ASP Dilinde Sql Injection Ve Korunma

MUSTAFA SERHAT DÜNDAR, <SERHAT AT SERHATDUNDAR DOT COM>,
21/06/2009

SQL NEDİR?

SQL bir dildir. Web sayfalarının veritabanları ile ilişkisini kurar. Structured Query Language kelimelerinin baş harflerinden oluşur. Web sayfasının veritabanın'dan veri çekmesi, veri yazdırması, güncellemesi gibi işleri görür. SQL veritabanında bilgiler tablolarda saklanır. Tablolarda bilgilerin saklandığı sütunlar vardır. Her tablo ve sütunun bir ismi vardır. SQL enjeksiyon bu tabloların bir ya da daha fazlasında değişiklik yapma manasına gelir.

Temel olarak 2 şekilde meydana gelen bu saldırılar :

Form injection : Veri girebildiğimiz tüm formlar.

URL injection : http://www.saldirilacakadres.com/yazilar.asp?NO=10

En çok kullanılan kodlar :

INSERT - Veri Ekler.

DELETE - Veri Siler.

ORDER BY – Şuna şuna göre sırala anlamındadir.
SELECT – Malum, seç.
UPDATE – Güncelle.
AND – Matematikte olduğu gibi her iki durumunda sağlanması veya sağlanmaması. Ve.
OR – Veya. Her iki durumda biri sağlanır, diğeri sağlanmaz veya tam tersi gerçekleşir.

Form'a yönelik saldırılarda girilen verilerin doğruluğunu veritabanından kontrol eden herhangi bir <form> arıyoruz.

Url'ye yönelik saldırılarda ise "=" kısmı bizi ilgilendiriyor. Saldırgan yöntemlerimiz = karakteri üzerine olacak.

SQL CÜMLECİKLERİNE YÖNELİK BLACKLİSTİNG UYGULANMASI VE OLUŞTURDUĞU ZAAF

Bu yazımızda Sql Inj. hakkında hiç birşey bilmediğinizi varsayarak yola çıkıyorum. Tabii ki asp bilmeniz gereklidir. Sql injection`u sql server`ın meta karakteri olan (`) meydana getiriyor. Sql server`da bundan etkilenmemek için tek tırnak yerine " kullanmamız gerekiyor. Bu yazımızda sql servera tek tırnağı çift tırnak gibi algılamasını sağlatacak fonksiyonu yazacağız. Bu sayede güvelik açığımız kalmayacak.

Tabii ki bundan önce en önemli güvenlik zaaflarından biri olan blacklisting yöntemini anlatacağım.

Blacklisting = Kara Liste Yöntemi

Yani bir takım sql cümleciklerini kara listeye ekleyerek çalışmalarını engelliyoruz. Bu yöntem sql injection`u engelleme de en büyük zaafları oluşturur (xss içinde blacklisting vardır ve O da aynı şekilde güvenlik zaafıdır).

Öncelikle en basit şekilde yazılmış üyelik sistemimizi inceleyelim.

Yazıyı daha iyi anlamanız açısından üyelik sistemini biraz açıklayalım :

Giriş yapıp içeriğini görüntülemek istediğimiz sayfa olan sayac.asp'dir.

```
<body><html>
```

```
<!–#include file="ayar.asp"–>
```

```
<center><table border="0" width="100%" cellpadding="0” style="border-collapse: collapse" height="99”>
```

```
<%
IF Not Session ("girdinmihocam") = “evetgirdim" Then
%>

<form action="giris.asp" method="post">
<tr><td><p align="right"> Kullanıcı adı : </td>
<td> <input type="text" name="kullanici" size="20”></td></tr>
<tr><td><p align="right"> Parola : </td>
<td> <input type="text" name="parola" size="20”></td></tr>
<tr><td><p aling="right"> </td>
<td> <input type="submit" value="Giriş"></td></tr></form>
```

```
<%
Else
%>

<tr><td> </td><td>Hoşgeldiniz</td></tr>
<center><tr><td>& ; ; ;nbsp;</td><td>
<p align="center"> <a href="logout.asp" style="text-decoration: none">
<font color="#FF0000”>Oturumu Kapat</font></a></td></center>

<%
END IF
%>

</table></center></body></html>
```

Include ettiğimiz ayar.asp :

```
<%
Set baglanti = Server.Createobject("ADODB.Connection")
baglanti.open “Provider=Microsoft.Jet.Oledb.4.0;Data Source=" &
Server.MapPath("uyelik.mdb")
%>
```

