Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29
Questions: XSS

Q1. What is the most basic indication that a cross-site scripting attack might be possible on a website?

The most basic indication that a cross-site scripting (XSS) attack might be possible on a website is when user inputs are not properly sanitized or validated before they are displayed on the website. This means that if a website takes user-generated content and directly inserts it into web pages without proper filtering or encoding, it can create an opening for an XSS attack.

Q2. What is one example of language that can be leveraged in an XSS attack?

One example of a language that can be leveraged in an XSS attack is JavaScript. In an XSS attack, malicious JavaScript code is injected into a web page and then executed in the context of victim's browser. This can allow an attacker to steal user data, manipulate the appearance of the website, or perform other malicious actions on behalf of the user.

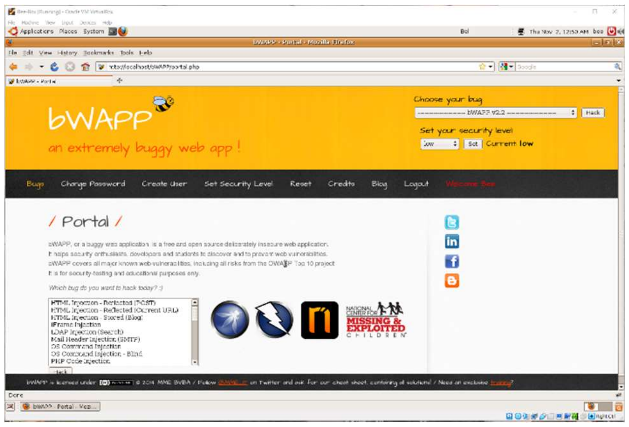
Q3. If the languages can be leveraged in an XSS attack, what shall we do then? Never use such languages?

Its not necessary to completely avoid using languages like JavaScript or other technologies that can be leveraged in XSS attacks. Instead, you should follow best practices to protect you web applications from XSS vulnerabilities. Here are some key measures to take:

a. Input Validation: Always validate and sanitize user inputs to ensure that they do not contain malicious code.
b. Output Encoding: Encode or escape user-generated content before it is displayed on web pages. This ensures that even if an attacker injects malicious code, it will be treated as data rather than exe code.
c. Content Security Policy (CSP): Implement a content security Policy to control which scripts can be executed on your website. CSP headers can help prevent the execution of unauthorized scripts.
d. Regular Security Audits: Periodically perform security audits and vulnerability assessments on your website to identify and address any potential XSS vulnerabilities.

Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29
Activity 1: XSS Attack

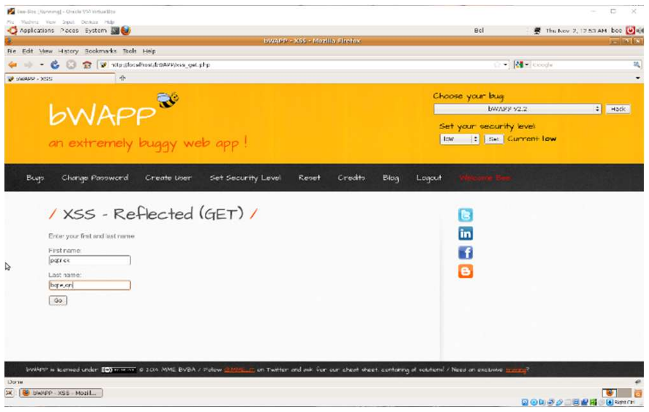
1. Go to http://www.bwapp.com/login.php
2. Login: bee password: bug
3. From "choose your bug." Drop-down list, select "Cross-Site Scripting – Reflected (GET)"
4. Get security level to low and click Hack
5. Provide different combinations of first name and last name and look for the potential vulnerability.


http://localhost/bWAPP/portal.php

/ Portal /

bWAPP, or a buggy web application, is a free and open source deliberately insecure web application.
It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.
bWAPP covers all major known web vulnerabilities, including all risks from the OWASP Top 10 project.
It is for security-testing and educational purposes only.

