Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 28, S. 213-219, Kasım 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 28, pp. 213-219, November 2021

Research Article

Windows Tabanlı Uygulamalarda SQL Enjeksiyon Siber Saldırı Senaryosu ve Güvenlik Önlemleri

İsa Avcı[1*], Murat Koca[2], Merve Atasoy[3]

[1*] Karabuk Üniversitesi, Mühendislik Fakültesi, Bilgisayar Bölümü, İstanbul, Türkiye, (ORCID: 0000-0001-7032-8018), isaavci@karabuk.edu.tr
[2] Bilim Sanayi ve Teknoloji İl Müdürü, Hakkari, Türkiye (ORCID: 0000-0002-6048-7645), muratkoca30@gmail.com
[3] Karabuk Üniversitesi, Mühendislik Fakültesi, Bilgisayar Bölümü, İstanbul, Türkiye, (ORCID: 0000-0001-9400-5694), merveatasoy48@gmail.com

(1st International Conference on Applied Engineering and Natural Sciences ICAENS 2021, November 1-3, 2021)

(**DOI:** 10.31590/ejosat.995697)

ATIF/REFERENCE: Avcı, İ., Koca, M. & Atasoy, M. (2021). Windows Tabanlı Uygulamalarda SQL Enjeksiyon Siber Saldırı Senaryosu ve Güvenlik Önlemleri. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 213-219.

Öz

Teknolojilerin son yıllarda hızla gelişmesi ile birlikte şirketlere yapılan siber saldırılarda artışlar yaşanmaktadır. Bu saldırılardan en önemlilerinden birisi SQL enjeksiyon saldırılarıdır. Şirketlerin ve kurumların en önemli verilerinin tutulduğu yerler veri tabanlarındadır. Veri tabanlarında tutulan veri çeşidi ve miktarı her geçen gün artmaktadır. Çeşitli siber saldırı yöntemlerle savunmasız veri tabanlarına sızmak mümkündür. Gerekli önlemler alınmazsa özellikle de şirketlerin kendi bünyesinden bu sistemlere sızmak çok zor değildir. Profesyonel anlamda hizmet sağlayan web uygulamalarının birçoğu SQL altyapısını kullanan veri tabanı sorgu yapılarını tercih etmektedir. Birçok web tabanlı program kullanıcı isteğine bağlı SQL altyapısı kullanarak sonuçları döndürmektedir ve geliştiricisine bağlı olarak farklı tasarımlarda kullanıcıların hizmetine sunulmaktadır. Ancak web tabanlı uygulamalarda saldırgan tarafından sisteme girilen bazı zararlı cümleciklerle SQL sorgularına enjeksiyon işlemi yapılarak sistem manipüle edilebilmektedir. Sızma işlemi sonrası elde edilen gizli bilgiler kötüye kullanılabilir ve hatta kayıtlar silinip uygulamaya ya da sunucuya zarar verilebilir. Bunlara ilave olarak şirketlerin ve kurumların verilerinin çalınması ile büyük ekonomik zararlara uğramaları kaçınılmazdır. Bu çalışmada SQL enjeksiyon açığı bulunan sistemlerin tespitine dair sızma yöntemleri ve alınabilecek güvenlik önlemleri sunulmuştur. ASP.NET platformu MSSQL tabanlı SQL enjeksiyon açığı bulunan bir web projesi üzerinde saldırı örneği ve analizi gösterilmiştir. Ayrıca web tabanlı uygulamalarda alınabilecek güvenlik önlemleri ve çözüm önerileri sunulmuştur.

Anahtar Kelimeler: SQL Enjeksiyon, Saldırı Tespiti, Savunma Yöntemleri, Siber Güvenlik.

SQL Injection Cyber-Attack Scenario and Security Measures in Windows-Based Applications

Abstract

With the rapid development of technologies in recent years, there has been an increase in cyber attacks on companies. One of the most important of these attacks is SQL injection attacks. The places where the most important data of companies and institutions are kept are in databases. The type and amount of data kept in databases are increasing day by day. It is possible to infiltrate vulnerable databases through various cyber-attack methods. If the necessary precautions are not taken, it is not very difficult to infiltrate these systems, especially from the companies themselves. Many of the web applications that provide professional services prefer database query structures that use SQL infrastructure. Many web-based programs return results by using SQL infrastructure upon user request and are offered to users in different designs depending on the developer. However, in web-based applications, the system can be manipulated by injecting SQL queries with some harmful phrases entered into the system by the attacker. Confidential information obtained after the infiltration process may be misused, and even the records may be deleted, and the application or server may be damaged. In addition to these, it is inevitable for companies and institutions to suffer great economic losses by stealing their data. In this study, infiltration methods for the detection of systems with SQL injection vulnerability and security measures that can be taken are presented. An attack example and analysis are demonstrated on an ASP.NET platform MSSQL-based web project with SQL injection vulnerability. In addition, security measures and solutions that can be taken in web-based applications are presented.

Keywords: SQL Injection, Attack Detection, Defense Methods, Cybersecurity.

