NTP Working Group D. Sibold
Internet-Draft PTB
Obsoletes: 5906 (if approved) S. Roettger
Intended status: Standards Track TU-BS
Expires: January 29, 2013 July 30, 2012

Network Time Protocol: autokey Version 2 Specification draft-sibold-autokey-00

Abstract

This document describes a security protocol that enables authenticated time synchronization using Network Time Protocol (NTP). Autokey Version 2 obsoletes NTP autokey protocol (RFC 5906) which suffers from various security vulnerabilities. Its design considers the special requirements that are related to the task of precise timekeeping.

Requirements Language

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 RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 29, 2013.

Copyright Notice

Table of Contents

1. Introduction

In NTP [RFC5905] the autokey protocol [RFC5906] was introduced to provide authenticity to NTP servers and to ensure integrity of time synchronization. It is designed to meet the specific communication requirements of precise timekeeping. Its basic design is a combination of PKI and a pseudo-random sequence of symmetric keys, the so-called autokeys of which each are valid for one packet only. This design maintains the stateless nature of NTP and therefore does not compromise timekeeping precision.

This document focuses on a new definition of the autokey protocol for NTP, autokey version 2. The necessity to renew the autokey specification arises from various severe security vulnerabilities that have been found in a thorough analysis of the protocol [Roettger]. The new specification is based on the same assumptions as the original autokey specification. In particular, the prerequisite is that precise timekeeping can only be accomplished with stateless time synchronization communication, which excludes standard security protocols like IPSec or TLS. This prerequisite corresponds with the requirement that a security mechanism for timekeeping must be designed in such a way that it does not degrade the quality of the time transfer [I-D.ietf-tictoc-security-requirements].

1.1. Differences from the original autokey

Autokey version 2 is a major redraft of the original autokey specification. It is intended to mitigate security vulnerabilities of the original specification and it is based on the suggestions in the analysis of Roettger [Roettger]. The major changes are:

- The bit length of server seed and cookie has been increased.
- The utilized hash algorithms are negotiable.
- The IP addresses of the synchronization partners in the calculation of the cookie have been replaced by the public key of the NTP client.
- The identity schemes for the verification of the NTP server authenticity have been replaced by a hierarchical public key infrastructure (PKI) based on X.509 certificates.
- Compatibility with the current autokey specification is not given.
- The term proventication is not used, i.e., authorization and time synchronization are disentangled.

Discussion

The client verifies the authenticity of the server via PKI infrastructure. To this end, it has to verify the certification chain up to a trusted authority which, in the context of the PKI, is a certification authority (CA). Proventication may be established if the trusted authority is also the NTP stratum 1 server. See also the discussion in Section 6.2.

2. Security Threats

A profound analysis of security threats and requirements for NTP and Precision Time Protocol (PTP) can be found in the I-D [I-D.ietf-

tictoc-security-requirements].

3. Objectives

The objectives of the autokey specifications are as follows:

- Authenticity: Autokey enables the client to authenticate its NTP server or peer.
- Integrity: Autokey protects the integrity of time synchronization packets via a message authentication code (MAC) or a hash-based message authentication code (HMAC).
- Confidentiality: Autokey does not provide confidentiality protection of the NTP packets.
- Modes of operation: All operational modes of NTP are supported (Client-Server, symmetric, broadcast).
- Hybrid mode: Both secure and insecure communication modes are possible for NTP servers and clients, respectively.
- Compatibility: Interoperation with autokey version 1 and the symmetric key scheme described in [RFC1305] is not given. Insecure NTP associations are not affected.
- Leap seconds are not in the scope of autokey.

4. Terms and abbreviations

- Throughout this document the term "autokey" refers to autokey version 2.

5. Autokey Overview

In autokey, authenticity and integrity of NTP packets are ensured by an attached key ID and a message authentication code (MAC). The MAC is calculated with a so-called "autokey" which is a symmetric key that is valid for one packet only. The MAC is given by

MAC = H(autokey || NTP packet),

where || indicates concatenation and in which H is a hash algorithm on which client and server agree during the association message (ASSOC) exchange. The key ID uniquely identifies the autokey. The autokeys are calculated for each NTP packet according to:

autokey = H(key ID || cookie),

in which H is a hash function on which client and server have to

agree (during ASSOC) and which is not necessarily identical to the one used for the MAC calculation. The cookie is a 128 bit secret

between client and server. It is exchanged during the cookie message protocol sequence (COOK). The cookie is calculated by the server via

cookie = MSB_128 (H(server seed || public key of client)).

The same hash algorithm H is utilized as in the calculation of the autokey. The function MSB_128 cuts off the 128 most significant bits of the result of the hash function. The server seed is a 128 bit random value of the server, which has to be kept secret. The cookie thus never changes. To comply with 4.5.3 in [I-D.ietf-tictoc-security-requirements] the server seed has to be changed periodically. The server does not keep a state of the client. Therefore it has to recalculate the cookie each time it receives a request from the client. To this end, the client has to attach its public key to each request (see Section 6.4).

Discussion

Alternative cookie calculation: Instead of using the client's public key for the cookie calculation, the hash value of the public key can be used. This has the advantage that during the time request message the client only needs to send the hash of its public key and not the whole public key itself.

6. Protocol Sequence

6.1. Association Message

The protocol sequence starts with the association message, in which the client sends an NTP packet with an extension field of type association. It contains the hostname of the client and a status word which contains the algorithms used for the signatures and the status of the connection. The response contains the hostname of the server and the algorithms for the signatures. Client and server MUST agree upon the employed MAC and hash algorithms.

