Analisis Aplikasi Cryptowallet Tiruan Terhadap Indikasi Android Malware

Adenta Rubian Qiyas Syahwidi[1)], Setiyo Cahyono[2)], Ray Novita Yasa[3)]

1) Rekayasa Perangkat Lunak Kripto, Politeknik Siber dan Sandi Negara, adenta.rubian@student.poltekssn.ac.id

2) Rekayasa Perangkat Lunak Kripto, Politeknik Siber dan Sandi Negara, setiyo.cahyono@poltekssn.ac.id

3) Rekayasa Perangkat Lunak Kripto, Politeknik Siber dan Sandi Negara, ray.novita@poltekssn.ac.id

Abstrak

Pemanfaatan teknologi smartphone dalam hal ini Android penggunaannya semakin tahun semakin meningkat. Begitu juga dengan perkembangan teknologi pada bidang ekonomi yang dalam hal ini adalah aset kripto. Semakin tahun semakin banyak pemilik aset kripto yang menyimpan aset kriptonya sebagai mata uang digital. Aset tersebut disimpan pada suatu wadah cryptowallet yang salah satunya berbentuk aplikasi android. Pengembang perangkat lunak dapat menggunakan kesempatan tersebut untuk mengambil keuntungan dengan membuat aplikasi atau software yang berbahaya bagi pengguna. Adanya tiruan aplikasi yang hampir mirip dengan aplikasi aslinya tentunya meresahkan bagi pengguna awam yang kurang tahu mengenai perbedaannya. Bisa saja aplikasi tersebut disisipi oleh suatu code yang berbahaya. Dalam penelitian ini dilakukan analisis terhadap aplikasi tiruan cryptowallet milik perusahaan MetaMask dengan menggunakan analisis statis dan analisis dinamis. Penelitian ini bertujuan untuk mengetahui karakteristik dan dampak perilaku bagi pengguna android khususnya pemilik aset kripto dengan maksud agar menambah informasi dan pengetahuan bagi pengguna khususnya terkait adanya aplikasi tiruan yang beredar di lingkungan umum.

Kata kunci: Analisis Dinamis (1), Analisis Statis (2), Android (3), Aplikasi Tiruan (4), *Malware* (5)

Abstract

Every year, there is a growth in the use of smartphone technology, specifically Android. A crypto asset in this context is an example of how technology is advancing in the world of economics. Over time, an increasing number of owners of crypto assets have begun to store such assets as digital money. These assets are kept in a container called a crypto wallet, one of which is an android app. Software developers can take advantage of this opportunity by creating applications or software that are harmful to users. For normal users who do not know the difference, the existence of a fake application that is almost identical to the original application is undoubtedly troubling. The application might have been injected with alicious code. In this research, both static and dynamic analysis were used to examine a fake cryptowallet application that belonged to the MetaMask company. This research intends to inform users of android, particularly those who possess cryptocurrency assets, on the characteristics and behavioral impacts that is hoped to better inform users about the existence of fake applications that are spreading in public.

Keywords: Android (1), Fake Application (2), Malware (3), Static Analysis (4), Dynamic Analysis (5)

1. PENDAHULUAN

Indonesia Honeynet Project BSSN mencatat terdapat serangan *malware* sebanyak ribuan kasus setiap bulan. Serangan *malware* tertinggi pada tahun 2020 di bulan Februari dengan 36.321 kasus [1]. Sehingga malware menjadi salah satu serangan yang paling banyak dicatat oleh BSSN. *Malware* atau *malicious software* merupakan suatu perangkat lunak yang biasa disebut *software* yang bisa digunakan untuk mengganggu perangkat komputer, mengumpulkan informasi sensitif atau mendapatkan akses ke sistem komputer [2]. Jenis jenis *malware* meliputi *virus, worm, logic bomb, trojan, sniffers, adware, spyware* dan beberapa jenis *malware* lainnya [2].

Perkembangan teknologi khususnya pada sektor ekonomi berkembang pesat dalam beberapa tahun terakhir. Salah satu perkembangannya adalah mata uang kripto yang lebih dikenal dengan *cryptocurrency*. *Cryptocurrency* merupakan aset digital asli yang berasal dari sistem *blockchain open assets public* dengan berbagai macam keuntungan pada bidang ekonomi [3]. *Cryptocurrency* ini sangat populer di seluruh belahan dunia, termasuk di Indonesia. Tercatat di Indonesia saja, 2.6% dari penduduk Indonesia yang berkisar 7.2 juta orang memiliki *cryptoasset* [4]. Pengguna dapat mendapatkan aset kripto melalui *mining*, *trading*, dan bisa juga dengan menggunakan investasi untuk menjaga nilai dari koin.

