Spring Security ve HTTP Response Headers

Security HTTP Response Headers

- Http response içeriğine eklenen bir takım **güvenlikle ilgili header**'lar vardır
- Spring Security bu header'ların **response'a kolaylıkla eklenmesi**ni sağlar
- Bu security header'ları istemci/tarayıcı tarafından **process ediliyorsa anlamlı**dır

```
<security:http>
   <security:headers/>
</security:http>
```

Default eklenen header'lar:

- Cache Control
- Content Type Options
- HTTP Strict Transport Security
- X-Frame-Options
- X-XSS-Protection

Security HTTP Response Headers

- Her bir header aşağıdaki gibi **ayrı ayrı** da kontrol edilebilir

```
<security:http>
    <security:headers>
        <security:cache-control />
        <security:content-type-options />
        <security:hsts />
        <security:frame-options />
        <security:xss-protection />
    </security:headers>
</security:http>
```

Security HTTP Response Headers

- **Cache Control**
  - Modern tarayıcılar artık **secure bağlantı**da elde edilen **içeriğin cache'lenmesini** de desteklemektedir
  - Bu bir güvenlik açığı yaratabilir
  - Bu header güvenliğe tabi içeriğin tarayıcı tarafından cache'lenmesini önler

```
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
```

Security HTTP Response Headers

- **Content Type Options**
  - Tarayıcılar **content sniffing** yaparak content tipi belirtilmeyen response'un content tipini tespit etmeye çalışırlar
  - Bu durum **birden fazla content tipi**ne ait içerik barındıran dosyalarda güvenlik açığı oluşturabilir
  - Örneğin bir postscript dokümanın içerisinde aynı zamanda XSS saldırısında bulunan bir javascript kodu da yer alabilir

```
X-Content-Type-Options: nosniff
```

Security HTTP Response Headers

- **HTTP Strict Transport Security (HSTS)**
  - HTTPS ile erişilmesi gereken bir siteye hiçbir şekilde HTTP ile erişilmemelidir
  - Aksi durum potansiyel olarak **man in the middle saldırısı**na kapı aralanabilir
  - Örneğin bir saldırgan ilk request'i intercept edip asıl site yerine kendi sitesine yönlendirme yapabilir
  - HSTS domain, tarayıcıya ilk andan itibaren HTTP yazılsa bile siteye HTTPS ile erişmesi gerektiğini söyler

Security HTTP Response Headers

- **HTTP Strict Transport Security (HSTS)**
  - Yöntemlerden biri, bir hostu preload ederek tarayıcıya HSTS domain olduğunu söylemektir
  - Diğer bir yöntem ise strict transport security header'ı kullanarak bir domain'i belirli bir süre HSTS olarak kabul etmesini sağlamaktır

```
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
```

Security HTTP Response Headers

- **X-Frame-Options**
  - Bir site'nin **frame içerisine gömülmesi**ne izin vermek security açıklarına yol açabilir
  - Örneğin bir CSS yanıltmacası ile kullanıcı sizin uygulamanızda bir butona tıkladığını düşünürken aslında saldırgana erişim izni veriyor olabilir
  - Bu tür saldırılara **clickjacking** adı verilir
  - Bu header ile tarayıcı, sitenin frame içerisinde render edilmesine izin vermez

```
X-Frame-Options: DENY
```

Security HTTP Response Headers

- **X-XSS-Protection**
  - Bazı tarayıcılar reflected xss saldırılarını filtreleme kabiliyetine sahiptir
  - XSS saldırısı tespit edildiği vakit bu içerik response'dan çıkarılır
  - Bu durum güvenlik açığına yol açabilir
  - Spring Security'nin tercihi filtreleme yerine içeriği tamamen bloklamaktır

```
X-XSS-Protection: 1; mode=block
```

Security HTTP Response Headers

- **Custom header** yazmak da mümkündür

```
<security:http>
    <security:headers>
        <security:header name="X-Content-Security-Policy"
value="default-src 'self'" />
        <security:header name="X-WebKit-CSP" value="default-
src 'self'" />
    </headers>
</security:http>
```

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com