[*] Sorumlu Yazar: isaavci@karabuk.edu.tr

1. Giriş

Günümüzde birçok firma profesyonel ve kurumsal olarak hizmet veren servis sağlayıcı firmalardan veri tabanı sistemleri için hizmet almaktadır. Tercihlerin bu yönde şekillenmiş olması sebebiyle veri tabanı sistemleri oldukça önemli hale gelmiştir. En kritik veriler veri tabanlarında tutulurken, tutulan veri çeşidi ve veri miktarı da her geçen gün artmaktadır. Yalnızca Kurumsal Kaynak Planlaması (ERP), Müşteri İlişkileri Yönetimi (CRM) sistemleri değil artık doküman yönetimi, Ürün Yaşam Döngüsü Yönetimi (PLM) gibi uygulamalar da veri tabanlarını kullanmaktadır. Ürün geliştirme yapılan PLM sistemlerinde solidworks gibi uygulamalarda verilerini yine veri tabanlarında tutulmaktadır.

Kişilerin temel hak ve özgürlüklerini korumak için kişisel verileri firmaların yükümlülükleri ve uyacakları kurallar Kişisel Verileri Koruma Kanunu (KVKK) ile belirlenmiştir. KVKK kişisel verilerin işlenmesinden özel hayatın gizliliğine kadar kişileri koruma altına almayı ve bu kuralları göz ardı eden kurum ya da kuruluşları cezalandırmayı hedeflemektedir. Bu açıdan bakıldığından veri tabanlarındaki verilerin gizliliği ve korunması önem arz etmektedir. Kurumlar kurallara uymadığı taktirde ciddi yaptırımlarla karşılaşmaktadır.

Veri tabanlarında seçme, ekleme, silme, güncelleme gibi işlemleri yapabilmemizi sağlayan yapıya (Yapılandırılmış Sorgu Dili) SQL adı verilmiştir. SQL, Windows ve web tabanlı uygulamalarda sıklıkla kullanılan Oracle, MYSQL, PostgreSQL, MSSQL gibi veri tabanı yönetim sistemlerinin temelini oluşturmaktadır. Web tabanlı uygulamalar SQL aracılığıyla haberleşme işlemlerini gerçekleştirirler. Saldırganlar tarafından haberleşme işlemi esnasında veri tabanı ve sistem zafiyetleri ele alınarak sistem kötüye kullanılabilir. Buna ilave olarak kişisel ya da kurumsal veriler saldırganların eline geçmektedir.

SQL enjeksiyon zararlı olmayan SQL cümleciklerinin arasına zararlı kelimeler yerleştirerek sistemi manipüle etme işlemidir. Bu yöntem çok basit bir saldırı yöntemi olmasına rağmen yıllardır popülaritesini hiç yitirmemiştir. SQL enjeksiyon ile kullanıcı adı ve şifre bilgisine ihtiyaç duymadan sisteme girebilmek mümkündür. Web uygulamaları ara yüz geçişleri esnasında sorgulama dilini kullanır ve geliştiricisinin tasarımına bağlı olarak kullanıcıya farklı şekillerde sunulur. SQL enjeksiyon işlemi tam da bu esnada gerçekleştirilmektedir. Saldırgan tarafından uygulamada kullanılan kullanıcı giriş alanlarına ya da tarayıcı adres çubuğuna kötücül cümleciklerin eklenmesiyle meydana gelmektedir. Saldırgan elde ettiği bilgiler ışığında farklı senaryolar üreterek veri tabanındaki verilerin bir kısmı ya da tamamına ulaşabilir. Yönetici şifresini ele geçirerek veri tabanını silebilir, sistemi kapatabilir, ya da elde ettiği verileri kullanarak çeşitli sitelerde verileri ücret karşılığında ilgililerin kullanımına açık hale getirebilir (Boyd ve Keromytis, 2004).

SQL enjeksiyon açıklarını tespit etmek başta yapılması gereken işlem olacaktır. Sonrasında enjeksiyon açığı kullanılarak yapılan bir saldırıyı tespit edecek olan bir mekanizma ve saldırıyı kayıt altına alıp engelleyecek adımlar dizisi bu çalışmada gerçekleştirilmiştir. ASP.NET tabanlı MSSQL alt yapısını kullanan bir uygulama üzerinde web uygulamalarında saldırı çeşitlerinden biri olan SQL enjeksiyon yöntemi kullanılarak saldırı gerçekleştirilmiş ve analiz işlemi yapılmıştır. Ayrıca, SQL enjeksiyon saldırı çeşidine karşı alınabilecek güvenlik önlemleri analiz edilmiştir.

2. Literatür Taraması

Teknolojinin günden güne gelişmesi, dolayısıyla internet kullanımı ve bilginin küreselleşmesi konuları arasında doğrusal bir ilişki bulunmaktadır. İnternet kullanımına olan talep arttıkça istenilen bilgiye internet vasıtasıyla zamandan ve mekândan bağımsız olarak erişilmektedir. Türkiye İstatistik Kurumu (TÜİK) verilerine göre internet kullanımı her geçen gün yaygınlaşmaktadır. Buna etmen teknoloji çağında yaşıyor olmamız ve çevresel etkenler örnek olarak gösterilebilir. İnternet kullanım oranı bir önceki yıla kıyasla 2020 yılında %75,3' ten %79' a yükselmiştir (TÜİK, 2020). 2020 yılı verilerine göre dünya çapındaki internet kullanıcı sayısı ise 4,66 milyar olarak açıklandı (Datareportal, 2020). Dünya çapında İnternetin yaygın olarak kullanılmasıyla web tabanlı uygulamalarda bulunan verilerin gizliliğinin ve güvenliğinin sağlanması daha kritik hale gelmiştir. Kişiye özel ya da kurumsal bilgiler bu uygulamalarda barındırıldığından kaynakların güvenliği sağlanmalıdır (Vural ve Sağıroğlu, 2008).

