Web Shell Uygulamalarından Korunun

Oğuzhan YILMAZ, <oguzhan at maestropanel dot com>, maestropanel.com, 11/09/2012

Giriş

Piyasada Shared Web Hosting sunucuları için en büyük tehditlerden bir tanesi de yanlış konfigürasyonlardan yararlanan kötü niyetli ASP, .Net veya PHP uygulamaları bilinen adıyla Web Shell. Her ne kadar işin erbabı özel olarak hazırlanmış shell uygulamaları kullanıp Special Defacement istatistiğini arttırsa da piyasada bebeler için olan shell uygulamaları da iş görebiliyor (Örneğin: *AspxSpy,CyberSpy5,EFSO.ASP,R57Shell.PHP,C99Shell.PHP gibi*).

Bu yazıda kötücül script'lerden korunmak için Web Sunucularında temel olarak yapılması gereken bazı ayarlamalara değineceğim.

.Net Güvenliği

Net ile geliştirilmiş olan Shell uygulamaları yoğun olarak Framework içindeki **System.IO**, **Microsoft.Win32**,**System.Net**,**System.ServiceProcess**,**System.Management** sınıf kütüphanelerini kullanır.
Bu sınıf kütüphanelerinin Shared Hosting'de güvenli bir şekilde çalışması için içinde Code Access Security dediğimiz XML dosyaları vardır.

> .Net 4.0 için `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config` çindedir

Eğer Web sunucunuzu .Net Shell'lerin den korumak istiyorsanız bu sınıf kütüphanelerinin güveliğini kesinlikle sınırlamanız gerekmektedir. Bunun en basiti de varsayılan olarak **Full** gelen .Net Trust Level'ini **Medium**'a çekmektir.
Medium'a çektiğiniz taktirde Shell uygulamalarının sistem derinliklerine ulaşması büyük ölçüde engellenmiş olur. Tabi bu durum normal uygulamalarında çalışmasını engelleyebilir bu durumda *trust.config* dosyalarında bazı oynamalar yapabilirsiniz duruma göre.

Burada dikkat edilmesi gereken Code Access Security izinleri verirken tüm Framework sürümlerinde (v2.0 ,v4.0) ve platformlarında (x68, x64) verilmesidir.

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web_mediumtrust.config
C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG\web_mediumtrust.config
C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\web_mediumtrust.config
C:\Windows\Microsoft.NET\Framework\v2.0.50727\CONFIG\web_mediumtrust.config
```

Örnek olarak **FileIOPermission**'ı inceleyelim. Aşağıdaki kural uygulamanın sadece çalıştığı dizinde işlem yapmasına izin verir. Böylelikle Shell bir üst klasöre çıkma istediğinde **SecurityException** alır.

```
<IPermission class="FileIOPermission" version="1" Read="$AppDir$"
Write="$AppDir$" Append="$AppDir$" PathDiscovery="$AppDir$" />
```

Burada ufak bir not düşmekte fayda var:
Medium Trust'a çektiğiniz .NET Level'i içinde bazı temel .NET kütüphaneleri yoktur. Bunları aşağıda veriyorum.

```
System.Data.OleDb, System.Configuration, System.Net, System.Net.Mail
```

bunların PermissionSet dediğimiz parametrelerini elle girmeniz gerekir ki normal .net uygulamalarınız sorunsuz çalışabilsin.

Trust config'de yapılan değişiklikler (Buradaki ilgili config web_mediumtrust.config oluyor) bir alt uygulama tarafından ezilebilir (override) bunun için **allowOverride** özelliğini "false" olarak işaretlemek gerekir yoksa web sitesinin en alt seviyedeki web.config root web.config'den aldığı miras neticesinde Trust Level değerini Full olarak değiştirir ve güvenlik yüzeyinin artmasına neden olabilir.

Bunu önlemek için:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\web.config içinden

<location allowOverride="true">

değerini

<location allowOverride="false">

şeklinde değiştirin.
```

Diğer önemli olan konu ise sunucu üzerindeki NTFS (IT Procular ACL demeyi sever. Ben haklar diyorum) kullanıcı hakları.
IIS'in Impersonate (bir kullanıcının başka bir kullanıcı gibi hareket etmesi olayı) özelliği Anonymous kullanıcı seviyesinden User Grubu kullanıcı seviyesine kadar çıkabilen bazı özellikler içerir bu nedenle User grubuna yetki verilmiş yerlere direkt erişim sağlanabilecek hale gelebilir.

Bu nedenle gerekmeyen her yerde User grubunun haklarını kaldırmanız Shell uygulamalarını sunucunun diskinin orasına burasına erişmesini büyük ölçüde engeller.

