TICTOC Working Group
Internet Draft
Intended status: Informational
Expires: April 2012

Tal Mizrahi
Marvell
Karen O'Donoghue
ISOC
October 24, 2011

TICTOC Security Requirements
draft-mizrahi-tictoc-security-requirements-00.txt

Abstract

As time synchronization protocols are becoming increasingly common and widely deployed, concern about their exposure to various security threats is increasing. This document defines a set of requirements for security solutions for time synchronization protocols, focusing on the IEEE 1588 and NTP. This document also discusses the security impacts of time synchronization protocol practices, the time synchronization performance implications of external security practices, the dependencies between other security services and time synchronization.

Table of Contents

1. Introduction

As time synchronization protocols are becoming increasingly common and widely deployed, concern about the resulting exposure to various security threats is increasing. If a time synchronization protocol is compromised, the applications it serves are prone to a range of possible attacks including Denial-of-Service or incorrect behavior.

This document focuses on the security aspects of the Precision Time Protocol ([IEEE 1588]) and the Network Time Protocol ([NTPv4]). The Network Time Protocol was defined with an inherent security protocol, defined in [NTPv4] and in [AutoKey]. The IEEE 1588 includes an experimental security protocol, defined in Annex K of the standard, but this Annex was never formalized into a fully defined security protocol.

This document attempts to add clarity to the time synchronization protocol security requirements discussion by addressing a series of questions. It is expected that this document will evolve into possibly two documents including one on requirements and one providing clarity around the additional questions raised below. Until the discussion has matured sufficiently, it will be captured in this document. The four primary questions addressed by this draft include:

(1) What are the threats that need to be addressed for the time synchronization protocol, and thus what security services need to be provided? (e.g. a malicious NTP server or PTP master)

(2) What external security practices impact the security and performance of time keeping, and what can be done to mitigate these impacts? (e.g. an IPSec tunnel in the synchronization traffic path)

(3) What are the security impacts of time synchronization protocol practices? (e.g. on-the-fly modification of timestamps)

(4) What are the dependencies between other security services and time synchronization? (e.g. which comes first - the certificate or the timestamp?)

It is expected that the final version of this document will define a set of requirements for security solutions for time synchronization protocols, focusing on the IEEE 1588 and NTP.

2. Conventions Used in this Document

2.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

This document describes security requirements, and thus requirements are phrased in the document in the form "the security mechanism MUST/SHOULD/...". Note, that the phrasing does not imply that this document defines a specific security mechanism, but defines the requirements that every security mechanism should comply to.

This document refers to both PTP and NTP. For the sake of consistency, throughout the document the term "master" applies to both a PTP master and an NTP server. Similarly, the term "slave" applies to both PTP slaves and NTP clients. The general term "clock" refers to masters, slaves and PTP Transparent Clocks (TC). The term "protocol packets" is refers generically to PTP and NTP messages.

2.2. Abbreviations

BC Boundary Clock

MITM Man In The Middle

NTP Network Time Protocol

OC Ordinary Clock

PTP Precision Time Protocol

TC Transparent Clock

3. Security Threats

The following section defines the security threats that are discussed in subsequent sections.

3.1. Packet interception and manipulation

A packet interception and manipulation attack results when a Man-In-The-Middle (MITM) attacker intercepts timing protocol packets, alters them and relays them to their destination, allowing the attacker to maliciously tamper with the protocol. This can result in a situation where the time protocol is apparently operational but providing intentionally inaccurate information.

3.2. Spoofing

In spoofing, an attacker masquerades as a legitimate node in the network. For example, an attacker can impersonate the master, allowing malicious distribution of false timing information. As with packet interception and manipulation, this can result in a situation where the time protocol is apparently operational but providing intentionally inaccurate information.

3.3. Replay attack

In a replay attack, an attacker records protocol packets and replays them at a later time. This can also result in a situation where the time protocol is apparently operational but providing intentionally inaccurate information.

3.4. Rogue master attack

In a rogue master attack, an attacker causes other nodes in the network to believe it is a legitimate master. As opposed to the spoofing attack, in the Rouge Master attack the attacker does not fake its identity, but rather manipulates the master election process. For example, in PTP, an attacker can manipulate the Best Master Clock Algorithm (BMCA), and cause other nodes in the network to believe it is the most eligible candidate to be a grandmaster.

3.5. Packet Interception and Removal

A packet interception and removal attack results when a Man-In-The-Middle attacker intercepts and drops protocol packets, preventing the destination node from receiving the timing information.

3.6. Packet delay manipulation

In a packet delay manipulation scenario, a Man-In-The-Middle attacker intercepts protocol packets, and relays them to their destination after adding a maliciously computed delay.

3.7. Cryptographic performance attacks

In cryptographic performance attacks, an attacker transmits fake protocol packet, causing high utilization of the cryptographic engine at the receiver, which attempts to verify the integrity of these fake packets.

