**Web'in Gizli Kapılarını Aralayan Anahtar: Dirsearch Nedir ve Nasıl Kullanılır?**

İnternette gezindiğimiz her web sitesi, aslında bir buzdağına benzer. Biz sadece suyun üzerindeki görünen kısmını, yani ana sayfayı, "Hakkımızda" veya "İletişim" gibi bağlantıları tıklarız. Peki ya suyun altındaki devasa kısım? İşte o kısımda, geliştiricilerin unuttuğu test sayfaları, yanlışlıkla açık bırakılan yönetim panelleri, hassas bilgiler içeren yapılandırma dosyaları veya gizli API uç noktaları olabilir. Bu gizli kapıları bulmak için kullanılan en etkili dijital maymuncuklardan biri de **Dirsearch**'tür.

Bu yazıda, Dirsearch'ün ne olduğunu, neden bu kadar önemli olduğunu, nasıl kurulduğunu ve en önemlisi, bir siber güvenlik uzmanı gibi onu nasıl etkili bir şekilde kullanabileceğinizi detaylı örneklerle ele alacağız.

**1. DİRSEARCH Aracı Nedir?**

**Dirsearch**, web sunucularında bulunan "gizli" veya doğrudan bağlantı verilmeyen dizinleri ve dosyaları keşfetmek için tasarlanmış, komut satırı tabanlı bir araçtır. Python ile yazılmıştır ve temel çalışma prensibi oldukça basittir: Belirlediğiniz bir hedef web sitesine karşı, önceden hazırlanmış kelime listelerini (wordlist) kullanarak binlerce olası dizin ve dosya adını dener. Sunucudan "200 OK" (Başarılı), "403 Forbidden" (Yasaklanmış) gibi olumlu bir yanıt aldığında, bu sonucu size raporlar.

Kısacası Dirsearch, bir web sunucusunun kapısını binlerce farklı anahtarla tek tek deneyerek hangilerinin açıldığını size gösteren bir otomasyon aracıdır.

**2. Neden Kullanılır?**

Bir siber güvenlik uzmanı veya sızma testi (penetration test) uzmanı için bir hedefin "saldırı yüzeyini" genişletmek en öncelikli adımdır. Dirsearch tam olarak bu amaca hizmet eder. Kullanım nedenlerini şu şekilde sıralayabiliriz:

* **Yönetim Panellerini Keşfetmek:** /admin, /login, /dashboard gibi standart yollarla bulunamayan yönetim panellerini tespit etmek için kullanılır.
* **Hassas Dosyaları Bulmak:** Yedek dosyaları (.bak, .zip, .old), yapılandırma dosyaları (.env, wp-config.php.bak), log dosyaları (.log) veya kaynak kodları ifşa eden dosyaları bulabilir.
* **Gizli API Uç Noktalarını Tespit Etmek:** Modern web uygulamaları API'ler üzerinden çalışır. Dirsearch, belgelenmemiş veya gizli kalmış API yollarını (/api/v1/users) ortaya çıkarabilir.
* **Unutulmuş İçerikleri Bulmak:** Geliştirme aşamasında oluşturulmuş ancak yayından kaldırılması unutulmuş test sayfalarını (test.php, info.php) bulmak için idealdir.
* **Dizin Listeleme Zafiyetini Sömürmek:** Bazı sunucular, bir dizin içinde index.html gibi bir varsayılan dosya bulunmadığında o dizindeki tüm dosyaları listeler. Dirsearch bu tür dizinleri bularak zafiyeti ortaya çıkarır.

**3. Nasıl Kurulum Yapılır?**

Dirsearch'ü kurmak oldukça basittir. Genellikle üç popüler yöntem kullanılır:

**Yöntem 1: pip ile Kurulum (En Kolay ve Önerilen)**

Python'un paket yöneticisi olan pip ile tek bir komutla kurabilirsiniz. Bu yöntem, bağımlılıkları da otomatik olarak yönetir.

pip3 install dirsearch

**Yöntem 2: Git ile Klonlayarak Kurulum**

Aracın en güncel sürümünü doğrudan GitHub deposundan çekmek isterseniz bu yöntemi kullanabilirsiniz.

# 1. Projeyi klonlayın

git clone https://github.com/maurosoria/dirsearch.git

# 2. Proje dizinine gidin

cd dirsearch

# 3. Gerekli Python kütüphanelerini kurun

pip3 install -r requirements.txt

Bu kurulumdan sonra aracı ./dirsearch.py komutuyla çalıştırırsınız.

