FUTURE WİTH SERDAR · BÖLÜM 63 EKİ

REA ile Tersine Mühendislik Pratik Kılavuz

Kaynak kodu elinde olmayan bir uygulamanın nasıl çalıştığını anlamak · Ekim 2026

Fikir şu: uygulamanın kaynak kodu sende yok, ama davranışı ortada. REA, ajanına “bu ikili dosyayı / bu uygulamayı / bu siteyi incele” diyebilme yeteneği veriyor. Sen soruyu soruyorsun, ajan REA'nın araçlarıyla hedefi açıyor, bulduğunu **kanıtıyla birlikte** geri getiriyor.

> **Önce şunu netleştirelim:** REA tek başına çalışan bir uygulama değil. Ajanına bağlanan bir **MCP sunucusu**. Yani Claude Code (ya da Codex, Cursor, Gemini CLI, Grok Build) kurulu değilse REA'nın yapacağı bir şey yok. Asıl işi senin ajanın yapıyor; REA ona gözlük takıyor.

Ön koşullar

- Desteklenen bir kodlama ajanı: **Claude Code**, Codex, Cursor, Gemini CLI, Grok Build
- **Node.js** 22.x (≥22.19), 24.x (≥24.11) veya 26+ ve **npm**
- Hedefe göre ek araç: yerel ikili dosyalar için **Hopper, Ghidra veya IDA**; web siteleri için Chrome ailesinden bir tarayıcı; JavaScript/Electron ve .NET için ek bir şey gerekmiyor
- İncelemeye **hakkın olan** bir hedef (buna aşağıda ayrıca değiniyorum)

Kurulum: tek komut

```
npx rea-agents setup
```

Komut sana hangi ajanları kuracağını soruyor, yapacağı değişiklikleri önüne koyuyor ve onayını bekliyor. Onayladığında REA'nın MCP sunucusunu ve ona uygun çalışma talimatlarını ekliyor; mevcut yapılandırmanın **yedeğini alıyor**. Bittiğinde **ajanını yeniden başlat**, yoksa yeni araçları görmez.

> **Kurmadan önce adrese bak.** Aynı isimle sahte depolar türedi. Aracın kendi sayfası bile buna dikkat çekiyor: benzer bir depo adı ya da kopyalanmış bir README, o indirmenin resmi olduğunu göstermez. Resmi üç adres: `github.com/morluto/rea`, `rea.tools` ve npm'deki `rea-agents` paketi.

Terminalden kullanmak (isteğe bağlı)

Ajan üzerinden gitmek istemiyorsan aynı işler komut satırından da çalışıyor:

```
npx -y rea-agents@latest analyze-javascript-application /dosya/yolu --json
```

Sürekli kullanacaksan global kurup `rea` komutunu açabilirsin:

```
npm install --global rea-agents
rea --help
```

Neleri açabiliyor

Liste uzun; en çok işine yarayacak olanlar bunlar:

Hedef	Ne döndürüyor	Gereken
Yerel ikili dosyalar	Sözde kod, assembly, metinler, semboller, çağrılar	Hopper, Ghidra ya da IDA
JavaScript / Electron	Modüller, import'lar, kaynak haritaları, rotalar, IPC	Node.js + npm
.NET derlemeleri	Üstveri, CIL komutları, yerel bağımlılıklar	Ek araç yok
Web siteleri	Sayfa yapısı, betikler, ağ gözlemleri, ekran görüntüleri	Chrome ailesi tarayıcı
Android APK	Manifest, sınıflar, derlenmiş metotlar, referanslar	JADX + JDK
SQLite veritabanı	Tablolar, sütunlar, indeksler, görünümler, satırlar	Yerel kopya

İki önemli ayrım. Statik JavaScript ve .NET incelemesi dosyaları **uygulamayı çalıştırmadan** okuyor. Çalışma zamanı yakalaması ise hedefi **senin kullanıcı izinlerinle** çalıştırıyor ya da onunla etkileşime giriyor. Bunu bilerek seç. Her iki durumda da analiz **senin bilgisayarında** yapılıyor; dosyaların dışarı çıkmıyor.

Nasıl sorulur: işe yarayan kalıplar

Aracın resmi örneği şu biçimde ve bu kalıp gerçekten iyi çalışıyor: **anla → kanıtını göster → bana uyarla**.

