2025 Yılında Android APK Analizindeki En Son ve En Etkili 10 Teknik ve Trend

Yönetici Özeti

Bu rapor, 2025 yılında Android APK analizini şekillendiren en etkili teknik ve trendlere ilişkin ileriye dönük bir inceleme sunmaktadır. Mobil tehdit ortamı, gelişmiş yapay zeka destekli saldırılar, tedarik zinciri güvenlik açıkları ve artan gizlilik düzenlemeleriyle hızla değişmektedir.

2025'te etkili APK analizi, geleneksel statik ve dinamik yöntemlerin ötesine geçerek gelişmiş yapay zeka/makine öğrenimi yeteneklerini, derinlemesine davranışsal incelemeleri, sağlam bütünlük kontrollerini ve geliştirme yaşam döngüsü boyunca proaktif güvenlik önlemlerini bir araya getirmektedir.

Temel temalar arasında yapay zekanın analiz süreçlerini otomatikleştirmedeki ve geliştirmedeki yaygın etkisi, tedarik zinciri görünürlüğünün kritik gerekliliği, tasarımdan itibaren gizliliğe odaklanma ve donanım düzeyinde güvenliğin kullanılması yer almaktadır. Bu tekniklerin benimsenmesi, kuruluşların tırmanan siber tehditler karşısında güçlü bir güvenlik duruşu sergilemeleri, hassas verileri korumaları ve uygulama bütünlüğünü sağlamaları açısından hayati öneme sahiptir.

Giriş: Android APK Analizinin Gelişen Ortamı

Bu projenin temel amacı, bir Android APK dosyasının ayrıntılı bir analizini yaparak, uygulamanın iç yapısını, işleyişini ve temel özelliklerini kapsamlı bir şekilde ortaya koymaktır. İnceleme süreci, uygulamanın tasarım dosyaları (düzenler, resimler), tanımlanan izinler (örneğin, internet erişimi, depolama) ve uygulamanın temel kod yapısı gibi bileşenlerin detaylı olarak incelenmesini içerecektir. Bu analiz sayesinde,

uygulamanın hangi fonksiyonları yerine getirdiği, bu fonksiyonların nasıl çalıştığı ve her bir dosyanın veya bileşenin uygulamanın genel mimarisindeki rolü derinlemesine anlaşılacaktır.

Android APK analizinin geleneksel yöntemleri, derlenmiş kodu yürütmeden inceleyen statik analizi ve uygulamanın çalışma zamanı davranışını gözlemleyen dinamik analizi temel alır.[1] Uygulamanın nasıl çalıştığını anlamak için bir uygulamayı parçalara ayırma süreci olan tersine mühendislik, bu sürecin karmaşık ancak temel bir parçasıdır. Bu süreçte APKtool, JADX, Dex2jar, JD-GUI ve Radare2 gibi araçlar yaygın olarak kullanılmaktadır.[1]

Ancak, mobil uygulama güvenliği ortamı hızla değişmektedir. 2025 yılında, yapay zeka destekli saldırılarda, daha sıkı düzenlemelerde ve bağlı cihazların sayısında bir artışla karşı karşıya kalınmaktadır.[3] Bu durum, daha proaktif, akıllı ve entegre güvenlik analizi yaklaşımlarına doğru bir geçişi zorunlu kılmaktadır.

Android uygulama geliştirme alanındaki gelişmeler, örneğin Android 14 ve 15'in daha sıkı arka plan konum kısıtlamaları ve geçici izinler getirmesi [4], 2025'teki yeni API sürümleri [5] ve yerel kod için 16KB sayfa boyutlarına geçiş [6] gibi platform değişiklikleri, gelişmiş APK analizinin karmaşıklığını ve gerekliliğini doğrudan etkilemektedir. Bu durum, analiz araçlarının yalnızca tehdit ortamındaki değişikliklere değil, aynı zamanda temel Android platformundaki evrime de sürekli olarak uyum sağlaması gerektiğini göstermektedir.

Hem yasal uygulama geliştirmenin (örneğin, yapay zeka entegrasyonu, çoklu platform geliştirme) hem de kötü niyetli faaliyetlerin (yapay zeka destekli kötü amaçlı yazılımlar) artan karmaşıklığı, mobil güvenlikte bir "silahlanma yarışı" yaratmaktadır.[7] Bu, APK analizinin statik bir görev değil, sürekli yenilik gerektiren, sürekli adaptif bir süreç olduğu anlamına gelmektedir. Bu dinamik, analiz tekniklerinin hem yasal hem de kötü amaçlı uygulama tasarımındaki gelecekteki gelişmeleri takip etmesini ve hatta tahmin etmesini gerektirmektedir.

Tablo 1: 2025 Yılında Android APK Analizindeki En Son 10 Teknik ve Trende Genel Bakış

Trend Adı	Kısa Açıklama	APK Analizi İçin Birincil Fayda
Yapay Zeka Destekli Statik ve Dinamik Analiz	APK'larda güvenlik açığı tespiti, kötü amaçlı yazılım analizi ve anomali tanımlamasını otomatikleştirmek ve geliştirmek için yapay zeka/makine öğrenimi kullanılması.	Karmaşık güvenlik açıklarının ve yeni tehditlerin daha hızlı ve doğru bir şekilde tanımlanması, manuel çabanın ve yanlış pozitiflerin azaltılması.
Gelişmiş Obfüskasyon Giderme Teknikleri (LLM Destekli)	Obfüske edilmiş kodu tersine çevirmek ve okunabilirliği sağlamak için Büyük Dil Modelleri (LLM'ler) ve imza tabanlı yöntemlerin kullanılması.	Gizli güvenlik açıklarını ve fikri mülkiyet hırsızlığını belirlemek için obfüske edilmiş uygulama mantığının derinlemesine anlaşılmasını sağlar.
Gelişmiş Yazılım Tedarik Zinciri Güvenliği Analizi	Gizli güvenlik açıkları ve kurcalama için üçüncü taraf SDK'ların, açık kaynak bileşenlerinin ve derleme ortamlarının proaktif analizi.	Tehlikeye atılmış bağımlılıklardan kaynaklanan riskleri azaltır ve tüm yazılım tedarik zincirinde bütünlüğü sağlar, APK'lardaki "kör noktaları" giderir.
Gelişmiş Tehdit Tespiti İçin Davranışsal Analiz	Sahtekarlık, hesap ele geçirme veya kötü amaçlı faaliyetleri gösteren anormallikleri tespit etmek için uygulama ve kullanıcı davranışının yapay zeka destekli izlenmesi.	Geleneksel imza tabanlı tespiti aşan sofistike tehditleri, normal operasyonel kalıplardan sapmaları tanıyarak tanımlar.
Bütünlük Doğrulama ve Kurcalama Karşıtı Analiz	Bir APK'nın yayınlandıktan sonra değiştirilmediğini ve çalışma zamanında kötü amaçlı değişikliklerin önlendiğini sağlamak için sağlam mekanizmaların uygulanması.	Uygulamanın orijinalliğini ve güvenilirliğini garanti eder, kötü amaçlı değişikliklere ve fikri mülkiyet hırsızlığına karşı koruma sağlar.
Güvenlik Açığı Keşfi İçin Otomatik Fuzzing ve Sembolik Yürütme	Girişleri sistematik olarak test ederek ve yürütme yollarını keşfederek güvenlik açıklarını keşfetmek için otomatik tekniklerin kullanılması.	Özellikle Android Niyetleri gibi karmaşık süreçler arası iletişim (IPC) mekanizmalarında bulunması zor hataları ortaya çıkarır, genel güvenliği artırır.
Gizlilik Uyumluluğu ve	Gizlilik düzenlemelerine ve	Gelişen gizlilik yasalarına

