Public Notary Transparency S. Kent
Internet-Draft Independent
Intended status: Informational May 29, 2018
Expires: November 30, 2018

Attack and Threat Model for Certificate Transparency
draft-ietf-trans-threat-analysis-14

Abstract

This document describes an attack model and discusses threats for the Web PKI context for which security mechanisms to detect mis-issuance of web site certificates are being developed. The model provides an analysis of detection and remediation mechanisms for both syntactic and semantic mis-issuance. The model introduces an outline of attacks to organize the discussion. The model also describes the roles played by the elements of the Certificate Transparency (CT) system, to establish a context for the model.

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 https://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 November 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
  1.1. Conventions used in this document 8
2. Threats 8
3. Semantic mis-issuance 10
  3.1. Non-malicious Web PKI CA context 10
    3.1.1. Certificate logged 10
      3.1.1.1. Benign log 10
      3.1.1.2. Misbehaving log 11
      3.1.1.3. Misbehaving third party Monitor 12
    3.1.2. Certificate not logged 12
      3.1.2.1. CT-enabled browser 12
  3.2. Malicious Web PKI CA context 13
    3.2.1. Certificate logged 13
      3.2.1.1. Benign log 13
      3.2.1.2. Misbehaving log 14
      3.2.1.3. Misbehaving third party Monitor 14
    3.2.2. Certificate not logged 14
      3.2.2.1. CT-aware browser 15
  3.3. Undetected Compromise of CAs or Logs 15
    3.3.1. Compromised CA, Benign Log 16
    3.3.2. Benign CA, Compromised Log 17
    3.3.3. Compromised CA, Compromised Log 18
  3.4. Attacks Based on Exploiting Multiple Certificate Chains . 19
  3.5. Attacks Related to Distribution of Revocation Status . . 20
4. Syntactic mis-issuance 21
  4.1. Non-malicious Web PKI CA context 22
    4.1.1. Certificate logged 22
      4.1.1.1. Benign log 22
      4.1.1.2. Misbehaving log or third party Monitor 23
    4.1.2. Certificate not logged 23
      4.1.2.1. Self-monitoring Subject 23
    4.1.3. Situations Independent of Certificate Logging 24
      4.1.3.1. Self-monitoring Subject and Benign third party Monitor 24
      4.1.3.2. CT-enabled browser 24
  4.2. Malicious Web PKI CA context 24
    4.2.1. Certificate logged 25
      4.2.1.1. Benign log 25
      4.2.1.2. Misbehaving log or third party Monitor 25
      4.2.1.3. CT-enabled browser 25
    4.2.2. Certificate is not logged 26
5. Issues Applicable to Sections 3 and 4 26

  5.1. How does a Subject know which Monitor(s) to use? 26
  5.2. How does a Monitor discover new logs? 26
  5.3. CA response to report of a bogus or erroneous certificate 27
  5.4. Browser behavior 27
  5.5. Remediation for a malicious CA 27
  5.6. Auditing - detecting misbehaving logs 28
6. Security Considerations 29
7. IANA Considerations 29
8. Acknowledgments 29
9. References 30
  9.1. Normative References 30
  9.2. Informative References 30
  9.3. URIs 31
Author's Address 31

1. Introduction

Certificate transparency (CT) is a set of mechanisms designed to detect, deter, and facilitate remediation of certificate mis-issuance. The term certificate mis-issuance is defined here to encompass violations of either semantic or syntactic constraints. The fundamental semantic constraint for a certificate is that it was issued to an entity that is authorized to represent the Subject (or Subject Alternative) named in the certificate. (It is also assumed that the entity requested the certificate from the CA that issued it.) Throughout the remainder of this document we refer to a semantically mis-issued certificate as "bogus."

A certificate is characterized as syntactically mis-issued (aka erroneous) if it violates syntax constraints associated with the class of certificate that it purports to represent. Syntax constraints for certificates are established by certificate profiles, and typically are application-specific. For example, certificates used in the Web PKI environment might be characterized as domain validation (DV) or extended validation (EV) certificates. Certificates used with applications such as IPsec or S/MIME have different syntactic constraints from those in the Web PKI context.

There are three classes of beneficiaries of CT: certificate Subjects, CAs, and relying parties (RPs). In the initial focus context of CT, the Web PKI, Subjects are web sites and RPs are browsers employing HTTPS to access these web sites. The CAs that benefit are issuers of certificates used to authenticate web sites.

A certificate Subject benefits from CT because CT helps detect certificates that have been mis-issued in the name of that Subject. A Subject learns of a bogus certificate (issued in its name), via the Monitor function of CT. The Monitor function may be provided by the

Subject itself, i.e., self-monitoring, or by a third party trusted by the Subject. When a Subject is informed of certificate mis-issuance by a Monitor, the Subject is expected to request/demand revocation of the bogus certificate. Revocation of a bogus certificate is the primary means of remedying mis-issuance.

Certificate Revocations Lists (CRLs) [RFC5280] are the primary means of certificate revocation established by IETF standards. Unfortunately, most browsers do not make use of CRLs to check the revocation status of certificates presented by a TLS Server (Subject). Some browsers make use of Online Certificate Status Protocol (OCSP) data [RFC6960] as a standards-based alternative to CRLs. If a certificate contains an Authority Information Access (AIA) extension [RFC5280], it directs a relying party to an OCSP server to which a request can be directed. The data from this extension also may be used by a browser to request OCSP responses from a TLS server with which it is communicating [RFC6066][RFC6961].

RFC 5280 does not require inclusion of an AIA extension in certificates, so a browser cannot assume that this extension will be present. The Certification Authority and Browser Forum (CABF) baseline requirements and extended validation guidelines do mandate inclusion of this extension in EE certificates (in conjunction with their certificate policies). (See cabforum.org [1] for the most recent versions of these policies.)

In addition to the revocation status data dissemination mechanisms specified by IETF standards, most browser vendors employ proprietary means of conveying certificate revocation status information to their products, e.g., via a blacklist that enumerates revoked certificates (EE or CA). Such capabilities enable a browser vendor to cause browsers to reject any certificates on the blacklist. This approach also can be employed to remedy mis-issuance. Throughout the remainder of this document references to certificate revocation as a remedy encompass this and analogous forms of browser behavior, if available. Note: there are no IETF standards defining a browser blacklist capability.

Note that a Subject can benefit from the Monitor function of CT even if the Subject's certificate has not been logged. Monitoring of logs for certificates issued in the Subject's name suffices to detect mis-issuance targeting the Subject, if the bogus/erroneous certificate is logged.