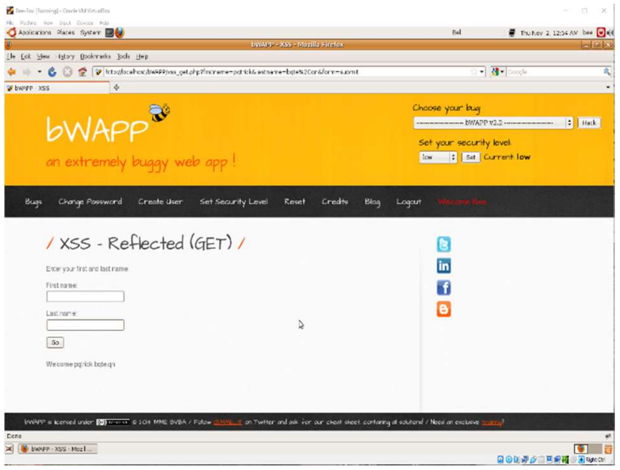
Which bug do you want to hack today? :)


/ XSS - Reflected (GET) /

Enter your first and last name:

First name: pqtrick

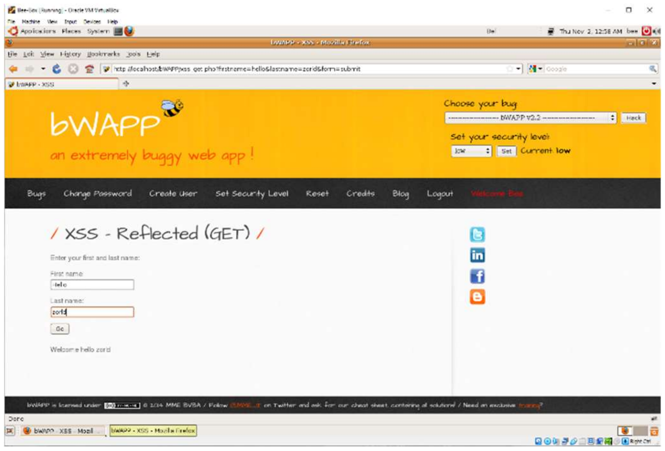
Last name: bqtean


http://localhost/bWAPP/xss_get.php?firstname=pqtrick&lastname=bqte%2Cqn&form=submit

/ XSS - Reflected (GET) /

Enter your first and last name:

Welcome pqtrick bqte,qn


http://localhost/bWAPP/xss_get.php?firstname=hello&lastname=zorld&form=submit

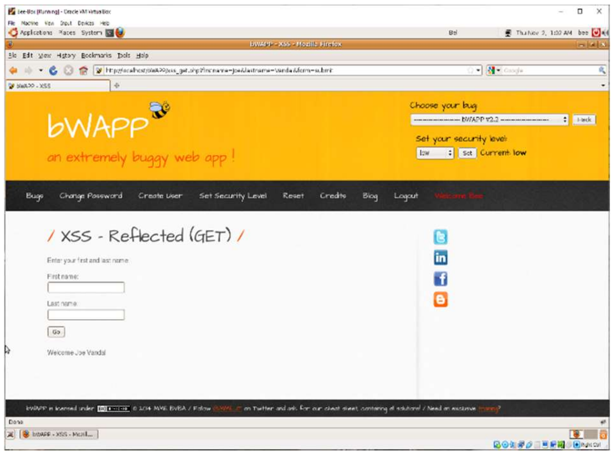
/ XSS - Reflected (GET) /

Enter your first and last name:

First name: hello

Last name: zorld

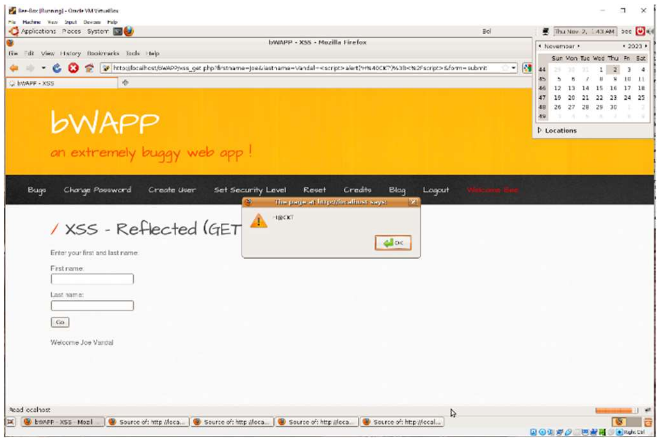
Welcome hello zorld


http://localhost/bWAPP/xss_get.php?firstname=Joe&lastname=Vandal&form=submit

/ XSS - Reflected (GET) /

Enter your first and last name:

