OPEN ACCESS
LEDGER, VOL. 1, NO.2, MAY 2022

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

LEDGER: Journal Informatic and Information Technology

Implementasi Static Analysis Dan Background Process Untuk Mendeteksi Malware Pada Aplikasi Android Dengan Mobile Security Framework

Aris Rafael Tambunan[1], Trihastuti Yuniati S.Kom., M.T[2], Yoso Adi Setyoko S.T., M.T[3]

[1]*Teknik Informatika, Institut Teknologi Telkom Purwokerto, Indonesia*
Jl. DI. Panjaitan No. 128 Purwokerto, Indonesia,
*Penulis Korespondensi: trihastuti@ittelkom-pwt.ac.id

Abstract

Android is the most widely used operating system among the many choices of operating systems for mobile devices with support by various applications to facilitate human life in their activities. Mobile Security Framework (MobSF) is an open-source automated testing framework capable of performing penetration tests, malware analysis, mobile application dynamic services with statistical analysis and in carrying out the analysis process will display the results of the android application. The purpose of this study is how to use the Mobile Security Framework (MobSF) as an analysis of static malware on android applications. Malware samples will be taken via the internet and analyze the malware using a static analysis method that reads malware information and then combined it with android background processes by installing directly on the android device. Researchers used Mobile security framework (MobSF) to analyze static security with parameters of malicious permissions, weak crypto, root detection, SSL bypass and domain malware check in X8Speeder application. The results of the analysis, detected Malware and has improper permissions. As seen, the game application does not require permission to modify the external storage of the smartphone. Therefore, it requires user accuracy so as not to rush or click something when surfing the Internet

Keywords: Mobile Security Framework, Static Analysis, Backgrund Proces Android, Android

Abstrak

Android menjadi sistem operasi yang paling banyak digunakan di antara sekian banyak pilihan sistem operasi untuk perangkat mobile dengan dukungan oleh aplikasi yang beragam guna memudahkan kehidupan manusia dalam beraktifitas Mobile Security Framework (MobSF) adalah framework pengujian otomatis bersifat open-source, yang mampu melakukan uji penetrasi, analisis malware, penilaian keamanan aplikasi seluler dengan analisis statis dan dinamis dalam melakukan proses analisis akan menampilkan hasil berupa laporan mengenai aplikasi android tersebut. Tujuan Penelitian ini adalaha bagaimana menggunakan Mobile Security Framwork (MobSF) sebagai analisis static malware pada aplikasi android. Sample malware akan diambil melalui internet dan menganalisis malware tersebut menggunakan metode analisis statik yang membaca informasi malware lalu dikombinasikan dengan back ground process android dengan menginstal live pada device android. Peneliti menggunakan Mobile security framework (MobSF) untuk menganalisis statik keamanan dengan parameter dangerous permissions, weak crypto, root detection, SSL bypass dan domain malware check pada aplikasi X8Speeder. Hasil dari analisis, terdeteksi Malware dan memiliki perizinan yang tidak sewajarnya. Seperti yang terlihat, aplikasi game tidak memerlukan izin untuk memodifikasi penyimpanan eksternal smartphone. Maka dari itu, diperlukannya ketelitian pengguna agar tidak sembarangan mengunduh atau mengklik sesuatu ketika berinternet.

Kata Kunci: Mobile Security Framework, Static Analysis, Backgrund Proces Android, Android

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

I. PENDAHULUAN

Seiring perkembangan teknologi informasi yang semakin maju, penggunan mobile phone yang berjenis smartphone atau ponsel pintar sangat berkembang pesat. Ada berbagai jenis sistem operasi yang digunakan smartphone seperti, IOS, Windows Phone, dan Android. Android menjadi sistem operasi yang paling banyak digunakan di antara sekian banyak pilihan sistem operasi untuk perangkat mobile dengan dukungan oleh aplikasi yang beragam guna memudahkan kehidupan manusia dalam beraktifitas. Namun dari sekian banyak aplikasi yang tersedia pada android, beberapa diantaranya tanpa disadari dapat membahayakan penguna smartpone sendiri, beberapa contoh yang dapat merugikan diantaranya aplikasi antivirus palsu, aplikasi game palsu, aplikasi aplikasi e-commerce palsu, aplikasi tiruan, dan lain sebagainya tersebar di internet. Alasan mengapa aplikasi-aplikasi tersebut dikategorikan berbahaya karena disusupi malware yang dapat mencuri, memata-matai, mensadap dan bahkan merusak data.

Malware atau Malicious Software adalah perangkat lunak yang dapat menyusup ke sistem operasi sehingga dapat merusak sistem operasi, memanfaatkan sumber daya tanpa sepengetahuan pemilik perangkat, bahkan mengumpulkan informasi pribadi untuk dibagikan ke pihak ketiga tanpa persetujuan pengguna. Malware-malware baru terus bermunculan seiring dengan perkembangan teknologi, baik dari segi platform maupun sistem operasi, dengan memanfaatkan celah keamanan dan kelalaian pengguna. Berkembangnya teknologi pun memicu dikembangkannya malware-malware baru, sehingga semakin banyak pihak yang dirugikan karena adanya malware-malware ini. Bahkan saat ini, malware telah dapat menjangkit hampir seluruh jenis sistem operasi. [1]. Sistem operasi android adalah sistem operasi yang berbasis linux untuk telepon seluler seperti smartphone dan tablet PC. Sistem android memiliki keunggulan, seperti sistem operasi bersifat open source, multitasking, kemudahan dalam notifikasi hingga banyaknya aplikasi atau software yang dapat dinikmati dengan menggunakan sistem android. Akan tetapi, salah satu keunggulan sistem android menjadi salah satu kelemahannya. Keunggulan sistem operasi bersifat open access, dimana disediakan platform terbuka bagi para pengembang (user) yang dimaksudkan agar user dapat menciptakan dan mengembangkan aplikasi mereka sendiri sehingga dapat digunakan pada bermacam perangkat seluler. Akan tetapi, hal ini malah menimbulkan kemudahan dalam pihak yang tidak bertanggung jawab untuk membangun dan mengembangkan malware menjadi aplikasi yang dapat masuk ke sistem android.

