



**MOBİL SIZMA TESTLERİ İÇİN UYGULAMA ÇATISI MODELİNİN  
GELİŞTİRİLMESİ**

**Berkecan ÖZGÜR**

**YÜKSEK LİSANS TEZİ  
BİLGİ GÜVENLİĞİ MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**HAZİRAN 2023**

## ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Berkecan ÖZGÜR

22/06/2023

# MOBİL SIZMA TESTLERİ İÇİN UYGULAMA ÇATISI MODELİNİN GELİŞTİRİLMESİ

(Yüksek Lisans Tezi)

Berkecan ÖZGÜR

GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2023

## ÖZET

Mobil cihaz kullanımı, mobil alandaki teknolojinin gelişmelerle yaygınlaşmıştır. Mobil uygulamaların kullanımı giderek arttıkça, uygulamaları test etmek için etkili yöntemlere de ihtiyaç artmaktadır. Güvenlik analistleri, mobil uygulamaları analiz etmek için; statik analiz, dinamik analiz, ağ analizi ve sunucu tarafı analizler gibi analiz yöntemleri kullanmaktadır. Her analiz aracının güçlü ve zayıf olduğu yönleri vardır. Bazı araçlar statik veya dinamik analizi öncelikleyebiliyorken, bazıları statik ve ağ analizini öncelikleyebilmektedir. Mobil uygulama penetrasyon testleri sürecinde, güvenlik analistleri analiz sürecinin sağlıklı ve standartlara uygun bir şekilde ilerlemesi için belirli rehberler kullanırlar. Bu rehberler: OWASP Mobil Uygulama Güvenlik Doğrulama Standardı (MASVS), OWASP Mobil Güvenlik Test Rehberi (MSTG) ve OWASP Mobil Uygulama Güvenlik Kontrol Listesi'dir. Uygulama kapsamında, Java tabanlı sunucu uygulaması ve React tabanlı ön yüz uygulaması geliştirilmiştir. OWASP standartlarını otomatize hale getiren ve bunları bir çerçeve olarak sunan kompakt bir araç sunulmaktadır. Araç analistlere, analiz edilen mobil uygulamanın zayıflık puanını hesaplamak için bir zayıflık puanlama arayüzü ve mobil uygulamanın potansiyel olarak sahip olabileceği zayıflıkları gösteren bir zafiyet raporu sunulmaktadır. Analistlere, konfigüratif kurallar ile puanlamaya etki edebilecekleri bir yapı sunulmaktadır. Önerilen uygulama çatısı, uygulamalar üzerinde gerçekleştirilen testlerde, potansiyel veri sızıntısı tespit sayısında MOBSF aracı ortalama 111,75 sayıda tespit yaparken, önerilen uygulama çatısı 192,90 tespit yapmıştır. Veri sızıntısına sebep olabilecek başka bir öncül olan tespit edilen URL sayısında MAPTF (Mobile Application Penetration Test Framework) ortalama 109,06 sayıda bulgu tespit etmiş, MOBSF ortalama 21,56 sayıda bulgu tespit etmiş, Qark ortalama 114,84 sayıda bulgu tespit etmiştir. Zafiyete sahip uygulamaların karşılaştırılmasında VirusTotal aracı temel olarak alınmıştır. MOBSF ile yapılan karşılaştırmalarda, VirusTotal tarafından belirlenen zafiyetli uygulamalar her iki araçta zafiyet tespit puanlarına uygun tutarlılıkta sonuç verirken, zafiyete sahip olmayan 6 uygulamayı MOBSF zafiyete sahip olarak kategorize ederken, MAPTF aracı potansiyel zafiyet tespit etmemiştir.

Bilim Kodu : 92403

Anahtar Kelimeler : Mobil Güvenlik, OWASP MASVS, OWASP MSTG, OWASP MASTG, Güvenlik, Statik Analiz, Zafiyet, Sızma Testi

Sayfa Adedi : 53

Danışman : Doç. Dr. İbrahim Alper DOĞRU

# DEVELOPMENT OF AN APPLICATION FRAMEWORK MODEL FOR MOBILE PENETRATION TESTING

(M. Sc. Thesis)

Berkecan ÖZGÜR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2023

## ABSTRACT

The use of mobile devices has become widespread with the advances in technology in the mobile space. As the use of mobile applications is increasing, the need for effective methods to test applications is also increasing. Security analysts use analysis methods such as static analysis, dynamic analysis, network analysis and server-side analysis to analyze mobile applications. Each analysis tool has its strengths and weaknesses. Some tools may prioritize static or dynamic analysis, while others may prioritize static and network analysis. In the process of mobile application penetration testing, security analysts use certain guidelines to ensure that the analysis process proceeds in a healthy and standardized manner. These guidelines are OWASP Mobile Application Security Verification Standard (MASVS), OWASP Mobile Security Testing Guide (MSTG) and OWASP Mobile Application Security Checklist. Within the scope of the application, a Java-based server application and a React-based front-end application were developed. A compact tool is presented that automates the OWASP standards and presents them as a framework. The study provides analysts with a vulnerability scoring interface to calculate the vulnerability score of the mobile application being analyzed and a vulnerability report showing the vulnerabilities that the mobile application could potentially have. Analysts are provided with a structure where they can influence the scoring with configurative rules. In the tests performed on the applications, the MOBSF tool detected an average of 111.75 potential data leaks, while the proposed application framework detected 192.90. In the number of detected URLs, which is another precursor to data leakage, MAPTF (Mobile Application Penetration Test Framework) detected 109.06 detections on average, MOBSF detected 21.56 detections on average, and Qark detected 114.84 detections on average. The VirusTotal tool was used as the basis for comparing vulnerable applications. In the comparisons made with MOBSF, the applications with vulnerabilities identified by VirusTotal gave results consistent with the vulnerability detection scores of both tools, while MOBSF categorized 6 applications without vulnerabilities as vulnerable, while the MAPTF tool did not detect potential vulnerabilities.

Science Code : 92403

Key Words : Mobile Security, OWASP MASVS, OWASP MSTG, OWASP MASTG, Security, Vulnerabilities, Static Analysis, Penetration Testing

Page Number : 53

Supervisor : Assoc. Prof. Dr. İbrahim Alper DOĞRU

## TEŞEKKÜR

Yüksek lisans çalışmalarımın her aşamasında kıymetli katkıları ve eleştirileriyle yol gösteren, beni her zaman çalışmalarımın ilerlemesi yönünde teşvik eden değerli danışmanım Doç. Dr. İbrahim Alper DOĞRU'ya teşekkür ederim. Ayrıca, bu süreç boyunca her türlü zorluğa rağmen desteklerini ve sevgilerini benden esirgemeyen eşim, annem, babam ve kardeşime tüm kalbimle sonsuz teşekkür ediyorum.



## İÇİNDEKİLER

|                                                                                  | Sayfa |
|----------------------------------------------------------------------------------|-------|
| ÖZET .....                                                                       | iv    |
| ABSTRACT .....                                                                   | v     |
| TEŞEKKÜR .....                                                                   | vi    |
| İÇİNDEKİLER .....                                                                | vii   |
| ÇİZELGELERİN LİSTESİ .....                                                       | ix    |
| ŞEKİLLERİN LİSTESİ.....                                                          | x     |
| SİMGELER VE KISALTMALAR .....                                                    | xi    |
| 1. GİRİŞ.....                                                                    | 1     |
| 2. LİTERATÜRDE YER ALAN ÇALIŞMALARIN KİYASLANMASI..                              | 3     |
| 2.1. Statik Analiz .....                                                         | 3     |
| 2.2. Dinamik Analiz .....                                                        | 4     |
| 2.2. Ağ Analizi .....                                                            | 5     |
| 2.2. Hibrit Analiz.....                                                          | 5     |
| 3. MATERYAL VE YÖNTEM.....                                                       | 13    |
| 3.1. Tarama, Bilgi Eldesi, Raporlama ve Analiz Sürecinde Kullanılan Araçlar..... | 15    |
| 3.1.1. Apktool.....                                                              | 15    |
| 3.1.2. Androguard .....                                                          | 15    |
| 3.1.3. Androbugs.....                                                            | 16    |
| 3.1.4. Qark.....                                                                 | 16    |
| 3.1.5. Mobsf .....                                                               | 16    |
| 3.1.6. Drozer.....                                                               | 17    |
| 3.1.7. Burp Suite .....                                                          | 17    |
| 3.1.8. Nmap.....                                                                 | 17    |

|                                                                                  | <b>Sayfa</b> |
|----------------------------------------------------------------------------------|--------------|
| 3.1.9. Owasp Zed Attack Proxy.....                                               | 18           |
| 3.1.10. Virustotal.....                                                          | 18           |
| 3.2. Tarama, Bilgi Eldesi, Raporlama ve Analiz Sürecinde Kullanılan Yöntemler .. | 20           |
| 3.2.1. Statik analiz.....                                                        | 21           |
| 3.2.2. Dinamik analiz .....                                                      | 21           |
| 3.2.3. Ağ analizi .....                                                          | 22           |
| 3.2.4. Sunucu tarafı analizi .....                                               | 22           |
| 3.2.5. Zafiyet taraması .....                                                    | 23           |
| 3.2.6. Zafiyet taraması yorumlarının sunulması.....                              | 23           |
| 3.2.7. Anahtar kelime taraması .....                                             | 23           |
| 3.2.8. Kaynak kodlarına ayrıştırma aşaması.....                                  | 24           |
| 3.2.9. Analiz raporunun oluşturulması .....                                      | 24           |
| 3.2.10. Veri tabanı desteği .....                                                | 25           |
| 3.2.11. Kullanıcı arayüz desteği.....                                            | 25           |
| 3.2.12. REST API dokümantasyonu.....                                             | 26           |
| 3.2.13. Zafiyet temelli puanlama .....                                           | 26           |
| 3.2.14. Puanlama kurallarının özelleştirilmesi .....                             | 26           |
| 3.2.15. Analiz otomasyonu .....                                                  | 27           |
| 4. BULGULAR VE TARTIŞMA .....                                                    | 35           |
| 5. SONUÇ .....                                                                   | 41           |
| KAYNAKLAR.....                                                                   | 43           |
| EKLER .....                                                                      | 47           |
| EK-1. MAPTF Kapsam Matrisi.....                                                  | 48           |
| ÖZGEÇMİŞ .....                                                                   | 53           |

## ÇİZELGELERİN LİSTESİ

| Çizelge                                                                | Sayfa |
|------------------------------------------------------------------------|-------|
| Çizelge 2.1. Literatürdeki akademik çalışmalar .....                   | 8     |
| Çizelge 3.1. MAPTF kapsam matrisi .....                                | 30    |
| Çizelge 3.2. MAPTF puanlama zafiyetleri sınıflandırma aralıkları ..... | 32    |
| Çizelge 4.1. Mobil uygulama analiz araçlarının karşılaştırılması ..... | 36    |
| Çizelge 4.2. Mobil uygulama bazlı zafiyet puanı karşılaştırması.....   | 37    |



## ŞEKİLLERİN LİSTESİ

| Şekil                                                                       | Sayfa |
|-----------------------------------------------------------------------------|-------|
| Şekil 3.1. MAPTF Yetenek Haritası.....                                      | 14    |
| Şekil 3.2. MAPTF analiz kapsamında alt analiz yöntemleri ve araçları .....  | 19    |
| Şekil 3.3. MAPTF analiz adımları ve süreç akışı.....                        | 20    |
| Şekil 3.4. Puan güçlendirme mekanizması.....                                | 29    |
| Şekil 3.5. Puanlama süreci genel bakış .....                                | 32    |
| Şekil 3.6. Mobil Uygulama Güvenlik Analisti Arayüz Kullanımı gösterimi..... | 34    |

## KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

| Kısaltmalar  | Açıklamalar                                       |
|--------------|---------------------------------------------------|
| <b>APK</b>   | Android Package Kit                               |
| <b>CWE</b>   | Common Weakness Enumeration                       |
| <b>CVSS</b>  | Common Vulnerability Scoring System               |
| <b>CVE</b>   | Common Vulnerabilities and Exposures              |
| <b>HTTP</b>  | Hyper Text Transfer Protokol                      |
| <b>HTTPS</b> | Hypertext Transfer Protocol Secure                |
| <b>JWT</b>   | JSON Web Token                                    |
| <b>MAPTF</b> | Mobile Application Penetration Test Framework     |
| <b>MASVS</b> | Mobile Application Security Verification Standard |
| <b>MSTG</b>  | Mobile Security Testing Guide                     |
| <b>NIST</b>  | National Institute of Standards and Technology    |
| <b>NVD</b>   | National Vulnerability Database                   |
| <b>OWASP</b> | Open Web Application Security Project             |
| <b>REGEX</b> | Regular Expression                                |
| <b>REST</b>  | Representational State Transfer                   |
| <b>SDK</b>   | Software Development Kit                          |
| <b>TUIK</b>  | Türkiye İstatistik Kurumu                         |
| <b>YAML</b>  | YAML Ain't Markup Language                        |

## 1. GİRİŞ

Son yıllarda mobil cihazların hızla gelişmesi, bilginin istenildiği zaman ve mekândan bağımsız olarak kolaylıkla erişilebilmesini sağlamaktadır. GlobalStats tarafından 2023 yılında yayımlanan bir rapora göre, Android, iOS ve Windows Phone işletim sistemlerinin pazar payları şu şekildedir: Android %70,93, iOS %28,37, Windows Phone %0,02 ve diğer işletim sistemleri ise %0,67'dir [1]. Ayrıca, TÜİK tarafından 2022 yılında gerçekleştirilen bir bilgi teknolojisi kullanımı araştırmasına göre, evlerdeki internet erişim oranının %94,1 olduğu tespit edilmiştir. Bu oranın bir önceki yıl %92,0 olduğu görülmüştür [2].

Mobil uygulamaların pazar yerlerine sunulma sürecinde, pazar hizmet sağlayıcıları genellikle potansiyel güvenlik açıklarını tespit etmek amacıyla güvenlik analizleri yapmaktadır. Ancak, Play Store ve AppStore gibi uygulama mağazalarında yer alan uygulamalardaki güvenlik açıklarının tamamının tespit edilmesi mümkün olmamaktadır. Mobil cihaz kullanıcıları, hangi uygulamaların güvenli olduğunu ve hangilerinin kişisel verilerin kötüye kullanımına neden olabileceğini bilmek istemekte; aynı şekilde uygulama mağazası sağlayıcıları da düşük kaliteli veya kötü niyetli uygulamaların piyasaya sürülmesini engellemek için en etkili yöntemleri araştırmaktadır [3]. Güvenlik araştırmacıları, uygulamalardaki güvenlik açıklarını tespit etmek için güvenlik analiz araçlarından faydalanmaktadır. Mobil uygulamalar, yalnızca resmi pazar yerlerinden değil, üçüncü taraf mağazalardan ve geliştiriciler tarafından da dağıtılabilmektedir. Bu durumda, mobil kullanıcılar güvenilir ve güvenilmeyen kaynaklardan bu uygulamalara erişebilmektedir. Dolayısıyla, uygulama dağıtıcıları tarafından bu tür testlerin yapılmaması, mobil uygulamalarda güvenlik açıklarının ve zayıflıkların ortaya çıkmasına neden olabilmektedir [4].

Mobil uygulama güvenlik analistleri, bir mobil uygulamanın zafiyet durumunu değerlendirmek için çeşitli tekniklerden yararlanmaktadır. Bu teknikler arasında statik analiz, sunucu taraflı analiz, dinamik analiz, hibrit analiz ve ağ analizi bulunmaktadır. Güvenlik analistleri, bu teknikleri tek tek veya toplu olarak uygulayarak mobil uygulamanın güvenlik durumunu değerlendirebilmektedir. Ancak, bu aşamada, bu tekniklerin tamamını veya bir kısmını uygulamak için kabul edilmiş bir standart metodoloji bulunmaması nedeniyle zorlu ve kapsamlı bir süreç ortaya çıkmaktadır. Bu

değerlendirmelerin sonuçları, kullanıcılara hangi uygulamaların güvenli olduğu ve hangilerinin zafiyete sahip olduğu konusunda bilgi sunmaktadır. Böylece, kullanıcılar daha bilinçli bir şekilde mobil uygulamalarını seçebilmekte ve güvenlik açıklarına karşı daha korunaklı hale gelebilmektedirler.

Mobil uygulama penetrasyon testi sürecinde, güvenlik analistleri analiz sürecinin sağlıklı ve standartlara uygun bir şekilde ilerlemesi için belirli rehberleri kullanmaktadır. Bu rehberlerden bazıları OWASP Mobil Uygulama Güvenlik Doğrulama Standardı (MASVS), OWASP Mobil Güvenlik Test Rehberi (MSTG) ve OWASP Mobil Uygulama Güvenlik Kontrol Listesi'dir. Bu rehberler, analistlere mobil uygulamaların güvenlik açıklarını tespit etmek ve gidermek için gerekli adımları sunmaktadır.

Çalışma kapsamında, mobil uygulama güvenliği alanında önemli yenilikler sunulması hedeflenmiştir. Birinci olarak, mobil uygulamaların güvenlik analiz süreçlerini iyileştirerek analistlere daha etkili araçlar sunmaktır. İkinci olarak, güvenlik zafiyetlerini puanlama ve değerlendirme konusunda analistlere yardımcı olan bir zafiyet temelli puanlama ara yüzü sağlayarak, mobil uygulama güvenliği alanında bir iyileştirme sağlamaktır. Üçüncü olarak, analiz edilen mobil uygulamanın zayıflıklarını kapsamlı bir şekilde ortaya koyabilen ve analistlere daha detaylı bir bakış açısı sağlayan kapsamlı bir zafiyet raporu üretmektir. Dördüncü olarak, mobil uygulama güvenlik analizlerini esnek hale getirirken aynı zamanda analistlere özelleştirilebilir kurallar ve konfigürasyon seçenekleri sunarak analiz sürecini kişiselleştirebilme imkânı sunmaktır. Son olarak, veri tabanı entegrasyonu ile analiz raporlarının saklanması ve izlenmesini sağlamakta, böylece analiz sürecinin ilerleyişini gözlemlene olanağı sunmaktır.

Bu kapsamda kullanılan yöntemler ayrıntılı olarak sonraki bölümlerde açıklanmıştır. Çalışmanın kalan bölümleri şu şekildedir. Bölüm 2 ilgili çalışmaların genel bir özetini sunar, Bölüm 3 malzeme ve yöntemi tanımlar, Bölüm 4 deneysel sonuçları ve tartışmaları içerir ve son olarak, Bölüm 5 sonuçları kapsar.

## 2. LİTERATÜRDE YER ALAN ÇALIŞMALARIN KIYASLANMASI

Önerilen model kapsamında, mobil uygulama analizinde kullanılan analiz türleri, bu analiz türlerinin açıklamaları ve her analiz türü için kullanılan analiz araçları aşağıda belirtilmiştir.

### 2.1. Statik Analiz

Statik analiz, mobil uygulama güvenliği alanında önemli bir yöntemdir. Bu yöntem, uygulamanın bir mobil cihaza yüklenmeden önce sahip olduğu bilgilerin ve kodun analiz edilmesini sağlar. Bu analiz süreci, mobil uygulamaların güvenlik açıklarını tespit etmek ve zafiyetleri ortaya çıkarmak için kullanılır. Ayrıca, uygulamanın içerdiği potansiyel risklerin belirlenmesi ve geliştirme aşamasında hataların düzeltilmesi için önemli bir kaynak olarak değerlendirilir. Statik analiz, mobil uygulamaların kaynak kodu ve diğer dosyalarının detaylı bir şekilde incelenmesiyle gerçekleştirilir. Bu süreçte, uygulamanın içerdiği izinler, kullanılan API'ler, veri depolama yöntemleri, veri akışı ve güvenlik kontrolleri gibi faktörler titizlikle değerlendirilir [8]. Aynı zamanda, uygulamanın içerdiği potansiyel zayıf noktalar, yanlış kullanımlar ve kodlama hataları da tespit edilmeye çalışılır.

Bu çalışmada, Android uygulamalarının statik analizi için Qark [5], Androbugs [6] ve Apktool [7] gibi araçlardan yararlandık. Qark, uygulama paketinin içerdiği izinleri, bileşenleri ve potansiyel güvenlik açıklarını tespit etmek için kullanılan bir araçtır. Androbugs, uygulamanın kaynak kodunu analiz ederek, içinde bulunan potansiyel riskleri ve güvenlik açıklarını belirlemeye yardımcı olur. Apktool ise, uygulamanın APK dosyasını çözümleyerek kaynak kodunu elde etmemizi sağlar ve böylece detaylı bir inceleme yapmamızı mümkün kılar. Statik analiz yöntemiyle, mevcut uygulama içerisinde kullanılan izinlerin ve uygulama kaynak kodlarının detaylı bir analizi gerçekleştirilir. Bu analiz, uygulamanın kullanıcı gizliliği, veri güvenliği, kimlik doğrulama mekanizmaları, iletişim protokolleri ve diğer güvenlik önlemleri açısından değerlendirilmesini sağlar. Potansiyel güvenlik açıkları, kodlama hataları, zayıf noktalar ve yanlış kullanımlar tespit edilerek, bu sorunların giderilmesi ve uygulamanın daha güvenli hale getirilmesi için gerekli önlemler alınabilir. Statik analiz, mobil uygulama güvenliği sürecinde önemli bir adımdır ve

güvenlik açıklarının erken tespit edilmesi ve düzeltilmesine yardımcı olur. Bu şekilde, kullanıcıların finansal bilgileri, kişisel verileri ve diğer hassas bilgileri korunurken, uygulamanın güvenilir ve güvenli bir şekilde kullanılması sağlanır. Bu nedenle, statik analiz, mobil uygulama güvenliği sürecinin ayrılmaz bir parçası olarak dikkate alınmalı ve güvenlik standartlarının karşılanmasına katkıda bulunmalıdır.

