Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

JavaScript
Raphael Ritter

Ein Angebot der Berufsfachschule Uster und der Höheren Fachschule Uster

Agenda

- MLZ
  - Vorstellung
  - Bewertungsschema
- Security
  - SOP
  - CSP
  - Attacken
- Architektur

Bildungszentrum Uster
Höhere Berufsbildung
Uster
HBU

Rückblick

Unterlagen der HSR

Einige Inhalte sind aus den Folien / Übungen des CAS Frontend Engineering der Hochschule für Technik Rapperswil.

www.hsr.ch
http://hsr.ch/CAS-Front-End-Engineering.12432.0.html

MLZ

- Lest euch das Briefing bezüglich der MLZ kurz durch
  - Dies bildet lediglich einen fiktiven Rahmen zu den Kriterien für die MLZ in diesem Modul kommen wir anschliessend

Security

- JavaScript wird im Browser innerhalb einer geschützten Umgebung, einer sog. **Sandbox** ausgeführt
- Diese bewirkt, dass JavaScript im Allgemeinen nur Zugriff auf die Objekte des Browsers hat
- Nicht möglich ist
  - auf das lokale Dateisystem zugreifen (Dateien lesen oder schreiben)
  - Betriebssystem- oder Browsereinstellungen ändern
  - Software auf dem Client-Rechner installieren

- Achtung: Durch den Einsatz von AcitveX (im IE) können diese Beschränkung umgangen werden

Security

- Same Origin Policy (SOP)
  - Sicherheitskonzept, das es JavaScript nur dann erlaubt, auf Objekte einer anderen Webseite zuzugreifen, wenn diese derselben Herkunft (Origin) sind.
  - Als Herkunft wird dabei die Kombination aus Protokoll (zum Beispiel http oder https), Domain und Port definiert.
  - Nur wenn alle drei gleich sind, gilt die SOP als erfüllt und der Skript-Zugriff ist möglich
  - Kann mit CORS (Cross-Origin Resource Sharing) oder JSONP gezielt erlaubt werden

Security

Beispiele SOP

- Ein Skript unter http://example.com/pub/page.html versucht, auf ein Element in den folgenden Seiten zuzugreifen:

- http://beispiel.com/page.html
  - SOP ist nicht erfüllt, andere Domain

- http://example.org/page.html
  - SOP ist nicht erfüllt, andere Domain

Security

Beispiele SOP

- Ein Skript unter http://example.com/pub/page.html versucht, auf ein Element in den folgenden Seiten zuzugreifen:

- http://example.com:89/dir/another.html
  - SOP ist nicht erfüllt, warum ?

- https://example.com/dir/another.html
  - SOP ist nicht erfüllt, warum ?

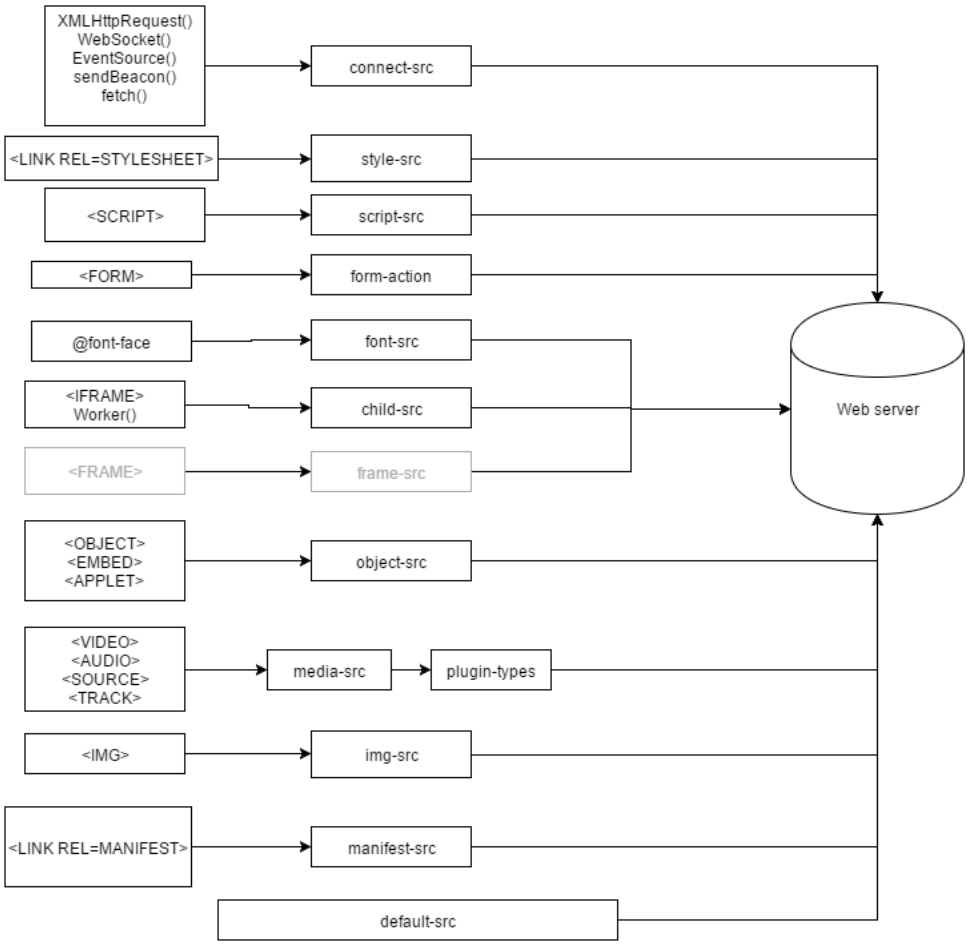
Security

CSP (Content Security Policy)