A relying party (e.g., browser) benefits from CT if it rejects a bogus certificate, i.e., treats it as invalid. An RP is protected from accepting a bogus certificate if that certificate is revoked, and if the RP checks the revocation status of the certificate. (An

RP also is protected if a browser vendor "blacklists" a certificate or places a CA on a "bad-CA-list", causing all certs issued by the CA to be treated as invalid.) An RP also may benefit from CT if the RP validates an SCT associated with a certificate (see 8.1.3 in [I-D.ietf-trans-rfc6962-bis]), and rejects the certificate if the Signed certificate Timestamp (SCT) [I-D.ietf-trans-rfc6962-bis] is invalid. If an RP verified that a certificate that claims to have been logged has a valid log entry, the RP probably would have a higher degree of confidence that the certificate is genuine. However, checking logs in this fashion imposes a burden on RPs and on logs. Moreover, the existence of a log entry does not ensure that the certificate is not mis-issued. Unless the certificate Subject is monitoring the log(s) in question, a bogus certificate will not be detected by CT mechanisms. Finally, if an RP were to check logs for individual certificates, that would disclose to logs the identity of web sites being visited by the RP, a privacy violation. Thus this attack model does not assume that all RPs will check log entries.

A CA benefits from CT when it (acting as a Monitor for its clients) detects a (mis-issued) certificate that represents the same Subject name as a legitimate certificate issued by the CA.

Note that all RPs may benefit from CT even if they do nothing with SCTs. If Monitors inform Subjects of mis-issuance, and if a CA revokes a certificate in response to a request from the certificate's legitimate Subject, then an RP benefits without having to implement any CT-specific mechanisms.

Also note that one proposal [I-D.ietf-trans-gossip] for distributing Audit information (to detect misbehaving logs) calls for a browser to send SCTs it receives to the corresponding website when visited by the browser. If a website acquires an inclusion proof from a log for each (unique) SCT it receives in this fashion, this would cause a bogus SCT to be discovered, and, presumably, trigger a revocation request.

Logging [I-D.ietf-trans-rfc6962-bis] is the central element of CT. Logging enables a Monitor to detect a bogus certificate based on reference information provided by the certificate Subject. Logging of certificates is intended to deter mis-issuance, by creating a publicly-accessible record that associates a CA with any certificates that it mis-issues. Logging does not remedy mis-issuance; but it does facilitate remediation by providing the information needed to enable detection and subsequent revocation of bogus certificates in some circumstances.

Auditing is a function employed by CT to detect misbehavior by logs and to deter mis-issuance that is abetted by misbehaving logs.

Auditing detects several types of log misbehavior, including failures to adhere to the advertised Maximum Merge Delay (MMD) and Signed Tree Head (STH) frequency count [I-D.ietf-trans-rfc6962-bis] violating the append-only property, and providing inconsistent views of the log to different log clients. The first three of these are relatively easy for an individual auditor to detect, but the last form of misbehavior requires communication among multiple log clients. Monitors ought not trust logs that are detected misbehaving. Thus the Audit function does not detect mis-issuance per se. The CT design identifies audit functions designed to detect several types of misbehavior. However, mechanisms to detect some forms of log misbehavior are not yet standardized.

Figure 1 (below) illustrates the data exchanges among the major elements of the CT system, based on the log specification [I-D.ietf-trans-rfc6962-bis] and on the assumed behavior of other CT system elements as described above. This Figure does not include the Audit function, because there is not yet agreement on how that function will work in a distributed, privacy-preserving fashion.

```
+----+          +---------+          +---------+
| CA |---[1-]-->|   Log   |<---[8]---| Monitor |
|    |          |         |          |         |
|    |<--[2]----|         |----[9]-->|         |
|    |          |         |          |         |
|    |---[3]--->|         |<--[10]---|         |
|    |          |         |          |         |--------+
|    |<--[4]----|         |---[11]-->|         |        |
|    |          |         |          +---------+        |
|    |          |         |                             |
|    |          |         |          +---------+        |
|    |          |         |<--[8]----|  Self-  |        |
|    |          |         |          | Monitor |        |
|    |          |         |---[9]--->|(Subject)|        |
|    |          |         |          |         |        |
|    |          |         |<--[10]---|         |      [12]
|    |          |         |          |         |        |
|    |          |         |---[11]-->|         |        |
|    |          +---------+          +---------+        |
|    |                                                  |
|    |          +---------+          +---------+        |
|    |----[5]-->| Website |---[7]--->| Browser |        |
|    |          |(Subject)|          +---------+        |
|    |<---[6]-->|         |<----------------------------+
+----+          +---------+
```

[1] Retrieve accepted root certs
[2] accepted root certs
[3] Add chain to log/add PreCertChain to log
[4] SCT
[5] send cert + SCTs (or cert with embedded SCTs)
[6] Revocation request/response (in response to detected
     mis-issuance)
[7] cert + SCTs (or cert with embedded SCTs)
[8] Retrieve entries from Log
[9] returned entries from log
[10] Retrieve latest STH
[11] returned STH
[12] bogus/erroneous cert notification

Figure 1: Data Exchanges Between Major Elements of the CT System

Certificate mis-issuance may arise in one of several ways. The ways by which CT enables a Subject (or others) to detect and redress mis-issuance depends on the context and the entities involved in the mis-issuance. This attack model applies to use of CT in the Web PKI context. If CT is extended to apply to other contexts, each context

will require its own attack model, although most elements of the model described here are likely to be applicable.

Because certificates are issued by CAs, the top level differentiation in this analysis is whether the CA that mis-issued a certificate did so maliciously or not. Next, for each scenario, the model considers whether or not the certificate was logged. Scenarios are further differentiated based on whether the logs and monitors are benign or malicious and whether a certificate's Subject is self-monitoring or is using a third party Monitoring service. Finally, the analysis considers whether a browser is performing checking relevant to CT. The scenarios are organized as illustrated by the following outline:

```
   Web PKI CA - malicious vs non-malicious
      Certificate - logged vs not logged
            Log - benign vs malicious
            Third party Monitor - benign vs malicious
            Certificate's Subject - self-monitoring (or not)
            Browser - CT-supporting (or not)
```

The next section of the document briefly discusses threats. Subsequent sections examine each of the cases described above. As noted earlier, the focus here is on the Web PKI context, although most of the analysis is applicable to other PKI contexts.

1.1. Conventions used in this document

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 [RFC2119].

2. Threats

A threat is defined, traditionally, as a motivated, capable adversary. An adversary who is not motivated to attack a system is not a threat. An adversary who is motivated but not "capable" also is not a threat. Threats change over time; new classes of adversaries may arise, new motivations may come into play, and the capabilities of adversaries may change. Nonetheless, it is useful to document perceived threats against a system to provide a context for understanding attacks (even though some attacks may be the result of errors, not threats). Even if the assumptions about adversaries prove to be incorrect, documenting the assumptions is valuable.

As noted above, the goals of CT are to deter, detect, and facilitate remediation of attacks that result in certificate mis-issuance in the Web PKI. (CT also may facilitate detection and remediation of errors that result in certificate mis-issuance.) Such attacks can enable an

attacker to spoof the identity of TLS-enabled web sites. Spoofing enables an adversary to perform many types of attacks, e.g., delivery of malware to a client, reporting bogus information, or acquiring information that a client would not communicate if the client were aware of the spoofing. Such information may include personal identification and authentication information and electronic payment authorization information. Because of the nature of the information that may be divulged (or misinformation or malware that may be delivered), the principal adversaries in the CT context are perceived to be (cyber) criminals and nation states. Both adversaries are motivated to acquire personal identification and authentication information. Criminals are also motivated to acquire electronic payment authorization information.

