**SQL INJECTION**

Muhammed Fatih YILMAZ

[mfth78@hotmail.com](mailto:mfth78@hotmail.com)

**İÇERİK**

* SQL injection zafiyeti ve giderilmesi

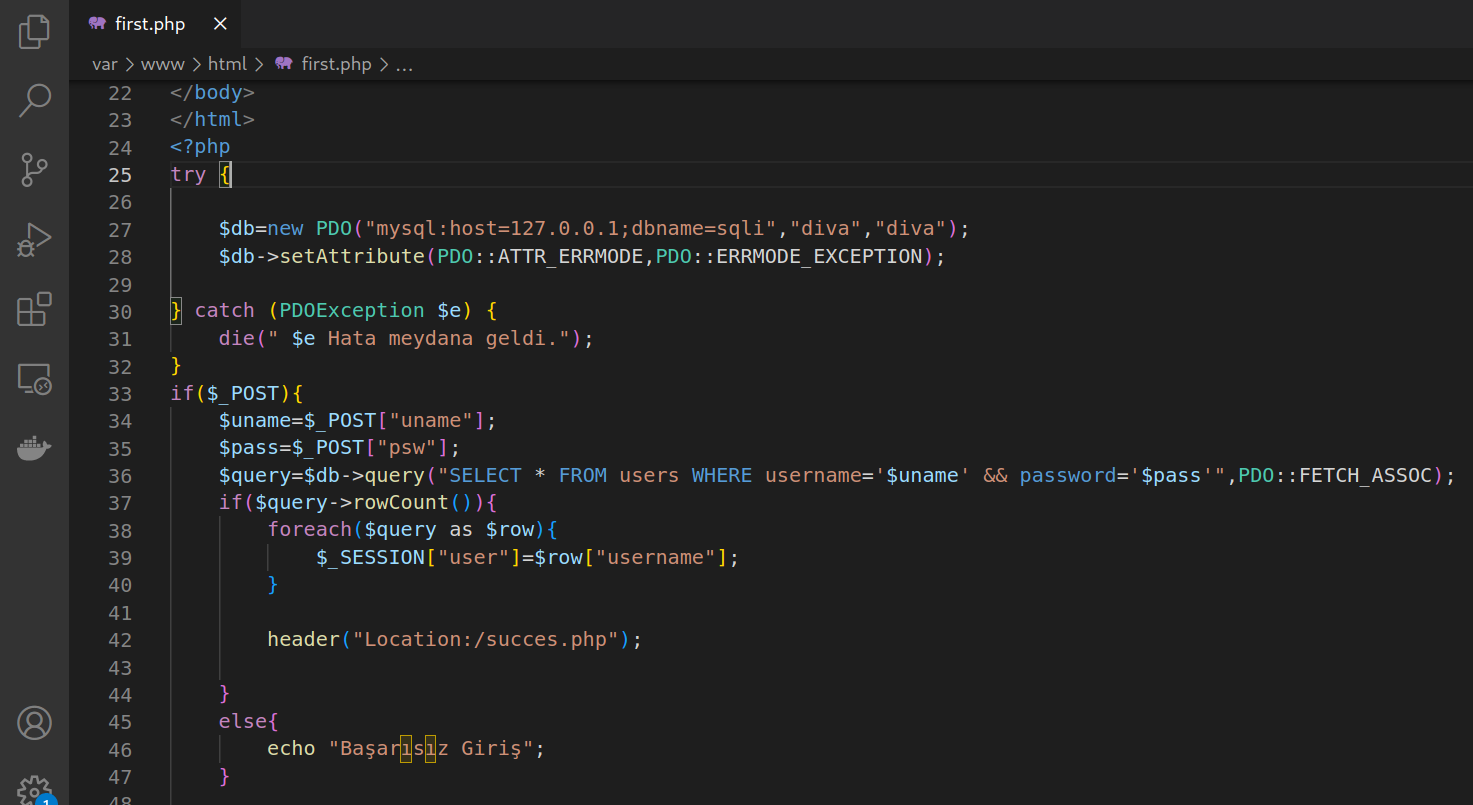
Kodumuzda gördüğünüz üzere parametreleri direkt sorguya verdiğimizden şu ifade gerçekleşir

