Gazi University Journal of Science
GU J Sci Part C
4(2):49-64 (2016)

Mobil Kötücül Yazılımlar ve Güvenlik Çözümleri Üzerine Bir İnceleme

Anıl UTKU[1,♠], İbrahim Alper DOĞRU[2]

[1] *Gazi Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, Maltepe, Ankara.*

[2] *Gazi Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü, Teknikokullar, Ankara.*

Başvuru: 19/01/2016 Düzeltme:04/04/2016 Kabul: 06/04/2016

ÖZ

Günümüzde mobil cihazlar her zaman ve her yerde farklı çeşitlerdeki servislere erişme imkânı sağlayarak hayatımızın önemli bir parçası haline gelmişlerdir. Son zamanlarda GSM, GPRS, Bluetooth ve Wi-Fi gibi mobil cihazlar tarafından kullanılan bağlantıların sayısının artmasıyla birlikte mobil cihazların zaman ve mekân kısıtlamaları ortadan kalkmıştır. Bu sebeple mobil iletişim kanallarını ve hizmetlerini istismar eden güvenlik açıklarının sayısında ve çeşitliliğinde artış yaşanmaktadır. Bu çalışma kapsamında mobil cihazlar için güvenlik çözümleri üzerine araştırmalar yapılarak kapsamlı bir bakış açısı sunma hedeflenmiştir. Mobil uygulamalardaki güvenlik açıkları, tehditler ve güvenlik çözümleri üzerine odaklanılmıştır. Kötücül yazılım tespit yöntemleri, mimariler, toplanan veriler ve işletim sistemlerine dayalı olarak mobil cihazları korumaya yönelik yaklaşımlar incelenmiştir.

Anahtar kelimeler: *Mobil kötücül yazılım, kötücül yazılım tespit yöntemleri, mobil güvenlik.*

A Review on Mobile Malware and Security Solutions

ABSTRACT

Nowadays, mobile devices have become an important part of our lives by providing the opportunity to access to different kind of services every time and everywhere. Recently, time and place constrains of mobile devices have been disappeared with increasing number of connections that is used by mobile devices such as GSM, GPRS, Bluetooth, Wi-Fi. Thus, number and kind of security vulnerabilities that misuse mobile communication channels and services have increased recently. In this study, it was targeted some researches on security solutions for mobile devices to present a comprehensive view. It was focused on security vulnerabilities, threats, and solutions on mobile applications. It was analyzed approaches which protect mobile devices based on malware detection techniques, architectures, collected data, and operating systems.

Keywords: *Mobile malware, malware detection techniques, mobile security.*

1. GİRİŞ

Günümüzde mobil cihazlar kişisel bilgisayarların sağladığı IEEE 802.11, Bluetooth, GSM, GPRS, UMTS ve HSPA gibi bir çok bağlantı özelliğini kullanıcılarına sunmaktadır. Bu gibi çekici özellikleri sebebiyle yaygınlaşan mobil cihazlar, saldırganların hedefi durumuna gelmişlerdir. Başlangıçta standart işletim

♠Corresponding author, *e-mail: anilutku@gazi.edu.tr*

sistemi ile piyasaya sürülen mobil cihazlar, saldırganların yakaladıkları güvenlik açıklarıyla daha büyük saldırılar gerçekleştirmelerine imkân sağlamaktaydı. Son zamanlarda Symbian, Windows Mobile, Android ve iPhone cihazları için geliştirilen işletim sistemleri önemli bir pazar oranı elde etmişlerdir. IMS 2011 raporuna göre mobil cihazların sayısı 420 milyonu geçmiş olsa da mobil kötücül yazılımların sayısı, kişisel bilgisayarlara yönelik kötücül yazılımların sayısından daha azdır. Ancak ilerleyen zamanlarda bu sayının artarak kişisel bilgisayarlara yönelik kötücül yazılım sayısını geçmesi beklenmektedir. Kullanıcıların mobil cihazlarına yükledikleri üçüncü parti uygulamalar, hassas internet alışverişleri ve bankacılık uygulamalarında tehdit olarak ön plana çıkmaktadır [1].

Son on yılda öğretim, ulaşım ve trafik, çevrimiçi arama ve satın alma, haber ve eğlence, sosyal ağlar, eğitim, çevresel izleme ve sağlık yönetimi gibi alanlarda pek çok akıllı telefon uygulaması geliştirilmiştir. Mobil iletişim cihazlarının ve uygulama servislerinin gelişmesi çevrimiçi kullanıcı davranışlarında radikal bir değişim yaşanmasına neden olmuştur. Akıllı telefonlar, mobil iletişim için birincil cihaz haline gelmektedir. Bu nedenle, güvenli mobil ve kablosuz ağ uygulamaları, internet tabanlı iş ortamlarında çalışan işletmeler için de çok önemlidir [2]. Mobil cihazların güvenlik saldırılarına karşı hassas olmaları ve elde edilebilecek bilgilerin çeşitliliği kötücül uygulama yazarlarını cezbetmektedir. Kullanıcıların mobil cihazlar üzerinden gerçekleştirdiği çevrimiçi bankacılık ve alışveriş gibi hassas finansal işlemler, kötü niyetli kullanıcılar için önemli bir kazanç kapısı olabilmektedir. Ayrıca Android işletim sistemini kullanan mobil cihazların önemli bir pazar payına sahip olması ve Android sistemlerin açık kaynak kodlu çekirdek yapısına sahip olması kötücül uygulama yazarlarının Android platformuna yoğunlaşmasına neden olmuştur. Google'ın üçüncü parti uygulamaların geliştirilmesini teşvik eden pazarlama stratejisinin bir sonucu olarak kötücül uygulamaların oluşturulması ve yayımlanması saldırganlar için kolaylaşmaktadır [3].

İnsanların günlük yaşamlarında mobil iletişim hizmetleriyle yüksek düzeyde etkileşim halinde olup akıllı telefonlarında kısa mesajlarını, adreslerini, fotoğraflarını, konum bilgilerini ve hatta kurumsal verilerini sakladıkları için bu cihazlar veri hırsızlığının temel hedefi haline gelmişlerdir. Kötücül yazılımlar şifre dinleme, klavye dinleme, ağ izleme, arka kapı erişimi, casusluk ve rootkiting gibi amaçlar için tasarlanmıştır. Kötücül amaçlarla veriye erişim davranışları genel olarak bir hedefe bulaşma, belirlenen kötücül faaliyetleri gerçekleştirme ve diğer cihazlara yayılma başlıkları altında incelenebilir [2].

2. LİTERATÜRDEKİ ÇALIŞMALAR

Mobil cihazların günlük hayata getirdikleri kolaylıklar ve yenilikler sayesinde, her zaman ve her yerde kullanıldıkları ve yaygınlaştıkları gerçeği yadsınamaz. Günümüz mobil cihazları, 10 yıl önceki kişisel bilgisayarlara oranla daha güçlüdürler. Ayrıca mobil cihazlar taşınabilirlikleri sayesinde kullanıcılar için daha ilgi çekicidirler. Ayrıca sundukları uygulamalar ile kullanıcı etkileşimlerini en üst seviyeye çıkarmayı hedeflemektedirler. Ancak, bu popülerlik ciddi güvenlik ve gizlilik tehditleri ile çeşitli kötü niyetli faaliyetleri de beraberinde getirmektedir. Kötü niyetli faaliyetler kullanıcıdan gizlenmekte ve arka planda faaliyete geçmektedir.

Android işletim sistemini kullanan mobil cihaz sayısının yüksek bir pazar oranına sahip olması nedeniyle kötücül yazılım geliştiriciler bu platformu hedef almaktadırlar. Ayrıca Android işletim sisteminin Blackberry, iOS ve Windows işletim sistemlerinden farklı olarak açık kaynak kodlu olması Android işletim sistemini saldırıların hedefi konumuna getirmiştir. Bu gibi ilgi çekici özellikleri sebebiyle Android işletim sistemini hedef alan kötücül yazılımları tespit etmeye yönelik yapılan çalışmalar incelenmiştir.

Dini ve ark. tarafından 2012 yılında yapılan çalışmada, Android kötücül yazılımlar için çok seviyeli bir anomali tespit sistemi geliştirilmiştir. Geliştirilen sistem, Android sistemleri çekirdek seviyesinde ve kullanıcı seviyesinde izleyerek makine öğrenmesi yöntemleri kullanarak kötücül davranışların normal davranışlardan ayırt edilmesini sağlamaktadır. Önerilen yöntemin ilk prototipinin gerçek kötücül yazılımları tespit ettiği ortaya konulmuştur [4].

Khune ve Thangakumar tarafından 2012 yılında yapılan çalışmada, ağ üzerindeki kötücül davranışları tespit etmek için derinlemesine analizler sağlayan Android cihazlar için bulut tabanlı bir saldırı tespit ve kurtarma sistemi önerilmiştir. Geliştirilen mekanizma, bulut ortamında sanallaştırılmış ve senkronize edilmiş sanal bir cihaz ile analizler yapmaktadır. Analiz edilen simüle edilmiş cihaz, saldırı durumunda cevap oluşturabilmek için çoklu paralel algılama motorları, bellek tarayıcıları ve anomali tespit edildiğinde verilecek sistem çağrılarını içermektedir. Saldırı durumunda yapılacak eylemler arasında cihazda yüklü olan aracı birimin, gerekli durumlarda cihazdaki bilgileri kurtarma işlemi vardır [5].

La Polla ve ark. tarafından 2013 yılında yapılan çalışmada, mobil cihazlar için güvenlik çözümleri üzerine araştırmalar yapılarak kapsamlı bir bakış açısı sunmak hedeflenmiştir. 2004-2011 yılları arasında kullanıcı uygulamalarındaki güvenlik açıkları, tehditler ve güvenlik çözümleri üzerine odaklanılmıştır. Farklı kategoriler halinde algılama ilkeleri, mimariler, toplanan veriler ve işletim sistemlerine dayalı olarak mobil cihazları korumaya yönelik yaklaşımlar özellikle IDS tabanlı modeller ve araçlar üzerinde durularak incelenmiştir. Bu sınıflandırma ile her bir yaklaşımın benimsediği modelin altında yatan kolay ve özlü görünüm açığa çıkarılmaya çalışılmıştır [1].

Rastogi ve ark. tarafından 2013 yılında yapılan çalışmada, AppsPlayground adı verilen sistem ile akıllı telefon uygulamalarını otomatik olarak analiz eden bir yapı sunulmuştur. AppsPlayground bu amaçla, farklı algılama ve otomatik arama tekniklerini içeren çoklu bileşenleri bütünleştirmektedir. İyi huylu ve kötücül

uygulamaları içeren deneyler kullanılarak geliştirilen sistem değerlendirilmiştir [6].

Shabtai ve ark. tarafından 2014 yılında yapılan çalışmada, mobil uygulamaların ağ davranışlarındaki sapmaları tespit etmek için davranış tabanlı yeni bir anomali tespit sistemi sunulmuştur. Önerilen sistemin temel amacı mobil cihaz kullanıcılarını ve hücresel altyapı şirketlerini kötü niyetli uygulamalardan korumaktır. Bu amaç mobil cihazları kötücül saldırılardan ya da cihaza yüklü kötücül uygulamalardan korumak ve popüler uygulamalara enjekte edilen kötücül kod parçalarının belirlenmesi adımlarıyla gerçekleştirilmektedir. Google Play Store üzerinde bulunan uygulamaların kendilerini güncelleme yetenekleri ile mobil kötücül yazılımların tespit edilmesine çalışılmıştır. Kötücül yazılım tespiti yalnızca uygulamaların ağ trafiği örüntülerine dayanarak yapılmıştır. Her uygulama için kendine özgü trafik örüntüsünü temsil eden bir örnek cihaz üzerinden öğrenilmiş ve yarı denetimli makine öğrenmesi yöntemleri, normal davranış örüntülerini öğrenme ve uygulamanın beklenenden farklı olan davranış sapmalarını tespit etmek için kullanılmıştır [7].

