**🧠 1. SQL Injection Nedir?**

**SQL Injection**, saldırganın bir web uygulamasının veritabanı sorgularına **kötü niyetli SQL kodu enjekte etmesi** ile oluşan bir güvenlik açığıdır.

📌 Amacı:

* Yetkisiz veri okuma
* Giriş atlama (login bypass)
* Veri değiştirme/silme
* Veritabanı yönetimi ele geçirme

**🧪 2. SQL Injection Nasıl Çalışır?**

Güvensiz bir sorgu örneği:

SELECT \* FROM users WHERE username = '$user' AND password = '$pass';

Saldırgan, şu girişi yaparsa:

* Kullanıcı adı: admin' --
* Şifre: (boş)

Uygulama şu sorguyu çalıştırır:

SELECT \* FROM users WHERE username = 'admin' --' AND password = '';

-- SQL'de yorum satırı açar → Şifre kontrolü iptal edilir.

**🎯 3. Hangi Durumlarda SQL Injection Mümkündür?**

| **Durum** | **Açıklama** |
| --- | --- |
| 📥 Kullanıcı girdisi filtrelenmemişse | Form alanı, URL parametresi, arama kutusu vb. |
| ⚠️ SQL sorgusu doğrudan birleştiriliyorsa | Parametre bağlama yapılmamış |
| 🔍 Hatalı SQL çıktısı veriliyorsa | Sistem hataları saldırgana ipucu verir |
| 🔄 Eski/patchsiz veritabanı sistemleri | Zafiyet içeren sürümler hedef alınabilir |

**🔎 4. SQL Injection Türleri**

| **Tür** | **Açıklama** |
| --- | --- |
| 🔓 **Classic SQLi** | Giriş atlama, veri okuma (örnek: ' OR '1'='1) |
| 🕵️ **Error-based SQLi** | Hatalı sorgular üzerinden bilgi toplama |
| 📊 **Union-based SQLi** | UNION SELECT ile farklı tablo verilerini alma |
| ⌛ **Time-based Blind SQLi** | IF(condition, SLEEP(5), 0) ile veri var mı yok mu öğrenme |
| 📦 **Out-of-Band SQLi** | DNS/HTTP ile dış sunucuya veri sızdırma |

**🛠️ 5. SQL Injection Payload Örnekleri**

| **Amaç** | **Payload** |
| --- | --- |
| Giriş atlama | ' OR '1'='1 |
| Tüm tabloyu listeleme | ' UNION SELECT username, password FROM users -- |
| Hata alma | ' AND 1=CONVERT(int, (SELECT TOP 1 name FROM sysobjects)) -- |
| Zaman tabanlı kör SQLi | ' OR IF(1=1, SLEEP(5), 0) -- |

**🔐 6. SQLi'den Korunma Yöntemleri**

| **Yöntem** | **Açıklama** |
| --- | --- |
| ✅ **Prepared statements (parametrik sorgular)** | SQL komutunu sabitle, veriyi dışarıdan al |
| ✅ **Input validation** | Kullanıcı girdileri filtrelenmeli (beyaz listeleme) |
| ✅ **ORM kullanımı** | Framework seviyesinde sorgu yönetimi |
| ✅ **Veritabanı yetkilendirmesi** | Uygulama kullanıcılarına sınırlı yetki |
| ✅ **Hata mesajlarını gizleme** | SQL hata çıktıları gösterilmemeli |

**📋 7. Örnek Uygulama: DVWA (Damn Vulnerable Web App)**

Kali Linux'ta kolayca kurulabilir. SQLi testleri için idealdir.

**🔧 Kurulum:**

sudo apt install apache2 php mariadb-server

git clone https://github.com/digininja/DVWA.git /var/www/html/dvwa

**🔓 SQLi Sayfası:**

* http://localhost/dvwa/vulnerabilities/sqli/
* Kullanıcı giriş formuna ' OR '1'='1 yazıldığında giriş başarılı olur

**📷 8. Örnek Terminal/Browser Çıktısı**

metin, ekran görüntüsü, yazı tipi, grafik içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**🎯 9. Öğrenim Kazanımları**

* SQL Injection mantığını uygulamalı öğrenme
* Sorgu yapılarında zafiyetli kalıpları tanıma
* Güvenli kodlama ve input filtrelemenin önemini kavrama
* Web güvenlik testlerinin nasıl yapıldığını öğrenme