**Web'in Sinsi Tehdidi XSS'e Karşı Akıllı Kalkan: XSStrike Nedir ve Nasıl Ustalıkla Kullanılır?**

Web uygulamalarının en yaygın ve en sinsi güvenlik açıklarından biri şüphesiz Cross-Site Scripting, yani XSS'tir. Bir saldırganın, sizin güvendiğiniz bir web sitesi aracılığıyla tarayıcınızda zararlı JavaScript kodları çalıştırmasına olanak tanıyan bu zafiyet, oturum bilgilerinizi çalıp hesabınızı ele geçirmekten, sizi sahte sitelere yönlendirmeye kadar birçok kötü amaca hizmet edebilir. Peki, bu sinsi tehdidi tespit etmek için basitçe <script>alert(1)</script> yazıp denemek yeterli mi? Çoğu zaman hayır. İşte bu noktada, bir XSS zafiyetini bir dedektif gibi akıllıca arayan **XSStrike** devreye giriyor.

Bu yazıda, XSStrike'ın neden sıradan bir tarayıcıdan çok daha fazlası olduğunu, nasıl kurulduğunu ve bu aracı bir siber güvenlik uzmanı gibi kullanarak web uygulamalarının gizli XSS zafiyetlerini nasıl ortaya çıkarabileceğinizi adım adım, örneklerle anlatacağız.

**1. XSStrike Aracı Nedir?**

**XSStrike**, yalnızca XSS zafiyetlerini tespit etmeye odaklanmış, Python tabanlı gelişmiş bir tarama aracıdır. Onu diğer araçlardan ayıran en temel özelliği "akıllı" olmasıdır. XSStrike, körü körüne payload (saldırı kodu) göndermek yerine, hedefin davranışını analiz eder ve buna göre hareket eder.

**XSStrike'ın Akıllı Yaklaşımı:**

* **Context Analizi (Bağlam Analizi):** Bir parametrenin yansıdığı HTML kodunu analiz eder. Kod bir <script> etiketi içinde mi, bir HTML etiketinin niteliğinde mi, yoksa bir yorum satırında mı yansıyor? Bu bağlama göre en uygun XSS payload'unu kendisi üretir.
* **Payload Üretimi:** Kendi içinde güçlü bir payload üretme motoru vardır. Basit alert() denemelerinin ötesinde, olay yöneticileri (onmouseover, onerror vb.), farklı kodlama teknikleri ve karmaşık JavaScript kodları içeren payload'lar oluşturur.
* **WAF Tespiti ve Bypass:** Hedefte bir Web Application Firewall (WAF - Web Uygulama Güvenlik Duvarı) olup olmadığını tespit etmeye çalışır ve bazı bilinen WAF'ları atlatmak için özel olarak tasarlanmış payload'lar kullanır.
* **Akıllı Fuzzer:** Sadece payload göndermez, aynı zamanda parametreleri analiz ederek onları fuzz'lar (rastgele veriyle test eder) ve beklenmedik davranışları tespit eder.

**2. Neden Kullanılır?**

XSStrike, XSS avcıları için bir İsviçre çakısı gibidir. Kullanım nedenleri şunlardır:

* **Yüksek Başarı Oranı:** Akıllı context analizi sayesinde, standart tarayıcıların gözden kaçıracağı karmaşık XSS zafiyetlerini bulma olasılığı daha yüksektir.
* **Zaman Tasarrufu:** Manuel olarak yüzlerce farklı payload denemek yerine, otomasyonla en etkili payload'ları saniyeler içinde test eder.
* **WAF'ları Atlatma Yeteneği:** Güvenlik duvarları tarafından korunan sistemlerde bile zafiyet bulma şansını artırır.
* **Kapsamlı Tarama:** Bir web sitesini baştan sona tarayarak (crawl özelliği ile) enjekte edilebilir tüm noktaları kendi kendine keşfeder.
* **DOM XSS Tespiti:** Sadece sunucu tabanlı (Reflected, Stored) XSS'i değil, aynı zamanda istemci tarafında tetiklenen DOM tabanlı XSS zafiyetlerini de bulabilen bir tarayıcı motoruna sahiptir.