Mobile Security Framework (MobSF) adalah framework pengujian otomatis bersifat open-source, yang mampu melakukan uji penetrasi, analisis malware, penilaian keamanan aplikasi seluler dengan analisis statis dan dinamis dalam melakukan proses analisis akan menampilkan hasil berupa laporan mengenai aplikasi android tersebut. Pada penelitian [2] yang pernah dilakukan sebelumnya dengan judul "Analisis Dan Deteksi Malware Dengan Metode Hybrid Analysis Menggunakan Framework Mobsf" menggunakan metode hybrid analysis yakni gabungan dari static analysis dan dynamic analysis pada MobSF dengan sampel malware Bouncing Golf dan Riltok. Peneliti menganalisa behavior kedua malware dan mendaptkan hasil Bouncing Golf melakukan pemantauan dan merekam aktivitas yang dilakukan pengguna smartphone android. Sedangkan malware Riltok memiliki karakteristik dalam mencuri data kartu kredit dengan menampilkan halaman billing palsu. Berdasarkan permasalahan yang telah diuraikan, maka pada penelitian ini penulis akan buat sebuah sistem keamanan aplikasi android dengan menggunakan Mobile Security Framework (MobSF) untuk dapat mengetahui adanya malware berbahaya pada aplikasi android. Dengan menggunakan Mobile Security Framework (MobSF) akan dilakukan static analysis dengan parameter analisis dangerous permissions, weak crypto, root detection, ssl bypass, domain malware check dikombinasikan dengan background proses pada beberapa aplikasi android yang ada di internet dan sampel yang sudah mengandung malware, yang nantinya akan dilakukan analisa behavior dari malware tersebut lebih lanjut mengenai hasilnya yang berupa laporan dan source code program.

II. TINJAUAN PUSTAKA

Penelitian mengenai analisis dan deteksi malware aplikasi android bukan yang pertama kali dilakukan, namun terdapat beberapa hal yang akan dikembangkan pada penelitian ini. Dalam bab ini, penulis menjabarkan penelitian – penelitian terkait dengan penelitian ini.yang akan menjadi referensi. Edward Tansen, Deris Wahyu Nurdiarto dengan judul "Analisis Dan Deteksi Malware Dengan Metode Hybrid

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

Analysis Menggunakan Framework Mobsf" membahas tentang malware Bouncing Golf dan Riltok pada android, Informasi diperoleh dari Virus Total yaitu sampel malware Bouncing Golf memiliki nilai checksum SHA256 atau Secure Hash Algorithm 55123ed4982fa1 35dbeda49969ab68444125143e36930fe1612d367f2fa615fc. Deteksi dengan rasio dari 61 antimalware dan hasil 24 antimalware dapat mendeteksi sampel malware Bouncing Golf File sampel malware berukuran 14.07 MB dan tipe file APK. Informasi didapatkan dari VirusTotal yaitu sampel malware Riltok memiliki nilai checksum SHA256 atau Secure Hash Algorithm 0497b6000a7a23e9e9b97472bc2d3799caf49cbbea1627ad4d87ae6e0b7e2a98. Deteksi dengan rasio dari 61 antimalware dan hasil 40 antimalware dapat mendeteksi sampel malware Riltok. File sampel malware berukuran 992.55 KB dan tipe file APK. Kedua malware dilakukan static analysis dan dynamic analysis dengan hasil analisis sampel malware Bouncing Golf dan Riltok menggunakan MobSF dengan metode static analysis dan dynamic analysis menggunakan virtual lab (emulator) android dari Genymotion dengan versi android 8.1 dan API level 27 (Oreo) telah suskses mendapatkan beberapa karakteristik dari kedua malware. Bouncing Golf melakukan pemantauan dan merekam aktivitas yang dilakukan pengguna smartphone android. Sedangkan malware Riltok memiliki karakteristik dalam mencuri data kartu kredit dengan menampilkan halaman billing palsu. Kedua malware memiliki karakteristik yang sama, yaitu data akan dikirimkan ke server C&C(Command & Control Server). [2] Aan Kartono, Anang Sularsa, Setia Juli Irzal Ismail dengan judul "Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan Mobsf" membahas beberapa malware pada android yaitu Blackmart versi 0.99 dengan keterangan Aplikasi penyedia aplikasi android, Krep.imtd.ywtjexf versi 3.0 yakni Aplikasi bank palsu, Earthquake versi 1.1.5 Aplikasi pendeteksi gempa. Hasil pengujian dengan Framework MobSF dari ketiga sampel tersebut adalah sebagai berikut. Blackmart mendapakan function dan beberapa kode yang dicurigakan, mendapatkan sebuah alamat atau url saat membuka aplikasi yaitu "https://api.airpush.com/inappads/inappadcall.php", serta hasil scan virustotal 44% terdeteksi sebagai malware. Krep.imtd.ywtjexf mendapakan function dan beberapa kode yang dicurigakan, mendapatkan sebuah alamat atau url berbentuk phising yaitu "https://play.googleapis.com/play/log", hasil scan virustotal 61% terdeteksi sebagai malware. Earthquake mendapakan function dan beberapa kode yang dicurigakan, mendapatkan sebuah alamat atau url saat membuka aplikasi yaitu "http://pari.securedapinetworks.com/api/input.php?type=Master&data=" hasil scan virustotal 0% terdeteksi sebagai malware. [1][2][3] Achmad Farhan Febrianto, Avon Budiono, S.T., M.T., Ahmad Almaarif., S.Kom., M.T. dengan judul "Analisis Malware Pada Sistem Operasi Android Menggunakan Metode Traffic Analysis".

Mobile Security Framework

Mobile Security Framework (MobSF) adalah framework pengujian otomatis bersifat open-source, yang mampu melakukan uji penetrasi, analisis malware, penilaian keamanan aplikasi seluler dengan analisis statis dan dinamis dalam melakukan proses analisis akan menampilkan hasil berupa laporan mengenai aplikasi android tersebut. MobSF dapat digunakan untuk analisis keamanan yang efektif dan cepat dari aplikasi seluler Android, iOS dan Windows dan mendukung binari (APK, IPA & APPX) dan kode sumber zip. MobSF dapat melakukan pengujian aplikasi dinamis saat runtime untuk aplikasi Android dan memiliki kemampuan fuzzing API Web yang didukung oleh CapFuzz, pemindai keamanan khusus API Web. MobSF memiliki UI grafis dalam bentuk layanan web. Layanan web terdiri dari dasbor yang menyajikan hasil analisis, situs dokumentasinya sendiri, emulator terintegrasi & API yang memungkinkan pengguna memicu analisis secara otomatis. MobSF dihosting di server lokal, data sensitif tidak pernah berinteraksi dengan cloud.[6].

