Testing Session Timeout (Oturum Zaman Aşımının Test Edilmesi)

Summary (Özet)

Bu faz testlerinde, uygulamanın kullanıcının belirli bir süre boşta kaldığında bir kullanıcıyı otomatik olarak kaydetmesini kontrol eder, aynı oturumu “yeniden kullanmanın” mümkün olmadığından ve hiçbir hassas verinin tarayıcı önbelleğinde saklanmamasını sağlar.

Tüm uygulamalar oturumlar için boşta veya hareketsizlik zaman aşımı uygulamalıdır. Bu zaman aşımı, kullanıcı tarafından herhangi bir etkinlik olmaması durumunda, belirli bir oturum ID için web uygulaması tarafından alınan son HTTP talebinden bu yana oturumu kapatma ve
geçersiz kılma, oturumu kapatma ve geçersiz kılma süresinin aktif kalması durumunda aktif kalacağı süreyi tanımlar. En uygun zaman aşımı, güvenlik (kısa zaman aşımı) ile kullanılabilirlik (daha uzun zaman aşımı) arasında bir denge olmalıdır ve büyük ölçüde uygulama tarafından
ele alınan verilerin hassasiyet seviyesine bağlıdır. Örneğin, halka açık bir forum için 60 dakikalık bir gün izin kabul edilebilir, ancak bu kadar uzun bir süre bir ev bankacılığı uygulamasında (maksimum 15 dakikalık bir zaman aşımı önerildiği) çok fazla olacaktır. Her durumda, zaman aşımına dayalı bir günlüğü uygulamayan herhangi bir uygulama, belirli bir işlevsel gereklilik tarafından gerekli olmadıkça, güvenli görülmemelidir.

Boş zaman aşımı, bir saldırganın başka bir kullanıcıdan geçerli bir oturum kimliği tahmin etme ve kullanma şansını sınırlar ve belirli koşullar altında kamu bilgisayarlarını oturum yeniden kullanımından koruyabilir. Bununla birlikte, saldırgan belirli bir oturumu kaçırabilirse, boş zaman aşımı saldırganın eylemlerini sınırlamaz, çünkü oturumu daha uzun süre aktif tutmak için periyodik olarak oturumda faaliyet oluşturabilir.

Oturum zaman aşımı yönetimi ve son kullanma sunucu tarafı uygulanmalıdır. Müşterinin kontrolü altındaki bazı veriler oturum zaman aşımını uygulamak için

kullanılırsa, örneğin zaman referanslarını izlemek için çerez değerlerini veya diğer istemci parametrelerini kullanmak için kullanılırsa (örneğin, zaman içinde oturum süresini uzatmak için dakika sayısı), bir saldırgan bunları oturum süresini uzatmak için manipüle edebilir. Bu nedenle, uygulama hareketsizlik süresini sunucunun bir tarafına izlemeli ve zaman aşımının sona ermesinden sonra, mevcut kullanıcının oturumunu otomatik olarak geçersiz kılmak ve istemcide depolanan her veriyi silmek zorundadır.

Kullanıcının uygulamadan çıkış yapmayı unutması durumunda, bir saldırgan tarafından izinsiz erişim sağlamak için istismar edilebilecek zayıflıkları ortaya çıkarmaktan kaçınmak için her iki eylem de dikkatli bir şekilde uygulanmalıdır. Daha spesifik olarak, günlük bırakma işlevine gelince, tüm oturum belirteçlerinin (örneğin çerezlerin) uygun şekilde imha edilmesini veya kullanılamaz hale getirilmesini ve oturum belirteçlerinin yeniden kullanılmasını önlemek için uygun kontrollerin sunucu tarafının zorunlu kılınmasını sağlamak önemlidir. Bu tür eylemler düzgün bir şekilde gerçekleştirilmezse, bir saldırgan meşru bir kullanıcının oturumunu "yeniden dirmek" ve onu taklit etmek için bu oturum belirteçlerini tekrar oynatabilir (bu saldırı genellikle 'çerez tekrarı' olarak bilinir). Tabii ki, hafifletici bir faktör, saldırganın bu jetonlara (kurbanın PC'sinde depolanan) erişebilmesi gerektiğidir, ancak çeşitli durumlarda bu imkansız veya özellikle zor olmayabilir.

