Analisis Malware AhMyth Pada Android Menggunakan Metode *Reverse Engineering*

1st Muhammad Ridwan Hidayat
Direktorat Universitas Telkom Purwokerto
Universitas Telkom Purwokerto
Purwokerto, Indonesia
20102066@ittelkom-pwt.ac.id

2nd Wahyu Adi Prabowo, S.Kom., M.B.A., M.Kom.
Direktorat Universitas Telkom Purwokerto
Universitas Telkom Purwokerto
Purwokerto, Indonesia
wahyup@telkomuniversity.ac.id

3rd Trihastuti Yuniati, S.Kom., M.T.
Direktorat Universitas Telkom Purwokerto
Universitas Telkom Purwokerto
Purwokerto, Indonesia
trihastutiy@telkomuniversity.ac.id

Abstrak — Penggunaan smartphone berbasis Android yang meningkat pesat membawa risiko keamanan privasi dan data pengguna. Salah satu ancaman utama adalah malware berjenis Remote Access Trojan (RAT), seperti AhMyth, yang memungkinkan penyerang mengakses perangkat secara ilegal. Penelitian ini bertujuan untuk menganalisis kemampuan malware AhMyth ketika disematkan pada aplikasi WhatsApp GB yang telah dimodifikasi oleh pihak ketiga. Metode yang digunakan adalah reverse engineering untuk membandingkan perubahan kode aplikasi sebelum dan sesudah disusupi malware. Analisis dilakukan secara otomatis menggunakan Mobile Security Framework (MobSF) dan manual menggunakan JADX. Hasil analisis menunjukkan bahwa penyisipan malware menyebabkan perubahan pada struktur kode aplikasi, termasuk penambahan izin baru seperti ACCESS_BACKGROUND_LOCATION, READ_SMS, dan PROCESS_OUTGOING_CALLS, yang memungkinkan pengumpulan data sensitif seperti lokasi pengguna, pesan SMS, dan informasi panggilan. Malware juga meningkatkan risiko privasi dengan akses tersembunyi ke kamera, mikrofon, dan file pengguna tanpa memerlukan interaksi langsung dari korban.

Penelitian ini diharapkan dapat meningkatkan kesadaran mengenai ancaman keamanan pada aplikasi Android yang tidak resmi, memberikan pemahaman tentang potensi bahaya malware, dan menjadi acuan bagi penelitian lebih lanjut dalam bidang analisis malware.

Kata kunci — AhMyth, WhatsApp GB, reverse engineering, MobSF, keamanan Android

I. PENDAHULUAN

Penggunaan Smartphone android sekarang ini meningkat secara signifikan di seluruh dunia. Menurut survei Databoks pengguna smartphone diprediksi akan mencapai 89% populasi di tahun 2025, dan akan terus meningkat seiring berjalannya waktu. Ini dikarenakan ponsel pintar semakin terjangkau, sehingga meningkatkan minat dari para penggunanya juga [1]. Karena jumlah yang banyak ini, muncul kasus-kasus tentang keamanan privasi dan informasi, sebab kurangnya kesadaran diri mengenai malware, spam, spoofing/phising, hingga mengunggah tentang hal yang bersifat pribadi seperti nomor telepon, dan alamat rumah [2].

Dalam era teknologi saat ini, internet telah menjadi bagian yang umum dalam kehidupan sehari-hari, membantu manusia dalam berbagai kegiatan dan aktivitas. Kemudahan yang ditawarkan oleh internet telah menghubungkan berbagai perangkat seperti komputer dan smartphone, memungkinkan mereka saling berinteraksi melalui jaringan ini. Manfaat internet sangat besar bagi kehidupan manusia, namun di balik segala keuntungannya, internet juga dapat membawa bencana bagi para penggunanya, khususnya melalui penyebaran malware. Malware adalah bentuk kejahatan yang sering muncul dalam jaringan komputer. Salah satunya adalah "backdoor", yang merupakan jenis virus Trojan Horse. Backdoor ini bisa berkembang di dalam perangkat yang terinfeksi dan memungkinkan penyerang untuk masuk ke sistem tanpa sepengetahuan pemiliknya. Biasanya, malware yang diinstal di dalam backdoor dikenal sebagai "Remote Access Trojan" (RAT). Jenis malware ini menjadi ancaman serius karena dapat mengambil alih kendali perangkat pengguna dan bahkan mencuri data pribadi atau informasi sensitif lainnya. Keberadaan backdoor memberi kesempatan kepada penyerang untuk secara diam-diam mencuri informasi, merusak sistem, atau melakukan aksi jahat lainnya tanpa sepengetahuan pemilik perangkat [3].

Salah satu bentuk kejahatan siber yang semakin sering terjadi adalah penipuan melalui file berekstensi Android Package Kit (APK). APK merupakan format aplikasi yang digunakan pada perangkat berbasis Android. Penipuan ini biasanya dilakukan melalui chat atau obrolan di platform media sosial seperti WhatsApp, dengan berbagai modus yang sangat beragam, seperti modus undangan pernikahan, cek resi paket, tagihan BPJS Kesehatan, surat tilang kepolisian, dan modus lainnya. Kronologisnya, penipuan ini melibatkan pengiriman file APK yang menjanjikan hal-hal menarik atau

penting kepada calon korban. Begitu calon korban mengunduh dan menginstal aplikasi tersebut, pelaku akan mendapatkan akses ilegal ke perangkat korban. Dengan akses ini, pelaku dapat dengan mudah menyadap data penting, seperti kode One Time Password (OTP), pin, dan password dari layanan perbankan mobile atau dompet digital (e-wallet) milik korban. Dampaknya sangat merugikan, karena begitu pelaku mendapatkan data sensitif korban, mereka akan menyalahgunakannya untuk mencuri seluruh saldo yang ada di akun perbankan mobile atau e-wallet korban. Hal ini dapat menyebabkan kerugian finansial yang besar bagi korban dan mengganggu keamanan serta privasi pribadi mereka [4].