Pemilik aset kripto dapat menyimpan aset mereka pada suatu wadah dengan nama dompet kripto atau yang lebih populer dikenal dengan *cryptowallet*. Wadah tersebut bisa ditemukan dan dipasang dari *website* penyedia layanan dan bisa juga melalui pengunduh aplikasi *smartphone* baik Android maupun iOS. Penyedia layanan terkait *cryptowallet* tersebut semakin banyak ditemukan pada aplikasi yang beredar. Namun, dengan adanya kemudahan dalam mengakses *cryptowallet*, terdapat kekhawatiran lain yakni adanya suatu *malware*. Tercatat adanya

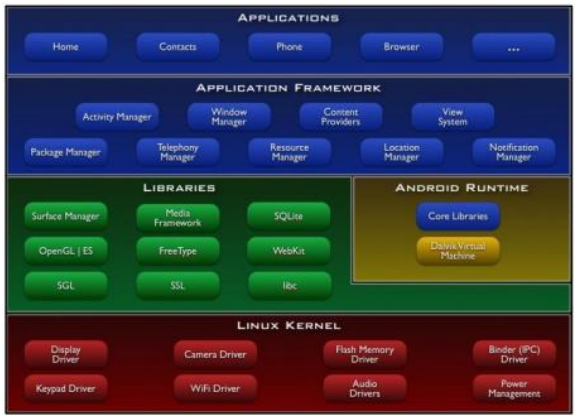
serangan terkait *banking malware* yang menginfeksi 300.00 pengguna Google Play sejak bulan Juli 2021 hingga November 2021 [5]. *Malware* tersebut berbentuk beragam seperti aplikasi kesehatan, QR *codes*, PDF *scanning* dan bahkan berupa aplikasi yang menyerupai aplikasi resmi dengan mengambil akun bank milik korban. Dengan adanya serangan tersebut membuat pemilik aset kripto harus berhati-hati dalam menjaga aset mereka.

Pada penelitian ini akan dilakukan analisis karakteristik dan perilaku serta dampak dari aplikasi tiruan *cryptowallet* pada *smartphone* khususnya android. Sampel diunduh dari sumber terbuka yakni Play Store dan berbentuk aplikasi android yang menyerupai aplikasi penyedia jasa *cryptowallet* milik perusahaan MetaMask. Metode yang dilakukan adalah dengan menggunakan analisis statis untuk meneliti *source code* dan analisis dinamis untuk melihat aktivitas dari sampel yang diteliti. Hasil yang diperoleh nantinya dapat memberikan informasi dan himbauan terkait aplikasi yang hendak digunakan untuk menyimpan aset kripto.

2. LANDASAN TEORI

2.1. Android

Android berdasarkan penelitian milik J.A Shaheen dan rekannya merupakan sistem operasi *open source* berbasis Linux kernel [6]. Sistem operasi ini berjalan dengan lima *layer*. Berdasarkan penelitian J.A. Shaheen, *layer* tersebut terbagi seperti pada Gambar 1.



Gambar 1. Layer Android berdasarkan [6]

Dijelaskan bahwa struktur android terdiri dari 5 *layer* yang meliputi:

a. *Application layer* merupakan *layer* yang bertanggung jawab pada SMS, *maps*, *browser*, kontak dan beberapa lainnya. Menggunakan bahasa java. Berinteraksi langsung dengan pengguna.
b. *Application framework layer* merupakan layer yang berinteraksi dengan aplikasi yang digunakan. *Layer* ini berjalan diatas manajemen fungsi dasar dari *resource manager*, *activity manager*, *package manager* dan beberapa lainnya.
c. *Libraries layer* merupakan *layer* yang membawa sekumpulan instruksi untuk mengarahkan perangkat android kita dalam menangani berbagai tipe data. Contohnya, perekam dari berbagai macam format video dan audio ditangani oleh Media Framework Library, seperti Webkit, Sqlite, Openssl.
d. *Android runtime layer* merupakan *layer* yang terletak pada level yang sama dengan lapisan *library* juga terdapat lapisan *Android Runtime* dan juga sekumpulan *Library Java* yang dikhususkan untuk Android. *Programmer* aplikasi android membuat aplikasinya menggunakan bahasa pemrograman Java.
e. Kernel *layer* merupakan *layer* yang merupakan inti dari sistem operasi berbasis UNIX. Kernel merupakan lapisan terdalam yang berada pada sistem operasi baik itu Linux maupun Android.

2.2. *Cryptocurrency*

Cryptocurrency atau mata uang kripto merupakan aset digital yang menggunakan teknologi *blockchain* untuk mengamankan transaksi [7]. Mata uang kripto ini dipopulerkan pada tahun 2008 dengan nama anonim "Satoshi Nakamoto", yang mengeluarkan *white paper* berisikan implementasi dari mata uang digital yang dinamakan *Bitcoin* dan menggunakan teknologi *blockchain*. *Cryptocurrency* ini menggunakan mekanisme *peer-to-peer* untuk melakukan transaksi dan secara efektif menghilangkan orang ketiga yang merupakan institusi keuangan. Dengan menggunakan ponsel dan internet sudah bisa terhubung langsung dengan perusahaan pemiliknya.