Örneğin: IceWarp Merak Mail'in çalıştığı klasörün (`C:\Program Files (x86)\IceWarp`) bir üst klasörden aldığı hak miraslarını iptal edip sadece **SYSTEM** ve **Administrator** grup haklarına sahip olmasının sağlanması gerekir ki herkes her önüne gelen config dosyalarını okuyamasın. (Tabi servisin kullandığı başka kullanıcılar varsa bunu da göz önünde bulundurmak gerekir.)

- Sunucudaki klasörlerin hak kontrollerini yapmak için AccessEnum (http://technet.microsoft.com/en-us/sysinternals/bb897332) aracını kullanabilirsiniz.

Aynı şekilde IIS'inde unutmamak gerekir.
IIS'i korumak için `C:\Windows\System32\inetsrv\config` klasörü içindeki *.config uzantılı dosyaların haklarından **Users** grubunu kaldırmak doğru olur. Gerçi varsayılan olarak öyle geliyor fakat yine de bazı uygulamalar kendi kafasına göre değiştirebiliyor.

Bazı Shell uygulamalarından mevcut disklerin listelenmesi Group Policy'den engellenebilir. Şöyle ki;

Windows Tuşu + R'ye bastıktan sonra gpedit.msc yazın ve **User Configuration** > **Administrative Templates** > **Windows Components** > **Windows Explorer** > *Hide these specific drives in My Computer* > Restrict All Drives seçeneğini "Enable" yapıp "Restrict All Driver" komutunu verin. Bu Bilgisayarıma tıkladığınızda size Diskleri göstermez biraz rahatsız edici ama yine de ek bir önlem.

Shell uygulamaları sunucu üzerinde herhangi bir işlem yapmak için genelde sistemde hazır olan çalıştırılabilir uygulamaları kullanırlar (cmd.exe, cacls.exe, ping.exe gibi) bu hem Code Access Security'i atlatmak hemde daha az kod yazmak adına iyidir. Bu nedenle sunucunuzdaki C:\Windows\System32*.exe ve C:\Windows\Syswow64*.exe klasörlerinde *.exe uzantılı dosyaların erişimlerini minimum'a indirmek doğru bir hamle olur.
Bu kritik klasördeki *.exe erişimlerini değiştirmek için öncelikle dosyaların sahipliğini Administrator'a çevirmek gerekir.

Sistemde Administrator hakları ile giriş yapmışken aşağıdaki komutu verin:

```
takeown /F c:\Windows\system32*.exe
```

Bu komut System32 klasöründeki tüm .exe uzantılı dosyaların sahipliğini Administrator'a çevirecektir (Default olarak TrustedInstaller gelir)

Daha sonra aşağıdaki komutu çalıştırın:

```
calcs c:\Windows\System32*.exe /E /D Users
```

Bu komut da System32 klasöründeki tüm .exe uzantılı dosyaların haklarından Users grubunun kaldırır.

PHP Güvenliği

PHP ile geliştirilmiş olan shell uygulamaları içinde **php.ini**'de yapılacak bir kaç ayar var.

Öncelikle sakat fonksiyonları kapatmakla başlamak gerekir. PHP Shell script'lerini incelerseniz çeşitli zaafları sömürmek için kullanılan temel fonksiyonların olduğunu keşfedersiniz yani sorunu kökünden çözmek için bunları tespit edip kapatmak mantıklı bir hareket olacaktır. İşte o fonksiyonlar;

```
exec, dl, passthru, shell_exec, system, eval, popen, fsockopen,
proc_open, proc_get_status, proc_nice, proc_terminate, show_source,
stream_socket_server, symlink, link, lchgrp, lchown, chown, chgrp,
posix_initgroups, posix_kill, posix_mkfifo, posix_mknod, and
posix_setegid, posix_seteuid, posix_setgid, posix_setpgid, posix_setsid,
posix_setuid
```

php.ini dosyasında "**disable_functions**" değerinin karşısına yukarıdaki satırı ekledikten sonra bir çok shell script'ini elemine etmiş oluyorsunuz.

Bir başka ayarda **allow_url_fopen**'ın Off olması buna bağlı olarak da **allow_url_include** 'un da Off olması web sitesi güvenliği ve selameti için iyi olur.

PHP için kilit noktalardan bir tanesi de her web sitesi için müstakil bir php.ini dosyasının yaratılması ve siteyi onun üzerinden çalıştırmak olacaktır. Böylece her sitenin özel **upload_tmp_dir**'ı disable_functions'ı veya disabled_classes'ı olabilir

Referanslar

- http://msdn.microsoft.com/en-us/library/ff649487.aspx
- http://msdn.microsoft.com/en-us/library/dd984947.aspx
- http://msdn.microsoft.com/en-us/library/system.security.permissions.fileiopermission.aspx
- http://wiki.maestropanel.com/MaestroPanel-PHP-Ayarları.ashx

- http://blog.oguzhan.info/?p=20