Dalam penelitian ini, akan dilakukan analisis malware yang disisipkan pada sebuah aplikasi. Aplikasi yang digunakan pada penelitian ini adalah aplikasi Whatsapp GB. Aplikasi ini adalah hasil modifikasi dari pihak ketiga. Aplikasi inilah yang akan digunakan untuk melakukan eksploitasi pada smartphone android menggunakan alat bernama AhMyth. AhMyth adalah sebuah alat Administrasi Jarak Jauh atau Remote Administration Tool (RAT) sumber terbuka yang kuat, dirancang untuk mengakses data informatif dari perangkat android. AhMyth ini juga yang digunakan sebagai malware untuk melakukan pengujian ekslpoitasi ini.

Metode yang digunakan dalam penelitian ini adalah Reverse Engineering, adalah proses untuk memahami dan mengungkap cara kerja sebuah aplikasi dengan menganalisis cara operasinya, serta mempelajari struktur dan fungsinya [5]. Selanjutnya, hasil penelitian dianalisis dengan mengidentifikasi perbedaan dalam kode program, baik secara manual maupun otomatis. Analisis otomatis dilakukan menggunakan MobSF (Mobile Security Framework) dengan pendekatan analisis statis.

II. KAJIAN TEORI

A. Reverse Engineering

Reverse engineering, atau dalam bahasa Indonesia dikenal sebagai rekayasa balik, merupakan suatu proses untuk mengungkap dan memahami prinsip-prinsip kerja dari sebuah produk teknologi yang sudah ada. Proses ini dilakukan dengan menganalisis secara mendalam bagaimana suatu sistem, objek, atau perangkat berfungsi. Tujuan utamanya adalah untuk mengetahui bagaimana teknologi tersebut bekerja, dengan melakukan penelitian terhadap struktur, mekanisme, atau fungsi dari sistem yang diteliti. Melalui metode ini, para peneliti atau teknisi dapat memperoleh wawasan yang lebih dalam mengenai cara kerja teknologi yang ada, baik untuk kepentingan pengembangan lebih lanjut maupun untuk tujuan analisis dan evaluasi [6]

B. Whatsapp GB

WhatsApp GB merupakan hasil modifikasi dari WhatsApp Plus, yang juga merupakan aplikasi hasil modifikasi. WhatsApp Plus sendiri telah diblokir oleh pihak WhatsApp. Aplikasi tidak resmi ini hanya bisa dijalankan pada platform Android. Namun, WhatsApp GB tidak tersedia di Play Store. Aplikasi ini hanya dapat diunduh melalui file APK dari sumber pihak ketiga, seperti situs web. Mengunduh aplikasi dari sumber pihak ketiga tentu lebih berisiko dibandingkan jika mengunduhnya melalui Play Store [7].

C. AhMyth

AhMyth adalah alat eksploitasi yang beroperasi dengan arsitektur client-server, dirancang untuk mengakses perangkat Android melalui pemasangan payload di sistem target. Dengan menggunakan antarmuka GUI AhMyth, pengguna dapat mengakses berbagai informasi penting dari perangkat yang disusupi, seperti lokasi, sistem file, foto, rekaman suara, kontak, dan detail SMS. Selain itu, alat ini juga memungkinkan pemantauan aktivitas perangkat secara real-time, menjadikannya sangat efektif dalam melacak dan mengeksploitasi data perangkat Android secara menyeluruh [8].

D. JADX

Jadx adalah alat yang berguna dalam mendekompilasi file APK Android menjadi struktur yang lebih mudah dipahami, seperti kode sumber, folder sumber daya, dan tanda tangan APK. Dengan menggunakan Jadx, file APK yang sebelumnya hanya bisa dieksekusi dapat dipecah menjadi komponen-komponen yang dapat dianalisis lebih lanjut. Jadx tersedia dalam dua bentuk: versi Command Line (CLI), yang dioperasikan melalui perintah di terminal, dan versi Graphical User Interface (GUI), yang dikenal sebagai Jadx-gui, dengan tampilan antarmuka grafis yang lebih ramah pengguna. Kedua versi ini memberikan fleksibilitas bagi pengguna dalam melakukan dekompilasi APK, baik melalui perintah langsung di terminal maupun melalui antarmuka visual yang lebih mudah digunakan [9].

E. Mobile Security Framework (MobSF

Mobile Security Framework (MobSF) adalah sebuah framework pengujian otomatis yang bersifat open-source. Dirancang khusus untuk melakukan analisis menyeluruh terhadap aplikasi Android, MobSF memiliki kemampuan yang luas dalam menganalisis baik secara statis maupun dinamis. Proses analisis ini melibatkan penggunaan bahasa pemrograman Python untuk mengonfigurasi MobSF. Hasil dari analisis yang dilakukan oleh MobSF ditampilkan dalam bentuk laporan yang memberikan informasi detail mengenai aplikasi Android yang sedang dianalisis. Tujuan utama dari MobSF adalah untuk mendeteksi dan menganalisis keberadaan malware pada aplikasi Android, sehingga

dapat memberikan pemahaman yang lebih baik mengenai kelemahan keamanan yang mungkin ada [21].

F. Malware

Malware, yang juga dikenal sebagai perangkat lunak berbahaya, merujuk pada perangkat lunak yang diciptakan tanpa izin pemiliknya dengan tujuan untuk menyusup atau merusak sistem komputer, server, atau jaringan komputer. Saat ini, malware juga menyerang smartphone, termasuk yang menggunakan sistem operasi Android, karena popularitasnya di kalangan pengguna. Pada bulan Mei 2013, Google melaporkan bahwa terdapat 900 juta perangkat Android aktif di seluruh dunia, dengan lebih dari 48 miliar aplikasi yang telah diunduh dari toko resmi Google Play. Meskipun toko aplikasi Google Play dianggap sebagai sumber aplikasi yang aman, data statistik penyebaran malware di Indonesia pada tahun 2014 mengungkapkan adanya empat jenis malware yang menjadi ancaman serius. Pertama, ada Trojan Horse (kuda Troya), yang menyembunyikan dirinya dalam aplikasi yang seharusnya sah. Setelah diunduh dan diinstal oleh pengguna, Trojan Horse akan mencuri data pribadi atau mengendalikan perangkat tanpa sepengetahuan pengguna. Kedua, Worm (cacing) adalah jenis malware yang dapat menggandakan dirinya sendiri dan menyebar melalui jaringan, mempengaruhi banyak perangkat secara bersamaan. Worm umumnya digunakan untuk mengirim spam, mencuri informasi, atau menghancurkan data. Ketiga, Spyware (perangkat lunak mata-mata) dirancang untuk secara diam-diam mengumpulkan informasi pribadi pengguna dan aktivitas online mereka, termasuk kata sandi, riwayat penjelajahan, dan data sensitif lainnya. Terakhir, Ransomware (perangkat lunak pengeksploitasi) mengenkripsi data pada perangkat pengguna dan meminta tebusan agar data tersebut dapat dikembalikan. Jika tidak membayar, pengguna akan kehilangan akses ke data mereka. Mengingat ancaman ini, penting bagi pengguna smartphone Android untuk selalu berhati-hati, memperbarui keamanan perangkat secara teratur, dan hanya menginstal aplikasi dari sumber yang terpercaya [11]

