ASP.NET Web Uygulamalarında Güvenlik

Cengiz HAN

cengiz@cengizhan.com
www.cengizhan.com

1

ASP.NET Web Uygulamalarında Güvenlik

- Konu 1: Web Uygulaması Güvenliği Temel Kavramlar
- Konu 2: Windows Tabanlı Kimlik Denetimi
- Konu 3: Form Tabanlı Kimlik Denetimi

2

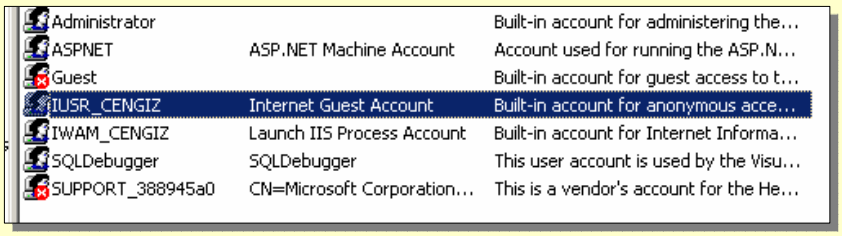
Konu 1: Temel Güvenlik Kavramları

- **Kimlik Denetimi (Authentication)**
  - **Kullanıcıların tanımlanması aşaması**
- **Yetkilendirme (Authorization)**
  - **Kullanıcının kimliği doğrultusundan erişim haklarını belirleme aşaması**

3

Anonim (Anonymous) Erişim

- **Internet üzeriden bulunan web sitelerinin çoğunda anonim erişim kullanılır.**
- **Sitenin her bölümünün herkese açık olduğu ve sitenin gizli veya kişiye özel bilgi içermediği durumlarda...**
- **ASP.NET Web Uygulamaları anonim erişim için taklit etme (Impersonation) yöntemini kullanır.**
- **Kimliksiz erişim yapan kullanıcıya genel bir kullanıcı hesabı atanması yapılarak yetkilendirme yapılır.**
- **Varsayılan olarak bu kullanıcı hesabının adı ISUR_*makineadi* şeklindedir.**



4

Kimlik Denetimi ile Erişim

- **ASP.NET Kimlik Denetimi Yöntemleri**
  - **Microsoft Passport**
  - **Windows tabanlı**
  - **Form tabanlı**

5

Uygun Kimlik Denetim Sistemini Seçmek

- **Anonim**
  - Tanıtım siteleri
  - Kullanım klavuzu
- **Windows tabanlı**
  - Intranet uygulamaları
- **Form tabanlı**
  - Alış-veriş sitesi

6

Web.Config Dosyasının Güvenlik Ayaları Açısından Temel Yapısı

- **<authentication>**
  - **Kimlik denetim sisteminin belirlenir**
- **<authorization>**
  - **Yetkilendirme işlemi için**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<system.web>
    <authentication mode="Windows|Forms|Passport|None">
        <forms name="formismi"loginUrl="girissayfasiadresi"
            protection="All|None|Encryption|Validation"
            timeout="30" path="/"
            requireSSL="true|false"
            slidingExpiration="true|false">
            <credentials passwordFormat="Clear|SHA1|MD5">
                <user name="kullaniciadi" password="sifre"/>
            </credentials>
        </forms>
        <passport redirectUrl="internal"/>
    </authentication>

    <authorization>
        <allow users="kullanıcıların listesi"
            roles="rollerin listesi"
            verbs="eylemlerin listesi" />
        <deny users="kullanıcıların listesi"
            roles="rollerin listesi"
            verbs="eylemlerin listesi" />
    </authorization>