Granüler Veri Akışı Analizi	kullanıcı rızasına uyumu sağlamak için bir uygulamanın hassas kullanıcı verilerini nasıl topladığını, işlediğini ve ilettiğini analiz etme.	(örneğin, GDPR, CCPA) uyumu sağlar ve olası veri sızıntısı yollarını belirler, kullanıcı güvenini artırır.
Donanım Destekli Güvenlik Analizi ve TEE'ler	Güvenli yürütme ortamları için TrustZone gibi donanım özelliklerini kullanma ve uygulamalar içindeki uygulamalarını analiz etme.	Kritik operasyonlar ve hassas veriler için daha yüksek düzeyde izolasyon ve koruma sağlar, yazılım tabanlı saldırıların güvenliği tehlikeye atmasını zorlaştırır.
Sanallaştırma Tabanlı Kötü Amaçlı Yazılım Analizi ve Tespiti	Geleneksel güvenlik önlemlerini atlatmak için cihaz içi sanallaştırmayı kullanan kötü amaçlı uygulamaları analiz etme ve tespit etme.	Geleneksel imza tabanlı tespiti aşan sofistike tehditleri, normal operasyonel kalıplardan sapmaları tanıyarak tanımlar.
Sürekli Güvenlik Analizi İçin DevSecOps Entegrasyonu	Güvenlik uygulamalarını ve otomatik analiz araçlarını tüm yazılım geliştirme yaşam döngüsü (SDLC) boyunca entegre etme.	Güvenliği sola kaydırarak güvenlik açıklarını daha erken tespit eder ve düzeltir, maliyetleri azaltır, yayın döngülerini hızlandırır ve işbirliğine dayalı bir güvenlik kültürü oluşturur.

2025 Yılında Android APK Analizindeki En Son 10 Teknik ve Trend

1. Yapay Zeka Destekli Statik ve Dinamik Analiz

Yapay zeka destekli statik ve dinamik analiz, Android APK'larındaki güvenlik açıklarını ve kötü amaçlı kalıpları otomatik olarak tespit etmek ve bu tespiti geliştirmek için makine öğrenimi algoritmalarını kullanır. Statik analiz araçları, yürütme olmaksızın kod kalıplarını analiz etmek ve olası hataları belirlemek için yapay zekayı kullanırken, dinamik analiz, kötü amaçlı yazılım veya istismarları gösterebilecek anormallikler ve şüpheli faaliyetler için çalışma zamanı davranışını izlemek amacıyla yapay zekayı kullanır. Bu yaklaşım, tehdit tespitinin hızını ve doğruluğunu önemli ölçüde artırır, insan

analistlerin üzerindeki yükü azaltır ve yanlış pozitifleri en aza indirir.

2025 yılında yapay zeka, kuruluşların tehditleri nasıl tespit ettiğini, bunlara nasıl yanıt verdiğini ve bunları nasıl hafiflettiğini dönüştüren modern güvenlik operasyonlarının temel taşı haline gelmektedir.[9] Yapay zeka destekli güvenlik açığı taraması, istismar edilmeden önce güvenlik boşluklarını proaktif olarak belirleyebilir ve azaltabilir.[3] Yapay zeka destekli kötü amaçlı yazılım tespiti, kod kalıplarını, yürütme davranışlarını ve tehdit istihbarat verilerini analiz ederek yeni kötü amaçlı yazılım varyantlarını tanımlayabilir.[3] Yapay zeka destekli kötü amaçlı yazılım ve kimlik avı saldırılarındaki artış göz önüne alındığında bu durum kritik öneme sahiptir.[3] Yapay zeka ayrıca, uyumluluğu sürekli izleyerek ve denetim raporları oluşturarak yönetişim, risk ve uyumluluk (GRC) süreçlerini kolaylaştırır.[9] Android Studio'nun kendisi de test ve çökme analizi için Gemini ile yapay zekayı entegre etmekte, önerilen düzeltmeler sunmakta ve kullanıcı yolculuğu testlerini otomatikleştirmektedir.[6]

2025'te yapay zeka destekli saldırıların ve kötü amaçlı yazılımların [3] artması, yapay zeka destekli güvenlik analizi araçlarının [3] benimsenmesini doğrudan zorunlu kılmaktadır. Bu durum, savunma amaçlı yapay zekanın saldırı amaçlı yapay zeka yeteneklerine karşı sürekli olarak gelişmesi gereken bir "yapay zeka silahlanma yarışı" yaratmaktadır.[7] Tehditler daha sofistike hale geldikçe, onları tespit etmek için kullanılan araçlar da aynı derecede gelişmiş olmalıdır.

Yapay zekanın Android Studio gibi geliştirme araçlarına entegrasyonu [6], yalnızca güvenliğe odaklanan yapay zekadan, tüm uygulama yaşam döngüsü boyunca yerleşik yapay zekaya doğru bir geçişi işaret etmektedir. Bu durum, geliştirme, test ve güvenlik arasındaki sınırları bulanıklaştırmaktadır. Bu durum, geliştiricilerin en baştan daha güvenli uygulamalar oluşturmak için yapay zeka destekli yeteneklere sahip olacağı ve potansiyel olarak APK analizi için saldırı yüzeyini azaltacağı anlamına gelmektedir. Bu, güvenlik açıklarının geliştirme sırasında önlenmesine yardımcı olan proaktif bir yaklaşımdır.

