Сессии

Лекция 6

Определение сессии

- Web-сессия – специальный ключ (идентификатор сессии), хранящийся клиентской программой и сопоставляемый с пользовательскими данными на стороне сервера.
- Сессии предназначены для хранения и передачи данных отдельного пользователя между динамическими страницами одного ресурса.

Причины появления сессий

- Web-сервер каждый раз при обращении к очередной странице инициализирует новую HTTP-транзакцию без возможности связывания старых данных пользователя с вызовом новой динамической страницы.

Способы хранения ключа сессии на стороне клиента

Ключ (идентификатор сессии) на стороне клиента может храниться двумя способами:

- В HTTP-Cookie (относительно безопасно)
- Как часть URL (небезопасно)

Хранение сессии на стороне сервера

- Способы хранения данных внутри сессии никак не регламентируются и могут быть представлены различными способами: в структурных файлах сервера, в БД различного типа и т.д.
- Данные сессии хранятся на сервере в виде массива.

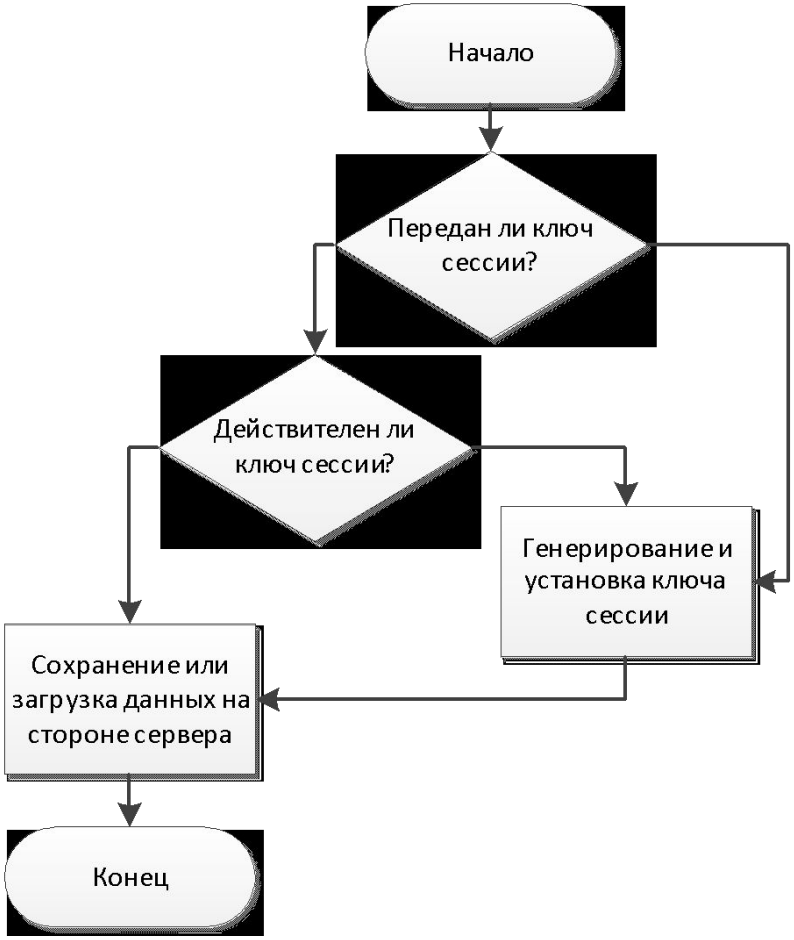
Использование сессий

Сессии могут использоваться для тех же целей, что и HTTP Cookie, но с сохранением данных на стороне сервера.

С помощью сессий также реализуются следующий функционал:

- Подтверждение авторизации пользователя на сайте
- Хранение временных данных, вводимых из нескольких форм
- Корзина в интернет-магазине
- Хранение символьного значения картинки из CAPTCHA

Механизм работы сессий



- Ключ считывается с сервера либо из строки URL, либо из Cookie
- На этапе проверки ключа проверяется, существуют ли данные на сервере, сопоставляемые с этим ключом.
- Ключ генерируются таким образом, чтобы исключить возможность его подмены.

Сессии в PHP

- Ключ PHP-сессии, сохраняемый в Cookie, именуется по-умолчанию PHPSESSID.
- Значения сессии хранятся в суперглобальном массиве $_SESSION

Сохранение данных сессии

```
session_start(); /* инициализация механизма работы сессии */
$_SESSION["user"] = "pavel"; /* Сохранение переменной сессии "user" со значением "pavel" */
if(isset($_SESSION["user"]) && !empty($_SESSION["user"]))
{
    print "Пользователь авторизован";
}
else
{
    print "Требуется авторизация";
}
```

Удаление данных сессии

- Для того, чтобы удалить переменную сессии, достаточно опустошить индекс ассоциативного суперглобального массива $_SESSION:

```
unset($_SESSION["items"]);
// Пользователь удалил все товары из корзины покупок
```

Удаление всей сессии

- Для того, чтобы удалить полностью текущую сессию пользователя, можно вызвать функцию session_destroy():

```
session_destroy();
// Пользователь нажал кнопку "Выход", а значит более он не авторизован
```

Функции работы с идентификатором и его именем

- Функция, возвращающая текущий идентификатор сессии:

```
string session_id ([ string $id ] )
```

- Функция, возвращающая или устанавливающая (если указана переменная $name) имя переменной Cookie для идентификатора сессии:

```
string session_name ([ string $name ] )
```

Безопасность сессии

Узнав идентификатор сессии, злоумышленник может получить доступ к учётной записи другого пользователя. Способы получения идентификатора:

- Перехватка GET-запроса через HTTP-заголовок **Http-Referer** на сайте, на который осуществлён переход (в том случае, если сессия хранится как часть URL)
- Перехватка Cookies путём внедрения JavaScript-кода на одну из страниц сайта через **XSS** уязвимость ресурса или дополнительные привилегии администратора
- Прослушивание HTTP-трафика

Безопасность сессии

Существуют следующие методы обеспечения безопасности сессии:

- Запрет на хранение идентификатора сессии в части URL (только в Cookies)
- Экранирование символов, способных внедрить HTML-код на страницу (можно использовать функцию **htmlspecialchars()**)
- Использование протокола **SSL** над HTTP

Другие технологии защиты сессии

- Ограничение срока действия сессии
- Привязка к IP-адресу пользователя, его браузеру и другим характеристикам

Функция md5()

```
string md5 ( string $str  )
```

- Вычисляет MD5-хэш строки str используя алгоритм MD5 RSA Data Security, Inc. и возвращает этот хэш.

```
<?php
$str = 'яблоко';

if (md5($str) === '1afa148eb41f2e7103f21410bf48346c') {
    echo "Вам зеленое или красное яблоко?";
}
?>
```

Лабораторная работа

- Создать скрипт авторизации пользователя на сайте при помощи сессий
- Создать форму авторизации (логин и пароль)
- Реализовать алгоритм авторизации, сопоставляя данные с теми, которые установлены внутри файла скрипта.
- Пароль хранить в переменной, предварительно зашифровав значение при помощи функции `md5()`.