Ankara Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Ağ Tabanlı Teknolojiler ve Uygulamaları Proje Dokümanı

Tuğrul Özgün

22290082

Github: https://github.com/Forcipus/Network-Danger-Scanner

Video:
https://drive.google.com/drive/folders/1vZmjpvsaXMHb73BNbsNRhw6uwMTP3VEn?usp=drive_link

***Önemli*:** Ben video uzun olduğu için parça parça çektim ancak bu parçaları birleştirip tek bir video haline getirince çözünürlük ne yazık ki azaldı. Okunulabilirlikde sorun yaşanabileceği için final videosu ile birlikte çektiğim kısa parçaları da yukarıdaki drive a ekledim. Sırası ile 1-24 kısa videolar, Final.mp4 ise bunların birleştiği tek videodur.

Web Tabanlı Tehdit Toplayıcı

Dijitalleşmenin hızla artmasıyla birlikte, kurum ve bireylerin karşılaştığı siber tehditlerin çeşitliliği ve karmaşıklığı da önemli ölçüde yükselmiştir. Günümüzde zararlı yazılımlar, oltalama (phishing) saldırıları, DDoS tehditleri ve veri ihlalleri gibi olaylar, sistem güvenliğini ciddi biçimde tehdit etmektedir. Bu tür saldırılara karşı savunma oluşturmanın temel taşlarından biri tehdit istihbaratıdır.

Tehdit istihbaratı, bir IP adresi, alan adı veya dosya hakkındaki geçmiş aktiviteleri, zararlı davranış örüntülerini ve potansiyel risk seviyelerini inceleyerek saldırıları önceden tespit etmeye yardımcı olur. Ancak bu bilgilerin farklı platformlarda dağınık halde bulunması, güvenlik uzmanları için verimli analiz yapmayı zorlaştırır.

Bu noktada geliştirilen “Web Tabanlı Tehdit Toplayıcı” projesi, bu problemi çözmeyi hedefler. Farklı açık kaynak tehdit istihbaratı servislerinden (VirusTotal, AbuseIPDB,

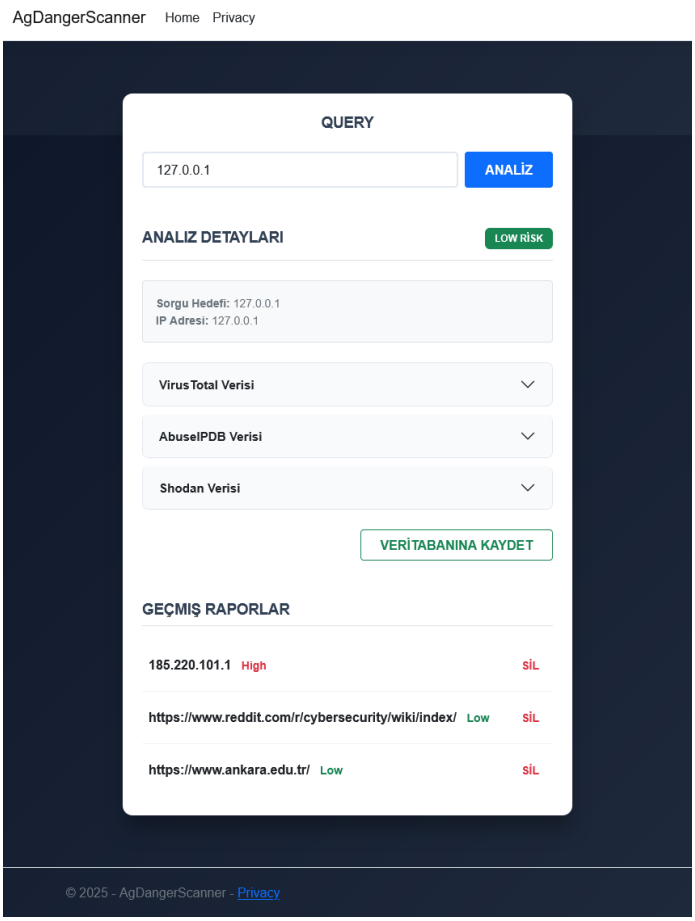
Shodan gibi) gelen veriler, tek bir web tabanlı panelde toplayarak; kullanıcı, bir IP adresi veya domain hakkında geçmiş kötüye kullanım raporlarını, açık port bilgilerini ve genel tehdit puanını tek bir ekrandan görüntüler.