Malware

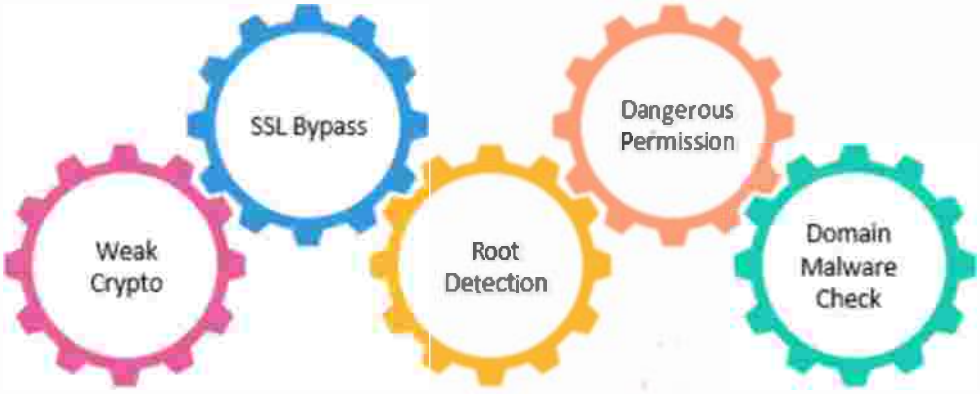
Malware atau Malicious Software merupakan sebuah program atau software jahat yang sengaja dibuat dengan tujuan tertentu oleh pembuatnya yang dapat mengakibatkan berbagai kerugian pada pengguna yang menjadi sasaran dari pembuat malware tersebut. Kata malware merupakan istilah umum yang digunakan

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

untuk software maupun program yang dirancang bertujuan untuk menyusup atau merusak sebuah sistem secara diam-diam [3].

Static Analyss

Metode static analysis dilakukan tanpa benar-benar menjalankan programnya dan lebih seperti menyelidiki apa yang terjadi pada source code dengan tujuan utama yaitu untuk mengetahui kode berbahaya seperti apa yang tertanam dalam aplikasi tersebut. [10] MobSF dapat melakukan static analysis, mulai dari certificate information, application permissions, apkid anlysis, browseble activity, network security, manifest analysis, code analysis, Niap Analysis, App Security Score Calculation, and Risk Calculation. Analisis statis mengacu pada dekompilasi APK aplikasi ke file Java dan XML yang sesuai. Untuk melakukan analisis statis, penganalisis statis harus memiliki fitur untuk memeriksa XML dengan benar dan presisi.



Gambar 2. 1 Parameter Static Analysis

Dari hasil analisis statis maka didapat hasil daftar malwords yang sering muncul sebagai daftar string berbahaya dengan tingkat resiko masing-masing, setelah mendapatkan daftar malwords, peneliti akan menganalisisnya [7]. Gambar 2.1. menunjukkan parameter pada analisis static.

Android Background Process

Android background process adalah bagian penting dalam aplikasi Android, yang responsif bagi pengguna Anda dan ramah pengguna di platform Android yang dimana untuk mengetahui proses yang berjalan di belakang layar pengguna smartphone android. Untuk mengetahui background task pada android dapat menggunakan aplikasi Advance Task Manager yang akan di lakukan penulis untuk menganalisa proses background task pada android. Adanya aplikasi yang berjalan di balik layar sebenarnya bukan dikonotasikan Negatif seperti virus, spyware atau konotasi miring lainnya. Program yang berjalan di balik layar yang notabene tidak ditampakkan oleh pengembang aplikasi maupun sistem operasi itu sendiri bertujuan untuk menyembunyikan

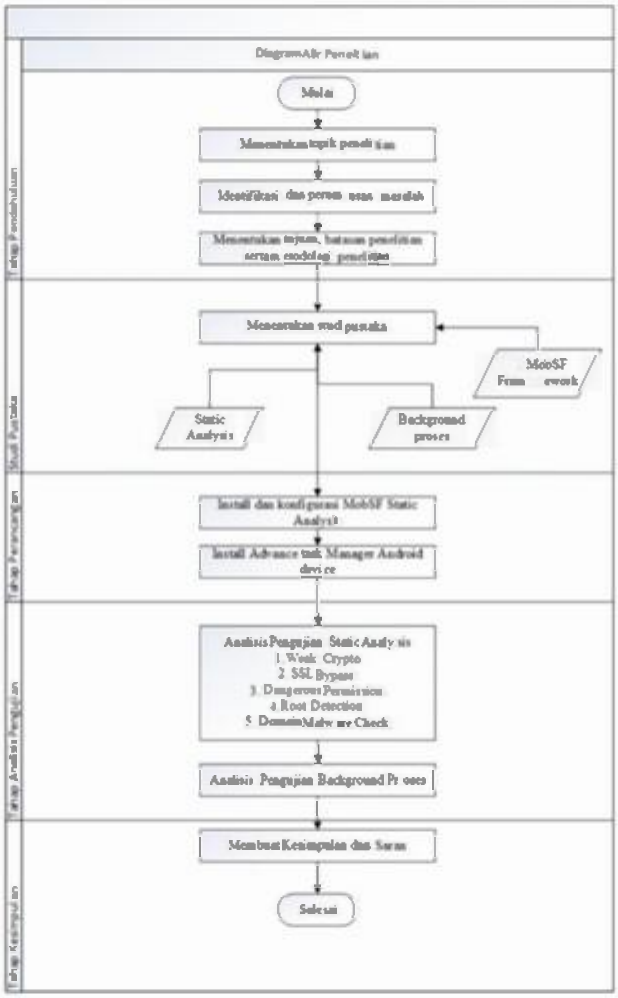
III. METODE PENELITIAN

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

Metodologi yang digunakan pada penelitian ini adalah kuantitatif dengan uji analisis dengan uji analisis aplikasi speedhack game menggunakan Mobile Security Framework (MobSF) dan obyek penelitian yaitu keamanan aplikasi speed hack game X8Speeder dengan tujuan penelitian ini mendukung sistem keamanan pada android. Kriteria yang akan dijadikan sebagai parameter penelitian adalah dangerous permissions, weak crypto, root detection, SSL bypass dan domain malware check.

A. Diagram Alir Penelitian

Proses penelitian pada perancangan analisis deteksi malware ini memiliki beberapa tahapan, yaitu dengan pengerjaan dari satu sistem dibuat secara berurutan dengan diawali pendahauluan, studi pustaka, tahap perancangan, tahap analisis pengujian, serta tahap kesimpulan. Alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

1. Tahap pendahuluan, mengidentifikasi masalah yang dibahas dalam penelitian yaitu analisi deteksi malware dengan mengkombinasikan metode static analysis dengan background proses.
2. Studi Pustaka. Memperoleh dokumentasi penelitian-penelitian sebelumnya dari berbagai sumber yang tersedia seperti jurnal, skripsi, buku, website, maupun sumber lain yang memiliki keterkaitan dengan permasalahan yang dihadapi.
3. Tahap perancangan. Tahap ini peneliti Model perancangan sistem menjelaskan mengenai cara kerja sistem secara terstuktur.

