عنوان: **CAPTCHAfa**
نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۲:۴۶
آدرس: www.dotnettips.info
گروه‌ها: CAPTCHA, reCAPTCHA, CAPTCHAfa, ASP.Net, MVC, ASP.Net MVC

حتماً با CAPTCHA آشنا هستید. فرایندی که در طی آن متنی نمایش داده می‌شود که عمدتاً فقط یک انسان قادر به درک و پاسخگویی به آن است. با این کار از ارسال داده‌های بیهوده توسط ربات‌ها جلوگیری می‌شود.
reCAPTCHA ایده‌ای است که با نمایش کلمات واقعی و اسکن شده از کتاب‌های قدیمی، بخشی از مشکلات را حل کرده و از کاربران اینترنت برای شناسایی کلماتی که رایانه توانایی خواندن آن‌ها را ندارد استفاده می‌کند. (شکل زیر)

با وارد کردن درست هر کلمه، بخشی از یک کتاب، روزنامه، و یا مجله‌ی قدیمی در رایانه شناسایی و به فرمت دیجیتال ذخیره می‌شود. به این شکل شما در دیجیتالی کردن متون کاغذی سهیم هستید.
پروژه‌ی reCAPTCHA توسط گوگل حمایت می‌شود و در این آدرس قرار دارد.
به تازگی تیمی از دانشکده فنی دانشگاه تهران به همراه انستیتو تکنولوژی ایلینویز شیکاگو، پروژه ای بر همین اساس اما برای متون فارسی با عنوان CAPTCHAfa تولید کرده اند (شکل زیر) که در این آدرس در دسترس است. امیدوارم این پروژه به گونه ای تغییر کنه که برای دیجیتالی کردن متن‌های پارسی استفاده بشه. در حال حاضر، این پروژه از کلماتی از پیش تعریف شده استفاده می‌کنه.

متاسفانه این پروژه در حال حاضر فقط توسط برنامه‌های PHP قابل استفاده است. از این رو بر آن شدم تا اون رو برای برنامه‌های ASP.NET (هم Web Forms و هم MVC) آماده کنم. برای استفاده از CAPTCHAfa نیاز به یک کلید خصوصی و یک کلید عمومی دارید که از این آدرس قابل دریافت است.
کدهای پروژه‌ی Class Library به شرح زیر است.

```
// =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
//                    Captchafa demo for ASP.NET applications
//                                 by: Behrouz Rad
// =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
namespace Captcha
{
    using System;
    using System.Collections.Generic;
    using System.IO;
    using System.Net;
    using System.Text;
    using System.Web;

    public class Captchafa
    {
        private static readonly string PRIVATE_KEY = "your private key";
        private static readonly string PUBLIC_KEY = "your public key";
        private static readonly string CAPTCHAFA_API_SERVER = "http://www.captchafa.com/api";
        private static readonly string CAPTCHAFA_VERIFY_SERVER =
"http://www.captchafa.com/api/verify/";

        private IDictionary<string, string> CaptchafaData
        {
            get
            {
                HttpContext httpContext = HttpContext.Current;

                string remoteIp = httpContext.Request.ServerVariables["REMOTE_ADDR"];
                string challenge = httpContext.Request.Form["captchafa_challenge_field"];
                string response = httpContext.Request.Form["captchafa_response_field"];

                IDictionary<string, string> data = new Dictionary<string, string>()
                {
                    {"privatekey" , PRIVATE_KEY },
                    {"remoteip"   , remoteIp    },
                    {"challenge"  , challenge   },
                    {"response"   , response    }
                };

                return data;
            }
        }

        public static string CaptchafaGetHtml()
        {
            return string.Format("<script type=\"text/javascript\"
src=\"{0}/?challenge&k={1}\"></script>", CAPTCHAFA_API_SERVER, PUBLIC_KEY);
        }

        public bool IsAnswerCorrect()
        {
            string dataToSend = this.CaptchafaPrepareDataToSend(this.CaptchafaData);

            string result = this.CaptchafaPostResponse(dataToSend);

            return result.StartsWith("true");
        }

        private string CaptchafaPrepareDataToSend(IDictionary<string, string> data)
        {
            string result = string.Empty;
            StringBuilder sb = new StringBuilder();

            foreach (var item in data)
            {
                sb.AppendFormat("{0}={1}&", item.Key, HttpUtility.UrlEncode(item.Value.Replace(@"\",
string.Empty)));
            }

            result = sb.ToString();

            sb = null;

            result = result.Substring(0, result.LastIndexOf("&"));

            return result;
        }

        private string CaptchafaPostResponse(string data)
        {
            StreamReader reader = null;
```

```
            Stream dataStream = null;
            WebResponse response = null;
            string responseFromServer = string.Empty;

            try
            {
                WebRequest request = WebRequest.Create(CAPTCHAFA_VERIFY_SERVER);

                request.Method = "POST";
                request.ContentType = "application/x-www-form-urlencoded";

                byte[] byteData = Encoding.UTF8.GetBytes(data);

                request.ContentLength = byteData.Length;

                dataStream = request.GetRequestStream();

                dataStream.Write(byteData, 0, byteData.Length);

                dataStream.Close();

                response = request.GetResponse();

                dataStream = response.GetResponseStream();

                reader = new StreamReader(dataStream);

                responseFromServer = reader.ReadToEnd();
            }
            finally
            {
                if (reader != null)
                {
                    reader.Close();
                }

                if (dataStream != null)
                {
                    dataStream.Close();
                }

                if (response != null)
                {
                    response.Close();
                }
            }

            return responseFromServer;
        }
    }
}
```

استفاده در پروژه‌های ASP.NET Web Forms

ابتدا ارجاعی به فایل Captchafa.dll ایجاد کنید و سپس در روال Page_Load، کد زیر را قرار دهید. این کار برای تزریق اسکریپ CAPTCHAfa به صفحه استفاده می‌شود.

```
if (!IsPostBack)
{
    litCaptcha.Text = Captchafa.CaptchafaGetHtml();
}
```

litCaptcha، یک کنترل Literal است که اسکریپت تولید شده، به عنوان متن آن معرفی می‌شود.

