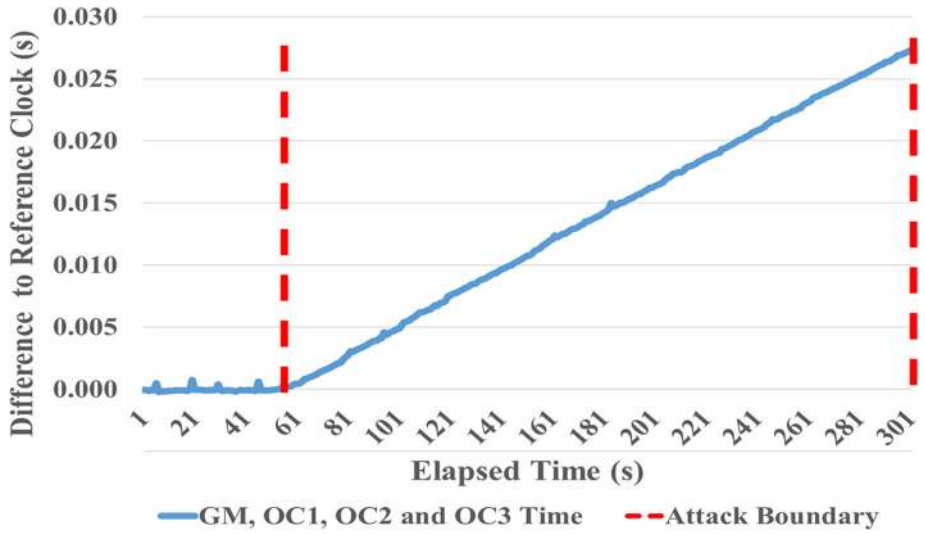

Fig. 4 Impact of BMCA or time source degradation attack on slaves

decreases their timestamps by a value that is incremented by 100 μs per second. Similarly, Fig. 6 shows the impact of asymmetric delay attack when an attacker intercepts each Sync message and holds it for 20 ms, before forwarding it to its destination. Since many applications require a smooth and monotonically increasing time base, PTP daemons were designed to take this feature into account, especially when the time error introduced is within the preconfigured threshold. Figure 6 showed that the PTP daemon increased the slave clock frequency gradually, starting from 60 s into the experiment, due to the introduced asymmetric delay, to meet the master clock frequency. At the 100th second, the PTP daemon realized that the slave clock frequency became faster than the master clock frequency and subsequently decreased the slave clock frequency gradually until both clock frequencies were close to each other. This experiment was conducted by adding a network impairment emulator device between the GM and TC1 to affect all slaves, or between TCs to affect some slaves (see Fig. 3). The emulator device is able to intercept and delay/manipulate the content of specific packets (i.e., Sync/Follow_Up messages) and then forward them to their destination.

3) Interfere with the clock synchronization process via master spoofing or replay attacks: With each of

