Subresource Integrity (SRI)

(-) Birebir uygulanmamıştır.

a. Subresource Integrity Nedir?

Web uygulamaları js kütüphanelerini, css dosyalarını yada resim dosyalarını third party konumlardan dahil edebilmekteler. Örneğin bir web uygulaması jquery kütüphanesini CDN ile uygulamaya dahil edebilir. Fakat şayet CDN sunucusu hack'lenirse saldırgan CDN sunucusundaki javascript dosyalarına kendi javascript kodlarını girebilir ve böylece istemci varsayılan olarak gelen javascript dosyasını alarak cross site scripting saldırısına maruz kalabilir. Başka bir örnek olarak bir web uygulaması css dosyasını CDN ile uygulamaya dahil edebilir. Fakat şayet CDN sunucusu hack'lenirse saldırgan CDN sunucusundaki css dosyalarına kendi css kodlarını girebilir ve böylece istemci varsayılan olarak gelen css dosyasını alarak sayfası komple tahrif edilmiş halde sayfayı görüntüleyebilir. Bu şekilde deface saldırısına maruz kalabilir.

Third party sunuculardа oluşabilecek en ufacık bir güvenlik zafiyetine karşın web geliştiricisinin yapabileceği önlem tarayıcıda çalışacak olan "Subresource integrity" methodunu kullanmaktır. Subresource Integrity methodu ile tarayıcılar üçüncü parti konumlardan gelen dosyaların bütünlüğünün bozulup bozulmadığı test eder. Şayet third party bir içerik manipulasyona uğramışsa tarayıcı içeriği çalıştırılmaz ve böylece olası bir saldırının önüne geçilmiş olur.

Tarayıcının web uygulaması açıkken third party konumlardan gelen dosyaların bütünlüğüne dönük test yapabilmesi için html koduna "integrity" attribute'u eklenmelidir:

```
integrity="[hash algorithm]-[base64 encoded cryptographic hash value]
```

Syntax'a göre integrity keyword'ü önce hash algoritmasının adını, sonra third party içeriğin orijnal halinin hash'ini alır. Örneğin;

```
<script src="https://code.jquery.com/jquery-2.1.4.min.js" integrity="sha384-
R4/ztc4ZlRqWjqIuvf6RX5yb/v90qNGx6fS48N0tRxiGkqveZETq72KgDVJCp2TC"
crossorigin="anonymous"></script>
```

Olası bir CDN sunucusu hack'i karşısında manipule edilmiş javascript dosyasının hash'i ile web uygulamasında mevcut olan hash eşleşmeyeceğinden tarayıcı third party konumdan gelen js kütüphanesini çalıştırmayacaktır ve olası bir saldırının önüne geçilmiş olacaktır.

Not: crossorigin attribute'u "anonymous" değerini aldığında istemci tarafındaki çerez bilgisinin herhangi bir üçüncü parti sunucuya gönderilmesi önlenmiş olur. "use-credentials" değerini aldığında ise üçüncü parti sunuculara gönderilmesinin önü açılmış olur.

Subresource Integrity (SRI) desteği her tarayıcıda yoktur. Bu desteği veren tarayıcılar aşağıdaki adresten öğrenilebilir:

http://caniuse.com/#feat=subresource-integrity

b. Subresource Integrity Methodunun Uygulanması Örneği

Aşağıda subresource integrity uygulanmamış bir kaynak kullanım örneği ile subresource integrity uygulanmış bir kaynak kullanım örneği yer almaktadır:

Subresource Integrity is not implemented

<script src="https://code.jquery.com/jquery-2.1.4.min.js"></script>

Subresource Integrity is implemented.

<script src="https://code.jquery.com/jquery-2.1.4.min.js" integrity="sha384-R4/ztc4ZlRqWjqIuvf6RX5yb/v90qNGx6fS48N0tRxiGkqveZETq72KgDVJCp2TC" crossorigin="anonymous"></script>

Web uygulamasındaki third party kaynaklara Subresource Integrity uygulayarak third party sunucularda doğabilecek zafiyetlere ve o sunuculardan gelebilecek saldırılara karşı korunmuş oluruz.

Kaynaklar

https://www.netsparker.com/blog/web-security/subresource-integrity-SRI-security/

https://developer.mozilla.org/tr/docs/Web/HTML/CORS_settings_attributes

http://caniuse.com/#feat=subresource-integrity