SELECT \* FROM users WHERE username='admin' && password='topsecretpw'

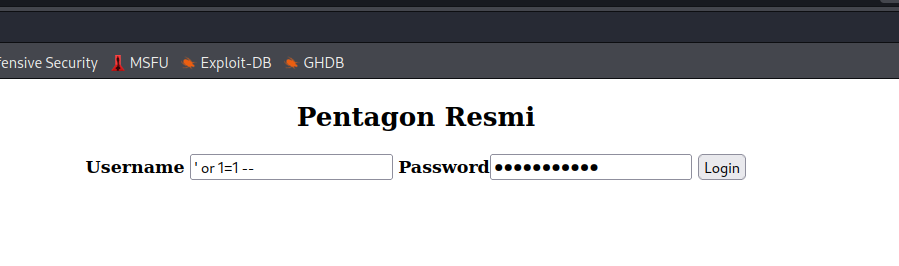
Eğer biz sorgudaki ifadelerimiz yerine ‘ or 1=1 – ifadesini yazdığımızda

SELECT \* FROM users WHERE username='’or 1=1 --' && password='’or 1=1--'

Mysql databasemizde bu ‘ (tırnak) işareti ile sql sorgusundaki ifadeden çıkıp kendi istediğimiz or 1=1 her türlü true olacağı için kendi sorgumuzu doğru kabul edilebilir hale getirdik, -- işareti ile de diğer kısımları yorum olarak mysql algıladı ve bu sayede database de istediğimiz kullanıcıya erişim sağlamış olduk.



