Haliç Üniversitesi Fen Bilimleri Dergisi 2021, 4/2: 77-109
DOI: 10.46373/hafebid.936695

Android İşletim Sistemlerinde Zaman Tabanlı İzin Yaklaşımı ile Saldırı Önleme Sistemi

Halil İbrahim BEDİR[1*], Muhammed Ali AYDIN[2], Abdül Halim ZAİM[1]

[1]İstanbul Ticaret Üniversitesi, Bilgisayar Mühendisliği, İstanbul, Türkiye
Orcid: 0000-0002-3690-4429, (https://orcid.org/0000-0002-0233-064X)
[2]İstanbul Üniversitesi -Cerrahpaşa, Bilgisayar Mühendisliği, İstanbul, Türkiye
Orcid: 0000-0002-1846-6090

Geliş Tarihi: 12.05.2021
***Sorumlu Yazar e mail:** bediribrahim@outlook.com **Kabul Tarihi:** 08.06.2021

Atıf/Citation: Bedir, H. İ., Aydın, M. A., Zaim, A. H., "Android İşletim Sistemlerinde Zaman Tabanlı İzin Yaklaşımı ile Saldırı Önleme Sistemi", Haliç Üniversitesi Fen Bilimleri Dergisi 2021, 4/2: 77-109.

Araştırma Makalesi/ ***Research Article***

Özet

Pazarın %72.2'sini oluşturan Android işletim sistemi, en çok tercih edilen ve kullanılan mobil işletim sistemidir. Kullanıcılar, birçok platformdan farklı uygulamaları bu işletim sistemine rahatlıkla kurabilir. Akıllı cihazlar son zamanlarda çok fazla özel kullanıcı bilgisini taşıdığından dolayı siber saldırıların daha sık hedefi haline gelmiştir. Andorid işlem sistemi, uygulamaların mobil cihazda ulaşabileceği alanları kısıtlamak, özel kullanıcı bilgisi olan konum, kamera, şifre ve rehber gibi kullanıcı gizliliğini korumak için birçok izin çeşidi kullanır. Bu çalışmada, mevcut izin kontrol sistemindeki eksiklikleri göstermek için kötü amaçlı bir uygulama üretilmiştir ve bu uygulama üzerinden birkaç siber saldırı gerçekleştirilmiştir. Bu sorunların sebebinin, kullanıcıların istenen izinlerin niyetini tam olarak anlamadığından ve mevcut izin sisteminin kullanıcıya başka izin onaylama seçenekleri önerememesinden dolayı olduğu görülmüştür. Daha sonra, uygulama çalıştırırken kullanıcının ilgili fonksiyonu ne kadar zaman kullandığını belirleyerek daha kolay izin kontrolü sağlayabilen, müdahaleci olmayan, iznin süresini belirleyen zaman tabanlı bir izin kontrol sistemi önerilir.

Anahtar Kelimeler: Android,Android uygulamaları,Android güvenlik,APK ,İzin sistemi,API seviyesi

Attack Prevention System with Time Based Permission Approach in Android Operating Systems

Abstract

Android operating system, which constitutes 72.2% of the market, is the most preferred and used mobile operating system. Users can easily install different applications from many platforms onto this operating system. Since smart devices carry a lot of private user information, they have become the target of cyberattacks. The Andorid transaction system uses many types of permissions to restrict the areas that applications can access on the mobile device and to protect user privacy such as location, camera, password and directory with private user information. In this study, a malicious application was created to show the shortcomings in the current permission control system, and several cyber attacks were carried out on this application. It seems that the cause of these problems is because users do not fully understand the intent of the requested permissions and the current permission system is unable to suggest other permission options to the user. Then, a non-intrusive, time-based permission control system that determines the duration of the permission is recommended, It can provide easier permission control by determining how long the user has used the relevant function while running the application.

Keywords: Android, Android applications, Android security, APK, Permission system, API level

1.Giriş

Günümüzde insanlar eğlence ve iş için giderek daha fazla cep telefonu kullanmakta, bu nedenle cep telefonları, iletişim, şifre ve e-posta gibi daha fazla özel kullanıcı bilgisi taşımaktadır. Android İşletim Sistemi, pazar yerinin%72,2'sine sahip olması sebebiyle mobil cihazlar için en popüler platformdur. Bu pazarda bu ihtiyacı karşılamak için bireysel geliştiriciler ve şirketler tarafından üçüncü taraf uygulamaları geliştirmede ciddi bir yükseliş bulunmaktadır. Android İşletim Sistemi, açık kaynak ve birçok platformda uygulama pazarının öncüsüdür. Sonuç olarak, bağımsız geliştiricilerin kendi uygulamalarını geliştirmelerine ve dağıtmalarına izin verir. Ayrıca, Android platformu, uygulamaların sistem kaynaklarına ve cihaz donanımına erişmesini sağlayan büyük

ölçekli bir uygulama karma arabirimi (API) ile üçüncü taraf geliştirme sağlar.

Android uygulamaları, kullanılacak uygulamalar için gerekli tüm dosyaları, kitaplıkları ve meta verileri içeren sıkıştırılmış Android paketleri (APK) olarak yüklenir. Bu APK'lar, pazar yerinden Paket Yöneticisi sistem hizmetini çağırarak yüklenebilir [1]. Örneğin, Google Play uygulaması aracılığıyla Google market mağazasından bir uygulama yüklenebilir. Paket Yöneticisini çağırmanın başka bir yöntemi de uygulamanın APK'sını cihaza kopyalamak ve Android işletim sisteminin onu başlatmasını istemektir.

Kullanıcının gizliliğini korumak için Android platformu, uygulamayı hassas bilgilere erişmek için izinler istemeye zorlar [2, 3]. Bu nedenle, uygulama yükleme işlemine devam etmek için Android işletim sistemi, kullanıcıdan uygulamanın gerekli izinlerini açıkça vermesini ister. Kullanıcı izinleri vermeyi reddederse, yükleme işlemi iptal edilecektir. Ancak çoğu kullanıcı, listelenen izinleri tam olarak anlamadan izin istemlerini kabul eder. Sonuç olarak, bazı uygulamalar yalnızca kullanıcı verilerini toplamak için ekstra izinler gerektirebilir, bu da kullanıcı gizliliğini ihlal edebilir ve kötü niyetli eylemleri artırabilir [4]. Bu nedenle, Android güvenliği büyük ölçüde izin sistemi mekanizmasının verimliliğine bağlıdır.

Bu makalede aşağıdaki konular işlenmiştir;

- Mevcut mobil sistemdeki izin kontrol mekanizması yeterince güvenli mi?
- İzinlerin doğru bir şekilde verilip verilmediği?
- Kullanıcının izin seçiminde zaman tercihi, önerilen izin sisteminin geliştirilmesinde ne ölçüde yardımcı olur?
- Kullanıcının bilgi güvenliğini müdahaleci olmayan ve kullanıcı dostu bir şekilde nasıl temin ederiz?

Bu amaçla, mevcut izin sisteminin yetersizliklerini göstermek için bazı kavram kanıtı saldırıları gerçekleştirilir. Hâlâ sistem tarafından tespit edilemeyen birçok iznin kötüye kullanımı durumu vardır. Cihazın farklı fonksiyonlarına yapılan sızma işlemlerinde eylem

modellerindeki ve cihazın performansındaki farklılıklar izin sistemini daha da iyileştirmek için fizibilite sağlar. Ardından, mevcut sistem için önerilen zaman tabanlı izin sistemi akışının mevcut sistemden ne kadar verimli olduğu gösterilmiştir.

Android izin sistemi, sistem kaynaklarını ve kullanıcıların gizliliğini korumak için bir erişim kontrol mekanizması sunan temel güvenlik bileşenidir. Fakat onaylanan izinler sayesinde uygulamalar, kullanıcıların en hassas verilerine istedikleri zaman ulaşabilirler. Kullanıcılar verilen izinlerin uygulamalar tarafından ne zaman ve nasıl kullanıldığını bilememektedir. Mevcut izin sisteminin yetersizliği bu saldırıları kısıtlayamaz. Bu makaledeki amaç, kullanıcıların mahrem verilerinin korunması için yeni bir zaman tabanlı izin yaklaşımı ile güvenliğin arttırılması ve saldırıların kısıtlanmasıdır.

Önerilen bu sistem, mevcut izin sisteminin işleyişini bozmadan sadece zaman tabanlı bir akış ve kontrol mekanizması ile ilişkilendirilmiştir. Olası ataklardan içeri sızma ve kullanıcıların hassas verilerini kullanma ile meydana gelebilecek zararı minimuma indirmek için tasarlanmıştır.

1.1 Literatür İncelemesi

Android izin sisteminin 2008'de tanıtıldığından bugüne kadar, bileşenlerinin resmi bir modeli eşliğinde kapsamlı bir analizini sunmaktadır. Analizin sonuçları, orijinal sürümden bu yana izin sayısında sürekli bir artış olduğunu ortaya koyuyor ve bazı izin kategorilerinde yedi kat artış olduğu görülmektedir. İzin sisteminin gelişimini ve buna bağlı güvenlik sorunlarını uygulamaların bakış açısından incelemek için en iyi Android uygulamalarının son beş yıllık sürümleri için bir örnek olay yürütülmüştür. Bazı uygulamalar 2020 sürümüne kadar izin kullanımında%73,33 artış gösterdi. Ek olarak, vaka çalışmasının sonuçları hem satıcılar hem de geliştiriciler tarafından izin dağıtımının anlaşılmasına katkıda bulunur. Son olarak, izne dayalı güvenlik geliştirmeleriyle ilgili bir sonuç olarak Android izin sisteminin çeşitli güvenlik sorunları ile karşı karşıya olduğunu ortaya koymaktadır [5].