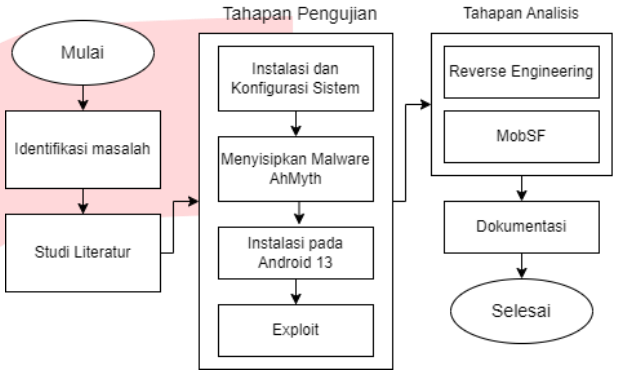
G. Backdoor

Backdoor adalah sebuah mekanisme rahasia yang tidak didokumentasikan yang memungkinkan seorang administrator untuk masuk ke dalam sistem guna melakukan pemecahan masalah atau pemeliharaan. Dalam situasi ketika masalah muncul, backdoor sering kali menjadi solusi yang efektif. Backdoor ini disisipkan secara diam-diam ke dalam kode sistem atau program sehingga pengguna tidak menyadari keberadaan backdoor di dalam sistem. Dengan adanya backdoor, pihak yang memiliki akses ke backdoor dapat masuk dan mendapatkan akses ke sistem pengguna, bahkan dalam beberapa kasus, mereka dapat mengakses seluruh sistem tersebut. Namun, penting untuk dicatat bahwa penggunaan backdoor dalam konteks keamanan sering kali tidak etis dan melanggar privasi dan integritas sistem yang terpengaruh [12]

III. METODE

A. Tahapan Penelitian

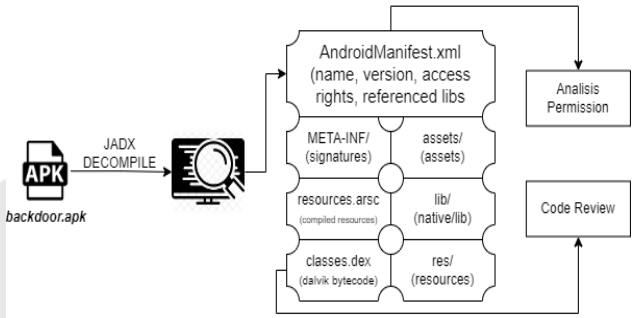
Penelitian ini dilaksanakan secara terstruktur dan sistematis melalui beberapa tahapan. Diagram Alir Penelitian dapat dilihat pada Gambar 1.



Gambar 1
(Diagram Alir Penelitian)

B. Alur *Reverse Engineering*

Penelitian ini dilaksanakan secara terstruktur dan sistematis melalui beberapa tahapan. Diagram Alir Penelitian dapat dilihat pada Gambar 1.



Gambar 2
(Alur Reverse Engineering)

Berdasarkan pada gambar 2 setelah aplikasi yang telah disusupi malware didekompilasi menggunakan JADX, hasilnya berupa kumpulan file yang mencerminkan struktur inti dari aplikasi Android. Salah satu file utama yang dihasilkan adalah AndroidManifest.xml, yang berfungsi sebagai berkas manifest aplikasi, memuat informasi penting tentang aplikasi dan izin-izin yang diminta oleh aplikasi pada perangkat Android. Selain itu, dekompilasi juga menghasilkan beberapa file lain, termasuk META-INF, assets, resources.arsc, lib, res, dan classes.dex. File META-INF menyimpan metadata terkait berkas APK, sedangkan assets berisi aset tambahan seperti file konfigurasi atau data terkait aplikasi. Resources.arsc mengandung sumber daya aplikasi, termasuk gambar, tata letak, dan string. Folder lib

menyimpan file biner dari library atau modul tambahan yang digunakan oleh aplikasi, sementara direktori res menyimpan sumber daya tambahan seperti ikon, gambar, dan file XML. Dan terakhir, berkas classes.dex berisi kode aplikasi yang sudah terkompilasi.

IV. HASIL DAN PEMBAHASAN

Hasil dari analisis otomatis dan manual dibandingkan untuk menemukan kesamaan serta perbedaan yang signifikan. Tujuan utama dari perbandingan ini adalah untuk memahami bagaimana perubahan kode mempengaruhi perilaku aplikasi serta memastikan keakuratan hasil yang diperoleh dari kedua metode tersebut.

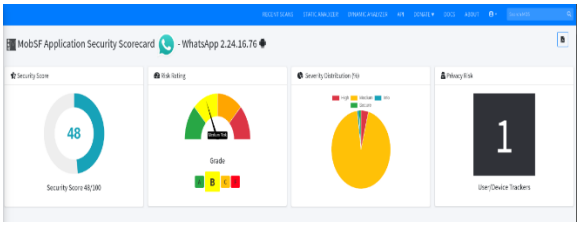
A. Hasil Analisis Otomatis Menggunakan MobSF (Analisis Statis)

Pendekatan pertama adalah analisis otomatis menggunakan Mobile Security Framework (MobSF), di mana aplikasi yang telah dimodifikasi diperiksa secara menyeluruh melalui pemindaian otomatis. Hasil dari analisis ini mencakup perbandingan sebelum dan sesudah aplikasi disusupi malware untuk mengidentifikasi perubahan dalam aspek keamanan.