Üyelik.mdb içinde üyeler adında bir tablomuz var.5 adet sütundan oluşuyor.id – kullanıcı – parola – giris_tarihi ve onay

Giris.asp sayfamız :

```
<!– #include file="ayar.asp" –>
<%
kullanici = Request.Form("kullanici")
parola = Request.Form("parola")
If kullanici="" or parola="" Then
Response.Write “Lütfen boş alan bırakmayın !"
Response.End
END IF
sql="Select * From uyeler Where onay=1 and kullanici = `"& kullanici &"` and parola = `"
&parola& “`"
Set Kontrol = Baglanti.ExeCute(sql)
IF Kontrol.eof Then
Response.Write “Böyle bir kullanıcı yok be gülüm"
Else
Session ("girdinmihocam") = “evetgirdim"
Session ("id") = Kontrol("id")
```

```
Session ("kullanici") = Kontrol("kullanici")
Response.Redirect “../fentanyl/index/index.asp"
END IF
%>
```

Şimdi gelelim asıl mevzuya.

Sayac.asp sayfasını çağırdığızda karşınıza kullanıcı adı – parola soran sayfa geliyor.

Buraya normalde admin-admin yazarak giriş yaparız ve index.asp`ye yönleniriz. Daha sonra geri dönüp sayfayı yenilediğimizde sayac.asp`ye giris yapmış oluruz.

(Bu kendi kodladığım bişey. Böyle bi saçmalığa gerek yok aslında :)

Boş alan bırakarak giriş yapmaya çalışınca `Lütfen boş alan bırakmayın` hatası alıyoruz.Peki bunu sağlayan kodlar neler?

```
KodIf kullanici="" or parola="" Then

Response.Write “Lütfen boş alan bırakmayın !"

Response.End
END IF
```

Eğer kullanici değeri boş bir değere eşitse `lütfen boş bırakmayın` yazdırıyoruz ve response.end komutuyla komutu durduruyoruz. End if ile şart cümleciğimizi kapatıyoruz.

Bu mantıkla biz bir blacklisting yapabiliriz.

Eğer kullanıcı=x ise `lütfen sql denemeyin` gibi bir mantıkla bunu uygulayacağız.

Çok bilindik ve en basit sql cümleciklerinden biri olan `OR` ile giriş yapmayı deneyelim.

Kullanıcı adı ve şifre değeri yerine bunu yazınca giriş yapmış oluyoruz. Demekki üyelik sistemimiz bu cümleciğe karşı bir zaaf taşıyor.

Bunu engellemek için daha doğrusu engellediğini düşünenlerin en çok düştüğü hata tek tek cümlecikleri üyelik sistemine tanıtmak.

Örnek :

```
KodIf kullanici="`OR`" or parola="`OR`" Then
Response.Write “Lütfen sql denemeyin !"
Response.End
END IF
```

Bu iki kod parçasını sayfanıza eklediğizde artık `OR` kodu ile giriş yapılamayacak. Peki bu güvenlimi ? Diyelimki bütün bilindik sql kodlarını tek tek blacklisting yaptınız güvende olduğunuzu düşünüyorsunuz peki sizi hacklemeye düşünen kişi ya sizden biraz farklı düşünüyorsa ?

`OR` bu kod ile giriş yapamıyoruz peki (`OR`) parantez içindeki kod ? Sonuna eklediğim boşluklar dahil bunu kullanıcı adı ve parola alanına aynen yazıyoruz. Tabiki parantez işaretleri hariç.

Denediğiniz zaman bu kod ile gayet güzel bir şekilde giriş yapıldığını göreceksiniz. Hadi diyelim bu ihtimalide düşünüp bunuda engellediniz. Peki aşağıda parantez içinde vereceğim kodların hepsini nasıl engelleyebilirsiniz ? Kodun sonundaki boşluk sayısı değiştikçe sizin yaptığınız blacklisting çuvallıyor.

(`OR`)
(`OR`)
(`OR`)
(`OR`)

yaptığım şey sadece boşluk sayısını arttırmak.Her koyduğum boşluk için yeni bir sql açığı meydana geliyor. Demekki cümleciğe yönelik blacklisting uygulaması aslında büyük bir güvenlik zaafıymış.

Şimdi aklınızdan `Peki ben giris.asp içinde boşluk karakteri yazılmasını engellersem?` diye düşünebilirsiniz. Tamam hadi sizi kırmayalım ve boşluk karakterini de engelleyelim. Peki sizce bütün bilindik sql cümleciklerini engelleyebilir misiniz?

