VERİ TABANLARINI BEKLEYEN TEHLİKELER: ŞIRINGA EDİLEN SQL İFADELERİ

Ömer Utku ERZENGİN[1], Gürhan ÖZDEMİR[2]
ouerzengin@hotmail.com ghan1@hotmail.com

Giriş

İnternet üzerinden veri paylaşımı arttıkça veri güvenliğinin sağlanması önemli bir olgu haline gelmiştir. Bilişim sistemlerini çökertmek, yavaşlatmak üzerine yapılan saldırılarla birlikte değerli olan bilgilerin çalınmasına yönelik saldırılar daha tehlikeli hale gelmiştir. Değerli olan bilgilere günümüzde örnek olarak kredi kartı bilgileri ve banka hesapları verilebilir (Gavin, 2005).

Dış saldırılara karşı güvenlik duvarı (firewall), antivirus ve antispyware gibi çeşitli programlar, SSL (secure socket layer) benzeri şifreleme yöntemleri ve kullanıcı hakların kısıtlanması gibi yaptırımlar mevcuttur (Burgess, 2003). Veri tabanından bilgi çalınmasına karşı yapılabilecek hazır yazılım tarzında çok fazla ürün bulunmaktadır. Tasarımlarına bağlı olarak güvenlik duvarları genelde iç bilgisayarlara her türlü hakkı verirken, SSL tarzında uygulamalar verinin şifrelenmesi ve başka kişiler tarafından okunmasını engellenmesi amacına yönelik olduğundan veri tabanlarına karşı yapılan saldırılara yaptırımları çok fazla değildir.

Veri sorgulamasına ve girişine izin verilen her hangi bir kullanıcı sistemin içinde kabul edilir. Veritabanı uygulamalarında veritabanı genelde sunucu bilgisayar (server) üzerinde bulunmakta ve istemci bilgisayarlardan (client) veri girişi veya sorgulaması yapılmaktadır.

İnternete servis vermek üzere sunucu tabanlı uygulamalardan ASP (Active Server Page) ve PHP (Hypertext Preprocessor) en fazla bilinenleridir. ASP ve PHP tabanlı web programları, derlenmiş kendi başına çalışabilen (executable binary) dosya yaratmadan web üzerinde kullanım sağlarlar (Yuhanna, 2005). ASP ve PHP sunucu tabanlı uygulamalarla beraber çalışan SQL (Structured Query Language) veri tabanları veri girişini ve sorgulamasını kolaylaştırır. SQL veri tabanlarına MySQL, PostgreSQL, MS SQL ve Oracle örnek gösterilebilir. Microsoft Internet Explorer veya Netscape gibi herhangi bir tarayıcı (browser) aracılığıyla bu veri tabanlarına veri girilebilir veya veri tabanlarından veri sorgulanabilir (Achour, 2005).

Veri girişinin ve sorgusunun hızlandırılması amacıyla sunucu tabanlı bilgisayarlarda PHP ve ASP gibi kolay hayata geçirilen uygulamalar bazı temel kurallara dikkat edilmezse veri tabanına karşı saldırılara açık vermektedir (Gavin 2005).

Sunucu temelli SQL veri tabanları ile beraber kullanılan PHP ve ASP uygulamaları programcılık hatalarından (bug) başka güvenlik sorunları vardır (Castano, 1995). ASP ve PHP temelli uygulamalarda veri tabanlarındaki değerli bilgilere izinsiz ulaşmaya olanak veren tasarım boşlukları bulunabilmektedir. Veri girişi ve sorgulaması için geliştirilen uygulamanın bir an evvel uygulamaya konulma isteği güvenlik açıklarını ortaya çıkarmaktadır. Veri tabanlarında ihtiyaç duyulan güvenliğinin sağlanması,

kullanılacak uygulamanın tamamlanma süresini uzatmaktadır. Tasarım aşamasında uygulamada yapılan güvenlik açıklıkları, uygulamanın kullanılması sırasında sorunlara yok açabilmektedir (Anley, 2003, Harper 2002). Veri tabanlarındaki açıklar kredi kartı bilgileri gibi değerli bilgilerin veya adli sicil gibi kişisel bilgilerin çalınmasına neden olmaktadır.