</system.web>
</configuration>
```

7

Konu 2: Window Tabanlı Kimlik Denetimi

- **Window Tabanlı Kimlik Denetimi Nedir?**
  - Windows tabanlı kimlik denetimi Windows işletim sistemi üzerine kurulu çalışır ve sunucu bilgisayar üzerindeki kullanıcı listesi ile kimlik denetimi yapar.

8

Windows Kimlik Denetiminin Etkinleştirilmesi

Adım Adım Etkinleştirme

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <system.web>
        <authentication mode="Windows" />
        <authorization>
            <deny users="?"/>
        </authorization>
    </system.web>
</configuration>
```

Alt Klasör ve Dosyalar için Yetkilendirme Ayarları

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <location path="yonetim">
        <system.web>
            <authorization>
                <allow roles="BizimSirket/Yoneticiler"/>
                <deny user="*" />
            </authorization>
        </system.web>
    </location>
.....
```

9

Bir Kullanıcı için İzinlerin Verilmesi ve Kısıtların Konulması

- **ASP.NET Web.config dosyasındaki <authorization> düğümü içerisindeki kullanıcı listesine bakarak kullanıcının uygulama üzerindeki yetkilerine karar verir.**
- **ASP.NET <authorization> düğümü içerisinde kullanıcıların erişim yetkilerini kontrol ederken yetkilendirme bildirimlerinden kullanıcı ile ilk uyuşanı kullanır.**
- **Bu sebeple istenen kullanıcılara yetki verildikten sonra mutlaka <deny> düğümü ile önceki yetkilendirme düğümleri ile uyuşmayan kullanıcılar için kısıtlama yapılmalıdır. Bunu ise yıldız (*) karakteri vererek sağlayabiliriz.**

```
<authorization>
    <deny users="?"/>
</authorization>
```

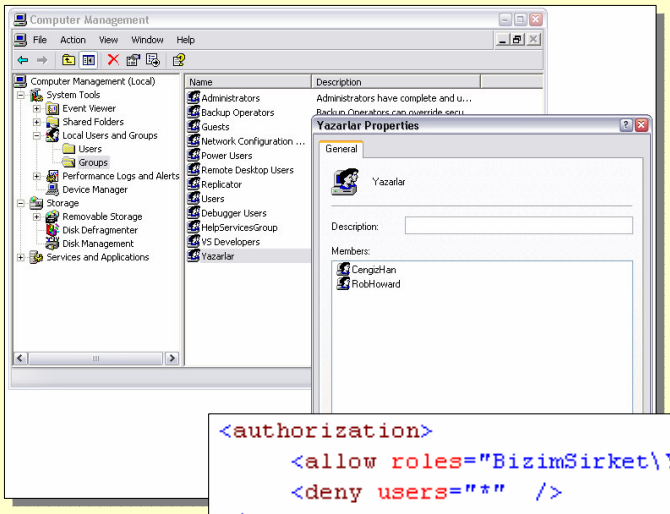
```
<authorization>
 <allow users="BizimSirket\RobHoward, BizimSirket\CengizHan" />
 <deny users="*"  />
</authorization>
```

```
</authorization>
```

10

Rol Tabanlı Kimlik Denetimini Kullanmak

- Her kullanıcıya yetki verilmek ile uğraşmadan belli bir rol üzerinde yetkilendirme işlemi gerçekleştirilir.
- Kullanıcılarda bu rolleri üstlendirilir.




```
<authorization>
      <allow roles="BizimSirket\Yazarlar" />
      <deny users="*"  />
</authorization>
```

11

Rol Tabanlı Kimlik Denetimini Kullanmak

- Sadece **Stajerler** grubundakilerin ve **anonim** erişim ile bağlanmak isteyenlerin uygulamayı kullanmasının kısıtlanması, diğer tüm kullanıcıların uygulamaya erişebilmesini sağlayan bir <authorization> düğümü :

```
<authorization>
      <deny roles="BizimSirket\Stajerler" />
      <deny user="?" />
      <allow user="*" />
</authorization>
```

12

Kullanıcı Bilgisini Okumak

- **Uygulamaya giriş yapmış olan kullanıcınınn bilgilerini okumak User nesnesinin Identity özelliği kullanılır. Identity özelliği kullanıcı adı ve rol bilgilerini içeren bir nesne döndürmektedir.**

