1.1.1 Clickjacking Saldırılarına Karşı Koruma Eksikliği (Missing X-Frame-Options Header) (CWE-1021)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Kullanıcılara fark ettirmeden istemediği eylemleri gerçekleştirtme

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Clickjacking web site ziyaretçilerinin farkına varmadan farklı bir web sayfa öğesine tıklamaları sonucu yaşanan saldırılara denir. Birçok clickjacking saldırı türü vardır. Bunlar arasından çoğu metot olarak html iframe'lerle alakalı sömürü (exploitation) yolunu takip ederler ve bu saldırılara karşı önlemler de sayfa frame'leme üzerine yoğunlaşır.

Clickjacking saldırı türlerinden birini ifade edecek olursak örneğin bir saldırganın tıklanabilir bir nesnenin (örn; butonun veya linkin) üzerinde transparan (şeffaf / görünmez) bir iframe koyması bir clickjacking saldırısı türüdür. Bu clickjacking saldırısında sadece tıklanabilir nesne sayfada görünür, fakat bu tıklanabilir nesnenin üzerinde şeffaf / görünmez bir iframe vardır. Dolayısıyla bir kullanıcı tıklanabilir nesneye tıkladığında tıklanabilir nesne yerine üzerindeki şeffaf / görünmez iframe'e tıklamış olur. Böylece kullanıcı istemediği bir eylemi gerçekleştirebilir.

Clickjacking için bahsedilen saldırı türüne dair bir senaryo örneği vermek gerekirse örneğin bir ziyaretçi zararlı bir web sitesinde bir formu kapamak için butona tıklamak ister. Ziyaretçi butona tıkladığında butona tıkladığını düşünür, fakat bunun yerine üzerindeki şeffaf iframe'e tıklar ve bir truva atı indirir, veya banka hesabına para transfer eder, veya bilgisayarındaki yerleşik mikrofonunu ve webcam'ini açar. Bu örnek özelinde zararlı web sitesi bilinen bir legal web sitesinin sahte kopyası olabilir. Bu clickjacking saldırısı türünü yapabilmek için saldırgan zararlı web sitesini internette eposta yoluyla veya benzer farklı yollarla paylaşabilir. Daha sonra kullanıcılar sitedeki kapat butonuna tıkladıklarında iframe'e tıklamış olurlar ve böylece saldırganın istediği eylemi gerçekleştirmiş olurlar.

Aynı saldırı türüne dair bir başka senaryo örneği vermek gerekirse legal bir web site içerikleri zararlı bir web sitesinde iframe'lenerek kullanılabilir. Örneğin legal Facebook sitesinin like ve share butonları zararlı bir web sitesinde tıklanabilir bir nesnenin üzerine şeffaf olarak

konulabilir. Böylece kullanıcılar zararlı web sitesinde tıklanabilir nesneye tıkladıklarında aslında zararlı web sitesindeki içerik için Like veya Share butonlarından birine basmış olurlar. Kullanıcıların bu beğenme veya paylaşma işlemi kullanıcı facebook profillerine yansır, bu şekilde şüpheli içerik yayılabilir. Zararlı web site sahibi like veya share kasarak zararlı web sitesine daha fazla kullanıcı ve potansiyel kurban çekebilir. Bu clickjacking saldırı senaryosunda önceki senaryoya nazaran doğrudan legal bir web sitenin suistimal edilmesi söz konusudur.

Clickjacking tek tip bir saldırı değildir. Geniş bir çeşitlilikte atak vektörüne ve tekniğine sahip bir saldırıdır. Genellikle "UI redress" saldırısı (kullanıcı arayüzü yerine koyma saldırısı) olarak adlandırılır. Saldırılar üst üste binen içeriğin kullanımına bağlı olarak genel itibariyle iki kategoriye ayrılabilir. Overlay-based (kaplama bazlı) saldırılar, ki bu en popüleridir, bir de şeffaf / görünmez iframe'lerde sayfaları gömme, ki bu en yaygın kullanılan teknik yaklaşımdır. Overlay-based (kaplama bazlı) clickjacking'de birkaç adet ana kategori mevcuttur.

- Tamamen transparan kaplama: Bu metotta transparan legal bir web sayfası özenle hazırlanmış zararlı bir web sitesinde nesnelerin üzerine yerleştirilir. Legal web sayfası görünmez bir iframe içerisinde zararlı web sitesinde yüklenir ve z-index'i yüksek değerde tutularak görünen zararlı web site sayfasının üzerinde konumlandırılır.

- Kırpma: Bu saldırı türünde saldırgan görünen zararlı web site sayfası üzerindeki transparan frame sayfasının sadece belirli parçalarını kaplama olarak kullanır. Saldırının amacına bağlı olarak bu örneğin butonların önüne görünmez linkler konulması olabilir. Böylece umulandan farklı bir eylem gerçekleştirilir.

