Attack Strategies and Countermeasures in Transport-Based Time Synchronization Solutions

Attack strategies and countermeasures in transport-based time synchronization solutions

Diana Gratiela Berbecaru, Antonio Lioy

Abstract The security, availability, and accuracy of time information transmitted over transport networks are getting increased attention since more applications in different domains require secure and accurate time. In this paper, we classify security attacks affecting the transport-based time synchronization architectures, such as the one currently designed and developed in the ROOT (Rolling Out OSNMA for the Secure Synchronization of Telecom Networks) project. We indicate security attack classes applying to different views of the architecture, namely time distribution, network management, hardware, and software views. We then concentrate on the software view and indicate preliminary results we have obtained by experimenting with software tampering attacks on a dedicated device employed for time distribution. As a countermeasure against such attacks, we exploit the Trusted Platform Module and specialized software for remote attestation, which has the ability to verify that the mentioned device remains in a good state for the duration of its computation. We consider these tests as a first step toward deploying software integrity controls on the specialized nodes handling time synchronization in the ROOT project.

Key words: cybersecurity, time synchronization, security attacks, remote attestation

Diana Gratiela Berbecaru
Politecnico di Torino, Dip. di Automatica e Informatica, Corso Duca degli Abruzzi 24, 10129, Torino (ITALY), e-mail: diana.berbecaru@polito.it

Antonio Lioy
Politecnico di Torino, Dip. di Automatica e Informatica, Corso Duca degli Abruzzi 24, 10129, Torino (ITALY), e-mail: lioy@polito.it

1 Introduction

Smart industry, power grids, vehicular applications, critical Internet of Things, industrial automation services, as well as finance and banking applications are increasingly requiring accurate and secure time synchronization solutions. Such solutions have long been a fundamental prerequisite for the operation of telecommunications networks, and will be more important than ever in 5G networks. The main 5G synchronization requirements and solutions have been sketched in [1] and are currently studied in the research project ROOT [2], which aims to experiment different types of cyberattacks and evaluate their impact on a transport-based time distribution architecture.

The emerging solution for time synchronization in Time-sensitive networks (TSNs) is one that exploits GNSS (Global Navigation Satellite System) services and transport domain networks. GNSS services include Galileo, GPS and BDS (BeiDou Navigation Satellite System) and are offered by several space agencies. The distribution of time over transport networks has been addressed in several ITU-T recommendations [3]. These recommendations define reference synchronization networks, where the synchronization is created by Enhanced Primary Reference Time Clocks (ePRTCs), which are typically based on GNSS technology and terrestrial accurate clocks, and where the reference timing signal is carried across a network of clocks. The time synchronization reference is typically carried through the PTP (Precision Time Protocol) protocol [4], or Network Time Protocol (NTP) protocol [5] over packet switched networks. NTP is suitable for large and dynamic networks, as it fulfills the requirements of distributed systems that require an accuracy of a few miliseconds over wide area networks. The PTP instead is designed for TSNs that often use PTP-aware hardware to provide an accuracy of few microseconds and even nanoseconds. PTP distinguishes between different clock types. One of these clocks is the grandmaster clock (GM), which provides the time reference for the network. The ePRTC is typically the source of time for the PTP grandmaster. The other ordinary clocks are called slaves, and they exchange time synchronization with the GM.

While many papers address the security of PTP protocol [10] [11] [12] or even the more accurate profile White Rabbit PTP (WR-PTP) protocol [16] [6], we found that less work has been done on the classification of attacks in transport-based time synchronization networks. In particular, an attacker may exploit multiple attack vectors. The attacks can target the network protocols exploited for the time distribution [7], the network elements connected to the time distribution devices, or the software (and the configuration) running on the above-mentioned entities. Moreover, we observe that the devices are operating at different levels in the time synchronization network. The work closest to ours is described in [13], in which both the network attacks as well as the advanced internal attacks targeting the software or the software configuration are being considered. Individuating possible attack points, as well as adequate countermeasures against such attacks are of crucial importance in time synchronization networks. In this work, we analyze the security attacks applying to

a transport-based synchronization network, similar to the ones currently considered in the ROOT project [15].

