1.1.1 HSTS Kullanımı Eksikliği (Missing HSTS Header) (CWE-346)

Açıklık Önem Derecesi: Orta

Açıklığın Etkisi: SSLStrip saldırılarına karşı savunmasız kalma

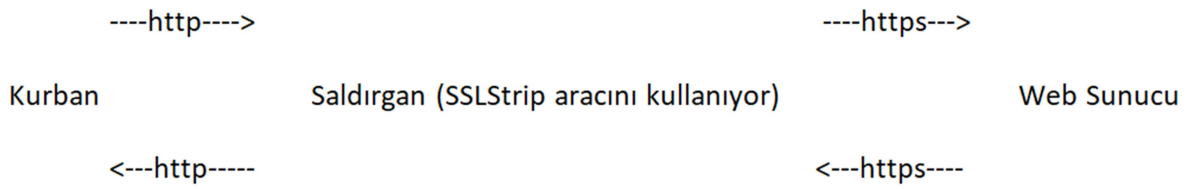
Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

SSLStrip, diğer adıyla HTTPS-den-Http-ye-İndirgeme kullanıcıların web uygulamasını https üzerinden kullanması gerekirken http üzerinden kullanmasına sebep olan bir saldırı türüdür. Saldırı kabaca yerel ağdaki bir kullanıcı ile web sunucu arasına yerel ağdaki saldırganın MITM (araya giren adam saldırısı) yaparak girmesiyle gerçekleşir. Normalde kullanıcılar bir websitesine bağlanmak istediklerinde genellikle adres çubuğuna web sitesinin adresini https ön ekini eklemeden girerler. Örn; www.deneme.com gibi. Bunun sonucunda aradaki adam kullanıcının bağlantı talebini alır, https yapar ve ilgili web sunucuya öyle gönderir. Saldırgan bu sırada kullanıcının paketini şifresiz aldığından içindeki bilgileri okuyabilirdir. Saldırgan aldığı bağlantı paketini sunucuya https yaparak gönderdikten sonra sunucu https olarak yanıtı saldırgana gönderir. Saldırgan gelen https yanıtını http yaparak kullanıcıya gönderir ve böylece kullanıcı halen http üzerinden web uygulamasını kullanır durumda kalır. Saldırgan kullanıcıdan gelen her http paketini okur, https yapar ve web sunucuya gönderir. Web sunucudan gelen her https yanıtını ise http yapar ve kurbana gönderir. Web sunucusu saldırganla arasında olan https trafiği dolayısıyla bir problem görmez ve bunun sonucunda saldırgan kullanıcıdan gelen her http paketini okuyarak kullanıcı adı , şifre, banka kart numarası , son kullanım tarihi , CVV kodu ve TC Kimlik numarası gibi birçok kritik bilgilere sahip olabilir hale gelir.

Bu saldırının gerçekleşmesini önlemek için web sunucusunu sadece HTTPS kullanacak şekilde yapılandırmak yetmemektedir. Çünkü zaten iletişim önceki senaryoda bahsedildiği gibi yerel ağda aradaki adam olan saldırgan ve web sunucu arasında HTTPS üzerinden gerçekleşmektedir. Şekil 1'de saldırının çalışma şekli gösterilmiştir.



Şekil 17. Saldırının Anatomisi

Burada kullanıcıyı HTTPS-den-HTTP-ye-İndirgeme saldırısından korumak için kullanıcı tarayıcılarını, http olarak adres çubuğuna girilen web uygulama adreslerini https olarak düzeltmeye ve o şekilde bağlantı talebinde bulunmaya zorlamak gerekir. Böylelikle yerel ağda kullanıcı ve web sunucu arasına giren saldırgan kullanıcıdan gelen https bağlantı talebini şifreli olarak alacağından kullanıcı verilerini göremez, elde edemez ve değiştiremez.

SSLStrip (HTTPS-to-HTTP-Downgrading) saldırıları hem HTTP ve HTTPS'i beraber kullanan web uygulamalarını hem de sadece HTTPS kullanan web uygulamalarını etkileyen bir saldırı türüdür. Çünkü bu saldırı türü web sunucunun davranışına (sunduğu hizmete) göre (yani http ve https'i beraber kullanmasına ya da yalnızca https kullanmasına göre) değişen bir saldırı türü değildir. Bu saldırı türü istemcinin davranışına göre değişen bir saldırı türüdür. Eğer istemci HTTPS yerine HTTP üzerinden bağlantı talebinde bulunursa web sunucu ister sadece HTTPS kullansın ister HTTP ve HTTPS'i beraber kullansın aradaki saldırgan aldığı paketi https yaparak göndereceğinden web sunucu tarafında fark eden bir şey olmayacaktır ve saldırgan, kullanıcıdan paketleri http olarak alarak saldırısını sürdürebilecektir. Teknik olarak sadece HTTPS kullanan web uygulamaları HTTP ve HTTPS'i birlikte kullanan web uygulamalarına göre daha az risk altındadırlar. Çünkü sadece HTTPS kullanan web uygulamalarında sadece ilk isteğin http üzerinden olması ihtimali söz konusudur. Ancak bu, SSLStrip adı verilen saldırı yöntemine karşı sadece HTTPS kullanan web uygulamalarını yine de savunmasız kılar.

Sanılanın aksine sadece sunucu tarafta https yönlendirmesi yetmez, istemci tarafta da https yönlendirmesi şarttır. Aksi takdirde araya giren saldırganlar yanlışlıkla gelen http taleplerini sslstrip gibi araçlarla avlayabilirler ve kullanıcıları http üzerinden haberleştirerek trafiklerindeki hassas bilgileri elde edebilirler. Bu nedenle kullanıcı web tarayıcılarına HTTPS kullanan web uygulamalarına HTTP üzerinden bağlanma girişimlerinde HTTPS üzerinden bağlan direktifi (HSTS yanıt başlığı) verilmesi gerekmektedir.