- Gizli kaplama: Bu saldırıda saldırgan 1 px x 1 px ebatlarında zararlı bir içerik içeren iframe oluşturur ve fare imlecinin katmansal olarak hemen altına yerleştirir. iframe fare imlecini takip eder. Herhangi bir tıklamada bu tık zararlı web sayfasında işleyecektir.

- Tıklama Event'inin Düşmesi: Zararlı bir web sitesinde sayfanın önüne zararlı web sayfasını tamamen kapatacak şekilde legal bir web sayfası iframe'i koyulur. Saldırgan, fare imleci css event özelliğini iframe'de gösterilen (üstte gösterilen) sayfa için none yapar.

  CSS:

  ... { pointer-events: none; }

Böylece tıklamalar üstte görünen sayfada çalışmaz ve tıklama geldiğinde bu tıklama legal web sayfa kaplamasının altındaki zararlı web sayfasına düşer. Zararlı web sayfasındaki nesneler için herhangi bir pointer-events tanımlamaları olmayacağından varsayılan olarak tıklamalar zararlı web sayfası içeriğinde çalışır olacaktır ve saldırgan ön yüzdeki görünen legal web sayfasında tıklanacak yerlerin konumsal olarak altına zararlı unsurlar koyarak tıklamaların arkadaki zararlı web sayfa içeriğinde işlemesi ile zararlı faaliyetler yürütebilir.

- v.b.

Clickjacking saldırılarında zararlı bir web sitede saldırgana ait zararlı kişisel iframe'ler ile faaliyetler yürütülmesi yolu vardır, zararlı bir web sitede legal bir web sitenin iframe'lenerek kullanılması / suistimal edilmesi yolu vardır, ve legal bir web sitenin hack'lenmesi (ele geçirilmesi) sonucu legal web siteye clickjacking yapan iframe'ler yerleştirilmesiyle yine legal web sitesinin kullanılması / suistimal edilmesi yolu vardır. Legal web siteleri clickjacking'e karşı koruma sağladıklarında iframe'ler yoluyla yabancı başka web uygulama adreslerinde veya - şayet legal web uygulamaya sızılmışsa - iframe'ler yoluyla legal web uygulama adresinin kendisinde içeriklerinin kullanılması / suistimal edilmesi (exploit edilmesi) yolu önlenir, ve legal web uygulama sahipleri ile legal web uygulama kullanıcıları clickjacking saldırılarına karşı korunur.

Legal web uygulama sahipleri web uygulamalarının clickjacking adı verilen saldırılarla suistimal edilmemesi için önlem uyguladıklarında saldırganlar legal web uygulamada kar elde edemezler, legal web uygulamayı zarara uğratamazlar ve legal web uygulamadaki diğer kullanıcıları zarara uğratamazlar. Kısaca web uygulamalarının clickjacking saldırılarında kullanılmasına mani olmuş olurlar.

Kurum web uygulamasında "clickjacking saldırılarına karşı önlem alınmaması (cwe-346)" açıklığı tespit edilmiştir:

:::::BULGU:::::

Açıklığın Önlemi:

Legal web sitelerini iframe'leyerek kullanan saldırganlara karşı legal web siteleri ve kullanıcıları clickjacking saldırılarından x-frame-options yanıt başlığı ile korunabilirler. Bu güvenlik başlığı genel manasıyla şu şekilde korur: Web tarayıcı ekranlarına yüklenen web sayfalarında kullanılan iframe'ler içeriklerini almak için ilaveten bir web sitesine talep yaparlar. Gelen yanıt paketinde X-Frame-Options başlığı yer alırsa web tarayıcıların iframe'i ekrana yükleyip yüklememesi gelen paketteki X-Frame-Options ayarına göre yapılır.

iframe'lerin yaptığı taleplere yanıt olarak gelen paketlerde X-Frame- Options yanıt başlığı istemci tarafta web tarayıcılara direktif verir ve iframe'in içeriğinin web tarayıcı ekranına yüklenip yüklenmeyeceğini belirtir. Aldığı üç farklı konfigürasyonla bunu yapar: DENY, SAMEORIGIN ve ALLOW-FROM. X-Frame-Options ile bu konfigürasyonlardan birini kullanan legal bir web uygulama içeriklerinin iframe'lenerek çeşitli zararlı lokasyonlarda (web uygulamalarda) kullanılması durumlarında içeriklerinin o yerlerde görüntülenmesi sırası geldiğinde web tarayıcı tarafından görüntülenmemesi gerektiğini veya sadece belirli yerlerde görüntülebileceğini kısıt olarak koyabilir. Bu sayede legal web uygulama sahipleri, içeriklerinin keyfi olarak herhangi bir yerde iframe'lenmesini önler ve uygulamalarının clickjacking saldırılarına karşı kullanımını / suistimalini engeller.