بررسی صحت مقدار وارد شده توسط کاربر (مثلاً در روال Click یک دکمه) به صورت زیر است.

```
Captchafa captchaFa = new Captchafa();

bool isAnswerCorrect = captchaFa.IsAnswerCorrect();

if (isAnswerCorrect)
{
    // پاسخ صحیح است
}
else
{
   // پاسخ صحیح نیست
```

```
}
```

استفاده در پروژه‌های ASP.NET MVC

ابتدا ارجاعی به فایل Captchafa.dll ایجاد کنید. در ASP.NET MVC بهتره تا فرایند کار رو در یک HTML helper کپسوله کنیم.

```
public static class CaptchaHelper
{
    public static MvcHtmlString Captchafa(this HtmlHelper htmlHelper)
    {
        return MvcHtmlString.Create(Captcha.Captchafa.CaptchafaGetHtml());
    }
}
```

بررسی صحت مقدار وارد شده توسط کاربر (پس از ارسال فرم به Server) به صورت زیر است.

```
[HttpPost]
[ActionName("Index")]
public ViewResult CaptchaCheck()
{
    Captchafa captchaFa = new Captchafa();

    bool isAnswerCorrect = captchaFa.IsAnswerCorrect();

    if (isAnswerCorrect)
    {
        ViewBag.IsAnswerCorrect = true;
    }
    else
    {
        ViewBag.IsAnswerCorrect = false;
    }

    return View();
}
```

و در نهایت، کدهای View (از سینتکس موتور Razor استفاده شده است).

```
@using CaptchafaDemoMvc.Helpers;

@{
    ViewBag.Title = "Index";
}

<form action="/" method="post">

@Html.Captchafa();

<input type="submit" id="btnCaptchafa" name="btnCaptchafa" value="آزمایش" />

@{
    bool isAnswerExists = ViewBag.IsAnswerCorrect != null;
}

@if (isAnswerExists)
{
    if ((bool)ViewBag.IsAnswerCorrect == true)
    {
       <span id="lblResult">پاسخ صحیح است</span>
    }
    else
    {
       <span id="lblResult">پاسخ صحیح نیست</span>
    }
}
</form>
```

دموی پروژه رو در این آدرس قرار دادم. پروژه‌ی نمونه نیز از این آدرس قابل دریافت است.

پ.ن: به زودی برخی بهبودها رو بر روی این پروژه انجام میدم.

نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۹

اول دیدم هر چی تست میکنم میگه صحیح نیست! ... بعد دیدم باید فاصله باشه بین دو کلمه که صحیح است! در صورتی که تو recaptcha اصلی فاضله اهمیت نداره! ... چرا اینجوریه؟!

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۳۸

چون در زبان فارسی، حروف وقتی به هم بچسبن، شکلشون تغییر می‌کنه. بنابراین طبیعیه که "مسلسل" با "مسل سل" فرق کنه.

نویسنده: ایمان نعمتی
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۴۰

کار جالبی کردی بهروز جان
اما به نظر میرسه خود پروژه خیلی جالب نباشه
اینکه از یه سری کلمات از پیش تعیین شده استفاده میکنه و برای نمایش هم فقط دوتا خط به شکل بیضی روی کلمات میندازه زیاد جالب نیست به نظرم. شبیه به ری‌کپتچا خواسته کار کنه اما خب چیز خوبی از کار درنیامده

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۵۲

برای شروع، کار بسیار ارزشمندیه. ضمن اینکه تنوع کلمات مورد استفاده بسیار بالاست. خیلی مشتاقم که زودتر برای دیجیتالی کردن متون فارسی بشه ازش استفاده کرد، اما نیاز به یک متولی قوی و پشتیبانی خوب داره. مثل کاری که سایت گنجور انجام میده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۵۳

نوشتن مشابه آن با دات نت زیاد سخت نیست. برای نمونه « بررسی نحوه برنامه نویسی سایت نستعلیق آنلاین » می‌تونه ایده خوبی برای شروع باشه. من این مطلب رو به همراه مطلب « تبدیل عدد به حروف » کنار هم گذاشتم و یک captcha فارسی ازش تهیه کردم.

نویسنده: ایمان نعمتی
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۹:۵۳

چه جالب!
امیدوارم استفاده از کپتچای فارسی در سایت‌های فارسی مرسوم بشه

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۵/۱۶ ۹:۵۶

با سلام. جهت حل مشکل کلمات از پیش تعریف شده، می‌توان از Google Translate استفاده کرد.
متاسفانه هنوز فرصت نکردم تا جزییات را دقیق ببینم، اما امیدوارم تلفظ و پخش کلمات بصورت صوتی خوب عمل کند!

