A8 Cross Site Request Forgery Lab

We learned about the dangers of CSRF in the last lecture. Now let's work with it. In this lab we'll explore a vulnerable page and then learn how to protect it.

Preparing

1. For learning purposes, this CSRF attack happens to use an iframe. Please go back and remove any iframe preventions that you may have added for clickjacking protection.
2. Next, double-check that you are NOT logged in. Hit our website Using IE 11 or below. If you see your name in the upper-right, go ahead and click 'log out'.
3. Look at the Orders table in the database. Look at the last order. Does it look alright to you? _____.
4. Now open a new browser tab[‡]. In it visit localhost:8888/csrf.html. Click the CSRF link. A CSRF attack was mounted against you but you were not logged in so no harm, no foul.
5. Look at the Orders table again. Is the last order still okay? _____ It should be.

Mounting the attack

6. Go back and log in.
7. Visit localhost:8888/csrf.html again in a separate tab. Once again you were hit with an invisible CSRF attack. But since you were logged in, it should have succeeded. Let's go look.
8. One more look at the Orders table should show that you now have a new order for a large amount of money.

Imagine if this were real. If all the stars aligned just right and someone were to hit the attack page, all their paid-for goodies would be shipped to the bad guys! Ouch!

Bonus! If you want to see how all this is done, go and look at the source for csrf.js in the Evil site. You'll see it is rendering pages in a protected portion of our site.

Hardening the site with a CAPTCHA or re-entering a password

The synchronizer token pattern will not help in this situation since our attacker has figured out how to walk through legitimate pages in proper order. Instead, let's eiter implement a CATCHA or ask the user to re-enter their password. You and your partner decide which you'd like to do.

If you choose to use a captcha, you could write your own or go with a publicly available one. We're going to use a publicly available one.

9. Go to the Checkout.aspx page. Add this line just under the <@ Page …> directive:

```
<%@ Register TagPrefix="recaptcha" Namespace="Recaptcha"
Assembly="Recaptcha" %>
```

10. Put this somewhere on the page. Maybe just above the "Place order" button would be a good place.

```
<recaptcha:RecaptchaControl ID="recaptcha" runat="server"
 PublicKey="6Lf8LMkSAAAAAEKR0lkvKnlaU-7IZwy6Ayo3Re8C"
 PrivateKey="6Lf8LMkSAAAAANEpuHz0RMOTa-TtLdxyuDEIScP7" />
```

11. Finally, put this at the top of the button's click event:

```
recaptcha.Validate();
if (! Page.IsValid)
```

[‡] Some browsers have built-in CSRF protection. You'll need to try Firefox 18 or below, IE 11 or below, Chrome 12 or below, or Opera 8 or below.

```
{
    lblErrorMessage.Text = "Not able to validate you. Try again.";
    return;
}
```

12. Run and test your site. Make sure you can still manually order goods and that the CAPTCHA works. Once you can, you're ready to try the attack again.
13. If you chose to have the user re-enter their password, take these steps:
14. Add a password field just above the Place Order button. Open Checkout.aspx and alter the code to look like something like this:

```
<fieldset>
  <asp:Label ID="lblPassword" runat="server"
      AssociatedControlID="txtPassword">
    For security reasons, please re-enter your password:
  </asp:Label><br />
  <asp:TextBox ID="txtPassword" TextMode="Password"
      CssClass="textEntry" runat="server" /><br />
  <asp:Button ID="btnPlaceOrder" runat="server" Text="Place Order"
      OnClick="btnPlaceOrder_Click"
      ValidationGroup="ShippingValidationGroup" />
</fieldset>
```

15. Then bind the btnPlaceOrder_Click() method and add some code like this:

```
var password = txtPassword.Text;
if (! System.Web.Security.Membership.ValidateUser(
          User.Identity.Name, password))
{
  lblErrorMessage.Text = "Bad password. Please try again.";
  return;
}
```

16. These things will require that a valid password has been entered before the user can proceed.

Re-running the attack

17. Reopen your attack site.
18. Try the attack again like you did in steps one through six above.
19. Check your orders. This time the attack should fail. Did it? _____

When you can legitmately order goods and cannot order them through evil means, you can be finished.