Tercatat pada situs resmi milik *coinmarketcap* [8], terdapat 1.583 mata uang kripto yang digunakan untuk transaksi. Koin yang paling populer adalah *Bitcoin* dengan *marketcap* sebesar $903.48 Milyar menjadikan koin dengan *marketcap* paling besar yang ada pada daftar *cryptocurrencies*. Aset kripto ini dapat didapatkan melalui *mining*, *trading* dan bisa juga dengan melakukan *investing* untuk membuat nilainya semakin besar tiap tahunnya. Dengan adanya aplikasi tiruan yang banyak beredar pada internet khususnya pada Play Store tentunya menimbulkan kekhawatiran terhadap pemilik aset kripto dalam melakukan transaksi.

2.3. Malware

Malware adalah singkatan dari *malicious software* yang berarti perangkat lunak yang mencurigakan atau berbahaya. *Malware* ini memiliki beberapa pengertian berdasarkan beberapa jurnal. Sajedul Talukder dan Zahidur Talukder yang mengatakan bahwa *malware* merupakan perangkat lunak berbahaya yang berfungsi untuk mengganggu aktifitas mesin, mengambil informasi sensitif atau mengambil akses sistem computer [9]. Muhammad

Bilal Mirza dan mengatakan bahwa malware adalah perangkat lunak berbahaya yang dapat menyusup masuk melalui instalasi program atau perangkat lunak [10].

2.4. Teknik Analisis *Malware*

Malware analysis atau analisis *malware* adalah proses penentuan tujuan dan fungsi dari suatu sampel *malware* tertentu [11]. Sampel *malware* ini banyak jenisnya mulai dari *trojan, virus, worm* dan beberapa sampel *malware* lain. Dengan ditemukannya *malware* berjenis baru, maka proses analisis *malware* ini diperlukan. Untuk mengetahui jenis dari *malware* tersebut dan bagaimana cara kerjanya. Analisis *malware* dibagi menjadi tiga yakni analisis statik, analisis dinamis, dan analisis *hybrid* [12] dengan penjelasan sebagai berikut:

a. Analisis statis merupakan teknik analisis yang dilakukan dengan melakukan pembongkaran malware dan meneliti *source code* tanpa mengeksekusi *malware* tersebut [12].
b. Analisis dinamis dilakukan dengan cara melakukan pengamatan terhadap perilaku dari sistem yang didapati melalui hasil eksekusi *malware* [12].
c. Teknik analisis *hybrid* ini merupakan teknik analisis yang menggabungkan analisis dinamis dan analisis statis [12]. Analis melakukan penelitian terhadap *malware* dengan cara membedah dan meneliti *signature* beserta *source code* dari *malware*. Kemudian dilakukan penelitian dengan memantau atau monitoring terhadap sistem ketika *malware* dieksekusi.

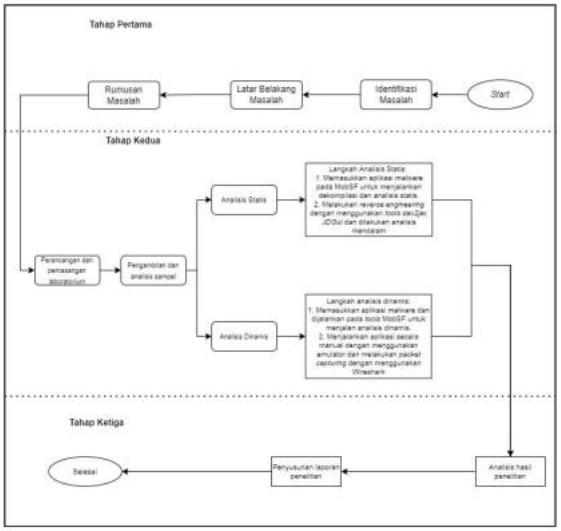
2.5. Sandbox

Sandbox merupakan lingkungan jaringan yang terpisah atau terisolasi dan memiliki tampilan yang sama dengan lingkungan atau *environment* milik *end-user* yang digunakan untuk melakukan eksekusi kode atau aplikasi yang bersifat *malicious* [13]. Analisis menggunakan *sandbox* ini memiliki keuntungan yakni mampu mendapatkan informasi lebih detail terkait perilaku dari suatu *malware* yang menjangkit suatu sistem. Tentunya akan lebih mendetail lagi terkait informasi yang didapat apabila dilakukan bersamaan dengan analisis statis terhadap *malware* [14].