نکته دیگر اینکه در مورد پیچیدگی تصویر CAPTCHA، بهترین روش استفاده از نقطه و پاره خط جهت تشویش تصویر است. معمولاً

بیشترین خطاها در OCR مربوط به این دو شکل پایه‌ی هندسی هستند. البته این موضوع بیشتر در مورد کلمات لاتین مطرح است.
شاید در مورد کلمات فارسی بیضی هایی که دوستان روی متن انداختند پیشچیدگی را افزایش دهد. آقای ایمان نعمتی لطفاً فراموش نکنیم که هدف از تشویش تصویر، سختی کار برای انسان نیست بلکه پیچیده کردن کار روبات است.
در مجموعه بسیار موضوع خوبی است. اگر دوستانی علاقه به این موضوع داشته باشند ایده و مطلب بسیار خوبی وجود دارد، هرچند پیاده سازی نشده است اما فکر می‌کنم ارزش انتشار و مطالعه را داشته باشد. البته جهت پیاده سازی آن چالشی بر سر راه نیست و فقط کمی وقت آزاد و البته تخصص نیاز دارد.

این مطلب را در این سایت میشه منتشر کرد؟ (با توجه به اینکه هنوز پیاده سازی ندارد و صرفاً بیان الگوریتم است.)

نویسنده: هوشنگ
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۱:۱۱

"با وارد کردن درست هر کلمه، بخشی از یک کتاب، روزنامه، و یا مجله‌ی قدیمی در رایانه شناسایی و به فرمت دیجیتال ذخیره می‌شود. به این شکل شما در دیجیتالی کردن متون کاغذی سهیم هستید. "

این منطقی نیست ، با همان شیوه ای که کلمه تایپ شده کاربر با عکس مقایسه میشه ، به همان طریق میشه متون کاغذی رو دیجیتالی کرد و نیازی به استفاده از این شیوه برای دیجتالی کردن نیست

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۲:۴

دوست من.
تمامی کلمات در یک متن (مخصوصاً اگر اون متن قدیمی باشه و برخی حروف یک کلمه پاک یا ناخوانا باشند) توسط رایانه و برنامه‌های OCR قابل تشخیص نیست.
مدل‌های CAPTCHAیی مانند reCAPTCHA که بر اساس دیجیتالی کردن متون پیاده سازی شدند، دو کلمه رو به کاربر نشان میدن:
1) کلمه ای که توسط رایانه به درستی تشخیص داده شده
2) کلمه یا بخشی از اون که رایانه نتونسته اون رو به درستی تشخیص بده.
در صورتی که کاربر، کلمه ای که توسط رایانه تونسته تشخیص داده بشه و برای رایانه مشخص هست رو صحیح وارد کنه، فرض بر این گرفته میشه که کلمه‌ی دوم یا بخشی از کلمه‌ی دوم که قابل تشخیص نبوده رو هم تونسته به درستی وارد کنه.
پس از این، میانگینی از تعداد عبارت ورودی کاربران برای کلمه‌ی نامشخص گرفته میشه و عبارتی که بیشترین فراوانی رو داشته به عنوان مورد قابل قبول ثبت میشه.

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۲۸ ۱۲:۵۰

کامپوننت رو آپدیت کردم. نسخه‌ی اولیه بر اساس وفاداری به کدهای PHP و برگردانی از اونها بود. در نسخه‌ی جدید، متد CaptchafaGetHtml به GetHtml تغییر پیدا کرد، مدیریت خطاها اضافه شد و کلیدهای خصوصی و عمومی باید در زمان استفاده به کامپوننت پاس داده شوند. نسخه‌ی جدید را از لینک انتهای مقاله دریافت کنید.

نویسنده: بابک
تاریخ: ۱۳۹۳/۰۸/۲۸ ۱۵:۲۳

آدرس ظاهرا به آدرس زیر تغییر کرده است:

http://www.captchafa.net

نویسنده: محمد رضا صفری

تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۱۲

فکر کنم که کلا پروژه تعطیل شده ...

عنوان: **بررسی تصویر امنیتی (Captcha) سایت - قسمت اول**
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۸/۲۸ ۲۱:۳۰
آدرس: www.dotnettips.info
برچسب‌ها: Security, CAPTCHA, CAPTCHAfa, handler

تصویر امنیتی و یا کپچا برای تشخیص و احراز انسان بودن استفاده کننده استفاده میشود و بصورت تصویری که استخراج نوشته‌های درون آن برای روبوت‌ها بسیار سخت و یا نشدنی است ایجاد میشود و دارای انواع و اقسام متفاوتی است. در این میان برای استفاده از این امکان نمونه هایی در زبانهای مختلف تهیه شده که بسته به سلیقه و نیاز مورد استفاده قرار گرفته شده است. در این مقاله قصد داریم با بررسی تصویر امنیتی که در وبلاگ کنونی استفاده شده آنرا تا حدودی بازسازی کنیم.

تصویر آشنای کاربران سایت برای ورود به قسمت مدیریت که در صفحه مورد نظر از تصویر امنیتی استفاده شده است:

با توجه به آدرس لینک تصویر مشخص است که برای تولید این تصویر از هندلر استفاده شده و با توجه به پارامترهایی که به آن داده شده تصویر مورد نظر را ایجاد میکند و با فرمت مشخصی بر میگرداند:


```
http://www.dotnettips.info
/CaptchaImage/Show?
text=H2yL5iOXIuu0dsBvu%2F405AnXkeacis3dMQ%2FHXAH8h4A%2BjwDM0f7%2FRZMHvBG5pXYJ
```

```
&foreColor=%231B0172
&fontSize=12
&fontName=Tahoma
```

پارامترهای پاس داده شده به ترتیب ذیل برای اهدافی قرار داده شده اند:

نام	مورد استفاده	مقدار
text	متنی که بصورت کد شده حاوی متن مورد استفاده تصویر امنیتی است	H2yL5iOXIuu0dsBvu%2F405AnXkeacis3dMQ%2FHXAH8h4A%2BjwDM0f7%2FRZMHvBG5pXYJ
foreColor	رنگ مربوط به نوشته‌های تصویر	%231B0172
fontSize	سایز فونت	12
fontName	نام فونت	Tahoma

بنظر اهم پارامتر مورد نیاز همان متن است که بصورت کد شده در بالا به آن اشاره شد. برای این منظور باید کلاسی تهیه شود که حاوی متدهای:

-1 انتخاب یه عدد تصادفی از بین دامنه ای که به آن داده میشود.

