

Trust Revoked — Practical Evaluation of OCSP- and CRL-Checking Implementations

Manuel Koschuch^(✉) and Ronald Wagner

Competence Centre for IT-Security, FH Campus Wien, University of Applied Sciences, Favoritenstrasse 226, 1100 Vienna, Austria

manuel.koschuch@fh-campuswien.ac.at, ronald.wagner@rowag.at
<https://fh-campuswien.ac.at/it-security>

Abstract. When deploying asymmetric cryptography robust ways to reliably link a public key to a certain identity have to be devised. The current standard for doing so are X.509v3 certificates. They are used in HTTPS and SSH as well as in code-, e-mail-, or PDF-signing. This widespread use necessitates the need for an efficient way of revoking such certificates in case of a compromised private key. Two methods are currently available to deal with this problem: the older Certificate Revocation Lists (CRL), and the newer Online Certificate Status Protocol (OCSP). In this work we perform a practical evaluation of how different software like web-browsers or PDF viewers deal with OCSP, in particular when the OCSP server cannot be reached. We find widely varying behavior, from silently accepting any certificates to completely blocking access. In addition we search an existing data-set of X.509v3 HTTPS certificates for revocation information, finding that almost 85 % of them contain neither CRL nor OCSP information, thereby rendering any practical revocation attempt nearly useless.

Keywords: OCSP · CRL · X.509v3 · Browser · Evaluation

1 Introduction

The recently (4/2014) published *Heartbleed*¹ bug is only the last in a long running series of attack vectors [1] against one of the foundations of secure Internet communication, the TLS protocol. This bug has gained special notoriety due to the fact that it allows to extract a server's private key, requiring the affected server to replace the leaked key and, consequently, also to establish new certificates and revoke the old ones.

Revocation of a certificate prior to the natural end of its validity is, at least in theory, well supported by the X.509v3 standard, using mechanisms like *Certificate Revocation Lists* (CRLs) and the *Online Certificate Status Protocol* (OCSP). In practice, however, things look quite different, and the way these protocols are implemented in different frameworks varies by a good degree.

¹ <http://heartbleed.com/>.

In this work we present the first results of our preliminary comparison of different browsers (like Internet Explorer, Chrome, and Firefox), software (Java, Flash installation packages, Adobe Acrobat), and operating systems (Windows, Ubuntu), with the goal to determine how the different systems react when they are unable to verify the revocation status of a given certificate using either CRLs or OCSP.

In addition to this we also try to quantify how many certificates in practice actually contain revocation information, using an existing data-set from the ZMap project [2].

To give context to our results we start by giving an overview of the X.509v3 certificate and the mechanisms used in CRLs and OCSP in Sect. 2. Section 3 then details our experimental approach and presents our preliminary results. Finally, Sect. 4 summarizes our results and provides a short outlook on future work to be done in this area and also on currently available (or planned) alternatives to CRL and OCSP.

2 X.509 Certificates

Asymmetric cryptography solves the key distribution problem present with symmetric algorithms, but creating a new one by doing so: the need to verify the authenticity and integrity of an entity’s public key. Almost all systems in wide use today use certificates for this purpose, binding an identity (be it a real name, a mail address, or a domain name) to a public key. Figure 1 gives a schematic overview of the contents of such a certificate, as specified by the X.509v3 standard ([3], last updated in [4]). The *subject* field contains information about the owner of the public key present in the *subjectPublicKeyInfo* field, *issuer* specifies the trusted third party having signed the certificate (that is, all the fields with a bold frame in Fig. 1) in the *signatureValue* field, while finally the *extensions* field contains an arbitrary number of other information, marked as either *critical* (meaning that implementations which don’t understand or implement this extension have to abort processing the certificate) or *non-critical*.

Usually the lifetime of a certificate (and, consequently, of the public key associated with this certificate) is limited by dates given in the *validity* field, which can range from several months for individual end-user certificates up to several decades for CA certificates.

However, in practice it may be necessary to revoke a key at an earlier point in time, for example due to compromise of the private key, compromise of the CA, and so on (see [3, 5.3.1] for an enumeration of more possible reasons). To achieve this, two mechanisms are available in X.509v3: Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP).

