**XSS Avının Geleceği: DalFox Nedir ve Nasıl Bir Usta Gibi Kullanılır?**

Cross-Site Scripting (XSS), web'in en eski ve en inatçı zafiyetlerinden biridir. Ancak web uygulamaları geliştikçe, XSS'i bulma yöntemleri de gelişmek zorunda kaldı. Artık basit bir <script>alert(1)</script> payload'u göndermek, karmaşık JavaScript framework'leri ve Web Uygulama Güvenlik Duvarları (WAF) ile korunan modern sitelerde genellikle işe yaramaz. İşte bu yeni nesil zorluklara karşı geliştirilmiş, daha akıllı, daha hızlı ve daha kapsamlı bir XSS avcısı var: **DalFox**.

Bu yazıda, DalFox'un neden "yeni nesil bir XSS tarayıcısı" olarak anıldığını, onu diğer araçlardan ayıran özelliklerini, nasıl kurulduğunu ve en önemlisi, bu güçlü aracı kullanarak en karmaşık XSS zafiyetlerini bile nasıl ortaya çıkarabileceğinizi pratik örneklerle ele alacağız.

**1. DalFox Aracı Nedir?**

**DalFox**, Go programlama diliyle yazılmış, son derece gelişmiş ve hızlı bir XSS (Cross-Site Scripting) tarayıcısı ve parametre analizi aracıdır. Ünlü siber güvenlik araştırmacısı Hahwul tarafından geliştirilen DalFox, sadece bilinen payload'ları göndermekle kalmaz, aynı zamanda bir hedefin yapısını analiz ederek potansiyel enjeksiyon noktalarını kendi kendine keşfeder.

**DalFox'u Özel Kılan Başlıca Yetenekler:**

* **Parametre Madenciliği (Parameter Mining):** Belki de en güçlü özelliğidir. DalFox, bir URL'yi analiz eder, JavaScript dosyalarını inceler ve formları tarayarak test edilebilecek "gizli" veya bariz olmayan parametreleri kendi kendine bulur.
* **Gelişmiş DOM XSS Taraması:** Arka planda bir "headless browser" (gerçek bir tarayıcının arayüzsüz versiyonu) çalıştırarak, JavaScript kodunun çalışmasıyla ortaya çıkan DOM tabanlı XSS zafiyetlerini tespit edebilir.
* **Blind XSS Tespiti:** Bir payload enjekte ettiğinizde anında bir sonuç görmediğiniz, ancak bu payload'un daha sonra bir yönetici panelinde veya bir log okuyucusunda tetiklendiği "Blind XSS" (Kör XSS) senaryolarını test etmek için yerleşik desteği vardır.
* **Statik Analiz:** JavaScript dosyalarını indirip içeriklerini analiz ederek potansiyel olarak tehlikeli kod parçacıklarını veya zafiyet ipuçlarını bulabilir.
* **Hız ve Verimlilik:** Go dili sayesinde, eş zamanlı olarak çok sayıda isteği hızlı bir şekilde göndererek taramaları kısa sürede tamamlar.

**2. Neden Kullanılır?**

DalFox, XSS testi yapan herkes için vazgeçilmez bir araçtır çünkü:

* **Kapsamlı Keşif Sağlar:** Manuel olarak asla bulamayacağınız parametreleri ve enjeksiyon noktalarını sizin için keşfederek saldırı yüzeyini inanılmaz derecede genişletir.
* **Zaman Kazandırır:** Otomatik parametre keşfi ve hızlı tarama yetenekleri sayesinde, saatler sürebilecek manuel analiz işini dakikalara indirir.
* **Yüksek Tespit Oranına Sahiptir:** Reflected, Stored, DOM ve Blind XSS dahil olmak üzere çok çeşitli XSS türlerini test eder, bu da zafiyet kaçırma olasılığını azaltır.
* **Modern Web Uygulamalarına Uyum Sağlar:** Karmaşık JavaScript framework'leri (React, Vue, Angular) üzerine kurulu sitelerde bile etkili bir şekilde çalışabilir.
* **Otomasyona Uygundur:** "Pipe" modu sayesinde, diğer keşif araçlarının (örneğin subfinder, gau, httpx) çıktılarını doğrudan alıp işleyebilir, bu da onu büyük ölçekli taramalar için mükemmel kılar.

**3. Nasıl Kurulum Yapılır?**

DalFox'u kurmanın birkaç popüler ve kolay yolu vardır.

**Yöntem 1: Go ile Kurulum (En Güncel ve Tavsiye Edilen)**

Eğer sisteminizde Go kurulu ise, bu en basit yöntemdir.

go install github.com/hahwul/dalfox/v2@latest

Bu komut, DalFox'un en son sürümünü indirip kuracak ve dalfox komutunu doğrudan kullanmanıza olanak tanıyacaktır.