برای مثال یه عد از بین 100 تا 9999 که بصورت تصادفی انتخاب میشود

-2 تبدیل به معادل حرفی آن

برای مثال از عدد "7062" به حروف آن یعنی "هفت هزار و شصت و دو" استخراج شود

-3 کد کردن حرف تولید شده

حرف "هفت هزار و شصت و دو" را کد کرده و بصورت متنی شبیه

" H2yL5iOXIuu0dsBvu%2F405AnXkeacis3dMQ%2FHXAH8h4A%2BjwDM0f7%2FRZMHvBG5pXYJ"

تبدیل کند

-4 دیکد کردن و استخراج

البته این مورد با توجه به استفاده ای که ما داریم نیازی به پیاده سازی نیست

در کنار تصوریر امنیتی از یک فیلد مخفی نیز برای نگهداری مقدار کد شده برای مقایسه با ورودی کاربر استفاده شده که در قسمت بعد ضمن ایجاد نمونه کاربردی بیشتر با قسمتهای مختلف آن آشنا میشویم.

نظرات خوانندگان

نویسنده: ahmadalli
تاریخ: ۱۳۹۱/۰۸/۳۰ ۲۱:۳۹

از نظر تئوری هر کدی قابل دی‌کد شدن هست. بهتر نیست برای کد اصلی از hash استفاده کنیم و hash کدی که کاربر وارد می‌کنه رو با hash اصلی مقایسه کنیم؟

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۸/۳۰ ۲۳:۴۴

با توجه به قسمت دوم از قسمت دیکد کردن استفاده نشده ولی چون از این نوع رمزنگرای استفاده شده در مواقعی ممکن است از دیکد کردن نیز استفاده کرد ولی به هرحال این رمزگذاری رو میشه بدلخواه تغییر داد.

عنوان: **بررسی تصویر امنیتی (Captcha) سایت - قسمت دوم**
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۸/۳۰ ۲۱:۳۰
آدرس: www.dotnettips.info
گروه‌ها: Security, CAPTCHA, CAPTCHAfa, handler

در ادامه بررسی تصویر امنیتی سایت مواردی که باید پیاده سازی شود برای مورد اول میتوان کلاس زیر را در نظر گرفت که متدهایی برای تولید اعداد بصورت تصادفی در بین بازه معرفی شده را بازگشت میدهد:

```
// RandomGenerator.cs
using System;
using System.Security.Cryptography;

namespace PersianCaptchaHandler
{
    public class RandomGenerator
    {
        private static readonly byte[] Randb = new byte[4];
        private static readonly RNGCryptoServiceProvider Rand = new RNGCryptoServiceProvider();

        public static int Next()
        {
            Rand.GetBytes(Randb);
            var value = BitConverter.ToInt32(Randb, 0);
            if (value < 0) value = -value;
            return value;
        }
        public static int Next(int max)
        {
            Rand.GetBytes(Randb);
            var value = BitConverter.ToInt32(Randb, 0);
            value = value%(max + 1);
            if (value < 0) value = -value;
            return value;
        }
        public static int Next(int min, int max)
        {
            var value = Next(max - min) + min;
            return value;
        }
    }
}
```

و برای تبدیل عدد تصادفی بدست آمده به متن نیز از این کلاس میتوان استفاده کرد که به طرز ساده ای این کار را انجام میدهد:

```
// NumberToString.cs
namespace PersianCaptchaHandler
{
    public class NumberToString
    {

        #region Fields
        private static readonly string[] Yakan = new[] { "","صفر", "یک", "دو", "سه", "چهار", "پنج",
"شش", "هفت", "هشت", "نه" };
        private static readonly string[] Dahgan = new[] { "", "", "","بیست", "سی", "چهل", "پنجاه",
"شصت", "هفتاد", "هشتاد", "نود" };
        private static readonly string[] Dahyek = new [] { "","ده", "یازده", "دوازده", "سیزده", "چهارده",
"پانزده", "شانزده", "هفده", "هجده", "نوزده" };
        private static readonly string[] Sadgan = new [] { "", "","یکصد", "دوصد", "سیصد", "چهارصد",
"پانصد", "ششصد", "هفتصد", "هشتصد", "نهصد" };
        private static readonly string[] Basex = new [] { "", "هزار", "میلیون", "میلیارد", "تریلیون" };
        #endregion

        private static string Getnum3(int num3)
        {
            var s = "";
            var d12 = num3 % 100;
            var d3 = num3 / 100;
            if (d3 != 0)
                s = Sadgan[d3] + " و ";
            if ((d12 >= 10) && (d12 <= 19))
```

```
            {
                s = s + Dahyek[d12 - 10];
            }
            else
            {
                var d2 = d12 / 10;
                if (d2 != 0)
                    s = s + Dahgan[d2] + " و ";
                var d1 = d12 % 10;
                if (d1 != 0)
                    s = s + Yakan[d1] + " و ";
                s = s.Substring(0, s.Length - 3);
            }
            return s;
        }

        public static string ConvertIntNumberToFarsiAlphabatic(string snum)
        {
            var stotal = "";
            if (snum == "0") return Yakan[0];

            snum = snum.PadLeft(((snum.Length - 1) / 3 + 1) * 3, '0');
            var l = snum.Length / 3 - 1;
            for (var i = 0; i <= l; i++)
            {
                var b = int.Parse(snum.Substring(i * 3, 3));
                if (b != 0)
                    stotal = stotal + Getnum3(b) + " " + Basex[l - i] + " و ";
            }
            stotal = stotal.Substring(0, stotal.Length - 3);
            return stotal;
        }
    }
}
```

