Mobil Uygulama Güvenliği 2023

Türkçe Çeviri Tarihi: Eylül 2023

İçindekiler

Bilgilendirme

Standart Hakkında

Uygulama güvenliği doğrulama standardı; yazılım tasarımcılar, yazılım geliştiriciler, test yapanlar, güvenlik uzmanları ve hatta müşteriler için uygulama güvenliği gereksinimlerinden oluşan bir listedir.

Copyright and License

Proje Liderleri

- Andrew van der Stock
- Daniel Cuthbert
- Jim Manico
- Josh C Grossman
- Mark Burnett

Değerlendirenler ve Katkıda Bulunanlar

- Osama Elnaggar
- Erlend Oftedal
- Serg Belkommen
- David Johansson
- Tonimir Kisasondi
- Ron Perris
- Jason Axley
- Abhay Bhargav
- Benedikt Bauer
- Elar Lang
- ScriptingXSS
- Philippe De Ryck
- Grog's Axle
- Marco Schnüriger
- Jacob Salassi
- Glenn ten Cate
- Anthony Weems
- bschach
- javixeneize
- Dan Cornell
- hello7s
- Lewis Ardern
- Jim Newman
- Stuart Gunter
- Geoff Baskwill
- Talargoni
- Ståle Pettersen
- Kelby Ludwig
- Jason Morrow
- Rogan Daw

aiseclab.org

AiSecLab Türkçe Çeviri Ekibi

Mentor: Cihan ÖZHAN
Furkan Berk KOÇOĞLU
Şevval Ayşe KENAR
Amine Nur YEŞİL

M1: Sahtekâr Kimlik Kullanımı

Tehdit Ajanları

Uygulamaya Özel

Mobil uygulamalarda sabit kodlanmış kimlik bilgilerinden ve uygunsuz kimlik bilgileri kullanımından yararlanan tehdit aracıları, kamuya açık veya özel olarak oluşturulmuş araçları kullanan otomatik saldırıları içerebilir. Bu tür ajanlar, potansiyel olarak sabit kodlanmış kimlik bilgilerini tespit edip kullanabilir veya kimlik bilgilerinin uygunsuz kullanımından kaynaklanan zayıflıklardan yararlanabilir.

Atak Vektörleri

İstismar edilebilirlik **KOLAY**

Saldırganlar hem sabit kodlanmış kimlik bilgilerindeki hem de kimlik bilgilerinin uygunsuz kullanımındaki güvenlik açıklarından yararlanabilir. Bu güvenlik açıkları belirlendikten sonra saldırgan, sabit kodlanmış kimlik bilgilerini kullanarak mobil uygulamanın hassas işlevlerine yetkisiz erişim sağlayabilir. Ayrıca kimlik bilgilerini kötüye kullanabilirler; örneğin, uygunsuz şekilde doğrulanmış veya saklanmış kimlik bilgileri yoluyla erişim elde ederek meşru erişim ihtiyacını atlayabilirler.

Güvenlik Zayıflığı

Yaygınlık **ORTAK**
Tespit Edilebilirlik **KOLAY**

Sabit kodlanmış kimlik bilgilerinin kullanılması ve uygun olmayan şekilde kullanılması gibi kimlik bilgisi yönetiminin kötü uygulanması, ciddi güvenlik zayıflıklarına yol açabilir. Kapsamlı bir güvenlik testi süreci bu sorunları tespit etmeyi amaçlamalıdır. Örneğin, güvenlik testçileri, mobil uygulamanın kaynak kodunda veya herhangi bir yapılandırma dosyasında sabit kodlanmış kimlik bilgilerini belirlemeye çalışmalıdır.

Teknik Etkiler

Etki **ŞİDDETLİ**

Yetersiz kimlik bilgisi yönetimi birçok önemli teknik etkiye yol açabilir. Yetkisiz kullanıcılar, mobil uygulamadaki veya arka uç sistemlerindeki hassas bilgilere veya işlevlere erişim sağlayabilir. Bu, veri ihlallerine, kullanıcı gizliliğinin kaybına, dolandırıcılık faaliyetlerine ve yönetim işlevlerine olası erişime yol açabilir.

Ticari Etkiler

Etki **ŞİDDETLİ**

Sabit kodlanmış kimlik bilgileri ve uygunsuz kimlik bilgileri kullanımı da dahil olmak üzere, zayıf kimlik bilgisi yönetiminin iş üzerindeki etkisi önemli olabilir:

- İtibar zarar
- Bilgi Hırsızlığı

- Sahtekarlık
- Verilere Yetkisiz Erişim

Uygunsuz Kimlik Bilgisi Kullanımına Karşı Savunmasız Mıyım?

Mobil uygulamalar sabit kodlanmış kimlik bilgileri kullandığında veya kimlik bilgileri kötüye kullanıldığında güvenli olmayan kimlik bilgisi yönetimi ortaya çıkabilir. Mobil uygulamanızın savunmasız olabileceğini gösteren bazı göstergeler şunlardır:

Sabit Kodlanmış Kimlik Bilgileri- Mobil uygulamanın, uygulamanın kaynak kodunda veya herhangi bir yapılandırma dosyasında sabit kodlanmış kimlik bilgileri içermesi, güvenlik açığının açık bir göstergesidir.

Güvenli Olmayan Kimlik Bilgisi İletimi- Kimlik bilgileri şifreleme olmadan veya güvenli olmayan kanallar aracılığıyla aktarılıyorsa bu bir güvenlik açığına işaret edebilir.

Güvenli Olmayan Kimlik Bilgisi Depolama- Mobil uygulama kullanıcı kimlik bilgilerini cihazda güvenli olmayan bir şekilde saklıyorsa bu bir güvenlik açığını temsil edebilir.

Zayıf Kullanıcı Kimlik Doğrulaması- Kullanıcı kimlik doğrulaması zayıf protokollere dayanıyorsa veya kolay atlamaya izin veriyorsa, bu bir güvenlik açığının işareti olabilir.

Uygunsuz Kimlik Bilgisi Kullanımını Nasıl Önleyebilirim?

Güvenli olmayan kimlik bilgisi yönetiminden kaçınmak, sabit kodlanmış kimlik bilgilerinin kullanılmamasını ve kullanıcı kimlik bilgilerinin doğru şekilde işlenmesini içerir.

Sabit Kodlanmış Kimlik Bilgilerini Kullanmaktan Kaçının

Sabit kodlanmış kimlik bilgileri saldırganlar tarafından kolayca keşfedilebilir ve yetkisiz kullanıcılar için kolay bir erişim noktası sağlar. Mobil uygulamanızın kodunda veya yapılandırma dosyalarında sabit kodlanmış kimlik bilgilerini kullanmaktan her zaman kaçının.

Kullanıcı Kimlik Bilgilerini Doğru Şekilde Kullanın

Kullanıcı kimlik bilgileri her zaman güvenli bir şekilde saklanmalı, iletilmeli ve kimlik doğrulaması yapılmalıdır:

- İletim sırasında kimlik bilgilerini şifreleyin.
- Kullanıcı kimlik bilgilerini cihazda saklamayın. Bunun yerine güvenli, iptal edilebilir erişim belirteçleri kullanmayı düşünün.
- Güçlü kullanıcı kimlik doğrulama protokollerini uygulayın.
- Kullanılan API anahtarlarını veya belirteçlerini düzenli olarak güncelleyin ve döndürün.

Örnek Saldırı Senaryoları

Aşağıdaki senaryolar, mobil uygulamalarda kimlik bilgilerinin uygunsuz kullanımını göstermektedir:

Senaryo 1: Sabit Kodlanmış Kimlik Bilgileri: Bir saldırgan, mobil uygulamanın kaynak kodundaki sabit kodlanmış kimlik bilgilerini keşfeder. Bu kimlik bilgilerini, uygulama veya arka uç sistemlerindeki hassas işlevlere yetkisiz erişim sağlamak için kullanırlar.

Senaryo 2: Güvenli Olmayan Kimlik Bilgisi İletimi: Bir saldırgan, mobil uygulama ile arka uç sistemleri arasında

güvenli olmayan şekilde iletilen kimlik bilgilerine müdahale eder. Ele geçirilen bu kimlik bilgilerini meşru bir kullanıcının kimliğine bürünmek ve yetkisiz erişim elde etmek için kullanırlar.

Senaryo 3: Güvenli Olmayan Kimlik Bilgisi Depolama: Bir saldırgan, kullanıcının cihazına fiziksel erişim sağlar ve depolanan kimlik bilgilerini mobil uygulamadan çıkarır. Saldırgan bu kimlik bilgilerini kullanıcının hesabına yetkisiz erişim sağlamak için kullanır.

Referanslar

OWASP

- https://owasp.org/www-project-top-ten/

Dış Kaynaklar

- https://cwe.mitre.org/

M2: Yetersiz Tedarik Zinciri Güvenliği

Tehdit Ajanları

Uygulamaya Özel

Bir saldırgan, mobil uygulama tedarik zincirindeki güvenlik açıklarından yararlanarak uygulama işlevselliğini manipüle edebilir. Örneğin, bir saldırgan, mobil uygulamanın kod tabanına kötü amaçlı kod ekleyebilir veya oluşturma işlemi sırasında arka kapıları, casus yazılımları veya diğer kötü amaçlı kodları eklemek için kodu değiştirebilir.

Bu, saldırganın verileri çalmasına, kullanıcıları gözetlemesine veya mobil cihazın kontrolünü ele geçirmesine olanak tanıyabilir. Ayrıca bir saldırgan, mobil uygulamaya veya arka uç sunuculara erişim sağlamak için üçüncü taraf yazılım kitaplıklarındaki, SDK'lardaki, satıcılardaki veya sabit kodlanmış kimlik bilgilerindeki güvenlik açıklarından yararlanabilir.

Bu, yetkisiz veri erişimine veya manipülasyonuna, hizmet reddine veya mobil uygulamanın veya cihazın tamamen ele geçirilmesine yol açabilir.

Atak Vektörleri

İstismar edilebilirlik **ORTALAMA**

Yetersiz Tedarik Zinciri güvenlik açığından yararlanmanın birden fazla yolu vardır; örneğin, içeriden gelen bir tehdit aracısı veya bir saldırgan, uygulamanın geliştirme aşaması sırasında kötü amaçlı kod enjekte edebilir, ardından kötü amaçlı kodu güvenilir olarak imzalamak için uygulama imzalama anahtarlarını veya sertifikalarını tehlikeye atabilir.

Başka bir şekilde, bir tehdit aracısı üçüncü taraf kitaplıklarındaki veya uygulamada kullanılan bileşenlerdeki güvenlik açıklarından yararlanabilir.

Zayıf Güvenlik

Yaygınlık **ORTAK**
Tespit edilebilirlik **ZOR**

Yetersiz Tedarik Zinciri güvenlik açığı, güvenli kodlama uygulamalarının eksikliği, yetersiz kod incelemeleri ve testlerin uygulamaya güvenlik açıklarının dahil edilmesine yol açması nedeniyle oluşur.

Yetersiz tedarik zinciri güvenlik açıklarının diğer nedenleri arasında yetersiz veya güvenli olmayan uygulama imzalama ve dağıtım süreci, üçüncü taraf yazılım bileşenleri veya kitaplıklarındaki zayıflık, veriler, şifreleme, depolama için yetersiz güvenlik kontrolleri veya hassas verilerin yetkisiz erişime açık hale getirilmesi yer alır.

Teknik Etkiler

Etki **ŞİDDETLİ**

Bir saldırganın yetersiz tedarik zinciri güvenliğinden başarıyla yararlanması halinde bunun teknik etkisi ciddi olabilir. Spesifik teknik etki, istismarın niteliğine bağlıdır ancak şunları içerebilir:

Veri İhlali: Saldırgan, oturum açma kimlik bilgileri, kişisel veriler veya finansal bilgiler gibi hassas verileri çalabilir. Veri ihlali, etkilenen kişiler için kimlik hırsızlığı veya mali dolandırıcılık gibi uzun vadeli sonuçlar doğurabilir.

Kötü Amaçlı Yazılım Bulaşması: Saldırgan, mobil uygulamaya, kullanıcının cihazına bulaşıp verileri çalabilecek veya kötü amaçlı faaliyetler gerçekleştirebilecek kötü amaçlı yazılım ekleyebilir. Kötü amaçlı yazılımın tespit edilmesi ve kaldırılması zor olabilir ve kullanıcının cihazına ve verilerine ciddi zarar verebilir.

Yetkisiz Erişim: Saldırgan, mobil uygulamanın sunucusuna veya kullanıcının cihazına erişim sağlayarak verileri değiştirme veya silme gibi yetkisiz faaliyetler gerçekleştirebilir. Bu, veri kaybına, hizmet kesintisine veya diğer teknik sorunlara neden olabilir.

Sistemin Ele Geçirilmesi: Saldırganın mobil uygulamanın tüm sistemini tehlikeye atması, sistem üzerindeki kontrolün tamamen kaybolmasına yol açabilir. Bu, uygulamanın kapanmasına, önemli miktarda veri kaybına ve mobil uygulama geliştiricisinin itibarının uzun vadede zarar görmesine neden olabilir.

Ticari Etkiler

Etki **ŞİDDETLİ**

Bir saldırganın yetersiz tedarik zinciri güvenliğinden başarıyla yararlanması durumunda bunun iş üzerindeki etkisi önemli olabilir. Spesifik iş etkisi, istismarın niteliğine ve kuruluşun büyüklüğüne, sektörüne ve genel güvenlik duruşuna bağlıdır ancak şunları içerebilir:

Finansal Kayıplar: Kuruluş, saldırı sonucunda ihlalin araştırılmasının maliyeti, etkilenen kişilere bildirimin maliyeti veya yasal çözümlerin maliyeti gibi mali kayıplara maruz kalabilir. Müşterilerin mobil uygulamaya olan güvenini kaybetmesi ve uygulamayı bırakması durumunda da kuruluş gelir kaybedebilir.

İtibar Hasarı: Saldırı sonucunda kuruluşun itibarı zarar görebilir ve bu durum kuruluşun markasına ve müşteri güvenine uzun vadede zarar verebilir. Bu, gelirin azalmasına ve yeni müşteri çekmenin zorlaşmasına neden olabilir.

Yasal ve Düzenleyici Sonuçlar: Kuruluş, saldırı sonucunda para cezaları, davalar veya devlet soruşturmaları gibi yasal ve düzenleyici sonuçlarla karşı karşıya kalabilir. Bu sonuçlar kuruluşta ciddi mali ve itibar kaybıyla sonuçlanabilir.

Tedarik Zincirinin Bozulması: Saldırı, kuruluşun tedarik zincirini bozabilir ve mal veya hizmetlerin teslimatında gecikmelere veya kesintilere yol açabilir. Bu, finansal kayıplara ve kuruluşun itibarının zarar görmesine neden olabilir.

'Yetersiz Tedarik Zinciri Güvenlik Açığı'na Karşı Savunmasız Mıyım?

Özellikle üçüncü taraf geliştiriciler tarafından geliştirilen mobil uygulamaları kullanıyorsanız veya üçüncü taraf kitaplıklara ve bileşenlere güveniyorsanız, yetersiz tedarik zinciri güvenlik açığına karşı savunmasız kalmanız mümkündür. Güvenlik açığı aşağıdakiler gibi çeşitli nedenlerden dolayı ortaya çıkabilir:

Üçüncü Taraf Bileşenlerde Güvenlik Eksikliği: Kitaplıklar veya çerçeveler gibi üçüncü taraf bileşenler, saldırganların yararlanabileceği güvenlik açıkları içerebilir. Mobil uygulama geliştiricisi üçüncü taraf bileşenlerini gerektiği gibi incelemez veya güncel tutmazsa uygulama saldırılara karşı savunmasız kalabilir.

Kötü Amaçlı İçeriden Tehdit: Kötü niyetli bir geliştirici veya tedarikçi gibi içerideki kötü niyetli kişiler, mobil uygulamaya kasıtlı olarak güvenlik açıkları getirebilir. Bu, geliştiricinin tedarik zinciri sürecinin yeterli güvenlik kontrollerini ve izlemesini uygulamaması durumunda ortaya çıkabilir.

Yetersiz Test ve Doğrulama: Mobil uygulama geliştiricisi uygulamayı kapsamlı bir şekilde test etmezse saldırılara açık hale gelebilir. Geliştirici ayrıca tedarik zinciri sürecinin güvenliğini doğrulamakta başarısız olabilir ve bu da uygulamada güvenlik açıklarına yol açabilir.

Güvenlik Bilincinin Eksikliği: Mobil uygulama geliştiricisi yeterli güvenlik bilincine sahip değilse tedarik zinciri saldırılarını önlemek için gerekli güvenlik kontrollerini uygulayamayabilir.

'Yetersiz Tedarik Zinciri Güvenlik Açığı'nı Nasıl Önleyebilirim?

- Güvenlik açıklarını belirlemek ve azaltmak için mobil uygulama geliştirme yaşam döngüsü boyunca güvenli kodlama uygulamaları, kod incelemesi ve testler uygulayın.
- Saldırganların kötü amaçlı kod imzalamasını ve dağıtmasını önlemek için uygulama imzalama ve dağıtım süreçlerinin güvenli olmasını sağlayın.
- Güvenlik açığı riskini azaltmak için yalnızca güvenilir ve doğrulanmış üçüncü taraf kitaplıkları veya bileşenleri kullanın.
- Saldırganların uygulamadaki güvenlik açıklarından yararlanmasını önlemek amacıyla uygulama güncellemeleri, yamalar ve sürümler için güvenlik kontrolleri oluşturun.
- Olayları zamanında tespit etmek ve bunlara yanıt vermek için güvenlik testleri, taramalar veya diğer teknikler aracılığıyla tedarik zinciri güvenliği olaylarını izleyin ve tespit edin.