To make use of bogus web site certificates, an adversary must be able to direct a TLS client to a spoofed web site, so that it can present the bogus certificate during a TLS handshake. An adversary may achieve this in various ways, e.g., by manipulation of the DNS response sent to a TLS client or via a man-in-the-middle attack. The former type of attack is well within the perceived capabilities of both classes of adversary. The latter attack may be possible for criminals and is certainly a capability available to a nation state within its borders. Nation states also may be able to compromise DNS servers outside their own jurisdiction.

The elements of CT may themselves be targets of attacks, as described below. A criminal organization might compromise a CA and cause it to issue bogus certificates, or it may exert influence over a CA (or CA staff) to do so, e.g., through extortion or physical threat. A CA may be the victim of social engineering, causing it to issue a certificate to an inappropriate Subject. (Even though the CA is not intentionally malicious in this case, the action is equivalent to a malicious CA, hence the use of the term "bogus" here.) A nation state may operate or influence a CA that is part of the large set of "root CAs" in browsers. A CA, acting in this fashion, is termed a "malicious" CA. A nation state also might compromise a CA in another country, to effect issuance of bogus certificates. In this case the (non-malicious) CA, upon detecting the compromise (perhaps because of CT) is expected to work with Subjects to remedy the mis-issuance.

A log also might be compromised by a suitably sophisticated criminal organization or by a nation state. Compromising a log would enable a compromised or rogue CA to acquire SCTs, but log entries would be suppressed, either for all log clients or for targeted clients (e.g., to selected Monitors or Auditors). It seems unlikely that a compromised, non-malicious, log would persist in presenting multiple views of its data, but a malicious log would.

Finally, note that a browser trust store may include a CA that is intended to issue certificates to enable monitoring of encrypted browser sessions. The inclusion of a trust anchor for such a CA is intended to facilitate monitoring encrypted content, via an authorized man-in-the-middle (MITM) attack. CT is not designed to counter this type of locally-authorized interception.

3. Semantic mis-issuance

3.1. Non-malicious Web PKI CA context

In this section, we address the case where the CA has no intent to issue a bogus certificate.

A CA may have mis-issued a certificate as a result of an error or, in the case of a bogus certificate, because it was the victim of a social engineering or a technical attack. In the case of an error, the CA should have a record of the erroneous certificate and be prepared to revoke this certificate once it has discovered and confirmed the error. In the event of a technical attack, a CA may have no record of a bogus certificate.

3.1.1. Certificate logged

3.1.1.1. Benign log

The log (or logs) is benign and thus is presumed to provide consistent, accurate responses to requests from all clients.

If a bogus (pre-)certificate has been submitted to one or more logs prior to issuance to acquire an embedded SCT, or post-issuance to acquire a standalone SCT, detection of this mis-issuance is the responsibility of a Monitor.

3.1.1.1.1. Self-monitoring Subject

If a Subject is tracking the log(s) to which a certificate was submitted, and is performing self-monitoring, then it will be able to detect a bogus (pre-)certificate and request revocation. In this case, the CA will make use of the log entry (supplied by the Subject) to determine the serial number of the bogus certificate, and investigate/revoke it. (See Sections 5.1, 5.2 and 5.3.)

3.1.1.1.2. Benign third party Monitor

If a benign third party monitor is checking the logs to which a certificate was submitted and is protecting the targeted Subject, it will detect a bogus certificate and will alert the Subject. The

Subject, in turn, will ask the CA to revoke the bogus certificate. In this case, the CA will make use of the log entry (supplied by the Subject) to determine the serial number of the bogus certificate, and revoke it (after investigation). (See Sections 5.1, 5.2 and 5.3.)

3.1.1.2. Misbehaving log

In this case, the bogus (pre-)certificate has been submitted to one or more logs, each of which generate an SCT for the submission. A misbehaving log probably will suppress a bogus certificate log entry, or it may create an entry for the certificate but report it selectively. (A misbehaving log also could create and report entries for bogus certificates that have not been issued by the indicated CA (hereafter called "fake"). Unless a Monitor validates the associated certificate chains up to roots that it trusts, these fake bogus certificates could cause the Monitors to report non-existent semantic problems to the Subject who would in turn report them to the purported issuing CA. This might cause the CA to do needless investigative work or perhaps incorrectly revoke and re-issue the Subject's real certificate. Note that for every certificate submitted to a log, the log must verify a complete certificate chain up to one of the roots it accepts. So creating a log entry for a fake bogus certificate marks the log as misbehaving.

3.1.1.2.1. Self-monitoring Subject & Benign third party Monitor

If a misbehaving log suppresses a bogus certificate log entry, a Subject performing self-monitoring will not detect the bogus certificate. CT relies on an Audit mechanism to detect log misbehavior, as a deterrent. It is anticipated that logs that are identified as persistently misbehaving will cease to be trusted by Monitors, non-malicious CAs, and by browser vendors. This assumption forms the basis for the perceived deterrent. It is not clear if mechanisms to detect this sort of log misbehavior will be viable.

Similarly, when a misbehaving log suppresses a bogus certificate log entry (or report such entries inconsistently) a benign third party Monitor that is protecting the targeted Subject also will not detect a bogus certificate. In this scenario, CT relies on a distributed Auditing mechanism [I-D.ietf-trans-gossip] to detect log misbehavior, as a deterrent. (See Section 5.6 below.) However, a Monitor (third-party or self) must participate in the Audit mechanism in order to become aware of log misbehavior.

If the misbehaving log has logged the bogus certificate when issuing the associated SCT, it will try to hide this from the Subject (if self-monitoring) or from the Monitor protecting the Subject. It does so by presenting them with a view of its log entries and STH that

does not contain the bogus certificate. To other entities, the log presents log entries and an STH that include the bogus certificate. This discrepancy can be detected if there is an exchange of information about the log entries and STH between the entities receiving the view that excludes the bogus certificate and entities that receive a view that includes it, i.e., a distributed Audit mechanism.

If a malicious log does not create an entry for a bogus certificate (for which an SCT has been issued), then any Monitor/Auditor that enrounters the bogus certificate (and SCT) will detect this when it checks with the log for log entries and STH (see Section 3.1.2.)

3.1.1.3. Misbehaving third party Monitor

A third party Monitor that misbehaves will not notify the targeted Subject of a bogus certificate. This is true irrespective of whether the Monitor checks the logs or whether the logs are benign or malicious/conspiring.

Note that independent of any mis-issuance on the part of the CA, a misbehaving Monitor could issue false warnings to a Subject that it protects. These could cause the Subject to report non-existent semantic problems to the issuing CA and cause the CA to do needless investigative work or perhaps incorrectly revoke and re-issue the Subject's certificate.