Kurum uygulamasının kullanıcı web tarayıcılarına https üzerinden bağlan direktifi veren HSTS başlığı gönderimini yapmadığı tespit edilmiştir.

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

(Örnek Bulgu)

(Spring Framework configure metodu)

(HSTS enable eden kod satırı bulunmamakta)

Şekil XXX. Strict-Transport-Security Eksikliği

Açıklığın Önlemi:

C# uygulamalarda HSTS aktifleştirme güvensiz yapılandırma ve güvenli yapılandırma şu şekildedir:

C#:

```
// GÜVENSİZ YAPILANDIRMA

// Calling IApplicationBuilder.UseHsts() with
// Default Configuration Provides Insufficient
// Defense

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    app.UseHsts(); // Bu satırdaki kullanım ile varsayılan olarak
                   // 30 gün süreli olur ki bu yetersizdir, ayrıca
                   // IncludeSubdomians direktifini aktifleştirmez.

    /* Var olan ek ayarlar burada yer alır*/
}
```

C#:

```
// GÜVENLİ YAPILANDIRMA

// Properly Configuring HSTS via ConfigureServices,
// and Invoking It with IApplicationBuilder.UseHsts()

public void ConfigureServices(IServiceCollection services)
{
    services.AddHsts(options =>
    {
        options.Preload = true;
        options.IncludeSubDomains = true;           // Tüm Subdomain'lerde
                                                    // hsts etkinleşir.
        options.MaxAge = TimeSpan.FromDays(365);    // 1 Yıl süreli
    });

    /* Mevcut ek ayarlarınız burada yer alır */
}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    app.UseHsts(); // ConfigureServices'deki ayarlar ile güvenli çalıştırılır.

    /* Mevcut ek ayarlarınız burada yer alır */
}
```

Alternatif olarak hsts yapılandırma dosyasından etkinleştirilebilir.

XML: // NOT: IIS 10, versiyon 1709 ve sonrası için bu yöntem uygulanabilir.

```
<!-- GÜVENLİ YAPILANDIRMA -->

<!-- Setting HSTS Header in Application Configuration File -->

<site name="siteName" id="1">
    <application ...>
        <!-- application tags -->
    </application>
    <bindings>
        <!-- port binding -->
    </bindings>
    <hsts enabled="true" max-age="31536000" includeSubDomains="true"
redirectHttpToHttps="true" />
</site>
```

Alternatif olarak kod içerisinde manuel ekleme metodu ile hsts uygulanabilir:

C#:

```
// GÜVENLİ YAPILANDIRMA

// Manuel Olarak Kodda HSTS Ekleme

Response.AddHeader("Strict-Transport-Security", "max-age=31536000;
includeSubDomains");
```

PHP uygulamalarda HSTS aktifleştirme şu şekilde yapılabilir.

```
// GÜVENLİ YAPILANDIRMA

// Manuel Olarak Kodda HSTS Ekleme

header("Strict-Transport-Security: max-age=31536000; includeSubDomains");
```

Java uygulamalarda - Spring Framework'ünde - HSTS aktifleştirme ayarı şu alternatif yöntemlerden biri ile yapılabilir:

Java

```
// Setting an HSTS Header in an HTTP Response
response.setHeader("Strict-Transport-Security", "max-age=31536000;
includeSubDomains");
```

XML

```
// Spring - Setting HSTS via Configuration Web.Xml
<http>
    <!-- This is a default value -->
    <headers>
        <hsts
            include-subdomains="true"
            max-age-seconds="31536000" />
    </headers>
</http>
```

JBoss

```
// JBoss - Setting HSTS via Configuration Web.Xml
<system.webServer>
    <httpProtocol>
        <customHeaders>
            <add name="Strict-Transport-Security" value="max-age=31536000;
includeSubDomains"/>
        </customHeaders>
    </httpProtocol>
</system.webServer>
```

Tomcat

```
// Tomcat - Enable Header Security in Web.Xml
<filter>
    <filter-name>httpHeaderSecurity</filter-name>
    <filter-
class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-class>
    <init-param>
        <param-name>hstsMaxAgeSeconds</param-name>
        <param-value>31536000</param-value>
    </init-param>
    <init-param>
        <param-name>includeSubDomains</param-name>
        <param-value>true</param-value>
    </init-param>
</filter>
```

Referanslar:

1. Web Penetration Testing in Kali Linux, sayfa 122-127
2. https://blog.appcanary.com/2017/http-security-headers.html#hsts
3. https://www.tbs-certificates.co.uk/FAQ/en/hsts-iis.html
4. https://hstspreload.org/
5. https://security.stackexchange.com/questions/64979/mitigating-sslstrip-by-only-serving-a-site-over-https
6. http://sectools.org/tool/sslstrip/
7. https://www.cyberciti.biz/faq/nginx-send-custom-http-headers/
8. https://geekflare.com/tomcat-http-security-header/
9. https://docs.spring.io/spring-security/site/docs/current/reference/html/headers.html
10. https://spring.io/guides/gs/securing-web/
11. https://spring.io/blog/2013/08/23/spring-security-3-2-0-rc1-highlights-security-headers
12. https://www.dailyrazor.com/blog/glassfish-vs-tomcat/
13. http://www.edu4java.com/en/servlet/servlet1.html
14. https://learn.microsoft.com/en-us/iis/configuration/system.applicationhost/sites/sitedefaults/hsts
15. https://learn.microsoft.com/en-us/iis/get-started/whats-new-in-iis-10-version-1709/iis-10-version-1709-hsts