ASP Security: Ways to Avoid Attack

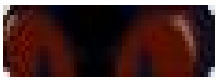
امنیت در ASP.NET

راههای مقابله با نفوذ در *ASP.Net*

نویسنده : *Hashem(S)*

منبع : **WWW.DEVX.COM**

تاریخ : ۱۳۸۳/ ۷/ ۱۰



Hashem(S)

مقدمه:

امروزه استفاده از صفحات**ASP.NET**یکی از وسایل تولید برنامه های کابردی وب**(Applitaction)** است.ولی این صفحات دارای ضعفهای فراوان می باشد.امیدوارم این مقاله بتواند برای طراحان سایتها کمکی باشد تا صفحاتی با ضریب امنیت بالاتر طراحی شود.

شاید افزایش امنیت برنامه های کاربردی وب چندان به چشم نیاید ولی شاید یکی از راههای نفوذ هکران عزیز !!!! ما می باشد.طراحی برنامه های کابردی وب با**ASP.NET**آنچنان آسان نیست.این نوع طراحی امنیت و امکانات مخفی زیادی داره اما آیا ما با آنها می توانیم کار کنیم در ضمن این قابل ذکر است که هکرها همیشه یک قدم از ما جلوترند.

یکی از راههای مقابله با آنها دانستن روش کار هکرهاست یعنی دانستن روشهای نفوذ حالا با بستن راه ها می شود تا حدی جلو آنها را گرفت.

بررسی راهها شاخه های زیادی داره اما به طور کل می توان ۸ روش است.

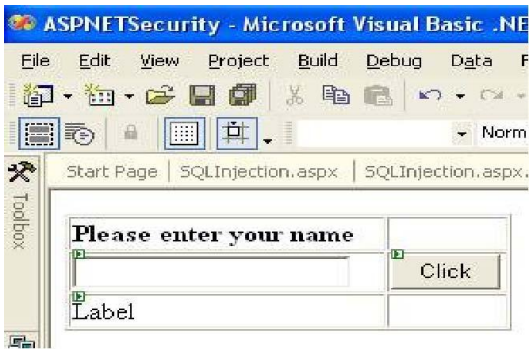
روش اول: (XSS) Cross - Sit Scripting

این روش ایجاد اختلال در صفحات وب با**ASP**توسط**Script**های قابل اجرا است.**XSS**یکی از بیشترین روش حمله می باشد.یک هکر برای حمله**XSS**به راحتی با تزریق یک**Script**در میان صفحه وب از طریق قسمت ورودی های کاربران و کلیک دکمه تایید بدون چک کردن لازم می تواند صفحه شما را مورد حمله قرار دهد.موقعی که داده ها که همراه**Script**است در وب پویا ذخیره می شود این داده می تواند برای هر کاربر دیگر در مرورگر وبش بار شود.

برای فهمیدن بهتر**XSS**مثالی می زنیم:

در یک صفحه وب که از یک کار بر سوال میشود**User Name**خود را در کادری در صفحه وارد کند از طریق این کادر**(شکل ۱)**وقتیکه ما نام کاربر را وارد کنیم دکمه را کلیک می کنیم حال ما می توانیم به جای نام کاربر در کادر یک خط دستور**Script**وارد کنیم برای مثل: **(شکل۲)**

```
<script> alert (“HELLO THERE”); </script>
```



شکل۱- آماده وارد کردن یک رشته کاراکتری

شکل۲- این یک رشته کاراکتری نیست. بلکه یک حمله با Script است.

در مورد زمانی که این**Script**را با زدن دکمه **Click**اجرا می شود و**(شکل۳)**پدید می آید.