```
lblUser.Text = User.Identity.Name
lblAuthenticationType.Text = User.Identity.AuthenticationType
lblIsAuthenticated.Text = User.Identity.IsAuthenticated
```

13

Konu 3: Form Tabanlı Kimlik Denetimi

- **Form Tabanlı Kimlik Denetimi Nedir?**
  - **Form tabanlı kimlik denetimi programcının düzenlediği bir Web Formu ile kimlik bilgilerinin kontrol edilmesi esasına dayanır.**
  - **Web.config dosyasındaki kullanıcı listesine veya programcı tarafından düzenlenen ayrı bir veritabanına göre kimlik denetimi ve yetkilendirme yapılır.**
  - **Form tabanlı kimlik denetiminde uygulamaya erişmek için kullanıcıların herhangi bir ağa üye olmalarına ihtiyaç yoktur.**
  - **Form tabanlı kimlik denetimi daha çok herkese açık internet uygulamalarında kullanılır.**

14

Form Tabanlı Kimlik Denetiminin Etkinleştirilmesi

- Form tabanlı kimlik denetimi programcıya kendi kullanıcı veritabanını oluşturma imkanı tanır.
- Adım Adım Etkinleştirme
  1. Web.config dosyasından kimlik denetim sistemi Forms olararak ayarlanır.
  2. Kullanıcı kimlik bilgilerinin (kullanıcı adı, şifre) girileceği bir Web Form oluşturulur.
  3. Kullanıcı kimlik bilgilerinin saklanması için isteğe seçime bağlı olarak programcının belirlediği parametreler çerçevesinde bir veritabanı oluşturulur veya Web.config dosyasında gerekli tanımlamalar yapılır.
  4. Oluşturulan Web Formuna kullanıcıların kimlik bilgilerinin denetlenmesi için gerekli kod yazılır. Buradaki denetleme için yazılacak kod parçası kullanıcı listesinin özel bir veritabanında mı, yoksa Web.config dosyasında mı tutulduğuna göre değişir.

15

Web.config Üzerindeki İşlemler