Bu projenin farklı açık kaynak tehdit istihbaratı servislerinden elde edilen verileri tek bir noktada toplayarak kullanıcıların IP adresi veya alan adı (domain) bazlı güvenlik analizleri yapmasını sağlamaktır.Böylece kullanıcı, bir adresin geçmiş kötüye kullanım raporlarını, açık port bilgilerini ve genel tehdit durumunu hızlı bir şekilde değerlendirebilecektir.

Uygulama, kullanıcıların belirli bir IP adresi veya domain hakkında tehdit istihbaratı toplamasını sağlayan web tabanlı bir arayüz sunar.
Kullanıcı, web arayüzü üzerinden bir IP adresi veya domain adı girerek sorgulama yapar.
Sistem bu isteği backend'e gönderir.
Arka uç, ilgili açık kaynak tehdit istihbaratı API'lerine sorgular göndererek gelen yanıtları birleştirir.
Sonuçlar tek bir JSON yanıtı olarak frontend'e döner ve kullanıcıya okunabilir bir dashboard formatında sunulur.

Dashboard üzerinde IP'nin:

- Tehdit puanı,
- Açık port bilgileri
  gibi veriler görselleştirilir.
- Bu bilgiler bir veri tabanına kaydedilebilir.



Mimari Genel Yapı

Uygulama, üç ana bileşenden oluşur:

Frontend Katmanı

Geliştirilen projenin frontend (ön yüz) katmanı, kullanıcı ile sistem arasındaki tüm etkileşimi yöneten, verileri görselleştiren ve backend API ile haberleşen dinamik bir yapıdır. Proje dokümanında belirtildiği üzere, bu katman kullanıcı dostu, hızlı ve modüler bir deneyim sunmak amacıyla tasarlanmıştır.

Aşağıda, frontend mimarisini ve işleyişini adım adım detaylandırılmıştır:

1. Teknolojik Yapı ve Tasarım Prensipleri

Frontend, **A**SP.NET yapısı üzerinde Veritabanı işlemleri, nesne yönelimli programlama prensiplerine uygun olarak `MongoService` adlı özel bir servis katmanı üzerinden kapsüllenmiştir. Bu katman, veritabanı bağlantı dizelerini (Connection String) `appsettings.json` üzerinden güvenli bir şekilde çekerek şu işlemleri yönetir:

- **Dinamik Arayüz:** Sayfa yenilenmeden veri güncellenmesini sağlayan asenkron JavaScript (AJAX/Fetch) mantığı kullanılmıştır.
- **Modern Görünüm:** Kullanıcıya verileri tablo veya kart biçiminde sunan bir dashboard tasarımı benimsenmiştir.

2. Kullanıcı Etkileşimi ve Sorgu Süreci

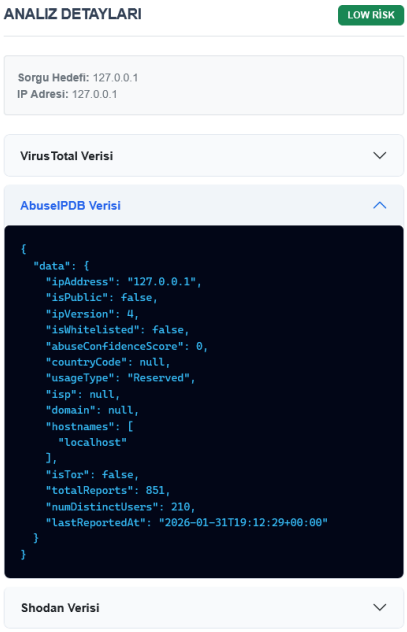
Kullanıcının sistemle girdiği etkileşim şu adımları izler:

- **Veri Girişi:** Kullanıcı, ana ekrandaki arama çubuğuna analiz etmek istediği bir IP adresi, alan adı (domain) veya URL bilgisini girer.
- **İstek Tetikleme:** "Analiz Et" butonuna basıldığında, JavaScript form verisini yakalar ve backend'deki `lookup` endpoint'ine bir HTTP GET isteği gönderir.
- **Yükleme Durumu:** Sorgu devam ederken kullanıcıya "Analiz ediliyor..." gibi bir görsel geri bildirim verilerek uygulamanın yanıt verdiği hissettirilir.

3. Veri Görselleştirme ve Dashboard Bileşenleri

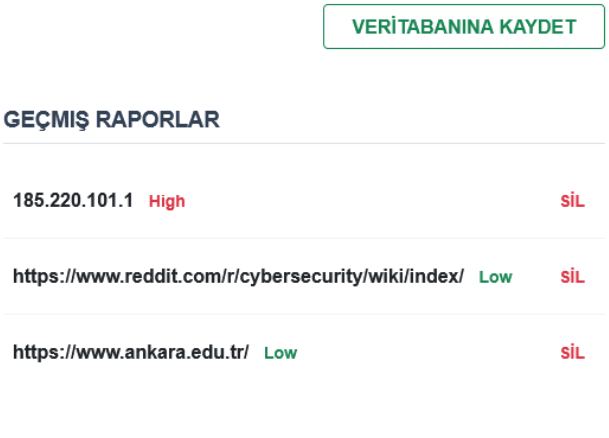
Backend'den dönen normalize edilmiş JSON yanıtı, frontend tarafında parçalanarak ilgili alanlara yerleştirilir. Dashboard üzerinde şu kritik bilgiler görselleştirilir:

- **Genel Risk Scoreu:** Hesaplanan risk skoruna göre "Low", "Medium" veya "High" şeklinde renkli risk scoreu gösterilir.
- **Collapsible (Akordiyon) Paneller:** Farklı API kaynaklarından (VirusTotal, AbuseIPDB, Shodan) gelen devasa veriler, sayfa karmaşasını önlemek adına açılır-kapanır paneller içinde sunulur.
- **Ham Veri Gösterimi:** Teknik detayları incelemek isteyen uzmanlar için API'lerden gelen ham JSON verisi, formatlanmış kod blokları içinde görüntülenir.



4. Kayıt ve Geçmiş Rapor Yönetimi

Frontend aynı zamanda verilerin kalıcılığını da kullanıcı seviyesinde yönetir:



- **Veritabanı Entegrasyonu:** Kullanıcı "Veritabanına Kaydet" butonuna bastığında, analiz özeti backend üzerinden MongoDB'ye gönderilir.
- **Geçmiş Raporlar Listesi:** Sayfanın alt kısmında yer alan bu bölüm, daha önce yapılan sorguları listeleyerek kullanıcının eski analizlerine hızlıca dönmesini sağlar.
- **Silme İşlemi:** Kullanıcı, artık ihtiyaç duymadığı raporları liste üzerinden silebilir; bu işlem eşzamanlı olarak backend API aracılığıyla veritabanından da kaldırılır.

Backend Katmanı

Uygulamanın beyni görevini görür.

1. Backend Mimari Yapısı ve Controller Mantığı

Uygulamanın arka ucu, RESTful API prensiplerine göre tasarlanmıştır. Tüm istekler `ThreatIntelController` sınıfı tarafından karşılanır:

- **İstek Yönetimi:** Frontend'den gelen IP veya alan adı (domain) sorguları JSON formatında alınır.

- **Adres Çözümleme (DNS Resolution):** Eğer kullanıcı bir domain girerse, sistem .NET'in yerleşik DNS kütüphanelerini kullanarak bu adresi bir IP adresine çözümler.

