Proposal for Better Revocation Model of SSL Certificates

Mohammed Almeshekah

meshekah@gmail.com

September 27, 2013

1 Introduction

There are multiple initiatives that have emerged in the past couple of years to address some of the inherit limitation of the CA-based trust model and SSL certificates. One of the these limitations is the current revocation models. In this document we present a proposal to address such limitations. The proposal is divided into two main phases; short-term solution and long-term solution.

2 Current Challenges with Revocation Models

There are significant problems with the way that Firefox is currently evaluating certificate to check for revocation. The current revocation checking in FireFox "CertVerifier" procedure is the following, check the end-entity certificate only by pinging the OCSP server, if no response is received fail safely and assume the certificate is trusted.

Traditionally, there are only two methods to revoke a certificate; *Certificate Revocation Lists (CRLs)* and *Online Certificate Status Protocol (OCSP)*. In the next paragraphs we discuss the major challenges with these two current methods in terms of security, privacy, performance and usability.

2.1 Security Challenges

- Nonsensical security properties of revocation checking of end-entity certificates: In most cases, a malicous actor that is trying to use a revoked certificate to attack a browser user will be able to turn off the revocation checking for the certificate he is using.
- Nonsensical security properties of revocation checking for intermediate CA certificates: We don't check for revocation of intermediate certificates at all except for the case of EV. A bad intermediate CA

certificate is extremely dangerous, so it is important to check revocation of them; sadly, the intermediates we do check revocation for (EV intermediate certificates) are the ones that are the least likely to cause our users security problems. And, those revocation checks suffer from the same problem that revocation checking of end-entity certificates currently has: an attacker can usually just block the check and prevent us from seeing that the certificate has been revoked.

- The two models, and specifically OCSP, is be design and single point of failure model. If the CA's server does not respond with the CRL or the OCSP response the revocation decision cannot be established. Moreover, OCSP servers are known to have a not very good up time []. As a result of that, major browsers implement a *soft-fail* approach where is a response in not received the certificate will be accepted [].

2.2 Privacy Challenges

- The CA learns the IP address, location, a subset of the user's browsing history, and other sensitive information about the user through the OCSP to its servers.

2.3 Performance Challenges

- Revocation checking through OCSP and CRL requests is way too slow.

2.4 Usability Challenges

- Many captive portals with HTTPS login pages work very poorly in Firefox because we stall for 30+ seconds waiting for the OCSP response for the captive portal that is being blocked by the captive portal until you log in.

- Confusing UX for EV certificates: If we fail to get revocation information via OCSP/CRL fetching for an EV certificate, then we do not show the certificate as an EV certificate. This is particularly problematic for cases when a web app is designed to be used offline (e.g. using AppCache), but even normal websites like paypal.com are affected by this. This inconsistency in the security indicators devalues the security indicators.

3 Interesting Data

A script was written to extract some information from the Alexa top 1 million website revocation dynamics. The following data points where interesting:

- In figure 1 Below we show the distribution of the revocation reasons for the Alexa top 1 million websites.
- In figure 2 and figure 3 we show the certificate types in the same group of million web pages.
- There were 987 CRLs referenced by the Alexa top million sites. However, out of these there were 215 CRLs who have no revoked certificates.
- The total number of revoked certificates in these CRLs is (2,650,548).
- Only 19 CRLs (out of 772) were touched by 70.39% of the one million connections touched. These CRLs revoked 125,429 certificates only. Their total size was 4.2 MB.
- If we include the information of the top 102 CRLs that revoked 492,238 certificates, we will be satisfying 93.24% of the connections. The total size of those is 18.3 MB.
- 369 CRLs were referred to in CAs certificates (determined by Basic Constraint). Only 223 were responsive.
- The total number of revoked certificates in those 223 CRLs were 389,633.

We also analyzed the revocation information propose by Google in their CRLSet solution. The following interesting data point were gathered from Google's latest CRLSet:

- The latest update of Google's CRLSet have (24,156) revoked certificates in their set.
- These certificates were revoked by 46 CAs only.

