Garrett Reeve

Deangelo Aguilar

Tests iteration 3

XSS to test

* "<script>alert(1)</script>"@example.com
* <a href='mailto:USER\_EMAIL'>email me</a>
* foo'onclick='alert'foo='@example.com
* http://example.com/index.php?user=<script>window.onload = function() {var AllLinks=document.getElementsByTagName("a");AllLinks[0].href = "http://badexample.com/malicious.exe";}</script>
* No hidden pages at the time of this build, no need to worry about accessing unwanted pages.

Ways to prevent XSS on email field

* Allow only entries that include @ and .com or .gov or .edu
* Don’t allow any special characters
  + Examples above rely on special characters like < > ? = - + etc.

SQL injection tests

* For input fields that talk to database
  + 105 OR 1=1
  + " or ""="
  + 105; DROP TABLE Suppliers
    - This one must be tested on a testing enviorment not prod

Protect against SQL injection

Need to use params that limit visibility and depth by attacker

* These are some examples of using params
  + txtUserId = getRequestString("UserId");  
    txtSQL = "SELECT \* FROM Users WHERE UserId = @0";  
    db.Execute(txtSQL,txtUserId);
  + txtNam = getRequestString("CustomerName");  
    txtAdd = getRequestString("Address");  
    txtCit = getRequestString("City");  
    txtSQL = "INSERT INTO Customers (CustomerName,Address,City) Values(@0,@1,@2)";  
    db.Execute(txtSQL,txtNam,txtAdd,txtCit);

**Tests on 11/11/21**

Testing the login page with the following commands run in the email space

* "<script>alert(1)</script>"@example.com
* <a href='mailto:USER\_EMAIL'>email me</a>
* foo'onclick='alert'foo='@example.com
  + All above tests failed as we have added filtering for email field and not allowed certain special characters
  + Testing username/email entries with the following tags
    - 1' or '1' = '1
    - SELECT \* FROM Users WHERE Username='1' OR '1' = '1' AND Password='1' OR '1' = '1'
    - SELECT \* FROM Users WHERE ((Username='$username') AND (Password=MD5('$password')))
    - 1' or '1' = '1'))/\*
    - Foo
      * Username/email entry tests see to be strong
  + Testing using an existing username or username that fits params
    - Unable to get passwords to display or to grab any info

WireShark

Testing of account creation page

* + - 1' or '1' = '1
    - SELECT \* FROM Users WHERE Username='1' OR '1' = '1' AND Password='1' OR '1' = '1'
    - Various SELECT statements were tested to try and display table info

Unable to grab through SQL injection

Able to see packets through Wireshark, Username is sent in plain text when account is created.

This is issue if malicious hackers wanted to grab usernames by sniffing the IP.

Able to ping page multiple times without being blocked or timed out. DDOS is possible.

Have not tested if hashing works when being sent, hashing is stored sent but not sure if hashed before transmitting.

No plain text was seen when doing testing and using sniffer.

Wire shark was used to sniff packets locally on test environment.

Conclusion

What needs to be addressed

* Sniffed packets do include lots of data
  + Allows for following of which pages send to which
  + May be able to intercept packets and change data
    - Actually, this is theoretically possible, not sure if the intercepted packets can be used to retrieve data.
    - With packet intercept possible to take down server that holds database with large quires that loop
* No password recovery system in place
  + This will bring issues such as spam from calling email links
  + May be able to intercept packet as new password is being sent to server and then changing data
  + This is all theoretical as no actual server is set up and is all local
* SQL injections on login page seem to be safe
  + Email field is very well filtered
  + Password field seems less safe as it allows for special characters
    - No fix for this as long as the email is processed first and then just check for match without possibility of running script in password field