- Ein Standard für Web Security der primär zum Ziel hat XSS zu verhindern
- Eine CSP besteht aus verschiedenen Rules welche von den jeweiligen Browsern interpretiert und umgesetzt wird
- Es gibt dazu einen Working Draft von der W3C

https://content-security-policy.com/

Security



Mapping between HTML5 and JavaScript features and Content Security Policy controls
Based on CSP Level 3 draft, 2015, http://w3c.github.io/webappsec-csp/
Drawn by Pawel Krawczyk https://ipsec.pl

Übungen

Übung 1.1 selbständig lösen

Security

SSL (Secure Sockets Layer) / TLS (Transport Layer Security)

- Mit SSL / TLS wird der gesamte Kanal verschlüsselt
- TLS ist der Nachfolger von SSL
- Sind asynchrone Verfahren
  - Beide haben einen Privaten und Öffentlichen Schlüssel
  - Es gibt ein Handshake in welchem die Public Schlüssel ausgetauscht werden
  - Durch Handshake ist ein gemeinsames Geheimnis gebildet
  - Mit diesem Geheimnis werden alle Nachrichten auf dem Kanal verschlüsselt
- Am besten immer SSL / TLS verwenden um sich vor Lauschattacken zu schützen!

Security

Authentifizierung

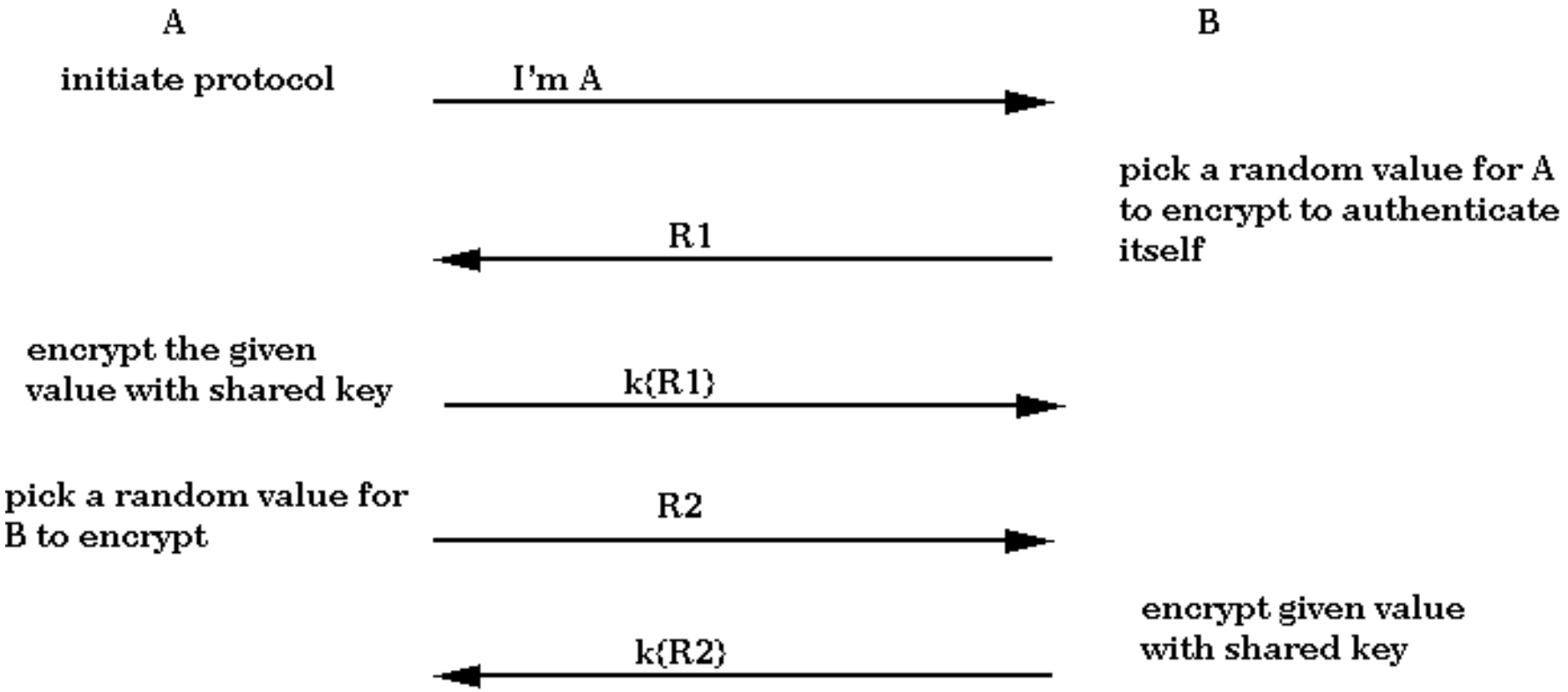
- Wer bist du?
  - Es muss sichergestellt werden, dass man auch der ist den man behauptet zu sein