- **Güvenilir Kaynak/Referans:** Build38 Blog [3], Qwiet.ai [9], Kaspersky [10], Malwarebytes [7], Google Android Developers Blog.[6]

2. Gelişmiş Obfüskasyon Giderme Teknikleri (LLM Destekli)

Gelişmiş obfüskasyon giderme teknikleri, bir uygulamanın fikri mülkiyetini korumak ve tersine mühendisliği engellemek amacıyla koduna kasten eklenen karmaşıklığı tersine

çevirmeyi amaçlar. DalvikFLIRT gibi imza tabanlı eşleştirme ile birleştirilmiş Büyük Dil Modelleri (LLM) destekli yeniden yazımlar, daha önce aşılamaz olan kodun otomatik güvenlik analizini mümkün kılar. LLM'ler, obfüske edilmiş kodu daha okunabilir bir biçimde yeniden yazmak için bağlamdan ve yapıdan anlam çıkarırken, DalvikFLIRT, adlar obfüske edilmiş olsa bile bilinen kütüphane bileşenlerini benzersiz yapısal ve davranışsal imzalarıyla tanımlar. Bu ikili yaklaşım, uygulamanın gerçek mantığının daha derinlemesine anlaşılmasını sağlar.

2025 yılında obfüskasyon, Android uygulama geliştiricileri için önemli bir savunma mekanizması olmaya devam etmekte, güvenlik araştırmacıları için tersine mühendisliği zorlaştırmaktadır.[11] ProGuard ve R8 gibi araçlar yaygın olarak kullanılmakta olup, ticari araçlar dize şifreleme ve kontrol akışı obfüskasyonu gibi daha gelişmiş özellikler sunmaktadır.[11] Gelişmiş obfüskasyonun aşılması, otomatik güvenlik analizi ve mantıksal güvenlik açığı keşfi için kritik öneme sahiptir.[11] Çoklu ajan sistemi ve özyinelemeli aşağıdan yukarıya analiz kullanan LLM destekli yeniden yazımlar, şifreli kodu anlaşılır versiyonlara dönüştürebilir, anlamlı adlar sağlayabilir ve karmaşık yapıları basitleştirebilir.[11] Bu, Ghidra'nın birleşik bir kontrol akışı grafiği için işleyebildiği yerel kod ve JNI referanslarının analizinde temeldir.[13]

Gelişmiş obfüskasyon tekniklerinin yaygın kullanımı [11], özellikle yapay zeka/LLM'leri kullanan sofistike obfüskasyon giderme yöntemlerinin geliştirilmesini ve gerekliliğini doğrudan tetiklemektedir.[11] Etkili obfüskasyon giderme olmadan, APK'nın mantığının daha derinlemesine güvenlik analizi pratik olmaktan çıkmaktadır. Bu, bir uygulamanın gerçek işlevselliğini ve potansiyel güvenlik açıklarını ortaya çıkarmak için gerekli bir adımdır.

LLM destekli obfüskasyon gidermenin başarısı [11], özellikle mantıksal hatalar için otomatik güvenlik açığı keşfinde önemli bir sıçramayı ifade etmektedir. Obfüske edilmiş kodu okunabilir hale getirerek, güvenlik araçları daha sonra obfüskasyonu kaldırılmış kod üzerinde geleneksel statik analiz kalıplarını veya hatta yapay zeka destekli güvenlik açığı tespitini uygulayabilir. Bu, daha kapsamlı ve otomatik bir güvenlik değerlendirmesine yol açmaktadır. Bu, obfüskasyon gidermenin diğer gelişmiş analiz teknikleri için kritik bir etkinleştirici görevi gördüğü ve daha önce erişilemeyen kod üzerinde işlev görmelerini sağladığı anlamına gelmektedir.

- **Güvenilir Kaynak/Referans:** Ostorlab Blog [11], Doverunner Blog [12], REMY HAX.[13]

3. Gelişmiş Yazılım Tedarik Zinciri Güvenliği Analizi

Gelişmiş yazılım tedarik zinciri güvenliği analizi, üçüncü taraf bileşenler, açık kaynak kütüphaneleri ve bir Android APK oluşturmak için kullanılan tüm derleme ortamı aracılığıyla ortaya çıkan riskleri belirlemeye ve azaltmaya odaklanır. Bu, Yazılım Malzeme Listelerini (SBOM'lar) titizlikle incelemeyi, bileşen zehirlenmesini tespit etmeyi ve ticari ikili dosyalardaki sızdırılmış geliştirici sırlarını veya güvensiz tasarımları izlemeyi içerir. Amaç, tüm bağımlılıklar hakkında tam görünürlük kazanmak ve geliştirme aşamasından dağıtıma kadar bunların bütünlüğünü sağlamaktır.

2025 yılında, yazılım tedarik zincirini güvence altına almak, büyük ölçekli saldırılara karşı savunmasızlığı nedeniyle en önemli önceliklerden biridir.[9] Android ve iOS SDK'larının %60'ından fazlası, kısmi veya eksik SBOM'lara sahip opak ikili dosyalardır; bu durum görünürlüğü azaltmakta ve güvenlik çabalarını engellemektedir.[14] Bu şeffaflık eksikliği, saldırganların bileşenleri zehirlemesine ve mobil tedarik zincirini tehlikeye atmasına olanak tanımaktadır.[14] Sızdırılmış geliştirici sırları (örneğin, sabit kodlanmış kimlik bilgileri, API anahtarları) vakaları geçen yıl %12 artmış ve ticari ikili dosyalar genellikle güvensiz tasarım sergilemektedir.[15] Kuruluşlar, görünürlük ve üçüncü taraf bağımlılıklarının sürekli izlenmesi için yapay zeka destekli SBOM'ları giderek daha fazla kullanmaktadır.[9] Geleneksel CVE takibinin (NVD) ötesine geçerek, sır ifşaları ve derleme ortamı kurcalaması gibi risklerin tam yelpazesine odaklanma ihtiyacı vurgulanmaktadır.[15]

Üçüncü taraf SDK'lardaki opak ikili dosyaların ve eksik SBOM'ların yaygınlığı [14], mobil tedarik zincirindeki görünürlüğün azalmasına doğrudan yol açmaktadır. Bu durum, saldırganların bileşenleri zehirlemesini mümkün kılmaktadır.[14] Bu şeffaflık eksikliği, birçok tedarik zinciri güvenlik açığının temel nedenidir. Bu dinamik, APK analizinin yalnızca uygulamanın kendisini değil, aynı zamanda tüm bağımlılıklarını ve derleme sürecini de kapsamasını zorunlu kılmaktadır.

NVD aracılığıyla yalnızca CVE'leri takip etmekten, "sır ifşaları, derleme ortamının kurcalanması ve dosya çürümesi gibi yazılım tedarik zincirine yönelik risklerin tam yelpazesine" daha geniş bir odaklanmaya geçiş [15], tedarik zinciri güvenliğinin olgunlaştığını göstermektedir. Bu durum, APK analizinin bilinen güvenlik açıklarının ötesine geçerek, uygulamanın nasıl oluşturulduğu ve bağımlılıklarının nasıl yönetildiği konusundaki sistemik riskleri proaktif olarak belirlemesi gerektiğini ifade etmektedir. Bu, reaktif yamalamadan proaktif risk yönetimine doğru bir paradigmatik değişimdir.

