

Letzte Vorlesung

Themen der letzten Vorlesung:

- Sicherheit in Web-Anwendungen
- Sichere Übergabe von Werten per HTTP
- Cross-Site-Scripting
- SQL Injection

Weitere Sicherheitsaspekte

Weitere Sicherheitsrisiken:

- Belauschen einer Verbindung
- Verfälschen der übertragenen Daten
- "Identitätsklau"

Verwende sicheres Protokoll mit folgenden Zielen:

- *Server-Authentifizierung* Client soll sicher sein, dass der Server auch wirklich derjenige ist, der er vorgibt zu sein
- *Client-Authentifizierung* Entsprechendes für den Client
- *Verschlüsselung* Niemand außer Client und Server soll es möglich sein, die Daten zu lesen

SSL und TLS

Für Web-Anwendungen wird das *Secure Socket Layer (SSL)* verwendet. Eigenschaften:

- Wird von (fast) allen Web-Servern und Browsern unterstützt
- Sämtlicher Aufwand wird an den Web-Server delegiert
- Zur Benutzung keine Änderungen an Web-Applikation erforderlich

Nachfolger von SSL: *Transport Layer Security (TLS)*

In der Praxis spricht man häufig nur von SSL, meint damit aber SSL (in den verschiedenen Versionen) *oder* TLS!

SSL und HTTP

HTTP kann über SSL betrieben werden. Als well-known Port wird Port 443 verwendet.

Um das Protokoll zu unterscheiden wird als URL-Schema

`https://...`

verwendet.

Anwendungen und SSL

SSL liegt in der Protokollhierarchie oberhalb von TCP.

Um über SSL zu kommunizieren verwenden Programme *SSL-Sockets* anstatt normaler Sockets.

SSL verwendet zwei Schichten:

- *record layer* Zerlegt, komprimiert und verschlüsselt Daten mit symmetrischem Verschlüsselungsverfahren. Versieht Nachrichten mit einer Art Seriennummer
- Darüber: Protokolle, um Verschlüsselungsparameter auszutauschen (SSL-Handshake), zu ändern, Warnungen und Fehler zu übertragen, Anwendungs-Daten übertragen.

Verschlüsselungsverfahren

SSL verwendet sowohl *symmetrische Verschlüsselungsverfahren* als auch *Public-Key-Verfahren*

Es folgt: Kurzer Exkurs zum Thema Verschlüsselung

Public-Key-Verfahren

Public-Key-Verfahren können zum Verschlüsseln und zum Signieren von Daten verwendet werden

Jeder Teilnehmer braucht Schlüsselpaar bestehend aus:

- *Öffentlicher Schlüssel* dient zum Verschlüsseln
- *Privater Schlüssel* dient zum Entschlüsseln

In der Praxis: Der private Schlüssel kann nicht aus dem öffentlichen Schlüssel berechnet werden

Beispiele: RSA, Diffie-Hellmann

Verschlüsselung mit Public-Key-Verfahren

A sendet verschlüsselte Nachricht an B:

- A verwendet den öffentlichen Schlüssel von B, um die Nachricht zu verschlüsseln
- B verwendet seinen privaten Schlüssel, um die Nachricht zu dekodieren

Verschlüsselung mit großen Primzahlen:

- Multiplikation von zwei Primzahlen ist einfach
- Zerlegung in Primfaktoren dagegen sehr aufwändig

Verschlüsselter Text ist ein Produkt, das zur Entschlüsselung in Primfaktoren zerlegt werden muss

Symmetrische Verschlüsselung

Bei *symmetrischer Verschlüsselung* kommt nur ein Schlüssel zum Einsatz, den beide Teilnehmer kennen müssen

Vorteil: sehr schnell (im Vergleich zu Public-Key-Verfahren)

Nachteil: Schlüssel muss ausgetauscht werden

Beispiele: DES, 3DES, IDEA, AES, ...

