عنوان: **بررسی Microsoft Anti-Cross Site Scripting Library**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۸/۲۰ ۱۵:۵۳:۲۵
آدرس: www.dotnettips.info
برچسب‌ها: Security

هنگام نمایش اطلاعات در وب باید اطلاعات خام دریافتی از کاربر را encode کرده و سپس نمایش داد تا از حملات XSS یا cross site scripting attacks در امان ماند. مثلا وبلاگی را طراحی کرده‌اید و یک نفر اطلاعات زیر را بجای توضیحات ارسال کرده است:

```
<SCRIPT>alert('XSS')</SCRIPT>
```

اگر اطلاعات به همین شکل دریافت و بدون تغییر هم نمایش داده شود، یک ضعف امنیتی برای سایت شما به‌حساب خواهد آمد. (بحث دزدیدن اطلاعات کوکی و امثال آن از این طریق با معرفی HttpOnly cookies در IEهای جدید و فایرفاکس 3 به بعد تقریبا منتفی شده است اما می‌توانند با ارسال انبوهی اسکریپت، مشاهده صفحه را با crash کردن مرورگر کاربران همراه کنند)
مایکروسافت برای این منظور Microsoft Anti-Cross Site Scripting Library را ارائه داده است. نمونه بهبود یافته HttpUtility.HtmlEncode که در فضای نام System.Web موجود است.

در اینجا قصد داریم این کتابخانه را با لیست زیر آزمایش کنیم:
http://ha.ckers.org/xss.html
در همان صفحه اگر دقت کنید، لیست حملات را به صورت یک فایل xml هم ارائه داده است:
http://ha.ckers.org/xssAttacks.xml
برای خواندن این فایل xml در دات نت روش‌های زیادی وجود دارد منجمله XML serialization .

ساختار این فایل به شکل زیر است:

```
<?xml version="1.0" encoding="UTF-8"?>
<xss>
<attack>
 <name>x1</name>
 <code>x2</code>
 <desc>x3</desc>
 <label>x4</label>
 <browser>x5</browser>
</attack>
.
.
.
```

بنابراین شیء نمایانگر آن می‌تواند به صورت لیستی از کلاس زیر باشد:

```
public class attack{
      public string name { get; set; }
      public string code { get; set; }
      public string desc { get; set; }
      public string label { get; set; }
      public string browser { get; set; }
  }
```

برای دریافت این لیست و بارگذاری فایل xml مربوطه با استفاده از روش XML serialization خواهیم داشت:

```
using System.Collections.Generic;
using System.IO;
using System.Xml.Serialization;

      public static List<attack> DeserializeFromXML(string path)
      {
          XmlRootAttribute root = new XmlRootAttribute("xss");
          XmlSerializer deserializer =
              new XmlSerializer(typeof (List<attack>),root);
          using (TextReader textReader = new StreamReader(path))
          {
              return (List<attack>)deserializer.Deserialize(textReader);
          }
      }
```

در ادامه فرض بر این است که ارجاعی از اسمبلی AntiXssLibrary.dll به پروژه اضافه شده است، همچنین فایل xssAttacks.xml فوق نیز در کنار فایل اجرایی برنامه ، مثلا یک برنامه کنسول قرار گرفته است:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Text;
using Microsoft.Security.Application;

      private static void testMethod()
      {
          StringBuilder sb = new StringBuilder();
          sb.AppendFormat("<html>{0}", Environment.NewLine);
          sb.AppendFormat("<body>{0}", Environment.NewLine);

          List<attack> data = XMLParser.DeserializeFromXML("xssAttacks.xml");
          foreach (attack atk in data)
          {
              string cleanSafeHtmlInput = AntiXss.HtmlEncode(atk.code);
              sb.AppendFormat("{0}<br>{1}", cleanSafeHtmlInput, Environment.NewLine);
          }

          sb.AppendFormat("</body>{0}", Environment.NewLine);
          sb.AppendFormat("</html>");

          File.WriteAllText("out.htm", sb.ToString());
      }
```

پس از اجرای تابع فوق، خروجی ما یک فایل html خواهد بود به نام out.htm . آنرا در مرورگر خود باز کنید. بدون هیچ مشکلی باز خواهد شد و خروجی امنی را مشاهده خواهید کرد. برای مشاهده اثر واقعی این کتابخانه، قسمت AntiXss.HtmlEncode را از کد فوق حذف کنید و یکبار دیگر برنامه را اجرا کنید. اکنون فایل نهایی را در مرورگر باز کنید. با انبوهی از alert های جاوا اسکریپتی مواجه خواهید شد که اهمیت کتابخانه فوق را جهت ارائه خروجی امن در صفحات وب مشخص می‌سازد.

## نظرات خوانندگان

نویسنده: Salar
تاریخ: ۱۳۸۷/۰۸/۲۱ ۰۹:۲۴:۰۰

روش جالب برای تست این کلاس بود. ممنون

اینجا مثالهایی از سایر توابع هم هست
http://blogs.msdn.com/cisg/archive/2008/08/26/what-is-microsoft-antixss.aspx

نویسنده: sff
تاریخ: ۱۳۹۱/۰۶/۱۳ ۳:۱۴

<SCRIPT>alert('XSS')</SCRIPT>

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۳ ۹:۰

احسنت! بزرگ شدی!