- Synchrone Verfahren
  - Hat gemeinsames Geheimnis

- Asynchrone Verfahren
  - Jeder hat sein eigenes Geheimnis und damit erstellen wir ein gemeinsames
  - Wird oft mit Zertifikaten gearbeitet
  - In der Praxis oft bevorzugt, da flexibler und sicherer

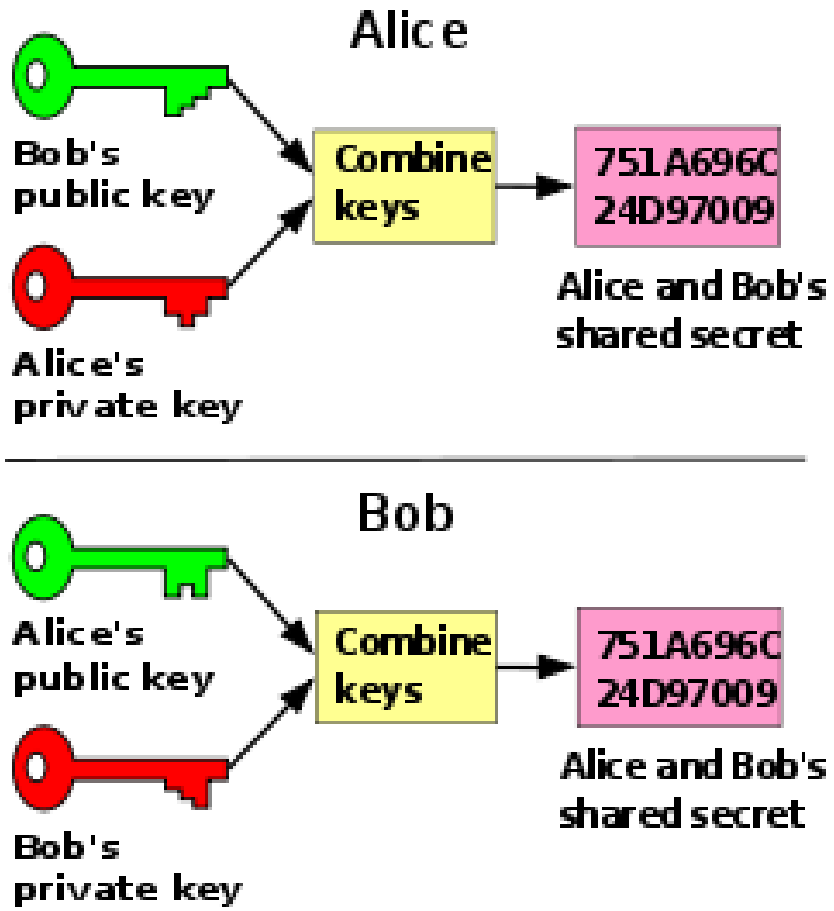
Security

Authentifizierung



Security

Asymmetrische Authentifizierung



Security

Mehrfaktorauthentisierung

- Nutzung mehrerer Faktoren mit unterschiedlichen Eigenschaften in den meisten Angriffsszenarien
  - Etwas was du weisst (Password)
  - Etwas was du hast (Keykarte)
  - Das was du bist (Biometrie, Körpermerkmale, Stimme, Gangmuster)
  - Z.B.
    - Password: Anfällig gegen Phishing, weniger gegen Diebstahl
    - Keykarte: Anfällig gegen Diebstahl, weniger gegen Phishing

Nutzung mehrerer unabhängiger Kommunikationskanäle

- Schutz gegen Unterwanderung eines Kanals (Man-in-the-middle)