شکل۳- بعد از کلیک**Script**در فرم۲

این یک مثال بود اما**Script**های وجود دارد که خسارات زیادی میزند حتی بعضی از آنها می توانند مقدار**Cookie**های و یا اطلاعات کاربران را که از این سرور استفاده کرده اند حتی اگر از سایتهای دیگر استفاده کرده باشند.خوشبختانه در**ASP.NET V1.1** دارای سیستم می باشد که می تواند**Script**ها را شناسایی کند و پیغام **(شکل۴)**را می دهد.این ساختار در سیستم درونی**ASP.NET V1.1**تعبیه شده که

شکل۴

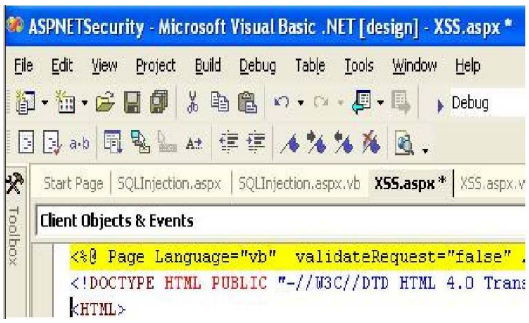
بتواند جلوی نفوذ باScriptرا بگیرند روش آن بصورت این است که به کلمه کلیدی<script>را در کلمه ورودی می گردد ودر صورت یافتن آن پیغام می دهد.
اما این سیستم آنچنان هم قوی نیست می گویید چرا ؟ چون راههایی است که به راحتی می شود این سیستم را فریب داد مثلا یکی از این راهها با اضافه کردن یک کاراکترNULL (%00)قبل ازScriptدردستور<Script>برای مثال قبل می شود :

```
<%00script> alert (“HELLO THERE”); </script>
```

پس با این سیستم هم می شود جلو نفوذ کامل را گرفت.یک روش خوب استفاده کردن سرور از روش HTML ENCODE()است که کارش این است که ورودی ها را به رمز (کد) تبدیل می کند با رشته های HTML ENCODEمی باشد.حال ورودی کاربر با استفاده از تابع زیر رمز گذاری می شود :

```
Private Sub Button1_Click(ByVal sender As System.Object,ByVal e As System.EventArgs)_
 Handles Button1.Click
lblMessage.Text = Server.HtmlEncode(txtName.Text)
End Sub
```

خوب حالااگرکاربریکScriptواردکند HTMLENCODE ورودی را گرفته آن را به شکل رشته کارکتری در آورده و در مرور گر نشان می دهد نه به صورت اجرایی شما اگر به هر دلیلی خواستید خاصیت محافظ درASP.NETV1.1 را از کار بیندازید کافی است خاصیتValidateRequetرا به حالتFalseدر آورید . (شکل ۵)
در این حالت خاصیت محافظت را به صورت دستی غیر فعال می کنیم اما ما پیشنهاد این کار را نمی کنیم بالاخره شاید بتواند جلوی هکر های تازه کار بگیرد گفتم شاید فقط شاید!!!!!!!



شکل۵

روش دوم: SQL LNJECTION

دومین روش بسیار ساده که با یک اکسپلویت معروف با نامSQLمی باشد.جالب این که کمتر هکر ها از آن استفاده می کنند بهتر بگویم این روش فقط در بین هکرهای با سابقه رواج دارد تا غیر حرفه ایها.
جالبتر که تعداد افرادی که این نقطه ضعف را می دانند بسیار کم هستند وحتی درباره آن نشنیده اند.حالا بگذارید با یک مثال طرز کار با این روش که با وارد کردن یکSQL است کار می کند را توضیحی کوتاه بدهیم:
فرض کنید یک صفحه داریم که برایLogin(وصل شدن)بایدUsernameو پسورد را وارد کنیم.
در صفحات وب پویا یک تابع(Function) بسیار معمولی وجود دارد.بیشتر طراحان سایت(مخصوصا تازه شروع به طراحی برنامه های بانک اطلاعاتی در وبها کرده اند)تابعی به صورت زیر می نویسند:

```
Private Sub btnLogin_Click(ByVal sender As System.Object,ByVal e As System.EventArgs) _
Handles btnLogin.Click
SqlConnection1.Open()
Dim str As String = "SELECT * FROM Users WHERE UserID='" _
& txtName.Text & "' AND Password='" & _
txtPassword.Text & "'"
Dim comm As New SqlCommand(str, SqlConnection1)
Dim reader As SqlDataReader = comm.ExecuteReader()
If Not reader.HasRows Then _
Response.Write("Login failed. Please try again")
```

```
While reader.Read()
Response.Write("Hello " & reader("UserName"))
End While
End Sub
```

