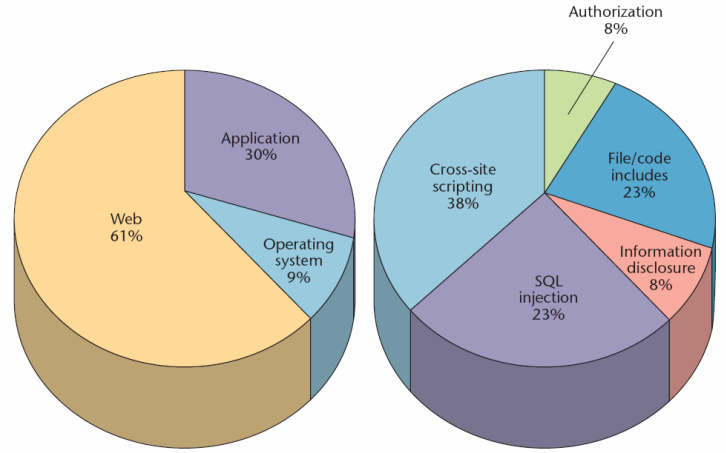
Web-Applikationen: Angriffe – und wie Sie sich dagegen verteidigen

Dr. Marc Rennhard
Dozent für Informationssicherheit
Institut für angewandte Informationstechnologie
Zürcher Hochschule Winterthur
marc.rennhard@zhwin.ch

Gemeldete Schwachstellen Mai 2006



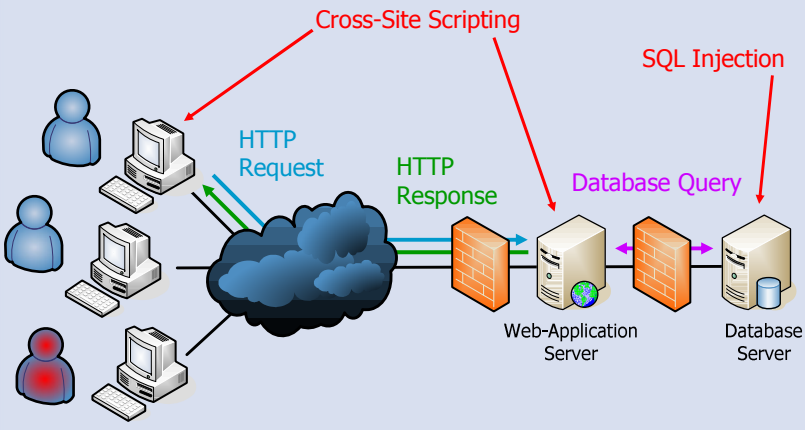
Quelle: IEEE Security & Privacy Magazine, July/August 2006

GO OUT
klubschule MIGROS
SONICWALL
interway
ZHW
IT - Security Forum #6
InfoWeek.ch
IT-Security
Blickpunkt KMU

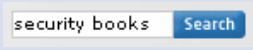
Wieso Web-Applikationen?

- Sind heute sehr verbreitet
  - e-Banking, e-Commerce, Informationsportale...
- Werden immer komplexer → anfälliger auf Schwachstellen
  - Dynamischer statt statischer Inhalt
  - Mehrere Komponenten auf der Serverseite
  - Client- und serverseitige Scripts
- Zugriff auf potentiell hochsensitive Daten
  - Kundendaten, finanzielle Information...
  - Via Web-Applikation hat auch der Benutzer/Angreifer darauf Zugriff

Web-Applikationen - Angriffspunkte



SQL Injection

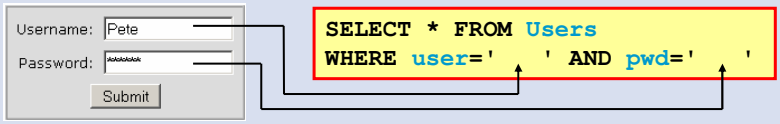
- Web-Applikationen verwenden oft Datenbanken
  - Benutzer-/Kundendaten, Produktdaten...
- Benutzer greifen indirekt auf die Datenbank zu:
  - Eingaben mittels Web-Formular 