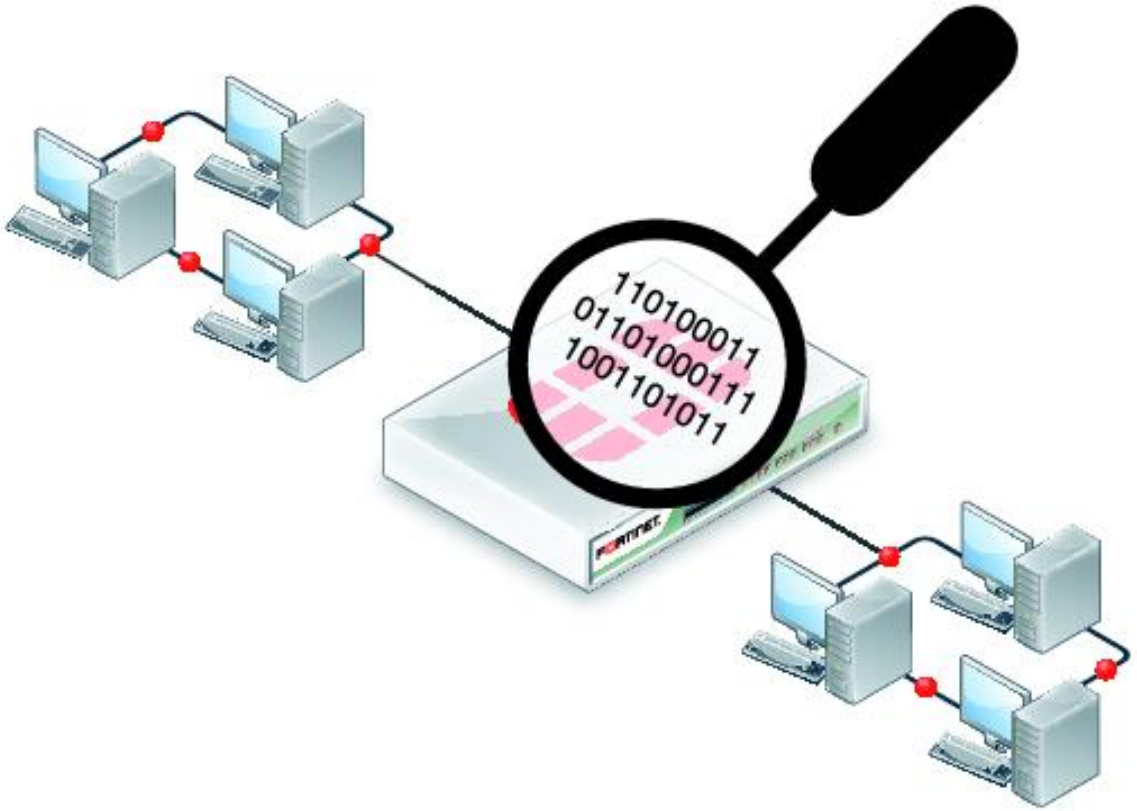
Security

Hash Algorithmen

- Erstellung eines eindeutigen Prüfwerts für einen Wert oder eine Nachricht
- Der Prüfwert für jede Nachricht muss einzigartig sein
- Dieser Prüfwert ist eindeutig und nicht umkehrbar
  - Es gibt keine Möglichkeit von einem Hashwert aus herauszufinden was der Ursprungswert ist
  - Wird aber vom gleichen Ursprungswert mit dem selber Hashalgorithmus wieder ein Hash generiert kommt dabei wieder der selbe Prüfwert heraus
- Bekannte Hash Algorithmen
  - MD5 (veraltet), SHA-1 (veraltet), SHA-2 (empfohlen)
- Hashwerte werden beispielsweise immer benutzt um Passwörter abzuspeichern

