1.1.1 Aşırı İzinli Content Security Policy (Overly Permissive Content Security Policy) (CWE-346)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: XSS saldırılarına maruz kalma, Clickjacking saldırılarına maruz kalma, İçerik Sahteciliği saldırılarına maruz kalma

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Content-Security-Policy başlığı mevcut web sayfasındaki bir script, bir gömülü child (alt) frame, bir gömülü parent (üst) frame veya bir resmin içerik kaynağının güvenli ve izinli olmasını denetler ve zorlar. Eğer bir web sayfası içerisinde içeriğin kaynağı katı Content-Security-Policy politikasına bağlı kalmıyorsa bu içerik web tarayıcı tarafından reddedilir. Katı Content Security Policy politikasındaki bir kusur web uygulama kullanıcılarını XSS, Clickjacking, Content Forgery (İçerik Sahteciliği) ve dahası saldırılara maruz bırakabilir.

Örneğin Content-Security-Policy başlığı şu değerle doldurulursa

```
# Güvensiz Kullanım

style-src * 'unsafe-inline'; script-src * 'unsafe-inline'; img-src *; connect-src *; frame-src *; object-src: *; font-src: *; media-src: *
```

tüm dış sunuculardan kurum web uygulamasına kaynak gelebilir ve çalışabilir izni verilmiş olur. Bu oldukça esnek kurala Javascript dosyalarının dışarıdaki google sunucularından gelişine izin ver, başka hiçbir sunucuya izin verme kısıtı konulmak istenirse CSP başlığı şu şekilde güncellenir:

```
# Güvensiz Kullanım

style-src * 'unsafe-inline'; script-src https://www.google.com.tr 'unsafe-inline'; img-src *; connect-src *; frame-src *; object-src: *; font-src: *; media-src: *
```

Kurum web uygulama içerisinde script etiketleri ve style etiketleri aralarında javascript ve css kodları olacak şekilde kullanım halindelerse 'unsafe-inline'lar ile onların çalışırlığı izni sürdürülmüş olacaktır, **ancak 'unsafe-inline' halen kurum web uygulamasını güvensiz kılmaktadır.** Bu nedenle kurum web uygulamasını güvende tutacak metot uygulamadaki tüm <script>…</script> ve <style>…</style> bloklarının html <head etiketleri arasında dosya.css ve dosya.js şeklinde dosya dahil etme yöntemiyle kullanılmasıdır, yani 'unsafe-inline'ların kaldırılmasıdır.

```
# script-src Güvenli Kullanım, Diğerleri Güvensiz Kullanım

style-src *; script-src https://www.google.com.tr; img-src *; connect-src *;
frame-src *; object-src: *; font-src: *; media-src: *
```

Eğer uygulamada script ve style'ları dosya halinde toparlamada teknik zorluklar yaşanacaksa bu durumda yine 'unsafe-inline'lar kaldırılmalıdır ve uygulamadaki <script>…</script> ve <style>…</style> etiketleri içeriklerinin hash (özet) hali örneğin sha256 algoritmasıyla alınıp hash (özet) değeri base64 ile kodlanarak CSP başlığındaki script-src ve style-src'a verilmelidir. Böylece uygulamadaki satır arası olan bu <script>…</script> ve <style>…</style> blokları CSP başlığınca tanımlı olacaklardır ve saldırı sonrası ilave gelebilecek tanımsız bloklara (<script>…</script> ve <style>…</style>'lara) karşı engelleme önlemi sürmüş olacaktır.

```
# script-src Güvenli Kullanım, Diğerleri Güvensiz Kullanım

style-src *; script-src https://www.google.com.tr 'sha256-
B2yPHKaXnvFWtRChIbabYmUBFZdVfKKXHbWtWidDVF8='; img-src *; connect-src *;
frame-src *; object-src: *; font-src: *; media-src: *
```

Content-Security-Policy modern web tarayıcılar tarafından güvenilir medya, resim, script, frame ve dahası içeriklerin bir göstergesi olarak kullanılır. Eğer bu politika çok geniş tanımlanırsa güvensiz içeriğin engellenmesinde etkisiz kalır. Bu duruma "Aşırı İzinli Content Security Policy (CWE-346)" açıklığı adı verilir.

Kurum web uygulamada Content-Security-Policy'nin aşırı geniş izinde tanımlandığı tespit edilmiştir.

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

Açıklığın Önlemi:

Content-Security-Policy web uygulamalarda

- back-end (arka-uç) kaynak kodlarında,
- konfigürasyon dosyalarında veya
- front-end (ön-uç) html sayfaların <head> bölümündeki <meta> etiketlerinde

tanımlanabilir.

Tavsiyeler ise şu şekildedir:

- CSP başlığındaki style-src, script-src, img-src, connect-src, frame-src, object-src, font-src, media-src ayarları uygulama hangi uzak sunucu bağlantılarına ihtiyaç duymaktaysa onları gösterecek adreslerle ayarlanmalıdır.
- * (wildcard), unsafe-inline ve unsafe-eval kullanımlarından sakınılmalıdır.
- Ayrıca web sayfalardaki ayrı müstakil dosya olarak toparlanamamış satır arası <script>…</script> ve <style>…</style> kod bloklarının CSP politikasına dahil edilebilmesi için hash (özet) hallerinin alınması ve base64 ile kodlanarak o şekilde csp politikasına eklenmesi gerekmektedir.

Bu şekilde kurum web uygulaması çalışırlığı bozulmamış olacaktır ve aynı zamanda web uygulama güvenli olacaktır.

Referanslar:

1. https://cwe.mitre.org/data/definitions/346
2. https://www.invicti.com/web-vulnerability-scanner/vulnerabilities/an-unsafe-content-security-policy-csp-directive-in-use/
3. https://www.invicti.com/web-vulnerability-scanner/vulnerabilities/wildcard-detected-in-domain-portion-of-content-security-policy-csp-directive/
4. https://www.invicti.com/web-vulnerability-scanner/vulnerabilities/wildcard-detected-in-scheme-portion-of-content-security-policy-csp-directive/
5. https://www.invicti.com/web-vulnerability-scanner/vulnerabilities/wildcard-detected-in-port-portion-of-content-security-policy-csp-directive/
6. https://vulncat.fortify.com/en/detail?category=HTML5&subcategory=Overly%20Permissive%20Content%20Security%20Policy#Java%2fJSP