3. METODE PENELITIAN

Penelitian yang dilakukan menggunakan penelitian deskriptif kualitatif yang dilakukan dengan mengumpulkan data dengan kondisi objek yang alamiah, kemudian data dilakukan analisis dan didukung dengan teori serta wawasan penelitian untuk menerangkan dan menjelaskan secara rinci terkait penelitian yang dilakukan [15]. Kajian dilakukan dengan mengumpulkan berbagai referensi berupa jurnal, buku, artikel dan beberapa sumber lainnya yang mendukung penelitian. Analisis dilakukan dengan menganalisis *source code* dan perilaku terhadap sistem dengan menggunakan *static analysis* dan *dynamic analysis*.

Desain pada penelitian ini menggunakan metodologi yang ditunjukkan pada Gambar 2.



Gambar 2. Tahapan Penelitian

Penjelasan dari alur tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.1. Tahapan Pertama

Tahap pertama dalam penelitian dilakukan identifikasi terhadap suatu masalah. Penyusunan latar belakang masalah diambil sesuai dengan kejadian yang terjadi ditambahkan dengan studi literatur yang mendukung. Dari kedua hal tersebut, dirumuskan permasalahan untuk penelitian yang dilakukan yaitu melakukan analisis terhadap aplikasi tiruan MetaMask menggunakan *static analysis* dan *dynamic analysis*.

3.2. Tahapan Kedua

Tahapan kedua ini terdiri dari perancangan laboratorium penelitian, pengambilan sampel, proses analisis statis dan analisis dinamis. Tahapan yang dimaksud adalah sebagai berikut:

1. Bagian yang pertama adalah perancangan dan pembangunan dari laboratorium penelitian. Dilakukan penyesuaian spesifikasi dan kebutuhan dalam pembangunan laboratorium.
2. Bagian yang kedua adalah pengambilan sampel aplikasi. Sampel aplikasi diunduh dari sumber terbuka Play Store sebanyak 3 aplikasi. Penentuan sampel aplikasi didasari atas adanya laporan pada sebuah forum di sosial media mengenai temuan 3 aplikasi imitasi dari aplikasi *cryptowallet* bernama MetaMask.
    a. Bagian yang ketiga adalah dilakukan analisis statis. Analisis statis dilakukan dengan meneliti *sourcecode* dan file *android manifest* pada aplikasi dengan menggunakan *tool* MobSF. Dalam [16] terdapat langkah

dalam melakukan analisis statis pada aplikasi mobile khususnya android. Tahapan tersebut meliputi:*Pre-Processing Phase.*

Pada fase ini, aplikasi diekstraksi untuk mendapatkan data mengenai *classes, manifest file, metadata information* dan *media files*. Pada penelitian ini, aplikasi diekstrasi menggunakan beberapa *tools* meliputi Jadx-GUI dan MobSF.

b. *Feature Extraction Phase*

Fase selanjutnya bergantung pada hasil yang didapatkan pada fase sebelumnya. Dilakukan penelitian untuk mendeteksi dan mengetahui detail terkait aplikasi yang diteliti. Fitur yang akan diamati dalam penelitian ini adalah *Android Permissions*, *API*, *strings*, dan *Domain*.

Pengambilan parameter dijelaskan sebagai berikut:

i. *Android Permissions*

Aplikasi *mobile* khususnya pada sistem operasi android memiliki permintaan akses atau yang sering ditemui ketika menginstal aplikasi pertama kali adalah *android permissions.*

ii. *Application Programming Interface* (API)

Android API dapat diintegrasikan dengan informasi sensitif pada *device* seperti lokasi, *device id,* kontak dan penyimpanan. Sehingga android API ini dapat menjadi hal yang cukup vital mengingat dapat mengakses informasi sensitif.

iii. *String* Analysis

Strings merupakan hal yang paling penting dalam melakukan analisis statis pada sebuah aplikasi yang tersimpan pada "classes.dex" [17] karena di dalamnya merupakan kumpulan *source code* dari sebuah aplikasi.

iv. *Domain Check*

Penelitian dilanjutkan dengan melakukan analisis terkait *domain* yang ada pada *source code.* Hal tersebut dilakukan pengecekan terkait IP *Address, geolocation* dan indikasi terkait apakah domain tersebut bersifat *malicious*.

c. *Detection Phase*

Pada tahap ini dilakukan pengecekan kembali, dan informasi yang didapat bisa digunakan untuk acuan pada analisis dinamis.

