Testing for Exposed Session Variables (Maruz Kalan Oturum Değişkenleri için Test)

Summary (Özet)

Session Tokens (Cookie, SessionID, Hidden Field), açığa çıkarsa, genellikle bir saldırganın bir
kurbanı taklit etmesini ve başvuruya gayrimeşru olarak erişmesini sağlayacaktır. Özellikle istemci tarayıcısı ve uygulama sunucuları arasında geçiş yaparken, her zaman dinlemeden korunmaları önemlidir.

Buradaki bilgiler, taşıma güvenliğinin genel olarak verilerden ziyade hassas Oturum Kimliği verilerinin aktarımı için nasıl uygulandığıyla ilgilidir ve sitenin sunduğu verilere uygulanan önbellekleme ve taşımacılık politikalarından daha katı olabilir.

Kişisel bir vekil kullanarak, her bir istek ve yanıt hakkında aşağıdakileri tespit etmek mümkündür:

- Kullanılan protokol (örneğin, HTTP vs. HTTPS)
- HTTP Başlıkları
- Mesaj Beden (örneğin, POST veya sayfa içeriği)

Oturumlu kimlik verileri istemci ve sunucu arasında her geçtiğinde, protokol, önbellek ve gizlilik direktifleri ve vücut incelenmelidir. Ulaştırma güvenliği, GET veya POST taleplerinde, mesaj organlarında veya geçerli HTTP istekleri üzerinden diğer yollarla geçen Oturum Kimliklerini ifade eder.

Test Objectives (Test Hedefleri)

- Uygun şifrelemenin uygulanmasını sağlayın.
- Kıyaslama yapılandırmasını gözden geçirin.
- Kanalın ve yöntemlerin güvenliğini değerlendirin.

How to Test (Nasıl Test Edilir)

Testing for Encryption & Reuse of Session Tokens Vulnerabilities (Encrypton Testi ve Oturum Tokenlerinin Güvenlik Açıklarının Yeniden Kullanımı)

Kulak misafirilikten korunma genellikle SSL şifrelemesi ile sağlanır, ancak diğer tünelleme veya şifrelemeyi içerebilir. Oturum kimliğinin şifreleme veya kriptografik hashing'inin, ile temsil edilebilecek veriler değil, Oturum kimliğinin korunmasından ayrı olarak düşünülmesi
gerektiği belirtilmelidir.

Oturum Kimliği, bir saldırgan tarafından erişim sağlamak için uygulamaya sunulabilirse, o zaman bu riski azaltmak için transit olarak korunmalıdır. Bu nedenle, şifrelemenin, kullanılan mekanizmadan bağımsız olarak (örneğin, gizli bir form alanı) ne olursa olsun, Oturum Kimliğinin geçtiği herhangi bir istek veya yanıt için hem varsayılan hem de uygulanmasını sağlamalıdır. Değiştirme gibi basit kontroller `https://` ile `http://` Uygulama ile etkileşim sırasında, güvenli ve teminatsız siteler arasında yeterli ayrışmanın uygulanıp uygulanmadığını belirlemek için form yayınlarının
değiştirilmesi ile birlikte yapılmalıdır.

Kullanıcının Oturum Kimlikleri ile izlendiği ancak güvenliğin bulunmadığı (örneğin, kayıtlı bir kullanıcı indirmesini hangi kamuya açık belgelere dikkat ederek) sitede bir unsur varsa, farklı bir Oturum kimliğinin kullanılmasının şart olduğunu unutmayın. Bu nedenle, müşteri farklı bir tanesinin kullanılmasını sağlamak için güvenli olmayan unsurlardan güvenli olmayan unsurlara geçerken Oturum Kimliği izlenmelidir.

> Kimlik doğrulaması her başarılı olduğunda, kullanıcı şunları almayı beklemelidir:
>
> - Farklı bir oturum tokeni
> - HTTP İsteği her yaptıklarında şifreli kanal üzerinden gönderilen bir token

Testing for Proxies & Caching Vulnerabilities (Proxies ve Önbellekler için Test)

Proxies ayrıca başvuru güvenliğini gözden geçirirken de dikkate alınmalıdır. Çoğu durumda, müşteriler başvuruya kurumsal, ISS veya diğer vekiller veya protokol farkında ağ geçitleri (örneğin, Güvenlik Duvarları) aracılığıyla erişecektir. HTTP protokolü, aşağı akışlı vekillerin davranışlarını kontrol etmek için direktifler sağlar ve bu direktiflerin doğru uygulanması da değerlendirilmelidir.

