TRANS
Internet-Draft
Intended status: Experimental
Expires: April 30, 2015

L. Nordberg
NORDUnet

October 27, 2014

Gossiping in CT
draft-linus-trans-gossip-ct-00

Abstract

This document describes gossiping in Certificate Transparency [RFC6962].

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 April 30, 2015.

Copyright Notice

Table of Contents

1. Introduction

Gossiping in Certificate Transparency (CT) can be split up in three pieces:

- A general gossip protocol. This document uses [draft-linus-trans-gossip] for a general gossip protocol.

- Gossip strategy and policy - what data to gossip and how to deal with incoming gossip information.

- Gossiping rules, i.e. what type of data and with whom to gossip.

The scope for this document is the last point, the gossiping rules.

2. Problem

Gossiping about what's known about CT logs helps solving the problem of detecting malicious logs showing different views to different clients, a.k.a. the partitioning attack.

The separate problem of how to disseminate information about a log misbehaving in other ways may be helped by gossiping but poses a

potential threat to the privacy of end users. Gossiping about log data linkable to a specific log entry and through that to a specific site has to be constrained to using the gossiping message format and gossiping transports for sending sensitive data only to particular recipients.

3. Who should gossip

- TLS clients using PKIX (i.e. web browsers, MTA:s, MUA:s, XMPP clients)
- CT auditors and CT monitors

4. What kind of data to gossip about

This section describes what type of log data to gossip.

4.1. Signed Tree Heads

All CT clients SHOULD gossip about Signed Tree Heads (STH's) with as many other CT clients as possible.

Gossiping about STH's enables detection of logs presenting more than one view of the log.

An STH contains: - the size of the tree being signed - a timestamp indicating the time when the tree was signed - the merkle tree hash of the tree being signed - a signature made by the log

An STH received from a client may indicate the following about that client: - gossiping - using CT, as late as the timestamp and tree size indicate - talking, indirectly, to the log indicated by the tree hash - software being used and software version

Which STH's to send and how often is part of gossiping strategy and out of scope for this document.

[TBD gossip about inclusion proofs and consistency proofs too?]

STH's are sent to a preconfigured gossip service in a [draft-linus-trans-gossip] GOSSIP-MSG message with 'gossip-data' as a JSON object [RFC7159] with the following content:

- sths: array of [RFC6962] Signed Tree Head's

4.1.1. Web browsers

Web browsers SHOULD send STH's to web servers using Transparency Gossiping [draft-linus-trans-gossip] by sending GOSSIP-MSG messages to a gossip service. Web browsers SHOULD use the [draft-linus-trans-gossip-transport-https] transport and MAY use other transports as well.

Which web servers STH's will be sent to depends on which web servers the chosen transports are connected to and those web servers capability and willingness to convey gossip. This is handled by the gossip transports.

Web browsers MAY register as a gossip transport themselves and perform the sending and receiving of gossip messages using connections already in use.

4.1.2. CT monitors

CT monitors SHOULD send STH's to web servers using Transparency Gossiping [draft-linus-trans-gossip] by sending GOSSIP-MSG messages to a gossip service.

CT monitors SHOULD use as many transports as possible.

4.1.3. MTA:s

TBD

4.1.4. MUA:s

TBD

4.1.5. XMPP clients

TBD

4.2. Illegitimate Signed Certificate Timestamps

If a TLS client detects misbehaviour of a log related to a given Signed Certificate Timestamp (SCT) it MAY send that SCT to the web server it got the SCT from. A corresponding X.509 certificate chain MAY be sent along with the SCT. The [draft-linus-trans-gossip-transport-https] messaging format SHOULD be used for this.

SCT's and corresponding X.509 certificates are sent to a preconfigured gossip service in a [draft-linus-trans-gossip] GOSSIP-MSG message with 'gossip-data' as a JSON object [RFC4627] with the following content:

- entry: An array of objects consisting of
  - sct: An [RFC6962] Signed Certificate Timestamp

  - x509_chain: An array of base64-encoded X.509 certificates. The first element is the end-entity certificate, the second chains to the first and so on.

The 'x509_chain' element can be empty or include as many certificates part of the same chain as available.

Note that 'gossip-data' is base64-encoded.

5. Security considerations

- TODO expand on why gossiping STH's is ok
- TODO expand on why gossiping SCT's is bad for privacy in the general case

6. Open questions

- TODO active vs. passive participants

7. IANA considerations

TBD

8. Contributors

TBD

9. Normative References

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

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

[draft-linus-trans-gossip]
"Transparency Gossip", n.d..

[draft-linus-trans-gossip-transport-https]
"Transparency Gossip HTTPS transport", n.d..

Author's Address

Linus Nordberg
NORDUnet

Email: linus@nordu.net