3.1.2. Certificate not logged

If the CA does not submit a pre-certificate to a log, whether a log is benign or misbehaving does not matter. The same is true if a Subject is issued a certificate without an SCT and does not log the certificate itself, to acquire an SCT. Also, since there is no log entry in this scenario, there is no difference in outcome between a benign and a misbehaving third party Monitor. In both cases, no Monitor (self or third-party) will detect a bogus certificate based on Monitor functions and there will be no consequent reporting of the problem to the Subject or by the Subject to the CA based on examination of log entries.

3.1.2.1. CT-enabled browser

If a browser rejects certificates without SCTs (see Section 5.4), CAs may be "encouraged" to log the certificates they issue. This, in turn, would make it easier for Monitors to detect bogus certificates. However, the CT architecture does not describe how such behavior by browsers can be deployed incrementally throughout the Internet. As a result, this attack model does not assume that browsers will reject a

certificate that is not accompanied by an SCT. In the CT architecture certificates have to be logged to enable Monitors to detect mis-issuance, and to trigger subsequent revocation. Thus the effectiveness of CT is diminished in this context.

3.2. Malicious Web PKI CA context

In this section, we address the scenario in which the mis-issuance is intentional, not due to error. The CA is not the victim but the attacker.

3.2.1. Certificate logged

3.2.1.1. Benign log

A bogus (pre-)certificate may be submitted to one or more benign logs prior to issuance, to acquire an embedded SCT, or post-issuance to acquire a standalone SCT. The log (or logs) replies correctly to requests from clients.

3.2.1.1.1. Self-monitoring Subject

If a Subject is checking the logs to which a certificate was submitted and is performing self-monitoring, it will be able to detect the bogus certificate and will request revocation. The CA may refuse to revoke, or may substantially delay revoking, the bogus certificate. For example, the CA could make excuses about inadequate proof that the certificate is bogus, or argue that it cannot quickly revoke the certificate because of legal concerns, etc. In this case, the CT mechanisms will have detected mis-issuance, but the information logged by CT may not suffice to remedy the problem. (See Sections 4 and 6.)

A malicious CA might revoke a bogus certificate to avoid having browser vendors take punitive action against the CA and/or to persuade them to not enter the bogus certificate on a vendor-maintained blacklist. However, the CA might provide a "good" OCSP response (from a server it operates) to a targeted browser instance as a way to circumvent the remediation nominally offered by revocation. No component of CT is tasked with detecting this sort of misbehavior by a CA. (The misbehavior is analogous to a log offering split views to different clients, as discussed later. The Audit element of CT is tasked with detecting this sort of attack.)

3.2.1.1.2. Benign third party Monitor

If a benign third party monitor is checking the logs to which a certificate was submitted and is protecting the targeted Subject, it will detect the bogus certificate and will alert the Subject. The Subject will then ask the CA to revoke the bogus certificate. As in 3.2.1.1.1, the CA may or may not revoke the certificate and it might revoke the certificate but provide "good" OCSP responses to a targeted browser instance.

3.2.1.2. Misbehaving log

A bogus (pre-)certificate may have been submitted to one or more logs that are misbehaving, e.g., conspiring with an attacker. These logs presumably issue SCTs, but will hide the log entries from some or all Monitors.

3.2.1.2.1. Monitors - third party and self

If log entries are hidden from a Monitor (third party or self), the Monitor will not be able to detect issuance of a bogus certificate.

The Audit function of CT is intended to detect logs that conspire to delay or suppress log entries (potentially selectively), based on consistency checking of logs. (See 3.1.1.2.2.) If a Monitor learns of misbehaving log operation, it alerts the Subjects that it is protecting, so that they no longer acquire SCTs from that log. The Monitor also avoids relying upon such a log in the future. However, unless a distributed Audit mechanism proves effective in detecting such misbehavior, CT cannot be relied upon to detect this form of mis-issuance. (See Section 5.6 below.)

3.2.1.3. Misbehaving third party Monitor

If the third party Monitor that is "protecting" the targeted Subject is misbehaving, then it will not notify the targeted Subject of any mis-issuance or of any malfeasant log behavior that it detects irrespective of whether the logs it checks are benign or malicious/conspiring. The CT architecture does not include any measures to detect misbehavior by third-party monitors.

3.2.2. Certificate not logged

Because the CA is presumed malicious, it may choose to not submit a (pre-)certificate to a log. This means there is no SCT for the certificate.

When a CA does not submit a certificate to a log, whether a log is benign or misbehaving does not matter. Also, since there is no log entry, there is no difference in behavior between a benign and a misbehaving third-party Monitor. Neither will report a problem to the Subject.

A bogus certificate would not be delivered to the legitimate Subject. So the Subject, acting as a self-Monitor, cannot detect the issuance of a bogus certificate in this case.

3.2.2.1. CT-aware browser

If careful browsers reject certificates without SCTs, CAs may be "encouraged" to log certificates (see section 5.4.) However, the CT architecture does not describe how such behavior by browsers can be deployed incrementally throughout the Internet. As a result, this attack model does not assume that browsers will reject a certificate that is not accompanied by an SCT. Since certificates have to be logged to enable detection of mis-issuance by Monitors, and to trigger subsequent revocation, the effectiveness of CT is diminished in this context.

3.3. Undetected Compromise of CAs or Logs

Sections 3.1 and 3.2 examined attacks in the context of non-malicious and malicious CAs, and benign and misbehaving logs. Another class of attacks might occur in the context of a non-malicious CA and/or a benign log. Specifically these CT elements might be compromised and the compromise might go undetected. Compromise of CAs and logs was noted in Section 2, as was coercion of a CA. As noted there, a compromised CA is almost equivalent to a malicious CA, and thus the discussions in Section 3.2 are applicable. Section 3.4 explores the undetected compromise of a CA in the context of attacks designed to issue a bogus certificate that might avoid revocation (because the certificate would appear on distinct certificate paths).

The section focuses on undetected compromise of CAs. Such compromises warrant some additional discussion, since some relying parties may see signed objects issued by the legitimate (non-malicious) CA, others may see signed objects from its compromised counterpart, and some may see objects from both. In the case of a compromised CA or log the adversary may have access to the private key used by a CA to sign certificates, or used by a log to sign SCTs and STHs. (An attacker might not have access to a CA or log private key per se. The attacker may be able to cause a CA to issue bogus certificates, or a log to generate bogus objects, and not have a record of them. The DigiNotar [2] case is an example of this sort of attack on a CA.) Because the compromise is undetected, there will be

no effort by a CA to have its certificate revoked or by a log to shut down the log.

3.3.1. Compromised CA, Benign Log

In the case of a compromised (non-malicious) CA, an attacker uses the purloined private key to generate a bogus certificate (that the CA would not knowingly issue). If this certificate is submitted to a (benign) log, then it subject to detection by a Monitor, as discussed in 3.1.1.1. If the bogus certificate is submitted to a misbehaving log, then an SCT can be generated, but there will be no entry for it, as discussed in 3.1.1.2. If the bogus certificate is not logged, then there will be no SCT, and the implications are as described in 3.1.2.

