

در پروژه های بزرگ نرم افزاری، از قدیم بحث تامین امنیت پروژه، یکی از چالش های مهم بوده است. از دیدگاه شخصی بنده، یک مدیر نرم افزار یا حتی یک توسعه دهنده برنامه های تحت وب، لازم است علاوه بر صرف وقت مطالعاتی و آشنایی و تسلط بر مباحث طراحی معماری سیستم های تحت وب، که از اهمیت بالا و مقیاس بزرگی برخوردارند آشنایی لازم را با چالش های امنیتی در پیاده سازی اینگونه سیستم ها داشته باشد. امنیت در یک سیستم بزرگ و ارائه دهنده خدمات، باعث می شود تا کاربر علاوه بر یک تجربه کاربری (user experience) خوب از سیستم که حاصل پیاده سازی صحیح سیستم می باشد، اعتماد ویژه ای به سیستم مذکور داشته باشد. گاهی کاربران به علت بی اعتمادی به شرایط امنیتی حاکم بر یک سیستم، از تجربه کاربری خوب یک سیستم چشم پوشی می کنند. اهمیت این مسئله تا جاییست که گول های تکنولوژی دنیا همچون Google درگیر این چالش می باشند 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 جهت های مختلف وارد کند.

در این سری از مقالات، بنده سعی دارم تا حد توان در رابطه با چالش های امنیتی موجود در زمینه توسعه برنامه های 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 چالش های 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 برنامه های تحت وب نمود پیدا میکند. به طور کلی و خلاصه، این آسیب پذیری به فرد نفوذ کننده اجازه تزریق اسکریپت هایی را به صفحات وب، می دهد که در سمت کاربر اجرا می شوند (Client Side scripts) . در نهایت این اسکریپت ها توسط سایر افرادی که از صفحات مورد هدف قرار گرفته بازدید می کنند اجرا خواهد شد.

هدف از این نوع حمله :

بدست آوردن اطلاعات کوکی ها و سشن های کاربران (مرتبط با آدرسی که صفحه آلوده شده در آن قرار دارد) است. سپس فرد نفوذ کننده متناسب با اطلاعات بدست آمده می تواند 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 می تواند دیتابیس سایت یا هر جای دیگری که داده ها توسط سایت یا برنامه تحت وب بازبینی می شوند و نمایش داده می شوند باشد. اما اینکه چگونه کدهای مخرب در منابع یاد شده ذخیره میشوند؟

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

وجود نداشته باشد و نوشته های کاربران که می تواند شامل کدهای مخرب باشد مستقیماً در دیتابیس ذخیره شده و بدون هیچ اعتبار سنجی نمایش داده شود چه اتفاقی رخ خواهد داد؟ مسلماً با بازدید صفحه آلوده شده، کدهای مخرب بر روی مرورگر شما اجرا و کوکی های 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 نمی شود! بلکه از ضعف امنیتی بخش هایی همچون بخش جستجو وب سایت، بخش های نمایش پیغام خطا و ... استفاده میشود ... اما به چه صورت؟

در بسیاری از سایت ها، انجمن ها و سیستم های سازمانی تحت وب، مشاهده می شود که مثلاً 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 بعضاً در بسیاری از سیستم ها این عبارت قبل از نمایش اعتبار سنجی نمی شود که آیا شامل کدهای مخرب می باشد یا خیر. همین امر سبب میشود تا اگر عبارت جستجو شامل کدهای مخرب باشد، آن ها به همراه نتیجه ی 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 می کند اجرا می شود، 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 می شود؛ 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 کافیه فرد مهاجم لینک مخرب را به هر شکلی به فرد مورد هدف بدهد (مثلاً 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 می شود)، کدها بر روی مرورگرش اجرا شده و کوکی های سایت مذکور برای مهاجم ارسال خواهد شد ... به این نوع حمله XSS، نوع انعکاسی می گویند که کاملاً از توضیحات فوق الذکر، دلیل این نامگذاری مشخص می باشد.

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 می رساند. کاربران مورد هدف کافی بود تا فقط یک ایمیل دریافتی از هکر را باز کنند تا کوکی های سایت یاهو برای هکر ارسال شده و دسترسی ایمیل های فرد قربانی برای هکر فراهم شود ... (در حال حاضر این باگ در یاهو وجود ندارد).

چگونگی جلوگیری از این آسیب پذیری

در این سری از مقالات کدهای پیرامون سرفصل ها و مثال ها 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 زبان ها و تکنولوژی های تحت وب یکسان می باشند.

خوشبختانه کتابخانه ای 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 می کند. 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 می شوند. این داستان برای Url ها هم که به صورت پیش فرض encode میشوند صدق می کند. ولی گاهی وقتی شما برای ورود اطلاعات مثلاً از یک ادیتور WYSWYG استفاده می کنید و نیاز دارید داده ها را بدون encoding رندر کنید. آنگاه به ناچار مجاب بر اعمال یک سری سیاست های خاص تر بر روی داده مورد نظر برای رندر می شوید و نمی توانید از encoding توکار فوق الذکر استفاده کنید. آنگاه این کتابخانه در اعمال سیاست های جلوگیری از بروز این آسیب پذیری می تواند برای شما مفید واقع شود.

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

این کتابخانه مجموعه ای از توابع کد کردن عبارات است که از مواردی همچون Html, XML, Url, Form, LDAP, CSS, JScript and VBScript پشتیبانی می کند. استفاده از آن بسیار ساده می باشد. کفایت 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.