3. Bagian yang keempat adalah dilakukan analisis dinamis. Analisis dinamis dilakukan dengan beberapa tahap yang meliputi:

a. *Debugging application*

Debugging application dilakukan dengan menggunakan *tool* MobSF yang terintegrasi dengan Genymotion untuk membantu dalam proses analisis dinamis. Hasil yang didapatkan adalah berupa informasi terkait perilaku dari aplikasi yang diteliti.

b. *Executing Application*

Pada tahap ini akan dilakukan analisis dinamis dengan menjalankan aplikasi pada emulator dan dijalankan secara manual dengan melihat kondisi perangkat sebelum aplikasi dijalankan dan sudah dijalankan lalu dilakukan pencatatan pada perbedaan yang ditemukan.

c. *Network traffic analysis*

Analisis dinamis dilanjutkan dengan *packet capturing* menggunakan aplikasi Wireshark. Aplikasi dijalankan secara manual pada emulator *android Nox Player*. Hasil yang diharapkan pada analisis dinamis adalah berupa *packet transaction* yang kemudian dilakukan analisis.

3.3. Tahapan Ketiga

Tahap ketiga merupakan tahapan yang paling akhir. Pada tahap ini dilakukan analisis terhadap hasil simulasi dan pengujian yang telah dilakukan. Hasil simulasi berdasarkan analisis statis dan dinamis diolah kembali dan disesuaikan. Hasil dari tahap analisis ini berupa karateristik dan perilaku serta dampak dari aplikasi tiruan MetaMask pada pengguna android. Hasil yang diperoleh dapat memberikan informasi serta himbauan terkait aplikasi yang hendak digunakan untuk menyimpan aset kripto. Kemudian dilakukan penyusunan laporan berdasarkan tahapan yang sudah dilakukan pada sebelumnya. Laporan ini disusun dengan data yang diperoleh dan diberikan kesimpulan dari penelitian yang telah dilakukan.

4. HASIL DAN PEMBAHASAN

4.1. Analisis Statis

1. MetaMask Sampel 1

Sampel pertama yang dilakukan penelitian adalah dengan nama paket "era.ability.okay" dan nilai hash 530c036cb68a738ae02ca5eda2917b32. Rincian dari sampel aplikasi pertama tertera pada Tabel 1. Dilakukan analisis statis dengan menggunakan *tool* MobSF dan Jadx-GUI sehingga didapatkan data sebagai berikut:

A. *Pre-Processing Phase*

Tahap ini merupakan tahap awal ketika melakukan analisis statis. Prosesnya dimulai dari melakukan dekompilasi pada sampel aplikasi 1 yang hendak diteliti menggunakan tool MobSF. Setelah fase *pre-processing phase*, dilanjutkan dengan fase *feature*

extraction phase dan *detection phase*.

B. *Feature Extraction Phase*

Tahap ini merupakan tahapan selanjutnya setelah dekompilasi. Hasil dari ekstrasi yang didapatkan dengan menggunakan *tool* MobSF meliputi *android permissions, API Calls, malicious string/ source code,* dan *domain address.* Hasilnya adalah sebagai berikut:

a. *Permissions*

Setelah dilakukan analisis statis dengan memasukkan sampel ke *tool* MobSF, didapati informasi terkait *permissions.* Terlihat aplikasi asli memiliki 23 *permissions*, sedangkan sampel aplikasi 1 memiliki kurang lebih 28 *permissions*. Dengan *permissions* yang berbeda sejumlah 17 *permissions*.

b. *Application Programming Interface* (API)

Didapatkan informasi terkai API dari sampel aplikasi 1, ditemukan bahwa API milik aplikasi asli berjumlah 30, sedangkan untuk sampel aplikasi berjumlah 23. Ditemukan perbedaan 12 API milik aplikasi asli tidak terdapat pada sampel aplikasi dan terdapat 3 API milik sampel aplikasi yang tidak ada pada aplikasi asli. Berdasarkan hal tersebut maka 3 API yang berbeda tersebut, menjadi fokus utama selain API lain saat dilakukan analisis statis.

c. Source code analysis

d. *Domain Checker*

Hasil dari tahap ini tidak didapati bahwa domain yang ditemukan tidak berbahaya.

Tabel 1. Informasi Sampel Aplikasi 1

No	Info	Keterangan
1	*Nama Aplikasi*	*MetaMask*
2	*Nama Paket*	*era.ability.okay*
3	*Main Activity*	*broom.december.decide.yxuygreds*
4	*Target SDK*	*27*
5	*Minimal SDK*	*20*
6	*Android Version Support*	*8.1 and up*
7	*Developer*	-
8	*MD5*	*530c036cb68a738ae02ca5eda2917b32*
9	*SHA1*	*569754483243730d16b47f09afa8a67c42c76947*
10	*SHA256*	*201a62940058ba3068282562cd29af15521e4f2b8612fc3005c4e8c70142b4a3*
11	*Versi Aplikasi*	*1.0*