**Yöntem 2: GitHub "Releases" Üzerinden İndirme**

Go kurmak istemiyorsanız, projenin [GitHub Releases sayfasından](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgithub.com%2Fhahwul%2Fdalfox%2Freleases) işletim sisteminize (Linux, Windows, macOS) uygun, önceden derlenmiş çalıştırılabilir dosyayı indirebilirsiniz.

**4. Kullanım Parametreleri Nedir?**

DalFox'un birçok güçlü parametresi vardır. İşte en sık kullanılanlar:

* **Çalışma Modları:**
  + dalfox url [URL]: Tek bir URL'yi tarar.
  + dalfox file [DOSYA\_ADI]: Bir dosyadaki URL listesini tarar.
  + dalfox pipe: Standart girdiden (stdin) gelen URL'leri tarar (otomasyon için ideal).
* **Temel Parametreler:**
  + -b veya --blind: Blind XSS testi için kullanılacak callback URL'sini (örneğin XSS Hunter veya Interactsh adresi) belirtir.
  + -p veya --param: Sadece belirtilen parametreleri test eder. (Örn: -p q,search)
  + --data: POST isteği ile veri göndermek için kullanılır.
  + --cookie: Kimlik doğrulaması için kullanılacak çerezi (cookie) belirtir.
  + -H veya --header: Özel HTTP başlıkları ekler. (Örn: -H "Authorization: Bearer token")
  + -w veya --worker: Eş zamanlı çalışacak thread sayısını belirtir. Varsayılan 100'dür.
  + -o veya --output: Sonuçları belirtilen dosyaya kaydeder.
* **Gelişmiş Parametreler:**
  + --mining-dom: DOM'u analiz ederek ek parametreler bulmaya çalışır.
  + --mining-js: JS dosyalarını analiz ederek ek parametreler bulmaya çalışır.
  + --deep-dom-inspection: DOM XSS için daha derinlemesine bir analiz yapar.

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

DalFox'un gerçek gücünü bu pratik senaryolarla görelim.

**Örnek 1: Tek Bir URL'yi Akıllıca Tarama**

Bu, en temel kullanımdır, ancak DalFox arka planda parametre keşfi de yapar.

dalfox url "http://hedefsite.com/search.php?q=test"

* **Açıklama:** DalFox, sadece q parametresini test etmekle kalmaz. Aynı zamanda search.php sayfasının kaynak kodunu ve bağlantılı JS dosyalarını analiz ederek q dışında başka enjekte edilebilir parametreler olup olmadığını da araştırır ve bulduklarını test eder.

**Örnek 2: DOM Parametrelerini Otomatik Keşfedip Test Etme**

Modern JavaScript uygulamalarında parametreler her zaman URL'de görünmez.

dalfox url http://hedefsite.com --mining-dom --mining-js

* **Açıklama:** Bu komut, hedefsite.com adresini ziyaret eder ve --mining-dom ile --mining-js parametreleri sayesinde, sayfanın JavaScript kodları içinde kullanılan gizli parametreleri (location.hash, document.cookie gibi) ve JS değişkenlerini bularak bunları XSS için test eder. Bu, manuel olarak fark edilmesi çok zor olan DOM XSS zafiyetlerini bulmak için paha biçilmezdir.

**Örnek 3: Blind XSS Zafiyetini Test Etme**

Bir iletişim formunun bir yönetici paneline veri gönderdiğini varsayalım.

dalfox url "http://hedefsite.com/feedback" --data "message=hello" -b your-interactsh-url.com

* **Açıklama:**
  + --data "message=hello": message parametresini POST isteği ile test eder.
  + -b your-interactsh-url.com: DalFox, message parametresine kendi Blind XSS payload'unu enjekte eder. Bu payload, your-interactsh-url.com adresine bir istek gönderecek şekilde tasarlanmıştır. Eğer bir yönetici bu mesajı okuduğunda payload tetiklenirse, sizin Interactsh sunucunuza bir bildirim gelir ve Blind XSS zafiyetini doğrulamış olursunuz.

**Örnek 4: Otomasyon Zinciri ile Toplu Tarama (Profesyonel Yöntem)**

Bu, bug bounty avcılarının en sevdiği yöntemdir.

gau hedefsite.com | httpx -silent | dalfox pipe -o dalfox\_results.txt

* **Açıklama (Adım Adım):**
  1. gau hedefsite.com: gau aracı, hedefsite.com alan adına ait, arşivlenmiş ve bilinen tüm URL'leri (parametreleriyle birlikte) bulur.
  2. |: gau'nun çıktısını (yüzlerce URL) httpx'e gönderir.
  3. httpx -silent: Bu URL'lerden sadece "canlı" olanları (200 OK gibi yanıt verenleri) filtreler.
  4. |: Canlı URL listesini dalfox pipe'a gönderir.
  5. dalfox pipe: Standart girdiden gelen her bir URL'yi alır ve XSS için tarar, sonuçları dalfox\_results.txt dosyasına kaydeder.

