Papers, Please...
X.509 Certificate Revocation in Practice

Manuel Koschuch[1] and Ronald Wagner[2]
[1]*Competence Centre for IT-Security, FH Campus Wien - University of Applied Sciences, Favoritenstrasse 226, 1100 Vienna, Austria*
[2]*FH Campus Wien - University of Applied Sciences, Favoritenstrasse 226, 1100 Vienna, Austria*

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

Abstract: X.509v3 certificates are the current standard of verifiable associating an entity with a public key, and are widely used in different networking applications: from HTTPS in browsers, SSH connections, to e-mail, PDF and code signing. This wide usage also necessitates the existence of a robust, reliable way to detect and deal with compromised or otherwise invalid certificates. Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP) are the two mechanisms currently deployed to handle revoked certificates. In this position paper we present preliminary results of our research into the practical use of these protocols, using an existing data-set to show that almost 85% of certificates currently in use contain no revocation information, and compare different browsers under different operating systems as to their dealing with unreachable OCSP servers. We find that browser behaviour in this case ranges from opening the site without any warnings whatsoever to totally blocking it, indicating no clear default reaction and no reliable behaviour.

1 INTRODUCTION

The recently (4/2014) published *Heartbleed* bug [1] is only the last in a long running series of attack vectors (Meyer and Schwenk, 2013) 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.

In this position paper 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 (Durumeric et al., 2013b).

This position paper is now structured as follows: Section 2 gives an overview of the X.509v3 certificate and the mechanisms used in CRLs and OSCP. Section 3 then details our experimental approach and presents our preliminary results. Finally, Section 4 summarizes our results and provides a short outlook on future work to be done in this area.

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

[1]http://heartbleed.com/

Koschuch M. and Wagner R..
Papers, Please... - X.509 Certificate Revocation in Practice.
DOI: 10.5220/0005113800360040
In *Proceedings of the 5th International Conference on Data Communication Networking* (DCNET-2014), pages 36-40
ISBN: 978-989-758-042-0

standard (Cooper et al., 2008), last updated in (Yee, 2013). 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 Figure 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 (Cooper et al., 2008, 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).

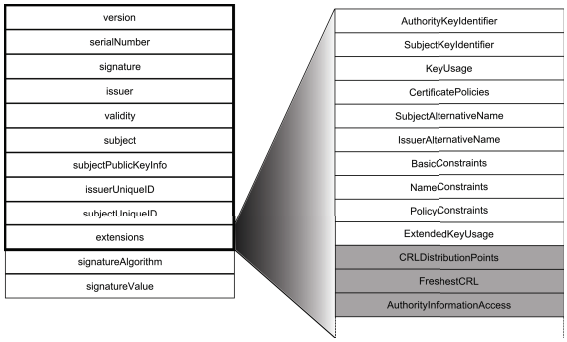

Figure 1: X.509v3 certificate with some selected extensions. Parts covered by the signature are indicated by a bold frame, fields that contain CRL or OCSP information have a grey background. (cf. (Cooper et al., 2008))

2.1 Certificate Revocation Lists

Certificate Revocation Lists (CRLs) are the older method of revoking certificates, first defined in (Housley et al., 1999), with the latest update in (Cooper et al., 2008). The main idea behind this 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 Figure 1) and thus cannot be manipulated by an adversary after issuing the certificate.

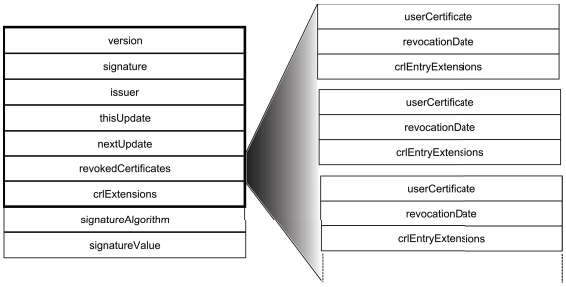

Figure 2: Format of a certificate revocation list, parts covered by the signature are indicated by a bold frame. (cf. (Cooper et al., 2008))

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 (Santesson et al., 2013), tries to alleviate some of the CRL's problems by adopting an interactive "challenge-response"-like approach (see Figure 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 Figure 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 signed[2]

[2]note that there is a possible response that can be sent without signing: the status code "3", meaning *tryLater*, which lead to subtle attacks against this protocol (Marlinspike, 2009).

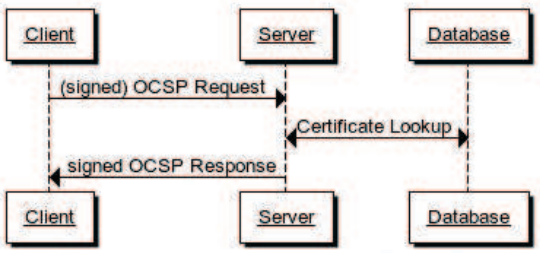

Figure 3: Schematic representation of an OCSP protocol run. The request *may* be signed, the response, when containing actual data, *must* be signed. (cf. (Santesson et al., 2013))

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 OSCP responder, which now has to handle each individual request. A possible way to alleviate this problem is to employ *OCSP stapling*, as defined in (Eastlake, 2011): 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.

