# Testing for Client-side URL Redirect ( İstemci Tarafı URL Yönlendirme Testi)

## Summary (Özet)

Bu bölüm, açık yönlendirme olarak da bilinen istemci tarafı URL yeniden yönlendirmesini nasıl kontrol edeceğinizi açıklar. Bir uygulama, kötü amaçlı olabilecek harici bir URL'ye yol açan bir bağlantıyı belirten kullanıcı kontrollü girdiyi kabul ettiğinde var olan bir giriş doğrulama kusurudur. Bu tür bir güvenlik açığı, bir kimlik avı saldırısını gerçekleştirmek veya bir kurbanı bir enfeksiyon sayfasına yönlendirmek için kullanılabilir.

Bu güvenlik açığı, bir uygulama bir URL değeri içeren ve sterilize etmeyen güvenilmez girdiyi kabul ettiğinde ortaya çıkar. Bu URL değeri, web uygulamasının, saldırgan tarafından kontrol edilen kötü amaçlı bir sayfa gibi kullanıcıyı başka bir sayfaya yönlendirmesine neden olabilir.

Bu güvenlik açığı, bir saldırganın başarılı bir şekilde bir kimlik avı dolandırıcılığı başlatmasını ve kullanıcı kimlik bilgilerini çalmasını sağlayabilir. Yönlendirme gerçek uygulamadan
kaynaklandığından, kimlik avı girişimleri daha güvenilir bir görünüme sahip olabilir.

İşte bir kimlik avı saldırı URL'sinin bir örneği.

`http://www.target.site?#redirect=www.fake-target.site`

Bu URL'yi ziyaret eden kurban otomatik olarak yönlendirilir. `fake-target.site` Bir saldırganın, kurbanın kimlik bilgilerini çalmak için amaçlanan siteye benzeyen sahte bir sayfa yerleştirebileceği yer.

Açık yönlendirme, uygulamanın erişim kontrol kontrol kontrollerini atlayacak ve saldırganı normalde erişemeyecekleri ayrıcalıklı işlevlere iletecek bir URL oluşturmak için de kullanılabilir.

## Test Objectives (Test Hedefleri)

- URL'leri veya yolları işleyen enjeksiyon noktalarını belirleyin.

- Sistemin yönlendirebileceği yerleri değerlendirin.

## How To Test (Nasıl Test Edilir)

Testçiler bu tür bir güvenlik açığını manuel olarak kontrol ettiklerinde, önce istemci tarafı kodunda uygulanan istemci tarafı yönlendirmeleri olup olmadığını belirlerler. Bu yönlendirmeler, JavaScript örneği vermek için uygulanabilir, `window.location` Nesne. Bu, tarayıcıyı sadece bir dize atayarak başka bir sayfaya yönlendirmek için kullanılabilir. Bu aşağıdaki snippet'te gösterilmiştir:

```
var redir = location.hash.substring(1);
if (redir) {
   window.location='http://'+decodeURIComponent(redir);
}
```

Bu örnekte, senaryo değişkenin herhangi bir doğrulamasını yapmaz. `redir` Sorgu dizesi aracılığıyla kullanıcı tarafından sağlanan girişi içeren. Kodlama şekli uygulanmadığı için bu geçerli olmayan girdiye aktarılır `windows.location` Nesne, URL yönlendirmesi güvenlik açığı oluşturmak.

Bu, bir saldırganın kurbanı sadece aşağıdaki sorgu dizesini göndererek kötü niyetli bir siteye yönlendirebileceği anlamına gelir:

`http://www.victim.site/?#www.malicious.site`

Hafif bir modifikasyonla, yukarıdaki örnek snippet JavaScript enjeksiyonuna karşı savunmasız olabilir.

```
var redir = location.hash.substring(1);
if (redir) {
   window.location=decodeURIComponent(redir);
}
```

Bu, aşağıdaki sorgu dizeyi göndererek kullanılabilir:

`http://www.victim.site/?#javascript:alert(document.cookie)`

Bu güvenlik açığını test ederken, bazı karakterlerin farklı tarayıcılar tarafından farklı muamele gördüğünü düşünün. Referans olarak, DOM tabanlı XSS'ye bakın.