عنوان: افزودن هدرهای Content Security Policy به برنامه‌های ASP.NET
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۹ ۱۳:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security

مرورگرهای جدید تحت زیر مجموعه‌ای به نام Content Security Policy، قابلیت‌های توکاری را اضافه کرده‌اند تا حملاتی مانند XSS را حتی در برنامه‌ی وبی که برای این نوع حملات تمهیداتی را درنظر نگرفته‌است، خنثی کنند. این قابلیت‌ها به صورت پیش فرض فعال نبوده و نیاز است برنامه نویس صراحتا درخواست فعال شدن آن‌ها را از طریق افزودن تعدادی هدر مشخص به Response، ارائه دهد. در ادامه این هدرها را بررسی خواهیم کرد.

غیرفعال کردن اجرای اسکریپت‌های inline

عمده‌ی حملات XSS زمانی قابلیت اجرا پیدا می‌کنند که مهاجم بتواند به طریقی (ورودی‌های اعتبارسنجی نشده)، اسکریپتی را به درون صفحه‌ی جاری تزریق کند. بنابراین اگر ما به مرورگر اعلام کنیم که دیگر اسکریپت‌های inline را پردازش نکن، سایت را تا حد زیادی در مقابل حملات XSS مقاوم کرده‌ایم. این قابلیت به صورت پیش فرض خاموش است؛ چون به طور قطع فعال سازی آن بسیاری از سایت‌هایی را که عادت کرده‌اند اسکریپت‌های خود را داخل صفحات وب مدفون کنند، از کار می‌اندازد. این نوع سایت‌ها باید به روز شده و اسکریپت‌ها را از طریق فایل‌های خارجی js، به سایت و صفحات خود الحاق کنند.
برای فعال سازی این قابلیت، فقط کافی است هدرهای زیر به Response اضافه شوند:

```
Content-Security-Policy: script-src 'self'
X-WebKit-CSP: script-src 'self'
X-Content-Security-Policy: script-src 'self'
```

سطر اول به زودی تبدیل به یک استاندارد W3 خواهد شد؛ اما فعلا فقط توسط کروم 25 به بعد پشتیبانی می‌شود. سطر دوم توسط مرورگرهایی که از موتور WebKit استفاده می‌کنند، پشتیبانی می‌شود و سطر سوم مخصوص فایرفاکس است و IE 10 به بعد.
بعد از فعال شدن این قابلیت، فقط اسکریپت‌هایی که از طریق دومین شما به صفحه الحاق شده‌اند، قابلیت اجرا را خواهند یافت و کلیه اسکریپت‌های مدفون شده داخل صفحات، دیگر اجرا نخواهد شد. در این حالت اگر از CDN برای الحاق اسکریپتی استفاده می‌کنید، مثلا مانند الحاق jQuery به صفحه، نیاز است مسیر آن‌را صراحتا در این هدر ذکر کنید:

```
Content-Security-Policy: script-src 'self' https://youcdn.com
X-WebKit-CSP: script-src 'self' https://yourcdn.com
X-Content-Security-Policy: script-src 'self' https://yourcdn.com
```

علاوه بر آن حتی می‌شود پردازش تمام منابع مورد استفاده را نیز مانند تصاویر، شیوه‌نامه‌ها، فایل‌های فلش و غیره، به دومین جاری محدود کرد:

```
Content-Security-Policy: default-src 'self' https://youcdn.com
X-WebKit-CSP: default-src 'self' https://yourcdn.com
X-Content-Security-Policy: default-src 'self' https://yourcdn.com
```

بدیهی است پس از آشنایی با این مورد، احتمالا در پروژه‌های جدید خود از آن استفاده کنید (چون inlineهای فعلی شما را کاملا از کار می‌اندازد).

نحوه‌ی اضافه کردن هدرهای Content Security Policy به برنامه‌های ASP.NET

روشی که با هر دو برنامه‌های وب فرم و MVC کار می‌کند، تهیه یک HTTP module است؛ به شرح ذیل:

```
using System;
using System.Web;

namespace AntiXssHeaders
{
```

```
    public class SecurityHeadersConstants
    {
        public static readonly string XXssProtectionHeader = "X-XSS-Protection";
        public static readonly string XFrameOptionsHeader = "X-Frame-Options";
        public static readonly string XWebKitCspHeader = "X-WebKit-CSP";
        public static readonly string XContentSecurityPolicyHeader = "X-Content-Security-Policy";
        public static readonly string ContentSecurityPolicyHeader = "Content-Security-Policy";
        public static readonly string XContentTypeOptionsHeader = "X-Content-Type-Options";
    }

    public class ContentSecurityPolicyModule : IHttpModule
    {
        public void Dispose()
        { }

        public void Init(HttpApplication app)
        {
            app.BeginRequest += AppBeginRequest;
        }

        void AppBeginRequest(object sender, EventArgs e)
        {
            var app = (HttpApplication)sender;
            var response = app.Context.Response;
            setHeaders(response);
        }

        private static void setHeaders(HttpResponse response)
        {
            response.Headers.Set(SecurityHeadersConstants.XFrameOptionsHeader, "SameOrigin");

            // For IE 8+
            response.Headers.Set(SecurityHeadersConstants.XXssProtectionHeader, "1; mode=block");
            response.Headers.Set(SecurityHeadersConstants.XContentTypeOptionsHeader, "nosniff");

            //todo: Add /Home/Report --> public JsonResult Report() { return Json(true); }

            const string cspValue = "default-src 'self';";
            // For Chrome 16+
            response.Headers.Set(SecurityHeadersConstants.XWebKitCspHeader, cspValue);

            // For Firefox 4+
            response.Headers.Set(SecurityHeadersConstants.XContentSecurityPolicyHeader, cspValue);
            response.Headers.Set(SecurityHeadersConstants.ContentSecurityPolicyHeader, cspValue);
        }
    }
}
```

