**Nuclei**

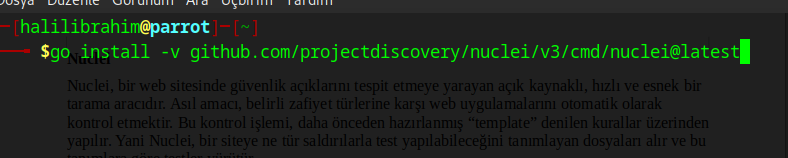
Nuclei, bir web sitesinde güvenlik açıklarını tespit etmeye yarayan açık kaynaklı, hızlı ve esnek bir tarama aracıdır. Asıl amacı, belirli zafiyet türlerine karşı web uygulamalarını otomatik olarak kontrol etmektir. Bu kontrol işlemi, daha önceden hazırlanmış “template” denilen kurallar üzerinden yapılır. Yani Nuclei, bir siteye ne tür saldırılarla test yapılabileceğini tanımlayan dosyaları alır ve bu tanımlara göre testler yürütür.

Örneğin bir web sitesine girdiğinizde sadece gördüğünüz ana sayfa değil, arka planda çalışan çok sayıda servis olabilir. Kullanıcı giriş sayfası, yönetici paneli, dosya yükleme alanları, veritabanı bağlantıları veya API uç noktaları gibi. Nuclei, bu gibi noktaları tespit edip, yaygın bilinen açıklara karşı test eder.

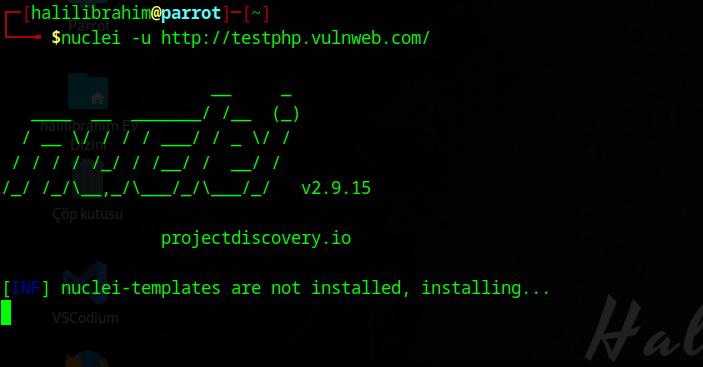
Bazı örnek kontrolleri şöyle düşünebiliriz:

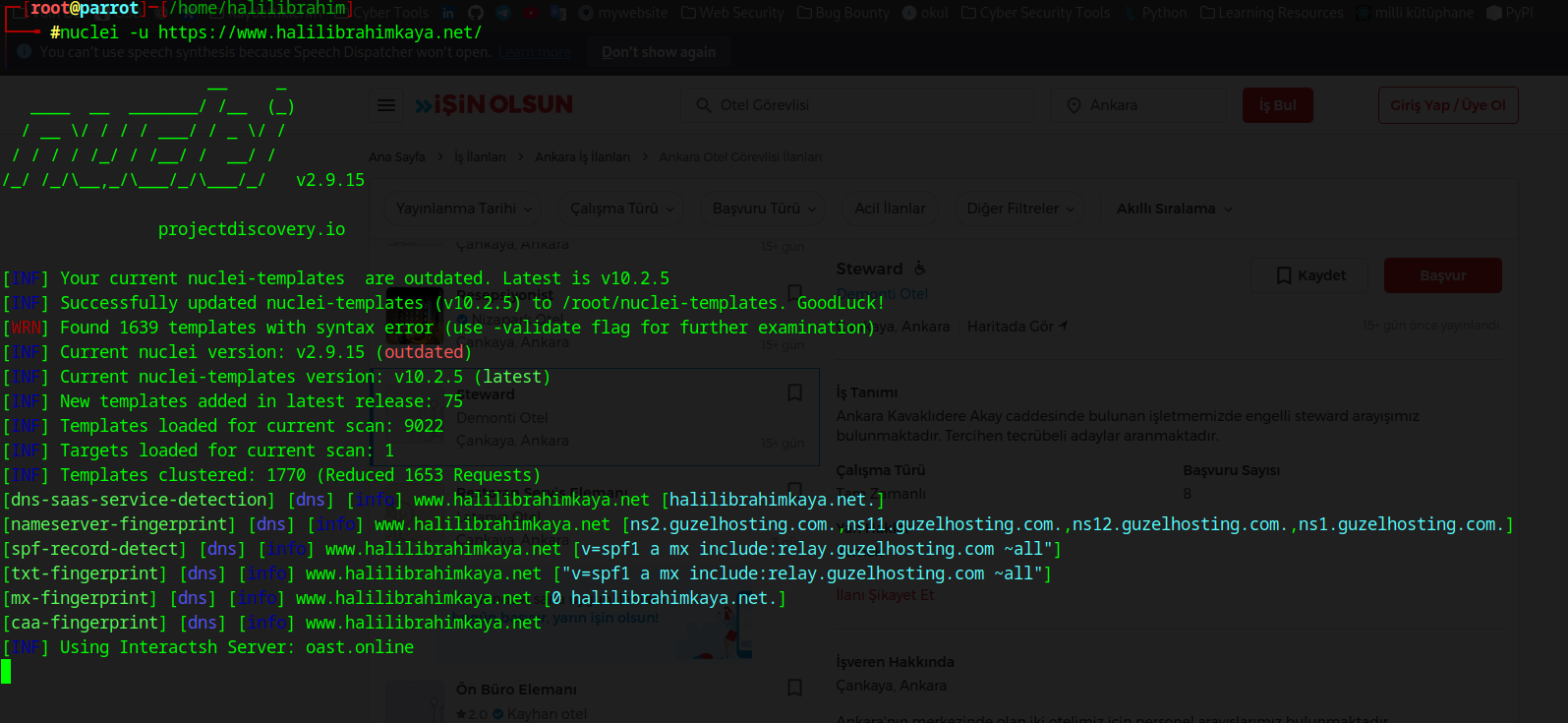
* Giriş sayfası brute-force saldırılarına açık mı?
* Site içinde hassas dosyalar (yapılandırma dosyaları, anahtarlar) herkese açık mı?
* Kullanılan yazılım bileşenleri güncel mi, eski sürümlerden dolayı güvenlik riski var mı?
* Yaygın zafiyetler (XSS, SQLi, RCE gibi) bu sitede var mı?

