عنوان: **محدود کردن کاربرها به آپلود فایل‌هایی خاص در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۶ ۰:۱
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security, ASP.Net MVC

یک ادیتور آنلاین را تصور کنید که کاربران در قسمت ارسال تصویر آن قرار است فقط فایل‌های jpg، png و gif ارسال کنند و نه مثلا فایل test.aspx و موارد مشابه. در اینجا برای محدود کردن نوع فایل‌های آپلود شده می‌توان از فیلترهای سفارشی ASP.NET MVC کمک گرفت:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web.Mvc;

namespace SecurityModule
{
    public class AllowUploadSpecialFilesOnlyAttribute : ActionFilterAttribute
    {
        readonly List<string> _toFilter = new List<string>();
        readonly string _extensionsWhiteList;
        public AllowUploadSpecialFilesOnlyAttribute(string extensionsWhiteList)
        {
            if (string.IsNullOrWhiteSpace(extensionsWhiteList))
                throw new ArgumentNullException("extensionsWhiteList");

            _extensionsWhiteList = extensionsWhiteList;
            var extensions = extensionsWhiteList.Split(',');
            foreach (var ext in extensions.Where(ext => !string.IsNullOrWhiteSpace(ext)))
            {
                _toFilter.Add(ext.ToLowerInvariant().Trim());
            }
        }

        bool canUpload(string fileName)
        {
            if (string.IsNullOrWhiteSpace(fileName)) return false;

            var ext = Path.GetExtension(fileName.ToLowerInvariant());
            return _toFilter.Contains(ext);
        }

        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            var files = filterContext.HttpContext.Request.Files;
            foreach (string file in files)
            {
                var postedFile = files[file];
                if (postedFile == null || postedFile.ContentLength == 0) continue;

                if (!canUpload(postedFile.FileName))
                    throw new InvalidOperationException(
                        string.Format("You are not allowed to upload {0} file. Please upload only these
files: {1}.",
                                      Path.GetFileName(postedFile.FileName),
                                      _extensionsWhiteList));
            }

            base.OnActionExecuting(filterContext);
        }
    }
}
```

توضیحات کدهای فوق:

برای تهیه فیلتر محدود سازی نوع فایل‌های قابل ارسال به سرور، با ارث بری از ActionFilterAttribute شروع خواهیم کرد.
سپس با تحریف متد OnActionExecuting آن، توسط filterContext.HttpContext.Request.Files می‌توان به کلیه فایل‌های درحال ارسال به سرور در طی درخواست جاری، دسترسی یافت.
به این ترتیب از طریق مقدار خاصیت postedFile.FileName می‌توان به پسوند فایل در حال ارسال رسید و بر این اساس امکان

ارسال فایل‌های غیرمجاز را در نیمه راه با صدور یک استثناء سلب کرد.

برای استفاده از این فیلتر سفارشی تهیه شده نیز می‌توان به نحو زیر عمل کرد:

```
[AllowUploadSpecialFilesOnly(".jpg,.gif,.png")]
public ActionResult ImageUpload(HttpPostedFileBase file)
```

در اینجا پسوند فایل‌های مجاز قابل ارسال، توسط یک کاما از هم جدا خواهند شد.

یک نکته تکمیلی:

اگر کاربر قرار است تنها تصویر ارسال کند، بررسی پسوند فایل لازم است اما کافی نیست. برای این منظور می‌توان از کلاس Image واقع شده در فضای نام System.Drawing نیز کمک گرفت:

```
public static bool IsImageFile(HttpPostedFileBase photoFile)
{
    using (var img = Image.FromStream(photoFile.InputStream))
    {
        return img.Width > 0;
    }
}
```

در اینجا اگر فایل ارسالی تصویر نباشد، به صورت خودکار یک استثناء صادر خواهد شد.

نظرات خوانندگان

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۴/۱۷ ۱۴:۵

سلام
همانطور که گفتید چک کردن پسوند فایل الزامی است ولی کافی نیست.برای همین از کلاس Image استفاده کردید که اگر فایل ارسال شده یک فایل تصویری نبود استثنایی صادر شود. حالا برای فایل‌ها با فرمت‌های دیگه چیکار کنیم؟ (مثلا ممکن است بخواهیم به کاربر اجازه ارسال فایل‌های zip و یا rar بدهیم و مواردی از این دست).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۷ ۱۴:۱۶

برای بررسی محتوایی خاص، نیاز به parser مخصوص این نوع فایل‌ها است تا بتواند بررسی کند محتوای دریافتی معتبر است یا نه و عموما کسی محتوای این نوع فایل‌ها را بررسی نمی‌کند. بنابراین بررسی پسوند در اکثر موارد کافی است.
مشکل این نیست که فایل rar واقعا rar است یا نه. مشکل این است که این فایل ارسالی، قابلیت اجرا را از طریق فراخوانی آدرس آن در مرورگر، نداشته باشد. بحث بررسی پسوند هم به همین دلیل است. فایل aspx را می‌شود از طریق فراخوانی در مرورگر بر روی سرور اجرا کرد ولی فایلی با پسوند rar این قابلیت را ندارد.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۵/۰۹ ۲:۱

ممنون ، استثنایی که صادر می‌شود را چطور میشه ؟ در بدنه‌ی Action مربوطه مدیریت کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۹ ۸:۲۳

لازم نیست مدیریت کنید. هدف این بوده که نظم جاری را تغییر دهد. این استثناء توسط ELmah دریافت و ثبت خواهد شد. همچنین کاربر به یکی از صفحات پیش فرض خطای برنامه هدایت می‌شود و متوجه خواهد شد که خطایی رخ داده است.
ولی در کل می‌شود IExceptionFilter را نیز پیاده سازی و مدیریت کرد:

```
public class CustomFilter : FilterAttribute, IExceptionFilter
{
    public void OnException(ExceptionContext filterContext)
    {
```

نویسنده: رضوی
تاریخ: ۱۳۹۱/۰۵/۲۸ ۸:۵۳

سلام

چند نکته تکمیلی برای امنیت بیشتر
-1 بهتر است که فایل روی دیتابیس ذخیره شود، در غیر اینصورت نام فایل و پسوند توسط سیستم تعیین شود.
-2 نام و پسوند ارسالی در نظر گرفته نشود.
-3 با اندازه گیری سایز فایل GIF نمیتوان به معتبر بودن آن اطمینان داشت.

http://www.pyxsoft.com/en/hacking-hackers.html
/http://www.exploit-db.com/exploits/16181

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۷ ۱۱:۱۶

یک نکته‌ی تکمیلی

نحوه‌ی فقط خواندنی کردنی دسترسی به فایل‌های یک پوشه (گرفتن دسترسی اجرا)

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