Eylem Kaydedici, kullanıcının eylem dizilerini kaydeder. Ardından, Grafik Oluşturucu, eylem dizilerinden kullanıcı eylem modelini oluşturur. Grafik Eşleştirici, kullanıcı eylem modeli ve paylaşılan kaynakların kesin referans etkileşim grafikleriyle eşleşir ve mevcut kaynaklara erişimin verilip verilmeyeceğine karar verir [6].

Esneklik ve izolasyon sağlayan genişletilebilir izin eklentileri için bir çerçeve olan Dalf'ı öneriyoruz. Dalf'ın altında yatan fikir, izin eklentilerinin kendilerinin uygulama olarak ele alınması gerektiğidir. Bu yaklaşım, eklentilerin durumu korumasına ve Android'in işlem izolasyon mekanizmaları tarafından kısıtlanırken cihazın konumu gibi sistem kaynaklarına erişmesine olanak tanır [7].

Uygulamalar, mobil işletim sistemleri için çok önemli bileşenlerdir. Uygulama ortamı ve uygulamaların çeşitliliği bir tercih sebebi haline gelmiştir. Kullanıcılar yalnızca iOS cihazları için App Store'dan uygulama indirebilirken, Android'in güvenilir uygulamaları indirme ve yükleme platformu Google Play Store'dur. Ayrıca, kullanıcılar Android uygulamalarını diğer kaynaklardan yükleyebilir. Kullanıcılar, belirli bir uygulama için verilen izne bağlı olarak mobil cihazlarına bir APK dosyası yükleyebilir. İşte bu noktada güvenlik tehditleri ortaya çıkıyor.Bu çalışmada, bu tehditlerin bir incelemesi sunulmaktadır. Bu tehditler, klonlanmış kötü amaçlı Android uygulamaları tarafından başlatılır. Klonlanan uygulamalar çoğunlukla üçüncü taraf Android uygulama mağazaları tarafından barındırılmaktadır. Bu tehditler üçüncü taraf uygulama mağazalarından başlatıldığı için, özellikle Çin, Rusya ve Avrupa'da ikamet edenler için çok ciddi tehditler bulunmaktadır [8].

Android uygulamalarının artan kullanımı, yeni kullanıcıları gizli verilere yetkisiz erişim açısından sürekli tehdit altında tutmuştur. Antivirüs yazılımı bile bir Android cihazının dosya sistemine veya platform sınırlamaları nedeniyle yüklü uygulamaların dinamik davranışına erişemez veya izleyemez. Cihaz güvenliği açısından ciddi sonuçları vardır. Kullanıcı tarafından algılanmadan kötü amaçlı bir uygulama dosyaları indirebilir ve başka kaynaklara iletebilir. Bu çalışmada, bir

çalışma Zamanı algılama ve önleme sistemi öneriyor ve uyguluyoruz. Önerilen sistem, kullanıcının önceden bilgisi olmadan cihazın kaynaklarına erişmeye çalışan uygulamaları kilitleyerek ikinci bir güvenlik seviyesi sağlayacaktır. Yeni sistemin, hassas bilgileri toplamak için uygulamaların tüm saldırılarının üstesinden gelebileceğini, meşru uygulamaların kullanımı ve işletim sisteminin performansı üzerinde küçük etkiler yaratabileceğini göstermektedir [9].

Apex, bir kullanıcıya telefon kaynaklarının farklı uygulamalar tarafından kullanımını kısıtlamak için çeşitli seçenekler sunar. Kullanıcı bazı izinleri verebilir ve diğerlerini reddedebilir. Bu, kullanıcının, uygulama tarafından sağlanan işlevselliğin bir kısmını kullanmasına izin verirken, yine de kritik veya maliyetli kaynaklara erişimi kısıtlar. Apex ayrıca, kullanıcının kaynakların kullanımına çalışma zamanı kısıtlamaları getirmesine izin verir. Son olarak, kullanıcı, bir uygulamanın kullanımına bağlı olarak kaynakların kullanımını, örneğin her gün gönderilen SMS mesajlarının sayısını sınırlamak isteyebilir. Apex'in anlamını ve bu kısıtlamaları tanımlamak için kullanılan politika modelini tanımlıyoruz [10].

FBI Müdür Yardımcısı Steven Chabinsky şunu söyledi: "Siber suçlar, yüksek düzeyde organize olmuş suç örgütlerinin mesleği haline geldikçe, kolluk kuvvetleri, tehdidin artan karmaşıklığını ele almak için yaklaşımlarını yeniliyorlar. Harekete geçmezsek, siber tehdit varoluşsal bir tehdit olabilir, yani ülkemizin varlığına meydan okuyabilir veya ulusumuzun potansiyelini önemli ölçüde değiştirebilir. Yeterli zaman, motivasyon ve finansman verildiğinde, kararlı bir düşmanın hedeflenen bir sisteme her zaman girebileceğine ikna oldum.[11]

Siber güvenlik, sistemleri ve interneti düzenlerken, yönetirken ve kullanırken bireylerin alabilecekleri özene ve yaptıkları sonuçlara dayanır. Siber güvenlik değerlendirme sorununa çözüm bulmak için çok sayıda çaba gösterilmiş ve çeşitli çerçeveler oluşturulmuştur. Kısıtlamalar, yeni teknolojiler ve tesis sınırlamaları gibi farklı yönlerden kaynaklanmaktadır. Güvenlik sorunları, genellikle güvenlik gereksinimleri ve diğer faydalar arasında bir değiş tokuş olarak kabul edilir. [12]

İzin sistemlerine yeni bir yaklaşımı savunuyoruz: her izin için bağımsız olarak en uygun izin verme mekanizmasını seçilmesidir. Yaklaşımımız, her iznin benzersiz gereksinimlerini ve kısıtlamalarını hesaba katar. Bu, tüm izinlere tek bir izin verme mekanizması uygulayan mevcut platformlardan bir ayrımdır. İzin verme mekanizmaları arasından seçim yapmak için bir dizi yönerge sağlıyoruz.

- Platform tasarımcılarının kullanabileceği izin verme mekanizmalarını sıralıyor ve güçlü ve zayıf yönlerini tartışıyoruz. Geçmiş literatürden bir dizi kullanılabilirlik ilkesini dikkate alıyor ve izin verme mekanizmalarını kullanıcı deneyimlerinin kalitesine göre sıralıyoruz.
- Her izin için en uygun izin verme mekanizmasını seçmek için bir ön karar prosedürü geliştiriyoruz.
- Karar prosedürümüzü bir dizi akıllı telefon uygulama iznine uyguluyoruz. En iyi kullanıcı deneyimlerini sağlayan izin verme mekanizmalarının izinlerin çoğuna uygulanabileceğini gördük.[13]

Makale, gelen her SMS'e yanıt gönderen bir otomatik yanıt uygulaması olan "SMS otomatik yanıtlama" gibi SMS ile ilgili çeşitli Android uygulamalarını incelemektedir. Bu otomatik yanıt özelliği uygulamada uygulanmıştır. SMS denetleyicisi, gelen her SMS'e yanıt gönderen bir otomatik yanıt özelliği olan SMS ile ilgili bir Android Hizmetidir. Bu otomatik yanıt özelliği bu uygulamada uygulanmıştır. Veri tabanımızda 3 iletişim numarası kaydedeceğiz, yeni SIM takıldığında bu oluşturulan mesaj tetiklenecek ve kaydedilen metin 3 iletişim numarasına gönderilecektir.[14]

2. Materyal ve Metot

2.1 Mevcut android izin sistemi

Android'in güvenlik mimarisinin temel bir tasarımı, hiçbir uygulamanın kullanıcıyı, diğer uygulamaları veya işletim sistemini olumsuz etkileyebilecek herhangi bir işlemi gerçekleştirememesidir. Bu nedenle, uygulamanın bileşenlere, hassas verilere ve belirli sistem özelliklerine

erişim istekleri Android izin sistemi tarafından düzenlenir. Tablo1'de Android izin sisteminin gelişimini göstermektedir. Her sürüm için Tablo1 de Android sürüm kod adını, sürüm aralıklarını, API düzeyi aralıklarını, koruma düzeyi başına izin sayısını ve her sürüme dahil edilen toplam izin sayısını gösterir. Her yeni sürümde kullanıcılardan daha fazla izin istenmektedir.