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 (Durumeric et al., 2013a), containing a total of 66,335,624 HTTPS certificates.

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.

Second, we performed a preliminary analysis of how different frameworks under different operating systems react to an unreachable OCSP responder. In particular, we tested the following software components:

- Internet Explorer 8.0.6001.18702 for Windows XP
- Internet Explorer 11.0.9600.17041 for Windows 7
- Firefox 28.0 for Windows 7 and Windows XP
- Firefox 26.0 for Ubuntu 13.04
- Safari 5.1.7 for Windows 7 and Windows XP
- Opera 20.0.1387.91 for Windows 7 and Windows XP
- Opera 12.16 for Ubuntu 13.04
- Chrome 34.0.1847.116 for Windows 7, Windows XP, and Ubuntu 13.04
- Outlook 14.0.7116.5000 for Windows 7
- Java 7u55 for Windows 7, Windows XP, and Ubuntu 13.04
- Adobe Acrobat Professional 8.0.0 for Windows 7
- Adobe Flash Player Installation 13.0.0.182 for Windows 7

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.

Table 1 gives an overview of our preliminary findings, where the first half describes the behaviour without the firewall, the second half with blocking of the OCSP URL. Each cell gives the results for the website with the valid and revoked certificate, respectively, separated by a ||. Possible reactions of the browsers where blocking of the website, thereby impeding access altogether (×), opening the websites without any warnings (✓) or presenting a dialogue-box for the user to choose the preferred action (□). The results vary wildly depending on browser and operating system, with Chrome effectively ignoring OCSP altogether (as is also detailed in (Langley, 2014) 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.

Table 1: Comparison of the reactions of the browsers tested when OSCP was reachable or blocked, respectively. ✓ indicates that the page was displayed, × that the page was blocked, and □ that the user was prompted on how to proceed.

OCSP reachable	valid certificate ‖ revoked certificate		
	Windows 7	Windows XP	Ubuntu 13.04
Internet Explorer 8.0.6001.18702		✓ ‖ ×	
Internet Explorer 11.0.9600.17041	✓ ‖ ×		
Firefox 28.0	✓ ‖ ×	✓ ‖ ×	
Firefox 26.0 for Ubuntu			✓ ‖ ×
Safari 5.1.7	✓ ‖ □	✓ ‖ □	
Opera 20.0.1387.91	✓ ‖ ×	✓ ‖ ×	
Opera 12.16 for Ubuntu			✓ ‖ ×
Chrome 34.0.1847.116	✓ ‖ ✓	✓ ‖ ×	✓ ‖ ✓
OCSP blocked	**valid certificate ‖ revoked certificate**		
	Windows 7	Windows XP	Ubuntu 13.04
Internet Explorer 8.0.6001.18702		✓ ‖ ×	
Internet Explorer 11.0.9600.17041	✓ ‖ ×		
Firefox 28.0	✓ ‖ ✓	✓ ‖ ✓	
Firefox 26.0 for Ubuntu		✓ ‖ ✓	
Safari 5.1.7	✓ ‖ □	✓ ‖ □	
Opera 20.0.1387.91	✓ ‖ ×	✓ ‖ ×	
Opera 12.16 for Ubuntu			✓ ‖ ✓
Chrome 34.0.1847.116	✓ ‖ ✓	✓ ‖ ×	✓ ‖ ✓

- The validity of the signature on a PDF document is considered 'unknown' when OCSP and CRL access is blocked.
- An e-mail signature is shown as 'valid' in Outlook 2010 when OCSP and CRL are 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 both OCSP and CRL are 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, asking the user, to downright blocking access to the specific web-page.

In addition to that, by using the existing data-set from (Durumeric et al., 2013a) 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 preliminary 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.

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

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. (2008). RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Technical report.

Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. (2013a). Analysis of the HTTPS certificate ecosys-

tem. In *Proceedings of the 13th Internet Measurement Conference*.

Durumeric, Z., Wustrow, E., and Halderman, J. A. (2013b). ZMap: Fast Internet-wide scanning and its security applications. In *Proceedings of the 22nd USENIX Security Symposium*.

Eastlake, D. (2011). RFC6066 - Transport Layer Security (TLS) Extensions: Extension Definitions. Technical report.

Housley, R., Ford, W., Polk, W., and Solo, D. (1999). RFC2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile. Technical report.

Langley, A. (2014). No, don't enable revocation checking. `https://www.imperialviolet.org/2014/04/19/revchecking.html`.

Marlinspike, M. (2009). Defeating OCSP with the Character '3'. `http://www.thoughtcrime.org/papers/ocsp-attack.pdf`.

Meyer, C. and Schwenk, J. (2013). SoK: Lessons Learned from SSL/TLS Attacks. In Kim, Y., Lee, H., and Perrig, A., editors, *Information Security Applications - 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers*, volume 8267 of *Lecture Notes in Computer Science*, pages 189–209. Springer.

Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and Adams, C. (2013). RFC6960 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. Technical report.

Yee, P. (2013). RFC6818 - Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Technical report.