Jacob Kilver

CS 6035: Introduction to Information Security

Project 4: Web Security

7 Dec 2016

# Target 1: XSRF

The main vulnerability here is the cross-site request forgery prevention mechanism is weak. First, the mt\_rand() method is used, which is known to not produce cryptographically secure values (<http://php.net/manual/en/function.mt-rand.php>). In fact, if I am reading the code properly, only one “challenge” value is created, so the key is essentially constant. Second, when attempting to create this attack and guess the response, the expected value is actually printed out in the message saying the attack was prevented. This is a very poor security choice.

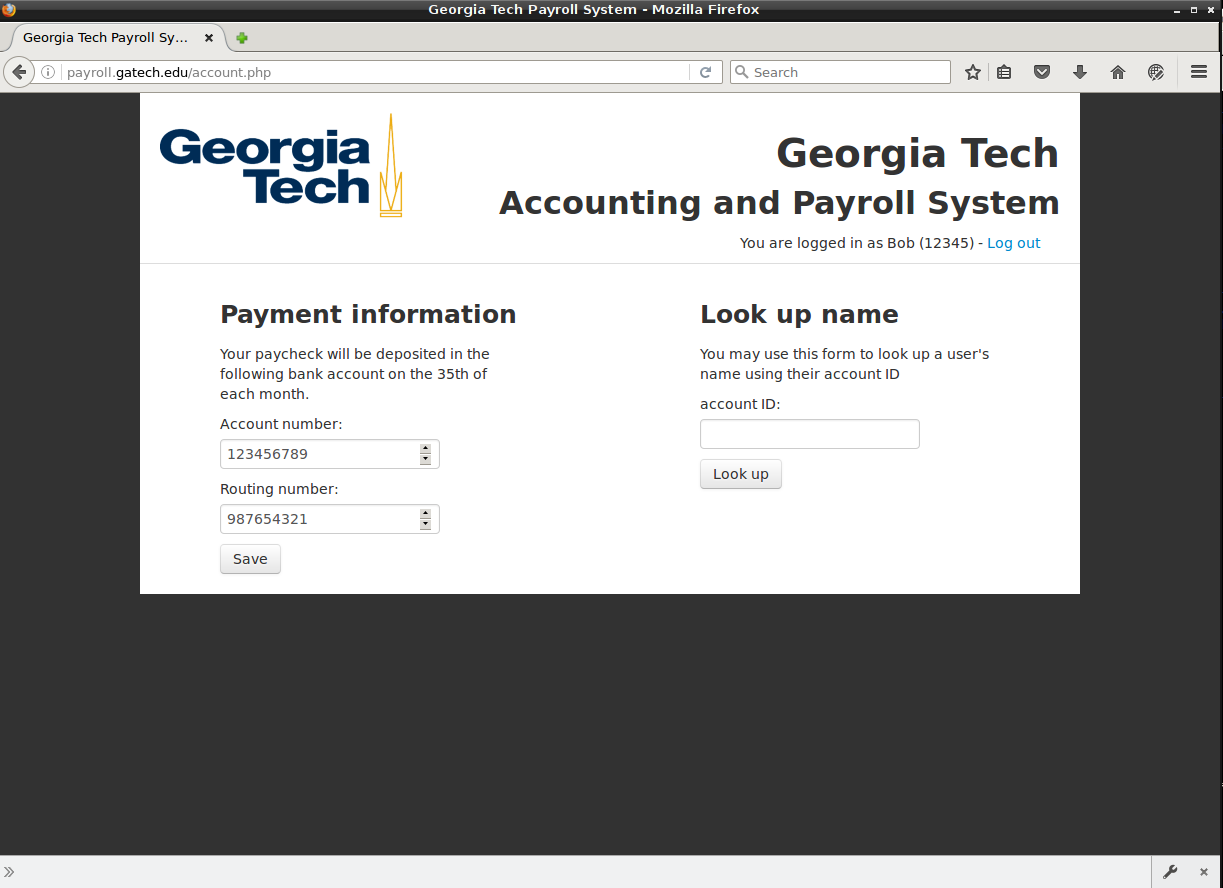


Figure : Screenshot of account page before attack is executed

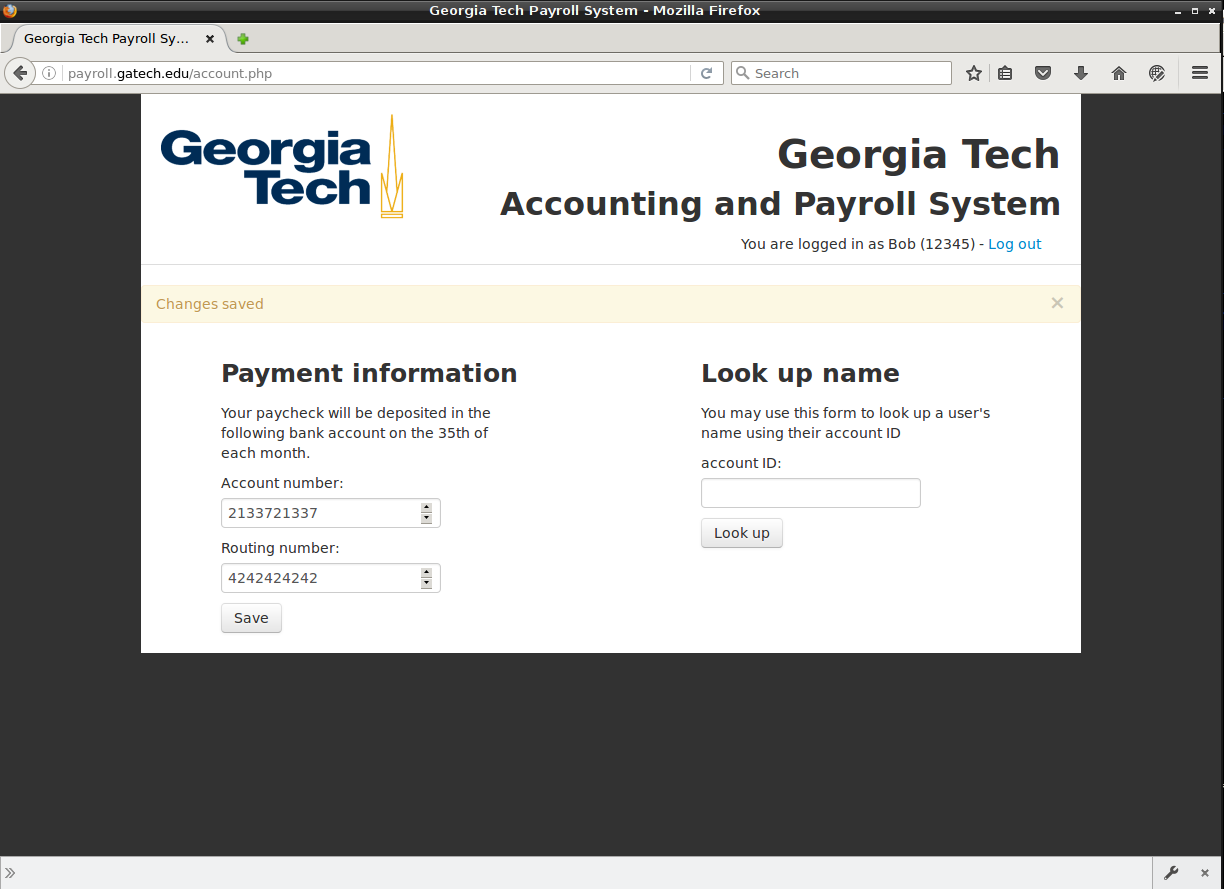


Figure : Screenshot of account page after attack is executed. This was accomplished by simply navigating to the attack page’s HTML on the VM

# Target 2: XSS

In this case the weakness is that user input is not scrubbed. All user input needs to be considered as being malicious and must be validated (<https://www.sitepoint.com/php-security-cross-site-scripting-attacks-xss/>). Proper validation of user input could prevent this attack.