Örnek Atak Senaryoları

Senaryo 1: Kötü Amaçlı Yazılım Ekleme

Bir saldırgan, geliştirme aşamasında popüler bir mobil uygulamaya kötü amaçlı yazılım enjekte eder. Saldırgan daha sonra uygulamayı geçerli bir sertifikayla imzalar ve uygulama mağazasının güvenlik kontrollerini atlayarak uygulamayı uygulama mağazasına dağıtır. Kullanıcılar, oturum açma kimlik bilgilerini ve diğer hassas verilerini çalan virüslü uygulamayı indirip yüklüyor. Saldırgan daha sonra çalınan verileri dolandırıcılık veya kimlik hırsızlığı yapmak için kullanır ve bu da kurbanlara ciddi mali zararlar verirken, uygulama sağlayıcısının itibarının zedelenmesine neden olur.

Referanslar

OWASP

- https://owasp.org/www-project-kubernetes-top-ten/2022/en/src/K02-supply-chain-vulnerabilities
- https://owasp.org/www-project-dependency-check/

Dış Kaynaklar

- https://cwe.mitre.org/

M3: Güvenli Olmayan Kimlik Doğrulama/Yetkilendirme

Tehdit Ajanları

Uygulamaya Özel

Kimlik doğrulama ve yetkilendirme güvenlik açıklarından yararlanan tehdit aracıları, bunu genellikle mevcut veya özel olarak oluşturulmuş araçları kullanan otomatik saldırılar aracılığıyla gerçekleştirir.

Atak Vektörleri

İstismar edilebilirlik **KOLAY**

Düşman, kimlik doğrulama veya yetkilendirme şemasındaki güvenlik açıklarını anladıktan sonra bu zayıflıklardan iki yoldan biriyle yararlanabilir. Hizmet isteklerini doğrudan mobil uygulamanın arka uç sunucusuna göndererek, mobil uygulama ile herhangi bir doğrudan etkileşimi atlatarak kimlik doğrulamayı taklit edebilir veya atlayabilirler veya kimlik doğrulama kontrolünü başarıyla geçtikten sonra uygulamaya meşru bir kullanıcı olarak giriş yapabilir ve ardından kimlik doğrulamayı zorunlu kılabilirler. Yönetim işlevlerini yürütmek için güvenlik açığı bulunan bir uç noktaya göz atın. Her iki yararlanma yöntemi de genellikle saldırganın sahip olduğu cihaz veya botnet'ler içindeki mobil kötü amaçlı yazılımlar aracılığıyla gerçekleştirilir.

Güvenlik Zayıflığı

Yaygınlık **ORTAK**
Tespit edilebilirlik **ORTALAMA**

Mobil uygulamalardaki zayıf yetkilendirme ve kimlik doğrulama planlarını test etmek için test uzmanları tarafından çeşitli stratejiler kullanılabilir. Yetkilendirme için, test uzmanları mobil uygulamaya karşı ikili saldırılar gerçekleştirebilir ve özellikle mobil uygulama "çevrimdışı" moddayken yalnızca daha yüksek ayrıcalıklara sahip bir kullanıcı tarafından yürütülebilmesi gereken ayrıcalıklı işlevleri yürütmeye çalışabilir. Test uzmanları ayrıca, arka uç sunucusuna yönelik hassas işlevsellik için karşılık gelen POST/GET istekleri dahilinde düşük ayrıcalıklı bir oturum belirteci kullanarak herhangi bir ayrıcalıklı işlevi yürütmeye çalışmalıdır.

Yetersiz veya eksik yetkilendirme planları, potansiyel olarak bir saldırganın, mobil uygulamanın kimliği doğrulanmış ancak daha düşük ayrıcalıklara sahip bir kullanıcıyı kullanarak yetki sahibi olmaması gereken işlevleri yürütmesine izin verebilir. Bu ayrıcalık yükseltme saldırısı riski, yetkilendirme kararları uzak bir sunucu yerine mobil cihaz içinde alındığında artar; bu, çoğunlukla çevrimdışı kullanılabilirliğin mobil gereksinimleri nedeniyle ortaya çıkabilen bir senaryodur.

Zayıf kimlik doğrulama şemaları açısından, test uzmanları, mobil uygulama "çevrimdışı" moddayken, çevrimdışı kimlik doğrulamayı atlamayı ve ardından çevrimdışı kimlik doğrulama gerektirmesi gereken işlevleri yürütmeyi amaçlayan ikili saldırılar gerçekleştirebilir. Test uzmanları ayrıca, mobil uygulama işlevselliğine yönelik herhangi bir POST/GET isteğinden oturum belirteçlerini kaldırarak herhangi bir arka uç sunucu işlevini anonim olarak yürütmeye çalışmalıdır.

Zayıf veya eksik kimlik doğrulama şemaları, bir saldırganın mobil uygulama veya mobil uygulama tarafından kullanılan arka uç sunucusu içindeki işlevleri anonim olarak yürütmesine izin verebilir. Mobil uygulama kimlik doğrulamasındaki bu zayıflıklar, mobil cihazın genellikle kısa şifreleri veya 4 haneli PIN'leri teşvik eden giriş formu faktörü nedeniyle oldukça yaygındır.

Mobil uygulamalar, büyük ölçüde değişen kullanılabilirlik gereksinimleri nedeniyle, geleneksel web kimlik doğrulama planlarından farklılaşabilen benzersiz kimlik doğrulama gereksinimleriyle karşı karşıyadır. Kullanıcıların çevrimiçi

olmasının ve bir arka uç sunucusuyla gerçek zamanlı olarak kimlik doğrulaması yapmasının beklendiği geleneksel web uygulamalarının aksine, mobil uygulamaların, mobil internet bağlantılarının güvenilmezliği veya öngörülemezliği nedeniyle çevrimdışı kimlik doğrulamayı gerektiren çalışma süresi gereksinimlerini karşılaması gerekebilir. Bu gereksinim, geliştiricilerin mobil kimlik doğrulamayı uygularken dikkate alması gereken faktörleri önemli ölçüde etkileyebilir.

Teknik Etkiler

Etki **ŞİDDETLİ**

Bir sistemdeki zayıf yetkilendirme ve kimlik doğrulamanın teknik etkisi, büyük ölçüde yürütülen aşırı ayrıcalıklı işlevsellik türüne bağlı olarak geniş kapsamlı, önemli ve benzer olabilir. Yetersiz yetkilendirme söz konusu olduğunda, örneğin uzaktan veya yerel yönetim işlevlerinin aşırı ayrıcalıkla yürütülmesi, sistemlere veya hassas bilgilere erişime zarar verebilir.

Zayıf kimlik doğrulamanın teknik yansımaları, çözümün bir eylem isteği gerçekleştiren kullanıcıyı tanımlayamadığı durumlarda ortaya çıkar. Bu, kullanıcının kimliği belirlenemediği için kullanıcı etkinliğinin hemen kayıt altına alınamamasına veya denetlenememesine neden olabilir. Bu kimlik doğrulama eksikliği, bir saldırının kaynağını tespit etme, altta yatan istismarların doğasını anlama veya gelecekteki saldırıları önlemek için stratejiler geliştirme konusundaki yetersizliğe katkıda bulunur.

Ayrıca, kimlik doğrulamadaki hatalar aynı zamanda temeldeki yetkilendirme hatalarını da ortaya çıkarabilir. Kimlik doğrulama kontrolleri başarısız olduğunda çözüm, kullanıcının rolüne ve ilgili izinlere yakından bağlı olan kullanıcının kimliğini doğrulayamaz. Bir saldırganın hassas işlevleri anonim olarak yürütebilmesi, temeldeki kodun, eylem isteğinde bulunan kullanıcının izinlerini doğrulamadığını gösterir. Sonuç olarak, kodun anonim olarak yürütülmesi hem kimlik doğrulama hem de yetkilendirme kontrollerindeki hataların altını çizer.

Ticari Etkiler

Etki **ŞİDDETLİ**

Yetersiz kimlik doğrulama ve yetkilendirmenin iş üzerindeki etkisi genellikle en azından aşağıdakilerle sonuçlanacaktır:

- İtibar Hasarı
- Bilgi Hırsızlığı
- Sahtekâr
- Verilere Yetkisiz Erişim

'Güvensiz Kimlik Doğrulama/Yetkilendirmeye' Karşı Savunmasız Mıyım?

Kimlik doğrulama ve yetkilendirme arasındaki farkı anlamak, mobil uygulama güvenliğini değerlendirmede çok önemlidir. Kimlik doğrulama, bir kişiyi tanımlarken, yetkilendirme, tanımlanan kişinin belirli bir eylem için gerekli izinlere sahip olup olmadığını doğrular. Yetkilendirme kontrollerinin mobil cihaz isteği kimlik doğrulamasını hemen takip etmesi gerektiğinden, bu iki husus birbiriyle yakından ilişkilidir.

Arayanın kimliği belirlenmeden gelen bir istek üzerinde yetkilendirme kontrolleri yapmak neredeyse imkânsız olduğundan, bir kuruluş bir mobil cihazdan talep edilen bir API uç noktasını çalıştırmadan önce bir kişinin kimliğini doğrulayamadığı zaman güvenli olmayan yetkilendirme meydana gelebilir.

Güvenli olmayan yetkilendirmenin bazı basit göstergeleri şunlardır:

- **Güvenli Olmayan Doğrudan Nesne Referansı (IDOR) güvenlik açıklarının varlığı-** Bir IDOR güvenlik açığının fark edilmesi, kodun uygun bir yetkilendirme kontrolü yapmadığını gösterebilir.
- **Gizli Uç Noktalar-** Geliştiriciler, gizli işlevselliğe yalnızca uygun role sahip bir kullanıcı tarafından erişileceğini varsayarak, arka uç gizli işlevlerine ilişkin yetkilendirme kontrollerini ihmal edebilir.
- **Kullanıcı Rolü veya İzin İletimleri-** Mobil uygulamanın, bir isteğin parçası olarak kullanıcının rollerini veya izinlerini bir arka uç sistemine aktarması durumunda, bu, güvenli olmayan bir yetkilendirme sinyali verebilir.

Benzer şekilde, mobil uygulamalar da çeşitli güvenli olmayan kimlik doğrulama belirtileri gösterebilir:

- **Anonim Arka Uç API Yürütmesi-** Uygulamanın, erişim belirteci sağlamadan bir arka uç API hizmeti isteğini yürütme yeteneği, güvenli olmayan kimlik doğrulamaya işaret edebilir.
- **Parolaların veya Paylaşılan Sırların Yerel Olarak Depolanması-** Uygulama, herhangi bir parolayı veya paylaşılan sırları cihazda yerel olarak saklıyorsa, bu, güvenli olmayan bir kimlik doğrulamanın işareti olabilir.
- **Zayıf Parola Politikası-** Basitleştirilmiş bir parola girme işleminin kullanılması, kimlik doğrulamanın güvenli olmadığı anlamına gelebilir.
- **FaceID ve TouchID gibi Özelliklerin Kullanımı-** FaceID veya TouchID gibi özelliklerin kullanılması, güvenli olmayan kimlik doğrulamanın göstergesi olabilir.

'Güvensiz Kimlik Doğrulama ve Yetkilendirmeyi' Nasıl Önleyebilirim?

Hem güvenli olmayan kimlik doğrulamayı hem de yetkilendirmeyi önlemek için zayıf kalıplardan kaçınmak ve güvenli önlemleri güçlendirmek çok önemlidir.

Zayıf Modellerden Kaçının
Güvenli Olmayan Mobil Uygulama Kimlik Doğrulama Tasarım Modellerinden kaçınılmalıdır:

- Bir web uygulamasını mobil eşdeğerine taşıyorsanız, mobil uygulamaların kimlik doğrulama gereksinimlerinin web uygulaması bileşeninin kimlik doğrulama gereksinimleriyle eşleştiğinden emin olun. Web tarayıcısından daha az faktörle kimlik doğrulaması yapmak mümkün olmamalıdır.
- Yerel kullanıcı kimlik doğrulaması, istemci tarafında atlama güvenlik açıklarına yol açabilir. Uygulama, verileri yerel olarak depoluyorsa, jailbreak'li cihazlarda, çalışma zamanı manipülasyonu veya ikili değişiklik yoluyla kimlik doğrulama rutini atlanabilir. Çevrimdışı kimlik doğrulamanın zorlayıcı bir iş gereksinimi olması durumunda, mobil uygulamaya yönelik ikili saldırıların önlenmesine ilişkin ek kılavuza başvurun.
- Mümkün olduğunda tüm kimlik doğrulama isteklerini sunucu tarafında gerçekleştirin. Başarılı kimlik doğrulamanın ardından uygulama verileri mobil cihaza yüklenecek ve uygulama verilerinin yalnızca başarılı kimlik doğrulama sonrasında kullanılabilirliği sağlanacak.
- İstemci tarafında veri depolama gerekiyorsa, kullanıcının oturum açma bilgilerinden güvenli bir şekilde elde edilen bir şifreleme anahtarını kullanarak verileri şifreleyin. Ancak ikili saldırılar yoluyla verilerin şifresinin çözülmesine ilişkin ek riskler de vardır.
- "Beni Hatırla" işlevi asla bir kullanıcının şifresini cihazda saklamamalıdır.
- Mobil uygulamalar ideal olarak, kullanıcı tarafından mobil uygulama içinde iptal edilebilecek cihaza özel bir kimlik doğrulama belirteci kullanmalıdır; böylece çalınan/kaybolan bir cihazdan kaynaklanan yetkisiz erişim riskleri azaltılır.
- Kullanıcı kimlik doğrulaması için cihaz tanımlayıcıları veya coğrafi konum da dahil olmak üzere yanıltıcı olabilecek değerleri kullanmaktan kaçının.
- Mobil uygulamalarda kalıcı kimlik doğrulama, isteğe bağlı olarak uygulanmalı ve varsayılan olarak etkinleştirilmemelidir.
- Mümkün olduğunda, kullanıcıların kimlik doğrulama şifreleri için 4 haneli PIN numaraları sağlamasına izin vermekten kaçının.

Kimlik Doğrulamayı Güçlendirin

- Geliştiriciler, tüm istemci tarafı yetkilendirme ve kimlik doğrulama kontrollerinin kötü niyetli kullanıcılar tarafından atlanabileceğini varsaymalıdır. Bu kontrollerin sunucu tarafında güçlendirilmesi kritik öneme sahiptir.
- Çevrimdışı kullanım gereksinimleri nedeniyle mobil uygulamaların yerel kimlik doğrulama veya yetkilendirme kontrolleri yapması gerekebilir. Bu gibi durumlarda geliştiriciler, yetkisiz kod değişikliklerini tespit etmek için yerel bütünlük kontrolleri yapmalıdır. İkili saldırıları tespit etme ve bunlara tepki verme konusunda ek kılavuza başvurun.
- Biyometrik olarak kilitlenmiş sırların kilidini açmak ve oturum belirteçleri gibi hassas kimlik doğrulama malzemelerini güvenli bir şekilde korumak için FaceID ve TouchID'yi kullanın.

Güvenli Olmayan Yetkilendirmenin Engellenmesi

Güvenli olmayan yetkilendirmeyi önlemek için:

- Arka uç sistemleri, kimliği doğrulanmış kullanıcının rollerini ve izinlerini bağımsız olarak doğrulamalıdır. Mobil cihazdan gelen hiçbir role veya izin bilgisine güvenmeyin.
- Tüm istemci tarafı yetkilendirmesinin atlanabileceğini varsayalım, bu nedenle mümkün olduğunda sunucu tarafı yetkilendirme kontrollerini güçlendirin.
- Mobil uygulamanın kodunda çevrimdışı yetkilendirme kontrolleri gerekiyorsa geliştiriciler, yetkisiz kod değişikliklerini tespit etmek için yerel bütünlük kontrolleri yapmalıdır.

Örnek Atak Senaryoları

Aşağıdaki senaryolar, mobil uygulamalardaki kimlik doğrulama veya yetkilendirme kontrollerinin zayıf olduğunu göstermektedir:

Senaryo 1: Gizli Hizmet İstekleri: Geliştiriciler, mobil uygulamanın işlenmek üzere arka uca gönderdiği bir hizmet isteğini yalnızca kimliği doğrulanmış kullanıcıların oluşturabileceğini varsayar. İsteğin işlenmesi sırasında sunucu kodu, gelen
isteğin bilinen bir kullanıcıyla ilişkili olduğunu doğrulamaz. Bu nedenle, saldırganlar hizmet isteklerini arka uç hizmete gönderir ve çözümün meşru kullanıcılarını etkileyen işlevleri anonim olarak yürütür.

Senaryo 2: Arayüz Güveni: Geliştiriciler, mobil uygulamalarında belirli bir işlevin varlığını yalnızca yetkili kullanıcıların görebileceğini varsayarlar. Bu nedenle, yalnızca yasal olarak yetkilendirilmiş kullanıcıların mobil cihazlarından hizmete yönelik talepte bulunabilmelerini bekliyorlar. İsteği işleyen arka uç kodu, istekle ilişkili kimliğin hizmeti yürütme
yetkisine sahip olduğunu doğrulama zahmetine girmez. Bu nedenle, saldırganlar oldukça düşük ayrıcalıklı kullanıcı hesaplarını kullanarak uzaktan yönetim işlevlerini gerçekleştirebilmektedir.

Senaryo 3: Kullanılabilirlik Gereksinimleri: Kullanılabilirlik gereksinimleri nedeniyle, mobil uygulamalar 4 haneli uzunluktaki şifrelere izin verir. Sunucu kodu, parolanın karma sürümünü doğru şekilde saklıyor. Bununla birlikte, parolanın çok kısa olması nedeniyle, bir saldırgan, gökkuşağı karma tablolarını kullanarak orijinal parolaları hızlı bir şekilde çıkarabilecektir. Sunucudaki şifre dosyasının (veya veri deposunun) güvenliği ihlal edilirse, bir saldırgan kullanıcıların şifrelerini hızlı bir şekilde tespit edebilecektir.