Web ortamında sunulan hizmetler sayesinde saldırganlar sistem açıkları ile kolayca sisteme sızabilmekte ve elde ettikleri bilgileri kendi amaçlarına yönelik kullanmaktadırlar (Kara, 2020). Bu şekilde olan yasa dışı erişimler yalnızca kurum, kuruluş ve kullanıcılara değil ülke ekonomileri üzerinde de zarara sebebiyet vermektedir. Bağımsız araştırma firması Uluslararası Veri Kurumuna (IDC) açıkladığı rakamlara göre Türkiye 2017 yılında siber suçlara 233 milyar dolar harcamıştır (Türkiye'nin Siber Güvenlik Pazarı, n.d.). Rakamlar ele alındığında uygulamalardaki güvenliğin önemi bir kez daha anlaşılmaktadır (Aydoğdu ve Gündüz, 2016).

Veri tabanlarında barındırılan web uygulamalarına ait veriler diğer kısımlarla etkileşimli olarak çalışan betiklerden oluşmaktadır. Bu tarz uygulamalarda sunucu ile kullanıcı isteklerine ait gönderilen parametreler senkronize çalışmaktadır (Çağlayan, 2004). Kullanıcılara ait istekler web sunucuları üzerinde gönderildiğinden veri tabanı sunucuları saldırganların açık hedefi haline gelmektedir.

Web tabanlı uygulamalar avantajları sebebiyle birçok alanında öncü firma tarafından tercih edilir olmuştur. Her yerden erişimin kolaylıkla sağlanabilmesi, bekleme süresi dolayısıyla zaman kaybını azalttığı ve işlemlerin daha kolay süreçler ile sonuçlanmasını sağlaması gibi sebeplerden dolayı kullanımı artmıştır (Işık, 2013). Gün geçtikçe web uygulamalarının kullanımının artmasından dolayı web saldırılarında da aynı paralellikte artış yaşanmıştır.

Literatürde SQL enjeksiyon saldırı çeşidiyle alakalı birçok çalışmaya rastlanmaktadır (The Open Web Application Security Project). OWASP sitesinde SQL enjeksiyon saldırı yöntemini kullanıcının uygulamaya girdiği veriler aracılığıyla çalışacak olan sorguya yeni bir yapı eklemesinden, enjekte etmesinden oluştuğunu belirtmiştir. Buna ek olarak saldırganın veri tabanı üzerinde ekleme, silme ve güncelleme işlemlerini yapabileceğinden ve işletim sistemine komutlar verebileceğinden bahsetmiştir (OWASP, 2021).

Khaleel Ahmad ve arkadaşları araştırmalarında SQL enjeksiyonunu yetkisiz erişimlerle veri tabanından yararlanma yöntemi olarak tanımlamışlar ve SQL saldırı çeşitlerini sınıflara ayırmışlardır (Khanna ve Verma, 2018). Halfond ve arkadaşları yaptıkları çalışmada saldırıların ciddi güvenlik tehdidi oluşturduğunun altını çizmişlerdir. Araştırmacıların SQL enjeksiyon saldırıları ele almak için birçok yöntem önerdiğini fakat bu yaklaşımların problemin tamamını kapsamadığını öne sürerek farklı türlerdeki SQL saldırılarını yaptıkları bu çalışmada

detaylandırarak açıklamışlardır (Halfond et al., 2006). Buehrer ve arkadaşları araştırmalarında SQL enjeksiyon saldırısını Ayrıştırma Ağacı kullanarak nasıl engelleneceğine dair literatüre katkıda bulunmuşlardır (Buehrer ve ark., 2005).

Altıntaş veritabanı yönetim sistemlerinin tamamında enjeksiyon açıklarının oluşabileceğini söylemiştir. Ele aldığı veri tabanı çeşitlerinin kıyaslamalarını yapmış ve saldırıların tespiti, kaydedilmesi ve engellenmesi konusunda öneriler sunmuştur (Altıntaş, 2019). Saldırıların tespitinde doğuruluk analizi J48 kural tabanlı algoritma kullanılarak sağlanmıştır (Ross ve ark., 2017). Enjeksiyon saldırılarını en aza indirmek için tek bir yöntem önerilmiş olsa da tek bir çözümle saldırıların engellenemeyeceği savunmuştur (Mouli ve Jevitha, 2016). E-ticaret sitesine yapılmış olan enjeksiyon saldırılarında kullanıcı girişleri 10 farklı kategoriye ayrıldı ve zararlı karakterlerden kaçınmak alınabilecek güvenlik önlemleri arasında kabul edildi (Soewito ve ark., 2018).

3. Materyal ve Metot

3.1. SQL Enjeksiyon

SQL enjeksiyonu zararlı olmayan SQL cümleciklerinin arasına zararlı kelimeler yerleştirerek sistemi manipüle etme işlemidir. SQL enjeksiyonu veri tabanı katmanında oluşan bir güvenlik açığıdır (Bravenboer ve ark., 2010). Bu saldırı yöntemi en tehlikeli siber saldırılardan biridir (Natarajan ve Subramani, 2012). Saldırgan uygulama programcısı tarafından amaçlanandan farklı bir veri tabanı isteğiyle sonuçlanan kullanıcı girdisi sağlamaktadır. Kodunu bir web uygulamasına enjekte etmek bu konuda uzman olan kişiler tarafından kolaylıkla saldırı yapabilmektedir (Ron ve ark., 2015). Kişisel bilgilere erişilmesinde en zararlı güvenlik açıklarından biridir. Enjeksiyonları yalnızca ilişkisel veri tabanlarında oluşmaz ve Not only SQL (NoSQL) veri tabanları da enjeksiyon saldırılarına maruz kalabilmektedir (Kareem ve ark., 2021). Her bir geliştirmenin parçası olarak sızma testlerinin yaptırılması ve zafiyetlerin kapatılması gerekmektedir.