6.2. Certificate Message

In this step, the client receives the certification chain up to the trusted authority (TA). To this end, the client requests the certificate for the subject name (hostname) of the NTP server. The response contains the certificate with the issuer name. If the issuer name is different from the subject name, the client requests the certificate for the issuer. This continues until it receives a certificate which is issued by a TA. The client recognizes the TA because it has a list of certificates which are accepted as TAs. The client has to prove that each issuer is authorized to issue new certificates. To this end, it has to prove that the X.509v3 extension contains the field "CA:TRUE". With the established certification chain the client is able to verify the server

signatures and, hence, the authenticity of the server messages with extension fields is ensured.

Discussion

Note that this certification chain is a priori independent of the time synchronization chain, because the TA and the NTP root are not inevitably identical. This has consequences if proventication is required (Requirement 4.1.2 in [I-D.ietf-tictoc-security-requirements]). In this case, proventication can be ensured only if the NTP root server is also a recognized TA, hence a CA.

6.3. Cookie Message

The client requests a cookie from the server, which is used to calculate the autokeys. The request includes the public key of the client. The public key is used by the server to calculate the cookie. The response of the server contains the cookie encrypted with the public key.

6.4. Time request message

The client request includes a new extension field "time request" which contains its public key. The server needs the public key to recalculate the cookie for the client. The response is a normal NTP packet without extension field.

7. Hash and MAC algorithms

Hash algorithms are used for the calculation of cookie, autokey and MAC.

7.1. Hash Function for Cookie and Autokey

The hash algorithm utilized for the calculation of the cookie and the autokey is negotiated during the association message exchange (Section 6.1). The client MUST request SHA-1 or a stronger hash function. The server also MUST provide SHA-256.

7.2. Hash Function for the Message Authentication Code

The hash function for the MAC is negotiated during the association message exchange in Section 6.1. Client and server SHOULD negotiate a Keyed-Hash Message Authentication Code [RFC2104].

8. Server Seed Considerations

The server has to calculate a random seed which has to be kepted secret and which has to be changed periodically.

8.1. Server Seed Function

8.2. Server Seed Live Time

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

The client has to verify the validity of the certificates during the certification message exchange (Section 6.2). Since it generally has no reliable time during this initial communication phase, it is impossible to verify the period of validity of the certificates. Therefore, the client MUST use one of the following approaches:

- The TA and the dependent certificates are trusted by default. Usually this will be the case in corporation networks.
- The client ensures that the certificates are not revoked. To this end, the client uses the Online Certificate Status Protocol (OCSP) defined in [RFC6277].
- The client requests a different service to get an initial time stamp in order to be able to verify the certificates' periods of validity. To this end, it can, e.g., use a secure shell connection to a reliable host. Another alternative is to request a time stamp from a Time Stamping Authority (TSA) by means of the Time-Stamp Protocol (TSP) defined in [RFC3161].

11. Acknowledgements

12. References

12.1. Normative References

[I-D.ietf-tictoc-security-requirements] Mizrahi, T. and K. O'Donoghue, "TICTOC Security Requirements", Internet-Draft draft-ietf-tictoc-security-requirements-02, June 2012.

[RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

[RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3161] Adams, C., Cain, P., Pinkas, D. and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

[RFC5905] Mills, D., Martin, J., Burbank, J. and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC5906] Haberman, B. and D. Mills, "Network Time Protocol Version 4: Autokey Specification", RFC 5906, June 2010.

[RFC6277] Santesson, S. and P. Hallam-Baker, "Online Certificate Status Protocol Algorithm Agility", RFC 6277, June 2011.

12.2. Informative References

[Roettger] Roettger, S., "Analysis of the NTP Autokey Procedures", February 2012.

Appendix A. TICTOC Security Requirements

The following table compares the autokey specifications against the tictoc security requirements [I-D.ietf-tictoc-security-requirements].

Section	Requirement from I-D tictoc security-requirements-02	Type	Autokey V2
4.1	Authentication of sender.	MUST	OK
	Authentication of master.	MUST	OK
	Proventication	MUST	Open 1)
	Authentication of slaves.	SHOULD	OK
	PTP: Authentication of TCs.	SHOULD	N/A
	PTP: Authentication of Announce messages.	SHOULD	N/A
4.2	Integrity protection.	MUST	OK
	PTP: hop-by-hop integrity protection.	MUST	N/A
	PTP: end-to-end integrity protection.	SHOULD	N/A
4.3	Protection against DoS attacks.	MUST	NTP 2)
4.4	Replay protection.	MUST	NTP 2)
4.5	Security association.	MUST	OK
	Unicast and multicast associations.	MUST	OK
	Key freshness.	MUST	OK
4.6	Performance: no degradation in quality of time transfer.	MUST	OK
	Performance: lightweight.	SHOULD	YES
	Performance: storage, bandwidth.	MUST	OK
4.7	Confidentiality protection.	MAY	NO
	Protection against delay attacks.	MAY	NO
4.9	Secure mode.	MUST	NTP? 3)
	Hybrid mode.	MAY	YES

1) Refer to discussion in Section 6.2. 2) These requirements are fulfilled by the NTP on-wire protocol. 3) Has still to be checked.

Authors' Addresses

Dieter Sibold
Physikalisch-Technische Bundesanstalt
Bundesallee 100
Braunschweig, D-38116
Germany

Phone: +49-(0)531-592-8420
Email: dieter.sibold@ptb.de

Stephen Roettger
Technische Universitaet Braunschweig

Email: stephen.roettger@googlemail.com