Studiengang Informatik
apsi 2020

Fachhochschule Nordwestschweiz
A. Wagner, M. Schläpfer, R. Wagner

Übungsblatt 6 — 19. Oktober 2020

Aufgabe 6-1: Secure Session Management

In der Vorlesung haben wir Session Management im Zusammenhang mit HTTP besprochen. Beantworten Sie dazu folgende Fragen.

(a) Weshalb ist Session Management bei Verwendung von HTTP so wichtig? Für welche Anwendungsfälle?

(b) Welche Möglichkeiten kennen Sie für das Session Management?

(c) Was ist Session Hijacking? Wie könnte ein Angreifer hierfür vorgehen? Was können Sie dagegen machen?

(d) Welche Eigenschaften muss ein Session Identifier erfüllen?

(e) Ein Freund erzählt Ihnen, dass er als Session Identifier einen SHA-256 Hash des aktuellen Timestamps und des Benutzernamens verwendet. Wie beurteilen Sie dieses Vorgehen? Was raten Sie Ihrem Freund (ausser den Besuch der apsi Vorlesung)?

Aufgabe 6-2: HTTP Parameter

Das HTTP erlaubt das Übermitteln von Parametern mit dem Request. Beantworten Sie dazu folgende Fragen:

(a) Welche beiden fundamentalen Möglichkeiten/Methoden stehen für die Übermittlung von Parametern zur Verfügung?

(b) Was sind die Vor- und Nachteile der beiden Methoden.

(c) Der Freund aus Aufgabe 6-1 erzählt Ihnen weiter, dass er der Einfachheit halber die Login Credentials mittels der GET Methode überträgt. Er argumentiert, dass dies ja kein Problem sei, da er die Verbindung ja mittels TLS 1.3 abgesichert habe. Wie beurteilen Sie diese Aussage?

Aufgabe 6-3: CGI

In der Vorlesung haben Sie CGI kennen gelernt. In dieser Aufgabe werden Sie ein einfaches CGI-Script schreiben. Hierfür müssen Sie sich eine Testumgebung mit Apache und aktivem `modcgi` aufbauen. Sie können hierfür auch einen entsprechenden Docker-Container verwenden. Im Übungsordner auf dem apsi-host finden Sie ein vorbereitetes Dockerfile, welches Sie verwenden können. Gehen Sie dazu in den Ordner und führen Sie folgende Anweisungen aus:

```
$ docker build -t cgi_machine .
$ docker run -v "$PWD/www":/var/www/html -p 8080:80 -d cgi_machine
    /usr/sbin/apache2ctl -D FOREGROUND
```

Damit steht Ihnen nun ein Webserver mit Perl und CGI zur Verfügung. Sie können Ihre HTML und Script Files nun einfach im Ordner www, welcher generiert wurde ablegen. Achten Sie bei ausführbaren Dateien darauf, dass diese auch die entsprechenden Berechtigungen besitzen:

```
$ sudo chmod +x www/ihr-script.cgi
```

In Ihrem Host System können Sie nun den Browser Ihrer Wahl verwenden, um auf den Webserver zuzugreifen:

`http://localhost:8080/[ihr-script.cgi]`

Nun, da Sie einen laufenden Webserver zur Verfügung haben, lösen Sie die folgenden Aufgaben.

(a) Schreiben Sie ein CGI-Script, welches ein Cookie setzt und als HTML Output anzeigt, welche Cookies gesendet wurden. Rufen Sie die entsprechende Seite mehrmals auf.

(b) Schreiben Sie nun ein CGI-Script mit folgenden Eigenschaften:

- Der Benutzer kann in zwei Zuständen sein: eingeloggt oder ausgeloggt.
- Der Default ist ausgeloggt.
- Der HTML Output zeigt eine Seite, auf welcher sich der Benutzer mittels Formular einloggen kann. Wenn sich der Benutzer einloggt, wird ein Cookie mit einem Session Identifier gesetzt. Darauffolgende Requests zeigen dem Benutzer an, dass er eingeloggt ist.

- Wenn der Benutzer eingeloggt ist, wird ein Link angezeigt, mittels welchem er sich ausloggen kann. Dabei wird das Cookie gelöscht (mit leerem Wert überschrieben). Anschliessende requests zeigen dem Benutzer wieder an, dass er ausgeloggt ist.

Folgendes Kurztutorial könnte für Sie hilfreich sein:

`http://krum.rz.uni-mannheim.de/cgi-tut.html`