و برای کد کردن و دیکد کردن یعنی موارد سوم و چهارم مقاله قبلی، متن بدست آمده را که بعنوان قسمتی از آدرس تصویر در ادامه آدرس هندلر معرفی شده می‌آید تبدیل به یک string بی معنی برای بازدیدکننده میکند:

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace PersianCaptchaHandler
{
    public class Encryptor
    {
        #region constraints
        private static string Password { get { return "Mehdi"; } }
        private static string Salt { get { return "Payervand"; } }
        #endregion

        public static string Encrypt(string clearText)
        {
            // Turn text to bytes
            var clearBytes = Encoding.Unicode.GetBytes(clearText);

            var pdb = new PasswordDeriveBytes(Password, Encoding.Unicode.GetBytes(Salt));

            var ms = new MemoryStream();
            var alg = Rijndael.Create();

            alg.Key = pdb.GetBytes(32);
            alg.IV = pdb.GetBytes(16);

            var cs = new CryptoStream(ms, alg.CreateEncryptor(), CryptoStreamMode.Write);

            cs.Write(clearBytes, 0, clearBytes.Length);
            cs.Close();

            var encryptedData = ms.ToArray();

            return Convert.ToBase64String(encryptedData);
        }
        public static string Decrypt(string cipherText)
        {
```

```
            // Convert text to byte
            var cipherBytes = Convert.FromBase64String(cipherText);

            var pdb = new PasswordDeriveBytes(Password, Encoding.Unicode.GetBytes(Salt));

            var ms = new MemoryStream();
            var alg = Rijndael.Create();

            alg.Key = pdb.GetBytes(32);
            alg.IV = pdb.GetBytes(16);

            var cs = new CryptoStream(ms, alg.CreateDecryptor(), CryptoStreamMode.Write);

            cs.Write(cipherBytes, 0, cipherBytes.Length);
            cs.Close();

            var decryptedData = ms.ToArray();

            return Encoding.Unicode.GetString(decryptedData);
        }
    }
}
```

و نیز برای اعتبار سنجی عدد دریافتی از کاربر میتوان از عبارت با قاعده زیر استفاده کرد:

```
// Utils.cs
using System.Text.RegularExpressions;

namespace PersianCaptchaHandler
{
    public class Utils
    {
        private static readonly Regex NumberMatch = new Regex(@"^([0-9]*|\d*\.\d{1}?\d*)$",
RegexOptions.Compiled);
        public static bool IsNumber(string number2Match)
        {
            return NumberMatch.IsMatch(number2Match);
        }
    }
}
```

برای استفاده نیز کافیست که هندلر مربوط به این کتابخانه را بطریق زیر در وب کانفیگ رجیستر کرد:

```
<add verb="GET" path="/captcha/" type="PersianCaptchaHandler.CaptchaHandler, PersianCaptchaHandler,
Version=1.0.0.0, Culture=neutral"  />
```

و در صفحه مورد نظرتان بطریق زیر میتوان از یک تگ تصویر برای نمایش تصویر تولیدی و از یک فیلد مخفی برای نگهداری مقدار کد شده معادل عدد تولیدی که در هنگام مقایسه با عدد ورودی کاربر مورد نیاز است استفاده شود:

```
<!-- ASPX -->
<dl>
    <dt>تصویر امنیتی</dt>
    <dd>
        <asp:Image ID="imgCaptchaText" runat="server" AlternateText="CaptchaImage" />
        <asp:HiddenField ID="hfCaptchaText" runat="server" />
        <asp:ImageButton ID="btnRefreshCaptcha" runat="server" ImageUrl="/img/refresh.png"
            OnClick="btnRefreshCaptcha_Click" />
        <br />
        <asp:TextBox ID="txtCaptcha" runat="server" AutoCompleteType="Disabled"></asp:TextBox>
    </dd>
    <dd>
        <asp:Button ID="btnSubmit" runat="server" Text="ثبت" OnClick="btnSubmit_Click" />
    </dd>
</dl>
<asp:Label ID="lblMessage" runat="server"></asp:Label>
```

در زمان لود صفحه، تصویر امنیتی مقدار دهی میشود و در زمان ورود عدد توسط کاربر با توجه به اینکه کاربر حتما باید عدد وارد کند با عبارت با قاعده این اعتبار سنجی انجام میشود:

```
protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
                SetCaptcha();
        }

        private void SetCaptcha()
        {
            lblMessage.Text =
            txtCaptcha.Text = string.Empty;

            var newNumber =
                RandomGenerator.Next(100, 999)
                ;

            var farsiAlphabatic =
NumberToString.ConvertIntNumberToFarsiAlphabatic(newNumber.ToString());

            hfCaptchaText.Value =
                HttpUtility
                .UrlEncode(
                    Encryptor.Encrypt(
                        farsiAlphabatic
                    )
                );

            txtCaptcha.Text = string.Empty;
            imgCaptchaText.ImageUrl =
                "/captcha/?text=" + hfCaptchaText.Value;
        }
```

و بعد از ورود عدد از سمت کاربر از متد تبدیل به حروف استفاده کرده و این مقدار تولیدی با مقدار فیلد مخفی مقایسه میشود:

```
private string GetCaptcha()
        {
            var farsiAlphabatic = NumberToString.ConvertIntNumberToFarsiAlphabatic(txtCaptcha.Text);

            var encryptedString =
                HttpUtility
                .UrlEncode(
                    Encryptor.Encrypt(
                        farsiAlphabatic
                    )
                );

            return encryptedString;
        }

        private bool ValidateUserInputForLogin()
        {
            if (!Utils.IsNumber(txtCaptcha.Text))
            {
                lblMessage.Text = "تصویر امنیتی را بطور صحیح وارد نکرده اید";
                return false;
            }

            var strGetCaptcha =
                GetCaptcha();

            var strDecodedVAlue =
                hfCaptchaText.Value;

            if (strDecodedVAlue != strGetCaptcha)
            {
                lblMessage.Text = "کلمه امنیتی اشتباه است";
                SetCaptcha();
                return false;
            }
            return true;
        }
```

```
        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            if (!ValidateUserInputForLogin()) return;
            lblMessage.Text = "کلمه امنیتی درست است";
        }

