源碼掃描注意項目

1. XSS 弱點防護:
   Acton 收到之參數 需做過濾檢查
   ParamChecker.Check(str)
   或
   AntiXssEncoder.HtmlEncode(str, true)

2. SESSION FIXATION

防護措施

- 禁止將 Session ID 使用 URL (GET) 方式來傳遞
- 設定加強安全性的 Cookie 屬性：HttpOnly (無法被 JavaScript 存取)
- 設定加強安全性的 Cookie 屬性：Secure (只在 HTTPS 傳遞，若網站無 HTTPS 請勿設定)
- 在需要權限的頁面請使用者重新輸入密碼

Eg: 登出時

```
Session.Clear();
Session.Abandon();
Session.RemoveAll();

    if (Request.Cookies["ASP.NET_SessionId"] != null)
    {
        Response.Cookies["ASP.NET_SessionId"].Value = string.Empty;
        Response.Cookies["ASP.NET_SessionId"].Expires =
DateTime.Now.AddMonths(-20);
    }

    if (Request.Cookies["AuthToken"] != null)
    {
```

```
            Response.Cookies["AuthToken"].Value = string.Empty;
            Response.Cookies["AuthToken"].Expires =
    DateTime.Now.AddMonths(-20);
        }
```

3. ajax post 需使用 ValidateAntiForgeryToken

```
var token = $('input[name="__RequestVerificationToken"]').val();
     $.ajax({
         type: "POST",
         url: "@Url.Action("MultiCreate")",
         async: false,
         data: {
             __RequestVerificationToken: token,
```

4. XXE 漏洞 防護 (Improper Restriction of XXE Ref)

   XmlDocument doc = new XmlDocument();
   doc.XmlResolver = null; //增加此設定

5. Missing Column Encryption

   解法: using (SqlConnection conn = new SqlConnection(@"Data Source=資料庫;Initial Catalog=SecurityDB;User Id=帳號;Password=密碼;Column Encryption Setting=Enabled"))

6. Client Potential XSS

   解法: 前端 JS 增加 htmlencode(s) 與 htmldecode(s) 做編碼
   例: $('#PayProductShow').html("<label class='control-label'>" + htmlencode(newPayProduct.text()) + "</label>")

7. Path Traversal

   解: filename.Replace("\", "").Replace("..", "")