**Yöntem 3: Kali Linux Üzerinde Kurulum**

Eğer Kali Linux kullanıyorsanız, Dirsearch genellikle sistemin paket depolarında mevcuttur ve apt ile kolayca kurulabilir.

sudo apt update && sudo apt install dirsearch

**4. Kullanım Parametreleri Nedir?**

Dirsearch'ün gücü, sunduğu esnek parametrelerden gelir. İşte en sık kullanılan parametreler:

* -u veya --url: Hedef web sitesinin URL'sini belirtir. (Örn: -u https://example.com)
* -w veya --wordlist: Taramada kullanılacak kelime listesinin yolunu belirtir. Belirtilmezse kendi varsayılan listesini kullanır.
* -e veya --extensions: Aranacak dosya uzantılarını belirtir. (Örn: -e php,html,js,bak)
* -t veya --threads: Tarama hızını artırmak için aynı anda çalışacak iş parçacığı (thread) sayısını belirtir. (Örn: -t 50)
* -r veya --recursive: Bir dizin bulduğunda, o dizinin içinde de tarama yapmaya devam eder. (Örn: /admin bulursa, /admin/ altında da tarama yapar.)
* -x veya --exclude-status-codes: Sonuçlarda gösterilmeyecek HTTP durum kodlarını belirtir. Gürültüyü azaltmak için kullanışlıdır. (Örn: -x 404,403)
* -o veya --output: Tarama sonuçlarını belirtilen dosyaya kaydeder. (Örn: -o ciktilar.txt)
* --full-url: Çıktıda sadece bulunan yolu (/admin) değil, tam URL'yi (https://example.com/admin) gösterir.

**5. Etkili Kullanım İçin 5 Örnek ve Açıklamaları**

Teoriyi pratiğe dökme zamanı! İşte Dirsearch'ü etkili bir şekilde kullanmanızı sağlayacak 5 farklı senaryo:

**Örnek 1: Hızlı ve Temel Tarama**

Bu, bir hedefe yönelik ilk keşif adımıdır. Aracın varsayılan kelime listesini kullanarak genel bir tarama yaparız.

dirsearch -u https://hedefsite.com -e php,html,js

* **Açıklama:** Bu komut, hedefsite.com adresinde .php, .html ve .js uzantılı dosyaları ve yaygın dizin adlarını arar. Bir sızma testinin başında, hedef hakkında hızlıca bir fikir edinmek için idealdir.

**Örnek 2: Hızlı, Derinlemesine ve Gürültüsüz Tarama**

Büyük bir siteyi tararken hızı artırmak, daha derine inmek ve alakasız sonuçları filtrelemek önemlidir.

dirsearch -u https://buyukwebsitesi.com -t 60 -r -x 404,403

* **Açıklama:**
  + -t 60: Taramayı 60 iş parçacığı ile yaparak ciddi şekilde hızlandırır.
  + -r: Bir dizin bulduğunda (örneğin /uploads/), onun alt dizinlerini de tarayarak daha derinlere iner.
  + -x 404,403: "Bulunamadı" (404) ve "Yasaklandı" (403) hatalarını sonuçlarda göstermeyerek sadece erişilebilir ve potansiyel olarak ilginç kaynaklara odaklanmamızı sağlar.

**Örnek 3: Yedek ve Yapılandırma Dosyalarını Hedef Alma**

En kritik bulgular genellikle unutulmuş yedek ve yapılandırma dosyalarından gelir.

dirsearch -u https://hedefsite.com -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e bak,zip,sql,txt,old,swp,conf,yml -o backup\_scan.txt

* **Açıklama:**
  + -w ...: Varsayılan kelime listesi yerine, daha kapsamlı olan dirbuster kelime listesini kullanırız.
  + -e ...: Aramayı özellikle .bak (yedek), .zip (arşiv), .sql (veritabanı dökümü), .conf (yapılandırma) gibi tehlikeli olabilecek uzantılarla sınırlarız.
  + -o backup\_scan.txt: Sonuçları daha sonra analiz etmek üzere bir dosyaya kaydederiz.

**Örnek 4: API Keşfi**

Modern uygulamaların bel kemiği olan API'ları hedef alan bir tarama.

dirsearch -u https://api.hedef.com -w /path/to/my/api\_wordlist.txt -e json --full-url

* **Açıklama:**
  + -u https://api.hedef.com: Taramayı doğrudan API subdomain'ine yönlendiririz.
  + -w .../api\_wordlist.txt: users, posts, v1, v2, auth gibi API'lara özgü kelimeler içeren özel bir kelime listesi kullanırız.
  + --full-url: Sonuçları tam URL olarak gösterir, böylece bulunan API uç noktalarını doğrudan tarayıcıda veya başka bir araçta test etmek kolaylaşır.

