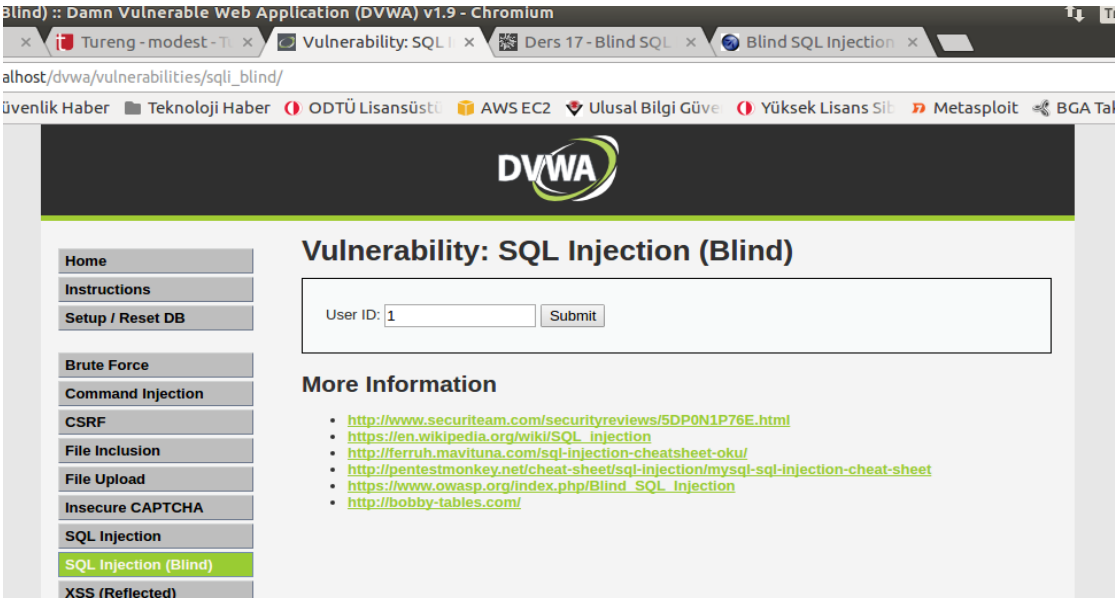
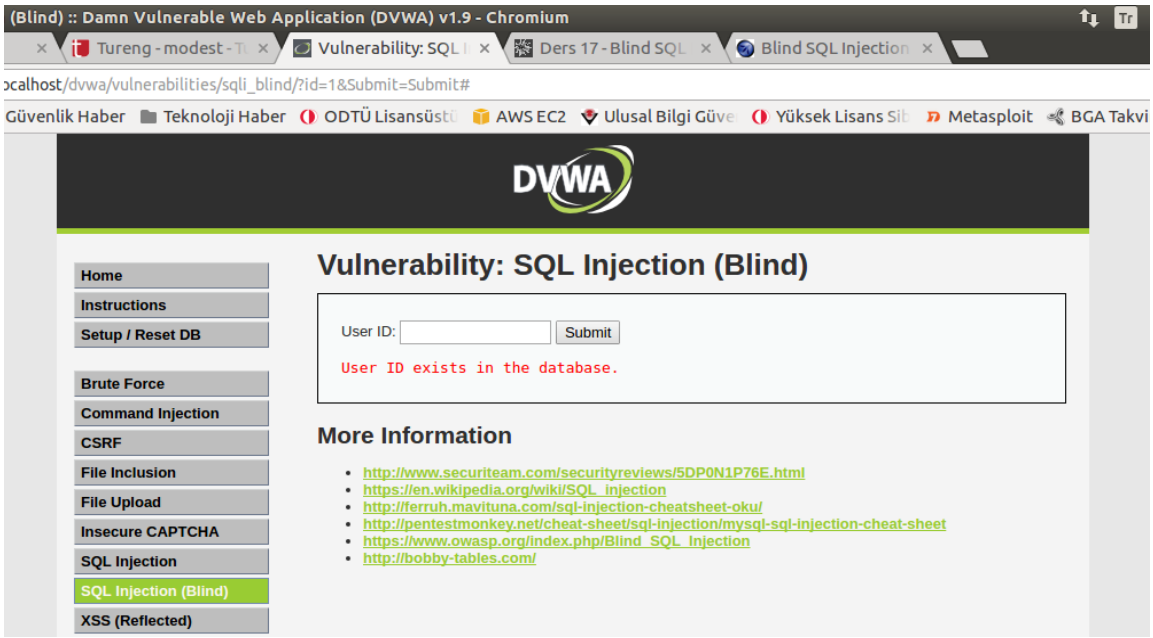
**DVWA Blind SQL Injection Hakkında EK NOT**

Blog sitendeki Blind SQL Injection (Low Level) makalesi içerisinde belirttiğin üzere Blind SQL Injection saldırılarında UNION kullanılamaz demişsin ama nedenini tam izah etmemişsin. O nedenle bu belge bunu izah etmek için hazırlanmıştır.

Hatırlarsan SQL Injection saldırılarında UNION keyword'üyle ekstradan bir sorgu ekliyorduk ve böylece while() döngüsü bizim için bir kez daha dönüyordu. Böylece eklemlediğimiz sorgudaki kritik bilgiler (veritabanı adı, tablo adı, kolon adı, kullanıcı adı - şifre bilgilerini) mevcut sql sorgusunun çıktısının altına yansıyordu. Peki Blind SQL Injection saldırısında neden UNION kullanamıyoruz? Bunun nedeni web sayfasının veritabanından gelen çıktıyı ekrana yansıtmıyor oluşundandır. Bu olayı DVWA'nın Blind SQL Injection sayfasını inceleyerek açıklayalım:




Metin kutusuna girilen veri sql sorgusuna eklenecektir.



