Project 4 Report.pdf

gtid drozen3

Target 2: XSS-password theft

10 points: Identify vulnerability and briefly explain why it is vulnerable.

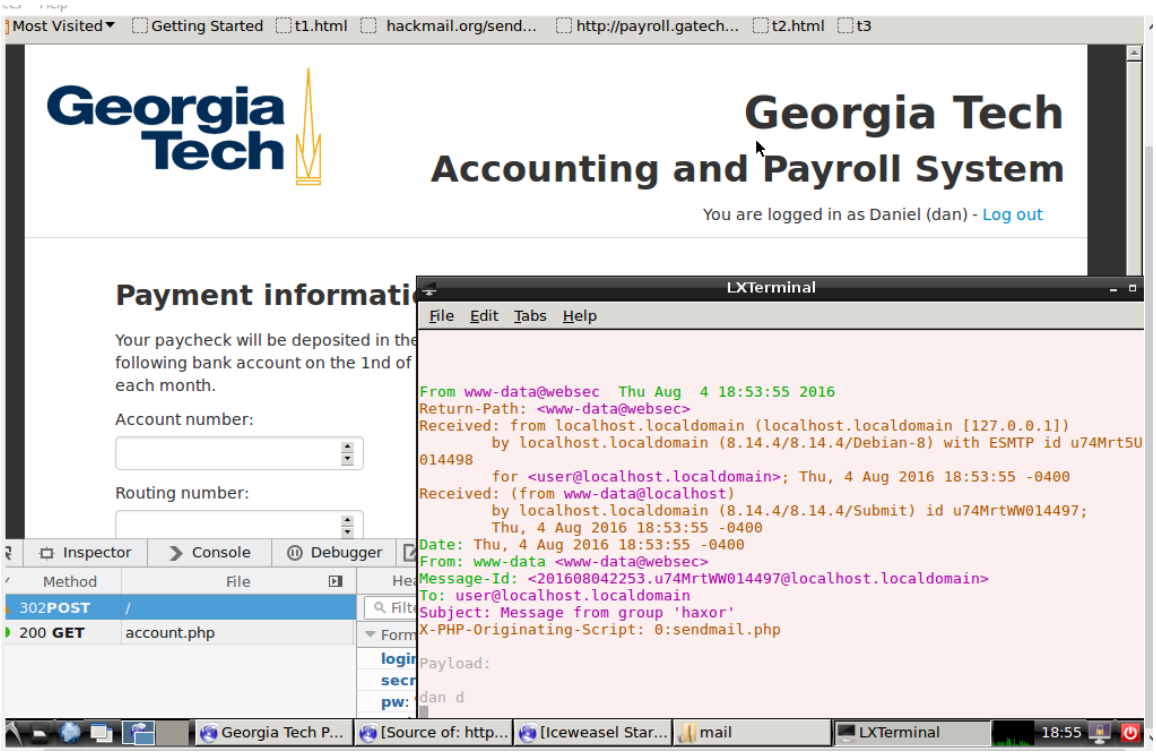
Target XSS

The vulnerable code is in index.php:line 43

because filling the “Name” textbox allows you to register an account with only a name. Now when someone clicks the “Log In” button when account ID and password fields are blank, it will fire the script that was entered into Name textbox.

20 points: Able to steal the username/password and send it via the mail.

my code will pre-enter the username and password as dan/d (please sign up with these credentials first). and then submit the form and then an email with these stole credentials will be sent as shown below. Also the site looks exactly the same as the legitimate site.



10 points: the exploited webpage is cosmetically the same as original site.

Target 3: SQL Injection (30 points)

<u>10 points: Identify vulnerability and briefly explain why it is vulnerable.</u>

Target **SQL Injection**

The vulnerable code is in auth.php:line 52

because In the vulnerable line in auth.php:

```
$sql = "SELECT user_id, name, eid FROM users WHERE eid='$username' AND
password='$hash'";
```

```
If we set $$username = enteredName + "' OR 'x'='x";
```

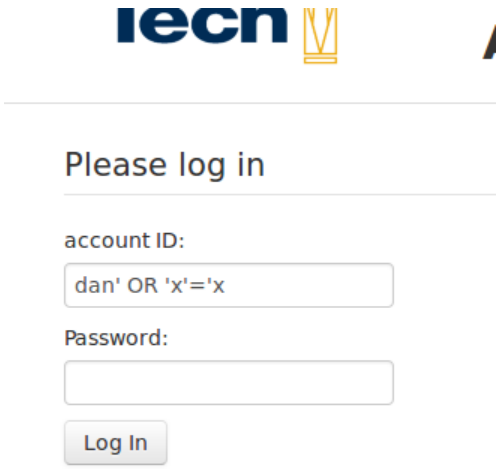
We can make a query such as

```
$sql = "SELECT user_id, name, eid FROM users WHERE eid=' enteredName + "' OR
'x'='x";
' AND password='$hash'";
```

which will result in an injection attack that will bypass the password requirement and always evaluate to true.

This occurs because 'x'='x' always evaluates to true.

eg.



clicking Log In will result in: