1.1.1 Çerez İçeriğinin Korunaksız Olması (Insecure Cookie) (CWE-614)

Açıklık Önem Derecesi: Orta

Açıklığın Etkisi: Oturum Bilgisinin Ele Geçirilmesi

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Web uygulamalarının kullanıcılara dağıttıkları çerezlerde mutlaka HttpOnly bayrağı yer almalıdır. Bu bayrak web uygulamalarında istemci tarafındaki javascript kodlarının kullanıcı çerezlerine erişimini engellemektedir. Böylece web uygulamalarında ileride çıkabilecek olası XSS zafiyetlerine karşı kullanıcı çerezlerinin üçüncü taraf konumlara (saldırganların sunucularına) gidişinin önü kesilmiş olur.

Web uygulamalarının kullanıcılara dağıttıkları çerezlerde ikinci olarak eğer uygulama SSL/TLS üzerinden çalışıyor ise "Secure" bayrağı yer almalıdır. Bu bayrak çerezin sadece SSL/TLS ile şifrelenmiş taleplerin içerisinde gitmesini sağlar. Örneğin kullanıcı arp zehirleme gibi araya giren adam saldırılarına maruz kalırsa uygulama trafiği https yerine http üzerinden aktığı sıralarda araya girme saldırısı yapan saldırganlar kullanıcının çerezini açık bir şekilde okuyabilirler. Fakat web uygulamasında çerezlerde "Secure" bayrağı kullanılırsa kullanıcı uygulama trafiği http üzerinden aktığı sıralarda çerezini web sunucusuna sadece https üzerinden trafik akarken göndereceğinden göndermeyecektir ve aradaki adam (saldırgan) bu sebeple çerezi elde edemeyecektir. Böylece kullanıcı çerezleri araya giren adam saldırılarından korunmuş olacaktır.

Web uygulamalarının kullanıcılara dağıttıkları çerezlerde üçüncü olarak "SameSite" bayrağı yer almalıdır. Bu bayrak kullanıcıların Siteler Arası Talep Sahtekarlığı saldırılarına maruz kaldığı vak'alar meydana geldiğinde fark etmeden web uygulama üzerinde işlemler gerçekleştirmelerini önler. Yani çerezin saldırganlarca sunulan sahte taleplerde (yan sekmelerden) kullanımını engeller.

Not:

CSRF (Siteler Arası Talep Sahtekarlığı) saldırılarına önlem olarak sunulan CSRF jetonları SameSite bayrağı önleminin bir alternatifi konumunda değildirler. Ayrı bir kademede güvenlik önlemidir. Dolayısıyla her iki ayrı kademede yer alan önlemi

uygulamak en güvenli kullanımdır.

Kurum web uygulamasında [[[[[[[[[[[[[[[HttpOnly]]]]]]]]]]]]]]] ve [[[[[[[[[[[[[[[Secure]]]]]]]]]]]]]]] ve [[[[SameSite]]]]]] bayraklarının kullanılmadığı tespit edilmiştir:

:::::::::BULGU:::::::::

Açıklığın Önlemi:

a) Apache Sunucularda Çerezlere HttpOnly, Secure ve SameSite Ekleme

Apache sunucularda çerezlere HttpOnly, Secure ve SameSite bayraklarını eklemek için sistem türüne göre httpd.conf ya da apache2.conf yapılandırma dosyası açılmalıdır:

Terminal:

```
> sudo su
> nano /etc/apache2/apache2.conf
```

Ardından yapılandırma dosyasının en altına aşağıdaki satır eklenmelidir:

```
Header edit Set-Cookie ^(.*)$ "$1; HttpOnly; Secure; SameSite=Strict"
```

Son olarak yapılandırma dosyası kaydedilmelidir ve apache servisi yeniden başlatılmalıdır:

Terminal:

```
> service apache2 restart
```

b) IIS Sunucularda Çerezlere HttpOnly, Secure ve SameSite Ekleme

IIS sunucularda ise web.config adlı yapılandırma dosyası açılmalıdır ve system.web etiketi içerisindeki httpCookies etiketi özellikleri "true" yapılmalıdır.

```
<httpCookies httpOnlyCookies="true" requireSSL="true">
```

IIS sunucularda SameSite bayrağı için sistemin IIS 7 ve üzeri olması, ASP.NET rolünün aktif (enable) olması ve ayrıca "Url Rewrite" modülünün (tool'unun) sistemde yüklü olması

gerekmektedir. Bu koşullar sağlandığı takdirde web.config dosyası açılmalıdır ve system.web etiketi içerisine şu kod satırları eklenmelidir.

```
<!--
    SameSite Bayrağı Ekleme

    IIS sunucusu versiyon 7 ve üzeri değilse,
    ASP.NET rolü enable edilmemişse ve Url
    Rewrite yazılımı sisteme yüklenmemişse
    bu eklenen kodlar çalışmayacaktır.
-->

<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>
```

IIS ardından servisi yeniden başlatılmalıdır.

c) Nginx Sunucularda Çerezlere HttpOnly, Secure ve SameSite Ekleme

Nginx sunucularda aynı işlemi yapmak için ise ssl.conf ya da default.conf dosyası açılmalıdır ve dosyada yer alan proxy_cookie_path satırına bayraklar aşağıdaki gibi eklenmelidir:

```
proxy_cookie_path / "/; HttpOnly; secure; SameSite=Strict";
```

Ardından nginx servisi yeniden başlatılmalıdır.

Not:

Eğer uygulama SSL/TLS kullanmıyorsa sadece HttpOnly ve SameSite bayraklarını eklemek için

Apache sunucularda belirtilen satır

```
Header edit Set-Cookie ^(.*)$ "$1; HttpOnly; SameSite=Strict"
```

IIS sunucularda belirtilen satır

```
<httpCookies httpOnlyCookies="true" requireSSL="false">
```

ve SameSite için IIS rewrite kuralı web.config'de aynı şekilde,

Nginx sunucularda ise belirtilen satır

```
proxy_cookie_path / "/; HttpOnly; SameSite=Strict";
```

kullanılmalıdır.

Referanslar:

1. http://www.php.net/manual/en/session.configuration.php
2. https://geekflare.com/httponly-secure-cookie-apache/
3. https://msdn.microsoft.com/en-us/library/ms228262(v=VS.80).aspx
4. https://geekflare.com/httponly-secure-cookie-nginx/
5. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie
6. https://security.stackexchange.com/questions/153835/is-samesite-attribute-redundant-on-httponlycookie
7. https://serverfault.com/questions/849888/add-samesite-to-cookies-using-nginx-as-reverse-proxy
8. https://stackoverflow.com/questions/38954821/preventing-csrf-with-the-same-site-cookie-attribute
9. https://docs.microsoft.com/en-us/iis/extensions/url-rewrite-module/creating-rewrite-rules-for-the-urlrewrite-module
10. https://www.whitehatsec.com/blog/session-cookie-secure-flag-java/
11. https://www.owasp.org/index.php/SecureFlag
12. https://geekflare.com/secure-cookie-flag-in-tomcat/

13. https://tomcat.apache.org/tomcat-8.5-doc/ssl-howto.html
14. https://docs.microsoft.com/tr-tr/aspnet/samesite/system-web-samesite
15. https://www.sjoerdlangkemper.nl/2016/04/14/preventing-csrf-with-samesite-cookie-attribute/
16. http://whatis.techtarget.com/definition/third-party-cookie
17. http://www.ravelrumba.com/blog/third-party-cookies/
18. https://blog.appcanary.com/2017/http-security-headers.html
19. https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/cross-site-request-forgery-inlogin-form/
20. https://wiki.owasp.org/index.php/Testing_for_cookies_attributes_(OTG-SESS-002)