Tablo 1. Android Platformu Resmi Bültenleri

Android Sürüm İsmi	Versiyon	API Seviyesi	Tehlikeli İzin	Normal	İmza	İmza veya Sistem	Toplam İzinler
Android 11	11	30	30	47	48	42	167
Q	10	29	30	45	42	41	158
Pie	9	28	27	42	38	41	148
Oreo	8.0-8.1	26-27	26	39	38	41	144
Nougat	7.0-7.1	24-25	24	35	35	41	135
Marshmallow	6	23	24	35	35	40	131
Lollipop	5.0-5.1	21-22	24	32	24	40	120
KitKat Watch	4.4W	20	24	32	18	40	113
KitKat	4.4	19	23	32	18	40	112
Jelly Bean	4.1-4.3.1	16-18	23	29	16	36	104
Ice Cream Sandwich	4.0.1-4.0.4	14-15	20	29	14	35	98
Honeycomb	3.0.x-3.2	11-13	19	29	12	35	95
Gingerbread	2.3-2.3.5	9-10	19	29	11	35	94
Froyo	2.2.x	8	18	27	11	35	87
Eclair	2.0-2.1	5-7	18	26	9	33	86
Donut	1.6	4	18	26	9	32	85
Cupcake	1.5	3	17	25	8	31	81
Base	1-1.1	1-2	17	24	7	27	73

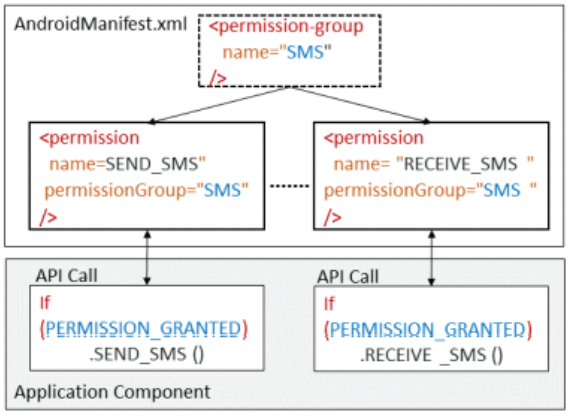
İzinler, karakter dizileri olarak benzersiz bir şekilde tanımlanır (ör. Android.permission.READ_SMS). Bir izin bir nesneye bağlıysa, nesneye ancak uygulamaya izin verildikten sonra başka bir uygulama tarafından erişilebilir.

2.1.1 Manifest dosyası

Her uygulama bir Manifest dosyası içerir. Android Manifest.xml dosyası, bir Android uygulamasının kök klasöründe bulunur. Uygulama tarafından kullanılan izinleri ve uygulama hakkındaki meta verileri içerir. Sisteme veya diğer uygulama kaynaklarına veya verilerine erişmek için verilmesi beklenen uygulama izinleri, AndroidManifest.xml dosyasında "<uses-permission> "etiketi ile etiketlenir. Ayrıca, Android Manifest.xml dosyası, uygulama tarafından kendi bileşenlerini korumak için yeniden alınan izinleri içerir. Bu izinler "<permission> "etiketiyle bildirilir. Ayrıca, Manifest dosyası uygulamanın bileşenlerini içerir. Uygulamanın her bileşeni, uygulamanın kullandığı izinler dâhil olmak üzere temel özelliklerini bildirmelidir.

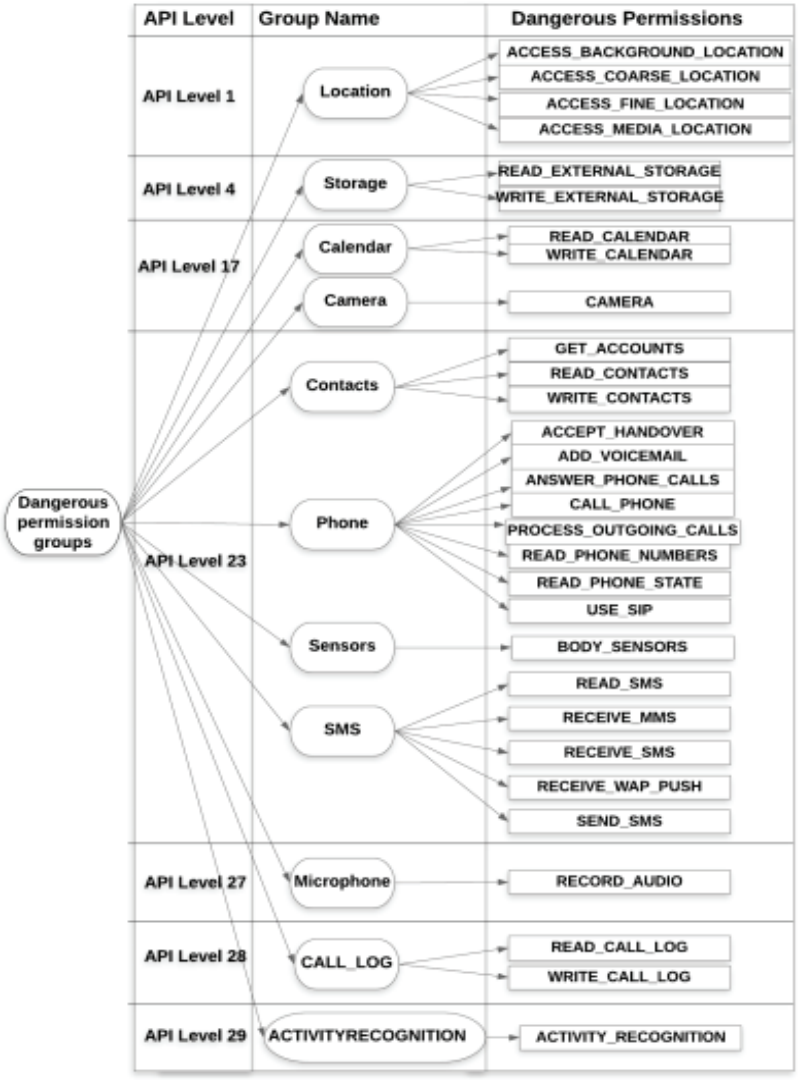
2.1.2 İzin grubu

İzin grubu, uygulamanın izinlerinin mantıksal bir sınıflandırmasıdır. Manifest dosyasında "<permission-group>" etiketi olarak tanımlanır. Ardından," <permission>" etiketinin içinde bir "android: permGroup "öğesi bildirilerek bu gruba izinler eklenebilir. (Şekil 1)'de Manifest dosyasında bir izin grubu oluşturmanın bir örneğini gösterilmektedir. SMS grubuna iki izin, SEND_SMS ve RECEIVE_SMS eklendi.



Şekil 1. Android Gruplama İzinlerinin Bir Örneği

Android izin sistemi, aşağıda da açıklandığı gibi tüm tehlikeli izinleri izin gruplarına ayırır. Buna göre, tehlikeli türündeki tüm izinler için, bu iznin ait olduğu yerde bir izin grubu vardır. (Şekil 2)'de Android'in 11 tehlikeli izin grubu gösterilmektedir.



Şekil 2. Android Tehlikeli İzin Grupları

2.1.3 İznin uygulanma süreci

Uygulama, kısıtlanmış verilere veya kısıtlanmış eylemlere erişim gerektirebilecek işlevler sunuyorsa, izinleri onaylamaya gerek kalmadan eylemleri gerçekleştirip gerçekleştiremeyeceğini kullanıcı belirlemektedir. Uygulamanın bir fonksiyonunu kullanmak için kısıtlanmış verilere erişmesi veya sınırlı eylemler gerçekleştirmesi gerektiğine kullanıcı karar verirse, uygun izinleri onaylaması gerekmektedir. Yükleme zamanı izinleri olarak bilinen bazı izinler, uygulama yüklendiğinde otomatik olarak verilir. Çalışma zamanı izinleri olarak bilinen diğer izinler, uygulamanın bir adım daha ileri gitmesi ve çalışma zamanında izin istenmesi gerekmektedir.[15]

Uygulama Android 6.0 veya daha yüksek bir sürümde çalışıyorsa, sistem izin verme işlemini aşağıdaki şekilde işler:

- Normal izinler, yükleme sırasında otomatik olarak verilir.
- İmza izinleri için, her iki uygulama da kullanıcı onayı olmadan aynı sertifika ile imzalanırsa sistem hemen izinleri verir. Aksi takdirde, imza izinleri yükleme sırasında verilir.
- Sistem tehlikeli izin verebilmek için öncelikle talep edilen iznin izin grubunu kontrol eder. İzin grubu bir izin vermediyse, çalışma zamanında kullanıcıya bir iletişim kutusu gösterilerek izin istenir. Grup izin onayı verilmişse, sistem tarafından sadece istenen izne erişim verilir.
- İzin denetleyicisi, çalışma zamanı izni ile ilgili işlemeyi kontrol eder. Android 9 ve önceki sürümlerde, izin denetleyicisinin işlevleri paket yükleyicisine yerleştirilmiştir. Ancak Android 10'da bu modül paket yükleyiciden bağımsız bir varlık olarak çalışmaktadır.
- İzin denetleyicisi, çalışma zamanı izinlerinin, izin gruplamasının ve izin kullanımı izleme ve rollerinin verilmesini denetler.

2.2 Kötü amaçlı bir android uygulama ile mevcut izin sisteminin testi ve saldırı tespiti

Bu çalışma Kali Linux ve Android Stüdyo kullanılarak yapılmıştır [16,17]. Mevcut Android izin sisteminin eksikliklerinin gösterilmesi için Kali Linux üzerinde oluşturulan zararlı bir Android uygulaması Andorid Stüdyodaki emülatöre yüklenilmiştir. Bu yükleme sonrası uygulamaya Kali Linux üzerinden saldırı komutları verildi. Her saldırı sonrası uygulamanın cihaz üzerindeki performansını ölçülmüştür.

Kali Linux üzerinden bir Android uygulaması oluşturulması için sırayla Tablo2 de belirtilen komutlar girilmiştir.