1. Security Score

Security Score pada MobSF adalah indikator numerik yang menilai tingkat keamanan sebuah aplikasi berdasarkan analisis yang dilakukan oleh MobSF. Skor ini mencakup berbagai aspek keamanan, seperti izin yang diminta aplikasi (permissions), kerentanan dalam kode (vulnerabilities), praktik pengkodean yang tidak aman, serta masalah konfigurasi seperti kebocoran kunci API. Skor yang lebih tinggi menunjukkan aplikasi yang lebih aman, sementara skor rendah mengindikasikan adanya potensi risiko keamanan yang perlu diperbaiki.

Gambar 3
(Security score sebelum disusupi malware)



Gambar 4
(Security score sesudah disusupi malware)

Penurunan Security Score setelah penyisipan malware mengindikasikan bahwa MobSF mendeteksi adanya kerentanan tambahan, meskipun sedikit. Ini menunjukkan bahwa malware tersebut belum sepenuhnya berhasil mengeksploitasi aplikasi secara besar-besaran, tetapi cukup untuk mengurangi keamanan keseluruhan. Risk Rating yang tidak berubah dapat menandakan bahwa metode eksploitasi malware mungkin berfokus pada elemen yang lebih tersembunyi atau bersifat pasif, sehingga tidak langsung memengaruhi keseluruhan profil keamanan aplikasi. Namun, Privacy Risk yang tetap stabil dapat disebabkan oleh fakta bahwa malware tersebut menggunakan izin yang sudah dimiliki aplikasi sebelumnya, tanpa menambah izin baru yang terlalu mencurigakan.

Secara keseluruhan, perubahan dalam Security Score dan hasil analisis menunjukkan bahwa malware memang berdampak pada aplikasi, meski belum pada tingkat yang sepenuhnya mengubah profil keamanan dan privasi. Distribusi kerentanan yang tetap sama menunjukkan bahwa malware mungkin tidak menambah kerentanan baru yang terdeteksi, tetapi tetap dapat memengaruhi keamanan aplikasi secara keseluruhan.

2. Analisis Permission

Analisis permission adalah proses evaluasi terhadap izin-izin (permissions) yang diminta oleh suatu aplikasi. Pada aplikasi Android, izin ini biasanya mengontrol akses aplikasi terhadap fitur-fitur atau data sensitif perangkat seperti lokasi, kamera, mikrofon, kontak, penyimpanan, dan lainnya. Izin-izin ini penting untuk memastikan keamanan dan privasi pengguna, karena aplikasi yang memiliki akses tidak terbatas bisa menyalahgunakan data atau melakukan aktivitas yang tidak diinginkan.

Berikut adalah izin yang ditambahkan setelah disisipkan malware, yaitu:

Ada beberapa izin yang ditambahkan setelah disisipkan malware, yaitu:

- android.permission.ACCESS_BACKGROUND_LOCATION

  Fungsi: Memberikan akses bagi aplikasi untuk mengakses lokasi pengguna bahkan ketika aplikasi berjalan di latar belakang.

  Potensi Penyalahgunaan oleh Malware: Malware dapat memanfaatkan izin ini untuk melacak lokasi pengguna tanpa mereka sadari. Ini berguna bagi malware yang ingin memantau pergerakan pengguna secara terus-menerus, bahkan saat aplikasi tidak aktif di layar depan.

- android.permission.PROCESS_OUTGOING_CALLS

  Fungsi: Mengizinkan aplikasi untuk memonitor atau memodifikasi panggilan keluar.

  Potensi Penyalahgunaan oleh Malware: Malware bisa menggunakan izin ini untuk

mencatat informasi tentang panggilan keluar, mengalihkan panggilan ke nomor lain, atau bahkan merekam panggilan tanpa sepengetahuan pengguna.

- android.permission.READ_SMS
  Fungsi: Memungkinkan aplikasi untuk membaca pesan SMS yang diterima oleh perangkat.
  Potensi Penyalahgunaan oleh Malware: Malware dapat menggunakan izin ini untuk mengakses pesan pribadi pengguna, seperti kode verifikasi yang dikirim oleh bank atau layanan lainnya, yang berpotensi digunakan untuk keperluan peretasan atau penipuan.
- android.permission.REQUEST_IGNORE_BATTERY_OPTIMISATIONS
  Fungsi: Memberikan izin kepada aplikasi untuk meminta pengecualian dari optimasi baterai.
  Potensi Penyalahgunaan oleh Malware: Dengan izin ini, malware dapat tetap aktif di latar belakang tanpa batasan waktu, meskipun perangkat dalam mode hemat daya. Ini memungkinkan malware untuk menjalankan proses berkelanjutan yang mungkin tidak disadari oleh pengguna.
- android.permission.WRITE_SECURE_SETTINGS
  Fungsi: Memungkinkan aplikasi mengubah pengaturan sistem yang bersifat sensitif atau aman.
  Potensi Penyalahgunaan oleh Malware: Izin ini biasanya terbatas pada aplikasi sistem atau perangkat yang di-root. Jika dimanfaatkan oleh malware, izin ini memungkinkan perubahan pada pengaturan keamanan perangkat, yang dapat melemahkan pertahanan sistem atau membuka celah untuk serangan lainnya.
- android.permission.WRITE_SETTINGS
  Fungsi: Mengizinkan aplikasi untuk memodifikasi pengaturan perangkat, seperti volume, kecerahan layar, atau mode jaringan.
  Potensi Penyalahgunaan oleh Malware: Malware dapat menggunakan izin ini untuk mengubah perilaku perangkat tanpa sepengetahuan pengguna, yang dapat menyebabkan pengalaman pengguna terganggu atau bahkan memanipulasi fungsi perangkat untuk kepentingan jahat.
- android.permission.WRITE_SMS
  Fungsi: Memberikan akses kepada aplikasi untuk menulis atau mengirim pesan SMS.
  Potensi Penyalahgunaan oleh Malware: Malware yang memiliki izin ini dapat mengirim pesan SMS tanpa diketahui pengguna. Ini bisa digunakan untuk penyebaran malware melalui tautan di SMS atau untuk mengirim pesan ke nomor premium yang berpotensi merugikan pengguna secara finansial.