3.2. SQL Enjeksiyon Siber Saldırı Yöntemi

SQL Enjeksiyon saldırılarının saldırganlar tarafından nasıl yapıldığı detaylı olarak bu bölümde verilmektedir. SQL enjeksiyon yöntemi parola bilgisine sahip olmayan saldırgan tarafından yazılımın açığını kullanarak bu bağlantı üzerinden sisteme zarar vermeye çalışmaktadır. Kullanıcı adı ve parola girdiğimiz bir uygulama ekranı tasarlandığı zaman, bilgiler bu alana girildiğinde SQL Server' a Şekil 1'de görüldüğü gibi bir SQL sorgusu yapılmaktadır.

```
Select * from users where username='admin' and password='12345'
```

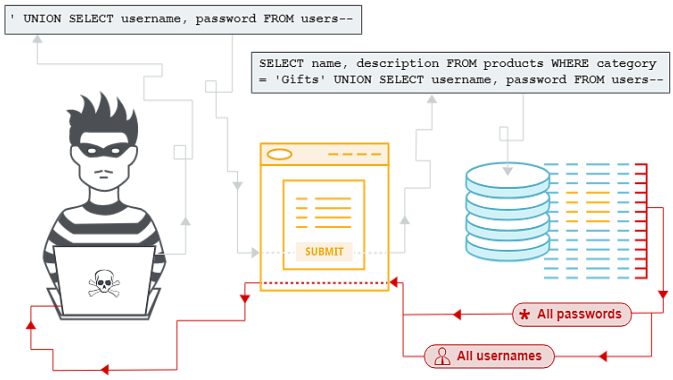
Şekil 1. Kullanıcı Girişi Esnasında Arka Planda Gönderilen Sorgu.

Şekil 1.' de gönderilen sorgu yerine Şekil 2.' deki bir sorgu gönderilmiş olsaydı sisteme basit bir şekilde sızma işlemi gerçekleştirilmiş olacaktır.

```
Select * from users where username='admin' and password='12345' or '1'='1'
```

Şekil 2. Kullanıcı Girişi Esnasında Enjeksiyon Yöntemi Ile Sistemi Manipüle Eden Sorgu.

Sistemlerde enjeksiyon açıkları sayesinde diske format atma, Dosya Transfer Protokolü' ne (FTP) backup atma ve sunucuya zararlı yazılım atma işlemlerine kadar daha birçok işlem bu açık kullanılarak gerçekleştirilmektedir.

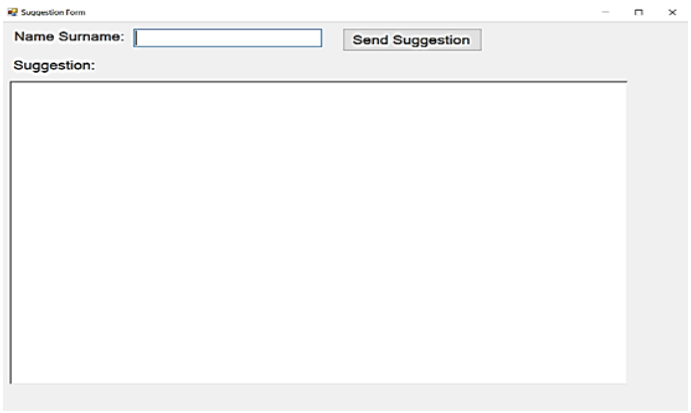


Şekil 3. Web tabanlı SQL Enjeksiyon Çerçevesi.

3.3. Windows Tabanlı Enjeksiyon Açığı Bulunan Uygulama Senaryosu

Bir şirkette üst yönetici yazılım ekibine bir talimat verdiğini ve çalışanlardan öneri toplama ile alakalı olarak hızlı bir uygulama geliştirilmesi talep edildiğini varsayalım. Hızlı olmak adına yazılım geliştirme ekibinin yaklaşık bir saatte ara yüz tasarladığını ve bu işlemi yaparken de dikkatsizlik sonucu enjeksiyon açığı oluşan bir uygulama ekranının kullanıcılara sunulduğu durum gerçekleştirilir.

Windows form uygulama ekranı Şekil 4'de görüldüğü gibi oluşturulmuştur. Uygulama kullanıcı öneride bulunduktan sonra kullanıcıya öneri ID' sini döndürmektedir.



Şekil 4. Windows Tabanlı Uygulama Ekranı.

3.4. SQL Enjeksiyon Açığı Bulunan Uygulamaya Saldırı Yöntemi

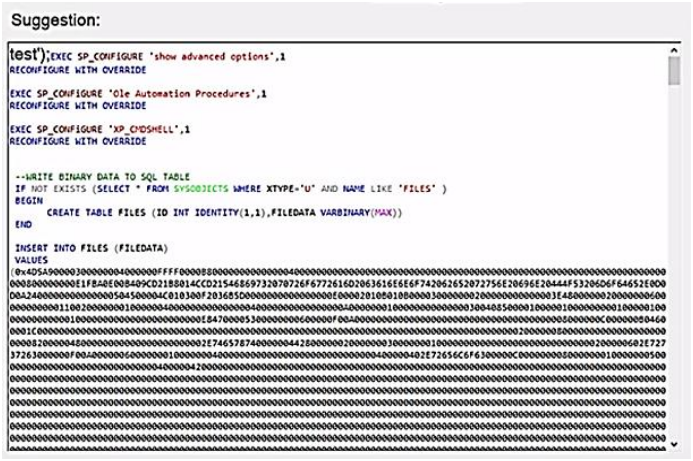
Açık bulunan uygulamaya saldırı yapıldı ve saldırı sonucunda sistem yöneticisinin (SA) parolası ele geçirilmiştir. Parola ele geçirildikten sonra veri tabanını silmekten, kayıtların değiştirilmesine kadar tüm işlemler saldırganın kontrolüne geçmiştir.

Bu saldırının sonucunda SA, SQL Server parolasını klavyeden girdiğinde ilgili bilgiyi yakalayıp ele geçirmeyi amaçlamıştır. Aşağıdaki adımlar uygulama boyunca izlenmiştir.

1. Zararlı yazılım geliştirildi.
2. Yazılımın exe'si binary olarak saklandı.
3. Enjeksiyon açığı kullanılarak bu dosya sunucuya atıldı.

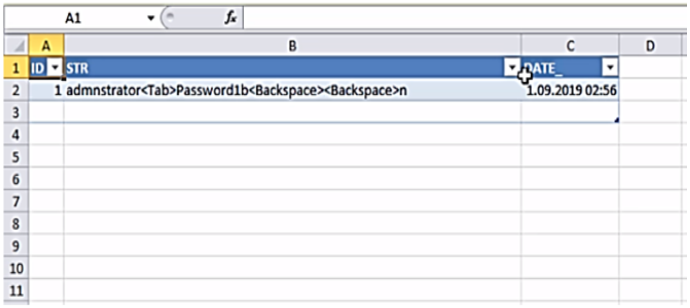
4. Sunucuya dosya kaydedildi.
5. Sunucuya kullanıcı açıldı.
6. Sunucuda keylogger uygulaması çalıştırıldı.
7. Keylogger uygulamasından gelen klavye bilgileri sql tablosuna aktarıldı.
8. Excel ile bağlanılarak klavye bilgileri izlendi ve yönetici parolası ele geçirildi.

Tüm bu işlemler SQL koduna dönüştürülmüş ve tamamı Şekil 5.' de gösterildiği gibi enjeksiyon açığı bulunan Windows tabanlı uygulamadan gönderilmektedir.



Şekil 5. Geliştirilen Kodların Enjeksiyon Yöntemi ile Sisteme Gönderildiği Windows Tabanlı Uygulama Ara Yüzü.

Bu kod sayesinde geliştirilen keylogger uygulaması sunucuya gizlenir, yeni bir kullanıcı ve tablo oluşturulur. Yöneticinin klavyesi dinlenir, girdiği parola ve kullanıcı adı tabloya insert edilir. Saldırgan excel' den bağlanıp girdiğinde Şekil 6.' da görüldüğü üzere admin parolası ele geçirilmiştir.



Şekil 6. SA Kullanıcı Adı ve Parolasını İçeren Tablonun Excel Görünümü.

4. SQL Enjeksiyon Saldırılarına Karşı Alınacak Güvenlik Önlemleri

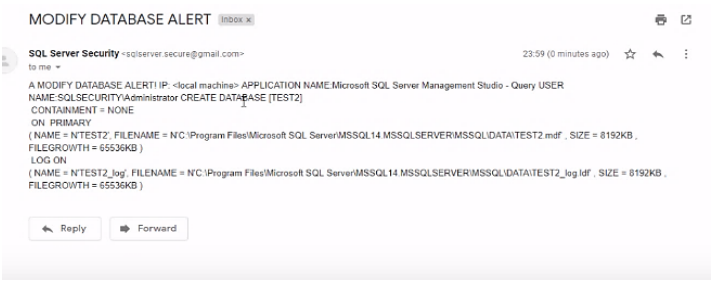
Sistem üzerinde bir şekilde açık olabilir ve kendi geliştirmediğimiz uygulamanın güvenlik açısından garantisini veremeyiz. Bu uygulamalar dışarıdan satın alınmışta olabilmektedir veya kendimiz tarafından bireysel olarak geliştirilmiş olabilmektedir. Bunun üzerine sıklıkla penetrasyon testleri yapılmalıdır ki bu geliştirilen yazılımlarda açık olup olmadığı tespit edilmesi açısından önem arz etmektedir (Chen ve ark., 2021). Fakat bu işlem sonucunda da garanti sağlanması mümkün değildir. En nihayetinde yönetici ve sistem sorumlusu bu sistemlerden sorumludur. Bu anlamda yöneticinin kendi önlemlerini alabiliyor olması gerekmektedir. Böylece, eğer herhangi bir kullanıcı sisteme SQL enjeksiyon açığını kullanarak kullanıcı açıyorsa yöneticinin bundan anında haberdar olması gerekmektedir (Alenezi ve ark., 2021).

4.1. Mail ile Veri Tabanı Yöneticisine Haber Verme

Sisteme yeni bir kullanıcı eklendiği zaman çalışan, kullanıcının hangi bilgisayardan ne zaman eklendiğini, kullanıcı adı bilgisini anlık olarak veri tabanı yöneticisine haber verme işlemini gerçekleştiren bir tetikleyici oluşturulur. Tetikleyici sisteme eklendiğinde Server Objects başlığı altında oluştuğu gözlemlenir (Kara, 2020). Tetikleyici sisteme eklendikten sonra enjeksiyon açığı kullanılarak sistemde kullanıcı açmaya çalışan saldırganın denemeleri sonrasında veri tabanı yöneticisine düşen mail Şekil 7.' da gösterilmiştir.

Şekil 7. Yöneticiye Sisteme Yeni Kullanıcı Eklendiğine Otomatik Dair Gönderilen Mail.

Aynı zamanda veri tabanında kayıtlar siliniyor, ekleniyor ya da değiştiriliyor olduğu durum içinde yeni bir alarm oluşturulur. SQL Enjeksiyon açığından faydalanarak sistemde bu işlemleri yapan saldırganın girişiminden sonra Şekil 8.'de ki mail veri tabanı yöneticisine gönderilmektedir.



Şekil 8. Veri Tabanı Kayıtlarının Değiştirildiğine Dair Yöneticiye Otomatik Gönderilen Mail.

4.2.İşlemlerin Belirli Bilgisayarlardan Yapılmasına İzin Vermek

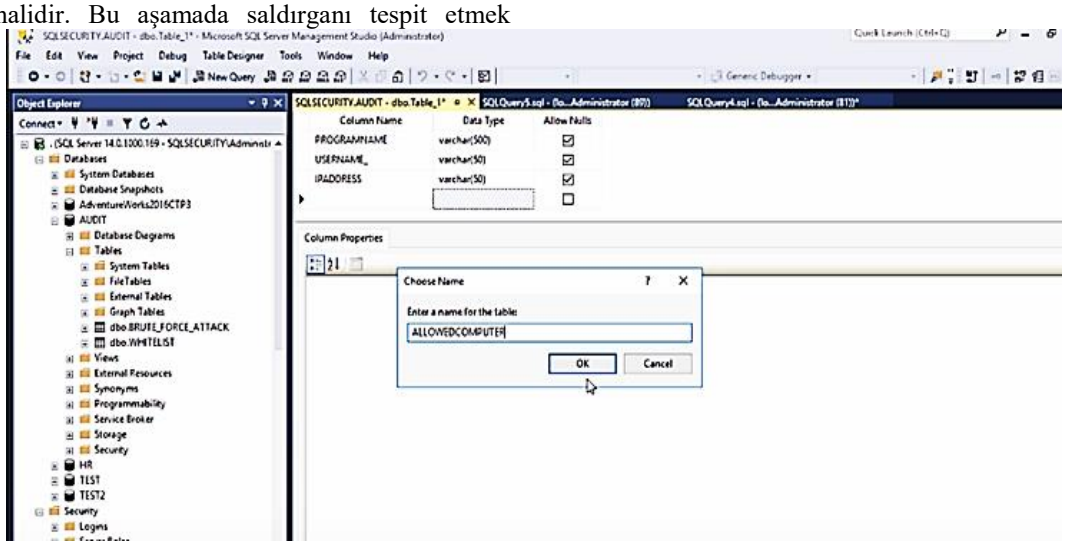
Mail ile veri tabanı admin' e haber verme işlemi saldırganı tespit etmek adına yararlı bir süreç olacaktır. Fakat saldırganın veri tabanına müdahalesi sonucu veri kaybı yaşanmış ve her şey için çok geç olabilmektedir. Bu aşamada en mantıklı işlem saldırganın müdahalesinin engellenmesi olmaktadır. Bu aşamada ne kadar kullanıcı adı ve parola biliniyor olsa bile sadece yetkili bilgisayarlardan işlem yapılmasına izin vermek doğru olacaktır. SQL'e bir AUDIT tablosu eklenir ve tabloda ID, IPADRESS alanlarına yer verilmektedir (Özarpa ve ark., 2020).

Create, drop ve alter işlemlerinde yalnızca bu tablodaki IP' lere izin verilen; diğer IP' lerden gelen istekleri engelleyen bir uyarıcı oluşturulur. Bu sayede yetkisi olmayan kullanıcıların eline kullanıcı adı ve parola bilgisi geçmiş olsa bile saldırı işlemlerinin engellenmesi sağlanır.

4.3. System Administrator Parolasının Ele Geçirildiği Şüphesi Oluştuğu Durum

Yönetici parolalarının korunması ve saklanması siber saldırılar açısından önem arz etmektedir. Yönetici parolasının ele geçirildi -ği düşünülüyorsa fakat herhangi bir kanıt yok ise bununla alakalı akla gelen ilk şey parolanın değiştirilmesi olacaktır. Bir şekilde sisteme bağlanan yazılımlar içerisinde parola bilgisi yer alıyor olabilir. Parolayı değiştirmek yazılımların kaynak kodlarını ya da bağlantı cümleciklerinde değişikliğe sebebiyet verecektir ve bu da problemli bir süreç olabilmektedir.

Bir diğer nokta ise parola ele geçirildiyse yeniden ele geçirilmesi ihtimalidir. Bu aşamada saldırganı tespit etmek saldırıyı engellemekten daha önemli bir durumdur. Bu aşamada SA kullanıcısı ile bağlanan uygulamaları ve SA kullanıcısı ile bağlanabilecek bilgisayarları kontrol edip bu bilgisayarlar ya da bu uygulamalar dışında bağlantı sağlayabile durumu söz konusuysa yöneticiye mail gönderecek bir uyarı sistemi bu problemleri ortadan kaldıracaktır.



Şekil 9. AUDIT Tablosu ve Alanları.

Şekil 9.' de ki gibi bir AUDIT tablosu oluşturuldu ve yöneticinin ilgili bilgileri bu tabloya eklenir. Bu adımlardan sonra bir uyarıcı oluşturulmaktadır ve oluşturulan bu uyarıcı sonrasında yönetici kendi makinesinden girdiğinde sisteme herhangi bir mail düşmeyecek fakat başka makineden giriş denendiği durumda Şekil 10.' da görülen bilgilendirme maili ayarlanan mail adresine SQL Server Management Studio (SSMS) tarafından gönderilecektir.

Şekil 10. Yetkisiz Giriş Tespit Edildiğine Dair Sistem Tarafından Otomatik Gönderilen Mail.

Bu işlem sonrasında SQL sunucusunda SA parolasının hangi makine tarafından ele geçirilmeye çalışıldığı ve kullanılan program bilgisi hakkında veriler toplanmaktadır. Yazılan bir uygulama ile uyarıcı belirli bilgisayarlardan ve belirli uygulamalardan sisteme SA ile giriş yapıldığında tespit etme mekanizması kurulumu gerçekleştirilir.

4.4. Tüm Sistemlerde Üzerinde SQL Enjeksiyonuna Karşı Alınması Gereken Genel Önlemler

Bu çalışmada verilen detaylı bilgiler ışığında, sonuç olarak, veritabanı yönetim sistemlerinin yüksek düzeyde güvenliğini sağlamak için; Literatürdeki çalışmalar ve örnek uygulamalar incelenmiş ve sızmaya karşı geliştirilen modellere dayalı olarak alınması gereken önlemler aşağıda verilmiştir (Avcı, 2021).

- Veritabanındaki tablo ve tablo alanlarındaki isimlerin tahmin edilmesi kolay olmamalıdır.
- Web formlarında parametrik sorguların kullanılması tercih edilmelidir.
- Web formlarında giriş kontrollerine veri girişi yapılırken bu veriler doğrulanmalı ve giriş uzunluğu kontrol edilmelidir.
- Web formlarında kullanılan ve veri tabanında bir kayıt satırını temsil eden değerler (Query-String değeri) için formlar arası geçişlerde bu değerlerin sayısal değer olup olmadığının kontrol edilmesi gerekmektedir.
- SQL tabanlı web uygulamalarında kullanıcı veri girdikten sonra veri tabanına gönderilen sorgular filtrelenmeli ve zararlı içerik olarak algılanabilecek karakterler kaldırılmalı veya değiştirilmelidir.

• SQL sunucularında oluşacak hataların, saldırganın site hakkında bilgi toplamasını ve açık araştırma yapmasını kolaylaştıracağı için web formlarında görüntülenmesi engellenmelidir.

• Uygulamalarda kullanılan veri tabanını yazma, okuma, silme gibi özellikler sadece yetkili bir yönetici tarafından gerçekleştirilmelidir.

• Web uygulamalarında sorguları formlarda yazmak yerine veritabanı tarafında saklı bir prosedür olarak gerçekleştirilmelidir.

• SQL enjeksiyon saldırı yönteminde kullanılabilecek kelimeleri (seç, ekle, sil, güncelle vb.) olası sızma girişimleri hakkında bilgi vermeyi engelleyecek bir fonksiyon ile filtreleme yapılmalıdır.

• Veritabanında kullanılacak yöneticilerin sınırlı yetkilere sahip bir kullanıcı hesabı ile çalıştırılmasına özen gösterilmelidir.

• Kullanılmayan saklı yordamlar ve yönetici hesapları kaldırılmalıdır.

• Sistem nesnelerine genel erişim verilmemeli, gerekirse kullanıcı bazında yetki verilmelidir.

• Web uygulaması ve veritabanı sunucularının bulunduğu sistem, donanım veya yazılım tabanlı güvenlik duvarı ile saldırılara karşı korunmalıdır.

• Veri girişleri yapılırken tüm SQL deyimleri bir logbook'a yazılmalı ve kontrol edilmelidir.

• Her veri girişinden sonra verilerin yedeklenmesi güvenlik açısından önemlidir.

5. Sonuç

Şirket ve kurumların en önemli verileri veri tabanlarında saklanmaktadır. Veritabanlarında tutulan verilerin türü ve veri boyutu her geçen gün artmaktadır. Savunmasız veritabanlarına çeşitli şekillerde sızmak mümkündür. Gerekli önlemler alınmadığı takdirde bu sistemlere özellikle içeriden sızmak zor değildir. Profesyonel hizmet veren web uygulamalarının birçoğu SQL altyapısını kullanan veritabanı sorgulama yapılarını tercih etmektedir. Web tabanlı birçok program, kullanıcının isteği üzerine SQL motorunu kullanarak sonuç döndürür ve geliştiriciye bağlı olarak farklı tasarımlarda kullanıcıya sunulur. Ancak web tabanlı uygulamalarda saldırgan tarafından sisteme girilen bazı zararlı ifadelerle SQL sorguları enjekte edilerek sistem manipüle edilebilmektedir. Sızma işlemi sonrasında elde edilen gizli bilgiler kötüye kullanılabilir, hatta kayıtlar silinebilir ve uygulama veya sunucu zarar görebilir. Bu zararlar diski biçimlendirmek, FTP yedeği oluşturmak, veritabanı sunucusuna kötü amaçlı yazılım atmak olarak sıralanabilir ve paralel olarak çoğaltılması mümkündür. Bu çalışmada verilen önlemler saldırı tipine göre artırılıp azaltılabilir. İlgili önlemler alınırsa SQL enjeksiyon saldırıları büyük ölçüde önlenmiş olacaktır. Tüm bu nedenlerle geliştirilen web tabanlı uygulamalarda sızma ve saldırılara karşı önlem alınmalıdır. Güvenli yazılım geliştirme hususları her aşamada dikkate alınmalıdır. Bu saldırılara karşı güvenli yerli ve milli uygulamalar geliştirilmelidir. Saldırıları önlemek için kaynak ve güvenli siber istihbarat servisleri kullanılmalı ve akademik çalışmalar desteklenmelidir.