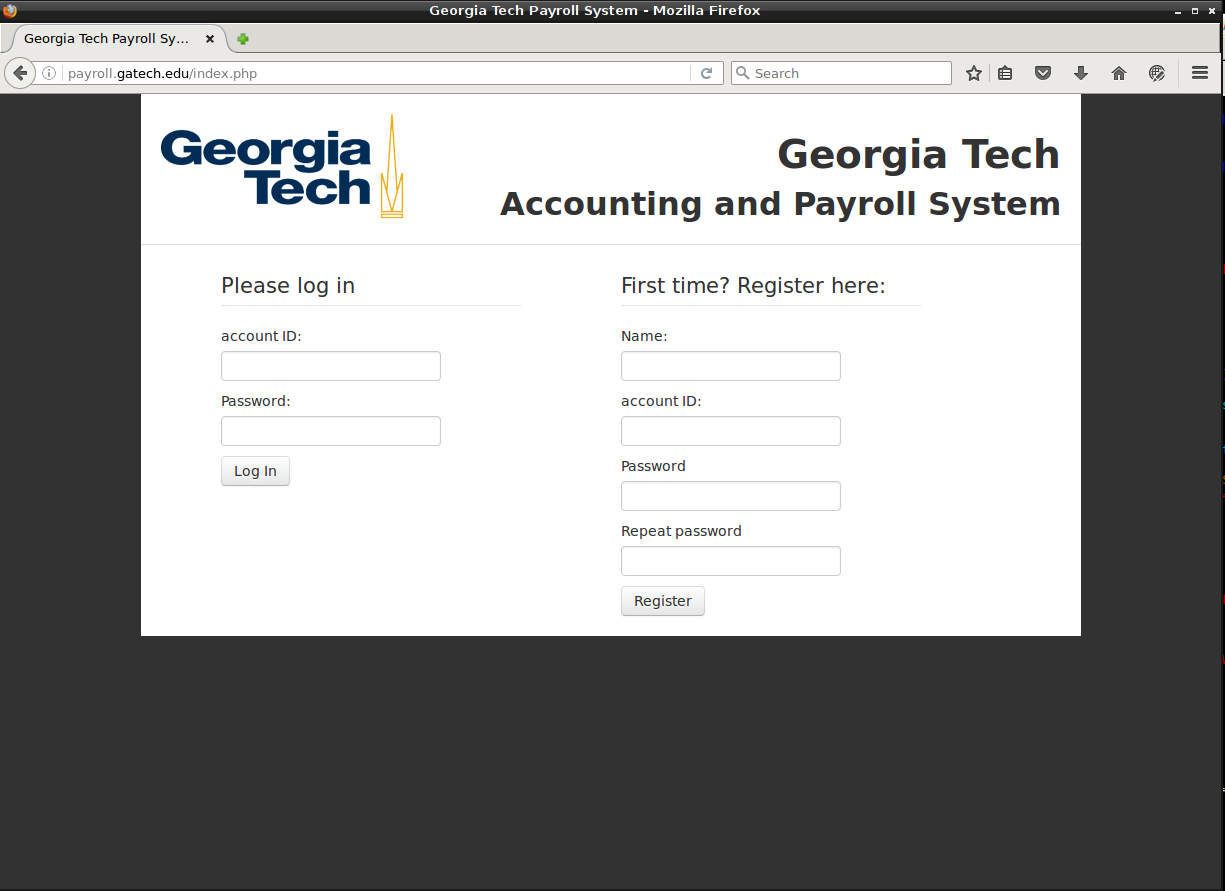


Figure : Malicious login page. Note that the look is identical and the address is the same as the normal login page

