Public Notary Transparency
Internet Draft
Expires: December 2015
Intended Status: RFC -- Informational

Kent, S.
BBN Technologies

June 2015

Threat Analysis for Certificate Transparency

draft-ietf-trans-threat-analysis-00.txt

Abstract

This document describes a threat model for the Web PKI context in which security mechanisms to detect mis-issuance of web site certificates will be developed. The threat model covers both syntactic and semantic mis-issuance, using a taxonomy of threats starting with whether the mis-issuance was done by the CA maliciously or not; then whether or not the certificate was logged; and then whether the log(s) or monitor(s) are benign or malicious, whether the certificate subject is self-monitoring and whether a client is doing any checks.

Status of this Memo

Table of Contents

Attack Model and Discussion of Detection and Mitigation Options

1. Introduction

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.

Certificates are issued by CAs. So 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 term certificate mis-issuance is defined here to encompass issuance of a syntactically incorrect certificate or issuance of a certificate to an unauthorized party (semantically incorrect). (Throughout the remainder of this document we refer to a semantically incorrect certificate as ''bogus.'') This analysis addresses semantic mis-issuance first, then syntactic mis-issuance.

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.

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, (If there are many logs, it may not be feasible for a Subject to track all of them.) In this case, the CA will make use of the log entry (supplied by the Subject) to determine the serial number of the mis-issued certificate, and revoke it (after investigation). (See Notes 1 and 2.)

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, (If there are many logs, it may not be feasible for a Monitor to track all of them.) 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 Notes 1 and 2.)

2.1.1.2. Malicious or conspiring log

In this case, the bogus (pre-)certificate has been submitted to one or more logs that are either simply malicious or are conspiring with the attacker. It is assumed that the logs issue SCTs (in an attempt to fool browsers and/or Monitors). In this context, a log probably will suppress a bogus certificate log entry. (This case encompasses the scenario in which a log creates an entry for the certificate but reports it selectively.)

Note that a malicious log also could create and report entries for bogus certificates that have not been issued by the indicated CA. These could cause the Monitor to report non-existent semantic problems to the Subject who would in turn report them to the (apparently) issuing CA. This might cause the CA to do needless investigative work or perhaps incorrectly revoke and re-issue the Subject's certificate.

2.1.1.2.1. Self-monitoring Subject

If a malicious or conspiring log suppresses a bogus certificate log entry, a Subject performing self-monitoring will not detect the bogus certificate. In this scenario, CT relies on a ''gossiping'' mechanism to detect this sort of log misbehavior, as a deterrent. It is not clear if such a mechanism is viable if there are very large numbers of self-monitoring Subjects.

2.1.1.2.2. Benign third party Monitor

Because a malicious or conspiring log will suppress a bogus certificate log entry, 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 ''gossiping'' mechanism to detect this sort of log misbehavior, as a deterrent. However, a Monitor (third party or self) must participate in the gossiping mechanism in order to become aware of log misbehavior.

2.1.1.3. Malicious or conspiring third party Monitor

A third party Monitor that is conspiring with the entity that caused the mis-issuance, or a Monitor that is simply malicious 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 malicious 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 malicious/conspiring 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 malicious/conspiring third party Monitor. In both cases, there will be no reporting of the problem to the Subject 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 believed 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. (A failure to log the certificate might be the result of an operations error, or evidence of an attack.)

2.1.2.2. Careful browser

If a browser rejects certificates without SCTs and notifies the Subject and/or the issuing CA when no SCT is provided, this form of mis-issuance will be detected (see Note 3.) However, it is not clear how such behavior by browsers can be deployed incrementally throughout the Internet. Also, there is an obvious potential for DDoS attacks if browsers can be tricked into contacting CAs and/or Subjects based on this behavior. If, when an SCT is not provided, clients do not reject certificates and do not notify the CA or the Subject, this form of mis-issuance will not be detected unless the Subject is self-monitoring (See 2.1.2.1 and Note 3.)

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.

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. (If there are many logs, it may not be feasible for a Subject to track all of them.) 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 Notes 2 and 4.)

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. (If there are many logs, it may not be feasible for a Monitor to track all of them.) 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.

2.2.1.1.3. Malicious or conspiring third party Monitor

If the third party Monitor that is "protecting" the targeted Subject is malicious or is conspiring with the entity that caused the mis-issuance, then it will not notify the targeted Subject irrespective of whether the logs it checks are benign or malicious/conspiring.

2.2.1.2. Malicious or conspiring log

The bogus (pre-)certificate may have been submitted to one or more logs that are conspiring with the attacker. These logs may or may not issue SCTs, but will hide the log entries from some or all Monitors. In this case Monitors (third party and self) cannot detect issuance of a bogus certificate based on monitoring these logs.