2.1 Certificate Revocation Lists

Certificate Revocation Lists (CRLs) are the older method of revoking certificates, first defined in [5], with the latest update in [3]. The main idea behind this

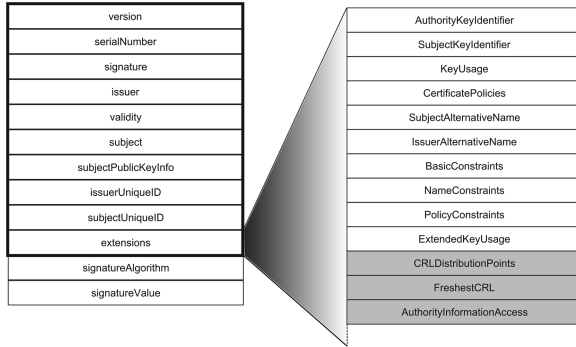


Fig. 1. X.509v3 certificate with some selected extensions. Parts covered by the signature are indicated by a bold frame, fields that contain CRL or OCSP information have a grey background (cf. [3]).

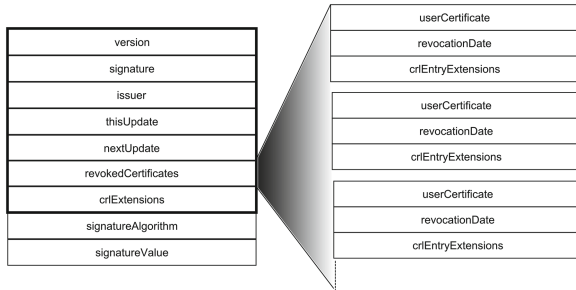


Fig. 2. Format of a certificate revocation list, parts covered by the signature are indicated by a bold frame (cf. [3]).

approach is simple: the Certification Authority (CA) periodically publishes a signed list containing all revoked certificates, or, in potentially shorter intervals, so called “delta lists” containing only the differences to the last full update. The certificates issued by this CA contain the address of the CRL distribution in the *CRLDistributionPoints* field for full CRLs and the *FreshestCRL* field for delta lists, respectively. Both fields are covered by the CA’s signature (as depicted in Fig. 1) and thus cannot be manipulated by an adversary after issuing the certificate.

Figure 2 gives an overview of the contents of such a CRL, where the parts covered by the CA’s signature are again indicated by a bold frame. The *revokedCertificates* field contains a list of certificate serial numbers together with the corresponding revocation date.

This approach suffers from two main problems: for one, it doesn’t scale very well. Once a certificate is added to a CRL, it becomes virtually impossible to remove it again, even if its regular validity has already expired (since there are still implementations, like for example mailing applications, that can and do also work with expired certificates), resulting in ever-growing lists that have to be delivered to each requesting client that subsequently has to parse the entire list.

On the other hand, the periodic issuing of CRLs creates periods of time where a revoked certificate might not have been added to the list yet and is thus still considered valid by client applications.

2.2 Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP), as defined in [6], tries to alleviate some of the CRL’s problems by adopting an interactive “challenge-response”-like approach (see Fig. 3). When a client wants to determine the validity of a certificate, it sends a (possibly signed) *OCSPRequest* to the responder given in the *AuthorityInformationAccess* field (see also Fig. 1), containing the serial number as well as a hash of the issuer’s name and public key of the certificate in question.

The OCSP responder then looks up the requested certificate in its database and replies with a signed² response, indicating whether this particular certificate has been revoked or not. While basically scaling better than the CRL approach, additional load is put on the CA’s OCSP responder, which now has to handle each individual request. A possible way to alleviate this problem is to employ *OCSP stapling*, as defined in [7]: here the certificate owner (which in practice usually is the website’s server in the case of HTTPS) periodically requests a validation of his own certificate from the CA and sends this validation together with his certificate to connecting clients. Since the validation is signed by the CA, a malicious server is unable to forge this information.

This reduces the pressure on the responder, but again introduces uncertainty periods, where a revoked certificate is still considered valid by the client.

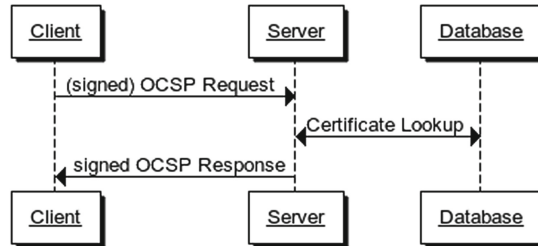


Fig. 3. Schematic representation of an OCSP protocol run. The request *may* be signed, the response, when containing actual data, *must* be signed (cf. [6]).

3 Practical Evaluation

Our practical evaluation was twofold: first, we were interested in how many certificates provide the location of a CRL, how many provide an OCSP responder, and corresponding combinations of these two values. For this we used the data-set collected in [8], containing a total of 66,335,624 HTTPS certificates.