SQL sorgusu satır döndürdüğünde ekrana yukarıdaki gibi “kayıt var” bildirimi gelecektir. Görüldüğü üzere dönen satırın verileri ekrana yansıtılmadı. Sadece veritabanında eşleşen bir satırın var olduğunu ifade eden bir bildirim yansıdı. Demek ki bu web sayfasının arkaplanında şöyle bir kodlama çalışıyormuş:

```
$sql = "Select … From tablo1 WHERE kolon1 = '1'
$result=mysqli_query($myConnection, $sql)

if ($row = mysqli_fetch_array($result) ){
        echo "User ID exists";
}
```

Yani sql sorgusu ne satır döndürürse döndürsün ekrana veritabanından gelen çıktı yansıtılmıyor, ekrana sorgudan satır dönmüşse kayıt var çıktısı yansıtılıyor. Dolayısıyla input olarak

```
1' UNION Select ….
```

girilseydi kodlama aşağıdaki gibi olacaktı:

```
$sql = "Select … From tablo1 WHERE kolon1 = '1' UNION Select ….
$result=mysqli_query($myConnection, $sql)

if ($row = mysqli_fetch_array($result) ){
        echo "User ID exists";
}
```

Yani UNION arzuladığımız satırı döndürse de ekrana basılmayacaktı. Ekrana sadece kayıt var bildirimi basılacaktı. Şayet web sayfası arkaplanında aşağıdaki kodlama çalışsaydı

```
$sql = "Select … From tablo1 WHERE kolon1 = '1' UNION Select ….
$result=mysqli_query($myConnection, $sql)

while ($row = mysqli_fetch_array($result) ){
        echo "Name : " . $row[....] . "\n";
        echo " Surname : " . $row[...] . "\n";
        echo " Age : " . $row[...] . "\n";
}
```

o zaman UNION ile ekleyeceğimiz sorgu while döngüsünü bir defa daha çalıştıracaktı ve UNION ile eklenen sorgunun döndüreceği kayıt son iterasyonda ekrana basılacaktı.

İncelediğimiz DVWA web sayfasının sunduğu mekanizmaya göre arkaplanda while değil, if çalışmakta olduğundan UNION'ı injection olarak kullanabilsek de UNION sorgusundan dönen veritabanı adları, tablo adları, kolon adları bilgilerini öğrenemeyeceğiz demektir. İşte bu noktada Blind SQL Injection yöntemi devreye giriyor. Blind SQL Injection AND operatörü ile yapılan bir saldırı türüdür.

Input:

1' AND ascii(substring((select ... from ... limit 0,1),1,1)) >= 90 #

Normalde sadece 1 değeri girildiğinde sql sorgusu satır dönüyordu ve if'e girip kayıt var deniyordu. AND ile eklenen koşul ile AND'in sağı true mu dönüyor false mu dönüyor kontrolü yaparız. Eğer AND'in sağ tarafı true dönerse satır yine dönecektir ve ekrana “kayıt var” gelecektir. Eğer AND'in sağ tarafı false dönerse (true AND false) olacağı için WHERE koşulu iptal olacaktır ve ekrana “kayıt var” bildirimi gelmeyecektir. Böylece kontrollü bir şekilde 90 sayını adım adım ilerletebiliriz ve AND'in sağ tarafı true, true, true gelirken, yani ekrana “kayıt var”, “kayıt var”, “kayıt var” bildirimleri gelirken AND'in sağı false geldiğinde deriz ki bundan bir önceki sayı doğru olanmış.

// Yani yaptığımız injection kodunun true döndürüp döndürmediğini ekrana bakarak anlarız.
// Ekran normal çıktı veriyorsa injection kodu true döndürüyor, normal çıktı vermiyorsa
// injection kodu false döndürüyor demektir.

Doğru olan sayıyı bulduktan sonra sayının ASCII tablosunda karşılık gelen harfine bakarak taradığımız veritabanı isimlerinden birinin ilk harfini bulmuş oluruz. Sonra injection kodundaki parametreleri oynayarak ikinci harfi, üçüncü harfi,... buluruz ve ilk veritabanı ismini tespit ederiz. Oradan ikinci veritabanı ismini, üçüncü veritabanı ismini,... harf harf buluruz. Daha sonra tablo isimlerini, kolon isimlerini harf harf buluruz. Ve en nihayetinde username kolonun içerdiği string'leri ve password kolonunun içerdiği string'leri harf harf buluruz. Böylece blind sql injection yöntemiyle kullanıcı adı ve şifre bilgilerine erişmiş oluruz.

**Blind SQL Injection Kodunun İzahı**

Aşağıdaki blind sql injection kodu ile taranan harfin ascii()'si alınır ve >= operatörünün kıskacında ascii()'nin döndürdüğü sayı belirlenir.

Input:

1' AND **ascii(**substring((select ... from ... limit 0,1),1,1)**)** >= 90 #

Böylece tespit edilen sayının ASCII tablosunda tekabül ettiği harfe bakılarak harf tespit edilir.

Yararlanılan Kaynak

https://www.owasp.org/index.php/Blind_SQL_Injection