        protected void btnRefreshCaptcha_Click(object sender, ImageClickEventArgs e)
        {
            SetCaptcha();
        }
```

در آخر این پروژه در کدپلکس قرار داده شده، و مشتاق نظرات و پیشنهادات شما هستم و نیز نمونه مثال بالا ضمیمه شده است

نظرات خوانندگان

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۹/۰۱ ۱۳:۲۸

تشکر از مطلب مفیدتون.
جناب نصیری کلاسی برای تبدیل عدد به حروف نوشته اند که استفاده از اون کلاس در این پروژه ، زیبایی کار را دو چندان می‌کند!

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۱ ۱۳:۳۸

پروژه در کدپلکس
پروژه در سایت

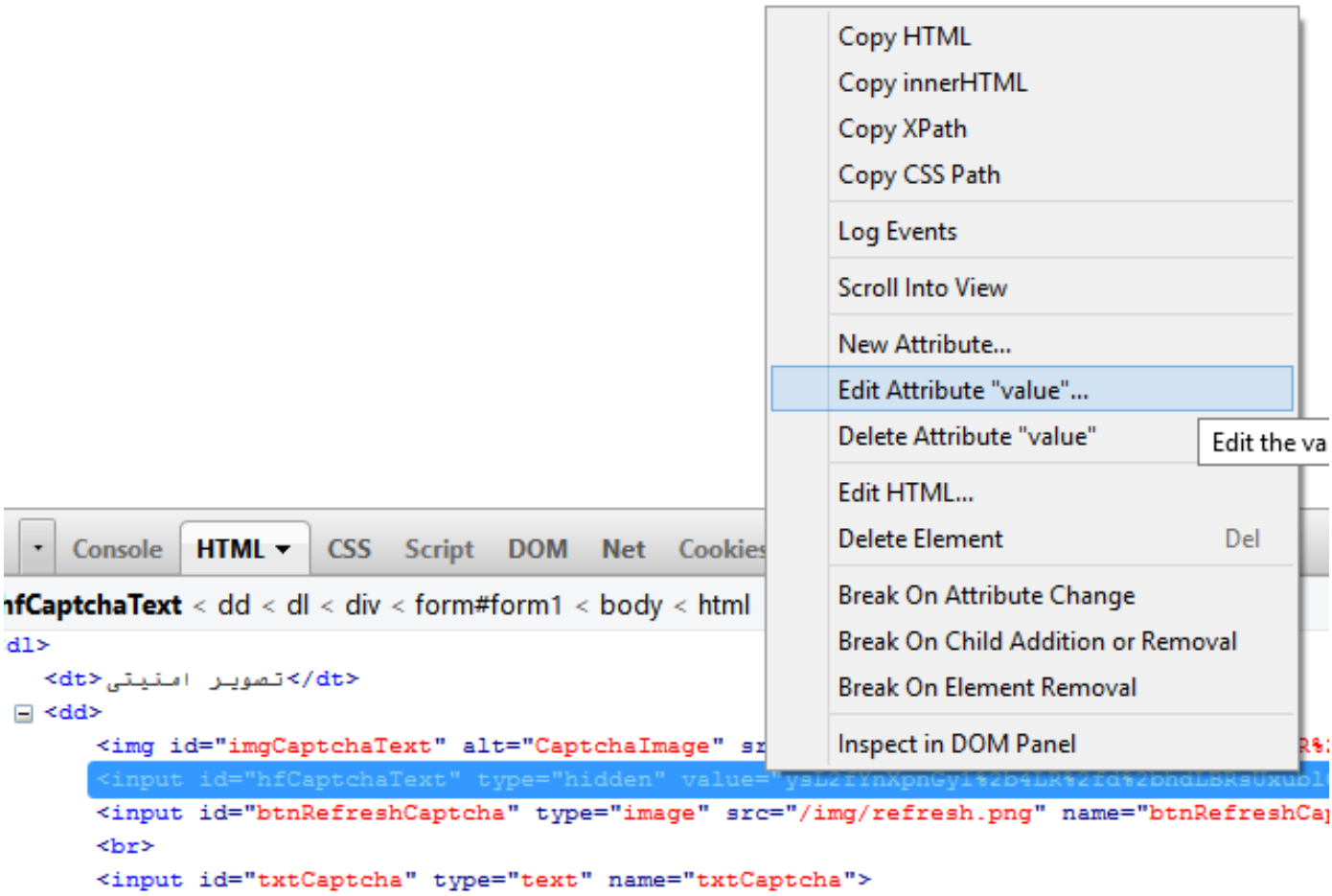
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۱ ۱۳:۴۰

بسیار ممنون، با اجازه مهندس نصیری این کار انجام میشه.
لطفا درخواست رو در قسمت مربوط به پروژه در سایت مطرح کنید تا یک تاریخچه مطلوبی برای پروژه ایجاد بشه.

نویسنده: رحمت اله رضایی
تاریخ: ۱۳۹۱/۰۹/۰۶ ۱۱:۲۹

مشکلی که این روش دارد این است که با یک بار بارگذاری صفحه، خواندن عکس توسط انسان و یادداشت متن رمز گذاری شده می‌توان مقدار اولیه عکس را هر بار با ویرایش متن رمزگذاری شده به سرور ارسال کرد و تایید گرفت! مشکلی که سایت فعلی هم دارد.

برای حل این مشکل چه پیشنهادی دارید؟



نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۶ ۱۴:۱۹

ممنون از نظرتون،
موضوع مورد نظر، قسمتی است که متن تصویر برای این کپچا در آن نگهداری میشود،
1. در صورتی که بخواهیم از توانایی‌های سمت سرور کمک گرفت: بنظرم برای اعتبار سنجی میتوان برای کابران در بانک یک کد یونیک به ازای اولین هیت کاربر در سایت ایجاد کرد و این کد را بصورت رمزنگاری شده در کلاینت(کوکی) نگهداشت و مواردی از این دست را با این کد اعتبارسنجی کرد.
2. در صورتی که بخواهیم در سمت کلاینت همه چیز هندل شود : مطمئنا بهتر است که این فیلد مخفی در کنار باکس ورودی قرار داده نشود و برای نگهداری این فیلد از جاوا اسکریپت یا کوکی و از هردو کمک گرفت به این صورت که این مقدار کد شده بصورت مقداری غیر از این و با نامی بدون ربط در متغیری که از سمت سرور مقدار دهی میشود و فقط در سرور خوانده میشود تعویض کرد.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۷ ۱۷:۵

لطفا راهنمایی کنید چطور پروژه رو دانلود کنم؟

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۷ ۱۷:۵۶

برای دمو از این لینک میتونید استفاده کنید +
و همچنین در کدپلکس persiacaptchahandler.codeplex.com
و در همین سایت هم PersianCaptchaHandler

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۲:۴۳

بهترین راه استفاده از Expiration date هستش،به این صورت که به متن رمزنگاری شده یک تاریخ انقضای 3دقیقه ای داد،بدون استفاده از دیتابیس و کوکی.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۲:۵۷

البته اگر از بازه عددی کوچکی استفاده بشه که ممکنه این تاریخ هم بلا استفاده بمونه، همچنین این تاریخ قرار هست کجا نگهداری بشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۴

در تمام قسمت‌های رمزنگاری شده سایت جاری از یک salt با طول عمر یک روز استفاده می‌شود. این salt به کلید رمزنگاری اطلاعات اضافه می‌شود. بنابراین اطلاعات رمزنگاری شده توسط آن کلید، فقط در طی روز جاری قابل رمزگشایی خواهند بود.
برای مثال لینک یک پیغام خصوصی سایت را در جایی ذخیره کنید. تا پایان روز کار می‌کند. روز بعد به صورت خودکار منقضی شده (چون salt فردا چیز دیگری است) و اطلاعات روز قبل توسط کلید جدید قابل رمزگشایی نیست. به این ترتیب شخص نیاز خواهد داشت تا لینک جدیدی را در صفحه مداخل مرتبط دریافت کند.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۵

در واقع باید متن رو مثلا شامل دو بخش،بخش اول،و بخش دوم هم تاریخ و زمان فعلی هستش که با یه splitter از هم جدا شدن و بعد رمزنگاری کرد،و در سمت سرور ابتدا بخش دوم رو با تاریخ و زمان فعلی(submit شدن فرم) مقایسه کرد و درصورتیکه اختلاف اونها بیشتر از 3 دقیقه(یا 2 یا ...) بود captcha غیر معتبر هستش.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۷

یک روز برای یک captcha زمان زیادی نیست؟

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۸

البته بنظرم برای نگهداری نکردن تاریخ در سرور میشه salt رو همین تاریخ امروز در نظر گرفت

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۱۰

این لایه اول کلی هست (در تمام قسمت‌های سایت). تست کنید ببینید مطابق تصویر بالایی که ارسال شده می‌تونید در مورد کپچای سایت جواب بگیرید یا نه ...

نویسنده: کیارش سلیمان زاده

تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۱۸

