

# Angriffe auf Web 2.0

Jürgen Key, Peter Steiert

**OWL-Security 2007**

Paderborn, 25.4.2007

# Agenda

- **Definition „Web 2.0“**
- **Abgrenzung „Web 1.0“ - „Web 2.0“**
- **Schematische Darstellung einer Web 2.0 Anwendung**
- **Angriffsszenarien Web 1.0 - 2.0**
  - Klientseitige Angriffe
  - Man-in-the Middle Angriffe
  - Serverseitige Angriffe
  - Reine Web 2.0 Angriffe
  - Gemischt (Phishing)
  - Privatsphäre (Datenschutz)
- **Zusammenfassung: Angriffe Web 1.0 - 2.0**
- **Gegenmaßnahmen**

## Definition „Web 2.0“

Begriff geprägt von [Tim O'Reilly](#):

- ***"Web 2.0 is the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform."***

# Abgrenzung „Web 1.0“ - „Web 2.0“:

## Web 1.0:

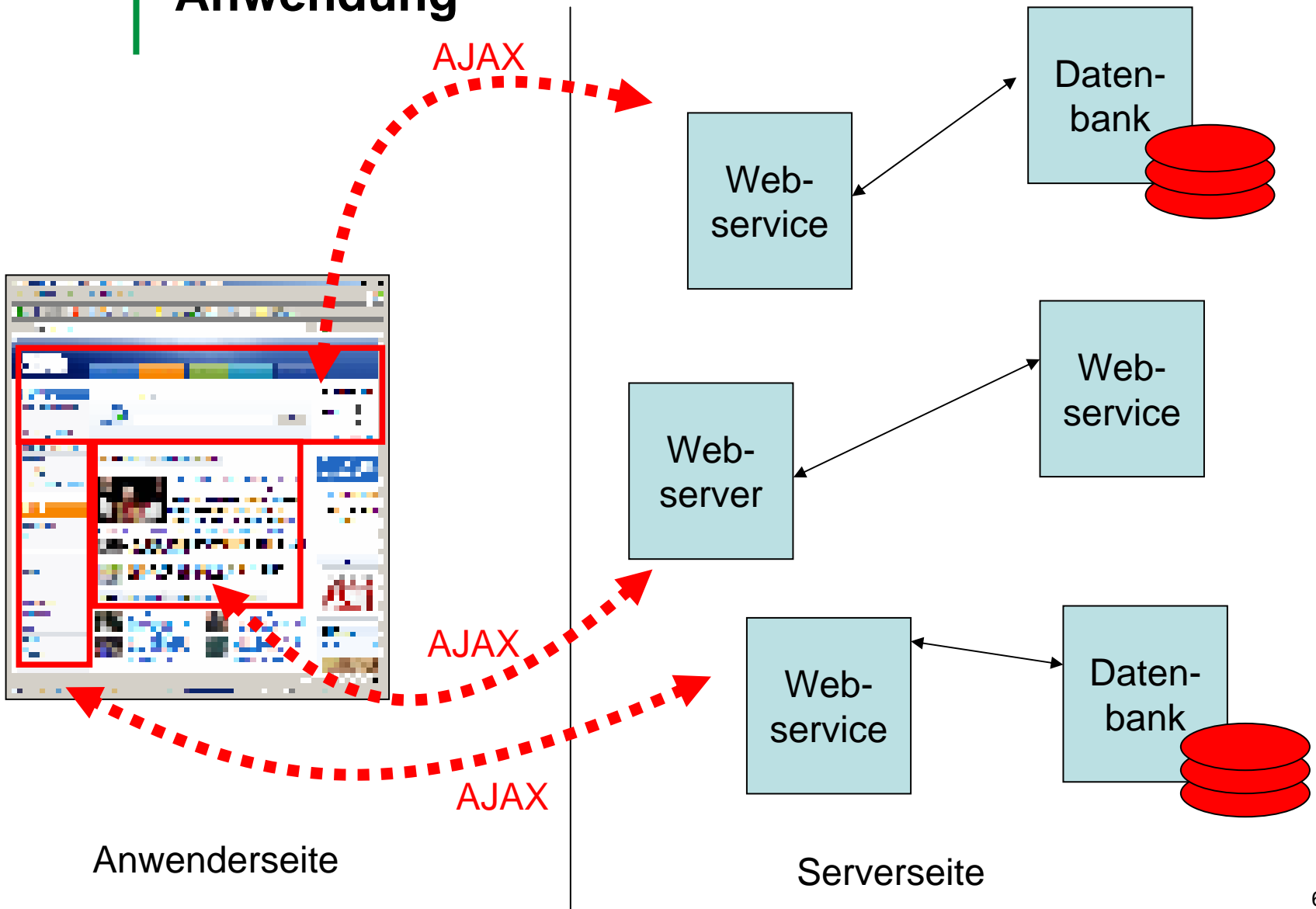
- **Daten, Darstellung, Steuerung meist durch monolithische Anwendung realisiert:**
  - Applikationslogik nahezu ausschließlich serverbasiert
  - Browser dienen nur als Anzeigegeräte
- **Hauptanwendungsgebiet:**
  - “Seitenbasierte”-Anwendungen
  - Interaktionsmöglichkeit zwischen Nutzer und Applikation sind vollständig übertragene, in sich unveränderliche HTML Seiten, die jedoch dynamisch vom Server generiert werden können
  - Strukturierte Darstellung erfolgt durch den Browser (Internet Explorer, Firefox, Opera, ...)
- **Weiterentwicklung der Darstellung und Verarbeitung**
  - > **Browserunterstützt: CSS, Javascript, J-Script, Active-X, etc.**
  - > **Plugins: Java, Shockwave, PDF**