Seo ve ark. tarafından 2014 yılında yapılan çalışmada, kötücül yazılımların özellikleri tartışılmış ve olası mobil saldırı senaryoları açıklanmıştır. Ayrıca Android uygulamalarının güvenlik açıklarını tespit eden statik analiz aracı DroidAnalyzer sunulmuştur. Çalışmada çeşitli mobil kötücül yazılımlar ile bankacılık, uçuş izleme, rezervasyon, ev ve ofis izleme gibi potansiyel olarak hedef olabilecek uygulamalar DroidAnalyzer ile analiz edilmiştir [8].

Chen ve ark. tarafından 2015 yılında yapılan çalışmada, internetteki iletişim özelliklerine göre mobil kötücül yazılımların anormal davranışlarının tespit edilebilmesi için yeni bir yöntem önerilmektedir. Önerilen yöntem ile kötücül yazılımların uzak sunuculara verilerin iletimi için kurdukları dış bağlantılar ve iletişim paketleri izlenerek anormal davranışlar tespit edilmektedir. Önerilen yöntem HTTP POST / GET paketlerini belirlemek ve hassas verilerin iletimi için kontrol ve uzak sunucu doğrulaması aşamalarından oluşan bir kontrol yapısı içermektedir. Kötücül yazılımların ve normal yazılımların davranış özellikleri, geliştirilen yöntemin etkinliğini doğrulamak için karşılaştırılmıştır. Sonuçlar, önerilen yöntemin etkili bir şekilde karmaşık algılama yapıları oluşturulmasına gerek kalmadan anormal internet davranışlarını tanımladığını ortaya koymuştur. Geliştirilen yöntem tüm mobil işletim sistemi platformlarındaki anomalileri saptamaktadır [2].

Feizollah ve ark. tarafından 2015 yılında yapılan çalışmada, etkili bir algılama sistemi geliştirmek amacıyla mevcut olan yüzlerce özellik içinden bir özellik alt kümesi seçilmiştir. Mobil kötücül yazılım tespitinde özellik seçimi bakış açısı ile 2010 ve 2014 yılları arasında yayınlanan 100 adet araştırma çalışması incelenmiştir. Mevcut özellikler statik özellikler, dinamik özellikler, hibrid özellikler ve uygulamaların metadataları şeklinde dört gruba ayrılmıştır. Ayrıca, kullanılan değerlendirme önlemlerinin ve analizlerinin yanı sıra son araştırmalarda kullanılan veri kümeleri de tartışılmıştır [9].

Arankumar ve ark. tarafından 2015 yılında yapılan çalışmada, mobil ortamlarda yaşanabilecek olası saldırılar ve son kullanıcıların gizlilikleri için kullanılabilecek mekanizmalar açıklanmıştır. Ayrıca mobil ortamlarda kullanılan farklı gizlilik sağlama yöntemleri ve ilerleyen zamanlarda yaşanabilecek zorluklar açıklanmıştır [10].

3. MOBİL KÖTÜCÜL YAZILIMLAR

Mobil cihaz piyasasında Android, iOS, Windows Phone ve BlackBerry gibi çok sayıda farklı mobil işletim sistemi vardır. CNET tarafından yapılan araştırmada 2013 yılının üçüncü çeyreğinde toplam 261.100.000 cihazın % 81.3' ünün Android işletim sistemi kullanan cihaz olduğu belirlenmiştir. Bu araştırma Android işletim sistemini kullanan cihazların mobil cihaz piyasasına egemen olduğunu göstermiştir. Android işletim sisteminin bu popülaritesi yapılan saldırı sayısında artışa neden olmaktadır. F-Secure raporunda, Android kötücül yazılım sayısının 2012 yılında tüm kötücül yazılımların % 79' unu, 2011 yılında % 66.7' sini, 2010 yılında ise sadece % 11.25' ini oluşturduğunu göstermiştir. Benzer şekilde, Symantec Android kötücül yazılım sayısının Haziran 2012 ve Haziran 2013 arasında neredeyse dört kat arttığını belirtmiştir. Android zararlı yazılımlardaki muazzam artışın nedeni Android' in açık kaynak kodlu bir işletim sistemi olmasıdır [9].

Kötücül yazılımlar sistemlere müdahale etmek, cihazları devre dışı bırakmak, kullanıcı bilgilerini elde etmek veya mobil cihazları uzaktan kontrol etmek için tasarlanmış kod parçalarıdır [1]. Bu yazılımlar internetten indirilen dosyalar ya da harici medya kaynaklarından edinilen dosyalar yoluyla eriştikleri sistemlerin açıklarını denetler ve savunmasız sistemleri devre dışı bırakır. Kötücül yazılımların en sofistike ve tehlikeli olanlarından adware olarak bilinen reklam amaçlı olanlarına kadar birçok çeşidi vardır. En tehlikeli ve sofistike olan kötü amaçlı yazılımlar cihaz üzerindeki kişisel verilere erişmenin yanı sıra mobil cihazları hackleme yeteneğine sahiptirler. Saldırganların bazı kötücül yazılımlar yoluyla mali çıkar sağladıkları tespit edilmiştir. Bu kötücül yazılımlar yükleme sonrasında, kullanıcıların bilgisi olmadan belirli numaralara kısa mesaj göndermekte ve telefon aramaları gerçekleştirmektedir. 2013 yılında yayınlanan bir rapor, bazı saldırganların kötü amaçlı yazılımlar yoluyla ayda 12.000 dolar kadar kazanç sağladığını göstermektedir. Botnet olarak anılan kötü niyetli uygulamalar ise daha tehlikelidirler. Bu uygulamalar cihazlara bulaştıktan sonra saldırgan, cihazlara erişim ve cihazdaki uygulamaları kontrol ederek kötü niyetli faaliyetler gerçekleştirebilir. Son zamanlarda, saldırganların mobil cihazlara yönelik yeni bir yaklaşım geliştirdikleri görülmüştür. Bugüne kadar saldırganlar kullanıcıların bilgisi olmadan kötü niyetli etkinlikleri gerçekleştiriyorken yeni yaklaşımla kötü niyetli uygulamaları indirmek için kullanıcıların ilgilerini çekerek Android cihazları bir kanal gibi kullanarak kullanıcıları yönlendirdikleri tespit edilmiştir [9].

Kötücül yazılımların kitlesel bir iletişim aracı haline gelen internet üzerindeki yayılma oranı endişe verici rakamlara ulaşmıştır. Günümüzdeki mobil kötücül yazılımlar kullanıcıların kişi listelerinin ya da cihazda bulunan bilgilerinin aktarımı, özel SMS ya da MMS mesajları gönderilmesi, uzaktan erişim sağlama ve cihazların tamamen kilitlenmesi gibi yeteneklere sahiptir [11]. Kötücül yazılımlar virüsler, solucanlar, Truva atları, rootkitler ve botnetler olarak gruplandırılabilir. Virüsler kendilerini çoğaltarak dosyalara bulaşan kötücül yazılım türüdür. Solucanlar, herhangi bir kullanıcı müdahalesi olmadan var olan ağ üzerinde, farklı taşıma mekanizmaları kullanarak bir cihazdan diğerine kendisini kopyalayan programlardır. Truva atları ise bazı işlevleri gerçekleştirme hedefiyle piyasaya sürülen kötü amaçlı kod parçaları gizlenmiş yazılımlardır. Rootkitler işletim sistemlerini etkileyerek uzaktaki saldırganlara yönetim yetkisi veren yazılımlardır. Botnetler ise uzaktaki bir saldırgana tam yönetim hakkı verilen birden fazla cihazdan oluşur. Mobil kötücül yazılımlar, kullanıcıların virüslü ek dosyaları, Bluetooth yoluyla alınan dosyalar, MMS veya SMS ile birlikte gelen dosyalar ve çeşitli Web sitelerine yönlendirilen linklere tıklanması gibi çeşitli ve farklı yollarla yayılabilir. Akıllı telefonlara yönelik olan kötücül yazılımların temel hedefi, kullanıcıların kişisel verilerinin çalınması olarak gösterilebilir. Mart 2011' de Google kötücül yazılım içerdiği tespit edilen uygulamaların Android marketten ve kullanıcıların cihazlarına uzaktan bağlanılarak silindiğini duyurmuştur. Android uygulamalarının markete yüklenirken izlenmemesi, kötücül yazılım geliştiricilerin Android işletim sistemlerini hedef almalarına sebep olmuştur [1].

Mobil kötücül yazılımlar yayılma davranışları, uzaktan kontrol edilme durumları ve kötücül saldırı davranışlarına göre kategorize edilebilir. Yayılma davranışı kötücül yazılımın hedef cihaza nasıl iletildiğini ifade etmektedir. Uzaktan kontrol edilme durumu, kötücül yazılımın enfekte olduğu mobil cihazın uzak sunuculardan yönetilmesini ifade etmektedir. Saldırı davranışı ise kötücül yazılımın kurban cihaza ulaştıktan sonra Bluetooth gibi farklı iletişim kanalları üzerinden nasıl saldırılar gerçekleştirebileceğini ifade etmektedir [12]. Kötücül yazılımlar cihazlara yüklendikten sonra depolanmış verilere erişme, cihaz fonksiyonlarına müdahale etme ya da güvenlik açıkları oluşturarak yetkisiz erişim sağlama gibi faaliyetler yürütebilirler [13]. Kötücül yazılımlar yemleme (phishing), casusluk (spyware), gözetim saldırıları, Premium arama ve SMS (diallerware), mali amaçlı saldırılar, solucan tabanlı saldırılar ve botnetler gibi tehditler oluşturabilmektedir. Yemleme saldırıları gerçek gibi görünen e-posta veya SMS' ler vasıtasıyla kullanıcıların kredi kartı, hesap ya da kimlik bilgilerinin elde edilmesidir. Casusluk saldırıları kullanıcı davranışlarının izlenmesi ve kullanıcı bilgilerinin elde edilmesidir. Elde edilen bilgiler (konum gibi) daha sonra reklam mailleri gönderme gibi amaçlar için kullanılabilmektedir. Gözetim saldırıları mobil cihazların sahip olduğu mikrofon, kamera ve GPS gibi sensörler kullanılarak kullanıcıların izlenmesidir. Diallerware saldırıları kullanıcıların haberleri olmadan arka planda premium aramalar yaparak ve SMS' ler göndererek kullanıcılara finansal açıdan zarar vermeyi amaçlamaktadır. Finansal saldırılar kullanıcıların bankacılık ya da çevrimiçi alışveriş işlemlerinde ortadaki adam saldırısı (man-in-the-middle) olarak bilinen yöntemle araya girerek kullanıcıların kredi kartı bilgilerinin çalınmasıdır. Solucan tabanlı saldırılar bir cihazdan diğerine kullanıcıların müdahalesi olmadan ağ üzerinde farklı araçlar kullanarak kendisini kopyalan solucanlar vasıtasıyla gerçekleştirilmektedir. Botnet saldırıları ise saldırganın uzaktan kontrol edebildiği ve kötücül yazılımlardan etkilenmiş bir dizi mobil cihaz yoluyla gerçekleştirilmektedir [12].

4. MOBİL CİHAZLARA YÖNELİK SALDIRILAR VE TESPİT YÖNTEMLERİ

Kişisel bilgisayarlar ve mobil cihazlar benzerlikler gösteriyor olsalar da güvenlik konusunda birkaç önemli farklılık ön plana çıkmaktadır. İlk olarak kötücül yazılım üreticileri mobil cihazlar üzerinden maddi kazanç sağlamayı hedeflemektedir. Kişisel bilgisayar ortamlarına göre kolay bir şekilde mobil ortamlarda yapılan bu aktivitelere premium aramalar ve Premium SMS' ler örnek verilebilir. İkincisi akıllı telefonlarda yapılan her bir aktivitenin operatör tarafında bir ücretlendirmesi vardır. Bu aktivite kullanıcının bilgisi dâhilinde yapılsa da, kötücül yazılımlar tarafından kullanıcının bilgisi dâhilinde olmadan yapılsa da ücretlendirme yapılmaktadır.