  - Daten werden zum Webserver gesendet und dort von einem Skript übernommen
  - Skript verwendet die Daten in einer Datenbankabfrage, generiert eine Webseite und sendet diese zum Benutzer
- Die vom Benutzer eingegebenen Daten werden als Teil des HTTP Requests gesendet
  - GET /path/search.php?string=security books HTTP/1.0

SQL Injection Beispiel - Login

- Benutzerverwaltung: Tabelle Users

user	pwd	email
Pete	perObINa	peter@pan.org
John	hogeldogel	john@wayne.us



  - Skript login.php erhält die eingegebenen Zugangsdaten und verwendet diese in einer SQL Query
  - Benutzername/Passwort existiert: Query gibt eine Zeile zurück → der Benutzer ist identifiziert und erhält Zugang
  - Andernfalls gibt die Query keine Zeile zurück und der Zugang wird verwehrt

SQL Injection Beispiel - Attacke

- Ziel des Angreifers: Zugang zum System ohne Kenntnis von Benutzername/Passwort
  - Dazu wählt er die eingegebenen Daten so, dass die SQL Query in jedem Fall mindestens eine Zeile zurückgibt
- Bei Logins funktioniert dies häufig mit ' or "='

```
SELECT * FROM Users
WHERE user='' or ''='' AND pwd='' or ''=''
```

immer TRUE

  - SQL Query gibt alle Zeilen der Tabelle Users zurück, meist wird login.php nur die erste Zeile davon beachten
  - Der Angreifer wird von login.php als der dieser Zeile entsprechende Benutzer identifiziert und erhält Zugang

SQL Injection – Weiteres Beispiel

- Benützer wählt Produktkategorie via Drop-List:
  - GET /path/showproducts.php?category_id=17 HTTP/1.0

```
SELECT name, description FROM Products WHERE catID=
```

- Der Angreifer möchte zusätzlich auch alle Benutzer und Passwörter erhalten
  - GET /path/showproducts.php?category_id=17 UNION SELECT user, pwd FROM Users HTTP/1.0

```
SELECT name, description FROM Products WHERE catID=17
UNION SELECT user, pwd FROM Users
```

- Fazit: Fast beliebiger Datenzugriff durch Manipulation der SQL Queries möglich
  → enormes Schadenspotential

Cross-Site Scripting (XSS)

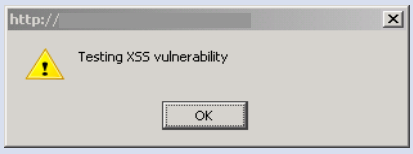
- Javascript: In Webseiten eingefügter Code, der im Browser ausgeführt werden
  - Wertet die Funktionalität von Webseiten massiv auf
- Oft enthalten dynamisch generierte Webseiten die von einem Benutzer eingegebenen Daten
  - Produktresultatseiten, Google etc. zeigen den eingegebenen Suchstring an
- Bei XSS nutzt ein Angreifer dieses Feature aus:
  - Angreifer sendet in einem Web-Formular ein Javascript zum Server
  - Server baut das Javascript in die dynamisch generierte Webseite ein
  - Sobald die Webseite im Browser angezeigt wird, wird das Script ausgeführt

Testen auf XSS Schwachstellen

- Eingabe eines einfachen Javascripts in verschiedenen Feldern von Web-Formularen:

```
<script>alert("Testing XSS vulnerability");</script>
```

- Bei Erfolg öffnet sich ein Popup-Fenster



  - Dies bedeutet, der Webserver übernimmt vom Benutzer eingegebene Javascripts in generierte Webseiten
  - → Proof-of-Concept für die Verwundbarkeit des Webservers auf XSS Attacken
  - → Angreifer kann damit prinzipiell jedes Javascript einfügen

GO OUT IT-SECURITY HOSTING
klubschule MIGROS
SONICWALL
interway
z:w
IT - Security Forum #6
InfoWeek.ch
IT-SecuriTy.
Blickpunkt KMU

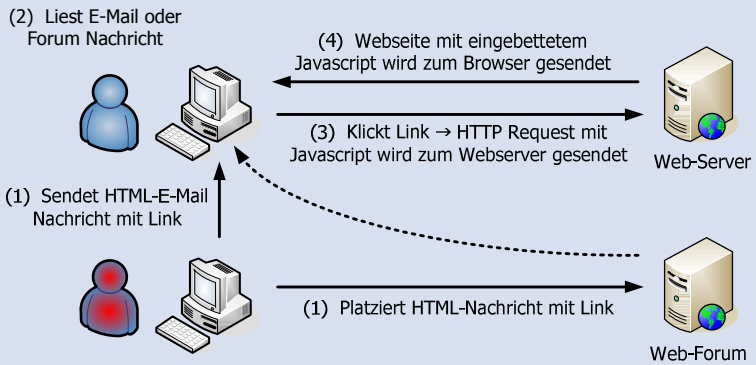
Was kann man mit XSS tun?

