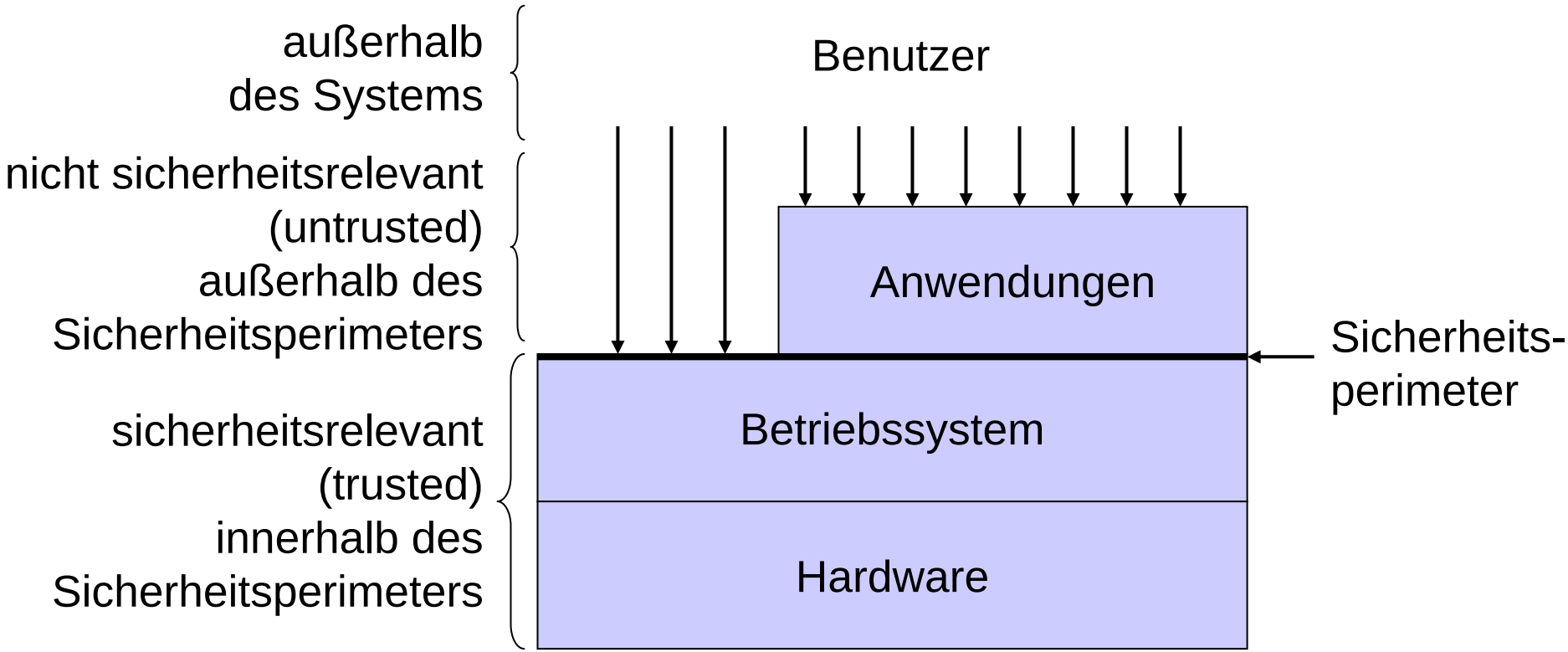
“The reference monitor is moving into the web page”

[Brendan Eich, Mozilla]