**3. Nasıl Kurulum Yapılır?**

XSStrike'ı kurmak oldukça basittir ve genellikle git ile klonlama yöntemi kullanılır.

# 1. Projeyi GitHub'dan klonlayın

git clone https://github.com/s0md3v/XSStrike.git

# 2. Proje dizinine gidin

cd XSStrike

# 3. Gerekli Python kütüphanelerini kurun (Python 3 gereklidir)

pip3 install -r requirements.txt

Kurulum tamamlandıktan sonra aracı python3 xsstrike.py komutuyla çalıştırabilirsiniz.

**4. Kullanım Parametreleri Nedir?**

XSStrike, taramalarınızı özelleştirmeniz için çeşitli parametreler sunar. İşte en sık kullanılanlar:

* -u veya --url: Hedef URL'yi belirtir. Taranacak parametreleri içermelidir.
* --data: POST metodu ile yapılacak taramalar için gönderilecek veriyi belirtir.
* --crawl: Hedef web sitesinde yeni linkler keşfetmek için siteyi tarar ve bulduğu tüm parametreleri test eder.
* --fuzzer: Parametreleri analiz etmek ve olası zafiyetleri bulmak için akıllı fuzzer'ı çalıştırır.
* -t veya --threads: Tarama hızını artırmak için kullanılacak thread sayısını belirtir.
* -l veya --level: Taramanın ne kadar derine ineceğini belirtir (crawl ile kullanılır). Varsayılan 2'dir.
* --headers: HTTP isteğine özel başlıklar (Cookie, User-Agent vb.) eklemek için kullanılır.
* --timeout: Bir isteğin ne kadar süre sonra zaman aşımına uğrayacağını belirtir.
* --skip-dom: DOM XSS taramasını atlamak için kullanılır.

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

Şimdi XSStrike'ın gücünü ortaya çıkaran pratik senaryolara bakalım.

**Örnek 1: Basit Bir GET Parametresini Test Etme**

En temel kullanım, bir URL'deki arama parametresi gibi tek bir noktayı test etmektir.

python3 xsstrike.py -u "http://hedefsite.com/search.php?q=test"

* **Açıklama:** Bu komut, XSStrike'a search.php sayfasındaki q parametresini hedeflemesini söyler. XSStrike, q parametresinin sayfada nereye ve nasıl yansıdığını analiz edecek, bu bağlama uygun payload'lar üretecek ve bir XSS zafiyeti olup olmadığını raporlayacaktır.

**Örnek 2: Bir Giriş Formunu (POST İsteği) Test Etme**

Web sitelerindeki birçok form (giriş, yorum, iletişim) veriyi POST metoduyla gönderir.

python3 xsstrike.py -u "http://hedefsite.com/login.php" --data "username=test&password=test"

* **Açıklama:**
  + -u "http://hedefsite.com/login.php": Formun gönderildiği adresi belirtir.
  + --data "username=test&password=test": Formdaki username ve password alanlarının test edileceğini belirtir. XSStrike, bu iki parametreyi de ayrı ayrı en etkili XSS payload'ları ile test edecektir.

**Örnek 3: Bütün Bir Web Sitesini Otomatik Olarak Tarama**

Büyük bir web sitesinde manuel olarak test edilecek parametre bulmak zordur. crawl özelliği burada devreye girer.

python3 xsstrike.py -u http://hedefsite.com/ --crawl -l 3

* **Açıklama:**
  + --crawl: XSStrike, hedefsite.com ana sayfasından başlayarak sitedeki tüm linkleri takip eder.
  + -l 3: Taramanın 3 seviye derine inmesini söyler (Ana Sayfa -> Sayfa 2 -> Sayfa 3). Bu sayede site haritasını çıkarır ve keşfettiği her URL'deki tüm parametreleri otomatik olarak XSS için test eder.