**Örnek 5: Sanal Barındırma (Virtual Host) Taraması**

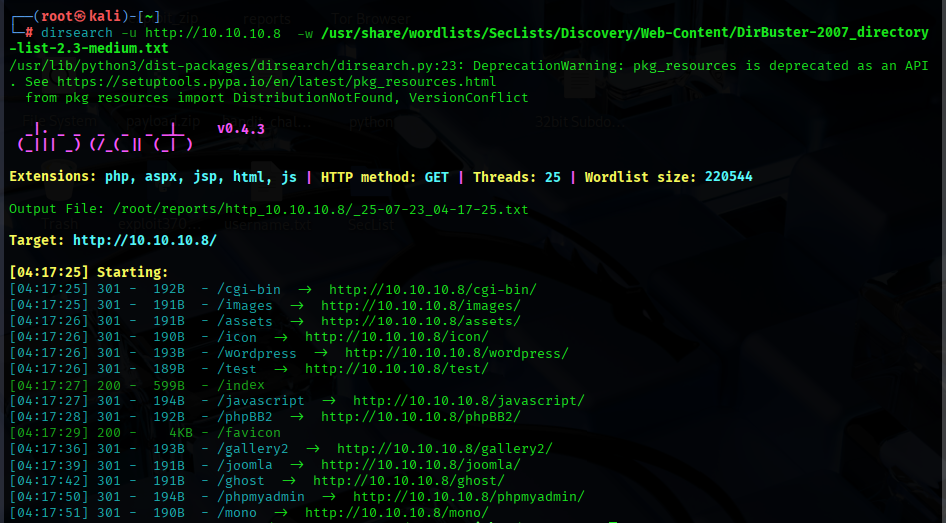
Bazen bir IP adresinde birden fazla web sitesi barındırılır. Dirsearch, belirli bir siteyi hedeflemek için "Header" kullanabilir.

dirsearch -u http://192.168.1.100 -H "Host: farkli-site.com"

* **Açıklama:**
  + -u http://192.168.1.100: Taramayı doğrudan sunucunun IP adresine yaparız.
  + -H "Host: farkli-site.com": Sunucuya gönderilen isteğe bir Host başlığı ekleriz. Bu, sunucuya "Ben aslında farkli-site.com adresine ulaşmak istiyorum" demenin bir yoludur. Bu sayede, aynı IP'de barınan doğru web sitesinin dizinlerini taramış oluruz.

**Örnek: Dirsearch ile Dizin Keşfi**

Dirsearch aracı, hedef web sunucusunda erişilebilir dizin ve dosyaları tespit etmek amacıyla kullanılan güçlü bir brute-force dizin tarayıcısıdır. Aşağıda yer alan ekran görüntüsünde, http://10.10.10.8/ adresine karşı bir tarama gerçekleştirilmiştir.

  
Bu çıktıda Dirsearch aracının -w parametresiyle bir wordlist (dizin listesi) kullanarak hedefe GET isteği gönderdiği ve 301 durum kodlarıyla yönlendirme yapılan dizinleri tespit ettiği görülmektedir. Tarama sonucunda /cgi-bin/, /images/, /assets/, /wordpress/, /phpmyadmin/ gibi birçok potansiyel saldırı yüzeyi ortaya çıkarılmıştır. Özellikle phpmyadmin gibi yönetici paneline erişim sağlayabilecek dizinler, sistemin güvenliği açısından kritik önem taşır. Aracın çok iş parçacıklı (25 threads) çalıştırılması sayesinde hızlı bir tarama gerçekleştirilmiş ve çıktılar /root/reports/ dizinine kaydedilmiştir.

**Sonuç**

Dirsearch, basit gibi görünen ama doğru kullanıldığında inanılmaz derecede güçlü sonuçlar üreten bir araçtır. Bir web uygulamasının güvenliğini test ederken, görünürdeki bağlantıların ötesine geçerek gizli kalmış zafiyetleri ve varlıkları ortaya çıkarmak için vazgeçilmezdir. Ancak unutulmamalıdır ki, bu tür araçlar büyük bir sorumluluk gerektirir. Dirsearch'ü **yalnızca yasal olarak test etme izniniz olan sistemlerde**, etik hackerlık ve sızma testi faaliyetleri kapsamında kullanmalısınız. Aksi takdirde yasa dışı bir eylem gerçekleştirmiş olursunuz.

Şimdi bu dijital anahtarı alın ve sorumlu bir şekilde web'in gizli dehlizlerini keşfetmeye başlayın