Hybrid-Verfahren

In der Praxis kommen häufig *hybride Verfahren* zum Einsatz (auch bei SSL). Das heißt:

- Erzeuge einen *Sitzungsschlüssel*
- Verschlüssele den Sitzungsschlüssel mit einem Public-Key-Verfahren
- Verschlüsselten Sitzungsschlüssel übertragen
- Verwende den Sitzungsschlüssel, um die Nutzdaten der Anwendung mit einem symmetrischen Verfahren zu verschlüsseln

⇒ *Schnellere Übertragung großer Datenmengen*

Zertifikate

SSL verwendet *Zertifikate*, die einen Server/Dienst eindeutig ausweisen

Eine *Certificate Authority (CA)* gibt die Zertifikate heraus und bestätigt die Echtheit

Ein Zertifikat besteht aus:

- Öffentlicher Schlüssel
- Name (des Servers oder Clients)
- Verfallsdatum
- Name der CA
- Digitale Unterschrift der CA

Certificate Authorities (1)

Der Benutzer entscheidet selbst, welche Certificate Authority sein Zertifikat *signiert*.

- Benutzer erstellt *Certificate Signing Request (CSR)*, das die grundlegenden Informationen des Zertifikats enthält
- Benutzer sendet CSR an die Certificate Authority
- CA überprüft Angaben im CSR anhand von Personalausweis, Handelsregisterauszügen, ...
- Certificate Authority signiert Zertifikat mit dem CA-Zertifikat. Das Zertifikat kann veröffentlicht und benutzt werden.

Clients (Browser) enthalten die CA-Zertifikate von bekannten und vertrauenswürdigen Certificate Authorities, die zur Überprüfung der digitalen Signatur verwendet werden

Digitale Signatur

Wie wird signiert?

- A verschlüsselt Daten mit dem eigenen geheimen Schlüssel
- A sendet die verschlüsselten Daten an B
- B entschlüsselt die Daten mit dem öffentlichen Schlüssel von A

Was garantiert die digitale Signatur?

- Authentizität: B weiß, dass A (und kein anderer) signiert hat
- Nicht fälschbar: Nur A hat Zugang zum privaten Schlüssel
- Nicht übertragbar: Die Signatur gilt nur für diese Daten
- Manipulationssicher: Bei Änderungen der Daten kann B nicht mehr entschlüsseln
- Nicht zu leugnen: B braucht keine Hilfe von A zur Überprüfung

SSL-Handshake (1)

Nach dem Aufbau der TCP-Verbindung beginnt der SSL-Handshake. Dieser Handshake besteht aus bis zu vier Schritten:

- Client sendet `client_hello`, Server antwortet mit `server_hello`. Inhalt der Nachrichten: Version des SSL-Protokoll, Session-Id, unterstützte Verschlüsselungsverfahren, Pre-Master-Secret (Zufallszahl)
- Optional: Der Server weist sich mit seinem Zertifikat beim Client aus.

Certificate Authorities (2)

Ein *selbst-signiertes Zertifikat* ist eine Alternative zu einem CA-signierten Zertifikat.

Vorteil: Kostenlos, sinnvoll wenn es nur auf die Verschlüsselung ankommt

Signaturgesetz (SigG):

- CA kann sich bei der Bundesnetzagentur akkreditieren lassen
- Akkreditierte CAs können *qualifizierte Zertifikate* ausgeben
- Qualifizierte Zertifikate sind der eigenhändigen Unterschrift gleichgestellt

SSL-Handshake (2)

Weitere Schritte im SSL-Handshake:

- Optional: Der Client authentifiziert sich gegenüber dem Server, d.h. sendet sein Zertifikat an den Server. Client überprüft das Zertifikat des Servers
- Abschluss: Aus dem Pre-Master-Secret wird das Master-Secret abgeleitet und daraus der Sitzungsschlüssel

Die höhere SSL-Protokollschicht kann nun Daten verschlüsselt übertragen