Kişisel bilgisayarlarla karşılaştırıldığında güvenlik sorunlarının mobil cihazlarda farklılık gösterdiğini destekleyen diğer bir durum ise mobil cihazların her zaman ve her yerden internete bağlanabilmelerini sağlamak için kullanılan farklı teknolojilerdir. Mobil cihaz güvenliğinin, kişisel bilgisayar güvenliğinden ayrılan noktaları literatürde aşağıdaki başlıklar altında incelenmiştir.

- Hareketlilik: Cihazlar kullanıcılarıyla birlikte her zaman ve her yere taşınabilirler. Bu da cihazların çalınması ya da zarar görmesi riskini doğurur.
- Güçlü kişiselleştirme: Mobil cihazlar genel olarak kullanıcısına özeldir ve kullanıcısı tarafından kişiselleştirilmiştir.
- Güçlü bağlanabilirlik: Kullanıcılar mobil cihazlarını kullanarak e-posta gönderebilir, banka hesaplarını kontrol edebilir, internet hizmetlerine ve Web sitelerine erişebilir. Ancak bu bağlantılar yoluyla ya da SMS ve MMS gibi servislerde kötücül yazılımların cihazlarına bulaşma riskini yaşarlar.
- Teknolojik kapasite: Mobil cihazların aynı anda birçok teknolojik alt yapıyı kullanıyor olması, saldırganlara da farklı yöntemler geliştirmesi için fırsat vermektedir.
- Azaltılmış yetenekler: Mobil cihazlar küçültülmüş bilgisayarlar olsalar da örnek olarak tam manasıyla bir klavyeye sahip olmamaları eksiklik olarak düşünülebilir.

Mobil cihazların sınırlı CPU ve hafıza donanımları, kişisel bilgisayarlarda gerçek zamanlı uygulamalarda kullanılan saldırı tespit sistemlerinin tam anlamıyla uygulanamamasına neden olur. Ayrıca mobil cihazların sınırlı batarya kapasitesi, kişisel bilgisayarlardaki güvenlik çözümlerinin mobil cihazlara uygulanamamasına neden olur [1].

Mobil cihazlara yönelik saldırılar kötücül yazılım saldırıları, spyware ve grayware olmak üzere üç farklı saldırı türünden oluşmaktadır. Kötücül yazılımlar kullanıcı verilerini elde etmek, kullanıcıları rahatsız etmek ya da cihazlara zarar vermek amacıyla mobil cihazlara erişmektedir. Kötücül yazılımlar kullanıcılar tarafından cihazlarına yüklendikten sonra saldırganlar, sistemde bulunan güvenlik açıklarından yararlanarak kullanıcı bilgilerine erişebilir ayrıca uzaktan yetkisiz erişimler sağlayabilir. Truva atları, solucanlar, botnetler ve virüsler gibi farklı çeşitleri bulunan kötücül yazılımlar başta Amerika Birleşik Devletleri olmak üzere birçok ülkede yasa dışı bir faaliyet olarak belirlenmiştir ve bu ihlali gerçekleştirenler hakkında hapis cezaları uygulanmaktadır. Spyware belirli bir süre boyunca kullanıcı cihazlarından konum bilgisi ve metin mesajı bilgileri toplayan yazılım türüdür. Kullanıcı cihazlarına fiziksel erişim elde eden saldırganlar, kişiye özgü spyware yazılımlarını kullanarak kullanıcıların bilgisi olmadan cihazlara yazılım yüklemesi gerçekleştirirler. Kişiye özgü casus yazılımlar kullanılarak saldırganlara hedef cihazdaki bilgiler sızdırılmaktadır. Kullanıcı izni olmadan casus yazılımların yüklenmesi yasadışı bir faaliyet olarak ön plana çıkmaktadır. Grayware yazılımları pazarlama stratejileri ya da kullanıcı profilleri oluşturmak için mobil cihazlardan bilgi toplamaktadırlar. Grayware' ler kullanıcı verilerini sızdırsalar da bu yazılımları kullanan şirketlerin amacı kullanıcılara zarar vermek değildir. Grayware aktiviteleri için herhangi bir cezai yaptırım olmasa da kullanıcılar veri gizlilik ihlalleri için itirazda bulunabilir. Uygulama marketleri grayware yazılımlarını tespit ettiklerinde bu yazılımları marketten kaldırma ya da içeriğine göre faaliyetini sürdürme kararı alabilirler [14].

4.1. Saldırıların hedefleri

Mobil cihazlara yönelik saldırıların hedefleri gizlilik, sniffing, hizmet reddi ve yüksek fatura üzerine olabilir.

- Gizlilik saldırıları, telefonların çalınması ya da kaybedilmesi durumlarında ortaya çıkabilmektedir. Mobil cihazlar, laptoplardan daha küçük olmaları sebebiyle kaybolma ya da çalınma durumları daha sık yaşanabilmektedir. Mobil cihazın kaybolması ya da çalınması durumunda cihaza üçüncü kişiler tarafından casus yazılımlar yüklenmesi ya da kişinin mesajlarının okunması ve kişi listelerine erişilmesi gibi durumlar gizlilik ihlallerine neden olabilmektedir.
- Sniffing saldırıları mikrofon, kamera, GPS gibi sensörler kullanılarak yapılan saldırılardır. Bu sensörler uygulama çeşitliliği sağlamalarının yanı sıra kullanıcı gizliliğini tehlikeye atabilmektedir. Bir mobil cihaz saldırgan tarafından ele geçirilirse, saldırgan cihazda depolanan verilere erişmek ve aynı zamanda kullanıcının eylemlerini kaydetmek için sensörleri kullanabilir.
- Hizmet reddi saldırıları: Denial-of-Service (DoS) saldırıları ile saldırgan, bir servisin veya cihazın hizmet verememesine neden olur. Akıllı telefonlara yönelik DoS saldırıları sınırlı donanım özellikleri nedeniyle güçlü bağlantı ve düşük kapasite ile gerçekleştirilir.
- Yüksek fatura saldırıları, kurbanın hesabına ek ücret yansıtılması veya saldırganların kurbanların hesabına ekstra ücret devretmesi ile gerçekleşir. Bazı kablosuz servislerde ödeme başına kullanım sözleşmeleri düzenlendiği için bu saldırılar, kablosuz akıllı telefonlara yönelik gerçekleştirilmektedir [1].

4.2. Kötücül yazılım tespit yöntemleri

Kötücül yazılım tespit yöntemleri, analizcilere riskleri ve kötücül kod parçaları ile ilişkili hedefleri anlamak için yardımcı olur. Bu şekilde elde edilen bakış açıları kötücül yazılım geliştirme konusundaki yeni trendleri anlamada ya da gelecek tehditler için önleyici tedbirler almak için kullanılabilir. Kötücül yazılım analizinden elde edilen özellikler grubu, bilinmeyen kötücül yazılımların tespit edilmesinde ya da bilinen kötücül yazılım ailelerinin sınıflandırılmasında kullanılabilir. Kötücül yazılımlar, arka planda çalışan kod parçacıklarının, yazılımın fonksiyonlarının ve davranışlarının analizleri yapılarak tespit edilebilir.

Blackberry, iOS ve Windows işletim sistemlerinden farklı olarak Android işletim sistemi açık kaynak kodlu yapısı sebebiyle uygulamaların kaynak koduna erişilmesine izin vermektedir. Uygulama içi istenen izinler, API çağrıları ve imza analizi yöntemleri ile kötücül yazılım tespiti gerçekleştirilebilmektedir. Bu sebeple mobil kötücül yazılım tespiti üzerine yapılmış çalışmalar Android işletim sistemi üzerine odaklanmaktadır.

Android kötücül yazılım tespit yöntemleri statik analiz, dinamik analiz, izin tabanlı analiz, imza tabanlı analiz ve makine öğrenmesi yöntemlerine dayalı analizler olarak kategorize edilebilir.

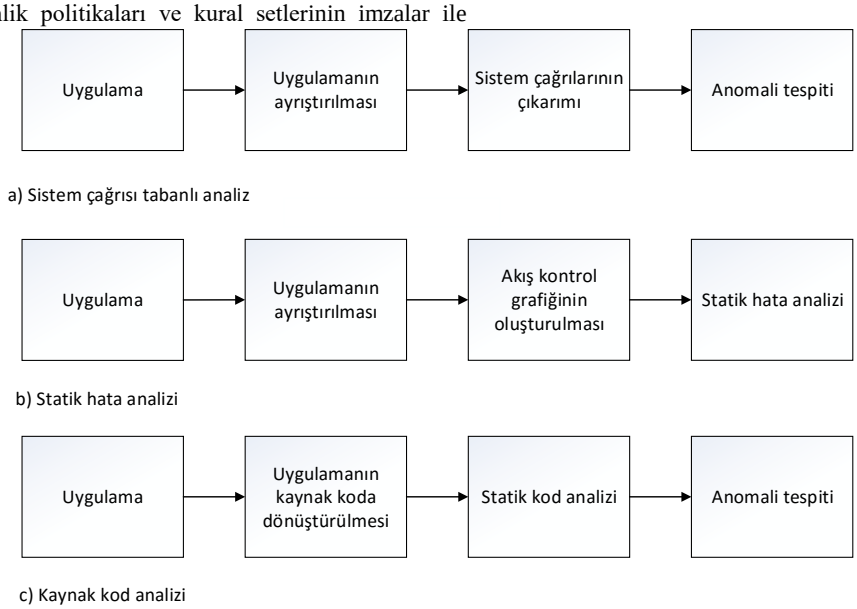
4.2.1. Statik analiz

Statik analizler uygulamaların çalıştırılmadan, yazılım özelliklerinin ya da kaynak kodlarının incelenmesi yoluyla gerçekleştirilir. Statik analiz yöntemleri, uygulamaların cihaza enfekte olmadan analiz edilmesini sağlar. Statik analiz yöntemleri, Android uygulamalarının manifest dosyalarından Uygulama Programlama Arayüzü (API) çağrılarını ve izin bilgilerini alarak kötücül yazılım tespiti gerçekleştirir. Statik analiz yöntemlerinde genel olarak düşük seviyeli kod parçalarına ayırma, sistem çağrıları, hassas API' lerin kullanımı, depolama aygıtlarına erişim gibi önemli özelliklerin çıkarımı ile örüntü veya imza tabanlı sınıflandırıcı uygulamaları gerçekleştirilmektedir [15]. Statik analiz yöntemleri ile kötücül kod parçaları içeren, kullanıcı etkileşimi olmadan maddi zararlara neden olan ve dinamik olarak şifreli kod parçaları yükleyen uygulamalar tespit edilebilmektedir [16].

Statik analiz yöntemleri kullanılarak gerçekleştirilen AndroidLeaks ile farklı Android marketlerinden indirilen 24.350 uygulama analiz edilmiş ve potansiyel olarak gizlilik sorunları yaratabilecek 7.414 uygulama tespit edilmiştir. Manuel yapılan doğrulamalar ile 2.342 uygulamanın mikrofon ile kaydedilen ses dosyaları, GPS konum bilgileri, Wi-Fi verileri ve telefon verileri dâhil olmak üzere özel verileri sızdırma eğiliminde olduğu tespit edilmiştir. Statik analiz yöntemlerinin avantajları verimlilik ve ölçeklenebilirlik, belirli örüntülere dayalı olan basit tahmin yöntemleri ile öğrenme tabanlı modellere dayalı olmalarıdır [15]. Kötücül kod parçalarının uygulama yazılımlarının içine gizlenerek gömülmesi ya da şifrelenmesi statik analiz tekniklerini zorlaştırmaktadır. Statik analiz yöntemleri, geleneksel antivirüsler tarafından kullanılan kötüye kullanım tespiti ve anomali tespiti şeklinde kategorize edilebilir [17].

