Session Management und Cookies

Max Tafelmayer

Motivation

- HTTP ist ein zustandsloses Protokoll
- Je Seitenaufruf muss eine neue Verbindung aufgebaut werden
- Server kann keinen Zusammenhang zwischen verschiedenen Anfragen herstellen
- IP-Adresse zur Identifizierung ist nicht zuverlässig (Proxy, Router)

Session Management

- Ziel: Identifizierung eines Users über mehrere Seitenaufrufe hinweg
- Lösung: Verwendung einer Session ID
  - 1. Beim Start der Session wird eine eindeutige Session ID erzeugt
  - 2. Die Session ID wird in der Antwort auf die Anfrage mitgeschickt
  - 3. Bei jeder weiteren Anfrage wird die Session ID wieder an den Server geschickt
- Die Daten der Session werden auf dem Server zur jeweiligen Session ID gespeichert
- Methoden zur Speicherung: Filesystem, Speicher oder Datenbank

Session ID

- Session ID ist ein String: b3ecbc1d1e6a6a4e5724ab140ec63374
- Eine Session ID muss folgende Anforderungen erfüllen:
  - nicht vorhersehbar
  - nicht reproduzierbar
  - ausreichende Länge
- Drei Methoden zur Übertragung:
  - URL Parameter: `http://www.example.com/article.php?id=1&sessionid=12345678`
  - Formularfelder: `<input type="hidden" name="sessionid" value="12345678" />`
  - Cookies
- Cookies sind die sicherste und transparenteste Methode

Cookies

- Bestandteil des HTTP Protokolls:
  - Response Header: `Set-Cookie: sessionid=12345678; expires=Mon, 09-Jan-2005 08:00:00 GMT; max-age=86400; path=/; domain=.example.com; version="1";`
  - Request Header: `Cookie: sessionid=12345678; $path=/; $version="1";`
- Persistent Cookies: haben Ablaufdatum, gespeichert auf der Festplatte des Users
- Session Cookies: werden beim Schließen des Browsers gelöscht, befinden sich nur im Speicher des Users
- Gegen die Verwendung von Cookies spricht:
  - schlechter Ruf, ungenaue Identifizierung, Privatsphäre, Anonymität, Werbung
- Cookies stellen kein Sicherheitsrisiko dar

Authentifizierung

- Sessions werden oft zur Authentifizierung verwendet
- Die Session ID dient dann als Zugangsberechtigung
- Oft werden keine weiteren Maßnahmen getroffen um die Rechte eines Users zu prüfen

→ Session IDs sind deswegen ein beliebtes Angriffsziel

Gefahren

- Interception: Abfangen der Session ID → HTTPS
- Prediction: Vorhersage der Session ID → guter Generator
- Brute-Force: Berechnung der Session ID → lange Session ID
- Session Fixation: vorherige Erzeugung der Session ID
- Session Hijacking: (versehentliche) Preisgabe der Session ID
- Zuständigkeit:
  - Webserver: Interception, Prediction und Brute-Force
  - Webapplikation: Session Fixation und Session Hijacking

Session Fixation und Session Hijacking

	Session Fixation	**Session Hijacking**
Zeitpunkt	Attacke vor dem Login	Attacke nach dem Login
Dauer des Zugangs	einmalig, temporär oder langfristig	normalerweise einmalig
Aufrechterhaltung	evtl. nötig bis zum Login	keine Aufrechterhaltung
Ausgenutzte Schwachstellten	XSS (Domain), Meta Tag (Domain), Link, Formular, Session Adoption, Server (Domain), DNS Server, Netzwerkverkehr	XSS, HTTP Referer, Netzwerkverkehr
Bereiche für einen Angriff	Netzwerkverbindung, alle Webserver in der Domain, DNS Server	Netzwerkverbindung, Webserver

Gegenmaßnahmen allgemein

- XSS verhindern
- Session ID an IP-Adresse und Daten im Header binden
- Session ID an das SSL Zertifikat des Users binden
- Timeouts für Session IDs verwenden
- Session ID nach Logout oder Timeout auch auf dem Server löschen
- Vor wichtigen Aktionen noch mal die nötigen Rechte prüfen
- Keine „Remember me“ Funktion verwenden
- In Shared-Hosting-Umgebungen Session IDs nicht in einem gemeinsamen Ordner speichern

Gegenmaßnahmen zu Session Fixation

- Die Session erst nach erfolgreichem Login starten
- Session IDs von Usern vor dem Login ignorieren
- Session ID bei jedem Login erneut erzeugen
- Session ID in Intervallen erneut erzeugen
- Falls keine gültige Session ID vorhanden ist den User als nicht eingeloggt behandeln

Fazit

- Wenn Sessions zur Authentifizierung verwendet werden müssen innerhalb der Webapplikation Maßnahmen gegen einen Missbrauch der Session ID getroffen werden
- Vor sensiblen Aktionen sollte immer eine erneute Authentifizierung durchgeführt werden
- Der Grad der Sicherheit muss im Einklang mit dem Risiko eines Einbruchs und der gewünschten Usability der Webapplikation sein