در این کد به سادگی اطلاعات وارد شده توسط کاربران در روی صفحه وب آشکار است و از فرمولهای رشته ها درSQLبرای برنامه نویسی استفاده شده است برای گرفتن داده های کاربر.

باید در طراحی صفحات کمترین اطلاعات از کاربران در صفحه درج نشود مثلا در بعضی سایت هاUsername کاربران مشخص است.بهترین راه کنترل طول ورودی که کاربر وارد می کند است تا مطمئن شویم ورودی طولانی تر از حد معمول نباشد. با وجود این کدها نامرتب خطر بزرگی ما رو تهدید می کند.وقتی نام کاربران مشخص باشد یعنی برای هکر ها نصف کار را انجام دادیم !! مطمئن باشید چیز هایی وجود دارد که آنها به جای پسورد وارد کنند و به بانک اطلاعات شما دست بیابند.برای مثل (شکل۶)

در این شکل با وارد کردن چند کلمه کلیدیSQLمی توان اطلاعات رو دستکاری کرد و بدون داشتن پسورد وارد شد .

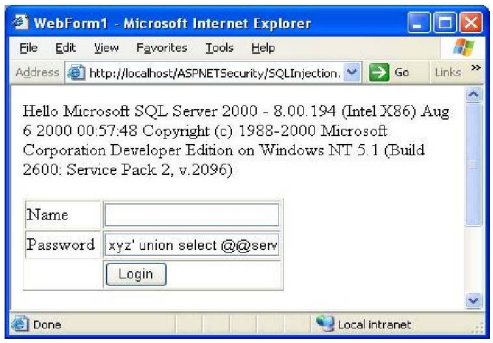
یکی دیگر از اکسپلویت هایی که هکرها از آن برای حمله استفاده می کنندSQL Unionاست.به این صورت کار می کند که به دنبال رشته کاراکتری وارد شده درUsernameیا پسورد و اجرا آن با زدن دکمهClick(شکل۷)می توان اطلاعات مهم و زیادی در مورد سرور در یافت کرد.

کدSQLUnionبصورت زیر می تواند باشد:

```
xyz' union select @@servername, @@servicename,
@@version –
```

یک روش بسیار خوب بروی فرمولبندی رشته هایSQL استفاده از موضوعParametersدرSQLCommandاست.علت برتری این معبر ایجاد شده این است کهADO.NETنمی تواند عمل جانشینی(Substitution)را انجام دهد.در این معبر پارامترهایSQL Serverهمان جا عمل جانشینی و خطا یابی و یا تایید رخ می دهد.

شکل۶- با داشتن نام کاربر بدون داشتن پسورد وارد شوند.



شکل۷- کاربر بجای پسورد در فرمLoginیک رشته دستوراتSQLبرای آشکار کردن اطلاعات لازم برای حمله استفاده کرده.