`OR`
`or
`or“`
““or“or`

Tamamen kafanızdan türetebileceğiniz zilyar tane sql cümleciği var.

Bu verdiğim cümleciklerin tamamı bu yazıyı yazarken kendi deneyerek 2-3 sn`de bulduğum şeyler. Sizde deneme yanılma ile sql kodları bularak bu tip basit üyelik sistemlerini aşabilirsiniz.

Demekki ne olursa olsun üyelik sistemimizi blacklisting uygulamalarına emanet etmeyeceğiz.

Peki ne yapmalıyız ?

KARAKTERE YÖNELİK BLACKLİSTİNG UYGULAMASI

Tek tırnak ile çift tırnağı ve diğer zaaf oluşturabilecek kritik işaretleri değiştirecek bir fonksiyon bizim işimizi görür :

```
<%
function guvenlik(degistir)
degistir = replace(degistir,chr(13),"")
degistir = replace(degistir,chr(34),"")
degistir = replace(degistir,chr(39),"")
degistir = replace(degistir,",","")
degistir = replace(degistir,"/","")
degistir = replace(degistir,"\","")
degistir = replace(degistir,"?","")
degistir = replace(degistir,"*","")
degistir = replace(degistir,"‘","")
degistir = replace(degistir,"OR","")
degistir = replace(degistir,"AND","")
degistir = replace(degistir,"%","")
degistir = replace(degistir,"&","")
degistir = replace(degistir,"<","")
degistir = replace(degistir,"$","")
degistir = replace(degistir,">","")
degistir = replace(degistir,"=","")
degistir = replace(degistir,"!","")
degistir = replace(degistir,"–","")
degistir = replace(degistir,"#","")
degistir = replace(degistir,"like","")
degistir = replace(degistir,"drop","")
degistir = replace(degistir,"create","")
degistir = replace(degistir,"modify","")
degistir = replace(degistir,"rename","")
degistir = replace(degistir,"alter","")
degistir = replace(degistir,"cast","")
degistir = replace(degistir,"join","")
degistir = replace(degistir,"union","")
degistir = replace(degistir,"where","")
degistir = replace(degistir,"insert","")
degistir = replace(degistir,"delete","")
degistir = replace(degistir,"update","")
guvenlik = degistir
end function
%>
```

Ziyaretçi "delete", "update", "rename", "like", "?" gibi karakterleri kullanarak giriş yapmaya çalıştığında biz bu değerleri içi boş değere dönüştürüyoruz.

Bu kod parçası sıradan bir güvenlik sistemidir fakat ilk aşamada bizim işimizi görür. Giriş yapılan paneldeki form'u denetleyen kod parçasıdır.

Bunun ikinci ve çok benzer bir uygulanışı daha vardır :

Gelen verileri temizleyen bir fonksiyon kullanmaktır.

```
Function Temizleyelim(Verimiz)
Temizleyelim = Replace(Verimiz, "'", "''")
Verimiz = Temizleyelim
End Function
```

Filtreden Geçmeden Doğrudan Veriyi Çekmek : `Verimiz = REQUEST.FORM("Verimiz")`

Filtreli Hali : `Verimiz = Temizleyelim(Request.Form("Verimiz"))`

REQUEST.QUERYSTRİNG FİLTRELEME

Örnek olarak yazilar.asp şeklinde bir sayfanızın olduğunu düşünelim. yazilar.asp sayfasının içinden yazilar_Guncelle.asp?NO=1 şeklinde bir güncelleme fonksiyonumuz olsun.

request.querystring kısmına aşağıdaki şekilde filtreleme uygulamazsanız, link üzerinde çeşitli oynamalar ile Sql İnjection saldırısına maruz kalabilirsiniz.

* En çok kullanılan sql injection yöntemidir.

Sizin koymuş olduğunuz kod şu şekilde ise; NO = REQUEST.QUERYSTRING("NO") bu kodu aşağıda verdiğimiz şekilde değiştirin :

```
NO = REQUEST.QUERYSTRING("NO")
IF Not IsNumeric(REQUEST.QUERYSTRING("NO")) THEN
response.write "Bu NOda bir yazi yok, yoksa Sql mi deniyorsun? (:"
response.end
END IF
```

Buradaki kodumuz özetle eğer NO=x kısmında X yerine numeric yani rakam olmayan bir karakter girildiğinde bunu engellemekten ibarettir.

M.SERHAT DÜNDAR