ASP.NET Header Bilgileri

Onur Yılmaz, Aralık 2009, WGT E-Dergi 3. Sayı

ASP.NET ile geliştirilmiş web uygulamalarına yapılan isteklere dönen cevaplar neticesinde, uygulamanın ASP.NET Version bilgisini öğrenmek mümkündür.

```
HTTP/1.1 200 OK =>
Connection => close
Date => Tue, 08 Dec 2009 00:39:17 GMT
Server => Microsoft-IIS/6.0
X-Powered-By => PleskWin
X-AspNet-Version => 2.0.50727
Cache-Control => private
Content-Type => text/html; charset=utf-8
Content-Length => 19752
```

Bu bilginin dışarıya sızdırılmasının iki temel sonucu vardır;

- Her yapılan istek sonucunda ASP.NET Version bilgisi cevap olarak döneceği için, normalden daha fazla ziyaretçi alan uygulamalar için ekstra bant genişliğinin oluşması demektir.
- İkinci ve bizim inceleyeceğimiz sorun ise bu tarz bir bilginin sızması sonucunda uygulamaya yönelik spesifik atakların yapılmasının kolaylaşacak olmasıdır.

ASP.NET Version Bilgisinin Sızdırılmasının Sonuçları

ASP.NET Version bilgisinin öğrenilmesi sonucunda uygulamanın kendisinden kaynaklanan güvenlik açıklıklarına yaklaşımın değişeceği gibi, ASP.NET`in kendisinden kaynaklanan güvenlik açıklıklarının denenmesi konusunda daha spesifik yaklaşılacaktır.

ASP.NET platformu, güvenlik açısından, diğer platformlara göre birçok konuda iyi olmasına rağmen[1], ASP.NET`in güvenlik için getirdiği özelliklerin yeterli olmadığı durumlarda bulunmaktadır. Örneğin; ASP.NET istek denetimini[2] yeterli gören bir geliştirici, uygulamasına bu denetimden başka herhangi bir güvenlik denetimi yapan fonksiyon / bileşen eklemediyse, versiyon bilgisinin elde edilmesi sonucunda uygulamadaki birçok alana, ASP.NET`in kendisinden kaynaklanan girdi denetimi zayıflığını kullanarak[3] atak yapması ve başarıya ulaşması kaçınılmazdır.

Bu noktada geliştiricinin yanlışıyla beraber ASP.NET`in kendisinden kaynaklanan zayıflığının birleşimi sonucunda atak yapılsa da, bazı durumlarda geliştiricinin elinde olmayan durumlarda oluşabilir.

Geliştirici, uygulamadaki kimlik denetimiyle ulaşılabilen bir dizini güvenli olarak addederken, ASP.NET Framework 1.0 ve 1.1`de bulunan güvenlik zafiyetiyle[4] bu denetimi aşarak ilgili dizine ulaşmak mümkün olmaktadır.

Geliştiricinin, web uygulama güvenliği konusunda kaynak kod tarafında yeterince bilgili ve dikkatli olduğunu varsayarsak, platformun kendisinden kaynaklanan zayıflıkları takip etmediği ve ilgili çözümleri uygulamaya entegre etmediği müddetçe, uygulamanın yeterince güvenli olduğunu söylemek yanlış olacaktır.

ASP.NET Version Bilgisinin Sızmasını Engellemek

ASP.NET konfigürasyon dosyası olan web.config`in, <system.web> elementi sınırları içerisine;

```
<httpRuntime enableVersionHeader="false" />
```

satırını eklemeniz, uygulamanızdan ASP.NET Version bilgisinin sızmasını engelleyecektir. [5]

Aynı şekilde eğer ASP.NET MVC yapısı kullanıyor ve ASP.NET MVC Version bilgisinin de sızmasını engellemek istiyorsanız, global.asax dosyasında ki Application_Start olayının içerisine;

```
MvcHandler.DisableMvcResponseHeader = true; //C#
MvcHandler.DisableMvcResponseHeader = True //VB
```

satırını eklemeniz yeterli olacaktır.

Son Durum

Gerekli konfigürasyonlar yapıldıktan sonra uygulamamıza tekrar istek göndererek sonucu kontrol edebiliriz. [6]

```
HTTP/1.1 200 OK =>
Connection => close
Date => Tue, 08 Dec 2009 01:27:51 GMT
Server => Microsoft-IIS/6.0
X-Powered-By => PleskWin
Cache-Control => private
Content-Type => text/html; charset=utf-8
Content-Length => 19752
```

Referanslar

[1] http://ferruh.mavituna.com/aspnet-guvenligi-ve-platform-tasarimi-oku/
[2] http://docs.google.com/View?id=d4w2g9c_22d5dv22gr
[3] http://www.microsoft.com/technet/security/bulletin/ms07-040.mspx
[4] http://www.securityfocus.com/bid/11342/info
[5] http://dotnetperls.com/x-aspnet-version
[6] http://www.webconfs.com/http-header-check.php