This sort of attack may be most effective if the CA that is the victim of the attack has issued a certificate for the targeted Subject. In this case the bogus certificate will then have the same certification path as the legitimate certificate, which may help hide the bogus certificate. However, means of remedying the attack are independent of this aspect, i.e., revocation can be effected irrespective of whether the targeted Subject received its certificate from the compromised CA.

A compromised (non-malicious) CA may be able to revoke the bogus certificate if it is detected by a Monitor, and the targeted Subject has been notified. It can do so only when the serial number of the bogus certificate is made known to this CA and assuming that the bogus certificate was not issued with an Authority Information Access (AIA) or CRL Distribution Point (CRL DP) extension that enables only the malicious twin to revoke the certificate. (The AIA extension in the bogus certificate could be used to direct relying parties to an OCSP server controlled by the malicious twin. The CRL DP extension could be used to direct relying parties to a CRL controlled by the malicious twin.) If the bogus certificate contains either extension, the compromised CA cannot effectively revoke it. However, the presence of either of these extensions provides some evidence that an entity other than the compromised CA issued the certificate in question. (If the extensions differ from those in other certificates issued by the compromised CA, that is suspicious.)

If the serial number of the bogus certificate is the same as for a valid, not-expired certificate issued by the CA (to the target or to another Subject), then revocation poses a problem. This is because revocation of the bogus certificate will also invalidate a legitimate certificate. This problem may cause the compromised CA to delay revocation, thus allowing the bogus certificate to remain a danger for a longer time.

The compromised CA may not realize that the bogus certificate was issued by a malicious twin; one occurrence of this sort might be regarded as an error, and not cause the CA to transition to a new key pair. (This assumes that the bogus certificate does not contain an AIA or CRL DP extension that wrests control of revocation from the compromised CA.) If the compromised CA does determine that its private key has been stolen, it probably will take some time to transition to a new key pair, and reissue certificates to all of its legitimate Subjects. Thus an attack of this sort probably will take a while to be remedied.

Also note that the malicious twin of the compromised CA may be capable of issuing its own CRL or OCSP responses, without changing any AIA/CRL DP data present in the targeted certificate. The revocation status data from the evil twin will appear as valid as those of the compromised CA. If the attacker has the ability to control the sources of revocation status data available to a targeted user (browser instance), then the user may not become aware of the attack.

A bogus certificate issued by the malicious CA will not match the SCT for the legitimate certificate, since they are not identical, e.g., at a minimum the private keys do not match. Thus a CT-aware browser that rejects certificates without SCTs (see 3.1.2.2) will reject a bogus certificate created under these circumstances if it is not logged. If the bogus certificate is logged it is subject to detection by Monitors. Because the CA is presumed to be malicious the CA may delay revocation or try to suppress revocation status (see Section 3.5) even when confronted with evidence of issuance of the bogus certificate. In this case, even browsers that require an SCT will still accept the bogus certificate until they become aware of its revocation status.

3.3.2. Benign CA, Compromised Log

A benign CA does not issue bogus certificates, except as a result of an accident or attack. So, in normal operation, it is not clear what behavior by a compromised log would yield an attack. If a bogus certificate is issued by a benign CA (under these circumstances) is submitted to a compromised (non-malicious) log, then both an SCT and a log entry will be created. Again, it is not clear what additional adverse actions the compromised log would perform to further an attack on CT.

It is worth noting that if a benign CA was attacked and thus issued one or more bogus certificates, then a malicious log might provide split views of its log to help conceal the bogus certificate from targeted users. Specifically, the log would show an accurate set of

log entries (and STHs) to most clients, but would maintain a separate log view for targeted users. This sort of attack motivates the need for Audit capabilities based on "gossiping" [I-D.ietf-trans-gossip]. However, even if such mechanisms are employed, they might be thwarted if a user is unable to exchange log information with trustworthy partners.

3.3.3. Compromised CA, Compromised Log

As noted in 3.4, an evil twin CA may issue a bogus certificate that contains the same Subject name as a legitimate certificate issued by the compromised CA. Alternatively, the bogus certificate may contain a different name but reuse a serial number from a valid, not revoked certificate issued by that CA.

An attacker who compromises a log might act in one of two ways. It might use the private key of the log only to generate SCTs for a malicious CA or the evil twin of a compromised CA. If a browser checks the signature on an SCT but does not contact a log to verify that the certificate appears in the log, then this is an effective attack strategy. Alternatively, the attacker might not only generate SCTs, but also pose as the compromised log, at least with regard to requests from targeted users. In the latter case, this "evil twin" log could respond to STH requests from targeted users, making appear that the compromised log was offering a split view (thus acting as a malicious log). To detect this attack an Auditor needs to employ a gossip mechanism that is able to acquire CT data from diverse sources, a feature not yet part of the base CT system.

An evil twin CA might submit a bogus certificate to the evil twin of a compromised log. (The same adversary may be controlling both.) The operator of the evil twin log can use the purloined private key to generate SCTs for certificates that have not been logged by its legitimate counterpart. These SCTs will appear valid relative to the public key associated with the legitimate log. However, an STH issued by the legitimate log will not correspond to a tree (maintained by the compromised log) containing these SCTs. Thus checking the SCTs issued by the evil twin log against STHs from the compromised log will identify this discrepancy. As noted above, if an attacker uses the key to generate log entries and respond to log queries, the effect is analogous to a malicious log.)

An Auditor checking for log consistency and with access to bogus SCTs, might conclude that the compromised log is acting maliciously, and is presenting a split view to its clients. In this fashion the compromised log may be shunned and forced to shut down. However, if an attacker targets a set of TLS clients that do not have access to the legitimate log, they may not be able to detect this

inconsistency. In this case CT would need to rely on a distributed gossiping audit mechanism to detect the compromise (see Section 5.6).

3.4. Attacks Based on Exploiting Multiple Certificate Chains

Section 3.2 examined attacks in which a malicious CA issued a bogus certificate and either tried to prevent the Subject from detecting the bogus certificate, or reported the bogus certificate as valid, to at least some relying parties, even if the Subject requested revocation. These attacks are limited in that if the bogus certificate is not submitted to a log, then it may not be accepted by CT-aware browsers, and submitting the bogus certificate to a log increases the chances that the CA's malicious behavior will be detected.