Senaryo 4: Güvenli Olmayan Doğrudan Nesne Referansı: Bir kullanıcı, bir aktör kimliği ve bir OAuth taşıyıcı jetonu içeren bir arka uç REST API'sine API uç noktası isteğinde bulunur. Kullanıcı, aktör kimliğini gelen URL'nin bir parçası olarak ekler ve erişim belirtecini isteğe standart bir başlık olarak ekler. Arka uç, taşıyıcı jetonun varlığını doğrular ancak taşıyıcı
jetonla ilişkili aktör kimliğini doğrulayamaz. Sonuç olarak kullanıcı, REST API isteğinin bir parçası olarak aktör kimliğini ayarlayabilir ve diğer kullanıcıların hesap bilgilerine ulaşabilir.

Senaryo 5: LDAP rollerinin iletimi: Bir kullanıcı, standart bir oAuth taşıyıcı belirtecinin yanı sıra kullanıcının ait olduğu LDAP gruplarının listesini içeren bir üstbilgi içeren bir arka uç REST API'sine API uç noktası isteğinde bulunur. Arka uç isteği taşıyıcı jetonunu doğrular ve ardından hassas işlevselliğe devam etmeden önce gelen LDAP gruplarını doğru grup üyeliği açısından inceler. Ancak arka uç sistemi, LDAP grup üyeliğinin bağımsız bir şekilde doğrulanmasını

gerçekleştirmez ve bunun yerine kullanıcıdan gelen LDAP bilgilerine güvenir. Kullanıcı, gelen başlıkta ince ayar yapabilir ve keyfi olarak herhangi bir LDAP grubunun üyesi olacak şekilde rapor verebilir ve yönetim işlevlerini gerçekleştirebilir.

Referanslar

OWASP

- https://owasp.org/www-project-top-ten/

Dış Kaynaklar

- https://cwe.mitre.org/

M4: Yetersiz Giriş/Çıkış Doğrulaması

Tehdit Ajanları

Uygulamaya Özel

Bir mobil uygulamada kullanıcı girişleri veya ağ verileri gibi harici kaynaklardan gelen verilerin yetersiz doğrulanması ve temizlenmesi, ciddi güvenlik açıklarına neden olabilir. Bu tür verileri doğru şekilde doğrulama ve temizleme konusunda başarısız olan mobil uygulamalar, SQL enjeksiyonu, Komut Enjeksiyonu ve siteler arası komut dosyası çalıştırma (XSS) saldırıları dahil olmak üzere mobil ortamlara özel saldırılar yoluyla istismar edilme riskiyle karşı karşıyadır.

Bu güvenlik açıkları, hassas verilere yetkisiz erişim, uygulama işlevselliğinin manipülasyonu ve tüm mobil sistemin potansiyel olarak tehlikeye atılması gibi zararlı sonuçlara yol açabilir.

Yetersiz çıktı doğrulaması, veri bozulmasına veya sunumda güvenlik açıklarına yol açarak, kötü niyetli aktörlerin kötü amaçlı kod yerleştirmesine veya kullanıcılara gösterilen hassas bilgileri değiştirmesine olanak tanıyabilir.

Atak Vektörleri

İstismar edilebilirlik **ZOR**

Yetersiz giriş/çıkış doğrulaması, uygulamamızı SQL enjeksiyonu, XSS, komut enjeksiyonu ve yol geçişi dahil olmak üzere kritik saldırı vektörlerine maruz bırakır. Bu güvenlik açıkları, yetkisiz erişime, veri manipülasyonuna, kod yürütülmesine ve tüm arka uç sisteminin tehlikeye atılmasına neden olabilir.

Güvenlik Zayıflığı

Yaygınlık **ORTAK**
Tespit edilebilirlik **KOLAY**

Yetersiz giriş/çıkış doğrulama güvenlik açığı, bir uygulamanın kullanıcı girişini düzgün bir şekilde kontrol edip temizlememesi veya çıktı verilerini doğrulayıp temizlememesi durumunda ortaya çıkar. Bu güvenlik açığından aşağıdaki şekillerde yararlanılabilir:

Yetersiz Giriş Doğrulaması: Kullanıcı girişi tam olarak kontrol edilmediğinde, saldırganlar beklenmedik veya kötü amaçlı veriler girerek girişleri manipüle edebilir. Bu, güvenlik önlemlerini atlayabilir ve kod yürütme güvenlik açıklarına veya yetkisiz sistem erişimine yol açabilir.

Yetersiz Çıkış Doğrulaması: Çıktı verileri uygun şekilde doğrulanmazsa ve arındırılmazsa saldırganlar, kullanıcıların tarayıcıları tarafından çalıştırılan kötü amaçlı komut dosyaları enjekte edebilir. Bu, siteler arası komut dosyası çalıştırma (XSS) saldırılarına, veri hırsızlığına, oturumun ele geçirilmesine veya görüntülenen içeriğin manipülasyonuna yol açabilir.

Bağlamsal Doğrulama Eksikliği: Belirli bağlamın veya beklenen veri formatlarının dikkate alınmaması, SQL enjeksiyonu veya format dizesi güvenlik açıkları gibi güvenlik açıklarına neden olabilir. Bunlar, doğrulanmamış kullanıcı girişinin doğrudan veritabanı sorgularına dahil edilmesi veya biçim dizesi işlevlerinde uygunsuz şekilde işlenmesiyle ortaya çıkar ve saldırganların sorguları değiştirmesine veya rastgele kod yürütmesine olanak tanır.

Veri Bütünlüğünün Doğrulanamaması: Veri bütünlüğü doğrulanmazsa uygulama, verilerin bozulmasına veya yanlış işlenmesine karşı savunmasız hale gelir. Saldırganlar, kritik sistem değişkenlerine müdahale edebilir veya uygulamanın işlevselliğini bozacak hatalı biçimlendirilmiş veriler sunabilir.

Bu güvenlik açıkları genellikle uygulama mantığındaki hatalardan, doğrulama kontrollerinin eksik uygulanmasından, güvenlik bilincinin eksikliğinden veya yetersiz test ve kod inceleme uygulamalarından kaynaklanır.

Teknik Etkiler

Etki **ŞİDDETLİ**

Yetersiz giriş/çıkış doğrulama güvenlik açığının, etkilenen uygulama üzerinde çeşitli teknik etkileri olabilir:

Kod Yürütme: Kötü niyetli bir aktör, güvenlik önlemlerini atlayarak uygulamanın ortamında yetkisiz kod yürütmek için bu güvenlik açığından yararlanabilir.

Veri İhlalleri: Yetersiz doğrulama, saldırganların girişi değiştirmesine olanak tanıyarak, potansiyel olarak yetkisiz erişime ve hassas verilerin çıkarılmasına yol açabilir.

Sistemin Ele Geçirilmesi: Saldırganlar, temeldeki sisteme yetkisiz erişim sağlayabilir, onu tehlikeye atabilir ve potansiyel olarak kontrolü ele geçirebilir.

Uygulamanın Bozulması: Kötü niyetli girdiler kesintilere, çökmelere veya veri bozulmasına neden olarak uygulamanın güvenilirliğini ve işlevselliğini etkileyebilir.

İtibar Hasarı: Bu güvenlik açığından başarıyla yararlanılması, veri ihlalleri ve müşteri güveninin kaybı nedeniyle itibarın zarar görmesine neden olabilir.

Yasal ve Uyumluluk Sorunları: Yetersiz doğrulama, yasal yükümlülüklere, düzenleyici cezalara ve veri koruma düzenlemelerine uyulmamasına yol açabilir.

Ticari Etkiler

Etki **ŞİDDETLİ**

Yetersiz giriş/çıkış doğrulama güvenlik açığının önemli teknik ve ticari sonuçları vardır. Uygulama açısından bakıldığında etkiler şunları içerir:

Kod Yürütme: Saldırganlar, yetkisiz kod yürütmek için bu güvenlik açığından yararlanabilir, bu da potansiyel olarak sistem güvenliğinin aşılmasına ve yetkisiz erişime yol açabilir.
Veri İhlalleri: Yetersiz doğrulama, saldırganların girişi değiştirmesine olanak tanıyarak veri ihlallerine ve hassas bilgilere yetkisiz erişime neden olur.
Sistem Kesintileri: Güvenlik açığından yararlanılması, uygulama çökmelerine, kararsızlığa veya veri bozulmasına neden olarak hizmet kesintilerine ve operasyonel verimsizliklere yol açabilir.
Veri Bütünlüğü Sorunları: Yetersiz doğrulama, sistemin güvenilirliğinden ve bütünlüğünden ödün vererek verilerin bozulmasına, yanlış işlenmesine veya hatalı çıktılara neden olabilir.

İş tarafında ise etkiler şunlardır:

- İtibar Hasarı: Güvenlik açığının başarılı bir şekilde kullanılması, veri ihlallerine, sistem kesintilerine ve müşteri güvensizliğine yol açarak kuruluşun itibarına ve marka imajına zarar verebilir.
- Yasal ve Uyumluluk Sonuçları: Yetersiz doğrulama nedeniyle veri koruma düzenlemelerine uyulmaması, yasal yükümlülüklere, düzenleyici cezalara ve olası mali kayıplara yol açabilir.

- Finansal Etki: Güvenlik açığından kaynaklanan veri ihlalleri veya sistem kesintileri, olay müdahalesi, iyileştirme maliyetleri, yasal ücretler ve potansiyel gelir kaybı nedeniyle mali kayıplara neden olabilir.

'Yetersiz Giriş/Çıkış Doğrulaması'na Karşı Savunmasız Mıyım?

Bir uygulama aşağıdaki nedenlerden dolayı yetersiz giriş/çıkış doğrulamasına karşı savunmasız olabilir:

- Giriş Doğrulaması Eksikliği: Kullanıcı girişinin doğru şekilde doğrulanmaması, uygulamayı SQL enjeksiyonu, komut enjeksiyonu veya XSS gibi enjeksiyon saldırılarına maruz bırakabilir.
- Yetersiz Çıkış Temizleme: Çıkış verilerinin yetersiz şekilde temizlenmesi, XSS güvenlik açıklarına yol açarak saldırganların kötü amaçlı komut dosyaları eklemesine ve yürütmesine olanak tanır.
- Bağlama Özel Doğrulamanın İhmali: Veri bağlamına dayalı belirli doğrulama gereksinimlerinin dikkate alınmaması, yol geçiş saldırıları veya dosyalara yetkisiz erişim gibi güvenlik açıkları oluşturabilir.
- Yetersiz Veri Bütünlüğü Kontrolleri: Uygun veri bütünlüğü kontrollerinin yapılmaması, verilerin bozulmasına veya yetkisiz değişikliklere yol açarak güvenilirlik ve güvenlikten ödün verilmesine neden olabilir.
- Zayıf Güvenli Kodlama Uygulamaları: Parametreli sorguların kullanılması veya verilerden kaçış/kodlama gibi güvenli kodlama uygulamalarının ihmal edilmesi, giriş/çıkış doğrulama güvenlik açıklarına katkıda bulunur.

'Yetersiz Giriş/Çıkış Doğrulamasını' Nasıl Önleyebilirim?

“Yetersiz Giriş/Çıkış Doğrulaması” güvenlik açıklarını önlemek için:

- **Giriş Doğrulaması:** Katı doğrulama tekniklerini kullanarak kullanıcı girişini doğrulayın ve sterilize edin. Giriş uzunluğu kısıtlamaları uygulayın ve beklenmeyen veya kötü amaçlı verileri reddedin.
- **Çıkış Sterilizasyonu:** Siteler arası komut dosyası çalıştırma (XSS) saldırılarını önlemek için çıktı verilerini uygun şekilde temizleyin.
- Verileri görüntülerken veya iletirken çıktı kodlama tekniklerini kullanın.
- **Bağlama Özel Doğrulama:** Yol geçişi veya ekleme gibi saldırıları önlemek için veri bağlamına (ör. dosya yüklemeleri, veritabanı sorguları) dayalı olarak özel doğrulama gerçekleştirin.
- **Veri Bütünlüğü Kontrolleri:** Veri bozulmalarını veya yetkisiz değişiklikleri tespit etmek ve önlemek için veri bütünlüğü kontrolleri uygulayın.
- **Güvenli Kodlama Uygulamaları:** SQL enjeksiyonunu önlemek için parametreli sorgular ve hazırlanmış ifadeler kullanmak gibi güvenli kodlama uygulamalarını izleyin.
- **Düzenli Güvenlik Testi:** Güvenlik açıklarını belirlemek ve gidermek için sızma testleri ve kod incelemeleri de dahil olmak üzere düzenli güvenlik değerlendirmeleri gerçekleştirin.

Örnek Atak Senaryoları

Senaryo 1: Kötü Amaçlı Giriş Yoluyla Uzaktan Kod Yürütme

Saldırgan, uygun giriş doğrulaması ve temizliği olmayan bir mobil uygulamayı tanımlar. Beklenmeyen karakterler içeren kötü amaçlı bir girdi oluşturarak uygulamanın davranışından yararlanırlar. Yetersiz doğrulama nedeniyle uygulama girişi yanlış yöneterek güvenlik açıklarına yol açar. Saldırgan, rastgele kodu başarıyla yürüterek cihazın kaynaklarına ve hassas verilere yetkisiz erişim sağlar.

Senaryo 2: Yetersiz Çıkış Doğrulaması Yoluyla Enjeksiyon Saldırıları

Saldırgan, çıktı doğrulama ve temizleme işlemlerinin yetersiz olduğu bir mobil uygulamayı tespit eder. Kullanıcı tarafından oluşturulan içeriğin veya güvenilmeyen verilerin işlendiği bir giriş noktasından yararlanırlar. Saldırgan, kod veya komut dosyaları (ör. HTML, JavaScript, SQL) içeren kötü amaçlı girdiler oluşturarak çıktı doğrulama eksikliğinden yararlanır. Hazırlanan girdiyi kullanıcı etkileşimi yoluyla gönderen uygulama, girdiyi doğrulayamıyor veya temizleyemiyor, bu da enjekte edilen kodun veya istenmeyen işlemlerin yürütülmesine izin veriyor. Saldırgan, siteler arası komut dosyası oluşturma (XSS) veya SQL enjeksiyonu gibi enjeksiyon tabanlı saldırıları başarıyla yürüterek uygulamanın bütünlüğünü tehlikeye atar ve hassas bilgilere erişim sağlar.

Senaryo 3: Hatalı Çıkış Yoluyla Uzaktan Kod Yürütme

Saldırgan, kullanıcı tarafından sağlanan verileri işleyen ve dinamik çıktı üreten bir mobil uygulamayı tanımlar. Saldırgan, uygulamanın yetersiz çıktı doğrulamasından yararlanarak özel olarak biçimlendirilmiş veriler hazırlar. Saldırgan, hatalı biçimlendirilmiş verileri doğrudan etkileşim yoluyla veya açığa çıkan bir API'den yararlanarak uygulamaya gönderir. Uygulama, oluşturulan çıktıyı düzgün bir şekilde doğrulamakta veya arındırmakta başarısız olduğundan, saldırganın hazırlanmış verilerinin kod yürütmesine veya istenmeyen eylemleri tetiklemesine olanak tanır. Saldırgan, bu güvenlik açığından yararlanarak uzaktan kod yürütmeyi başarır ve mobil cihaz, kaynakları veya hassas veriler üzerinde kontrol sahibi olur.

Referanslar

OWASP

- https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html
- https://owasp.org/www-community/vulnerabilities/Improper_Data_Validation#:~:text=Omitting%20validation%20for%20even%20a,incomplete%20or%20absent%20input%20validation

Dış Kaynaklar

- https://cwe.mitre.org/
- https://cwe.mitre.org/data/definitions/20.html

M5: Güvenli Olmayan İletişim

Tehdit Ajanları

Uygulamaya Özel

Çoğu modern mobil uygulama, bir veya daha fazla uzak sunucuyla veri alışverişinde bulunur. Veri aktarımı gerçekleştiğinde, genellikle mobil cihazın operatör ağından ve internetten geçer; kabloyu dinleyen bir tehdit aracısı, düz metin olarak veya kullanımdan kaldırılmış bir şifreleme protokolü kullanılarak iletiliyorsa verileri yakalayabilir ve değiştirebilir. Tehdit ajanlarının hassas bilgileri çalmak, casusluk yapmak, kimlik hırsızlığı ve daha fazlası gibi farklı amaçları olabilir. Aşağıdaki tehdit aracıları mevcuttur.

- Yerel ağınızı paylaşan bir düşman (güvenliği ihlal edilmiş veya izlenen Wi-Fi);
- Hileli taşıyıcı veya ağ cihazları (yönlendiriciler, baz istasyonları, proxy'ler vb.); veya
- Mobil cihazınızda kötü amaçlı yazılım.

Atak Vektörleri

İstismar edilebilirlik **KOLAY**

Modern uygulamalar SSL/TLS gibi kriptografik protokollere yanıt verirken bazen uygulamalarında aşağıdaki gibi kusurlar bulunabilir:

- Kullanımdan kaldırılmış protokollerin ve/veya hatalı yapılandırma ayarlarının kullanılması;
- Kötü SSL sertifikalarını kabul etmek (kendinden imzalı, iptal edilmiş, süresi dolmuş, yanlış ana bilgisayar…); veya
- Tutarsızlık (yalnızca kimlik doğrulama gibi belirli iş akışlarında SSL/TLS'ye sahip olmak).

Güvenlik Zayıflığı

Yaygınlık **ORTAK**
Tespit Edilebilirlik **ORTALAMA**

Modern mobil uygulamalar ağ trafiğini korumayı hedeflerken çoğu zaman uygulamalarında tutarsızlıklar yaşanmaktadır.
Bu tutarsızlıklar, verileri ve oturum kimliklerini ele geçirmeye açık hale getiren güvenlik açıklarına yol açabilir. Bir uygulamanın aktarım güvenliği protokollerini kullanması, onun doğru şekilde uygulandığı anlamına gelmez. Temel kusurları belirlemek için telefondaki ağ trafiğini gözlemleyebilirsiniz. Ancak daha ince kusurları tespit etmek, uygulamanın tasarımına ve yapılandırmasına daha yakından bakmayı gerektirir.