# Abgrenzung „Web 1.0“ - „Web 2.0“:

## Web 2.0 – Web Services der zweiten Generation

- **Eigenschaften von Web 2.0 Anwendungen:**
  - Weiterhin lauffähig im Browser
  - Verlagerung der Applikationslogik vom Server in den Browser
    - Manipulation der dynamischen Daten durch Javascript, J-Script
  - Teilinhalte werden durch verschiedene Webservices bereitgestellt
  - zusätzlich: verstärkte Einbindung dynamischer Datenquellen (Datenbanken, Verzeichnisdienste, andere Webanwendungen, etc.)
  - Gemeinsame Software-Schnittstelle: AJAX (Asynchronous Java and XML)
  - Datenaustausch zwischen den Web-Diensten mithilfe von XML
  - Darstellung der Daten (X)HTML, CSS
- **Bsp: Routenplaner kapseln die Verarbeitung einzelner Services:**
  - Service für Kartendaten
  - Service für Wegfindung, Wegberechnung
  - Service für Darstellung
  - etc.

# Schematische Darstellung einer Web 2.0 Anwendung



# Angriffsszenarien Web 1.0 - 2.0

- **Klientseitige Angriffsszenarien:**
  - absichtlicher/unabsichtlicher Bezug von Mal-Ware, Viren, Trojanern
    - Auslieferung über Webseiten/Internetforen/etc. die unzureichend gesichert sind. <sup>1.)</sup>
    - Ausnutzung von **lokalen** Schwachstellen der gängigen Systeme (Stack-/Heapoverflows), um sich im System zu verankern. <sup>2.) 3.)</sup>

Ablauf: Der Anwender öffnet eine Webseite, die hinterlegte Elemente werden geladen, beim Laden verarbeiten entsprechende browserinterne Programmteile, externe Programme oder Plugins die geladenen Dateien.

Haben diese Schwächen, so lässt sich fremder Programmcode einschleusen, der auf dem lokalen System arbeitet.

Beispiele:

1.) <http://www.heise.de/newsticker/meldung/88285>

2.) <http://www.heise.de/security/news/meldung/87829>

3.) <http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/88060&words=Microsoft%20Patch>

# Angriffsszenarien Web 1.0 - 2.0

- **Klientseitige Angriffsszenarien:**
  - Ausnutzung von Schwachstellen bei der klientenseitigen Verarbeitung der Daten
    - Rendering zur Darstellung von Webseiten (Javascript-, J-Scriptmanipulation)
    - Vorstufe zu Phishing-Angriffen

## Ablauf:

Der Anwender öffnet eine Webseite, die hinterlegten Elemente, inklusive Javascript-/Jscriptcode, werden geladen.

Der Javascriptcode enthält Schadroutinen, die die Verarbeitung beeinflussen und im Browser ausgeführt werden.

z.B.: Umleitung der Session (Man-in-the Middle) auf andere Server, Transfer von Daten (Pin, TAN, etc.) an andere Server.

