## **ExtensionHound Nedir?**

**ExtensionHound**, bir web sunucusunda barınan dosyaların **uzantılarına (file extensions)** odaklanarak, sistemin potansiyel olarak **yanlış yapılandırılmış ya da açıkta bırakılmış dosyalarını** tespit etmeye yarayan bir keşif (recon) aracıdır.

Özellikle .bak, .git, .swp, .old, .zip, .tar.gz, .env gibi **hassas veya yedek dosya uzantılarına sahip içerikleri** bulmak için kullanılır.  
Bu tür dosyalar çoğunlukla geliştiriciler tarafından yanlışlıkla sunucuda bırakılır ve içeriklerinde parola, yapılandırma bilgileri ya da kaynak kod bulunabilir.

## **Neden Kullanılır?**

* Web uygulaması üzerinde yanlışlıkla bırakılmış veya gizlenmemiş **geliştirici dosyalarını** tespit etmek için,
* Özellikle **yedekler**, **yapılandırma dosyaları** veya **kod kalıntıları** gibi, dış dünyaya açık olmaması gereken dosyaları yakalamak için,
* Recon aşamasında, sistemin nasıl çalıştığına dair ipuçları toplamak amacıyla.

Kısacası, saldırganların hedefteki sistem hakkında bilgi toplamasına neden olabilecek dosya türlerini tespit eder.

## **Kullanım Örneği**

Önce hedef URL’yi ve aranacak dosya türlerini belirlemeniz gerekir. Basit bir örnek:

* python3 extensionhound.py -u [https://target.com](https://target.com/)

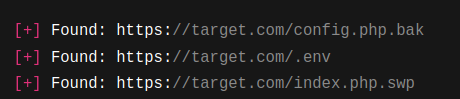
Bu komut, https://target.com adresinde yaygın ve hassas bilinen dosya uzantılarını arar.

## Parametreler

* -u **veya** --url:  
  Tarama yapılacak hedef URL’yi belirtir.  
  (Örnek: -u https://example.com)
* -e **veya** --extensions:  
  Taranacak dosya uzantılarını virgülle ayırarak belirtir.  
  (Örnek: -e bak,old,log)
* -w **veya** --wordlist (opsiyonel):  
  Kullanılacak özel bir kelime listesi dosyasını belirtir.  
  Böylece sadece belirli dosya adlarını uzantılarıyla tarayabilirsiniz.
* -o **veya** --output:  
  Bulunan sonuçların yazılacağı çıktı dosyasını belirtir.  
  (Örnek: -o found.txt)
* --timeout:  
  Her HTTP isteği için zaman aşımı süresi (saniye).
* --threads:  
  Aynı anda kaç bağlantı kurulacağını belirtir. Tarama hızını artırmak için kullanılır.

## **Elde Edilebilecek Sonuçlar**

Tarama sonucunda, hedef sunucuda erişilebilir olan dosya yolları listelenir. Örneğin:



u gibi dosyaların içeriği incelendiğinde, genellikle şunlar ortaya çıkabilir:

* Veritabanı bağlantı bilgileri
* API anahtarları
* Kullanıcı adı/parolalar
* Kaynak kod ve mantık hataları
* Versiyon bilgileri ve konfigürasyon detayları

Bunlar da daha büyük zafiyetlerin (örneğin RCE, SQLi) kapısını aralayabilir.