2 Transport-based time synchronization solutions

Upcoming telecom networks can respond to time synchronization requirements granting nanosecond accuracy by deploying specialized devices combining GNSS timing receivers with terrestrial clocks (cesium or rubidium) that support specific transport protocols, such as PTP, NTP (Network Time Protocol), or WR-PTP protocols. In [1], the authors describe some possible synchronization solutions, some of them are implemented in the RAN (Radio Access Network) domain, while others are implemented in the transport domain. The focus of this paper is on the latter category of solutions, although, in general, a combination of techniques in both domains would be needed to create a robust and reliable solution. However, as mentioned in [1] "distributing time synchronization over the same transport network infrastructure used for user data has the benefits of providing the same level of robustness and redundancy for timing as for the user traffic itself".

Two PTP profiles, G.8275.1 (PTP with full timing support from the network) and G.8275.2 (PTP with partial timing support from the network), have been defined for the use of PTP in telecom networks [3]. PTP is designed for infrastructure networks that often use PTP-aware hardware, providing clock synchronization accuracy down to microsecond and even nanosecond level.

To achieve high accuracy (such as, from 65 ns to 130 ns for 5G front-haul applications) a high-level network architecture exploiting a centralized master clock has been described in [15], and its functionality is shortly resumed below. A Centralized Gransmaster Clock (C-GMC) generates a time reference by combining different time sources. In particular, the C-GMC uses a GNSS receiver combined with a co-located Cesium Atomic Clock (Cs AC). To provide increased robustness and resilience in case the common time reference is not available due to temporary malfunctioning or intentional attacks, multiple reference clocks are distributed across the network, namely the so-called Distributed GMC (D-GMC). The D-GMCs also combine multiple time sources to generate a time reference, typically a GNSS receiver and a less expensive) Ove-Controlled Cristal Oscillator (OCXO) or Rubidium (Rb) clock. Note also that typically there are at least two C-GMCs (primary and secondary), as well as primary and secondary (backup) D-GMCs, with similar configurations but placed in different physical locations. The time synchronization signal is distributed, e.g. over fiber optical channels, to the devices connected to the network (as shown in Fig. 1). The telecom network operators implement the C-GMC and D-GMC devices at several operational levels. For example, in the architecture shown in [15] and used as reference in Fig. 2, the C-GMC and D-GMC devices occur at the regional level (HL 3) and at the metro aggregation level (HL 4) connected to the network nodes of the telecom operator. At the hierarchical level (HL) 5, which is the most distributed level where mobile base stations connect, operates a simpler time

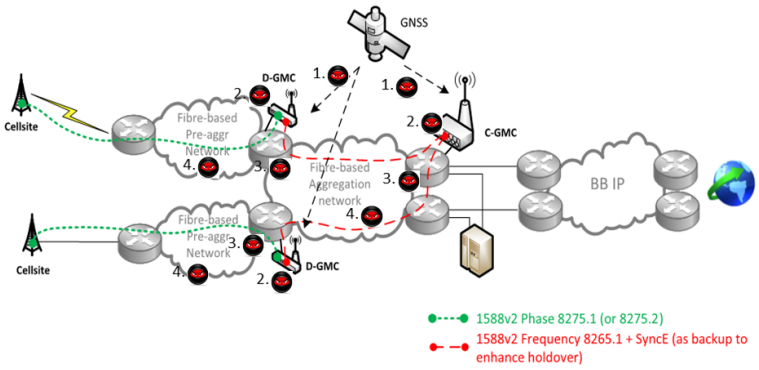

Fig. 1 High-level architecture showing possible phase synchronization distribution, by exploiting C-GMC and D-GMC devices. Possible input attack points targeting the C-GMC and D-GMC devices (both the hardware and running software), their communication with the GNSS satellites, the network nodes (e.g., routers), or the protocols (e.g., PTP or WR-PTP) employed for time distribution are indicated.

distribution device (called BC/GM), which also exploits a GNSS receiver for time calculation, but Cs AC or Rb clocks are not employed at this level.

3 Classification of attacks

Attacks (unintentional or intentional) against time synchronization networks may have a devastating effect. For example, in 2013, a PTP infrastructure glitch forced Eurex (a famous international stock exchange) to postpone its market opening, because an incorrect leap second calculation caused in erroneous synchronization of their critical systems [29]. The cyberattacks have evolved in time, a recent work of Alghamdi and Schukat [26] indicate of particular concern nowadays the advanced persistent threats (APTs) that may affect the PTP networks. The APTs, similar to the Stuxnet attack of 2010 that had a significant impact on industrial control systems [27], target a small number of power users within the target organization with malicious software, for example, malware on secondary memory devices, i.e., USB sticks. Then, they propagate themselves across the organization by exploiting software flaws.