Welcome Joe Vandal


http://localhost/bWAPP/xss_get.php?firstname=Joe&lastname=Vandal<script>alert('H%40CK')%3B<%2Fscript>&form=submit

The page at http://localhost says:

H@CK

/ XSS - Reflected (GET

Enter your first and last name:

Welcome Joe Vandal

Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29

Challenge I: Modified XSS Attack

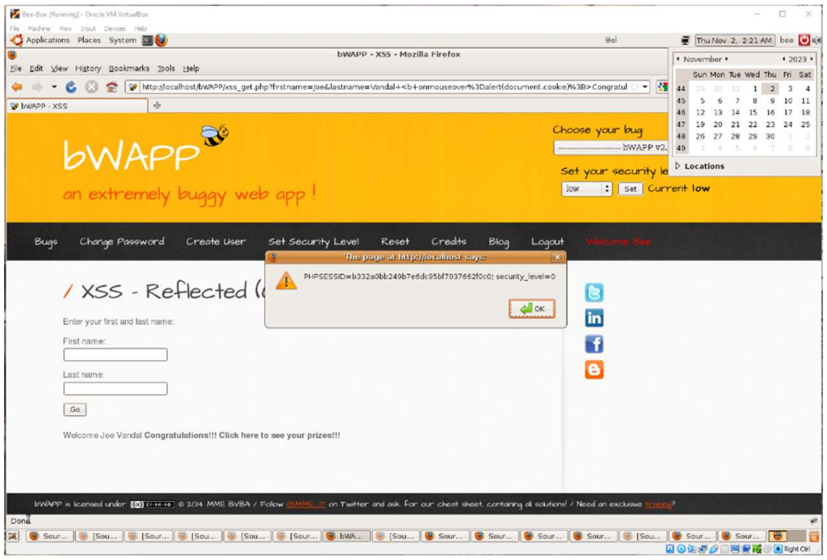
As a Continuation to previous activity, try to inject some input parameters such that

1. Inputs values are returned back to the user itself with the welcome message as follows:

   Welcome message as follows:

   Welcome Joe Vandal Congratulations !!! Click here to see your prizes!!!

2. When user performs mouse-over action on Congratulations!!! Click here to see your prizes!!! , it displays cookie information for the user as an alert.



```
http://localhost/bWAPP/xss_get.php?firstname=Joe&lastname=Vandal+<b+onmouseover%3Dalert(document.cookie)%3B>Congratul
```

The page at http://localhost says:

```
PHPSESSID=b332a0bb249b7e6dc95bf7037662f0c0; security_level=0
```

