عنوان: **ELMAH و حملات XSS**
نویسنده: داود زینی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۳:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, ASP.Net MVC, ELMAH, xss

ASP.NET جهت مقابله با حملات XSS بطور پیش‌فرض از ورود تگ‌های HTML جلوگیری می‌کند و در صورتی که ورودی کاربر شامل این تگ‌ها باشد، HttpRequestValidationException صادر می‌گردد. لاگ کردن و بررسی این خطاها جهت آگاهی از وجود حمله بی اهمیت نیست. اما متأسفانه ELMAH که به عنوان معمول‌ترین ابزار ثبت خطاها کاربرد دارد این نوع Exceptionها را ثبت نمی‌کند. دلیل آن هم این است که ELMAH در رویه‌های درونی خود اقدام به خواندن ورودی‌های کاربر می‌کند و در این هنگام اگر ورودی کاربر نامعتبر باشد، Exception مذکور صادر می‌شود و فرصتی برای ادامه روند و ثبت خطا باقی نمی‌ماند. به هر حال ضروری است که این نقیصه را خودمان جبران کنیم. راه حل افزودن یک فیلتر سفارشی برای ثبت خطاها به شکل زیر است (ASP.NET MVC):

```
public class ElmahRequestValidationErrorFilter : IExceptionFilter
{
    public void OnException(ExceptionContext context)
    {
        if (context.Exception is HttpRequestValidationException)
           ErrorLog.GetDefault(HttpContext.Current).Log(new Error(context.Exception));
    }
}
```

فیلتر فوق باید در Global.asax معرفی شود:

```
public static void RegisterGlobalFilters (GlobalFilterCollection filters)
{
    filters.Add(new ElmahRequestValidationErrorFilter());
    filters.Add(new HandleErrorAttribute());
}
```

به این ترتیب HttpRequestValidationException هم بعد از این در سیستم ELMAH ثبت خواهد شد.

نظرات خوانندگان

نویسنده: یاسر مرادی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۳

با سلام، در چک لیست ASP.NET MVC مورد زیر وجود دارد
آیا مورد زیر کماکا معتبر است ؟
- فیلتر پیش فرض مدیریت خطاها حذف و بجای آن از ELMAH استفاده شود.

با سپاس

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۲۶

- به سطر HandleErrorAttribute پیش فرض نیازی نیست (البته اگر تنظیمات وب کانفیگ درستی داشته باشید). در قسمت 16 سری MVC توضیح دادم. وجود آن سبب می‌شود که ELMAH اصلا کار نکند و خطای مدیریت نشده‌ای به آن ارجاع داده نشود (چون قبلا مدیریت شده).
- بله. همچنان ELMAH معتبر است. نکته فوق را هم اضافه کنید، کاملتر خواهد شد.

نویسنده: داود زینی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۲۸

سلام
این چک لیست توسط آقای نصیری تهیه شده بود. این مورد هم از نظر من معتبر است.
با تشکر

عنوان: **آشنایی با چالش های امنیتی در توسعه برنامه‌های تحت وب، بخش اول**
نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۳/۰۷/۱۱ ۸:۵۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security, xss, Cross-site scripting

در پروژه‌های بزرگ نرم افزاری، از قدیم بحث تامین امنیت پروژه، یکی از چالش‌های مهم بوده است. از دیدگاه شخصی بنده، یک مدیر نرم افزار یا حتی یک توسعه دهنده‌ی برنامه‌های تحت وب، لازم است علاوه بر صرف وقت مطالعاتی و آشنایی و تسلط بر مباحث طراحی معماری سیستم‌های تحت وب، که از اهمیت بالا و مقیاس بزرگی برخوردارند آشنایی لازم را با چالش‌های امنیتی در پیاده سازی اینگونه سیستم‌ها داشته باشد. امنیت در یک سیستم بزرگ و ارائه دهنده خدمات، باعث می‌شود تا کاربر علاوه بر یک تجربه کاربری (user experience) خوب از سیستم که حاصل پیاده سازی صحیح سیستم می‌باشد، اعتماد ویژه‌ای به سیستم مذکور داشته باشد. گاها کاربران به علت بی اعتمادی به شرایط امنیتی حاکم بر یک سیستم، از تجربه کاربری خوب یک سیستم چشم پوشی می‌کنند. اهمیت این مسئله تا جاییست که غول‌های تکنولوژی دنیا همچون Google درگیر این چالش می‌باشند و همیشه سعی بر تامین امنیت کاربران علاوه بر ایجاد تجربه کاربری خوب دارند. پس عدم توجه به این موضوع میتواند خسارات وارده جبران ناپذیری را به یک سیستم از جهت‌های مختلف وارد کند.

در این سری از مقالات، بنده سعی دارم تا حد توان در رابطه با چالش‌های امنیتی موجود در زمینه توسعه برنامه‌های تحت وب، مطالبی را منتشر کنم. از این رو امیدوارم تا این سری از مقالات برای دوستان مفید واقع گردد.

در این سری از مقالات چالش‌های امنیتی زیر مورد بحث و بررسی واقع خواهند گردید

XSS , LDAPi ,RFI ,LFI ,SQLi ,RFD ,LFD ,SOF ,BSQLI ,DNN ,BOF ,CRLF ,CSRF ,SSI ,PCI ,SCD ,AFD ,RCE

در بخش اول از این سری مقالات ، به بررسی آسیب پذیری **Cross-site scripting** میپردازیم .

واژه XSS مخفف Cross-site scripting ، نوعی از آسیب پذیریست که در برنامه‌های تحت وب نمود پیدا میکند. به طور کلی و خلاصه، این آسیب پذیری به فرد نفوذ کننده اجازه تزریق اسکریپت‌هایی را به صفحات وب، می‌دهد که در سمت کاربر اجرا می‌شوند (Client Side scripts) . در نهایت این اسکریپت‌ها توسط سایر افرادی که از صفحات مورد هدف قرار گرفته بازدید می‌کنند اجرا خواهد شد.

هدف از این نوع حمله :

بدست آوردن اطلاعات کوکی‌ها و سشن‌های کاربران (مرتبط با آدرسی که صفحه آلوده شده در آن قرار دارد) است. سپس فرد نفوذ کننده متناسب با اطلاعات بدست آمده می‌تواند به اکانت شخصی کاربران مورد هدف قرار گرفته، نفوذ کرده و از اطلاعات شخصی آن‌ها سوء استفاده کند .

به صورت کلی دو طبقه بندی برای انواع حملات Cross-site scripting وجود دارند.

حملات XSS ذخیره سازی شده (Stored XSS Attacks) :

در این نوع ، کدهای مخرب تزریق شده، در سرور سایت قربانی ذخیره میشوند. محل ذخیره سازی می‌تواند دیتابیس سایت یا هر جای دیگری که داده‌ها توسط سایت یا برنامه تحت وب بازیابی می‌شوند و نمایش داده می‌شوند باشد. اما اینکه چگونه کدهای مخرب در منابع یاد شده ذخیره میشوند؟

فرض کنید در سایت جاری آسیب پذیری مذکور وجود دارد. راه‌های ارسال داده‌ها به این سایت چیست؟ نویسندگان میتوانند مطلب ارسال کنند و کاربران میتوانند نظر دهند. حال اگر در یکی از این دو بخش بررسی‌های لازم جهت مقابله با این آسیب پذیری

وجود نداشته باشد و نوشته‌های کاربران که می‌تواند شامل کدهای مخرب باشد مستقیما در دیتابیس ذخیره شده و بدون هیچ اعتبار سنجی نمایش داده شود چه اتفاقی رخ خواهد داد؟ مسلما با بازدید صفحه آلوده شده، کدهای مخرب بر روی مرورگر شما اجرا و کوکی‌های سایت جاری که متعلق به شما هستند برای هکر ارسال میشود و ...

حملات XSS منعکس شده (Reflected XSS Attacks) :

در این نوع از حمله، هیچ نوع کد مخربی در منابع ذخیره سازی وبسایت یا اپلیکیشن تحت وب توسط فرد مهاجم ذخیره نمی‌شود ! بلکه از ضعف امنیتی بخش‌هایی همچون بخش جستجو وب سایت، بخش‌های نمایش پیغام خطا و ... استفاده میشود ... اما به چه صورت؟

در بسیاری از سایت‌ها، انجمن‌ها و سیستم‌های سازمانی تحت وب، مشاهده می‌شود که مثلا در بخش جستجو، یک فیلد برای وارد کردن عبارت جستجو وجود دارد. پس از وارد کردن عبارت جستجو و submit فرم، علاوه بر نمایش نتایج جستجو، عبارت جستجو شده نیز به نمایش گذاشته میشود و بعضا در بسیاری از سیستم‌ها این عبارت قبل از نمایش اعتبار سنجی نمی‌شود که آیا شامل کدهای مخرب می‌باشد یا خیر. همین امر سبب میشود تا اگر عبارت جستجو شامل کدهای مخرب باشد، آن‌ها به همراه نتیجه‌ی جستجو اجرا شوند.

اما این موضوع چگونه مورد سوء استفاده قرار خواهد گرفت؟ مگر نه اینکه این عبارت ذخیره نمیشود پس با توضیحات فوق، کد فقط بر روی سیستم مهاجم که کد جستجو را ایجاد می‌کند اجرا می‌شود، درست است؟ بله درست است ولی به نقطه ضعف زیر توجه کنید ؟

```
www.test.com/search?q=PHNjcmlwdD5hbGVydChkb2N1bWVudC5jb29raWUpOzwvc2NyaXB0Pg==
```

این آدرس حاصل submit شدن فرم جستجو وب‌سایت test (نام وب‌سایت واقعی نیست و برای مثال است) و ارجاع به صفحه نتایج جستجو میباشد. در واقع این لینک برای جستجوی یک کلمه یا عبارت توسط این وبسایت تولید شده و از هر کجا به این لینک مراجعه کنید عبارت مورد نظر مورد جستجو واقع خواهد شد. در واقع عبارت جستجو به صورت Base64 به عنوان یک query String به وبسایت ارسال می‌شود؛ علاوه بر نمایش نتایج، عبارت جستجو شده نیز به کاربر نشان داده شده و اگر آسیب پذیری مورد بحث وجود داشته باشد و عبارت شامل کدهای مخرب باشد، کدهای مخرب بر روی مرورگر فردی که این لینک را باز کرده اجرا خواهد شد!

در این صورت کافیست فرد مهاجم لینک مخرب را به هر شکلی به فرد مورد هدف بدهد (مثلا ایمیل و ...). حال در صورتیکه فرد لینک را باز کند (با توجه به اینکه لینک مربوط به یک سایت معروف است و عدم آگاهی کاربر از آسیب پذیری موجود در لینک، باعث باز کردن لینک توسط کاربر می‌شود)، کدها بر روی مرورگرش اجرا شده و کوکی‌های سایت مذکور برای مهاجم ارسال خواهد شد ... به این نوع حمله XSS ، نوع انعکاسی می‌گویند که کاملا از توضیحات فوق الذکر، دلیل این نامگذاری مشخص می‌باشد.

اهمیت مقابله با این حمله :

برای نمونه این نوع باگ حتی تا سال گذشته در سرویس ایمیل یاهو وجود داشت. به شکلی که یکی از افراد انجمن hackforums به صورت Private این باگ را به عنوان Yahoo 0-Day XSS Exploit در محیط زیر زمینی و بازار سیاه هکرها به مبلغ چند صد هزار دلار به فروش می‌رساند. کاربران مورد هدف کافی بود تا فقط یک ایمیل دریافتی از هکر را باز کنند تا کوکی‌های سایت یاهو برای هکر ارسال شده و دسترسی ایمیل‌های فرد قربانی برای هکر فراهم شود ... (در حال حاظر این باگ در یاهو وجو ندارد).

چگونگی جلوگیری از این آسیب پذیری

در این سری از مقالات کدهای پیرامون سرفصل‌ها و مثال‌ها با ASP.net تحت فریم ورک MVC و به زبان C# خواهند بود. هر چند کلیات مقابله با آسیب پذیری هایی از این دست در تمامی زبان‌ها و تکنولوژی‌های تحت وب یکسان میباشند.

خوشبختانه کتابخانه‌ای قدرتمند برای مقابله با حمله مورد بحث وجود دارد با نام AntiXSS که میتوانید آخرین نسخه آن را با فرمان زیر از طریق nugget به پروژه خود اضافه کنید. البته ذکر این نکته **حائز اهمیت** است که Asp.net و فریم ورک MVC به صورت توکار تا حدودی از بروز این حملات جلوگیری می‌کند. برای مثال به این صورت که در View ها شما تا زمانی که از MvcHtmlString استفاده نکنید تمامی محتوای مورد نظر برای نمایش به صورت Encode شده رندر می‌شوند. این داستان برای Url ها هم که به صورت پیش فرض encode میشوند صدق می‌کند. ولی گاها وقتی شما برای ورود اطلاعات مثلا از یک ادیتور WYSWYG استفاده می‌کنید و نیاز دارید داده‌ها را بدون encoding رندر کنید. آنگاه به ناچار مجاب بر اعمال یک سری سیاست‌های خاص‌تر بر روی داده مورد نظر برای رندر می‌شوید و نمی‌توانید از encoding توکار فوق الذکر استفاده کنید. آنگاه این کتابخانه در اعمال سیاست‌های جلوگیری از بروز این آسیب پذیری می‌تواند برای شما مفید واقع شود.

```
PM> Install-Package AntiXSS
```

این کتابخانه مجموعه‌ای از توابع کد کردن عبارات است که از مواردی همچون Html, XML, Url, Form, LDAP, CSS, JScript and VBScript پشتیبانی می‌کند. استفاده از آن بسیار ساده می‌باشد. کافیست ارجاعات لازم را به پروژه خود افزوده و به شکل زیر از توابع ارائه شده توسط این کتابخانه استفاده کنید:

```
…
var reviewContent = model.UserReview;
reviewContent = Microsoft.Security.Application.Encoder.HtmlEncode(review);
…
```

امیدوارم در اولین بخش از این سری مقالات، به صورت خلاصه مطالب مهمی که باعث ایجاد فهم کلی در رابطه با حملات Xss وجود دارد، برای دوستان روشن شده و پیش زمینه فکری برای مقابله با این دست از حملات برایتان به وجود آمده باشد.

نظرات خوانندگان

نویسنده: مجید و مسعود منظوری
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۲:۱۳

سلام
جایی دیدیم که نوعی از حملات هست که هکر تو یکی از پوشه‌های سایت (مثلا پوشه Images) یک اسکریپت یا یک صفحه asp قرار میده و کاربر رو به سایت خودش هدایت میکنه، یا اینکه یک عکس نشون میده
ظاهرا این حملات بیشتر برای استفاده از سایت هدف برای تبلیغات به کار میره
راه مقابله با این حملات چی میتونه باشه؟
این حملات به نام حملات CLRF شناخته میشن گویا

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۲:۵۴

دسترسی اجرایی را از این پوشه‌ها با تنظیم ذیل در وب کانفیگ سایت، بگیرید:

```
<location path="upload">
  <system.webServer>
    <handlers accessPolicy="Read" />
  </system.webServer>
</location>
```

اطلاعات بیشتر: ^ و ^

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۳:۲۳

در واقع آپلود فایل‌های مخرب نوعی حفره امنیتی در توسعه اپلیکیشن هست که موجب سوء استفاده میتونه واقع بشه . این حفره امنیتی به Unrestricted File Upload معروفه که با نکته ای که آقای نصیری ذکر کردن قابل حل هست در asp net (لینک‌های ارجاعی رو مطالعه بفرمایید)

در رابطه با ارجاع کاربران به سایت هدف نوع حمله همون CLRF هست که شامل xss هم میشه .
مقابله با این نوع حملات ساده و استوار بر دو اصل اساسی هست :
1 : همیشه بر این قانون که به ورودی داده‌های کاربر اعتماد نکنید استوار باشید
2 : تمامی ورودی‌های کاربران را که قرار است مورد مشاهده در عموم و خصوص کاربران در مرورگر باشد تا حد ممکن پاکسازی کنید
مطالعه بیشتر ^