In our work, we classify first the attacks affecting time distribution networks, by considering also other possible attacks points. With respect to the architecture shown in Fig. 1, from a preliminary analysis we observe that an adversary may pick one or a combination of the following attack scenarios:

- attacks against communication between GNSS (satellites) to GNSS-enabled nodes (e.g., D-GCM, C-GCM, . . .) such as:

  - calculate and broadcast a false GNSS signal (spoofing attacks [28]), or retransmit a real but delayed GNSS signal (meaconing attack)
  - denial of service (including jamming attacks [9])
- attacks (hardware or software) against the specific D-GCM or C-GCM devices, such as:
  - physically breach the device, then exploit the direct access to the hardware
  - modify the software running on the device, through malware, memory scraping, side channel attacks, software backdoor, or software tampering attacks
  - modify the configuration of the software running on the device
  - install malicious code/software, either through injection attacks (SQL, XSS) or through worms/trojans, ransomware, rootkits, botnets, or malicious network functions
  - exploitation of buffer overflows (in operating system or software installed)
- attacks targeting the underlying protocols (typically, PTP or WR-PTP) used for the distribution of timing information, such as: denial of service (DoS) attacks, man in the middle (MITM) attacks, replay attacks, or delay attacks [16] [17].
- attacks targeting the underlying network protocols used for the management of the time distribution devices, such as: DoS attacks, MITM attacks, or replay attacks. Examples of protocols employed for this purpose are the Secure Shell (SSH) [18] or Transport Layer Security (TLS) [19] for remote access, the Remote Authentication Dial-In User Service (RADIUS) [20] for authentication, or Simple Network Management Protocol (SNMPv3) [21]. The Internet Control Message (ICMP) protocol is supported typically in any IP-based network, as well as the TCP / UDP transport protocols. Network protocols are prone to manipulation of network configuration parameters or of the security data configuration (cryptography keys, digital certificates, security policies or access rules). Thus, specific protocol may be employed to check their correctness, e.g. OCSP (Online Certificate Status Protocol) protocol used to validate the digital certificates in real time [22]. They might even be wrongly configured or poorly configured, e.g., operate with default authentication credentials, easily guessed by the attackers.

To perform a more accurate security analysis, we have further decomposed the architecture shown in Fig. 1 into several views, trying to individuate the specific attacks applying to each view. In particular, we have considered the time distribution view, the network (management) view, the hardware view, and the software view.

Insider versus Outsider attacks. In classifying the attacks, we have to consider also the attacker's location. We distinguish among Outsider (or External) attacks and Insider (or Internal) attacks.

Outsider attacks are carried out by entities (nodes) which do not belong to the timing distribution or the network managements networks. External attacks can cause serious damage, for example in case the attacker manages to inject false timing data into the network, or if the attacker can inject spurious data into the network to consume network resources and launch a DoS attack. Some attacks are less likely to occur, or even impossible for external attackers. For example, the node tampering

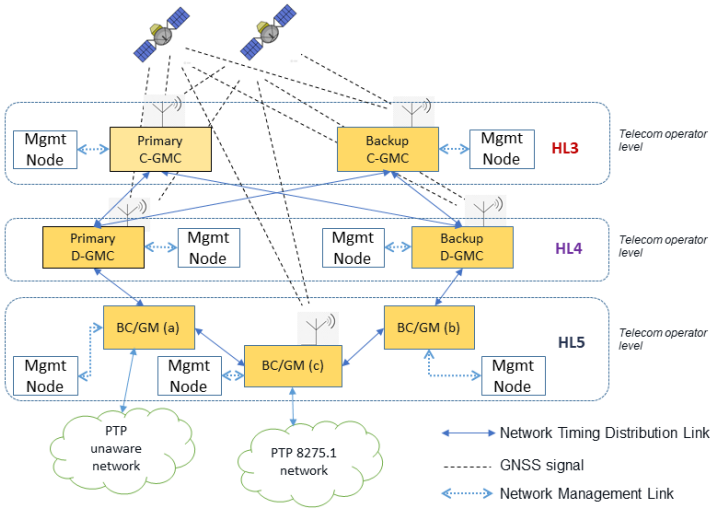

Fig. 2 Timing distribution and network management views used in the security analysis of the transport-based time synchronization architecture.

attack assumes an attacker can have physical access to the device, so it cannot be performed by an external attacker.