```
<authentication mode="Windows|Forms|Passport|None">
   <forms name="formismi"
          loginUrl="girissayfasiadresi"
          protection="All|None|Encryption|Validation"
          timeout="30" path="/"
          requireSSL="true|false"
          slidingExpiration="true|false">
      <credentials passwordFormat="Clear|SHA1|MD5">
         <user name="kullaniciadi" password="sifre"/>
      </credentials>
   </forms>
</authentication>
```

16

Web.config Üzerindeki İşlemler

- **<authentication>**
  - **mode:** Kimlik denetim sistemi (Forms)
- **<forms>**
  - **name:** Çerezin (cookie) ismini belirler.(varsayılan : **.ASPXAUTH**)
  - **loginUrl:** Giriş sayfası.
  - **protection:** Kullanıcı bilgisayarında saklanacak olan çerezin (cookie) güvenliğinin nasıl sağlanacağını belirler.
    - **All, Encryption, Validation, None.** Varsayılan olarak **All** değerini alır. All değeri ile çereze yazılacak verilerin güvenliği en iyi şekilde sağlanmaktadır.
  - **timeout:** Kimlik denetimi çerezlerinin kaç dakika geçerli olacağını belirler. (30)
  - **path:** Oluşturulacak çerezlerin yol tanımlaması için kullanılır. Varsayılan değeri / dır.
  - **requireSSL:** Kimlik denetim çerezinin transferi için güvenli bağlantı gerekip gerekmediğini belirler.
  - **slidingExpiration:** Alabileceği değeler **true** ve **false**' dur.
    - **true** değerini aldığında kullanıcıdan gelen her istek ile zaman aşımı (timeout) süresi geri sayımı en baştan tekrar başlar.
    - .NET Framework 1.0 da varsayılan değer true idi, .NET Framework 1.1 versiyonunda varsayılan değer **false**' dur.

17

Web.config Üzerindeki İşlemler

- **<credentials> düğümü içerisine eklenen <user> düğümleri ile Web.config dosyası içerisinde kullanıcı tanımlaması yapılabilir.**
- **<credentials>**
  - **passwordFormat :** Kullanıcı şifrelerine uygulanan karakter şifreleme (encrypt) yöntemi belirlenir. SHA1, MD5 ve Clear. Varsayılan değer SHA1' dir.
- **<users>**
  - **name :** Kullanıcının adını belirler.
  - **password :** Kullanıcının şifresini belirler. Belirlenen passwordFormat değeri ile uyumlu bir değer almalıdır.
- Bu şekilde bir uygulamada programcı yada bir yönetici tarafından Web.config dosyasına ekleme yapılmasıyla çalışabilir.
- Kullanıcıların kendi kullanıcı hesaplarını oluşturabileceği ve kimlik bilgilerinin yönetimini kendileri yapabileceği bir uygulama için kullanılamaz.

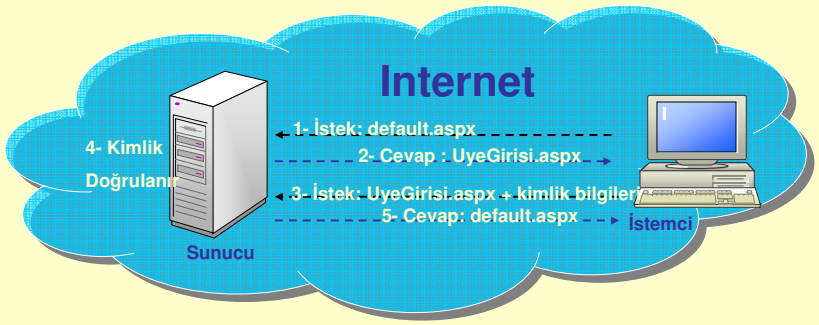
18

Kullanıcı Girişini Yapmak

- **Form tabanlı kimlik denetiminde kullanıcının girişi yapması için bir Web Form oluşturulmalıdır.**

Kullanıcı Adı:
Şifre:
Bu bilgisayarda hatırla
Giriş Yap
[lblMsg]

```
<form id="Form1" method="post" runat="server">
    Kullanıcı Adı: <asp:TextBox id="txtKullanici" runat="server"></asp:TextBox><BR>
    Şifre: <asp:TextBox id="txtSifre" runat="server" TextMode="Password"></asp:TextBox><BR>
    <asp:CheckBox id="chkHatirla" runat="server" Text="Bu bilgisayarda hatırla"></asp:CheckBox>
    <asp:Button id="btnGirisYap" runat="server" Text="Giriş Yap"></asp:Button><BR>
    <asp:Label id="lblMsg" runat="server"></asp:Label>
</form>
```



19

Web.config Dosyasındaki Kullanıcı Listesi ile Kullanıcı Girişi Yapmak

```
<authentication mode="Forms">
     <forms loginUrl="UyeGirisi.aspx">
          <credentials passwordFormat="Clear">
               <user name="cengiz" password="han"></user>
               <user name="rob" password="howard"></user>
          </credentials>
     </forms>
</authentication>
<authorization>
     <deny users="?" />
</authorization>
```

```
If FormsAuthentication.Authenticate(txtKullanici.Text, txtSifre.Text) Then
    FormsAuthentication.RedirectFromLoginPage(txtKullanici.Text, chkHatirla.Checked)
Else
    lblMsg.Text = "Kullanıcı adı ve/veya şifre yanlış."
End If
```

```
Public Shared Function Authenticate( _
   ByVal name As String, _
   ByVal password As String _
   ) As Boolean
```

```
Public Overloads Shared Sub RedirectFromLoginPage( _
   ByVal userName As String, _
   ByVal createPersistentCookie As Boolean _
)
```