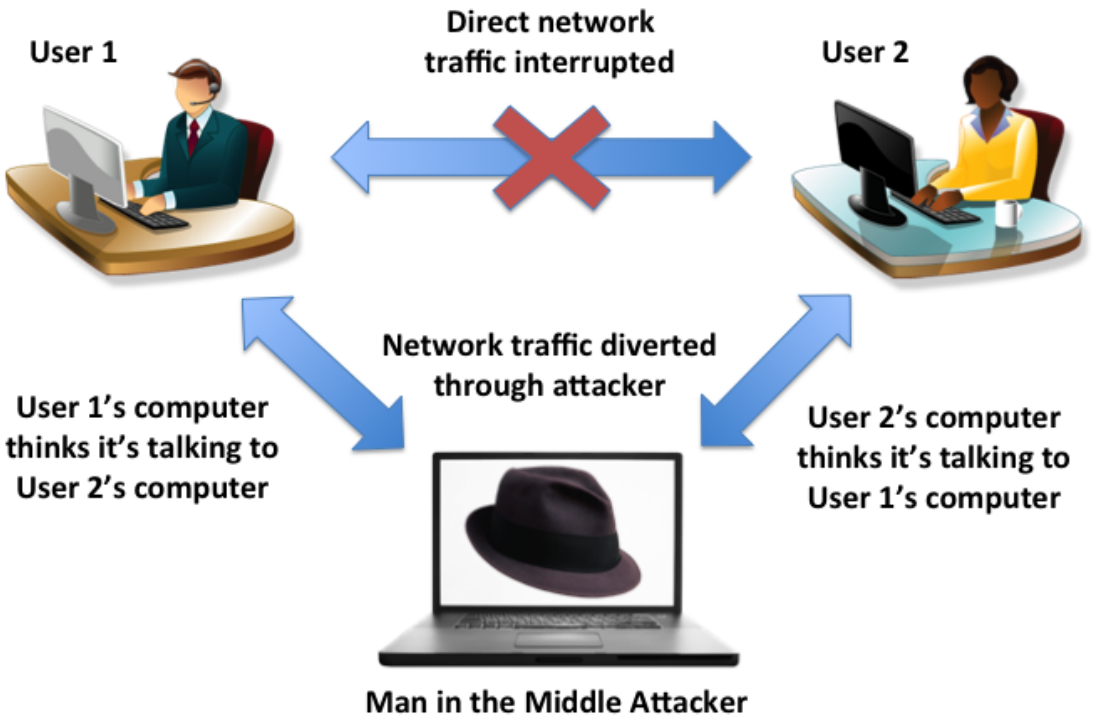
Security

Sniffing



Security

Man in the Middle

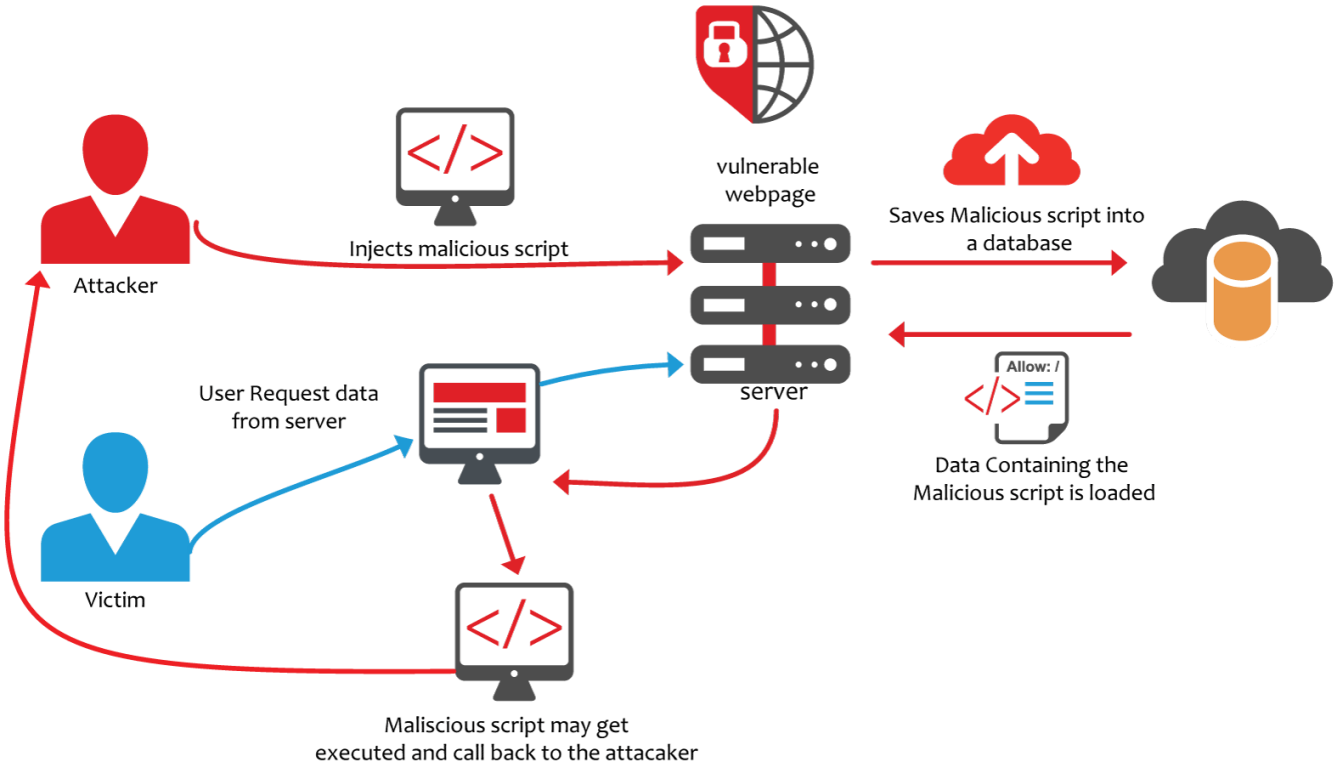


Security

- Cross-Site Scripting (XSS) ist eine Art HTML Injection.
- Cross-Site Scripting tritt dann auf, wenn eine Webanwendung Daten annimmt, die von einem Nutzer stammen, und diese Daten dann an einen Browser weitersendet, ohne den Inhalt zu überprüfen.
- Damit ist es einem Angreifer möglich, auch Skripte indirekt an den Browser des Opfers zu senden und damit Schadcode auf der Seite des Clients auszuführen

Security

Cross Site Scripting (XSS) Beispiel



Security

Was sind mögliche Konsequenzen von erfolgreichen XSS Attacken?

- Cookies mit sensitiven Daten des Benutzers können gestohlen werden. z.B. Session ID
- Der Angreifer registriert einen Keyboard Event Handler um an sensitive Daten des Benutzers zu gelangen. z.B. User/Pwd oder Kreditkarten Infos
- Phishing der Angreifer erstellt mittels DOM Manipulation eine Fake Anmeldemaske um an die Login Daten des Benutzers zu gelangen.

Security

Wie kann XSS verhindert werden?