4 Important Considerations

To have a good revocation model(s) the following issues must be taking into consideration.

- **Security:** Does the new model introduce new trust anchors, maintain the current trust anchors or reduces/limits the current trust anchors?
- **Security:** Does the new model has false positives/negatives?
- **Security:** How easy it is to add/delete/modify revocation information?
- **Privacy:** Does the new model violates the user's privacy?
- **Deployability:** Does the new model require changes at the user's side (browsers), the server's side and/or the CA side? Requiring changes at the user's side only require a handful of major browsers

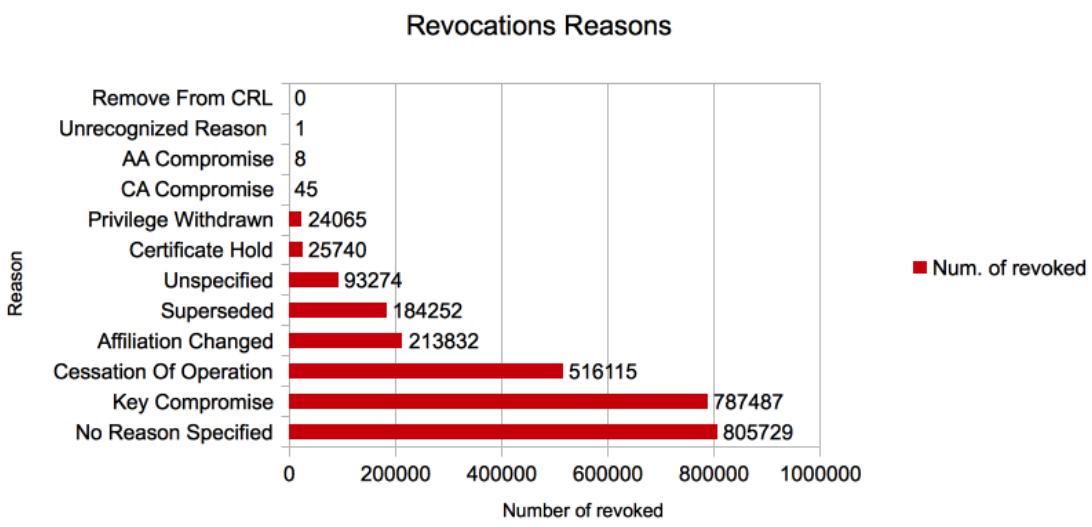


Figure 1: Revocation Reasons

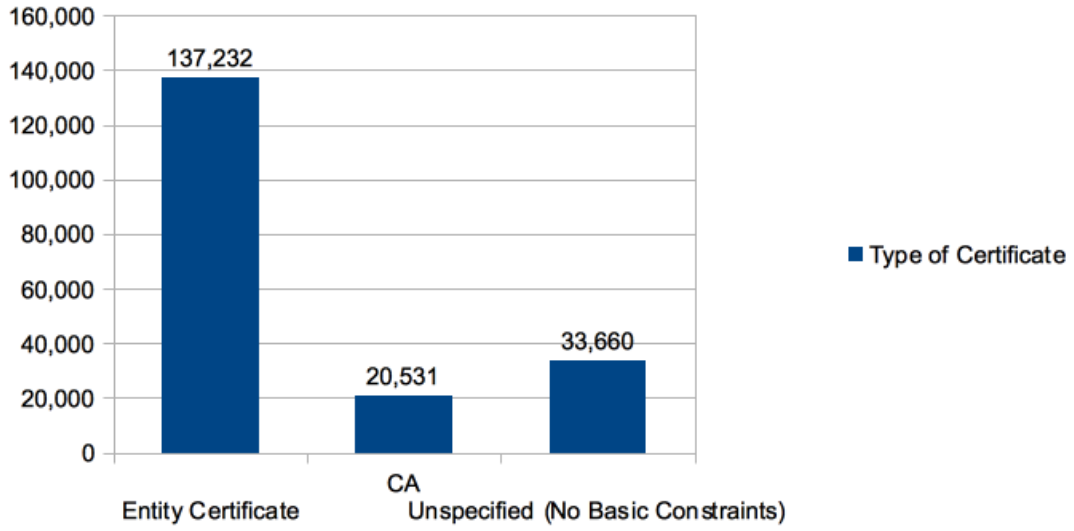
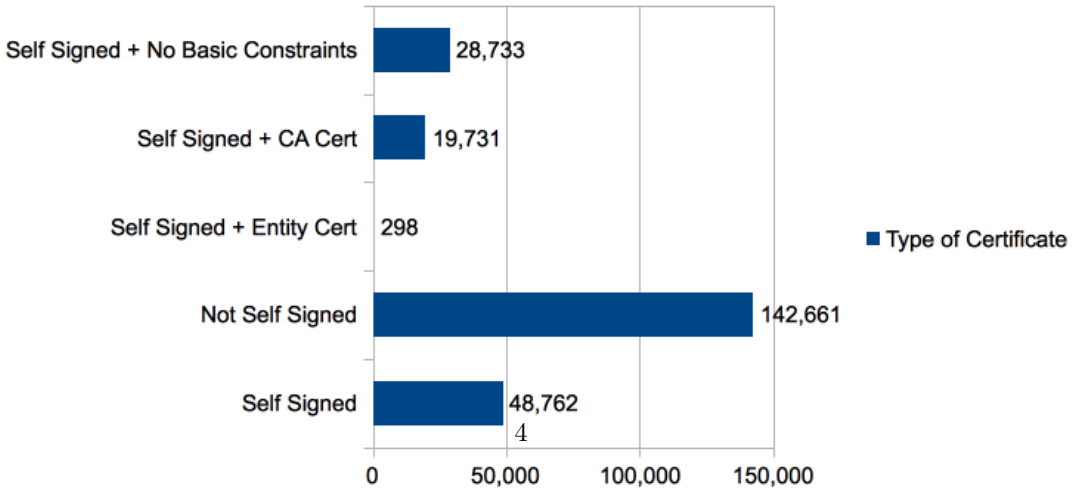


Figure 2: Certificate Types 1



vendor to update theit clients. Changes at the CA level impact a couple of hundered CAs. However, changes at the servers' side require changes at a scale of hundereds of millions of machines.

- **Deployability:** Does the new model requires bootstrapping or can it be imcemently deployed?
- **Deployability:** Is the new model scalable to the Internet size?
- **Deployability:** Does the new model depend on implementing other protocols such as DNSSEC.
- **Usability:** Does the new model require user interaction? Also, does it affect the user's experience?
- **Performace:** Does the new model have a performace impact on opening and SSL encrypted page. In other words, does the model introdce an extra communication step or does it bundle this information with something else.
- **Performace:** Does the model adds an extra storage/computation overhead on the the user's side, the server's side and/or the CA side?

5 Short-term Proposal