- Kötüye kullanım tespiti: Kötüye kullanım tespiti, güvenlik politikaları ve kural setlerinin imzalar ile eşleştirilmesine dayalı olarak kötücül yazılım tespiti gerçekleştirir.
- Anomali tespiti: Anomali tespitinde bilinen kötücül yazılımları öğrenme ve bilinmeyen kötücül yazılımları tahmin etmek için makine öğrenmesi algoritmaları kullanılır. Bu yaklaşım örüntülerin yerine kötücül faaliyetlerin kullanıldığı durumlar için uygundur. Burada yöntemler uygulamaların şüpheli davranışlarını belirlemek için kullanılır ve daha sonra gözlemlenen imzalar normal davranışlar sergileyen uygulamaların imzaları ile karşılaştırılır. Destek vektör makinesi (SVM) gibi sınıflandırıcılar ile ağ eğitilerek kötücül ve normal davranışlar ayırt edilebilmektedir [17].

Chandramohan ve ark. tarafından 2012 yılında yapılan çalışmada statik analiz yöntemleri Şekil 1' de görüldüğü gibi sistem çağrıları tabanlı, statik hata analizi ve kaynak kod analizi şeklinde kategorize edilmiştir [18].



Şekil 1. Statik analiz yöntemleri [18]

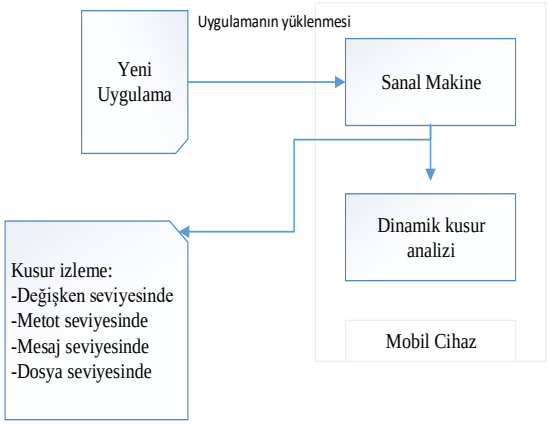
- Sistem çağrısı tabanlı analiz: Mobil uygulamalar ilk olarak IDA Pro gibi araçlar kullanılarak ayrıştırılmıştır. Sistem çağrılarının elde edilmesini sağlayan bu araçtan sonra anomali tespiti yapmak ve kötücül faaliyet gösteren uygulamaları sınıflandırmak için Centroid makinelere geçilmiştir [18].
- Statik hata analizi: Egele ve ark. tarafından 2011 yılında yapılan çalışmada, iOS uygulamalarının binary dosyaları üzerinde statik hata analizi gerçekleştirilmiştir. Yapılan çalışmada üçüncü parti yazılım geliştiriciler tarafından oluşturulmuş iOS uygulamalarının yaratabileceği tehditler üzerinde durulmuştur. Çalışmalar gizlilik ihlallerini otomatik olarak tespit edebilen PiOS adı verilen bir araç geliştirilerek yürütülmüştür. PiOS aracı uygulamaların hassas bilgilere erişim kontrollerini ve ağ üzerinden bilgi aktarım durumlarını belirlemek için statik analiz kullanmaktadır. PiOS yazılımı ilk olarak uygulamaların binary dosyalarından akış kontrol grafiği (CFG) oluşturmaktadır. CFG tarafından yürütülecek binary dosyalarının çıkarımı için IDA Pro yazılımı kullanılmıştır. CFG ile sınıf hiyerarşisi oluşturulmakta ve metot çağrıları çözümlenmektedir. Uygulamaların erişebilirlik analizleri CFG' ler ile gerçekleştirilmiştir. iOS uygulamaları tarafından cihaz ID' leri, adres defteri, konum bilgisi, fotoğraf galerisi, e-posta hesap bilgileri, Wi-Fi bağlantı bilgileri, arama detayları, Safari tarayıcısının ayarları ve tarama geçmişi ile klavye önbelleğine erişilebilmektedir. 1.400 'den fazla iPhone uygulaması için yapılan çalışmanın sonuçları, uygulamaların yarısından fazlasının cihaz ID'sini sızdırdığı tespit edilmiştir. Bu bilgi sızıntısı üçüncü parti uygulama geliştiricilerin, kullanıcıların uygulama tercihleri ve kullanım alışkanlıklarını elde etmesine neden olmaktadır. Çalışmada ayrıca çoğu uygulamanın adres defteri, konum bilgisi, telefon numarası, Web tarayıcı geçmişi

ve fotoğraflar için de bilgi sızıntısı oluşturduğu tespit edilmiştir [19].

- Kaynak kod analizi: Enck ve ark. tarafından 2011 yılında yapılan çalışmada, Android Market uygulamalarının güvenilirliği incelenmiştir. 21 milyon satır kod, akış kontrol analizi, veri akış analizi, yapısal analiz ve semantik analizler ile güvenlik açıklarına karşı incelenmiştir. Analiz sonuçları, uygulamaların % 50' den fazlasının telefon numarası gibi bilgilerin yanı sıra Uluslararası Mobil Ekipman Kimliği (IMEI), Entegre Devre Kartı Kimliği (ICC-ID) ve Uluslararası Mobil Abone Kimliği (IMSI) bilgilerini sızdırdığını göstermiştir. Ayrıca uygulama geliştiricilerinin güvenli yazılım geliştirme süreçlerini uygulamadıkları tespit edilmiştir [20].

4.2.2. Dinamik analiz

Dinamik analizler uygulamaların davranışlarını izlemek için izole bir ortamda uygulamanın yürütülmesini gerektirir. Statik analiz yöntemlerinin aksine uygulama yürütüldüğü için kötücül davranışların gözlemlenmesini sağlar [21]. Dinamik analizler ağ etkinliği izleme, dosya değişikliklerini izleme ve sistem çağrıları gibi yöntemlerle gerçekleştirilebilir. Android uygulamaları, telefonların arama özellikleri hariç bütün yazılım ve donanım özellikleriyle mobil cihaz emülatörleri vasıtasıyla masaüstü bilgisayarlarda çalıştırılabilir. Test amaçlı emülatörler Android Sanal Aygıt (AVD) yapılandırmalarını desteklemektedir. Uygulamalar emülatör üzerinde çalıştırıldığında, ağ durumu, video oynatma, veri depolama gibi tüm hizmetlere erişim sağlayabilmektedir. Konsol çıktısı olarak yapılan telefon çağrıları, SMS' ler ve olay günlükleri elde edilebilir [17]. Dosyalar üzerindeki değişiklikler, ağ etkinlikleri, süreçler ve sistem çağrısı izleme gibi uygulamalar, dinamik analiz ile elde edilmektedir. Şekil 2' de dinamik analiz görülmektedir [5].



Şekil 2. Dinamik analiz [5]

Korumalı alanlar, kötücül yazılımların etkilemesine maruz kalmadan güvenilmeyen uygulamaları çalıştırmak için korunan ortamlardır. Örneğin, izole edilmiş bir sanal makine kötü niyetli uygulamaları test etmek için bir korumalı alan (sandbox) olarak kullanılabilmektedir. Araştırmacılar, bu sayede bulaşma veya hassas bilgilerin sızdırılması riski olmadan zararlı aktiviteleri gözlemleyebilmektedir [4].

Isohara ve ark. tarafından 2011 yılında yapılan çalışmada, Android kötücül yazılım tespiti için çekirdek tabanlı bir davranış analiz yöntemi sunulmuştur. Önerilen sistem, log dosyası toplama modülü ve log dosyası analiz modülünü içermektedir. Linux katmanında yer alan log dosyası toplama modülü tüm sistem çağrılarını kaydetmektedir. Süreç yönetimi ve dosyaların giriş çıkış durumları kötücül yazılım tespiti için önemli olduğundan, log aktivitelerinden sonra bir filtreleme mekanizması ile hedef uygulamanın faaliyetleri filtrelenmektedir. Veri analizi modülünde, kötücül aktiviteleri tespit etmek için imzalar ile aktiviteler, log analiz modülü tarafından karşılaştırılmaktadır. Analiz sonuçları geliştirilen prototip ile kötücül yazılım tespitinde başarı elde edildiği ortaya konulmuş ancak log analiz modülünün mobil cihazlar için aşırı kaynak tüketimine sebep olduğu belirtilmiştir [22].

4.2.3. İzin tabanlı analiz

Sandbox ortamında çalışan uygulamalar, belirli verilere erişmek için izinlere ihtiyaç duyarlar. Android uygulamaları yüklenirken kullanıcılarına uygulamanın etkinliğini sürdürebilmesi için gerekli olan bazı izinleri kabul edip etmediğini sormaktadır [21]. Bu izinler Android uygulamaları analiz edilirken önemli rol oynamaktadır. İzinler yüklenen her uygulama için Manifest.xml dosyasında listelenir. Uygulamalar yüklenirken verilen izinler uygulama davranışlarını kısıtlar ve güvenlik açıklarını azaltır. Android cihazlardaki kaynak kullanımları bu izin kümelerine dayalıdır [17].

Johnson ve ark. tarafından 2012 yılında yapılan çalışmada, Android marketten otomatik olarak uygulama indiren bir sistem geliştirilmiştir. Kategorilere göre indirilecek uygulamaları aramak için farklı algoritmalar kullanılmıştır. Statik analiz ile uygulamaların işlevselliğine göre gerekli izinler edilmiştir. İzinlerin isimleri Android kaynak kodunda aranmış ve istenen izinlerin verilip verilmediğini belirlemek için izinler API çağrıları ile eşleştirilmiştir. Uygulama dosyalarında bulunan izinler AndroidManifest.xml dosyası ile karşılaştırılarak ekstradan istenen izinler analiz edilmektedir [23].

4.2.4. İmza tabanlı analiz

Android kötücül yazılımları tespit etmek için kullanılan imza tabanlı algılama sistemleri, kötücül yazılımlardan toplanan imza verileriyle uygulama verilerinin karşılaştırmasını yapmaktadır. Bilinen kötücül yazılımları tespit etmek için bir imza olarak Java kodu ve sınıfları kullanılır. Bununla birlikte, bilinmeyen kötücül yazılımlar tespit edilememektedir [9].

Zheng ve ark. tarafından 2013 yılında yapılan çalışmada, kötücül yazılımları otomatik olarak toplayan, yöneten, analiz eden ve ayıklayan DroidAnalytics adı verilen imza tabanlı bir sistem geliştirilmiştir. Geliştirilen sistem ile 102 farklı aileden 150.368 kötücül yazılım

analiz edilmiş ve 2.494 adet kötücül yazılım tespit edilmiştir [24].

4.2.5. Makine öğrenmesi yöntemlerine dayalı analizler

Android kötücül yazılım türevlerinin istikrarlı yükselişi sebebiyle bu yazılımların etkili bir şekilde tespit edilmesi hayati önem taşımaktadır. Araştırmacılar imza tabanlı yöntemlerin sınırlamalarını aşmak için makine öğrenmesi yöntemlerini kullanmaya başlamışlardır. Uygulamalardan toplanan verilere dayalı olarak makine öğrenmesi algoritmaları ile sistemler eğitilmektedir ve bu sayede yeni kötü amaçlı yazılımları tespit etmek için gerekli olan anomaliler belirlenebilmektedir [25]. Makine öğrenmesi yöntemlerinde, özellik seçimi ilk ve en önemli adımlardandır. Android uygulamaları izinler, Java kodları, belgelendirmeler, cihaz ve ağ üzerindeki davranışlar ile uygulama davranışı gibi çeşitli elemanlardan oluşmaktadır. Özellik seçiminin faydalarından bazıları şunlardır:

- Özellik seçimi veri setlerinin boyutunu azaltmak için yapılmaktadır. Az veri ile kolayca veri görselleştirmesi yapılabilmektedir.
- Veri kümesinin azaltılması sadece deney maliyetinden tasarruf etmeye değil aynı zamanda gerçek zamanlı uygulamalarda işlem süresinden tasarruf etmeyi sağlar. Özelliklerin kullanışlı alt kümelerinin seçilmesi, eğitim aşamasında makine öğrenmesi algoritmalarının çalışma zamanını azaltmaktadır.
- Özellik seçimi ile makine öğrenmesi algoritmalarında daha doğru sonuçların alınması sağlamak için veri setlerinden gürültülü ve ilgisiz veriler kaldırılır.