In general, if a CA is discovered to be acting maliciously, its certificates will no longer be accepted, either because its parent will revoke its CA certificate, its CA certificate will be added to browsers' blacklists, or both. However, a malicious CA may be able to obtain an SCT for each bogus certificate that it issues and continue to have those certificates accepted by relying parties even after its malicious behavior has been detected. It can do this by creating more than one path validation chain for the certificates, as shown in Figure 2.

```
          +-----------------+    +-----------------+
          |       CA A      |    |      CA B       |
          +-----------------+    +-----------------+
                         \          /
                          \        /
          CA certificate 1 \      / CA certificate 2
                            \    /
                       +----------------+
                       |  malicious CA  |
                       +----------------+
                               |
                               | bogus EE certificate
                               |
                      +------------------+
                      | targeted Subject |
                      +------------------+
```

Figure 2: Multiple Certificate Chains for a Bogus Certificate

In Figure 2, the malicious CA has been issued CA certificates by two different parent CAs. The parent CAs may be two different trust anchors, or one or both of them may be an intermediate CA (i.e., it is subordinate to some trust anchor). If both parent CAs are

intermediate CAs, they may be subordinate to the same trust anchor or to different trust anchors. The malicious CA may have obtained certificates from the two parents by applying to them for the certificates, or by compromising the parent CAs and creating the certificates without the knowledge of the CAs. If the malicious CA applied for its certificates from these CAs, it may have presented false information as input to the CA's normal issuance procedures, with the result that the CAs do not realize that a certificate with the same subject name and public key has been issued by another CA.

Because there are two certificate path validation chains, the malicious CA could provide the chain that includes CA A when submitting a bogus certificate to one or more logs, but an attacker (colluding with the malicious CA) could provide the chain that includes CA B to targeted browsers. If the CA's malicious behavior is detected, then CA A and browser vendors may be alerted (e.g., via the CT Monitor function) and revoke/blacklist CA certificate 1. However, CA certificate 2 does not appear in any logs, and CA A is unaware that CA B has issued a certificate to the malicious CA. Thus those who detected the malicious behavior may not discover the second chain and so may not alert CA B or browser vendors of the need to revoke/blacklist CA certificate 2. In this case, targeted browsers would continue to accept the bogus certificates issued by the malicious CA, since the certificate chain they are provided is valid and because the SCT issued for the bogus certificate it the same irrespective of which certificate chain is presented.

This sort of attack might be thwarted if all intermediate (i.e., CA) certificates had to be logged. In that case CA certificate 2 might be rejected by CT-aware browsers.

3.5. Attacks Related to Distribution of Revocation Status

A bogus certificate that has been revoked may still appear valid to a browser under certain circumstances. In part this is because the revocation information seen by a relying party is partly under the control of the CA and/or the certificate subject. As a result, different relying parties may be presented with different revocation information. This is true irrespective of whether revocation is effected via use of a CRL or OCSP. Additionally, an attacker can steer a browser to specific revocation status data via various means, preventing a targeted browser from acquiring accurate revocation status information for a bogus certificate.

The bogus certificate might contain an AIA extension pointing to an OCSP server controlled by the malicious CA (or the attacker). As noted in Section 3.2.1.1.1, the malicious CA could send a "good" OCSP response to a targeted browser instance, even if other parties are

provided with a "revoked" response. A TLS server can supply an OCSP response to a browser as part of the TLS handshake [RFC6961], if requested by the browser. A TLS server posing as the entity named in the bogus certificate also could acquire a "good" OCSP response from the malicious CA to effect the attack. Only if the browser relies upon a trusted, third-party OCSP responder, one not part of the collusion, would these OCSP-based attacks fail.

The bogus certificate could contain a CRL distribution point extension instead of an AIA extension. In that case a site supplying CRLs for the malicious CA could supply different CRLs to different requestors, in an attempt to hide the revocation status of the bogus certificate from targeted browser instances. This is analogous to a split-view attack effected by a CT log. However, as noted in Section 3.2.1.1 and 3.2.1.1.1, no element of CT is responsible for detecting inconsistent reporting of certificate revocation status data. (Monitoring in the CT context tracks log entries made by CAs or Subjects. Auditing is designed to detect misbehavior by logs, not by CAs per se.)

The failure of a bogus certificate to be detected as revoked (by a browser) is not the fault of CT. In the class of attacks described above, CT achieves its goal of detecting the bogus certificate when that certificate is logged and a Monitor observes the log entry. Detection is intended to trigger revocation, to effect remediation, the details of which are outside the scope of CT. However the SCT mechanism is intended to assure a relying party that certificate has been logged, is susceptible to being detected as bogus by a Monitor, and presumably will be revoked if detected as such. In the context of these attacks, because of the way revocation may be implemented, the assurance provided by the SCT may not have the anticipated effect.

This type of attack might be thwarted in several ways. If a malicious CA is discovered, a browser vendor might blacklist it by public key (not by its serial number and the name of the parent CA or by a hash of the certificate). This approach to revocation would cause CA certificate 2 to be rejected as well as CA certificate 1. However none of these mechanisms are part of the CT specification [I-D.ietf-trans-rfc6962-bis] nor general IETF PKI standards (e.g., [RFC5280]).

4. Syntactic mis-issuance

4.1. Non-malicious Web PKI CA context

This section analyzes the scenario in which the CA has no intent to issue a syntactically incorrect certificate. As noted in Section 1, we refer to a syntactically incorrect certificate as erroneous.

4.1.1. Certificate logged

4.1.1.1. Benign log

If a (pre)certificate is submitted to a benign log, syntactic mis-issuance can (optionally) be detected, and noted. This will happen only if the log performs syntactic checks in general, and if the log is capable of performing the checks applicable to the submitted (pre)certificate. (A (pre)certificate should be logged even if it fails syntactic validation; logging takes precedence over detection of syntactic mis-issuance.) If syntactic validation fails, this can be noted in an SCT extension returned to the submitter.

If the (pre)certificate is submitted by the non-malicious issuing CA, then the CA should remedy the syntactic problem and re-submit the (pre)certificate to a log or logs. If this is a pre-certificate submitted prior to issuance, syntactic checking by a log could help a CA detect and avoid issuance of an erroneous certificate. If the CA does not have a record of the certificate contents, then presumably it was a bogus certificate and the CA should revoke it.

If a certificate is submitted by its Subject, and is deemed erroneous, then the Subject should contact the issuing CA and request a new certificate. If the Subject is a legitimate subscriber of the CA, then the CA will either have a record of the certificate content or can obtain a copy of the certificate from the Subject. The CA will remedy the syntactic problem and either re-submit a corrected (pre-)certificate to a log and send it to the Subject or the Subject will re-submit it to a log. Here too syntactic checking by a log enables a Subject to be informed that its certificate is erroneous and thus may hasten issuance of a replacement certificate.

If a certificate is submitted by a third party, that party might contact the Subject or the issuing CA, but because the party is not the Subject of the certificate it is not clear how the CA will respond.

This analysis suggests that syntactic mis-issuance of a certificate can be avoided by a CA if it makes use of logs that are capable of performing these checks for the types of certificates that are submitted, and if the CA acts on the feedback it receives. If a CA uses a log that does not perform such checks, or if the CA requests

checking relative to criteria not supported by the log, then syntactic mis-issuance will not be detected or avoided by this mechanism. Similarly, syntactic mis-issuance can be remedied if a Subject submits a certificate to a log that performs syntactic checks, and if the Subject asks the issuing CA to fix problems detected by the log. (The issuer is presumed to be willing to re-issue the certificate, correcting any problems, because the issuing CA is not malicious.)

