Softwaresicherheit

x.509 Certificate Validation: challenges, pitfalls, bugs

Gruppenmitglieder: Julian Hinxlage (7010922)
Edgar Schkrob (7012009)
Johannes Theiner (7010923)

Semester: Wintersemester 2019/2020
letzte Änderung: 20. Januar 2022

Inhaltsverzeichnis

1 Grundlagen **1**
1.1 Allgemein . . . 1
1.2 Struktur . . . 1
1.3 Zertifikatskette . . . 3

2 Validierung von Extensions **4**
2.1 Key Usage . . . 4
2.1.1 KCI Attacke . . . 4
2.2 Critical Extensions . . . 5
2.3 Hostname Validation . . . 5
2.4 Basic Constraints . . . 5
2.5 Name Constraint . . . 6

3 Weitere Probleme **7**
3.1 Validity Dates . . . 7
3.2 Certificate Chain Validation . . . 7
3.3 Certificate Revocation . . . 7
3.4 Redirection . . . 7
3.4.1 StartEncrypt . . . 8
3.5 Weak Algorithm . . . 8
3.6 Verwendung von Bibliotheken . . . 8

4 nicht technische Probleme **9**
4.1 Verbindungssicherheit . . . 9
4.2 Vertrauen in die Zertifikatskette . . . 9
4.3 IDN homograph attack . . . 9
4.4 Extended Validation . . . 10

5 Fazit **11**

1 Grundlagen

1.1 Allgemein

X.509 ist ein Standard der ITU-T[1]itut, der zur Erstellung digitaler Zertifikate in Public-Key-Infrastrukturen verwendet wird. Dieser wird auch als ISO/IEC 9594-8 bezeichnet. Das X.509-Zertifikat wird in Protokollen wie TLS, SSL, und HTTPS angewendet. Außerdem wird es im S/MIME-Standard benutzt, der im Bereich der E-Mail-Übertragung eingesetzt wird.

Ein X.509-Zertifikat wird von Zertifikatsstellen ausgestellt, diese sind nach einem hierarchischen System angeordnet. Für gewöhnlich verfügen Webbrowser über eine Liste mit Zertifikatsstellen, die als vertrauenswürdig gelten. Zertifikatstellen können Zertifikat auch für ungültig erklären. Dafür wird das betroffene Zertifikat in eine Sperrliste verzeichnet, die von den meisten Programmen überprüft wird.

Die aktuellste Version von X.509 ist X.509v3. Das X.509v3-Zertifikat ist durch Profile erweiterbar.

1.2 Struktur

Das X.509-Zertifikat weißt eine festgelgete Struktur auf. Dies sieht wie folgt aus:

```
Zertifikat
    Version
    Seriennummer
    Algorithmen-ID
    Aussteller
    Gültigkeit
        von
        bis
    Zertifikatinhaber
    Zertifikatinhaber-Schlüsselinformationen
        Public-Key-Algorithmus
        Public Key des Zertifikatinhabers
    Eindeutige ID des Ausstellers
    Eindeutige ID des Inhabers
    Erweiterungen
```

[1]International Communication Union Telecommunication Standardization Sector

```
        ...
Zertifikat-Signaturalgorithmus
Zertifikat-Signatur
```