Dieter Gollmann
TU Hamburg-Harburg

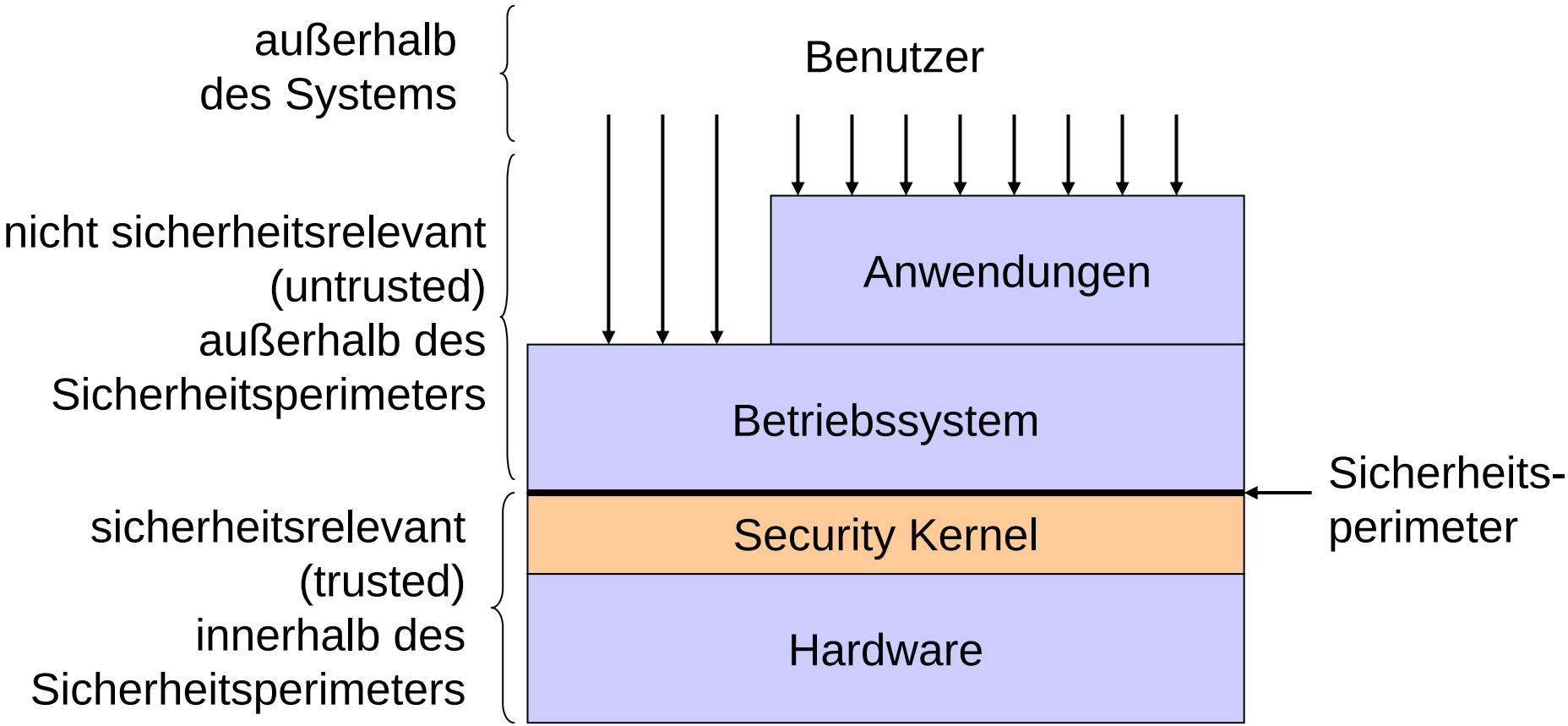
Computersicherheit, 1988

Morrie Gasser, Building a Secure Computer System, Van Nostrand Reinhold



Computersicherheit+, 1988

Morrie Gasser, Building a Secure Computer System, Van Nostrand Reinhold



Sicherheitsstrategie, 1988

- Höhere Sicherheit durch kleinen, verifizierbaren Security Kernel, der den Reference Monitor implementiert.
- Discretionary & Mandatory Access Control.
- Sicherheit wird von den unteren Systemschichten garantiert.
- Den Anwendungen muss nicht vertraut werden.
- Die Verteidiger bunkern sich im Security Kernel ein.