Kaynakça

Alenezi, M., Nadeem, M., Asif, R. (2021). SQL injection attacks countermeasures assessments. Indonesian Journal of Electrical Engineering and Computer Science 2021, 21(2), 1121-1131. doi: 10.11591/ijeecs.v21.i2.

Altıntaş, B. (2019). Master Thesis a Security Comparison of Oracle, Security Comparison of Oracle, SQL Server and MYSQL Database Management System Against SQL Injection Attack Vulnerablities. Master Thesis, Yasar University, İzmir, Turkiye.

Avcı, İ. (2021). Investigation of Cyber-Attack Methods and Measures in Smart Grids. Sakarya University Journal of Science, 25 (4), 1049-1060. DOI: 10.16984/saufenbilder.955914.

Aydoğdu, D., Gündüz, M. S. (2016). Web uygulama güvenliği açıklıkları ve güvenlik çözümleri üzerine bir araştırma. 1, 1–7.

Boyd, S. W., Keromytis, A. D. (2004). SQLrand: Preventing SQL Injection Attacks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3089, 292–302, https://doi.org/10.1007/978-3-540-24852-1_21

Bravenboer, M., Dolstra, E., Visser, E. (2010). Preventing Injection Attacks with Syntax Embeddings. Science of Computer Programming, 75(7), 473–495. https://doi.org/10.1016/j.scico.2009.05.004.

Buehrer, G., Weide, B. W., Sivilotti, P. A. G. (2005). Using Parse Tree Validation to Prevent SQL Injection Attacks. SEM 2005 - Proceedings of the 5th International Workshop on Software Engineering and Middleware, 106–113.

Chen, D. et al. (2021). Sql injection attack detection and prevention techniques using deep learning. Journal of Physics: Conference Series, IOP Publishing, 012055.

Çağlayan, İ. (2004). Yeni Web Teknolojileri Ve Web Uygulamaları. Master Thesis, İstanbul Kültür Üniversitesi, İstanbul, Turkiye.

Digital2020. (2021) https://datareportal.com/reports/digital-2020-october-global-statshot (Erişim tarihi:11.08.2021)

Halfond, W. G. et al. (2006). A classification of SQL-injection attacks and countermeasures. Proceedings of the IEEE international symposium on secure software engineering, IEEE, 13-15.

Kara, İ. (2020). Web Hackleme Saldırıları. Ejovoc, vol. 10, 1-6.

Kareem, F. Q. et al. (2021). SQL injection attacks prevention system technology. Asian Journal of Research in Computer Science, 13-32. doi: 10.9734/AJRCOS/2021/v10i330242.

Khanna, S., Verma, A. K. (2018). Classification of SQL injection attacks. Advances in Intelligent Systems and Computing, 518,463–469.https://doi.org/10.1007/978-981-10-3373-5_46

Işık, D. (2013). Üniversite Kütüphanelerinde Web 2.0 Teknolojilerinin Kullanımı ve Web Tabanlı Kullanıcı Eğitimi İçin Öneriler. Türk Kütüphaneciliği, vol. 27(1): 100-116.

Mouli, V. R., Jevitha, K. P. (2016). Web Services Attacks and Security- A Systematic Literature Review. Procedia Computer Science, 93(September), 870–877. https://doi.org/10.1016/j.procs.2016.07.265

Natarajan, K., Subramani, S. (2012). Generation of Sql-Injection Free Secure Algorithm to Detect and Prevent Sql-Injection Attacks, Procedia Technology, 4, 790–796. https://doi.org/10.1016/j.protcy.2012.05.129.

Özarpa, C., Kara, S. A., Avcı, İ. (2020). Siber Güvenlik Savunma Hiyerarşisinde Yeni Bir Eğitim Modeli. 4. Uluslararası Eğitim ve Değerler Sempozyumu, ISOEVA-2020, Karabük, Türkiye, 939-947.

Ron, A., Shulman-Peleg, A., Bronshtein, E. (2015). No SQL, No Injection? Examining NoSQL Security. http://arxiv.org/abs/1506.04082.

Ross, K., Moh, M., Moh, T., & Yao, J. (2018). Multi-source data analysis and evaluation of machine learning techniques for SQL injection detection. Proceedings of the ACMSE 2018 Conference.

SQLInjection|.(2021).https://owasp.org/wwwcommunity/attacks/SQL_Injection (Erişim tarihi:01.09.2021)

Soewito, B. et al. (2018). Prevention Structured Query Language Injection Using Regular Expression and Escape String. Procedia Computer Science, 135, 678-687.https://doi.org/10.1016/j.procs.2018.08.218

TÜİK.(2020) https: //tuikweb.tuik.gov.tr/PreHaberBultenleri.do?id=33679 (Erişim tarihi:15.08.2021)

Vural, Y., Sağiroğlu, Ş. (2008). Kurumsal bilgi güvenliği ve standartları üzerine bir inceleme. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, Vol. 23, Issue 2, 507–522.