Public Notary Transparency
Internet Draft
Expires: January 2016
Intended Status: Informational

S. Kent
BBN Technologies

July 2015

Attack Model for Certificate Transparency
draft-ietf-trans-threat-analysis-02

Abstract

This document describes an attack model for the Web PKI context in which security mechanisms to detect mis-issuance of web site certificates will be 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

Table of Contents

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 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 two classes of beneficiaries of CT: certificate Subjects 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.

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. (If a browser vendor distributes a "blacklist" of bogus certificates or CAs that mis-issued certificates and modifies the browser to reject any certificates on the list or for which the issuing CA is on the list, this too remedies 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 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 is also protected if a browser vendor "blacklists" a certificate or a CA as noted above.) An RP also may benefit from CT if the RP validates an SCT associated with a certificate, and rejects the certificate if the SCT is invalid. If an RP verified that a

certificate that claims to have been logged has a valid log entry, the RP 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 assumes that RPs will not check log entries.

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.

Logging [TRANS] 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 revocation of bogus certificates in some circumstances.

Auditing is a function employed by CT to detect mis-behavior by logs. Auditing is intended to deter mis-issuance that is abetted by misbehaving logs. Auditing detects log mis-behavior by noting inconsistencies between SCTs issued by a log and the log entries published by the log. (A failure to make available log entries in a timely fashion, consistent with the MMD for the log is also a form of misbehavior that can be detected by the Audit function.) Such inconsistencies indicate log misbehavior and suggest that Monitors ought not trust such logs. Thus the Audit function does not detect mis-issuance per se. There is no agreed-upon Audit function design for CT at the time of this writing. As a result, the model merely notes where Auditing is needed to detect certain classes of attacks.