C. *Detection Phase*

Tahapan ini dilakukan penelitian ulang dan peninjauan ulang terkait hasil analisis pada tahap sebelumnya. Hasil yang didapati adalah sebagai berikut:

a. *Android Permissions*

Hasil pengamatan mendapati adanya *string* yang menggunakan android.permission.READ_PHONE_STATE untuk membaca serial number milik perangkat yang cukup mencurigakan untuk dilakukan oleh aplikasi atau dapat juga digunakan untuk meminta nomor telepon pengguna beserta daftar panggilan yang ada pada kontak. Kemudian adanya permission meminta perangkat terkait lokasi pasti dengan menggunakan android.permission.ACCESS_FINE_LOCATION dan android.permission.ACCESS_COARSE_LOCATION serta selalu meminta update terkait lokasi perangkat. Lalu adanya permission android.permission.ACCESS_NETWORK_STATE yang meminta kondisi terkini terkait koneksi perangkat.

b. Android *Application Programming Interface* (API)

Hasil dari analisis android ditemukan adanya string yang mencurigakan dengan menggunakan API Loading Native Code untuk meminta *temporary directory* dari *user profile* dan membuat sebuah *temporary*. Ditemukan juga adanya string yang meminta *gain access privileged* dari perangkat, adanya API *get installed application* yang menggunakan *intent* untuk mendapatkan email dari perangkat serta mengakses Whatsapp pengguna, adanya string yang memanfaatkan *clipboard* perangkat yang kemungkinan digunakan untuk melakukan *copy* dan *paste* isi yang ada pada *clipboard*, adanya string yang meminta untuk membaca pesan yang diterima.

c. *Domain Checker*

Pada tahap ini tidak ditemukan adanya domain yang mencurigakan.

2. MetaMask Sampel 2

Sampel kedua yang dilakukan penelitian adalah dengan nama paket "muffin.breeze.royal" dan nilai *hash* ba4bb3c9ba686611674deddf3b0929d0.

Rincian dari sampel aplikasi kedua tertera pada Tabel 2. Analisis statis dengan menggunakan *tool* MobSF dan Jadx-GUI sehingga didapatkan data sebagai berikut:

A. *Pre-Processing Phase*

Sampel aplikasi 2 akan dianalisis menggunakan *tool* MobSF kemudian dilanjutkan dengan fase *feature extraction phase* dan *detection phase.*

Tabel 2. Informasi Sampel Aplikasi 2

No	Info	Keterangan
1	*Nama Aplikasi*	*MetaMask*
2	*Nama Paket*	*muffin.breeze.royal*
3	*Main Activity*	*range.consider.garbage.vkzwh*
4	*Target SDK*	*27*
5	*Minimal SDK*	*20*
6	*Android Version Support*	*8.1 and up*
7	*Developer*	-
8	*MD5*	ba4bb3c9ba686611674deddf3b0929d0
9	*SHA1*	5f06ad620334b5c6c45d72c63556d569bfc636f8
10	*SHA256*	e8601ae0ce75ed76963aee761bccfd65d4057dabda7ad5209603c8adb88268e4
11	*Versi Aplikasi*	*1.0*

B. *Feature Extraction Phase*

Tahap ini merupakan tahapan selanjutnya setelah dekompilasi. Hasil dari ekstrasi yang didapatkan dengan menggunakan *tool* MobSF meliputi *android permissions, API Calls, malicious string/ source code,* dan *domain address.* Hasilnya adalah sebagai berikut:

a. *Permissions*

Setelah dilakukan analisis statis dengan memasukkan sampel ke *tool* MobSF, didapati informasi terkait *permissions.* Aplikasi asli memiliki 23 *permissions*, sedangkan sampel aplikasi 2 sama seperti sampel aplikasi 1 memiliki kurang lebih 28 *permissions*. Dengan *permissions* yang berbeda sejumlah 17 *permissions.*

b. *Application Programming Interface* (API)

Pada sampel aplikasi 2 ditemukan API milik aplikasi asli berjumlah 30, sedangkan untuk sampel aplikasi berjumlah 23. Perbedaan 12 API milik aplikasi asli tidak terdapat pada sampel aplikasi dan terdapat 3 API milik sampel aplikasi yang tidak ada pada aplikasi asli menjadi fokus utama selain API lainnya saat dilakukan analisis statis.

c. *Source code analysis*

Hasil dari proses analisis menemukan adanya beberapa string yang mencurigakan pada file berupa parameter seperti membaca kondisi perangkat, ditemukan adanya perintah yang mencurigakan meminta angka MacAddress WiFi dari perangkat. Berdasarkan string tersebut diindikasikan bahwa meminta perangkat untuk mengirimkan lokasi yang menunjukkan adanya indikasi permintaan access previledge pada file io/netty/util/internal/j.java.

e. *Domain Checker*

Hasilnya *domain* yang ditemukan tidak berbahaya.