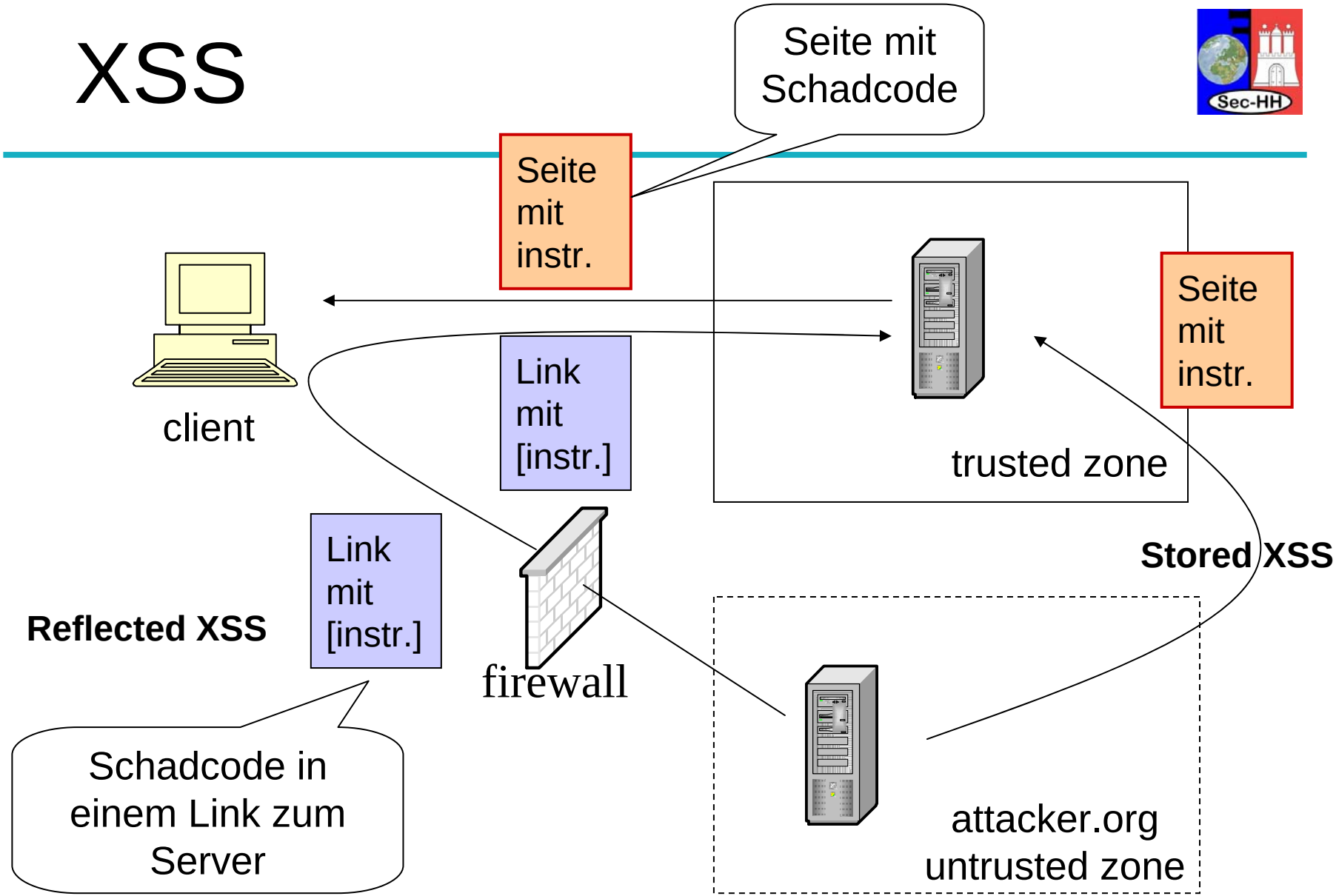
Computersicherheit, 2008

- Cross-Site Scripting
  - #1 in 2007 OWASP Top Ten Vulnerabilities
  - #1 in CVE seit 2005
- SQL Injection
  - #2 in CVE seit 2006
- Cross-Site Request Forgery
- JavaScript Hijacking
- DNS Rebinding
- …