Beispiel:

```
Certificate:
    Data:
        Version: 3 \(0x2\)
        Serial Number: 1 \(0x1\)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=AT, ST=Steiermark, L=Graz, O=TrustMe Ltd, OU=Certificate Authority, CN=C
        Validity
            Not Before: Oct 29 17:39:10 2000 GMT
            Not After : Oct 29 17:39:10 2001 GMT
        Subject: C=AT, ST=Vienna, L=Vienna, O=Home, OU=Web Lab, CN=anywhere.com/Email=xyz@
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: \(1024 bit\)
                Modulus \(1024 bit\)\:
                    00:c4:40:4c:6e:14:1b:61:36:84:24:b2:61:c0:b5:
                    d7:e4:7a:a5:4b:94:ef:d9:5e:43:7f:c1:64:80:fd:
                    9f:50:41:6b:70:73:80:48:90:f3:58:bf:f0:4c:b9:
                    90:32:81:59:18:16:3f:19:f4:5f:11:68:36:85:f6:
                    1c:a9:af:fa:a9:a8:7b:44:85:79:b5:f1:20:d3:25:
                    7d:1c:de:68:15:0c:b6:bc:59:46:0a:d8:99:4e:07:
                    50:0a:5d:83:61:d4:db:c9:7d:c3:2e:eb:0a:8f:62:
                    8f:7e:00:e1:37:67:3f:36:d5:04:38:44:44:77:e9:
                    f0:b4:95:f5:f9:34:9f:f8:43
                Exponent: 65537 \(0x10001\)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                email:xyz@anywhere.com
            Netscape Comment:
                mod_ssl generated test server certificate
            Netscape Cert Type:
                SSL Server
    Signature Algorithm: md5WithRSAEncryption
        12:ed:f7:b3:5e:a0:93:3f:a0:1d:60:cb:47:19:7d:15:59:9b:
        3b:2c:a8:a3:6a:03:43:d0:85:d3:86:86:2f:e3:aa:79:39:e7:
        82:20:ed:f4:11:85:a3:41:5e:5c:8d:36:a2:71:b6:6a:08:f9:
```

```
cc:1e:da:c4:78:05:75:8f:9b:10:f0:15:f0:9e:67:a0:4e:a1:
4d:3f:16:4c:9b:19:56:6a:f2:af:89:54:52:4a:06:34:42:0d:
d5:40:25:6b:b0:c0:a2:03:18:cd:d1:07:20:b6:e5:c5:1e:21:
44:e7:c5:09:d2:d5:94:9d:6c:13:07:2f:3b:7c:4c:64:90:bf:
ff:8e
```

1.3 Zertifikatskette

Eine Zertifikatskette ist eine Liste aus Zertifikaten, die zur Feststellung der Stammzertifizierungsstelle verwendet wird. Beim Empfang eines Zertifikates wird die Echtheit durch das Aufrufen des Ausstellerzertifikates überprüft. Diese wird durch das Zertifikat dessen Ausstellers überprüft. Dieser Vorgang wird fortgesetzt bis das Zertifikat der Stammzertifizierungsstelle aufgerufen wird, welches grundsätzlich als vertrauenswürdig gilt und nicht weiter überprüft wird.

2 Validierung von Extensions

2.1 Key Usage

Die Key Usage Erweiterung eines Zertifikat bestimmt den Zweck des Public Key. Die Nutzung des Schlüssels wird dadurch limitiert. Diese Erweiterung sollte immer als kritisch markiert werden, da die Key Usage immer überprüft werden muss.

Es gibt folgende Werte:

- digitalSignature
- nonRepudiation
- keyEncipherment
- dataEncipherment
- keyAgreement
- keyCertSign
- cRLSign
- encipherOnly
- decipherOnly

Dabei kann ein Zertifikat auch mehrere Werte für die Key Usage annehmen.

Das digitalSignature Bit ist gesetzt, wenn der Key zum Prüfen von Signaturen, die nicht ein Zertifikat sind, eingesetzt wird. Das keyCertSign Bit ist gesetzt, wenn eine Signatur eines weiteren Zertifikat geprüft werden soll. [1]

Die Problematik entsteht, wenn die Key Usage nicht richtig während der Zertifikat Validierung überprüft wird. Dies wurde bei der KCI Attacke ausgenutzt.

2.1.1 KCI Attacke

Die Key Compromise Impersonation(KCI) Attacke ist eine Man in the Middle Attacke. Dabei fügt der Angreifer ein Client Zertifikat dem Opfer hinzu (z.B. bei Hotel Wifi soll häufig ein Client Zertifikat installiert werden). Dann kann der Angreifer sich als der Server ausgeben. Dazu braucht er nur den private key des hinzugefügten Zertifikates. Eigentlich sollte es nicht möglich sein mit einem Client Zertifikat einen Server zu authentifizieren. Da aber in der Implementation vieler Zertifikat Valierungen die key usage nicht korrekt überprüft wurde, kann eine Zertifikatskette erzeugt werden und damit der Server authentifiziert werden. Der Angreifer muss zusätzlich den Traffic des Opfers zu sich umleiten, dies ist z.B. mit arp spoofing möglich.[2]

2.2 Critical Extensions

Die Erweiterbarkeit des x.509 Protokoll wird über Erweiterungen gewährleistet. Wenn unterschiedliche Software, die unterschiedliche Versionen des Protokolls benutzt kann es dazu kommen, dass eine Erweiterung unbekannt ist. In diesem Fall wird die Erweiterung ignoriert. Ist die Erweiterung allerdings als "critical" markiert muss das Zertifikat mit einer unbekannten Erweiterung als ungültig behandelt werden.