## 2.2. Dinamik Analiz

Mobil uygulamaların değerlendirilmesi ve incelenmesi amacıyla kullanılan dinamik analiz yöntemi, uygulamanın gerçek zamanlı çalışma sürecindeki performansını değerlendirmek için bir araç olarak kullanılmaktadır. Bu yöntem, mobil uygulamanın çalışması esnasında ortaya çıkan hataları ve hata mesajlarını tespit etmeyi hedeflemektedir. Aynı zamanda, uygulamanın davranışının gerçek zamanlı olarak test edilmesi ve analiz edilmesi için sanal veya fiziksel bir cihaza yüklenmesini mümkün kılar. Bu sayede, uygulamanın anormal davranışlara yol açabilecek çeşitli girdiler ve testlerle değerlendirilmesi sağlanabilmektedir [9].

Dinamik analiz yöntemi, mobil uygulamalarda yaygın olarak kullanılan araçlardan biri olan Frida ile gerçekleştirilebilir [10]. Frida, uygulamanın çalışma sürecindeki davranışını izlemek ve analiz etmek için kullanılan bir araçtır. Bu araç, uygulama içerisindeki veri akışını takip edebilir, bellek işlemlerini izleyebilir ve gerektiğinde uygulamanın kodunu değiştirebilir. Bu şekilde, uygulamanın dinamik davranışını anlama ve potansiyel güvenlik açıklarını tespit etme imkânı sağlanır. Dinamik analiz için bir diğer önemli araç ise Drozer'dır [11]. Drozer, Android uygulamalarının güvenlik açıklarını tespit etmek için kullanılan bir test çerçevesidir. Bu araç, uygulamanın gerçek zamanlı davranışını izleme, istemci-sunucu trafiğini analiz etme, girdileri manipüle etme ve diğer güvenlik testlerini gerçekleştirme gibi işlemlere sahiptir. Drozer, mobil uygulamaların güvenlik açıklarını tespit etme ve giderme konusunda etkili bir araç olarak kabul edilmektedir. Dinamik analiz yöntemi, mobil uygulamaların güvenlik açısından değerlendirilmesinde önemli bir rol oynamaktadır. Uygulamanın gerçek zamanlı çalışma sürecindeki davranışının ayrıntılı bir şekilde incelenmesi, potansiyel güvenlik açıklarının tespit edilmesini ve uygulamanın güvenliğinin artırılmasını sağlamaktadır. Bu nedenle, mobil uygulama geliştirme sürecinde dinamik analiz yöntemi göz önünde bulundurulmalı ve güvenlik testlerinin yapılmasında önemli bir araç olarak kullanılmalıdır.

### 2.3. Ağ Analizi

Mobil uygulamalar, internet protokollerinin kullanımıyla cihaz ve hedef sunucu arasındaki iletişimi sağlarlar. Bu, istemci-sunucu iletişiminin güvenlik düzeyini ve verilerin nasıl iletileceğini belirler. Örneğin, bir bankacılık uygulaması kullanırken, sunucu giriş bilgilerinizi doğrulamazsa sisteme erişim yapılamayacaktır [12]. Ağ analizi, verilerin iletim sırasındaki güvenliğin belirlenmesini, potansiyel veri sızıntılarının, veri ihlallerinin, destek hizmetlerinin kontrolünü, kullanılan şifreleme protokollerini ve bu protokollerdeki zayıflıkları belirlemeyi mümkün kılar. Bilinen güvenlik açıkları olan protokollerin kullanılmaması bir gereklilik olarak ortaya çıkmaktadır. Ağ trafiğini analiz etmek için Burp Suite [13] ve Wireshark [14] gibi araçlar kullanılabilir.

### 2.4. Hibrit Analiz

Hibrit analiz yöntemi, hem statik, dinamik ve ağ analizi tekniklerinin birleştirilmesini, beraber kullanılmasını anlatmak için kullanılan bir analiz türüdür. Bu yaklaşım, tekil analiz şekillerinin her birinin özelliklerini taşır, aynı zamanda daha yüksek bir karmaşıklığa ve konfigürasyona sahiptir. Bu karmaşıklık nedeniyle, uygulama için ek kaynaklar gerekebilmektedir. Ayrıca, hibrit analiz, kaynak kodunun statik analiz ile incelemesinden, uygulamaların çalışma zamanında dinamik olarak incelemesinden ve ağlar üzerinden iletilen verilerin incelemesinden de bilgi toplama kapasitesine sahiptir [15].

Literatürdeki akademik çalışmalar hakkında bilgiler aşağıda verilmiştir.

Acheampong ve Kaur, Gana'daki en büyük dört banka için Android mobil bankacılık uygulamalarındaki güvenlik açıklarını tespit etmeyi amaçlamıştır. Bu çalışmada hem otomatik hem de manuel testlerden oluşan hibrit bir yaklaşım kullanarak finans kurumları için mobil uygulamalarda kapsamlı sızma testi gerçekleştirmeyi amaçlamışlardır [16].

Almohaini, bu çalışma da yüksek doğrulukla fidye yazılımlarını tespit etmek için anti virüs sistemi ile statik analiz sistemini entegre eden, statik ve dinamik analizi birleştiren bir hibrit tespit yöntemi sunmaktadır [17].

Özdemir and Çavşı Zaim 2021 yılında, Android işletim sistemi mimarisini kapsamında, çeşitli zafiyet sınıfları için farklı saldırı senaryoları geliştirmişler ve daha sonra bu tür saldırıların etkisinin azaltılması için öneriler sunmuşlardır [18].

Nilsson, çalışma kapsamında, Android uygulamaları üzerinde yapılacak test süreçleri ile ilgili için bir şablon oluşturmuştur. Araştırmanın sonuçları, mobil uygulamanın güvenliğini içeriğine yönelik öneri veya çözüm sunmamaktadır [19].

Williams ve arkadaşları 2020 yılında, Ulusal Zafiyet Veri tabanından alınan örneklerle, çalışmada kapsamında zafiyetlere zamansal yaklaşımları ortaya çıkarmak, zafiyet değişimini incelemek ve gelecekteki zafiyetleri tahmin yöntemi oluşturmak için çalışma yapmıştır. Entegre veri madenciliği için bir çerçeve olarak difüzyon tabanlı bir hikâye anlatma tekniği olan bir evrim modeli (TSEM) önerilmiştir. [20]

Nunes, Mobil uygulama sızma testi ve mobil kötü amaçlı yazılım analizi için gereksinimleri analiz etmeye odaklanan, mobil kötü amaçlı yazılım analizi ve mobil sızma testi için bir çerçeve önermiş ve uygulanmıştır [21].

Alanda ve arkadaşları, Android işletim sistemlerinin ve uygulamalarının güvenliğini artırmayı amaçlayan mobil uygulamalardaki güvenlik açıklarının belirlenmesi ve değerlendirilmesi için bir yöntem önermiş ve bu açıkların giderilmesine yönelik öneriler sunmuştur. Bu araştırmanın potansiyel sınırlaması, test edilen uygulamaların örneklem büyüklüğünün oldukça küçük olması nedeniyle sonuçların daha geniş bir mobil uygulama kümesinde denenmesi gerekmektedir [22].

Kim 2019 yılında, Android işletim sistemi için bir potansiyel saldırıların tespit için sistem verilerini kullanan bir uygulama geliştirmiştir. Uygulama, günlük analizi ve saldırı ağacını kullanarak mobil saldırıları öncesi ve sonrası olarak iki aşamada tespit etmiş ve sınıflandırma yapmıştır [23].

Yeom ve Won, çalışmaları kapsamında bir kullanıcının mobil cihazındaki tüm uygulamaları otomatik olarak decompile etme prosedürünü oluşturulması ve cihazdaki açıkların listesi ve matrisi oluşturularak aralarındaki uygulama zafiyetlerinin belirlenmesine yönelik bir yöntemin geliştirilmesi amaçlanmıştır [24].

Mishra ve arkadaşları 2022 yılında, çalışma kapsamında, gereksiz izinleri sınırlandırarak ve arka plan veri koruma hizmetiyle kullanıcı verilerini koruyarak uygulama işlevselliğini

ve kullanıcı gizliliğini sağladığı bir uygulama ortaya koymuştur. Önerilen çerçevenin potansiyel sınırlaması, uygulamaların arka plan veri koruma hizmetiyle yeniden paketlenmesini gerektirmesidir [25].

Guyton, çalışma kapsamında, Android cihazlar üzerindeki ağla ilgili verileri analiz etmek ve normal davranıştan sapma belirlemek için karar ağacı algoritması gibi makine öğrenme tekniklerini kullanmıştır. Android intentlerinin zafiyet tanımlama için ayırt edici bir özellik olarak etkililiğini değerlendirmek için çalışma yapmıştır [26].

Abraham ve arkadaşları, bu çalışmada potansiyel zararlı yazılımların tespitini de içeren mobil uygulamaların güvenliğini kapsamlı bir şekilde değerlendirmek için dinamik ve statik analiz tekniklerinin bir arada kombine ederek zafiyet taraması yöntemleri de içeren puan tabanlı bir raporlama aracı ortaya koymuştur [27].

Bhatt 2018 yılında çalışma kapsamında, iOS uygulamalarında gizlilik ihlallerini tespit etmek ve sınıflandırmak için dinamik ve statik analiz teknikleri kombinasyonunu kullanarak, uygulama izin kümelerinin analizini yapıp, her uygulama için risk puanlarını oluşturmuştur [28].

Benkhelifa, çalışma kapsamında, çağrılar, coğrafi konum, multimedia ve tarayıcı geçmişi gibi farklı kaynaklardan veri haritalaması yapmayı amaçlayan kapsamlı bir mobil adli araç sunmayı hedeflemiştir [29].

Lee ve arkadaşları 2016 yılında ortaya koydukları HybriDroid isimli çalışma kapsamında, Android uygulamalar için bir statik analiz çerçevesi ortaya koymuştur. Java ve Javascript 'in betiklerini analiz ederek hataları ve güvenlik açıklarını tespit ve hata bulma işlemi yapmıştır.

Jadhav, önerilen araştırma kapsamında, mobil cihaz güvenliği için mobil cihaz penetrasyon testi çerçevesi ve platformunu sunar. Zafiyet analizi için güvenli sandbox ortamı kullanmıştır. Çalışma zafiyet depoları, günlük toplama ve destek ve mobil zafiyet analizi ve mobil uygulama penetrasyon testi yapmak için araç içermektedir. Mobil uygulama penetrasyon testi teknikleri ile eğitilmesi ve OWASP Mobile Top 10 açıklıklarını tespit etmesini amaçlamıştır.

Aşağıdaki Çizelge 2.1’de literatürdeki akademik çalışmalar hakkında özet bilgi yer almaktadır.

Çizelge 2.1. Literatürdeki akademik çalışmalar

| Çalışmanın Adı                                                                            | Hedef Platform | Yöntem                                                                                                                                                                                                             | Çalışmanın Olumlu ve Olumsuz Yönleri                                                                                                                                                                        | Referans |
|-------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Automatic And Manual Discovery of Vulnerabilities in Selected Android Mobile Applications | Android        | Bu çalışma hem otomatik hem de manuel testlerden oluşan hibrit bir yaklaşım kullanarak finans kurumları için üst düzey mobil uygulamalarda kapsamlı sızma testi gerçekleştirmeyi amaçlamaktadır.                   | Uygulamanın yayınlanmasından önce maksimum güvenliğinin sağlanması için daha verimli yöntem olarak otomatik ve manuel testlerin birleştirildiği hibrit yaklaşım önerilmektedir.                             | [16]     |
| Hybrid-Based Analysis Impact on Ransomware Detection for Android Systems                  | Android        | Uygulama fide yazılımlarını tespit etmek için anti virüs sistemi, statik analiz sistemini entegre eden, dinamik analiz kullanan, statik ve dinamik analizi birleştiren hibrit tespit yöntemi sunar.                | Android fide yazılımların statik, dinamik analiz yöntemlerini kullanarak tespiti için hibrit bir yöntem sunulmaktadır. Önerilen yöntemin sadece 10 aileden fide yazılım örnekleri üzerinde test edilmiştir. | [17]     |
| Investigation Of Attack Types in Android Operating System                                 | Android        | Bu çalışma, Android işletim sistemi mimarisini araştırır, çeşitli zafiyet sınıfları için çeşitli saldırı senaryoları geliştirir ve daha sonra bu tür saldırıların etkisinin azaltılması için öneriler sunar.       | Android OS güvenliğini incelemekte, farklı saldırı senaryoları oluşturulmakta ve bu saldırıların önlenmesi için öneriler sunulmaktadır. Mobil uygulama analiz programlarının kullanımı önerilmiştir.        | [18]     |
| Penetration Testing of Android Applications                                               | Android        | Potansiyel tehditleri belirlemek için bir tehdit modeli sunulmuştur. Test, Android işletim sistemindeki Spotify uygulamasına odaklanmıştır. Test derinliği, saldırı ile ilişkili potansiyel riskler açıklanmıştır. | Uygulama Android uygulamaları üzerinde yapılacak araştırmalar için şablon oluşturmasıdır hedefindedir. Araştırmanın sonuçları, uygulamaların güvenliğine içerisine yönelik öneri veya çözüm sunmamaktadır.  | [19]     |

Çizelge 2.1. (devam) Literatürdeki akademik çalışmalar

| Çalışmanın Adı                                                    | Hedef Platform | Yöntem                                                                                                                                                                                                                                                                                                         | Çalışmanın Olumlu ve Olumsuz Yönleri                                                                                                                                                                                                                                                                                                                                                                                                                    | Referans |
|-------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| A vulnerability analysis and prediction framework                 | Android<br>IOS | NVD' den alınan örneklerle, zamansal yaklaşımları ortaya çıkarmak ve gelecekteki zafiyetleri tahmin etmek için çalışma yapılmıştır. Entegre veri madenciliği için bir çerçeve olarak difüzyon tabanlı olan konusal olarak denetlenen bir evrim modeli (TSEM) önerilmiştir.                                     | Çalışma, Ulusal Zayıflıklar Veri tabanına uygulanmıştır. Bu nedenle, araştırmanın kapsamında belirli bir işletim sistemi hedef alınmamış, çeşitli işletim sistemleri üzerinde model test edilmiştir. Bu modelin genişletilebilirliği ve test edilebilirliği için önemlidir. Ayrıca NVD üzerinden girdilerini aldığı için geniş bir kapsamda çalışılmıştır.                                                                                              | [20]     |
| Droids - A Framework for Security Analysis of Mobile Applications | Android        | Mobil uygulama sızma testi ve mobil kötü amaçlı yazılım analizi için gereksinimleri analiz etmeye odaklanarak, mobil kötü amaçlı yazılım analizi ve mobil sızma testi için bir çerçeve önerilmiş ve uygulanmıştır.                                                                                             | Bu çalışma, güvenlik açıklarını belirlemek için mevcut araçlar ve metodolojilerden yararlanmaktadır. Fakat, analizin doğruluğu açısından geniş bir temelde tarama yapılması önceliklendirilmemiştir. Araştırma, yapıldığı tarihe yakın ortaya çıkan en son güvenlik tehditlerini taranmamıştır.                                                                                                                                                         | [21]     |
| Mobile Application Security Penetration Testing Based on OWASP    | Android        | Android uygulamalarındaki güvenlik açıklarını tespit etmek için öneriler sunulmaktadır. Bu amaçla Android beş farklı mobil uygulamayı test etmek için uygulanan; hazırlık, bilgi toplama, raporlama olmak üzere beş aşamadan oluşan OWASP Mobile Top 10 belgelerine dayalı sızma testi yaklaşımı önerilmiştir. | Araştırma, Android işletim sistemlerinin ve uygulamalarının güvenliğini artırmayı amaçlayan mobil uygulamalardaki güvenlik açıklarının belirlenmesi ve değerlendirilmesi için bir yöntem ve bu açıkların giderilmesine yönelik öneriler sunmaktadır. Bu araştırmanın potansiyel sınırlaması, test edilen uygulamaların örneklem büyüklüğünün oldukça küçük olması nedeniyle sonuçların daha geniş bir mobil uygulama kümesinde denenmesi gerekmektedir. | [22]     |

Çizelge 2.1. (devam) Literatürdeki akademik çalışmalar

| Çalışmanın Adı                                                                       | Hedef Platform | Yöntem                                                                                                                                                                                                                                                    | Çalışmanın Olumlu ve Olumsuz Yönleri                                                                                                                                                                                                                                                                     | Referans |
|--------------------------------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Attack Detection Application with Attack Tree for Mobile System using Log Analysis   | Android        | Araştırma yöntemi, Android OS için bir saldırı tespit uygulamasının geliştirilmesidir. Uygulama, günlük analizi ve saldırı ağacını kullanarak mobil saldırıları öncesi ve sonrası olarak iki aşamada tespit etme ve sınıflandırma yapmaktadır.            | Araştırma sadece smishing ve backdoor gibi iki tür saldırıyı test etmektedir, bu mobil cihazlar üzerinde yapılabilecek farklı saldırı türlerinin küçük bir örneğidir. Bu, önerilen yaklaşımın diğer saldırı türlerine karşı etkisiz olabileceği düşünülmektedir.                                         | [23]     |
| Vulnerability Evaluation Method through Correlation Analysis of Android Applications | Android        | Mobil cihazdaki tüm uygulamaları otomatik olarak decompile etme prosedürünü oluşturulması ve cihazdaki açıkların listesi ve matrisi oluşturularak aralarındaki uygulama zafiyetlerinin belirlenmesine yönelik yöntem geliştirilmesi amaçlanmıştır.        | AndroidManifest dosyası ve Java kaynak kodunun analizine dayanarak, uygulama arası risklerin görselleştirilmesi ve korelasyon analizi kullanılarak saldırıları veri bozulmasını ve sızmasını tanımlamaya ve önlemeye daha verimli ve etkili bir yol sunar. Ancak, bulunmayan açıklıkları kapsamayabilir. | [24]     |
| Privacy Protection Framework for Android                                             | Android        | Yaklaşım, kişisel verilerin alınmasını önlemek için ortak filtreleme ve izin kümesinin, veri madenciliği algoritmaları kullanarak sınırlandırmayı amaçlar. Kötü niyetli uygulamanın riskli API çağrılarını toplamasına ve sunmasına olanak sağlamaktadır. | Çerçevenin potansiyel sınırlaması, uygulamaların arka plan veri koruma hizmetiyle yeniden paketlenmesini gerektirmesidir, bu tüm mobil uygulamalar için mümkün olmayabilir veya uygulamanın işlevselliği ile ilgili sorunlara sebep olabilir.                                                            | [25]     |
| A Survey of Android Security Threats and Machine Learning                            | Android        | Araştırma, Android cihazlar üzerindeki ağla ilgili verileri analiz etmek ve normal davranıştan sapma belirlemek için karar ağacı algoritması gibi makine öğrenme tekniklerini kullanmıştır.                                                               | Çalışma kapsamında, karar ağacı ve makine öğrenmesi tekniklerinin kullanılması zafiyet tespiti için önemlidir.                                                                                                                                                                                           | [26]     |