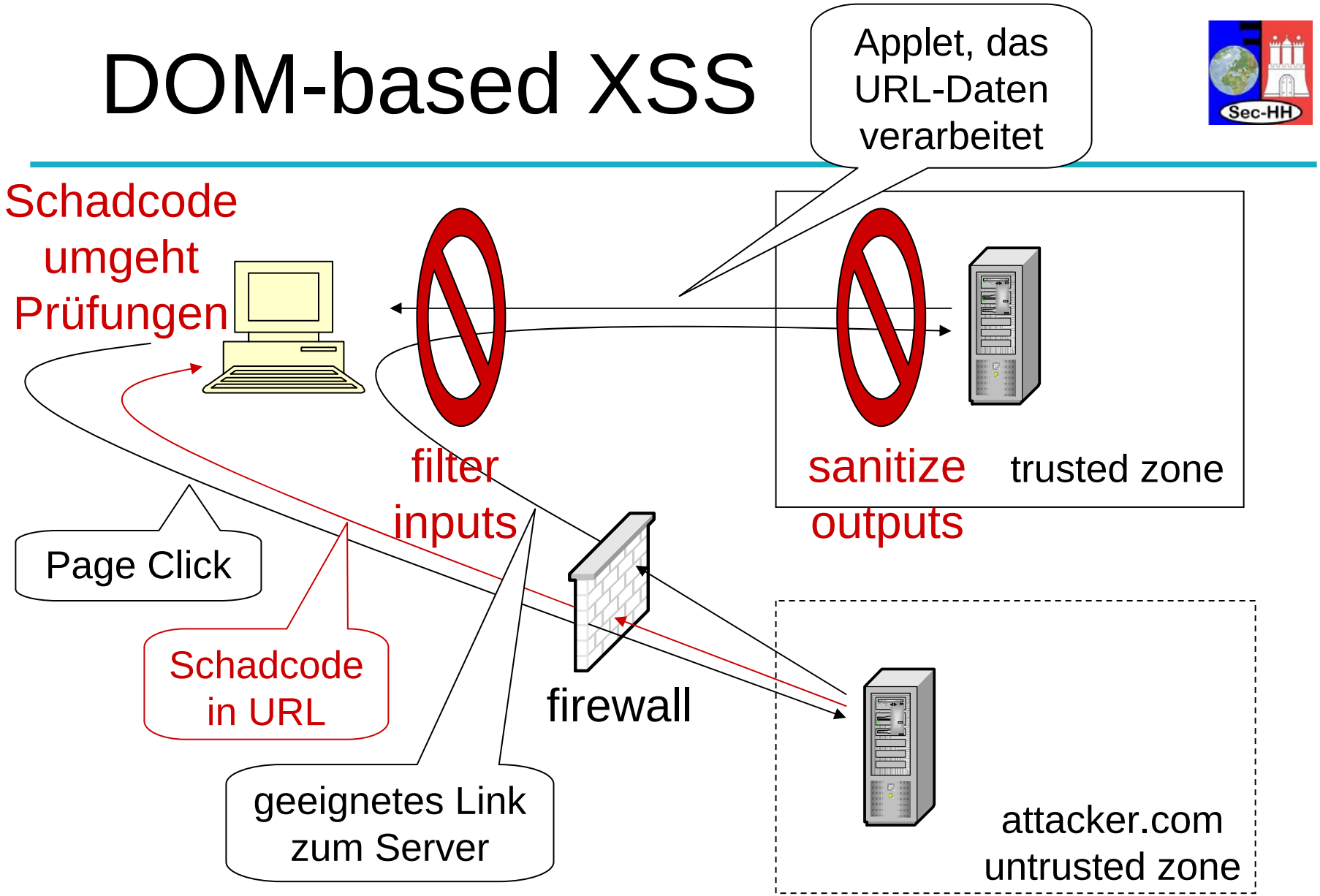
Cross-Site Scripting (XSS)

- Beteiligt: Angreifer, Client (Opfer), Server (welchem der Client „vertraut“).
- Angreifer platziert Schadcode auf einer Webseite am Server (stored XSS: Diskussionsforum) oder auf einer Webseite des Angreifers in einem Link zum Server (reflected XSS: Page-not-found).
- Besucht ein Benutzer die Seite des Angreifers, folgt der Browser im Client automatisch dem Link zum Server.
- Stored XSS: Benutzer besucht Seite am Server.
- In beiden Angriffen ist der Schadcode in der Antwort des Servers enthalten und wird mit den Rechten des Servers ausgeführt.
- Ziel: z.B. Cookie Stealing.
- Umgeht ‚origin based security policy‘ des Clienten.

