

**ANKARA ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BLM4537 - IOS ile Mobil Uygulama Geliştirme**



**Metehan ULUOCAK - 21290433**

GitHub: <https://github.com/Metehan-Uluocak/SLIP-IOS-WEB/tree/master/slip>  
Youtube: <https://youtu.be/5DC3C-EMWns>

## **Security Leak Intelligence Platform**

### **Kurumsal Güvenlik Zafiyet Takip ve Analiz Sistemi**

#### **Özet**

Bu proje, şirketlerin kullandığı yazılım/servis/altyapılarla ilişkili güvenlik sızıntılarını (leak) otomatik olarak tespit edip, normalize ederek, analiz edilebilecek bir biçimde saklayan ve güvenlik ekiplerinin erken uyarı olmasını sağlayan bir platformdur. Sistem; scraper/harvester, merkezi API (SlipAPI — .NET 8 / EF Core), veri tabanı (mysql) ve çok platformlu istemci (Flutter — slip uygulaması) bileşenlerinden oluşur.

#### **Amaç**

- Açık kaynak güvenlik duyuruları, güvenlik forumları, CVE/OSV verileri ve hedeflenen sosyal medya (ör. LinkedIn paylaşımları, güvenlik blogları) kaynaklarından otomatik veri toplamak.
- Toplanan veriyi bağlam (context), etiket, kaynak URL, yayın tarihi, etki derecesi gibi meta verilerle zenginleştirip merkezi bir depoda tutmak.
- Güvenlik ekiplerine; filtrelenebilir, aranabilir ve rol-bazlı erişimle sunulan Dashboard ve raporlar sağlamak.
- İnsan-hatalarını azaltmak için otomatik önceliklendirme (TF-IDF/isim-entity eşleştirme / basit risk scoring) ve uyarı mekanizmaları sunmak.

#### **Temel Fonksiyonlar (Detaylı)**

##### **1) Sızıntı Toplama & İşleme**

- Kaynaklar:** Önceden kurum tarafından tanımlanmış ve gelecekte de tanımlanabilecek haber siteleri, güvenlik blogları, CVE kaynakları ve sosyal medya. Ancak bu proje için güvenlik sebebiyle Mock Data kullanılmıştır.
- Normalizasyon:** Ham içerik → Leak entity (platform, başlık, özeti, tam içerik, yayın tarihi, kaynak URL, etiketler, hash, confidence score).
- Doğrulama:** Aynı içeriğin tekrarı/duplicate olması kontrol edilir (hash/URL normalization).

## 2) Veri Modeli

- **Leak:** id, platformName, title, summary, publishDate, sourceUrl, sourceName
- **Source:** id, name, url, description
- **Platform:** id, name, description
- **User:** id, name, email, role, password

## 3) API & Backend

- SlipAPI (.NET 8, EF Core): Controllers: LeaksController, SourcesController, PlatformsController, UsersController.

- Örnek endpoint'ler:

- GET /api/leaks[?platform=&from=&to=&q=]
- POST /api/leaks
- GET /api/sources
- POST /api/sources
- GET /api/platforms
- POST /api/platforms
- POST /api/users/auth (login/logout)

## 4) Frontend (Flutter – slip)

- **Dashboard:** filtrelenebilir liste, detay görünümü, hızlı arama.
- **Yönetim:** Sources, Platforms CRUD (Admin/Analyst yetkileri).
- Auth ve rol yönetimi; oturum yönetimi ve logout (AuthController).
- Mobil/desktop uyumlu responsive tasarım.

## Roller & Yetkilendirme

- **Admin:** Kullanıcı/rol yönetimi, kaynak/platform yönetimi, tüm verileri görüntüleme.
- **Analist:** Kaynak/platform ekleme/düzenleme, sizıntıları görüntüleme ve etiketleme.
- **User:** Sadece görüntüleme (read-only).

## **Proje Planı (geliştirilmiş, 8 haftalık sprintler)**

- Hafta 1:** Proje analizi ve gereksinim belirleme
- Hafta 2:** Veritabanı tasarımı ve backend altyapısının oluşturulması
- Hafta 3:** Mock datanın geliştirilmesi ve veri çekme testleri
- Hafta 4:** Dashboard ve haber listeleme arayüzünün tasarımı
- Hafta 5:** Kaynak (Sources) yönetimi modülünün geliştirilmesi
- Hafta 6:** Platform modülünün geliştirilmesi
- Hafta 7:** Rol yönetimi, yetkilendirme ve güvenlik testleri
- Hafta 8:** Test, hata düzeltme ve yayına hazırlık

## **Notlar**

- **Backend kontrolörleri:** SlipAPI/Controllers altında LeaksController.cs, SourcesController.cs, PlatformsController.cs, UsersController.cs mevcut.
- **Veri erişimi:** SlipAPI/Data/AppDbContext.cs; migrationlar SlipAPI/Migrations altında.
- **Flutter istemci:** slip/lib/controllers içinde AuthController vb. Mevcut.