در زیر این روش را برایLoginپیاده سازی کرده ایم:

```
Private Sub btnLogin_Click(ByVal sender As System.Object,ByVal e As System.EventArgs) _
Handles btnsecureLogin.Click
SqlConnection1.Open()
 Dim str As String = "SELECT * FROM Users WHERE " & _
"UserID=@userID AND Password=@password"
Dim comm As New SqlCommand(str, SqlConnection1)
comm.Parameters.Add("@userID", txtName.Text)
comm.Parameters.Add("@password", txtPassword.Text)
Dim reader As SqlDataReader = comm.ExecuteReader()
If Not reader.HasRows Then Response.Write("Login failed. Please try again")
While reader.Read()
Response.Write("Hello " & reader("UserName"))
End While
End Sub
```

روش سوم: تایید ورودی های کاربر توسط خود برنامه نویس وب

در این روش باید اطلاعاتی از پیش معین شده را در اختیار کاربران قرار دهیم تا کاربر فقط بتواند همانها را وارد و انتخاب کند. در این روش با استفاده از اسکرول بار ها و **check box**ها در ورودی ها و محدود کردن گزینه های ورودی و قرار دادن مقدار پیش فرض در ورودی ها. اما این روش برای ورودی های قابل پیش بینی و محدود کاربرد دارد مثل:تاریخ تولد و نام کشور و نام شهر و .... .
در ورودی های که نمی توان از حالت قابل پیش بینی استفاده کرد هم در کلاینت و هم در سرور برای آن ورودی یک مقدار پیش فرض در کادر ورودی کلاینت و در بانک سرور قرار دهید.

روش چهارم: استفاده از روش مخلوط کردن(Hahsing)پسوردهای ذخیره شده

یک روش خوب برای ذخیره پسود ها دربانک اطلاعاتی سرور استفاده از روش**Hashing**(درهم) است.
Hashigدارای دو روش است:
یک: روش پردازش برای نگارش داده ها(متن های بدون رمز)برای این کار از یک رشته بایتی منحفر به فرد با طول ثابت استفاده می شود.به این رشته بایتی با طول ثابت**Hash**گویند.
دوم: روش ایستا در این روش تمام اطلاعات توسط یک متد مرز گذاری و ذخیره می شود.
در مورد ذخیره پسوردها در بانک اطلاعات باید گفت این اطلاعات ذخیره شده ارزش**Hash**هر یک از پسورد ها است و برتری بالاتری برای ذخیره پهنای پسورد دارد.وقتکه یک کاربر پسورد خود را برای**Login**شدن وارد می کند ارزش**Hash**پسورد محاسبه شده و سپس با ارزش**Hash**ذخیره شده مقایسه می شود.
اما باز مشکلات فرا وانی است اگر هکرها به اطلاعات دست یابند درست است به پسوردهای اصلی دسترسی ندارد ولی هنوز می تواند مقدار آنها را تغییر دهد یا با ایجاد چند کاربر برای خود(در حالت ایستا) به فرمول رمز گذاری دست یابد.
تابع زیر استفاده می کند ازالگوریتم عددی درهم سازی(hash) **SHA1** استفاده شده :

```
Public Function ComputeHashValue(ByVal data() As Byte) As Byte()
 Dim hashAlg As SHA1 = SHA1.Create
 Dim hashvalue() As Byte = hashAlg.ComputeHash(data)
 Return hashvalue
 End Function
```

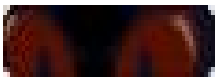
میشود پسورد ها را با فراخوانی تابع بالا تبدیل به ارزش**Hash**کرد:

```
Dim hashValue() As Byte
hashValue = ComputeHashValue(Encoding.ASCII.GetBytes(txtPassword.Text))
```

می شود این کار را برای همه اطلاعات و یا فقط پسورد به کار برد.