- **Güvenilir Kaynak/Referans:** Security Buzz [14], ISACA Now Blog [15], Qwiet.ai.[9]

4. Gelişmiş Tehdit Tespiti İçin Davranışsal Analiz

Android APK analizinde davranışsal analiz, kullanıcı ve uygulama davranışındaki anormallikleri tespit etmek için yapay zekayı kullanmayı içerir. Bu anormallikler, sahtekarlık, hesap ele geçirme veya sofistike kötü amaçlı yazılımların varlığını gösterebilir. Bu yaklaşım, "normal" operasyonel kalıpları anlayarak ve çalışma zamanında şüpheli sapmaları işaretleyerek imza tabanlı tespitin ötesine geçer.

2025 yılında yapay zeka destekli davranışsal analiz, sahtekarlığı ve hesap ele geçirmeyi önlemek için kullanıcı davranışındaki anormallikleri tespit eden önemli bir gelişmedir.[3] Örneğin, bir cihazın bir uygulamaya alışılmadık bir konumdan eriştiğini veya insan dışı bir davranış sergilediğini tanıyabilir.[3] Bu, bireysel kullanıcılara özel olarak tasarlanmış, sosyal mühendislik saldırılarının başarı oranlarını artıran yapay zeka tarafından oluşturulan kimlik avı kampanyalarına karşı özellikle etkilidir.[3] İnsan hataları önemli bir zayıf halka olmaya devam ettiğinden, bu tür analizler geleneksel yöntemleri aşan sofistike tehditleri belirlemeye yardımcı olur.[3] Hassas verilerin korunması ve gizlilik yasalarına uyum sağlanması için kritik öneme sahiptir.[16]

Yapay zeka destekli kimlik avı ve sosyal mühendislik saldırılarının artan karmaşıklığı [3], yapay zeka destekli davranışsal analiz ihtiyacını doğrudan tetiklemektedir.[3] Geleneksel imza tabanlı tespit, son derece kişiselleştirilmiş ve gelişen saldırı vektörlerine karşı yetersiz kalmakta, bu da davranışsal anormallikleri daha güvenilir bir tehlike göstergesi haline getirmektedir.

Davranışsal analizin etkinliği, yalnızca kötü amaçlı yazılım tespitinin ötesine geçerek sahtekarlık ve hesap ele geçirme gibi daha geniş güvenlik sorunlarını da kapsamaktadır.[3] Bu durum, APK analizinin, çalışma zamanı davranışsal izleme ile birleştirildiğinde, yalnızca uygulamanın kodu için değil, aynı zamanda kullanıcılarla ve cihaz ortamıyla etkileşimi için de daha bütünsel bir güvenlik duruşu sağlayabileceğini göstermektedir. Bu, bir uygulamanın gerçek dünya güvenliğini anlamak için kritik bir tamamlayıcı tekniktir ve çalışma zamanı izleme ile entegrasyon gerektirmektedir.

- **Güvenilir Kaynak/Referans:** Build38 Blog [3], Qwiet.ai [9], Appaloosa.io [16], Sidekick Interactive.[17]

5. Bütünlük Doğrulama ve Kurcalama Karşıtı Analiz

Bütünlük doğrulama ve kurcalama karşıtı analiz, bir Android APK'sının orijinal sürümünden bu yana değiştirilmediğinden veya bozulmadığından emin olmak ve çalışma zamanında kötü amaçlı değişiklikleri önlemek için sağlam mekanizmaların uygulanmasını içerir. Bu, platform düzeyinde bütünlük API'lerini, dijital imzaları ve yetkisiz değişiklikleri tespit etmek ve bunlara yanıt vermek için kendi kendini koruma teknolojilerini kullanmayı kapsar.

2025 yılında Google Play Bütünlük API'si, Android 13+ cihazlar için daha sıkı davranışsal değişiklikler getirmiş olup, bütünlük kararları almak için uygulamaların Google Play tarafından yüklenmesini veya güncellenmesini gerektirmektedir.[18]

MEETS_STRONG_INTEGRITY kararı artık son bir yıl içinde bir güvenlik güncellemesi gerektirmektedir.[18] Bu, platformun uygulama orijinalliğini sağlamaya olan bağlılığını vurgulamaktadır. Saldırganlar, geleneksel doğrulama süreçlerini atlatmak için meşru uygulamaları kötü amaçlı özelliklerle sık sık yeniden paketlemektedir.[14] Güvenli kod obfüskasyonu [19] ve Çalışma Zamanı Uygulama Kendi Kendini Koruma (RASP) [3] kritik öneme sahiptir; RASP, gerçek zamanlı tehditleri izlemekte ve kötü amaçlı kod yürütmesini önlemektedir. Dijital imzalar, Play Store'daki uygulamaların yetkisiz değiştirilmesini ve kötü amaçlı güncellemeleri önlemek için kritik öneme sahiptir.[20]

Google Play Bütünlük API'sinin daha sıkı gereksinimleri [18], geliştiricileri uygulamalarında daha güçlü bütünlük doğrulamasına doğru doğrudan itmektedir. Bir uygulama Google Play tarafından yüklü olarak tanınmazsa veya yakın zamanda güvenlik güncellemelerinden yoksunsa, güçlü bütünlük kararları almayacak, bu da güveni ve işlevselliği etkileyecektir. Bu durum, geliştiricileri bu bütünlük kontrollerini doğal olarak destekleyen ve geçen uygulamalar oluşturmaya zorlamaktadır.

Platform düzeyinde bütünlük kontrolleri (Google Play Bütünlük API'si) güçlenirken, saldırganlar aynı zamanda bu resmi kanalları atlatmak için "yan yükleme" ve "meşru uygulamaların yeniden paketlenmiş versiyonlarını" [14] kullanmaktadır. Bu durum, harici analiz yöntemlerinin standart uygulama mağazası korumalarını aşan tehditleri hesaba katması gereken sürekli bir kedi-fare oyununu vurgulamaktadır. Bu, APK analizinin hem resmi hem de gayri resmi dağıtım vektörlerini bütünlük kontrolleri için değerlendirmesi gerektiği anlamına gelmektedir.