Insider attacks occur when an individual or a group within an organization seeks to disrupt operations or exploit organizational assets. It is known that many attacks originate from within the network by authorized users [23]. This may take the form of a disgruntled employee, an abusive administrator, or a user trying to gain access to privileged information. Internal DoS attacks appear also when the genuine nodes of the timing distribution network behave in an unintended way. An insider can cause significant damage if he can inject fake (timing) data into the timing distribution network, or if he launches other types of network attacks, such a DoS attack, or a replay attack.

Time distribution view. This view (shown in Fig. 2) deals with the logical devices and the communications links exploited in the distribution of timing information over the networks. Here we may consider as well the attacks targeting the GNSS signals, e.g., spoofing, jamming [8]. The attacks may target the individual devices at each level (HL3, HL4 or HL5), or the **timing distribution links** and protocols, like PTP or WR-PTP.

Several attacks, including MITM attacks and DoS attacks have been discovered against the NTP and PTP protocols. The attacks against PTP are usually categorized as DoS attacks that might be carried out at various network layers, and MITM attacks, including clock masquerade, replay and filtering attacks [10] [24] [25] [26]. In addition, a novel class of insider threats has been experimentally demonstrated in [10] including two variants of DoS spamming attacks capable to incorrectly steer or permanently skew the slave's clock, as well as a master clock takeover attack.

Network management view. This view (shown also in Fig. 2) is composed of the logical devices and the communications links exploited in the management of the C-GMC and D-GMC devices. The network attacks address the C-GMC and D-GMC devices reachable via the **network management links** from the management node(s) or the communication links themselves.

Note that the network management links are often different from the timing distribution links. They may exploit different physical cables, different input/output ports, and network interfaces of the C-GMC or D-GMC devices. In this context, an attacker could try to perform a DoS attacks against a network management link either by physically interrupting link (in case of an internal attacker) or by exploiting vulnerabilities in the network management communication protocols. An attacker could try to perform interception of the network management traffic exchanged between the management node and the C-GMC or D-GMC device to capture sensitive data, such as usernames and passwords, or change cryptographic keys used in the configuration of the devices. Some examples of attacks against typical protocols employed for network management are for instance the ARP spoofing attack [35], the ping flooding attack [36], the DNS attacks [37], or the TLS attacks [38] [39]. Nowadays, several countermeasures exist to such attacks, including the exploitation of firewalls, intrusion detection systems, and even network forensics to produce evidences with probative value in case of incidents [40].

Hardware view. As indicated in [30], physical security is important because "with regards to attacking an electronic device, any successful physical breach, fundamentally compromises its security. Once an adversary can perform reverse engineering, the security of the device is fundamentally broken". Through reverse engineering an attacker might produce circuits that impair proper functioning of electronic devices, and the manufacture of unlicensed/unapproved duplicates, called "clones" or "ghost devices" [30]. To mitigate reverse engineering risks, one could identify and secure a sufficient set of individual hardware components of the device. This is also the idea behind the Trusted Platform Module (TPM), which a computer chip broadly available in most modern computers and in several motherboards that can securely store data used to authenticate the platform.

Apart from the GNSS antenna input, other interfaces can be considered critical and need to be adequately protected. For example, the PPS (Pulse-Per-Second) interface used by the C-GMC/D-GMC to receive external reference from a GNSS receiver, and the 10 MHz interface to receive external reference from Atomic clocks, or Rb or OCXO oscillators. In this view, we need to also consider the communication medium (physical cables). Both the devices as well as the communication medium are subject to several physical attacks, unintentional (accidental) damages, or failures and malfunctions.

