

# Web Security

## Assignment 2

Christoph Schmidl  
s4226887  
Data Science  
c.schmidl@student.ru.nl

April 18, 2017

1. Do the following two lessons in WebGoat (you do not have to submit any notes regarding these exercises to blackboard):

- Session Management Flaws - Hijack a Session

The screenshot shows the 'Hijack a Session' lesson page in WebGoat. At the top, there is a navigation bar with buttons: 'Show Source', 'Show Solution', 'Show Plan', 'Show Hints', and 'Restart Lesson'. Below this, a red banner reads 'Congratulations. You have successfully completed this lesson.' followed by a paragraph explaining session security. The 'General Goal(s):' section states: 'Try to access an authenticated session belonging to someone else.' The main content area is titled 'Sign In' and contains a form with the text 'Please sign in to your account.' and '\*Required Fields'. The form has two input fields: '\*User Name:' and '\*Password:', each with a text box. A 'Login' button is at the bottom of the form.

Session hijacking, sometimes also known as cookie hijacking is the exploitation of a valid computer session - sometimes also called a session key - to gain unauthorized access to information or services in a computer system. In particular, it is used to refer to the theft of a magic cookie used to authenticate a user to a remote server. In this particular lesson the value of the cookie named "WEAKID" can be bruteforced by using a tool like JHijack. To find a value to start with you can use the SessionID Analysis Tab of WebScarab. You do not know the identity of the user's session you are going to hijack.

- Session Management Flaws - Session Fixation

## Session Fixation

[Show Source](#)
[Show Solution](#)
[Show Plan](#)
[Show Hints](#)
[Restart Lesson](#)

**Congratulations. You have successfully completed this lesson.**

STAGE 1: You are Hacker Joe and you want to steal the session from Jane. Send a prepared email to the victim which looks like an official email from the bank. A template message is prepared below, you will need to add a Session ID (SID) in the link inside the email. Alter the link to include a SID.

**You are: Hacker Joe**

**Mail To:** jane.plane@owasp.org

**Mail From:** admin@webgoatfinancial.com

**Title:**

<b>Dear MS. Plane</b> <br><br>During the last week we had a few problems with our database. We have received many complaints regarding incorrect account details. Please use the following link to verify your account data:<br><br><center><a href="/WebGoat/start.mvc#attack/2007866518/1800"> Goat Hills Financial</a></center><br><br>We are sorry for the any inconvenience and thank you for your cooperation.<br><br><b>Your Goat Hills Financial Team</b></center> <br><br></center>

2. Inspect 5 websites for which you got login credentials

- Do these website use HSTS (HTTP Strict Transport Security)?
- Does the session ID cookie of these websites use *Secure* and/or *HttpOnly*?

Make a table of these websites and their support for HSTS, the name of the cookie and the usage of *Secure* and *HttpOnly* for that cookie.

**Solution:**

Website	HSTS	Cookie Name	Secure	HttpOnly
facebook.com	yes	multiple cookies (c_user, xs, sb)	yes, yes, yes	no, yes, yes
mail.ru.nl	no	cadata	yes	yes
linkedin.com	no	li_at	yes	yes
github.com	yes	_gh_sess	yes	yes
gmx.net	yes	905a7a5991a00898953878290a55d118	yes	yes

Table 1: Website Information