Deswegen sollte bei der Erstellung von Zertifikaten alle wichtigen Erweiterungen als "critical" markiert werden. Bei der Validierung müssen Zertifikate mit unbekannten Erweiterungen abgelehnt werden.[3]

2.3 Hostname Validation

Zertifikate enthalten den Hostname, für den das Zertifikat ausgestellt ist. Es ist wichtig, dass dieser korrekt überprüft wird. Über die Verifizierung das Hostnames wird sichergestellt, dass die Verbindung auch mit dem vorgesehenem Server aufgebaut wurde. Ohne eine Überprüfung kann jedes gültige Zertifikat für jeden Hostname genutzt werden.

Der Hostname kann an mehreren Stellen im Zertifikat auftreten, im "subject" Feld und in der "Subject Alternative Name" Erweiterung. Die Hostname Validierung kann für jede Anwendung der x.509 Zertifikate spezifisch sein. Die Validierung ist also nicht einheitlich und ist nicht im x.509 Standard definiert. Bei dem TLS[2] Protokoll, welches von HTTPS Webseiten verwendet wird, entspricht der Hostname der Domain der Webseite. Bei Email Zertifikaten ist der Hostname die Email Adresse.[3]

2.4 Basic Constraints

Die "Basic Constraints" Erweiterung gibt an, ob das Zertifikat ein CA[3] Zertifikat ist oder nicht. Nur, wenn ein Zertifikat ein CA Zertifikat ist kann es genutzt werden, um andere Zertifikate zu erstellen. In diesem Fall wird auch angegeben, wie viele Zertifikate noch in der Zertifikatskette folgen können.

Bei der Validierung muss also jedes Zertifikat in der Kette überprüft werden. Der Wert für CA muss true sein und die Anzahl der folgenden Zertifikate darf nicht größer sein als der entsprechende Wert. Eine Zertifikatskette, bei der das nicht zutrifft, muss als ungültig gehandhabt werden.[3]

[2]Transport Layer Security
[3]Certificate Authority

2.5 Name Constraint

Die ”Name Constraint” Erweiterung gibt Limitierungen für CA Zertifikaten an. Über die Limitierungen wird bestimmt, welche Art von Zertifikaten in der Kette folgen können.

Für jedes Zertifikat in der Kette muss überprüft werden, ob der Typ nicht im darüberliegenden Zertifikat limitiert wird.[3]

3 Weitere Probleme

3.1 Validity Dates

Ein Zertifikat ist nur für einen bestimmten Zeitraum gültig. Der Zeitraum wird über die Felder ”notBefore” und ”notAfter” definiert. Bei der Prüfung eines Zertifikats müssen beide Zeitstempel mit der aktuellen Zeit abgeglichen werden. Ein Zertifikat muss abgelehnt werden, wenn die aktuelle Zeit kleiner als ”notBefore” oder größer als ”notAfter” ist. Wenn dies nicht überprüft wird, kann ein abgelaufenes Zertifikat weiter genutzt werden.[3]

3.2 Certificate Chain Validation

Alle Schritte bei der Überprüfung eines Zertifikates müssen für alle Zertifikate in der Kette durchgeführt werden. Auch die digitalen Signaturen müssen überprüft werden. Jedes Zertifikat, außer das Root-Zertifikat, muss mit dem public key des darüberliegendem Zertifikats signiert werden. Root Zertifikate sind selbst signiert. Bei einer ungültigen Signatur muss das Zertifikat abgelehnt werden.[3]

3.3 Certificate Revocation

Wenn ein Zertifikat von der CA als ungültig erklärt wird, wird ein Eintrag in die Widerrufliste hinzugefügt. Bei der Überprüfung eines Zertifikats muss diese Liste überprüft werden. Wenn es ein Eintrag für das aktuelle Zertifikat gibt, muss dieses abgelehnt werden. Dies ist ein zusätzlicher Schritt, der weitere Anforderungen an die Infrastruktur stellt.

