**Web Güvenliğinin Otomatik Gözcüsü: Kapsamlı Zafiyet Tarayıcısı Wapiti**

Bir siber güvenlik uzmanı bir web uygulamasını test etmeye başladığında, karşısında devasa bir kontrol listesi bulunur: SQL Enjeksiyonu, Cross-Site Scripting (XSS), Dosya Dahil Etme (File Inclusion), Komut Enjeksiyonu ve daha nicesi... Bu zafiyetlerin her birini manuel olarak test etmek, özellikle büyük ve karmaşık sitelerde haftalar sürebilir. Peki ya tüm bu temel kontrolleri sizin için otomatik olarak yapan, bulgularını düzenli bir rapor halinde sunan bir yardımcınız olsaydı? İşte bu yardımcının adı **Wapiti**'dir.

Bu yazıda, Wapiti'nin ne olduğunu, neden sızma testlerinde değerli bir ilk adım olduğunu, nasıl kurulduğunu ve bu kapsamlı tarayıcıyı en verimli şekilde nasıl kullanabileceğinizi pratik örneklerle keşfedeceğiz.

**1. Wapiti Aracı Nedir?**

**Wapiti**, bir web uygulamasının güvenliğini denetlemek için tasarlanmış, açık kaynaklı ve komut satırı tabanlı bir "black-box" (kara kutu) zafiyet tarayıcısıdır. "Kara kutu" olması, testleri gerçekleştirmek için uygulamanın kaynak koduna ihtiyaç duymadığı anlamına gelir. Tıpkı gerçek bir saldırgan gibi, uygulamaya dışarıdan bakar ve onunla etkileşime girerek zafiyetleri bulmaya çalışır.

Wapiti'nin çalışma prensibi iki ana adımdan oluşur:

1. **Crawl (Tarama/Örümcekleme):** İlk olarak, hedef web sitesini bir örümcek gibi gezer. Ana sayfadan başlayarak tüm bağlantıları takip eder, formları, giriş alanlarını, URL parametrelerini ve test edilebilecek diğer tüm noktaları keşfederek sitenin bir haritasını çıkarır.
2. **Attack (Saldırı/Test):** Haritasını çıkardığı bu noktalara, önceden tanımlanmış zafiyet modüllerini kullanarak saldırı kodları (payload) gönderir. Sunucudan gelen yanıtları analiz ederek bir zafiyetin var olup olmadığını tespit eder.

Wapiti, aşağıdaki gibi çok sayıda kritik zafiyeti test edebilir:

* SQL Injection (SQL Enjeksiyonu)
* Cross-Site Scripting (XSS)
* File Inclusion / Disclosure (Dosya Dahil Etme / İfşası - LFI/RFI)
* Command Execution (Komut Enjeksiyonu)
* XXE Injection (XML External Entity)
* CRLF Injection (HTTP Yanıt Bölme)
* Ve daha birçokları...

**2. Neden Kullanılır?**

Wapiti, siber güvenlik uzmanlarının araç setinde önemli bir yer tutar:

* **Kapsamlı ve Hızlı İlk Değerlendirme:** Bir sızma testinin başında, hedef hakkında hızlıca bir fikir edinmek ve "düşük asılı meyveleri" (low-hanging fruits), yani kolayca bulunabilecek bariz zafiyetleri toplamak için mükemmeldir.
* **Otomasyon ve Verimlilik:** Manuel olarak saatler sürecek testleri dakikalar içinde otomatik olarak gerçekleştirerek zamandan büyük tasarruf sağlar.
* **Geniş Zafiyet Yelpazesi:** Tek bir araçla onlarca farklı zafiyet türünü tarayabilme imkanı sunar.
* **Detaylı Raporlama:** Tarama sonunda bulguları özetleyen, anlaşılır ve profesyonel görünümlü HTML raporları oluşturabilir. Bu raporlar, bulguları geliştiricilere veya müşterilere sunmak için idealdir.
* **Erişilebilirlik:** Tamamen ücretsiz ve açık kaynaklıdır, bu da onu herkes için erişilebilir kılar.

**3. Nasıl Kurulum Yapılır?**

Wapiti, Python ile yazılmıştır ve genellikle pip paket yöneticisi ile çok kolay bir şekilde kurulur.

