Phishing Lab

There's not a whole lot that developers can do to prevent phishing; it is completely between the hacker and the victim. They never even come to our site. What can we do to prevent it? Nothing. But what can we do to reduce the chances that our customers will be fooled? Quite a bit.

Creating an uncomplicated site

First, if our customers see simple URLs on our site and then an attacker tempts them with a link with a complicated URL, they'll be less likely to click on it. Something won't seem right. We can create very simple URLs with route mapping.

1. Go to the site and add any product to your cart. Once you do that, you should be looking at your cart. The URL will be something like http://localhost:7777/ViewCart.aspx. That URL can be improved with a route.
2. To see an example, open global.asax. Look at the routes already in there.
3. Add a new one that looks like this:
   ```
   routes.MapPageRoute("ViewCart",
       "Cart",
       "~/ViewCart.aspx"
       );
   ```
4. Now restart the app and navigate to http://localhost:7777/Cart See a cart?
5. Try to add a product to your cart. Notice that you can still see your items, but the URL is still the old one. Let's fix that.
6. Go to Product.aspx.cs. Find the AddToCart button's click event. Notice that the last thing in there is a Response.Redirect. Change it to send the customer to the new route we've created.
7. Run and test. Once you've got them going to the new route, you can move on.

Routes and default values

The routes for the blog have already been created for you. But in certain situations you may need default values. The Blog routes show you how to provide default values. There's no need to actually change anything in the next few steps. We just want you to see examples of defaults.

8. Navigate to the blog section of our website. Reply to one of the blog entries. Examine the URLs as you're doing all this.
9. Go into global.asax and look at these routes:
   ```
   routes.MapPageRoute("BlogReply",
       "Blog/Reply/{BlogEntryId}",
       "~/BlogEntryReply.aspx",
       false,
       new RouteValueDictionary() { { "BlogEntryId", 1 } }
       );
   routes.MapPageRoute("Blog",
       "Blog/{StartBlogEntryId}/{NumberOfBlogEntries}",
       "~/Blog.aspx",
       false,
       new RouteValueDictionary() {
         { "StartBlogEntryId", 0 },
         {"NumberOfBlogEntries", 4}
       }
   ```

```
       );
```
10. Note the RouteValueDictionary. These are to provide default values if needed.

Sending an email

Phishers will often trick surfers into entering their old passwords into a fake 'change your password' page. If the user changes his or her password on our legitimate site, we should send an email telling them so. This way, if they ever get tricked by a phisher and then fail to get an email, they'll be alerted to the chicanery.

11. Edit the Account/ChangePassword.aspx page. Find the ChangeUserPassword control. Add an event by clicking on the lightning bolt at the top of the properties window.
12. Find the ChangedPassword event and double-click on it to create the new event handler.
13. Add some code like this:
```
try
{
    var to = Membership.GetUser().Email;
    var messageBody = string.Format(
       "Your password was changed at {0}.",
       DateTime.Now.ToString());
    EmailSender.Send(to, "Password changed", messageBody);
}
catch (NullReferenceException ex)
{
    //TODO: Log the error ... cannot find the user.
}
```
14. Now run and test. Change your password and check your email to see if you've got a notification that your email was changed.[*]
15. Go back in and change the message to be nicely formatted with HTML and some more informative text. In addition, it is very important for us to greet the user by name in all of our emails. So make sure you personalize it with the user's name.
16. Run and test it again.

Once you're receiving emails on password change, you can be finished.

[*] A corporate firewall that blocks SMTP will cause this to fail. Consult your System Admin to get the corporate sending settings. It's often just mail.companysite.com.