- **Normalizasyon:** Farklı servislerden gelen ham ve karmaşık veriler, sistemin iç standartlarına uygun tek bir JSON yanıtına dönüştürülür.

2. Üçüncü Taraf API Entegrasyonları (Multi-Source Intelligence)

Backend katmanı, kapsamlı bir tehdit istihbaratı (Multi-Source Intelligence) sağlamak amacıyla üç farklı küresel servis ile eşzamanlı (asenkron) olarak haberleşir. Her bir servisin kendine özgü uzmanlık alanı, projenin dinamik risk hesaplama algoritmasında farklı ağırlıklara sahiptir:

- **VirusTotal API:** 70'den fazla antivirüs tarayıcısı ve URL/alan adı engelleme servisinden gelen verileri birleştiren bir platformdur. Projemizde, bir IP veya domain'in küresel güvenlik motorları tarafından "zararlı" (malicious) olarak işaretlenip işaretlenmediğini anlamak için kullanılır.

- **AbuseIPDB API:** Tamamen IP tabanlı kötüye kullanım raporlarına odaklanan bir veritabanıdır. Özellikle spam, DDoS, hacking denemeleri ve kaba kuvvet (brute-force) saldırıları yapan IP adreslerinin geçmiş kayıtlarını tutar.

- **Shodan API:** İnternete bağlı cihazların "arama motoru" olarak bilinir. Bir sunucunun hangi portlarının dış dünyaya açık olduğunu, hangi işletim sistemini ve servis banner'larını (sürüm bilgileri vb.) kullandığını tespit eder.

Her servis için güvenli API anahtarı (API Key) yönetimi yapılır ve sonuçlar `IHttpClientFactory` üzerinden yönetilen HTTP istemcileriyle çekilir.

3. Veri İşleme ve Risk Skorlama Algoritması

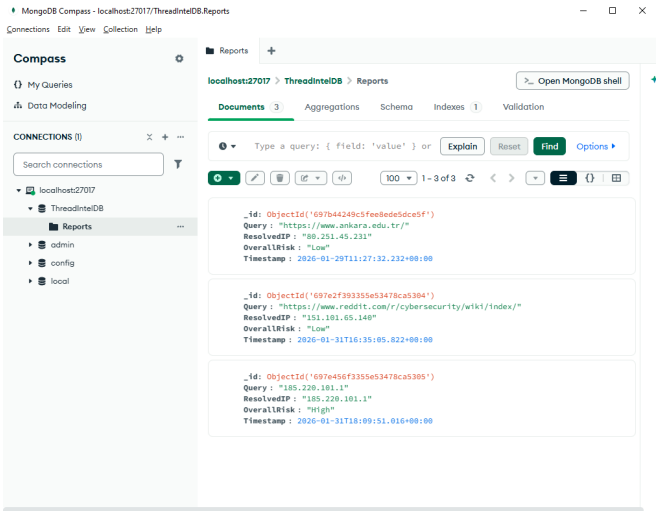
Sistem, her üç kaynaktan gelen verileri normalize ederek bir **Overall Risk** (Genel Risk) etiketi üretir. Bu hesaplamada kullanılan kritik özellikler şunlardır:

- **AbuseIPDB - Abuse Confidence Score:** Bu servisten gelen "güven skoru" (0-100 arası), risk puanlamasının temelini oluşturur. Eğer bir IP adresinin geçmişteki kötüye kullanım raporları yoğunsa ve skor 50'nin üzerindeyse, sistem doğrudan **"High Risk"** ataması yapar.

- **VirusTotal - Malicious Engine Count:** API yanıtındaki `last_analysis_stats` alanı incelenerek, kaç farklı güvenlik motorunun bu hedefi "zararlı" olarak işaretlediği kontrol edilir. Motor sayısının artışı (örneğin 3 ve üzeri tespit), AbuseIPDB skoru düşük olsa bile risk seviyesini yukarı çeker.