- DENY

Bir web uygulama DENY ayarını kullandığında hiçbir adreste (kendi adresi dahil) içeriklerinin iframe olarak sunulmasına izin vermez.

```
x-frame-options: DENY
```

Eğer teşebbüs edilirse içeriğini iframe'e yanıt olarak verdiği zaman X-Frame- Options başlığı da geleceğinden web tarayıcı gelen içeriği yüklemez. Böylece olası üçüncü taraf web siteleri üzerinden veya kendi olası hack'lenmiş web sitesi üzerinden gelebilecek clickjacking saldırılarına karşı web tarayıcılara sayfa yükletmeme ile koruma sağlar.

- SAMEORIGIN

Bir web uygulama SAMEORIGIN kullandığında sadece kendi adresinde içeriklerinin iframe olarak sunulmasına izin verir. Diğer hiçbir adres altında içeriklerinin iframe olarak sunulmasına izin vermez.

```
x-frame-options: SAMEORIGIN
```

Eğer üçüncü taraf web adreslerde teşebbüs edilirse içeriğini iframe'e yanıt olarak verdiği zaman X-Frame-Options başlığı da geleceğinden web tarayıcı gelen içeriği yüklemez. Böylece olası üçüncü taraf web siteleri üzerinden gelebilecek clickjacking saldırılarına karşı web tarayıcılara sayfalarını yükletmeme ile koruma sağlar.

- ALLOW-FROM

Bir web uygulama ALLOW-FROM kullandığında belirttiği üçüncü taraf web adresinde içeriklerinin iframe olarak kullanılmasına izin verir. Diğer hiçbir üçüncü taraf adreste ise içeriklerinin iframe olarak sunulmasına izin vermez.

```
x-frame-options: ALLOW-FROM https://www.domain.com
```

Eğer belirtilenin dışındaki üçüncü taraf web adreslerde teşebbüs edilirse içeriğini iframe'e yanıt olarak verdiği zaman X-Frame-Options başlığı da geleceğinden web tarayıcı diğer tanımlanmamış üçüncü taraf web adreslerde gelen içeriği yüklemez

X-Frame-Options DENY, SAMEORIGIN ve ALLOW-FROM değerlerini alır. Bunlar arasından DENY en sağlam konfigürasyondur. SAMEORIGIN ile halen clickjacking saldırısı yapılabilir. Çünkü zafiyetli web uygulamaya sızılabilir ve zafiyetli web uygulamada bir içerik iframe olarak zafiyetli uygulamanın bir başka yerine yansıtılabilir. Örneğin yansıtıldığı yerde iframe opak olacağı için görmeyen kullanıcılar arkasındaki görünen nesneye tıklamak istediklerinde iframe'e tıklayabilirler ve zafiyetli web uygulamada farklı bir eylem gerçekleştirebilirler. SAMEORIGIN bu noktada fayda etmeyecektir ve clickjacking'e bu şekilde kullanıcılar maruz kalabilecektir. Dolayısıyla en garanti çözüm DENY'dir. Fakat eğer web uygulamanın bazı içerikleri iframe olarak yine web uygulamanın farklı yerlerinde kullanılıyorsa en düşük mertebe olarak SAMEORIGIN kullanılabilir. Bu optimal değer ile hem web uygulama çalışırlığı sürdürülmüş olur hem de belli kademe güvenlik sağlanmış olur. Eğer üçüncü taraf bir bilinen web sunucuda web uygulama içeriği iframe olarak kullanılacaksa ALLOW-FROM ile sadece o üçüncü taraf adreslere izin verilebilir ve böylece web uygulama içeriği belirtilen üçüncü taraf web adreslerde iframe olarak kullanılabilir. Yine ALLOW-FROM optimal değeri ile hem web uygulama ve bileşenleri çalışırlığı sürdürülmüş olur hem de belli kademe güvenlik sağlanmış olur. X-Frame-Options aldığı üç ayar ile clickjacking'e karşı belli kademelerde güvenlik sağlar.

[*] Bilgi:

Legal web siteler hack'lenirlerse (ele geçirilirlerse) clickjacking saldırısı yapan iframe'ler legal web sitelere de yerleştirilebilir. Bu durumda X-Frame-Options legal web sitelerini hem aynı originden (adresten) hem de farklı originlerden (web adreslerden) gelecek clickjacking saldırılarına karşı koruyacak şekilde yapılandırılabilir. X-Frame-Options: DENY bu işe yarar.