The Audit function of CT is intended to detect logs that conspire to suppress log entries, based on consistency checking of logs and use of a ''gossip'' mechanism. If a Monitor learns of malfeasant log operation, it SHOULD alert the Subjects that it is protecting. The Monitor SHOULD also avoid using such a log. However, unless a gossip mechanism proves effective in detecting such misbehavior,, CT cannot be relied upon to detect this form of mis-issuance. (See Note 5 below.)

2.2.1.2.1. Malicious or conspiring third party Monitor

A conspiring third party Monitor will not notify the targeted Subject of any mis-issuance or of any malfeasant log behavior that it detects.

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 (pre-)submit a certificate to a log, whether a log is benign or malicious/conspiring does not matter. Also, since there is no log entry, there is no difference in behavior between a benign and a malicious/conspiring third party Monitor. Neither will report a problem to the Subject.

2.2.2.1. Self-monitoring Subject

A Subject performing self-monitoring will be able to detect the lack of SCT and notify the CA about the bogus certificate and request revocation. The CA may refuse to revoke, or may substantially delay revoking, the bogus certificate. It could make excuses about inadequate proof that the certificate is bogus, or argue that it cannot quickly revoke the certificate because of local, legal concerns, etc. In this case, the CT mechanisms have detected mis-issuance, but the information logged by CT does not help remedy the problem. (See Notes 2 and 4.)

2.2.2.2. Careful browser

If clients reject certificates without SCTs and notify the Subject and/or the issuing CA when no SCT is provided, this form of mis-issuance will be detected (see Note 3.) If, when an SCT is not provided, clients do not reject certificates and do not notify the CA or the Subject, this form of mis-issuance will succeed unless the Subject is self-monitoring (See 2.2.2.1 and Note 3.) However, it is not clear how such behavior by browsers can be deployed incrementally throughout the Internet. Also, there is an obvious potential for DDoS attacks if browsers can be tricked into contacting CAs and/or Subjects based on this behavior.

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 will be noted in the SCT 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 a malformed 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 it 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 malformed and thus may hasten issuance of a replacement certificate.

. If a (pre-)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. Malicious or conspiring 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 ''gossip'' mechanism could reveal 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 (pre-)certificate to one or more logs and then send the result to the Subject, or send the certificate to the Subject, who will re-submit it to one or more logs.

3.1.1.4. Careful browser

If TLS clients reject erroneous certificates and notify the Subject and/or the issuing CA, then syntactic mis-issuance will be detected (see Note 3.) 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. This argues for syntactic checking 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 Subjects or Monitors performing the requisite checks.

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. Malicious or conspiring log or third party Monitor

A malicious or conspiring 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/conspiring 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 would accrue to them from syntax checks perform by logs.

If a TLS clients 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 TLS client 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 client 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, the Monitor function cannot detect the issuance of an erroneous certificate (based on examination of logs). Thus there is no difference between a benign or a malicious/conspiring log or a benign or conspiring/malicious Monitor. A self-Monitoring Subject also will not detect the error based on examination of log entries. (A Subject MAY detect a syntax error by examining the certificate returned to the Subject.) 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. Notes Applicable to Sections 2 and 3

1. 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 mitigate 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. It also is not clear how a 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.

2. A CA being presented with evidence of a bogus certificate, in the form of a log entry, 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, 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).

3. If a TLS client were 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 (non-conspiring) logs as a means of evading detection.

4. A targeted Subject might request 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 them as bogus still represents serious collateral damage. If this specification were to require that a browser can be configured to

reject a specific, bogus certificate identified by a Monitor, then the bogus certificate could be rejected in that fashion. This mitigation 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 certificates based on information provided by an external entity such as a Monitor. Moreover, the same or another malicious CA could issue new bogus 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 mitigate this form of attack, even though it provides a basis for detecting such attacks.

5. 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 s 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 client has no way to know that the logged certificate is not visible to Monitors. Only if Monitors and clients reject certificates that contain SCTs from conspiring logs (based on info from an audit) will CT be able to deter use of such logs. Thus the means by which a Monitor performing an audit function detects such logs, and informs TLS clients must be specified for this to be effective.

Absent a ''gossip'' 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. Developing such a mechanism is not easy. The mechanism SHOULD 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 a gossip 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 TLS clients to reject SCTs issued by such a log? Absent a description of a mitigation 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. It 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 this 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 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 certificate targeting that Subject, as noted above. It is RECOMMENDED 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-00, (December 2014), work in progress.

[EXTVAL] Kent, S., ''Syntactic and Semantic Checks for Extended Validation Certificates,'' draft-kent-trans-extended-validation-cert-checks-00 (December 2014), 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