3. Code Analysis

No	Issue	Severity (Sebelum)	Severity (Sesudah)
1	The App logs information. Sensitive information should never be logged.	info	info
2	SHA-1 is a weak hash known to have hash collisions.	warning	warning
3	The App uses an insecure Random Number Generator.	warning	warning
4	App can read/write to External Storage. Any App can read data written to External Storage.	warning	warning
5	This App copies data to clipboard. Sensitive data should not be copied to clipboard as other applications can access it.	info	info
6	Files may contain hardcoded sensitive information like usernames, passwords, keys, etc.	warning	warning
7	This App may have root detection capabilities.	secure	secure
8	The file or SharedPreference is World Readable. Any App can read from the file	high	high
9	IP Address disclosure	warning	warning
10	MD5 is a weak hash known to have hash collisions.	warning	warning
11	This app has capabilities to prevent tapjacking attacks.	secure	secure
12	App uses SQLite Database and executes raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also, sensitive information should be encrypted and written to the database.	warning	warning
13	App creates temp file. Sensitive information should never be written into a temp file.	warning	warning
14	The file or SharedPreference is World Writable. Any App can write to the file	high	high
15	Debug configuration enabled. Production builds must not be debuggable.	high	high
16	The App uses the encryption mode CBC with PKCS5/PKCS7 padding. This configuration is vulnerable to padding oracle attacks.	high	high
17	This App uses SSL certificate pinning to detect or prevent MITM attacks in secure communication channel.	secure	secure
18	Calling Cipher.getInstance("AES") will return AES ECB mode by default. ECB mode is known to be weak as it results in the same ciphertext for identical blocks of plaintext.	high	high
19	Insecure WebView Implementation. Execution of user-controlled code in WebView is a critical Security Hole.	warning	warning

Gambar 5
(Perbandingan Severity sebelum dan sesudah disisipkan malware)

Dalam keamanan aplikasi, severity membantu mengidentifikasi prioritas dalam penanganan masalah. Penjelasannya:

- Info prioritasnya rendah karena dampaknya minimal.
- Warning mengindikasikan risiko yang perlu perhatian tetapi tidak darurat.
- High menunjukkan ancaman serius yang perlu ditangani segera.
- Secure menandakan area di mana aplikasi sudah memiliki proteksi yang memadai

Dan dari hasil analisis yang dilakukan oleh MobSF, tidak ada perubahan severity dari sebelum ke sesudah di sisipkan malware.

B. Hasil Analisis Manual

Pendekatan kedua adalah analisis manual, di mana aplikasi didekompilasi untuk melihat secara langsung perubahan pada struktur source code. Proses ini memungkinkan penelusuran lebih mendalam terhadap kode yang telah dimodifikasi atau ditambahkan oleh malware. Perbandingan antara kode asli dan kode setelah modifikasi membantu dalam mengidentifikasi aktivitas yang tidak sah atau berbahaya.

1. Analisis Permission

   Analisis permission adalah proses evaluasi terhadap izin-izin (permissions) yang diminta oleh suatu aplikasi. Pada aplikasi Android, izin ini biasanya mengontrol akses aplikasi terhadap fitur-fitur atau data sensitif perangkat seperti lokasi, kamera, mikrofon, kontak, penyimpanan, dan lainnya. Izin-izin ini penting untuk memastikan keamanan dan privasi pengguna, karena aplikasi yang memiliki akses tidak terbatas bisa menyalahgunakan data atau melakukan aktivitas yang tidak diinginkan.

   Berikut adalah izin yang ditambahkan setelah disisipkan malware, yaitu:

   Ada beberapa izin yang ditambahkan setelah disisipkan malware, yaitu:

   - android.permission.ACCESS_BACKGROUND_LOCATION

     Fungsi: Memberikan akses bagi aplikasi untuk mengakses lokasi pengguna bahkan ketika aplikasi berjalan di latar belakang.

     Potensi Penyalahgunaan oleh Malware: Malware dapat memanfaatkan izin ini untuk melacak lokasi pengguna tanpa mereka sadari. Ini berguna bagi malware yang ingin memantau pergerakan pengguna secara terus-menerus, bahkan saat aplikasi tidak aktif di layar depan.
   - android.permission.PROCESS_OUTGOING_CALLS

     Fungsi: Mengizinkan aplikasi untuk memonitor atau memodifikasi panggilan keluar.

     Potensi Penyalahgunaan oleh Malware: Malware bisa menggunakan izin ini untuk mencatat informasi tentang panggilan keluar, mengalihkan panggilan ke nomor lain, atau bahkan merekam panggilan tanpa sepengetahuan pengguna.
   - android.permission.READ_SMS

     Fungsi: Memungkinkan aplikasi untuk membaca pesan SMS yang diterima oleh perangkat.

     Potensi Penyalahgunaan oleh Malware: Malware dapat menggunakan izin ini untuk mengakses pesan pribadi pengguna, seperti kode verifikasi yang dikirim oleh bank atau layanan lainnya, yang berpotensi digunakan untuk keperluan peretasan atau penipuan.
   - android.permission.REQUEST_IGNORE_BATTERY_OPTIMISATIONS

     Fungsi: Memberikan izin kepada aplikasi untuk meminta pengecualian dari optimasi baterai.

     Potensi Penyalahgunaan oleh Malware: Dengan izin ini, malware dapat tetap aktif di latar belakang tanpa batasan waktu, meskipun perangkat dalam mode hemat daya. Ini memungkinkan malware untuk menjalankan proses berkelanjutan yang mungkin tidak disadari oleh pengguna.
   - android.permission.WRITE_SECURE_SETTINGS

     Fungsi: Memungkinkan aplikasi mengubah pengaturan sistem yang bersifat sensitif atau aman.

     Potensi Penyalahgunaan oleh Malware: Izin ini biasanya terbatas pada aplikasi sistem atau perangkat yang di-root. Jika dimanfaatkan oleh malware, izin ini memungkinkan perubahan pada pengaturan keamanan perangkat, yang dapat melemahkan pertahanan sistem atau membuka celah untuk serangan lainnya.
   - android.permission.WRITE_SETTINGS

     Fungsi: Mengizinkan aplikasi untuk memodifikasi pengaturan perangkat, seperti volume, kecerahan layar, atau mode jaringan.