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

4. Tahap Analisis Pengujian. Pada tahap pengujian dilakukan dua tahap, yang pertama yaitu seperti pada menggunakan metode analisis statis dengan Mobile Security Framework (MobSF), lalu dilanjutkan dengan menganalisa Back_ground proses.
5. Kesimpulan. Pada tahap ini penulis mengevaluasi keseluruhan tahapan penelitian dan mendok umentasi.kan dalam bentuk laporan.

B. Blok Diagram

Input: meng-upload file aplikasi speedhack game dengan format APK ke mobile security framework, tunggu hingga proses upload selesai. Proses: mobile security framework akan melak ukan proses analisis statik dengan parameter yaitu dangerous permissions, weak crypto, root detection, ssl bypass dan domain malware check. Output: saat proses telah selesai, maka akan muncul hasil pengujian file tersebut yang berupa laporan dan skor keamanan.

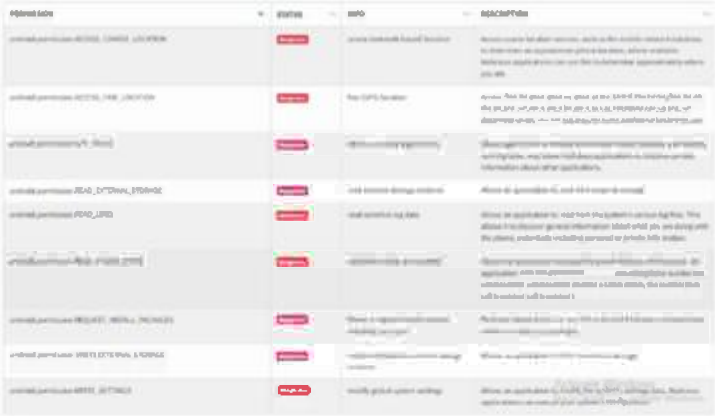
Gambar 3.2 Blok Diagram

IV. Hasil dan Pembahasan

A. Hasil Static Analysis

1. 1. *Dangerous Permission*

Pada gambar 4.2 yang menunjukkan halaman analisis file aplikasi X8Speeder pada mobile security framework (MobSF) bagian application permission



Gambar 4. 2 Halaman Analisis X8Speeder Bagian Application Permission

Pada aplikasi X8Speeder terdapat 9 dangerous permission yaitu:

1.android.permissionACCESS_COARSE_LOCATION yang mengizinkan mengakses perkiraan kasar dengan tingkat akurat dalam jarak 3 kilometer persegi atau sekitar 1.2 mil persegi.

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

2. android.permission.ACCES_FINE_LOCATION yang mengizinkan mengakses lokasi akurat biasanya berada dalam jarak sekitar 50 meter (160 kaki) dan terkadang akurat seperti dalam beberapa meter (10 kaki) atau yang lebih baik.

3. android.permission.GET_TASK yang mengizinkan aplikasi mengakses mengambil informasi tentang tugas yang sedang dan baru berjalan. Dapat mengizinkan aplikasi berbahaya untuk menemukan informasi pribadi tentang aplikasi lain

4. android.permission.READ_EXTERNAL_STORAGE yang mengizinkan aplikasi membaca penyimpanan eksternal.

5. android.permission.READ_LOGS yang mengizinkan aplikasi membaca dari berbagai file log sistem. Ini memungkinkannya untuk menemukan informasi umum tentang apa yang dilakukan dengan smartphone, kemungkinan termasuk informasi pribadi.

6. android.permission.READ_PHONE_STATE yang mengizinkan aplikasi mengakses fitur telepon perangkat. Aplikasi dengan izin ini dapat menentukan nomor telepon dan nomor seri telepon ini, apakah panggilan aktif, nomor yang terhubung ke panggilan dan sebagainya.

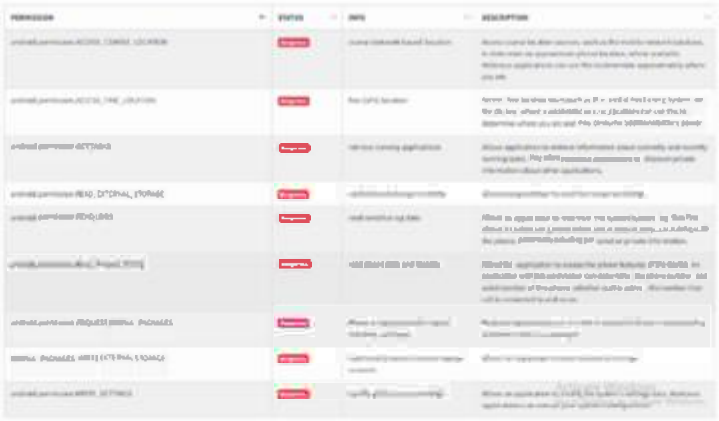
7. android.permission.REQUEST_INSTALL_PACKAGES yang mengizinkan aplikasi Aplikasi berbahaya dapat menggunakan ini untuk mencoba dan mengelabui pengguna agar menginstal paket berbahaya tambahan.

8. android.permission.WRITE_EXTERNAL_STORAGE yang mengizinkan aplikasi membaca memodifikasi dana tau menghapus ke penyimpanan eksternal.

9. android.permission.WRITE_SETTINGS yang mengizinkan aplikasi mengubah data setelan sistem. Aplikasi berbahaya dapat merusak konfigurasi sistem android.

1.2. *Weak Crypto*

Pada gambar 4.2 yang menunjukkan halaman analisis file aplikasi X8Speeder pada mobile security framework (MobSF) bagian application permission



Gambar 4. 2 Halaman Analisis X8Speeder Bagian *Weak Crypto*

Pada aplikasi X8Speeder terdapat 3 *weak crypto* yang terdiri dari 1 high severity yaitu aplikasi menggunakan mode enkripsi ECB dalam algoritma enkripsi kriptografi. Mode ECB diketahui lemah karena menghasilkan ciphertext yang sama untuk blok plaintext yang identik. Dengan standar:

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

1. *Common Weakness Enumeration (CWE)* adalah daftar yang menampilkan keberadaan bug pada software atau hardware yang berbahaya, terdeteksi CWE-327: Penggunaan algoritma kriptografi rusak atau berisiko

2. *Open Web Application Security Project (OWASP)* Top 10 adalah sebuah panduan bagi para developers dan security team tentang kelemahan-kelemahan pada web apps yang mudah diserang dan harus segera disiasati, terdeteksi M5: Kriptografi tidak memadai.