² Note that there is one possible response that can be sent without signing: the status code “3”, meaning *tryLater*, which lead to subtle attacks against this protocol [10].

Table 1. Evaluated software packages, the individual cells give the version we used, with “-” indicating that no testing of this software was conducted under the given operating system.

Software	Operating system		
	Windows XP	Windows 7	Ubuntu 13.04
Internet explorer	8.0.6001.18702	11.0.9600.17041	-
Firefox	28.0		26.0
Safari	5.1.7		-
Opera	20.0.1387.91		12.16
Chrome	34.0.1847.116		
Outlook	-	14.0.7116.5000	-
Java	7u55		
Adobe acrobat professional	-	8.0.0	-
Adobe flash player installation	-	13.0.0.182	-

From these, 9, 833, 063 (roughly 15 %) contain a CRL entry, 9, 295, 779 (approx. 14 %) an OCSP entry, 9, 249, 263 (again approx. 14 %) contained both, and 56, 456, 045 (that is almost 85 %) contained neither (note that from the 9, 295, 779 certificates containing an OCSP entry, only 7, 130, 220 (that is approx. 11 % of the total number of certificates) actually contain the string ‘OCSP’ in the corresponding *authorityInfoAccess* field). So we start with the insight that only about every fifth HTTPS certificate actually contains revocation information.

Subsequently we performed a preliminary analysis of how different frameworks under different operating systems react to an unreachable OCSP responder. Table 1 gives an overview of the software tested, together with the corresponding operating system and version number.

For the browsers we used the two HTTPS demo sites from <https://www.pki.dfn.de/crl/globalocsp/>. <https://info.pca.dfn.de/> uses a valid certificate containing OCSP information, the certificate of the site <https://revoked-demo.pca.dfn.de/> is revoked, which again can be verified using OCSP. Both sites were accessed using the browser’s default settings, first without any modifications to the network connection, then with an active Checkpoint Gaia R76 firewall blocking access to the OCSP URL given in the certificates.

Figure 4 gives an overview of our preliminary findings, where the individual quadrants (starting in the upper right corner and then proceeding clockwise) represent the cases of

- (I) browsers that *accept* a *revoked* certificate
- (II) browsers that *block* a *revoked* certificate
- (III) browsers that *block* a *valid* certificate
- (IV) browsers that *accept* a *valid* certificate.

In addition each quadrant is divided into one part for the case when access to the OCSP responder is possible, and another one when this access is blocked.

Certificate Valid		Certificate Revoked	
OCSP Reachable	OCSP Blocked	OCSP Reachable	OCSP Blocked
Internet Explorer 8.0.6001.18702/WinXP Internet Explorer 11.0.9600.17041/Win7 Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Safari 5.1.7/WinXP Safari 5.1.7/Win7 Opera 20.0.1387.91/WinXP Opera 20.0.1387.91/Win7 Opera 12.16/Ubuntu 13.04 Chrome 34.0.1847.116/WinXP Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	Internet Explorer 8.0.6001.18702/WinXP Internet Explorer 11.0.9600.17041/Win7 Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Safari 5.1.7/WinXP Safari 5.1.7/Win7 Opera 20.0.1387.91/WinXP Opera 20.0.1387.91/Win7 Opera 12.16/Ubuntu 13.04 Chrome 34.0.1847.116/WinXP Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	Safari 5.1.7/WinXP Safari 5.1.7/Win7 Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Safari 5.1.7/WinXP Safari 5.1.7/Win7 Opera 12.16/Ubuntu 13.04 Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04
Accept	Accept	Accept	Accept
Blocked		Blocked	
OCSP Reachable OCSP Blocked		OCSP Reachable OCSP Blocked	
Certificate Valid		Certificate Revoked	
Internet Explorer 8.0.6001.18702/WinXP Internet Explorer 11.0.9600.17041/Win7 Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Opera 20.0.1387.91/WinXP Opera 20.0.1387.91/Win7 Opera 12.16/Ubuntu 13.04 Chrome 34.0.1847.116/WinXP		Internet Explorer 8.0.6001.18702/WinXP Internet Explorer 11.0.9600.17041/Win7 Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Opera 20.0.1387.91/WinXP Opera 20.0.1387.91/Win7 Opera 12.16/Ubuntu 13.04 Chrome 34.0.1847.116/WinXP	
Blocked		Blocked	

Fig. 4. Comparison of the reactions of the browsers tested when OCSP was reachable or blocked, respectively. The quadrant on the upper left contains the browsers that accepted a valid certificate when OCSP was blocked/reachable. The quadrant on the upper right contains the browsers that accepted a revoked certificate when OCSP was blocked/reachable. The browsers indicated in bold are those that always performed according to what one would expect to be the correct behavior (i.e. blocking a revoked certificate and accepting a valid one).