**Örnek 4: Akıllı Fuzzer ile Gizli Zafiyetleri Keşfetme**

Bazen bir XSS, basit bir payload ile tetiklenmez. Parametrenin nasıl işlendiğini anlamak gerekir.

python3 xsstrike.py -u "http://hedefsite.com/profile.php?user\_id=1" --fuzzer

* **Açıklama:** --fuzzer parametresi, XSStrike'a user\_id parametresine çeşitli karakterler ve diziler göndererek sunucunun nasıl tepki verdiğini analiz etmesini söyler. Bir hata mesajı, beklenmedik bir içerik veya farklı bir HTTP yanıtı alırsa, bu durumu bir anomali olarak işaretler ve bu anomaliyi tetikleyen payload'ı size gösterir. Bu, manuel testlerde gözden kaçabilecek zafiyetleri bulmak için harikadır.

**Örnek 5: WAF'ı Atlatmak İçin Cookie ve Header Kullanımı**

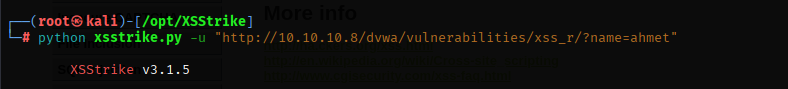
Bazı siteler sadece giriş yapmış kullanıcılara özel sayfalarda zafiyet barındırır veya belirli bir User-Agent bekler.

python3 xsstrike.py -u "http://hedefsite.com/dashboard.php?module=home" --headers "Cookie: session=xyz123abc; User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36"

* **Açıklama:** --headers parametresi, isteklere özel başlıklar eklememizi sağlar.
  + Cookie: session=xyz123abc: XSStrike'ın, sanki geçerli bir kullanıcı oturumuyla istek yapıyormuş gibi davranmasını sağlar. Bu, kimlik doğrulaması gerektiren sayfaları test etmek için zorunludur.
  + User-Agent: ...: Sunucuyu, isteğin sıradan bir tarayıcıdan geldiğine inandırarak bazı temel WAF kurallarını atlatmaya yardımcı olabilir.

**⚔️ Örnek: XSStrike ile XSS Zafiyeti Analizi**

XSStrike, özellikle **Cross-Site Scripting (XSS)** zafiyetlerini tespit etmek ve analiz etmek için geliştirilmiş gelişmiş bir XSS keşif aracıdır. Aşağıda, http://10.10.10.8 üzerindeki DVWA uygulamasında yer alan bir XSS endpoint’ine yönelik yapılan testin başlangıcı görülmektedir.

  
Bu çıktıda xsstrike.py Python dosyası çalıştırılarak, -u parametresiyle hedef URL belirtilmiştir. Parametre değeri olarak name=ahmet kullanılmıştır. XSStrike, bu parametre üzerinden yansıyan XSS açıklarını otomatik olarak test edecek, filtre bypass denemeleri yapacak ve potansiyel yüklerin başarılı olup olmadığını raporlayacaktır. Kullanılan sürüm v3.1.5’tir. Bu araç sadece payload denemekle kalmaz; aynı zamanda DOM analizi, filtreleme sistemi tespiti gibi gelişmiş XSS analizi özelliklerine de sahiptir.

**Sonuç**

XSStrike, XSS zafiyetlerini bulma sürecini otomatikleştiren, ancak bunu körü körüne değil, akıllıca yapan güçlü bir araçtır. Context analizi, WAF tespiti ve akıllı fuzzer gibi yetenekleri, onu her siber güvenlik uzmanının veya etik hacker'ın cephaneliğinde bulunması gereken bir araç haline getirir. Web uygulamalarının güvenliğini sağlamak, bu tür sinsi zafiyetleri proaktif bir şekilde bulup düzeltmekle başlar.

Unutmayın ki, XSStrike gibi güçlü araçlar büyük bir sorumlulukla gelir. Bu aracı **kesinlikle ve yalnızca test etme yetkinizin ve izninizin olduğu sistemlerde** kullanmalısınız. Bilginizi ve yeteneklerinizi siber güvenliği artırmak için etik bir şekilde kullanmak, bir profesyonelin en önemli ilkesidir.