```
Public Overloads Shared Sub RedirectFromLoginPage( _
   ByVal userName As String, _
   ByVal createPersistentCookie As Boolean, _
   ByVal strCookiePath As String _
)
```

20

Veritabanında Kayıtlı Olan Kullanıcı Listesi ile Kullanıcı Girişi Yapmak

- **<credentials> düğümünün kullanılmasına gerek yoktur. Bu düğüm sadece FormsAuthentication.Authenticate fonksiyonu tarafından kullanılır.**

```
<authentication mode="Forms">
      <forms loginUrl="UyeGirisi.aspx" />
</authentication>
<authorization>
      <deny users="?" />
</authorization>
```

```
If KimlikDenetle(txtKullanici.Text, txtSifre.Text) Then
    FormsAuthentication.RedirectFromLoginPage _
    (txtKullanici.Text, chkHatirla.Checked)
Else
    lblMsg.Text = "Kullanıcı adı ve/veya şifre yanlış."
End If
```

```
Function KimlikDenetle(ByVal kullanici As String, _
 ByVal sifre As String) As Boolean
    'veritabanı kontrol işlemlerini yap
    'eğer kimlik geçerli ise
    '   return true
    'aksi halde
    '   return false
    Return True 'her koşulda giriş yap
End Function
```

21

Kullanıcıyı Uygulamadan Çıkarmak

- **Önce oturum çerezleri ve kalıcı çerezler kaldırılır.**
- **Daha sonra ise sonucun sayfaya yansıması için şu anda bulunulan sayfa tekrar çağrılır.**

```
FormsAuthentication.SignOut()
Response.Redirect(Request.RawUrl)
```

- Request.RawUrl şuanda bulunulan sayfanın adresini döndürür.
- Örneğin
  - **http://cengiz/b2b-web/siparislerim.aspx** sayfasında bu özellik **/b2b-web/siparislerim.aspx** değerini döndürür.

22

Alt Klasörlere Erişim İznini Ayarlamak

- Alt klasöre farklı bir erişim yetkisi vermek için
  - Alt klasör içerisinde de bir Web.config dosyası oluşturulabilir.
  - Ana dizindeki Web.config dosyası ile farklı dosya ve alt klasörler için yetkilendirme yapılabilir.
    Bunun için **<configuration>** düğümü içerisindeki **<location>** düğümü kullanılır.
- Web uygulamasının kimlik denetim sistemi sadece ana klasörde bulunan Web.Config dosyası ile değiştirilebilir.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<system.web>
     <!-- buraya diğer düğümler eklenebilir -->
     <authorization>
          <allow users="CengizHan"></allow>
          <allow users="RobHoward"></allow>
          <deny users="*" />
     </authorization>
     <!-- buraya diğer düğümler eklenebilir -->
</system.web>
</configuration>
```

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<!-- buraya diğer düğümler eklenebilir -->
<location path="yonetim">
     <system.web>
          <authorization>
               <allow users="CengizHan" />
               <allow users="RobHoward" />
               <deny users="*" />
          </authorization>
     </system.web>
</location>
<!-- buraya diğer düğümler eklenebilir -->
</configuration>
```

```
<location path="Siparislerim.aspx">
     <system.web>
          <authorization>
               <deny users="?" />
          </authorization>
     </system.web>
</location>
```

23

Gözden Geçirme

1. ASP.NET ile hangi kimlik denetim sistemleri kullanılabilir.
2. Windows tabanlı kimlik denetim sistemi hangi durumlarda kullanılmalıdır?
3. Form tabanlı kimlik denetim sistemi hangi durumlarda kullanılmalıdır?
4. Form tabanlı kimlik denetim sistemi nasıl oluşturulur? Açıklayınız.
5. Form tabanlı kimlik denetim sisteminde bir alt klasörün güvenlik ayarları nasıl ayarlanır.

24