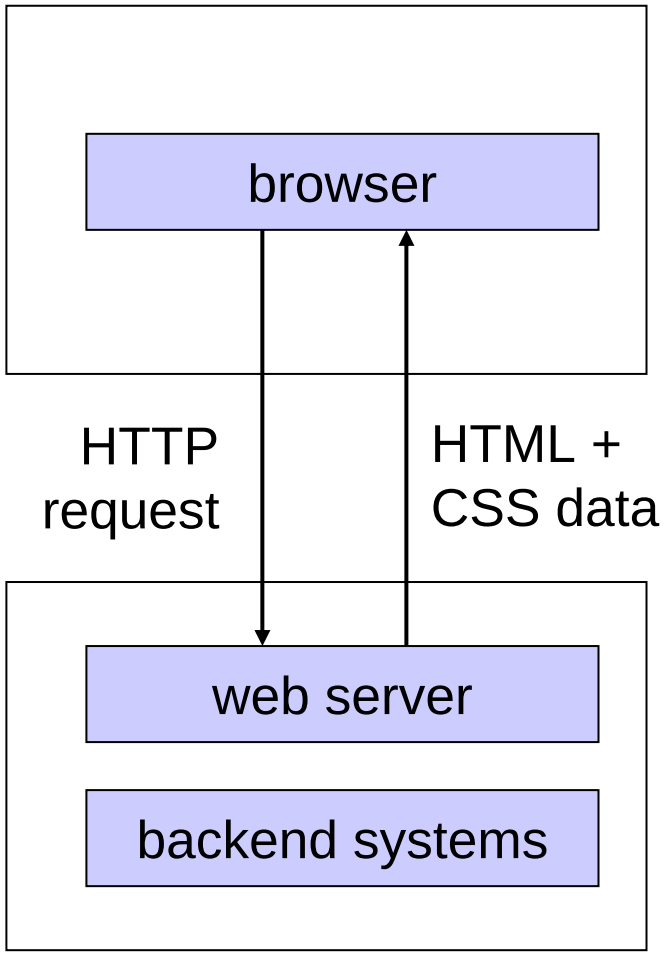
XSS

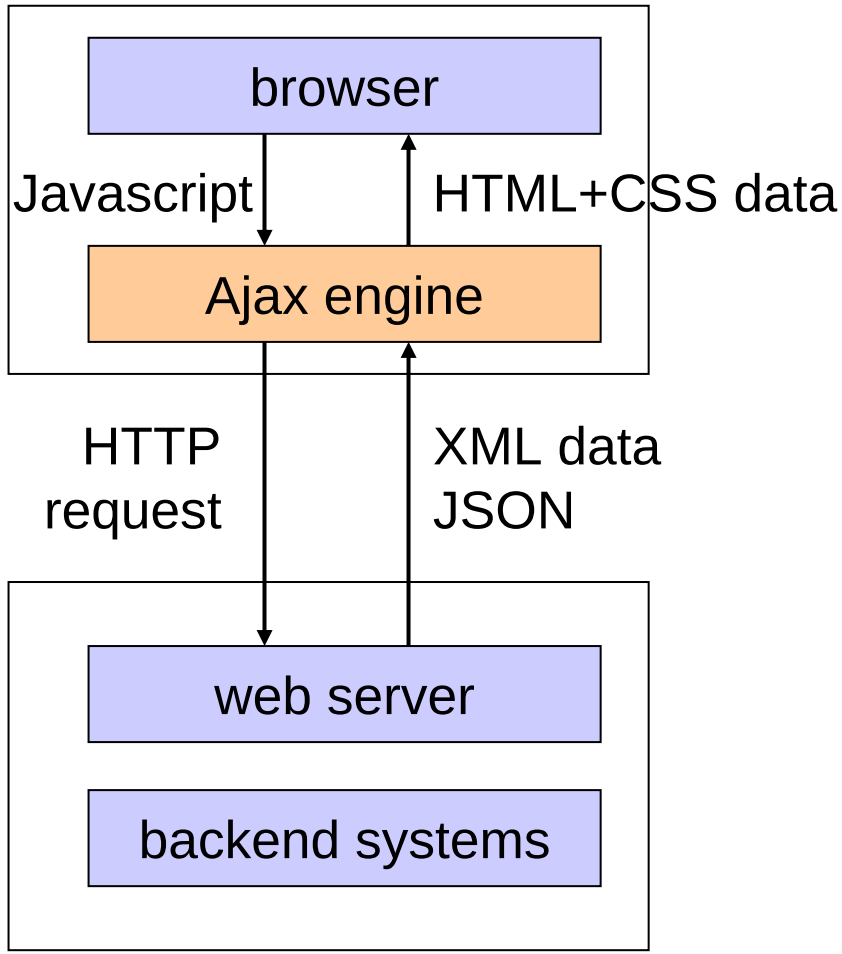

Seite mit Schadcode
Seite mit instr.
client
Link mit [instr.]
trusted zone
Seite mit instr.
Stored XSS
Reflected XSS
Link mit [instr.]
firewall
Schadcode in einem Link zum Server
attacker.org
untrusted zone

DOM-based XSS



Web 1.0 & Web 2.0


browser
HTTP
request
HTML +
CSS data
web server
backend systems


browser
Javascript
HTML+CSS data
Ajax engine
HTTP
request
XML data
JSON
web server
backend systems

JavaScript Hijacking

- Benutzer besucht Seite des Angreifers, die Schadcode in einem Link zum Server enthält.
- Schadcode erfragt sensitive Daten des Benutzers.
- Der Browser erhält diese Daten, falls eine authentifizierte Verbindung zum Server besteht.
- Weiterer Schadcode ändert einen Objektkonstruktor (JSON → JavaScript), so dass die Daten zum Angreifer weitergeleitet werden.
- Schadcode wird im Kontext der Seite des Angreifers ausgeführt; daher ist das Weiterleiten „berechtigt“.

DNS Rebinding

- Same origin policy: Applet kann nur zu dem Server eine Verbindung aufbauen, von dem es geladen wurde.
- Um eine Verbindung aufzubauen benötigt der Browser eine IP Adresse des Servers.