Figure 1 (below) illustrates the data exchanges among the major elements of the CT system, based on the log specification [TRANS] 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 (embedded in pre-cert, if applicable)
       [ 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 - careful (or not)
```

The following sections examine each of these cases. As noted above, the focus here is on the Web PKI context, although most of the analysis is applicable to other PKI contexts.

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. Semantic mis-issuance

2.1. 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 attack or an attack such as the one that affected DigiNotar [https://www.vasco.com/company/about_vasco/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx]. 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 an attack, a CA may have no record of a bogus certificate.

2.1.1. Certificate logged

2.1.1.1. 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 mis-issuance is the responsibility of a Monitor.

2.1.1.1.1. 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 4.1, 4.2 and 4.3.)

2.1.1.1.2. 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 4.1, 4.2 and 4.3.)

2.1.1.2. Mis-behaving 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. In this scenario, a log probably will suppress a bogus certificate log entry, or it may create an entry for the certificate but report it selectively. (A mis-behaving 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 mis-behaving.)

2.1.1.2.1. Self-monitoring Subject

If a mis-behaving 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 mis-behaving will cease to be trusted by Monitors and by non-malicious CAs.) It is not clear if such a mechanism will be viable if there are a very large number of self-monitoring Subjects.

2.1.1.2.2. Benign third party Monitor

Because a mis-behaving log will suppress 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 [gossip] to detect log misbehavior, as a deterrent. (See Section 4.6 below.) However, a Monitor (third-party or self) must participate in the Audit mechanism in order to become aware of log misbehavior.

If the mis-behaving log has logged the bogus certificate when it 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 sees the bogus certificate will detect this when it checks with the log for log entries and STH (see Section 2.1.2.)

2.1.1.3. Mis-behaving third party Monitor

A third party Monitor that mis-behaves 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 mis-behaving 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.

2.1.2. Certificate not logged

If the CA does not submit a pre-certificate to a log, whether a log is benign or mis-behaving 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 mis-behaving 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.

2.1.2.1. 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. The Subject SHOULD notify the CA if the Subject believes that its certificate was supposed to be logged. 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.

2.1.2.2. Careful browser

If a browser rejects certificates without SCTs (see Section 4.4.), CAs may be "encouraged" to log the certificates they issue. This, in

turn, would make it easier for Monitors to detect bogus certificates. However, it is not clear how such behavior by browsers can be deployed incrementally throughout the Internet. As a result, this model does not assume that browsers will reject a certificate that is not accompanied by an SCT. In the absence of logging, browsers generally do not benefit from CT, since certificates have to be logged to enable detection of mis-issuance by Monitors, and to trigger subsequent revocation.

2.2. 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.

2.2.1. Certificate logged

2.2.1.1. 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.

2.2.1.1.1. 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. 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 does not help remedy the problem. (See Sections 3 and 5.)

2.2.1.1.2. 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 2.2.1.1.1, the CA may or may not revoke the certificate.

2.2.1.2. Mis-behaving log

A bogus (pre-)certificate may have been submitted to one or more logs that are mis-behaving, e.g., conspiring with an attacker. These logs may or may not issue SCTs, but will hide the log entries from some or all Monitors.

2.2.1.2.1. 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 2.1.1.2.2.) 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 4.6 below.)

2.2.1.3. Mis-behaving third party Monitor

If the third party Monitor that is "protecting" the targeted Subject is mis-behaving, 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.

2.2.2. 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 mis-behaving does not matter. Also, since there is no log entry, there is no difference in behavior between a benign and a mis-behaving 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.

2.2.2.1. Careful browser

If careful browsers reject certificates without SCTs, CAs may be "encouraged" to log certificates (see section 4.4.) However, it is not clear how such behavior by browsers can be deployed incrementally throughout the Internet. As a result, this model does not assume that browsers will reject a certificate that is not accompanied by an SCT. In the absence of logging, browsers generally do not benefit from CT, since certificates have to be logged to enable detection of mis-issuance by Monitors, and to trigger subsequent revocation.

3. Syntactic mis-issuance

3.1. Non-malicious Web PKI CA context

This section analyzes the scenario in which the CA has no intent to issue a syntactically incorrect certificate. Throughout the remainder of this document we refer to a syntactically incorrect certificate as "erroneous".

3.1.1. Certificate logged

3.1.1.1. 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, and if the CA has a record of the certificate content, 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 helps 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.

Bottom line: 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.)

3.1.1.2. Mis-behaving 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 mis-behavior 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 which 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.

3.1.1.3. Self-monitoring Subject and Benign third-party Monitor

If a Subject or benign 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.

3.1.1.4. Careful 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 4.) 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. This argues for syntactic checking to be performed by other elements of the CT system, e.g., logs and/or Monitors.

3.1.2. 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.

3.2. 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.

3.2.1. Certificate logged

3.2.1.1. Benign log

Because the CA is presumed to be malicious, the CA may cause the log to not perform checks, in one of several ways. (See [DOMVAL] and [EXTVAL] for more details on validation checks and CCIDs).

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. In this fashion the CA can prevent the log from performing checks, and the SCT and log entry will not contain an indication of a failed check.

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.

3.2.1.2. Mis-behaving log or third party Monitor

A mis-behaving log or third party Monitor will either not perform syntactic checks or not report any problems that it discovers. (See 3.1.1.2 for further problems).

3.2.1.3. Self-monitoring Subject and Benign third party Monitor

Irrespective of whether syntactic checks are performed by a log, a malicious CA will acquire an embedded SCT, or post-issuance will acquire a standalone SCT. 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 action to remedy the problem.

3.2.1.4. Careful browser

As noted above (3.1.1.4) many browsers fail to perform thorough syntax checks on certificates. Such browsers would benefit from having such checks performed by a log and reported in the SCT. (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 them from syntax checks perform by logs.

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 approach noted in 3.2.1.1 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 strategies noted above could be thwarted.

3.2.2. Certificate is not logged

Since certificates are not logged in this scenario, a Monitor (third-party or self) cannot detect the issuance of an erroneous certificate. Thus there is no difference between a benign or a malicious/conspiring log or a benign or conspiring/malicious Monitor. (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.

4. Issues Applicable to Sections 2 and 3

4.1. 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. It is not clear whether every Monitor MUST offer to track every Subject that requests protection. Absent such a guarantee, how do Subjects know which set of Monitors will provide "sufficient" coverage? If a Subject acts as its own Monitor, this problem is solved for that Subject.

4.2. 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.

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

A CA being presented with evidence of a bogus or erroneous certificate, in the form of 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 revocation nor to verify that the certificate to be revoked is bogus or erroneous, based on applicable criteria. 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).

4.4. 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. 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.

4.5. 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 CA hot 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 remedy this form of attack, even though it provides a basis for detecting such attacks.

4.6. Auditing - detecting mis-behaving 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 mis-behaving 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 mis-behaving 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 mis-behaving 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 must not rely on a CA or RA database for its reference information or use certificate discovery protocols; this information must 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. A Monitor must not rely on certificate discovery mechanisms to build the list of valid certificates since such mechanisms might result in bogus or erroneous certificates being added to the list.

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.

5. Security Considerations

A 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.

6. IANA Considerations

None.

7. Acknowledgments

The author would like to thank David Mandelberg and Karen Seo for help with the editing and formatting, and other members of the TRANS working group for reviewing this document.

8. References

8.1. Normative References

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

8.2. Informative References

[TRANS] Laurie, B., Langley, A., Kasper, E., Messeri, E., Stradling, R., "Certificate Transparency," draft-ietf-trans-rfc6962-bis-07 (March 9, 2015), work in progress.

[DOMVAL] Kent, S., "Syntactic and Semantic Checks for Domain Validation Certificates," draft-kent-trans-domain-validation-cert-checks-01, (December 2014), work in progress.

[EXTVAL] Kent, S., "Syntactic and Semantic Checks for Extended Validation Certificates," draft-kent-trans-extended-validation-cert-checks-01 (December 2014), work in progress.

[gossip] Nordberg, L, Gillmore, D, "Gossiping in CT," draft-linus-trans-gossip-ct-01, (March 9, 2015), work in progress

Author's Addresses

Stephen Kent
BBN Technologies
10 Moulton Street
Cambridge MA 02138
USA

Phone: +1 (617) 873-3988
Email: skent@bbn.com

Copyright Statement