روش پنجم: رمز گذاری داده های مهم برای جلو گیری از نفوذ پذیری

صفحه وبها و برنامه های کابردی وب ساخته شده با**ASP.NET** می توانند در بعضی مواقع برای ذخیره اطلاعات مفید باشند.مثلا:برای ذخیره رشته ها در بانکهای اطلاعات از فایل**Web.config**سریعتر نسبت به **Hard-code**در برنامه های کاربردی.در بانکهای اطلاعاتی سرور همیشه باید تغییرات و اصلاحات و کامپایل مجدد انجام داد بنابراین ذخیره اطلاعات به صورت رشته کاراکتری ساده(این رشته کاراکتری می تواند اطلاعاتی در مورد کاربران ویا پسورد ها باشند)در فایل**Web.config**کاری بسیار خطرناک است و اصلا پیشنهاد نمی شود چرا که فایل**Web.config**در مستندات(Document) **XML**ذخیره شده که این فایل یک فایل متنی ساده است و به راحتی در دسترس دیگران(**هکرهای عزیز!!!!**) قرار می گیرد.
یه راه ساده و آسان رمزگذاری داده های مهم(و غیر مهم چون گاهی همان ها هم خطرناکند)و ذخیره آن به صورت رمز متنی در میان فایل**Web.config**است.
دو الگوریتم برای پنهان سازی وجود دارد :
۱- متقارن(**Symmetric**) **۲-** نامتقارن(**ASymmetric**)



Hashem(S)

۱-الگوریتم متقارن: رمز گذاری و بازکردن رمز اطلاعات با استفاده از یک کلید مشترک باری همه داده ها و همه کاربران.این روش بسیار ساده و کم هزینه است پس دارای امنیت کمتری می باشد چرا؟؟؟؟ چون نفوذ گر بعد از نفوذ ودسترسی به فایل ما با درست کردن چند کاربر برای خود و کمی مهارت می تواند با مشاهده تغییرات کلید رمز ما را به راحتی کشف کند و آن وقت .....

۲-الگوریتم نا متقارن: رمز گذاری و رمز گشایی اطلاعات با چند کلید.این کلیدها دو دسته هستند کلید شخصی و کلید عمومی.رمزگذاری داده ها از یک کلید عمومی استفاده می کند که فقط برای رمزگشایی از یک کلید شخصی استفاده می کند و بر عکس.این الگوریتم دارای پیچیدگی های محاسباتی است و پر هزینه پس دارای امنیت بیشتری نسبت به نوع اول است.

لیست۱ نمایش می دهد آرایه ای به نام**Rijndael**را که در الگوریتم رمزگذاری متقارن قرار گرفته و لیست۲ نشان می دهد آرایه ای به نام**RSA**را که در الگوریتم رمز گذاری نا متقارن قرار گرفته است.
توابعی که در زیر می بینید در لیستهای ۱و۲ باعث رمز گذاری داده ها در برنامه های کاربردی وب به شکلی مخصوص و فایل**XML**می شود.لیست۳ که مشاهده می کنید یک تابع پشتیبان است که از توابع استفاده شده در لیست ۱و۲بهره می برد.
این تابع پشتیبان باعث تبدیل رشته کاراکتری به آرایه بایتی می شود.تبدیل رشته کاراکتری "1 2 3 4 5 6" به آرایه بایتی{1,2,3,4,5,6}.
برای رمز گذاری نامتقارن نیازمند سازنده کلیدهای عمومی**(Public Key)** وکلید شخصی**(Private Key)** هستیم:

استفاده از نامتقارن

```
==========For Asymmetric use============
Dim publicKey, privateKey As String
Dim RSA As New RSACryptoServiceProvider()
publicKey = RSA.ToXmlString(False) ' get public key
privateKey = RSA.ToXmlString(True) ' get private key
```

رمز گذاری نامتقارن

```
==========Asymmetric Encryption==========
Dim cipherText as String = AsymmetricEncryption _
(txtAsyPlainText.Text, publicKey)
```

رمز گشایی نامتقارن

```
==========Asymmetric Decryption============
Dim plainText as String = AsymmetricDecryption _
(txtAsyCipherText.Text, privateKey)
```

برای رمز گذاری متقارن نیاز داریم به کلیدی با ۱۶ بیت و بردار مقدار دهی**(IV)**:

```
==========Symmetric============
Dim Key, IV as String
Key="1234567890123456"
IV ="1234567890123456"
Dim cipherText As String = SymmetricEncryption _
(txtSyPlainText.Text, Key, IV)

==========Symmetric============
Dim plainText as String = SymmetricDecryption _
(txtSyCipherText.Text, Key, IV)
```

