Testing for Padding Oracle (Oracle Dolgu Testi)

Summary (Özet)

Bir dolgu kahini, istemci tarafından sağlanan şifreli verilerin şifresini çözen, örneğin istemcide depolanan dahili oturum durumunun ve şifre çözmeden sonra dolgunun geçerliliğini açıklayan bir uygulamanın bir işlevidir. Bir dolgu kahininin varlığı, bir saldırganın şifrelenmiş verilerin şifresini çözmesine ve bu kriptografik işlemler için kullanılan anahtar hakkında bilgi sahibi olmadan rastgele verileri şifrelemesine izin verir. Bu, şifrelenmiş verilerin bütünlüğü uygulama tarafından varsayılırsa, mantıklı verilerin açıklanmasına veya ayrıcalıklı tırmanma güvenlik açıklarına yol açabilir.

Blok şifreleri yalnızca belirli boyutlarda bloklarda verileri şifreler. Yaygın şifreler tarafından kullanılan blok boyutları 8 ve 16 bayttır. Boyutun kullanılmış şifrenin blok boyutunun birden fazlasına uymadığı veriler belirli bir şekilde doldurulmalıdır, böylece şifre çözücü dolguyu sıyırabilir. Yaygın olarak kullanılan bir dolgu şeması PKCS#7'dir. Kalan baytları dolgu uzunluğunun değeriyle doldurur.

Example 1 (Örnek 1)

Pading 5 bayt uzunluğundaysa, bayt değeri `0x05` Düz metinden sonra beş kez tekrarlanır.

Bir hata koşulu, dolgu, kullanılmış dolgu şemasının sözdizimi ile eşleşmezse mevcuttur. Bir uygulama, istemci tarafından sağlanan şifreli veriler için bu özel dolgu hatası koşulunu sızdırırsa, bir dolgu kabı bulunur. Bu, istisnaları açığa çıkararak olabilir (örneğin. `BadPaddingException` Java'da) doğrudan, müşteriye veya zamanlama davranışı gibi başka bir yan kanal tarafından gönderilen yanıtlardaki ince farklılıklarla.

Kriptografinin belirli çalışma modları, şifreli metinde biraz çevirmenin, bit çevirmenin de düz metinde çevrilmesine neden olduğu bit-fliping saldırılarına izin verir. CBC şifreli verilerin n-th bloğunda biraz dolaşmak, (n+1)-ik bloğundaki aynı parçanın şifreli verilerde çevrilmesine neden olur. Çözülmüş şifreli şifreli metnin n-th bloğu bu manipülasyon tarafından çöpe atılır.

Dolma kahin saldırısı, bir saldırganın şifreleme anahtarı hakkında bilgi sahibi olmadan şifreli verilerin şifresini çözmesini sağlar ve yopher'ı dolgu kafilesine ustaca manipüle edilmiş şifreli metinler göndererek ve onunla iade edilen sonuçları gözlemleyerek kullanır. Bu, şifreli verilerin gizliliğinin kaybına neden olur. Örneğin, istemci tarafında depolanan oturum verileri durumunda saldırgan, uygulamanın iç durumu ve yapısı hakkında bilgi alabilir.

Bir dolgu kahin saldırısı, bir saldırganın kullanılmış anahtar ve şifreyi bilmeden keyfi düz metinleri şifrelemesini de sağlar. Uygulama, şifreli verilerin bütünlüğünün ve gerçekliğinin verildiğini varsayarsa, bir saldırgan iç oturum durumunu manipüle edebilir ve muhtemelen daha yüksek ayrıcalıklar kazanabilir.

Test Objectives (Test Hedefleri)

- Padding'e dayanan şifreli mesajları belirleyin.
- Şifreli mesajların dolgusunu kırmaya ve daha fazla analiz için iade edilen hata mesajlarını analiz etmeye çalışın.

How To Test (Nasıl Test Edilir)

Black-Box Testing (Siyah-Kutu Testi)

Önce yastıklı kahrolus için olası girdi noktaları tespit edilmelidir. Genel olarak aşağıdaki koşullar yerine getirilmelidir:

1. Veriler şifrelenir. İyi adaylar rastgele görünen değerlerdir.
2. Bir blok şifre kullanılır. Çözülmüşlerin uzunluğu (Base64 sıklıkla kullanılır) şifreli metin, 8 veya 16 bayt gibi yaygın şifre blok boyutlarının bir katıdır. Farklı şifre metinleri (örneğin, farklı
   oturumlar veya oturum hallerinin manipülasyonu ile toplanmış) ortak bir bölücüyü uzun süre paylaşır.