3.4 Redirection

Bei jedem Besuch einer Webseite muss das Zertifikat überprüft werden. Dies trifft auch bei einer Weiterleitung zu. Nachdem eine Webseite den Nutzer weiterleitet, muss auch das Zertifikat des folge Webseite überprüft werden.

3.4.1 StartEncrypt

Bei der CA StartEncrypt konnten gültige Zertifikate ausgestellt werden, ohne Kontrolle über die Domain zu haben. So war es möglich z.b ein gültiges Zertifikat für Facebook

von StartEncrypt zu erhalten.

Zum beweisen der Kontrolle der Domain muss eine Datei auf den Server geladen werden. Der Server von StartEncrypt prüft dann, ob die Datei verfügbar ist. Auf einigen Webseiten ist es möglich als Benutzer Dateien hochzuladen. Dadurch war es möglich die Datei, die von StartEncrypt verlangt war, auf der Webseite hochzuladen. Die Prüfung von StartEncrypt ist also erfolgreich und das Zertifikat wird ausgegeben.

Auch ist der Server Weiterleitungen gefolgt. So Konnte die Datei auch auf einer anderen Domain liegen, wenn von der ursprünglichen Domain dorthin weitergeleitet wurde.[4]

3.5 Weak Algorithm

Zertifikate können verschiedene Algorithmen für das Signatur verfahren verwenden. Dabei ist das eigentliche Signaturverfahren, sowie die Hashfunktion wichtig. Viele Signaturverfahren unterstützen verschiedene Schlüssellängen. Daher ist auch die Wahl der Schlüssellänge wichtig.

Es wurde längere Zeit lang die md5 Hashfunktion verwendet. Md5 hat nur eine Bitlänge von 56 und gilt als gebrochen. Zertifikate, die Md5 verwenden sind daher nicht sicher. Ein Zertifikat mit einer gültigen Unterschrift könnte daher ohne den private key erstellt werden.

Die Wahl der verschiedenen Verfahren ist wichtig und für die Sicherheit von Zertifikaten essenziell.

3.6 Verwendung von Bibliotheken

Auch das Verwenden einer Bibilothek schützt nicht unbedingt vor diesen Problemen. Viele Bibilotheken verwenden schlechte Standardwerte oder sind so kompliziert aufgebaut das nur wenige Programmierer diese verstehen und korrekt anwenden können.[5]

4 nicht technische Probleme

4.1 Verbindungssicherheit

War eine verschlüsselte Verbindung über SSL[4] mit einer Webseite aufgebaut zeigten viele Webbrowser in der URL[5] Zeile ein grünes Schloss mit dem Text "sicher". Viele Nutzer assoziieren "sicher" fälschlicherweise mit "vertrauenswürdig" und vertrauen sofort solchen Webseiten. [6]

4.2 Vertrauen in die Zertifikatskette

Nicht nur dem Gesprächspartner muss vertraut werden auch der ausstellenden CA muss vertraut werden. So versuchte Kazakhstan seit 2015 die eigenen Bürger über ein Root Zertifikat auszuspionieren. Zuerst wurde versucht dieses Zertifikat über entsprechende Anträge in die verschiedenen Browser einzufügen, als dies scheiterte wurden die Büger von den ISP[6]s gezwungen dieses selbst zu installieren, um das Internet verwenden zu dürfen. Apple, Google und Mozilla blockieren inzwischen dieses Zertifikat.[7]

4.3 IDN homograph attack

Die eindeutige Identifizierung von Texten bei denen mehrere Alphabete verwendet werden können gestaltet sich schwierig wenn ähnliche Zeichen enthalten sind. So wird seit den ersten Versionen von DNS[7] ASCII[8] zur Kodierung von Domainnamen verwendet. Um auch nicht ASCII Zeichen rückwärtskompatibel verwenden zu können wurde 2003 Punnycode entwickelt. Dabei werden Zeichen die nicht im ASCII Zeichensatz enthalten sind in Codepunkte umgewandelt.

Die starke optische Ähnlichkeit zwischen einigen Zeichen verschiedener Alphabete macht sich ein IDN[9] homograph Angriff zu Nutze. Dabei werden einige Zeichen ausgetauscht um authentisch Zeichenketten(etwa URLs) fälschen zu können.

