Testing for Browser Cache Weaknesses (Tarayıcı Önbellek Zayıflıkları için test)

Summary (Özet)

Bu aşamada test cihazı, uygulamanın tarayıcıya hassas verileri tutmaması talimatını doğru bir şekilde doğru bir şekilde kontrol eder. Tarayıcılar bilgileri önbellekleme ve tarih için saklayabilirler.
Önbellekleme performansı artırmak için kullanılır, böylece daha önce görüntülenen bilgilerin tekrar indirilmesi gerekmez. Tarih mekanizmaları kullanıcı rahatlığı için kullanılır, böylece kullanıcı kaynağın alındığı sırada tam olarak ne gördüklerini görebilir. Kullanıcıya hassas bilgiler (iletleri, kredi kartı bilgileri, Sosyal Güvenlik Numarası veya kullanıcı adı gibi) görüntülenirse, bu bilgiler önbelleğe para kazanma veya geçmiş için saklanabilir ve bu nedenle tarayıcının önbelleğini inceleyerek veya tarayıcının
Arka düğmesine basarak geri alınabilir.

Test Objectives (Test Hedefleri)

- Uygulamanın hassas bilgileri istemci tarafında saklayıp saklamadığını gözden geçirin.
- Erişim izni olmadan gerçekleşebilirse gözden geçirin.

How to Test (Nasıl Test Edilir)

Browser History (Tarayıcı Tarihi)

Teknik olarak, **Geri** düğme bir tarihtir ve bir önbellek değildir (bkz. HTTP: Tarih Listelerinde Önbelleğe Kavuşma).
Önbellek ve tarih iki farklı varlıktır. Bununla birlikte, daha önce görüntülenen hassas bilgileri sunmanın aynı zayıflığını paylaşırlar.

İlk ve en basit test, hassas bilgileri uygulamaya girmek ve oturum açmaktan oluşur. Daha sonra test cihazı, daha önce görüntülenen hassas bilgilerin

doğrulanmamışken erişilip erişilemeyeceğini kontrol etmek için tarayıcının **Arka** düğmesini tıklar.

Geri düğmesine basarak test cihazı önceki sayfalara erişebilir, ancak yeni sayfalara erişemiyorsa, o zaman bir kimlik doğrulama sorunu değil, bir tarayıcı geçmişi sorunudur. Bu sayfalar hassas veriler içeriyorsa, uygulamanın tarayıcının saklamasını yasaklamadığı anlamına gelir.

Kimlik doğrulamanın mutlaka teste dahil olması gerekmez. Örneğin, bir kullanıcı bir bültene kaydolmak için e-posta adresini girdiğinde, bu bilgiler düzgün bir şekilde ele alınmadığı takdirde geri alınabilir.

Geri tuşa geçsen hassas verileri göstermekten geri tuşlanabilir. Bu şöyle yapılabilir:

- Sayfayı HTTPS üzerinden teslim etmek.
- Ayar `Cache-Control: must-revalidate`

Browser Cache (Tarayıcı Önbellek)

Burada testçiler, uygulamanın tarayıcı önbelleğine herhangi bir hassas veri sızdırmadığını kontrol eder. Bunu yapmak için, bir proxy (OWASP ZAP gibi) kullanabilir ve oturuma ait sunucu yanıtları aracılığıyla arayabilir ve hassas bilgileri içeren her sayfa için sunucunun tarayıcıya herhangi bir veriyi önbellememesi için talimat verdiğini kontrol edebilirler. HTTP yanıt başlıklarında böyle bir direktif aşağıdaki direktiflerle birlikte verilebilir:

- `Cache-Control: no-cache, no-store`
- `Expires: 0`
- `Pragma: no-cache`

Bu direktifler genellikle sağlamdır, ancak ek bayraklar gerekli olabilir. `Cache-Control` Dosya sistemindeki sürekli olarak bağlantılı dosyaları daha iyi önlemek için başlık. Bunlar şunları içerir:

- `Cache-Control: must-revalidate, max-age=0, s-maxage=0`

```
HTTP/1.1:
Cache-Control: no-cache
```

```
HTTP/1.0:
Pragma: no-cache
Expires: "past date or illegal value (e.g., 0)"
```

Örneğin, testçiler bir e-ticaret başvurusunu test ediyorsa, kredi kartı numarası veya başka bir finansal bilgi içeren tüm sayfaları aramalıdır ve tüm bu sayfaların uyguladığını kontrol etmelidirler. `no-cache` Direktif. Kritik bilgiler içeren ancak tarayıcıya içeriklerini önbellememeleri için talimat vermezse, hassas bilgilerin diskte saklanacağını bilirler ve bunu tarayıcı önbelleğindeki sayfayı arayarak iki kez kontrol edebilirler.

Bu bilgilerin depolandığı tam konum, istemci işletim sistemine ve kullanılan tarayıcıya bağlıdır. İşte bazı örnekler:

- Mozilla Firefox:
  - Unix/Linux: `~/.cache/mozilla/firefox/`
  - Windows: `C:\Users\<user_name>\AppData\Local\Mozilla\Firefox\Profiles\<profile-id>\Cache2\`
- Internet Explorer:
  - `C:\Users\<user_name>\AppData\Local\Microsoft\Windows\INetCache\`
- Krom:
  - Windows: `C:\Users\<user_name>\AppData\Local\Google\Chrome\User Data\Default\Cache`
  - Unix/Linux: `~/.cache/google-chrome`

Reviewing Cached Information (Gözden Geçirme Bilgileri)

Firefox, önbelleğe alınmış bilgileri görüntülemek için işlevsellik sağlar, bu da bir testçi olarak yararınıza olabilir. Tabii ki endüstri ayrıca Chrome, Internet Explorer veya Edge için tercih edebileceğiniz veya ihtiyaç duyabileceğiniz çeşitli uzantılar ve harici uygulamalar üretti.

Önbellek detayları FirefoxFirefox, ChromeChrome ve Edge gibi çoğu modern tarayıcıda geliştirici araçları aracılığıyla da mevcuttur. Firefox ile URL'yi kullanmak da mümkündür `about:cache` Önbellek detaylarını kontrol etmek için.

Check Handling for Mobile Browsers (Mobil Tarayıcılar için ele almayı kontrol edin)

Önbellek direktiflerinin ele alınması mobil tarayıcılar için tamamen farklı olabilir. Bu nedenle, testçiler temiz önbelleklerle yeni bir tarama oturumu başlatmalı ve yukarıda belirtilen kavramları yeniden test etmek veya ayrı olarak test etmek için Chrome'un Aygıt Modu veya Firefox'un Yanıtlayıcı Tasarım Modu gibi özelliklerden yararlanmalıdır.

Ek olarak, ZAP ve Burp Suite gibi kişisel vekiller test cihazının hangisini belirtmesini sağlar. `User-Agent` Örümcekleri/çaprazları tarafından gönderilmelidir. Bu, bir mobil tarayıcıyla eşleşecek şekilde ayarlanabilir `User-Agent` İskele ve test edilen uygulama tarafından hangi önbellek direktiflerinin gönderildiğini görmek için kullanılır.

Gray-Box Testing (Gri-Kutu Testi)

Her iki senaryoda da test uzmanları sunucu yanıt başlıklarına ve HTML koduna tam erişime sahip olduğundan, test metodolojisi kara kutu durumuna eşdeğerdir. Ancak, gri kutu testinde test uzmanı, yalnızca kimliği doğrulanmış kullanıcıların erişebildiği hassas sayfaları test etmelerine olanak tanıyacak hesap kimlik bilgilerine erişebilir.

Tools (Araçlar)

- OWASP Zed Attack Proxy

References (Referanslar)

Whitepapers ()

- Caching in HTTP