DFTI Worksheet 6
Introduction to web security

Part I - Cross-site scripting

This section allows you to explore the security risks of cross-site scripting, in which crackers can link to your site and inject HTML or JavaScript into your site to force your site to perform an unintended action - and SQL injection.

This tutorial will allow you to perform a cross-site scripting attack on HitTastic!, and then modify the script to prevent the attack happening.

Please follow these instructions EXACTLY. We will go through it together in the class.

Questions

1. Load up FIREFOX (or if not available, EDGE). For security reasons, ensure that you do not have any other websites showing on FIREFOX/EDGE, as otherwise your session ID from these sites will be stolen and logged on Edward2.
   In FIREFOX/EDGE, login to the pre-prepared HitTastic! site on Neptune, at

   http://ephp.solent.ac.uk/~ephp001/

   Login to HitTastic! with the user whose username we give you.

   See http://ephp.solent.ac.uk/dfti/users.php for the list of users and passwords. For example, if your user is TimWilson, you would use TimWilson/egg882.

   Now, on another tab on FIREFOX/EDGE, access this "phishing" page.
   https://edward2.solent.ac.uk/dfti/phish.html

   Do not do anything with this page until instructed.
   When instructed, choose one of the songs to download (Woop, Madonna, Michael Jackson or Oasis). What happens?

2. Although you appear to have downloaded the song, what has happened is that a cross-site scripting attack has taken place, and your session ID has been sent to Edward2 and logged in a database. You will see this as the list

of stolen session IDs will mount up on the presentation on the overhead projector.

At this point, your tutor will explain how the cross-site scripting attack has taken place.

3. Now load CHROME. (The idea is that sessions are specific to a particular browser, so by loading CHROME, you are simulating another website user). Try to access HitTastic! using the URL above. You will find you can't get in, as your session is specific to Firefox.
4. You can edit cookies in the browser, but plugins are needed for this. What we will do instead is use a simulation of editing the cookies file. Go to this URL (in CHROME)
   http://ephp.solent.ac.uk/~ephp001/setsession.php

   and choose one of the session IDs shown at

   https://edward2.solent.ac.uk/dfti/stolencookies.html

   What happens?

5. Now buy music at the legitimate user's expense !

6. We will then demo how to prevent the attack taking place by modifying the code. Once we have edited the code live to prevent the attack, log out of the site (both on FIREFOX/EDGE and CHROME), and try doing the attack again by logging back in on FIREFOX/EDGE and following the link on the "phishing" page again. Does it work now?

Part II - SQL Injection

This section allows you to explore the security risk of SQL injection, in which crackers can link to your site and inject SQL into your site to force your site to perform an unintended action.

Questions

a) Ensure that your session variable in your login script is set this way:

```
$_SESSION["gatekeeper"] = $row["username"]
```

and not this way

```
$_SESSION["gatekeeper"] = $un;
```

or

```
$_SESSION["gatekeeper"] = $_POST["username"]
```

In other words, it should set the session variable to the username field of the row returned from the SELECT statement in the login script.

b) Go to your login page on your HitTastic! site. Login with the following:

```
Username: jbloggs
Password: password' OR '1=1
```

What happens? Can you see why? We will explain what is happening after everyone has got this far.

c) Modify the login script to use prepared statements to prevent the SQL injection occurring. Try out the SQL injection attack again.

d) Return to your HitTastic! work and change all your scripts which use database queries to use prepared statements.

If you finish this, then try out the earlier extra topic: Worksheet 4a - functions and associative arrays (under additional material - see the website). This topic will be very useful to you if you wish to become a PHP developer after university.