Zafiyetli Kod Bloğu



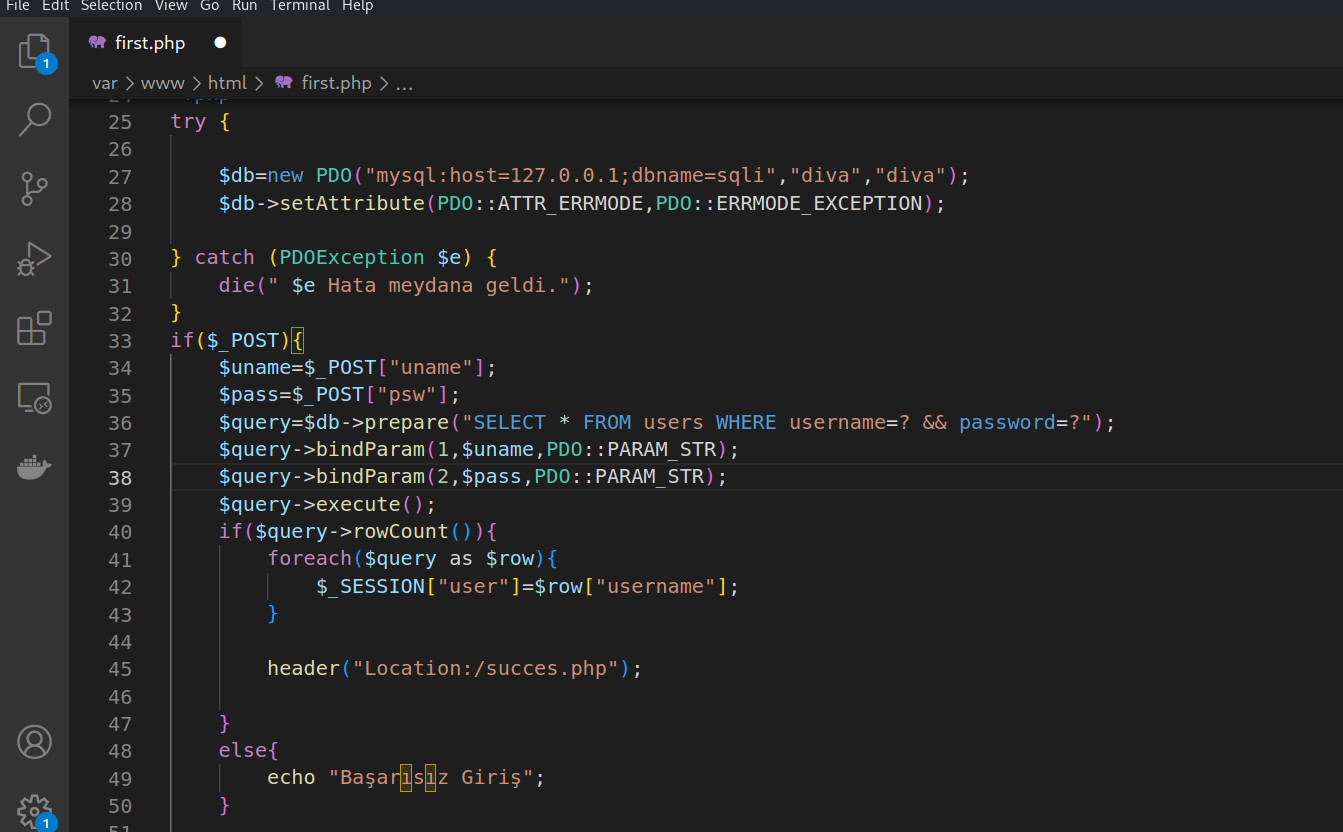
SQL İnjection Deneme

metin içeren bir resim

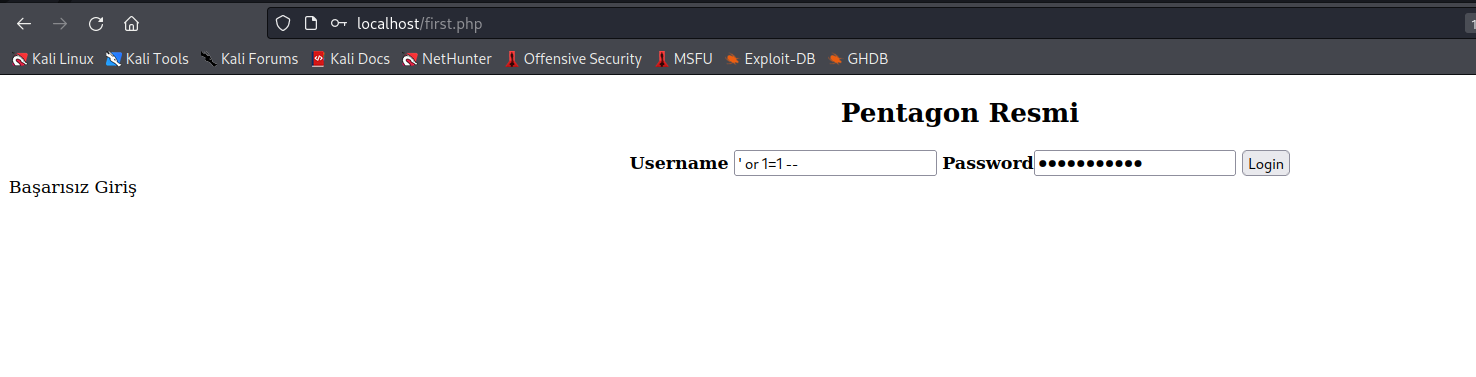
Açıklama otomatik olarak oluşturuldu

Zafiyet Gerçekleşmesi Sonucu

Bunu engellemek için PDO da aşağıdaki görselde kodda da gözüktüğü üzere prepare fonksiyonu yer almaktadır. Bu fonksiyon SQL sorgusunu oluşturur ve direkt değişkeni almak yerine ? ile değişken yerini tutup bunu databaseye gönderir. Database bu sorgumuzu ayrıştırır, ve execute etmeden saklar. Yani buna bağlı dönecek sonuçlar hazırdır. Daha sonra sorguda ? ile gösterilen parametreleri bindParam komutuyla uygulayarak sorguyu çalıştırır. bindParam da parametrelerin nereye hangi sırayla string mi integer mi olacağını belirleyerek SQL injection zaafiyetini engellemiş oluruz.



Zafiyet Engellenmiş Kod Bloğu



Zafiyet Başarısız