Teknik Etkiler

Etki **ŞİDDETLİ**

Bu kusur, hesabın ele geçirilmesine, kullanıcının kimliğine bürünülmesine, PII veri sızıntılarına ve daha fazlasına yol açabilecek kullanıcı verilerini açığa çıkarabilir; örneğin bir saldırgan, kullanıcı kimlik bilgilerini, oturumu, 2FA belirteçlerini ele geçirebilir ve bu da daha ayrıntılı saldırılara kapı açabilir.

Ticari Etkiler

Etki **ORTA**

En azından hassas verilere bir iletişim kanalı aracılığıyla müdahale edilmesi gizlilik ihlaline yol açacaktır. Kullanıcının gizliliğinin ihlali aşağıdaki sonuçlara yol açabilir:

- Kimlik Hırsızı;
- Dolandırıcılık veya
- İtibar Hasarı.

'Güvensiz İletişime' Karşı Savunmasız Mıyım?

- Bu risk, A noktasından B noktasına veri almanın ancak bunu güvenli olmayan bir şekilde gerçekleştirmenin tüm yönlerini kapsar. Mobilden mobile iletişimleri, uygulamadan sunucuya iletişimleri veya mobilden başka bir şeye iletişimleri kapsar. Bu risk, bir mobil cihazın kullanabileceği tüm iletişim teknolojilerini içerir: TCP/IP, WiFi, Bluetooth/Bluetooth-LE, NFC, ses, kızılötesi, GSM, 3G, SMS vb.
- Tüm TLS iletişim sorunları buraya gider. Tüm NFC, Bluetooth ve WiFi sorunları buraya gider.
- Öne çıkan özellikleri arasında bir tür hassas verinin paketlenmesi ve cihazın içine veya dışına iletilmesi yer alıyor. Hassas verilere bazı örnekler arasında şifreleme anahtarları, parolalar, özel kullanıcı bilgileri, hesap ayrıntıları, oturum belirteçleri, belgeler, meta veriler ve ikili dosyalar yer alır. Hassas veriler cihaza bir sunucudan geliyor olabilir, bir uygulamadan sunucuya geliyor olabilir veya cihaz ile yerel başka bir şey (ör. NFC terminali veya NFC kartı) arasında gidiyor olabilir. Bu riskin belirleyici özelliği iki cihazın varlığı ve aralarında bazı verilerin geçmesidir.
- Veriler cihazın kendisinde yerel olarak depolanıyorsa bu #Güvensiz Veridir. Oturum ayrıntıları güvenli bir şekilde iletilirse (örneğin, güçlü bir TLS bağlantısı aracılığıyla) ancak oturum tanımlayıcının kendisi kötüyse (belki tahmin edilebilir, düşük entropi vb.), o zaman bu bir iletişim sorunu değil, #Güvensiz Kimlik Doğrulama sorunudur.
- Güvenli olmayan iletişimin olağan riskleri veri bütünlüğü, veri gizliliği ve kaynak bütünlüğü ile ilgilidir. Veriler aktarım sırasında değişiklik tespit edilemeden değiştirilebiliyorsa (örneğin, ortadaki adam saldırısı yoluyla), bu, bu riskin iyi bir örneğidir. Gizli veriler, iletişimler gerçekleştiği anda gözlemlenerek (ör. gizlice dinlemek) veya görüşmeyi gerçekleşirken kaydederek ve daha sonra ona saldırarak (çevrimdışı saldırı) açığa çıkarılabilir, öğrenilebilir veya elde edilebilirse, bu da güvenli olmayan bir iletişim sorunudur. Bir TLS bağlantısının düzgün şekilde kurulamaması ve doğrulanamaması (örneğin, sertifika kontrolü, zayıf şifreler, diğer TLS yapılandırma sorunları) güvensiz iletişimde ortaya çıkar.

'Güvensiz İletişimi' Nasıl Önleyebilirim?

Genel En İyi Uygulamalar

- Ağ katmanının güvenli olmadığını ve gizlice dinlenmeye açık olduğunu varsayalım.
- Mobil uygulamanın verileri bir arka uç API'sine veya web hizmetine iletmek için kullanacağı taşıma kanallarına SSL/TLS uygulayın.
- Bir uygulama tarayıcı/web kiti aracılığıyla bir rutin çalıştırdığında, üçüncü taraf analiz şirketleri, sosyal ağlar vb. gibi dış kuruluşların SSL sürümlerini kullanarak hesap oluşturun.
- Kullanıcının oturum kimliğini açığa çıkarabileceğinden karışık SSL oturumlarından kaçının.
- Uygun anahtar uzunluklarına sahip güçlü, endüstri standardı şifre paketlerini kullanın.
- Güvenilir bir CA sağlayıcısı tarafından imzalanan sertifikaları kullanın.
- Kötü sertifikalara (kendinden imzalı, süresi dolmuş, güvenilmeyen kök, iptal edilmiş, yanlış ana makine..) asla izin vermeyin.
- Sertifika sabitlemeyi düşünün.
- Her zaman SSL zinciri doğrulamasını zorunlu tutun.

- Yalnızca anahtarlıktaki güvenilir sertifikaları kullanarak uç nokta sunucusunun kimliğini doğruladıktan sonra güvenli bir bağlantı kurun.
- Mobil uygulama geçersiz bir sertifika tespit ederse kullanıcıları kullanıcı arayüzü aracılığıyla uyarın.
- Hassas verileri alternatif kanallar (ör. SMS, MMS veya bildirimler) üzerinden göndermeyin.
- Mümkünse, hassas verilere SSL kanalına verilmeden önce ayrı bir şifreleme katmanı uygulayın. SSL uygulamasında gelecekteki güvenlik açıklarının keşfedilmesi durumunda, şifrelenen veriler gizlilik ihlaline karşı ikincil bir savunma sağlayacaktır.
- Geliştirme döngüleri sırasında, güvenilmeyen sertifikalara izin vermek için SSL doğrulama yöntemlerini geçersiz kılmaktan kaçının; bunun yerine kendinden imzalı sertifikalar veya yerel bir geliştirme sertifikası yetkilisi (CA) kullanmayı deneyin.
- Güvenlik değerlendirmeleri sırasında, herhangi bir trafiğin düz metin kanallarından geçip geçmediğini görmek için uygulama trafiğinin analiz edilmesi önerilir.

IOS'a Özel En İyi Uygulamalar

IOS'un en son sürümündeki varsayılan sınıflar, SSL şifre gücü anlaşmasını çok iyi yönetir. Geliştiriciler, geliştirme engellerini aşmak için bu varsayılanları atlamak üzere geçici olarak kod eklediğinde sorun ortaya çıkar. Yukarıdaki genel uygulamalara ek olarak:

- Sertifikaların geçerli olduğundan ve başarısız bir şekilde kapatıldığından emin olun.
- CFNetwork'ü kullanırken, güvenilir istemci sertifikalarını belirlemek için Güvenli Aktarım API'sini kullanmayı düşünün. Neredeyse tüm durumlarda, daha yüksek standart şifreleme gücü için NSStreamSocketSecurityLevelTLSv1 kullanılmalıdır.
- Geliştirme sonrasında, tüm NSURL çağrılarının (veya NSURL sarmalayıcılarının), NSURL sınıf yöntemi setAllowsAnyHTTPSCertificate gibi kendinden imzalı veya geçersiz sertifikalara izin vermediğinden emin olun.
- Aşağıdakileri yaparak sertifika sabitlemeyi kullanmayı düşünün: Sertifikanızı dışa aktarın, uygulama paketinize ekleyin ve güven nesnenize bağlayın. NSURL yöntemini kullanarak Connection:willSendRequestForAuthenticationChallenge: artık sertifikanızı kabul edecek.

Android'e Özel En İyi Uygulamalar

- Uygulamanın org.Apache.http.conn.ssl.AllowAllHostnameVerifier veya SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER gibi tüm sertifikaları kabul etmesine izin verebilecek geliştirme döngüsünden sonra tüm kodları kaldırın. Bunlar tüm sertifikalara güvenmeye eşdeğerdir.
- SSLSocketFactory'yi genişleten bir sınıf kullanılıyorsa, sunucu sertifikasının doğru şekilde kontrol edilmesi için checkServerTrusted yönteminin doğru şekilde uygulandığından emin olun.
- Geçersiz SSL sertifikalarına izin vermek için onReceivedSslError değerini geçersiz kılmaktan kaçının.

Örnek Atak Senaryoları

Sızma test uzmanlarının bir mobil uygulamanın iletişim güvenliğini incelerken sıklıkla keşfettiği birkaç yaygın senaryo vardır:

Sertifika denetiminin olmaması: Mobil uygulama ve uç nokta, güvenli bir kanal oluşturmak için başarıyla bağlanır ve TLS anlaşması gerçekleştirir. Ancak mobil uygulama, sunucunun sunduğu sertifikayı incelemez ve mobil uygulama, sunucunun kendisine sunduğu sertifikayı koşulsuz olarak kabul eder. Bu, mobil uygulama ile uç nokta arasındaki her türlü karşılıklı kimlik doğrulama özelliğini yok eder. Mobil uygulama, TLS proxy'si aracılığıyla ortadaki adam saldırılarına karşı hassastır.

Zayıf el sıkışma anlaşması: Mobil uygulama ve uç nokta, bağlantı anlaşmasının bir parçası olarak bir şifre paketine başarılı bir şekilde bağlanır ve üzerinde anlaşmaya varır. İstemci, düşman tarafından kolayca çözülebilecek zayıf şifrelemeyle sonuçlanan zayıf bir şifre paketi kullanmak için sunucuyla başarılı bir şekilde anlaşır. Bu, mobil uygulama ile uç nokta arasındaki kanalın gizliliğini tehlikeye atar.

Gizlilik bilgileri sızıntısı: Mobil uygulama, kişisel olarak tanımlanabilir bilgileri SSL/TLS yerine güvenli olmayan kanallar aracılığıyla bir uç noktaya iletir. Bu, mobil uygulama ile uç nokta arasındaki gizlilikle ilgili verilerin gizliliğini tehlikeye atar.

Kimlik bilgileri sızıntısı: Mobil uygulama, kullanıcı kimlik bilgilerini SSL/TLS yerine güvenli olmayan kanallar aracılığıyla bir uç noktaya iletir. Bu, bir saldırganın bu kimlik bilgilerini açık metin olarak ele geçirmesine olanak tanır.

İki Faktörlü kimlik doğrulamayı atlama: Mobil uygulama, SSL/TLS yerine güvenli olmayan kanallar aracılığıyla bir uç noktadan bir oturum tanımlayıcısı alır. Bu, saldırganın ele geçirilen oturum tanımlayıcısını kullanarak iki faktörlü kimlik doğrulamayı atlamasına olanak tanır.

Referanslar

OWASP

- https://owasp.org/

Dış Kaynaklar

- https://cwe.mitre.org/

M6: Yetersiz Gizlilik Kontrolleri

Tehdit Unsurları

Uygulamaya Özel

Gizlilik kontrolleri, isimler ve adresler, kredi kartı bilgileri, e-posta ve IP adresleri, sağlık, din, cinsellik ve siyasi görüşlerle ilgili bilgiler gibi Kişisel Olarak Tanımlanabilir Bilgilerin (Personally Identifiable Information (PII)) korunmasıyla ilgilidir.

Bu bilgiler saldırganlar için çeşitli nedenlerden dolayı değerlidir. Örneğin, bir saldırgan şunları yapabilir:

- Dolandırıcılık yapmak için kurbanın kimliğine bürünmek,
- Kurbanın ödeme verilerini kötüye kullanmak,
- Kurbana hassas bilgilerle şantaj yapmak veya
- Kurbanın kritik verilerini yok ederek veya manipüle ederek kurbana zarar vermek.

Genel olarak, PII sızdırılabilir (bir başka deyişle, gizlilik ihlali), manipüle edilebilir (bütünlük ihlali) veya yok edilebilir/engellenebilir (kullanılabilirlik ihlali).

Saldırı Vektörleri

İstismar Edilebilirlik **ORTALAMA**

PII için tipik kaynaklar iyi korunur, örneğin uygulamanın sanal alanı, sunucu ile ağ iletişimi, uygulamanın günlükleri ve yedekleri. URL sorgu parametreleri ve pano içeriği gibi diğer bazı kaynaklar daha az korumaya sahiptir ancak yine de erişilmesi zordur.

Dolayısıyla PII elde etmek için saldırganın önce başka bir seviyede güvenliği ihlal etmesi gerekir. Saldırganlar ağ iletişimini gizlice dinleyebilir, dosya sistemine, panoya veya günlüklere bir trojan ile erişebilir veya mobil cihazı ele geçirip analiz etmek için bir yedek oluşturabilir. PII, mobil cihazlarda mevcut olan tüm yollarla depolanabilen, işlenebilen ve iletilebilen bir veri olduğundan, onu çıkarma veya manipüle etme olasılıkları çok çeşitlidir.

Güvenlik Zafiyetleri

Görülme sıklığı **YAYGIN**
Tespit edilebilirlik **KOLAY**

Neredeyse tüm uygulamalar bir tür PII işlemektedir. Hatta birçoğu kendi amaçlarını yerine getirmek için ihtiyaç duyduklarından daha fazlasını toplar ve işler, bu da onları iş ihtiyaçları olmayan bir hedef olarak daha çekici hale getirir.
Gizlilik ihlali riskleri, geliştiriciler tarafından PII'nin dikkatsizce ele alınması nedeniyle artmaktadır. PII her zaman bir saldırganın iletişim ve depolama ortamına erişebileceği ihtimali göz önünde bulundurularak işlenmelidir.
Bu nedenle, topladığı bazı kişisel veriler bir saldırganı yeterince güvenli olmayan bir depolama veya iletim ortamı aracılığıyla bu verileri manipüle etmeye veya kötüye kullanmaya sevk edebiliyorsa, o uygulama gizlilik ihlallerine karşı savunmasızdır.

Teknik Etkiler

Etki **DÜŞÜK**

Gizlilik ihlallerinin genellikle sistem üzerinde bir bütün olarak çok az teknik etkisi vardır. Yalnızca PII kimlik doğrulama verileri gibi bilgiler içeriyorsa, izlenebilirlik gibi belirli bazı evrensel güvenlik özelliklerini etkileyebilir.

Kullanıcı verileri manipüle edilirse, bu durum sistemi o kullanıcı için kullanılamaz hale getirebilir. Uygun sanitizasyon ve hata işleme eksikse, kötü biçimlendirilmiş veriler yoluyla backend de bozulabilir.

Ticari Etkiler

Etki **ŞİDDETLİ**

Bir gizlilik ihlalinin sahip olduğu iş etkisinin kapsamı ve ciddiyeti, etkilenen kullanıcıların sayısına, etkilenen verilerin kritikliğine ve ihlalin gerçekleştiği yerde geçerli olan veri koruma düzenlemelerine büyük ölçüde bağlıdır. Gizlilik ihlallerinin iş üzerindeki etkisi genellikle asgari olarak aşağıdakilerle sonuçlanacaktır:

Yasal düzenlemelerin ihlali: Düzenlemeler, gizlilik kontrolleriyle ilgili en büyük sorundur. GDPR (Avrupa), CCPA (Kaliforniya, ABD), PDPA (Singapur), PIPEDA (Kanada), LGPD (Brezilya), 2018 Veri Koruma Yasası (İngiltere), POPIA (Güney Afrika), PDPL (Çin), kullanıcılarının verilerini korumayan şirketlere karşı bilinen yaptırımlara sahip ilgili düzenlemelere örnektir.

Mağdurların açtığı davalar nedeniyle maddi zarar: Bir gizlilik ihlalinden kişisel olarak etkilenen herkes, ihlalin gerçekleşmesine izin veren uygulama sağlayıcısına dava açabilir. Bu davalar, geçerli yasal düzenlemelere ve sağlayıcının yeterli ve güncel koruma mekanizmalarına sahip olduğunu gösterme becerisine bağlı olarak başarılı olabilir.

İtibar kaybı: Bir gizlilik ihlali kullanıcıları büyük ölçekte etkiliyorsa, muhtemelen medyada yayınlanır ve böylece uygulamanın sağlayıcısı için olumsuz bir kamuoyu oluşturur. Sonuç olarak, uygulamanın ve hatta aynı sağlayıcının diğer ilgisiz ürünlerinin satışları ve kullanımı düşebilir.

PII'nin kaybolması veya çalınması: Çalınan gerçek bilgiler, uygulamanın sağlayıcısına yönelik saldırılar için bile kötüye kullanılabilir. Örneğin, belirli kullanıcı verileri, bir kurbanı taklit ederek sağlayıcıya sosyal mühendislik saldırısı yapmak için kullanılabilir.

'Yetersiz Gizlilik Kontrollerine' Karşı Savunmasız Mıyım?

Bir uygulama yalnızca kişisel olarak tanımlanabilir bilgilerin bir türünü işliyorsa Yetersiz Gizlilik Kontrollerine karşı savunmasız olabilir. Bu durum neredeyse her zaman söz konusudur: İstemci uygulamaların bir sunucu tarafından görülebilen IP adresleri, uygulamaların kullanım günlükleri ve çökme raporları veya analizlerle birlikte gönderilen meta veriler çoğu uygulama için geçerli olan PII'dır. Genellikle, bir uygulama kullanıcılarından hesaplar, ödeme verileri, konumlar ve daha fazlası gibi ek, daha hassas PII toplar ve işler.