3.8. DoS attacks

There are many possible Layer 2 and Layer 3 Denial of Service attacks. As the target's availability is compromised, the timing protocol is affected accordingly.

3.9. Time source spoofing (e.g. GPS fraud)

In time source spoofing, an attacker spoofs the accurate time source of the master. For example, if the master uses a GPS based clock as its reference source, an attacker can spoof the GPS satellites, causing the master to use a false reference time.

4. Security Requirements

4.1. Clock Identity Authentication

Requirement

The security mechanism MUST provide a means for each clock to authenticate the sender of a protocol packet.

Discussion

In the context of this document, authentication refers to:

- Identification: verifying the identity of the peer clock.
- Authorization: verifying that the peer clock is permitted to play the role that it plays in the protocol. For example, some nodes may be permitted to be masters, while other nodes are only permitted to be slaves or TCs.

  The following subsections describe 4 distinct cases of clock authentication.

4.1.1. Authentication and Proventication of Masters

Requirement

The security mechanism MUST support a proventication mechanism, to be used in cases where end-to-end authentication is not possible.

Discussion

Slaves and transparent clocks authenticate masters in order to ensure the authenticity of the time source.

In some cases a slave is connected to an intermediate master, that is not the primary time source. For example, in PTP a slave can be connected to a Boundary Clock (BC), which in turn is connected to a grandmaster. A similar example in NTP is when a client is connected to a stratum 2 server, which is connected to a stratum 1 server. In both the PTP and the NTP cases, the slave authenticates the intermediate master, and the intermediate master authenticates the primary master. This inductive authentication process is referred to in [AutoKey] as proventication.

4.1.2. Authentication of Slaves

Requirement

The security mechanism SHOULD provide a means for a master to authenticate its slaves.

Discussion

Slaves are authenticated by masters in order to verify that the slave is authorized to receive timing services from the master.

Authentication of slaves prevents unauthorized clocks from receiving time services, and also reduces unnecessary load on the master clock, by preventing the master from serving unauthorized clocks. It could be argued that the authentication of slaves could put a higher load on the master then serving the unauthorized clock. This tradeoff will need to be discussed further.

4.1.3. PTP: Authentication of Transparent Clocks

Requirement

The security mechanism for PTP SHOULD provide a means for a master to authenticate the TCs.

Discussion

Transparent clocks are authenticated by peer masters, slaves and TCs.

Authentication of TCs, much like authentication of slaves, reduces unnecessary load on the master clock and peer TCs, by preventing the master from serving unauthorized clocks. It also prevents malicious TCs from attacking the protocol by manipulating the correctionField. It could also be argued that the authentication could result in a higher load then merely serving the unauthorized devices. This tradeoff will need to be discussed further.

4.1.4. PTP: Authentication of Announce Messages

Requirement

The security mechanism for PTP MUST support authentication of Announce messages.

Discussion

Master election is performed in PTP using the Best Master Clock Algorithm (BMCA). Each Ordinary Clock (OC) announces its clock attributes using Announce messages, and the best master is elected based on the information gathered from all the candidates. Announce messages must be authenticated in order to prevent malicious master attacks.

Note, that this subsection specifies a requirement that is not necessarily included in 4.1.1. or in 4.1.2. , since the BMCA is initiated before clocks have been defined as masters or slaves.

4.2. Data integrity

Requirement

The security mechanism MUST protect the integrity of protocol packets.

Discussion

While subsection 4.1. refers to ensuring WHO sent the protocol packet, this subsection refers to ensuring that the packet arrived intact. The integrity protection mechanism ensures the authenticity and completeness of data from the data originator.

4.2.1. PTP: Hop-by-hop vs. End-to-end Integrity Protection

Requirement

A security mechanism for PTP MUST support hop-by-hop integrity protection.

Requirement

A security mechanism for PTP SHOULD support end-to-end integrity protection.

Discussion

Specifically in PTP, when protocol packets are subjected to modification by TCs, the integrity protection can be enforced in one of two approaches, end-to-end or hop-by-hop.

4.2.1.1. Hop by Hop Integrity Protection

Each hop that needs to modify a protocol packet:

- o Verifies its integrity.
- o Modifies the packet, i.e., modifies the correctionField.
- o Re-generates the integrity protection, e.g., re-computes a Message Authentication Code.

In the hop-by-hop approach, the integrity of protocol packets is protected by induction on the path from the originator to the receiver.

This approach is simple, but allows malicious TCs to modify protocol packets.

4.2.1.2. End to End Integrity Protection

In this approach, the integrity protection is maintained on the path from the originator of a protocol packet to the receiver. This allows the receiver to validate the protocol packet without the ability of intermediate TCs to manipulate the packet.

Since TCs need to modify the correctionField, a separate integrity protection mechanism is used specifically for the correctionField.

The end-to-end approach limits the TC's impact to the correctionField alone, while the rest of the protocol packet is protected on an end-to-end basis.

