observations report

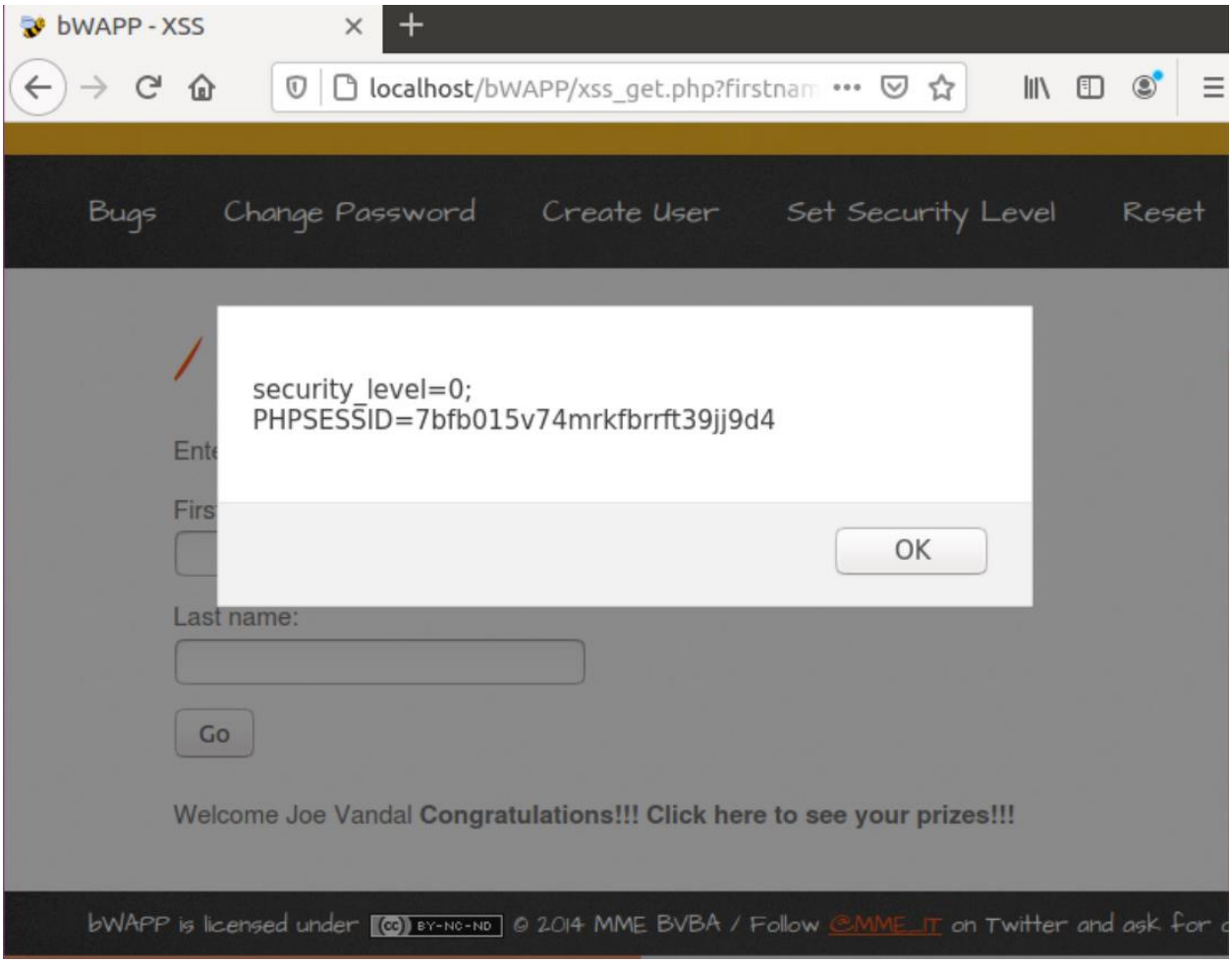
Name: Jihui.Sheng

Course: Cpt_S 427

ID: 11539324