Metot

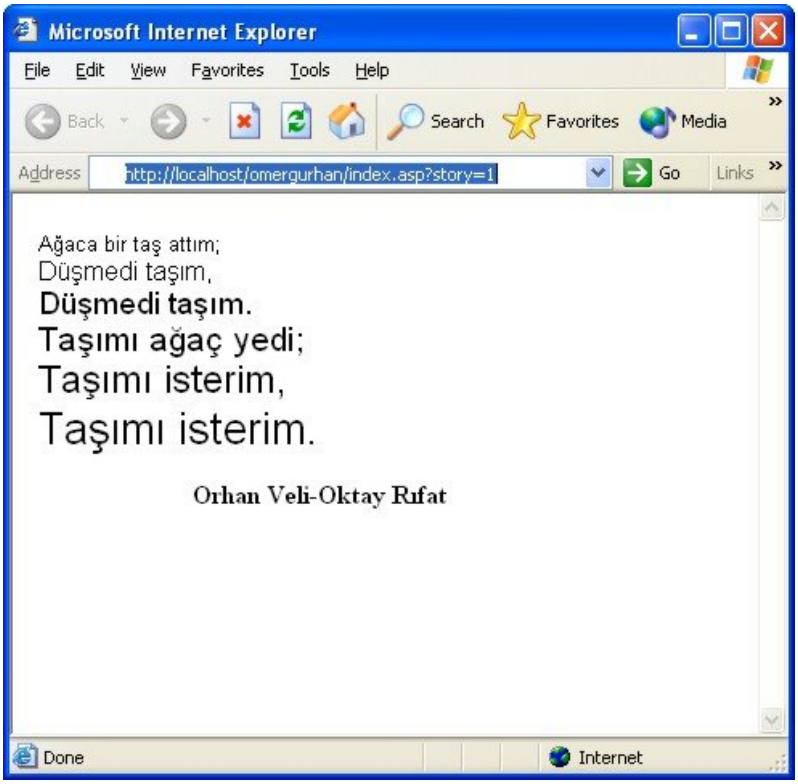
Bu çalışmanın amacı SQL şırıngalarına karşı korunmak için veritabanlarında uyulması gereken güvenlik kurallarıdır. Ayrıca çalışmanın diğer bir amacı veritabanlarına veri girilirken veya veri sorgulanırken kullanılan SQL komutlarının hata veya saldırganlar tarafından değiştirilmesiyle ortaya çıkabilecek zararı engellemektir.

Örnek Bir Veritabanı SQL Şırınga İşlemi

Kullanacağımız örnekte ASP ve SQL Sunucu kurulmuş, ayarları yapılmamıştır.

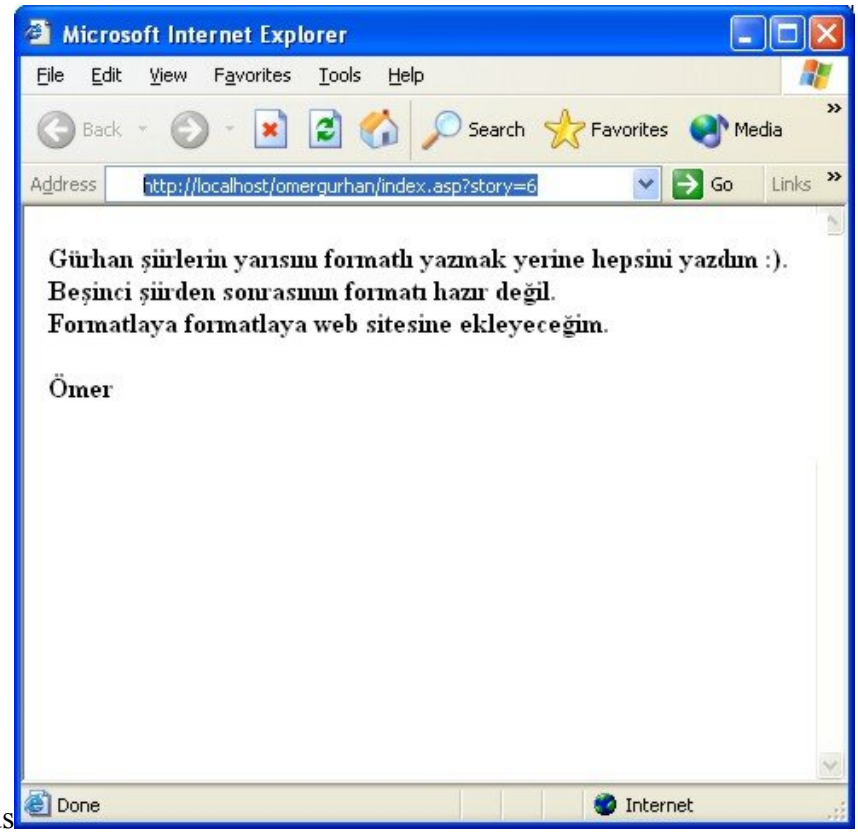
SQL sorgulama komutları temelde "Select *değişken_ismi1, değişken_ismi2* from *veritabanı* where *koşular* " şeklindedir. Kullanılan bilgiler değişken isimleri altında tablolar halinde veritabanlarına konulmaktadır. Bilgi sızdırmak isteyen herhangi saldırgan bu SQL komutunu değiştirmek zorundadır. Bilgi sızdırmak isteyen kişinin bilmesi gereken en önemli kısım değişken ismi, tablo ismi, ve veritabanı ismidir.