Software view. In this view, we consider the entire software stack running on the C-GMC and D-GMC devices, including the operating system, the management software (such as, for the SNMP, TLS, RADIUS protocols) and the specialized deamons employed for time synchronization, such as `ptpd` [31], `gpsd` [32], or `ppsi` [33]. In general, software attacks - sometimes called low-level attacks - rely on characteristics of the hardware, compiler or operating system used to execute software pro-

grams to make these programs misbehave, or to extract sensitive information from them [34]. The software attacks are getting increased attention nowadays, as well as the countermeasure that can be employed to defend from the software tampering and/or its configuration. Possible solutions capable to detect intentional changes in the software running on the C-GMC or D-GMC devices could exploit the Trusted Computing (TC) principles, for instance by taking advantage of a TPM for secure boot, protected storage for sensitive data (keys) and cryptographic operations, and to perform remote attestation [41].

4 Preliminary experiments with software tampering attacks

On the practical side, we have considered the software view and we have experimented practical attacks against the modification of the software (or its configuration) on a device equipped with TPM. Our final goal is to test the feasibility of remote attestation software on the specialized devices employed in time synchronization.

The Trusted Computing Group establishes the TPM architecture and all its functionalities. Two versions of the TPM have been standardized and adopted, version 1.2 (considered obsolete) and 2.0 released respectively in 2003 and 2014. The TPM has the task of saving aggregates of measures, that will be used in the remote attestation process. The aggregate measurements are stored in special registries in the TPM called Platform Configuration Registers (PCRs). Their primary use is to provide a cryptographic record of the software state. The PCRs can be extended via one-way hash functions (e.g, SHA256), and can be read to report their state through a procedure called *quote*, which is an attestation. A TPM attestation is basically *a proof of the software state* that an attester (sometimes called Prover or Agent) can send remotely to a so-called Verifier, via a remote attestation protocol.

When the system boots up, PCRs in the TPM are reset. The control then passes to CRTM (Core Root of Trust of Measurement), which calculates the hash of the BIOS and extends this hash to PCRs. The control is then passes to BIOS, which calculates the hash of the operating system (that is the kernel image), and extends this hash to PCRs. Then the system boots up.

Before any software (such as the `ptpd`, `gpsd`, or `ppsi` daemons) is started, the hash over its binary is calculated and is added to a so-called stored measurement list (SML) maintained by the kernel. This hash is also extended to the PCRs. Before any software runs, its fingerprint is recorded into the PCRs in an incremental fashion, and thus cannot be reverted back. So a system's SML has a list of hashes (one for every piece of executable), and the PCR has the incremental hash of all these hashes. Note that at system boot up, the kernel calculates the hash of BIOS and its own hash, and puts them into the SML. Thus we can say that measurement list together with PCR completely give the measurement of the runtime behavior of the system. Once the digest of the measurements is securely stored in the TPM, it is possible

to create an integrity report (protected with specific keys) that will be transferred to the (remote) Verifier via a remote attestation protocol.

We have tried to attest the software stack on a dedicated Raspberry Pi 4 device equipped with an Infineon TPM v2 and pre-configured with the user space TC software (e.g., TPM2 TSS v2.4.0, TPM2 Tools v4.2, TPM2 TSS Engine v1.1.0) and the Integrity Measurement Architecture (IMA) security subsystem for Linux [42]. For remote attestation, besides IMA, we have installed on the device part of the Keylime [43] software, which provides trusted computing services. In particular, on the device we have installed the *Prover* part (or *Agent*) of the remote attestation protocol.

In the experimental testbed we have used an additional PC, running the dedicated software components needed for the software integrity verification, namely *Registrar*, used for key management, and *Verifier*.

After completing the Keylime installation, we have created whitelist and excludelist indicating the software to be measured. For example, we have selected the `ptpd` daemon to be measured. We simulated an attack in which we have tried to modify the executable of the `ptpd` daemon, by replacing it with the one of the `mv` command). We have observed that the *Verifier* indicates an error specifying that the hash of the executable file has not been verified. Subsequently, we have tried to modify the code of the `ptpd` daemon and we installed it at the same path. Also in this case, the *Verifier* has correctly indicated that the `ptpd` daemon has not been successfully verified.

5 Conclusions

This paper investigates possible attack strategies in transport-based time synchronization architectures. In addition to the attack types dealing with time distribution in packet switched networks described in RFC 7384, we indicate other possible attack points applying to different views (including the network management, hardware, and the software ones) of such architectures. While this paper presents some experimental results with regard to software attack implementation, further research will explore these threats in more depth on the devices exploited in the ROOT project. We note that combinations of different types of attacks, e.g., spoofing or jamming attacks, software attacks, and timing distribution protocol attacks, targeting simultaneously different actors in a transport-based time synchronization solutions have not been investigated yet, but are definitely worth to address in the future.