- Encoding (HTML escaping), dabei werden Zeichen wie <> in < > umgewandelt.
- Clientseitiges Encoding verschiedene Buildin Methoden und Properties encoden den Inhalt automatisch:
  - *node*.textContent = **userInput**
  - *element*.setAttribute(*attribute*, **userInput**)
- Verwenden eines Template Systems oder Web Application Framework das "Auto-escaping" unterstützt und "Context-aware" ist
- Validation nur gewisse HTML Elemente werden erlaubt (Whitelisting) oder verboten (Blacklisting)

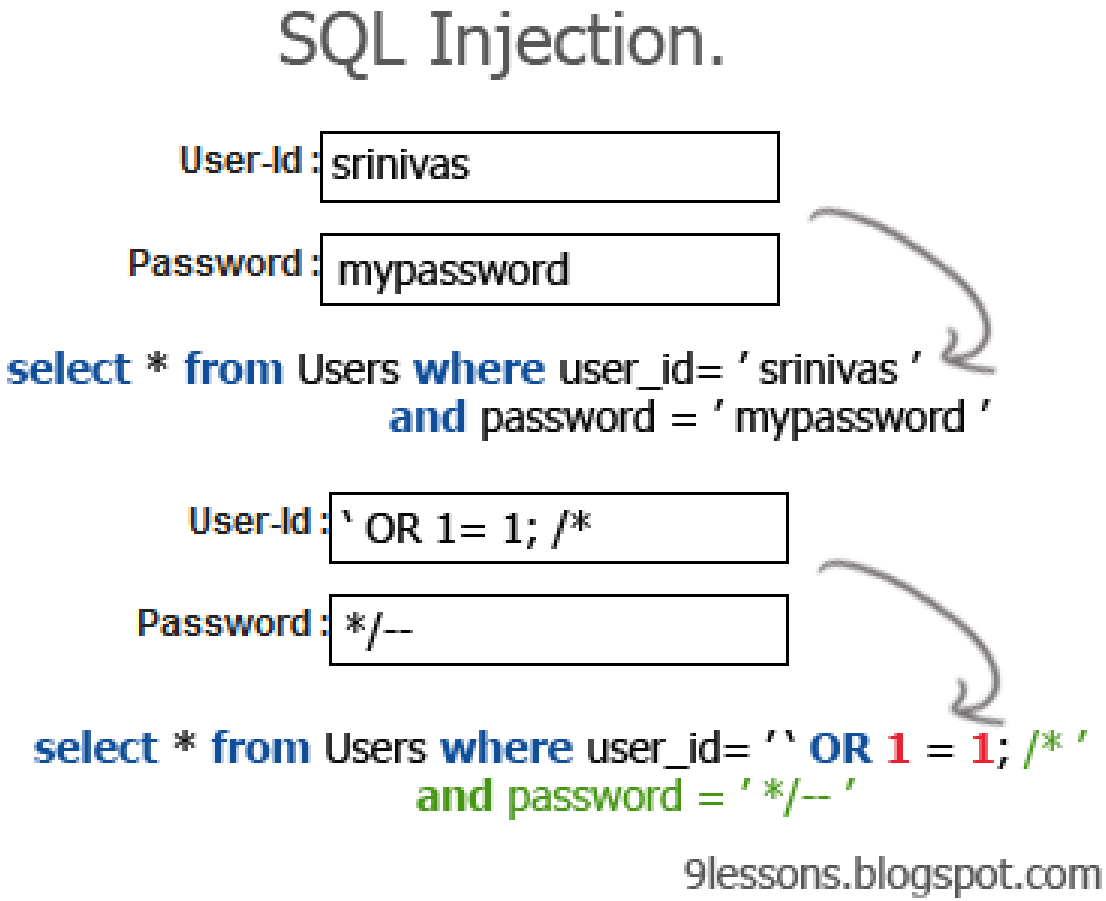
Security

Warum und wie erschwert CSP das Cross Site Scripting (XSS)?

- Inline Sources werden nicht erlaubt (ausser 'unsafe-inline‘ ist definiert)
- Fremde Sourcen werden nicht erlaubt (ausser in CSP definiert)

