

CSRF – Cross-Site Request Forgery (1)

- **Angreifer lässt autorisierten Benutzer eine manipulierte URL aufrufen - Beispiel:**

```
http://www.server.com/user.php?  
action=new&id=foo&pass=bar
```

- **Benutzer bringt Rechte mit**
- **Request in URL wird mit Benutzerrechten ausgeführt**
- **Bedingung: User muss authentifiziert ein, z.B. durch:**
 - manuellen Login
 - automatischer Login (Browser-Feature)
 - Cookie-Anmeldung
 - ...

CSRF – Cross-Site Request Forgery (2)

- **XSS**
- **Phishing**
 - „Bitte entsperren Sie Ihr Online-Banking-Konto mittels Pin & Tan (auf unserer gemeinsamen Seite).“
- **Social Hacking**
 - „Guck Dir mein/e tolle/s Seite/Bild an....“
- **URL-Spoofing**
 - Kurz-URL-Dienste
 - <http://jump.to/myEvilHack>
- **html-Emails**
 - externes Nachladen von Bildern

Cross-Site Scripting (XSS)

- „Einbetten“ von Schadcode/Skripten in vertrauenswürdige Webpages/URLS (z.B. mittels Hidden Frame)

```
http://host/a.php?variable="><script>alert('Hello unsuspecting world')</script>
```

Ermöglicht:

- **Client-seitig**
 - Ausführen von Skripten (z.B. JavaScript) im Browser des Opfers
 - Cookie-Stealing
- **Server-seitig**
 - Einbinden externer Libraries & lokales Ausführen dieser eingebunden Skripte
 - Mißbrauch von Webcrawlern für anonyme Angriffe

Cross-Frame / Cross-Window Scripting

- Ähnlich zu XSS
- Besondere Schwachstelle im M\$ Internetexplorer
- Jscript kann auf alle Elemente einer Website zugreifen falls gleicher Ursprung („Same-Origin-Prinzip“)
- Bugs in IE erlaubten Zugriff auch ohne Same-Origin
- Script öffnet (z.B.) Hilfeseite mit lokalen Rechten
- Übergabe von Parametern möglich

<http://www.securityfocus.com/archive/1/298748/2002-11-02/2002-11-08/2>

Application Denial of Service

- „klassisches DoS“
 - Auslasten eines Webservers mit Traffic
 - Auslasten von Datenbankverbindungen
- **Application DoS**
 - Dauerhaftes Belegen bestimmter Ressourcen
 - Blockieren fremder Accounts durch:
 - zu häufige Fehlanmeldung
 - Server sperrt (temporär) den Account
 - User hat keinen Zugang mehr
 - Beantragen eines neuen Passworts
 - Server verschickt neues Passwort
 - User muss neues Passwort erst aus Postfach holen

Google Hacking (1)

- **Google-Anfragen mit Ausnutzen verschiedener (größtenteils) unbekannter Parameter**
- **Boolsche Verknüpfungen**
 - AND, OR, NOT
- **Search-Operators:**
 - intitle, inurl, intext, inanchor site, filetype, author, language restrictions, daterange, numrange, related, phonebook, groups, define, cache, info ...
- **Suchen von Directory Listings, bestimmter Verzeichnisse / Dateien (z.B. `.htusers`)**
- **Nutzung von Google-Übersetzungstool als Proxy**

Google Hacking (2)

- **Möglichkeit Firmen-interne Intranet-/Hilfe-Seiten zu finden**
 - Informationen über Netzwerkstruktur
 - Server-Adressen
 - Telefonlisten
 - ...
- **Hilfsmittel für Social Hacking**
- **Auffinden von persönlichen/beruflichen Emails, Kalenderdaten, Adressbüchern über Outlook-pst-Dateien**
- **Weitere Anwendungen:**
 - Auffinden von Exploit-Opfern (Server-Versionen, etc.)
 - Network-Mapping
 - ...

<http://johnny.ihackstuff.com/>

Broken Authentication

- **Schwachstellen auf Softwareseite:**
 - Unzureichende Sicherung von Passwortdateien
 - Unzureichende Sicherung von Benutzerlisten
 - „Remember my password“-Funktionen
 - „I forgot my password“-Funktionen
 - Unverschlüsselte Übertragung von Logindaten
 - Übergabe von Logindaten mittels 'GET' in der Adresszeile
 - Passwortänderung ohne Re-Authentifikation
 - ...
- **Auf Userseite:**
 - Unsichere Passwörter
 - Ungeschütztes Speichern von Passwörtern im Browser

Improper Information Handling

- **Ausgeben von Fehlermeldungen im Browser**
 - Fehlercodes
 - Syntaxfehler in der Software
 - Speicherzugriffe
 - „Forbidden“ statt „Not found“
 - ...
-
- **Angabe von exakten Softwareversionsnummern**
- **Unsicheres Speichern sensibler Daten (siehe „Broken Authentication“)**

Fin

Fragen?