زیرا پیغامهای**SOAP**(Simple Access Protocol پروتکل مبتنی بر**XML**برای تبادل اطلاعات ساختار یافته و نوع دادهها در وب) فرستاده می شود در متن بدون رمز سرویسهای وب می توانند همچنین استفاده کنند از فرم رمز گذاری شده در عوض استفاده از**SSL**(**Secure Sockets Layer** پروتکلی که توسط شرکت **Netscap** طراحی شده و برای رمز گذاری با کلید عمومی استفاده می شود و برای کار های مالی طراحی شده ولی برای امکانات و

سرویسهای دیگر هم کار می کند)از ورود مسیرهای ارتباطی نگهداری می کند از رمز گذاری می توان برای نگهداری از اطلاعات مهم مثل شمارهcredit cardها از دید دیگران می شود استفاده کرد.

روش ششم: ذخیره اطلاعات مهم در رجیستری

برای رمز گذاری دادهای که به صورت دستی وارد می شوند از رجستری در ذخیره اطلاعات مهم استفاده کرد.برای مثال ممکن برای پیکربندی وب سرور شما به بانک اطلاعات تان با قبول شناسه ها در ویندوز از راه دورLoginمی شوید و همچنین ممکن در پیکربندی شما استفاده بشه از برنامه های کاربردی وب با استفاده ازImpersonate, تخصیص یکUserو پسورد خاص این ارتباط را بر قرار کنیم.

```
<identity impersonate=”true”
username=”someuser”
password=”tosecret”
/>
```

بنابراین ذخیرهUsernameو پسورد درWeb.configبه شکل متن بدون رمز کاری بسیار خطر دارد یک کار نسبتا مناسب ذخیره کردن ازUsernameو پسورد در رجیستری است.
در پیروی سری کارهای انجام شده برای ذخیره رشته ها در بانک اطلاعات در فایلWeb.configو استفاده از نرم افزارASPNET_SETREG.exe که محصول شرکت ماکرو سافت است می توانUsername و پسورد را در رجیستری ذخیره نمود.

1- برای دانلودaspnet_setreg.exe می توانید به آدرس زیر مراجعه کرد:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q329290

2- ساختن یک کاربر جدید برای ویندوز کامپیوترتان. من می توانم با صدا کردن
Username: "ASPNETUSER "با پسورد "secret"

3- اضافه کردن عبارت<appSetting>در میان فایلWeb.configاین تغییر را ذخیره می کنیم .

```
<configuration>
<appSettings>
<add key="Distributor" value="workstation id=F16;packet size=4096;integrated
security=SSPI;data
source=F16;persist security info=True;initial catalog=Distributor" />
</appSettings>
```

4- دراین کد ما , ما می توانیم رشته کاراکتری معین راکه در فایلWeb.configاستفاده شده را باز یافت کنیم .

```
Dim connStr As String = ConfigurationSettings.AppSettings("Distributor")
Dim Conn As New SqlConnection(connStr)
```

5- استفاده دیگری که می شود از aspnet_setreg.exe کرد اضافه کردن Username و پسورد به عنوان کاربر که در ASP.NI برای معرفی کاربر در رجستری :

```
C:\>aspnet_setreg -k:Software\ASPNetApp\Identity -u:ASPNETUSER -p:secret
```

6- همچنین می شود در ویندوز از پیغامهای طولانی در صفحه پرینت گرفت. در پایین پیغامی را می بینید که دو خط آن پر رنگتراز بقیه است ما نیاز داریم که فقط این دو خط را در فایل متنی ذخیره کنیم.

```
Please edit your configuration to contain the following:
userName="registry:HKLM\Software\ASPNetApp\Identity\ASPNET_SETREG,userName
"
password="registry:HKLM\Software\ASPNetApp\Identity\ASPNET_SETREG,password"
```

