عنوان مقاله : سنجش قدرت گذرواژه موضوع : برنامه نویسی وب و سی شارپ تاریخ انتشار: تابستان ۸۷

چکیده:

گذرواژه (رمزعبور) یکی از متداول ترین روشهای اعتبارسنجی است. قدرت گذرواژه میزان مقاومت آن در برابر حملات Brute force و sholder surfing میباشد . قدرت گذرواژه تابع فضای تنوع و تعداد کاراکترهای آن است.

گذرواژه ها از روشهای مختلف hack میشوند استفاده از keystroke logger ، روش shoulder surfing ، پیدا کردن یک گذرواژه نوشته شده ، حملات brute force ، روش phishing ، ارسال email های جعلی به منظور خواستن رمز عبور، متداولترین روشهای لو رفتن گذرواژه هستند. برای مقابله با این روشها ا ساس نیاز به آموزش داریم. با اگر کاربر دانش کافی در انتخاب و بکارگیری گذرواژه داشته باشد روشهای مشابه phishing نمیتواند موثر واقع شود اگر کاربر بداند که در صورت دریافت Email از طرف کسی که خود را administrator یا مدیر سیستم مینامد هرگز نباید گذرواژه خود را اعلام کند، خطر لو رفتن گذرواژه از این روش کاهش می یابد. استفاده از آنتی ویروس می تواند خطر اجرایی بودن برنامه های keystroke logger را کاهش دهد. با استفاده از پروتکلهای امن امکان لو رفتن گذرواژه از طریق packet sniffing از میان می رود.

برای مقابله با روش shuolder surfing (نگاه کردن شخص دیگر هنگام وارد کردن رمز عبور) باید کاربر گذرواژه را سریع تایپ کند و همچنین گذرواژه نباید توالی چند کاراکتر روی صفحه کلید باشد برنامه باید از قبول کاراکترهای متوالی به عنوان گذرواژه جلوگیری کند.

برای مقابله با brute force (امتحان کردن کلیه حالتهای ممکن) باید کابر گذرواژه خود را از مجموعه کاراکتر های حرفی عددی، حوف بزرگ و کوچک و کاراکترهای خاص انتخاب نماید با وسیع بودن فضای احتمال تعداد حالت های ممکن افزایش مییابد و الگوریتمهای brute force را غیر موثر میسازد.

محاسبات

اگر بدانیم گذرواژه عددی است چهار رقمی و دفعات تلاش ناموفق برای ورود در سامانه اعتبارسنجی نامحدود است، میتوانیم ده به توان چهار حالت یعنی ۱۰۰۰۰ حالت ممکن را امتحان کنیم این کار با استفاده از برنامه بسیار ساده خواهد بود. اگر فضای انتخاب کاربر کلیه حروف کوچک زبان انگلیسی و اعداد صفر تا نه باشد و طول گذرواژه ما شش کاراکتر باشد برای هر کاراکتر سی و شش حالت (26+10)خواهیم داشت لذا طبق اصل ضرب ۳۶ به توان ۶ طول فضای نمونه خواهد بود که برابر است با ۲۱۷۶۷۸۲۳۳۶ که یک عدد ده رقمی است. اگر کاربر امکان استفاده از حروف بزرگ و کاراکترهای ویژه را داشته باشد این فضا بیشتر خواهد شد.

طرح مسئله:

می خواهیم یک برنامه وب بنویسیم که پس از دریافت گذرواژه بطور اتوماتیک و بدون refresh کردن صفحه قدرت گذرواژه را محاسبه کرده و آن را نشان دهد همچنین باید در صورت مواجه شدن با توالی کاراکترها یا بکارگیری کاراکترهای یکسان گذرواژه را ضعیف ارزیابی کند. برنامه باید یک تصویر را بصورت دینامیک آماده کند که در آن با افزایش قدرت گذرواژه طول نوارسبز افزایش یابد.