تو لایه‌های بعدی چطور تست میکنید؟
طریقه ایجاد salt رو میشه بیشتر توضیح بدید؟
چیزی رو در دیتابیس هم ذخیره می‌کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۲۱

salt مورد استفاده تاریخ ساده روز سیستم است؛ در بانک اطلاعاتی هم ذخیره نمی‌شود.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۴۴

ممنون،
"تست کنید ببینید مطابق تصویر بالایی که ارسال شده می‌تونید در مورد کپچای سایت جواب بگیرید یا نه ..."
بله تست کردم.
بعد از اینکه مشخص شد salt متعلق به امروز هستش،چطور تست میکنید که آیا مطابق با تصویر بالا،متن کپی شده یا اصلی هستش؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۴۷

در مرحله بعد از یک کش 5 دقیقه‌ای با کلیدی بر اساس IP شخص استفاده می‌کنم.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۵۴

مگه IP نسبت به ISP برای بعضی‌ها یکسان نیست؟
در این صورت مشکلی پیش نمیاد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۵۸

تا الان که کسی مشکلی گزارش نکرده.
همچنین مقدار ذخیره شده در این کش با طول عمر کوتاه 5 دقیقه‌ای به ازای هر بار ریفرش صفحه متفاوت خواهد بود.

نویسنده: رحمت اله رضایی
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۴:۱۱

هفته پیش کپچای سایت این مشکل را داشت ولی الان برطرف شده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۶:۱۲

گیرم اصلا این کپچا نباشد. با anti forgery-token می‌خواهید چکار کنید؟ با ماژول آنتی داس چطور و خیلی مسایل دیگر.
خلاصه صرف اینکه در داخل یک مرورگر به کمک یک افزونه برای مدت کوتاهی توانستید اطلاعاتی را شبیه سازی کنید به این معنا نیست که قابلیت استفاده وسیع و خارج از مرورگر را هم به سادگی دارا است (مثلا بروت فورس پسورد کاربران). تا حد شبیه سازی کامل رفتار یک مرورگر باید پیش برید (و ... اصلا کار ساده‌ای نیست).

نویسنده: رحمت اله رضایی

تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۱:۵۴

هدف من از طرح این مساله این بود که نمایش تصویر امنیتی بدون در نظر گرفتن این مورد هیچ فایده ای برای سایت نخواهد داشت و کار به ماژول آنتی داس و مشکلات مربوط به آن کشیده می‌شود (یک لایه عقب تر).
در واقع کپچا یک تصویر تزیینی می‌شود.

دستکاری اطلاعات ارسالی به سرور، فقط در اینجا مطرح نیست. عمده مورد استفاده آن در ویرایش idها (مثلا در DropDownList ها) ست. (ثبت نام مسکن برای شهری که هنوز ثبت نام آن فعال نشده، خرید بلیت جایگاه ویژه برای یک کنسرت، پرداخت بیمه با دستکاری هزینه و ...)

نویسنده: داود حنیفی
تاریخ: ۱۳۹۱/۰۹/۲۱ ۹:۵

با سلام
ببخشید این مورد تو vs2012 چطور کار میکنه؟
تست من تو vs2012 جواب نداد.
با تشکر از زحماتتون.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۲۱ ۹:۴۶

من این مثال رو با VS2012 باز کردم مشکلی نداشت

نویسنده: داود حنیفی
تاریخ: ۱۳۹۱/۰۹/۲۱ ۱۰:۳۴

ببخشید من اشتباه کردم.منظورم با (4.5 net.) بود.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۲۱ ۱۰:۵۰

با اون دات نت 4.5 هم مشکلی نداشت
اگه کارتون با نمونه پروژه تبدیل شده راه میوفته من اونو تو این لینک گذاشتم
PersianCaptcha_2012-12-11_10-51-20.rar

نویسنده: مسعود
تاریخ: ۱۳۹۱/۱۱/۲۶ ۱۶:۱۹

سلام
لازم است در ابتدا بابت این پروژه تشکر کنم.
من مشکلی با این کپچا دارم و آن این است که روی محیط تست خودم کار می‌کند اما زمانی که آن را پابلیش می‌کنم و روی سرور اصلی قرار می‌دهم، تصویر کپچا نمایش داده نمی‌شود.
لازم به ذکر است که از پروتوکل https استفاده می‌کنم.

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۶ ۱۶:۵۴

یک حدس از راه دور:

تنظیم پیش فرض این پروژه در وب کانفیگ مربوط به IIS6 است (system.web/httpHandlers). برای IIS7 تنظیم جداگانه‌ای (system.webServer/handlers) نیاز دارد.

نویسنده: مسعود
تاریخ: ۱۳۹۱/۱۱/۲۶ ۱۷:۲۳

سلام
ممنون از پاسختون.
اما IIS من ۷ است و موردی رو هم که فرمودید پیش از این چک کردم ولی تغییری حاصل نشده بود.

نویسنده: مسعود
تاریخ: ۱۳۹۱/۱۱/۲۶ ۱۷:۳۲

با راه اندازی مجدد IIS مشکل رفع شد.

ممنون

نویسنده: مرتضی مختاری
تاریخ: ۱۳۹۲/۱۰/۲۲ ۱۶:۱۶

سلام؛ بنده آخر متوجه نشدم که از کپچا به این صورت استفاده کنیم یا نه؟
استفاده از expiration time به نظر من جواب نمیده چون توسط ربات تویه عرض یک دقیقه میشه هزار تا comment ثبت کرد.فکر کنم استفاده از captcha به همین روش و استفاده از anti forgery-token هیچ مشکل امنیتی نداشته باشه. شما چطوری مشکل این روش رو برطرف کردید؟ با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۲۲ ۱۶:۲۱

نمونه بهبود یافته مطلب جاری در اینجا « نحوه ایجاد یک تصویر امنیتی (Captcha) با حروف فارسی در ASP.Net MVC »