

CSRF یا Cross Site Request Forgery به صورت خلاصه به این معنا است که شخص مهاجم اعمالی را توسط شما و با سطح دسترسی شما بر روی سایت انجام دهد و اطلاعات مورد نظر خود را استخراج کرده (محتویات کوکی یا سشن و امثال آن) و به هر سایتی که تمایل دارد ارسال کند. این کار عموماً با تزریق کد در صفحه صورت می‌گیرد. مثلاً ارسال تصویری پویا به شکل زیر در یک صفحه فوروم، بلاگ یا ایمیل:

```

```

شخصی که این صفحه را مشاهده می‌کند، متوجه وجود هیچگونه مشکلی نخواهد شد و مرورگر حداکثر جای خالی تصویر را به او نمایش می‌دهد. اما کدی با سطح دسترسی شخص بازدید کننده بر روی سایت اجرا خواهد شد.

روش‌های مقابله:

هر زمانیکه کار شما با یک سایت حساس به پایان رسید، log off کنید. به این صورت بجای منتظر شدن جهت به پایان رسیدن خودکار طول سشن، سشن را زودتر خاتمه داده‌اید یا برنامه نویسی‌ها نیز باید طول مدت مجاز سشن در برنامه‌های حساس را کاهش دهند. شاید بپرسید این مورد چه اهمیتی دارد؟ مرورگری که امکان اجازتهای بازکردن چندین سایت با هم را به شما در tab های مختلف می‌دهد، ممکن است سشن یک سایت را در برگه‌ای دیگر به سایت مهاجم ارسال کند. بنابراین زمانیکه به یک سایت حساس لاگین کرده‌اید، سایت‌های دیگر را مرور نکنید. البته مرورگرهای جدید مقاوم به این مسایل شده‌اند ولی جانب احتیاط را باید رعایت کرد.

برای نمونه افزونه‌ای مخصوص فایرفاکس جهت مقابله با این منظور در آدرس زیر قابل دریافت است:

[CSRF Protector](#)

در برنامه خود قسمت Referrer header را بررسی کنید. آیا متد POST رسیده، از سایت شما صادر شده است یا اینکه صفحه‌ای دیگر در سایتی دیگر جعل شده و به برنامه شما ارسال شده است؟ هر چند این روش آنچنان قوی نیست و فایروال‌های جدید یا حتی بعضی از مرورگرها با افزونه‌هایی ویژه، امکان عدم ارسال این قسمت از header درخواست را میسر می‌سازند.

برنامه نویسی‌ها نباید مقادیر حساس را از طریق GET requests ارسال کنند. استفاده از روش POST نیز به تنهایی کارآمد نیست و آن‌را باید با random tokens ترکیب کرد تا امکان جعل درخواست منتفی شود. برای مثال استفاده از ViewStateUserKey در ASP.Net . جهت خودکار سازی اعمال این موارد در ASP.Net، اخیراً HTTP مازول زیر ارائه شده است:

[AntiCSRF - A Cross Site Request Forgery \(CSRF\) module for ASP.NET](#)

تنها کافی است که فایل dll آن در دایرکتوری bin پروژه شما قرار گیرد و در وب کانفیگ برنامه ارجاعی به این مازول را لحاظ نمائید.

کاری که این نوع مازول‌ها انجام می‌دهند افزودن نشانه‌هایی اتفاقی (random tokens) به صفحه است که مرورگر آن‌ها را بخاطر نمی‌سپارد و این token به ازای هر سشن و صفحه منحصر بفرد خواهد بود.

برای PHP نیز چنین تلاش‌هایی صورت گرفته است:

[/http://csrf.htmlpurifier.org](http://csrf.htmlpurifier.org)

مراجعی برای مطالعه بیشتر

[Prevent Cross-Site Request Forgery \(CSRF\) using ASP.NET MVC's AntiForgeryToken\(\) helper](#)

[Cross-site request forgery](#)

[Top 10 2007-Cross Site Request Forgery](#)

[CSRF - An underestimated attack method](#)

نظرات خوانندگان

نویسنده: نیما

تاریخ: ۱۳۸۷/۱۱/۰۲ ۱۳:۳۶:۰۰

سلام استاد نصیری
راستشو بگم از مطلبی که گذاشتین خیلی سر درنیوردم.
الان در این کد: `

نویسنده: محسن د.

تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۹:۴۰

سلام؛
توی توضیحات مربوط به مازول این خط رو متوجه نشدم
must ensure your GET requests are idempotent (i.e. the side-effects of multiple identical requests are the same
(as for a single request

این یعنی چی ؟

ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۶/۲۵ ۲۰:۱۰

ماژول معرفی شده فقط درخواست‌هایی از نوع Post رو محافظت می‌کنه و برای سایر درخواست‌ها (مثلا درخواست‌های Get)
کاربرد ندارد.

نویسنده: علی یگانه مقدم

تاریخ: ۲۳:۵ ۱۳۹۳/۱۱/۰۷

ممنون بابت این مطلب مفید
سوالی که دارم اینکه اگر این ماژول بالا استفاده بشه ما رو از استفاده ی Html.AntiForgeryToken بی نیاز می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۹ ۱۳۹۳/۱۱/۰۷

این مطلب برای وب فرم‌ها تهیه شد. در ASP.NET MVC نیازی به آن نیست.