The DACL on the registry key grants Full Control to System, Administrators, and Creator Owner.
If you have encrypted credentials for the <identity/> configuration section, or a connection string fo
<sessionState/> configuration section, ensure that the process identity has Read access to the registry key. Furthermore, if you have configured IIS to access content on a UNC share, the account u
access the share will need Read access to the registry key.
Regedt32.exe may be used to view/modify registry key permissions.
You may rename the registry subkey and registry value in order to prevent discovery.

7- محل فایل**Machine.config**در:

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CONFIG

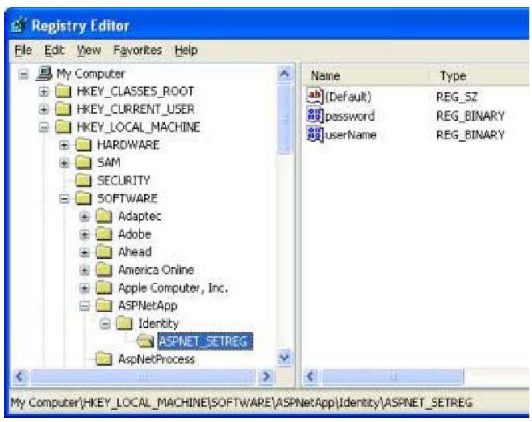
است و اصلاح کردن<**identity**>در فایل**Machine.config**به صورت زیر است:

```
<identity impersonate="true"
userName="registry:HKLM\Software\ASPNetApp\Identity\ASPNET_SETREG,userName"
password="registry:HKLM\Software\ASPNetApp\Identity\ASPNET_SETREG,password"
/>
```

ما می توانیم با جلوگیری از تغییر در فایل**Machine.config**به وسیله اصلاحات در فایل**Web.config**در برنامه های شما ومعین کردن هویت دیگر کاربران و شناسه های آنها.

8- شروع تغییر در رجستری و خاصیت آن در کامپیوتربه آدرس زیر در**Registery Editor** می رویم:

HKEY_LOCAL_MACHINE/SOFTWARE/ASPNetApp/Identity/ASPNET_SETREG



شکل۸- استفاده از Registery Editor
تغییرات ایجاد شده را در رجیستری می بینید

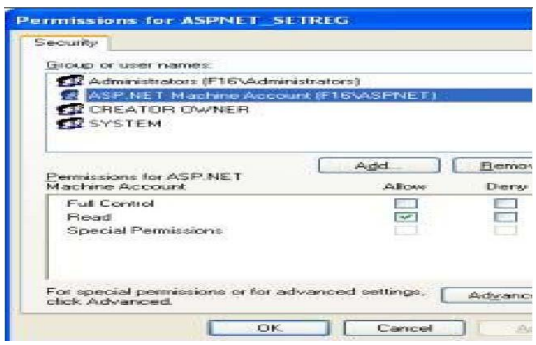
(با استفاده ازRegedt32)که در**(شکل ۸)**میبینید.

9- حالا راست کلیک روی کلید رجیستری **ASPNET_SETREG** کرده و**Permission**را انتخاب می کنیم. اضافه کردن یک کاربر در**ASPNET**و گذاشتن آن در قسمت**Read permission**(در شکل ۹ می بینید)

10- نسبت دادن کاربر**ASPNETUSER**به **Full Control**در قسمت**Machine Account**برای دسترسی مستقیم و درست به فایل **"Temporary ASP.Net"** در:

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322.

11- با استفاده از این برنامه توانستیم یک تولید کاربر با نام**ASPNETUSER**در رجیستری تولید کرده با امنیت بالا.



