So unterscheiden sich Clientzertifikate von Serverzertifikaten

In Zeiten von DevOps hat man als Entwickler immer mehr mit Zertifikaten zu tun. In meiner Laufbahn als Consultant habe ich bemerkt, dass es hierbei immer wieder zu den grössten Verwirrungen kommt. Ich möchte daher mein heutiges TechUp diesem Thema widmen.

Zuerst ein paar Grundbegriffe

SSL (Secure Socket Layer)

SSL ist ein von Netscape entwickeltes Verschlüsselungsprotokoll, welches 1994 in der Version 1.0 erstmalig erschien. 1995 wurde SSL 2.0, und nur ein Jahr später SSL 3.0 veröffentlich. Alle diese Versionen sind aber heute nicht mehr zulässig. 1996 hat Netscape die Versionskontrolle zur Entwicklung eines Internet-Standards an die IETF (Internet Engineering Task Force), übergeben.

TLS (Transport Layer Socket)

Nach der Übergabe entwickelte die IETF auf Basis von SSL 3.0 das verbessertes Protokoll TLS in der Version 1.0, welches 1999 erschien. Auch dieses wird mittlerweile aber nicht mehr unterstützt, weil es unter anderem nicht mehr dem Zahlungsverkehr-Standard (PCI DSS) entspricht. 2006 wurde die Version 1.1 von TLS herausgebracht. Da hier aber SHA-1 für die Signaturerstelllung verwendet wird, wird von der Nutzung abgeraten. 2008 wurde dann die noch heute gültige TLS 1.2 veröffentlicht. Seit 2018 gibt es aber mittlerweile auch schon TLS 1.3 welche neue Anforderungen für TLS 1.2 enthält.

Cipher Suites

Eine Cipher (Chiffre) ist einfach ein Algorithmus, oder eine Sammlung von Schritten, um mathematische Berechnungen durchzuführen (RSA). Anhand dieses Algorithmus werden die Nachrichten verschlüsselt. Es gibt für TLS 1.2 derzeit 37 Ciphers und für TLS 1.3 fünf Ciphers. Als Cipher Suite bezeichnet man eine Kombination von Chiffren. Es gibt vier verschiedene Arten von Chiffren:

- Key Exchange Algorithms (RSA, DH, ECDH, DHE, ECDHE, PSK)
- Authentication/Digital Signature Algorithm (RSA, ECDSA, DSA)
- Bulk Encryption Algorithms (AES, CHACHA20, Camellia, ARIA)
- Message Authentication Code Algorithms (SHA-256, POLY1305)

Eine Cipher Suite sieht nun beispielsweise so aus:

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

- TLS ist das Protokoll.
- ECDHE (Elliptic Curve Diffie Hellman) ist eine Chiffre, mit welcher während des Handshake die

Keys ausgetauscht werden.
- RSA ist der Authentifizierungsmechanismus.
- AES_128_GCM Ist der Bulk-Verschlüsselungsmechanismus (also die Verschlüsselung der Daten).
- SHA256 ist der Hashing Algorithmus.

X.509

X.509 ist ein Standard für eine Public-Key-Infrastruktur zur Erstellung digitaler Zertifikate. X.509 Zertifikate finden Anwendung bei TLS-Versionen verschiedener Übertragungsprotokolle (HTTPS, S/MIME).

Für was braucht es Zertifakte?

Zertifikate sind einerseits dazu da, sich um die Identifizierung von Benutzern oder Clients zu kümmern, und zusätzlich sind sie dafür verantwortlich, diese Kommunikation zu verschlüsseln. Man unterscheidet dabei zwischen zwei Arten von Zertifikaten; dem Client-Zertifikat (auch PKI) und dem Serverzertifikat (auch SSL-Zertifikat).

Verschlüsselung

Es gibt mehrere Wege, wie man Informationen verschlüsseln kann. Das einfachste ist, die Nachricht mit einem Schlüssel zu verschlüsseln und diese an den Empfänger zu schicken. Da dem Empfänger der gleiche Schlüssel vorliegt, kann er diese Nachricht wieder entschlüsseln.

