1.1.1 Spring Framework'te X-Frame-Options Kullanımı Eksikliği (Missing X-Frame-Options in Spring) (CWE-1021)

Açıklık Önem Derecesi: Orta

Açıklığın Etkisi: Clickjacking saldırılarına karşı savunmasız kalma

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 uygulamasında clickjacking saldırılarına karşı önlem alınmadığı tespit edilmiştir:

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

(Örnek Bulgu)

(Spring Framework configure metodu)

(X-Frame-Options enable eden kod satırı bulunmamakta)

Şekil XX. X-Frame-Options Eksikliği

Açıklığın Önlemi:

Java uygulamalarda - Spring Framework'ünde - x-frame-options ayarı şu şekilde uygulanabilir:

Java:

```
// Adding X-Frame-Options in Spring Security
// Using Java Configuration

@EnableWebSecurity
@Configuration
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .headers()
                .frameOptions();
    }
}
```

XML:

```
// Adding X-Frame-Options with parameters
// in Spring Security Using XML Configuration
<http>
    <!-- ... -->
    <headers>
        <frame-options policy="DENY"/>
    </headers>
</http>
```

Eğer uygulamada iframe'leme unsuru mevcutsa / kullanılıyorsa şu şekilde alternatif kısıt ayarı ile hem kullanılan iframe'ler kullanılmaya devam edilebilir hem de güvenlik sağlanabilir.

Java:

```
http.headers().frameOptions().sameOrigin();
```

XML:

```
// Adding X-Frame-Options with parameters
// in Spring Security Using XML Configuration

<http>
    <!-- ... -->
    <headers>
        <frame-options policy="SAMEORIGIN"/>
    </headers>
</http>
```

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.html#x-content-type-options
27. https://www.cyberciti.biz/faq/nginx-send-custom-http-headers/
28. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
29. https://geekflare.com/tomcat-http-security-header/
30. https://docs.spring.io/spring-security/site/docs/current/reference/html/headers.html
31. https://spring.io/guides/gs/securing-web/
32. https://spring.io/blog/2013/08/23/spring-security-3-2-0-rc1-highlights-security-headers
33. https://www.dailyrazor.com/blog/glassfish-vs-tomcat/
34. http://www.edu4java.com/en/servlet/servlet1.html
35. https://stackoverflow.com/questions/24182367/how-to-add-x-content-type-options-to-tomcat-configuration
36. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/script-src#Unsafe_inline_script
37. https://stackoverflow.com/questions/28647136/how-to-disable-x-frame-options-response-header-in-spring-security