Project name: Simple Moodle Improved

Description:
The site allows registered users to access content from a variety of courses. Users can register and access the content and the forum to ask user questions. A teacher account is used to enter new content within a course, while an administrator account is used to enter new courses into the database. The attacker's goal is to create a new course and find a way to insert an image called solution.jpg (residing in the images folder) within the created course. The attack should be divided into two phases: first, one must try to steal the php session from the administrator. Then he must exploit a vulnerability in order to obtain the username and password of the teacher account. At this point he can log in and find a way to load image in the course content. The image is considered to be not accessible directly from the browser by requesting its corresponding URL. The attacker should not try to enumerate all the possible password from the database, and login/signup form is considered to be injection-proof.

Vulnerabilities description:
There are three intended vulnerabilities. The first one is related to the forum functionality. Since any content is allowed, a malicious XSScript can be inserted to read and save an administrator's session cookie. Since the cookies are not saved as HTTP only, they are accessible via javascript. An external website can be used to register the session of each user, and the attacker can steal every session. Here the user input is considered to be always safa, and sanitization is not applied, thus violating the core defense idea to never thrust the user's input.
The second vulnerability is similar, but this time takes place in form of an SQL injection vulnerability. When the attacker is logged as administrator, he can use the find user functionality to get all the usernames and passwords in the website. Here the input is not filtered since the functionality is considered to be accessible only for administrators.
The last vulnerability resides in the course page. A logged teacher can add anything to the course, which includes an html <img> tag to show the target image. Here the input is again not filtered, but also the direct object reference is insecure, since access to the resource is not verified, and the attacker can request any resource in the website.

Authors:
Tommaso Bertoldi, Simone Peraro.