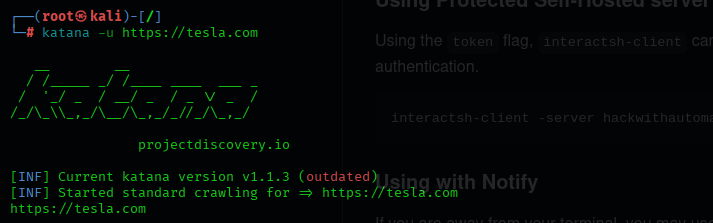
**Katana: Web'in Derinliklerindeki Gizli Yolları Kesen Samuray Kılıcı**

****

Bir web sitesini ziyaret ettiğimizde gördüğümüz, genellikle buzdağının sadece görünen kısmıdır. Ana sayfa, hakkımızda sayfası, iletişim formu... Peki ya geliştiricilerin unuttuğu test sayfaları, gizli API uç noktaları (endpoints) veya hassas bilgiler içeren JavaScript dosyaları? İşte bu gizli dijital patikaları keşfetmek, siber güvenlikte "saldırı yüzeyini" genişletmenin ve zafiyet aramanın ilk adımıdır. Bu keşif görevini modern, hızlı ve etkili bir şekilde yapmak için tasarlanmış bir araç var: **Katana**. Tıpkı bir samurayın keskin kılıcı gibi, Katana da bir web sitesinin katmanlarını keserek en derin ve gizli noktalarına ulaşmamızı sağlar. Bu yazıda, Katana'nın ne olduğunu, neden bu kadar önemli olduğunu, nasıl kurulup kullanılacağını ve onu bir siber güvenlik ustası gibi kullanmanızı sağlayacak pratik örnekleri inceleyeceğiz.

**1. Katana Aracı Nedir?**

**Katana**, Project Discovery tarafından geliştirilen, yeni nesil bir web tarama (crawling) ve örümcek (spidering) aracıdır. Temel amacı, bir web sitesini veya bir web uygulamaları listesini otomatik olarak gezerek ulaşılabilir tüm bağlantıları, JavaScript dosyalarını, API uç noktalarını ve diğer kaynakları keşfetmektir.

Go dili ile yazılmış olması sayesinde son derece hızlıdır ve iki temel modda çalışabilir:

* **Standart Mod:** Ham HTTP istekleri göndererek bir web sitesinin kaynak kodunu analiz eder ve buradaki linkleri takip eder.
* **Headless Mod:** Arka planda tam bir web tarayıcısı (Chrome) çalıştırarak siteyi ziyaret eder. Bu mod, içeriği JavaScript ile dinamik olarak yüklenen modern web uygulamalarını (React, Angular, Vue vb.) taramak için hayati öneme sahiptir.

Özetle Katana, bir hedefin haritasını çıkarmak için kullanılan güçlü bir keşif aracıdır.

**2. Neden Kullanılır?**

Web siteleri artık basit HTML sayfalarından ibaret değil. Karmaşık yapılar, JavaScript kütüphaneleri ve API'lerle dolu. Katana'nın kullanılmasının temel nedenleri şunlardır:

* **Saldırı Yüzeyi Keşfi:** Bir pentester veya bug bounty avcısı için ne kadar çok uç nokta, parametre ve dosya bulunursa, zafiyet bulma olasılığı o kadar artar. Katana, /admin, /backup, /api/v2/users gibi manuel olarak bulunması zor olan yolları otomatik olarak ortaya çıkarır.
* **Hassas Bilgi Tespiti:** JavaScript dosyaları içerisinde bazen API anahtarları, gizli uç noktalar veya geliştirici yorumları gibi hassas bilgiler unutulabilir. Katana, bu dosyaları toplayarak daha sonraki analizler için bir araya getirir.
* **Otomasyon ve Entegrasyon:** Tıpkı Naabu gibi, Katana da diğer araçlarla "pipe" yöntemiyle kusursuz bir şekilde çalışır. Subfinder ile bulunan alan adlarını doğrudan Katana'ya göndererek tüm bu siteleri otomatik olarak tarayabilirsiniz.
* **Modern Web Uygulamalarıyla Uyumluluk:** headless modu sayesinde, geleneksel tarayıcıların içeriğini göremediği, JavaScript'e bağımlı modern web sitelerinin haritasını çıkarabilir.

