**🧠 1. Corscanner Nedir?**

**Corscanner**, web uygulamalarında **Cross-Origin Resource Sharing (CORS)** yapılandırmalarını analiz ederek, kötü amaçlı domainlere izin verilmiş mi, Access-Control-Allow-Origin hataları var mı gibi kritik açıklara odaklanan bir güvenlik test aracıdır.

📌 CORS zafiyetleri, saldırganın başka bir domain üzerinden kullanıcıya ait verileri çekebilmesine yol açabilir.  
📌 Bu araç Python ile yazılmıştır ve hızlı tarama yapabilir.

**🎯 2. Ne İşe Yarar?**

| **Özellik** | **Açıklama** |
| --- | --- |
| 🌐 CORS misconfiguration tespiti | Güvensiz origin’lere izin verilip verilmediğini kontrol eder |
| 🧪 Access-Control-Allow-Origin analizi | Yanıt başlıklarını inceler |
| ⚠️ Kötüye kullanılabilir yapıların teşhisi | \*, null, subdomain.example.com gibi tehlikeli değerleri yakalar |
| 📄 URL listesiyle çalışır | Toplu test yapılabilir |
| 🧠 Red Team / Blue Team için ideal | Hem saldırgan hem de savunmacı bakışla kullanılabilir |

**⚙️ 3. Kurulum (Kali / Linux)**

**✅ 1. GitHub’dan indir:**

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

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**✅ 2. Gerekenleri yükle:**

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

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

Python 3 ile çalışır.

**🚀 4. Kullanım Komutları**

**✅ Tek bir hedef URL taraması:**

python3 corscanner.py -u https://example.com

**✅ Çoklu hedef taraması (dosya ile):**

python3 corscanner.py -i urls.txt

→ urls.txt dosyasına her satıra bir URL gelecek şekilde yazılır.

**✅ Sessiz mod (yalnızca sonuç):**

python3 corscanner.py -u https://example.com -q

**📋 5. Örnek Terminal Çıktısı**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**🧠 6. Neden CORS Güvenliği Önemlidir?**

* Access-Control-Allow-Origin: \* → Herhangi bir siteden veri çekmeye izin verir
* Access-Control-Allow-Credentials: true ile birleştiğinde **kullanıcıya ait oturum bilgileri sızabilir**
* Saldırgan, kendi sitesinden API çağrısı yaparak kurbanın oturumunu kötüye kullanabilir

**📊 7. Güvensiz CORS Yapılandırmalarına Örnekler**

| **Yapılandırma** | **Risk** |
| --- | --- |
| \* | Herkese açık erişim |
| null | Dosya:// gibi kaynaklara izin |
| evil.example.com | Tehlikeli subdomain whitelisti |
| example.com.evil.com | Yanıltıcı domain izni |
| CORS + credentials: true | Oturum çalma riski |

**🧪 8. Kullanım Senaryoları**

| **Senaryo** | **Açıklama** |
| --- | --- |
| 🔍 Web API testleri | REST/GraphQL API’lerin CORS açıklarını tarama |
| 🧠 Red Team analizleri | Kurbanın oturum bilgilerini çalmak için kullanılabilecek açıklıklar |
| 🛡️ Blue Team yapılandırma kontrolü | Yanlışlıkla izin verilen domainleri bulma |
| 🧾 Güvenlik raporu oluşturma | CORS konfigürasyon hatalarını belgelendirme |

**🎯 9. Öğrenim Kazanımları**

* CORS mantığını ve güvenli yapılandırmaları öğrenme
* CORS açıklıklarının nasıl tespit edildiğini uygulamalı görme
* Güvenli web API tasarımının temellerini kavrama
* Python tabanlı zafiyet tarayıcılarını kullanma deneyimi
* Red Team ve Blue Team senaryolarında analiz yetkinliği kazanma