Çizelge 2.1. (devam) Literatürdeki akademik çalışmalar

| Çalışmanın Adı                                                                                                | Hedef Platform | Yöntem                                                                                                                                                                                                                                                                                                                                          | Çalışmanın Olumlu ve Olumsuz Yönleri                                                                                                                                                                                                    | Referans |
|---------------------------------------------------------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Mobile Security Framework Project                                                                             | Android<br>IOS | Bu araştırma, potansiyel zararlı yazılımların tespitini de içeren mobil uygulamaların güvenliğini kapsamlı bir şekilde değerlendirmek için dinamik ve statik analiz tekniklerinin bir arada kombine ederek kullanır.                                                                                                                            | Önerilen çalışma, mobil uygulamaların statik ve dinamik analizlerinin yapılmasına olanak sağlamaktadır ve yapılan analizler sonucu elde edilen bulgu sayıları kullanıcı ile paylaşılmaktadır.                                           | [27]     |
| iABC: Towards a hybrid framework for analyzing and classifying behaviour of iOS applications using static and | IOS            | Çalışma, iOS uygulamalarında gizlilik ihallerini tespit etmek, sınıflandırmak için dinamik ve statik analiz teknikleri kombinasyonunu kullanarak, uygulama izin kümelerinin analizini yapıp, her uygulama için risk puanlarını ve sıralama algoritmaları kullanır. Elde edilen izin kümesine çeşitli makine öğrenme sınıflandırıcıları uygular. | Bu çalışma kapsamında statik, dinamik analiz, tersine mühendislik, makine öğrenme sınıflandırıcıları, çalışma zamanı değerlendirmesi, hibrit metodolojinin kullanımı elde edilen zafiyet tespit oranının ortaya çıkmasına olanak tanır. | [28]     |
| Framework for Mobile Devices Analysis                                                                         | Mobil Cihazlar | Önerilen çerçeve, çağrılar, multimedya ve tarayıcı geçmişi gibi kaynaklardan veri haritalamasını sağlamayı amaçlayan bir mobil araç sunar.                                                                                                                                                                                                      | Farklı kaynaklardan gelen verilerle veri haritalaması olumlu bir özelliktir.                                                                                                                                                            | [29]     |
| Hybridroid: Static analysis framework for Android hybrid                                                      | Android        | Android hibrit uygulamalar için bir statik analiz çerçevesidir. Android Java ve JavaScript'in betiklerini analiz ederek hataları ve güvenlik açıklarını tespit eder, hata bulma işlemi yapar.                                                                                                                                                   | Bu araştırmanın sınırı, yalnızca statik analiz üzerine odaklanması ve diğer tüm analizlerin kapsamamasıdır.                                                                                                                             | [30]     |
| Mobile Device Penetration Testing Framework And Platform For The Mobile Device Security Course                | Android        | Araştırma, mobil cihaz güvenliği kursu için mobil cihaz penetrasyon testi çerçevesi ve platformunu sunar. Zafiyet analizi için güvenli sandbox ortamı kullanılmıştır. Zafiyet depoları, log toplama ve destek ve mobil zafiyet analizi ve mobil uygulama penetrasyon testi yapmak için bir çerçeve içerir.                                      | Güvenli ve Sandbox ortamı sunar. Zafiyet analizi, güncel zafiyet deposu, log mesajları toplama ve mobil zafiyet analizi için bir çerçeve sunmaktadır. Statik analize odaklanmıştır.                                                     | [31]     |

İncelenen Kaynaklar; kitaplar, makaleler, konferans bildirileri ve sektörden alınan geri bildirimler sonucunda;

- Mobil uygulama güvenliği konusunda geliştiricilere yol göstermek amacıyla; OWASP Mobil Uygulama Güvenlik Doğrulama Standardı (MASVS), OWASP Mobil Güvenlik Test Rehberi (MSTG), OWASP Mobil Uygulama Güvenlik Kontrol Listesi rehberlerinin akademik çalışmalara yeterince dahil edilmediği,
- Mobil uygulama güvenlik analistlerinin, analiz sürecine katılmaları ve analiz sürecine aktif olarak etki etmelerine olanak sağlayan konfigüratif yapı eksikliği,
- Mobil uygulama sızma testlerini yapmak için ihtiyaç duyulan araçların, kurulum ve konfigürasyonları için yüksek efor harcanması gerekliliği,
- Zafiyet Temelli Skorelama yeteneğine sahip güvenlik araçlarının sayısının yeterli olmadığı veya doğruluk ve güvenilirlik açısından geliştirilebileceği

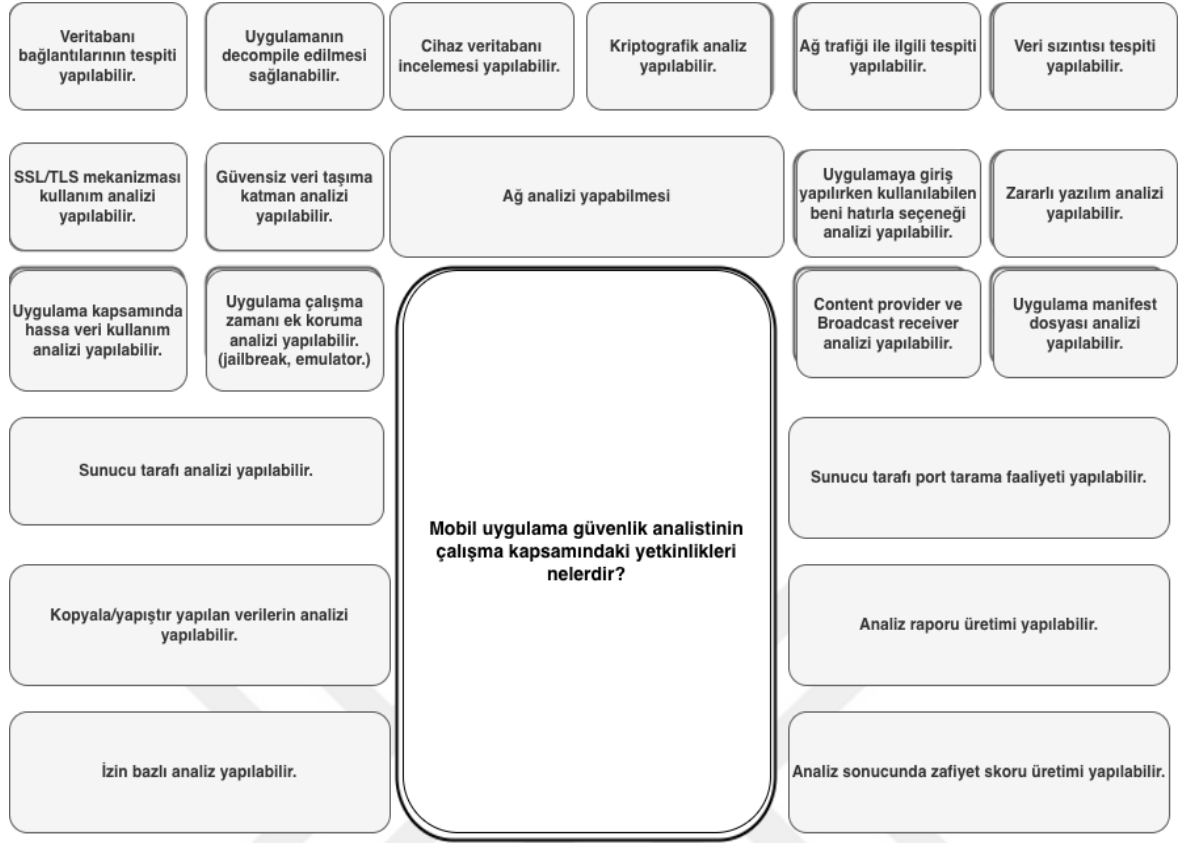
tespit edilmiştir.

### 3. MATERYAL VE YÖNTEM

Mevcut çalışma kapsamında üç temel hedef belirlenmiştir. Bu hedeflerden ilki, mobil uygulama analizinde kullanılan araçların aynı çatı altında toplanarak, bu araçları otomatize ederek aynı zamanda bu otomatize etme sürecine ek olarak geliştirilen altyapı özellikleri ile de kapsamlı bir mobil uygulama penetrasyon test aracı oluşturmaktır. MAPTF'in, temelde mobil uygulama analizi yapmak isteyen analistlerin, analiz sürecinde ihtiyaç duyacakları tüm gereksinimlerini kapsaması amaçlanmıştır. İkinci hedefimiz, analiz yapılacak uygulamanın potansiyel zayıflıklarının tespit edilmesi, zafiyet tarama faaliyetlerinin gerçekleştirilmesi ve anlamlı bir analiz raporu ortaya çıkartılmasıdır. Son ve en önemli hedefimiz ise zafiyet temelli puanlama kabiliyetidir. Bu puanlamayı yapabilmek için birçok girdiye ihtiyaç duyulmaktadır. Bunların başında farklı analiz araçlarının raporlarının girdi olarak verilmesi, uygulama APK dosyasının girdi olarak verilmesi, uygulamanın farklı araçlar ile zafiyet taramasına sokulması gelmektedir.

Mobil uygulama analizi yapmakta olan analistlere kolaylık sunmak amacımız doğrultusunda, güvenlik analistlerine tek bir uygulama aracılığıyla çok sayıda araca kolay erişim imkânı sunmak, analiz bilgilerini saklanması ve analiz edilen uygulamalardan oluşan raporları dışa aktarmaya olanak tanınması hedeflenmiştir. Ayrıca analistler, birçok analiz aracı yardımıyla taranmış uygulamaların zafiyet puanlarını oluşturabilmektedir. Mobil güvenlik testi yapan analistlerinin, MAPTF kapsamındaki yetenekleri Şekil. 3.1.'de MAPTF yetenek haritasında gösterilmektedir. Bunun yanı sıra uygulama, zafiyet puanlama algoritmalarıyla ve güvenlik analiz araçlarından elde edilen verilerin işlenmesiyle analistlere mobil uygulama davranışının anlamlandırma yardımcı olmayı amaçlamıştır.

Mevcut çalışma kapsamında geliştirilen araç için; Java 17, Spring Boot v2.6.15 kullanılarak arka uç geliştirme yapılmıştır. Analistlerin isteklerini iletecekleri ve sistemle etkileşime geçecekleri aynı zamanda raporlara ve puanlamalara erişebilecekleri arayüz geliştirmesi için React programlama dilinin 18.2.0 versiyonu kullanılarak ön uç geliştirilmesi yapılmıştır.



Şekil 3.1. MAPTF Yetenek Haritası

Bu çalışmada, mobil uygulama penetrasyon test tekniklerinde yaygın olarak kullanılan analiz araçları kullanılmıştır. Bu araçlar farklı şekillerde ve yöntemler ile mobil uygulama güvenlik analistleri tarafından girdi olarak sağlanan APK dosyalarına, test süreçlerinde uygulanmıştır. Uygulama kapsamında mobil uygulamalar için kullanılan statik analiz araçları Apktool, Qark, Androbugs ve Androguard'dır. Burp analiz aracı, mobil uygulamaların ağ trafiğini analiz etmek için kullanılmıştır. Bu araç vasıtasıyla, mobil uygulamanın ağ trafiği toplanmış ve rapor dosyası haline getirilmiştir ve analizler yapılmıştır. Sunucu tarafı port tarama faaliyetleri için NMAP ve yine sunucu tarafında tarama faaliyetleri için ZAP Proxy kullanılmıştır. Mobil uygulamanın hedef işletim sistemi sürümü ve minimum işletim sistemi sürümleri, açık kaynak bilgileri gibi diğer önemli veriler apk dosyasından elde edilmiştir. Bu tür bilgilerin eldesi Apktool analiz aracı vasıtasıyla yapılmıştır. Aşağıda uygulama dosyasından tarama, bilgi çıkarmak, raporlama ve analiz sürecinde kullandığımız araçlar verilmiştir.

### **3.1. Tarama, Bilgi Eldesi, Raporlama ve Analiz Sürecinde Kullanılan Araçlar**

Aşağıda uygulama kapsamında tarama faaliyetleri, bilgi çıkartma ve bilgi eldesi işlemleri, analiz raporlarının oluşturulması, uygulama analizlerinin yapılması ve zafiyet temelli puanlama aşamalarında kullanılan araçlar açıklanmıştır.

#### **3.1.1. Apktool**

Apktool [7], APK'ların tersine mühendisliğini yaparak kullanılabilir kaynak kod oluşturmaya olanak tanıyan açık kaynak bir araçtır. Apktool, kaynak kodlarının orijinal haline yakın bir şekilde ayrıştırılmasına olanak tanır ve sonrasında kaynak kod üzerinde yapılan değişikliklerle birlikte kaynak kodları birleştirip tekrar uygulama dosyası haline getirmeye olanak sağlar. Uygulamanın orijinal halini ayırtmak veya değiştirilmiş bir versiyonunu oluşturmak için kullanılmaktadır. Apktool, Android uygulamasının yapısını analiz etmek, Android uygulamasından yararlı bilgiler(metinler, resimler ve XML dosyaları) çıkarmak, Android uygulamasına özellik eklemek veya kaldırmak, kaynak kodunun değiştirilmesinden sonra uygulamayı yeniden oluşturmak gibi birçok görev için kullanışlıdır. Apktool, mobil güvenliği alanlarında, penetrasyon testi, statik analiz ve ters mühendislik gibi görevler için kullanılmaktadır.

#### **3.1.2. Androguard**

Androguard [32], Android uygulamalarının tersine mühendisliği ve analizi için yaygın şekilde kullanılan açık kaynak kodlu bir araçtır. Python ile yazılmıştır ve Android uygulamalarının davranış ve meta verilerinin incelenmesi ile ilgili bir dizi kütüphane, araç ve betik içermektedir. Androguard'ın sağladığı bazı özellikler şunlardır: APK dosyalarını kaynak kodunu elde etmek için ayrıştırma, APK dosyalarının byte kodunu çıkarmak için ayrıştırma, izinler, API çağrıları ve etkinlikler açısından uygulama davranışını analiz etmek ve paket adı, sürüm numarası ve izin listesi gibi meta verileri çıkarmaktır [32]. Bu özellikler, Androguard'ı mobil uygulama güvenliği ve analizi alanındaki araştırmacılar için değerli bir kaynak haline getirmektedir.

### 3.1.3. Androbugs

Androbugs [6], Android uygulamalarında güvenlik açıklarının belirlenmesine yardımcı olan Python tabanlı bir araçtır. Kod ve manifest dosyasındaki statik analiz yoluyla güvenlik açıklarını tespit etmek için kullanılır. Tespit edilen güvenlik açıklarının tutulması için veri tabanı desteği sunar. Araç, komut satırı arabirimi olarak veya bağımsız bir Java uygulaması olarak kullanılabilir ve raporlar oluşturma, meta verileri çıkarma, APK dosyalarını ayrıştırma gibi işlevleri sunar. Androbugs, mobil uygulama güvenliği alanında çalışanlar için Android uygulamalarındaki güvenlik açıklarının verimli bir şekilde belirlenmesine ve analizine olanak tanıyan bir araçtır.

### 3.1.4. Qark

Qark (Quick Android Review Kit) [5], Android uygulamalarında potansiyel güvenlik açıklarının belirlenmesine yönelik kullanılan bir analiz aracıdır. Python diliyle geliştirilmiştir. Güvenlik açıklarının bulunması sürecini otomatikleştirir ve keşfedilen bulgu hakkında raporlama yapar. Qark'ın belirleyebildiği güvenlik açıkları arasında güvensiz veri saklama, ikili koruma eksikliği ve zayıf SSL / TLS kullanımı gibi yaygın sorunlar ve bir uygulama veya organizasyona özel güvenlik açıkları bulunur. Qark, komut satırı arabirimi olarak kullanılabilen bir araçtır ve bulunan güvenlik açıkları hakkında raporlar oluşturma özelliğine sahiptir. Bu raporlar, güvenlik açıklarının giderilmesine yönelik çalışmaları bilgilendirmek içermektedir. Qark, mobil uygulama güvenliği alanında çalışan araştırmacılar ve analistler için Android uygulamalarındaki güvenlik açıklarını belirleme imkânı sunan bir araçtır.

### 3.1.5. Mobsf

Mobile Security Framework (MOBSF) [44], mobil uygulamaların güvenliğini birçok açıdan test etmek için geliştirilen açık kaynaklı bir araçtır. Hem Android hem de iOS uygulamalarını destekler. Statik ve dinamik analiz, zafiyet analizi gibi analiz yöntemlerini destekler. Farklı dosya formatları ile rapor çıktı desteği sunar. MOBSF uygulamasının Python dillerinde geliştirilmiştir. Sanal makine veya Docker sürümü ile dağıtımı mevcuttur. Web tabanlı bir kullanıcı arayüzü ve REST API desteği bulunmaktadır.

### 3.1.6. Drozer

Drozer [11], Android uygulamalarının güvenlik aşamalarını analiz etmek için kullanılan bir dinamik analiz aracıdır. Analistlere, hedeflenen uygulama ve alt yapı sistemiyle etkileşim kurma, zayıflıkları arama ve dinamik analiz yapma imkânı sunan bir etkileşim arayüzü sunar. Ayrıca, Python ile geliştirilmesi yapılmıştır. Güvenlik analiz için çalıştırılmak istenen komutlar, komut satırı arayüzü vasıtasıyla ve analiz türüne göre değişiklik gösteren parametreler ile analiz aracına girdi olarak sağlanmaktadır. Drozer üzerinde belirli görevler için özelleşmiş modüller bulunmaktadır. Örneğin uygulama kapsamında kullanılan ve kullanıcılardan istenilen izinleri listeleme, suiistimal etmeye açık yayın alıcılarını belirleme, manifest dosyası analizi, uygulama intentleri ile olan etkileşimler gibi. Drozer, Android uygulamalarını ve işletim sistemini analiz etmek ve etkileşim kurmak için bir dizi yöntem sunması nedeniyle mobil uygulama güvenliği analistlerinin ve araştırmacılarının farklı analiz aşamalarında kullandığı bir araçtır.

### 3.1.7. Burp Suite

Burp Suite [13], web trafiğini ve uygulamaların ağ üzerinden yaptığı iletişimi izlemek aynı zamanda yönetmek etmek için yaygın olarak bir araçtır. HTTP veya HTTPS trafiğini keserek araya girme, bulandırma ve otomatik ağ testleri yapma gibi görevler için çeşitli araçlar içermektedir. Burp Suite, Java programlama dili ile geliştirilmiştir. Geniş özellik seti ve esnekliği nedeniyle güvenlik analistleri ve araştırmacılar tarafından kullanılmaktadır. Burp Suite, web trafiğini ve uygulamaların ağ üzerinden yaptığı iletişimi tanımlamaya ve çözüm bulmaya yardımcı olması nedeniyle yaygın olarak kullanılan bir araçtır.

### 3.1.8. Nmap

Nmap [33], bir ağdaki ana bilgisayarların ve hizmetlerin keşfi ve durumlarının belirlenmesi için kullanılan bir ağ güvenlik aracıdır. C dillerinde geliştirilmiş olan Nmap ile, komut satırı aracı veya kullanıcı arayüzü vasıtasıyla etkileşime geçilebilir. Ağ yöneticileri, uygulama analistleri ve güvenlik uzmanları tarafından geniş çapta kullanılan Nmap, ağ haritalama, envanter ve güvenlik denetimi faaliyetlerinde kullanılır. Ayrıca, büyük ağların hızlı bir şekilde taraması ve zafiyet oluşturabilecek portların taraması için

kullanılabilir. Ağdaki diğer bilgisayarların keşfi ve hizmet tespiti için özelleştirme seçenekleri sunmaktadır. Nmap, ağdaki ana bilgisayarların işletim sistemi ve sürümünü tanımlayabilir ve güvenlik önlemleri ile diğer güvenlik tedbirlerinin varlığını tespit edebilir [33]. Bu sebeple Nmap, ağ güvenliği, port taraması ve ağ yönetimi analiz süreçlerinin birçok aşamasında kullanılan önemli bir araçtır.

### 3.1.9. Owasp Zed Attack Proxy

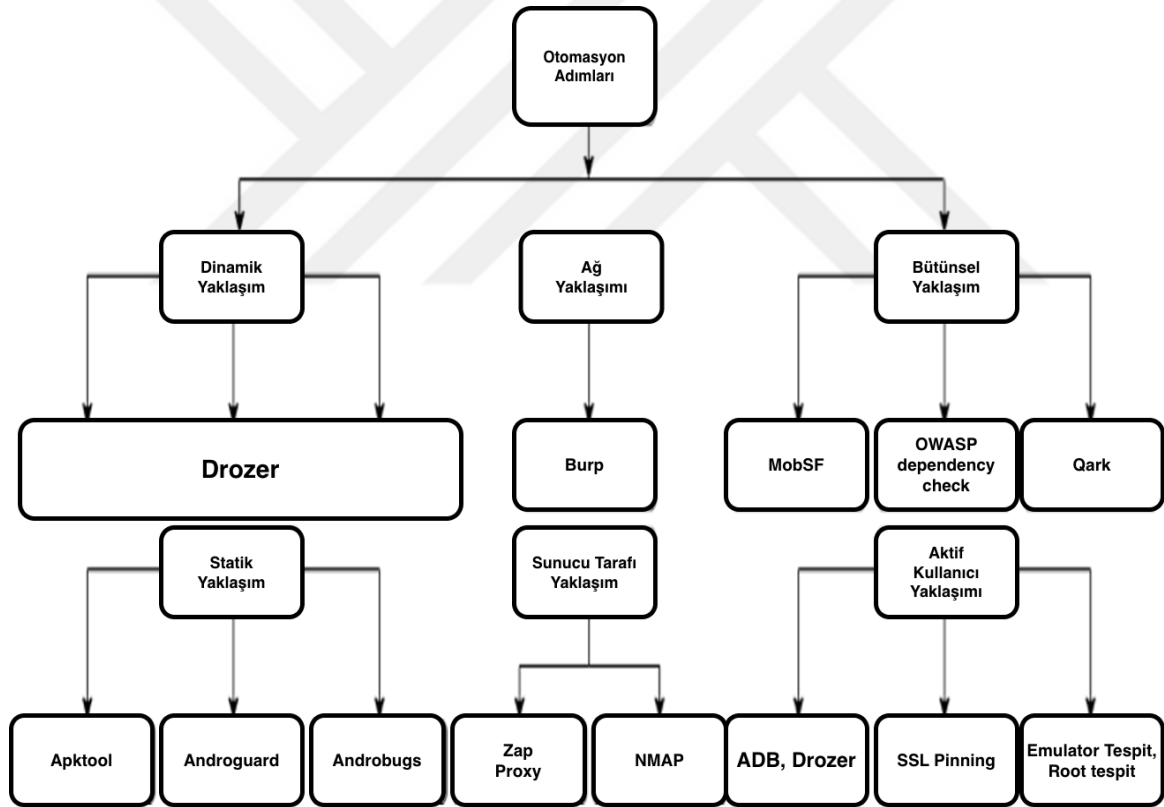
OWASP ZAP (Zed Attack Proxy) [34], güvenlik profesyonellerinin, geliştiricilerin ve uygulama analistlerinin ağ trafiğine giren uygulamalardaki zayıflıkları belirlemelerine, taramalarına ve çözmelerine yardımcı olan bir güvenlik aracıdır. ZAP aracının yansıma vekil sunucusu olarak kullanıldığı senaryolar yaygındır. Kullanıcının ağ üzerinden gönderilen istekleri ve yanıtları değiştirmesine olanak sağlar ve isteklerin taramalarını gerçekleştirir. ZAP, aktif ve pasif tarama, bulandırma ve web soket desteği gibi bir dizi yeteneğe sahiptir [34]. Ayrıca, API arayüzü sunmaktadır. Bu API arayüzleri sayesinde test ortamlarına entegre edilebilmektedir. Genel olarak ZAP, web ve mobil uygulamalarındaki güvenlik risklerini belirleme ve azaltma için kullanılan bir ağ aracıdır.