     Potensi Penyalahgunaan oleh Malware: Malware dapat menggunakan izin ini untuk mengubah perilaku perangkat tanpa sepengetahuan pengguna, yang dapat menyebabkan pengalaman pengguna terganggu atau bahkan memanipulasi fungsi perangkat untuk kepentingan jahat.
   - android.permission.WRITE_SMS

     Fungsi: Memberikan akses kepada aplikasi untuk menulis atau mengirim pesan SMS.

     Potensi Penyalahgunaan oleh Malware: Malware yang memiliki izin ini dapat mengirim pesan SMS tanpa diketahui pengguna. Ini bisa digunakan untuk penyebaran malware melalui tautan di SMS atau untuk mengirim pesan ke nomor premium yang berpotensi merugikan pengguna secara finansial.

2. META-INF

   Analisis pada file META-INF dilakukan untuk memeriksa tanda tangan digital atau informasi penting lainnya yang menunjukkan apakah file aplikasi telah dimodifikasi.

APKEASYT.SF	ANDROIDD.SF
Signature-Version: 1.0 Created-By: 1.0 (Android) SHA-256-Digest-Manifest: 3/UL2MDgbVZP93IdPz6XMnPPCeItk8LuJ0 fKLJOyJwc= X-Android-APK-Signed: 2, 3 Name: AndroidManifest.xml SHA-256-Digest: HGBXK+0TBJLEO2eag2bvz2n4O/dSEJopHW voYsuzkpk=	Signature-Version: 1.0 Created-By: 1.0 (Android) SHA-256-Digest-Manifest: DMhSaYmzHr9d0l5h9g+JIT7QC14SIofwxb 83VcTD4oY= X-Android-APK-Signed: 2, 3 Name: AndroidManifest.xml SHA-256-Digest: yuDONrMgD9meTSFAqksx2qFB+qmmsjzHH6 l11mEA0qQ=

Gambar 6
(Perubahan pada META-INF)

Perbedaan dalam nilai hash menunjukkan bahwa konten dari aplikasi ini telah berubah setelah penyisipan malware. Ini mengindikasikan bahwa ada modifikasi pada file manifest, yang dapat berupa penambahan, penghapusan, atau perubahan izin (permissions) dan konfigurasi lainnya. Dari table perbandingan di atas, dapat diketahui bahwa terdapat perubahan signifikan pada kedua file tesebut setelah disisipkan malware, yang dapat mencakup perubahan pada permissions dan atribut lain yang berpotensi membahayakan keamanan aplikasi. Modifikasi ini adalah karakteristik umum dari malware yang berusaha menyembunyikan aktivitas atau mengeksploitasi fungsi perangkat Android.

3. Resources

Menunjukkan konten dari resources, yang berfungsi sebagai lokasi penyimpanan untuk semua sumber daya dan file yang diperlukan oleh aplikasi. Dalam folder tersebut, tidak terdapat penambahan folder yang signifikan, namun terdapat tambahan berupa file classes7.dex.

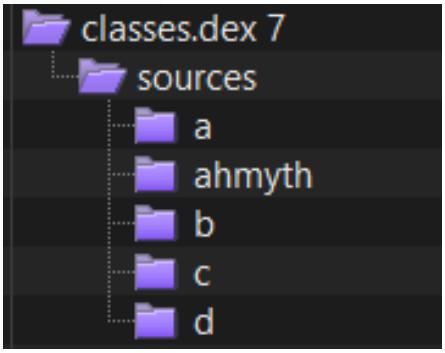
Gambar 7
(Resources sebelum dan sesudah)

menunjukkan konten dari resources, yang berfungsi sebagai lokasi penyimpanan untuk semua sumber daya dan file yang diperlukan oleh aplikasi. Dalam folder tersebut, tidak terdapat penambahan folder yang signifikan, namun terdapat tambahan berupa file classes7.dex.

4. Classes.dex

Analisis dilakukan terhadap file classes.dex yang berisi Dalvik bytecode, yaitu kode program yang dieksekusi oleh runtime Android. File classes.dex ini dianalisis untuk melihat perbedaan sebelum dan sesudah penyisipan malware AhMyth. Berikut adalah hasil analisis dari setiap classes.dex.

- Classes.dex
  Pada classes.dex di aplikasi original (sebelum disisipi malware), tidak ditemukan file tambahan dari AhMyth. Namun, setelah penyisipan AhMyth, ditemukan adanya sejumlah file tambahan, termasuk folder android/arch dan androidx/versionedparcelable yang merupakan indikasi perubahan signifikan akibat malware. Gambar dapat dilihat pada lampiran halaman 1 menunjukkan struktur classes.dex pada aplikasi sebelum dan sesudah disisipi malware. Terlihat jelas perbedaan jumlah file dan penambahan folder yang mengindikasikan modifikasi oleh AhMyth.
- Classes2.dex hingga Classes6.dex
  Pada classes2.dex, classes3.dex, classes4.dex, classes5.dex, dan classes6.dex, tidak ditemukan penambahan file ataupun folder setelah disisipi AhMyth. Analisis pada bagian-bagian ini mengindikasikan bahwa penyisipan malware tidak mempengaruhi struktur file classes.dex di lima file ini. Menunjukkan struktur dari classes2.dex hingga classes6.dex yang tidak mengalami perubahan apapun. File-file ini tetap utuh dan tidak mengandung file tambahan yang berkaitan dengan malware AhMyth.
- Classes7.dex
  Classes7.dex adalah file tambahan yang disisipkan oleh AhMyth. File ini berisi komponen-komponen penting untuk eksploitasi target, seperti koneksi jarak jauh dan layanan background yang bertugas menjalankan perintah dari server malware.

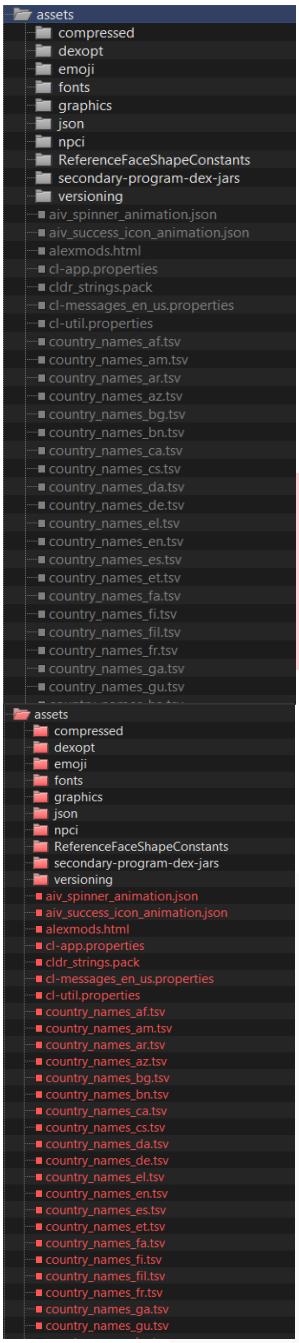


Gambar 8
(Classes7.dex)

Struktur classes7.dex sangat berbeda dari file classes lainnya, dengan adanya tambahan beberapa file penting yang memungkinkan komunikasi jarak jauh dan pengelolaan perangkat dari jarak jauh oleh AhMyth.