کد برنامه:

```
private void Page_Load(object sender, System.EventArgs e)
{
        Response.Cache.SetCacheability(HttpCacheability.NoCache);
        Response.Clear();
        string StrPass = Request.QueryString["pass"];
        if ((StrPass != null)&&(StrPass != ""))
        {
```

```
                try
                {
                        if(StrPass.Length>40)
                        {
                                PaintError("طول ورودی نادرست است");
                        }
                        string Str = "";
                        int i = CalcStrength(StrPass,out Str);
                        MyPaint(i,Str);
                }
                catch
                {
                        PaintError("خطا در فراخوانی تابع");
                }
        }
        else
        {
                PaintError("خطا ورودی تهی");
        }
}
private void MyPaint(int h,string StrMsg)
```

```
        {
                Bitmap picture = new Bitmap(280, 15);
                System.Drawing.Graphics g = System.Drawing.Graphics.FromImage(picture);
                int i, j;
                for (i = 0; i < 15; i++)
                {
                        for (j = 0; j < 80; j++)
                        {
                                picture.SetPixel(79 - j, i, Color.FromArgb(iRed(h, j), iGreen(h, j), 25));
                        }
                }
                Pen pen = new Pen(Color.FromArgb(192,213,244));
                Rectangle rect = new Rectangle(79,0,280,15);
                SolidBrush b = new SolidBrush(Color.FromArgb(192,213,244));
                SolidBrush blue = new SolidBrush(Color.Blue);
                g.DrawRectangle(pen, rect);
                g.FillRectangle(b, rect);
                Font Font1 = new Font("tahoma", 8, System.Drawing.FontStyle.Regular);
                System.Drawing.StringFormat MyAlignment = new StringFormat();
                MyAlignment.Alignment = StringAlignment.Far;
                g.DrawString(StrMsg, Font1, Brushes.Black, 250, 0, MyAlignment);
                Response.ContentType = "image/jpg";
```

```
        picture.Save(Response.OutputStream,System.Drawing.Imaging.ImageFormat.Jpeg);

        g.Dispose();

        picture.Dispose();

    }




    private void PaintError(string StrErr)

    {

        Bitmap picture = new Bitmap(280, 15);

        System.Drawing.Graphics g = System.Drawing.Graphics.FromImage(picture);

        Pen pen = new Pen(Color.FromArgb(192,213,244));

        Rectangle rect = new Rectangle(0,0,280,15);

        SolidBrush b = new SolidBrush(Color.FromArgb(192,213,244));

        SolidBrush blue = new SolidBrush(Color.Blue);

        g.DrawRectangle(pen, rect);

        g.FillRectangle(b, rect);

        Font Font1 = new Font("tahoma", 8, System.Drawing.FontStyle.Regular);

        System.Drawing.StringFormat MyAlignment = new StringFormat();

        MyAlignment.Alignment = StringAlignment.Far;

        g.DrawString(StrErr, Font1, Brushes.Black, 250, 0, MyAlignment);

        Response.ContentType = "image/jpg";

        picture.Save(Response.OutputStream,System.Drawing.Imaging.ImageFormat.Jpeg);
```

```
            g.Dispose();

            picture.Dispose();

        }



        int iRed(int h,int i)

        {

            int iout = 255 - ((h / 20) * i);

            if (iout <= 0) iout = 0;

            return iout;

        }

        int iGreen(int h, int i)

        {

            int iout = 1 + ((h / 10) * i);

            if (iout >= 255) iout = 255;

            return iout;

        }

        private int CalcStrength(string StrIN,out string StrOUT)

        {

            StrOUT = "";

            string StrChk = CheckSpecialCase(StrIN);

            if (StrChk != "ok")

            {
```

```
        StrOUT = StrChk;

        return 1;

    }

    int b1, b2, b3, b4;

    double d1, d2, d3, S;

    string SrtLowercase = "abcdefghijklmnopqrstuvwxyz";

    string SrtUpperrcase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    string StrSpecialChars = @"~`!@#$%^&*()_+-=,<.>/?':;|\]{}[" + "\"" ;

    string StrNumbers = "0123456789";

    char[] ArrLowerCase = new char[] { };

    char[] ArrUppercase = new char[] { };

    char[] ArrSpecialChars = new char[] { };

    char[] ArrNumbers = new char[] { };

    ArrLowerCase = SrtLowercase.ToCharArray();

    ArrUppercase = SrtUpperrcase.ToCharArray();

    ArrSpecialChars = StrSpecialChars.ToCharArray();

    ArrNumbers = StrNumbers.ToCharArray();

    b1 = StrIN.IndexOfAny(ArrLowerCase);

    b2 = StrIN.IndexOfAny(ArrUppercase);

    b3 = StrIN.IndexOfAny(ArrSpecialChars);

    b4 = StrIN.IndexOfAny(ArrNumbers);

    S = 0;
```

```
        if (b1 >= 0) S += 26;

        if (b2 >= 0) S += 26;

        if (b3 >= 0) S += 32;

        if (b4 >= 0) S += 10;

        d1 = (double)(StrIN.Length);

        d2 = System.Math.Log10(S) * d1 ;

        d3 = 30 + (2.1) * (d2 - 7);

        if (d3 <= 0) d3 = 1;

        d2 = System.Math.Round(d2);

        d3 = System.Math.Round(d3);

        if(d3<1) d3=1;

        StrOUT = AppraiseStrength((int)(d3));

        return ((int)(d3));

    }


    private string CheckSpecialCase(string StrIn)

    {

        string StrLower = StrIn.ToLower();

        if (StrIn.Length < 5) return "کوتاهی طول گذرواژه";

        if (StrIn.Length > 19) return "ok";

        string[] Str1 = new string[] { "test", "admin", "administrator",
```

```
"pass", "password", "user", "pa$$vword" };

for (int i = 0; i < Str1.Length; i++)

{

      if (StrLower == Str1[i].ToString()) return "قـابـل حدس";

}

string StrKeyboard = "qwertyuiop[]asdfghjkl;'zxcvbnm,./,mnbvcxz';lkjhgfdsa[poiuytrewq";

if (StrKeyboard.IndexOf(StrLower)>0) return "تـوالـی صفـحه کـلـیـد";

int  j, k, p1, p2;

p1 = 0;

p2 = 0;

char[] ArrTest = new char[] { };

ArrTest = StrIn.ToCharArray();

for (j = 0; j < ArrTest.Length - 1; j++)

{

      for (k = j + 1; k < ArrTest.Length; k++)

      {

             int d1 = System.Math.Abs(ArrTest[j] - ArrTest[k]);

             int d2 = System.Math.Abs(j - k);

             if ((d1 == 1)&&(d2 == 1)) p1++;

             if ((d1 == 0) && (d2 == 1)) p2++;


      }
```

```
        }
        if (p1 >= 3) return "استفاده از کاراکترهای متوالی";
        if (p2 >= 3) return "استفاده از کاراکترهای یکسان";
        return "ok";
    }
    private string AppraiseStrength(int i)
    {
        if (i<=5) return "غیر قابل قبول";
        if ((i>5)&&(i<=10)) return "بسیار ضعیف";
        if ((i>10)&&(i<=20)) return "ضعیف";
        if ((i>20)&&(i<=50)) return "متوسط";
        if ((i>50)&&(i<=70)) return "قوی";
        if (i>70) return "بسیار قوی";
        return "خطا در مقدار ورودی";
    }
```