### 3.1.10. Virustotal

VirusTotal [35], dosyaları URL'leri, virüsleri ve diğer kötü amaçlı yazılımları taramak için kullanıcılara olanak sağlayan bir web sitesi ve mobil uygulamadır. VirusTotal mobil uygulaması, cihazda yüklü olan uygulamaların güvenliğini ve kötü amaçlı davranışlarını tarama ve analiz etme özelliğini içermektedir. Bu özellik, bireysel ve kurumsal düzeyde mobil cihazların güvenliğini sağlamaya ve kötü amaçlı yazılımlardan korunmaya yardımcı olmaktadır. Tarama ve analiz yeteneğine ek olarak, VirusTotal, bir veri tabanı olarak bilinen kötü amaçlı yazılımlar ve zayıflıklar hakkında bilgi içermekte ve bu tehditleri paylaşma ve tartışma için bir topluluk platformu da sunmaktadır [35]. Genel olarak, VirusTotal, analistler ve araştırmacılar için bir kaynak olarak hedeflediği cihazların güvenli hale getirilmesine ve kötü amaçlı yazılımlardan korunmaya yönelik bir amaç taşımaktadır.

Bu çalışmanın amaçlarından birisi giriş bölümünde açıklandığı üzere analistlere, uygulama analizinde önceden belirlenen güvenlik araçları kullanarak bir dosya yükleme ve analiz

etme yetkinliğine sahip mobil uygulama analizi çerçevesini sunmaktır. Bu kapsamda, yüklenen dosya, seçilen analiz araçlarıyla bir analiz süreç çerçevesinden geçerek zafiyet ve analiz raporları üretir. MAPTF, analistlere, seçilen güvenlik analiz araçları kapsamında kullanabilecekleri analiz yöntemlerini sunar. Her analiz aracının, her özelliği geliştirilen çerçeve kapsamındaki kısıtlardan dolayı kullanılamayabilir. Örneğin, dinamik analiz aşamasında, güvenlik analiz araçları, uygulamanın yüklendiği cihaza erişim talebinde bulunmakta veya cihaz için kök yetkilerini istemektedir, bu aşamada çerçeve, analistlere uyarılara vererek veya sadece kullanabilecekleri özellikleri arka planda aktif ederek analizleri gerçekleştirmektedir. Farklı türde analizlere göre geliştirilen çerçevesinin etki alanındaki alt analiz yöntemleri ve bu analiz yöntemleri kapsamında kullanılabilecek araçlar Şekil 3.2’de gösterilmiştir.



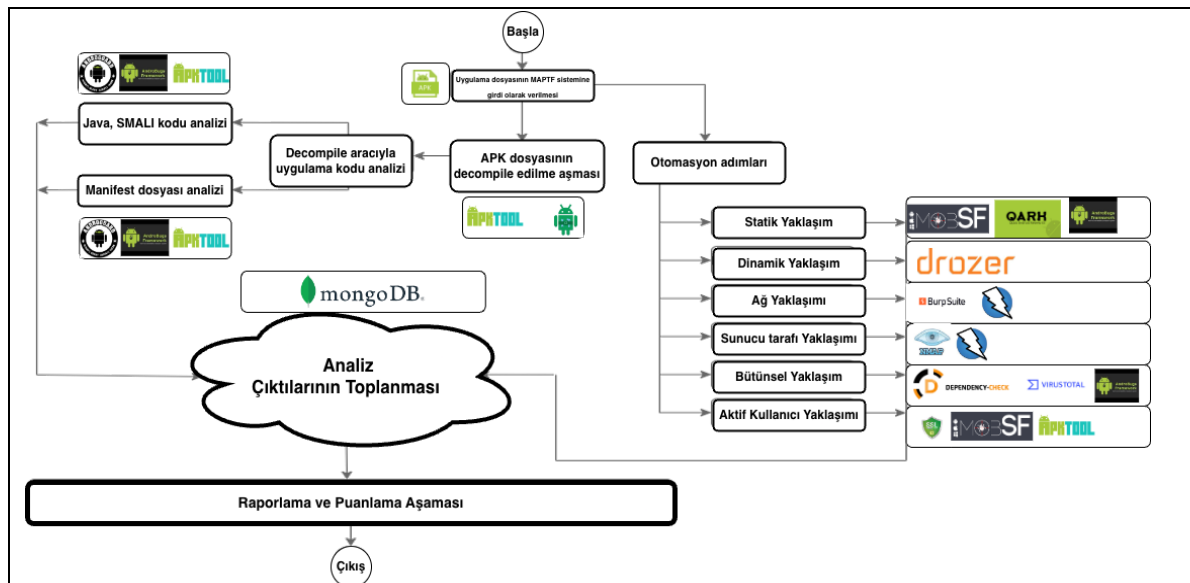
Şekil 3.2. MAPTF analiz kapsamında alt analiz yöntemleri ve araçları

Mobil uygulama güvenliği analizinde puanlama aşaması, önceden konfigürasyon dosyası aracılığıyla belirlenmiş ve güvenlik analistleri tarafından genişletebilen kurallara göre değerlendirilmesi ve yorumlanması sürecidir. Modelin çerçevesinde önemli olarak belirlenen anahtar alanlar, puanlama sürecine konfigürasyon dosyaları aracılığıyla sisteme

girdi olarak alınır. Örneğin, Android uygulamaları için kimlik doğrulama atlamalarına ilişkin açıklar, uygulamanın manifest dosyasının ayrıntılı incelenmesi, uygulamanın kaynak kodunun karıştırılması, hata ayıklama seçenekleri, kaynak kod içerisinde açık olarak yazılan şifrelerin bulunması, zararlı kabuk komutları, URL tarama ve zafiyet tespit süreçleri gibi anahtar alanlar bulunmaktadır. Bu anahtar alanların her birinin sonuçları, uygulamanın güvenliği hakkında bilgi vermek üzere puanlama algoritması tarafından konfigürasyon dosyasında belirtilen, ilgili kuralın karşılık geldiği CVSS puanına göre oluşturulur. Bu puanlar, dinamik, statik, ağ, otomatik analiz ve sunucu tarafı yöntemlerinin birleşimini kullanarak toplanır ve 100 puan üzerinden zafiyet puanı elde edilecek şekilde oranlanır. Ayrıca, Mobil Uygulama Sızma Testi Çerçevesi içindeki puanlama algoritması, temel olarak MASVS ve MSTG gibi OWASP standartlarını tanımlı kurallar olarak kapsamakta. Aynı zamanda uygulama güvenlik analisti tarafından farklı kurallar eklenerek esnek bir yapı oluşturulması amacıyla tasarlanmıştır.

### 3.2. Tarama, Bilgi Eldesi, Raporlama ve Analiz Sürecinde Kullanılan Yöntemler

Önerilen araç için geliştirilen web arayüzü, mobil uygulamaların kolay ve verimli bir şekilde analizini mümkün kılmakta ve güvenlik analistleri görsel ve kullanıcı dostu bir yaklaşım ortaya koymaktadır. Şekil 3.3'te önerilen araç için analiz adımları ve süreç akışı gösterilmiştir.



Şekil 3.3. MAPTF analiz adımları ve süreç akışı

Çalışma kapsamında, aşağıdaki başlıklar altında ele alınan yöntemler, mobil uygulama analizi türleri, analiz yöntemleri ve standartları, ayrıca OWASP mobil uygulama analizi süreçleri göz önünde bulundurularak detaylı olarak açıklanmıştır.

### 3.2.1. Statik analiz

Araştırma kapsamında, analist web arayüzü üzerinden statik analiz başlığı altında seçtiği analiz aracı ile analizini başlatır. Analist, uygulamayı hangi araçla analiz etmek istediğini belirler ve uygulamanın dosya yolunu ile seçilen araç tarafından oluşturulacak raporun kaydedileceği dosya yolunu seçer. Analiz işlemi, paralel olarak arka planda devam eder ve analiz tamamlandığında, oluşan rapor belirtilen dosya yoluna kaydedilir. Analist, analiz aracı olarak Qark, Apktool, Androbugs veya Androguard araçlarından birini seçebilir. Eğer Apktool seçilirse, mobil uygulama dosyasının çözülme işlemi yapılır. Diğer analiz araçları seçildiğinde, framework belirtilen analiz aracı ve kullanıcı tarafından girdi olarak verilen APK'yı seçilen analiz yöntemi ile arka planda analiz eder. Analiz sonucu, belirtilen dosya yoluna kaydedilen bir rapor şeklinde sunulur. Analist, bu raporu inceleyebilir veya tarama sonuçlarını diğer güvenlik analizi süreçlerinde kullanabilir. Ayrıca, bu analiz sonucu MAPTF uygulamasına girdi olarak sağlanabilir ve zafiyet puanının oluşturulma işleminde kullanılabilir. Analiz sonucu, MongoDB veri tabanına kaydedilir ve kullanıcıya ilgili uygulamayla ilgili önceki analizlere göre bilgi sağlar.

### 3.2.2. Dinamik analiz

Mobil Uygulama Sızma Testi Çerçevesi kapsamında, dinamik analiz süreçleri için belirlenen analiz araçları, dinamik analiz yöntemi olarak kullanıcıya seçenek olarak sunulmuştur. Dinamik analiz aracının seçimi ve analiz sürecinin başlatılması işlemi web arayüzü üzerinden yapılmaktadır. Analist, bu bölümde analiz etmek istediği uygulamanın hangi araçla analiz edileceğini seçer. Analist, analiz etmek istediği uygulamanın dosya yolunu ve seçilen araç tarafından oluşturulacak raporun kaydedileceği dosya yolunu seçer ve analiz işlemini başlatır. Analiz işleminde MOBSF aracı mobil Uygulama Sızma Testi çerçevesi kapsamında halihazırda desteklemektedir. MOBSF aracılığıyla analiz başlatıldığında, MOBSF mobil uygulama güvenlik analiz raporu araştırmacı tarafından belirtilen dosya yolunda otomatik olarak oluşturulacaktır. Analiz işlemi MAPTF web arayüzü üzerinden veya REST API aracılığıyla tetiklenebilmektedir.

### 3.2.3. Ağ analizi

Uygulama kapsamında, araştırmacıya Burp ağ analizi aracı ile analiz yapma imkânı sunulmuştur. Bu seçeneklere web arayüzü üzerinden veya REST API üzerinden erişilebilmektedir. Araştırmacı, istediği uygulamanın analizini hangi ağ analizi aracıyla yapmak istediğini seçebilir. Burp seçildiğinde, analistin ekranına komut satırı açılacak ve Burp uygulaması (daha önce yüklenmediyse "requirement.sh" dosyası aracılığıyla yüklenmelidir.) aracılığıyla başlatılacaktır. Araştırmacı daha sonra Burp kullanarak analizini gerçekleştirecektir. Bu ağ analizi sonucunda yapılan işlemler ve çıktılar tarama raporu olarak html formatında dışarıya aktarılacaktır. Elde edilen bu analiz süreç dosyası, puanlama arayüzü üzerinden bir uygulama için tetikleneceği zaman girdi olarak verilebilmektedir.

### 3.2.4. Sunucu tarafı analizi

Mobil uygulamanın sunucu tarafındaki bileşenlerinin değerlendirilmesi, mobil uygulama sunucu taraflı analiz olarak adlandırılır. Bu süreç, saldırılara açık olabilecek zayıflıkları veya açıkları tespit etmek için arka uç altyapısının ve sunucu tarafında bulunan API'lerin incelenmesi gibi unsurları içerebilir. Bu tür bir analizin ana amaçları, hassas verilerin korunmasını ve mobil uygulamanın optimal performansını garanti etmektir. Bu çalışmada, amacımız, port eşleme tarama yoluyla sunucu taraflı analiz yapmaktır. Girdi olarak, analist tarafından sağlanan uygulamanın IP adresini kullanmaktadır. Nmap aracını kullanarak bu IP adreslerinin port taramasını yapılmaktadır. TCP veya UDP açık port parametreleri ile ve aynı zamanda açık portların toplam sayısını tarayıp, bu tarama sonuçlarını analistlere rapor olarak belirledikleri analiz raporu dosya yolu üzerinden arayüz aracılığıyla sunulmaktadır. Daha önce analiz edilen mobil uygulamanın içerdiği IP adresleri de tablo biçiminde arayüz aracılığıyla araştırmacıya sunulmuştur. NMAP dışında OWASP ZAPProxy ile sunucu tarafı analiz işlemi yapılabilmektedir. Araştırmacının arayüz veya API'leri ile bu araç ile analiz yöntemini tetiklediğinde analistin komut satırı açılarak daha öncesinden sisteme yükleme OWASP ZAPProxy uygulaması açılmaktadır. Analist bu araç üzerinden mobil uygulama ile ilgili taramalarını yapabilmektedir. Gelecekteki çalışmalarda ZAPProxy analiz sonuçları girdi parametre olarak işlenecek ve bir rapor çıktısı sunulması amaçlanmaktadır.

### 3.2.5. Zafiyet taraması

Mobil uygulama güvenlik analizinin önemli bir ilk adımı olan zafiyet tarama, mevcut uygulama dosyasının bilinen zararlıları içerip içermediğini kontrol etmek için yapılan bir süreçtir. Eğer girdi olarak verilen mobil uygulama, virüs tarama yazılımı tarafından zararlı olarak tespit edilirse veya daha önce yapılan taramalarda eşleşen bir tekil imzasına sahip başka bir uygulama bulunursa, bu uygulama zararlı olarak adlandırılır. Bu kapsamda projede, analistler tarafından girdi olarak sağlanan APK dosyasına, VirusTotal sistemi aracılığıyla analiz ve tarama işlemi yapılmıştır. VirusTotal API arayüzleri APK dosyasının kaynak kodunun ikili formunu sorgulamak için kullanılmıştır. Gereken API\_KEY bilgisi, VirusTotal API belgelerinden referans alınarak kullanılmıştır. Bu analizin sonuçları puanlama aşamasında da kullanılmış ve analistlere rapor formunda sunulmuştur.

### 3.2.6. Zafiyet taraması yorumlarının sunulması

Tarama işlemleri sonucunda, VirusTotal sisteminde yapılan tarama faaliyeti ve bu işlem sonucunda kaydedilen uygulama ile ilgili araştırmacılar yorumlar girerek kendi bilgilerini sistem ile paylaşabilmektedir. Yorumlar aracılığıyla tespit edilen zafiyetle ilgili daha ayrıntılı bilgiler ve daha kapsamlı kullanım senaryolarıyla ilgili paylaşılan bilgiler incelenebilir. Bu yorumlar taraması yapılan uygulamanın elde edilen zafiyet taramaları dışında, daha ayrıntılı bir yorum veya görüş içermesi açısından önemlidir. Bu yorumlar MAPTF uygulama çerçevesi tarafından oluşturulan rapora dahil edilebilir veya uygulamanın zafiyet puanının hesaplanmasında kullanılabilir. Uygulama ve ilişkili verileri ayrıntılı bir şekilde inceleyerek, tanımlanan riskleri azaltmak ve uygulamanın genel güvenliği durumunu iyileştirmek ile ilgili bilgi sahibi olunmaktadır.

### 3.2.7. Anahtar kelime taraması

Mobil Uygulama Sızma Testi Çerçevesi kapsamında, güvenlik analistlerine kendi kural setleri oluşturma ve MAPTF tarafından kullanılan zafiyet puanlama mekanizmasını etkilemesine olanak tanır. Bu metodoloji, güvenlik analisti tarafından yazılan kuralları içeren bir YAML dosyasının kullanılması ile gerçekleştirilir ve uygulama çerçevesinin kaynak kodundan türetilir. Bu kurallar arasında, yaygın olarak kullanılan metotlardan birisi, verilen anahtar kelime ve kelime gruplarının uygulamanın içerisinde kullanılıp

kullanılmadığını ile ilgili yapılan taramalardır. Bu tarama süreci şu şekilde çalışmaktadır, eğer analist tarafından uygulama kapsamında kullanılıp kullanılmadığının kontrolü için belirtilen anahtar kelime, MAPTF kapsamında mobil uygulama güvenlik araçlarının ürettiği raporlar içerisinde bulunması durumuna göre puanlama yapılmaktadır. Bu arama işlemi REGEX şeklinde tanımlanan potansiyel sızıntıları, uygulama APK dosyasından ayrıştırılan kaynak kodlar içerisinde arama yöntemine dayanmaktadır. Arama şartının sağlanması durumunda kural tanımında belirtilen CVSS puanı, zafiyet puanına etki için kullanılmıştır.

### **3.2.8. Kaynak kodlarına ayrıştırma aşaması**

Bir mobil uygulamanın APK dosyasından kaynak kodunu çıkarmaya yönelik işleyen bir süreçtir. APK dosyalarını açık kaynak kodlu bir şekilde çözmek ve kaynak kodunu görüntülemek için, ücretsiz ve açık kaynak bir araç olan Apktool kullanılmıştır. APK dosyalarını açmak, uygulamanın nasıl çalıştığını anlamaya ya da güvenlik açıklarını belirlemeye yardımcı olabilecek bir süreçtir. Çalışma kapsamında, APK dosyalarını açmak, manifest dosyasından, orijinal kaynak kodlarından, SMALI dosyalarından veya kaynak dosyalarından bilgi çıkarmayı amaçladık. Bu özelliği kullanarak, anahtar kelime tarama, puanlama özellikleri ve manifest yorumlama gibi işlemler yapılmaktadır.

### **3.2.9. Analiz raporunun oluşturulması**

Araştırmacılar, Mobil Uygulama Sızma Testi Çerçevesi ile mobil uygulama analizi için kullanılan araçların yetkinliklerine sahip olduklarından, o araçların ürettiği raporlara da erişebilmektedirler. MAPTF uygulama çerçevesi aracılığıyla web arayüzünü kullanarak analizlerini başlattıklarında, analiz raporunun kaydedileceği dosya yolunu da belirtirler. Seçilen analiz tipi için rapor belirtilen dosya yoluna kaydedilir ve araştırmacı tarafından incelenebilir. Bu özellik sayesinde, araştırmacılar kullanıcı dostu web arayüzü aracılığıyla çeşitli analiz araçlarını kolayca kullanabilir, rapor üretme komutlarını bilmeye gerek kalmadan ve daha önce yapmış oldukları analizlere de kolayca erişebilirler.

### 3.2.10. Veri tabanı desteği

MAPTF uygulama çerçevesi ile kullanıcılar, belirtilen analiz tipleri için raporlar oluşturabilir ve bu raporları web arayüzünden veya analiz ortamlarına indirerek görüntüleyebilirler. Web arayüzü ile uygulama raporunun ne zaman oluşturulduğu hakkında bilgi sağlanmakta aynı zamanda mobil uygulama güvenlik analisti için takip edilebilirlik ve izlenebilirlik sağlamaktadır. Bu nedenle, uygulama çerçevesi varsayılan olarak veri tabanı desteği sunmaktadır. Veri tabanı olarak MongoDB kullanılmıştır. MongoDB belge tabanlı bir veri tabanıdır. Hedefimiz veri tabanı vasıtasıyla raporların saklanarak, araştırmacılar tarafından oluşturulan raporların takip edilebilirliğini sağlamaktır. Ayrıca, MAPTF uygulama güvenlik puanı web arayüzü aracılığıyla oluşturulduğunda, puan, analiz tarihi, kaç puan aldığı ve analiz için hangi analiz raporlarının kullanıldığı görüntülenebilmektedir.