5. Assets

Folder ini berfungsi menyimpan file tambahan yang digunakan oleh aplikasi, seperti file konfigurasi, gambar, atau resource lain yang tidak dikompilasi.

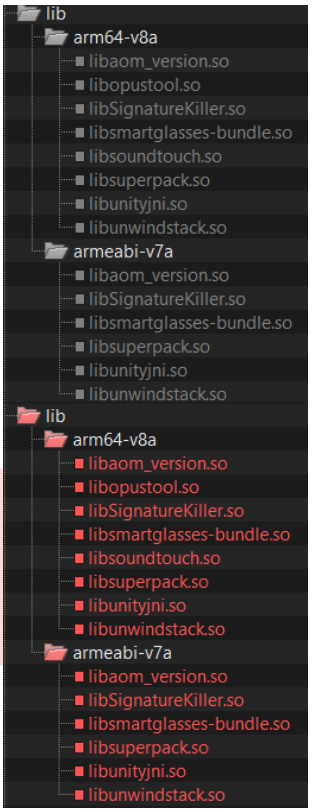


Gambar 9
(Assets)

menunjukkan bahwa tidak ada penambahan file atau folder apapun.

6. Lib

Folder ini berisi file yang berfungsi untuk menyimpan native libraries dalam format .so, biasanya untuk prosesor tertentu (ARM, x86, dll).

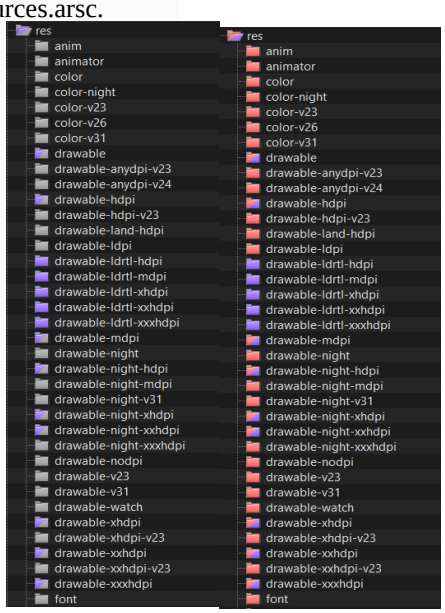


Gambar 10
(Folder Lib)

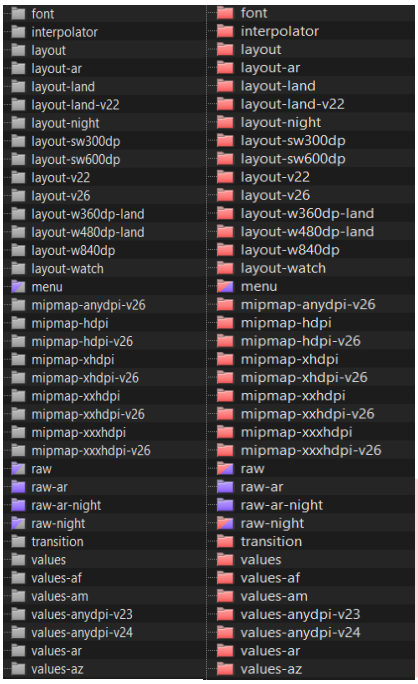
Pada gambar 10 folder lib tidak ada penambahan file atau folder apapun.

7. Res

Folder ini berfungsi untuk menyimpan resource mentah seperti gambar, layout XML, strings, dan file-file lainnya sebelum dikompilasi ke dalam resources.arsc.



Gambar 11
(Folder Res)



Gambar 12
(Folder Res)

Dapat dilihat pada gambar 11 dan gambar 12, terdapat penambahan dan pergantian file pada beberapa folder yang berwarna biru. Pergantian file yang dimaksud yaitu mengganti nama file.

C. Perbandingan Analisis Otomatis (MobSF) dan Manual

Kedua metode analisis, MobSF sebagai alat pemindaian otomatis dan analisis manual menggunakan hasil dekompilasi aplikasi, saling melengkapi dalam memberikan gambaran terkait perubahan pada aplikasi setelah disisipi malware. Berikut adalah perbandingan kedua metode tersebut:

1. Kemudahan dan Kecepatan Analisis
   - MobSF: Memberikan hasil secara cepat dan otomatis tanpa membutuhkan pemahaman mendalam tentang struktur kode. Dan bagus untuk analisis awal karena mampu mendeteksi izin (permissions), skor keamanan, dan kerentanan umum dalam waktu singkat.
   - Manual: Memerlukan waktu dan usaha lebih karena melibatkan pemeriksaan langsung pada kode yang telah didekompilasi. Meski lebih lambat, metode ini memungkinkan pengamatan yang lebih rinci terhadap perubahan kode.
2. Tingkat Kedalaman Analisis
   - MobSF: Berfokus pada analisis seperti skor keamanan, izin akses, dan kerentanan umum. Tidak mampu mendeteksi perubahan mendetail pada level source code, seperti penambahan kelas atau fungsi spesifik.
   - Manual: Mampu mengidentifikasi perubahan mendetail pada struktur kode, seperti penambahan kelas atau fungsi baru (contoh: ConnectionManager dan AdminReceiver).
3. Kemampuan Deteksi Malware
   - MobSF: Tidak secara langsung mengidentifikasi file atau kelas yang berisi kode malware, tetapi memberikan gambaran umum terkait kerentanan aplikasi yang dapat dimanfaatkan oleh malware.
   - Manual: Memungkinkan identifikasi langsung kode-kode spesifik yang digunakan oleh malware, memberikan pemahaman lebih mendalam tentang cara kerja malware.
4. Keterbatasan
   - MobSF: Analisis otomatis tidak mampu mendeteksi perubahan kecil atau file spesifik seperti classes7.dex.
   - Manual: Membutuhkan keahlian dari analis untuk memahami kode yang kompleks dan waktu yang lebih lama untuk menyelesaikan analisis.