Genel olarak, Oturum Kimliği asla şifrelenmemiş taşıma üzerinden gönderilmemeli ve asla önbellilmemelidir. Şifreli iletişimlerin hem varsayılan hem de Oturum Kimliklerinin herhangi bir transferi için zorunlu olduğundan emin olmak için uygulama incelenmelidir. Ayrıca, Oturum Kimliği geçtiğinde, orta ve hatta yerel önbellekler tarafından önlenmesini önlemek için direktifler yerinde olmalıdır.

Uygulama ayrıca hem HTTP/1.0 hem de HTTP/1.1 – RFC 2616 üzerindeki önbelleklerdeki verileri HTTP'ye atıfta bulunarak uygun kontrolleri tartışacak şekilde yapılandırılmalıdır. HTTP/1.1 bir dizi önbellek kontrol mekanizması sağlar. `Cache-Control: no-cache` Bir vekilin herhangi bir veriyi yeniden kullanmaması gerektiğini gösterir. Ne zaman `Cache-Control: Private` Uygun bir direktif gibi görünüyor, bu hala paylaşılmayan bir proxy'nin verileri önbelleğe almasını sağlar. Web-kafeler veya diğer paylaşılan sistemler söz konusu olduğunda, bu açık bir risk oluşturur. Tek kullanıcılı iş istasyonlarında bile, önbelleğe alınmış Session ID, dosya sisteminin bir uzlaşması veya ağ mağazalarının kullanıldığı yerlerde maruz kalabilir. HTTP/1.0 önbellekler tanımaz `Cache-Control: no-cache` Direktif.

> The (İngilizce) `Expires: 0`ve `Cache-Control: max-age=0`Kongların verileri açığa çıkarmamasını sağlamak için direktifler kullanılmalıdır. Oturumlu kimlik verileri, uygun önbellek direktiflerinin kullanımda olduğundan emin olmak için her bir istek / yanıt kodu incelenmelidir.

Testing for GET & POST Vulnerabilities (GET & POST Güvenlik Açıkları için Test)

Genel olarak, Oturum Kimliği Proxy veya Güvenlik Duvarı kayıtlarında maruz kalabileceğinden, GET talepleri kullanılmamalıdır. Ayrıca, diğer ulaşım türlerinden

çok daha kolay manipüle edilirler, ancak neredeyse her mekanizmanın müşteri tarafından doğru araçlarla manipüle edilebileceği belirtilmelidir. Ayrıca, çapraz site senaryo (XSS) saldırıları, kurbana özel olarak inşa edilmiş bir bağlantı göndererek en kolay şekilde sömürülmektedir. Bu, müşteriden POST olarak veri gönderilirse çok daha az olasıdır.

POST taleplerinden alan tüm sunucu kodu alıcı tüm kod, GET olarak gönderilirse verileri kabul etmediğinden emin olmak için test edilmelidir. Örneğin, aşağıdaki POST talebini göz önünde bulundurun (`http://owaspapp.com/login.asp`) sayfadaki bir günlüğü tarafından oluşturulur.

```
POST /login.asp HTTP/1.1
Host: owaspapp.com
[...]
Cookie: ASPSESSIONIDABCDEFG=ASKLJDLKJRELKHJG
Content-Length: 51

Login=Username&password=Password&SessionID=12345678
```

login.asp kötü uygulanıyorsa, aşağıdaki URL'yi kullanarak giriş yapmak mümkün olabilir: `http://owaspapp.com/login.asp?Login=Username&password=Password&SessionID=12345678`

Potansiyel olarak güvensiz sunucu tarafı komut dosyaları, her bir POST'u bu şekilde kontrol ederek tanımlanabilir.

Testing for Transport Vulnerabilities (Nakliye Güvenlik Açıkları için Test)

Müşteri ve Uygulama arasındaki tüm etkileşim en azından aşağıdaki kriterlere göre test edilmelidir.

- Oturum kimlikleri nasıl aktarılır? e.g., GET, POST, Form Sahası (gizli alanlar dahil)
- Oturum Kimlikleri her zaman varsayılan olarak şifreli taşıma üzerinden gönderilir mi?
- Oturum kimliklerini şifrelemeden göndermek için uygulamayı manipüle etmek mümkün mü? Ee.g., HTTPS'ye HTTP'yi değiştirerek?

- Oturum Kimliklerini geçen isteklere/responslara hangi önbellek kontrol direktifleri uygulanır?
- Bu direktifler her zaman mevcut mudur? Değilse, istisnalar nerede?
- Alın Oturum kimliğini içeren talepler kullanılır mı?
- POST kullanılırsa, GET ile kavrılabilir mi?

Refernace (Referanslar)

Whitepapers (Beyaz kağıtlar)

- RFCs 2109 & 2965 – HTTP Devlet Yönetimi Mekanizması [D. Kristol, L. Montulli]
- RFC 2616 – Hipekmet Aktarma Protokolü - HTTP/1.1