Aktuelle Browser & E-Mail Clients schützen vor diesem Angriff durch anzeigen von IDN Domains in Punnycode Form.[8]

[4]Secure Sockets Layer
[5]Uniform Resource Locator
[6]Internet Service Provider
[7]Domain Name System
[8]American Standard Code for Information Interchange
[9]Internationalized domain name

4.4 Extended Validation

Eine Erweiterung von x.509 sind EV[10] Zertifikate, diese enthalten zusätzliche Informationen über den Besitzer des Zertifikats. Nur ein registriertes Unternehmen kann ein solches Zertifikat, bei einer der wenigen CAs die ein solches austellen dürfen, beantragen.

In älteren Browsern wurde der Name des Unternehmens neben oder anstatt der URL angezeigt. Exsistieren nun zwei Unternehmen mit gleichem Namen sind die Zertifikate bis auf das Bundesland/den Bundesstaat für normale Nutzer identisch.[9]

[10] Extended Validation

5 Fazit

X.509 Zertifikate haben mehrere Einsatzgebiete. Durch die schrittweise eingeführten Erweiterungen ist der Standard mittlerweile unübersichtlich. Das hat zur Folge, dass bei der Validierung viele Einzelheiten beachtet werden müssen. Dadurch gibt es viele potenzielle Fehlerquellen bei der Implementierung. Daher ist es stark zu empfehlen eine einfache und sichere Library zu verwenden und die Validierung nicht selbst zu implementieren. Weiterhin basiert der Standard stark auf das Vertrauen in die CAs. Durch die CAs muss auch die Infrastruktur bereitgestellt werden.

Vorteilhaft ist die weite Verbreitung der Zertifikate. Der sichere Schlüssel Austausch, der public keys, wird über Zertifikate gewährleistet.

SSH[11] bietet eine alternative Lösung zu Zertifikaten. Bei dem SSH Protokoll werden die public keys bei der ersten Kommunikation als known host gespeichert. Bei jedem weiteren Verbindungsaufbau wird überprüft ob der richtige public key abgespeichert ist. Wenn sich der public key des Servers ändern sollte oder ein Angreifer versucht einen anderen Schlüssel einzuschleusen, wird dies von SSH Protokoll erkannt. Es ist also wichtig, dass beim ersten Verbindungsaufbau mit dem richtigen Server kommuniziert wird.

[11] Secure Shell

Literatur

[1] R. Housley, R. Laboratories, W. Polk u. a., „Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280, 25. Okt. 2020.

[2] C. Hlauschek, M. Gruber, F. Fankhauser und C. Schanes, „Prying open Pandora's box: KCI attacks against TLS," Techn. Ber., 2015.

[3] D. Cooper, NIST, S. Santesson u. a., „Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, 2008.

[4] R. Eikenberg. „Gratis-CA StartEncrypt beginnt mit Sicherheitsproblemen." (2. Juli 2016), Adresse: `https://www.heise.de/security/meldung/Gratis-CA-StartEncrypt-beginnt-mit-Sicherheitsproblemen-3252904.html` (besucht am 13. 11. 2019).

[5] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh und V. Shmatikok, „The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software," Techn. Ber., 2012.

[6] T. Hunt. „The Decreasing Usefulness of Positive Visual Security Indicators (and the Importance of Negative Ones)." (7. Mai 2018), Adresse: `https://www.troyhunt.com/the-decreasing-usefulness-of-positive-visual-security-indicators-and-the-importance-of-negative-ones/` (besucht am 02. 11. 2019).

[7] C. Cimpanu. „Apple, Google, and Mozilla block Kazakhstan's HTTPS intercepting certificate." (21. Aug. 2019), Adresse: `https://www.zdnet.com/article/apple-google-and-mozilla-block-kazakhstans-https-intercepting-certificate/` (besucht am 12. 11. 2019).

[8] X. Zheng. „Phishing with Unicode Domains." (14. Apr. 2017), Adresse: `https://www.xudongz.com/blog/2017/idn-phishing/` (besucht am 26. 10. 2019).

[9] I. Carroll. „Extended Validation Is Broken." (29. Apr. 2018), Adresse: `https://stripe.ian.sh/` (besucht am 06. 11. 2019).