Herhangi bir ASP tabanlı bir web sitesinden bir bilgi istenirken adres çubuğuna [http://localhost/omergurhan/index.asp?story=1] şeklinde bir bilgi yazılır. Bu bilgi genelde world wide web anlamına gelen "www" ile başlar. Örnekte kullanılan sunucu ve istemci, aynı bilgisayar üzerinde olduğundan "www" ibaresi yerine "localhost" gelmiştir. Adres çubuğundan yazılan bilgi bize Şekil 1'deki gibi bir web sayfasını döndürür.



Şekil-1 ASP tabanlı bir web sayfasından bilgi isteme

Veri tabanına sızmak isteyen kişi adres çubuğunda girilen ASP komutu "story=1" ifadesini "story=6" olarak değiştirirse Şekil 2'deki bilgi internet tarayıcısına dönecektir. Adres çubuğuna girilen bilgi [http://localhost/omergurhan/index.asp?story=6] şeklinde olacaktır.

a8n5x asus 

Şekil 2 ASP tabanlı bir web sayfasından açıkta olmayan bilgiyi isteme

"story" SQL ifadesinde değişkene denk gelmektedir. Adres çubuğuna "story=1" yerine "story=6" şeklinde bir bilgi yazılması açıkta olmayan veritabanı içeriğinin, veritabanını hazırlayanların istemi dışında elde edilmesini sağlamıştır. SQL komut sisteminde bu müdahale koşuların değiştirilmesiyle olmuştur. Değişkenlerin eşitlik parametrelerini değiştirmek en basit SQL şırınga yöntemidir. Bilgi sızdırmak isteyen saldırgan değişken eşitlik parametrelerine SQL komutları şırınga ederek, değişkenler içinden gerekli bilgilere ulaşabilmektedir.

Veri tabanından bilgi sızdırmak isteyen kişi koşulları değiştirerek adres çubuğuna [http://localhost/omergurhan/index.asp?story=3 AND biryer=3] şeklinde bir bilgi yazarsa, SQL sunucusu aşağıdaki benzer bir hata mesajını döndürür.

Error Type:
Microsoft OLE DB Provider for ODBC Drivers (0x80040E14)
[Microsoft][ODBC SQL Server Driver][SQL Server]Invalid column name 'biryer'.
//localhost/omergurhan/, line 33

SQL komutuyla istenen bu "biryer" değişken bilgisi veri tabanında olmadığından, SQL sunucu internet tarayıcısına hata mesajı döndürecektir. Bilgi sızdırmak isteyen kişi hata mesajı ile karşılaşacağını bildiğinden sonsuza kadar diğer değişken isimlerini deneme şansı olacaktır. Bilgi sızdırmak isteyen kişi, SQL komutuna şırınga ettiği, değişken isimlerini deneme yanılma metoduyla öğrenebilecektir.

Veritabanından bilgi sızdırmak isteyen kişi veritabanına normalde kullanılmayan bir karakteri bilgi olarak gönderebilir. Bu karakter bir çok bilgiyi açığa çıkartabilir. Adres çubuğuna [http://localhost/omergurhan/index.asp?story=3'] şeklinde bir bilgi yazılabilir. " story=3' " şeklindeki bilgi veri tabanının içinde kullanılan tabloların isimlerini internet tarayıcısına hata olarak gönderecektir. SQL suncudan web tarayıcısına dönen hata mesajı aşağıdaki gibi olacaktır.

Institute 2002, Error Type:
Microsoft OLE DB Provider for ODBC Drivers
[Microsoft][ODBC SQL Server Driver][SQL
the character string ' AND gurhan.aID= omer.aID'.
//localhost/omergurhan/.asp, line 25

SQL suncudan internet tarayıcısına dönen hata mesajı tablo isimlerini ortaya çıkarmıştır. SQL şırınga yöntemiyle, veri tabanında kullanılmayan karakterle, saldırgan veritabanları isimlerini öğrenebilmektedir. Veritabanında kullanılan veritabanı isimlerinin elde edilebilmesi herhangi bir koruma yoksa mümkündür. SQL şırınga yöntemleriyle değişken isimlerinin, tablo isimlerinin, veritabanı isimlerinin başka metodlarla da elde edilmesi mümkündür. Veritabanı isimlerinin elde edilebilme kısımı anlatılmayacaktır. Diğer SQL şırınga yöntemlerine değinilmeyecektir.

