1.1.1 XSS (Content-Sniffing) Saldırılarına Karşı Koruma Eksikliği (Missing X-Content-Type-Options Header) (CWE-16)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Hassas bilgilere yetkisiz erişim, Uzaktan kod çalıştırma

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Web tarayıcılar web sunuculardan gelen yanıt paketlerini okurlarken yanıt paketlerindeki Content-Type başlığına bakarlar ve ona göre paket içeriğini okuma / yorumlama ve ekrana sunma uygularlar. Fakat yanıt paketlerinde Content-Type başlığı bozuk değerde olursa, Content-Type başlığı paket içeriğiyle uyuşmazsa veya Content-Type başlığı hiç yer almazsa yanıt paketinin doğru okunabilmesi / yorumlanabilmesi ve ekrana sunulabilmesi amacıyla eski web tarayıcı yazılımları (özellikle Internet Explorer'lar) yanıt paketleri içerisindeki içerik üzerinde Content Sniffing prosedürünü uygularlar. Bu şekilde gelen paket içerisindeki içeriğin MIME türünü saptarlar ve bu bilgi doğrultusunda paketi okuma / yorumlama yapıp istemciye arayüzde sunarlar. Yani yanıt paketinde Content-Type referansı olmadan / kullanılmadan uygulama sayfası doğru bir şekilde görüntülenebilir olur.

Content-Type başlığının bozuk değerde olması, paket içeriğiyle uyuşmaz değerde olması veya hiç olmaması / eksik olması eski web tarayıcıların yanıt paketlerini ekrana sunması noktasında Content-Sniffing özelliğinin çalışmasını tetikler. Böylece paket Content-Type bilgisi paketin gövdesindeki veride Content Sniff'leme yapılmak suretiyle elde edilir ve paket okuması / yorumlaması buna göre yapılarak ekrana sunma gerçekleşir.

Bilgi:

Eski IE web tarayıcılar Content-Type yanıt başlığı doğru formatta olsa da veya doğru veri türünü gösterse de her halükarda Content Sniffing yapmaktadırlar ve Content Sniffing ile belirlenen veri türüne göre paket okuması / yorumlaması yapmaktadırlar. Content Sniffing özelliğini bu şekilde kullandıkları için XSS (Content - Sniffing) zafiyetinin sömürülmesi noktasında IE web tarayıcılar öndedirler.

Web tarayıcılardaki Content Sniffing özelliği web uygulamalara bir esneklik sunmak amacıyla tasarlanmıştır. Bu sayede web uygulamalarda paket türü bilgisi bozuk değerde olduğunda, paket gövdesindeki dökümanla aynı veri türünde olmadığında veya eksik olduğu durumlarda tarayıcılar yine de tür bilgisini kendileri saptayıp görüntüleme sunabilmekteler. Fakat bu aynı zamanda bir güvenlik riski teşkil etmektedir.

Örneğin bir web uygulamada girdi noktasına girilen geçersiz girdi durumunda json dosya yanıtıyla hata mesajı dönüldüğünde hataya sebep olan girdinin json yanıtında yer alması durumunda bu hata yanıtı, yani json dosyası kullanıcılarca ve/veya sistem yöneticisince eski web tarayıcılarda görüntülendiğinde içindeki xss payload'ları çalışabilir ve xss saldırısı yaşanabilir.

Örneğin bir web uygulamaya dosya yükleme mekanizması ile xss payload'ları içeren txt dosyası yüklenebilir. Sadece txt dosyası kabul eden dosya yükleme mekanizması ile bu şekilde sadece zararsız dosya alınıyor şeklinde düşünülebilir. Fakat txt dosyası kullanıcılarca ve/veya sistem yöneticisince eski web tarayıcılarda görüntülendiğinde content sniffing nedeniyle içindeki xss payload'ları çalışabilir ve xss saldırısı yaşanabilir.

XSS (Content-Sniffing) saldırısı kötü niyetli kullanıcının girdi olarak verdiği xss zararlı girdisinin geri sunulacağı sayfada normalde string olarak yansıması beklenirken web tarayıcının yanıt paketinde Content Sniff'leme yapması ve paketin türü bilgisini kullanıcı girdisi türü bilgisine göre değerlendirip ona göre paketi okuması / yorumlaması ile sunması yoluyla xss zararlı girdisinin çalışır halde tarayıcıya yansıtılmasına denir. XSS (Content-Sniffing) zafiyeti Content Sniffing yapma prosedürüne sahip web tarayıcılarda meydana gelir.

> Not:
>
> Tarayıcılarda Content Sniff'leme özelliği spesifik anlarda tetiklenir. Bunun için bir http spesifikasyonu vardır.

