1.1.1 Hassas Verilerin LocalStorage'da Depolanması (Client HTML5 Store Sensitive Data in Web Storage (CWE-922) (CWE-312)

Açıklık Önem Derecesi: Orta

Açıklığın Etkisi: Hassas verilerin ele geçirilebilmesi

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

localStorage bir HTML5 özelliğidir. İstemci tarafta veri depolamaya yarayan bir Javascript nesnesidir. Veriler key=value şeklinde çerez olarak istemci tarafın, yani web tarayıcının local storage bölümünde depolanırlar. localStorage temel olarak web geliştiricilerine Javascript kullanarak herhangi bir veriyi kullanıcı web tarayıcısında depolama imkanı sunar.

Örneğin verilen örnekte Javascript kodlaması ile kişisel bir bilginin web tarayıcının local storage'ına depolanışı gösterilmiştir:

test.html:

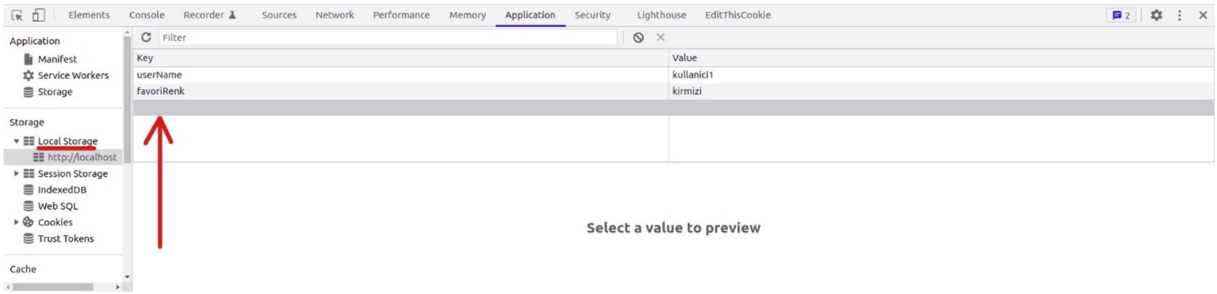
```
    <script>
    // a) Local Storage'da aşağıdaki söz dizimlerinden
    // biriyle veri depolaması yapılabilir.
    localStorage.userName = "kullanici1";
    localStorage.setItem("favoriRenk", "kirmizi");

    // b) Veri localStorage'da yer aldığında sonsuz dek
    // orada kalacaktır. Açıkça silinirse ancak o zaman gidecektir.
    alert(localStorage.userName + " 'in sevdigi renk " + localStorage.
favoriRenk + "dir.");

    // c) Local Storage'dan veri silmek kolaydır. Aşağıdaki
    // satırlar yorum olmaktan çıkarıldığında ilgili girdiler
    // Local Storage'dan silineceklerdir.

    //localStorage.removeItem("userName");
    //localStorage.removeItem("favoriRenk");
    </script>
```

Bu örnek bir html sayfa olarak web tarayıcıda görüntülendiğinde web tarayıcının F12 geliştirici araçları -> Application -> local storage bölümünde çerezlerin oluştuğu görülebilir.



Şekil XXX. Local Storage

Bu örnek web sayfanın oluşturduğu çerezleri silmek için her defasında elle c) adımında gösterildiği gibi silme prosedürü uygulamak gerekir (veya web tarayıcıda önbellek (cache) temizlemek gerekir). Bu durumdan dolayı örneğin localStorage kullanmaktansa bunun yerine HTML5'de bir diğer javascript nesnesi olan sessionStorage kullanılabilir (bkz. Şekil XXX'deki resimde sol sütunda Local Storage altında yer alan Session Storage seçeneği). sessionStorage localStorage'la aynı çalışır. Fakat bir özelliği hariç. sessionStorage web tarayıcı sekmesi kapandığı an web tarayıcıda depoladığı verileri otomatik bir şekilde siler. Böylece web tarayıcı önbelleğinden verileri silmek için localStorage'larda gerekli prosedür otomatik uygulanmış olur.

LocalStorage ile hassas veri depolamak sakıncalıdır. LocalStorage hakkındaki problem güvensiz oluşundandır. LocalStorage'lar web tarayıcılarda güvenli depolama mekanizması olarak kullanılmak üzere tasarlanmamışlardır. Basitçe anahtar/değer string'leri depolamak üzere tasarlanmışlardır. LocalStorage'da depolanan veriler XSS adı verilen saldırılara karşı tamamen savunmasızdırlar. Cookie'lerde HttpOnly bayrağı ile XSS'e karşı bir önlem vardır. Fakat Local Storage'da böylesi bir önlem yoktur.

Eğer hassas bir veri local storage'da depolanırsa ve web uygulamada gelecekte XSS açıklığı meydana gelirse XSS açıklığı yoluyla saldırganlar son kullanıcı ekranlarındaki web tarayıcılarda Javascript kodları çalıştırabilirler. Çalıştırdıkları javascript kodları ile son kullanıcıların web tarayıcılarındaki local storage'da yer alan tüm verileri kendi domain adresine gönderebilirler. Örneğin bir kullanıcının session token'ı (oturum jetonu) local storage'da yer alırsa bu durumda saldırgan kullanıcının oturumunu çalmış olacaktır ve kullanıcı adına web uygulamada işlemler yapabilecektir. Sonuç olarak hiçbir hassas veri local storage'da depolanmamalıdır.

Local Storage'da depolanmaması gereken hassas verilere örnek olarak şunlar gösterilebilir:

- User ID'ler
- Session ID'ler
- JWT (JSON Web Token) 'lar
- Kişisel Bilgiler
- Kredi Kart Bilgileri
- API Anahtarları
- v.b.

Kurum web uygulamasında hassas bir verinin (json token'ın) local storage'da depolandığı tespit edilmiştir:

:::::BULGU::::

Açıklığın Önlemi:

Javascript uygulamalarda veri depolamak için localStorage kullanımından kaçınılmalıdır. Eğer gerekiyorsa minimal bir yaklaşım gereği sessionStorage kullanılabilir. Herkese açık bir şekilde bilinmesi sakınca oluşturmayan tüm hassas nitelikte olmayan verileri depolamak için sessionStorage kullanılabilir. Fakat hassas veri depolamak için localStorage veya sessionStorage kullanılmamalıdır. Herkesçe bilinmesi sakınca oluşturan hassas nitelikteki verileri istemci tarafta (yani web tarayıcıda) depolamak için HttpOnly, Secure ve SameSite gibi güvenlik önlemlerine sahip Set-Cookie yanıt başlığı kullanılmalıdır. Örneğin session token (oturum jetonu) hassas verisi Cookies yanıt başlığında yer almalıdır.

Referanslar:

1. https://dev.to/rdegges/please-stop-using-local-storage-1i04
2. https://snyk.io/blog/is-localstorage-safe-to-use/
3. https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage
4. https://stackoverflow.com/questions/44133536/is-it-safe-to-store-a-jwt-in-localstorage-with-reactjs
5. https://dev.to/cotter/localstorage-vs-cookies-all-you-need-to-know-about-storing-jwt-tokens-securely-in-the-front-end-15id
6. https://stormpath.com/blog/where-to-store-your-jwts-cookies-vs-html5-web-storage
7. https://cheatsheetseries.owasp.org/cheatsheets/HTML5_Security_Cheat_Sheet.html#local-storage
8. https://cwe.mitre.org/data/definitions/922.html
9. https://cheatsheetseries.owasp.org/cheatsheets/HTML5_Security_Cheat_Sheet.html#local-storage
10.