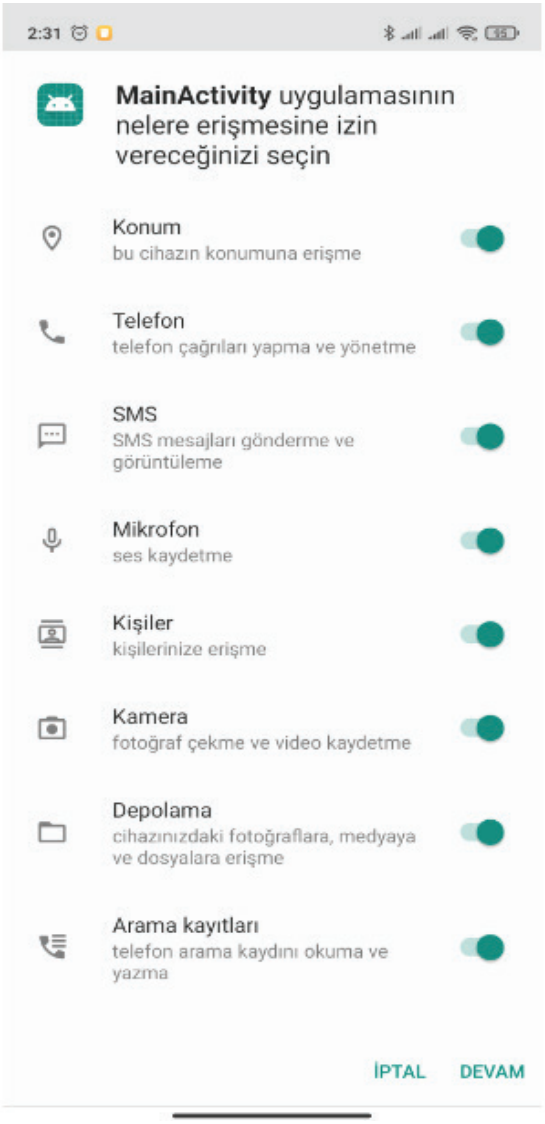
Tablo 2. Kali Linux Komutları

Kali Linux Komutları	**Açıklamalar**
msfvenom –"p android/meterpreter/reverse_tcp LHOST=192.168.1.106 LPORT=1920 R > Desktop/MainActivite.apk"	Bu komutla bir APK dosyası oluşturulur. Hangi ip ve port üzerinden sızma yapılacağı belirtilir.
"msfconsole "	Komutları girmek için console hazırlanır.
"use exploit/multi/handler"	Exploitler kullanılır.
"set payload android/meterpreter/reverse_tcp"	Payload dosyası ayarlanır.
"set LHOST 192.168.1.106"	Uygulamanın ip adresi ayarlanır.
"set LPORT 1920"	Uygulamanın portu ayarlanır.
"exploit "	Bu komutla sızma işlemi başlatılır.

İlgili komutlar girildikten sonra bir kötü amaçlı Android uygulaması oluşturuldu. Bu uygulama kullanıcı tarafından cihaza yüklendi. (Şekil 3)'te görüldüğü gibi exploit komutu verildikten sonra uygulama üzerine tanımlanan ip adresi (192.168.1.106) ve portu (1920) dinlemeye alınmıştır.

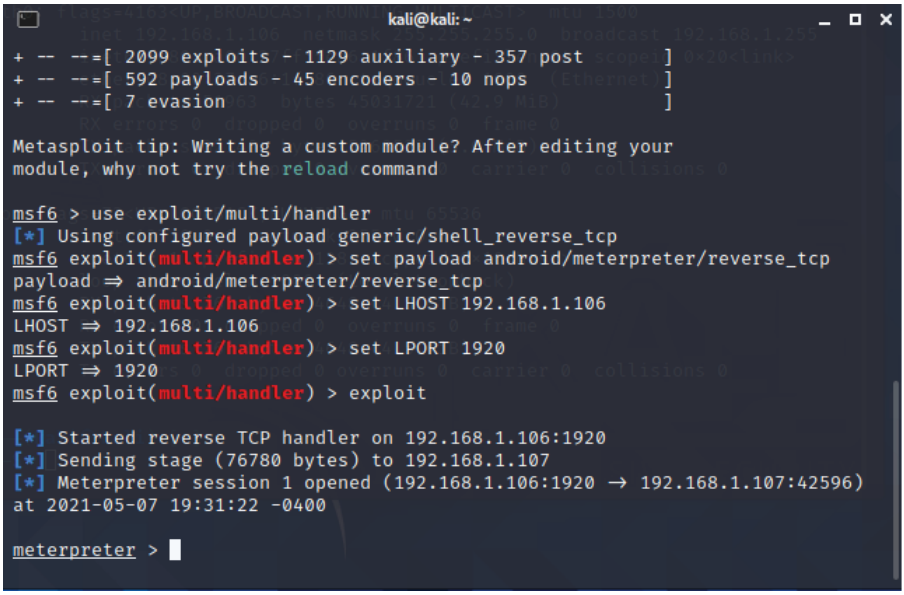
Şekil 3. Sızma Öncesi

Kullanıcı uygulamayı yüklerken (Şekil 4)'teki izinlere onay verdi. Kullanıcı ayarlardan izinleri kaldırana kadar uygulama verilen izinler kapsamında cihazın hassas verilerine ulaşabilecektir.



Şekil 4. Uygulamanın Talep Ettiği İzinler

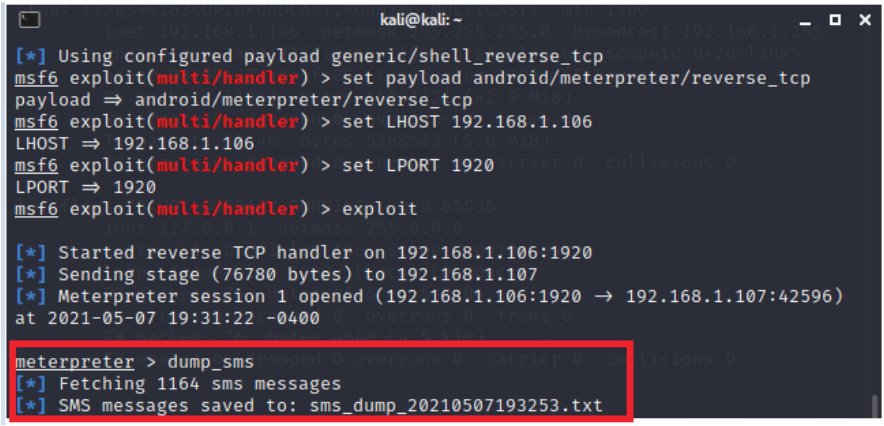
Kullanıcı uygulamayı çalıştırdığında (Şekil 5)'te görüldüğü gibi dinlemeye alınan ip adresi (192.168.1.106) ve portu (1920) aktif olmuştur. Bundan sonra uygulama üzerinden kullanıcının özel verilerine ulaşılabilir.



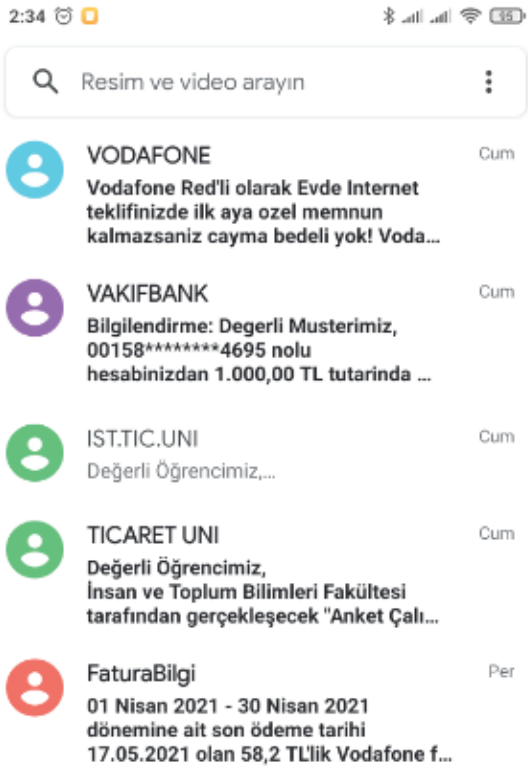
Şekil 5. Sızma Sonrası

2.2.1 Kullanıcının SMS'lerini elde etmek

Kullanıcı uygulamayı kapatsa bile SMS'leri görüntüleme (READ_ SMS) iznine onay verdiği için uygulama üzerinden kullanıcının SMS bilgilerine ulaşılabilir. Kali Linux üzerinden daha önceden dinlemeye alınan uygulamaya sızma komutları verildi. (Şekil 6)'da SMS'leri görüntüleme komutu verilmiştir ve bir .txt dosyası halinde tüm SMS'ler çekilmiştir. (Şekil 7)'de telefon üzerindeki kullanıcın SMS'leri gösterilmiştir. (Şekil 8)'de .txt dosyası üzerindeki elde edilen SMS'ler gösterilmiştir.