Sonuç olarak eski web tarayıcılardaki Content-Sniffing özelliği XSS saldırılarına imkan tanımaktadır. Eski web tarayıcılarda Content Sniffing yoluyla normalde çalıştırılabilir olmaması gereken unsurların çalıştırılabilir hale geçmesi XSS için elverişli bir yol açar. Örneğin txt dosyasındaki xss zararlı girdilerinin çalışması gibi veya json dosyasındaki xss zararlı girdilerinin çalışması gibi.

Kurum web uygulamasında Content-Sniffing özellikli web tarayıcı kullanan kullanıcıların ve kurum web uygulamasının kendisinin XSS (Content-Sniffing) saldırılarından korunmadığı tespit edilmiştir:

:::::BULGU:::::

Açıklığın Önlemi:

a) IIS Web Sunucular

IIS sunucularda konfigürasyon dosyası Web.config açılmalıdır ve httpprotocol etiketi içerisindeki customheaders etiketi içerisine gösterilen satır eklenmelidir.

```
<!-- GÜVENLİ YAPILANDIRMA -->

<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <system.webServer>
        <security>
            ...
        </security>
        <httpProtocol>
            <customHeaders>
                <add name="X-Content-Type-Options" value="nosniff">
                </add>
            </customHeaders>
        </httpProtocol>
    </system.webServer>
</configuration>
```

b) Apache Web Sunucular

Debian / Ubuntu tabanlı linux işletim sistemlerinde yer alan apache web sunucularda apache2.conf, RedHat / Centos tabanlı linux işletim sistemlerinde yer alan apache web sunucularda httpd.conf dosyası açılmalıdır ve dosya içeriğinin en altına belirtilen satır eklenmelidir.

```
Header set X-Content-Type-Options "nosniff"
```

c) Nginx Web Sunucular

Nginx web sunucularda nginx.conf konfigürasyon dosyası açılmalıdır ve dosya içeriğindeki http { ... } bloğu içerisine belirtilen satır eklenmelidir

```
add_header X-Content-Type-Options nosniff;
```

d) Tomcat v.b. Java Web Sunucular

Tomcat gibi hafif java uygulamalarını taşıyabilen servlet container'larda ya da GlashFish, JBoss, WebLogic gibi kompleks java uygulamalarını taşıyabilen servlet container'larda spring framework'ünü kullanan web uygulamalarının src/main/java/hello/ dizininde WebSecurityConfig.java adlı bir dosyasısı bulunur. Bu java dosyasında gösterilen java kod bloğuna "Eklenecek Kodlar Başlıyor" yorum satırı ile "Eklenecek Kodlar Bitti" yorum satırı arasındaki satırlar eklenmelidir:

```
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {

        // Eklenecek Kodlar Başlıyor

        http
        .headers()
        .contentTypeOptions();

        // Eklenecek Kodlar Bitti

    }
}
```

Sonuç

En nihayetinde yapılandırma dosyasında yapılan değişiklik sonrası web sunucusu yazılımı yeniden başlatılmalıdır. Böylelikle kullanıcıların gönderdiği http / https taleplerine karşılık web sunucudan dönen http / https yanıtlarında X-Content-Type-Options önlemi yer alır duruma gelecektir.

Referanslar:

1. https://en.wikipedia.org/wiki/Content_sniffing
2. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
3. https://www.geeksforgeeks.org/http-headers-x-content-type-options/
4. https://geekflare.com/http-header-implementation/
5. https://www.acunetix.com/vulnerabilities/web/cross-site-scripting-content-sniffing/
6. https://www.denimgroup.com/resources/blog/2019/05/mime-sniffing-in-browsers-and-the-security-implications/
7. https://hackerone.com/reports/363845
8. https://security.stackexchange.com/questions/57615/content-sniffing-xss-vulnerable-browsers
9. https://vulncat.fortify.com/en/detail?id=desc.dataflow.java.cross_site_scripting_content_sniffing
10. http://www.insiderattack.net/2014/04/configuring-secure-iis-response-headers.html
11. https://www.ryadel.com/en/iis-web-config-secure-http-response-headers-pass-securityheaders-io-scan/

12. https://stackoverflow.com/questions/16385541/content-security-policy-invalid-command
13. https://www.keycdn.com/blog/http-security-headers/
14. https://stackoverflow.com/questions/39502968/apache-difference-between-header-always-set-and-header-set
15. https://scotthelme.co.uk/a-new-security-header-referrer-policy/
16. https://blog.qualys.com/securitylabs/2016/01/07/open-redirection-a-simple-vulnerability-threatens-your-web-applications
17. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3175
18. https://blog.appcanary.com/2017/http-security-headers.html
19. https://www.netsparker.com.tr/blog/web-guvenligi/turkiyede-http-guvenlik-headerlerinin-kullanimi/