3.3 Malicious, Colluding CAs and/or Undetcted CA Compromises

Section 3.2 examined attacks in which a single CA might issue a bogus certificate. There is also the potential that two or more CAs might collude to issue a bogus certificate in a fashion designed to foil the remediation (but not detection) safeguards envisioned by CT. Undetected compromise of one or more CAs also could be used to effect equivalent attacks. The goal of such attacks is to trick a CT-aware browser into accepting a bogus certificate because it was accompanied by a valid SCT, while evading certificate revocation status indications. This section explores such scenarios.

These attacks may take place without the knowledge or assistance of trust anchors; any pair of intermediate CAs can effect this attack without the knowledge of superior CAs (which are presumed to be benign). (The following text refers to these as two CAs, because they might be represented by entities that are organizationally distinct, perhaps realized by different physical presences. However, because they share the same name and key pair, one also might view them as the same CA that appears in two certificate paths.) The two CAs must have the same Subject name and the same public key for the attack. (RFC 5280 does not explicitly preclude the creation of two CAs with the same name, so long as the parent CAs are distinct. Requirements for Subject name uniqueness apply individually to each CA but not across CA boundaries, as per Section 4.2.1.6. However, the Security Considerations section of RFC 5280 warns that name collisions could cause security problems.)

Two instances of a CA with the same name and key pair (call them CA-A and CA-B) could arise in several ways. Both might be malicious, created expressly for purposes of this attack. In this scenario an attacker needs to locate prospective parent CAs that are authorized to create subordinate CAs and that will not detect and reject an attempt to create CAs with the same name and key pair. The parents are presumed to be distinct and do not have name constraints that prevent both from certifying subordinate CAs with the same name. Compromise of a single parent to issue two distinct certificates with the same Subject name and public key would not achieve the goals of this attack. This is because the revocation of one instance of the bogus certificate would cause the second bogus certificate instance to be revoked as well. In this scenario CA-A issues the bogus certificate for X and logs it. CA-B will appear to be a valid issuer of the same certificate.

Another scenario calls for one CA (CA-A) to be the victim of an attack that results in issuing, and logging, the bogus certificate for X. The attacker then creates a second CA (CA-B) with the same name and public key, under a different parent, so that the bogus certificate for X will be valid under that CA as well.

Because the CA-A and CA-B have the same name and make use of the same key, a bogus certificate issued under either CA, targeting a Subject (X) is valid when processed under a certificate chain involving either CA-A or CA-B. One of the CAs (say, CA-A)ould log the certificate and acquire an SCT for it, while the other would not.

Because the bogus certificate is logged, it is subject to detection as such by a Monitor. Once the bogus certificate is detected it is anticipated that action will be taken to render it invalid. The bogus certificate itself might be revoked by CA-A that issued and logged it. This action masks the malicious intent of CA-A if it is malicious. If CA-A was the victim of an attack, its action is expected A browser vendor might add the bogus certificate to a blacklist maintained by the vendor, e.g., if CA-A failed to revoke it or maybe as a precautionary measure.

If CA-A is suspected of being malicious, e.g., because it has a history of using bogus certificates, the certificate of that CA might itself be revoked. This revocation might be effected by the parent of that CA (which is assumed to not be complicit), or by a browser vendor using a blacklist. Whether the proposed attack can achieve its goal depends on which revocation mechanism is employed, and which certificate or certificates are revoked.

3.3.1 Revocation of the Bogus Certificate

If the bogus (EE) certificate is revoked by CA-A, browsers should treat that certificate as invalid. However, a browser checking a CRL or OCSP response might not match this revocation status data against the bogus certificate when it is viewed as issued by CA-B. This is because revocation status checking is performed in the context of a certification path (during path validation). The bogus certificate (X) has two different certification paths and thus the revocation status data for each might be acquired and managed independently. (RFC 5280 does not provide implementation guidance for management of revocation data. It is known that some relying party implementations maintain such information on a per-certificate path basis, but others might not.)

Even if the bogus certificate contains an AIA extension pointing to an OCSP server the attack might still succeed. (As noted in the Section 1, RFC 5280 does not mandate inclusion this extension, but its presence is required by CABF requirements.) As noted in Section 3.2.1.1.1, a malicious CA could send a "good" OCSP response to a targeted browser instance, even if other parties are provided with a "revoked" response. Also note that 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 could acquire a "good" OCSP response from the

a malicious CA (e.g., CA-B) to effect the attack. Only if the browser relies upon a trusted, third-party OCSP responder, one not part of the collusion, would the attack fail.

The analysis above also applies to the use of CRLs to disseminate certificate revocation status data. The bogus certificate could contain a CRL distribution point extension instead of an AIA extension. In that case a site supplying CRLs for CA-B 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.)

If CA-A (who logged the certificate) does not revoke it, a browser vendor might enter the bogus certificate into a "blacklist". Unfortunately, there are no IETF standards for such blacklists. Thus it is conceivable that the revocation status data also might be managed in a path-specific fashion. If that were true, then the attack could succeed. However, if a vendor maintains revocation status data in a path-independent fashion, then the attack will fail. For example, if revoked certificates are identified by CA name and serial number, or a hash of the certificate, this attack would fail.

3.3.2 Revocation of a CA Certificate

If CA-A is viewed as acting maliciously, its parent might revoke that CA's certificate. Even though CA-B has the same name and uses the same public key, its certificates is distinct from that of CA-A, e.g., it was issued by a different parent and almost certainly has a different certificate serial number. Thus revocation of the certificate of CA-A does not affect the certificate of CA-B. In this case, the bogus EE certificate would be treated as valid when it appears in a certification path involving CA-B. Thus revocation the certificate for CA-A does not prevent this attack from succeeding.

A vendor also might choose to add the certificate of CA-A to its blacklist, e.g., if that CA refuses to revoke the bogus certificate. This also may not prevent the bogus certificate from being accepted by a browser. For example, if the CA certificate blacklist entry is analogous to a CRL entry (Subject name of the parent of the malicious CA and the serial number of the malicious CA's certificate), the colluding CA's certificate would still be valid in this case.