Şimdi Nuclei’yi kurmak için terminali açıp aşağıdaki komutu giriyoruz.



Şimdi bir örnek yapalım. Belirli bir adresi taramak için “nuclei -u <url>” komutunu kullanabiliriz.





Tarama başlatıldığında, ilk olarak Nuclei'nin kendi güncellemeleri kontrol ediliyor ve gerekli şablonlar (template) yükleniyor. Buradaki "1639 templates with syntax error" uyarısı, bazı şablonların eski ya da hatalı olabileceğini belirtiyor ama taramanın çalışmasını engellemez. Hedef olarak verilen site için toplam 9022 şablondan uygun olanlar uygulanıyor.

Elde edilen bulgular arasında sunucunun barındırdığı yazılımlar, sürümler ve eksik güvenlik başlıkları (security headers) yer alıyor. Örneğin:

* Sunucu Nginx kullanıyor (sürüm 1.19.0).
* PHP sürümü 5.6.40 olarak tespit edilmiş.
* Kullanılan teknoloji Dreamweaver olarak tanımlanmış.
* Ve en önemlisi: birçok HTTP güvenlik başlığının eksik olduğu raporlanmış.

Bu eksik başlıklar arasında Content-Security-Policy, X-Frame-Options, Cross-Origin-Resource-Policy gibi modern web uygulamalarında mutlaka yer alması gereken başlıklar bulunuyor. Bu başlıkların eksikliği, siteyi clickjacking, veri sızıntısı veya tarayıcı bazlı saldırılara açık hale getirebilir.

Özetle bu çıktı, Nuclei’nin bir site üzerinde **yüzlerce potansiyel zafiyeti otomatik olarak tarayabildiğini** gösteren güçlü bir örnektir. Özellikle bug bounty, sızma testi ya da siber güvenlik eğitimi alan birisi için oldukça yol göstericidir. Blog yazınızda bu çıktıyı, Nuclei'nin ne işe yaradığını gösteren ilk somut örnek olarak kullanabilirsiniz.

Şimdi Nuclei’de kullanabileceğimiz bazı parametrelerden bahsedelim.

* -u <target>  
  Belirli bir hedef URL üzerinde tarama yapar.  
  Örnek: nuclei -u https://example.com
* -l <list.txt>  
  Birden fazla hedefi dosyadan alarak tarar.  
  Örnek: nuclei -l targets.txt
* -t <template or directory>  
  Belirli bir şablonu veya şablon klasörünü kullanır.  
  Örnek: nuclei -t vulnerabilities/
* -tags <tag1,tag2>  
  Sadece belirtilen etiketlere sahip şablonları çalıştırır.  
  Örnek: nuclei -u site.com -tags cve
* -severity <level>  
  Belirtilen şiddet seviyesine göre filtreleme yapar.  
  Örnek: nuclei -u site.com -severity critical,high
* -etags <tag1,tag2>  
  Belirtilen etiketlere sahip şablonları **hariç tutar**.  
  Örnek: nuclei -u site.com -etags fuzz
* -o <output.txt>  
  Çıktıyı belirtilen dosyaya kaydeder.
* -silent  
  Sadece sonuçları gösterir, log mesajlarını gizler.
* -json  
  Çıktıyı JSON formatında verir (makine tarafından okunabilir).
* -proxy <http://127.0.0.1:8080>  
  Trafiği belirli bir proxy üzerinden geçirir (örneğin Burp Suite için).
* -rate-limit <sayi>  
  Saniyede yapılacak istek sayısını sınırlar.  
  Örnek: -rate-limit 100
* -retries <sayi>  
  Hatalı isteklerde yeniden deneme sayısını belirtir.
* -timeout <süre>  
  Her isteğin maksimum bekleme süresini milisaniye cinsinden ayarlar.