Şekil 6. SMS Görüntüleme Saldırısı



Şekil 7. Kullanıcının SMS'leri

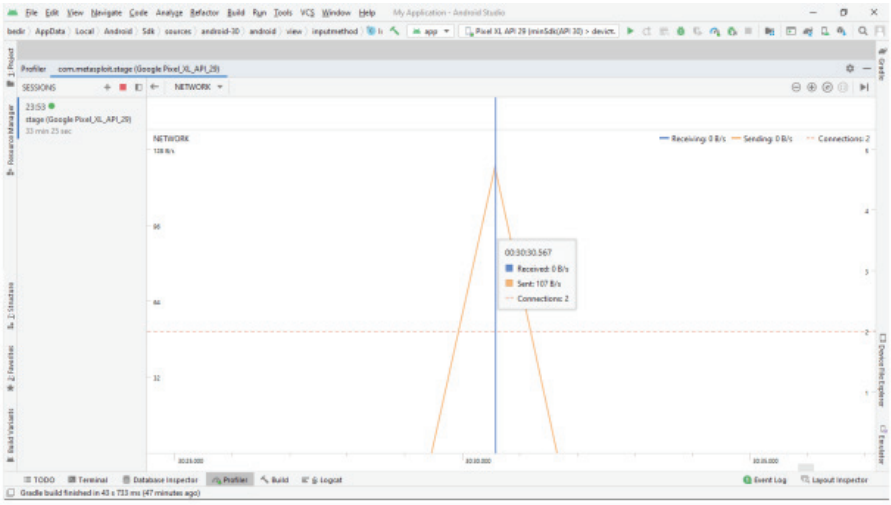
Şekil 8. Elde Edilen SMS'ler

2.2.2 SMS saldırısı sonrası uygulamanın performansı

Bu fonksiyonu kullanarak yapılan sızma işleminde internet (Şekil 9)'daki gibi olmaktadır. Uygulamanın bu fonksiyonunun kullanılmasından sonra;

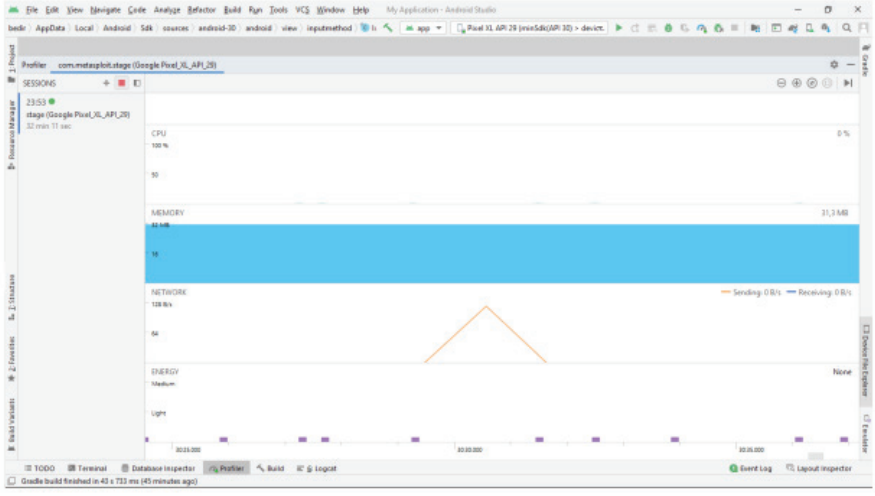
Received (Alınan): Dışarıdan verilen sızma komutuyla uygulamanın cihaza aldığı verinin 0 KB/s olduğu görülmektedir.

Sent (Gönderilen): Dışarıya iletilen veri 107 KB/s olduğu görülmektedir. Cihazdan dışarı gönderilen paket boyutudur.



Şekil 9. Sızma Anında Uygulamanın İnternet Performansı

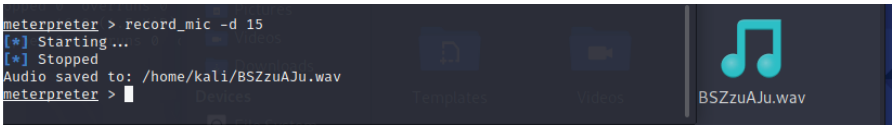
Bu fonksiyonu kullanarak yapılan sızma işleminde performans geneli (Şekil 10) gösterildiği gibi olmaktadır.



Şekil 10. Uygulamanın Genel Performansı

2.2.3 Uygulama üzerinden cihazın ses kaydı fonksiyonuna sızma

Kali Linux üzerinden (Şekil 11)'de belirtilen sızma komutu verilerek uygulamanın ses kaydı alma özelliği (RECORD_AUDIO) ile cihazın ortam sesi 15 saniye dinlenmiştir. .wav uzantısı ile ses dosyası kaydedilir.



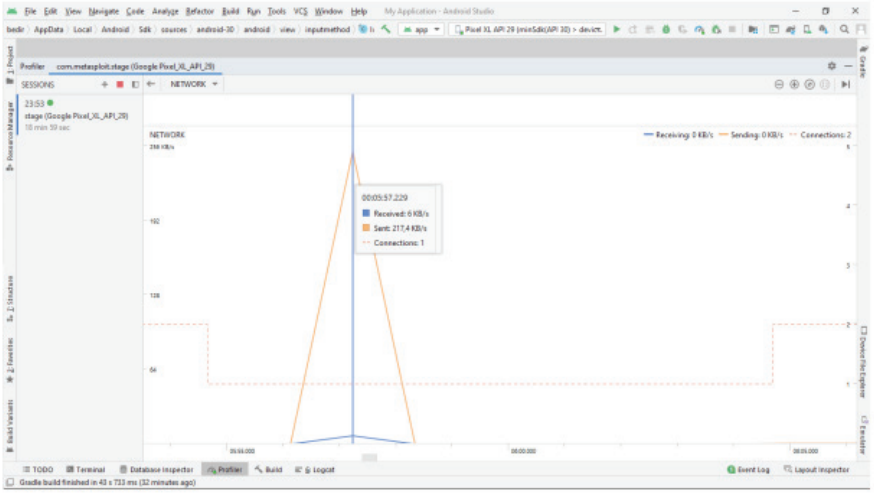
Şekil 11. Ses Kaydı Sızma Komutu

2.2.4 Ses kaydı saldırısı sonrası uygulamanın performansı

Bu fonksiyonu kullanarak yapılan sızma işleminde internet (Şekil 12) gösterildiği gibi olmaktadır. Uygulamanın bu fonksiyonunun kullanılmasından sonra;

Received (Alınan): Dışarıdan verilen sızma komutuyla uygulamanın aldığı verinin 6 KB/s olduğu görülmektedir.

Sent (Gönderilen): Dışarıya iletilen verinin 217,4 KB/s olduğu görülmektedir. Cihazdan dışarı gönderilen paket boyutudur.



Şekil 12. Ses Kaydı Sızması Anında Uygulamanın İnternet Performansı

2.3 Önerilen izin sistemi

Mevcut izin sisteminde kullanıcıya iki çeşit izin sunuluyor. Bu seçenekler izin ver seçeneği ve reddet seçeneğidir. Eğer kullanıcı uygulamaya izin verirse, uygulama izin verilen fonksiyonu kullanıcı ayarlardan düzenleyene kadar kullanabilir. Bu yüzden kötü amaçlı uygulamalara karşı güvenlik sağlanamaz. Kullanıcı reddet yaparsa uygulamayı istediği gibi rahat kullanamaz. Çünkü uygulama kullanımı süresinde sürekli ilgili fonksiyon kullanımı için kullanıcının önüne iletişim kutusu çıkaracaktır ve izin vermeye zorlayacaktır. Kullanıcı uygulamadan istediği verimi alamayacaktır. Bu hem kullanıcı dostu hem de güvenli olmayan bir durumdur.

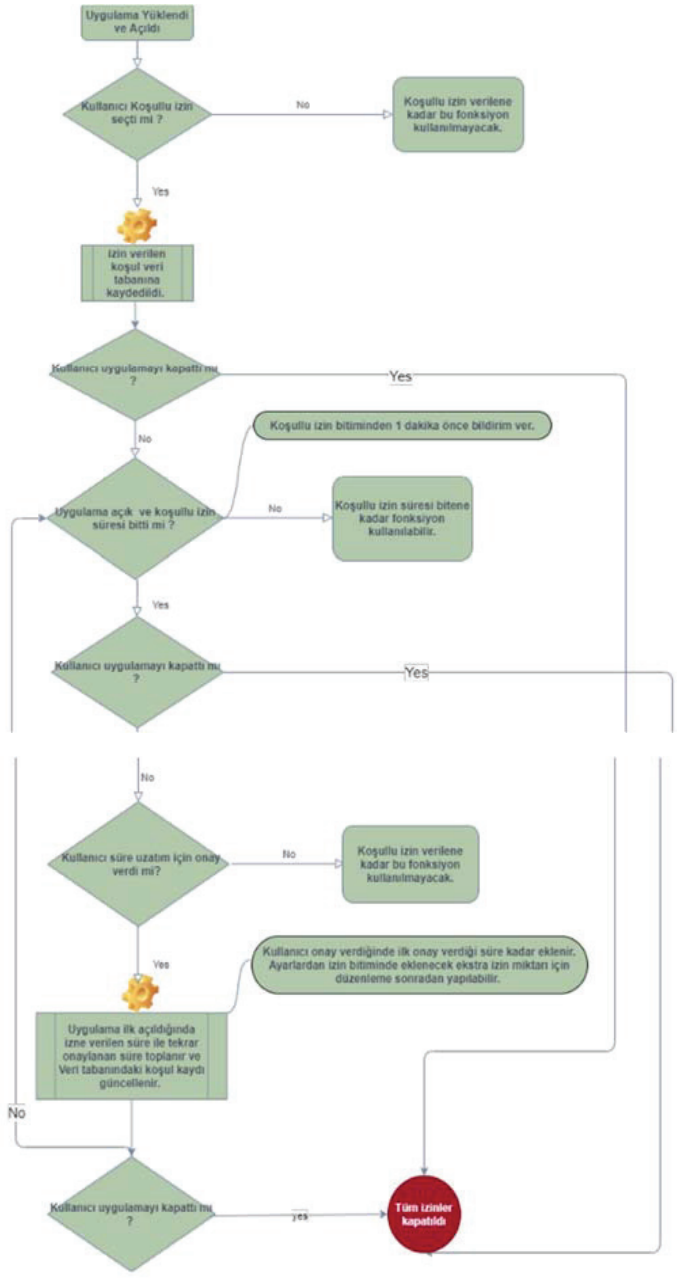
Yapılan araştırmalar sonucunda önerilen diğer yaklaşımlardan elde edilen bilgiler incelendiğinde zaman tabanlı izin sistemi, diğerler önerilerden farklı olarak eşsiz bir algoritma ve farklı akış kurgusu sayesinde kullanıcı güvenliğini ve deneyimini belirgin bir şekilde ön plana çıkardığı görülmektedir. Özellikle zaman kurgusu ve buna bağlı olarak oluşturulan akış, diğer yaklaşımlara göre müdahaleci olmayan ve cihazın performansını etkilemeyen bir sistem olarak sonuç vermektedir. Bu sistemde kullanıcıya uygulama izinlerini verirken başka bir seçenek sunulmuştur. Kullanıcı, uygulamanın izin listesinden bir fonksiyona izin verirken zaman seçeneklerinden birini seçer. Bu şekilde kullanıcı, uygulamanın o fonksiyonu ne kadar kullanacağını sınırlamış olur. Bu şekilde kullanıcı uygulamayı hem rahat hem de güvenli bir şekilde kullanabilecektir.