Acknowledgements

This work was developed within the ROOT project (www.gnss-root.eu) funded by the European GNSS Agency under the European Union's Horizon 2020 – G.A. n. 101004261.

References

1. Ruffini S, Johansson M, Pohlman B, Sandgren M (2021) 5G syncronization requirements and solutions. Ericsson Technology Review, https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/5g-synchronization-requirements-and-solutions
2. The ROOT (Rolling Out OSNMA for the Secure Synchronisation of Telecom Networks) Project, https://www.gnss-root.eu/
3. ITU-T Recommendations G.826x and G.827x series (G.8200-G.8299: Synchronization, quality and availability targets, https://www.itu.int/ITU-T/recommendations/index_sg.aspx?sg=15
4. IEEE 1588-2019 – IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, June 16, 2020, https://standards.ieee.org/standard/1588-2019.html
5. Mills D, Delaware U, Martin J, Burbank J, Kasch W (2010) Network Time Protocol Version 4: Protocol and Algorithms Specification. IETF RFC 5905
6. Girela-López F, López-Jiménez J, Jiménez-López M, Rodríguez R, Ros E, Díaz J, IEEE 1588 High Accuracy Default Profile: Applications and Challenges, In: IEEE Access, vol. 8, pp. 45211-45220, 2020, doi: 10.1109/ACCESS.2020.2978337
7. Mizrahi T (2014) Security requirements of time protocols in packet switched networks. IETF RFC 7384
8. Borio D, Dovis F, Kuusniemi H, Lo Presti L (2016) Impact and Detection of GNSS Jammers on Consumer Grade Satellite Navigation Receivers, in Proceedings of the IEEE, vol. 104, no. 6, pp. 1233-1245, June 2016. doi: 10.1109/JPROC.2016.2543266
9. Falletti E, Margaria D, Marucco G, Motella B, Nicola M, Pini M (2018) Synchronization of critical infrastructures dependent upon GNSS: Current vulnerabilities and protection provided by new signals. IEEE Systems Journal, 13(3), 2118-2129
10. DeCusatis C, Lynch RM, Kluge W, Houston J, Wojciak PA, and Guendert S (2020) Impact of Cyberattacks on Precision Time Protocol. In: IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 5, pp. 2172-2181, May 2020, doi: 10.1109/TIM.2019.2918597
11. Dalmas M, Rachadel H, Silvano G, Dutra C (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
12. 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. doi: 10.1109/TDSC.2017.2748583
13. Alghamdi W, Schukat M (2021) Precision time protocol attack strategies and their resistance to existing security extensions. Cybersecur 4, 12, doi: 10.1186/s42400-021-00080-y
14. Barreto S, Suresh A, Le Boudec J (2016) Cyber-attack on packet-based time synchronization protocols: The undetectable Delay Box. In: 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, pp. 1-6, doi: 10.1109/I2MTC.2016.7520408
15. Pini M, Minetto A, Vesco A, Berbecaru D, Contreras Murillo LM, Nemry P, De Francesca I, Rat B, Callewaert K (2021) Satellite-derived Time for Enhanced Telecom Networks Synchronization: the ROOT Project. accepted for publication at IEEE MetroAeroSpace 2021, Naples (Italy), June 23-25, 2021
16. Barreto S, Suresh A, Le Boudec J (2016) Cyber-attack on packet-based time synchronization protocols: The undetectable Delay Box. In: 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, 2016, pp. 1-6, doi: 10.1109/I2MTC.2016.7520408
17. 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 2009, pp. 1-6, doi: 10.1109/ISPCS.2009.5340224
18. Ylonen T, Lonvick C (2006) The Secure Shell (SSH) Connection Protocol. IETF RFC 4254
19. Rescorla E (2018) The Transport Layer Security (TLS) Protocol version 1.3. IETF RFC 8446