**3. Nasıl Kurulum Yapılır?**

Katana'yı kurmak, diğer Project Discovery araçları gibi oldukça basittir.

**Yöntem 1: Go ile Kurulum (Önerilen)**

Sisteminizde Go kuruluysa, bu en kolay yöntemdir.

1. Terminali açın ve şu komutu girin:

go install -v github.com/projectdiscovery/katana/cmd/katana@latest

1. Kurulum ~/go/bin/ dizinine yapılır. Bu dizinin sisteminizin PATH'ine ekli olduğundan emin olun.

**Yöntem 2: Binary (Hazır Derlenmiş) Dosya ile Kurulum**

1. [Katana GitHub Releases](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgithub.com%2Fprojectdiscovery%2Fkatana%2Freleases) sayfasına gidin.
2. İşletim sisteminize (Linux, Windows, macOS) uygun olan .zip dosyasını indirin.
3. Arşivden katana dosyasını çıkarın.
4. Çıkardığınız bu dosyayı komutların çalıştığı bir yola taşıyın (örneğin Linux için /usr/local/bin).

# Örnek Linux komutları

wget https://github.com/projectdiscovery/katana/releases/download/v1.0.4/katana\_1.0.4\_linux\_amd64.zip

unzip katana\_1.0.4\_linux\_amd64.zip

sudo mv katana /usr/local/bin/

**Headless Mod için Ek Kurulum:** Headless modun çalışabilmesi için sisteminizde **Google Chrome** tarayıcısının kurulu olması gerektiğini unutmayın.

**4. Kullanım Parametreleri**

Katana, esnek kullanım için birçok parametre sunar. İşte en yaygın olanları:

* -u veya -url: Taranacak tek bir URL'i belirtir.
* -list: İçinde satır satır taranacak URL'lerin bulunduğu bir dosyanın yolunu belirtir.
* -d veya -depth: Tarayıcının ne kadar derine ineceğini belirler. (Varsayılan: 2)
* -o veya -output: Çıktıyı belirtilen dosyaya kaydeder.
* -silent: Başlangıç banner'ı olmadan sadece bulunan URL'leri gösterir.
* -headless: JavaScript'in yoğun kullanıldığı siteler için headless tarama modunu aktif eder.
* -jc veya -js-crawl: JavaScript dosyalarının içindeki uç noktaları da ayrıştırır ve tarar.
* -kf veya -known-files: Bilinen hassas dosyaları/yolları (örneğin .git, /.env) da taramaya dahil eder.
* -ef veya -extension-filter: Sadece belirtilen uzantılara sahip URL'leri sonuç olarak gösterir (örneğin js,json,html).
* -fs veya -field-scope: Sadece belirtilen alan adından (scope) sonuçları gösterir. cdn, google.com gibi harici linkleri eler.

**5. Etkili Kullanım Örnekleri ve Açıklamaları**

Şimdi Katana'nın gücünü gerçek dünya senaryolarıyla görelim.

**Örnek 1: Standart ve Hızlı Tarama**

Bir web sitesinin haritasını hızlıca çıkarmak için en temel komuttur.

* **Komut:**

katana -u https://testphp.vulnweb.com -d 3 -silent

* **Açıklama:**  
  Bu komut, https://testphp.vulnweb.com adresini taramaya başlar. -d 3 parametresi ile ana sayfadan başlayarak 3 link derinliğine kadar ilerler. Yani ana sayfadaki bir linke, sonra o linkin içindeki bir linke ve sonra onun da içindeki bir linke tıklar. -silent ile sadece bulduğu URL'leri listeleyerek temiz bir çıktı sağlar.

**Örnek 2: JavaScript'e Bağımlı Modern Bir Siteyi Tarama**

Günümüzdeki React, Angular gibi framework'lerle yapılmış siteler, normal bir tarayıcıyla düzgün taranamayabilir. İşte burada headless modu devreye girer.