# Angriffsszenarien Web 1.0 - 2.0

- **Man-in-the Middle Angriffe:**
  - Zwischengeschaltete Systeme, die den Datenverkehr durchleiten
  - Schnittstellen, die Provider Behörden zur Überwachung vorhalten müssen
  - Kein spezifischer Web 2.0 Angriff, auch auf andere Dienste und Anwendungen (VoIP, Email, etc.) anwendbar
  - Auch SSL-Verschlüsselung ohne Clientverifikation kann durch Man-in-the Middle ausgehebelt werden

# Angriffsszenarien Web 1.0 - 2.0

- **Serverseitige Angriffe: SQL Injection**
  - Einschleusung von Fremddaten/Steuerbefehlen in Server-Systeme
  - Ältere Webapplicationen anfällig dafür

z.B.:

„`http://www.meinefirma.de/cgi-bin/shop.pl?ID=123456&datensatz=DATENSATZ=%22DELETE%20*%20from%20firma.addresse%22`“

**Versteckte Schadroutine:**

`Delete * from firma.addressen“`

weiteres Beispiel:

„`http://www.meinefirma.de/cgi-bin/shop.pl?ID=123456&DATENSATZ=%84Insert%20preis=%B41234%B4%20into%20firma.preise%20where%20preis%3E1400%93`“

**Versteckte Schadroutine:**

„`Insert preis='1234' into firma.preise where preis>1400“`

# Angriffsszenarien Web 1.0 - 2.0

- **Serverseitige Angriffe: Remote Exploitation (Stack/Heapoverflow)**
  - Einschleusung von Programmcode in Server-Systeme
  - Übernahme der Anwendung (Session) durch einen Angreifer
  - Starten von bössartiger Software auf dem System
  - Verschieben des Betriebssystems in Virtuelle Umgebungen (Red Pill)

# Angriffsszenarien Web 1.0 - 2.0

## Reine Web 2.0 Angriffe:

- **Ausnutzung von AJAX Verarbeitung:**
  - J - Javascript:
    - Manipulation der Verarbeitung
    - Übernahme/Umleitung der Sitzung
    - Vorgaukeln sicherer Verbindungen (SSL)
  
  - X - XML:
    - Einschleusung, Ausspionieren von Daten in den XML-Datenstrom
    - Ausnutzung von Parser Schwächen (z.B: XML, CSS, etc.)

# Angriffsszenarien Web 1.0 - 2.0

## Gemischte Angriffe – Phishingattacken:

- **Cross-site Scripting:**
  - Gezielte Ausnutzung von Javascript/J-Script häufig in Verbindung mit Social-Engineering und E-Mail-Spamming
  - Häufig betroffene Webseiten: Banken, Auktionsplattformen, Buchhändler, etc.
  - Kombination aus Server und Klientangriff

### Beispiel:

`http://www.meinebank.de/show_archives.php?subaction=showcomments&id=%3Cscript%3Ealert(document.cookie);%3C/script%3E%3E&archive=&start_from=&ucat=&&archive=&start_from=1`

### Original URL:

`http://www.meinebank.de/show_archives.php?subaction=showcomments&id=<script>alert(document.cookie);</script>&archive=&start_from=&ucat=&&archive=&start_from=1`

# Angriffsszenarien Web 1.0 - 2.0

## Gemischte Angriffe – Phishingattacken:

- **Ablauf:**

- *Möglichkeit 1:*

Empfängern werden E-Mails zugesendet, die Links auf original Webseiten enthalten, denen Angreifer Schadcode unterschieben können.

Im Text der Mail wird meist inhaltlich daraufhingewiesen, dass der Account kompromitiert wurde und deshalb Änderungen an persönlichen Daten (Login, Passwort, Pin, TANs) notwendig sind.

Wird der Link angewählt, leitet das Script den Browser automatisch zu Nachbildungen der Original-Webseiten. Alle Daten werden dann auf der Phishing-Webseite eingegeben.

- *Möglichkeit 2:*

In Foren oder auf Webseiten (Shopsystemen, Auktionsplattformen), in denen ein potentieller Verkäufer Text zu Artikeln hinterlegen kann, wird ein präparierter Link hinterlegt.

Wird der Link angewählt, leitet das Script den Browser automatisch zu Nachbildungen der Original-Webseiten. Alle Daten werden dann auf der Phishing-Webseite eingegeben.