RESMİ ÖRNEK

“Notes uygulamasında arama nasıl çalışıyor, anla; kanıtını göster; sonra kendi projem için benzer bir özellik yaz.”

1 · DAR SORU

“Bu uygulamada *şu* özellik hangi adımlarla çalışıyor? Sadece o akışı izle, tamamını analiz etme.”

2 · KANIT İSTE

“Her çıkarımın yanında hangi dosyaya/fonksiyona baktığını yaz. Emin olmadığın yerleri ayrıca işaretle.”

3 · SINIRI SOR

“Bu analizde neyi göremedin? Hangi kısım tahmin, hangi kısım doğrudan gözlem?”

4 · KARŞILAŞTIR

“Bu iki sürüm arasında *şu* davranış değişmiş. Farkı ve nerede değiştiğini çıkar.”

5 · KENDİ PROJENE TAŞI

“Anladığın mantığı benim projemde sıfırdan yaz. Onların kodunu kopyalama; davranışı tarif et ve testini de yaz.”

6 · DOĞRULAT

“Yazdığın uygulamayı çalıştır ve orijinaliyle aynı sonucu verdiğini göster.”

Sonucu nasıl okursun

REA'nın en değerli yanı burada: her sonucun yanında **kanıtını ve sınırını** veriyor. “Şöyle çalışıyor” deyip geçmiyor; neye bakarak böyle dediğini ve nerede emin olmadığını da yazıyor.

Pratikte şu üçünü ayır

- **Gözlem**: dosyada/çıktıda birebir görünen şey. Buna güvenebilirsin.
- **Çıkarım**: gözlemlerden kurulan açıklama. Mantıklı ama test etmeden kabul etme.
- **Bilinmeyen**: aracın göremediğini söylediği yer. Kararını buna dayandırma.

Bir aracın sana “bilmiyorum” diyebilmesi, her soruya kendinden emin cevap vermesinden kıymetlidir. Bu kılavuzdaki tek numara varsa o da budur: **önce bilinmeyenler listesini oku.**

Sık yapılan hatalar

“Her şeyi analiz et” demek

Dar soru net sonuç verir, geniş soru uzun ve işe yaramaz bir rapor verir. Tek bir özellikle başla.

Ajanı yeniden başlatmamak

Kurulum bittiğinde ajan hala eski araç listesiyle çalışıyor olur. “REA çalışmıyor” vakalarının çoğu bu.

Yerel ikili dosyaya sağlayıcısız girmek

Native analiz için Hopper, Ghidra ya da IDA'dan birini önceden kurup tanıtman gerekiyor. Yoksa o kategori boş döner.

Çıkan kodu olduğu gibi projene yapıştırmak

Hem lisans açısından risklidir hem de bağlamı olmayan kod projende çalışmaz. Davranışı anla, kendi kodunu yaz.

Sınır: bunu atlama

> **Öğrenmek için sökmek başka, bir ürünün özelliğini kopyalayıp satmak başka.** Aracın kendi beyanı da bunu söylüyor: REA; yasal araştırma, eğitim, birlikte çalışabilirlik, yazılım koruma ve **sahip olduğun ya da incelemeye yetkili olduğun** yazılımın analizi için. Başkasının ürününü açıp özelliğini kendi ürünün diye satmak bu çerçevenin dışında kalır ve o çizgiyi aştığında seni araç değil hukuk karşılar.

Kontrol listesi

- ☐ Ajanım kurulu ve çalışıyor (Claude Code / Codex / Cursor / Gemini CLI / Grok Build)
- ☐ Node.js sürümüm uyumlu (22.19+ / 24.11+ / 26+)
- ☐ `npx rea-agents setup` çalıştırdım, değişiklikleri okuyup onayladım
- ☐ Deponun adresini resmi üç kaynaktan doğruladım
- ☐ Ajanı yeniden başlattım
- ☐ Hedefim için gereken ek aracı kurdum (gerekiyorsa)
- ☐ Tek bir dar soruyla başladım, “her şeyi analiz et” demedim
- ☐ Çıktıdaki **bilinmeyenler** bölümünü okudum
- ☐ İncelediğim yazılımı incelemeye hakkım var