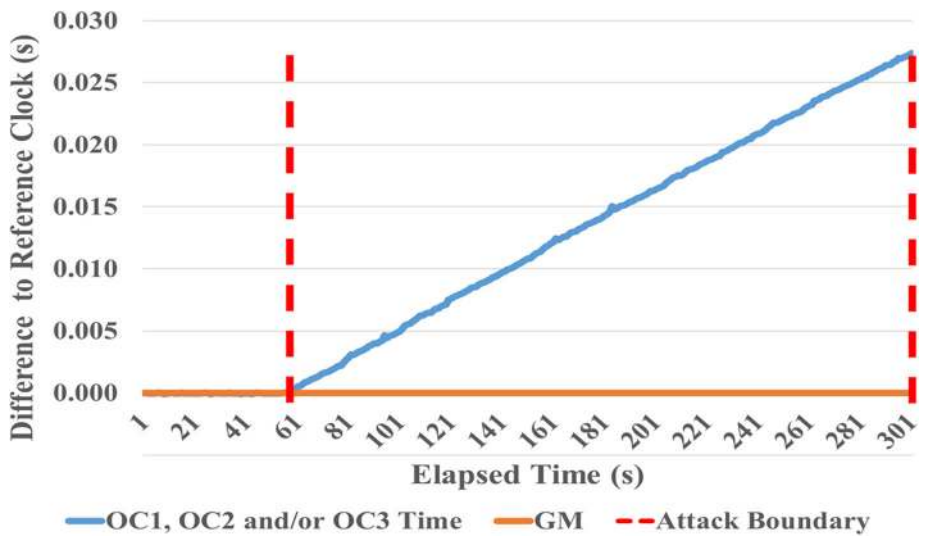

Fig. 5 Impact of packet content manipulation attack

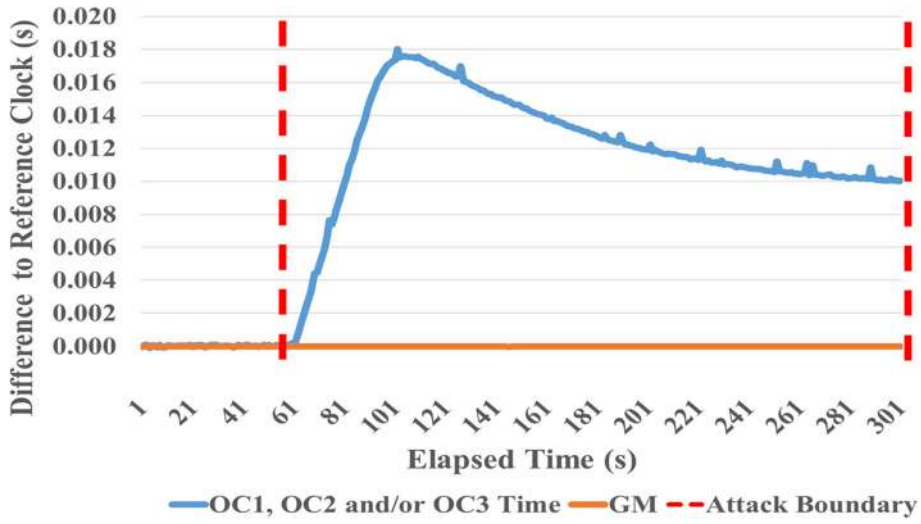

Fig. 6 Impact of asymmetric packet delay manipulation attack

these, a slave may receive valid and fresh sync messages as well as spoofed or replayed sync messages over time, making it swing between a synced and an unsynced state to the master. Figure 7 shows the impact of a replay attack when an attacker records the last Sync/Follow_Up messages sent by the master every 5 s and replays them to their destination. Such an attack can be performed by any of the existing slaves. It is worth noting that the PTP daemon in this experiment applied a clock reset instead of gradually adjusting the clock frequency, because the introduced time error exceeded the preconfigured threshold (i.e., if the time error is greater than 1 s).

4) Target a single PTP clock and manipulate it by using the slave spoofing attack: This has the lowest impact on a PTP network.
5) Launch a DoS attack or packet removal attack that makes affected slave clocks go into free-running mode: A denial of service can be relatively easily detected by a network or a slave, as it affects all network services. Figure 8 shows the impact of a DoS attack when an attacker prevents slave(s) from receiving the PTP messages. Here, the PTP slave clock will be in free-running mode and subsequently its frequency will be unstable, making its time ahead of and sometimes behind the reference clock. The emulator device is used here again to