4.1.1.2. Misbehaving log or third party Monitor

A log or Monitor that is conspiring with the attacker or is independently malicious, will either not perform syntactic checks, even though it claims to do so, or simply not report errors. The log entry and the SCT for an erroneous certificate will assert that the certificate syntax was verified.

As with detection of semantic mis-issuance, a distributed Audit mechanism could, in principle, detect misbehavior by logs or Monitors with respect to syntactic checking. For example, if for a given certificate, some logs (or Monitors) are reporting syntactic errors and some that claim to do syntactic checking, are not reporting these errors, this is indicative of misbehavior by these logs and/or Monitors.

Note that a malicious log (or Monitor) could report syntactic errors for a syntactically valid certificate. This could result in reporting of non-existent syntactic problems to the issuing CA, which might cause the CA to do needless investigative work or perhaps incorrectly revoke and re-issue the Subject's certificate.

4.1.2. Certificate not logged

If a CA does not submit a certificate to a log, there can be no syntactic checking by the log. Detection of syntactic errors will depend on a Subject performing the requisite checks when it receives its certificate from a CA. A Monitor that performs syntactic checks on behalf of a Subject also could detect such problems, but the CT architecture does not require Monitors to perform such checks.

4.1.2.1. Self-monitoring Subject

A Subject performing self-monitoring will be able to detect the lack of an embedded SCT in the certificate it received from the CA, or the lack of an SCT supplied to the Subject via an out-of-band channel. A Subject ought to notify the CA if the Subject expected that its certificate was to be logged. (A Subject would expect its certificate to be logged if there is an agreement between the Subject

and the CA to do so, or because the CA advertises that it logs all of the certificates that it issues.) If the certificate was supposed to be logged, but was not, the CA can use the certificate supplied by the Subject to investigate and remedy the problem. In the context of a benign CA, a failure to log the certificate might be the result of an operations error, or evidence of an attack on the CA.

4.1.3. Situations Independent of Certificate Logging

4.1.3.1. Self-monitoring Subject and Benign third party Monitor

If a Subject or benign third party Monitor performs syntactic checks, it will detect the erroneous certificate and the issuing CA will be notified (by the Subject). If the Subject is a legitimate subscriber of the CA, then the CA will either have a record of the certificate content or can obtain a copy of the certificate from the Subject. The CA SHOULD revoke the erroneous certificate (after investigation) and remedy the syntactic problem. The CA SHOULD either re-submit the corrected (pre)certificate to one or more logs and then send the result to the Subject, or send the corrected certificate to the Subject, who will re-submit it to one or more logs.

4.1.3.2. CT-enabled browser

If a browser rejects an erroneous certificate and notifies the Subject and/or the issuing CA, then syntactic mis-issuance will be detected (see Section 5.) Unfortunately, experience suggests that many browsers do not perform thorough syntactic checks on certificates, and so it seems unlikely that browsers will be a reliable way to detect erroneous certificates. Moreover, a protocol used by a browser to notify a Subject and/or CA of an erroneous certificate represents a DoS potential, and thus may not be appropriate. Additionally, if a browser directly contacts a CA when an erroneous certificate is detected, this is a potential privacy violation, i.e., the CA learns that the browser user is visiting the web site in question. These observations argue for syntactic checking to be performed by other elements of the CT system, e.g., logs and/or Monitors.

4.2. Malicious Web PKI CA context

This section analyzes the scenario in which the CA's issuance of a syntactically incorrect certificate is intentional, not due to error. The CA is not the victim but the attacker.

Note that irrespective of whether syntactic checks are performed by a log, a malicious CA can acquire an embedded SCT, or post-issuance will acquire a standalone SCT for an erroneous certificate. If

Subjects or Monitors perform syntactic checks that detect the syntactic mis-issuance and report the problem to the CA, a malicious CA may do nothing or may delay the action(s) needed to remedy the problem.

4.2.1. Certificate logged

4.2.1.1. Benign log

Because the CA is presumed to be malicious, the CA might cause the log to not perform checks (if it had chosen to do so), in one of several ways.

1. The CA may assert that the certificate is being issued w/o regard to any guidelines (the "no guidelines" reserved CCID).

2. The CA may assert a CCID that has not been registered, and thus no log will be able to perform a check.

3. The CA may check to see which CCIDs a log declares it can check, and chose a registered CCID that is not checked by the log in question.

4. The CA may submit a (pre-) certificate to a log that is known to not perform any syntactic checks, and thus avoid syntactic checking.

4.2.1.2. Misbehaving log or third party Monitor

A misbehaving log or third party Monitor will either not perform syntactic checks or not report any problems that it discovers. (See 4.1.1.2 for further problems). Also, as noted above, the CT architecture includes no explicit provisions for detecting a misbehaving third-party Monitor.

4.2.1.3. CT-enabled browser

As noted above (4.1.1.4), most browsers fail to perform thorough syntax checks on certificates. Such browsers might benefit from having syntax checks performed by a log and reported in the SCT, although the pervasive nature of syntactically-defective certificates may limit the utility of such checks. (Remember, in this scenario, the log is benign.) However, if a browser does not discriminate against certificates that do not contain SCTs (or that are not accompanied by an SCT in the TLS handshake), only minimal benefits might accrue to the browser from syntax checks perform by logs or Monitors.

If a browser accepts certificates that do not appear to have been syntactically checked by a log (as indicated by the SCT), a malicious CA need not worry about failing a log-based check. Similarly, if there is no requirement for a browser to reject a certificate that was logged by an operator that does not perform syntactic checks, the fourth attack noted in 4.2.1.1 will succeed as well. If a browser were configured to know which versions of certificate types are applicable to its use of a certificate, the second and third attack strategies noted above could be thwarted.

4.2.2. Certificate is not logged

Since certificates are not logged in this scenario, a third-party Monitor cannot detect the issuance of an erroneous certificate based on examination of log entries. However, if a Subject informs a Monitor of the syntactic criteria applicable to the certificate it is supplying, the Monitor can perform syntactic checks on behalf of the Subject. Thus there is no difference between a benign or a malicious/conspiring log or a benign or conspiring/malicious Monitor. (Also note that a Subject MAY detect a syntax error by examining the certificate returned to it by the Issuer.) However, even if errors are detected and reported to the CA, a malicious/conspiring CA may do nothing to fix the problem or may delay action.

5. Issues Applicable to Sections 3 and 4

5.1. How does a Subject know which Monitor(s) to use?

If a CA submits a bogus certificate to one or more logs, but these logs are not tracked by a Monitor that is protecting the targeted Subject, CT will not remedy this type of mis-issuance attack. If third-party Monitors advertise which logs they track, Subjects may be able to use this information to select an appropriate Monitor (or set thereof). Also, it is not clear whether every third-party Monitor must offer to track every Subject that requests protection. If a Subject acts as its own Monitor, this problem is solved for that Subject.

5.2. How does a Monitor discover new logs?

It is not clear how a (self-)Monitor becomes aware of all (relevant) logs, including newly created logs. The means by which Monitors become aware of new logs must accommodate self-monitoring by a potentially very large number of web site operators. If there are many logs, it may not be feasible for a (self-) Monitor to track all of them, or to determine what set of logs suffice to ensure an adequate level of coverage.