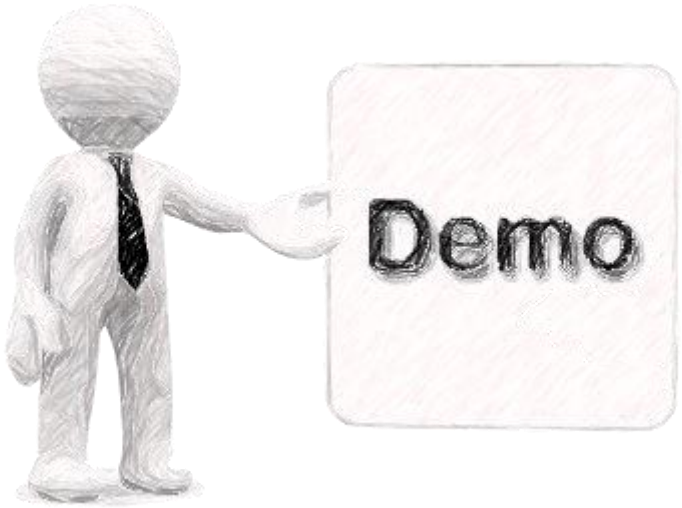
Security

SQL Injection



Google Maps

https://www.google.com/about/appsecurity/learning/xss/



Übungen

Übung 1.2 selbständig lösen

Architektur

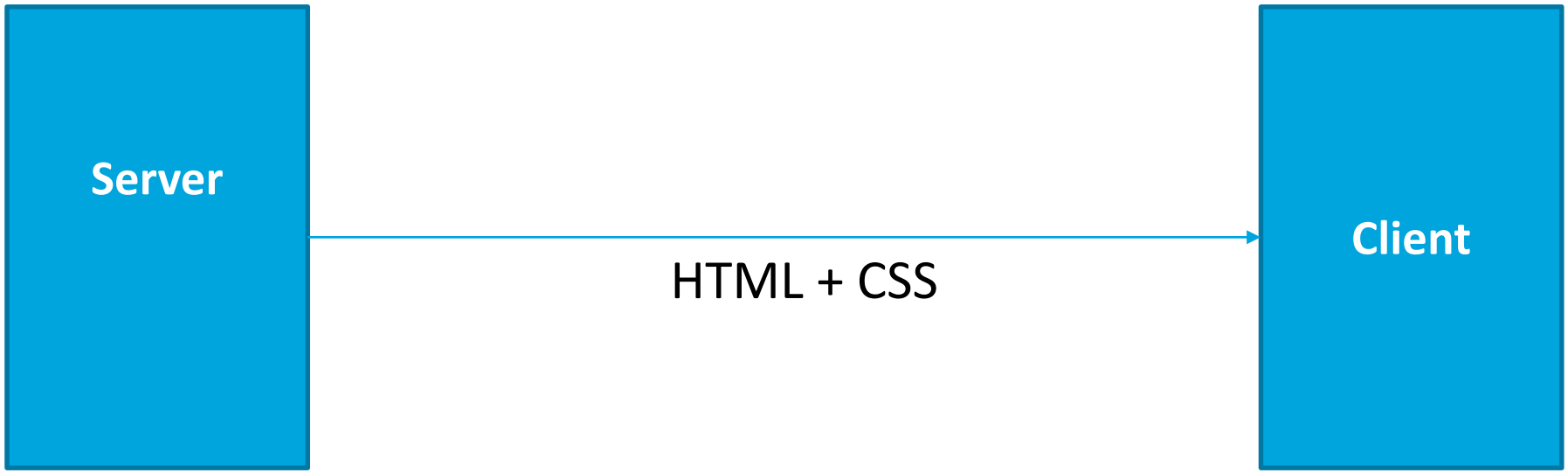
1990 – Blankes HTML
1993 – CGI
1995 – JavaScript
1995 – PHP
1996 – ASP
1999 – JSP
2002 – ASP.NET
2004 – Ruby on Rails
2005 – Die Wende!
2006 – JQuery
2008 – Working Draft HTML5
2009 – AngularJS
2010 – KnockoutJS / Backbone.JS
2016 – Angular 2

Architektur

Statische Seiten (ab 1990)

Architektur

Serverseitiges Rendering (bis 2005)