Kötücül yazılım tespitinde makine öğrenmesi yöntemlerinin etkinliğini incelemek için Feizollah ve ark. tarafından yapılan çalışma kapsamında iki adet analiz gerçekleştirilmiştir. MalGenome veri setinden kötücül ve kötücül olmayan 800' ün üzerinde Android uygulamasından ağ trafiği toplanmıştır. Veri kümesi 504.148 kayıt içermektedir. Komşulukların sayısı üç olarak belirlenmiş ve K-en yakın komşu sınıflandırıcısı kullanılmıştır. Analiz sonuçları farklı özelliklerin veri toplama süreci ve kullanılan sınıflandırıcının aynı olmasına karşı farklı sonuçlar verdiğini göstermiştir [9].

Nataraj ve ark. tarafından 2011 yılında yapılan çalışmada, kötücül yazılımları sınıflandırmak ve görselleştirmek için görüntü işleme yöntemlerine dayalı yeni bir yaklaşım sunulmuştur. Öklid uzaklığı yöntemi ile K-en yakın komşu yöntemi kötü amaçlı yazılım sınıflandırma için kullanılmıştır. Yapılan çalışmada kötücül yazılım aileleri benzer renk tonlarıyla eşleştirilerek standart görüntü özelliklerine göre yeni bir sınıflandırma yöntemi önerilmiştir. Kod ayrıştırma ya da uygulamanın çalıştırılması gerekmeden kötücül yazılımların tespit edildiği çalışmada 25 farklı aileden 9.458 kötücül azılım % 98 başarı oranıyla sınıflandırılmıştır [26].

Rieck ve ark. tarafından 2011 yılında yapılan çalışmada, makine öğrenmesi yöntemleri kullanılarak kötücül davranışları otomatik olarak analiz etmek için yeni bir yaklaşım sunulmuştur. Yapılan çalışmada çok sayıda kötücül yazılım örneği toplanarak bir sandbox ortamında kötücül yazılımların davranışları takip edilmiştir. Bilinen kötücül yazılımlarla benzer davranışlar sergileyen yeni kötücül yazılımlar kümeleme yöntemleri ile tanımlanmışlardır. Bilinmeyen kötücül yazılımlar sınıflandırma yöntemleri kullanılarak sınıflara atanmışlardır [28].

Kong ve ark. tarafından 2013 yılında yapılan çalışmada, kötücül yazılımların yapısal bilgilerine (fonksiyon çağrısı grafikleri) dayalı, otomatik kötücül yazılım sınıflandırılması gerçekleştiren yeni bir yaklaşım sunulmuştur. Her bir kötücül yazılım örneği için fonksiyon çağrıları çıkarıldıktan sonra fonksiyon çağrısı grafikleri arasındaki benzerlikler değerlendirilmiştir. Çalışmada Windows tabanlı 11 farklı aileden kötücül yazılımlar ile geliştirilen sistem test edilmiş ve yüksek sınıflandırma doğruluğu elde edilmiştir [27].

Nari ve ark. tarafından 2013 yılında yapılan çalışmada, kötücül yazılımların ağ davranışlarına dayalı olarak sınıflandırılması için yeni bir yaklaşım sunulmuştur. Uygulamaların ağ davranışlarına göre davranış profilleri oluşturulmuştur ve sınıflandırma algoritmaları kullanılarak kötücül yazılımlar sınıflandırılmıştır. Analiz sonuçları J48 karar ağacı yönteminin diğer sınıflandırma yöntemlerine göre daha iyi performans gösterdiği sonucuna varılmıştır [29].

Santos ve ark. tarafından 2013 yılında yapılan çalışmada, kötücül yazılımların statik ve dinamik analizlerinden elde edilen özellik kümesi kullanılarak OPEM adı verilen hibrid bir algılama sistemi geliştirilmiştir. Statik özellikler işlemsel kod modellemelerinden, dinamik özellikler ise sistem çağrıları ve yürütülen faaliyetlerin izlenmesi yoluyla elde edilmiştir. Geliştirilen sistem karar ağacı, k-en yakın komşu, Bayes ağları ve destek vektör makinesi gibi farklı makine öğrenmesi algoritmaları ile iki farklı veri seti üzerinde test edilmiştir. Test sonuçları, önerilen yaklaşımın hem statik analiz yöntemlerine hem de dinamik analiz yöntemlerine göre daha yüksek performans gösterdiği görülmüştür [30].

Islam ve ark. tarafından 2013 yılında yapılan çalışmada, statik ve dinamik özellikler kullanarak uygulamaları kötücül ya da iyicil olarak sınıflandırılmak için bir sistem geliştirilmiştir. Geliştirilen sistem 2003-2007 yılları ile 2009-2010 yılları arasında toplanan 2398'i kötücül olmak üzere toplam 2939 yazılımdan oluşan iki adet veri seti üzerinde destek vektör makinesi, IB1, karar ağacı ve random forest yöntemleri ile test edilmiştir. Test sonuçları başta random forest olmak üzere bütün sınıflandırma yöntemlerinin yüksek performans gösterdiğini ortaya koymuştur [31].

Android kötücül yazılım tespit yöntemleri statik analizler, dinamik analizler, izin tabanlı analizler, imza tabanlı analizler ve makine öğrenmesine dayalı analizler başlıkları altında incelenmiştir. Literatürde bulunan Android kötücül yazılım tespitine yönelik çalışmalar kullanılan teknik, algoritma, özellik, veriseti, değerlendirme metriği ve başarı oranı açısından karşılaştırmalı olarak Tablo 1' de verilmiştir [32-38].

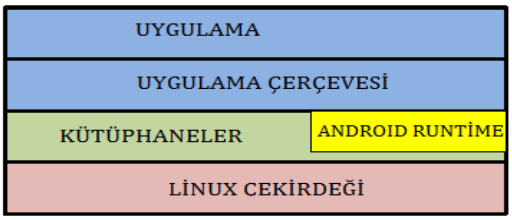
Tablo 1. Kötücül yazılım tespitine yönelik yapılan çalışmalar ve karşılaştırmaları

Makale	Teknik	Algoritma	Özellikler	Veriseti	Değerlendirme Metriği	Başarı Oranı
Kim ve ark. (2011)	Dinamik Analiz	x^2 Mesafesi	Şarj durumu	-	TPR	% 99 doğruluk oranı
Shabtai ve ark. (2012)	Dinamik Analiz	Naive Bayes, Karar Ağacı, K-Means, Logistic Regresyon	Çeşitli sistem özellikleri	20 iyicil oyun, 20 kötücül yazılım tespit uygulaması ve 4 kötücül uygulama	FPR, TPR, AUC, Accuracy	% 99' un üzerinde doğruluk oranı
Lu ve ark. (2013)	Dinamik Analiz	Naive Bayes, Chi-Square	Kötücül davranışlar	447 uygulama örneği	FPR, FNR, Sınıflandırma doğruluğu	% 89 doğruluk oranı
Yerima ve ark. (2014)	Makine Öğrenmesine Dayalı Analiz	Naive Bayes	Uygulama izinleri	Genome Project ve 100 iyicil uygulama	Hassasiyet, Doğruluk, TNR, FPR, TPR, FNR	% 93' ün üzerinde doğruluk oranı
Shen ve ark. (2014)	Statik Analiz	VF2	Android uygulama bileşenleri	Genome Project	Detection rate	% 86,36
Sheen ve ark. (2015)	İzin Tabanlı Analiz-Statik Analiz	Ensemble	Uygulama izinleri ve API çağrıları	Genome Project	Precision, Recall, F-measure	% 88 doğruluk oranı
Kabakuş ve ark. (2015)	İzin Tabanlı Analiz	Logistic Regresyon	Uygulama izinleri	Genome Project, Drebin	Accuracy (Doğruluk), Specificity (Çeşitlilik)	% 88 doğruluk ve % 92,5 çeşitlilik oranı

Literatürde Android kötücül yazılım tespiti için yapılmış olan çalışmalarda temel olarak statik ve dinamik analiz yöntemlerine odaklanılmıştır. Ancak hibrid bir sistem geliştirilmeyip bu yöntemlerin tek başına kullanılması elde edilen başarı oranının sınırlı kalmasına yol açmaktadır. Dinamik analiz çalışmalarında makine öğrenmesi yöntemleri kullanılarak geliştirilecek hibrid sistemlerde başarı oranının daha yüksek olacağı öngörülebilmektedir. Ayrıca kötücül yazılım tespiti için kullanılan verisetlerinin eskimesi sebebiyle günümüzdeki kötücül yazılım aktivitelerinin belirlenmesi zorlaşabilmektedir.

5. GÜVENLİK ÇÖZÜMLERİ

Mobil cihazlara yönelik tehditler ve güvenliğin sağlanması konusundaki zorluklar göz önüne alındığında, yeni güvenlik çözümlerinin geliştirilmesi gerekliliği ortaya çıkmaktadır. Bu güvenlik çözümleri mobil cihazlar için yeni ve güvenli bir sistem mimarisi oluşturulması ile mevcut güvenlik mimarilerinin yeniden değerlendirilmesini ve geliştirilmesini gerekli kılar. Android işletim sistemleri Şekil 3' de görüldüğü gibi Linux çekirdeği, kütüphaneler (Android Runtime), uygulama çözümü ve uygulama katmanları olmak üzere dört katmandan oluşmaktadır. Temel olarak Android Runtime kütüphaneleri Linux çekirdeği ile uygulamalar arasında bir köprü konumundadır.



Şekil 3. Android işletim sisteminin katmanları [39]

Android sistemlerde yaygın olarak kullanılan güvenlik özellikleri süreç ve dosya sistemi izolasyonu ile uygulama veya kod imzalamadır. Ancak Android işletim sistemlerinin temel güvenlik sorunu bir uygulamanın güvenli olup olmadığı kararının son kullanıcıya bırakılmasıdır. Google, Google Play üzerindeki uygulamaları tarayarak tek parça bir güvenlik katmanı ekliyor olsa da son kullanıcıların

analizlerini yapıp uygulamayı kurup kurmama konusunda karar vermeleri gerekmektedir. Google' ın mümkün oldukça fazla uygulama sunarak daha büyük büyük bir pazar payı elde etme stratejisi, kötücül uygulama geliştiricilere ortam hazırlamaktadır. Örnek olarak Google Play Store üzerinde bulunan droid09 isimli kötücül yazılım, banka bilgilerini taklit ederek kullanıcıların banka giriş bilgilerini elde etmeye çalışmaktadır. Günümüzde daha da sofistike bir hale gelen kötücül yazılımlar, sistemlerdeki güvenlik açıklarından yararlanarak ya da kullanıcıları kandırarak tehdit oluşturmaktadırlar [39].

Arabo ve Pranggono tarafından 2013 yılında yapılan çalışmada, Şekil 4' de görüldüğü gibi mobil cihazlar için güvenlik çözümlerinin eklendiği çok katmanlı bir mimari sunulmuştur [40].