- **Shodan - Open Ports and Services:** IP üzerinde tespit edilen açık port sayısı ve bu portlarda çalışan servislerin (VNC, RDP, Telnet vb.) kritikliği değerlendirilir. Shodan verilerinde açık port ve zafiyet barındırabilecek servislerin varlığı, sistemin risk etiketini en az **"Medium Risk"** seviyesine taşıyan önemli bir değişkendir.

4. Veri Tabanı Katmanı ve MongoDB Entegrasyonu



Geçmiş sorguların kalıcılığını sağlamak amacıyla asenkron bir kayıt sistemi kurulmuştur. Proje kapsamında toplanan istihbarat verileri, dış API servislerinden dinamik ve hiyerarşik JSON formatlarında gelmektedir. Bu verilerin geleneksel ilişkisel veritabanlarının (SQL) katı tablo yapısına uydurulması veri kaybına veya karmaşık dönüşüm süreçlerine yol açabilmektedir. Veritabanı işlemleri, nesne yönelimli programlama prensiplerine uygun olarak `MongoService` adlı özel bir servis katmanı üzerinden kapsüllenmiştir. Bu katman, veritabanı bağlantı dizelerini (Connection String) `appsettings.json` üzerinden güvenli bir şekilde çekerek şu işlemleri yönetir:

- **Veri Yazma (Create):** Kullanıcı "Veritabanına Kaydet" butonuna bastığında, analiz sonucu `InsertOneAsync` metoduyla MongoDB koleksiyonuna eklenir.

- **Veri Listeleme (Read):** Kullanıcı geçmiş raporları görüntülemek istediğinde, veritabanındaki tüm kayıtlar tarihe göre azalan sırada (`SortByDescending`) listelenerek dashboard'a yansıtılır.

- **Veri Silme (Delete):** Kullanıcının artık ihtiyaç duymadığı analizler, benzersiz döküman kimliği (ObjectId) üzerinden filtrelenerek sistemden kalıcı olarak kaldırılır.

Web Tabanlı Tehdit İstihbarat Toplayıcı projesi, siber güvenlik alanında artan tehditleri analiz etmek ve güvenlik farkındalığını artırmak amacıyla geliştirilmiştir. Proje, tamamen

ücretsiz ve açık kaynak API’leri kullanarak veri topladığı için herhangi bir maliyet gerektirmez.

Sonuç

Geliştirilen bu proje, siber güvenlik dünyasındaki veri dağınıklığını ortadan kaldıran ve analiz süreçlerini tek bir merkezde toplayan etkili bir çözüm sunmaktadır. ASP.NET Core mimarisi üzerinde inşa edilen sistemin en büyük başarısı, farklı API servislerinden gelen heterojen verileri normalize ederek anlamlı bir risk skoruna dönüştürebilmesidir. VirusTotal, AbuseIPDB ve Shodan gibi kritik kaynakların eş zamanlı entegrasyonu sayesinde, bir IP veya alan adının tehdit seviyesi saniyeler içinde tespit edilip kalıcı olarak MongoDB veri tabanına işlenebilmektedir. Bu yapı, güvenlik uzmanlarına sadece ham veri sunmakla kalmayıp, bu verileri yorumlayarak hızlı karar verme mekanizması sağlayan modüler bir altyapı oluşturmuştur.

Sonuç olarak, Web Tabanlı Tehdit İstihbarat Toplayıcı projesi hem maliyetsiz açık kaynak servislerin verimli kullanımını hem de veri yönetim prensiplerini başarıyla bir araya getirmiştir. Sistemin esnek şema yapısı, gelecekte yeni istihbarat kaynaklarının veya yapay zeka tabanlı risk tahmin modüllerinin eklenmesine olanak tanıyan genişletilebilir bir temel sunmaktadır. Proje, özellikle kısıtlı bütçeye sahip kurumlar ve bireysel kullanıcılar için siber güvenlik farkındalığını artırmayı hedefleyen, pratik uygulama değeri yüksek bir çalışma olarak tamamlanmıştır.