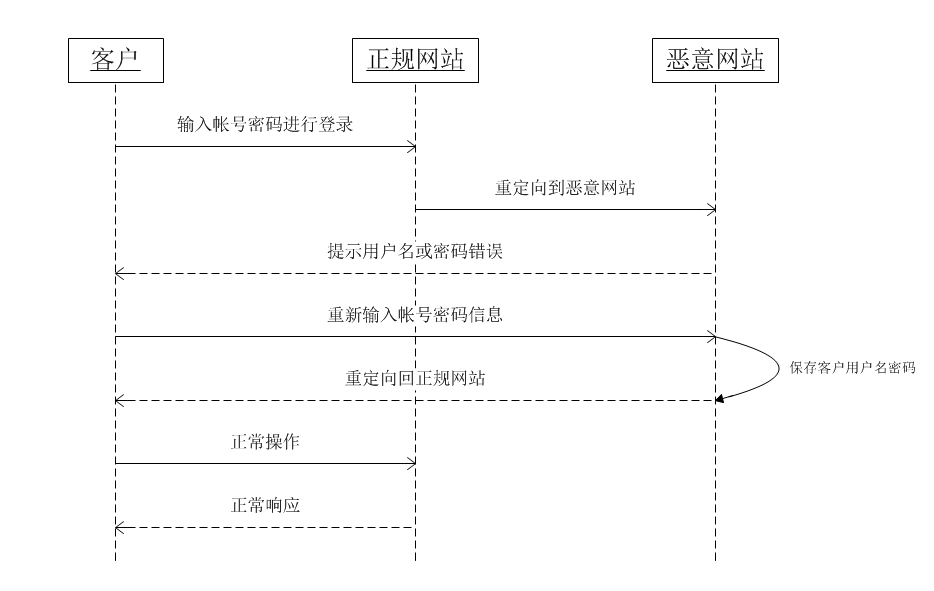
[Web安全相关（三）：开放重定向(Open Redirection)](http://www.cnblogs.com/Erik_Xu/p/5497479.html)

**简介**

　　那些通过请求（如查询字符串和表单数据）指定重定向URL的Web程序可能会被篡改，而把用户重定向到外部的恶意URL。这种篡改就被称为开发重定向攻击。

**场景分析**



　　假设有一个正规网站http://nerddinner.com/，还有一个恶意网站或钓鱼网站http://nerddiner.com/（注意：这里少了个n）。

　　一天，小白收到了别人发的链接：http://nerddinner.com/Account/LogOn?returnUrl=http://nerddiner.com。

　　1. 打开链接后进入了登录界面，小白输入了自己的帐号名密码进行登录。

　　2. 登录成功后重定向到了恶意网站。

　　3. 恶意网站是一个仿造正规网站的登录页面，并在上面提示用户名或密码错误。

　　4. 小白按照提示重新输入了帐号密码信息。

　　5. 恶意网站保存了客户的用户名密码，然后重定向会正规网站。

　　6. 小白继续平时正常的操作。

**防止开放重定向**

　　防止开发重定向只需要判断重定向的链接是本地的链接或者是合法的链接即可。

　　1. 如果登录链接和站点其他页面都在同一个域名，在ASP.MVC中可以用Url.IsLocalUrl(string url)来判断。

　　2. 如果登录链接和站点其他页面不在同一个域名，如单点登录，则需要自己去实现判断的逻辑。

**核心代码**

[复制代码](javascript:void(0);)

[HttpPost]

public ActionResult LogOn(LogOnModel model, string returnUrl)

{

//Your logon logic here.

FormsAuthentication.SetAuthCookie(model.UserName, false);

if (!string.IsNullOrEmpty(returnUrl)

&& Url.IsLocalUrl(returnUrl) //Comment out this code will cause open redirection

)

{

return Redirect(returnUrl);

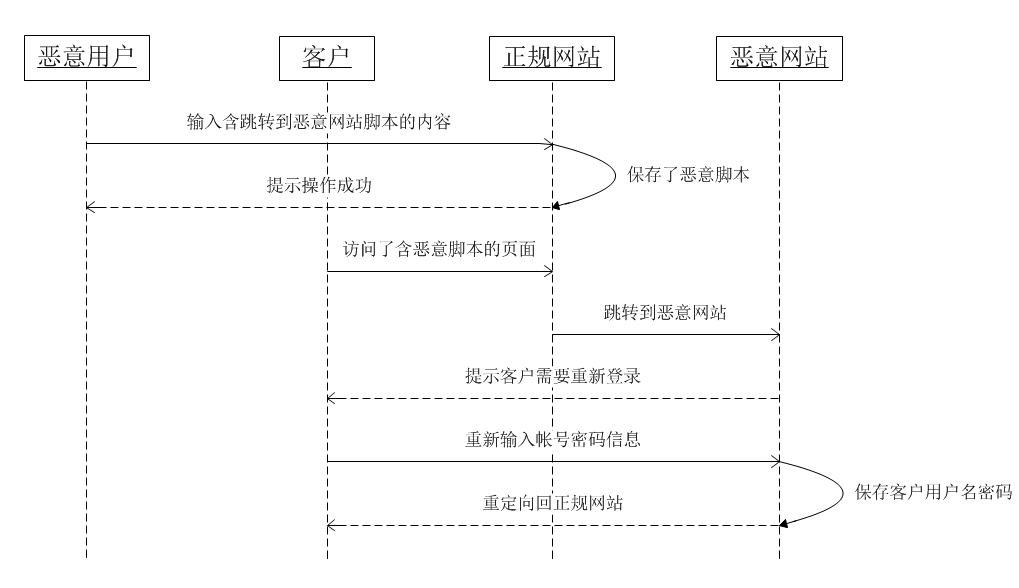
}

return RedirectToAction("Index", "Home");

}

[复制代码](javascript:void(0);)

**联想到XSS**



　　1. 恶意用户在正规网站下挂了跳转到恶意网站的脚本。

　　2. 普通用户访问到含恶意脚本的页面会跳转到恶意网站。

　　3. 恶意网站是一个仿造正规网站的登录页面，并在上面提示需要重新登录。

　　4. 小白按照提示重新输入了帐号密码信息。

　　5. 恶意网站保存了客户的用户名密码，然后重定向会正规网站。

　　注：这种方式每次访问含恶意脚本的页面都会跳转到恶意网站（提示重新登录），而开放重定向只会提示用户名密码错误一次，相对而言，开放重定向的无感知效果要好一点。

**源码下载**

　　如果大家想尝试开放重定向的效果，可以去下载代码，把判断本链接的语句Url.IsLocalUrl(returnUrl)注释掉，然后在登录页面加上想要跳转到的页面，如http://xxx.com/Account/LogOn?ReturnUrl=http://www.baidu.com。

　　下载地址：<https://github.com/ErikXu/OpenRedirection>