Şekil 4. Mobil cihazlar için güvenlik çözümleri eklenmiş mimari yapı [40]

- Son kullanıcılar: Son kullanıcıların güvenlik bilincine sahip olmaları her zaman önemlidir. Son kullanıcılarda aşağıdaki güvenlik adımları için farkındalık oluşturulması gerekmektedir.

- Kötücül yazılımlara ve virüslere karşı mobil cihazlara antivirüs yazılımları yüklemek ve otomatik güncellemeleri açık tutmak,
- Sadece güvenilir kaynaklardaki uygulamaları yüklemek,
- Uygulamaların resmi sitelerinden indirilmesi (Örneğin Instagram uygulaması sadece kendi Web sitesinden ya da yapımcısına dikkat edilerek marketten indirilmelidir),
- Uygulamaların yüklenmesi esnasında istenen izinlerin dikkatli bir şekilde incelenmesi,
- İşletim sisteminin ve uygulamaların güncellemelerinin kontrol edilmesi,
- Cihazların çalınması riskine karşı uzaktan yedek alma ve sıfırlama yazılımlarının yüklenmesi,
- Pozitif görüş bildirilen ve yüksek indirilme oranına sahip olan uygulamaların indirilmesi,
- Herhangi bir parola ya da şifreleme yöntemi kullanmayan açık kablosuz ağlarda hassas bilgilerle ilgili işlemlerin yapılmaması,
- Cihaz davranışındaki anomalilere karşı dikkatli olmak,
- Sosyal paylaşım siteleri üzerinde verilen linklere karşı dikkatli olmak son kullanıcıların alması gereken önlemler olarak söylenebilir.

- Mobil ağ operatörleri: Ağ operatörleri, müşterine karşı daha güvenli bir ortam hazırlamaktan sorumludur. Çoğu kötücül yazılımın ya da botnet cihazların bilgi sızdırma ya da faaliyet göstermek için SMS ve MMS gibi kanalları kullandığı düşünüldüğünde, ağ operatörlerinin mobil ağ üzerinden gelen ve giden mesaj trafiği için antivirüs yazılımları yüklemesi gerektiği ortaya çıkmaktadır. Ağ operatörlerinin, kötücül yazılımların yayılmasını önlemek için diğer operatörlerle bilgi, veritabanı ve uzmanlık paylaşımı yapması gerekmektedir.

- Uygulama Geliştiriciler: Uygulama geliştiricilerin, kendi uygulamalarında gerekli güvenlik önlemlerini almaları gerekmektedir. Verilerin http ya da TLS gibi ağlar üzerinden şifreli olarak gönderildiğinden emin olunmalıdır. Geliştiriciler, uygulamanın düzgün bir şekilde çalışması için gerekli olmadığı sürece istenen uygulama içi izinleri en aza indirmelidir. Android cihazlarda (CALL_PHONE) ve (SEND_SMS) gibi yaklaşık 100 adet izin isteği mevcuttur. Ayrıca uygulama geliştiriciler sadece gerekli olduğu durumlarda veri toplanacak şekilde uygulamalarını tasarlamalıdır. Kötücül uygulamalar genel olarak belirli ülkeler ya da belirli kullanıcı türlerinin ilgi alanlarına göre geliştirilmektedir. Örnek olarak Rusya için geliştirilmiş kötücül uygulamalar ile Çin için geliştirilmiş kötücül uygulamalar farklılık göstermektedir.

- Uygulama marketleri: Uygulama marketlerinin, uygulamaları piyasaya sürmeden önce titizlikle incelemeleri gerekmektedir. Uygulamalar kötü niyetli kod içerip içermedikleri, güvenilirlikleri, uygulama kılavuzunda açıklanan faaliyetleri gösterip göstermediği ve güvenlik açıkları konusunda dikkatli bir şekilde incelenmelidir [40].

6. KÖTÜCÜL YAZILIM TESPİTİNDE KULLANILAN VERİ SETLERİ

İlk kötücül yazılım olan FakePlayer' in 2010 yılında keşfedilmesinden sonra araştırmacılar kötücül yazılımların tespit edilebilmesi için mevcut kötücül yazılımların bilgilerinin elde edilmesinin önemli olduğunu belirlemişlerdir. Bu amaçla aktif kötücül yazılım örneklerinin toplanması için iki yaklaşım benimsenmiştir. Bunlardan ilki araştırmacılar ve antivirüs şirketlerinden Android kötücül yazılımlarla ilgili duyurular, tehdit raporları ve olay günlüklerini takip ederek yeni Android kötücül yazılımların bilgilerini elde etmek ve daha sonra bilgi alınan yerlerden kötücül yazılım talebinde bulunmaktır. İkincisi ise Android Market ya da üçüncü parti uygulamalar arasında kötücül yazılım taraması yapmaktır.

Felt ve ark. tarafından 2011 yılında yapılan çalışmada, IOS, Symbian ve Android sistemlerdeki kötücül

yazılımlarla ilgili olarak açık antivirüs veritabanlarından bilgi toplanmıştır. Yapılan çalışma kapsamında Ocak 2009 ile Haziran 2011 yılları arasında 46 farklı mobil kötücül yazılım türü tespit edilmiştir. Kötücül yazılım tespiti için Symantec, F-Secure, Fortiguard, Lookout, and Panda Security antivirüs firmalarının veritabanları incelenmiş ve kötücül yazılımlarla ilgili rapor ve açıklamalar takip edilmiştir. Çalışmanın amacı kötücül yazılımlarının arka planlarını görmek ve hedeflerini belirleyerek geliştirilecek savunma mekanizmasını şekillendirmektir [14].

Jiang ve Zhou tarafından 2012 yılında yapılan çalışmada, Ağustos 2010 ile Ekim 2011 arasında 49 kötücül yazılım ailesinden 1260 kötücül yazılım örneği toplanmıştır. Oluşturulan bu veri seti Android Malware Genom Projesi kapsamında Mayıs 2012' de yayınlanmıştır. Oluşturulan veri seti beş kıtadan 160'tan fazla üniversite, araştırma laboratuvarları ile firmalarla paylaşılmıştır [39]. Android Malware Genom projesi kapsamında üretilen veri seti literatürde bulunan ilk kötücül yazılım veri setidir. Araştırmacılar, geliştirdikleri uygulamaların sonuçlarını test etmek için bu veri setini kullanmışlardır [40].

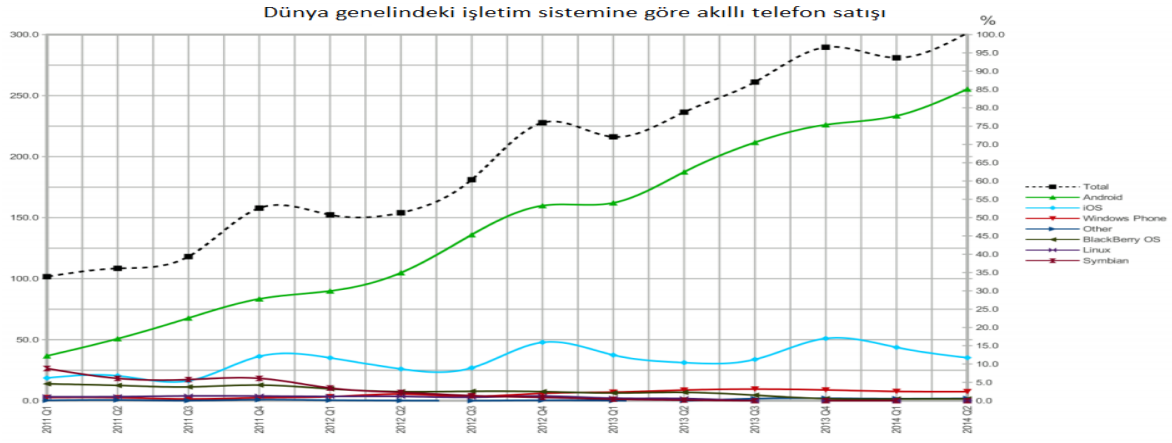
Arp ve ark. tarafından 2014 yılında yapılan çalışmada, gerçek Android uygulamaları ve gerçek kötücül yazılımlar kullanılarak bir veri seti oluşturulmuştur. Oluşturulan başlangıç veri kümesinde 131.611 adet kötücül ve kötücül olmayan yazılım mevcuttur. Veriler Ağustos 2010 ile Ekim 2012 tarihleri arasında toplanmıştır. Veri seti Google Play Store' dan edinilen 96150 uygulama, Çin uygulama marketlerinden edinilen 19545 uygulama, Rus uygulama marketlerinden edinilen 2810 uygulama ve Android web siteleri, kötücül yazılım forumları ve bloglar gibi diğer kaynaklardan elde edilen uygulamalardan oluşmaktadır. Oluşturulan final veri seti 123.453 kötücül olmayan uygulama ve 5.560 kötücül uygulamadan oluşmaktadır [41].

Lindorfer ve ark. tarafından 2014 yılında yapılan çalışmada, ANDRUBIS adı verilen Android sistemler için bir kötücül yazılım tespit sistemi geliştirilmiştir. ANDRUBIS Dalvik Sanal Makinası ve sistem düzeyindeki işlemlerle dinamik analizleri ve çeşitli uyarım teknikleri ile statik analizleri birleştirmektedir. ANDRUBIS sistemi geliştirilirken % 40' ı kötücül olmak üzere 1.000.000 Android uygulamasından bir veri kümesi oluşturulmuştur [42].

7. KÖTÜCÜL YAZILIM TESPİT ARAÇLARI

Mobil cihaz teknolojilerinin ilerlemesi, kullanıcılara mobil cihazlarında kullanabilecekleri farklı binlerce mobil uygulamanın ortaya çıkmasına neden olmuştur. Bu uygulamalar kullanıcıların her zaman ve her yerden erişimlerine açıktır. Mobil cihazlar tarafından sunulan hizmetler Uluslararası Telekomünikasyon Birliği' ne (ITU 2013) göre, dünya çapındaki cep telefonu aboneliklerinin artmasına katkı sağlamıştır. 2013 yılında mobil kullanıcı sayısı % 96' ya tekabül eden 6,8 milyar kullanıcıya ulaşmıştır [43].

Günümüzde kullanılmakta olan mobil cihazların çoğu farklı işletim sistemleri kullanmaktadır. Apple tarafından kullanılan IOS, Google tarafından desteklenen Android ile Blackberry, Symbian ve Windows Mobile mobil cihazlar için kullanılan işletim sistemlerine örnek olarak verilebilir. Bu beş popüler mobil işletim sistemi arasında Google'ın desteklediği Android işletim sistemi, mobil işletim sistemi pazarına hâkim olmuştur. Şekil 5' de mobil cihazların, işletim sistemlerine göre kullanım oranları görülmektedir. Android, 2013 yılının üçüncü çeyreğinde % 80 pazar payı ile diğer işletim sistemlerini geride bırakmıştır [44]. Android cihazlar açık kaynak olması, performans ve özelleştirme kolaylıkları açısından diğer işletim sistemlerini kullanan mobil cihazlara göre daha yüksek bir oranda tercih edilmektedir. Android işletim sistemindeki büyüme, Android cihazlar için kötücül yazılım tespiti konusundaki ilerlemeye de katkıda bulunmuştur [38]. Kaspersky Lab tarafından yapılan araştırmalar, mobil kötücül yazılımların % 98,05' inin Android platformu için geliştirilmiş olduğunu ortaya çıkarmıştır. Kötücül yazılımlar kullanıcıların kişisel bilgilerinin, etkinliklerinin, konumlarının depolanması, izinsiz SMS ve MMS gönderilmesi, bellek ve pil ömrünün kısaltılması gibi olumsuz durumlara yol açabilmektedir [45].