4.3. Availability

Requirement

The security mechanism MUST be resistant to DoS attacks from an external attacker.

Discussion

This requirement is attained by clock authentication, as described in 4.1. .

4.4. Replay Protection

Requirement

Protocol messages MUST be resistant to replay attacks.

4.5. Cryptographic Keys & Security Associations

4.5.1. Security Association

Requirement

The security protocol MUST support an association protocol where:

- Two or more clocks authenticate each other.
- The clocks generate and agree on a cryptographic session key.

Discussion

The security requirements in 4.1. and 4.2. require usage of cryptographich mechanisms, deploying cryptographic keys. A security association is an essential building block in these mechanisms.

4.5.2. Unicast and Multicast

Requirement

The security mechanism MUST support security association protocols for unicast and for multicast associations.

Discussion

A unicast protocol requires an association protocol between two clocks, whereas a multicast protocol requires an association protocol among two or more clocks, where one of the clocks is a master.

4.5.3. Key Freshness

Requirement

The cryptographic keys MUST be refreshed periodically.

Requirement

The association protocol MUST be invoked periodically, where each instance of the association protocol MUST produce a different session key.

4.6. Performance

Requirement

The security mechanism MUST be designed in such a way that it does not degrade the quality of the time transfer.

Requirement

The mechanism SHOULD be relatively lightweight, as client restrictions often dictate a low processing and memory footprint, and because the server may have extensive fan-out.

Requirement

The mechanism also SHOULD not require excessive storage of client state in the master, nor significantly increase bandwidth consumption.

4.7. Confidentiality

Requirement

The security mechanism MAY provide confidentiality protection of the protocol packets.

Discussion

In the context of time synchronization, confidentiality is typically of low importance, since timing information is typically not considered secret information.

Confidentiality can play an important role when service providers charge payment for time synchronization services, but these cases are rather esoteric.

Confidentiality can also prevent an MITM attacker from identifying protocol packets. Thus, confidentiality can assist in protecting the timing protocol against packet delay attacks, where the attacker selectively adds delay to time protocol packets.

4.8. Protection against packet delay attacks

Requirement

The security mechanism MAY include a means to detect packet delay attacks.

Requirement

The security mechanism MAY include a protection switching mechanism that allows a node that detects a delay attack to switch over to a secondary master.

5. Summary of Requirements

Section	Requirement	Type
4.1.	Authentication of sender.	MUST
	Proventication.	MUST
	Authentication of slaves.	SHOULD
	PTP: Authentication of TCs.	SHOULD
	PTP: Authentication of Announce messages.	SHOULD
4.2.	Integrity protection.	MUST
	PTP: hop-by-hop integrity protection.	MUST
	PTP: end-to-end integrity protection.	SHOULD

4.3.	Protection against DoS attacks.	MUST
4.4.	Replay protection.	MUST
4.5.	Security association.	MUST
	Unicast and multicast associations.	MUST
	Key freshness.	MUST
4.6.	Performance: no degradation in quality of time transfer.	MUST
	Performance: lightweight.	SHOULD
	Performance: storage, bandwidth.	MUST
4.7.	Confidentiality protection.	MAY
4.8.	Protection against delay attacks.	MAY

Table 1 Summary of Security Requirements

6. Additional security implications

This section will discuss additional security implications as outlined in the questions below. Contributions are welcome and encouraged.

- What external security practices impact the security and performance of time keeping? (and what can be done to mitigate these impacts?)

- What are the security impacts of time synchronization protocol practices? (e.g. on-the-fly modification of timestamps)

- What are the dependencies between other security services and time synchronization?

7. Issues for Further Discussion

This section will discuss additional issues as identified below. Again, contributions are welcome and encouraged.

- o Integrity - end-to-end vs. hop-by-hop.

- o Supporting a hybrid network, where some nodes are security enabled and others are not.

- o The key distribution is outside the scope of this document. Although this is a cardinal element in any security system, it is not a security requirement, and is thus not described here.

8. Security Considerations

The security considerations of network timing protocols are presented throughout this document.

9. IANA Considerations

There are no new IANA considerations implied by this document.

10. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

11. References

11.1. Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[NTPv4] Mills, D., Delaware, U., Martin, J., Burbank, J., Kasch, W., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[AutoKey] Haberman, B., Mills, D., "Network Time Protocol Version 4: Autokey Specification", RFC 5906, June 2010.

[Traps] Treytl, A., Gaderer, G., Hirschler, B., Cohen, R., "Traps and pitfalls in secure clock synchronization" in Proceedings of 2007 International Symposium for Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2007, pp. 18-24, 2007.

11.2. Informative References

[IEEE 1588] IEEE TC 9 Test and Measurement Society 2000, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com

Karen O'Donoghue
7167 Goby Lane
King George, VA 22485

Email: odonoghue@isoc.org