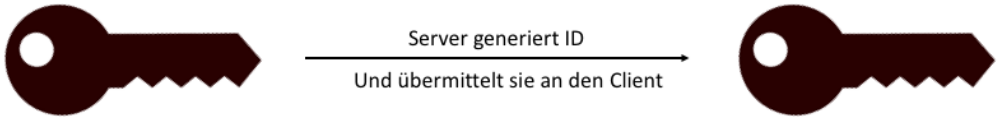
Sessions

Session

- Session bedeutet Sitzung
- Beschreibt eine stehende Verbindung zwischen Client und Server
- Session startet mit dem Login
- Session endet mit dem Logout

Session-ID

- zustandslose Protokolle (HTTP / HTTPS) verwenden die Session-ID um mehrere zusammengehörende Anfragen eines Benutzers zu erkennen.
- muss zufälliger Wert in grossem Wertebereich sein
- Wahrscheinlichkeit, zufällig auf eine gültige Session-ID zu stossen, muss verschwindend klein sein.



It is certainly not random and is based on a hash (default: md5) of these ingredients (see line 310 of code snippet):
IP address of the client
Current time
PHP Linear Congruence Generator - a pseudo random number generator (PRNG)
OS-specific random source - if the OS has a random source available (e.g. /dev/urandom)

Wie erkennen sich Client und Server

die Session-ID muss zwischen Client und Server ausgetauscht werden:

- **als Cookie im HTTP-Header**

  Empfohlene Lösung, Session steht auch statischen Seiten sowie Bilden zur Verfügung.

  Cookie-Informationen werden automatisch im HTTP-Header zwischen Client und Server übertragen

- **als Query-Parameter in der URL -> GET**
  http://www.example.com/index.php?sid=edb0e8665db4e9042fe0176a89aade16

  Sichtbar und deshalb sehr einfach zu lesen und zu ändern

  Session-ID wird in der Log-Datei des Servers aufgezeichnet

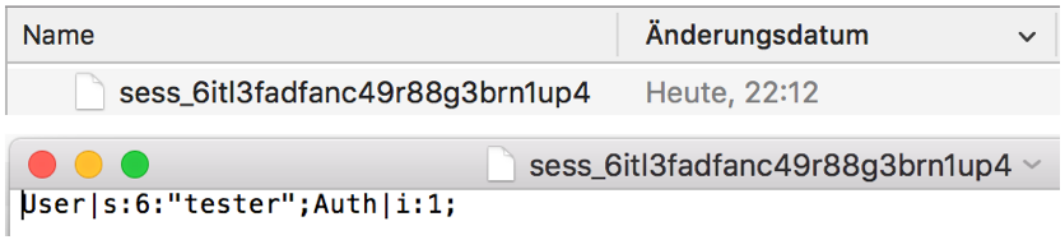
  Suchmaschinen indexieren mglw. mehrere Kopien ein und derselben Seite, weil unterschiedliche URL

- **als Datenteil eines Formulares -> POST**

  Umständlich, da jede Seite Daten per Post übermitteln muss

Wo werden Session-Daten gespeichert?

- in einem Verzeichnis auf dem Webserver
- Windwos: `c:/XAMPP/tmp`
- Mac: `/Applications/XAMPP/xampfiles/temp`
- `/Applications/MAMP/tmp/php`



Session starten

- `session_start();`

- Die Funktion muss in jedem PHP-Script aufgerufen werden, welches auf die Session zugreifen will.

In die Session schreiben

- $_SESSION ist ein superglobales, assoziatives Array, ähnlich wie $_GET oder $_POST. Das Array steht zur Verfügung, wenn die Funktion `session_start();` aufgerufen wurde.