C. *Detection Phase*

Tahapan ini dilakukan penelitian ulang dan peninjauan ulang terkait hasil analisis pada tahap sebelumnya. Hasil yang didapati adalah sebagai berikut:

a. *Android Permissions*

Pada sampel aplikasi 2 ditemukan adanya *string* yang menggunakan android.permission.READ_PHONE_STATE untuk membaca *serial number* milik perangkat yang cukup mencurigakan untuk dilakukan oleh aplikasi. *String* ini dapat juga digunakan untuk meminta nomor telepon pengguna beserta daftar panggilan yang ada pada kontak. Selanjutnya ditemukan adanya *permission* meminta perangkat terkait lokasi pasti dengan menggunakan android.permission.*ACCESS*_FINE_LOCATION dan android.permission.*ACCESS*_COARSE_LOCATION serta selalu meminta *update* terkait lokasi perangkat. Ada juga *permission* android.permission.*ACCESS*_NETWORK_STATE yang meminta kondisi terkini terkait koneksi perangkat.

b. Android *Application Programming Interface* (API)

Pada sampel aplikasi 2 ditemukan adanya *string* mencurigakan dengan menggunakan API *Loading Native Code* untuk meminta *temporary directory* dari *user profile* dan membuat sebuah *temporary file.* Dalam satu *line* tersebut ditemukan juga adanya *string* yang meminta *gain access privileged* dari perangkat serta adanya API *get installed application* yang menggunakan *intent* untuk mendapatkan *email* dari perangkat serta mengakses Whatsapp pengguna. Selain itu ditemukan juga adanya *string* yang memanfaatkan *clipboard* perangkat yang diduga digunakan untuk melakukan *copy* dan *paste* isi yang ada pada *clipboard* dan ditemukan

pula adanya *string* yang meminta untuk membaca pesan yang diterima.

c. *Domain Checker*
Hasil pengecekan tidak ditemukan adanya domain yang mencurigakan.

3. MetaMask Sampel 3
Sampel ketiga yang dilakukan penelitian adalah dengan nama paket "io.metamask" dan nilai *hash* 7e939043ce7015f274d0017b9fc4a7b6.
Rincian dari sampel aplikasi ketiga tertera pada Tabel 3.

Tabel 3. Informasi Sampel Aplikasi 3

No	Info	Keterangan
1	*Nama Aplikasi*	*MetaMask*
2	*Nama Paket*	*io.metamask*
3	*Main Activity*	*Io.metamask.SplashActivity*
4	*Target SDK*	*27*
5	*Minimal SDK*	*19*
6	*Android Version Support*	*10 and up*
7	*Developer*	-
8	*MD5*	bfb246d24728a65aa9dbf477c371a075
9	*SHA1*	11943c7c82767406396ced865197f8291b178215
10	*SHA256*	*774795e87753057dd9a8b31ea0560b14bf5e289bd452daa10e0bee1faa592e0c*
11	*Versi Aplikasi*	*3.2.0*

Analisis statis dengan menggunakan *tool* MobSF dan Jadx-GUI sehingga didapatkan data sebagai berikut:

A. *Pre-Processing Phase*
Pada tahap ini dilakukan dekompilasi pada sampel aplikasi 3 yang hendak diteliti menggunakan *tool* MobSF. phase dan *detection phase*.

B. *Feature Extraction Phase*
Hasil dari ekstraksi yang didapatkan dengan menggunakan *tool* MobSF meliputi *android permissions, API Calls, malicious string/ source code,* dan *domain address.* Hasilnya adalah sebagai berikut:

a. *Permissions*
Setelah dilakukan analisis statis dengan memasukkan sampel 3 ke *tool* MobSF, didapati informasi informasi terkait *permissions* sampel aplikasi memiliki kesamaan yang serupa dengan aplikasi asli sehingga membutuhkan analisis secara teliti untuk menemukan *source code* yang mencurigakan.

b. *Application Programming Interface* (API)
Tahap ini mendapatkan informasi terkait API dari sampel aplikasi 3. Sampel 3 merupakan sampel yang memiliki permissions dan API yang sama dengan aplikasi MetaMask asli sehingga dilakukan pengecekan dan penilaian mendalam pada sampel aplikasi 3.

c. *Source code analysis*
Proses analisis yang menemukan adanya beberapa string yang mencurigakan seperti pada file com/learnium/RNDeviceInfo/RNDeviceModule.java dengan menggunakan permissions "android.permission.READ_PHONE_STATE". Indikasi terhadap *string* tersebut adalah adanya upaya untuk meminta SN atau *serial number* dari perangkat. Selain itu pada lokasi yang sama ditemukan *string* tambahan yang mencurigakan Perintah tersebut diindikasikan untuk membaca nomor ID dari perangkat, seperti nomor IMEI dan nomor SIM.

d. *Domain Checker*
Penelitian dianjutkan dengna melakukan analisis terkait *domain* yang ada pada *source code* dan ditemukan tidak berbahaya.