Set your security level: low — Current: low

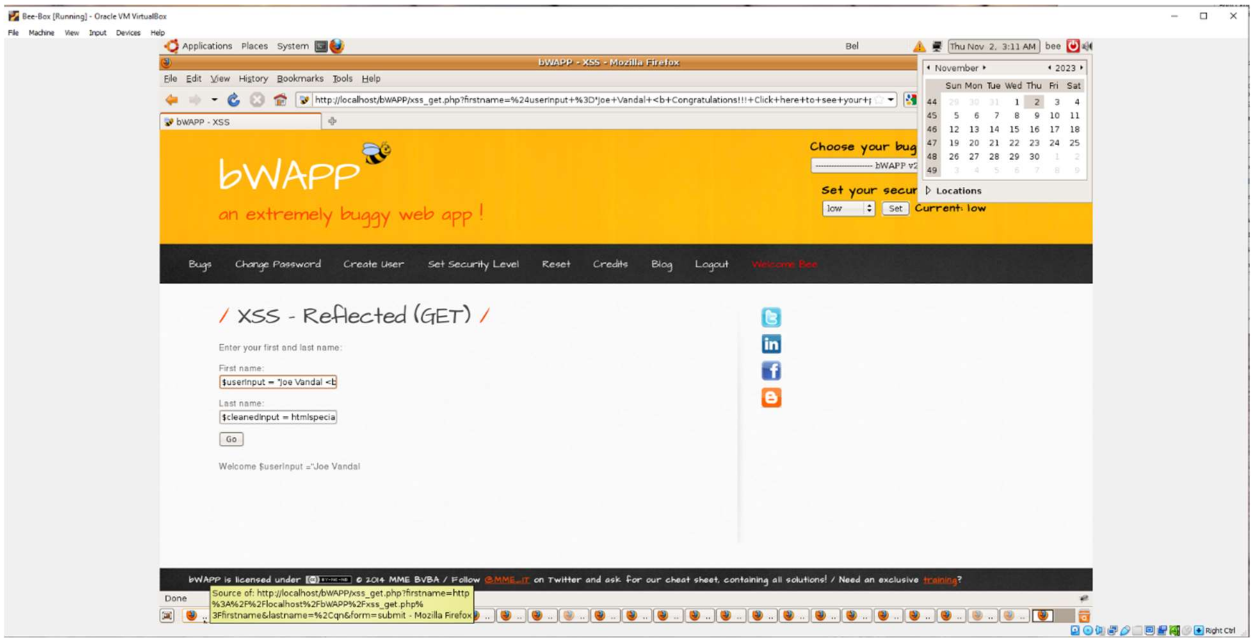
Enter your first and last name:

Welcome Joe Vandal **Congratulations!!! Click here to see your prizes!!!**

Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29
Challenge II: Mitigate Challenge I Attack

Try sanitizing inputs such that the scripts you executed in the challenge 1 acts just as a normal script.

For this challenge, the equivalent php file is located at: /var/www/html/bW APP/public.html/xss_get.php


Bee-Box [Running] - Oracle VM VirtualBox
bWAPP - XSS - Mozilla Firefox
bWAPP
an extremely buggy web app !
Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Bee

http://localhost/bWAPP/xss_get.php?firstname=%24userInput+%3D'Joe+Vandal+<b+Congratulations!!!+Click+here+to+see+your+

Choose your bug: bWAPP v2
Set your security level: low
Current: low

/ XSS - Reflected (GET) /

Enter your first and last name:

First name: $userInput = 'Joe Vandal <b

Last name: $cleanedInput = htmlspecia

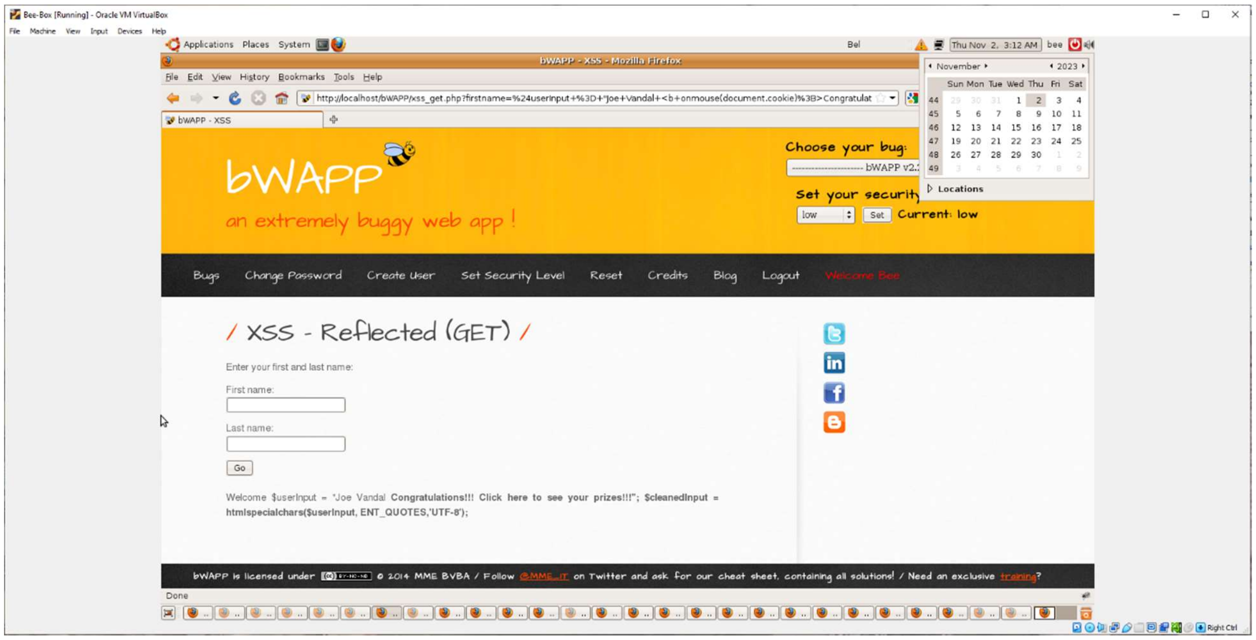
Go

Welcome $userInput ="Joe Vandal

bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?

Done

Source of: http://localhost/bWAPP/xss_get.php?firstname=http%3A%2F%2Flocalhost%2FbWAPP%2Fxss_get.php%3Ffirstname&lastname=%2Cqn&form=submit - Mozilla Firefox


Bee-Box [Running] - Oracle VM VirtualBox
bWAPP - XSS - Mozilla Firefox
bWAPP
an extremely buggy web app !
Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Bee

http://localhost/bWAPP/xss_get.php?firstname=%24userInput+%3D+"Joe+Vandal+<b+onmouse(document.cookie)%3B>Congratulat

Choose your bug: bWAPP v2.
Set your security level: low
Current: low

/ XSS - Reflected (GET) /

Enter your first and last name:

First name:

Last name:

Go

Welcome $userInput = "Joe Vandal **Congratulations!!! Click here to see your prizes!!!"; $cleanedInput = htmlspecialchars($userInput, ENT_QUOTES,'UTF-8');**

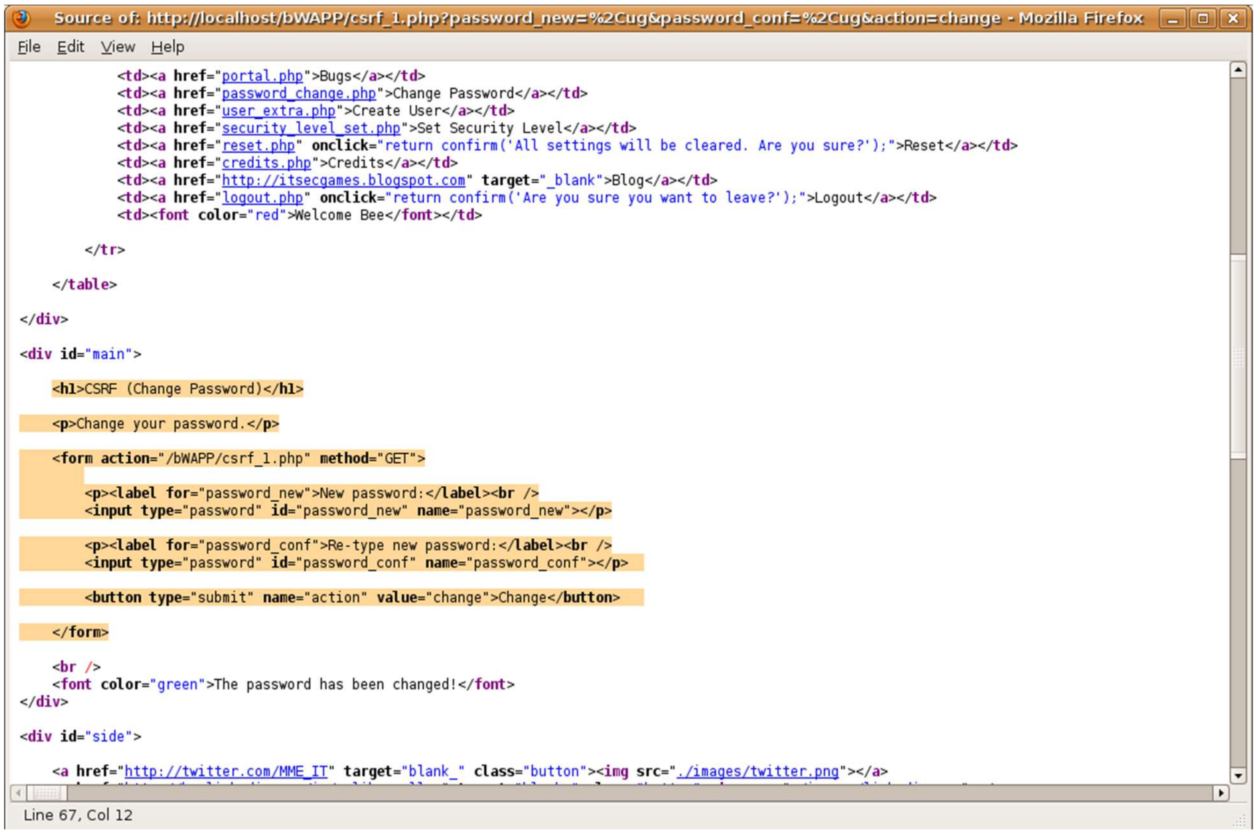
bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?