3. *OWASP The Mobile Application Security Verification Standard (MASVS)* adalah standar untuk keamanan aplikasi seluler, terdeteksi MSTGCRYPTO-2 artinya aplikasi menggunakan primitif kriptografi yang sesuai untuk kasus penggunaan tertentu, serta dikonfigurasi dengan parameter yang mematuhi praktik terbaik industri.

Lokasi file yaitu com/bytedance/sdk/openadsdk/core/a.java. Gambar 4.x menunjukkan kode pada com/bytedance/sdk/openadsdk/core/a.java.

```
public class a {
    public static String a(String str, String str2) {
        try {
            SecretKeySpec secretKeySpec = new SecretKeySpec(str2.getBytes(), "AES");
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(1, secretKeySpec);
            return Base64.encodeToString(cipher.doFinal(str.getBytes("utf-8")), 0);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
```

Gambar 4.4 Kode pada com/bytedance/sdk/openadsdk/core/a.java

Kode tersebut artinya membuat Cipher chiper dengan memanggil getInstance() metodenya dengan parameter yang diisi dengan jenis algoritma enkripsi yang digunakan yaitu AES dengan mode operasi ECB dan padding PKCS5. Terlebih dahulu masukkan kelas java yaitu javax.crypto. Cipher untuk membuat cipher. Advanced Encryption Standard (AES) merupakan algoritma yang menggunakan kunci yang sama (private key) untuk enkripsi dan dekripsi. Electronic Code Book (ECB) adalah suatu mode blok plainteks di-enkripsi secara bersamaan dan setiap blok plainteks di-enkripsi menggunakan kunci yang sama. Public Key Cryptographic Standard (PKCS) adalah penambahan data pada awal, tengah, atau akhir sebelum enkripsi, nilai tiap bit yang ditambahkan adalah banyak bit yang ditambahkan.

Pada aplikasi X8Speeder terdepat 3 warning severity yaitu:

1. MD5 adalah hash lemah yang diketahui memiliki tabrakan hash. Tabrakan hash adalah ada 2 atau lebih teks yang menghasilkan nilai hash yang sama. MD5 yang digunakan untuk password juga rentan terhadap serangan brute-force dengan waktu yang cepat karena hanya memiliki 128 bit. Serangan brute-force adalah serangan dengan mencoba segala kombinasi huruf, angka dan simbol agar didapatkan plaintext dari suatu hash. Dengan detail:

- Common Weakness Enumeration (CWE) terdeteksi CWE-327: penggunaan algoritma kriptografi rusak atau berisiko.
- Open Web Application Security Project (OWASP) Top 10 terdeteksi M5: kriptografi tidak memadai.
- The Mobile Application Security Verification Standard terdeteksi MSTG-CRYPTO-4 artinya aplikasi tidak menggunakan protokol atau algoritma kriptografi yang secara luas dianggap tidak digunakan lagi untuk tujuan keamanan.

Lokasi file yaitu pada:

ISSN : XXXX-XXXX (ONLINE)
DOI : 10.20895/LEDGER.V1I2.848

com/bytedance/bdtracker/ih.java
com/bytedance/bdtracker/j.java
com/bytedance/bdtracker/mm.java
com/bytedance/bdtracker/qf.java
com/bytedance/bdtracker/qj.java
com/bytedance/bdtracker/rv.java
com/bytedance/bdtracker/sd.java
com/bytedance/bdtracker/sh.java
com/bytedance/sdk/openadsdk/utils/j.java
com/tencent/mid/util/Util.java
com/tencent/stat/common/StatCommonHelper.java
com/x8zs/download/a.java
com/x8zs/download/i.java
com/x8zs/plugin/utils/MiscUtil.java
com/x8zs/plugin/utils/PkgUtil.java

```
private static MessageDigest a() {
    try {
        return MessageDigest.getInstance("md5");
    } catch (NoSuchAlgorithmException e) {
        return null;
    }
}
```

Gambar 4.5 Kode File MD5 pada Aplikasi X8Speeder

Kode MessageDigest.getInstance("md5") berarti string akan dienkripsi menggunakan MD5. Message digest adalah sebuah nilai yang dikenal juga sebagai kriptografi checksum atau secure hash. Message digest dimaksudkan untuk meningkatkan keamanan dalam transformasi data, kelas yang sering digunakan dalam aplikasi yang membutuhkan autentikasi user melalui password. Checksum adalah urutan angka dan huruf yang digunakan untuk memeriksa kesalahan data. Secure hash adalah fungsi hash yang bekerja satu arah, ini berarti pesan yang sudah diubah menjadi message digest tidak dapat dikembalikan menjadi pesan semula.

2. SHA-1 adalah hash lemah yang diketahui memiliki tabrakan hash. Tabrakan hash adalah ada 2 atau lebih teks yang menghasilkan nilai hash yang sama. SHA-1 yang digunakan untuk password juga rentan terhadap serangan brute-force dengan waktu yang cepat karena hanya memiliki 160 bit. Serangan brute-force adalah serangan dengan mencoba segala kombinasi huruf, angka dan simbol agar didapatkan plaintext dari suatu hash. Dengan standar:

- Common Weakness Enumeration (CWE) terdeteksi CWE-327: penggunaan algoritma kriptografi rusak atau berisiko.
- Open Web Application Security Project (OWASP) Top 10 terdeteksi M5: kriptografi tidak memadai.
- The Mobile Application Security Verification Standard terdeteksi MSTG-CRYPTO-4 artinya aplikasi tidak menggunakan protokol atau algoritme kriptografi yang secara luas dianggap tidak digunakan lagi untuk tujuan keamanan

1.3 *Root Detection*

Root detection adalah deteksi akses root terhadap perangkat android yang digunakan. Pada handphone yang sudah di-root membuat handphone rentan terhadap serangan spyware atau virus dan memungkinkan pengguna untuk memodifikasi aplikasi sehingga menyebabkan aplikasi rusak atau rentan terhadap kecurangan. [7] Pada aplikasi X8Speeder tidak memiliki kemampuan root detection