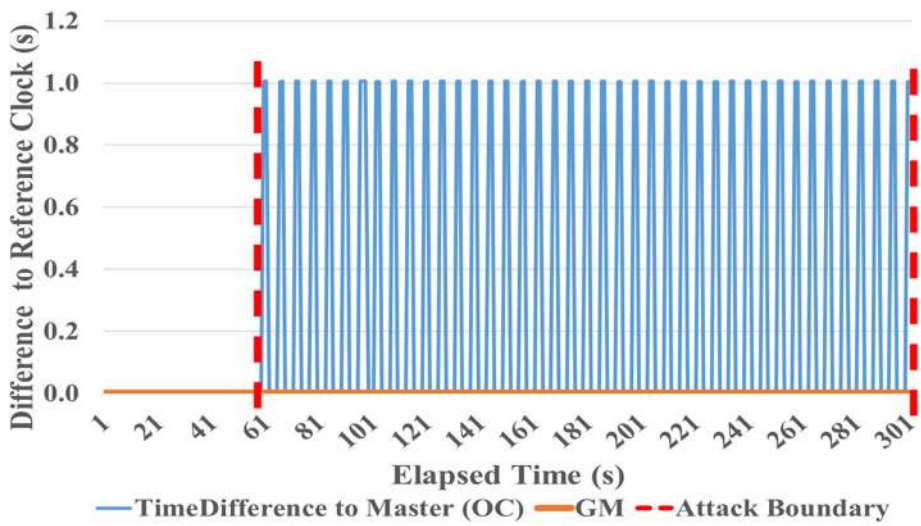

Fig. 7 Impact of replay attack

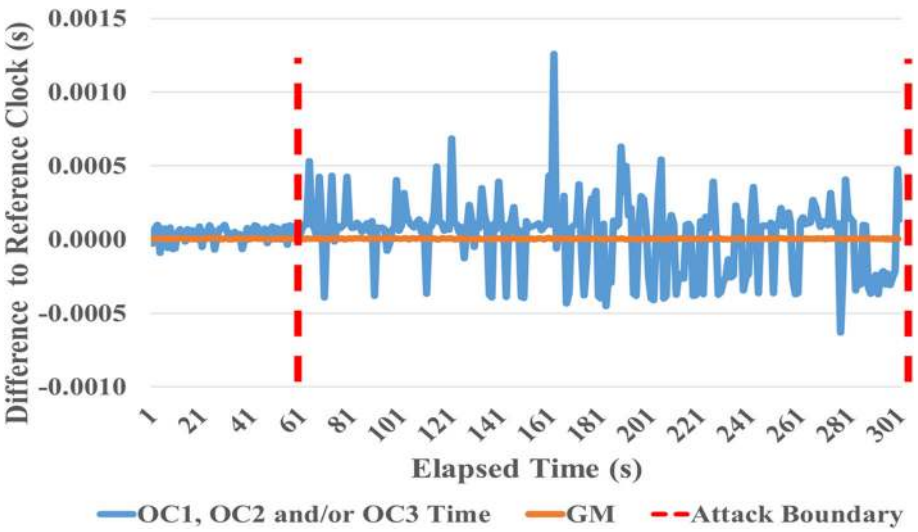

Fig. 8 Impact of DoS attack/packet removal attack

intercept and remove the Sync message, preventing them from being received by the slaves, which subsequently go into free-running mode.

Proposed security extension

The previous sections have shown that traditional protection methods cannot stop APTs for slave clock desynchronization. Therefore we propose to introduce a monitor unit called the trusted supervisor node (TSN) as shown in Fig. 9, which is able to monitor and analyze synchronized time packets sent by the master, as well as compare clock offsets provided by a large number of slave devices. The underlying concept is that although individual slave clocks are intrinsically inaccurate and likely to drift, a group of slaves might show a statistically significant deviation in their offsets if they are exposed to manipulated time packets (Alghamdi and Schukat 2017). Such a proposed method requires the existence of a management node to collect synchronization outputs (i.e., offset, delay, and frequency) from all slaves and compare these outputs with each other using a time series analysis. When the management node observes that some or all slaves start reporting abnormal values, an alarm would be raised to notify the network administrator about the affected clocks. Thus the proposed method provides a detection system rather than a prevention system. A similar approach has been suggested by (Moussa et al. 2020), but it lacks resilience against

