Cookie Not Marked as Secure , Cookie Not Marked as HttpOnly and Same-Site Cookie Not Implemented

Cookie Not Marked as Secure, Cookie Not Marked as HttpOnly ve SameSite Cookie Not Implemented kullanıcı çerezlerinin secure, httponly ve samesite flag'leri ile koruma altına alınmayışını ifade etmektedir. Secure, HttpOnly ve SameSite flag'leri birer Set-cookie başlığı anahtar kelimeleridir.

a. Set-Cookie Başlığı Nedir?

Set-cookie kullanıcı çerezlerini sunucudan kullanıcıya göndermek için kullanılan bir http response başlığıdır. Syntax'ı ise şu şekildedir:

> Set-Cookie: <key>=<value>; Expires=<expiryDate>; Secure; HttpOnly; SameSite="..."

Yukarıda önce çerez değişken ve değeri, sonra çerezin kullanım ömrü ve son olarak da çerezi denetim altına alan bayraklar listelenmiştir. Bayrakların açıklaması şu şekildedir:

Secure Flag

> Secure flag'i ile işaretlenmiş çerezler yalnızca https trafiği olduğunda istemci ve sunucu arasında gidip gelir. Eğer istemci ve sunucu arasında iletişim http 'ye dönerse çerez gönderimi engellenir. Böylece web uygulamasını kullanan kullanıcı çerezlerinin MITM yapan kişilerce okunabilmesinin önüne geçilmiş olur. Örneğin; https web uygulamalarda http üzerinden erişilir kılınmış kaynaklarda (javascript, css, v.b.) trafiğin http üzerinden akması sırasında çerezin gönderilmesini ve böylece aradaki saldırganın çerezi almasını önler.

HttpOnly Flag

> HttpOnly flag'i ile web uygulamasında istemci tarafındaki javascript kodlarının kullanıcı çerezine erişimi engellenir. Böylece ileride çıkabilecek olası xss zafiyetlerine karşı çerezlerin üçüncü parti konumlara gidişi engellenmiş olur.

SameSite Flag

> SameSite flag'i third party çerezlerin üçüncü taraf konumlara gidişini denetler. Mod olarak Lax ve Strict değerlerini alabilir. Lax gevşek moddur. Üçüncü taraf yerlere çerezin gidişine izin verir. Strict katı moddur. Üçüncü taraf yerlere çerezin gidişine izin vermez.
>
> SameSite flag'i çerezlerin çalınmasını önlemekten ziyade saldırganların kullanıcıları çerezleri vasıtasıyla istemediği bir eyleme zorlamasını önlemeye yarar. Yani SameSite bayrağı csrf saldırılarını önlemektedir. Ayrıntılı bilgi için bkz. Paketleme İçin Gözden Geçirilecekler / İnternetten Edinilmiş Kıymetli Bilgiler / CSRF Nedir ve Nasıl SameSite ile Tamamen Önlenir.docx

b. Cookie Not Marked as Secure Zafiyeti Nasıl Kapatılır? [High]

Cookie Not Marked as Secure zafiyeti Set-cookie başlığı ve Secure bayrağı ile kapatılılır. Bu başlığı ve bayrağını eklemek için sunucularda aşağıdakiler uygulanır.

```
# Apache sunucular için apache2.conf dosyası açılır ve aşağıdaki satır eklenir.
Header set Set-Cookie ^(.*)$ "$1; Secure"

# IIS sunucular için web.config dosyası açılır ve <system.web> tag'ı içerisinde yer alan
# <httpCookies> tag'ı içerisindeki requireSSL attribute'u aşağıdaki gibi true yapılır.
<httpCookies httpOnlyCookies="false" requireSSL="true">


# Nginx sunucular için ssl.conf ya da default.conf dosyası açılır ve aşağıdaki satıra
# Secure flag'i gösterildiği gibi eklenir.
proxy_cookie_path / "/; Secure";
```

Sunucular restart'lanır. Böylece http response başlıklarına set-cookie başlığı ve flag'leri eklenmiş olur.

c. Cookie Not Marked as HttpOnly Zafiyeti Nasıl Kapatılır? [Low]

Cookie Not Marked as HttpOnly zafiyeti Set-cookie başlığı ve HttpOnly bayrağı ile kapatılılır. Bu başlığı ve bayrağını eklemek için sunucularda aşağıdakiler uygulanır.

```
# Apache sunucular için apache2.conf dosyası açılır ve aşağıdaki satır eklenir.
Header set Set-Cookie ^(.*)$ "$1; HttpOnly"

# IIS sunucular için web.config dosyası açılır ve <system.web> tag'ı içerisindeki
# <httpCookies> tag'ı attribute'ları aşağıdaki gibi true yapılır.
<httpCookies httpOnlyCookies="true" requireSSL="false">


# Nginx sunucular için ssl.conf ya da default.conf dosyası açılır ve aşağıdaki satıra httpOnly
# flag'i gösterildiği gibi eklenir.
proxy_cookie_path / "/; HTTPOnly";
```

Sunucular restart'lanır. Böylece http response başlıklarına set-cookie başlığı ve flag'leri eklenmiş olur.

