

Güvenli ASP.NET Prodüksiyon Ortamı

Mesut Timur, Ağustos 2009, WGT E-Dergi 1. Sayı

Web Uygulaması Güvenliği

Web uygulaması güvenliği sağlanırken genellikle akıllara ilk gelenler çeşitli kod enjeksiyonu zafiyetleri ve girdi doğrulama temelli yaklaşımlardır. SQL Injection'ı kesmek için parametrik sorgular, XSS'i kesmek için output encoding, LFI & RFI için doğru şekilde girdi doğrulama ve CSRF için de kritik fonksiyonalteler için token uygulaması.

Oysa dünyanın en güvenli uygulamasını yazmış olsanız dahi, sunucu üzerinde tutulan bir klasörün izinlerini doğru şekilde düzenlemediyseniz , hack'lenmeniz işten bile değil ! Bundan dolayı işin kod kalitesi ile ilgili kısmını düşünürken, konfigürasyonel boyutunu da es geçmemek gerekmektedir.

ASP.NET

Günümüzde web uygulamaları bir çok programlama dili vasıtasıyla yazılabilmekte ve bir çok üretim ortamında çalışabilmektedir. Bu dillerden ve platformlardan önemli bir tanesi ise ASP.NET 'dir.

ASP.NET ortamında geliştirilmiş uygulamaların konfigürasyon bilgileri web.config isimli dosyada tutulur.

Web.config

Söz konusu XML dosyası uygulamanın durum bilgisi, güvenlik konfigürasyonu, derlenme ayarları ve uygulama-spesifik bilgiler yer alır. Bir web sunucu üzerindeki tüm uygulamalar aynı konfigürasyon dosyasını kullanmak zorunda değildir. Farklı uygulamalar için farklı Web.config dosyaları tanımlanabilir.

İlgili konfigürasyon dosyasındaki ayarların doğru şekilde belirlenmemesi durumlarında çeşitli güvenlik zafiyetleri ortaya çıkabilmektedir. Kimi güvenlik zafiyeti ise herhangi bir hatalı konfigürasyon beklemeden, varsayımlı ayarlarda meydana çıkmaktadır.

Potansiyel zafiyetlerin bir listesi şu şekildedir:

- . Hata mesajlarından ve uygulamadan çeşitli şekillerde bilgi sızmaları
- . Uygulama ile istemci arasındaki trafiğin ifşası
- . Yazılımın saldırganların kontrolüne geçmesi halinde, sunucunun tüm kontrolünün kaybedilmesi
- . Oturum ile ilgili konular
 - o Oturumun manipüle edilmesi
 - o Oturumun çalınması
 - o Zaman aşımı olmaması sonucu çalınan çerezin kullanılması
- . Web çerezleri ile ilgili konular

- o Çerezlerin XSS saldırısı ile çalınması
- o Çerezlerin ağ trafiği dinlenilerek çalınması
- . İstek hırsızlığı
- . Hassas dosyaların ifşası.
- . Trafiğin dinlenilmesi
- . Web.config içindeki kritik bilgilerin sızdırılması [1]

Web.config dosyasının doğru şekilde konfigüre edilmesiyle söz konusu zafiyetlerin bazılarında sonsuza dek, bazılarında ise bir çok durumda korunmak mümkün olacaktır.

Web.config dosyalarındaki problemlerin otomatik şekilde incelenmesini gerçekleştirmek üzere WCSA (Web.config Security Analyzer) hazırlandı. WCSA kendisine parametre olarak geçilen Web.config dosyalarını inceleyip varolan konfigürasyonel güvenlik problemlerini ortaya çıkarıyor.

Proje'nin ana sayfası <http://code.google.com/p/wcsa/> olup, download linklerine de erişebilirsiniz.

```

C:\Windows\system32\cmd.exe
C:\Users\Mesut\Executable>ASPauditor.exe SampleConfigs\n2-web.config
AA-00:Debugging Enabled-->compilation.true
AA-01:Clear-text credentials-->credentials.Clear
AA-30:Hardcoded Credentials Used-->user.
AA-02:Custom Errors Disabled-->customErrors.Off
Vulnerabilities with not specified(default) options are coming
AA-05:Cookieless Authentication may be enabled-->forms
AA-06:Doesn't Require SSL for Auth. Cookies-->forms
AA-07:Non-Unique Authentication Cookie Used-->forms
AA-08:Sliding Expiration Used-->forms
AA-09:Liberal Path Defined-->forms
AA-14:Web cookies are not HttpOnly-->httpCookies
AA-15:Web cookies doesn't require SSL-->httpCookies
AA-19:ViewState may not be encrypted-->pages
AA-21:roleManager cookies doesn't Require SSL-->roleManager
AA-22:roleManager Cookie Sliding Expiration Used-->roleManager
AA-26:roleManager cookie path is Liberal-->roleManager
AA-29:Your web application's trust level is higher than Minimal-->trust
C:\Users\Mesut\Executable>_

```

Çıktı olarak verdiği HTML raporda güvenlik zafiyetleri, zafiyetlerin tanımları, yapılması gereken güvenli konfigürasyon ve referanslar veriliyor.



Sonuç

Güvenlik bir bütün olarak düşünölmelidir. Güvenli web uygulamaları geliştirirken, güvenli kod pratiklerinin uygulanmasının yanında sürecin konfigürasyonel boyutu da göz önünde bulundurulmalıdır.

Referanslar

[1] <http://www.onuryilmaz.info/icerik/61-web-config-dosyasinin-icerigini-sifrelemek.aspx>