شکل۹- Read permission.تغییر و قرار دادن
ASPNETدر Permissionبرای خواندن کلیدها

روش هفتم: انجام چند سیستم داری قبل از Deploy برنامه های کاربردی وب

توضیح:

سیستم داری **Housekeeping** : روتین های از قبیل به روزسازی ساعت یا جمع آوری حافظه های آزاد شده که برای مرتب نگهداشتن سیستم محیط اجرا یک برنامه یا ساختارداده ای یک برنامه طراحی می شود.
پیگیری مرحله به مرحله**(tracing)**در برنامه های کابردی وب در**ASP.Net**بسیار ساده است با به کاربردن عبارت<**trace**>در فایل**Web.config**به اضافه در دستورات صفحه است.بنابراین موقعی تومی توانی با خواند **Deploy** در برنامه ها مطمئن شوی که**tracing** غیر فعال شدن در هردوی سطح صفحه و سطح برنامه که از کد زیر استفاده کنیم :

```
<trace enabled="false" requestLimit="10" pageOutput="false" traceMode="SortByTime"
localOnly="true" />
```

و همچنین خاموش کردن مد**debug**در فایل**Web.cofig**با کد:

```
<compilation defaultLanguage="vb" debug="true" />
```

در عبارت<**customErrors**>در**Web.config**یادتان باشد که مقدار مد را در حالت**"RemoteOnly"**قرار دهید به صورت زیر:

```
<customErrors mode="RemoteOnly" />
```

مقدار مد می تواند سه مقدار باشد :

۱- **"on"** = همیشه پیغامها را به صورت عادی نشان می دهد(توافقی)
۲- **"off"** = همیشه جزییات اطلاعات خطا های**ASP.Net** را نشان میدهد
۳- **"RemoteOnly"** = نمایش عادی پیغامها فقط کاربران نمی توانند اجرا کنند در محدوده وب سرور مان.

این تنظیم را پیشنهاد می کنم چون باعث می شود که جزییات اطلاعات کاربران نمایش داده نشود.
آخرین و نه کمترین کار حذف کردن**Solution** و فایلهای**Project**درفرم**deployment sever**است.اگر ما فیلتر **ISAPI**(این فیلتر یک فایل DLL که IIS ماکروسافت آن را برای اعتبار سنجی و بررسی درستی دخواستهای ISAPIدریافتی توسط IISاستفاده می کند)را قرار ندهیم بنابراین هکرها خیلی ساده حدس میزنند نام کاربر و دسترسی مستقیم داشتن به فرمهای ما در**WebBrowser**ها.

روش هشتم: استفاده از Session ها اما نه Cooki-less Session ها

اگه ما احتیاج داشته باشیم به نگهداشتن**Session** یک کاربر استفاده از موضوع**Session** برای ذخیره آن.موضوع**Session**در**ASP.Net**استفاده میکند از کوکی ها ذخیره شده در**Session ID** , کوکی بدست می آورد چیزای که مابین کلاینت و سرور رد و بدل می شود.موضوع**Session**دارای اطلاعات بسیار حساس وحیاتی که در طرف سرور ذخیره می شود.از این رو فقط اطلاعات**Session ID** که بدون حفاظ گذاشته شده اند ونه اطلاعات حساس و مهم.
ASP.Net پشتیبانی می کند از**Cookie-less Session**که ممکن است به نظراضافه**(temping)**بیاید بنابراین خیلی از کاربرها غیر فعال می کنند سیستم کوکی ها رو در مرورگرهاشون.استفاده می شود موضوع **Cookie-less Session**توسط هکرها در این حال هکر ها می توانند با استفاده از**URL** که تو در دسترس داری وکارگیری مرورگر ها کاری که نباید بکنند را انجام می دهند.
در آخر کار بایدهمیشه جلوگیری کرد از**Cookie-less Session**همین.

برداشت آخر:

در آخر تذکر می دهم تا می توانید مطالعه کنید کتابهایی در مورد شبکه و .NET. و XML.

با امید بهروزی و موافقیت شما
هاشم