Şekil 5. Dünya genelindeki işletim sistemine göre akıllı telefon satış oranları [44]

Mobil cihazlar üzerinde tespit edilen kötücül yazılımların sayısı, keşfedildikleri 2010 yılından beri önemli ölçüde artış göstermiştir. Kötücül yazılım tespiti amacıyla, mobil cihazlarda antivirüs yazılımları gibi geleneksel yaklaşımlar benimsenmiştir. Geleneksel antivirüs yaklaşımları imza tabanlı oldukları ve kullanıcı etkileşimlerini sürekli izledikleri için mobil cihazlar açısından kullanışlı olmamaktadır. Kötücül yazılım oluşturan kişilerin amacı bu yazılımları mümkün oldukça fazla cihaza bulaştırmak olduğundan Apple veya Google'ın kullandığı uygulama marketlerini hedef almaktadırlar. Ancak Android cihazlar üçüncü parti uygulamaların yüklenmesine izin verdikleri için risk ölçütü artış göstermektedir. Android hedefli kötücül yazılımlar Truva atı, casus yazılım, kök izni edinme, yükleyici ve botnet olarak gruplandırılabilir [46].

Android tabanlı cihazlarda kötücül yazılımların tespit edilmesi için kullanılan birçok araç mevcuttur.

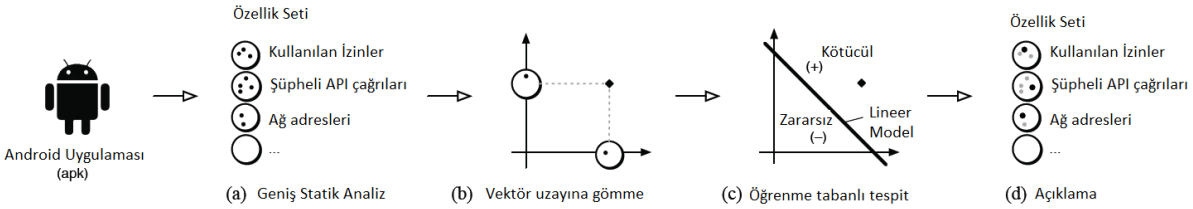
- Kirin, Android tabanlı sistemlerde kullanılan sertifika gerektiren mantık tabanlı bir araçtır.
- SCanDroid, Android uygulamalarındaki gizlilik ihlallerini tespit etmek statik analize dayalı bir yöntem sunmaktadır.
- TaintDroid, sistem üzerindeki bilgi sızma durumlarını izlemektedir. Özel veriler kusurlu sayılır ve bu verilerin Android cihazda bulunması durumunda kullanıcıya raporlanır.
- DroidBox gizlilik sızıntılarını tespit etmek için TaintDroid kullanmaktadır. Aynı zamanda Android API' leri izlenerek ve dosya sistemlerindeki ağ etkinliği, şifreli işlemler ve cep telefonu kullanım bilgileri gözlemlenmektedir.
- ComDroid, kullanıcının izni dışında gönderilen isteklerin tespit edilmesini sağlamaktadır.
- Crowdroid, kullanıcı etkileşimi gerektiren sistem çağrıları sırasında uygulamanın vermiş olduğu çağrı sayısını analiz ederek, Android akıllı telefonlar üzerinde Truva atı gibi kötücül yazılım analizi gerçekleştirmektedir.
- DroidRanger, üçüncü parti uygulama paketlerini ve kötücül yazılım sınıflandırmasını elde etmek için uygulamadan çıkarılan diğer özellikleri kullanmakta ve aynı zamanda kod tarafından yapılan sistem çağrılarını ayıklayarak girişimleri bu kodu gizlemek için aramaktadır.
- DroidMat, izinler, bileşenlerin dağıtımı, mesajlar ve API çağrıları da dâhil olmak üzere birçok özellik kullanır. Aynı zamanda iyi niyetli ya da kötü niyetli olarak uygulamaları sınıflandırmak için kümelemelerini farklı türde gerçekleştirmektedir.
- Andrubis, bilinmeyen uygulamaları analiz etmek için kullanılan bir platformdur. Kullanıcılarına, analiz işlemi bittikten sonra kötücül yazılım değerlendirmesi de dâhil olmak üzere ayrıntılı bir rapor sunmaktadır.
- MADAM, hem çekirdek düzeyinde hem de kullanıcı düzeyinde Android kötü amaçlı yazılım tespit etmek için 12 adet özellik kullanır. K-en yakın Komşu (KNN) algoritması kullanılarak MADAM ile başarılı 10 kötücül yazılım için % 93 doğruluk oranı elde edilmiştir.
- Andromaly, sürekli olarak akıllı telefon özelliklerini ve olayları izleyen bir ana bilgisayar tabanlı kötü amaçlı yazılım algılama sistemidir. Andromaly akıllı telefon ve faaliyetlerinden CPU kullanımı kadar, bu davranışları tanımlamak için 88 özellik gözlemleyerek kullanıcının davranışlarını izleyen makine öğrenmesi tekniklerine dayanmaktadır.
- RobotDroid, mobil cihazlar bilinmeyen kötü amaçlı yazılım tespit etmek için SVM sınıflandırıcısına dayanmaktadır.
- TStructDroid, izlenen verilerin analizi, teorik analiz, zaman serileri özelliği günlüğü, segmentasyon ve veri frekans bileşen analizi ve öğrenilen bir sınıflandırıcı kullanarak gerçek zamanlı kötücül yazılım algılama sistemi oluşturmaktadır. TStructDroid, %98 doğruluk hassasiyeti göstermektedir.
- STREAM, pil, bellek, ağ ve izin gibi seçilmiş özellikleri toplamak için bir Android emülatörü kullanır. Buradan elde edilen verilen makine öğrenmesi yöntemleri ile sınıflandırılır.
- A5, statik ve dinamik analiz yöntemlerini birlikte kullanan kötücül yazılım tespit araçlarındandır.
- Dendroid, Android kötücül yazılımlarının kod yapılarını sınıflandırmak ve analiz etmek için metin madenciliği ve bilgi erişim tekniklerini kullanan bir yaklaşımdır.
- DroidDolphin, kötü niyetli Android uygulamaları tespit etmek için büyük veri ve makine öğrenmesi yöntemlerine dayalı dinamik bir analiz çerçevesi sunmaktadır.
- DREBIN, donanım bileşenleri, izinler, uygulama bileşenleri ve API çağrıları gibi özelliklere göre makine öğrenmesi yöntemleri kullanarak cihaz üzerinde kötü amaçlı yazılım tespiti ve Android uygulamaların geniş bir analizini gerçekleştirir.

Kötücül yazılım tespit araçlarının kullandıkları yöntemlere göre karşılaştırmaları Tablo 2' de verilmiştir.

Tablo 2. Kötücül yazılım tespit araçlarının özellik karşılaştırmaları

Kötücül yazılım tespit aracı	Makine öğrenmesi	Manifest incelemesi	API analizi	İzin analizi
Kirin	X	X	X	
SCanDroid	X			
TaintDroid	X	X		
DroidBox	X	X		
ComDroid	X			
Crowdroid		X		
DroidRanger				
DroidMat				
Andrubis	X			
MADAM				
Andromaly		X		
RobotDroid		X		
TStructDroid		X		
STREAM		X		
A5	X			
Dendroid			X	X
DroidDolphin				
DREBIN				

Arp ve ark. Tarafından 2014 yılında yapılan çalışmada, Android tabanlı mobil cihazlar için bir kötücül yazılım tespit aracı olan DREBIN sunulmuştur [41]. DREBIN, Android işletim sistemini kullanan mobil cihazlarda, kötücül yazılımların belirlenmesini sağlayan bir tespit sistemidir. Şekil 6' da görüldüğü gibi DREBIN sınırlı kaynak kullanımı ile çalışma zamanında uygulamanın birçok özelliğini toplayarak geniş bir statik analiz gerçekleştirir. Bu özellikler kötücül yazılımların tipik örüntüleri ile otomatik olarak tanımlanır ve yöntemin kararlarını açıklamak için kullanılabilecek şekilde ortak bir vektör uzayına gömülür. 123.453 uygulama ve 5560 kötü amaçlı yazılım örneği için yapılan testler DREBIN' in % 94 başarı gösterdiğini ortaya koymuştur.



Şekil 6. DREBIN tarafından gerçekleştirilen analiz adımlarının şematik olarak gösterilmesi [4]

Android market üzerinde bulunan, Lookout Mobile Security ve McAfee' de dâhil olmak üzere popüler çözümlerin çoğu, kötücül yazılımlar tarafından kolayca işe yaramaz hale getirilebilmektedir. Mobil bir cihaz üzerindeki kötücül yazılımların tespit edilebilmesi için, DREBIN' in kötü niyetli faaliyet gösteren uygulamaların tipik hareketlerini belirlemesi gerekmektedir. Bu amaçla geliştirilen metot, farklı kaynaklardan özellik kümelerini ayıklamakta ve bir vektör uzayı ile geniş bir statik analiz kullanmaktadır. Bu işlem, Şekil 6 'da gösterilmiş ve aşağıda açıklanmıştır.

a) Geniş statik analiz: İlk adımda DREBIN, belirli bir Android uygulamasını denetler ve uygulamanın dex kodundan farklı özellik setlerini ayıklar.

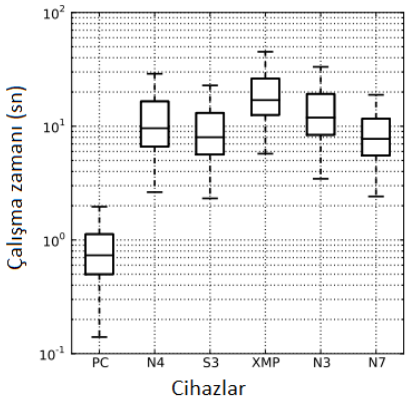
b) Vektör uzayına gömülmesi: Ayıklanan özellik kümeleri daha sonra özelliklerin örüntü ve birleşimlerine

göre geometrik olarak analiz edilebilen ortak bir vektör uzayına eşlenir.

c) Öğrenme-tabanlı algılama: Özellik setlerinin doğrusal olarak gömüldüğü Destek Vektör Makineleri gibi makine öğrenme teknikleri kullanılarak zararlı yazılımların tespit edilmesini sağlar.

d) Açıklama: Son aşamada, kötücül bir uygulama algılandığında, işlemleri açıklayan bir rapor kullanıcıya sunulmaktadır.

Mobil cihazların işlem gücü hızla artıyor olsa da hala normal masaüstü bilgisayarlara göre sınırlıdır. Sonuç olarak, bu cihazlar üzerinde doğrudan çalışması beklenen bir kötücül yazılım tespit sisteminin verimli bir şekilde çalışması beklenmektedir. DREBIN sistemi öğrenilen bir tespit modelinin bir Android uygulaması vasıtasıyla doğrudan mobil cihazları aktarılmasından oluşmaktadır. Bu sayede uygulama sadece 280 KB yer kaplamakta ve kaynak tüketimini minimum seviyede tutmaktadır. DREBIN' in uygulama performansı, Google Play Store' dan rastgele olarak seçilmiş 100 popüler uygulama ile karşılaştırılmıştır. Bu analiz için, dört akıllı telefon (Nexus 4, Galaxy S3, Xperia Pro Mini ve Nexus 3), bir tablet (Nexus 7) ve bir masaüstü bilgisayar dâhil olmak üzere çeşitli yaygın donanım yapılandırmalarını kapsayan aygıtlar seçilmiştir. Sonuçlar Şekil 7' de görüldüğü gibi DREBIN' in beş akıllı telefonda da ortalama 10 saniye içinde verilen bir uygulamayı analiz edebildiğini göstermiştir. Sony Xperia Mini Pro gibi eski modellerde, DREBIN' in ortalama 20 saniye içinde uygulamayı analiz edebildiği gösterilmiştir. Genel olarak, yapılan analizlerin tüm mobil cihazlarda 1 dakikadan daha kısa bir süre içinde tamamlandığı görülmüştür. Masaüstü bilgisayara (2.26 GHz 4GB RAM Core Duo) 100.000 adet uygulama için DREBIN' in taramayı bir günden daha az bir sürede tamamladığı ve uygulama başına 750 ms gibi bir analiz performansı elde ettiği ortaya konulmuştur. Yapılan analiz çalışmaları, 123.453 uygulama ve 5560 kötü amaçlı yazılım örneği için DREBIN' in % 94 başarı gösterdiğini ortaya koymuştur.