20. Rigney C, Willens S, Rubens A, Simpson W (2000) Remote Authentication Dial In User Service (RADIUS). IETF RFC 2865
21. Levi D, Meyer P, Stewart B (2002) Simple Network Management Protocol (SNMP) Application. IETF RFC 3413
22. Berbecaru D, Casalino M M, Lioy A (2013) FcgiOCSP: a scalable OCSP-based certificate validation system exploiting the FastCGI interface. Softw. Pract. Exper., 43: 1489-1518. doi: 10.1002/spe.2148
23. Internal Attacks and their Impact on Organizations, September 8, 2018, https://www.rewterz.com/vulnerability-management/internal-attacks-and-their-impact-on-organizations
24. Alghamdi W, Schukat M (2020) A Detection Model Against Precision Time Protocol Attacks, in 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2020, pp. 1-3, doi: 10.1109/ICCAIS48893.2020.9096742.
25. Alghamdi W, Schukat M (2020) Practical Implementation of APTs on PTP Time Synchronisation Networks, In: 31st Irish Signals and Systems Conference (ISSC), 2020, pp. 1-5, doi: 10.1109/ISSC49989.2020.9180157
26. Alghamdi W, Schukat M (2020) Cyber Attacks on Precision Time Protocol Networks—A Case Study, Electronics (MDPI), 9, 1398. doi: 10.3390/electronics9091398
27. Langner R (2011) Stuxnet: Dissecting a cyberwarfare weapon. IEEE Secur. Priv. 2011, 9, 49–51
28. Guenther C (2014) A survey of spoofing and countermeasures, Navigation, Journal of the Institute of Navigation, vol. 61, no. 3, pp. 159-177
29. Estrela PV, Neusüß S, Owczarek W (2014) Using a multi-source NTP watchdog to increase the robustness of PTPv2 in financial industry networks. In: 2014 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, Austin, TX, USA, 21–26 September 2014; pp. 87–92
30. Kröner U, Bergonzi C, Fortuny-Guasch J, Giuliani R, Littmann F, Shaw D, Symeonidis D (2010) Hardening of GNSS based trackers. Available at: https://publications.jrc.ec.europa.eu/repository/bitstream/JRC58733/reqno_jrc58733_st_report_on_hardening_of_gnss_based_trackers_release_final.pdf
31. ptpd - Precision Time Protocol daemon (1588-2008), June 2015, Available at: https://manpages.debian.org/stretch/ptpd/ptpd.8.en.html
32. The GPSD project. gpsd(8) Manual Page, March 2021. Available at: https://gpsd.io/gpsd.html
33. [36] PPSi: PTP Ported to Silicon, Wiki – Open Hardware Repository, August 2014. Available at: https://ohwr.org/project/ppsi/wikis/home
34. Piessens F, Verbauwhede I (2016) Software security: Vulnerabilities and countermeasures for two attacker models, In: Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 990-999
35. ARP spoofing – flaws in network security, Available at: https://www.ionos.com/digitalguide/server/security/arp-spoofing-attacks-from-the-internal-network/
36. Ping flood, Available at: https://www.ionos.com/digitalguide/server/security/ping-flood/
37. Kim TH, Reeves D (2020) A survey of domain name system vulnerabilities and attacks., J Surveill Secur Saf 2020;1:34-60, doi: 10.20517/jsss.2020.14
38. Stebila D, Attacks on TLS, Available at: https://www.douglas.stebila.ca/research/presentations/tls-attacks/
39. Berbecaru D, Lioy A (2007) On the Robustness of Applications Based on the SSL and TLS Security Protocols, In: Proceedings of the 4th European PKI Workshop: Theory and Practice (EuroPKI 2007), Palma de Mallorca (Spain), 28-30 June, 2007, vol. 4582 of Lecture Notes in Computer Science, Springer-Verlag, pp. 248-264, doi: 10.1007/978-3-540-73408-6_18
40. Berbecaru D (2018) On Creating Digital Evidence in IP Networks With NetTrack, Handbook of Research on Network Forensics and Analysis Techniques, IGI Global, doi: 10.4018/978-1-5225-4100-4.ch012

41. Ankergård SFJJ, Dushku E, Dragoni N (2021) State-of-the-Art Software-Based Remote Attestation: Opportunities and Open Issues for Internet of Things, Sensors 2021, 21, 1598. doi: 10.3390/s21051598
42. Kasatkin D, Zohar M (2017) Integrity Measurement Architecture. Available at: https://sourceforge.net/p/linux-ima/wiki/Home/
43. Keylime. Bootstrap & Maintain Trust on the Edge/Cloud and IoT. Available at: https://keylime.dev/