Bu tür bir saldırı için en yaygın senaryo, bazı özel bilgilere erişmek için kullanılan halka açık bir bilgisayardır (örneğin, web postası, çevrimiçi banka hesabı). Kullanıcı açıkça çıkış yapmadan
bilgisayardan uzaklaşırsa ve oturum zaman aşımı uygulamada uygulanmazsa, bir saldırgan tarayıcının "geri" düğmesine basarak aynı hesaba erişebilir.

Test Objectives (Test Hedefleri)

- Zor bir oturum zaman aşımı olduğunu doğrulayın.

How to Test (Nasıl Test Edilir)

Black-Box Testing (Siyah-Kutu Testi)

Logout işlevselliği için Test bölümünde görülen aynı yaklaşım, zaman aşımı gününüzü ölçerken uygulanabilir.Test metodolojisi çok benzer. İlk olarak, testçiler, örneğin, giriş yaparak ve zaman aşımının tetiklenmesini bekleyerek bir zaman

aşımının olup olmadığını kontrol etmelidir. Kayıt fonksiyonunda olduğu gibi, zaman aşımı geçtikten sonra, tüm oturum belirteçleri yok edilmeli veya kullanılamaz hale getirilmelidir.

Ardından, zaman aşımı yapılandırılırsa, test cihazlarının zaman aşımının istemci tarafından mı yoksa sunucu (veya her ikisi) tarafından uygulanıp uygulanmadığını anlamaları gerekir. Oturum çerezi kalıcı değilse (veya genel olarak, oturum çerezi zaman hakkında herhangi bir veri saklamaz), testçiler zaman aşımının sunucu tarafından uygulandığını varsayabilir. Oturum çerezi bazı zamanla ilgili veriler içeriyorsa (örneğin, zamanında oturum açın veya kalıcı bir çerez için son kullanma tarihi), müşterinin zaman aşımına maruz kalma işlemine dahil olması mümkündür. Bu durumda, testçiler çerezi (kriptografi olarak korunmazsa) değiştirmeye çalışabilir ve oturuma ne olacağını görebilir. Örneğin, testçiler çerez son kullanma tarihini gelecekte ayarlayabilir ve oturumun uzayıp uzayamayacağını görebilir.

Genel bir kural olarak, her şey sunucu tarafı kontrol edilmeli ve oturum çerezlerini önceki değerlere yeniden yerleştirerek, uygulamaya tekrar erişmek mümkün olmamalıdır.

Gray-Box Testing (Gri-Kutu Testi)

Testçinin şuna işaret etmesi gerekiyor:

- Kayıt dışı fonksiyon tüm oturum belirtecini etkili bir şekilde yok eder veya en azından onları kullanılamaz hale getirir,
- Sunucu, oturum durumunda uygun kontrolleri gerçekleştirir ve birsaldırganın daha önce tahrip edilen oturum tanımlayıcılarını tekrar oynatmasına izin vermez.
- Bir zaman aşımı uygulanır ve sunucu tarafından uygun şekildeuygulanır. Sunucu, istemci tarafından gönderilen bir oturumbelirtecinden okunan bir son kullanma süresi kullanırsa (ancak bu tavsiye edilmez), o zaman jetonun kriptografik olarak kurcalamadan korunması gerekir.

En önemli şeyin, uygulamanın sunucu tarafındaki oturumu geçersiz kılmak olduğunu unutmayın. Genel olarak bu, kodun uygun yöntemleri çağırması gerektiği anlamına gelir, örneğin. `HttpSession.invalidate()` Java ve `Session.abandon()` İçeride . NET. Çerezleri tarayıcıdan temizlemek tavsiye edilir, ancak kesinlikle gerekli değildir,

çünkü oturum sunucuda uygun şekilde
geçersiz kılınırsa, tarayıcıda çerezin bir saldırgana yardımcı olmaz.

Referances (Referanslar)

OWASP Resources (OWASP Kaynakları)

- Oturum Yönetimi Hile Sayfası