PII kullanan bir uygulama, diğer hassas veriler gibi bu verileri de ifşa edebilir. Bu en çok şu yollarla gerçekleşir:

- Güvensiz veri depolama ve iletişim (bkz. M5, M9),
- Güvensiz kimlik doğrulama ve yetkilendirme ile veri erişimi (bkz. M3, M1) ve
- Uygulamanın kum havuzuna içeriden yapılan saldırılar (bkz. M2, M4, M8).
- Diğer OWASP Mobile Top 10 riskleri, bir uygulamanın farklı saldırı vektörlerine karşı nasıl savunmasız olabileceği hakkında daha derin bilgiler sağlar.

(M5, M9 gibi ifadeler bölümleri işaret etmektedir.)

'Yetersiz Gizlilik Kontrollerini' Nasıl Önleyebilirim?

Var olmayan bir şeye saldırılamaz, bu nedenle gizlilik ihlallerini önlemek için en güvenli yaklaşım, işlenen PII miktarını ve çeşitliliğini en aza indirmektir. Bu, belirli bir uygulamadaki tüm PII varlıkları hakkında tam farkındalık gerektirir. Bu farkındalıkla birlikte aşağıdaki sorular değerlendirilmelidir:

- İşlenen tüm PII gerçekten gerekli mi, örneğin isim ve adres, cinsiyet, yaş?
- PII'nin bir kısmı daha az kritik bilgilerle değiştirilebilir mi, örneğin ince taneli konum kaba taneli konumla değiştirilebilir mi?
- PII'nin bir kısmı azaltılabilir mi, örneğin her dakika yerine her saat konum güncellemeleri?
- PII'nin bir kısmı anonimleştirilebilir veya bulanıklaştırılabilir mi, örneğin hashleme, kova oluşturma veya gürültü ekleme yoluyla?
- PII'nin bir kısmı belirli bir süre geçtikten sonra silinebilir mi, örneğin sadece son haftanın sağlık verileri saklanabilir mi?
- Kullanıcılar isteğe bağlı PII kullanımına rıza gösterebilir mi, örneğin daha iyi bir hizmet almak ama aynı zamanda ek riskin farkında olmak için?

Geriye kalan PII kesinlikle gerekli olmadıkça saklanmamalı veya aktarılmamalıdır. Depolanması veya aktarılması gerekiyorsa, erişim uygun kimlik doğrulama ve muhtemelen yetkilendirme ile korunmalıdır. Ayrıca özellikle kritik veriler için derinlemesine savunma düşünülmelidir. Örneğin, sağlık verileri uygulamanın kum havuzunda saklanmasının yanı sıra cihazın TPM'sinde mühürlenmiş bir anahtarla şifrelenebilir. Dolayısıyla, bir saldırgan sandbox kısıtlamalarını aşmayı başarırsa, veriler yine de okunamaz. Diğer OWASP Mobile Top 10 riskleri, hassas verilerin güvenli bir şekilde saklanması, aktarılması, erişilmesi ve başka şekillerde işlenmesi için önlemler önermektedir.

Tehdit modellemesi, belirli bir uygulamada gizlilik ihlallerinin meydana gelebileceği en olası yolları belirlemek için kullanılabilir. PII'yi güvence altına alma çabası daha sonra bunlara odaklanabilir.

Statik ve dinamik güvenlik kontrol araçları, hassas verilerin kaydedilmesi veya panoya ya da URL sorgu parametrelerine sızdırılması gibi yaygın tuzakları ortaya çıkarabilir.

Örnek Saldırı Senaryoları

Aşağıdaki senaryolar mobil uygulamalardaki yetersiz gizlilik kontrollerini göstermektedir:

Senaryo 1: Logların ve hata mesajlarının yetersiz temizlenmesi.

Logların ve hata mesajlarının raporlanması, verimli bir uygulamanın kalite güvencesi için çok önemlidir. Crash raporları ve diğer kullanım verileri, geliştiricilerin hataları düzeltmelerine ve uygulamalarının nasıl kullanıldığını öğrenmelerine yardımcı olur. Ancak, geliştiriciler bu verileri günlük veya hata mesajlarına dahil etmeyi seçerse günlükler ve hata mesajları PII içerebilir. Ayrıca, üçüncü taraf kütüphaneler de hata mesajlarında ve günlüklerinde PII içerebilir. Sorgunun veya sonucun bir kısmını ortaya çıkaran veri tabanı hataları sık karşılaşılan sorunlara bir örnektir. Bu, büyük olasılıkla çökme raporlarını toplamak ve değerlendirmek için kullanılan herhangi bir platform sağlayıcısı tarafından görülebilir. Hata ekranda görüntüleniyorsa kullanıcı tarafından ya da cihaz günlüklerini okuyabilen saldırganlar tarafından da görülebilir. Geliştiriciler özellikle ne kaydettikleri konusunda dikkatli olmalı ve hata mesajlarının kullanıcıya gösterilmeden veya bir sunucuya raporlanmadan önce temizlendiğinden emin olmalıdır.
(hata = exception)

Senaryo 2: URL sorgu parametrelerinde PII kullanımı.

URL sorgu parametreleri genellikle istek argümanlarını bir sunucuya iletmek için kullanılır. Bununla birlikte, URL sorgu parametreleri en azından sunucu loglarında, ancak genellikle web sitesi analizlerinde ve muhtemelen yerel tarayıcı geçmişinde de görülebilir. Bu nedenle hassas bilgiler asla sorgu parametreleri olarak iletilmemelidir. Bunun yerine, bir başlık veya gövdenin bir parçası olarak gönderilmelidirler.

Senaryo 3: Yedeklemelerde kişisel verilerin hariç tutulması/ hasFragileUserData ayarının yapılmaması.

Bir uygulama tarafından işlenen çoğu PII, kum havuzunda saklanır. Uygulama, cihaz yedeklemelerine hangi verilerin dahil edileceğini açıkça yapılandırmalıdır. Bir saldırgan bir cihazı ele geçirip bir yedek oluşturabilir veya başka bir kaynaktan bir yedek alabilir ve bu yedekten sanal alan içeriği çıkarılabilir.
Alternatif olarak, Android'de hasFragileUserData'yı 'true' olarak ayarlayarak, bir uygulama kaldırıldıktan sonra verilerini koruyabilir. Daha sonra aynı paket kimliği ile kötü amaçlı bir uygulama yüklemeyi başaran bir saldırgan bu verilere erişebilir.
Bu nedenle, geliştiricilerin niyetini şeffaf hale getirmek ve yedeklemeler yoluyla veya bir uygulamanın sonraki yüklemeleri arasında bilgi akışını kontrol etmek için her iki ayar da uygulamalar için açıkça ayarlanmalıdır.

Referanslar

OWASP

- User Privacy Protection - OWASP Cheat Sheet Series
- Mobile App User Privacy Protection - OWASP Mobile Application Security
- OWASP Top 10 Privacy Risks | OWASP Foundation
- OWASP Top 10 for Large Language Model Applications | OWASP Foundation
- OWASP Top 10 for Large Language Model Applications | OWASP Foundation

Dış Kaynaklar

- General Data Protection Regulation (GDPR) Compliance Guidelines

M7: Yetersiz Binary Koruma

Tehdit Unsurları

Uygulamaya Özel

Uygulama binary'lerini hedef alan saldırganlar çeşitli nedenlerle harekete geçerler.
Binary, bir saldırganın kötüye kullanabileceği ticari API anahtarları veya sabit kodlanmış kriptografik gizler gibi değerli gizli bilgiler içerebilir. Buna ek olarak, binary içindeki kod, örneğin önemli iş mantığını veya önceden eğitilmiş yapay zeka modellerini içerdiği için kendi başına değerli olabilir. Bazı saldırganlar da uygulamanın kendisini hedef almaz, ancak bir saldırıya hazırlanmak için ilgili arka ucun potansiyel zayıflıklarını keşfetmek için kullanabilir.
Saldırganlar bilgi toplamanın yanı sıra, ücretli özelliklere ücretsiz olarak erişmek veya diğer güvenlik kontrollerini atlamak için uygulama binary'lerini de manipüle edebilir. En kötü durumda, popüler uygulamalar kötü amaçlı kod içerecek şekilde değiştirilebilir ve şüphelenmeyen kullanıcılardan yararlanmak için üçüncü taraf uygulama mağazaları aracılığıyla veya yeni bir adla dağıtılabilir. Yaygın bir saldırı örneğinde, bir uygulamadaki ödeme tanımlayıcıları yeniden yapılandırılır, yeniden paketlenir ve uygulama mağazaları aracılığıyla dağıtılır. Ardından, kullanıcılar bu yetkisiz kopyayı uygulama mağazasından indirdiğinde, ödemeleri orijinal sağlayıcı yerine saldırgan alır.

Saldırı Vektörleri

İstismar Edilebilirlik **KOLAY**

Uygulama binary'leri genellikle uygulama mağazalarından indirilebilir veya mobil cihazlardan kopyalanabilir, bu nedenle binary saldırıları kurmak kolaydır.
Bir uygulama binary'si iki tür saldırıya maruz kalabilir:

- **Tersine mühendislik:** Uygulama binary'si derlenir ve gizli anahtarlar, algoritmalar veya güvenlik açıkları gibi değerli bilgiler için taranır.
- **Kod değiştirme:** Uygulama ikilisi, örneğin lisans kontrollerini kaldırmak, ödeme duvarlarını aşmak veya kullanıcı olarak başka avantajlar elde etmek için manipüle edilir. Alternatif olarak, uygulama kötü amaçlı kod içerecek şekilde manipüle edilebilir.

Güvenlik Zafiyetleri

Görülme sıklığı **YAYGIN**
Tespit edilebilirlik **KOLAY**

Tüm uygulamalar binary saldırılarına karşı savunmasızdır ve birçoğu günün birinde bir çeşit saldırıya maruz kalacaktır. Binary'lerinde hassas veriler veya algoritmalar kodlanmış olan uygulamalar binary saldırılara karşı özellikle savunmasızdır. Bu uygulamalar, potansiyel saldırganları saldırganın vazgeçeceği kadar uzun süre savuşturmak için gerekli karşı önlemleri almalıdır çünkü korumayı başarılı bir şekilde kırmanın maliyeti, bu başarıdan elde edilecek kazançtan daha pahalı olacaktır. Çoğu zaman, örneğin kopya koruması durumunda, uygulama satışlarından hedeflenen gelire ulaşılana kadar kırma işlemini uzatmak yeterlidir.

Tüm uygulamalar binary saldırılarına karşı savunmasızdır ve birçoğu günün birinde bir çeşit saldırıya maruz kalacaktır. Binary'lerinde hassas veriler veya algoritmalar kodlanmış olan uygulamalar binary saldırılara karşı özellikle savunmasızdır. Bu uygulamalar, potansiyel saldırganları saldırganın vazgeçeceği kadar uzun süre savuşturmak için gerekli karşı önlemleri

almalıdır çünkü korumayı başarılı bir şekilde kırmanın maliyeti, bu başarıdan elde edilecek kazançtan daha pahalı olacaktır. Çoğu zaman, örneğin kopya koruması durumunda, uygulama satışlarından hedeflenen gelire ulaşılana kadar kırma işlemini uzatmak yeterlidir.

Genel olarak, iOS uygulamaları gibi tamamen derlenmiş uygulamalar, Android uygulamalarında bulunan üst düzey bayt koduna kıyasla tersine mühendisliğe ve kod değiştirmeye daha az duyarlıdır (bu durumun PWA veya Flutter gibi platformlar arası teknolojilerle geliştirilen uygulamalar için geçerli olmayabileceğini unutmayın).

Özellikle popüler uygulamaların manipüle edilmesi ve uygulama mağazaları aracılığıyla yeniden dağıtılması muhtemeldir. Bu manipüle edilmiş kopyaları tespit etme ve kaldırma işlemi uzman şirketler tarafından sunulmaktadır, ancak uygulamalar içinde belirli tespit ve raporlama mekanizmaları ile de mümkündür.

Binary saldırıları önlemek için tamamen güvenilir mekanizmalar olmadığını unutmayın. Bunlara karşı savunma, karşı önlemlere yatırım yapan geliştiriciler ile bu önlemleri kıran saldırganlar arasında bir silahlanma yarışıdır. Dolayısıyla, her uygulama için cevaplanması gereken soru şudur: Binary saldırılara karşı önlem almak için ne kadar çaba gösterilmelidir?

Teknik Etkiler

Etki **ORTA**

Daha önce de belirtildiği gibi, bir binary saldırı ya tersine mühendislik olarak gerçekleşip uygulama binary'sinden bilgi sızdırabilir ya da kod kurcalama olarak gerçekleşip uygulamanın çalışma şeklini değiştirebilir.

Gizli bilgiler sızarsa, sistem genelinde hızlı bir şekilde değiştirilmeleri gerekir ki gizli bilgiler uygulamada sabit kodlanmışsa bu zordur. Binary'den bilgi sızması da backend'deki güvenlik açıklarını ortaya çıkarma potansiyeline sahiptir.

Bununla birlikte, manipülasyonun bir sistemin teknik sağlamlığı üzerinde daha da fazla etkisi vardır. Saldırganlar binary'leri manipüle ederek uygulamaların çalışma şeklini keyfi olarak değiştirebilir, örneğin kendi çıkarları için veya bu tür kötü niyetli taleplere karşı yeterince güçlendirilmemişlerse backend'leri rahatsız etmek için.

Ticari Etkiler

Etki **ORTA**

Ticari API'ler veya benzerleri için API anahtarlarının sızdırılması, büyük ölçekte kötüye kullanılmaları halinde önemli maliyetlere neden olabilir. Aynı durum, lisans kontrollerini kaldırmak veya işlevlerini rakip bir uygulama ile yayınlamak için kurcalanan uygulamalar için de geçerlidir. Her iki durumda da, bir uygulamayı kıran veya kişisel kullanım için bir API anahtarı çalan kişiler muhtemelen fark edilmeyecektir. Ancak büyük ölçekte, örneğin API anahtarları ve hatta işlevsellik diğer uygulamalarla sistematik olarak kullanıldığında, kötü niyetli rakipler önemli ölçüde daha düşük maliyetlere sahip oldukları için önemli bir avantaj elde edebilirler.

Büyük çabalarla geliştirilen algoritmalar veya yapay zeka modelleri gibi fikri mülkiyetin kamuya açık hale gelmesi veya kötü niyetli bir rakip tarafından çalınması durumunda uygulama geliştiricilerinin iş modeli daha da fazla tehdit altında kalabilir.

Özellikle kötü niyetli kodlarla yeniden dağıtılan popüler uygulamalar için büyük bir itibar kaybı ortaya çıkabilir. Uygulama sağlayıcısı, uygulamasının tahrif edilmiş bir kopyasının yeniden dağıtılmasını pek engelleyemese de olumsuz kamuoyu muhtemelen orijinal sağlayıcıya yönelecektir. Bu nedenle, bu riskin olasılığını azaltmak için yetkisiz kopyaların yeniden dağıtımı bir saldırgan için mümkün olduğunca zorlaştırılmalıdır.

' Yetersiz Binary Korumaya' Karşı Savunmasız Mıyım?

- Tüm uygulamalar binary saldırılara karşı savunmasızdır. Eğer uygulamanın binary'sinde hassas veriler veya algoritmalar kodlanmışsa ya da uygulama çok popülerse binary saldırılar özellikle zararlı hale gelebilir. Perdeleme, sırların yerel kodda kodlanması (Android için) veya benzeri ek koruyucu önlemler varsa, başarılı saldırıların gerçekleştirilmesi zorlaşır, ancak asla imkansız hale gelmez.
- Uygulamanın yeterince güvenli olup olmadığı, farklı binary saldırıların yaratabileceği iş etkisine bağlıdır. Saldırganlar için ne kadar teşvik ediciyse ve etkisi ne kadar büyük olacaksa, koruma için o kadar fazla çaba gösterilmelidir. Dolayısıyla, binary saldırılara karşı "güvenlik açığı" söz konusu uygulamaya özeldir.
- Hızlı bir kontrol için, geliştiriciler saldırganların kullanacağı benzer araçları kullanarak kendi uygulama binarylerini inceleyebilirler. MobSF, otool, apktool ve Ghidra gibi kullanımı oldukça kolay ve iyi belgelenmiş birçok ücretsiz veya uygun fiyatlı araç bulunmaktadır.

'Yetersiz Binary Korumasını' Nasıl Önleyebilirim?

Her uygulama için, binary'de herhangi bir kritik içerik bulunup bulunmadığı veya popülaritesinin binary korumasını zorunlu kılıp kılmadığı değerlendirilmelidir. Eğer evet ise, bir tehdit modelleme analizi en yüksek riskleri ve bunların gerçekleşmesi durumunda beklenen finansal etkilerini belirlemeye yardımcı olur. En ilişkili riskler için karşı önlemler alınmalıdır.

Uygulamalar her zaman güvenli olmayan yürütme ortamlarında çalıştırılır ve bu bilgiler her zaman sızdırılma veya manipüle edilme riski altında olduğundan, yalnızca çalışmak için ihtiyaç duydukları en az gerekli bilgileri almalıdır. Ancak belirli gizli bilgilerin, algoritmaların, güvenlik kontrollerinin ve benzerlerinin uygulamanın binary'sinde olması gerektiğini varsayarsak, farklı saldırılar farklı yollarla savuşturulabilir:

Tersine mühendislik: Tersine mühendisliği önlemek için uygulama binary'si anlaşılmaz hale getirilmelidir. Bu, birçok ücretsiz ve ticari perdeleme aracı tarafından desteklenmektedir. Uygulamaların bir kısmını yerel olarak derlemek (iOS ve Android) veya yorumlayıcılar ya da iç içe geçmiş sanal makineler kullanmak, tersine mühendisliği daha da zorlaştırır, çünkü birçok kod çözme aracı yalnızca bir dili ve binary formatını destekler. Bu tür bir gizleme, kodun karmaşıklığı ile tersine mühendisliğe karşı sağlamlık arasında bir değiş tokuştur, çünkü koddaki belirli dizelere veya sembollere dayanan birçok kütüphane tam gizlemeyle çalışmayacaktır. Geliştiriciler, önceki bölümdeki araçları kullanarak gizleme işlemlerinin kalitesini kontrol edebilirler.

