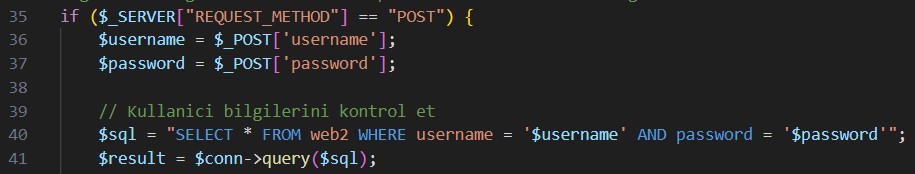
WEB UYGULAMALARINDA GÜVENLİK AÇIKLARI VE KAYNAK KOD İNCELEMELERİ

# GİRİŞ

Bu rapor, web uygulamanızda gerçekleştirilen penetrasyon testinin bulgularını ve analizini içermektedir. Test sırasında, login, register ve user.php sayfalarında çeşitli güvenlik açıkları tespit edilmiştir. Bu güvenlik açıklarının teknik detayları, tetiklenme yolları ve kaynak kodu analizleri aşağıda sunulmaktadır.

# LOGIN.PHP

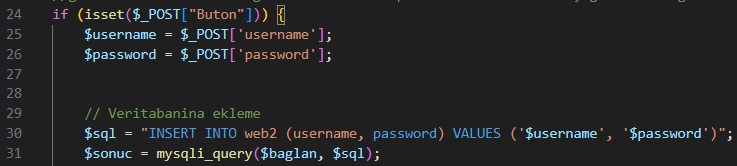
Bu web arayüzünde SQL açığı tespit edilmiştir. SQL injection yetkisiz erişim sağlayarak gizli kullanıcı bilgilerinin veri tabanından çekilmesine sebep olabilir. Bu açık, kaynak kodun şu satırlarından dolayı kaynaklanmıştır:



Buradaki SQL açığının sebebi web arayüzünden alınan username ve password kısımlarındaki girdinin direkt olarak SQL sorgusuna dahil edilmesinden kaynaklanır. Saldırgan, username veya password alanına “ admin’ OR 1=1 ” gibi zararlı SQL sorguları girerek zaafiyeti tetikletebilir ve kullanıcı girişini onaylayabilir. Bu sorgu 40.satırda işlem görür, username ve password girildikten sonra, password kısmına girilen ‘OR 1=1 sorgusu satırın sonuna eklenir. Yani ifade şuna dönüşür “username ve password doğru ise ya da 1=1 ise doğru döndür”. 1 her zaman 1’e eşit olacağından, username ve password girdilerinin ne olduğuna bakılmadan bu ifade çalışır.

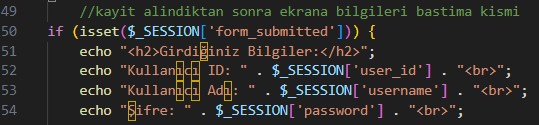
# REGİSTER.PHP

Bu web arayüzünde XSS ile SQL açıkları tespit edilmiştir. SQL açığının sebebi aşağıdaki kod satırlarından kaynaklanmıştır.



Register sayfasının kullanıcı adı ve parola kısmındaki girdiler direkt olarak SQL sorgularına dahil olduğundan dolayı saldırganların kötü niyetli SQL komutları enjekte etmesine olanak tanır ve veri tabanının yetkisiz erişim ve manipülasyonuna neden olabilir. Saldırganın password alanına girebileceği ‘OR 1=1 gibi SQL sorguları, 30.satırda, username ve password girildikten sonra aktif olur ve 1 her zaman 1’e eşit olacağından username ya da password girdilerine bakılmaksızın OR komutu çalışır. Böylece kullanıcı bilgileri doğru sayılır.

Kullanıcıdan alınan username veya password değerlerinin ekrana bastırılmasından kaynaklanan bir XSS açığı söz konusudur.

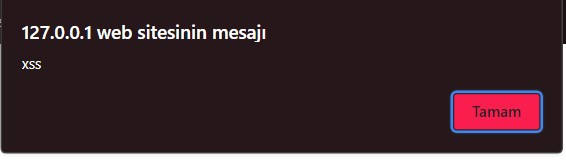


Saldırganlar username veya password kısımlarına <script>alert(‘xss’)</script> gibi Javascript komutlarını yazarak tarayıcıda çalıştırabilir ve zaafiyeti tetikletebilir. Bu şu şekilde gerçekleşir:

Kötü amaçlı kod, web uygulamasının veri tabanına veya içeriğine enjekte edilir ve sunucuda saklanır. Eğer kod anında çalıştırılmak üzere başka bir kullanıcıya gönderiliyorsa, bu da yansıtılmış (reflected) XSS olarak adlandırılır.

Burada kullanıcının girdiği username ve password değerleri, SQL sorgusuna doğrudan ekleniyor ve veri tabanına kaydediliyor. Daha sonra bu veriler, $\_SESSION değişkenlerine atanıyor ve HTML çıktısında kullanılıyor.

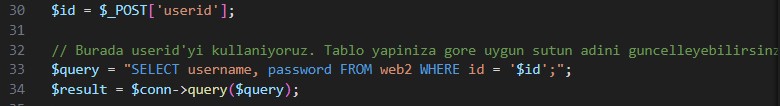
Örneği aşağıdaki gibidir.



# USER.PHP

Register sayfasında XSS tetiklemesi yapılmışsa, kullanıcı adı gibi bilgiler user.php sayfasında da XSS saldırısına neden olabilir. Zaafiyet tetiklendikten sonra ID kısmına, atanan ID bilgisi girildiğinde istek sunucuya gider ve veri tabanına kaydedilen o ID’ye ait bilgiler veya kötü niyetli yazılımlar çalıştırılarak zaafiyet sömürülür.

Bu arayüzde kaynak koddaki güvenliksiz SQL bağlantısı sebebiyle bir SQL açığı oluşmuştur. ID alanına girilen ‘ or’1’=’1 girdisi, 33.satırın sonunda çalıştırılacağı için komutun direkt olarak SQL sorgusuna dahil olmasından dolayı bütün kullanıcı bilgilerini ekrana bastıracaktır. Sorgunun etkili olmasındaki ilgili kod satırları aşağıda verilmiştir.



Bu raporda web sitelerinin açıkları ve bu zaafiyetlerin nasıl tetikleneceği konusunda bilgiler verilmiştir. Açıkların giderilmesi ve güvenli kod analizi veri güvenliği açısından önemlidir.