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

Sisteminizde Python 3 ve pip kuruluysa, tek yapmanız gereken aşağıdaki komutu çalıştırmaktır.

pip3 install wapiti3

*Not: Paket adı wapiti3'tür, çünkü aracın Python 3 ile uyumlu sürümünü belirtir.*

**Yöntem 2: Kali Linux (APT ile Kurulum)**

Eğer Kali Linux kullanıyorsanız, Wapiti genellikle sistemin paket depolarında mevcuttur.

sudo apt update && sudo apt install wapiti

Kurulumdan sonra, aracı doğrudan wapiti komutuyla çalıştırabilirsiniz.

**4. Kullanım Parametreleri Nedir?**

Wapiti'nin taramalarınızı özelleştirmenizi sağlayan birçok parametresi vardır. İşte en önemlileri:

* -u veya --url: Taranacak ana hedef URL'yi belirtir.
* -m veya --module: Sadece belirli test modüllerini çalıştırmak için kullanılır. Virgülle ayrılır. (Örn: -m "sql,xss,crlf")
* -x veya --exclude-module: Belirtilen test modüllerini tarama dışında bırakır.
* -s veya --start: Tarayıcının (crawler) başlayacağı URL'yi belirtir. Belirtilmezse, ana URL'den başlar.
* --scope: Taramanın kapsamını belirler. Örneğin, sadece belirtilen URL ve alt yollarıyla sınırlı kalmasını sağlar. (Örn: --scope-url "http://hedef.com/blog")
* --cookie: Kimlik doğrulaması gerektiren sayfaları taramak için kullanılacak oturum çerezini (cookie) içeren dosyayı veya doğrudan çerez dizesini belirtir.
* -p veya --proxy: Taramayı bir proxy üzerinden (örneğin Burp Suite) geçirmek için kullanılır.
* -f veya --format: Çıktı raporunun formatını belirtir (html, json, xml, txt).
* -o veya --output: Raporun kaydedileceği dosyanın adını ve yolunu belirtir.

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

Şimdi Wapiti'yi gerçek senaryolarda nasıl kullanacağımıza bakalım.

**Örnek 1: Temel Tarama ve HTML Raporu Oluşturma**

Bu, bir hedefe yönelik en yaygın ve temel kullanımdır.

wapiti -u http://testphp.vulnweb.com/ -f html -o vulnweb\_raporu.html

* **Açıklama:** Bu komut, http://testphp.vulnweb.com/ adresini tarar, tüm varsayılan zafiyet modüllerini çalıştırır ve bulguları vulnweb\_raporu.html adında, tarayıcıda açabileceğiniz düzenli bir HTML dosyasına kaydeder.

**Örnek 2: Odaklı ve Hızlı Tarama (Sadece SQLi ve XSS)**

Zamanınız kısıtlıysa veya sadece belirli zafiyet türlerinden şüpheleniyorsanız, taramayı sınırlayabilirsiniz.

wapiti -u http://hedefsite.com/ -m "sql,xss"

* **Açıklama:** -m "sql,xss" parametresi sayesinde Wapiti, diğer tüm modülleri atlayarak sadece SQL Enjeksiyonu ve Cross-Site Scripting zafiyetlerini arar. Bu, tarama süresini önemli ölçüde kısaltır.

**Örnek 3: Kimlik Doğrulaması Gerektiren Bir Alanı Tarama**

Birçok zafiyet, sadece giriş yapmış kullanıcıların erişebildiği yönetici panelleri veya profil sayfalarında bulunur.

# Önce tarayıcıdan geçerli bir session cookie'si kopyalanır.

wapiti -u "http://hedefsite.com/dashboard/" --cookie "sessionID=a1b2c3d4e5f6g7h8"

* **Açıklama:** Tarayıcınızın geliştirici araçlarından (F12) aldığınız geçerli oturum çerezini --cookie parametresiyle Wapiti'ye verirsiniz. Bu sayede Wapiti, sanki giriş yapmış bir kullanıcıymış gibi istekler göndererek kimlik doğrulaması gerektiren /dashboard/ alanını ve altındaki sayfaları tarayabilir.

**Örnek 4: Tarama Kapsamını Sınırlandırma**

Büyük bir sitede sadece yeni eklenen bir özelliği (örneğin /blog bölümü) test etmek isteyebilirsiniz.

wapiti -u http://buyuksite.com/ --scope-url "http://buyuksite.com/blog/"

* **Açıklama:** --scope-url parametresi, Wapiti'ye "Sadece http://buyuksite.com/blog/ adresi ve bunun altındaki linkleri takip et, sitenin geri kalanına dokunma" talimatını verir. Bu, taramanın alakasız yerlere gitmesini engeller ve hedefe odaklanmasını sağlar.