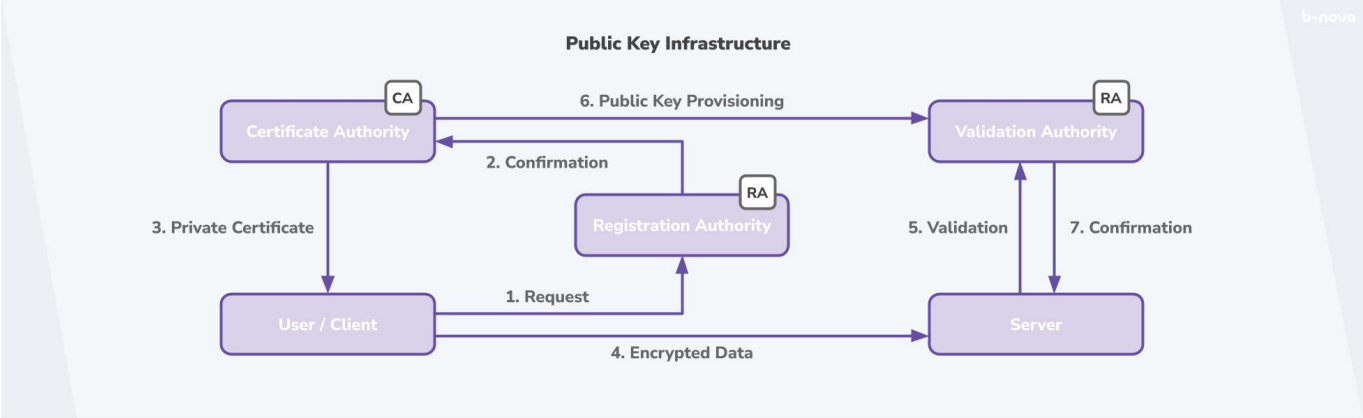
Im Netz ist diese Methode aber nicht wirklich anwendbar, denn sonst müsste man von jeder Webseite, mit der man sicher kommunizieren möchte, den Schlüssel kennen. Dies ist natürlich nicht möglich, denn man könnte den Schlüssel nur auf unverschlüsseltem Weg erhalten. Aus diesem Grund gibt es das Public-Key-Verfahren. Damit dieses Verfahren funktionieren kann, braucht es aber erstmals eine Infrastruktur.

Public Key Infrastructure

Wir alle haben im Zuge von Zertifikaten schon von PKI, oder Public Key Infrastructure gehört. Als PKI bezeichnet man in der Kryptologie ein System, welches digitale Zertifikate ausstellen, verteilen und prüfen kann. Die wichtigsten Bestandteile einer PKI sind:

- Zertifizierungsstelle (Certificate Authority, CA)
- Registrierungsstelle (Registration Authority, RA)
- Validierungsdienst (Valdiation Authority, VA)
- X.509 Digitale Zertifikate
- Verzeichnisdienste für Zertifikate
- Zertifikatssperrlisten

Die Funktionsweise einer PKI ist nun folgende:



Chain of Trust (Vertrauenskette)

Schauen wir uns an, wie in der Praxis mit einer PKI gearbeitet wird. Das höchste Gut einer PKI ist die Root-CA (Root Certificate Authority). Diese dient als “Vertrauensanker” und muss besonders geschützt werden. Wird die Root-CA gehackt, so ist die gesamte PKI nicht mehr vertrauenswürdig und alle von ihr ausgestellten Zertifikate müssen widerrufen werden. Aus diesem Grund haben auch nur wenige Personen Zugriff auf die Root-CA, welche übrigens NUR offline gehalten werden darf.

Aus diesem Grund gibt es CA-Hierarchien. Schauen wir uns an, welche CA’s es in dieser Hierarchie gibt.

Root-Zertifikate

Das Herzstück einer PKI. Jede CA hat nur wenige Root-Zertifikate. Der Public Key eines Root-Zertifikats wird beispielsweise auch im Browser hinterlegt, damit allen von der Root-CA ausgestellten Zertifikaten vertraut wird (Chrome CA’s).

Intermediate-Zertifikate

Ein Intermediate-Zertifkat (dt. Zwischenzertifikat) wird von einem Root-Zertifikat ausgestellt. Nachdem das Intermediate-Zertifkat durch das Root-Zertifikat signiert wurde, wird das Root Zertifikat zum Ausstellen weiterer Zertifikate nicht mehr gebraucht.

Serverzertifikate

Dieses Zertifikat wird von einem Intermediate-Zertifkat signiert. Es wird spezifisch für einen Benutzer oder Domainnamen ausgestellt.

Serverzertifikate werden zur Authentifizierung eines Servers genutzt. Ein SSL/TLS-Zertifikat (oder Serverzertifikat, wir nennen es einfachheitshalber im folgenden nur SSL-Zertifikat) ist auch die Basis für eine verschlüsselte Kommunikation mit einer Webseite. So werden beispielsweise Phishing-Attacken (Vortäuschen einer falschen Identität) verhindert.

Der Name SSL-Zertifikat ist in diesem Kontext irreführend. Anhand des Namens könnte man denken, dass die Zertifikate an ein bestimmtes Protokoll gebunden sind, was aber nicht der Fall ist. Vielmehr wird ein Zertifikat nur für die Umsetzung von SSL bzw. TSL verwendet.