* **Komut:**

katana -u https://react-shopping-cart-67954.firebaseapp.com/ -headless -jc

* **Açıklama:**  
  Bu komut, JavaScript tabanlı bir siteyi hedefler. -headless parametresi, Katana'nın arka planda bir Chrome tarayıcısı başlatmasını ve siteyi bir kullanıcı gibi "görmesini" sağlar. Bu sayede JavaScript'in oluşturduğu linkleri de keşfedebilir. -jc (JavaScript Crawl) ise bulduğu .js dosyalarının içine de bakarak oralardaki gizli yolları (endpoints) bulmaya çalışır.

**Örnek 3: Subfinder -> Httpx -> Katana Zinciri**

Project Discovery ekosisteminin gücünü birleştiren klasik ve çok güçlü bir otomasyon örneği.

* **Komut:**

subfinder -d example.com -silent | httpx -silent | katana -silent -d 3 -o example\_endpoints.txt

* **Açıklama:**
  1. subfinder -d example.com -silent: example.com'a ait tüm alt alan adlarını (subdomain) bulur.
  2. | httpx -silent: Bulunan alt alan adlarından hangilerinde aktif bir web sunucusu çalıştığını kontrol eder ve sadece çalışanları bir sonraki komuta aktarır.
  3. | katana -silent -d 3: Aktif web sunucusu olan her bir URL'i 3 derinliğinde tarar.
  4. -o example\_endpoints.txt: Tüm bu taramalar sonucunda bulunan binlerce URL'i example\_endpoints.txt dosyasına kaydeder. Bu komut, bir şirketin tüm web varlığının haritasını dakikalar içinde çıkarmanızı sağlar.

**Örnek 4: Sadece JavaScript ve JSON Dosyalarını Bulma**

Bazen tüm URL'ler yerine sadece belirli dosya türleriyle ilgileniriz. Örneğin, API anahtarları veya hassas bilgiler içerebilecek JavaScript dosyalarını toplamak isteyebiliriz.

* **Komut:**

katana -u https://example.com -headless -ef js,json

**Açıklama:**  
Bu komut, example.com sitesini headless modda tarar, ancak -ef js,json (extension filter) parametresi sayesinde sonuç olarak sadece .js ve .json uzantılı dosyaların URL'lerini listeler. Bu, hedefe yönelik analiz yapmak için mükemmel bir yöntemdir.

**Örnek 5: Kapsamlı Keşif ve Bilinen Zafiyetli Dosyaları Arama**

Bir hedef üzerinde mümkün olan en geniş kapsamlı taramayı yapmak ve bilinen zafiyetli yolları (örn: .env, /.git konfigürasyon dosyaları) da aramak istediğimizde bu komutu kullanırız.

* **Komut:**

katana -u https://example.com -d 4 -fs host -jc -kf -o full\_scan.txt

* **Açıklama:**
  + -d 4: Taramayı 4 derinliğe kadar indirir.
  + -fs host: Sadece example.com alan adı kapsamında kalır, dışarıya link vermez (field scope).
  + -jc: JavaScript dosyalarını da tarar.
  + -kf: Bilinen (known files) ve genellikle hassas bilgi içeren dosyaları/dizinleri de arar.
  + -o full\_scan.txt: Bulunan her şeyi full\_scan.txt dosyasına kaydeder. Bu, bir hedef hakkında toplanabilecek en detaylı URL listelerinden birini oluşturur.

**Sonuç: Keşif Süreçlerinizin Hızlandırıcısı**

Katana, basit bir web örümceğinden çok daha fazlasıdır. Modern web'in karmaşıklığına adapte olabilen, otomasyon zincirlerinin merkezine oturabilen ve siber güvenlik uzmanlarına inanılmaz bir hız ve verimlilik katan bir keşif platformudur. Saldırı yüzeyinizi haritalandırmak, gizli kalmış uç noktaları bulmak veya sadece bir web sitesinin yapısını anlamak istiyorsanız, Katana keskinliği ve hızıyla siber güvenlik araç kutunuzun en değerli parçalarından biri olmaya adaydır.