**Örnek 5: Taramayı Proxy Üzerinden Geçirerek Analiz Etme**

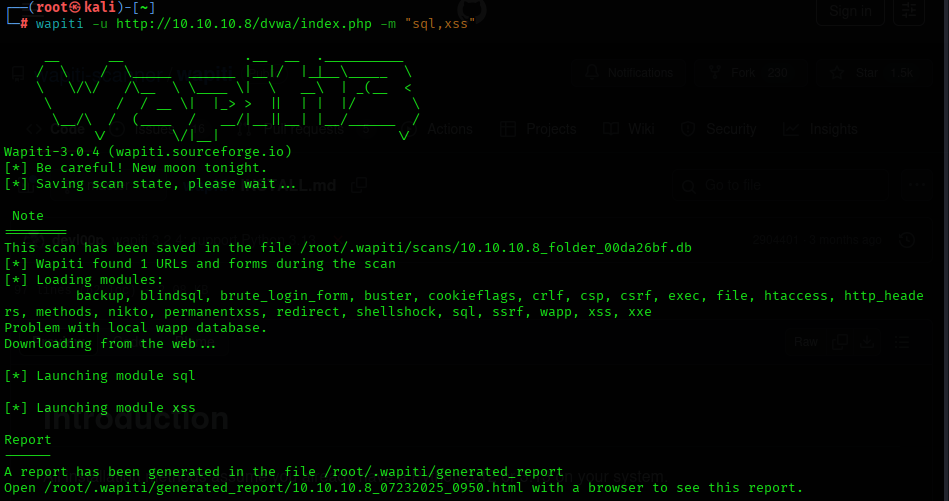
Wapiti'nin arka planda ne yaptığını tam olarak görmek ve istekleri/yanıtları analiz etmek isteyen ileri düzey kullanıcılar için idealdir.

wapiti -u http://hedefsite.com -p http://127.0.0.1:8080

* **Açıklama:** -p http://127.0.0.1:8080 komutu, Wapiti'nin tüm ağ trafiğini yerel makinede 8080 portunda çalışan bir araca (genellikle Burp Suite veya OWASP ZAP) yönlendirir. Bu sayede, Wapiti'nin gönderdiği her bir payload'ı ve sunucunun buna nasıl yanıt verdiğini canlı olarak izleyebilir, değiştirebilir ve analiz edebilirsiniz.

**🦂 Örnek: Wapiti ile SQL ve XSS Zafiyet Taraması**

**Wapiti**, açık kaynak kodlu, modüler yapıya sahip bir web güvenlik tarayıcısıdır. Web uygulamalarını dışarıdan (black-box) tarayarak birçok zafiyet türünü tespit edebilir. Aşağıdaki örnekte, DVWA üzerindeki bir hedef URL’de SQL ve XSS açıkları test edilmiştir.

  
Komut satırında wapiti -u http://10.10.10.8/dvwa/index.php -m "sql,xss" komutu çalıştırılmıştır. -u parametresiyle hedef URL, -m parametresiyle ise taranacak modüller (SQL ve XSS) belirtilmiştir. Wapiti, hedefteki formları ve URL’leri analiz ettikten sonra sql ve xss modüllerini sırayla çalıştırmıştır. Tarama tamamlandığında /root/.wapiti/generated\_report dizininde HTML formatında bir rapor oluşturulmuştur. Bu rapor, tarama sırasında tespit edilen zafiyetleri, ilgili HTTP istekleriyle birlikte detaylı olarak sunar. Wapiti'nin modüler altyapısı sayesinde çok sayıda zafiyet kategorisi (CSRF, SSRF, XXE, RCE vb.) taranabilir.

**Sonuç**

Wapiti, bir web uygulamasının güvenlik duruşuna dair hızlı ve kapsamlı bir ilk bakış sunan, son derece değerli bir otomasyon aracıdır. Tek bir komutla onlarca farklı zafiyeti tarayabilmesi ve sonuçları düzenli bir rapor halinde sunması, onu her seviyeden siber güvenlik uzmanı için pratik bir yardımcı yapar. Manuel testlerin ve derinlemesine analizin yerini tutmasa da, bir sızma testinin keşif aşamasında yolu aydınlatan güçlü bir fener görevi görür.

Her zaman olduğu gibi, bu tür güçlü tarayıcıları kullanırken büyük bir sorumluluk gerekir. Wapiti'yi **yalnızca ve kesinlikle test etme izniniz olan sistemlerde** etik kurallar çerçevesinde kullanarak, dijital dünyayı daha güvenli bir yer haline getirmeye yardımcı olabilirsiniz.