# Privatsphäre (Datenschutz)

- **Ausspionieren von Nutzerverhalten, Konkurrenz, etc.**
  - Ziel ist es, regelmäßigen Besuch von Nutzern zu registrieren (User-tracking, meist mit Cookies realisiert)
  - Javascript/J-Script können dazu dienen, zuletzt besuchte Seiten des Nutzers ausspionieren
  - Bei aktiven Cascading Style Sheets kann ein Tracking besuchter Webseiten ohne Einsatz von Scripting erfolgen <sup>1.)</sup> <sup>2.)</sup>

Beispiele:

<sup>1.)</sup> <http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/86113&words=CSS>

<sup>2.)</sup> <http://ha.ckers.org/weird/CSS-history.cgi>

# Zusammenfassung: Angriffe Web 1.0 - 2.0

- **Die Angriffsvektoren zwischen Web 1.0 und Web 2.0 haben sich wenig geändert**
- **Web 2.0 erfordert klientenseitig den Einsatz von Javascript/J-Script.**
  - Damit kommen zu den bereits bestehenden Web 1.0 Schwachstellen weitere Schwächen der Klienten (Browser) hinzu
- **Der wichtigste Schwachpunkt ist und bleibt der Mensch. Bewusste oder unbewusste Fehler entstehen durch**
  - Anwender
  - Programmierer
  - Zeitdruck und Unkenntnis verstärken dies bei Implementierung und Entwicklung

## Gegenmaßnahmen:

- **Schaffung Bewusstsein:**
  - **Firmendaten sind IHRE Daten !**
  - **Ihr wirtschaftlicher Erfolg ist an Ihre Daten gebunden !**
  - **IT-Sicherheit liegt mit in der Verantwortlichkeit des Managements!**
- **Immer stärkere Anforderungen an die IT-Sicherheit (z.B. KonTraG, BASEL II, Sarbanes-Oxley Act, etc).**
- **Schaffung eines „gesunden Misstrauens gegenüber unbekanntem Webseiten“**

# Gegenmaßnahmen:

## Einteilung in organisatorische und technische Sicherheit:

- **Auswahl an Organisatorischen Sicherheitsmaßnahmen:**
  - Festlegung von klaren Regeln (Policies) und Strukturen, wie mit Web 1.0 und 2.0 Anwendungen umzugehen ist
    - Einteilung in erforderliche und nichterlaubte Anwendungen
  - Festlegung der privaten Nutzung am Arbeitsplatz (Webmail ist ein Einfallstor für eine Vielzahl von Malware)
  - Regelmäßige Softwareupdates (Betriebssystem, Virens Scanner, Firewalls, etc.)
  - Regelmäßige Auditierung, Schulung, Tests auf Lücken
  - Ausnutzung der Sicherheitszonenfunktionalität für Browser (vor allem Internet Explorer), Blockierung von Javascript, J-Script bei verdächtigen Seiten. Viele Web 1.0 Seiten arbeiten ohne Scripting, Web 2.0 Seiten nicht.

# Gegenmaßnahmen:

## Auswahl an Technische Sicherheitmaßnahmen:

- **Webserversicherheit:**
  - Regelmäßige Wartung (Patches, Updates, etc.) der eigenen Dienste
  - Prüfung der eigenen Webapplikationen auf Cross-Site-Scripting Anfälligkeit
  - Blockierung von Javascript, J-Script wenn möglich
- **Klientsicherheit:**
  - Einsatz von Virens Scanner, Malwarescanner, etc., wenn möglich schon am Internetzugangspunkt. Zentraler Web-Proxy mit Filter für Viren/Trojaner.  
Nachteil:
    - geringerer Netzdurchsatz,
    - verschlüsselte Daten können nicht geprüft werden
- **Andere Dienste:**
  - Einsatz von SPAM Filtern, ein Grossteil der Phishingangriffe werden über SPAM initiiert  
**Nachteil:**
    - verschlüsselte Daten können nicht geprüft werden

# Vielen Dank für Ihre Aufmerksamkeit!

**Jürgen Key | Peter Steiert**

**NetSys.IT Information & Communication**

Weimarer Str. 28

98693 Ilmenau

Tel ++49.3677.203515

Fax ++49.3677.203515

E-Mail [jkey<AT>netsys-it.de](mailto:jkey@netsys-it.de) | [psteiert<AT>netsys-it.de](mailto:psteiert@netsys-it.de)

Web <http://www.netsys-it.de>