- Schreiben in die Session:

```
$_SESSION[‚username‘] = $username;
$_SESSION[‚loggedin‘] = true;
```


Aus der Session lesen

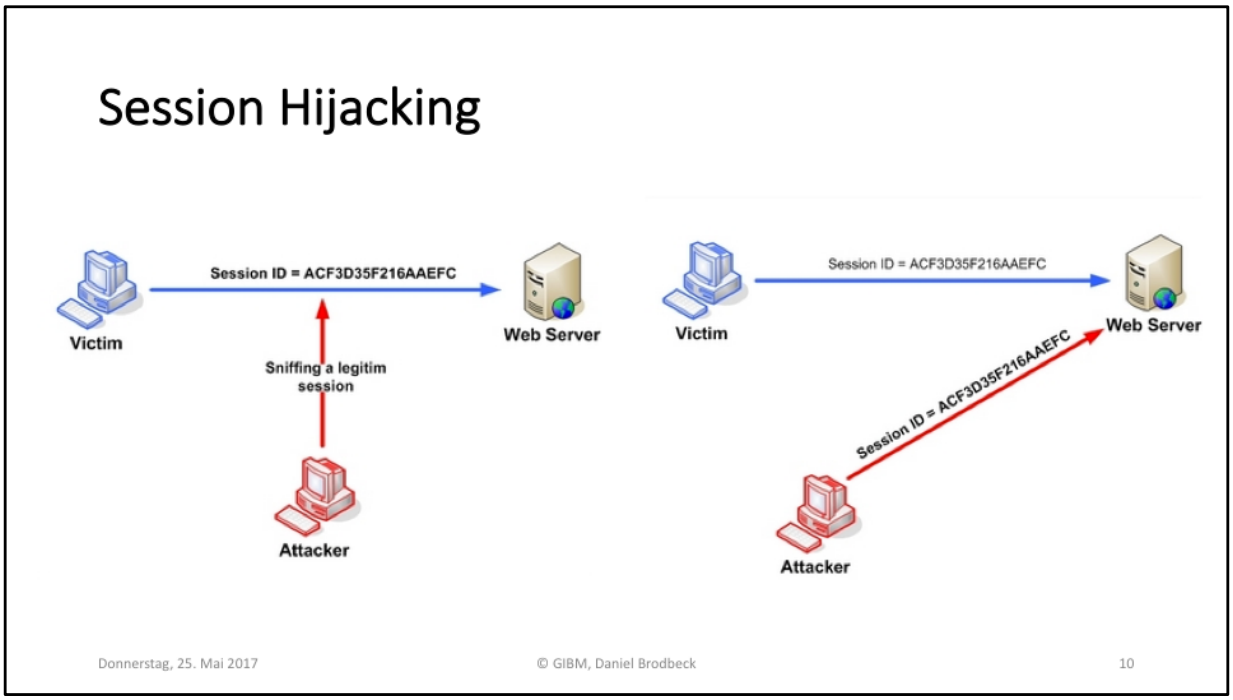
- Prüfen, ob die Session-Variable besteht:

```
if(isset($_SESSION[‚loggedin‘])){
     echo „Hallo “ . $_SESSION[‚username‘];
}
```