- **Güvenilir Kaynak/Referans:** Android Enterprise Community [18], Security Buzz [14], Clarion Technologies [19], Promon.[20]

6. Güvenlik Açığı Keşfi İçin Otomatik Fuzzing ve Sembolik Yürütme

Otomatik fuzzing, programı sistematik olarak hatalı veya beklenmedik girdilerle besleyerek çökmeleri veya beklenmedik davranışları tetiklemeyi ve böylece güvenlik açıklarını ortaya çıkarmayı içerir. Tamamlayıcı bir teknik olan sembolik yürütme, somut veriler yerine sembolik değerler kullanarak program yollarını analiz eder, bu da çok daha geniş bir yürütme durumu aralığının keşfedilmesine ve belirli kod yollarına veya güvenlik açıklarına yol açan girdilerin tanımlanmasına olanak tanır. Birleştirildiğinde, bu yöntemler derinleşimdeki güvenlik açıklarını keşfetmek için güçlü bir yaklaşım sunar.

2025 yılında fuzzing, güvenlik açığı keşfi için kritik bir teknik olmaya devam etmektedir. Açık kaynaklı, kapsama kılavuzlu bir fuzzing çerçevesi olan MALintent, Android Niyet işleyicilerindeki güvenlik açıklarını belirlemek için tasarlanmıştır.[21] Bu, derlenmiş kapalı kaynak Android uygulamalarında gri kutu fuzzing uygulayan ilk Niyet fuzzer'ıdır ve yeni kapsama enstrümantasyonu ve hata orakları kullanarak çökmeler, gizlilik ihlalleri ve bellek güvenliği sorunları bulur.[21] Birincil mesajlaşma mekanizması olan Niyetler, özellikle kötü amaçlı, düşük ayrıcalıklı uygulamalardan kullanıcı etkileşimi olmadan gönderilebildikleri için fuzzing için etkili giriş noktalarıdır.[21] 2025'teki araştırmalar, ikili dönüştürücüleri doğrulamak ve web uygulaması fuzzing'ini optimize etmek de dahil olmak üzere gelişmiş fuzzing tekniklerini keşfetmeye devam etmektedir.[22]

"Niyet işleyicilerinin" Android uygulamalarını fuzzing için "etkili giriş noktaları" olarak vurgulanması [21], 2025'te otomatik güvenlik açığı keşfi için belirli, yüksek etkili bir alanı işaret etmektedir. Bu, Niyetlerin uygulamalar arasında güven sınırlarını aşabilmesi ve hassas işlevselliği açığa çıkarabilmesi nedeniyle kritik öneme sahiptir. Bu, fuzzing'in Android'deki mimari güvenlik açığı vektörünü benzersiz bir şekilde hedefleyebileceği anlamına gelmektedir.

MALintent'in ikili enstrümantasyon (JVMTI) kullanarak *kapalı kaynak* Android uygulamalarında gri kutu fuzzing yapabilmesi [21], önemli bir ilerlemeyi temsil etmektedir. Bu, tescilli uygulamaların analizindeki büyük bir engeli aşmakta ve otomatik güvenlik testlerinin kapsamını genişletmektedir. Bu, daha önce erişilmesi zor olan Android uygulama ekosisteminin daha büyük bir kısmının daha derinlemesine analizini sağlayan temel bir metodolojik atılımdır.

- **Güvenilir Kaynak/Referans:** NDSS Sempozyumu 2025 Kabul Edilen Bildiriler [21], FuzzingPaper.[22]

7. Gizlilik Uyumluluğu ve Granüler Veri Akışı Analizi

Gizlilik uyumluluğu ve granüler veri akışı analizi, bir Android uygulamasının hassas kullanıcı verilerini nasıl topladığını, işlediğini, depoladığını ve ilettiğini titizlikle takip etmeyi içerir. Bu analiz, gelişen gizlilik düzenlemelerine (GDPR ve CCPA gibi) uyumu sağlar ve olası veri sızıntısı yollarını belirlerken, açık kullanıcı rızasına ve en az ayrıcalık ilkesine odaklanır.

2025 yılında Android izinleri, kullanıcı gizliliğine ve bağlamsal erişime artan bir vurgu ile daha incelikli hale gelmiştir.[4] Android 14 ve 15, daha sıkı arka plan konum kısıtlamaları ve geçici izinler getirerek, uygulamaların Play Store incelemesi sırasında izin ihtiyaçlarını daha net bir şekilde gerekçelendirmesini gerektirmiştir.[4] Gizlilik odaklı uygulama tasarımı, yalnızca uyumluluk için değil, aynı zamanda kullanıcı güvenini oluşturmak için de kritik öneme sahiptir; kullanıcıların %81'i veri işleme konusunda endişe duymaktadır.[17] GDPR veya CCPA gibi gizlilik yasalarına sıkı uyum zorunludur ve ihlaller için ağır para cezaları öngörülmektedir.[16] Granüler veri akışı analizi, ayrıcalıksız uygulamaların rıza olmadan özel verilere erişip erişemeyeceğini veya hassas verilerin yetkisiz havuzlara akıp akmadığını belirler.[21] Android 2025'teki yeni arama içi korumalar, aramalar sırasında riskli güvenlik eylemlerini, örneğin hassas verileri çalabilecek erişilebilirlik izinlerini vermeyi engellemeyi amaçlamaktadır.[24]

Kullanıcıların gizliliğe yönelik artan talepleri [17] ve giderek daha sıkı hale gelen düzenlemeler (GDPR, CCPA, HIPAA, FERPA) [16], Android izinlerinin evriminin ve granüler veri akışı analizinin gerekliliğinin temel itici güçleridir.[4] Uyumluluk eksikliği, önemli mali cezalara ve kullanıcı güveninin kaybına yol açabilir.

İzinler için "bağlamsal istemlere" [4] ve dolandırıcılık tespiti için "cihaz içi yapay zekaya" [24] yapılan vurgu, daha akıllı ve gizlilik koruyucu güvenlik mekanizmalarına doğru bir hareketi göstermektedir. Bu, APK analizinin yalnızca beyan edilen izinleri kontrol etmekle kalmayıp, izinlerin nasıl istendiğini ve dinamik olarak nasıl kullanıldığını ve cihaz içi işlemenin hassas verileri nasıl koruduğunu da değerlendirmesi gerektiği anlamına gelmektedir. Bu, gizliliğin uygulamanın ve işletim sisteminin

davranışına ve *mimarisine* dahil edildiğini, sadece statik bildirimlere değil.

- **Güvenilir Kaynak/Referans:** Google Security Blog [24], Appaloosa.io [16], Medium Blog [4], Clarion Technologies [19], Sidekick Interactive [17], NDSS Sempozyumu 2025

Kabul Edilen Bildiriler.[21]

8. Donanım Destekli Güvenlik Analizi ve Güvenilir Yürütme Ortamları (TEE'ler)

Donanım destekli güvenlik analizi, ARM TrustZone veya Intel Sanallaştırma Teknolojisi gibi özel donanım özelliklerini kullanarak Güvenilir Yürütme Ortamları (TEE'ler) oluşturmayı içerir. Bu TEE'ler, kriptografik anahtar yönetimi, güvenli önyükleme ve içerik şifre çözme gibi hassas işlemler için ana işlemci üzerinde güvenli, izole bir ortam sağlar ve bu da onları yazılım tabanlı saldırılara karşı oldukça dirençli hale getirir. Bir APK'nın bu donanım düzeyindeki korumalarla nasıl etkileşim kurduğunu veya bunları nasıl atlamaya çalıştığını analiz etmek çok önemlidir.