- Schafft es ein Angreifer, Javascript Code im Browser eines Opfers ausführen zu lassen, ist zB folgendes möglich:
  - Laden von weiterem Javascript Code von einem Server
  - Auslesen der aktuell verwendeten Session-ID und damit Übernahme einer Session (e-Commerce, e-Banking…)
  - Dynamisches Anpassen der Webseite bei der Darstellung, zB um den Login-Screen zu ersetzen
- Problem für den Angreifer:
  - Um Javascript im Browser eines anderen Benutzers ausführen zu lassen, muss dieser Benutzer selbst das Javascript an den Webserver senden
  - Wie kann der Angreifer das erreichen?

92

Einfügen von Javascript

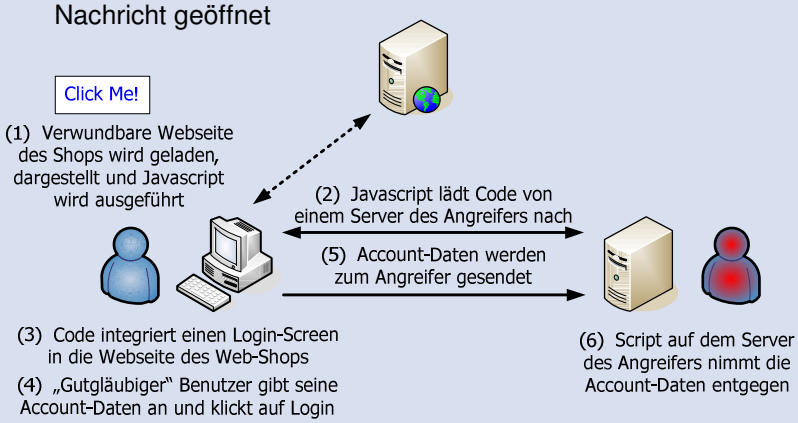
- Script wird per HTTP Request zum Server gesendet
  - GET /path/search.php?string=<script>..</script> HTTP/1.0
  - → Kann als Link in einem HTML-Dokument eingebaut werden (zB E-Mail, Nachricht in Web-Forum)
  - Ein Klick auf den Link lädt die Webseite mit dem Javascript



93

Konkrete Attacke mit XSS, Demo

- Opfer hat Account für einen Web-Shop, Angreifer möchte Account-Daten erhalten
- Angreifer hat ein entsprechendes Javascript in einem Link in einer Nachricht in einem Web-Forum platziert, Opfer hat die Nachricht geöffnet



Gegenmassnahmen

- Das Problem liegt darin, dass Server Daten vom Benutzer oft nicht überprüfen
- Folgende implementatorische Massnahmen sind wirkungsvoll:
  - Serverseitige Validierung aller Daten, welche vom Benutzer gesendet werden (White Lists)
  - Bereinigung sämtlicher Benutzerdaten, bevor diese in einer Webseite integriert und zurückgesendet werden
  - Bei SQL Injection: Benutzerdaten nicht direkt in SQL Queries einfügen sondern Daten einer Stored Procedure auf dem Datenbankserver übergeben

IT - Security Forum #6

Zusammenfassung

Wichtig für die Geschäftsleitung

- Angriffe auf Web-Applikationen gehören heute zu den dominierenden Attacken im Internet
- Entsprechende Schutzmechanismen in die IT-Sicherheitsstrategie aufnehmen

Wichtig für die IT - Abteilung

- Konsequentes Umsetzen der Schutzmechanismen
- Überprüfen der Sicherheit durch Penetration Tests

Wichtig für den Benutzer

- Generelle Wachsamkeit bei Klicks auf Links in HTML-E-Mails und Web-Foren