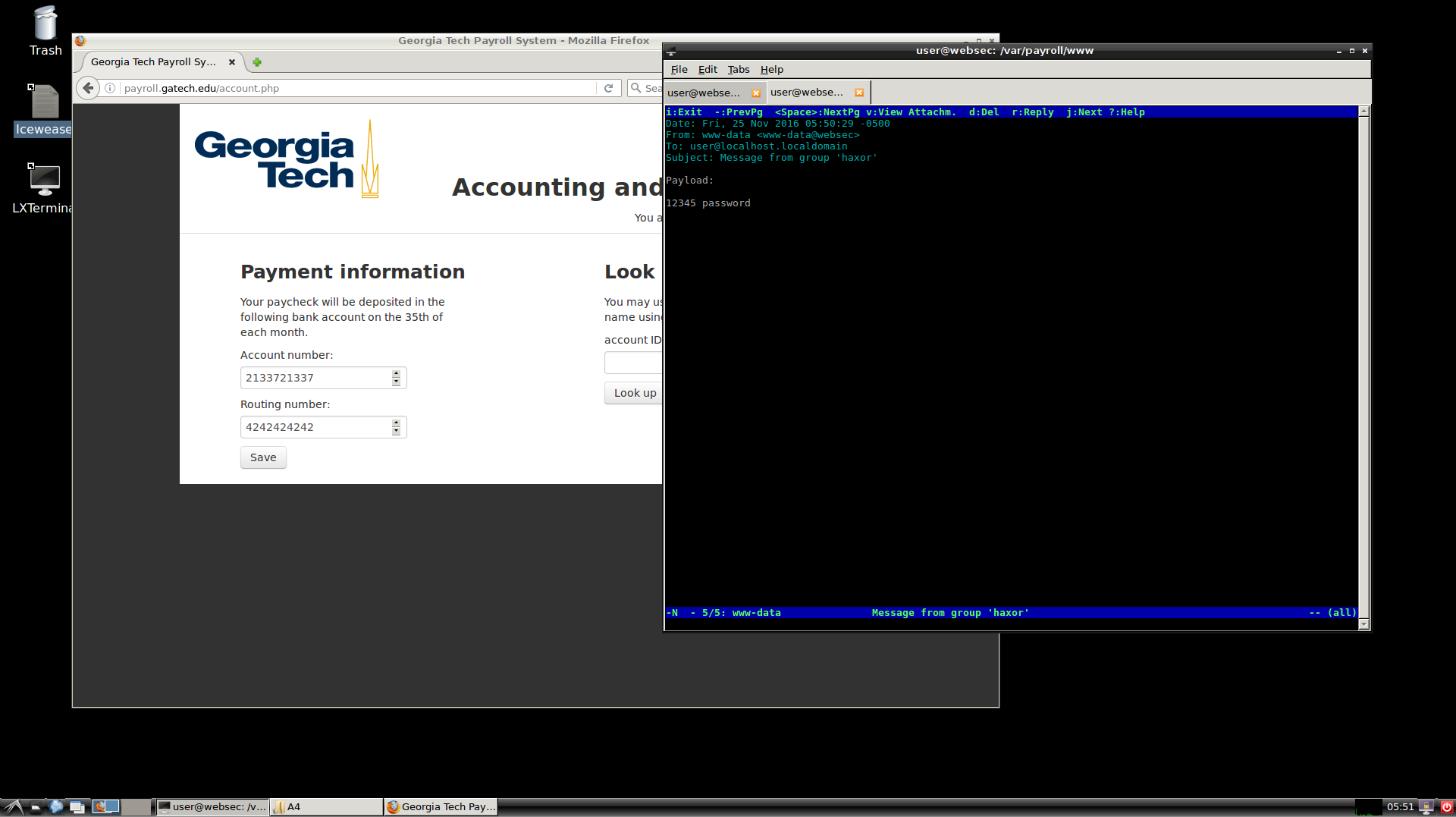


Figure : After successful login, email is sent with username and password information

# Target 3: SQL Injection

The vulnerability here is that the SQL query in index.php (or more specifically auth.php) is not properly validated before it is used. Single quotes are not removed from the input string. In fact, it seems that this website does not employ whitelisting as recommended. As such, it is possible to inject SQL commands into the query, allowing the user to login even without a password.