ARM'ın TrustZone teknolojisi, Android akıllı telefonlar ve IoT cihazları da dahil olmak üzere milyarlarca cihazda güvenliğin temelini oluşturmaktadır.[25] Güvenli bir işletim sistemi olan Trusty, Android için bir TEE sağlar ve Android işletim sisteminden izole edilmiş olsa da ana işlemci ve belleğe erişimi vardır.[26] Bu izolasyon, kötü amaçlı uygulamalara ve güvenlik açıklarına karşı koruma sağlar.[26] Üçüncü taraf Trusty uygulama geliştirme şu anda desteklenmese de, Trusty ortaklar için şeffaflık ve hata ayıklama kolaylığı sunmaktadır.[26] Gerçek dünya TrustZone yazılımının emülasyon yoluyla dinamik analizi, güvenlik açıkları bulmak için hem uygulanabilir hem de faydalıdır.[25] Synopsys, 2025'te donanım destekli doğrulama portföyünü genişleterek yazılım başlatma süreçlerini hızlandırmakta ve Android'in 10 dakikadan daha kısa sürede önyüklenmesini sağlamaktadır.[27] 2025-05-05 ve 2025-06-05 yama düzeyleri için Android güvenlik bültenleri, Qualcomm, Arm ve Imagination Technologies bileşenlerindeki güvenlik açıklarını ele alarak donanım düzeyinde güvenliğin devam eden önemini vurgulamaktadır.[28]

TEE'ler gibi donanım destekli güvenlik özelliklerine [25] kritik operasyonlar (örneğin, kriptografik anahtarlar, güvenli önyükleme) için artan bağımlılık, bu güvenli enklavlarla etkileşimleri inceleyebilen ve doğrulayabilen analiz tekniklerini doğrudan gerekli kılmaktadır. Bu donanım katmanlarını veya yazılım arayüzlerini hedefleyen saldırılar [25] önemli bir risk oluşturmaktadır. Bu, APK analizinin, uygulamanın donanım tabanlı güvenlik mekanizmalarını nasıl kullandığını veya bunları atlamaya çalıştığını anlaması gerektiği anlamına gelmektedir.

TEE'ler sağlam koruma sunarken, kapalı kaynak doğaları [26] ve ana Android işletim sistemiyle etkileşimlerinin karmaşıklığı [25], kapsamlı APK analizi için bir zorluk teşkil

etmektedir. Bu durum, gelecekteki analiz araçlarının Android uygulama katmanı ile temel donanım güvenliği arasındaki boşluğu, potansiyel olarak özel emülasyon veya donanım düzeyinde profil oluşturma araçları aracılığıyla [27] doldurması gerekeceğini göstermektedir. Bu, güvenlik analizi için bu kritik katmana görünürlük kazanma yönünde bir eğilimdir.

- **Güvenilir Kaynak/Referans:** Google Android Güvenlik Bülteni [28], Synopsys Basın Bülteni [27], Android Source [26], USENIX Security [25], Promon.[20]

9. Sanallaştırma Tabanlı Kötü Amaçlı Yazılım Analizi ve Tespiti

Sanallaştırma tabanlı kötü amaçlı yazılım analizi ve tespiti, cihaz içi sanallaştırmayı kullanarak kendi işlemleri için izole, gizli ortamlar oluşturan kötü amaçlı uygulamaları belirlemeye ve anlamaya odaklanır. Bu sofistike teknik, kötü amaçlı yazılımların meşru uygulamaları ele geçirmesine, gerçek zamanlı dolandırıcılık yapmasına ve geleneksel güvenlik önlemlerini atlatmasına olanak tanır. Bunu, kurbanın cihazında sanallaştırılmış bir katman içinde çalışarak yapar, bu da geleneksel çalışma zamanı izleme ile tespit edilmesini zorlaştırır.

2025 yılında, gelişmiş GodFather Android bankacılık trojanı gibi yeni Android kötü amaçlı yazılımlar, gerçek zamanlı dolandırıcılık için meşru mobil bankacılık ve kripto para uygulamalarını ele geçirmek amacıyla cihaz içi sanallaştırmayı kullanmaktadır.[31] Bu kötü amaçlı yazılım, bir sanallaştırma çerçevesi içeren kötü amaçlı bir "ana bilgisayar" uygulaması yükleyerek, kurbanın cihazında eksiksiz, izole bir sanal ortam oluşturur.[31] Bu, kimlik bilgilerini çalmak için geleneksel yer paylaşımı taktiklerinin ötesinde bir paradigma değişimi temsil etmektedir.[31] Bu tür kötü amaçlı yazılımları analiz etmek, sanallaştırma özelliklerini ve ana bilgisayar cihazıyla nasıl etkileşim kurduğunu anlamayı gerektirir.[31] Bilinmeyen APK'ların güvenli testi, birincil cihazdan izole ederek ve sanal bir ortamda çalıştırarak sanal alanlar veya APK emülatörleri kullanılarak yapılabilir.[1]

"Cihaz içi sanallaştırmayı" kullanan kötü amaçlı yazılımların ortaya çıkışı [31], 2025 için kritik yeni bir tehdit vektörüdür. Bu, mobil kötü amaçlı yazılımların nasıl çalıştığını ve tespitten nasıl kaçtığını temelden değiştirmektedir. Bu, basit yer paylaşımlarının ötesine geçerek, cihazın içinde tamamen izole, kötü amaçlı bir ortam oluşturmaktadır. Bu, geleneksel APK analiz yöntemlerinin gözden kaçırabileceği yeni ve gelişmiş bir saldırı tekniğidir.

Sanallaştırma tabanlı kötü amaçlı yazılımların yükselişi [31], bu gizli sanal katmanları tespit edebilen ve analiz edebilen gelişmiş sanal alan ve emülatör tabanlı analiz ortamlarının [1] kullanılmasını doğrudan gerekli kılmaktadır. Kötü amaçlı yazılım sanallaştırılmış bir ortamda bulunuyorsa, geleneksel dinamik analiz, kötü amaçlı yazılımın işlemlerini tam olarak ortaya çıkaramayabilir. Bu, analiz araçlarının bu tür ortamlarda çalışabilmesi veya bunları içermesi gerektiği anlamına gelmektedir.