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

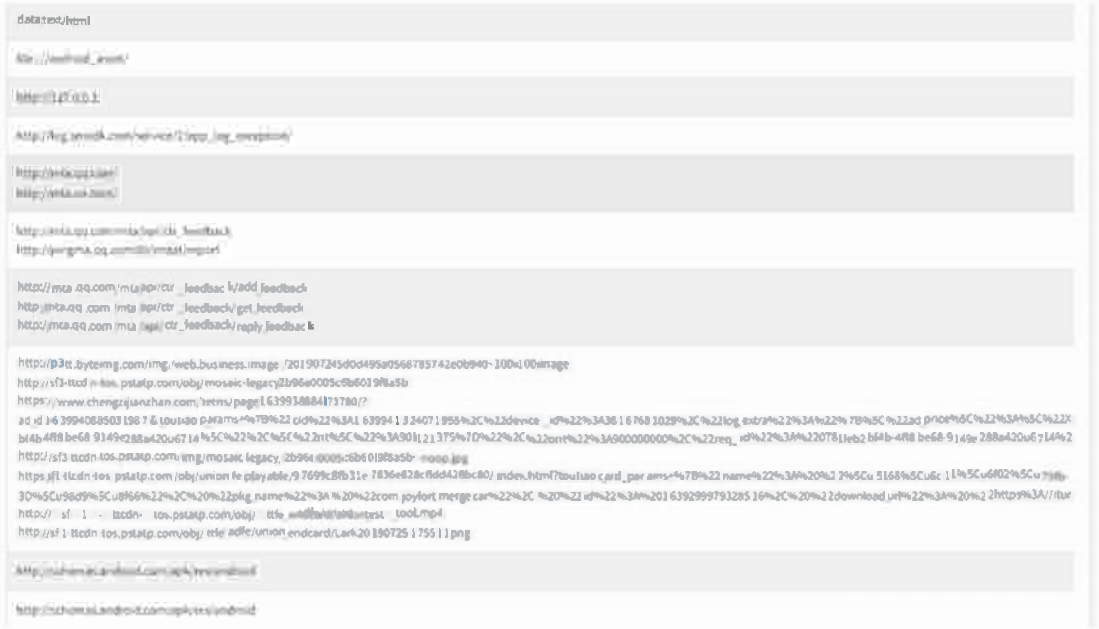
karena pada bagian code analysis tidak terdapat file atau kode tentang root detection seperti pada gambar 4.6

NO	ISSUE	SEVERITY	STANDARDS	FILES
12	This App may have root detection capabilities.	secure	OWASP MASVS: MSTG-RESILIENCE-1	com/bytedance/sdk/openadsdk/core/o.java

Gambar 4.6 Analisis File Aplikasi X8Speder Root Detection

1.4 SSL Bypass

Pada gambar 4.7 menunjukkan halaman hasil analisis file aplikasi X8Speeder pada mobile security framework (MobSF) bagian URLs.



Gambar 4.7 Analisis File Aplikasi X8Speeder Bagian URLs

Pada aplikasi X8Speeder terdapat url dengan protokol jaringan Hypertext Transfer Protocol (HTTP) dan Hypertext Transfer Protocol Secure (HTTPS). HTTPS adalah protokol jaringan dengan SSL. Secure Socket Layers (SSL) adalah lapisan keamanan tambahan untuk melindungi komunikasi data antara client dan server. Pada gambar 4.7 yang menggunakan protokol jaringan http, sebaiknya menggunakan HTTPS yang terdapat SSL. Pada bagian code analysis menunjukkan 2 *warning SSL Certificate*

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

8	Insecure Implementation of SSL. Trusting all the certificates or accepting self signed certificates is a critical Security Hole. This application is vulnerable to MITM attacks	High	**CWE:** CWE-295: Improper Certificate Validation **OWASP Top 10:** M3: Insecure Communication **OWASP MASVS:** MSTG-NETWORK-3	com/bytedance/bdtracker/bc.java com/bytedance/bdtracker/ex.java
9	Insecure WebView Implementation. WebView ignores SSL Certificate errors and accept any SSL Certificate. This application is vulnerable to MITM attacks	High	**CWE:** CWE-295: Improper Certificate Validation **OWASP Top 10:** M3: Insecure Communication **OWASP MASVS:** MSTG-NETWORK-3	com/bytedance/sdk/openadsdk/core/widget/webview/c.java

Gambar 4.8 Analisis SSL Certificate X8Speeder

1. Insecure Implementasi SSL Certificate yaitu mempercayai semua sertifikat atau menerima sertifikat yang ditandatangani sendiri adalah celah eamanan yang penting. Aplikasi ini rentan terhadap serangan MITM. Dengan standar:

- CWE-295: Improper Certificate Validation artinya Perangkat lunak tidak memvalidasi, atau salah memvalidasi, sertifikat. Ketika sertifikat tidak valid atau berbahaya, mungkin memungkinkan penyerang untuk menipu entitas tepercaya dengan mengganggu jalur komunikasi antara host dan klien. Aplikasi mungkin terhubung ke host jahat sambil meyakini bahwa ini adalah host tepercaya, atau perangkat lunak mungkin ditipu untuk menerima data palsu yang tampaknya berasal dari host tepercaya.

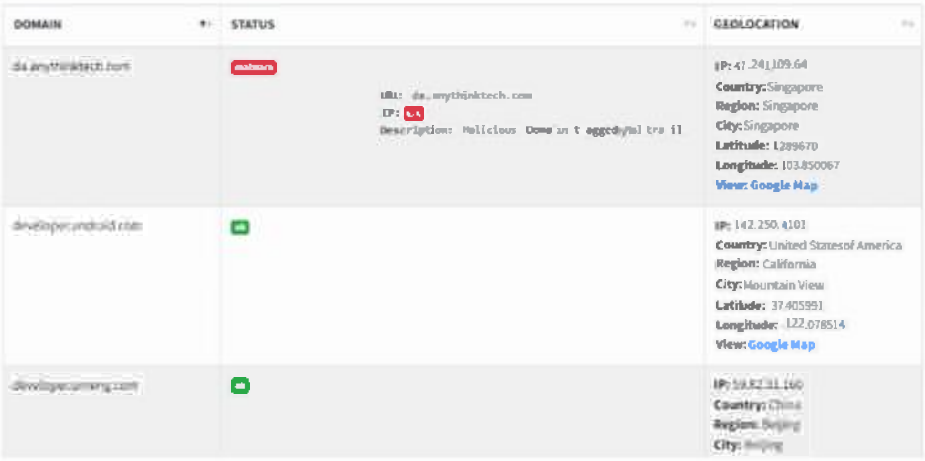
- OWASP Top 10: M3: Insecure Communication artinya risiko ini mencakup semua teknologi komunikasi yang mungkin digunakan perangkat seluler: TCP/IP, wifi, bluetooth, NFC, audio, infrared, GSM, 3G, SMS, dan lainnya.