5.3. CA response to report of a bogus or erroneous certificate

A CA being presented with evidence of a bogus or erroneous certificate, supported by a log entry and/or SCT, will need to examine its records to determine if it has knowledge of the certificate in question. It also will likely require the targeted Subject to provide assurances that it is the authorized entity representing the Subject name (subjectAltname) in question. Thus a Subject should not expect immediate revocation of a contested certificate. The time frame in which a CA will respond to a revocation request usually is described in the CPS for the CA. Other certificate fields and extensions may be of interest for forensic purposes, but are not required to effect. The SCT and log entry, because each contains a timestamp from a third party, is probably valuable for forensic purposes (assuming a non-conspiring log operator).

5.4. Browser behavior

If a browser is to reject a certificate that lacks an embedded SCT, or is not accompanied by an SCT transported via the TLS handshake, this behavior needs to be defined in a way that is compatible with incremental deployment. Issuing a warning to a (human) user is probably insufficient, based on experience with warnings displayed for expired certificates, lack of certificate revocation status information, and similar errors that violate RFC 5280 path validation rules [RFC5280]. Unless a mechanism is defined that accommodates incremental deployment of this capability, attackers probably will avoid submitting bogus certificates to (benign) logs as a means of evading detection.

5.5. Remediation for a malicious CA

A targeted Subject might ask the parent of a malicious CA to revoke the certificate of the non-cooperative CA. However, a request of this sort may be rejected, e.g., because of the potential for significant collateral damage. A browser might be configured to reject all certificates issued by the malicious CA, e.g., using a bad-CA-list distributed by a browser vendor. However, if the malicious CA has a sufficient number of legitimate clients, treating all of their certificates as bogus or erroneous still represents serious collateral damage. If this specification were to require that a browser can be configured to reject a specific, bogus or erroneous certificate identified by a Monitor, then the bogus or erroneous certificate could be rejected in that fashion. This remediation strategy calls for communication between Monitors and browsers, or between Monitors and browser vendors. Such communication has not been specified, i.e., there are no standard

ways to configure a browser to reject individual bogus or erroneous certificates based on information provided by an external entity such as a Monitor. Moreover, the same or another malicious CA could issue new bogus or erroneous certificates for the targeted Subject, which would have to be detected and rejected in this (as yet unspecified) fashion. Thus, for now, CT does not seem to provide a way to facilitate remediation of this form of attack, even though it provides a basis for detecting such attacks.

5.6. Auditing - detecting misbehaving logs

The combination of a malicious CA and one or more conspiring logs motivates the definition of an audit function, to detect conspiring logs. If a Monitor protecting a Subject does not see bogus certificates, it cannot alert the Subject. If one or more SCTs are present in a certificate, or passed via the TLS handshake, a browser has no way to know that the logged certificate is not visible to Monitors. Only if Monitors and browsers reject certificates that contain SCTs from conspiring logs (based on information from an auditor) will CT be able to detect and deter use of such logs. Thus the means by which a Monitor performing an audit function detects such logs, and informs browsers must be specified for CT to be effective in the context of misbehaving logs.

Absent a well-defined mechanism that enables Monitors to verify that data from logs are reported in a consistent fashion, CT cannot claim to provide protection against logs that are malicious or may conspire with, or are victims of, attackers effecting certificate mis-issuance. The mechanism needs to protect the privacy of users with respect to which web sites they visit. It needs to scale to accommodate a potentially large number of self-monitoring Subjects and a vast number of browsers, if browsers are part of the mechanism. Even when an Audit mechanism is defined, it will be necessary to describe how the CT system will deal with a misbehaving or compromised log. For example, will there be a mechanism to alert all browsers to reject SCTs issued by such a log? Absent a description of a remediation strategy to deal with misbehaving or compromised logs, CT cannot ensure detection of mis-issuance in a wide range of scenarios.

Monitors play a critical role in detecting semantic certificate mis-issuance, for Subjects that have requested monitoring of their certificates. A monitor (including a Subject performing self-monitoring) examines logs for certificates associated with one or more Subjects that are being "protected". A third-party Monitor must obtain a list of valid certificates for the Subject being monitored, in a secure manner, to use as a reference. It also must be able to identify and track a potentially large number of logs on behalf of

its Subjects. This may be a daunting task for Subjects that elect to perform self-monitoring.

Note: A Monitor should not rely on a CA or RA database for its reference information or use certificate discovery protocols; this information should be acquired by the Monitor based on reference certificates provided by a Subject. If a Monitor were to rely on a CA or RA database (for the CA that issued a targeted certificate), the Monitor would not detect mis-issuance due to malfeasance on the part of that CA or the RA, or due to compromise of the CA or the RA. If a CA or RA database is used, it would support detection of mis-issuance by an unauthorized CA.

As noted above, Monitors represent another target for adversaries who wish to effect certificate mis-issuance. If a Monitor is compromised by, or conspires with, an attacker, it will fail to alert a Subject to a bogus or erroneous certificate targeting that Subject, as noted above. It is suggested that a Subject request certificate monitoring from multiple sources to guard against such failures. Operation of a Monitor by a Subject, on its own behalf, avoids dependence on third party Monitors. However, the burden of Monitor operation may be viewed as too great for many web sites, and thus this mode of operation ought not be assumed to be universal when evaluating protection against Monitor compromise.

6. Security Considerations

An attack and threat model is, by definition, a security-centric document. Unlike a protocol description, a threat model does not create security problems nor does it purport to address security problems. This model postulates a set of threats (i.e., motivated, capable adversaries) and examines classes of attacks that these threats are capable of effecting, based on the motivations ascribed to the threats. It then analyses the ways in which the CT architecture addresses these attacks.

7. IANA Considerations

None.

8. Acknowledgments

The author would like to thank David Mandelberg and Karen Seo for their assistance in reviewing and preparing this document, and other members of the TRANS working group for reviewing it. Most of the text of Section 3.4 was provided by David Cooper, motivated by observations from Daniel Kahn Gilmor. Thanks also go to Daiming Li for her editorial assistance.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[I-D.ietf-trans-gossip] Nordberg, L., Gillmor, D., and T. Ritter, "Gossiping in CT", draft-ietf-trans-gossip-05 (work in progress), January 2018.

[I-D.kent-trans-extended-validation-cert-checks] Kent, S. and R. Andrews, "Syntactic and Semantic Checks for Extended Validation Certificates", draft-kent-trans-extended-validation-cert-checks-02 (work in progress), December 2015.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[RFC6961] Pettersen, Y., "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension", RFC 6961, DOI 10.17487/RFC6961, June 2013, <https://www.rfc-editor.org/info/rfc6961>.

9.3. URIs

[1] https://cabforum.org

[2] https://en.wikipedia.org/wiki/DigiNotar

Author's Address

Stephen Kent
Independent

Email: kent@alum.mit.edu