- **Güvenilir Kaynak/Referans:** The Hacker News [31], Sphinx Solution [32], Corellium Blog.[1]

10. Sürekli Uygulama Güvenliği Analizi İçin DevSecOps Entegrasyonu

DevSecOps entegrasyonu, yazılım geliştirme yaşam döngüsünün (SDLC) planlama ve tasarımdan kodlama, test ve sürekli bakıma kadar tüm aşamalarına güvenlik uygulamalarını ve otomatik analiz araçlarını yerleştirmeyi içerir. Bu "sola kaydırma" yaklaşımı, güvenlik açıklarını geliştirme sürecinin erken aşamalarında belirlemeyi ve düzeltmeyi, geliştirme, güvenlik ve operasyon ekipleri arasında işbirliğini teşvik etmeyi ve güvenli yazılım teslimini hızlandırmayı amaçlar.

2025 yılında DevSecOps entegrasyonu, güvenlik açıklarını dağıtımdan önce belirlemek için güvenlik uygulamalarını geliştirme iş akışlarına yerleştiren önemli bir stratejidir.[3] Otomatik güvenlik test araçları, hataların geliştirme döngüsünün her aşamasında ele alınmasını sağlar.[3] 2025'teki modern SDLC, çevik metodolojileri, DevOps uygulamalarını ve yapay zeka destekli geliştirme araçlarını birleştirerek, hızlı prototipleme ve sürekli entegrasyonu vurgulamaktadır.[33] Yapay zeka, otomatik kod üretimi, planlama için tahmine dayalı analiz ve yapay zeka destekli testler de dahil olmak üzere görevleri otomatikleştirerek, karar vermeyi iyileştirerek ve verimliliği artırarak SDLC'yi devrim niteliğinde değiştirmektedir.[34] DevSecOps, CI/CD boru hatlarını geliştirerek daha hızlı, daha güvenilir ve minimum kesinti süresiyle yayınları sağlar.[33] Bu, otomatik birim, entegrasyon ve regresyon testlerinin yanı sıra dağıtım öncesi güvenlik açığı taramalarını ve kod incelemelerini de içerir.[33] Sıfır Güven mimarisinin ve gelişmiş MFA'nın uygulanması, bu çerçeve içindeki temel güvenlik önlemleridir.[33]

DevSecOps, reaktif güvenlikten (dağıtım sonrası APK analizi) SDLC boyunca yerleşik proaktif, sürekli güvenliğe doğru temel bir değişimi temsil etmektedir.[3] Bu, APK analiz araçlarının ve tekniklerinin, yaşam döngüsünün sonunda bağımsız kontroller olmak yerine, otomatik CI/CD boru hatlarına entegre edilmesi gerektiği anlamına gelmektedir.

Bu, güvenlik açıklarının daha erken tespit edilmesini ve düzeltilmesini sağlayarak maliyetleri azaltır ve güvenli yazılım teslimini hızlandırır.

Yapay zeka, DevSecOps için önemli bir etkinleştiricidir.[33] Kod üretimini otomatikleştirmesi, riskleri tahmin etmesi, testleri yönlendirmesi ve CI/CD boru hatlarını optimize etmesi, güvenliğin hızlı bir şekilde entegre edilmesini sağlayarak DevSecOps modelini Android uygulama geliştirme için pratik ve verimli hale getirmektedir. Bu, yapay zekanın DevSecOps içindeki yalnızca başka bir araç değil, tüm sürekli güvenlik entegrasyon sürecini uygulanabilir ve etkili kılan dönüştürücü bir teknoloji olduğunu göstermektedir.

- **Güvenilir Kaynak/Referans:** Build38 Blog [3], Qwiet.ai [9], Sunbytes.io [33], Zignuts.[34]

Sonuç: 2025 Yılında Android Uygulama Güvenliği İçin Stratejik Zorunluluklar

2025 yılında Android APK'larının analizi artık statik, geliştirme sonrası bir görev değil, dinamik, sürekli ve yüksek düzeyde entegre bir süreçtir. Siber tehditlerin, özellikle yapay zekadan yararlanan ve yazılım tedarik zincirini hedef alanların artan karmaşıklığı, proaktif ve çok katmanlı bir güvenlik yaklaşımını zorunlu kılmaktadır.

Kuruluşlar için stratejik zorunluluklar şunları içermektedir:

- **Yapay Zekayı Temel Bir Güvenlik Etkinleştiricisi Olarak Benimsemek:** Akıllı statik/dinamik analizden ve obfüskasyon gidermeden davranışsal tehdit tespitine kadar, yapay zeka, gelişen tehditlere ayak uydurmak ve karmaşık analiz görevlerini otomatikleştirmek için vazgeçilmezdir.
- **Tedarik Zinciri Şeffaflığı ve Bütünlüğüne Öncelik Vermek:** Üçüncü taraf bileşenlerin derinlemesine analizi ve sağlam bütünlük doğrulaması, tehlikeye atılmış bağımlılıklardan kaynaklanan riskleri azaltmak ve dağıtılan uygulamaların orijinalliğini sağlamak için kritik öneme sahiptir.
- **Güvenliği SDLC Boyunca Entegre Etmek:** DevSecOps uygulamaları aracılığıyla güvenliği "sola kaydırmak", güvenlik açıklarının erken tespit edilmesini ve düzeltilmesini sağlayarak maliyetleri azaltır ve güvenli uygulamaların teslimini hızlandırır.
- **Platform ve Donanım Düzeyinde Korumalardan Yararlanmak:** Android'in gelişen gizlilik kontrollerini, TEE'lerini ve kurcalama karşıtı mekanizmalarını anlamak

ve kullanmak, gerçekten dayanıklı uygulamalar oluşturmak için temeldir.

Android APK analizinin geleceği, gelişmiş teknik yetenekleri sürekli, entegre bir güvenlik zihniyetiyle birleştiren bütünsel bir stratejide yatmaktadır. Bu en iyi teknikleri ve trendleri proaktif olarak benimseyen kuruluşlar, 2025 ve sonrasındaki sofistike tehditlere karşı uygulamalarını, kullanıcılarını ve hassas verilerini daha iyi koruyabileceklerdir.