Done

Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29

Activity II: CSRF Attack Part I

1. From "Choose your bug." Drop-down list, select "Cross-Site Request Forgery (Change Password)"
2. Set security level to low and click Hack
3. Change the password from "bug" to something else. You should be able to do it
4. Right Click on the browser and view the source of the webpage
5. Copy Web content as shown below



Source of: http://localhost/bWAPP/csrf_1.php?password_new=%2Cug&password_conf=%2Cug&action=change - Mozilla Firefox

```
            <td><a href="portal.php">Bugs</a></td>
            <td><a href="password_change.php">Change Password</a></td>
            <td><a href="user_extra.php">Create User</a></td>
            <td><a href="security_level_set.php">Set Security Level</a></td>
            <td><a href="reset.php" onclick="return confirm('All settings will be cleared. Are you sure?');">Reset</a></td>
            <td><a href="credits.php">Credits</a></td>
            <td><a href="http://itsecgames.blogspot.com" target="_blank">Blog</a></td>
            <td><a href="logout.php" onclick="return confirm('Are you sure you want to leave?');">Logout</a></td>
            <td><font color="red">Welcome Bee</font></td>

        </tr>

    </table>

</div>

<div id="main">

    <h1>CSRF (Change Password)</h1>

    <p>Change your password.</p>

    <form action="/bWAPP/csrf_1.php" method="GET">

        <p><label for="password_new">New password:</label><br />
        <input type="password" id="password_new" name="password_new"></p>

        <p><label for="password_conf">Re-type new password:</label><br />
        <input type="password" id="password_conf" name="password_conf"></p>

        <button type="submit" name="action" value="change">Change</button>

    </form>

    <br />
    <font color="green">The password has been changed!</font>
</div>

<div id="side">

    <a href="http://twitter.com/MME_IT" target="blank_" class="button"><img src="./images/twitter.png"></a>
```