Güvenlik mekanizmalarının kırılması: Gizleme aynı zamanda manipülasyona karşı da yardımcı olur, çünkü bir saldırganın güvenlik kontrollerini ve benzerlerini atlamak için kontrol akışını anlaması gerekir. Buna ek olarak, yerel güvenlik kontrolleri de backend tarafından zorlanmalıdır. Örneğin, korunan bir özellik için gerekli kaynaklar yalnızca bir kontrol yerel olarak ve backend'de başarılı olursa indirilmelidir. Son olarak, bütünlük kontrolleri kod değişikliklerini tespit edebilir ve örneğin bazı kaynakları silerek uygulama yüklemesini kullanılamaz hale getirebilir. Ancak, böyle bir bütünlük kontrolü de diğer yerel güvenlik kontrolleri gibi bulunabilir ve devre dışı bırakılabilir.

Yeniden dağıtım (kötü amaçlı kod ile): Bütünlük kontrolleri, örneğin başlangıçta, uygulama binary'lerinin yeniden dağıtımını ve değiştirilmesini de tespit edebilir. Bu ihlaller, yaygınlaşmadan önce uygulamanın yetkisiz kopyalarını bulmak ve uygulama mağazalarından kaldırmak için otomatik olarak bildirilebilir. Bu kullanım durumunu destekleyen uzman şirketler de vardır.

Örnek Saldırı Senaryoları

Senaryo 1 Sabit kodlanmış API anahtarları:

Bir uygulamanın her çağrı için küçük bir ücret ödemesi gereken ticari bir API kullandığını varsayalım. Bu çağrılar, kullanıcıların bu uygulama için ödediği abonelik ücreti ile kolayca ödenebilir. Ancak erişim ve faturalandırma için kullanılan API anahtarı, uygulamanın korumasız binary kodunda sabit kodlanmıştır. Erişim isteyen bir saldırgan, ücretsiz araçlarla uygulamada tersine mühendislik yapabilir ve gizli dizeye erişebilir. API erişimi yalnızca API anahtarı ile korunduğundan ve ek kullanıcı kimlik doğrulaması olmadığından, saldırgan API üzerinde serbestçe çalışabilir ve hatta API anahtarını satabilir. En kötü durumda, API anahtarları çok fazla kötüye kullanılabilir, uygulamanın sağlayıcısına önemli mali zarar verebilir veya en azından API erişimi oran sınırlıysa uygulamanın yasal kullanıcılarını engelleyebilir.

Senaryo 2: Ödeme ve lisans kontrollerinin devre dışı bırakılması:

Bir mobil oyun, uygulamasını ve ilk seviyelerini ücretsiz olarak yayınlayabilir. Kullanıcılar oyunu beğenirse, tam erişim için ödeme yaparlar. Daha sonraki seviyeler için tüm kaynaklar uygulama ile gönderilir. Bunlar yalnızca kullanıcı ödeme yaptığında lisansın indirildiği bir lisans kontrolü ile korunur. Bir saldırgan uygulamada tersine mühendislik yapabilir ve ödeme doğrulamasının nasıl gerçekleştiğini anlamaya çalışabilir. Uygulama binary'si yeterince korunmuyorsa, lisans kontrolünü bulmak ve onu statik bir başarı ifadesiyle değiştirmek kolaydır. Saldırgan daha sonra uygulamayı yeniden derleyebilir ve ücretsiz olarak oynatabilir veya hatta uygulama mağazalarında başka bir adla satabilir.

Senaryo 3: Hardcoded AI modelleri:

Konuşma veya serbest metin girdileri olarak verilen kullanıcı isteklerini yanıtlamak için bir yapay zeka içeren bir tıbbi uygulama varsayalım. Bu uygulama, çevrimdışı erişim sağlamak ve kendi indirme sunucularını barındırmaktan kaçınmak için özel ve kalite güvenceli yapay zeka modelini kaynak koduna dahil eder. Bu yapay zeka modeli, bu uygulamanın en değerli varlığıdır ve geliştirilmesi birçok kişinin yılını almıştır. Bir saldırgan bu modeli kaynak kodundan çıkarmaya ve rakiplerine satmaya çalışabilir. Uygulama binary'si yeterince korunmuyorsa, saldırgan yalnızca yapay zeka modeline erişmekle kalmaz, aynı zamanda nasıl kullanıldığını da öğrenebilir ve bu bilgileri yapay zeka eğitim parametreleriyle birlikte satabilir.

Referanslar

OWASP

- Mobile App Tampering and Reverse Engineering - OWASP Mobile Application Security
- iOS Tampering and Reverse Engineering - OWASP Mobile Application Security
- Android Tampering and Reverse Engineering - OWASP Mobile Application Security
- OWASP Reverse Engineering and Code Modification Prevention Project - OWASP
- OWASP Top 10 for Large Language Model Applications | OWASP Foundation

Dış Kaynaklar

- CWE - Common Weakness Enumeration (mitre.org)

M8: Hatalı Güvenlik Yapılandırmaları

Tehdit Unsurları

Uygulamaya Özel

Mobil uygulamalardaki hatalı güvenlik yapılandırmaları, güvenlik açıklarına ve yetkisiz erişime yol açabilecek güvenlik ayarlarının, izinlerin ve kontrollerin yanlış yapılandırılması anlamına gelir. Hatalı güvenlik yapılandırmalarından faydalanabilecek tehdit unsurları, hassas verilere yetkisiz erişim elde etmeyi veya kötü niyetli eylemler gerçekleştirmeyi amaçlayan saldırganlardır. Tehdit ajanları, cihaza fiziksel erişimi olan bir saldırgan, hedef savunmasız uygulama bağlamında yetkisiz eylemler yürütmek için cihazdaki hatalı güvenlik yapılandırmasından yararlanan kötü amaçlı bir uygulama olabilir.

Saldırı Vektörleri

İstismar edilebilirlik **ZOR**

Mobil uygulamalardaki hatalı güvenlik yapılandırmaları, aşağıdakiler de dahil olmak üzere çeşitli saldırı vektörleri aracılığıyla istismar edilebilir:

- Güvensiz varsayılan ayarlar: Mobil uygulamalar genellikle varsayılan yapılandırmalarla gelir ve bu yapılandırmalarda zayıf güvenlik ayarları veya gereksiz izinler etkin olabilir, bu da onları saldırılara karşı savunmasız hale getirir.
- Uygunsuz erişim kontrolleri: Hatalı yapılandırılmış erişim kontrolleri, yetkisiz kullanıcıların hassas verilere erişmesine veya ayrıcalıklı eylemler gerçekleştirmesine izin verebilir.
- Zayıf şifreleme veya hash: Yanlış uygulanan veya zayıf şifreleme ve hash algoritmaları hassas bilgilere erişim sağlamak için kullanılabilir.
- Güvenli iletişim eksikliği: SSL/TLS gibi güvenli iletişim protokollerinin kullanılmaması, hassas verileri gizli dinleme ve ortadaki adam saldırılarına maruz bırakabilir.
- Korumasız depolama: Hassas verilerin (parolalar veya API anahtarları gibi) düz metin veya zayıf şifreleme ile güvenli olmayan bir şekilde saklanması yetkisiz erişime yol açabilir.
- Güvensiz dosya izinleri: Uygulama dosyalarının herkes tarafından okunabilir ve/veya herkes tarafından yazılabilir izinlerle saklanması.
- Yanlış yapılandırılmış oturum yönetimi: Oturum yönetiminin yanlış yapılması, saldırganların meşru kullanıcıların kimliğine bürünmesine olanak tanıyarak oturumların ele geçirilmesine neden olabilir.

Güvenlik Zafiyetleri

Yaygınlık **ORTAK**
Tespit edilebilirlik KOLAY

Zaman kısıtlamaları, farkındalık eksikliği veya geliştirme sırasında insan hatası gibi faktörler nedeniyle mobil uygulamalarda hatalı güvenlik yapılandırmaları yaygındır. Manuel kod incelemesi, güvenlik testi veya otomatik tarama araçlarıyla güvenlik hatalarını tespit etmek nispeten kolaydır.
Hatalı güvenlik yapılandırmalarına örnek olarak şunlar verilebilir:

- Sürüm yapılarında hassas bilgileri açığa çıkarabilecek hata ayıklama özelliklerinin devre dışı bırakılmaması.
- HTTPS üzerinden güvenli iletişimi zorlamak yerine HTTP gibi güvensiz iletişim protokollerine izin verilmesi.

- Saldırganlara kolay erişim sağlayan varsayılan kullanıcı adları ve parolaların değiştirilmeden bırakılması.
- Yetkisiz kullanıcıların ayrıcalıklı eylemler gerçekleştirmesine izin veren yetersiz erişim kontrolleri.

Teknik Etkiler

Etki **ŞİDDETLİ**

Güvenlikle ilgili yanlış yapılandırmaların mobil uygulamalar üzerinde aşağıdakiler de dahil olmak üzere önemli teknik etkileri olabilir:

- Hassas verilere yetkisiz erişim: Yanlış yapılandırmalar saldırganların kullanıcı kimlik bilgileri, kişisel veriler veya gizli iş verileri gibi hassas bilgilere erişmesine izin verebilir.
- Hesap ele geçirme veya kimlik taklidi: Zayıf veya yanlış yapılandırılmış kimlik doğrulama mekanizmaları hesapların ele geçirilmesine veya meşru kullanıcıların kimliklerine bürünülmesine yol açabilir.
- Veri ihlalleri: Yetersiz güvenlik yapılandırmaları veri ihlallerine yol açarak hassas verilerin yetkisiz kişilerin eline geçmesine neden olabilir.
- Backend sistemlerinin tehlikeye girmesi: Mobil uygulamadaki yanlış yapılandırmalar, saldırganlara backend sistemlerini veya altyapısını tehlikeye atmaları için bir zemin sağlayabilir.

Ticari Etkiler

Etki **ŞİDDETLİ**

Hatalı güvenlik yapılandırmalarının aşağıdakiler de dahil olmak üzere ciddi ticari etkileri olabilir:

- Finansal kayıp: Yanlış güvenlik yapılandırmalarından kaynaklanan ihlaller, yasal cezalar, düzenleyici para cezaları ve kuruluşun itibarının zedelenmesi dahil olmak üzere mali kayıplara yol açabilir.
- Veri kaybı veya hırsızlığı: Yanlış yapılandırmalar hassas verilerin kaybolmasına veya çalınmasına neden olarak yasal ve mali sonuçlara yol açabilir.
- Kesinti ve aksama: Hatalı güvenlik yapılandırmalarının istismar edilmesi, kullanıcı deneyimini ve iş operasyonlarını etkileyerek uygulamanın kapalı kalma süresine, hizmet kesintisine veya işlevselliğin tehlikeye girmesine neden olabilir.
- Marka itibarının zarar görmesi: Kamuoyuna açıklanan güvenlik olayları kurumun itibarına zarar vererek müşteri güveninin kaybolmasına ve potansiyel iş kaybına yol açabilir.

Hatalı Güvenlik Yapılandırmalarına Karşı Savunmasız Mıyım?

Mobil uygulamalar, güvenlikle ilgili en iyi pratiklere uyacak şekilde düzgün yapılandırılmamışsa, hatalı güvenlik yapılandırmalarına karşı savunmasızdır. Yanlış güvenlik yapılandırmalarına karşı savunmasızlığın yaygın göstergeleri şunlardır:

- İncelenmemiş varsayılan ayarlar: Güvenlik ayarları, izinler ve varsayılan kimlik bilgileri gözden geçirilmeden varsayılan yapılandırmaların kullanılması.
- Güvenli iletişim eksikliği: Şifrelenmemiş veya zayıf şifrelenmiş iletişim kanallarının kullanılması.
- Zayıf veya eksik erişim kontrolleri: Hassas işlevlere veya verilere yetkisiz erişime izin verilmesi.
- Güncelleme veya yama yapılmaması: Uygulamaya veya temel bileşenlere gerekli güvenlik güncellemelerinin veya yamaların uygulanmaması.
- Hassas verilerin uygunsuz depolanması: Hassas verilerin düz metin veya zayıf korumalı formatlarda saklanması.

- Güvensiz dosya sağlayıcı yolu ayarları: Dahili uygulama kullanımı için olan bir dosya içerik sağlayıcısının diğer uygulamalara veya kullanıcılara açık olması, hassas verileri tehlikeye atabilir veya uygulama kaynaklarına yetkisiz erişime izin verebilir.
- Dışa aktarılan etkinlikler: Dahili uygulama kullanımı için olan bir etkinlik dışa aktarılır ve/veya taranabilir, bu da ek bir saldırı yüzeyi ortaya çıkarır.

Uygulamanızın hatalı güvenlik yapılandırmalarına karşı savunmasız olup olmadığını belirlemek için kod incelemesi, güvenlik testi ve yapılandırma analizi dahil olmak üzere kapsamlı bir güvenlik değerlendirmesi yapmalısınız.

Hatalı Güvenlik Yapılandırmalarını Nasıl Önleyebilirim?

Güvenli kodlama ve yapılandırma uygulamalarının takip edilmesi, mobil uygulamalardaki hatalı güvenlik yapılandırmalarının önüne geçilmesini sağlar. İşte bazı temel önlemler:

- Güvenli varsayılan yapılandırmalar: Varsayılan ayarların ve yapılandırmaların uygun şekilde güvence altına alındığından ve hassas bilgileri açığa çıkarmadığından veya gereksiz izinler sağlamadığından emin olun.
- Varsayılan kimlik bilgileri: Sabit kodlanmış varsayılan kimlik bilgilerini kullanmaktan kaçının.
- Güvensiz izinler: Uygulama dosyalarını world-readable ve/veya world-writable gibi aşırı izinli izinlerle depolamaktan kaçının.
- En az ayrıcalık ilkesi: Yalnızca uygulamanın düzgün çalışması için gerekli izinleri talep edin.
- Güvenli ağ yapılandırması: Açık metin trafiğine izin vermeyin ve mümkün olduğunda sertifika sabitleme kullanın.
- Hata Ayıklamayı Devre Dışı Bırak: Uygulamanın üretim sürümünde hata ayıklama özelliklerini devre dışı bırakın.
- Yedekleme modunu devre dışı bırak (Android): Android cihazlarda yedekleme modunu devre dışı bırakarak, uygulama verilerinin cihazın yedeklemesine dahil edilmesini önler ve uygulamadaki hassas verilerin cihaz yedeklemesinde saklanmamasını sağlarsınız.
- Yalnızca dışa aktarılması gereken etkinlikleri, içerik sağlayıcıları ve hizmetleri dışa aktararak uygulama saldırı yüzeyini sınırlayın.

Örnek Saldırı Senaryoları

Aşağıdaki senaryolar mobil uygulamalardaki hatalı güvenlik yapılandırmalarını göstermektedir:

Senaryo 1: Güvensiz varsayılan ayarlar. Bir mobil uygulama, zayıf güvenlik yapılandırmalarının etkin olduğu varsayılan ayarlarla yayınlanır. Bu, güvenli olmayan iletişim protokollerinin kullanılmasını, varsayılan kullanıcı adlarının ve parolaların değiştirilmeden bırakılmasını ve sürüm yapılarında hata ayıklama özelliklerinin devre dışı bırakılmamasını içerir. Saldırganlar, hassas verilere yetkisiz erişim elde etmek veya kötü amaçlı eylemler gerçekleştirmek için bu yanlış yapılandırmalardan yararlanır.

Senaryo 2: Güvensiz dosya sağlayıcı yolu ayarları. Bir mobil uygulama, dışa aktarılan bir dosya içerik sağlayıcısında root yolunu açığa çıkararak diğer uygulamaların kaynaklarına erişmesine izin verir.

Senaryo 3: Depolama izinlerine aşırı yetki verilmesi. Bir mobil uygulama, uygulama paylaşılan tercihlerini herkes tarafından okunabilir izinlerle depolayarak diğer uygulamaların bunları okumasına izin verir.

Senaryo 4: Dışa aktarılan etkinlik. Bir mobil uygulama, dahili kullanım için olan bazı etkinlikleri dışa aktararak saldırganlara uygulama için ekstra saldırı yüzeyi sağlar.

Senaryo 5: Gereksiz izinler. Bir mobil uygulama, temel işlevselliği için gerekli olmayan aşırı izinler talep eder. Örneğin, basit bir el feneri uygulaması kullanıcının rehberine, konumuna ve kamerasına erişim talep eder. Bu durum, uygulama verilen izinleri kötüye kullanabileceğinden veya hassas bilgileri istemeden sızdırabileceğinden kullanıcı verilerini gereksiz risklere maruz bırakır.

Referanslar

OWASP

- API8:2023 Security Misconfiguration - OWASP API Security Top 10
- A05 Security Misconfiguration - OWASP Top 10:2021

Dış Kaynaklar

- CWE - Common Weakness Enumeration (mitre.org)

M9: Güvensiz Veri Depolama

Tehdit Unsurları

Uygulamaya Özel

Bir mobil uygulamada güvensiz veri depolama, güvenlik açıklarından yararlanmayı ve hassas bilgilere yetkisiz erişim sağlamayı amaçlayan çeşitli tehdit unsurlarını cezbedebilir. Bu tehdit unsurları arasında değerli verileri elde etmek için mobil uygulamaları hedef alan yetenekli düşmanlar, ayrıcalıklarını kötüye kullanan kuruluş veya uygulama geliştirme ekibi içindeki kötü niyetli kişiler, siber casusluk yapan devlet destekli aktörler, veri hırsızlığı veya fidye yoluyla mali kazanç elde etmek isteyen siber suçlular, basit saldırılar için önceden oluşturulmuş araçları kullanan script kiddie'ler, kişisel bilgileri satmak için güvensiz depolamadan yararlanmak isteyen veri simsarları, rekabet avantajı elde etmeyi amaçlayan rakipler ve endüstriyel casuslar ve ideolojik güdüleri olan aktivistler veya hacktivistler yer almaktadır.