d. SameSite is Not Implemented Zafiyeti Nasıl Kapatılır? [Low]

SameSite is Not Implemented zafiyeti Set-cookie başlığı ve SameSite bayrağı ile kapatılılır. Bu başlığı ve bayrağını eklemek için sunucularda aşağıdakiler uygulanır.

```
# Apache sunucular için apache2.conf dosyası açılır ve aşağıdaki satır eklenir.
Header set Set-Cookie ^(.*)$ "$1; SameSite=strict"

# IIS 7 ve üzeri olan ve ayrıca "Url Rewrite" modülü (tool'u) yüklü olan IIS sunucular için
# web.config dosyası açılır ve <system.web> tag'ı içerisine aşağıdaki kod satırları eklenir.
<rewrite>
  <outboundRules>
    <rule name="Add SameSite" preCondition="No SameSite">
      <match serverVariable="RESPONSE_Set_Cookie" pattern=".*" negate="false" />
      <action type="Rewrite" value="{R:0}; SameSite=strict" />
      <conditions>
      </conditions>
    </rule>
    <preConditions>
      <preCondition name="No SameSite">
        <add input="{RESPONSE_Set_Cookie}" pattern="." />
        <add input="{RESPONSE_Set_Cookie}" pattern="; SameSite=strict" negate="true" />
      </preCondition>
    </preConditions>
  </outboundRules>
</rewrite>

# Nginx sunucular için ssl.conf ya da default.conf dosyası açılır ve aşağıdaki satıra SameSite
# flag gösterildiği gibi eklenir.
proxy_cookie_path / "/; SameSite=strict";
```

d. HttpOnly + Secure + SameSite Flag'leri ile Çerez Güvenliği Tam Anlamıyla Nasıl Sağlanır?

Tam çerez güvenliği hem httpOnly hem Secure hem de Samesite flag'lerini beraber kullanarak sağlanır.

```
Apache'de HttpOnly , Secure ve SameSite Flag
# ================================
# Apache sunucularda HttpOnly, Secure ve SameSite bayrakları için apache2.conf dosyası
# açılır ve aşağıdaki satır eklenir.
Header set Set-Cookie ^(.*)$ "$1; SameSite=strict;HttpOnly;Secure"

IIS'de HttpOnly , Secure ve SameSite Flag
# ================================
# IIS sunucularda HttpOnly ve Secure bayrakları için web.config dosyası açılır ve
# <system.web> tag'ı içerisindeki <httpCookies> tag'ı attribute'ları aşağıdaki gibi true
# yapılır.
<httpCookies httpOnlyCookies="true" requireSSL="true">
```

```
# IIS sunuculardа SameSite bayrağı için sistemin IIS 7 ve üzeri olması, ASP.NET rolünün
# enable olması ve ayrıca "Url Rewrite" modülünün (tool'unun) sistemde yüklü olması
# gerekmektedir. Bu koşullar sağlandığı takdirde web.config dosyası açılır ve <system.web>
# tag'ı içerisine aşağıdaki kod satırları eklenir.
<rewrite>
  <outboundRules>
    <rule name="Add SameSite" preCondition="No SameSite">
      <match serverVariable="RESPONSE_Set_Cookie" pattern=".*" negate="false" />
      <action type="Rewrite" value="{R:0}; SameSite=strict" />
      <conditions>
      </conditions>
    </rule>
    <preConditions>
      <preCondition name="No SameSite">
        <add input="{RESPONSE_Set_Cookie}" pattern="." />
        <add input="{RESPONSE_Set_Cookie}" pattern="; SameSite=strict" negate="true" />
      </preCondition>
    </preConditions>
  </outboundRules>
</rewrite>
# Not: IIS sunucusu versiyon 7 ve üzeri değilse, ASP.NET rolü enable edilmemişse ve Url
# Rewrite yazılımı sisteme yüklenmemişse bu eklenen kodlar çalışmayacaktır.
```

Nginx'de HttpOnly, Secure ve SameSite Flag

```
# ================================
# Nginx sunucular için ssl.conf ya da default.conf dosyası açılır ve aşağıdaki satıra httpOnly,
# Secure ve SameSite flag'i gösterildiği gibi eklenir.
proxy_cookie_path / "/; HttpOnly; secure; SameSite=strict";
```

Sunucular restart'lanır. Böylece http response başlıklarına set-cookie başlığı ve flag'leri eklenmiş olur.

Kaynaklar

https://blog.appcanary.com/2017/http-security-headers.html

https://geekflare.com/httponly-secure-cookie-apache/

https://msdn.microsoft.com/en-us/library/ms228262(v=VS.80).aspx

https://geekflare.com/httponly-secure-cookie-nginx/

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie

https://security.stackexchange.com/questions/153835/is-samesite-attribute-redundant-on-httponly-cookie

https://serverfault.com/questions/849888/add-samesite-to-cookies-using-nginx-as-reverse-proxy

https://stackoverflow.com/questions/38954821/preventing-csrf-with-the-same-site-cookie-attribute

https://docs.microsoft.com/en-us/iis/extensions/url-rewrite-module/creating-rewrite-rules-for-the-url-rewrite-module