Line 67, Col 12

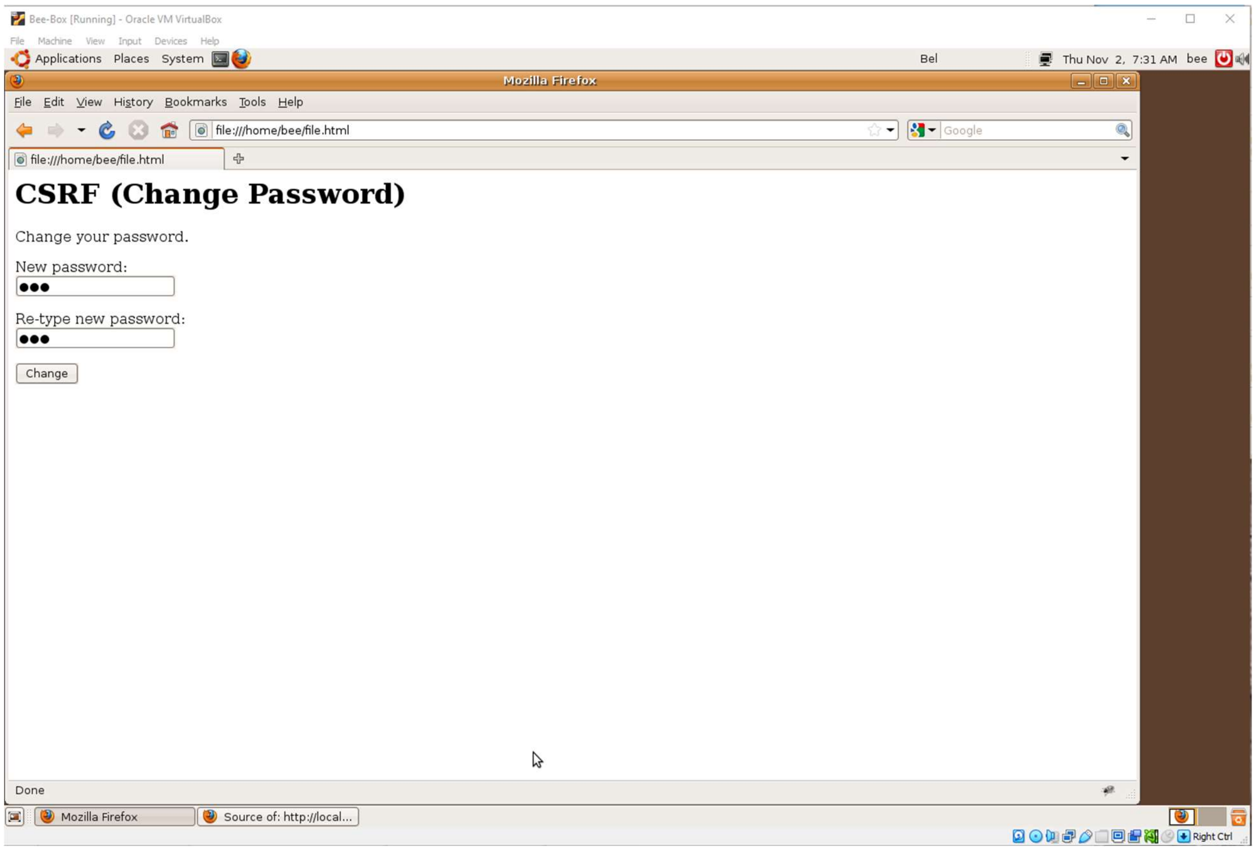
Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29

Activity II: CSRF Attack Part II

1. Modify the copied content as shown in the notes section below. Note the following changes:
   a. Action field in the form tag
   b. Addition of value attributes in the input tags
2. Save it as html file in desktop
3. Open the html file and press Change

Attackers tend to execute such process in the background silently where your password is being compromised secretly.

Mitigating Attack from Activity II


Bee-Box [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Thu Nov 2, 7:31 AM bee
Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///home/bee/file.html
Google
Done
Mozilla Firefox
Source of: http://local...
Right Ctrl

CSRF (Change Password)

Change your password.