### 3.2.11. Kullanıcı arayüz desteği

Mobil uygulama güvenlik analistleri , MAPTF uygulamasını, web arayüzüne üzerinden giriş yaparak, güvenlik analiz, başlatma sayfaları, analiz sonuçlarının görüntüleme sayfaları, uygulama zafiyet raporlama ve zafiyet puanlama sayfaları aracılığıyla kullanabilirler. Web arayüzü ve REST uç noktaları önüne koyulan JWT mekanizması ile sistem, araştırmacıların sisteme kaydolmadan kullanmasının önüne geçmiştir. Sistemde rol-tabanlı yetkilendirme mekanizması kullanılmıştır. Araştırmacılar, sisteme arayüz vasıtasıyla veya REST uç noktalarıyla kaydolarak giriş yapabilir. Yetkilendirme ve erişim işlemleri için JWT jeton mekanizması kullanılmıştır. Gelen her talebin, periyodik olarak üretilen JWT jetonları kullanılarak doğrulanması ile işlenmesi amaçlanmıştır. Uygulama içinde, mobil uygulama güvenliği analizlerinin yapılabileceği ve bu analizlerin takip edilebileceği bir panel geliştirilmiştir. Her tür analiz kendi paneline sahiptir ve bu panel üzerinden analizler başlatılabilir ve takip edilebilir. Ayrıca, çalışan sistemdeki kullanıcı yönetimi, uygulama metriklerinin izlenmesi, uygulama sağlık durumu bilgileri, uygulama yapılandırma bilgileri, uygulama günlük yapılandırmaları ve uygulama API'leri için arayüzleri sunan ekranlar bulunmaktadır.

### 3.2.12. REST API dokümantasyonu

API uç noktaları, uygulama kapsamında geliştirilen analiz özellikleri ve bu özelliklerle entegrasyon aktiviteleri için geliştirilmiştir. Veriler bu API uç noktaları aracılığıyla diğer sistemlerden, MAPTF sistemine veya MAPTF sisteminden başka bir sisteme aktarılabilirler. API uç noktaları, dokümantasyon ve entegrasyonu kolaylaştırmak için, OpenAPI özelleştirmesi olan Swagger kullanılarak belgelendirilmiştir.

### 3.2.13. Zafiyet temelli puanlama

MAPTF uygulama çerçevesi puanlama kural seti, MSTG [36], MASVS [37] ve OWASP Mobil Uygulama Güvenliği Kontrol Listesi [38] standartlarına dayalı oluşturulan temel kurallar, ayrıca mobil uygulama analistlerinin temel kurallara, konfigürasyon dosyaları vasıtasıyla ekleyecekleri özel kurallar ve diğer analiz araçlarından elde edilen tarama verilerinin beslenmesi ile oluşturulan bir puanlama mekanizması oluşturulmuştur. Bu puanlama mekanizmasında girdiler, uygulamanın kendisi, uygulama dosyalarının ayrıştırılması ile elde edilmiş kaynak kod dosyaları, farklı mobil uygulama güvenlik analiz araçlarından elde edilen raporlar ve zayıflık tarama araçlarından elde edilen sonuçlarıdır. Zafiyet temelli puanlamanın çıktısı; CWE öğelerine karşılık gelen CVSS puanlarının, varsayılan kurallar ve analist tarafından eklenen kuralların toplamı ve MAPTF uygulama çerçevesinin analistin isteğine göre 100 puan üzerinden oranlanması sonucu oluşan puandır. Elde edilen puan, birçok analiz araçlarından elde edildiğinden güvenilir ve OWASP tabanlı olması ve birçok güvenlik aracının ortak kesişim kümesi olması sebebiyle de gerçekçidir.

### 3.2.14. Puanlama kurallarının özelleştirilmesi

MAPTF uygulama çerçevesi kapsamında, MASVS, MSTG ve OWASP Mobil Uygulama Güvenliği Kontrol Listesi standartlarına dayalı bir kural kümesi oluşturduk. Bu kural kümesi, mobil uygulamaların olası ortak özelliklerini dikkate alarak oluşturulmuştur. Bu kural kümesi ile sisteme yüklenen mobil uygulama dosyası üzerinde analiz faaliyetleri yapılmaktadır. Aynı zamanda bu analiz faaliyetleri, analizi yapan analist tarafından farklı özelliklerle genişletilebilir veya daraltılabilir. Bu kural kümesi tamamen boşaltılabilir ve analiz faaliyeti sadece analistin kurallarıyla yapılabilir. Veya analist mevcut kuralların

ihtiyacını tam olarak karşılamadığını düşünürse, puanlama mekanizmasını ve puan ağırlığını değiştirmek için yeni kurallar ekleyebilir.

### 3.2.15. Analiz otomasyonu

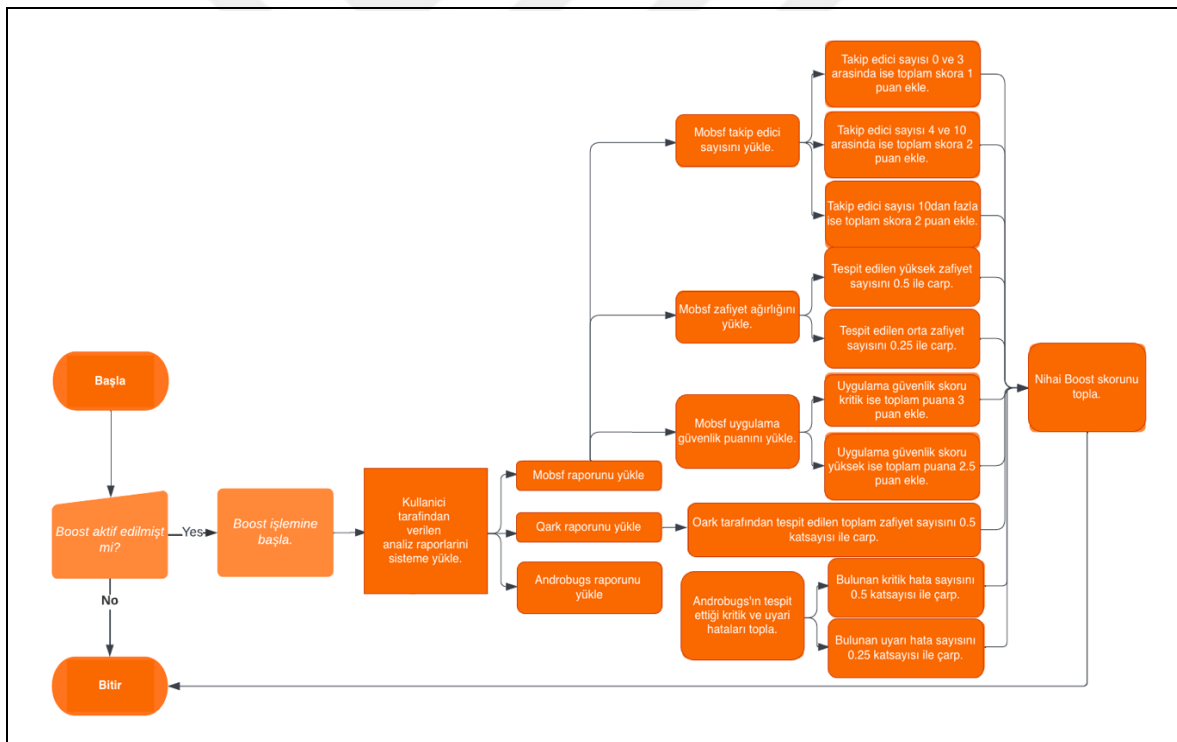
MAPTF uygulama çerçevesi, bir tek arayüz üzerinden, önceki maddelerde tanımlanan çeşitli analiz tür ve metotların analizini otomatik hale getirebilmektedir. Analiz edilecek uygulamanın dosya yolu ve araştırmacı tarafından sağlanan IP bilgileri, raporların oluşturulacağı dizin ile alınır. Bu bilgi uygulama çerçevesine analiz otomasyon yöntemine girdi olarak verilir ve gerekli tüm analiz araçları otomatik çalıştırılır. Analiz araçları adıyla oluşturulan bir dizin belirtilen dizinde açılır ve bu araçlardan elde edilen rapor çıktıları bu dizinde kaydedilir. Bu işlem arka planda paralel olarak gerçekleşir. Bu kapsamda, penetrasyon test analistlerine, SQL Enjeksiyon gibi belirli saldırı senaryolarını otomatik hale getirebilir, uygulamanın şifre veya diğer hassas verilerin saklanması, şifreli bilginin saklanması yapıldığı kod parçalarının incelenmesini, şifreleme algoritmalarının güncel olup olmadığının doğrulanmasını, kabuk komutlarının yürütülmesine olanak verilip verilmediğinin değerlendirilmesini, zararlı yazılım içerip içermediği tespitini, kaynak kodunun analizini ve root erişiminin varlığını kontrol etmek gibi özelleştirilmiş saldırı senaryolarını otomatik hale getirme süreçlerini sağlayan bir araç oluşturmaktadır. Ayrıca, çalışma kapsamında, analiz raporlarının cihazda saklanması sağlanmaktadır.

MAPTF uygulama çerçevesinin kullanımını kolaylaştırmak için, analistlerin geliştirme ortamında gerekli araçları ve kütüphaneleri bireysel olarak bulup yüklemelerine ihtiyacını ortadan kaldırmak için bir betik oluşturulmuştur. Bu betik çalıştırıldığında, analiz süreci için gerekli olan geliştirme kütüphaneleri, mobil uygulama güvenlik araçları ve kullanıcı arayüzü uygulamaları otomatik olarak yüklenir ve mobil uygulama güvenlik analistleri için her aracı tek tek edinmeden analiz sürecine hazır hale getirilmiştir. Bu aşama tamamlandıktan sonra, güvenlik analistleri geliştirme ortamındaki analiz uygulamasını çalıştırabilir ve web arayüzü veya REST uç noktaları üzerinden analiz yapabilir. Ayrıca, puanlama aşamasında, analistler mobil uygulama güvenlik analiz çerçevesi tarafından üretilen analiz raporlarını MAPTF uygulama göndererek zafiyet temelli puanlama algoritması ile puanlama yapmasını ve güvenilir sonuçlar elde etmelerini sağlamaktadır.

MAPTF zafiyet temelli puanlama mekanizmasının çalışma düzeni şu şekilde ilerlemektedir. İlk olarak, güvenlik analisti, kullanıcı bilgileriyle sisteme giriş yapar veya REST uç noktası aracılığıyla JWT jetonu elde eder. Daha sonra, güvenlik analisti, sağ üst menüden puanlama arayüzüne girer ve analiz etmek istediği APK dosyasının dosya yolunu ve analiz araçlarının çıktılarını içeren analiz raporlarının dosya yollarını belirler. Bu bilgi zafiyet temelli puanlama aşamasında girdi olarak kullanılacaklar.

Örnek olarak seçilen ‘DVBA’ uygulamasının puanlama aşamaları ve aldığı puanlar şu şekilde gerçekleşmiştir. MAPTF tarafından başlangıçta belirlenen kurallar veya varsa mobil uygulama analisti tarafından belirlenen kurallar alınarak her puan için maksimum CVSS puanı olan 10 puan alınıp toplam maksimum puan elde edilmiştir. Bu puan son aşamada 100 puan üzerinden ölçeklenecektir, seçilen uygulama için tanımlı 45 kural ile bu puan 450 olarak hesaplanmıştır. Daha sonra her bir konfigürasyon dosyasında tanımlı kural sisteme girdi olarak sokulma aşamasına geçilmiştir. Bu aşamada konfigürasyon dosyasında, toplam ‘puan etkisi’ maddesi pozitif olarak belirlenen maddeler ile eğer karşılaşırsa uygulama puanına negatif etki sağlamıştır. Eğer belirtilen kuralın ‘puan etkisi’ ‘NEGATIF’ veya ‘NEGATIVE\_POSITIVE’ ise toplam puana pozitif etki etmesi, eğer ‘NEUTRAL’ ise puana etki etmemesi sağlanmıştır. Bu örnekte ‘DVBA’ uygulaması için kurallar taramaya sokulduğunda uygulama buradan 111,89 puan almıştır. Bu puan çalışma kapsamında belirlenen ağırlık ölçekleme yöntemine girdi olarak sağlanmıştır. Bu aşamada elde edilen 111,89 puan son puana 10,66 etki etmesi sağlanmıştır. Daha sonra puan güçlendirme (Boosting) aşamasına geçilmiştir. Burada uygulama için sağlanan diğer analiz araçlarının raporları girdi olarak alınarak kombine bir puanlama yapılması amaçlanmıştır. Bu uygulama, Androbugs analiz raporu ile beslenen puan güçlendirici(Booster) aşamasından 1,25 puan, MOBSF analiz raporu ile beslenen puan booster aşamasından 3,0 puan ve Qark analiz raporu ile beslenen puan booster aşamasından 4,5 puan olmak üzere toplam 8,75 puan almıştır. Daha sonrasında uygulama zafiyet taraması için VirusTotal taramasına sokulmuştur. Bu tarama sonucunda uygulamanın ‘Android.W32.Bray.n’ zafiyetine sahip olduğu tespit edilmiştir. Bu aşamada zafiyet puanına, daha önce tanımlı bir tehdit içerdiği için 20,0 puan ve bu zafiyetin yaygınlığına göre yani güvenlik otoriteleri tarafından tespit edilme sayısına göre ölçekleme yaparak puan ekleme işlemi uygulanmıştır. Bu aşamada uygulama 26,5 puan almıştır. Elde edilen puanlar toplandığında ‘DVBA’ uygulamasının zafiyet puanı 100 üzerinden 45,91 olarak hesaplanmıştır.

Ayrıca, analist, MAPTF uygulama çerçevesi içinde tanımlanmış puanlama kurallarının olanların dışında ek puanlama kriterleri ekleyebilme yeteneğine sahiptir. Bu yetenek, uygulamadaki puan güçlendirme olarak adlandırılır. Analist bu özelliği aktif ederse, belirli yönlerden uygulamaya ek analizler yapılacak ve sonuçlar zafiyet puana pozitif olarak etki edecektir. Puan güçlendirme özelliği ile etkinleştirilecek ek analiz yöntemleri şunlardır: Eğer uygulama zararlı yazılım içeriyorsa, zayıflık olarak algılanmasını sağlayan araçların sayısı sayılacak ve bu sayıya dayalı bir ağırlık skoru hesaplanacak ve uygulamanın zafiyet puanına eklenecektir; başka bir özellik, MOBSF güvenlik analiz aracının çıktısını belirli bir ağırlık ile zafiyet puanına ekleyerek uygulamanın güvenilirliğini daha da artırmaktır; başka bir özellik, Androbugs aracı tarafından belirlenen kritik zafiyet ve uyarıların sayısını belirli bir ağırlık skoru ile çarpıp ve zafiyet puanına eklemektir. Ayrıca, Puan güçlendirme mekanizması akış şeması Şekil 3.4'te görülebilir.



Şekil 3.4. Puan güçlendirme mekanizması

MAPTF kapsamında, OWASP tarafından belirlenen MASVS, MSTG ve OWASP Mobil Uygulama Güvenliği Kontrol Listesi ile uyumlu olmasını ve bu uyumluluk ile ilgili maddelerle kontrol edilmesini sağlamak için MAPTF Kapsam Matrisi oluşturduk. Bu matris temel olarak 45 güvenlik kuralından oluşmaktadır. Bu matris yardımıyla, MAPTF kapsamında zayıflık olarak tanımlanmış maddeleri MASVS ile ilgili maddeleriyle eşledik.

Ayrıca, bu madde için analiz araçlarını, puanlama etkilerini, etki puanlarını, etki puanlarının hesaplanma yöntemini, bu madde için karşılık gelen CWE [39] maddesini, bu CWE maddesi için erişilebilecek bağlantıyı, bu CWE maddesi için ortaya çıkan zayıflıklar arasındaki en yüksek CVSS puanını ve zayıflığın içeriğini kontrol etmek için kullanılacak anahtar kelime listesini belirledik. Bu matrisi YAML dosya formatında girdi parametresi olarak sağladık ve çerçevedeki puanlama aşamasına girdik. MAPTF Kapsam Matrisi Yapısından alınan örneklem Çizelge 3.1'de verilmiştir. MAPTF kapsam matrisi ve matris kapsamında tanımlanan 45 kuralın tamamı EK-1 de verilmiştir.

Çizelge 3.1. MAPTF kapsam matrisi

| M<br>A<br>P<br>T<br>F<br>ID | MASV<br>S<br>A<br>L<br>T<br>M<br>A<br>D<br>D<br>E<br>S<br>İ | ZAFİYET<br>AÇIKLAM<br>ASI                            | KULL<br>ANIL<br>AN<br>GÜVE<br>NLİK<br>ARAÇ<br>LARI | ZAFİ<br>YET<br>SKOR<br>UNA<br>ETKİ<br>Sİ | YÖN<br>TEM | ETKİSİ       | CVSS<br>PUAN<br>I | CWE                             | ANA<br>HTA<br>R<br>KEL<br>İME<br>LER |
|-----------------------------|-------------------------------------------------------------|------------------------------------------------------|----------------------------------------------------|------------------------------------------|------------|--------------|-------------------|---------------------------------|--------------------------------------|
| 1                           | MSTG-<br>RESILIEN<br>CE-13                                  | Uygulama<br>Apk'sının<br>decompile<br>edilmesi.      | APKTOOL                                            | +/-                                      | NVD        | CRITICA<br>L | 9.0               | CWE-919,<br>Weakness<br>ID: 919 |                                      |
| 2                           | MSTG-<br>PLATFORM-1                                         | Uygulama<br>kapsamında,<br>gereksiz<br>kullanılan    | MOBSF<br>ANDROGUARD                                | -                                        | NVD        | CRITICAL     | 9.8               | CWE-250,<br>Weakness ID: 250    |                                      |
| 10                          | MSTG-AUTH-<br>9                                             | Uygulamada<br>2FA etkin mi<br>aktif edilmiş<br>mi?   | APKTOOL<br>MOBSF                                   | +                                        | NVD        | HIGH         | 7.5               | CWE-200,<br>Weakness ID:<br>200 | MultiFactorSe<br>ssion               |
| 23                          | MSTG-<br>PLATFORM-<br>1                                     | Content<br>providerlara<br>erişim izni<br>verilmemiş | ANDROGU<br>ARD                                     | -                                        | NVD        | LOW          | 3.5               | CWE-684,<br>Weakness ID:<br>684 | android:expo<br>rted="false"         |
| 28                          | MSTG-<br>CODE-2                                             | Uygulamada<br>debug<br>,<br>opsiyonları<br>açık mı?  | ANDROBU<br>GS                                      | -                                        | NVD        | MEDIUM       | 6.5               | CWE-117,<br>Weakness ID:<br>117 | android:allo<br>wBackup              |

Uygulama çatısını kullanmak için gerekli ön koşullar şu şekildedir: MAPTF'in çalışacağı sistem Linux veya MacOS işletim sistemine sahip olmalı, Java SDK 8 veya üzeri bir sürüm yüklü olmalı, Python 3 veya üzeri bir sürüm yüklü olmalı. Bunun dışındaki diğer ön koşullar, MAPTF kullanılırken çalıştırılması gereken "requirements.sh" dosyası vasıtasıyla yüklenecektir. Java SDK ve Python hedef sistem üzerinde yüklü değilse "requirements.sh" dosyası, bu kütüphane ve araçların yüklenmesinden sorumlu olacaktır.