Uygulama cihaza yeni yüklenirken (Şekil 13)'de gösterilen akıştaki yolu izleyecektir. Bu akış da;

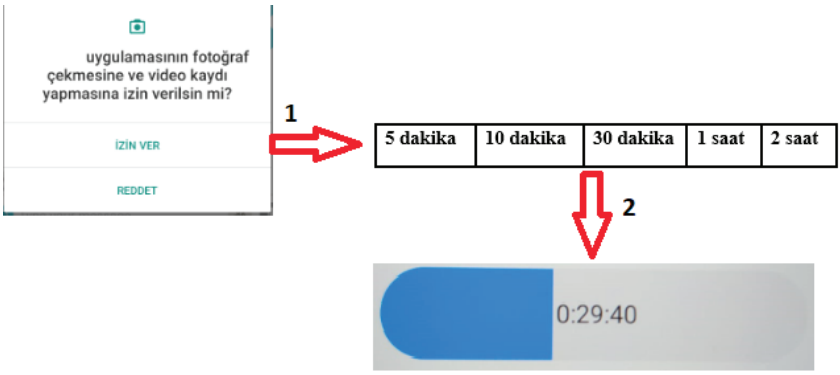
- Normal izinler, yükleme sırasında otomatik olarak verilir.
- İmza izinleri için, her iki uygulama da kullanıcı onayı olmadan aynı sertifika ile imzalanırsa sistem hemen izinleri verir. Aksi takdirde, İmza izinleri yükleme sırasında verilir.
- Sistemin tehlikeli izin verebilmesi için öncelikle talep edilen iznin izin grubunu kontrol eder. Uygulama daha önce bu gruba

tehlikeli bir izin verdiyse, kullanıcıyla herhangi bir etkileşim olmaksızın sistemin görev başına hemen verilir. Öte yandan, izin grubu bir izin vermediyse, çalışma zamanında kullanıcıya bir iletişim kutusu gösterilerek izin istenir.

- Kullanıcı, açılan iletişim kutusunda izin ver seçtiğinde (Şekil 14)'de gösterilen sırayla zaman paneli açılır ve sonra sürenin başladığını gösteren zaman göstergesi açılır. Bu gösterge sadece bilgilendirmek içindir ve ekrandan 3 saniye sonra kaybolacaktır.
- Seçilen izin veri tabanına kaydedilir.
- Zaman panelinden seçilen süre bitiminden 1 dakika önce kullanıcıya hatırlatma bildirimi verilir.
- Kullanıcı onay verirse ilk verdiği süre kadar tekrar izin verilir. Verilen bu süre ilk verilen süre ile toplanır ve veri tabanı güncellenir. Bunun amacı kullanıcının ilgili fonksiyonu uygulama açıkken ne kadar kullandığını hesaplayıp kullanıcı deneyimini belirlemektir.
- Kullanıcı bildirime zaman uzatımı için onay vermezse veya uygulamayı kapatırsa tüm izinler iptal edilir.



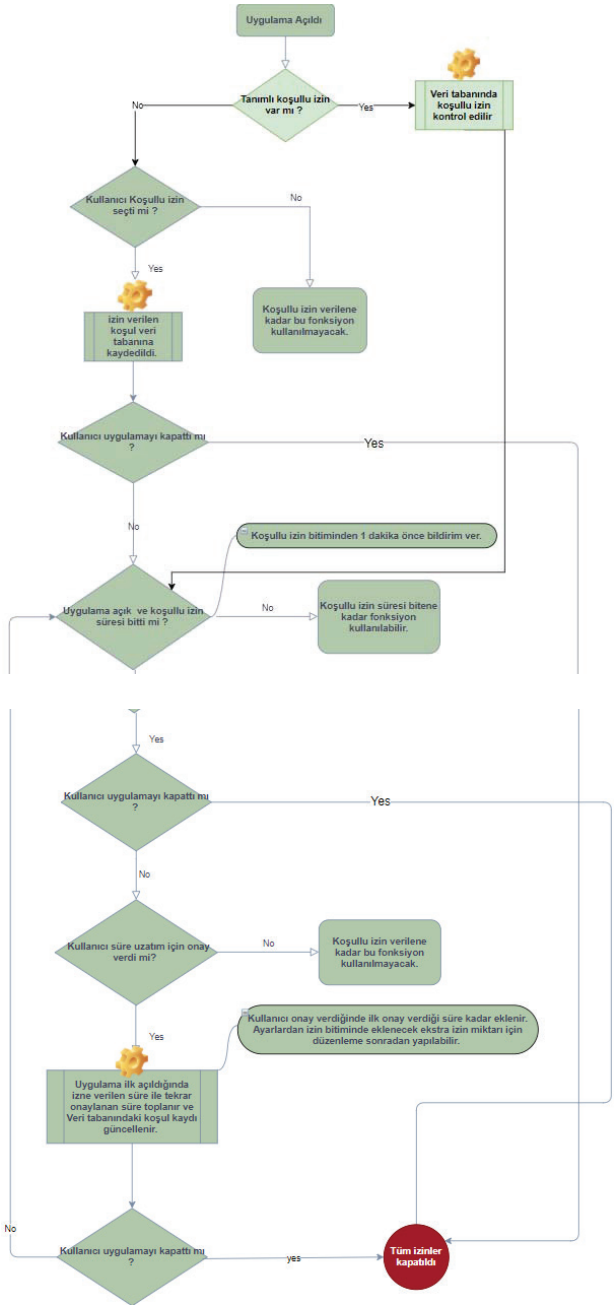
Şekil 13. Uygulamanın Yeni Yükleme Aşamasındaki İzin Akışı



Şekil 14. Zaman Paneli

Uygulama yeniden açıldığında kullanılacak olan yeni akış (Şekil 15)'te gösterilmiştir;

- Uygulama açıldığında daha önceden ilgili fonksiyonlar için veri tabanında tanımlı zaman var mı şeklinde kontrol yapılır. Eğer var ise o izin otomatik olarak veri tabanındaki zamana göre verilir. Buradaki amaç kullanıcıyı rahatsız etmeden kullanıcı deneyimini öne çıkarmaktır.
- Diğer süreç (Şekil 13)'te gösterildiği gibi devam edecektir.