To improve certificates revocation in FireFox we had to implement a short-term solution to the problem. The solution is divided into two processes; preloaded CRLSet and OCSP stapling. This document is concerned with implementation of CRLSet, while OCSP stapling has been done by David Keeler.

5.1 Overview

- CRLSet based on current information in the CRLs. Decisions are made based on the revocation reason.
- OCSP must-staple as an HTTP header.

5.2 What to Revoke

The CRLSet will consist of the list of certificates revoked in the PKIX. Based on Brian's proposal and the traditional X509 CRL extention we can group revoked certificates as in the table below based on the revocation reason. For the automation of the process we need to automatically decide which certificate to be included

Reason	**Description**	**Included?**
Unspecified (code: 0)	The reason of revoking is not specified (usually this is for business reasons).	No

Key Compromise (code: 1)	The key associated with the certificate was compromised.	No (OCSP must-staple)
CA Compromise (code: 2)	The CA has been compromised	Yes
Affiliation Change (code: 3)	The owner of this certificate is no longer affiliated with the issuer of this certificate	No
Superseded (code: 4)	Another certificate replaces this one	No (OCSP must staple)
Cessation of Operation (code: 5)	The CA who issued this certificate is ceased to operate	No (just add this CA to the set)
Certificate Hold (code: 6)	The certificate is currently not valid, but could become valid in the future	No
Remove from CRL (code: 8)	This only has meaning in delta CRLs where the certificate was on hold and now it should be removed	No
Privilege Withdrawn (code: 9)	The privilages granted to the subject has changed.	Yes
AA Compromise (code: 10)	Same as Key Compromise but for Authority Attribute certificates	No (OCSP must-staple)
Unrecognized code	If unstandardized code is specified.	No.