Session löschen

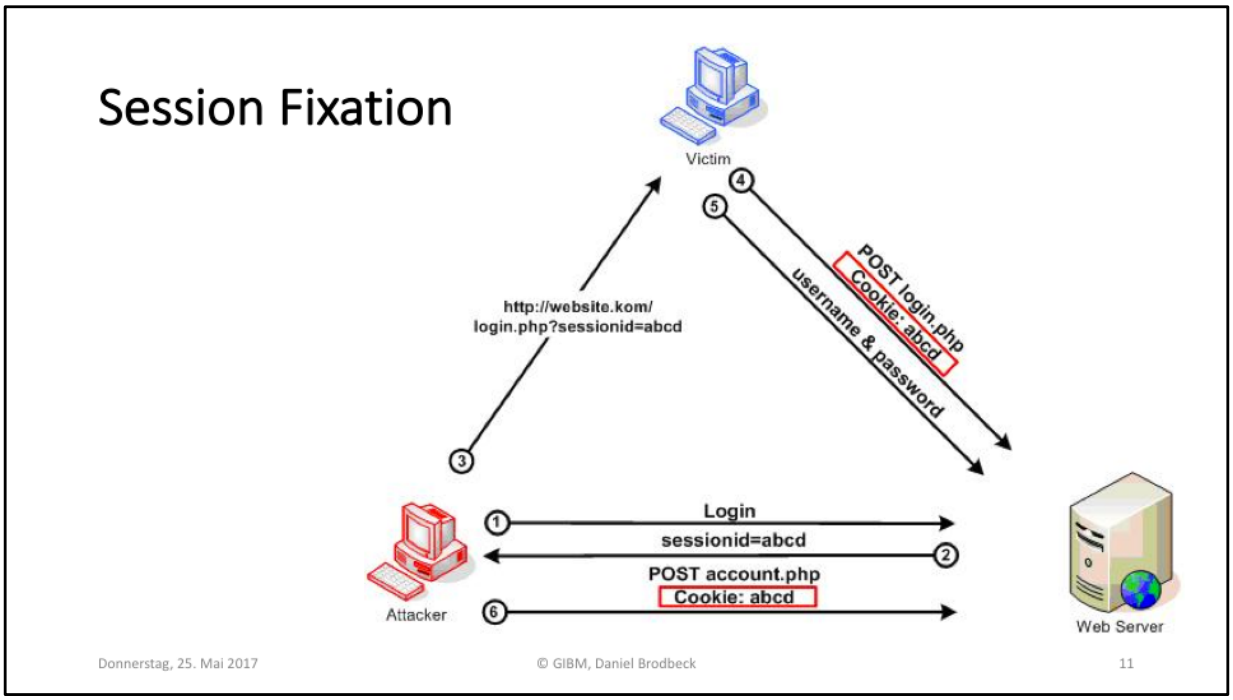
- Assoziatives Array leeren:
  `$_SESSION = array();`
  Die Session besteht weiterhin, lediglich die gespeicherten Informationen sind gelöscht.

- Session löschen:
  `session_destroy();`
  Nun besteht keine Verbindung mehr zwischen Client und Server.



Beschreibt den Angriff auf eine bestehende Sitzung.
Der Angreifer liest die Session-ID einer bestehende Verbindung aus. Er hat nun Zugang zu allen Informationen des ursprünglich angemeldeten Benutzers.

Massnahmen:
PHP: `session_regenerate_id();` nach jedem `session_start();`
Ersetzt die Session-ID durch eine neu erzeugte
HTTPS-Verschlüsselung erschwert das Auslesen der Session-ID



Beschreibt einen Angriff, indem ein Angreifer eine Session erstellt und diese einem anderen Benutzer zur Verwendung unterschiebt. Meldet sich der Benutzer mit dieser Session-ID am System an, hat auch der Angreifer Zugriff au alle Daten der Session.

Massnahmen:

Session-ID darf von der Applikation nicht über die URL angenommen werden. (siehe wichtige Parameter)

Session-Cookie darf nicht mit Javascript ausgelesen werden.
(siehe wichtige Parameter)

PHP: `session_regenerate_id();` nach jedem `session_start();`
Ersetzt die Session-ID durch eine neu erzeugte

Wichtige Parameter

Die folgenden Parameter können in der PHP.ini angepasst werden.
Zudem können sie mit `ini_get(,[Parameter]‘);` ausgelesen und mit `ini_set(,[Parameter]‘, [Wert]);` innerhalb eines Scripts gesetzt werden.
Diese Parameter erschweren das Session-Hijacking sowie die Session-Fixation.

session.use_cookies
Dürfen Session-Cookies angelegt werden (0, 1) (Default 1)
session.use_only_cookies
Schränkt die Session-Verwaltung auf Cookies ein (0,1) (Default 1)
session.use_trans_sid
Die Session-ID wir jedem Link angehängt ?PHPSESSID (0,1) (Default 0)
session.cookie_httponly
Ob der Session-Cookie NICHT mit Javascript (document.cookie) ausgelesen werden darf (0,1) (Default 1)