Alıntılanan çalışmalar

1. Apk Reverse Engineering | Compile Code to Readable Insights, erişim tarihi Haziran 28, 2025, https://www.corellium.com/blog/android-mobile-reverse-engineering
2. OWASP-MSTG/Document/0x05c-Reverse-Engineering-and ... - GitHub, erişim tarihi Haziran 28, 2025, https://github.com/boblone19/OWASP-MSTG/blob/master/Document/0x05c-Reverse-Engineering-and-Tampering.md
3. Mobile Application Security Trends 2025 - Build38, erişim tarihi Haziran 28, 2025, https://build38.com/blog/mobile-security/mobile-application-security-trends-2025-key-insights-and-emerging-strategies/
4. Mastering Android Permissions in 2025: Best Practices and New ..., erişim tarihi Haziran 28, 2025, https://medium.com/@vivek.beladia/mastering-android-permissions-in-2025-best-practices-and-new-trends-0c1058c12673
5. Features and APIs | Android Developers, erişim tarihi Haziran 28, 2025, https://developer.android.com/about/versions/16/features
6. What's new in Android ... - Android Developers Blog: Google I/O 2025, erişim tarihi Haziran 28, 2025, https://android-developers.googleblog.com/2025/05/google-io-2025-whats-new-in-android-development-tools.html
7. Malwarebytes Press Center - News & Events | Agentic AI Will ..., erişim tarihi Haziran 28, 2025, https://www.malwarebytes.com/press/2025/02/04/agentic-ai-will-revolutionize-cybercrime-in-2025-according-to-malwarebytes-state-of-malware-report
8. 21 Top Trends in Android App Development for 2025 - AppsRhino, erişim tarihi Haziran 28, 2025, https://www.appsrhino.com/blogs/top-trends-in-android-app-development
9. The Top 10 AppSec Trends Shaping Cybersecurity in 2025 - Qwiet AI, erişim tarihi Haziran 28, 2025, https://qwiet.ai/appsec-resources/the-top-10-appsec-trends-shaping-cybersecurity-in-2025/
10. ChatGPT-Mimicking Cyberthreats Surge 115% in Early ... - Kaspersky, erişim tarihi Haziran 28, 2025, https://www.kaspersky.com/about/press-releases/kaspersky-chatgpt-mimicking-

cyberthreats-surge-115-in-early-2025-smbs-increasingly-targeted
11. Bypassing Obfuscation in Android Apps: A Dual Approach with ..., erişim tarihi Haziran 28, 2025, https://blog.ostorlab.co/bypassing-obfuscation-android-app-dalvik-flirt-llm-powered-rewrites.html
12. Android App Obfuscation – Relevance in an insecure mobile application world, erişim tarihi Haziran 28, 2025, https://doverunner.com/blogs/android-app-obfuscation-guide/
13. Ghidra Is Best: Android Reverse Engineering | REMY HAX, erişim tarihi Haziran 28, 2025, https://remyhax.xyz/posts/android-with-ghidra/
14. Mobile Threats Surge in 2025: Phishing, Sideloading, and Supply ..., erişim tarihi Haziran 28, 2025, https://securitybuzz.com/cybersecurity-news/mobile-threats-surge-in-2025-phishing-sideloading-and-supply-chain-blind-spots/
15. ISACA Now Blog 2025 The 2025 Software Supply Chain Security ..., erişim tarihi Haziran 28, 2025, https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2025/the-2025-software-supply-chain-security-report
16. Mobile security, a strategic imperative in 2025 - Appaloosa.io, erişim tarihi Haziran 28, 2025, https://www.appaloosa.io/blog/mobile-security-imperative-2025
17. Data Security in Mobile Apps: Protecting User Privacy in 2025 and ..., erişim tarihi Haziran 28, 2025, https://www.sidekickinteractive.com/uncategorized/data-security-in-mobile-apps-protecting-user-privacy-in-2025-and-beyond/
18. Google Play Integrity API behavioral changes | Android Enterprise ..., erişim tarihi Haziran 28, 2025, https://www.androidenterprise.community/kb/announcements/google-play-integrity-api-behavioral-changes/11228
19. Top Security Measures for Mobile Apps in 2025: Stay Compliant in the US Market, erişim tarihi Haziran 28, 2025, https://www.clariontech.com/blog/top-security-measures-for-mobile-apps-in-2025-stay-compliant-in-the-us-market
20. App Threat Report 2025 Q1: Mitigations taken to protect Android and iOS - Promon, erişim tarihi Haziran 28, 2025, https://promon.io/security-news/app-threat-report-2025-q1
21. MALintent: Coverage Guided Intent Fuzzing Framework for Android, erişim tarihi Haziran 28, 2025, https://www.ndss-symposium.org/wp-content/uploads/2025-125-paper.pdf
22. Recent Papers Related To Fuzzing | FuzzingPaper - GitHub Pages, erişim tarihi Haziran 28, 2025, https://wcventure.github.io/FuzzingPaper/
23. NDSS Symposium 2025 Accepted Papers, erişim tarihi Haziran 28, 2025, https://www.ndss-symposium.org/ndss2025/accepted-papers/
24. What's New in Android Security and ... - Google Online Security Blog, erişim tarihi Haziran 28, 2025, https://security.googleblog.com/2025/05/whats-new-in-android-security-privacy

-2025.html
25. PARTEMU: Enabling Dynamic Analysis of Real-World TrustZone Software Using Emulation, erişim tarihi Haziran 28, 2025, https://www.usenix.org/conference/usenixsecurity20/presentation/harrison
26. Trusty TEE - Android Open Source Project, erişim tarihi Haziran 28, 2025, https://source.android.com/docs/security/features/trusty
27. Synopsys Expands the Industry's Highest Performance Hardware-Assisted Verification Portfolio to Propel Next-Generation Semiconductor and Design Innovation - Feb 13, 2025, erişim tarihi Haziran 28, 2025, https://news.synopsys.com/2025-02-13-Synopsys-Expands-the-Industrys-Highest-Performance-Hardware-Assisted-Verification-Portfolio-to-Propel-Next-Generation-Semiconductor-and-Design-Innovation
28. Android Security Bulletin—June 2025 | Android Open Source Project, erişim tarihi Haziran 28, 2025, https://source.android.com/docs/security/bulletin/2025-06-01
29. Android Security Bulletin—May 2025 | Android Open Source Project, erişim tarihi Haziran 28, 2025, https://source.android.com/docs/security/bulletin/2025-05-01
30. Android 16 Security Release Notes, erişim tarihi Haziran 28, 2025, https://source.android.com/docs/security/bulletin/android-16
31. New Android Malware Surge Hits Devices via Overlays, Virtualization Fraud, and NFC Theft, erişim tarihi Haziran 28, 2025, https://thehackernews.com/2025/06/new-android-malware-surge-hits-devices.html
32. What is an APK File & How to Install Safely (2025) - Sphinx Solutions, erişim tarihi Haziran 28, 2025, https://www.sphinx-solution.com/blog/apk-files-the-ultimate-guide/
33. Application Development Step-by-Step Guide in 2025 | Sunbytes, erişim tarihi Haziran 28, 2025, https://sunbytes.io/blog/application-development-guide-2025/
34. Top SDLC Methodologies in 2025 | Agile, DevOps & More - Zignuts, erişim tarihi Haziran 28, 2025, https://www.zignuts.com/blog/top-sdlc-methodologies-in-2025