For "Key Compromise" and "AA Compromise" we must provide website a way for them to revoke certificates. Currently, the only way to do that is to implement OCSP must-staple. To be able to push the CRLSet, we must provide websites with a method to protect themselves in cases such as key compromise.

5.3 Addressed and Remaining Challenges

The current short-term implementation is a preloaded CRLSet. A remaining challenges are the following:

- We need to be able to quickly update the CRLSet for revocation information. This is the next step of the solution by implementing the dynamic CRLSet.
- With a dynamic CRLSet we need to implement a process to detect if revocation information has not been updated for a while (i.e. some entity is actively blocking the update of the revocation information in FireFox).

5.4 Implementation

The CRLSet provided at Mozilla site has the following structure:

- Number of issuers.
- list of revoked certificate serial numbers grouped by the hash of the issuer's public key component. The issuers information starts with a space.

5.4.1 CRLSet

Data Structure

The combination of both the certificate's *issuer* information and the certificates *serial number* uniquely identifies a certificate. Therefore, we suggest the following data structure for the CRLSet that is pushed to the clients:

- Version.
- Number of issuers.
- For every group of certificates by the same issuers we will have the following:
  - The issuer's public key hash base64 encoded. This line should start with a leading space.
  - Number of revoked certificates for that issuer.
  - List of revoked certificates' serial numbers (one per line).

We choose to group the certificates by the *Subject Public Key Info* as this this uniquely identifies an issuer (in fact, a certificate) out of all others. This is not true for the issuer's name as intermediate CAs can have the same name.

This list must be signed by a special purpose signing key and the corresponding public key should be hardcoded in Firefox.

Updating

Firefox should automatically update the list once a day without having to restart the browser. If the list is not updated within the last 7 days a notification will be prompted to the user.

This CRLSet should be updated once every day. There is two ways to update the list of revoked certificates:

- Automatically pulling the CRLs from the list of URIs we have and checking whether anything have changed.
- When a CA manually notifies us of a revocations.

Also, the list of CRL URI we have should updated once a month by checking if there were any new CRLs.

Finally, for how long should we keep a revoked certificate in the list or should the list grow indefinately? One idea is to remove a certificate from the list after a certain threshold from the certificate expiration date. However, this will require to change the handling of expired certificates within Friefox to insure a hard-fail of connection when this threshold is reached.

6 Long-term Proposal

6.1 Overview

- Stop using CRLs and limit OCSP servers to server to server interactions.
- Implement CRLSet based on the CA-caused revocations.
- Use OCSP must-stample as X.509 extension and use it for server-caused revocations.

6.2 What to Revoke

Reason	**Description**	**Resposible Party**
Mis-issuance	The CA issued a certificate for a domain to somebody that doesn't own that domain.	Issuing CA
Technical issue with the certificate	There is a problem with the certificate such as a broken algorithm or key size that is too small or an important constraint is missing.	Issuing CA
Browser policy change	The browser changes its policy such that the certificate should no longer be considered trusted.	Issuing CA
Loss of domain ownership	The owner of a domain at the time a certificate was issued no longer owns the domain.	Issuing CA for EV; Current website owner otherwise.
Private key compromise	The legitimate owner of a certificate leaked his private key.	Current website owner
Customer's whim	The customer asked to have the certificate revoked for some non-security reason. The reason is probably unspecified.	Current website owner

Takeover of a domain	A new entity takes ownership of a domain for which certificates were previously issued to somebody else.	Current website owner.