- OWASP MASVS: MSTG-NETWORK-3 artinya mengangkut informasi sensitif melalui jaringan yang berpotensi membuat pengguna terkena serangan man-in-the-middle.

1.5 *Domain Malware Check*

Domain malware check ini berfungsi untuk menindai situs web domain memeriksa source code yang mencurigakan, media berbahaya, dan ancaman keamanan web lainnya yang disembunyikan ke dalam konten yang sah dan terletak di situs web.

ISSN : XXXX-XXXX (ONLINE)
DOI : 10.20895/LEDGER.V1I2.848



Gambar 4.9 Analisis File Aplikasi X8Speeder Bagian Domain Malware Check

Pada aplikasi X8Speeder terdapat beberapa domain salah satu diantaranya berstatus malware yang ditunjukkan pada gambar 4.9. Dengan standar:

1. URL da.anythinktech.com artinya url yang terindikasi
2. IPN/A artinya ip address tidak ditemukan
3. Malicious Domain tagged by Maltrail artinya domain da.anythinktech.com telah menjadi daftar hitam sebagai domain berbahaya oleh Maltrail. Maltrail sendiri adalah sistem deteksi lalu lintas berbahaya, memanfaatkan daftar (hitam) yang tersedia untuk umum yang berisi jejak berbahaya.

Tabel 4.1 Hasil Analisis Statik

No	Parameter Analisis Statik	Status
1.	Dangerous Permission	Yes
2.	Weak Crypto	Yes
3.	Root Detection	No
4.	SSL Bypass	Yes
5.	Domain Malware Check	Malware

Pada Tabel 4.1 dapat disimpulkan bahwa aplikasi X8Speeder memiliki dangerous permission yang meliputi:

dangerous permission yang meliputi:

1. android.permission.ACCESS_COARSE_LOCATION
2. android.permission.ACCES_FINE_LOCATION
3. android.permission.GET_TASK
4. android.permission.READ_EXTERNAL_STORAGE
5. android.permission.READ_LOGS
6. android.permission.READ_PHONE_STATE
7. android.permission.REQUEST_INSTALL_PACKAGES
8. android.permission.WRITE_EXTERNAL_STORAGE
9. android.permission.WRITE_SETTINGS

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

weak crptyo meliputi:
1. Aplikasi menggunakan mode enkripsi ECB dalam algoritma enkripsi kriptografi .

2. MD5 adalah hash lemah yang diketahui memiliki tabrakan hash yang termasuk warning severity

3. SHA-1 adalah hash lemah yang diketahui memiliki tabrakan hash yang termasuk warning severity

ssl bypass meliputi *warning SSL Certificate* :
1. Insecure Implementasi SSL Certificate mempercayai semua sertifikat atau menerima sertifikat yang ditandatangani sendiri adalah celah eamanan yang penting. Aplikasi ini rentan terhadap serangan MITM.

root detection yaitu tidak memiliki kemampuan root detection karena pada bagian code analysis tidak terdapat file atau kode tentang root detection.

domain malware check terdapat domain malware pada:
1. URL da.anythinktech.com
2. IP Address tidak terdeteksi
3. Malicious Domain tagged by Maltrail artinya dotnain da.anythinktech.com telah menjadi daftar hitam sebagai domain berbahaya oleh Maltrail.

B. Background Process Analysis

1. Live Instalasi X8Speeder

Pada tahap ini penulis akan secara langsung mendemokan behavioar apa saja yang dilak ukan malware setelah tertanam di android. Pada Gambar 4.10 merupakan alur instalasi pemasangan dari malware tersebut.

Gambar 4.10 Alur instalasi live demo android

Ketika malware tertanam di hp android. Terdapat beberap tindakan tambahan dari game dapat terlihat dari tabel berik ut:

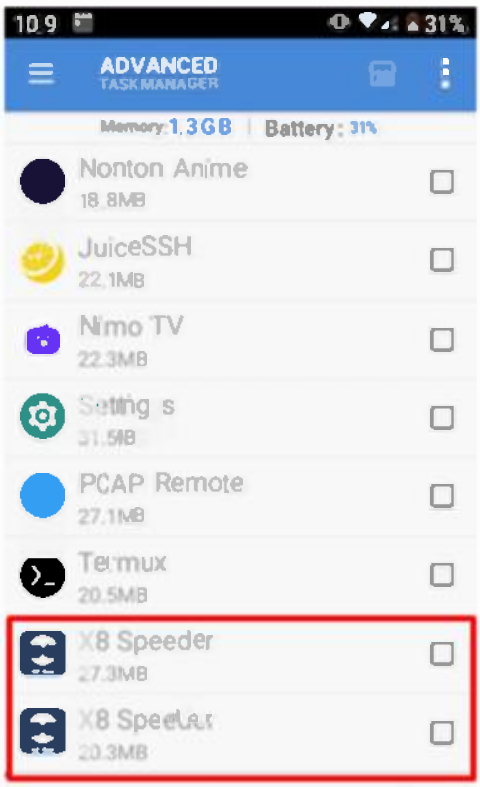
Tabel 4. 2 Informasi X8Speeder terisntal di android

No	Tindakan yang muncul	Terindikasi
1	Pop up iklan pada Game	*adware*
2	Background proses yang tidak berhenti	*malware*

2. Analisis Background Proses

Pada tahap ini akan di cek apakah ketika aplikasi Game yang dimainkan sudah di close background prosesnya apakah malware masih malukakan proses di belakang layar. Untuk dapat mengetahui hal tersebut penulis akan melak ukan analisa background proses dengan Advance Task Managaer pada live android seperti pada Gambar 4.11.

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848



Gambar 4.11 Analisis Background proses di Android

Dari hasil analisi background proses menggunakan Advance Task Manager dapat dilihat bahwa X8Speeder memiliki 2 proses app yang berbeda yang masih berjalan ketika game sudah di berhentikan yang tentunya kita ketahui melaui dangerous permission melakukan permission untuk install package tambahan dari 2 app yang berjalan ini dengan status dangerous.

V. KESIMPULAN

Berdasarkan penelitian yang dilakukan maka diambil kesimpulan sebagai berikut:

1. Penelitian ini berhasil menguji aplikasi android jenis APK di Mobile Security Framework (MobSF)
2. Hasil implementasi sampel apk X8Speeder menggunakan MobSF metode static analysis yaitu permission analysis menunjukan permission tidak pada tempatnya.
3. Hasil implementasi sampel apk X8Speeder menggunakan MobSF metode static analysis yaitu weak crypto masih menggunakan enkripsi yang lemah atau ketinggalan zaman.
4. Hasil implementasi sampel apk X8Speeder menggunakan MobSF metode static analysis yaitu SSL Bypass pada beberapa URL masih menggunakan HTTP dan Insecure Implementation SSL Certificate yang rentan terhadap serangan MITM.
5. Hasil implementasi sampel apk X8Speeder menggunakan MobSF metode static analysis yaitu Domain Malware Check terdapat status malware pada domain da.anythinktech.com yang di tag oleh Maltrail.
6. Hasil implementasi background proses android adanya pop-up iklan berulang-ulang yang berkarakteristik adware. Dan terdapat 2 proses app yang berbeda yang masih berjalan ketika game sudah di berhentikan.