New password:
●●●

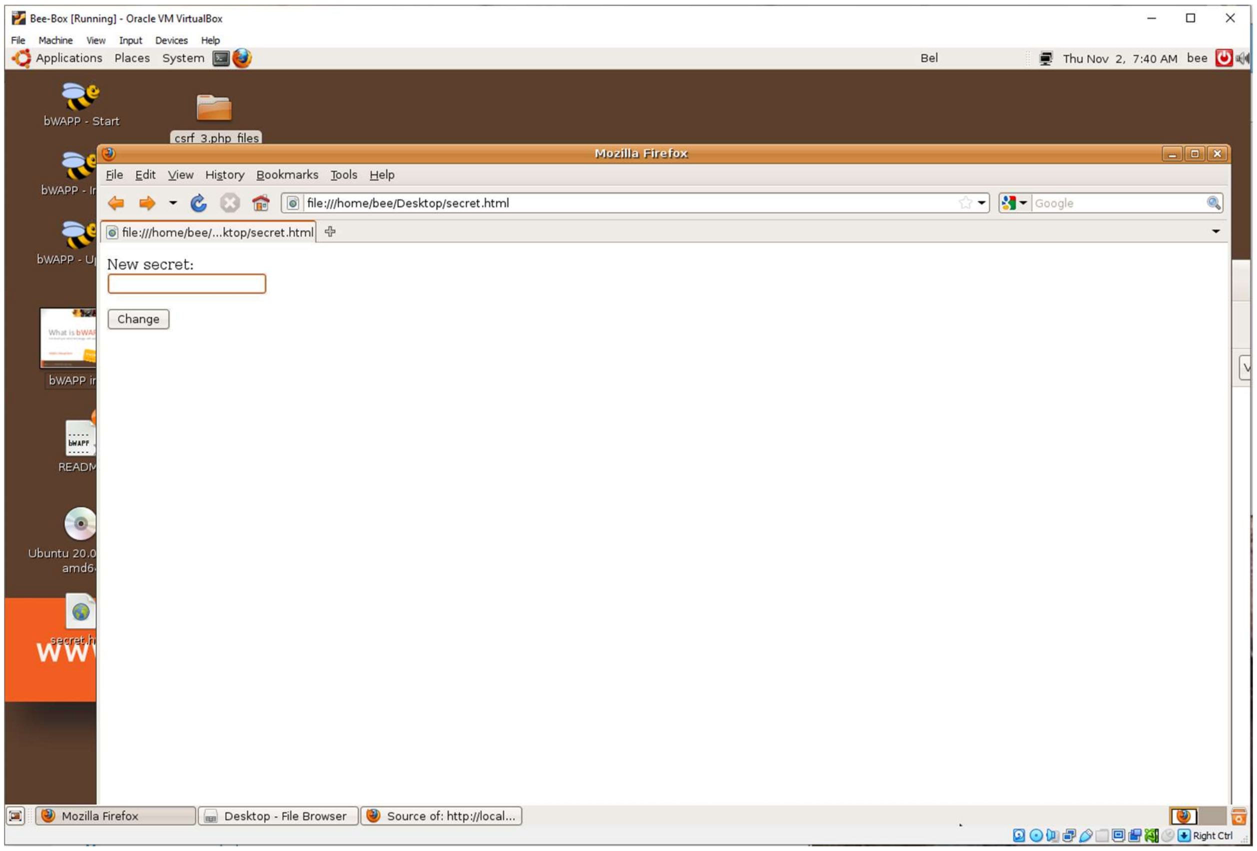
Re-type new password:
●●●

Change

Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29

Challenge III: Modified CSRF Attack

1. From Choose your bug, select Cross-Site Request Forgery(Change Secret)
2. Set security level to low and then press Hack
3. As we did in previous tutorial, create a HTML page that can be injected using CSRF, and execute html page such that it change the secret statement to something else.


Bee-Box [Running] - Oracle VM VirtualBox
Mozilla Firefox
file:///home/bee/Desktop/secret.html
New secret:
Change
csrf_3.php_files
Thu Nov 2, 7:40 AM

Questions: CSRF

What was the problem with the following pages?

1. Page where the password was changed
   a. It allowed for the user to force unwanted HTML file onto the webpage in which we had current authentication. The password was forcefully updated and changed
2. Page where the secret was changed
   a. This created the same situation as the password webpage but instead it prompted for a change in the secret instead of the password element

Can you think about the ways to mitigate the problems in the above pages?

Mark Strong-Shinozaki
HW05 – Observation Report
Professor: Ananth Jillepalli
Date: 10/29

Token Synchronization

- Attackers would not be able to make requests to the backend without valid tokens

Double-Submitting Cookies

- It is a stateless, easily implemented technique that sends random values twice

Same-Site Cookies

- Help defend against CSRF attacks by restricting the cookies sent alongside each request.

Challenge IV: Mitigate Challenge III Attack <>

Use one-time token implementation such that the system uses tokens that change as per the session thus, grabbing tokens from the web-application of the target victim doesn't work

- The page associated with this challenge is csrf_3.php located in /var/www/html/www.bwapp.com/public_html/