Example 2 (Örnek 2)

`Dg6W8OiWMIdVokIDH15T/A==` Base64'ün kod çözmesinden sonra sonuçlar `0e 0e 96 f0 e8 96 30 87 55 a2 42 03 1f 5e 53 fc`. . Bu rastgele ve 16 bayte uzunluğunda görünüyor.

Böyle bir girdi değeri adayı tespit edilirse, şifrelenmiş değerin biraz akıllıca kurcalanması için uygulamanın davranışı doğrulanmalıdır. Normalde bu Base64

kodlanmış değer, şifreli metne hazırlanan başlangıç vektörünü (IV) içerecektir. Bir düz metin verildi `p` ve blok boyutuna sahip bir şifre `n`, blok sayısı olacak `b = ceil( length(b) / n)`. . Şifreli ipin uzunluğu olacak `y=(b+1)*n` Başlangıç vektörü nedeniyle. Kalayın varlığını doğrulamak için, ipi çözer, ikinci-son bloğun son parçasını çevirin `b-1` (Şehitin en az önemli kısmı `y-n-1`), yeniden kodlayın ve gönderin. Ardından, orijinal dizeyi çöz, bloğun son parçasını çevirin `b-2` (Şehitin en az önemli kısmı `y-2*n-1`), yeniden kodlayın ve gönderin.

Şifreli dizenin tek bir blok olduğu biliniyorsa (IV sunucuda saklanır veya uygulama hardcoded IV'ü kötü bir uygulama kullanıyor), sırayla birkaç parça çevirme işlemi yapılmalıdır. Alternatif bir yaklaşım, rastgele bir blok hazırlamak ve eklenen bloğun son baytını mümkün olan tüm değerleri (0 ila 255) almak için çevirme parçaları olabilir.

Testler ve taban değeri, şifre çözme sırasında ve sonrasında en az üç farklı duruma neden olmalıdır:

- Cipher metni şifresini çözer, sonuçta veriler doğrudur.
- Cipher metni şifresini çözer, sonuçta veriler giyilir ve uygulama mantığında bazı istisna veya hata işleme neden olur.
- Şifreli metin şifre çözme dolgu hataları nedeniyle başarısız olur.

Yanıtları dikkatlice karşılaştırın. Özellikle pedingde bir şeylerin yanlış olduğunu belirten istisnalar ve mesajlar için arama yapın. Bu tür mesajlar ortaya çıkarsa, uygulama bir dolgu kahini içerir. Yukarıda
açıklanan üç farklı durum örtük olarak gözlemlenebilirse (farklı hata mesajları, zamanlama yan kanalları), bu noktada bir dolgu kıvrımı olma olasılığı yüksektir. Bunu sağlamak için dolgu kahin saldırısını gerçekleştirmeye çalışın.

Example 3 (Örnek 3)

- ASP.NET atar `System.Security.Cryptography.CryptographicException: Padding is invalid and cannot be removed.` Eğer şifresi çözülmüş bir şifreli metnin dolgusu kırılırsa.
- Java a `javax.crypto.BadPaddingException` Bu davada atılır.
- Şifre çözme hataları veya benzerleri olası dolgu orakları olabilir.

> Güvenli bir uygulama bütünlüğü kontrol eder ve yalnızca iki yanıta neden olacaktır: okve failed. . İç hata durumlarını belirlemek için kullanılabilecek hiçbir yan kanal yoktur.

Gray-Box Testing (Gri-Kutu Testi)

Müşteriden şifrelenmiş verilerin, yalnızca sunucu tarafından bilinmesi gereken tüm yerlerin şifresini çözdüğünü doğrulayın. Aşağıdaki koşullar bu kodla karşılanmalıdır:

1. Şifre metninin bütünlüğü, HMAC veya GCM veya CCM gibi doğrulanmış şifreli çalışma modları gibi güvenli bir mekanizma ile doğrulanmalıdır.
2. Şifre çözme ve daha fazla işlem sırasında tüm hata durumları tek tip olarak ele alınır.

Example 4 (Örnek 4)

Visualization of the decryption process

Tools (Araçlar)

- Bletchley
- PadBuster
- Padding Oracle Exploitation Tool (POET)
- Poracle
- python-paddingoracle

References(Referanslar)

- Wikepedia - Padding Oracle Attack
- Juliano Rizzo, Thai Duong, “Practical Padding Oracle Attacks”