- Authoritativer DNS Server löst in seiner Domain „abstrakte“ DNS Namen in „konkrete“ IP Adressen auf.
- Der Browser „vertraut“ dem DNS Server.
- Vertrauen ist schlecht für Sicherheit!

DNS Rebinding

- Client besucht attacker.org; der DNS Server des Angreifers löst diesen Namen zur IP Adresse des Angreifers mit kurzer Time-to-live auf.
- Angreifer bindet danach attacker.org an die IP Adresse des Opfers.
- Bösartiges Script verbindet sich dann zu attacker.org; wird nun zur IP Adresse des Opfers aufgelöst.
- Abwehr: Traue dem DNS Server nicht in Bezug auf Time-to-live; Pinning des Host Namen auf die ursprüngliche IP Adresse;
  - J. Roskind: Attacks against the Netscape browser. in RSA Conference, April 2001.

DNS Rebinding

- Flexible Autorisierung: Client Browser befragt bei Zugriffskontrollentscheidungen eine Policy, die vom DNS Server erhalten wurde.
- Ein bösartiger DNS Server kann so eine Verbindung zur IP Adresse des Opfers autorisieren.
- Abwehr: Statt die Policy vom DNS Server zu erfragen, prüfe ob die Maschine mit der angegebenen IP Adresse mit der Auflösung einverstanden ist.
  - Verwandt zu Reverse DNS Lookup.
  - Ähnlich zu Abwehrmaßnahmen gegen Bombing Angriffe in Netzwerksicherheit.