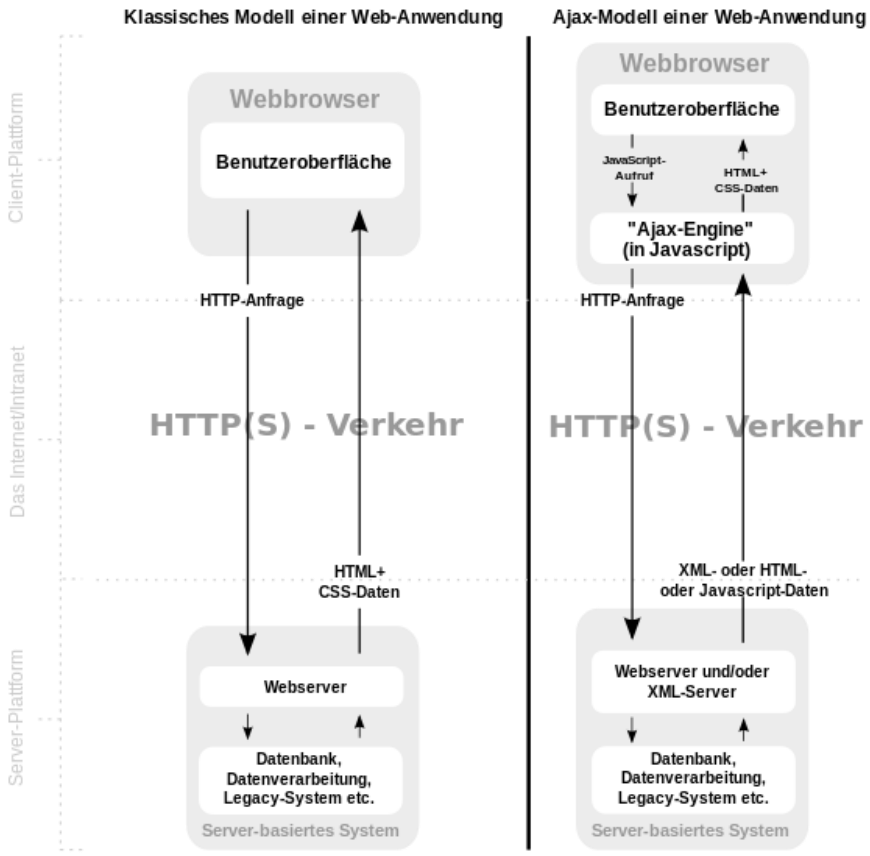


Architektur

Dynamisches nachladen von Seiten mit Ajax Calls

Architektur

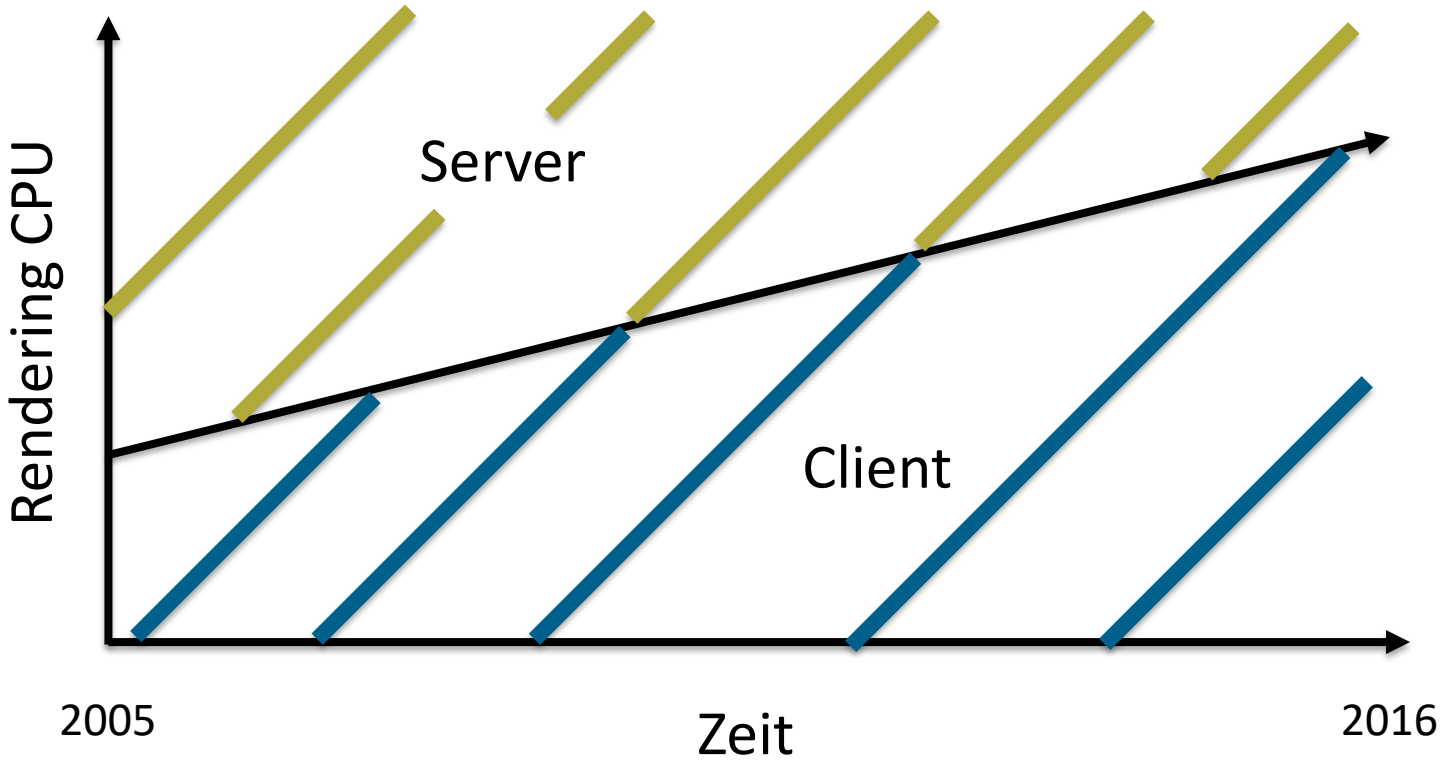
Dynamisches nachladen von Seiten mit Ajax Calls



Architektur

Single Page Applications

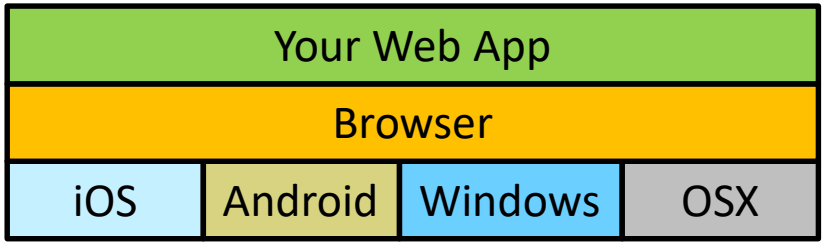
Architektur



Architektur

Vorteile des Webs

- Browser als Betriebssystem
  - Jedes Device hat ein Browser
  - Meta über iOS, Android, Windows, OSX
- Immer mehr browserbasierte Programme
  - Start 2004 mit Gmail
  - Visual Studio Online, Office 365 uvm.
- Ausbau Hardware Zugriff aus Browser



Übungen

Was hat sich im Laufe der Zeit mit den Websites verändert?
Was war die Erfindung, die SPA's überhaupt möglich machte?
Wohin streben die Entwicklungen des Browsers?

Recap