ISSN: XXXX-XXXX (ONLINE)
DOI: 10.20895/LEDGER.V1I2.848

REFERENSI

[1] I. K. A., Y. A. Setyoko, and A. Wijayanto, "Analisis Perbandingan Performansi MIPv6 (Mobile Internet Protocol v6) dan HMIPv6 (Hierarchical Mobile Internet Protocol v6) pada VANET (Vehicular Ad- Hoc Network)," *J. Sist. Inf.*, vol. 5, no. 1, pp. 17–26, 2019.

[2] Yankotinu Al Qod'rJonnata, Fahrudin Mukti Wibowo, and Iqsyahiro Kresna A, "Prototype Pendeteksi Daerah Rawan Kecelakaan Berbasis Internet of," *J. RESTI*, vol. 1, no. 10, pp. 4–10, 2021.

[3] H. V. A. Manoppo et al., "Analisa Malware Menggunakan Metode Dynamic Analysis Pada Jaringan Universitas Sam Ratulangi," vol. 9, no. 3, pp. 181–188, 2020.

[4] E. Tansen and D. W. Nurdiarto, "Analisis Dan Deteksi Malware Dengan Metode Hybrid Analysis Menggunakan Framework Mobsf," vol. 4, no 2, pp. 191–201, 2020.

[5] A. Kartono, A. Sularsa, and S. J. I. Ismail, "Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan Mobsf," vol. 5, no. 1, pp. 146–151, 2019.

[6] A. F. Febrianto, A. Budiono, P. Studi, S. Informasi, F. R. Industri, and U. Telkom, "Analisis Malware Pada Sistem Operasi Android Menggunakan Metode Network Traffic Analysis Malware Analysis in Android Operating System Using Network," vol. 6, no 2, pp. 7837–7844, 2019.

[7] L. Putra, Rizky, "Analisis Aktivitas Malware Pada Ram Android Dan Sandbox Environment," p. 76, 2019.

[8] A. S. Rusdi, N. Widiyasono, and H. Sulastri, "Analisis Infeksi Malware Pada Perangkat Android Dengan Metode Hybrid Analysis," Univ. Puter. Batam, vol. 46115, no. 24, p. 107, 2019.

[9] F. Nurindahsari and B. Parga Zen, "Analisis Statik Keamanan Aplikasi Video Streaming Berbasis Android Menggunakan Mobile Security Framework (Mobsf)," Cyber Secur. dan Forensik Digit., vol. 4, no. 2, pp. 63–80, 2022, doi: 10.14421/csecurity.2021.42.3373.

[10] B. A. Saputro, L. I. Alfitra, and R. B. Oktaviaji, "Analisis Malware Android Menggunakan Metode Reverse Engineering," J. Repos., vol. 2, no. 10, pp. 1331–1337, 2020, doi: 10.22219/repositor.v2i10.1061.

[11] S. Sinambela, A. R. Pangestu, and R. Ferriyanto, "Analisis Aplikasi Malware pada Android dengan Metode Statik," J. Ilm. Ilk. - Ilmu Komput. Inform., vol. 3, no. 2, pp. 88–94, 2020, doi: 10.47324/ilkominfo.v3i2.101.

[12] R. B. Hadiprakoso, N. Qomariasih, and R. N. Yasa, "Identifikasi Malware Android Menggunakan Pendekatan Analisis Hibrid Dengan Deep Learning," J. Teknol. Inf. Univ. Lambung Mangkurat, vol. 6, no. 2, pp. 77–84, 2021, doi: 10.20527/jtiulm.v6i2.82.

[13] K. Shirke and Medium.com, "Mobile Security Framework (MobSF) Static Analysis," www.medium.com, 2019. https://medium.com/@kshitishirke/mobile-security-framework-mobsf-static-analysis-df22fcdae46e (accessed Feb. 18, 2021).

[14] K. Alfalqi, R. Alghamdi, and M. Waqdan, "Android Platform Malware Analysis," Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 1, pp. 140–146, 2015, doi: 10.14569/ijacsa.2015060120.

[15] Google Developers, "Panduan pemrosesan latar belakang," developer.android.com. https://developer.android.com/guide/background.

[16] Julián Falcionelli, "Background Processing in Android," https://medium.com. https://medium.com/@_julian_falcionelli/background-processing-in-android-575fd4ecf769.

[17] Cwe.mitre.org, "CWE-327: Use of a Broken or Risky Cryptographic Algorithm," Cwe..Mitre.Org, 2021. https://cwe.mitre.org/data/definitions/327.html.

[18] Google Developers, "Izin di Android," https://developer.android.com. https://developer.android.com/guide/topics/permissions/overview.

[19] OWASP, "OWASP Mobile Application Security," owasp.org. https://owasp.org/www-project-mobile-app-security/.

[20] H. A. Nugroho and Y. Prayudi, "Penggunaan Teknik Reverse Engineering Pada Malware Analysis Untuk Identifikasi Serangan," Knsi, pp. 27–28, 2014.

[21] A. Arora, S. Garg, and S. K. Peddoju, "Malware detection using network traffic analysis in android based mobile devices," Proc. - 2014 8th Int. Conf. Next Gener. Mob. Appl. Serv. Technol. NGMAST 2014, pp. 66–71, 2014, doi: 10.1109/NGMAST.2014.57.

[22] B. Santoso, M. A. Ghofur, and J. Kuswanto, "Analysis of WhatsApp Mod User Awareness Information Security with Static Analysis Methods and Quantitative Methods," Pros. Semin. Nas. Sains Teknol. dan Inov. Indones., vol. 3, no. November, pp. 213–222, 2021, doi: 10.54706/senastindo.v3.2021.128.

[23] C. Hanifurohman and D. D. Hutagalung, "Analisis Statis Menggunakan Mobile Security Framework Untuk Pengujian Keamanan Aplikasi Mobile E-Commerce Berbasis Android," Sebatik, vol. 24, no. 1, pp. 22–28, 2020, doi: 10.46984/sebatik.v24i1.920.