[!] Uyarı:

X-Frame-Options'da ALLOW-FROM direktifi artık modern web tarayıcılarda çalışmamaktadır. Sadece DENY ve SAMEORIGIN direktifleri çalışmaktadır. Bu nedenle ALLOW-FROM kullanıldığında dikkatli olunmalıdır. Diğer türlü X-Frame-Options güvenlik önlemi bazı web tarayıcılarda beklenen korumayı gerçekleştirmeyebilir. Bu durum için X-Frame-Options başlığı Content-Security-Policy başlığı ile birlikte kullanılabilir. Content-Security-Policy ile de clickjacking güvenliği sağlanabilmektedir.

Clickjacking saldırılarından kurum web uygulama kullanıcılarını ve kurum web uygulamayı korumak için bir http güvenlik başlığı olan X-Frame-Options yanıt başlığı kullanılmalıdır:

a) IIS Web Sunucular

IIS web 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-Frame-Options" value="DENY">
                </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-Frame-Options "DENY"
```

c) Nginx Web Sunucular

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

```
add_header X-Frame-Options "DENY";
```

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()
        .frameOptions().deny(); // veya örn; sameOrigin();

        // Eklenecek Kodlar Bitti

    }
}
```

[!] Uyarı:

X-Frame-Options başlığı SAMEORIGIN değerini aldığı durumda kurum web uygulamasındaki sayfaları / içerikleri sunan iframe'ler sadece yine kurum web uygulamasında gösteriliyorsa çalışır durumda olacaktır.

Kurum web uygulamasındaki sayfaları / içerikleri sunan iframe'ler dışarıdan bir web uygulaması tarafından içerik olarak sunulmaya çalışılırsa dışardaki bu web uygulamasının kurum web uygulamasına ait sayfaları / içerikleri sunması engellenecektir. Eğer belirli web uygulamalarının kurum web uygulaması sayfalarını / içeriklerini iframe olarak sunabilmesine izin vermek istenirse (örn; farklı bir kurum web

uygulamasına) X-Frame-Options başlığı farklı web sunucular için belirtilen satırlardaki gibi kullanılmalıdır.

IIS web sunucular;

```
<!-- GÜVENLİ YAPILANDIRMA -->

<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <system.webServer>
        <security>
            ...
        </security>
        <httpProtocol>
            <customHeaders>
                <add name="X-Frame-Options" value="ALLOW-FROM
https://www.kurumwebuygulamasi2.com">
                </add>
            </customHeaders>
        </httpProtocol>
    </system.webServer>
</configuration>
```

Apache web sunucular

```
Header set X-Frame-Options "ALLOW-FROM https://www.kurumwebuygulamasi2.com"
```

Nginx web sunucular;

```
add_header X-Frame-Options "ALLOW-FROM https://www.kurumwebuygulamasi2.com";
```

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 Referrer-Policy önlemi yer alır duruma gelecektir.

Referanslar:

1. http://whatis.techtarget.com/definition/clickjacking-user-interface-or-UI-redressing-and-IFRAME-overlay
2. https://javascript.info/clickjacking
3. https://www.keycdn.com/blog/x-frame-options
4. https://securityboulevard.com/2019/08/clickjacking-attacks-what-they-are-and-how-to-prevent-them/
5. https://www.netsparker.com/blog/web-security/clickjacking-attacks/
6. https://developer.mozilla.org/en-US/docs/Web/CSS/pointer-events
7. https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html
8. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
9. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors
10. https://cure53.de/xfo-clickjacking.pdf
11. https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/insecure-frame-external/
12. https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-x-frame-options-header/
13. https://stackoverflow.com/questions/3332756/difference-between-window-location-href-and-top-location-href
14. http://seclab.stanford.edu/websec/framebusting/framebust.pdf
15. https://stackoverflow.com/questions/1192228/scrolling-an-iframe-with-javascript
16. https://www.w3schools.com/jsref/met_win_scrollby.asp
17. https://www.youtube.com/watch?v=2z4E9M8B4-g&ab_channel=TommyTessandori
18. http://www.insiderattack.net/2014/04/configuring-secure-iis-response-headers.html
19. https://www.ryadel.com/en/iis-web-config-secure-http-response-headers-pass-securityheaders-io-scan/
20. https://stackoverflow.com/questions/16385541/content-security-policy-invalid-command
21. https://www.keycdn.com/blog/http-security-headers/
22. https://stackoverflow.com/questions/39502968/apache-difference-between-header-always-set-and-header-set
23. https://scotthelme.co.uk/a-new-security-header-referrer-policy/
24. https://blog.qualys.com/securitylabs/2016/01/07/open-redirection-a-simple-vulnerability-threatens-your-web-applications
25. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3175
26. https://blog.appcanary.com/2017/http-security-headers.htm
27. https://www.netsparker.com.tr/blog/web-guvenligi/turkiyede-http-guvenlik-headerlerinin-kullanimi/