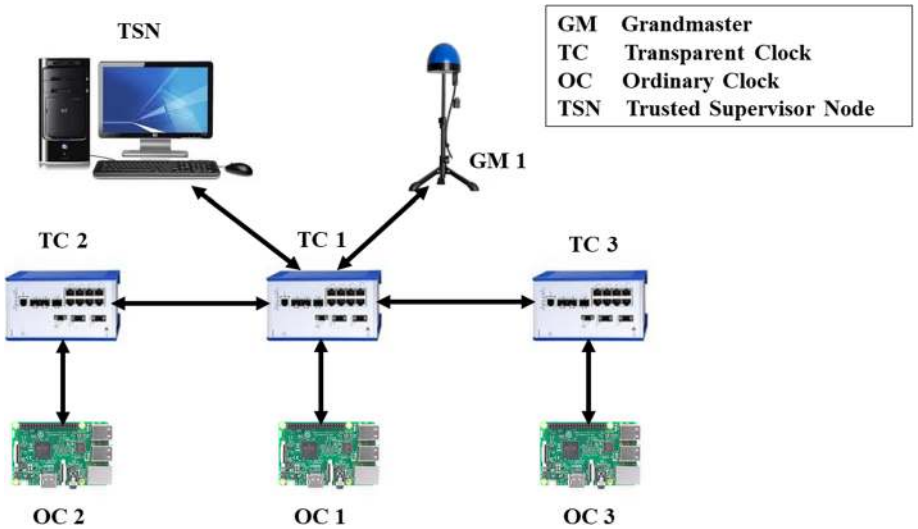

Fig. 9 Trusted Supervisor Node

targeted cyber-attacks, as it uses slave timestamps and requires deterministic network latencies in order to work. The proposed method will build on the ideas of Annex P Prong D. The time source degradation attack may show some resistance against the proposed security method, but this is still under investigation.

Conclusion and future work

This paper investigates the problem of advanced persistent threats to PTP networks. It distinguishes between attack strategies and attacker types as described in RFC7384, but further distinguishes between the spoofing and time source attack, the simple internal attack, and the advanced internal attack. This research takes into account the new security features of the emerging Annex P.

Our analysis shows that an internal attacker has a range of methodologies to compromise the time synchronization of PTP slaves, ranging from slave spoofing that targets individual slaves, to BCMA attacks that compromise all endpoints in a network. While prior research has validated that cryptographic security via MACsec or IPsec is a blunt instrument against most internal attacks, the previous sections have shown that infrastructure or protocol redundancy does not provide viable protection either. Moreover, all PTP infrastructure components (GM, BC, and TC) and even slave clocks (e.g., ordinary personal computers) can host an attacker, as shown in Fig. 2. This makes the comprehensive protection of a PTP network against malware infiltration, as for example exercised by Stuxnet, a very tedious task.

While this paper also presents some experimental findings with regard to attack implementation and their impact, further research will explore these threats in more depth. We are particularly interested in the exact behavior of different PTP client daemons in the presence of the aforementioned attacks using different parameters. These results will help us to reach our long-term goal, a PTP intrusion detection system based on the aforementioned trusted supervisor node to protect time synchronization networks against APTs.

Acknowledgments
Not applicable.

Authors' contributions
Waleed conducted the analysis and the experiments and the Michael read and approved the final manuscript. The author(s) read and approved the final manuscript.

Authors' information
Not applicable.

Funding
This work was supported by the Technical and Vocational Training Corporation, Saudi Arabia.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 11 February 2020 Accepted: 8 February 2021
Published online: 01 April 2021