**Örnek 5: Kimlik Doğrulaması Gerektiren Bir Sayfayı Tarama**

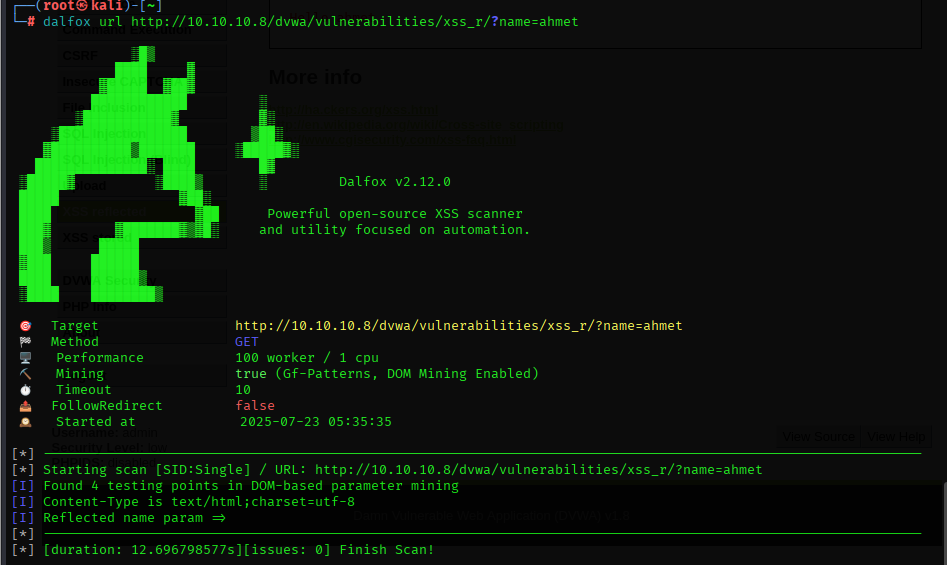
Birçok XSS zafiyeti, sadece giriş yapmış kullanıcıların erişebildiği "hesabım" gibi sayfalarda bulunur.

dalfox url "http://hedefsite.com/profile/settings" --cookie "session=abc123456789def"

* **Açıklama:** --cookie parametresi, DalFox'un her isteği belirtilen oturum çerezi ile birlikte göndermesini sağlar. Bu sayede DalFox, kimlik doğrulaması gerektiren /profile/settings sayfasına erişebilir ve oradaki parametreleri (örneğin, ad, soyad, adres alanları) XSS zafiyeti açısından test edebilir.

**🐺 Örnek: Dalfox ile Otomatik XSS Taraması**

**Dalfox**, XSS zafiyetlerini tespit etmek üzere geliştirilmiş, modern ve otomatik bir açık tarama aracıdır. DOM tabanlı analiz ve parametre madenciliği (parameter mining) gibi gelişmiş teknikleri destekler. Aşağıdaki ekran görüntüsünde, Dalfox’un DVWA uygulaması üzerindeki bir URL'ye uygulandığı görülmektedir.

  
Komut satırında dalfox url http://10.10.10.8/dvwa/vulnerabilities/xss\_r/?name=ahmet komutu çalıştırılarak hedef URL taramaya alınmıştır. Araç; 100 iş parçacığı (worker) ile çalıştırılmış, DOM tabanlı analiz (DOM Mining) etkinleştirilmiştir. Taramada 4 farklı test noktası (parameter) bulunmuş ve name parametresinin yansımalı (reflected) olduğu tespit edilmiştir. Bu, XSS zafiyetinin potansiyel olarak istismar edilebileceği anlamına gelir. Tarama yaklaşık 12.6 saniyede tamamlanmış ve herhangi bir hata olmadan başarıyla sonuçlanmıştır. Dalfox’un bu gibi testlerde otomatik, hızlı ve güvenilir sonuçlar sunması onu XSS taramaları için güçlü bir araç haline getirir.

**Sonuç**

DalFox, XSS zafiyetlerini tespit etme işini bir sonraki seviyeye taşıyan, modern ve vazgeçilmez bir araçtır. Akıllı parametre keşfi, çok yönlü tarama yetenekleri ve otomasyon dostu yapısıyla, hem yeni başlayanlar hem de deneyimli profesyoneller için muazzam bir güç sunar. Bir web uygulamasının güvenliğini sağlamak istiyorsanız, DalFox gibi araçlarla onu bir saldırganın gözünden test etmek en etkili yöntemlerden biridir.

Her zaman olduğu gibi, bu gücü büyük bir sorumlulukla kullanın. DalFox'u **yalnızca yasal olarak test etme izniniz olan sistemlerde** ve etik kurallar çerçevesinde kullanarak siber güvenlik dünyasına olumlu bir katkıda bulunun.