و یا در برنامه‌های ASP.NET MVC می‌توان یک فیلتر جدید را تعریف کرد و سپس آن‌را به صورت عمومی معرفی نمود:

```
//// RegisterGlobalFilters -> filters.Add(new ContentSecurityPolicyFilterAttribute());
public class ContentSecurityPolicyFilterAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var response = filterContext.HttpContext.Response;
        response.AddHeader("Content-Security-Policy", "script-src 'self'");
        // the rest ...
        base.OnActionExecuting(filterContext);
    }
}
```

در ماژول تهیه شده چند مورد دیگر را نیز مشاهده می‌کنید:

الف) X-XSS-Protection مربوط است به IE 8 به بعد

ب) تنظیم هدر X-Frame-Options به SameOrigin سبب می‌شود تا صفحات سایت شما دیگر توسط Iframeها در سایت‌های دیگر قابل نمایش نباشد و فقط در سایت جاری بتوان صفحه‌ای را از همان دومین در صورت نیاز توسط Iframeها نمایش داد.

ج) تنظیم X-Content-Type-Options به nosniff سبب می‌شود تا IE سعی نکند با اجرای یک محتوا سعی در تشخیص mime-type آن کند و به این ترتیب امنیت دسترسی و مشاهده اشیاء قرار گرفته در صفحه (و یا تزریق شده توسط مهاجمین) به شدت بالا خواهد رفت.

برای مطالعه بیشتر

Security through HTTP response headers

پروژه‌ی کاملی مخصوص افزودن هدرهای یاد شده

https://nwebsec.codeplex.com/

یک نکته تکمیلی

توصیه شده‌است تا دیگر از روال رویدادگردان PreSendRequestHeaders برای ارسال هدرها استفاده نکنید؛ چون با پردازش‌های غیرهمزمان تداخل ایجاد می‌کند.

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۹/۳۰ ۹:۲۳

آیا اگر اجرای Inline غیر فعال شود و پردازش تمام منابع مورد استفاده به دامین جاری محدود شود مهاجم نمی‌تواند یک ارجاع به اسکریپت مخرب خود به صفحه اضافه کند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۹:۲۷

اجرا نمی‌شود؛ مگر اینکه به هدر ارسالی مانند CDN ذکر شده در متن، آدرس آن اضافه و صراحتا مجوز استفاده از دومینی دیگر توسط برنامه صادر شود.

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۲:۲۲

یعنی کلیه اسکری‌ت‌ها حتی اسکری‌ت هایی مانند :

```
<script type="text/javascript">
        $(document).ready(function () {
            PopupForm.ShowForm({
                renderFormUrl : "/postreply/renderreplyform",
                .....
          });
        });
</script>
```

اجازه‌ی اجرا ندارند؟ و باید در یک فایل جدا به صفحه تزریق شوند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۳:۱۸

بله. این مثال را در کروم اجرا کنید:

AntiXssHeaders.zip

در صفحه اول آن

```
<script type="text/javascript">
        alert('test');
    </script>
```

درج شده

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۳:۳۷

ممنون از مثالتون. یک پروژه فروشگاه دارم که داخل هر صفحه‌ی آن پر است از اسکریپت‌های به قول شما مدفون شده. حتی فکر پیدا کردن و منتقل کردن هر کدوم از اونها به داخل یک فایل خارجی وحشت آور است.

نویسنده: رسول آذری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۳:۴۷

با سلام . تشکر

من با مرورگر ie9 تست کردم . ولی کد اجرا شد. مگه نگفتین واسه ie8 به بعد؟
و اینکه بعد از استفاده از این روش چطور میتونم از فایل‌های js و jquery خودم توی برنامه استفاده کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۴:۱۸

- رفتار IE در مورد X-XSS-Protection متفاوت است. یعنی باید واقعا تشخیص بدهد که اسکریپت در حال اجرا یک حمله محسوب می‌شود. یک مثال دریافت کوکی برای امتحان
- مانند قبل. در متن ذکر شده: « سایت‌ها باید به روز شده و اسکریپت‌ها را *از طریق فایل‌های خارجی js* ، به سایت و صفحات خود الحاق کنند »

نویسنده: رسول آذری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۴:۳۵

ممنون از پاسخ تون.
یه سری از کدها را میشه توی فایل خارجی قرار داد.
ولی کدهای جاوایی که گریدویو تولید میکنه،چی کار میشه کرد؟
با استفاده از این روش رویدادهایی مثل SelectedIndexChanged در dropdownlist ،
RowCommand در گریدویو ، updatepanel و scriptmanager از کار می‌افته.آیا راهی برای این مشکل وجود داره؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۵:۲

- به ASP.NET MVC کوچ کنید تا کنترل کاملی بر روی عناصر اضافه شونده به صفحه داشته باشید.
- با استفاده از jQuery به هر عنصری در صفحه می‌توان رویدادهای خاصی را انتساب داد یا حذف کرد. انجام اینکار از طریق یک فایل js الحاق شده به صفحه نیز میسر است.