References

Alghamdi W, Schukat M (2017) Advanced methodologies to deter internal attacks in PTP time synchronization networks. In: 28th Irish signals and systems conference (ISSC), Killarney, 20-21 June 2017. pp 1-6. IEEE
Baize E (2012) Developing secure products in the age of advanced persistent threats. IEEE Sec & Priv 10:88–92. https://doi.org/10.1109/MSP.2012.65
Chen D (2013) Secure 1588 in HeNB / Femtocell application. Paper presented at the Time & Sync in Telecoms, Lisbon
Chen T, Abu-Nimeh S (2011) Lessons from stuxnet. Comp 44:91–93
Cho DX, Nam HH (2019) A method of monitoring and detecting APT attacks based on unknown domains. Proc Comp Sci 150:316–323
Dadheech K, Choudhary A, Bhatia G (2018, 2018) De-militarized zone: a next level to network security. In: Second international conference on inventive communication and computational technologies (ICICCT), Coimbatore, pp 595–600
Dalmas M, Rachadel H, Silvano G, Dutra C (2015, 2015) Improving PTP robustness to the byzantine failure. In: IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS), Beijing, pp 111–114
DeCusatis C, Lynch RM, Kluge W, Houston J, Wojciak P, Guendert S (2019) Impact of Cyberattacks on precision time protocol. IEEE Trans Inst Meas 69:2172–2181. https://doi.org/10.1109/TIM.2019.2918597
Donoghue KO, Sibold D, Fries S (2017) New security mechanisms for network time synchronization protocols. In: IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS), California, pp 1–6
Estrela PV, Neusüß S, Owczarek W (2014, 2014) Using a multi-source NTP watchdog to increase the robustness of PTPv2 in financial industry networks. In: Precision clock synchronization for measurement, control, and communication (ISPCS), IEEE international symposium on, Austin. IEEE, pp 87–92
Garner GM (2008) IEEE 1588 Version 2, vol 8. ISPCS, Ann Arbor, pp 1–89
IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (2008) IEEE Std 1588–2008 (Revision of IEEE Std 1588–2002), pp 1–269. https://doi.org/10.1109/IEEESTD.2008.4579760
IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (2020) IEEE Std 1588–2019 (Revision ofIEEE Std 1588–2008), pp 1–499
IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security (2006) IEEE Std 802.1AE-2006, pp 1–150. https://doi.org/10.1109/IEEESTD.2006.245590
Itkin E, Wool A (2020) A security analysis and revised security extension for the precision time protocol. IEEE Trans Dep Sec Com 17:22–34. https://doi.org/10.1109/TDSC.2017.2748583
Knapp ED, Langill JT (2015) Industrial network security: securing critical infrastructure networks for smart grid, SCADA, and other industrial control systems
Koskiahde T, Kujala J (2016) PTP monitoring in redundant network. In: IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS), Stockholm, pp 1–5
Langner R (2011) Stuxnet: dissecting a cyberwarfare weapon. IEEE Sec Priv 9:49–51
Mills DL (1991) Internet time synchronization: the network time protocol. IEEE Trans Comm 39:1482–1493
Mizrahi T (2011) Time synchronization security using IPsec and MACsec. In: Proceedings of the International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication, Munich, pp 38–43
Mizrahi T (2014) Security requirements of time protocols in packet switched networks. In: RFC 7384 https://tools.ietf.org/html/rfc7384
Moussa B, Kassouf M, Hadjidj R, Debbabi M, Assi C (2020) An extension to the precision time protocol (PTP) to enable the detection of cyber attacks. IEEE Trans Ind Info 16:18–27. https://doi.org/10.1109/TII.2019.2943913
Neyer J, Gassner L, Marinescu C (2019) Redundant schemes or how to counter the delay attack on time synchronization protocols. In: IEEE international

symposium on precision clock synchronization for measurement, control, and communication (ISPCS), Portland, pp 1–6
Önal C, Kirrmann H (2012) Security improvements for IEEE 1588 annex K: implementation and comparison of authentication codes. In: International IEEE symposium on precision clock synchronization for measurement control and communication (ISPCS). IEEE, San Francisco, pp 1–6
Pathan Y, Dalvi A, Pillai A, Patil D, Reed D (2014) Analysis of selective packet delay attack on IEEE 1588 precision time protocol. Technical Report, University of Colorado at Boulder
Quintero-Bonilla S, Martín del Rey A (2020) A new proposal on the advanced persistent threat: a survey. App Scie 10:3874
Shannon J (2013) Improved techniques for time synchronization over WiFi and wireless sensor networks. Dissertation, National University of Ireland, Galway
Shannon J, Melvin H, Ruzzelli AG (2012) Dynamic flooding time synchronization protocol for WSNs. In: IEEE global communications conference (GLOBECOM). IEEE, Anaheim, pp 365–371
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: IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS), Portland, pp 1–6
Shpiner A, Revah Y, Mizrahi T (2013) Multi-path Time Protocols. In: IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS) Proceedings, Lemgo, pp 1–6
Stallings W (2006) Cryptography and network security: principles and practices. New Jersey: Pearson Education India
Vacca JR (2017) Computer and information security handbook. San Francisco: Morgan Kaufmann

Publisher's Note