Çok basit olarak yapılabilen SQL şırınga işlemleriyle veritabanına saldıran kişinin ulaştığı bilgi 3 ana başlık altında toplanabilir.

1. Veri tabanının iki tablodan oluştuğu ve tablo adlarının “gurhan” ve “omer” olduğu.
2. Bu tabloların içindeki değişkenlerin neler olduğu.
3. Veri tabanındaki iki tablonun bağlantılı olduğu.

Açığı olan herhangi bir veritabanına ulaşma şekli sonsuz olarak tarif edilebilir. Veritabanına sızmak isteyen kişi böyle açıktan, SQL sunucusuna birkaç değişik metodla ulaşabilecektir. Veritabanına ulaşımda kullanılan SQL şırınga metodlarının kombinasyonlarının kullanılmasıyla, veritabanındaki bilgilerin önemli bir kısmı kolayca açığa çıkacaktır. Alınabilecek temel önlemler tartışma kısmında anlatılmıştır.

Tartışma

SQL Şırınga Yöntemlerinde Korunmada Uyulması Gereken Bazı Kurallar

Veritabanlarına yönelik güvenlik çözümleri veya uyulması gereken kurallar aşağıda listelenmiştir.

1. Veri girişi ve sorgulamasında kullanılacak uygulamasının Web Tarayıcısı (Web Browser örnek Microsoft İnternet Explorer, Netscape) üzerinden girilirken adres çubuğunun kaldırılması.

Adres çubuğu internet tarayıcısından kaldırılarak bütün işlemlerin internet tarayıcısı üzerinden yapılmasının sağlanması, dışarıdan yapılacak müdahalelerin büyük bir kısmını engelleyecektir.

2. Veri tabanına bağlandıktan sonra SQL ifadelerine bağlı herhangi bir hata mesajının (error message) internet tarayıcısına veya uygulamaya yönlendirilmemesi.

Hata mesajının internet tarayıcısına yönlendirilmemesi veritabanına sızmak isteyen kişilerin hangi konumda olduklarını öğrenmemelerini sağlayacaktır.

3. ComboBox ve ListBox'lardaki veri tabanı ismi ile sunucu veri tabanı isminin aynı olmaması ve veri girişi yapılan yer adları ile veri tabanındaki değişken (variable) isimlerinin aynı olmaması.

İnternet tarayıcılarının tercihlerini bulunduran ComboBox ve ListbBox'lar veritabanı ve değişkenlerle aynı isime sahip olursa veritabanına ulaşılması kolaylaşacaktır.

4. Veri tabanlarında veri girişinde ve sorgulanmasında çapraz veritabanları ve tabloların kullanılması.

Veritabanından bilgi istendiğinde veya bilgi gönderildiğinde bir yerine bir kaç veri tabanı kullanılması ve bunlar arasında doğruluk karşılaştırılması yapılması güvenliği arttıracaktır.

5. Kullanıcı haklarının çapraz veri tabanları için ayrı ayrı tanımlanması.

Kullanılacak birden fazla veritabanı için her aşamada kullanıcı haklarının ayrı ayrı tanımlanması gerekmektedir. Eğer bu işlem bir tek veritabanı üzerinden yürütülürse ve güvenliğin sağlandığı veritabanına sızılacak olursa bütün veritabanının güvenliği tehlikeye düşecektir. Eğer diğer veritabanlarında da güvenlik tedbirleri olursa sızma işlemi gerçekleştirilmeyecektir. Diğer veritabanlarında güvenlik sağlayacağından bir veritabanına sızmak bilgi sızdırmak için yeterli olmayacaktır.

6. Veri girişi yapan kişinin okuma (read access), yazma (write access) hakkının olması değiştirme hakkının süreli olması.

7. Veri girişi yapan kişilerin yanlızca kendi girdikleri verileri okuyabilmesi.

6 ve 7 başlıkar temelde analizlere yönelik istatistksel veritabanları için geçerlidir. Daha önce girilmiş ve doğruluğu kabul edilmiş bilgilerin yanlışlıkla silinmesini engelleyecektir. Hatalı girişler için değişterme hakkı süreli olduğundan daha sonra doğru kabul edilen bilgi üst makamlarca değiştirilebilecektir.