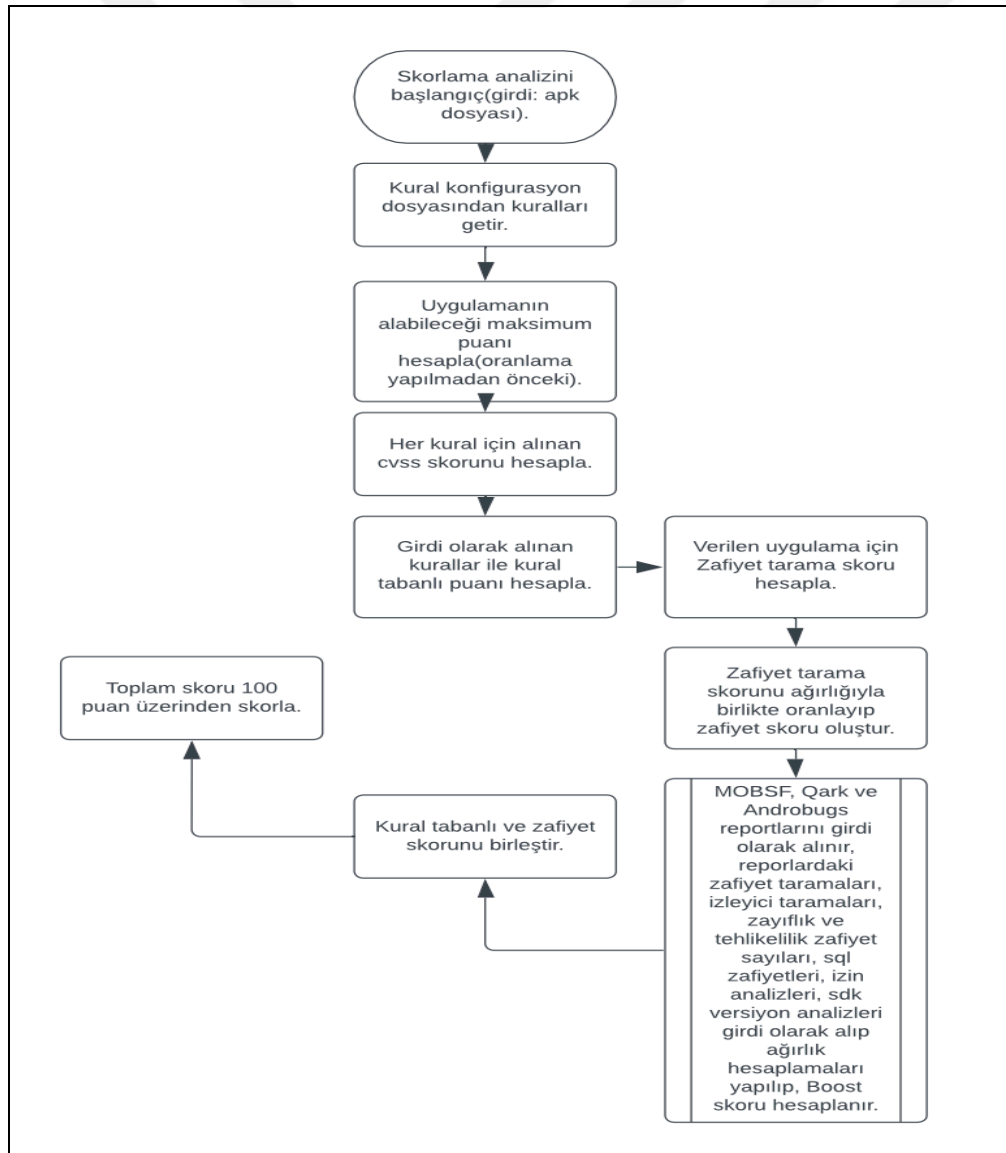
MAPTF uygulama çerçevesinin sağladığı analiz ve puanlama özellikleri, MASVS, MSTG ve OWASP Mobil Uygulama Güvenliği Kontrol Listesi standartlarına dayanmaktadır. Güvenlik analisti, kendi kurallarına dayalı bir puanlama altyapısı oluşturma imkanına sahiptir. Var olan temel MAPTF uygulama çerçevesi kurallarına kendi analiz kurallarını ekleyebilir ve bu kuralların puan üzerindeki etkisini yapılandırılabilir. Bunun için, kural tabanlı puanlamanın kontrol noktası olan yapılandırma dosyasında, kendi kuralları hakkında bu bilgileri sağlama imkanına sahiptir: zayıflığın tanımı, hangi araçla analiz yapılacağı, tanımlanmış kuralın zafiyet puanı üzerindeki etkisi, bu puanın nasıl hesaplandığı, bu kural maddesiyle ilgili CWE ögesi, CVSS puanı (geliştirme aşamasında belirlenen, varsayılan MAPTF kuralları içindeki zayıflıklar arasındaki en yüksek puan CVSS puanı olarak parametre olarak alınır), anahtar kelime aramalarının yapılması, bu kural kapsamında anahtar kelime araması yapılacaksa aranacak anahtar kelime, ilgili CWE ögesinin bağlantısı ve hedef işletim sistemi bilgisi. Bu kural tanımları sağlandıktan sonra, analist MAPTF temel kuralları ile kendi tanımladıkları kural veya sadece kendi tanımladıkları kural ile analiz işlemini başlatabilir ve zafiyet puanını oluşturabilir.

Önerilen çalışma ile oluşturulan puanların, çalışma kapsamındaki zafiyet sınıflandırmaları, puan aralıkları aynı zamanda web arayüzü üzerinde görselleştirme için kullanılan renkler Çizelge 3.2'deki gözükmemektedir. Sınıflandırma aşamalarında kullanılan aralıkların seçilme gerekçesi, belirlenen alt kategori sayısına göre basamakları eşit olarak bölmektir böylelikle puanlama aşamasının standartlaşması ve kolay anlaşılabilirlik açısından avantaj sağlayacağı değerlendirilmiştir. MAPTF ile yapılan uygulama puanlamasında seçilen yaklaşımın daha tutarlı bir sonuç elde edildiği tespit edilmiştir. MOBSF [27] çalışmasında uygulama zafiyet puanlaması için kullanılan tam çeyreklik kullanmama yöntemi yerine dağıtık aralık seçme stratejisi tercih edilmemiştir.

Çizelge 3.2. MAPTF puanlama zafiyetleri sınıflandırma aralıkları

| Puan aralığı (100 üzerinden) | Sınıflandırma           | Sınıflandırma Rengi |
|------------------------------|-------------------------|---------------------|
| 0-25                         | Risksiz                 | Yeşil               |
| 25-50                        | Düşük Potansiyel Riskli | Sarı                |
| 50-75                        | Potansiyel Riskli       | Turuncu             |
| 75-100                       | Riskli                  | Kırmızı             |

Çalışma kapsamında önerilen metot kullanılarak APK dosyasının zafiyet temelli puanlama işleminde geçtiği aşamalar ve her aşamada kullanılan puanlama yöntemleri Şekil 3.5'te görülmektedir.



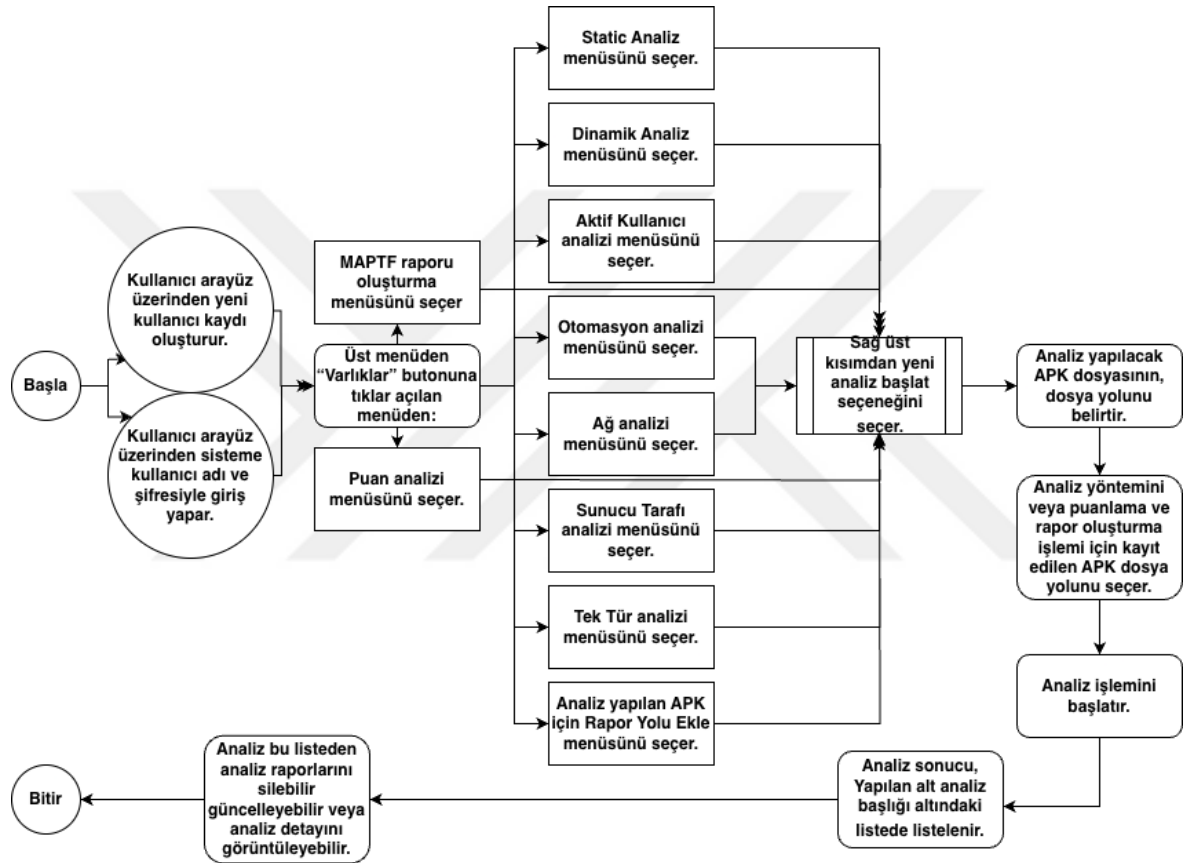
Şekil 3.5. Puanlama süreci genel bakış

Ulusal Güvenlik Açığı Veri tabanı (NVD), Ulusal ABD Ulusal Standartlar ve Teknolojisi Enstitüsü (NIST) tarafından yönetimi yapılan, bilgisayar sistemleri, ağlar ve uygulamaların zayıflıkları hakkında kapsamlı kayıtlar içeren, güvenlik profesyonelleri ve araştırmacılar tarafından yaygın olarak kullanılan kaynaktır [42]. Burada tutulan kayıtlar, ‘temel zayıflık girdileri’ olarak adlandırılır ve zayıflıkların şiddetini ve potansiyel etkisini içermektedir. MAPTF çalışmasının temel kural kümesinde, tanımlanan zayıflıkların CVSS puanlarını belirlemek için iki puanlama yöntemi de kullanılmaktadır [40]. Bunlardan yöntemlerden ilki, ‘NVD TARAFINDAN ÖNCEDEN HESAPLANMIŞ PUAN’dır. Bu puanlama türünde, zayıflığın ilgili sisteme etkisi NVD tarafından daha önce hesaplanmış ve paylaşılmış hesaplamalar kullanılarak yapılmıştır. İkinci yöntem ise, ‘NVD CVSS-V3-HESAP MAKİNESİ UYGULAMASI TARAFINDAN MANUEL HESAPLANAN PUAN’dır. Güvenlik analistleri, NVD'nin CVSS-V3-Calculator [41] uygulamasını kullanarak zayıflığın özelliklerini belirleyerek bir puan hesaplamasını sağlarlar. MAPTF kapsamında seçilen bu farklı puanlama yöntemleri, zayıflıkların tutarlı ve güvenilir bir değerlendirme temeline oturtmayı amaçlar.

İlgili Common Weakness Enumeration (CWE) maddesi ile Common Vulnerabilities and Exposures (CVE) veri tabanı arasındaki eşleme oluşturulamadığı veya CWE maddesinin tespit edilen zayıflıklar için CVSS skoru oluşturmadığı durumlarda CVSS-V3-Calculator tarafından manuel puan hesaplama yaklaşımı kullanıldı. Anlatılan yöntemin örnek kullanım senaryosunda CWE 676'nın puanlama değerlendirmesi yapılırken, ilgili CWE maddesine karşılık gelecek CVE maddesi bulunamamıştır. Bu sebeple, en yaygın olarak kullandığımız standart metod olan CVSS puanı elde edilememiştir. Bu aşamada, CVSS puanını hesaplamak için NVD'nin CVSS-V3-Calculator kullanılmıştır. Değerlendirme aşamasında, CVSS-V3-Calculator aracını etkili bir şekilde kullanmak için CWE 676 zafiyetin güvenlik yönlerinden ortaya çıkartabileceği potansiyel etkiler girdi olarak kullanılmıştır. CVSS tarafından belirlenen kılavuzlar doğrultusunda, temel skor metriklerinin hesaplanması için aşağıdaki nitelikler CWE 676 için seçilmiştir: Saldırı Vektörü (AV) Yerel olarak (AV: L) belirlendi, Saldırı Karmaşıklığı (AC) Düşük olarak (AC: L) atandı, Kullanıcı Etkileşimi (UI) Hiçbir şey olarak (UI: N) atandı, Kapsam (S) Değişmedi olarak (S: U) atandı, Gizlilik (C) Yüksek olarak (C: H) atandı, Bütünlük Etkisi (I) Yüksek olarak (I: H) atandı ve Kullanılabilirlik Etkisi (A) Düşük olarak (A: L) atandı. Zaman skor metrikleri için aşağıdaki nitelikler belirlendi: Açığın Kod Olgunluğu (E) Proof of Concept olarak (E: P) atandı, Giderme Seviyesi (RL) Çözüm olarak (RL: W) atandı ve

Rapor Güvenliği (RC) Doğrulandı olarak (RC: C). NVD CVSS-V3-Calculator aracı tarafından CWE 676'nın hesaplanan temel skoru 6,3 olarak tespit edilmiştir.

Şekil 3.6'da uygulama kapsamında mobil uygulama güvenlik analisti tarafından gerçekleştirilecek analizler için kullanılacak arayüz için oluşturulan Mobil Uygulama Güvenlik Analisti Arayüz Kullanımını göstermektedir.



Şekil 3.6. Mobil Uygulama Güvenlik Analisti Arayüz Kullanımı gösterimi

#### 4. BULGULAR VE TARTIŞMA

Bu bölümde, önerilen aracının deneysel sonuçları sunulmaktadır. MOBSF ve Qark MAPTF araçlar çeşitli güvenlik kriterleri açısından birbirleriyle karşılaştırılmıştır. Önerilen araç, MASVS, MSTG ve OWASP Mobil Uygulama Güvenliği Kontrol Listesi'ni otomatize olarak kural tabanlı yaklaşımla kapsamakta ve genişletilebilmesine olanak sağlamaktadır. Aynı zamanda analistlere analizi yapılan mobil uygulama için bir zafiyet puanı hesaplamakta bunu bir güvenlik analiz raporuyla sunmaktadır.

Mobil uygulama güvenlik araçları birbirleriyle karşılaştırılırken kullanılan parametreler şu şekildedir; analize girdi olarak sağlanan mobil uygulamada OWASP MASVS maddelerini ve bu maddelerin uygulanmasının kapsama yüzdesi, mobil uygulamada algılanan kötücül yazılım tespit sayısı, mobil uygulamada tespit edilen kötücül yazılım ayrıntı bilgisi, mobil uygulamada tespit edilen potansiyel veri sızıntı sayısı, mobil uygulamada tespit edilen URL Sayısı, mobil uygulamada tespit edilen analitik uygulamalarının sayısı, mobil uygulamada kullanılan güvensiz kütüphane bağımlılığı sayısı, mobil uygulamada tespit edilen dışa açık Activity sayısı, mobil uygulamada tespit edilen dışa açık Provider sayısı, mobil uygulamada tespit edilen dışa açık Receiver sayısı, mobil uygulamada tespit edilen kötücül yazılım yorum bilgileri, mobil uygulama analizi sonucunda zafiyet puanı oluşturma özelliği ve mobil uygulama analisti tarafından sağlanan genişletilebilir kural tabanlı puanlama. Karşılaştırmanın sonuçları Çizelge 4.1'de gösterilmiştir.

Çizelge 4.1. Mobil uygulama analiz araçlarının karşılaştırılması

|                                                                             | Qark      | MOBSF     | MAPTF                                                |
|-----------------------------------------------------------------------------|-----------|-----------|------------------------------------------------------|
| MASVS Kapsama Yüzdesi                                                       | Bilgi Yok | Bilgi Yok | %60 (%100'e kadar özel kuralar ile yükseltilebilir.) |
| Tespit Edilen Kötüçül Yazılım sayısı (Uygulama Başına Düşen )               | Bilgi Yok | 0.46      | 0.46                                                 |
| Kötüçül Yazılım Ayrıntısı Sunuluyor Mu?                                     | Hayır     | Evet      | Evet                                                 |
| Tespit Edilen Potansiyel Veri Sızıntı Sayısı (Uygulama Başına Düşen)        | 178.92    | 88.20     | 149.01                                               |
| Tespit Edilen URL Sayısı (Uygulama Başına Düşen)                            | 117.52    | 17.22     | 109.62                                               |
| Tespit Edilen Analitik Takip Edici Sayısı (Uygulama Başına Düşen)           | Bilgi Yok | 3.24      | 2.62                                                 |
| Tespit edilen güvensiz kütüphane bağımlılığı sayısı (Uygulama Başına Düşen) | Bilgi Yok | Bilgi Yok | 1.17                                                 |
| Tespit edilen Dışa Erişime Açık Activity Sayısı (Uygulama Başına Düşen)     | 2.90      | 2.71      | 2.90                                                 |
| Tespit edilen Dışa Erişime Açık Provider Sayısı (Uygulama Başına Düşen)     | 0.58      | 0.58      | 0.58                                                 |
| Tespit edilen Dışa Erişime Açık Receiver Sayısı (Uygulama Başına Düşen)     | 0.53      | 0.53      | 0.53                                                 |
| Kötüçül Yazılım Yorum Bilgileri Sunuluyor Mu?                               | Hayır     | Hayır     | Evet                                                 |
| Araç, Zafiyet Puanı Oluşturma Özelliğine Sahip mi?                          | Hayır     | Evet      | Evet                                                 |
| Genişletilebilir Kural Tabanlı Zafiyet Puanı Özelliğine Sahip Mi?           | Hayır     | Hayır     | Evet                                                 |

Çizelge 4.1'de sunulan karşılaştırma değerleri, 50 farklı mobil uygulama dosyası üzerinde yapılan testlerden elde edilen ortalama sonuçları yansıtmaktadır. Bu çalışmada kullanılan mobil test uygulamaları Google Play Store, ApkPure ve Aptoide gibi çeşitli uygulama marketlerinden, ayrıca bilinen güvensiz uygulamalardan seçilerek bir test havuzu oluşturulmuştur.

Çizelge 4.2. Mobil uygulama bazlı zafiyet puanı karşılaştırması

| Mobil Uygulama Adı               | VirusTotal Zararlı tespit edilme oranı(Tespit Sayısı / Toplam Güvenlik sağlayıcısı) | Uygulama Zafiyete Sahip Mi? | MOBSF Zafiyet Puanı (100 puan üzerinden) | MOBSF Sınıflandırma Kategorisi | MAPTF Zafiyet Puanı (100 puan üzerinden) | MAPTF Sınıflandırma kategorisi |
|----------------------------------|-------------------------------------------------------------------------------------|-----------------------------|------------------------------------------|--------------------------------|------------------------------------------|--------------------------------|
| 7-ElevenTH.apk                   | 0/65                                                                                | Hayır                       | 51/100                                   | Orta                           | 31.93/100                                | Düşük Potansiyel Risk          |
| InsecureBankv2.apk               | 10/65                                                                               | Evet                        | 13/100                                   | Kritik                         | 89.43/100                                | Riskli                         |
| AutoClicker-AutomaticTap.apk     | 0/65                                                                                | Hayır                       | 50/100                                   | Orta                           | 22.31/100                                | Risksiz                        |
| PIVAA.apk                        | 2/65                                                                                | Evet                        | 42/100                                   | Orta                           | 57.39/100                                | Potansiyel Risk                |
| AndroGoat.apk                    | 0/65                                                                                | Hayır                       | 47/100                                   | Orta                           | 23.28/100                                | Risksiz                        |
| Free-Bitcoin-Miner.apk           | 0/65                                                                                | Hayır                       | 50/100                                   | Orta                           | 28.02/100                                | Düşük Potansiyel Risk          |
| InjuredAndroid.apk               | 2/65                                                                                | Evet                        | 55/100                                   | Orta                           | 49.63/100                                | Düşük Potansiyel Risk          |
| Bitcoin-Bank.apk                 | 0/65                                                                                | Hayır                       | 50/100                                   | Orta                           | 26.89/100                                | Düşük Potansiyel Risk          |
| DVBA.apk                         | 1/65                                                                                | Evet                        | 49/100                                   | Orta                           | 45.91/100                                | Düşük Potansiyel Risk          |
| Wether-Local-Ethereum-Wallet.apk | 0/65                                                                                | Hayır                       | 50/100                                   | Orta                           | 31.39/100                                | Düşük Potansiyel Risk          |

Çizelge 4.2’de, MOBSF, MAPTF ve VirusTotal araçlarının örnek uygulamalar için analiz sonuçlarının karşılaştırılması gösterilmektedir. İlk sütunda analize konu mobil uygulamaların isimleri yer almaktadır. İkinci sütunda, 65 güvenlik sağlayıcısı tarafından VirusTotal aracılığıyla yapılan zararlı yazılım taramalarının sonuçlarını gösterilmektedir. Zafiyetleri tespit eden güvenlik sağlayıcılarının sayısı VirusTotal sütununda belirtilir. Üçüncü sütunda, MOBSF aracı tarafından oluşturulan zafiyet puanlarını gösterilmektedir. MOBSF tarafından düşük bir puan, uygulamanın daha yüksek güvensizlik riski taşıdığı anlamına gelmektedir. Dördüncü sütunda, MAPTF tarafından tespit edilen zafiyet temelli puanlama sonuçları gösterilmektedir. Bu sütun için, yüksek güvensizlik puanı, uygulamanın zafiyet riski taşıma ihtimalinin daha yüksek olduğu göstermektedir. Qark analiz araçlarının sonuç karşılaştırmasında olmamasının sebebi, Qark tarafından yapılan analiz sonucunda bir güvensizlik puanı oluşturulamamasından kaynaklanmaktadır. Bu nedenle, MOBSF, VirusTotal ve MAPTF araçları karşılaştırılarak sonuçlar gösterilmiştir.

Ayrıca önerilen araç CWE ve CVSS puanlaması kullanması sebebiyle güvenilir bir puanlama skalasına sahiptir. Kural tabanlı puanlama özellikleri kolayca mobil güvenlik analistleri tarafından genişletilebilir, bu da güvenlik analistlerine verimli bir araç olarak kullanılmasına olanak sağlamaktadır.