Bu tehdit unsurları zayıf şifreleme, yetersiz veri koruması, güvensiz veri depolama mekanizmaları ve kullanıcı kimlik bilgilerinin uygunsuz kullanımı gibi güvenlik açıklarından faydalanmaktadır. Mobil uygulama geliştiricileri ve kuruluşların, güvensiz veri depolama ile ilişkili riskleri azaltmak için sağlam şifreleme, güvenli veri depolama uygulamaları ve mobil uygulama güvenliği için en iyi uygulamalara bağlılık gibi güçlü güvenlik önlemleri almaları çok önemlidir.

Saldırı vektörleri

İstismar edilebilirlik **KOLAY**

Bir mobil uygulamadaki güvensiz veri depolama, kötü niyetli aktörlerin yararlanabileceği çeşitli saldırı vektörlerine karşı güvenlik açıklarını ortaya çıkarır. Saldırı vektörleri arasında cihazın dosya sistemine fiziksel veya uzaktan yetkisiz erişim, zayıf şifreleme veya eksikliğinden yararlanma, veri iletimlerini engelleme ve cihazda yüklü kötü amaçlı yazılım veya kötü amaçlı uygulamalardan yararlanma yer alır. Ayrıca, root edilmiş veya jailbreak yapılmış cihazlar saldırganlara güvenlik önlemlerini aşma ve hassas verilere doğrudan erişim sağlama fırsatı sunar. Diğer saldırı vektörleri arasında kullanıcıları kandırarak verilerine erişim sağlamaya veya uygulamanın davranışını manipüle etmeye yönelik sosyal mühendislik teknikleri yer almaktadır.

Genel olarak, bir mobil uygulamada güvensiz veri depolama, doğrudan veri çıkarmadan hassas bilgilerin ele geçirilmesine kadar çeşitli saldırılara yol açmakta ve mobil uygulama geliştirmede sağlam şifreleme, güvenli iletim protokolleri ve kapsamlı güvenlik önlemlerine duyulan kritik ihtiyacı vurgulamaktadır.

Güvenlik Zafiyetleri

Yaygınlık **ORTAK**
Tespit Edilebilirlik **ORTALAMA**

Bir mobil uygulamada güvensiz veri depolama, depolanan bilgilerin gizliliğini ve bütünlüğünü tehlikeye atabilecek çeşitli güvenlik zayıflıklarını kapsar. Bu zayıflıklar arasında, saldırganların hassas verilere kolayca erişmesine ve deşifre etmesine olanak tanıyan zayıf veya mevcut olmayan şifrelemenin kullanılması yer alır. Ayrıca, verilerin cihazın dosya sistemi içinde düz metin dosyaları veya korumasız veritabanları gibi kolay erişilebilir yerlerde saklanması, yetkisiz ayıklama veya manipülasyona maruz kalmasına neden olur. Yetersiz erişim kontrolleri ve kullanıcı kimlik doğrulama mekanizmaları sorunu daha da karmaşık hale getirerek yetkisiz kişilerin hassas verilere erişim sağlamasına olanak tanır.
Ayrıca, güvenli veri aktarım protokollerinin yokluğu, verileri mobil uygulama ile harici sunucular arasındaki iletişim sırasında ele geçirilmeye karşı savunmasız bırakmaktadır. Mobil uygulama veri depolamasındaki bu güvenlik zayıflıkları,

veri ihlalleri, yetkisiz erişim ve veri tahrifatı için fırsatlar yaratmakta ve bu riskleri azaltmak için sağlam şifreleme, güvenli depolama uygulamaları ve sıkı erişim kontrollerine duyulan kritik ihtiyacı vurgulamaktadır.

Teknik Etkiler

Etki **ŞİDDETLİ**

Bir mobil uygulamada güvensiz veri depolama, uygulamanın genel güvenliğini ve işlevselliğini zayıflatan önemli teknik etkilere sahip olabilir. Bu etkiler şunlardır:

- Veri ihlalleri: Güvensiz veri depolama, hassas bilgileri yetkisiz erişime ve veri ihlallerine açık hale getirir. Saldırganlar hassas verileri çıkarmak veya manipüle etmek için güvenlik açıklarından faydalanabilir, bu da potansiyel gizlilik ihlallerine ve gizli bilgi kaybına yol açabilir.

- Güvenliği ihlal edilmiş kullanıcı hesapları: Yetersiz veri depolama uygulamaları kullanıcı hesaplarının tehlikeye girmesine neden olabilir. Saldırganlar oturum açma kimlik bilgilerine veya güvenli olmayan bir şekilde saklanan kişisel bilgilere erişim sağlayarak yetkisiz hesap erişimine, kimlik hırsızlığına veya kullanıcı adına yetkisiz faaliyetlere yol açabilir.

- Veri tahrifatı ve bütünlük sorunları: Uygun veri koruma önlemleri olmadan, saldırganlar depolanan verileri değiştirebilir veya kurcalayabilir. Bu durum veri bütünlüğü sorunlarına, hatalı bilgilere veya uygulamanın veri depolarına kötü amaçlı içerik eklenmesine yol açabilir.

- Uygulama kaynaklarına yetkisiz erişim: Güvensiz veri depolama, saldırganların kritik uygulama kaynaklarına yetkisiz erişim elde etmesini sağlayabilir. Bu, uygulama içinde depolanan hassas dosyaları, yapılandırma dosyalarını veya kriptografik anahtarları içerir; bunlar uygulamanın işlevselliğini tehlikeye atmak veya temel sistemlerini istismar etmek için kullanılabilir.

- İtibar ve güven zedelenmesi: Bir uygulamanın güvenli olmayan veri depolama alanına sahip olduğu tespit edilirse, bu durum uygulama geliştiricisinin veya kuruluşun itibarına ve güvenine ciddi zarar verebilir. Kullanıcılar uygulamanın güvenliğine olan güvenlerini kaybedebilir, bu da kullanıcı tarafından benimsenmenin azalmasına ve potansiyel yasal ve düzenleyici sonuçlara yol açabilir.

- Uyumluluk ihlalleri: Güvensiz veri depolama, sektör düzenlemelerine ve veri koruma ile ilgili standartlara uyulmamasına yol açabilir. Uygulama geliştiricileri, kullanıcı verilerini yeterince koruyamaz ve güvenli veri depolama uygulamalarını sürdüremezlerse cezalara veya yasal işlemlere maruz kalabilirler.

Ticari Etkiler

Etki **ŞİDDETLİ**

Bir mobil uygulamada güvensiz veri depolamanın ticari etkisi önemli ve geniş kapsamlı olabilir. Aşağıda bazı temel ticari etkiler yer almaktadır:

İtibara zarar: Güvensiz veri depolama, veri ihlallerine ve kullanıcı hesaplarının tehlikeye girmesine yol açarak kuruluşun itibarına ve güvenine ciddi zarar verebilir. Veri ihlallerine ilişkin haberler hızla yayılabilir ve negatif reklama, müşteri memnuniyetsizliğine ve potansiyel iş kaybına neden olabilir.

Müşteri güveninin kaybı: Veri depolamanın güvenli olmaması nedeniyle hassas müşteri verileri tehlikeye girdiğinde, müşteriler kuruluşun bilgilerini koruma becerisine olan güvenlerini kaybedebilir. Bu güven kaybı müşteri sadakatinin azalmasına, müşteri kaybının artmasına ve genel müşteri memnuniyeti üzerinde olumsuz bir etkiye yol açabilir.

Yasal ve düzenleyici sonuçlar: Yetersiz veri depolama faaliyetleri, sektörel düzenlemelere ve veri koruma yasalarına uyulmamasına neden olabilir. Kuruluşlar, kullanıcı verilerini yeterince koruyamadıkları için para cezaları, cezalar veya davalar dahil olmak üzere yasal yansımalarla karşılaşabilir. Uyumluluk ihlalleri aynı zamanda kuruluşun itibarına ve müşteriler ile iş ortaklarının gözündeki güvenilirliğine de zarar verebilir.

Mali sonuçlar: Veri ihlalleri ve bunun sonucunda ortaya çıkan sonuçlar kuruluşlar için önemli mali sonuçlar doğurabilir. Bu, ihlalin araştırılması, etkilenen müşterilerin bilgilendirilmesi, kimlik hırsızlığına karşı koruma hizmetlerinin sağlanması, olası yasal anlaşmalar ve iş fırsatlarının kaybedilmesiyle ilgili maliyetleri içerir.

Rekabet dezavantajı: Günümüzün son derece rekabetçi ortamında, veri ihlalleri yaşayan veya güvensiz veri depolama ününe sahip kuruluşlar rekabet dezavantajıyla karşı karşıya kalabilir. Müşteriler verilerinin güvenliği konusunda giderek daha fazla endişe duyarlar ve hassas bilgileri koruma konusunda daha iyi bir geçmişe sahip olan rakipleri tercih edebilirler.

'Güvensiz Veri Depolamaya' Karşı Savunmasız Mıyım?

Bir mobil uygulamada güvensiz veri depolama ve istenmeyen veri sızıntısı çeşitli şekillerde ortaya çıkabilir ve olası gizlilik ihlallerine ve hassas bilgilere yetkisiz erişime yol açabilir. Aşağıda bu sorunların yaygın belirtileri yer almaktadır:

Erişim Kontrollerinin Eksikliği: Uygulama içindeki yetersiz erişim kontrolleri, yetkisiz kullanıcıların veya saldırganların cihazda veya uygulamanın veri tabanlarında depolanan hassas verilere erişmesine izin verebilir.

Yetersiz Şifreleme: Hassas verilerin düzgün bir şekilde şifrelenmemesi, bir saldırganın depolama konumuna erişim kazanması durumunda istenmeyen veri sızıntısına neden olabilir. Şifreleme olmadan veriler kolayca okunabilir ve istismar edilebilir.

İstemeden Verilerin Açığa Çıkması: Mobil uygulamalar hassas verileri uygulama logları, hata mesajları veya hata ayıklama özellikleri yoluyla istemeden açığa çıkarabilir ve yetkisiz kişilerin hassas bilgileri görüntülemesine veya yakalamasına izin verebilir.

Yetersiz Oturum Yönetimi: Yetersiz oturum yönetimi istenmeyen veri sızıntılarına yol açabilir. Oturum belirteçleri veya kullanıcı kimlik doğrulama bilgileri yeterince korunmaz veya yönetilmezse, hassas verilere yetkisiz erişime izin verecek şekilde ele geçirilebilir veya manipüle edilebilir.

Eksik Girdi Doğrulama: Veri doğrulama ve veri temizlemenin yetersiz olması istenmeyen veri sızıntılarına yol açabilir. Saldırganlar bu zayıflıktan faydalanarak kötü niyetli scriptler enjekte edebilir veya girdi alanlarını manipüle ederek hassas verilere ulaşabilir.

Bulut Depolama Hatalı Yapılandırmaları: Mobil uygulama veri depolama için bulut depolama hizmetlerini kullanıyorsa ve yapılandırmalar yanlış yönetiliyorsa veya yanlış yapılandırılmışsa, depolanan verilerin istenmeyen şekilde açığa çıkmasına veya yetkisiz erişime neden olabilir.

Üçüncü Taraf Kütüphane Güvenlik Açıkları: Mobil uygulamada kullanılan güvenli olmayan üçüncü taraf kütüphaneleri, istenmeyen veri sızıntılarına yol açabilecek güvenlik açıklarına sahip olabilir. Saldırganlar hassas bilgilere yetkisiz erişim elde etmek için bu güvenlik açıklarından faydalanabilir.

İstenmeyen Veri Paylaşımı: Uygulama içindeki veri paylaşım özelliklerinin yanlış kullanımı istenmeyen veri sızıntısına neden olabilir. Hassas veriler istenmeyen alıcılarla paylaşılırsa veya paylaşım süreci yeterince güvenli değilse, gizlilik ihlallerine yol açabilir.

'Güvensiz Veri Depolamayı' Nasıl Önleyebilirim?

Bir mobil uygulamada güvensiz veri depolanmasını önlemek ve hassas verilerin korunmasını sağlamak için aşağıdaki güvenlik önlemleri uygulanmalıdır:

Güçlü Şifreleme Kullanın: Hassas verileri hem bekleme hem de aktarım sırasında korumak için sağlam şifreleme algoritmaları ve uygulamaları uygulayın. Endüstri standardı şifreleme algoritmaları kullanın ve şifreleme anahtarlarının güvenli bir şekilde saklandığından ve yönetildiğinden emin olun.

Güvenli Veri İletimi: Mobil uygulama ve backend sunucuları arasında aktarım sırasında verileri korumak için güvenli iletişim protokollerini (örn. HTTPS, SSL/TLS) kullanın. Hassas verileri güvenli olmayan kanallar üzerinden göndermekten kaçının.

Güvenli Depolama Mekanizmaları Uygulayın: Hassas verileri yetkisiz kullanıcıların erişemeyeceği güvenli depolama konumlarında saklayın. Mobil işletim sistemi tarafından sağlanan Keychain (iOS) veya Keystore (Android) gibi platforma özgü güvenli depolama mekanizmalarını kullanın.

Uygun Erişim Kontrolleri Kullanın: Hassas verilere yetkisiz erişimi kısıtlamak için güçlü erişim kontrolleri uygulayın. Kullanıcıların kimliklerini güvenli bir şekilde doğrulayın, rol tabanlı erişim kontrolleri uygulayın ve hassas bilgilere erişim izni vermeden önce kullanıcı izinlerini doğrulayın.

Girdileri Doğrulayın ve Verileri Temizleyin: Enjeksiyon saldırılarını önlemek ve yalnızca geçerli ve beklenen verilerin depolandığından emin olmak için girdi doğrulama ve veri temizleme tekniklerini uygulayın. Kötü amaçlı kod ekleme veya istenmeyen veri sızıntısı riskini azaltmak için kullanıcı girdilerini doğrulayın.

Güvenli Oturum Yönetimi uygulayın: Rastgele oluşturulmuş oturum token'ları kullanmak, uygun oturum zaman aşımları ayarlamak ve oturum verilerini istemci ve sunucu tarafında güvenli bir şekilde depolamak gibi güvenli oturum yönetimi tekniklerini uygulayın.

Bağımlı Yazılımları Düzenli Olarak Güncelleyin ve Yama Uygulayın: Güvensiz veri depolamaya yol açabilecek güvenlik açıkları içerebilecekleri için tüm kütüphaneleri, çerçeveleri ve üçüncü taraf bağımlılıklarını güncel tutun. İlgili satıcılar tarafından sağlanan güvenlik yamalarını ve güncellemelerini düzenli olarak uygulayın.

Güncel Kalın: Mobil uygulama ortamındaki en son güvenlik tehditleri ve güvenlik açıkları konusunda güncel kalın. Ortaya çıkan risklerin zamanında azaltılmasını sağlamak için güvenlik forumlarını, güvenlik tavsiyelerini ve mobil platform güncellemelerini izleyin.

Örnek Saldırı Senaryoları

Mobil bir uygulamada güvensiz veri depolamanın olası örneklerini açıklayan birkaç örnek senaryo:

Parolaların Düz Metin Olarak Saklanması: Mobil uygulama, kullanıcı parolalarını yerel bir veri tabanında veya dosyada düz metin biçiminde saklar, bu da bir saldırganın cihaza yetkisiz erişim elde etmesi durumunda bu kimlik bilgilerini almasını ve kötüye kullanmasını kolaylaştırır.

Güvenli Olmayan Yerel Depolama: Mobil uygulama, kişisel olarak tanımlanabilir bilgiler (PII) gibi hassas kullanıcı verilerini uygun erişim kontrolleri veya şifreleme kullanmadan cihazda yerel olarak depolar. Bu, cihaza fiziksel erişimi olan herkesin verileri çıkarmasına ve görüntülemesine olanak tanır.

Önbelleğe Güvensiz Veri Alma: Mobil uygulama, kullanıcı kimlik doğrulama tokenları veya oturum bilgileri gibi hassas verileri uygun güvenlik önlemleri almadan önbelleğe alır. Bir saldırgan cihazın önbelleğine erişim kazanırsa, bu kimlik bilgilerini elde edebilir ve kullanıcının kimliğine bürünebilir.

Korumasız Log Kaydı: Mobil uygulama, kullanıcı eylemleri, API yanıtları veya hata mesajları gibi hassas verileri uygun güvenlik kontrolleri olmadan loglar. Bu durum, bir saldırganın cihaza erişim sağlaması veya günlük dosyalarını ele geçirmesi halinde hassas bilgilerin istenmeden açığa çıkmasına neden olabilir.

Güvensiz Bulut Depolama Yapılandırması: Mobil uygulama, kullanıcı verilerini depolamak için bulut depolama hizmetlerini kullanır ancak depolama izinlerini yanlış yapılandırarak depolanan bilgilere yetkisiz erişime izin verir. Bu durum veri sızıntısına veya hassas verilerin izinsiz olarak açığa çıkmasına neden olabilir.

Geçici Dosyaların Uygunsuz Kullanımı: Mobil uygulama hassas verileri işlemek veya depolamak için geçici dosyalar oluşturur, ancak daha sonra bu dosyaları düzgün bir şekilde işlemez ve silmez. Bu da hassas bilgileri açıkta ve yetkisiz erişime karşı savunmasız bırakır.