Şekil 7. DREBIN' in çalışma performansı [4]

DREBIN sistemi farklı kötücül yazılım tespit modellerini, makine öğrenmesi yöntemlerini kullanarak uygulamaktadır. Öğrenme teknikleri otomatik çıkarım modelleri için teoride güçlü bir araç sağlarken, pratikte veri temsili temelini gerektirmektedir. Bu durum DREBIN' in tespit modelinin kalitesinin kritik yapı taşının, kötücül ve iyicil uygulamaların bulunmasına bağlı olduğunu gösteren bir dezavantaj olarak ön plana çıkarmaktadır. Kötücül olmayan uygulamaların verilerini toplamak kolay olsa da kötücül yazılım örneklerinin verilerini toplamak teknik çaba gerektirmektedir. Bu sorunu aşmak için DroidRanger, AppsPlayground ve RiskRanker gibi çevrimdışı analiz yöntemleri ile otomatik olarak kötücül yazılım bilgileri ve güncellemeleri DREBIN için temsili veri setleri oluşturmada kullanılabilmektedir.

DREBIN' in bir diğer dezavantajı ise kötücül olmayan uygulamaların içine kötücül uygulamaların gizlenmesidir. Bu gibi yeniden paketleme, kodu tekrar düzenleme veya önemsiz kod ekleme gibi şaşırtmaca stratejileri, DREBIN' in performansını etkileyerek öğrenme ve tespit aşamasında sistemin yanılmasına neden olmaktadır.

8. SONUÇ

Birden çok bağlantı ve sensör gibi özellikler ile donatılmış mobil cihazların sayısındaki artış ile birlikte mobil kötücül yazılım sayısında da artış yaşanmaktadır. Mevcut sınırlı kaynaklar, güç ve işlem üniteleri gibi çok sayıda özellik, saldırganlar tarafından istismar edilebilmektedir. Bu çalışma kapsamında günümüzdeki mobil kötücül yazılım senaryoları tartışılmış, mobil cihazlara yönelik uygulama seviyesindeki kötücül yazılımlar kategorize edilmiş, saldırıların nasıl yürütüldüğü ve saldırganların amaçlarının ne olduğu tartışılmıştır. Ayrıca, saldırı tespiti ve güvenilir mobil platformlar temeline dayalı olarak mevcut mekanizmalara odaklanılarak mobil cihazlar için geçerli güvenlik çözümleri karşılaştırmalı olarak incelenmiştir.

KAYNAKLAR

[1] La Polla, M., Martinelli, F. ve Sgandurra, D., "A survey on security for mobile devices", IEEE Communications Surveys & Tutorials, Cilt 15, No 1, 446–471, 2013.

[2] Chen, P. S., Lin, S. ve Sun, C., "Simple and effective method for detecting abnormal internet behaviors of mobile devices", Information Sciences, Cilt 321, No C. 193-204, 2015.

[3] Damopoulos, D., Menesidou, S. A., Kambourakis, G., Papadaki, M., Clarke, N. ve Gritzalis, S., "Evaluation of Anomaly-Based IDS for Mobile Devices Using Machine Learning Classifiers", Security and Communication Networks, Cilt 5, No 1, 3-14, 2011.

[4] Dini, G., Martinelli, F., Saracino, A. ve Sgandurra, D., "MADAM: A Multi-Level Anomaly Detector for Android Malware", Computer Network Security, 240-253, 2012.

[5] Khune, R. S, ve Thangakumar, J., "A cloud-based intrusion detection system for Android

smartphones", Radar, Communication and Computing (ICRCC), 180-184, 2012.

[6] Rastogi, V., Chen, Y. ve Enck, W., "AppsPlayground: Automatic Security Analysis of Smartphone Applications", CODASPY'13, 2013.

[7] Shabtai, A., Tenenboim-Chekina, L., Mimran, D., Rokach, L., Shapira, B. ve Elovici, Y., "Mobile malware detection through analysis of deviations in application network behavior", Computers & Security, Cilt 43, 1-18, 2014.

[8] Seo, S., Gupta, A., Sallam, A. M., Bertino, E. ve Yim, K., "Detecting mobile malware threats to homeland security through static analysis", Journal of Network and Computer Applications, Cilt 38, 43-53, 2014.

[9] Feizollah, A., Anuar, N. B., Salleh, R. ve Abdul Wahab, A. W., "A review on feature selection in mobile malware detection", Digital Investigation, Cilt 13, 22-37, 2015.

[10] Arankumar, S., Srivatsa, M. ve Rajarajan, M., "A review paper on preserving privacy in mobile environments", Journal of Network and Computer Applications, Cilt 53, 74-90, 2015.

[11] Sawle, P. D. ve Gadicha, A. B., "Analysis of Malware Detection Techniques in Android", A Monthly Journal of Computer Science and Information Technology, Cilt 3, No 3, 176-182, 2014.

[12] He, D., Chan, S. ve Guizani, M., "Mobile application security: malware threats and defenses, Wireless Communications, IEEE, 22 (1). 138-144, 2015.

[13] Wu, F., Narang, H. and Clarke, D. (2014) An Overview of Mobile Malware and Solutions", Journal of Computer and Communications, Cilt 2, 8-17.

[14] Felt, A. P., Finifter, M., Chin, E., Hanna, S. ve Wagner, D., "A Survey of Mobile Malware in the Wild", SPSM '11 Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, 3-14, 2011.

[15] Shen, Y. C., Chien, R. ve Hung, S. H., "Toward Efficient Dynamic Analysis and Testing for Android Malware", IT CoNvergence PRActice (INPRA), Cilt 2, No 3, 14-23, 2014.

[16] Wang, X., Yang, Y. ve Zeng, Y., "Accurate mobile malware detection and classification in the cloud", SpringerPlus, 2015.

[17] Dua, L. ve Bansal, D., "Taxonomy: Mobile Malware Threats and Detection Techniques", Computer Science & Information Technology (CS & IT), 213-221, 2014.

[18] Chandramohan, M. ve Tan, H., "Detection of Mobile Malware in the Wild", Computer, Cilt 45, No 9, 65-71, 2012.

[19] Egele, M., Kruegel, C., Kirda, E. ve Vigna, G., "PiOS: Detecting Privacy Leaks in iOS Applications", Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS), 2011.

[20] Enck, W., Octeau, D., McDaniel, P. ve Chaudhuri, S., "A Study of Android Application Security", Proceedings of the 20th USENIX Security Symposium, 2011.

[21] Ramu, S., "Mobile Malware Evolution, Detection and Defense", EECE 571B, Term Survey Paper, 2012.

[22] Isohara, T., Takemori, K. ve Kubota, A., "Kernel-based Behavior Analysis for Android Malware Detection", Computational Intelligence and Security (CIS), 1011-1015, 2011.

[23] Johnson, R., Wang, Z., Gagnon, C. ve Stavrou, A., "Analysis of android applications' permissions", Software Security and Reliability Companion (SERE-C), 45-46, 2012.

[24] Zheng, M., Sun, M. ve Lui, J. C. S., "DroidAnalytics: A Signature Based Analytic System to Collect, Extract, Analyze and Associate Android Malware", Trust, Security and Privacy in Computing and Communications (TrustCom), 163-171, 2013.

[25] Gandotra, E., Bansal, D. ve Sofat, S., "Malware Analysis and Classification: A Survey", Journal of Information Seurity, Cilt 5, 56-64, 2014.

[26] Nataraj, L., Karthikeyan, S., Jacob, G. ve Manjunath, B., "Malware Images: Visualization and Automatic Classification", Proceedings of the 8th International Symposium on Visualization for Cyber Security, 2011.

[27] Rieck, K., Trinius, P., Willems, C. ve Holz, T., "Automatic Analysis of Malware Behavior Using Machine Learning", Journal of Computer Security, Cilt 19, 639-668, 2011.

[28] Kong, D. ve Yan, G., "Discriminant Malware Distance Learning on Structural Information for Automated Malware Classification", Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, 347-348, 2013.

[29] Nari, S. and Ghorbani, A., "Automated Malware Classification Based on Network Behavior", Proceedings of International Conference on Computing, Networking and Communications (ICNC), 642-647, 2013.

[30] Santos, I., Devesa, J., Brezo, F., Nieves, J. ve Bringas, P.G., "OPEM: A Static-Dynamic Approach for Machine Learning Based Malware Detection", International Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions, Cilt 189, 271-280, 2013.

[31] Islam, R., Tian, R., Battenb, L. ve Versteeg, S., "Classification of Malware Based on Integrated Static and Dynamic Features", Journal of Network and Computer Application, Cilt 36, 646-556, 2013.

[32] Kim, H., Shin, K.G., Pillai, P., "MODELZ: Monitoring, Detection, and Analysis of Energy-Greedy Anomalies in Mobile Handsets", IEEE Transactions on Mobile Computing, Cilt 10, No 7, 968-981, 2011.

[33] Shabtai, A., Kanonov, U., Elovici, Y., et al. "Andromaly: a behavioral malware detection framework for android devices", Journal of Intelligent Information Systems, Cilt 38, No 1, 161-190, 2012.

[34] Lu, Y., Zulie, P., Jingju, L., et al. "Android malware detection technology based on improved Bayesian Classification", 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, 1338–1341, 2013.

[35] Yerima, S.Y., Sezer, S., Muttik, I., "Android Malware Detection Using Parallel Machine Learning Classifiers", 2014 Eighth International Conference on Next Generation Mobile Applications, Services and Technologies, 37 – 42, 2014.

[36] Shen, T., Zhongyang, Y., Xin, Z., "Detect Android Malware Variants using Component Based Topology Graph", 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, 406-413, 2014.

[37] Sheen, S., Anitha, R., Natarajan, V., "Android based malware detection using a multifeature collaborative decision fusion approach", Neurocomputing, Cilt 151, 905-912, 2015.

[38] Kabakuş, A. T., Doğru, İ. A., Çetin, A., "APK Auditor: Permission-based Android malware detection system", Digital Investigation, Cilt 13, 1-14, 2015.

[39] Zhou Y. ve Jiang, X., "Dissecting Android Malware: Characterization and Evolution". 2012 IEEE Symposium on Security and Privacy, 95–109, 2012.

[40] Aydoğan, E., Genetik Programlama Kullanılarak Mobil Zararlı Yazılımların Otomatik Olarak Üretilmesi, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, 2014.

[41] Arp, D., Spreitzenbarth, M., Gübner, M., Gascon, H. ve Rieck, K., "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket", Network and Distributed System Security (NDSS) Symposium 2014, *2014.*

[42] Lindorfer, M., Neugschwandtner, M. ve Weichselbaum, L., "ANDRUBIS - 1,000,000 Apps Later: A View on Current Android Malware Behaviors", 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, 2014.

[43] Mas`ud, M. Z., Sahib, S., Abdollah, M. F., Selamat, S. R., ve Yusof, R., "Android Malware Detection System Classification", Research Journal of Information Technology, Cilt 6, No 4, 325-341, 2014.

[44] Van der Meulen, R. ve Rivera, J., "Gartner says smartphone sales accounted for 55 percent of overall mobile phone sales in third quarter of 2013, Press Release, 2013.

[45] Kabakuş, A. T., Doğru, İ. A., Çetin, A. (2015). Android Kötücül Yazılım Tespit ve Koruma Sistemleri. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Cilt 31, No 1, 9-16, 2015.

[46] Torregrosa, B., A framework for detection of malicious software in Android handled systems using machine learning techniques, Universitat Autònoma de Barcelona, 2015