Özet olarak, önerilen aracın mobil uygulama güvenlik analizi ve testi alanında bir değer ortaya koyması amaçlanmıştır. Güvenlik analistlerine mobil uygulamaları analiz etmek için güçlü ve esnek bir çerçeve sunmaktadır. Karşılaştırma sonuçları, mevcut araçların zayıf noktalarını ve zafiyet temelli puanlama yaparak etkili ve güvenilir bir çözüm ortaya koymaktadır. Çizelge 4.1 ve Çizelge 4.2’de verildiği üzere, MOBSF ve Qark araçlarıyla, tespit edilen zafiyet ve açıklık parametreleri üzerinden karşılaştırma yapıldığında; MAPTF uygulama çatısı, MASVS maddelerinin analizi ve kapsamasını varsayılan olarak gerçekleştirmesi sebebiyle diğer araçlardan pozitif olarak ayrılmaktadır. Ayrıca MAPTF aracı, tespit edilen potansiyel veri sızıntı sayısında, MOBSF aracından ortalama 50 adet fazla potansiyel zafiyet içeren bulgu tespit etmiştir. MAPTF aracı, mobil uygulamada tespit edilen gömülü URL sayısında MOBSF aracından ortalama 90 tane fazla bulgu tespit etmiştir. Bu madde kapsamında Qark aracı MAPTF aracından ortalama 5 fazla bulgu tespit etmiştir. MAPTF, Mobil uygulamaların analizinde kullanılan dışarı açılan aktivite parametresinde MOBSF uygulamasından ortalama 0.05 fazla bulgu tespit ederek Qark aracı ile aynı sonucu elde etmiştir. Mobil uygulamaların sızma testlerinde zafiyetli

içeriğinin tespiti için kullanılan MAPTF uygulaması MOBSF ile VirusTotal tespitleri üzerinden karşılaştırıldığında, zafiyetlere verilen puanlarda MAPTF in MOBSF e nazaran daha hassas sonuçlar verdiği gözlemlenmiştir. Örnek olarak Tablo-4 de de verilen veriler incelendiğinde, AutoClicker-Automatictap.apk isimli VirusTotal tarafından herhangi bir zafiyet içermeyen uygulamaya MOBSF aracı 100 puan üzerinden 50 zafiyet puanı verirken, bu puan MAPTF için 22.31'dir. Aynı şekilde Tablo-4 de başka bir uygulama karşılaştırma için seçilirse, Bitcoin-Bank.apk isimli mobil uygulama VirusTotal tarafından zafiyet tespit edilmemiştir ve MOBSF bu uygulamaya 50 puan vermiştir, MAPTF bu uygulamaya 26.89 zafiyet puanı vermiştir. Verilen örnekler, önerilen uygulama çatısının MAPTF in analistlere ve araştırmacılara güvenilir ve efektif bir kaynak olarak sunulması amaçlanmıştır.

Çalışmanın karmaşıklığının bir sınırlama olarak dikkate alınması önemlidir. Önerilen çalışma, etkili bir şekilde kullanmak için daha fazla kaynak ve uzmanlık gerektirebilir. Önerilen çalışma, OWASP kılavuzlarını otomatik hale getirme, zafiyet temelli puanlama yapması ve güvenlik analistleri tarafından özelleştirilebilen konfigürasyon temelli kurallar ile puanlamanın etkilenebilmesine olanak tanıma gibi geniş bir özellik ve yetenek seti sunmakla birlikte, mobil uygulama güvenlik analizi ve testi için etkili bir araç olması için tasarlanmıştır.



## 5. SONUÇ

Önerilen Çalışma, mobil uygulama bileşenleri hakkında statik, ikili analiz, bilgi toplama, bilgi manipülasyonu, uygulama kaynak koduna ve apk dosyalarından kaynak koduna erişim, Java kaynak kodu sınıflarının derlenmesi gibi işlemleri yapabilmektedir. Ayrıca, kaynak kod analizleri, kabuk komutlarını kontrol etme, zafiyet analizi, diğer uygulamalar ile arasındaki iletişimi kontrol etme, uygulamaya tanınan bellek erişimi kontrolü, mobil uygulama kaynak kodunun SQL Enjeksiyon saldırılarına olan hassasiyetini kontrol etme ve gömülü parolaları kontrol etme gibi işlemler yapabilmektedir.

Çalışma kapsamında, analistlere mobil uygulamalardaki güvenlik zafiyet puanlamasını etkilemek için kuralların özelleştirilmiş hallerini oluşturma olanağı sunulmuştur. Bu özellik, tanımlanmış temel kuralların ve bu kuralların ilgili parametrelerinin konfigürasyon dosyası vasıtasıyla değiştirilmesiyle kullanılmaktadır. Ayrıca, çalışmanın MongoDB veri tabanı entegrasyonu, tüm analiz türlerinin çıktılarının analiz zamanları ve analiz sonucuyla birlikte saklanması, web arayüzü ve REST uç nokta entegrasyonlarıyla dışarıdan erişilebilmesi çalışmanın analistlere sağladığı takip edilebilirliğini güçlendirmektedir.

Güvenlik analisti, çalışma kapsamında zafiyet temelli puanlama yaptığı uygulamaların sonuçlarını web arayüzü üzerinden mobil uygulamanın zafiyet sonucuna göre değişen renkler ile görselleştirir. Zafiyet temelli puan üretirken oluşturduğu uygulama güvenlik analiz raporunu da veri tabanında saklar. Analiz raporu, arayüz üzerinden, analiz yapılan uygulama detayı sayfası vasıtasıyla görüntüleyebilmektedir.

MAPTF uygulama çerçevesi, statik, dinamik, ağ, güvenlik açığı analizi, puanlama seçenekleri, APK kaynak kodlarına ayrıştırma, anahtar kelime tarama, şifreleme algoritması güvenliği kontrolü, port tarama aktiviteleri, manifest analizi, uygulama bileşeni tarama, izin analizi, analiz edilen uygulamada kullanılan analitik izleyicilerin denetimi ve uygulama kapsamında potansiyel hassas bilgi tarama gibi özellikler sunar. Ayrıca, MAPTF zafiyet temelli puanlama özelliğine sahiptir. Zafiyet temelli puanlama özelliğine girdi olarak alınan analiz kural setleri, güvenlik analisti tarafından konfigürasyon dosyası vasıtasıyla düzenlenebilir, değiştirilebilir, genişletilebilir. Bu özellik güvenlik analistinin zafiyet temelli puanlama aşamasına doğrudan etki etmesine olana sağlar.

MAPTF aracı kompleks yapısı ve ek olarak diğer mobil analiz araçlarının yüklenmesi gerekliliği limit olarak bulunurken, MASVS özelliklerini kapsamaması, kural tabanlı puanlama özellikleri ve birden çok güvenlik aracını kullanma yeteneğini kapsayan daha geniş bir mobil analiz özelliği yelpazesine sahip olması önemli bir avantajıdır. Aynı zamanda önerilen çalışma OWASP tarafından geliştirilen, uygulama dosyalarındaki bağımlılıkları tarayan ve daha önce tespit edilen bir açıklığa sahip kütüphane kullanımı olup olmadığının kontrolü yapan OWASP Dependency-Check uygulaması tarama faaliyetini uygulama başına ortalama 4 dakika da tamamlamaktadır [43]. Bunun dışında MAPTF uygulaması diğer uygulama araçları tarafından oluşturulan raporlarla da beslenerek sağlıklı bir zafiyet tarama raporu ve zafiyet tarama puanı oluşturmaktadır. Bu kapsamda kullanılan araçlardan birisi olan Qark aracı mobil uygulama tarama faaliyetini ortalama 2 dakikada tamamlamaktadır. Bu araçların kullanılması sebebiyle uygulama tarama faaliyetleri, rapor üretme aşaması ve puanlama aşaması zaman almaktadır. Bu maddeler, çalışma kapsamında limit olarak değerlendirilmektedir. Bu sorunlar çözüldüğü takdirde, sistemin daha büyük veri setleri üzerinde çalıştırılması ve test edilmesi daha doğru sonuçların elde edilmesi olarak sağlayacaktır.

Bu çalışmanın kapsamında, mobil uygulama analizi aşamalarını kolaylaştırmak için gerekli tüm araçları içeren, OWASP temelli yol gösterici olabilecek, zafiyet temelli puanlama yapabilen, kural temelli özelleştirilebilir uygulama analizleri yaparak analiz raporu sunabilen, mobil uygulamalar için otomatize penetrasyon testi çerçevesini oluşturulmuştur. Çalışma, mobil uygulamanın genel bir görünümünü sağlamak ve analistlere analiz süreçlerinde zaman ve karmaşıklık açısından avantaj kazanmalarına yardım etmek için zafiyet temelli puanlama tekniklerini kullanmıştır. Mevcut çalışmanın iyileştirilmesi ve gelecekte yapılacak çalışmalar için bazı öneriler bulunmaktadır. İlk olarak, MAPTF Uygulama çatısının Docker'a uyumlu hale getirilerek ve bulut ortamına taşınarak, ilk sistem kurulumu maliyetinden kurtulmak mümkün olabilir. İkinci olarak, MAPTF Uygulama çatısına farklı güvenlik analiz araçlarının entegrasyonu sağlanarak analiz ve puanlama süreci hassaslaştırılabilir ve genişletilebilir. Üçüncü olarak, kural temelli puanlama aşamasında kullanılan parametrelerin ağırlıklarının hesaplanmasında makine öğrenmesi yöntemlerinden faydalanılabilir. Bu sayede daha verimli ve etkili sonuçlar elde edilebilir. Gelecekte yapılacak çalışmalarda bu önerilerin değerlendirilmesi, çalışmanın gelişimine katkı sağlayabilir.

## KAYNAKLAR

1. Global Stats. (2023). Mobile Operating System Market Share Worldwide Research Report. *GStats*, Firenze.
2. Türkiye İstatistik Kurumu. (2022). Hanehalkı Bilişim Teknolojileri (BT) Kullanım Araştırması. *TUIK*, Ankara.
3. Rasthofer, S., Arzt, S., Kolhagen, M., Pfretzschner, B., Huber, S., Bodden E. ve Richter P. (2015). *DroidSearch: A Tool for Scaling Android App Triage to Real-World App Stores*. 2015 Science and Information Conference, Londra, 247–256.
4. Ding, C., Luktarhan, N., Lu, B. ve Zhang, W. (2021). A Hybrid Analysis-Based Approach to Android Malware Family Classification. *Entropy*, 23(8), 1009-1032.
5. İnternet: QARK (2015), Tool to look for several security related application vulnerabilities. URL: <https://github.com/linkedin/qark>, Son Erişim Tarihi: 09.04.2023.
6. İnternet: Androbugs (2015), AndroBugs Framework is an efficient Android vulnerability scanner that helps developers or hackers find potential security vulnerabilities in Android applications. URL: [https://github.com/AndroBugs/AndroBugs\\_Framework](https://github.com/AndroBugs/AndroBugs_Framework), Son Erişim Tarihi: 09.04.2023.
7. İnternet: Apktool (2010), A tool for reverse engineering Android apk files. URL: <https://github.com/iBotPeaches/Apktool>, Son Erişim Tarihi: 09.04.2023.
8. European Union Agency for Network and Information Security. (2017). Privacy and data protection in mobile applications, A study on the app development ecosystem and the technical implementation of GDPR. *ENISA*, Kandiye. 1-70.
9. Cai, H. ve Ryder, B. G. (2017). *Artifacts for Dynamic Analysis of Android Apps*. International Conference on Software Maintenance and Evolution, Şangay. 659–659.
10. İnternet: Frida (2013), Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers. URL: <https://frida.re/>, Son Erişim Tarihi: 09.04.2023.
11. İnternet: Drozer (2012), About The Leading Security Assessment Framework for Android. URL: <https://github.com/FSecureLABS/drozer>, Son Erişim Tarihi: 09.04.2023.
12. Zaman, M., Siddiqui, T., Amin, M. R. ve Hossain, M. S. (2015). *Malware Detection in Android by Network Traffic Analysis*. International Conference on Networking Systems and Security, 1-5.
13. İnternet: Burp Suite(2004). URL: <https://portswigger.net/burp>, Son Erişim Tarihi: 09.04.2023.
14. İnternet: Wireshark (1998). URL: <https://github.com/wireshark/wireshark>, Son Erişim Tarihi: 09.04.2023.

15. Shi, Y., You, W., Qian, K, Bhattacharya, P. ve Qian Y. (2016). *A Hybrid Analysis for Mobile Security Threat Detection*. IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, New York, 1-7.
16. Acheampong, E. O. ve Kaur, E. D. (2022). Automatic And Manual Discovery of Vulnerabilities in Selected Android Mobile Applications (Penetration Testing Approach). *International Research Journal of Modernization in Engineering Technology and Science*, 4(5), 1437-1458.
17. Almohaini, R., Almomani, I. ve AlKhayer, A. (2021). Hybrid-Based Analysis Impact on Ransomware Detection for Android Systems. *Applied Sciences*, 11(22), 10976-10999.
18. Özdemir, D. ve Zaim, H. Ç. (2021). Investigation Of Attack Types in Android Operating System. *Journal of Scientific Reports-A*, 46, 34-58.
19. Nilsson, R. (2020). *Penetration Testing of Android Applications*. Lisans Tezi, KTH/Skolan för elektroteknik och datavetenskap Üniversitesi, Stokholm, 37-80.
20. Williams, M. A., Barranco R. C., Naim, S. M., Dey, S., Hossain, M. S. ve Akbar, M. (2020). A vulnerability analysis and prediction framework. *Computers & Security*, 92(1), 142-148.
21. Nunes J. M. M. (2019). *DroidsF - A Framework For Security Analysis Of Mobile Applications*, Yüksek Lisans Tezi, Lizbon Teknik Üniversitesi, Bilişim ve Bilgisayar Teknolojileri Enstitüsü, Lizbon, 1-66.
22. Alanda, A., Satria, D., Mooduto, H. A. ve Kurniawan, B. (2020). Mobile Application Security Penetration Testing Based on OWASP. *IOP Materials Science and Engineering*, 846(1), 1-12.
23. Kim, D., Shin, D., Shin, D. ve Kim, Y. H. (2019). Attack Detection Application with Attack Tree for Mobile System using Log Analysis. *Mobile Networks and Applications*, 24, 184-192.
24. Yeom, C. ve Won, Y. (2019). Vulnerability Evaluation Method through Correlation Analysis of Android Applications. *Sustainability*, 11(23), 6637.
25. Mishra, B., Agarwal, A., Goel, A., Ansari, A. A. ve Gaur, P. (2022). Privacy Protection Framework for Android. *IEEE Access*, 10, 7973 - 7988.
26. İnternet: Guyton, F. (2018). A Survey of Android Security Threats and Machine Learning Techniques Used for Detection, URL: [https://www.researchgate.net/publication/326247336\\_A\\_survey\\_of\\_Android\\_security\\_threats\\_and\\_machine\\_learning\\_techniques\\_used\\_for\\_detection](https://www.researchgate.net/publication/326247336_A_survey_of_Android_security_threats_and_machine_learning_techniques_used_for_detection). Son Erişim Tarihi: 09.04.2023.
27. Abraham, A. (2018). *Automated Mobile Application Security Testing with Mobile Security Framework*, Nullcon GOA'16 Security Conference, Bogmalo.

28. Bhatt, A. J., Gupta, C. ve Mittal, S. (2018). iABC: Towards a hybrid framework for analyzing and classifying behaviour of iOS applications using static and dynamic analysis. *Journal of Information Security and Applications*, 41, 144-158.
29. Benkhelifa, E., Thomas, B. E., Tawalbeh, L. ve Jararweh, Y. (2016). Framework for Mobile Devices Analysis. *Procedia Computer Science*, 83, 1188-1193.
30. Lee S., Dolby, J. ve Ryu, S. (2016). *HybriDroid: Static analysis framework for Android hybrid applications*. 31st IEEE/ACM International Conference on Automated Software Engineering, Singapur, 250- 261.
31. Jadhav, S., Oh, T., Kim, Y. H. ve Kim, J. N. (2015). *Mobile Device Penetration Testing Framework and Platform for The Mobile Device Security Course*. 2015 17th International Conference on Advanced Communication Technology, PyeongChang, 675-680.
32. İnternet: Androguard (2011). URL: <https://github.com/androguard/androguard>, Son Erişim Tarihi: 09.04.2023.
33. İnternet: Nmap (1997). URL: <https://nmap.org/>, Son Erişim Tarihi: 09.04.2023.
34. İnternet: Zed Attack Proxy (2010). URL: <https://www.zaproxy.org/>, Son Erişim Tarihi: 09.04.2023.
35. İnternet: VirusTotal (2007), URL: <https://www.virustotal.com/>, Son Erişim Tarihi: 09.04.2023.
36. İnternet: OWASP Mobile Application Security Testing Guide (2018), MSTG. URL: <https://mas.owasp.org/MASTG/>, Son Erişim Tarihi: 09.04.2023.
37. İnternet: OWASP Mobile Application Security Verification Standard (2016), MASVS. URL: <https://mas.owasp.org/MASVS/>, Son Erişim Tarihi: 09.04.2023.
38. İnternet: OWASP Mobile Application Security Checklist (2020). URL: [https://mas.owasp.org/MAS\\_checklist/](https://mas.owasp.org/MAS_checklist/), Son Erişim Tarihi: 09.04.2023.
39. İnternet: Common Weakness Enumeration (1999), CWE. URL: <https://cwe.mitre.org/>, Son Erişim Tarihi: 09.04.2023.
40. İnternet: The Common Vulnerability Scoring System (2005), CVSS. URL: <https://www.first.org/cvss/>, Son Erişim Tarihi: 09.04.2023.
41. İnternet: Common Vulnerability Scoring System Calculator (2015), CVSS Calculator. URL: <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>, Son Erişim Tarihi: 09.04.2023.
42. İnternet: National Vulnerability Database (2000), NVD. URL: <https://nvd.nist.gov/>, Son Erişim Tarihi: 09.04.2023.
43. İnternet: OWASP Dependency-Check (2020), Dependency-Check. URL: <https://owasp.org/www-project-dependency-check/>, Son Erişim Tarihi: 09.04.2023.

44. İnternet: Mobile Security Framework (2015), MOBSF. URL: <https://github.com/MobSF/Mobile-Security-Framework-MobSF>, Son Erişim Tarihi: 09.04.2023.