Referanslar

OWASP

- OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation

Dış Kaynaklar

- CWE - Common Weakness Enumeration (mitre.org)

M10: Yetersiz Şifreleme

Tehdit Unsurları

Uygulamaya Özel

Tehdit unsurları, mobil uygulamalardaki güvensiz şifrelemeden istifade ederek hassas bilgilerin gizliliğine, bütünlüğüne ve gerçekliğine zarar verebilir. Bu tehdit unsurları arasında hassas verilerin şifresini çözmek için kriptografik algoritmaları veya uygulamaları hedef alan saldırganlar, kriptografik süreçleri manipüle eden veya şifreleme anahtarlarını sızdıran içerideki kötü niyetli kişiler, istihbarat amacıyla kriptanaliz yapan devlet destekli aktörler, değerli verileri çalmak veya finansal dolandırıcılık yapmak için yetersiz şifrelemeden yararlanan siber suçlular ve kriptografik protokollerdeki veya kütüphanelerdeki güvenlik açıklarından yararlanan saldırganlar yer alır.

Saldırı vektörleri

İstismar Edilebilirlik **ORTALAMA**

Mobil uygulamalardaki güvensiz şifrelemeye yönelik saldırı vektörü, hassas bilgileri korumak için kullanılan şifreleme mekanizmalarındaki açıklardan faydalanmayı içerir. Saldırganlar şifreleme algoritmaları, anahtar yönetimi veya uygulama kusurlarındaki zayıflıklardan faydalanmak için kriptografik saldırılar, brute force saldırıları veya side-channel saldırıları gibi çeşitli teknikler kullanabilir. Saldırganlar güvensiz şifrelemeyi hedef alarak şifrelenmiş verilerin şifresini çözmeyi, şifreleme süreçlerini manipüle etmeyi veya hassas bilgilere yetkisiz erişim sağlamayı amaçlar. Bu durum veri ihlallerine, kullanıcı hesaplarına yetkisiz erişime, gizliliğin tehlikeye girmesine ya da verilerin taklit veya tahrif edilmesine yol açabilir.

Güvenlik Zafiyetleri

Yaygınlık **ORTAK**
Tespit Edilebilirlik **ORTALAMA**

Bir mobil uygulamadaki güvensiz şifreleme, şifreleme önlemlerinin etkinliğini zayıflatabilecek ve hassas verilerin gizliliğini ve bütünlüğünü tehlikeye atabilecek güvenlik zayıflıklarını ortaya çıkarır. Bu zayıflıklar arasında zayıf şifreleme algoritmalarının veya zayıf anahtar uzunluklarının kullanılması, zayıf anahtar yönetimi pratikleri, şifreleme anahtarlarının yanlış kullanımı, güvenilir olmayan rastgele sayı üretimi, kriptografik protokollerin hatalı uygulanması veya kriptografik kütüphanelerdeki veya çerçevelerdeki güvenlik açıkları yer alabilir. Saldırganlar şifrelemeyi atlamak, kriptografik saldırılar gerçekleştirmek, verileri manipüle etmek veya şifrelenmiş bilgilere yetkisiz erişim sağlamak için bu zayıflıklardan faydalanabilir. Güvenli olmayan hash fonksiyonları ve kriptografik algoritmalar mobil uygulamalarda önemli güvenlik zayıflıklarına yol açmaktadır. Bu güvenlik açıkları ciddi veri ihlallerine ve hassas bilgilere yetkisiz erişime yol açabilir. Eski veya zayıf hash fonksiyonları kullanıldığında, saldırganlar bu açıklardan yararlanarak hashlenmiş veriyi tersine mühendislikle açabilir ve orijinal içeriği ortaya çıkarabilir. Mobil uygulamaları bu güvenlik risklerinden korumak için güçlü ve modern hash fonksiyonları ve kriptografik algoritmalar benimsemenin yanı sıra veri bütünlüğü ve gizliliğini sağlamak için şifreleme ve anahtar yönetiminde en iyi uygulamaları takip etmek çok önemlidir. Düzenli güvenlik denetimleri ve güncellemeleri de potansiyel tehditlere karşı en üst düzeyde koruma sağlamak için çok önemlidir.

Teknik Etki

Etki **ŞİDDETLİ**

Bu güvenlik açığı, hassas bilgilerin mobil cihazdan yetkisiz olarak alınmasına neden olacaktır.

Ticari Etkiler

Etki **ŞİDDETLİ**

Bir mobil uygulamada yetersiz şifreleme veya güvensiz hash fonksiyonları önemli ticari etkilere neden olabilir. İşte bazı potansiyel sonuçlar:

Veri İhlali: Zayıf veya yetersiz şifreleme, düşmanların mobil uygulama tarafından depolanan veya iletilen hassas verilerin gizliliğini tehlikeye atmasını kolaylaştırabilir. Bu durum bir veri ihlaline yol açarak kişisel tanımlanabilir bilgiler (PII), finansal detaylar veya fikri mülkiyet gibi hassas müşteri bilgilerinin açığa çıkmasına neden olabilir. Bu tür ihlaller yasal yükümlülüklere, düzenleyici cezalara, müşteri güveninin kaybına ve itibarın zedelenmesine yol açabilir.

Fikri Mülkiyet Kaybı: Yetersiz kriptografi, mobil uygulamaya gömülü özel algoritmaların, ticari gizliliklerin veya diğer fikri mülkiyetlerin korunmasını tehlikeye atabilir. Düşmanlar bu değerli bilgilerin şifresini çözüp çıkarabilirse, rakip şirketler tarafından rekabet avantajı için kullanılabilir veya karaborsada satılabilir.

Finansal Kayıplar: Yetersiz şifreleme çeşitli şekillerde mali kayıplara yol açabilir. Örneğin, ödeme işlemleri veya finansal veriler yanlış bir şekilde şifrelenirse, müşterileri dolandırıcılığa ve fonlarına yetkisiz erişime maruz bırakabilir. Ayrıca, güvenlik ihlallerinin araştırılması ve düzeltilmesi, etkilenen müşterilerin zararlarının karşılanması ve yasal sonuçların ele alınması ile ilgili maliyetler önemli olabilir.

Uyumluluk ve Yasal Sonuçlar: Birçok sektörde hassas bilgiler için güçlü şifreleme kullanılmasını zorunlu kılan özel veri koruma ve gizlilik düzenlemeleri vardır. Yetersiz şifreleme, bu düzenlemelere uyulmamasına neden olarak yasal sonuçlara, para cezalarına veya düzenleyici makamlar tarafından uygulanan yaptırımlara yol açabilir.

'Yetersiz Şifrelemeye' Karşı Savunmasız mıyım?

Güvensiz şifreleme ve güvensiz hash fonksiyonlarının bir mobil uygulamada ortaya çıkabileceği çeşitli yollar vardır:

Zayıf Şifreleme Algoritmaları: Mobil uygulama, zayıf veya saldırılara karşı savunmasız olduğu bilinen şifreleme algoritmaları kullanabilir. Bu algoritmalar bilinen zayıflıklara sahip olabilir, eski olabilir veya hassas verileri etkili bir şekilde korumak için gerekli güvenlik seviyesinden yoksun olabilir.

Yetersiz Anahtar Uzunluğu: Yetersiz anahtar uzunluğu şifreleme gücünü zayıflatabilir. Mobil uygulama kısa veya kolayca tahmin edilebilir şifreleme anahtarları kullanıyorsa, saldırganların brute force veya diğer kriptografik saldırılar yoluyla şifrelenmiş verilerin şifresini çözmesi daha kolay hale gelir.

Uygun Olmayan Anahtar Yönetimi: Şifreleme anahtarlarının güvenli olmayan bir şekilde saklanması veya düz metin olarak iletilmesi gibi kötü anahtar yönetimi pratikleri, anahtarları yetkisiz erişime maruz bırakabilir. Anahtarlara erişim sağlayan saldırganlar verilerin şifresini zorlanmadan çözebilir.

Hatalı Şifreleme Uygulaması: Şifreleme/şifre çözme işleminin kendisi yanlış uygulanabilir veya programlama kusurları içerebilir. Bu uygulama hataları, saldırganların şifreleme korumalarını atlamak veya zayıflatmak için kullanabilecekleri güvenlik açıklarını ortaya çıkarabilir.

Verilerin/Şifreleme Anahtarlarının Güvensiz Depolanması: Şifreleme anahtarları mobil cihazda düz metin olarak veya kolay erişilebilir yerlerde gibi güvenli olmayan bir şekilde saklanırsa, cihaza fiziksel veya yetkisiz erişimi olan saldırganlar anahtarları alabilir ve korunan verilerin şifresini çözebilir. Mobil uygulama zayıf bir şifreleme algoritması kullanıyorsa veya şifrelemeyi yanlış kullanıyorsa (örneğin zayıf bir anahtar kullanıyorsa veya tüm hassas verileri düzgün bir şekilde şifrelememişse); bu durum, şifrelemenin bir saldırgan tarafından kolayca atlanması veya şifresinin çözülmesi halinde verilerin tehlikeye girmesine neden olabilir.

Güvenli Aktarım Katmanı Eksikliği: Şifrelenmiş verileri ağlar üzerinden iletirken HTTPS gibi güvenli aktarım katmanı protokollerinin kullanılması çok önemlidir. Mobil uygulama güvenli taşıma protokollerini uygulayamazsa, şifrelenmiş veriler iletim sırasında ele geçirilmeye veya tahrif edilmeye karşı savunmasız olabilir.

Yetersiz Doğrulama ve Kimlik Doğrulama: Şifreleme sürecine dahil olan tarafların yetersiz doğrulama ve kimlik doğrulaması genel güvenliği zayıflatabilir. Uygun doğrulama olmadan, saldırganlar meşru varlıkları taklit edebilir, şifrelenmiş verileri ele geçirebilir ve tespit edilmeden manipüle edebilir.

Salting/Tuzlama Eksikliği: Hashing işleminden önce girdiye rastgele veri ekleme işlemi olan tuzlama, parolaların güvenliğini artırmak için çok önemlidir. Güvenli olmayan hash fonksiyonları tuzlamayı desteklemeyebilir veya zayıf tuzlama yöntemleri kullanabilir, bu da parola hashlerini önceden hesaplanmış tablolar veya brute force saldırıları gibi saldırılara açık hale getirir.

'Yetersiz Şifrelemeyi' Nasıl Önleyebilirim?

Mobil uygulamada "yetersiz şifreleme" güvenlik açıklarını önlemek için aşağıdaki en iyi uygulamaları (best practice) göz önünde bulundurun:

Güçlü Şifreleme Algoritmaları Kullanın: AES (Gelişmiş Şifreleme Standardı), RSA (Rivest-Shamir-Adleman) veya Eliptik Eğri Kriptografisi (ECC) gibi yaygın olarak kabul gören ve güvenli şifreleme algoritmaları uygulayın. Güncel kriptografik standartlarla güncel kalın ve kullanımdan kaldırılmış veya zayıf algoritmalardan kaçının.

Yeterli Anahtar Uzunluğu Sağlayın: Güçlü kriptografik dayanıklılık sağlamak için uygun uzunlukta şifreleme anahtarları seçin. Kullanılan özel şifreleme algoritmasını göz önünde bulundurarak anahtar uzunlukları için endüstri önerilerini takip edin.

Güvenli Anahtar Yönetimi Uygulamalarını Takip Edin: Şifreleme anahtarlarını güvenli bir şekilde saklamak için anahtar kasaları veya donanım güvenlik modülleri (HSM'ler) kullanmak gibi güvenli anahtar yönetimi teknikleri kullanın. Yetkili personele erişimi kısıtlamak, bekleyen anahtarları şifrelemek ve güvenli anahtar dağıtım mekanizmaları kullanmak dahil olmak üzere anahtarları yetkisiz erişime karşı koruyun.

Şifrelemeyi Doğru Şekilde Uygulayın: Yerleşik kriptografik kütüphanelere ve framework'lere bağlı kalarak mobil uygulamada şifreleme ve şifre çözme süreçlerini dikkatlice uygulayın. Hatalara ve güvenlik açıklarına daha yatkın oldukları için özel şifreleme uygulamalarından kaçının.

Şifreleme Anahtarlarının Güvenli Depolanması: Şifreleme anahtarlarının mobil cihazda güvenli bir şekilde saklandığından emin olun. Anahtarları düz metin olarak veya kolay erişilebilir yerlerde saklamaktan kaçının. İşletim sistemi tarafından sağlanan güvenli depolama mekanizmalarını kullanmayı veya donanım tabanlı güvenli depolama seçeneklerinden yararlanmayı düşünün.

Güvenli Aktarım Katmanı Kullanın: Şifrelenmiş verileri ağlar üzerinden iletmek için HTTPS (HTTP Secure) gibi güvenli aktarım katmanı protokollerini kullanın. Uygun sertifika doğrulaması uygulayın ve mobil uygulama ile backend sistemleri arasında güvenli iletişim kanalları sağlayın.

Doğrulama ve Kimlik Doğrulama: Şifreleme sürecine dahil olan tarafların bütünlüğünü ve gerçekliğini doğrulamak için güçlü doğrulama ve kimlik doğrulama mekanizmaları uygulayın. Sertifikalar, dijital imzalar veya kimlik doğrulama için kullanılan diğer mekanizmalar için uygun doğrulama gerçekleştirin.

Güvenlik Önlemlerini Düzenli Olarak Güncelleyin: Kriptografik kütüphaneler, çerçeveler ve platform sağlayıcılarının güvenlik güncellemeleri, yamaları ve önerileri hakkında bilgi sahibi olun. Belirlenen güvenlik açıklarını veya zayıflıkları gidermek için mobil uygulamayı ve temel kriptografik bileşenleri güncel tutun.

Güvenlik Testi Yapın: Kriptografik güvenlik açığı değerlendirmeleri, sızma testleri ve kod incelemeleri dahil olmak üzere kapsamlı güvenlik testleri gerçekleştirin. Test süreci sırasında keşfedilen zayıflıkları veya güvenlik açıklarını belirleyin ve düzeltin.

Endüstri Standartlarını ve En İyi Uygulamaları Takip Edin: Kriptografi ile ilgili endüstri standartları ve en iyi uygulamalar konusunda güncel kalın. NIST (Ulusal Standartlar ve Teknoloji Enstitüsü) ve IETF (İnternet Mühendisliği Görev Gücü) gibi kuruluşlar güvenli kriptografik uygulamalar için kılavuzlar ve öneriler sağlar.

Güçlü Hash Fonksiyonları Kullanın: SHA-256 veya bcrypt gibi yaygın olarak tanınan ve kriptografik olarak güvenli hash fonksiyonlarını seçin. Bu algoritmalar saldırılara karşı koymak ve yüksek düzeyde güvenlik sağlamak üzere tasarlanmıştır.

Tuzlama Uygulayın: Parolaları hash ederken her zaman güçlü bir rastgele tuz kullanın. Tuzlama, saldırganların parolaları kırmak için önceden hesaplanmış tabloları veya gökkuşağı tablolarını kullanmasını zorlaştırarak ekstra bir güvenlik katmanı ekler.

Anahtar Türetme İşlevlerini (KDF'ler) kullanın: Parola karma için PBKDF2, bcrypt veya scrypt gibi Anahtar Türetme İşlevlerini kullanın. Bu fonksiyonlar şifrelerden kriptografik anahtarları güvenli bir şekilde türetmek için özel olarak tasarlanmıştır ve brute-force saldırılarını yavaşlatmak için yineleme sayıları gibi ek güvenlik özellikleri sağlar.

Örnek Saldırı Senaryoları

Senaryo 1: Ortadaki Adam (MitM) Saldırıları- Bir saldırgan mobil uygulama ile sunucu arasındaki iletişimi keser. Zayıf kriptografi, saldırganların ele geçirilen verilerin şifresini çözmesini, değiştirmesini ve hedeflenen alıcıya iletmeden önce yeniden şifrelemesini sağlayabilir. Bu da yetkisiz erişime, veri manipülasyonuna veya kötü niyetli içeriğin enjekte edilmesine yol açabilir.

Senaryo 2: Brute-Force Saldırıları- Saldırganlar, verilerin şifresini çözmek için doğru olanı bulana kadar sistematik olarak çeşitli anahtar kombinasyonlarını dener. Zayıf kriptografi bu tür saldırılar için gereken süreyi kısaltarak hassas bilgilerin açığa çıkmasına neden olabilir.

Senaryo 3: Kriptografik Sürüm Düşürme Saldırıları- Mobil uygulamalar güvenli bağlantılar kurmak için birden fazla şifreleme protokolünü veya algoritmasını destekleyebilir. Zayıf şifrelemeye bir geri dönüş seçeneği olarak izin verilirse, saldırganlar bu zayıflıktan yararlanabilir ve uygulamayı zayıf şifreleme kullanmaya zorlayabilir. Sonuç olarak, ele geçirilen verilerin şifresini daha kolay çözebilir ve sonraki saldırıları başlatabilirler.

Senaryo 4: Anahtar Yönetimi Açıkları- Zayıf anahtar yönetimi uygulamaları mobil uygulamalarda kullanılan kriptografik sistemlerin güvenliğini zayıflatabilir. Örneğin, şifreleme anahtarları güvenli olmayan bir şekilde saklanırsa veya kolayca tahmin edilebilirse, saldırganlar anahtarlara yetkisiz erişim sağlayabilir ve şifrelenmiş verilerin şifresini çözebilir. Bu da veri ihlallerine ve gizlilik ihlallerine yol açabilir.

Senaryo 5: Kripto Uygulama Kusurları- Zayıf kriptografi, mobil uygulamanın kendisindeki uygulama kusurlarından da kaynaklanabilir. Bu kusurlar arasında kriptografik kütüphanelerin yanlış kullanımı, güvenli olmayan anahtar üretimi, uygun olmayan rastgele sayı üretimi veya şifrelemeyle ilgili işlevlerin güvenli olmayan şekilde ele alınması yer alabilir. Saldırganlar şifreleme korumalarını atlamak veya zayıflatmak için bu kusurlardan faydalanabilir.

Referanslar

OWASP

- OWASP Top Ten | OWASP Foundation

Dış Kaynaklar

- CWE - Common Weakness Enumeration (mitre.org)