Aufbau eines SSL-Zertifikats

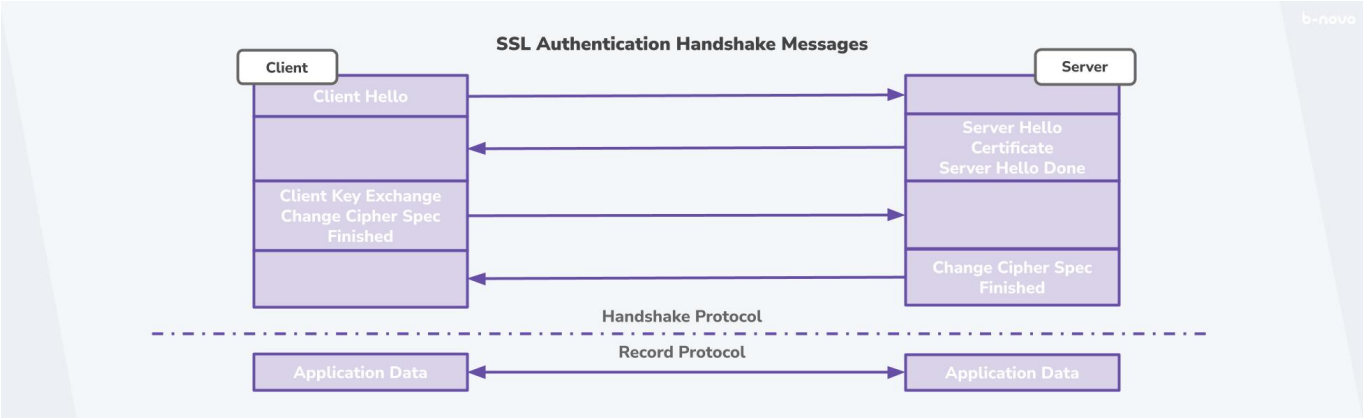
Ein SSL-Zertifikat enthält unter anderem die folgenden Informationen:

- Eindeutige Seriennummer
- Inhaber des Zertifikats (Unternehmen)
- Zertifizierte Domain
- Gültigkeit
- Verwendete Algorithmen
- Aussteller (CA = Certificate Authority)
- Signatur der CA
- Fingerabdruck
- Öffentlicher Schlüssel

Funktion SSL

Um eine sichere Verbindung aufzubauen, wird ein SSL/TLS-Handshake durchgeführt. Wir schauen uns das am Beispiel einer HTTPS-Verbindung eines Browsers mit einem Server an.

Nehmen wir an, du rufst die Seite b-nova.com auf.



1. Browser und Server sagen “hello” und einigen sich auf die Cipher Suite, welche für die Verschlüsselung genutzt werden soll.
2. Der Server sendet sein SSL-Zertifikat (Public Key inkludiert) zum Browser.
3. Der Browser überprüft das SSL-Zertifikat anhand des im Browser hinterlegten Root Zertifikats. Ist diese Überprüfung in Ordnung, erstellt der Browser einen Session Key und verschlüsselt diesen mit dem Public Key des Servers und schickt diesen anschliessend wieder zum Server zurück.
4. Der Server entschlüsselt den Session Key mit seinem Private Key. Ist diese Entschlüsselung erfolgreich, so ist der Handshake abgeschlossen. Bis zu diesem Zeitpunkt fand die asymmetrische Verschlüsselung statt, da Server und Browser ihre Nachrichten mit

verschiedenen Schlüsseln ver- und entschüsseln.
5. Ab diesem Zeitpunkt kommunizieren Browser und Server nur noch über den Session Key, welcher symmetrisch verschlüsselt wird.

Clientzertifikate

Ein Clientzertifikat dient zur Authentifizierung eines Benutzers oder “Clients”. Es dient allerdings nicht der Verschlüsselung der Kommunikation. Aus diesem Grund macht ein Clientzertifikat nur Sinn beim gleichzeitigen Einsatz eines Serverzertifikats, da dieses die Daten verschlüsselt. Clientzertifikate basieren, wie auch Serverzertifikate auf PKI.

Wenn man ein Clientzertifikat einsetzen will, so fordert der Server das Zertifikat als zusätzlichen Schritt beim Client an. Der Client schickt das Zertifikat mit und prüft dies gegen das bei ihm hinterlegte Ausstellerzertifikat.

Wir sehen in der Grafik, dass der Server ein “Certificate Request” macht. Damit fordert er vom Client das Clientzertifikat an. Beim Server selbst ist dann das Gegenstück dazu hinterlegt, damit er den Client authentifizieren kann.

Fazit

Hoffentlich konnte ich dir nun etwas näherbringen, wie Zertifikate im Allgemeinen funktionieren, und was Server- und Clientzertifikate ausmacht.