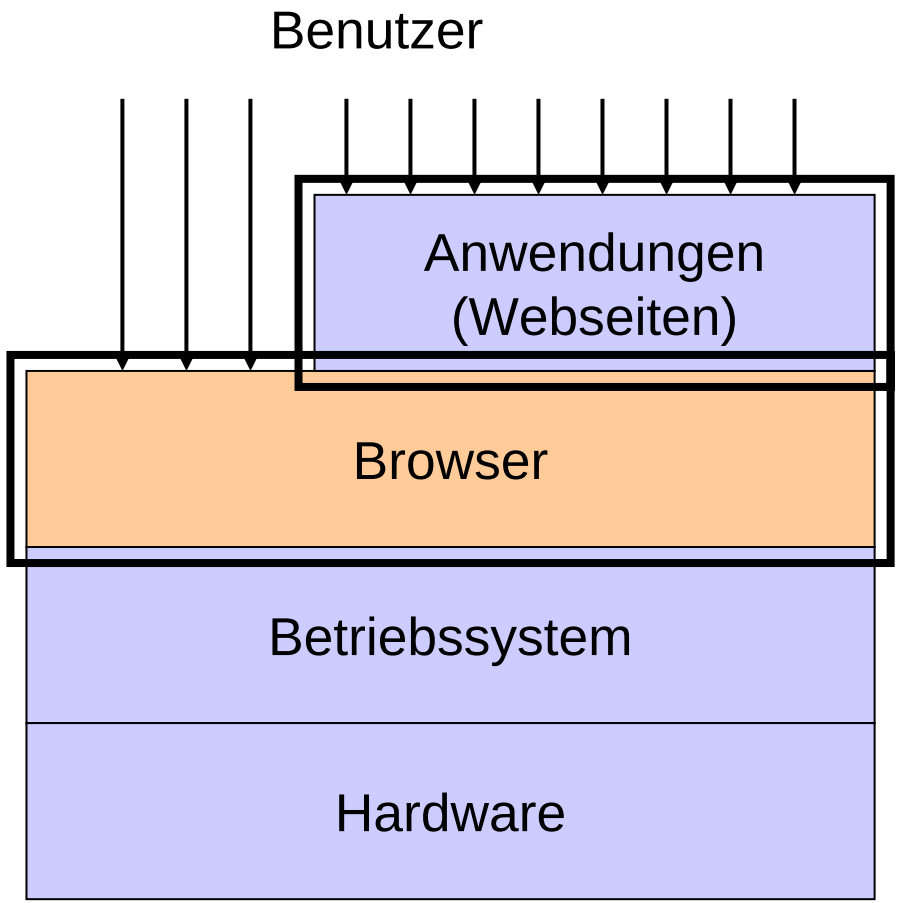
Computersicherheit, 2008

- Same Origin Policies: Wer darf Cookies lesen, wohin darf sich ein Applet verbinden?
- Reference Monitor im Browser (Sandbox).
- Die eben skizzierten Angriffe nützen Schlupflöcher in diesem Reference Monitor.

- Webseiten filtern Eingaben, z.B. um SQL Injection Angriffe abzuwehren.
- Reference Monitor in der Anwendungsebene; die Autoren von Anwendungen müssen entsprechende Filter einbauen.

Computersicherheit, 2008

Sicherheits-
perimeter?



Herausforderungen 2008

- Zugriffskontrolle – Mechanismen: Modellieren der Mechanismen im Browser
  - unter Berücksichtigung von Web 2.0, Plug-ins, Mashups, …
  - unter Hinzufügung geeigneter Mechanismen zur Authentifizierung des Ursprungs von Daten,
  - mit einer Untersuchung der notwendigen Bedingungen an „Authentifizierung“; reicht etwa „Wiedererkennen“?
- Zugriffskontrolle – Policies: Spezifikation und Durchsetzung von Cross Domain Policies.
  - Ajax Cross Domain Policies
  - HTTP Access Control Headers for Cross-Domain Policies, http://www.w3.org/TR/access-control/.

Fazit

- Die Verteidigungslinie gegen aktuelle Angriffe zieht sich hinauf in die Anwendungsebene.
- Die Verteidiger treffen den Angreifer vor den Toren.
- Mechanismen im traditionellen Security Kernel greifen kaum gegen die neuen Angriffe.
- Im Web kommt dem Browser eine zentrale Rolle in der Zugriffskontrolle zu.
- Ist Browsersicherheit die neue Betriebssystem-sicherheit?