**EKLER**

## EK-1. MAPTF Kapsam Matrisi

| MAPTF ID | MASVS SUBSTRACE                                                                          | EXPLANATION                                                                                                                                                                | TOOL                                | SCORE IMPACT | METHOD                      | Rating   | VULNERABILITY SCORE(CVSS) | CWE                                                                                                                             | KEYWORD(CONTAIN)                      | LINK(CVSS SCORE)                                                                            |
|----------|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------------|-----------------------------|----------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|---------------------------------------------------------------------------------------------|
| 1        | MSTG-PLATFORM-13<br>RESILIENCE-13                                                        | Uygulama Apk'nın decompile edilmesi.                                                                                                                                       | APKTOOL                             | NEUTRAL      | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.0                       | CWE-919: Weaknesses in Mobile Applications View ID: 919 Type: Implicit                                                          |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-919">https://www.openCVE.io/cve?cwe=CWE-919</a> |
| 2        | MSTG-PLATFORM-ARCH-10<br>ARCH-10                                                         | Uygulama kapsamında, gereksiz kullanılan izinlerin analizinin yapılması                                                                                                    | MOBSF ANDROGUARD                    | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.8                       | CWE-250: Execution with Unnecessary Privileges Weakness ID: 250 Abstraction: Base Structure: Simple                             |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-250">https://www.openCVE.io/cve?cwe=CWE-250</a> |
| 3        | MSTG-CODE-5<br>ARCH-1<br>ARCH-1A<br>ARCH-1B<br>ARCH-1C<br>ARCH-1D<br>ARCH-1E<br>CRYPTO-4 | Uygulama kötücü yazılım içerisinde mi?                                                                                                                                     | APKTOOL VIRUSTOTAL MOBSF ANDROGUARD | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.1                       | CWE-506: Embedded Malicious Code Weakness ID: 506 Abstraction: Class Structure: Simple                                          |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-506">https://www.openCVE.io/cve?cwe=CWE-506</a> |
| 4        | MSTG-STORAGE-9<br>STORAGE-9                                                              | Password bilgileri güvensiz saklanmış mı?                                                                                                                                  | APKTOOL MOBSF                       | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.8                       | CWE-922: Insecure Storage of Sensitive Information Weakness ID: 922 Abstraction: Class Structure: Simple                        |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-922">https://www.openCVE.io/cve?cwe=CWE-922</a> |
| 5        | MSTG-STORAGE-2<br>STORAGE-2                                                              | Kritik veriler uygulama dışında saklanmış mı?                                                                                                                              | APKTOOL ANDROGUARD OARK ADB         | POSITIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.1                       | CWE-359: Exposure of Private Personal Information to an Unauthorized Actor Weakness ID: 359 Abstraction: Base Structure: Simple |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-359">https://www.openCVE.io/cve?cwe=CWE-359</a> |
| 6        | MSTG-STORAGE-3<br>CODE-4                                                                 | Uygulama, Hassas veriler loglara yazılmış mı?                                                                                                                              | APKTOOL MOBSF OARK ANDROBUGS        | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | HIGH     | 7.8                       | CWE-117: Improper Output Neutralization for Logs Weakness ID: 117 Abstraction: Base Structure: Simple                           |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-117">https://www.openCVE.io/cve?cwe=CWE-117</a> |
| 7        | MSTG-STORAGE-5<br>CRYPTO-3                                                               | Uygulama, Klavye cache disable edilmiş mi?                                                                                                                                 | ADB MOBSF APKTOOL                   | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | MEDIUM   | 4.6                       | CWE-524: Use of Cache Containing Sensitive Information Weakness ID: 524 Abstraction: Base Structure: Simple                     | android:inputType="textNoSuggestions" | <a href="https://www.openCVE.io/cve?cwe=CWE-524">https://www.openCVE.io/cve?cwe=CWE-524</a> |
| 8        | MSTG-STORAGE-13                                                                          | Uygulamanın dosyaları MODE_WORLD_READABLE veya MODE_WORLD_WRITABLE yetkileri ile oluşturulmuş mu?                                                                          | APKTOOL MOBSF ANDROBUGS             | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | HIGH     | 7.1                       | CWE-749: Exposed Dangerous Method or Function Weakness ID: 749 Abstraction: Base Structure: Simple                              |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-749">https://www.openCVE.io/cve?cwe=CWE-749</a> |
| 9        | MSTG-CRYPTO-2<br>CODE-1                                                                  | Uygulama, şifreleme ilkelerinin kanıtlanmış uygulamalarını kullanır, MD-5 gibi algoritmaların kullanılması önerilmez. Zayıf veya uygun olmayan kriptografi kullanılmış mı? | OARK ANDROBUGS MOBSF                | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 10.0                      | CWE-916: Use of Password Hash With Insufficient Computational Effort Weakness ID: 261 Abstraction: Base Structure: Simple       |                                       | <a href="https://www.openCVE.io/cve?cwe=CWE-916">https://www.openCVE.io/cve?cwe=CWE-916</a> |
| 10       | MSTG-AUTH-9<br>AUTH-10                                                                   | Uygulama kapsamında, İkinci bir kimlik doğrulama ve 2FA uygulanmış mı aktive edilmiş mi?                                                                                   | APKTOOL MOBSF                       | POSITIVE     | PRE-CALCULATED SCORE BY NVD | HIGH     | 7.5                       | CWE-200: Exposure of Sensitive Information to an Unauthorized Actor Weakness ID: 200 Abstraction: Class Structure: Simple       | MultiFactorSession PhoneAuthProvider  | <a href="https://www.openCVE.io/cve?cwe=CWE-200">https://www.openCVE.io/cve?cwe=CWE-200</a> |

## EK-1. (devam) MAPTF Kapsam Matrisi

| MAPTF ID | MASVS SUBSTRACE                                     | EXPLANATION                                                                         | TOOL                          | SCORE IMPACT | METHOD                         | Rating   | VULNERABILITY SCORE(CVSS) | CWE                                                                                                                                | KEYWORD(CONTAIN)                                                                                                                                                                                                                                                            | LINK(CVSS SCORE)                                                                                              |
|----------|-----------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------|--------------|--------------------------------|----------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 11       | MSTG-AUTH-4<br>MSTG-AUTH-5<br>MSTG-AUTH-10          | Uygulama kapsamında, OAuth2 enable edilmiş mi?                                      | APKTOOL<br>ANDROGUARD<br>BURP | POSITIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 9.8                       | CWE-922: Insecure Storage of Sensitive Information<br>Weakness ID: 922<br>Abstraction: Class<br>Structure: Simple                  | AccountManager.get(this)<br>Bundle()                                                                                                                                                                                                                                        | <a href="https://www.openwv.it/cve/cve=CWE-922">https://www.openwv.it/cve/cve=CWE-922</a>                     |
| 12       | MSTG-NETWORK-1<br>MSTG-NETWORK-2<br>MSTG-STORAGE-13 | Uygulama, TLS communication enable edilmiş mi?                                      | MOBSF<br>QARK                 | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 9.8                       | CWE-319: Cleartext Transmission of Sensitive Information<br>Weakness ID: 319<br>Abstraction: Base<br>Structure: Simple             | SSLContext.getInstance("TLS")<br>SSLContext.getDefault()<br>(SSLContext.getDefault().setEnabledProtocols(TLS*))                                                                                                                                                             | <a href="https://www.openwv.it/cve/cve=CWE-319">https://www.openwv.it/cve/cve=CWE-319</a>                     |
| 13       | MSTG-PLATFORM-5<br>MSTG-PLATFORM-6                  | Uygulama, Webview ve JavaScript kod çalıştırma devre dışı bırakılmış mı?            | DROZER<br>MOBSF<br>APKTOOL    | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 10.0                      | CWE-94: Improper Control of Generation of Code (Code Injection)<br>Weakness ID: 94<br>Abstraction: Base<br>Structure: Simple       | WebView myWebView = (WebView) findViewById(R.id.webview)<br>myWebView.getSettings().setJavaScriptEnabled(true)<br>myWebView.getSettings().setAllowFileAccessFromFileUri(true)<br>myWebView.getSettings().setAllowUniversalAccessFromFileUri(true);<br>loadUrl("http://...") | <a href="https://www.openwv.it/cve/cve=CWE-94">https://www.openwv.it/cve/cve=CWE-94</a>                       |
| 14       | MSTG-PLATFORM-10                                    | Uygulama kapsamında, Webview kullanıldıktan sonra cache temizlenmesi yapılmamış mı? | APKTOOL<br>ANDROGUARD         | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 10.0                      | CWE-94: Improper Control of Generation of Code (Code Injection)<br>Weakness ID: 94<br>Abstraction: Base<br>Structure: Simple       | context.deleteDatabase("webview.db")                                                                                                                                                                                                                                        | <a href="https://www.openwv.it/cve/cve=CWE-94">https://www.openwv.it/cve/cve=CWE-94</a>                       |
| 15       | MSTG-PLATFORM-1                                     | Uygulama, BroadcastReceiver, intent kullanmış mı?                                   | APKTOOL<br>MOBSF<br>ANDROBUGS | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | MEDIUM   | 4.6                       | CWE-927: Use of Implicit Intent for Sensitive Communication<br>Weakness ID: 927<br>Abstraction: Intent<br>Structure: Simple        |                                                                                                                                                                                                                                                                             | <a href="https://nvd.nist.gov/vuln/detail/CVE-2001-0984">https://nvd.nist.gov/vuln/detail/CVE-2001-0984</a>   |
| 16       | MSTG-CODE-2<br>MSTG-CODE-4                          |                                                                                     | QARK<br>MOBSF<br>DROZER       | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | MEDIUM   | 6.5                       | CWE-117: Improper Output Neutralization for Logs<br>Weakness ID: 117<br>Abstraction: Base<br>Structure: Simple                     | Log.d(<br>print(<br>println(<br>log(<br>Log.i(<br>Log.info(<br>Log.debug(<br>System.out.println(<br>Log.v(<br>Log.d(<br>Log.w(<br>Log.e(<br>debuggable true                                                                                                                 | <a href="https://www.openwv.it/cve/cve=CWE-117">https://www.openwv.it/cve/cve=CWE-117</a>                     |
| 17       | MSTG-CODE-2<br>MSTG-CODE-4                          | Uygulama, debug modu açık olarak yayımlanmış mı?                                    | MOBSF<br>QARK<br>APKTOOL      | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | HIGH     | 7.5                       | CWE-200: Exposure of Sensitive Information to an Unauthorized Actor<br>Weakness ID: 200<br>Abstraction: Class<br>Structure: Simple |                                                                                                                                                                                                                                                                             | <a href="https://www.openwv.it/cve/cve=CWE-200">https://www.openwv.it/cve/cve=CWE-200</a>                     |
| 18       | MSTG-CODE-2<br>MSTG-CODE-4<br>MSTG-RESILIENCE-9     | Uygulama obfuscated edilmeden yayımlanmış mı?                                       | APKTOOL<br>QARK               | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 10.0                      | CWE-798: Use of Hard-coded Credentials<br>Weakness ID: 798<br>Abstraction: Base<br>Structure: Simple                               | minifyEnabled true                                                                                                                                                                                                                                                          | <a href="https://www.openwv.it/cve/cve=CWE-798">https://www.openwv.it/cve/cve=CWE-798</a>                     |
| 19       | MSTG-PLATFORM-2                                     | Uygulama XSS zafiyetine açık mı?                                                    | APKTOOL                       | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 9.9                       | CWE-693: Protection Mechanism Failure<br>Weakness ID: 693<br>Abstraction: Pillar<br>Structure: Simple                              | webView.loadUrl("javascript:initialize()")                                                                                                                                                                                                                                  | <a href="https://www.openwv.it/cve/cve=CWE-693">https://www.openwv.it/cve/cve=CWE-693</a>                     |
| 20       | MSTG-PLATFORM-1                                     | Uygulamanın MinSdk versionu belli bir threshold değerinden yüksek mi?               | ANDROBUGS<br>ANDROGUARD       | NEGATIVE     | PRE-CALCULATED SCORE<br>BY NVD | CRITICAL | 9.0                       | CWE-919: Weaknesses in Mobile Applications<br>View ID: 919<br>Type: Implicit                                                       |                                                                                                                                                                                                                                                                             | <a href="https://cwe.mitre.org/data/definitions/919.html">https://cwe.mitre.org/data/definitions/919.html</a> |

## EK-1. (devam) MAPTF Kapsam Matrisi

| MAPTF ID | MASSYS SUBSTRACE                                                  | EXPLANATION                                                                                                                                                                                               | TOOL                        | SCORE IMPACT | METHOD                      | Rating   | VULNERABILITY SCORE(CVSS) | CWE                                                                                                                                      | KEYWORD(CONTAIN)                                                                                                                                                             | LINK(CVSS SCORE)                                                                            |
|----------|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------------|-----------------------------|----------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 21       | <a href="#">MSTG-STORAGE-9</a>                                    | Uygulama kapsamında, keywordlerin kullanım tespiti ile zafiyete açık kelimelerin kullanımı yapılmış mı?                                                                                                   | APKTOOL                     | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 10.0                      | CWE-798: Use of Hard-coded Credentials<br>Weakness ID: 798<br>Abstraction: Base<br>Structure: Simple                                     | admin<br>key<br>password<br>etc.                                                                                                                                             | <a href="https://www.openwcc.io/cve/cwe=CWE-798">https://www.openwcc.io/cve/cwe=CWE-798</a> |
| 22       | <a href="#">MSTG-PLATFORM-1</a><br><a href="#">MSTG-STORAGE-9</a> | Uygulama kapsamında, imza bazi izin kullanılmış mı?                                                                                                                                                       | ANDROBUGS<br>ANDROGUARD     | POSITIVE     | PRE-CALCULATED SCORE BY NVD | MEDIUM   | 5.3                       | CWE-732: Incorrect Permission Assignment for Critical Resource<br>Weakness ID: 732<br>Abstraction: Class<br>Structure: Simple            | android.protection.level="signature"                                                                                                                                         | <a href="https://www.openwcc.io/cve/cwe=CWE-732">https://www.openwcc.io/cve/cwe=CWE-732</a> |
| 23       | <a href="#">MSTG-PLATFORM-1</a><br><a href="#">MSTG-STORAGE-9</a> | Uygulamada, content providerlara erişimi izni verilmiş mi?                                                                                                                                                | ANDROBUGS<br>ANDROGUARD     | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | LOW      | 3.5                       | CWE-684: Incorrect Provision of Specified Functionality<br>Weakness ID: 684<br>Abstraction: Class<br>Structure: Simple                   | android.exported="false"                                                                                                                                                     | <a href="https://www.openwcc.io/cve/cwe=CWE-684">https://www.openwcc.io/cve/cwe=CWE-684</a> |
| 24       | <a href="#">MSTG-STORAGE-4</a>                                    | Uygulamalar arasında veri güvenliği şekilde paylaşılmış mı?                                                                                                                                               | ANDROBUGS<br>QARK<br>MOBSF  | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.1                       | CWE-359: Exposure of Private Personal Information to an Unauthorized Actor<br>Weakness ID: 359<br>Abstraction: Base<br>Structure: Simple | use "content://" URLs, not "file://" URLs.                                                                                                                                   | <a href="https://www.openwcc.io/cve/cwe=CWE-359">https://www.openwcc.io/cve/cwe=CWE-359</a> |
| 25       | <a href="#">MSTG-STORAGE-10</a>                                   | Uygulama içerisinde gömülü email adresleri var mı?                                                                                                                                                        | MOBSF<br>APKTOOL            | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.8                       | CWE-312: Cleartext Storage of Sensitive Information<br>Weakness ID: 312<br>Abstraction: Base<br>Structure: Simple                        | mail regex scan                                                                                                                                                              | <a href="https://www.openwcc.io/cve/cwe=CWE-312">https://www.openwcc.io/cve/cwe=CWE-312</a> |
| 26       | <a href="#">MSTG-NETWORK-6</a>                                    | Uygulama kapsamında kullanılan endpointlerin port numaralarının yapılması.                                                                                                                                | NMAP                        | NEUTRAL      | PRE-CALCULATED SCORE BY NVD | HIGH     | 7.4                       | CWE-300: Channel Accessible by Non-Endpoint<br>Weakness ID: 300<br>Abstraction: Class<br>Structure: Simple                               |                                                                                                                                                                              | <a href="https://www.openwcc.io/cve/cwe=CWE-300">https://www.openwcc.io/cve/cwe=CWE-300</a> |
| 27       | <a href="#">MSTG-NETWORK-2</a>                                    | Uygulama, güvenli olmayan kaynaktan sertifika kabul edilmiş mi?                                                                                                                                           | APKTOOL<br>MOBSF            | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.8                       | CWE-295: Improper Certificate Validation<br>Weakness ID: 295<br>Abstraction: Base<br>Structure: Simple                                   | setAllowsAnyHTTPSVerifier(OS)<br>AllowAllHostnameVerifier(Android)                                                                                                           | <a href="https://www.openwcc.io/cve/cwe=CWE-295">https://www.openwcc.io/cve/cwe=CWE-295</a> |
| 28       | <a href="#">MSTG-CODE-2</a><br><a href="#">MSTG-CODE-4</a>        | Uygulamada, debug opsiyonları açık bırakılmış mı?                                                                                                                                                         | APKTOOL<br>ANDROBUGS<br>ADB | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | MEDIUM   | 6.5                       | CWE-117: Improper Output Neutralization for Logs<br>Weakness ID: 117<br>Abstraction: Base<br>Structure: Simple                           | android.allowBackup<br>android.debuggable                                                                                                                                    | <a href="https://www.openwcc.io/cve/cwe=CWE-117">https://www.openwcc.io/cve/cwe=CWE-117</a> |
| 29       | <a href="#">MSTG-STORAGE-9</a>                                    | Uygulama kapsamında, hardcoded olarak saklanan veriler anlamlıdır mı? şekilde belirli keywordler ile password, key, salt, IV gibi mi saklanmış?                                                           | APKTOOL                     | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.8                       | CWE-312: Cleartext Storage of Sensitive Information<br>Weakness ID: 312<br>Abstraction: Base<br>Structure: Simple                        | password regex scan                                                                                                                                                          | <a href="https://www.openwcc.io/cve/cwe=CWE-312">https://www.openwcc.io/cve/cwe=CWE-312</a> |
| 30       | <a href="#">MSTG-AUTH-2</a><br><a href="#">MSTG-AUTH-12</a>       | Uygulamada, endpoint e yapılan isteklerde, headerler extract edilerek içerisindeki api key içermemesi veya güvenli kriptografik yöntemlerle encrypt edilerek saklanması mı?(INFO Potential API Key found) | QARK                        | NEGATIVE     | PRE-CALCULATED SCORE BY NVD | CRITICAL | 9.8                       | CWE-319: Cleartext Transmission of Sensitive Information<br>Weakness ID: 319<br>Abstraction: Base<br>Structure: Simple                   | builder.addHeader("Authorization", "Bearer "+ token)<br>@Header("Authorization: MY_API_KEY")<br>@Header("api_key") String api_key<br>@Header("secret_key") String secret_key | <a href="https://www.openwcc.io/cve/cwe=CWE-319">https://www.openwcc.io/cve/cwe=CWE-319</a> |



## EK-1. (devam) MAPTF Kapsam Matrisi

| MAPTF ID | MASVS SUBSTRACE                      | EXPLANATION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | TOOL                       | SCORE IMPACT | METHOD                                          | Rating   | VULNERABILITY SCORE(CVSS) | CWE                                                                                                                                | KEYWORD(CONTAIN)                          | LINK(CVSS SCORE)                                                                                                                                                                                                                              |
|----------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|--------------|-------------------------------------------------|----------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 41       | <a href="#">NO MATCHED SUBSTRACE</a> | Uygulama, mobil ağ veritabanını kullanarak cihazın yaklaşıp lokasyonuna erişim sağlıyor mu?                                                                                                                                                                                                                                                                                                                                                                                                                                | MOBSF                      | NEGATIVE     | PRE-CALCULATED SCORE BY NVD                     | CRITICAL | 9.8                       | CWE-922: Insecure Storage of Sensitive Information<br>Weakness ID: 922<br>Abstraction: Class<br>Structure: Simple                  | android.permission.ACCESS_COARSE_LOCATION | <a href="https://www.openview.id/cve/cwe=CWE-922">https://www.openview.id/cve/cwe=CWE-922</a>                                                                                                                                                 |
| 42       | <a href="#">NO MATCHED SUBSTRACE</a> | Uygulama kullanıcının kişisel profil verisini kullanıyor mu?                                                                                                                                                                                                                                                                                                                                                                                                                                                               | ANDROGUARD<br>MOBSF        | NEGATIVE     | SELF-CALCULATED SCORE BY ONLINE CVSS CALCULATOR | MEDIUM   | 4.0                       | CWE-926: Improper Export of Android Application Components<br>Weakness ID: 926<br>Abstraction: Variant<br>Structure: Simple        | android.permission.READ_PROFILE           | <a href="https://nvd.nist.gov/vuln:metrics/cvss/v3.1:calculator/?vector=AV:L/AC:L/PR:NI/EN:SU/CL:L/NA:NA&amp;version=3.1">https://nvd.nist.gov/vuln:metrics/cvss/v3.1:calculator/?vector=AV:L/AC:L/PR:NI/EN:SU/CL:L/NA:NA&amp;version=3.1</a> |
| 43       | <a href="#">NO MATCHED SUBSTRACE</a> | Uygulama, takip(tracker) sistemi kullanıyor mu?(Google Admob, Google Analytics, Google Tag Manager)                                                                                                                                                                                                                                                                                                                                                                                                                        | ANDROBUGS<br>QARK<br>MOBSF | NEGATIVE     | PRE-CALCULATED SCORE BY NVD                     | HIGH     | 7.5                       | CWE-200: Exposure of Sensitive Information to an Unauthorized Actor<br>Weakness ID: 200<br>Abstraction: Class<br>Structure: Simple |                                           | <a href="https://www.openview.id/cve/cwe=CWE-200">https://www.openview.id/cve/cwe=CWE-200</a>                                                                                                                                                 |
| 44       | <a href="#">NO MATCHED SUBSTRACE</a> | Uygulama, receiverPermission izni olmadan broadcast paylaşım yapmıy mı(Broadcast send without receiverPermission)                                                                                                                                                                                                                                                                                                                                                                                                          | ANDROBUGS<br>QARK          | NEGATIVE     | PRE-CALCULATED SCORE BY NVD                     | MEDIUM   | 6.1                       | CWE-925: Improper Verification of Intent by Broadcast Receiver<br>Weakness ID: 925<br>Abstraction: Variant<br>Structure: Simple    | android.permission.READ_PHONE_STATE       | <a href="https://nvd.nist.gov/vuln:metrics/cvss/v3.1:calculator/?vector=AV:L/AC:L/PR:NI/EN:SU/CL:L/NA:NA&amp;version=3.1">https://nvd.nist.gov/vuln:metrics/cvss/v3.1:calculator/?vector=AV:L/AC:L/PR:NI/EN:SU/CL:L/NA:NA&amp;version=3.1</a> |
| 45       | <a href="#">NO MATCHED SUBSTRACE</a> | Uygulamada intent gereksinimi uygun olmayan, Intent.setClass() içinde bir method kullanılarak mı sağlanıyor? (For security reasons, the Intent objects you supply here should almost always be explicit intents, that is specify an explicit component to be delivered to through Intent.setClass. A malicious application could potentially intercept, redirect and/or modify this Intent. Pending Intents retain the UID of your application and all related permissions, allowing another application to act as yours.) | QARK                       | NEGATIVE     | PRE-CALCULATED SCORE BY NVD                     | MEDIUM   | 4.6                       | CWE-927: Use of Implicit Intent for Sensitive Communication<br>Weakness ID: 927<br>Abstraction: Variant<br>Structure: Simple       |                                           | <a href="https://nvd.nist.gov/vuln/detail/CVE-2001-0984">https://nvd.nist.gov/vuln/detail/CVE-2001-0984</a>                                                                                                                                   |



*Gazili olmak ayrıcalıktır*