Browsers that always perform according to the usual expectations (i.e. blocking revoked certificates and allowing valid ones) are marked in bold. The results vary wildly depending on browser and operating system, with Chrome effectively ignoring OCSP altogether (as is also detailed in [9] and basically stems mainly from usability reasons in practice).

To summarize our findings with the other software tested:

- The signed Java web start application we tested (<https://pki.pca.dfn.de/guira/guira.jnlp>) ran in every browser without any warnings, whether OCSP was blocked or not.
- The validity of the signature on a PDF document is considered ‘unknown’ when OCSP access is blocked.
- An e-mail signature is shown as ‘valid’ in Outlook 2010 when OCSP is blocked, but appears as ‘not verifiable’ when examining the signature details of the message.
- The installation of the signed Flash player executable for Windows 7 works without any warning whatsoever when OCSP is blocked.

4 Conclusions and Future Work

In this work we performed a very preliminary evaluation of the reaction of different browsers and other software using certificates on how they react to blocked revocation checking. We find that it is next to impossible to give a consistent

picture of how software reacts to inaccessible OCSP and/or CRL URLs, with everything from quietly ignoring this fact to asking the user on how to proceed to downright blocking access to the specific web-page. In addition to that, by using the existing data-set from [8] we find almost 85 % of HTTPS certificates don't contain any revocation information at all, thereby rendering this approach to deal with compromised keys next to useless in practice.

Our next steps will be to perform a more thorough testing of the different browsers with respect to the reaction of blocked CRLs (something we only did very inconsistently in this work) as well as of other software making use of certificates. But our current results already imply that the current practice of dealing with compromised keys, be it OCSP or CRLs, does not suffice to actually avoid users from mistakenly trusting compromised certificates. This result and conclusion is also in-line with those of some of the people behind the Chromium and Chrome project, resulting in Chrome completely having given up on CRL and OCSP and using their own CRLSet approach [11], effectively an offline CRL that is periodically pushed to the end-devices by the browser implementer. We also plan to evaluate the effectiveness of this approach in practice considering a real-world set of revoked certificates.

Acknowledgements. Manuel Koschuch is being supported by the MA23 - Wirtschaft, Arbeit und Statistik - in the course of the funding programme "Stiftungsprofessuren und Kompetenzteams für die Wiener Fachhochschul-Ausbildungen".

References

1. Meyer, C., Schwenk, J.: SoK: lessons learned from SSL/TLS attacks. In: Kim, Y., Lee, H., Perrig, A. (eds.) WISA 2013. LNCS, vol. 8267, pp. 172–189. Springer, Heidelberg (2014)
2. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: fast internet-wide scanning and its security applications. In: Proceedings of the 22nd USENIX Security Symposium, pp. 605–620. USENIX Association, Berkeley (2013)
3. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC (2008)
4. Yee, P.: RFC6818 - Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC (2013)
5. Housley, R., Ford, W., Polk, W., Solo, D.: RFC2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC (1999)
6. Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: RFC6960 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC (2013)
7. Eastlake, D.: RFC6066 - Transport Layer Security (TLS) Extensions, Extension Definitions. RFC (2011)
8. Durumeric, Z., Kasten, J., Bailey, M., Halderman, J.A.: Analysis of the HTTPS Certificate Ecosystem. In: Proceedings of the 13th Internet Measurement Conference, pp. 291–304. ACM, New York (2013)

9. Langley, A.: No, don't enable revocation checking (2014). <https://www.imperial-violet.org/2014/04/19/revchecking.html>
10. Marlinspike, M.: Defeating OCSP with the Character '3' (2009). <http://www.thoughtcrime.org/papers/ocsp-attack.pdf>
11. Langley, A.: Revocation checking and Chrome's CRL (2012). <https://www.imperialviolet.org/2012/02/05/crlsets.html>

E-Business and Telecommunications
11th International Joint Conference, ICETE 2014,
Vienna, Austria, August 28-30, 2014, Revised Selected
Papers
Obaidat, M.S.; Holzinger, A.; Filipe, J. (Eds.)
2015, XXIII, 538 p. 204 illus. in color., Softcover
ISBN: 978-3-319-25914-7