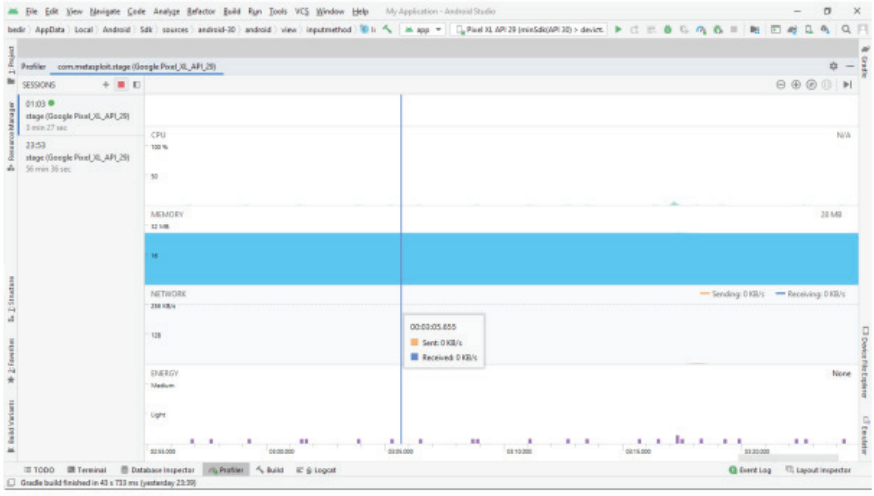


Şekil 15. Uygulama Yeniden Açıldığında Kullanılacak İzin Akışı

2.3.1 Önerilen izin sisteminin güvenlik testi ve performansı

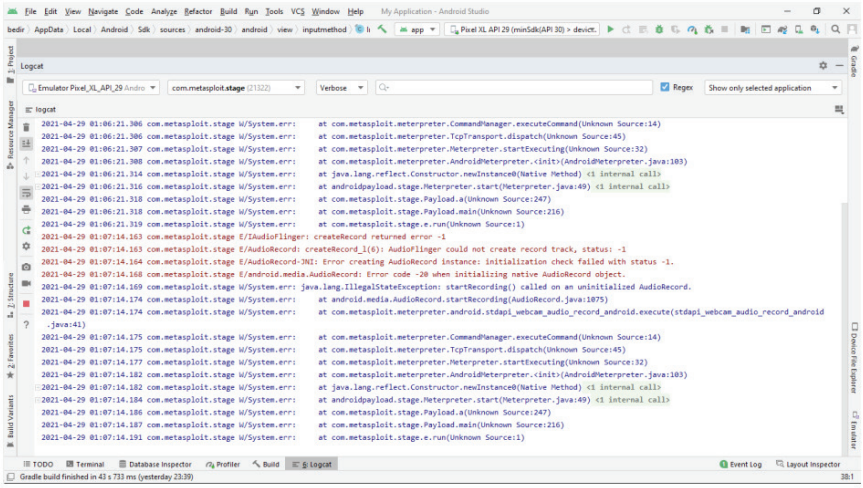
Mevcut izin sisteminin güvenlik testindeki gibi Kali Linux üzerinden sızma testi yapılmıştır. Android stüdyo üzerindeki emülatör kullanılarak uygulamanın performansı (Şekil 16)'da gösterilmiştir. Tehlikeli izin grubunda bulunan "RECORD_AUDIO" üzerinden sızma denenmiştir. Aynı zamanda önerilen izin sisteminin yapılan saldırıları önlediğini (Şekil 17) ve (Şekil 18)'de Log kayıtları ve Kali Linux ekran görüntülerinde gösterilmiştir.

Kullanıcı uygulamayı kapattığında ya da verilen izin süresi dolduğunda Kali Linux üzerinden bir sızma komutu verilmiştir. Uygulamanın üzerinde herhangi bir işlem başlatılamadığı görülmektedir. Bu şekilde sızma işleminin önlendiği görülmüştür.

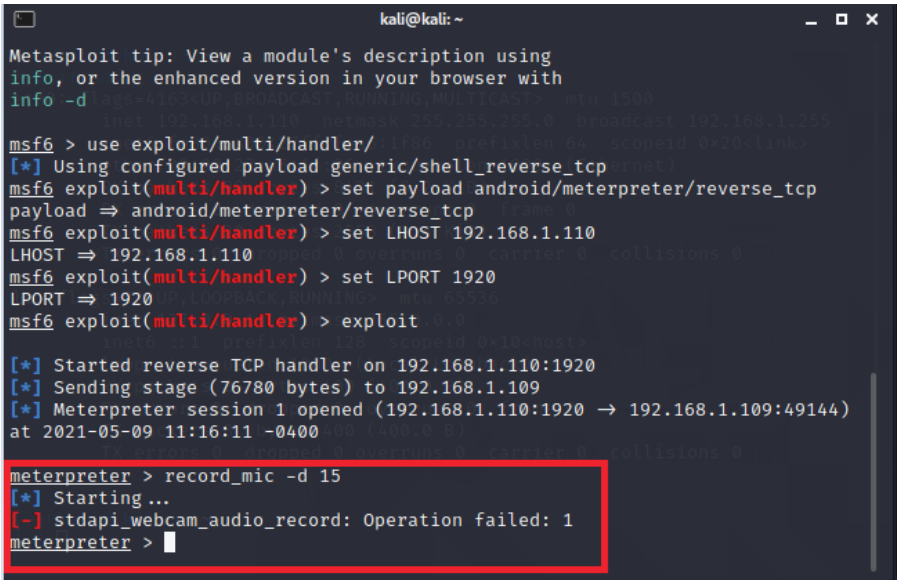


Şekil 16. Sızma Anında Uygulamanın Performansı

Kırmızı ile belirtilen Log kayıtlarında sızma işleminin başarısız olduğu görülmektedir.



Şekil 17. Sızma Anında Log Kayıtları



Şekil 18. Sızma İşleminin Başarısız Olması

3. Bulgular ve Tartışma

Eğer bir uygulama kötü amaçlı ise tehlikeli izinleri çok kapsamlı istemektedir. Bu izinler genelde kullanıcının bilgi güvenliğini tehlikeye atan izin gruplarıdır. Bunların bazıları Tablo3'te isimleri ve etki alanları ile birlikte gösterildi [18].

Tablo 3. Tehlikeli İzinler ve Etkileri

İzin İsmi	**İzin Açıklaması**	**İznin Yapabileceği Etkiler**
READ_CONTACTS	Uygulama bütün rehbere ulaşabilir.	Rehberdeki kişilerin bilgilerini görüntüleyebilir.
GET_ACCOUNTS	Bu izin telefonda bulunduğunuz hesaplara erişim izni de sağlar.	Google, Facebook, Instagram ve benzer diğer hesaplar.
ACCESS_FINE_LOCATION	Uygulama sürekli cihazın konumuna ulaşabilir.	Hırsızlar sizin evde olmadığınız zamanı öğrenebilirler.
RECORD_AUDIO	Uygulama etrafta olan tüm sesi kaydedebilir.	Reklam işlemleri için kullanıcıların tercihleri belirlenebilir.
BODY_SENSORS	Bu izin belli sensörlerden toplanan sağlık verilerinize erişime olanak sağlar.	Eğer vücut sensörü kullanan bir aksesuar kullanıyorsanız, uygulama sağlığınız hakkındaki verileri toplayabilir.
READ_SMS	Kayıtlı SMS mesajlarına ulaşılabilir.	Trojan bu özelliği kullanarak kullanıcının haberi olmadan para transferi gerçekleştirebilir.
READ_EXTERNAL_STORAGE	SD kartını veya diğer depolama birimlerini okuyabilir.	Uygulama cihazınızda bulunan her türlü dosyayı okuyabilir.

Birçok kullanıcı uygulamanın nasıl çalıştığını veya izinleri neden istediğini bilmemektedir. Bu yüzden mevcut izin sisteminin yetersizliği kullanıcıya başka seçenekler sunamamaktadır. Aynı zamanda yapılan araştırmalara göre önerilen diğer izin sistemi yaklaşımlarından elde edilen bilgiler incelendiğinde eksik taraflar aşağıda belirtilmiştir;

- Eylem kaydedici, kullanıcının eylem dizilerini kaydeder. Ardından, Grafik oluşturucu, eylem dizilerinden kullanıcı eylem modelini oluşturur. Grafik eşleştirici, kullanıcı eylem modeli ve paylaşılan kaynakların kesin referans etkileşim grafikleriyle eşleşir ve mevcut kaynaklara erişimin verilip verilmeyeceğine karar verir.[6] Kullanıcının bu eylemlerini sürekli kaydetmek cihazın performansını ve veri alanını kötü yönde etkileyecektir. Tüm Android modelleri için farklı ara yüzler olabilir. Bu yüzden her model için farklı bir eylem kaydedici algoritmaları oluşturmak gerekebilir.
- Esneklik ve izolasyon sağlayan genişletilebilir izin eklentileri için bir çerçeve olan Dalf'ı öneriyoruz. Dalf'ın altında yatan fikir, izin eklentilerinin kendilerinin uygulama olarak ele alınması gerektiğidir. Bu yaklaşım, eklentilerin durumu korumasına ve Android'in işlem izolasyon mekanizmaları tarafından kısıtlanırken cihazın konumu gibi sistem kaynaklarına erişmesine olanak tanır.[7] Bu eklentiler üzerinden izin sistemi kurmak kullanıcının işlemini yavaşlatacaktır. Çünkü kullanıcı bir fonksiyonu kullanmak istediğinde izin onay sürecinin işleyişi hızlı olmalıdır. Ama burada güvenlik için izin süreç hızının performansından fedakârlık edilmiştir.
- Apex, bir kullanıcıya telefon kaynaklarının farklı uygulamalar tarafından kullanımını kısıtlamak için çeşitli seçenekler sunar. Kullanıcı bazı izinleri verebilir ve diğerlerini reddedebilir. Bu, kullanıcının, uygulama tarafından sağlanan işlevselliğin bir kısmını kullanmasına izin verirken, yine de kritik veya maliyetli kaynaklara erişimi kısıtlar. Apex ayrıca, kullanıcının kaynakların kullanımına çalışma zamanı kısıtlamaları getirmesine izin verir.

Son olarak, kullanıcı, bir uygulamanın kullanımına bağlı olarak kaynakların kullanımını, örneğin her gün gönderilen SMS mesajlarının sayısını sınırlamak isteyebilir. Apex'in anlamını ve bu kısıtlamaları tanımlamak için kullanılan politika modelini tanımlıyoruz [10]. Burada fonksiyonun ne kadar kullanılacağı sınırlandırılmıştır. Kullanıcı bir SMS fonksiyonu için günlük 10 SMS yollama sınırı eklese bile uygulama sürekli SMS'leri okuyabilecektir. Bu durum güvenlik ihlalinin önüne geçemeyecektir.

Önerilen zaman tabanlı izin sistemi diğer çalışmalara kıyasla;

- Tüm Android cihazlar için uygulanabilir.
- Kullanıcı deneyimini esas alan iş akışı sayesinde cihazın performansını minimum şekilde etkilemektedir.
- Zaman tabanlı olduğu ve uygulama kapandığında tüm izinler iptal edildiği için cihazın güvenliği yüksek oranda sağlanmış olmaktadır.