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, berikut adalah kesimpulan yang diperoleh:

A. Kemampuan Malware AhMyth pada Sistem Operasi Android

Malware AhMyth memiliki kemampuan signifikan dalam memantau dan mengendalikan perangkat Android dari jarak jauh, yang dapat dirinci sebagai berikut:

- Pemantauan Data
  Malware ini mampu mengakses berbagai data sensitif pada perangkat, seperti lokasi, log panggilan, pesan, dan file media.
- Kontrol Jarak Jauh
  Malware memberikan akses bagi penyerang untuk mengendalikan perangkat secara tersembunyi, termasuk mengambil foto, merekam audio, atau menjalankan perintah tertentu tanpa sepengetahuan pengguna.

Dari kemampuan tersebut, dapat disimpulkan bahwa malware AhMyth dirancang dengan tujuan utama untuk spionase dan kendali perangkat secara jarak jauh, dengan memanfaatkan izin yang sudah ada maupun izin tambahan yang disisipkan selama proses injeksi.

B. Perubahan Kode Aplikasi Sebelum dan Sesudah Disisipi Malware AhMyth

Penyisipan malware AhMyth membawa dampak yang cukup signifikan pada kode aplikasi yang menjadi target, meliputi:

- Penambahan Class dan Fungsi Baru
  Beberapa class baru, seperti MainService, ConnectionManager, dan AdminReceiver, ditemukan telah ditambahkan ke dalam aplikasi. Class-class ini berfungsi untuk mendukung operasi latar belakang dan komunikasi dengan server malware. Penambahan tersebut teridentifikasi melalui analisis dekompilasi manual menggunakan JADX.
- Penambahan Izin Akses (Permissions)

Malware menambahkan sejumlah izin baru yang memungkinkan akses dan kontrol terhadap perangkat, seperti izin untuk membaca lokasi, log panggilan, atau data pada penyimpanan eksternal. Izin tambahan ini meningkatkan potensi risiko keamanan perangkat.

- Penambahan File classes7.dex
  File baru ini mengandung kode yang berkaitan dengan aktivitas malware. Keberadaan file ini tidak terdeteksi oleh analisis otomatis menggunakan MobSF, namun dapat ditemukan melalui analisis manual menggunakan JADX.

REFERENSI

[1] T. Danny Soesilo, S. Irawan, P. Studi Bimbingan dan Konseling, and U. Kisten Satya Wacana, "Pengaruh Penggunaan Smartphone Terhadap Interaksi Sosial Remaja," Salatiga, 2022.

[2] R. Akraman, C. Candiwan, and Y. Priyadi, "Pengukuran Kesadaran Keamanan Informasi Dan Privasi Pada Pengguna Smartphone Android Di Indonesia," J. Sist. Inf. Bisnis, vol. 8, no. 2, p. 115, 2018, doi: 10.21456/vol8iss2pp115-122.

[3] M. Alvian, H. Nasution, and A. T. Laksono, "Investigasi Serangan Backdoor Remote Access Trojan (RAT) Terhadap Smartphone," JURIKOM (Jurnal Ris. Komputer), vol. 7, no. 4, pp. 505–510, 2020, doi: 10.30865/jurikom.v7i4.2301.

[4] A. R. Sali and D. M. Sari, "Kampanye Sosial Pencegahan Fenomena Penipuan File . APK," JURRSENDEM, vol. 2, no. 2, pp. 51–59, 2023, doi: https://doi.org/10.55606/jurrsendem.v2i2.1549.

[5] N. Widiyasono, H. Mubarok, and A. Fatwa MF, "Analisis Malware Ahmyth pada Platform Android Menggunakan Metode Reverse Engineering," Gener. J., vol. 6, no. Vol. 6 No. 2 (2022): Generation Journal, pp. 114–123, 2022, doi: 10.29407/gj.v6i2.17749.

[6] Y. A. Utomo, S. J. I. Ismail, and T. Zani, "Membangun Sistem Analisis Malware Pada Aplikasi Android Dengan Metode Reverse Engineering Menggunakan Remnux," eProceedings …, vol. 4, no. 3, pp. 2000–2012, 2018, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/7164%0A https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/download/7164/7052

[7] N. Krisdayanti and I. Gunawan, "Analisis Keamanan Aplikasi Chat Android Pihak Ketiga Atau Non Playstore Menggunakan Digital Forensics," Sekol. Tinggi Teknol. Ronggolawe Cepu, vol. 16, no. 2, pp. 1–5, 2022, [Online].

[8] J. Tomy, J. Thomas, N. Jacob, and M. L. Varghese, "Securing Android Phones against Potential Thefts : AhMyth Android RAT," vol. 2, no. Vol. 2 No. 1 (2020): NCECA 2020, pp. 2–7, 2020, [Online]. Available: https://ajcejournal.in/nceca/article/view/27

[9] D. Hindarto, "Perbandingan Kinerja Akurasi Klasifikasi K-NN, NB dan DT pada APK Android," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 1, pp. 486–503, 2022, doi: 10.35957/jatisi.v9i1.1542.

[10] A. Kartono and A. Sularsa, "Membangun Sistem Pengujian Keamanan Aplikasi," eProceedings Appl. Sci., vol. 5, no. 1, pp. 146–151, 2019.

[11] Y. A. Utomo, S. J. I. Ismail, and T. Zani, "Membangun Sistem Analisis Malware Pada Aplikasi Android Dengan Metode Reverse Engineering Menggunakan Remnux," eProceedings …, vol. 4, no. 3, pp. 2000–2012, 2018, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/7164%0A https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/download/7164/7052

[12] S. Supri, Royana, and Munawaroh, "Implementation Of Backdoor Metasploit Framework For Android Using Windows," SATIN - Sains dan Teknol. Inf., vol. 9, no. 1, 2023, doi: 10.33372/stn.v9i1.952.