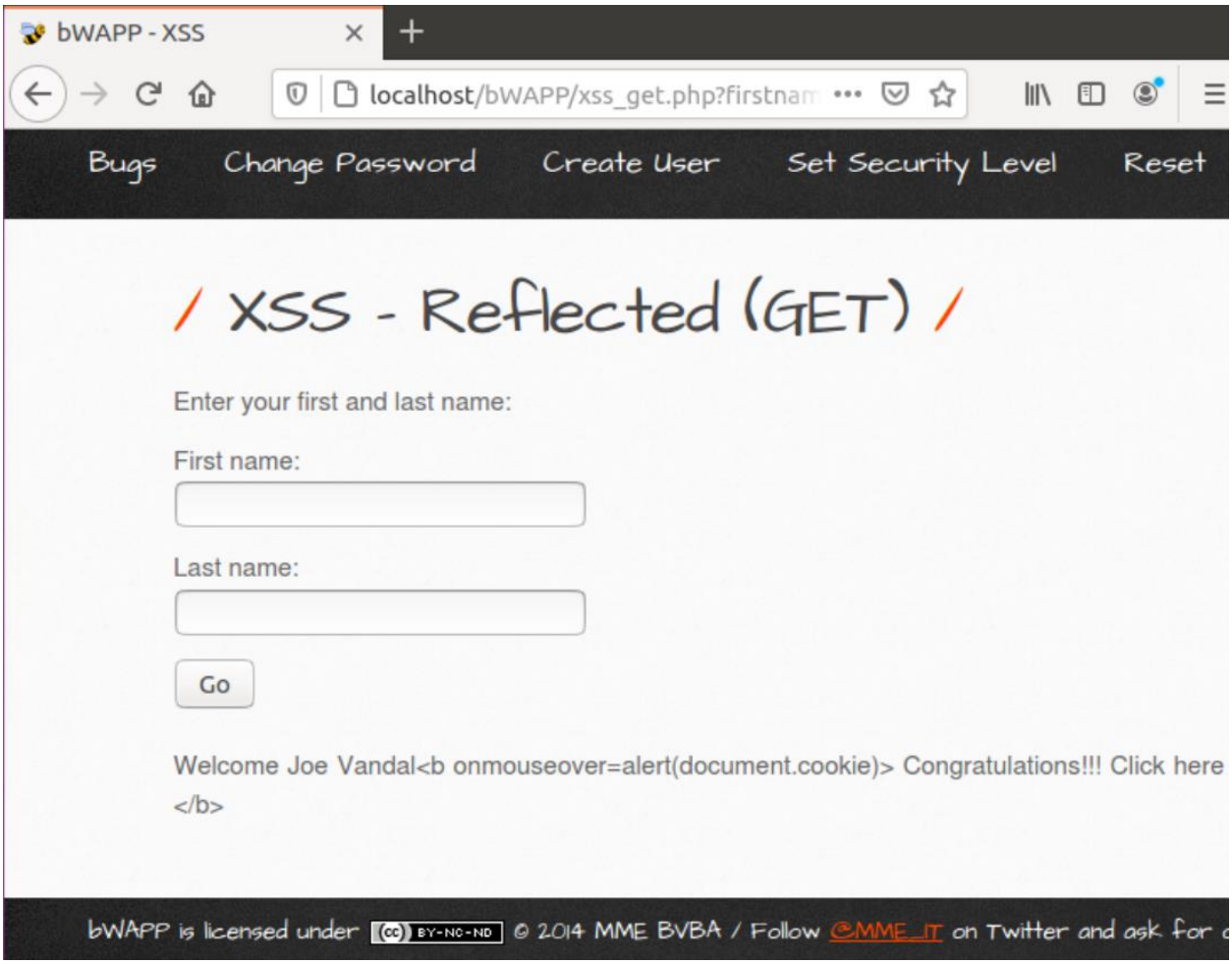
Screenshots for the tutorial activities:

Challenge I: Modified XSS Attack:



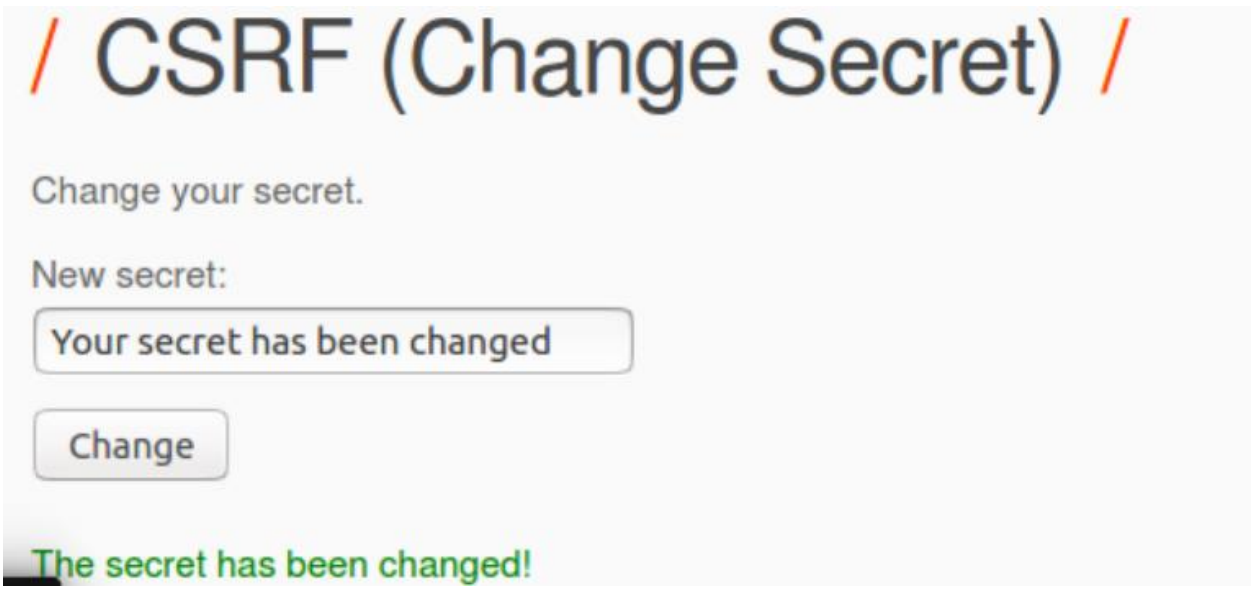
Challenge II: Mitigate Challenge I Attack:

```
function xss($data)
{

    switch($_COOKIE["security_level"])
    {

        case "0" :

            $data = htmlspecialchars($data,ENT_QUOTES,"UTF-8");
            break;
```



Challenge III: Modified CSRF Attack:

```
<div id="main"> overflow
  <h1> ... </h1>
   <p>Change your secret.</p>
  <form action="http://www.bwapp.com/csrf_3.php" method="POST">
   <p>
      <label for="secret">New secret:</label>
      <br>
      <input id="secret" type="text" name="secret" value="Your
      secret has been changed">
    </p>
    <input type="hidden" name="login" value="bee">
    <button type="submit" name="action" value="change">Change
    </button>
  </form>
  <br>
  <font color="green">The secret has been changed!</font>
</div>
```



Remarks: I do not know how to do this step, I tried to change the code and save it. The page will display the value of value in the input box. Does this mean that the attacker will forge the same web page to obtain the user's account password?

Challenge IV: Mitigate Challenge III Attack：

```
else
{
    // If the security level is not MEDIUM or HIGH
    if($_COOKIE["security_level"] != "1" && $_COOKIE["security_level"] != "2")
    {
        if(isset($_REQUEST["token"]) and isset($_REQUEST["token"]) and $_REQUEST["token"] == $_SESSION["token"$
        {
            $login = $_REQUEST["login"];
            $login = mysqli_real_escape_string($link, $login);
```

```
    else
    {
        $message = "<font color=\"red\">Invalid token!</font>";
    }

lse

    // If the security level is MEDIUM or HIGH
```

```
if($_COOKIE["security_level"] != "1" && $_COOKIE["security_level"] != "2")
{
?>
        <input type="hidden" name="login" value="<?php echo $login;?>">
        <input type="hidden" name="token" id="token" value="<?php echo $_SESSION['token']?>">
<?php
}
```

Report any bugs, typos, broken links etc:

I want to talk about the problems that I encountered during the implementation of the tutorial:

1. The other Ubantu version is not work, my last try is 20.?? version, It not ask you to set the password for root.
2. the acquisition of the IP address:
   For example: If I got IP form curl is 134.49.258.160. Then I can't use http://134.49.258.160 to get Apache2 Ubantu Default Page. It actually need use http://localhost.
3. The install helper file is useful. But not specific enough. At first time I didn't realize that the root of the web server was under the var/www/html file. Maybe it can be more detailed, similar to telling us to use sudo cp -r bWAPP /var/www/html.
4. I encountered a connection error when downloading using install.php. It displays Connection failed: "Access denied for user'root'@'localhost'. Finally, I download by changing the root account and password, but it is different from the tutorial.

Skills that I learned from the tutorial:

1. XSS attack refers to an attack in which an attacker tampered with a web page through "HTML injection" and inserted a malicious script to control the user's browser when the user browsed the web page.
2. A CSRF attack involves an attacker forging a user's identity and deceiving the server, causing user information leakage and property damage.
3. If the content entered by the user can be echoed to the website, it indicates that there is an XSS vulnerability, and an attacker can exploit it through script injection.
4. The password change page can use identity verification to prevent CSRF attacks.
5. The solutions of XXS and CSRF include: correct input verification, special character encoding, and correct user and session authentication.
6. The reason XSS and CSRF are still a problem to this day is that developers lack the expertise to implement secure web applications. They did not make safety a high priority for product delivery.
7.