Bu makalede mevcut izin sisteminin birçok konuda yetersiz olduğu da gösterildi. Mevcut izin sisteminin yetersiz olduğu ana başlıklar;

1. Kullanıcıya sadece "izin ver" ve "reddet" seçeneklerinin sunulması
2. Eğer kullanıcı uygulamanın tehlikeli izin talebini onaylarsa uygulamanın kapalı olsa veya açık olsa da istediği zaman bu izin verilen Android fonksiyonunu kullanabilmesi
3. Kullanıcılar bir uygulamanın sistemsel olarak nasıl işlediğini bilmeyebilirler. Bu yüzden verilen izinlerin kullanıcıların manuel bir şekilde ayarlardan yönetmesini beklemek
4. Eğer kullanıcı uygulamanın tehlikeli izin talebini reddederse kullanıcının karşısına belli aralıklarla iletişim kutusu çıkararak kullanıcıyı izin vermeye zorlamak ve uygulamanın konforlu kullanılmaması
5. Kullanıcı uygulamanın talep ettiği izinleri onayladı. Daha sonra kullanıcının isteği ve bilgisi dışında uygulamanın sürekli bu fonksiyonları kötü amaçlı kullanması ve bu yüzden

gerektiğinden fazla cihazın performansını olumsuz etkilemesi. Özellikle internet daha sonra işlemci ve batarya üzerinde olumsuz performans etkileri görülmektedir.

Önerilen zaman tabanlı izin sistemi ile yukarıda belirtilen maddelere getirdiğimiz çözümler yukardaki belirtilen sıra ile aşağıda gösterilmiştir;

1. Uygulamanın talep ettiği izinler için kullanıcıya zaman tabanlı izin önerilmiştir. Kullanıcı "izin ver" seçeneğini seçince sonra açılan zaman panelinden iznin kullanılacağı süreyi kendisi belirlemektedir. Bu şekilde uygulamanın fonksiyonu kullanarak cihaz üzerinde kullanıcının bilgilerine ulaşabileceği süreyi kısıtlamıştır.
2. Zaman tabanlı izin sisteminde uygulama kapandığında tüm tehlikeli izinler iptal olur. Uygulama açık olsa dahi verilen izin süresi dolmuş ise bu fonksiyon uygulama tarafından kullanıcının onayı olmadan kullanılamaz.
3. Kullanıcı uygulamayı yeni yüklediğinde ilgili fonksiyon için izne bir süre belirler. Bu süre veri tabanına o fonksiyon özelinde kaydedilir. Kullanıcı uygulamaya yeniden girmek istediğinde ilgili fonksiyon için veri tabanında en son kullanılan süre baz alınarak fonksiyona bu izin süresi atanır. Bu şekilde kullanıcı deneyimi baz alınarak iznin manuel şekilde düzenlenmesinin önüne geçilmiştir.
4. Eğer kullanıcı "reddet" seçeneğini kullanırsa kullanıcının önüne sürekli iletişim kutusu çıkarılmayacak.
5. Zaman tabanlı bir izin verileceği için uygulamanın ilgili fonksiyonları kullanması kısıtlanacaktır. Sadece kullanıcının izne verdiği süre kadar bu fonksiyonlar kullanılacaktır. Bu şekilde hem cihazın performansı olumsuz etkilenmeyecek hem de bilgi güvenliği sağlanmış olacaktır.

Tehlikeli izinler için önerilen bu yapı sayesinde kullanıcıların bilgili güvenliğini sağlama hedeflendi. Yapılan testler sonucu önerilen zaman tabanlı izin sisteminin %100 koruma sağladığı söylenemez.

Çünkü kötü amaçlı uygulamalar izinlere verilen süre içinde yine kullanıcıya ait bilgilere ulaşabilirler. Bu çalışmada verilen izin süresinin sabit olmaması ve kullanıcının deneyimine göre belirlenmesi sızma işlemlerini kısıtlamış olacaktır.

4.Sonuç

Mobil sistem, sistem kaynaklarını yönetmek ve kullanıcı gizliliğini korumak için izinleri kullanır. Bu makalede, kaynak paylaşımının getirdiği bazı kavram kanıtı saldırılar gerçekleştirildi. Mevcut izin kontrol yöntemlerinin kaynak paylaşımının güvenliğini garanti edemediğini ve yetersiz olduğu Tablo 4'te sızma sırasında cihazın internet performansı ile gösterilmiştir. Uygulama kapalı ve kullanıcının bilgisi olmadan yapılan saldırıların sonuçları gösterilmiştir. Tablo 4'te görüldüğü gibi her sızma sonucunda en çok dışarı gönderilen veriler internet performansını etkilemiştir. Çünkü uygulama, kullanıcının bilgisi olmadan cihazdan elde ettiği verileri cihazın internetini kullanarak başka kişi veya kişilere aktarmıştır. Bu durum cihazın özellikle internetini ve daha sonra saldırıların artırılması ile bataryasını da etkilemiştir.

Tablo 4. Sızma Çeşitleri ve İnternet Performansları

İzin Sistemi	**Received (Alınan)**	**Sent (Gönderilen)**	**Sızma Çeşidi**
Mevcut İzin Sistemi	6 KB/s	217,4 KB/s	Ses Kaydı
Mevcut İzin Sistemi	0 KB/s	107 KB/s	SMS Verileri
Zaman Tabanlı İzin Sitemi	0 KB/s	0 KB/s	Ses Kaydı

Kullanıcı deneyimine dayalı, müdahaleci olmayan, kullanıcı dostu zaman tabanlı izin sistemi önerilmiştir. Önerilen izin sistemin performansını göstermek için kavram kanıtı saldırıları gerçekleştirildi. Kullanıcıya "izin ver" ve "reddet" seçeneklerinin dışında başka bir

seçenek olan izin süresi sunulmuştur. Kullanıcının belirlediği izin süresi ile kötü amaçlı uygulamaların özel bilgilere ulaşması kısıtlanmıştır. Testlerin sonucunda Tablo 4'te görüldüğü gibi "Zaman Tabanlı İzin Sistemi" üzerinden yapılan sızma hakkında gerçek veriler incelendiğinde masum olmayan Android uygulama dünyası için alınması gereken önlemleri küçümsememek gerekir. Bu makale, Android paylaşılan kaynaklarının izin kontrolü hakkındaki önerileri paylaşıyor. Zaman tabanlı izin yaklaşımı Android işletim sisteminin performans ve güvenlik açısından elini güçlendirecek bir yapıdır.

Kaynaklar

[1] Alepis E., Patsakis C., Unravelling security issues of runtime permissions in android. J. Hardw. Syst. Secur., vol. 3, no. 1, (2019), (pp. 45–63).

[2] Hatamian M., Engineering privacy in smartphone apps: A technical guideline catalog for app developers. IEEE Access, vol. 8, (2020), (pp. 35429–35445).

[3] Alenezi M. , Almomani I., Abusing Android permissions: A security perspective in Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT), (2017), (pp.1–6).

[4] Xiao J., Chen S., He Q, Feng Z., Xue X,An Android application risk evaluation framework based on minimum permission set identification. J. Syst. Softw., vol. Art. no. 110533163, (2020).

[5] Almomani I., Khayer A.,A Comprehensive Analysis of the Android Permissions System. IEEE Access ,vol. 8, (2020),(pp. 216671- 216688).

[6] Wu.H ,Qin Z.,Tian X.,Sun E.,Xu F., Zhong S., Broken Relationship of Mobile User Intentions and Permission Control of Shared System Resources. IEEE Access, (2019), Nanjing University.

[7] Raval N. ,Razeen A. , Machanavajjhala A. ,Cox L. , Warfield A.,Permissions Plugins as Android Apps. Proceedings of the 17th Annual International Conference on Mobile Systems. , (2019),(pp.180-192).

[8] Baykara M.,Çolak E. ,A review of cloned mobile malware applications for android devices. IEEE Access, (2018), Fırat Üniversitesi.

[9] Dar M., A Novel Approach to Enhance the Security of Android based Smart phones. 2017 International Conference on Innovations in information , (2017), Hindistan.

[10] Nauman M.,Khan S.,Zhang X. ,Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints., ASIACCS , (2010),Çin.

[11] Hur, A.,Razzaq ,A.,Ahmad,F.,Masood,M.,. Cyber Security: Threats, Reasons, Challenges, Methodologies and State of the Art Solutions for Industrial Applications, (2013), (pp.2718-2752).

[12] Thakur , K.,Qiu,M.,Gai, K.,Ali, L.,An Investigation on Cyber Security Threats and Security Models. IEEE Access, (2015), (p.p.307-311).

[13] Felt,A.P,Egelman,S.,Finifter,M.,Akhawe,D., How to ask for permission. Proceedings of the 7th USENIX conference on Hot Topics in Security, (2012).

[14] Tambe,V.,Chauhan,D.,Kulal,S.,Sherkhane,S., Offline Mobile Security. 2018 International Conference on Smart City and Emerging Technology (ICSCET),(2018),Hindistan

[15] https://developer.android.com/guide/topics/permissions/overview , (Erişim Tarihi: 15.03.2021).

[16] https://kali.org , (Erişim Tarihi: 17.03.2021).

[17] https://developer.android.com/studio , (Erişim Tarihi: 10.03.2021).

[18] https://www.kaspersky.com.tr/blog/android-permissions-guide/2956/, (Erişim Tarihi: 02.04.2021).