8. Veri girişi hakkı, okuma hakkı, değiştirme hakkı yanlızca bir yönetici (veya merkez) tarafından verilmesi ve hakkların verileceği kişilerin önceden belirlenmesi. Bu yöneticinin veri tabanı güvenlik denetleme (auditing) işine dahil edilmemesi.

Kullanıcıların hakları bir tek merkezden verilmeli ve verilen okuma, yazma, değiştirme hakları kayıt altına alınmalıdır.

9. Veri girişinde veri girenlerin kullandıkları SQL ifadelerinin (SQL statement) hepsinin ayrı ayrı seyir dosyasına (log file) yazılması.

Veri tabanıyla yapılacak her türlü işlem sunucu tarafından kayıt edilmelidir. Bu sayede veritabanında meydana gelecek hata takip edilebilecektir.

10. Tarayıcıdan gelebilecek SQL komutları uzunlukları yazılım tarafından kontrol edilmelidir.

Veritabanına sızmak isteyen birisi SQL komutunu değiştirecektir. İnternet üzeriden gelen SQL komut uzunlukları kontrol mekanizması altına alınacak olursa bir çok saldırgan devre dışı kalacaktır.

11. Güvenlik denetleyecilerin hatalı veri girişlerini, veri girişçisinin işi bittikten sonra veriyi kontrol etmesi.

İstatistiksel veri tabanlarında güvenlik denetçileri girilen veriyi kontrol ederek kullanıma hazır bir hale getirmeleri gerekmektedir.

12. Veri girişçi kendisine ait veri girişi parçasını bitirdikten sonra veri güvenlik denetçisinin yedekleme (backup) işlemi yapması.

Sayısal ortama kaydedilen bilgileri güvenlik altında tutmanın en iyi yolu yedeklenmesi ve yedeğin yedeklenmesidir.

13. İstatistiksel analiz için kullanılacak veritabanının, web uygulamalarından elde edilecek ve asıl olarak kullanılacak veri veritabanından izole edilmesi. Asıl verinin veri tabanına yönetici ve güvenlik denetçileri tarafından düzenlenmesi.

14. Asıl olarak kullanılacak veri tabanı üzerinde yönetici hariç hiç bir veri güvenlikçinin yazma ve silme hakkının olmaması, değiştirme hakkının ise aynı anda iki veya üç veri güvenlikçinin şifresi girildikten sonra veya veri güvenlikçisi ve yönetici şifresiyle elde edilebilmesi

Sonuç

Internet üzerinden geliştirilen uygulamalarda, veritabanları uygulamanın son noktasıdır. Bilgili ve deneyimli bir saldırgan eğer uygulama üzerinde bir boşluk bulursa veritabanındaki bilgilere ulaşmakta zorlanmayacaktır. Saldırgan elde edebileceği yetkilerle girişlerini standart kullanıcı gibi gösterecek ve saklanması kolaylaşacaktır. Veritabanlarını korumak için uygulanacak önlemler uygulamanın işleme konma ve kullanıma açılama süresini ayrıca maliyeti arttıracaktır. Artan süre ve maliyete karşılık eldeki değerli bilgilerin dışarı sızmasını engelleyecektir diğer bir deyişle güvenliğin derecesi artacaktır.

Kaynaklar

1 Achour, M., Betz, F., Dovgal, A., Hojtsy, G.(2005), PHP Manual, Erişim [http://www.php.net/manual/en/]. Erişim Tarihi: 24.12.2005

2. Anley, C. (2003) Advanced SQL Injection In SQL Server Applications, Erişim: [http://www.ngssoftware.com]. Erişim Tarihi: 18.06.2005

3 Burgess, R. C., Small, M. P. (2005) Computer Security In The Workplace, North Charleston, North Carolina: SEO Press

4 Castano, S., Fugini MG., (1995) Database Security, Milan, Addison-Wesley & ACM Press.

5 Friedl, S. (2005) SQL Injection Attacks by Example, Erişim: [http://www.unixwiz.net/techtips/sql-injection.html]. Erişim Tarihi: 24.12.2005.

6 Gavin, Ed. (2005) Protecting Your "Crown Jewels": New Approaches to Securing Critical Data, Erişim: [http://www.bitpipe.com/detail/RES/1129646348_201.html]. Erişim Tarihi: 18.12.2005

7 Harper, Mitchell. (2002) SQL Injection Attacks - Are You Safe? , Erişim:[http://www.sitepoint.com/article/sql-injection-attacks-safe]. Erişim Tarihi: 23.12.2005.

8 Yuhanna, N., Mayer, O., (2005) Protecting Information at the Source: Securing Database Environments Erişim: [http://www.bitpipe.com/detail/RES/1131639771_658.html]. Erişim Tarihi: 12.12.2005