C. *Detection Phase*
Tahapan ini dilakukan penelitian ulang dan peninjauan ulang terkait hasil analisis pada tahap sebelumnya. Hasil yang didapati adalah sebagai berikut:

a. *Android Permissions*
Hasil yang didapatkan dapat dari *source code* tersebut menggunakan android.permission.
*ACCESS*_NETWORK_STATE untuk memberikan akses informasi mengenai operator dari jaringan yang digunakan oleh pengguna Android *Application Programming Interface* (API).

b. Hasil dari analisis android API yang telah dilakukan, ditemukan terdapat beberapa *source code* atau *string* yang mencurigakan. Berdasarkan *string* yang ditemukan, diketahui aplikasi menggunakan API *Get Network Interface information* untuk meminta akses dalam mendapatkan informasi terkait *network information* hingga ke IP *address* bahkan hingga *host address* baik IPv4 maupun IPv6. Pada sampel aplikasi 3 ini belum diketahui apakah *malware* atau tidak. Sehingga membutuhkan analisis lanjutan berupa analisis dinamis.

c. *Domain Checker*
Setelah dilakukan pengecekan terhadap domain dan IP *Address* yang terdapat pada sampel dengan menggunakan VirusTotal, tidak terdapat adanya

indikasi yang dinilai berbahaya.

4.2. Analisis Dinamis

1. Sampel Aplikasi 1
   Hasil dari analisis dinamis sampel aplikasi 1 adalah sebagai berikut:
   a. Sampel aplikasi ini berjalan pada *background*, sehingga pengguna tidak mengetahui bagaimana aplikasi ini berjalan di latar belakang kecuali membuka pengaturan. Hal tersebut menimbulkan kemungkinan bahwa pengguna tidak menyadari bahwa aplikasi ini tidak terpasang pada perangkat dan membuat aplikasi ini akan tetap berada pada perangkat.
   b. Pengguna tidak dapat mencopot pemasangan dari aplikasi apabila sudah terpasang. Aplikasi akan tetap berjalan dan tidak dapat di*uninstall* kecuali dilakukan *reset* ulang sehingga pengguna dapat kehilangan data atau aplikasi lain yang sudah dipasang pada perangkat.
   c. Aplikasi mampu mendapatkan apapun yang disalin oleh pengguna pada *clipboard* mereka. Kemudian sampel aplikasi ini akan mencatat dan memungkinkan mengirimkan apa yang disalin.
   d. Aplikasi dapat mengakses aplikasi lain contohnya pesan dan kontak sehingga data apapun yang berada pada pesan dan kontak dapat diketahui oleh sampel aplikasi. Hal tersebut menimbulkan kemungkinan sampel aplikasi untuk mengirimkan data ke penyerang dan pengguna dapat kehilangan datanya.
   e. Aplikasi dapat mengirimkan data pengguna yang ada pada perangkat sehingga pengguna memungkinkan untuk kehilangan data yang bersifat sensitif. Terbukti dari adanya pengiriman paket ke tujuan tertentu yang berisikan *string* bertuliskan informasi dari perangkat.
2. Sampel Aplikasi 2
   Hasil dari analisis dinamis sampel aplikasi 2 serupa dengan sampel aplikasi 1.
3. Sampel Aplikasi 3
   Hasil dari analisis dinamis sampel aplikasi 3 yakni sebagai berikut:
   a. Aplikasi terlihat tidak dapat berjalan, namun aplikasi ini tetap mengirimkan paket keluar sehingga pengguna memungkinkan berpikir bahwa aplikasi ini rusak. Paket yang dikirimkan memungkinkan berisikan informasi sensitif dari pengguna, sehingga pengguna dapat kehilangan data yang dimiliki.
   b. Aplikasi ini dapat mendapatkan *access control* dari adanya informasi atas paket yang dikirimkan. Memungkinkan aplikasi ini dapat mengontrol dan mengirimkan berbagai data menuju penyerang sehingga pengguna dapat kehilangan data pribadinya.

5. KESIMPULAN

Berikut ini adalah detail dari kesimpulan atas hasil yang didapatkan dari analisis statis dan analisis dinamis yang telah dilakukan:

1. Ketiga sampel aplikasi yang diteliti merupakan *malware* dengan jenis *spyware*. Ketiga aplikasi tersebut mengirimkan informasi terkait pengguna ke penyerang tanpa diketahui oleh pengguna. Aplikasi tersebut mengirimkan informasi dari pengguna berdasarkan temuan adanya paket yang berisikan *string* terenkripsi yang ditujukan pada suatu *host* tertentu. Apabila dibandingkan dengan aplikasi MetaMask yang asli, tidak ditemukan adanya paket dengan isi seperti yang dijelaskan pada proses penelitian.
2. Ketiga sampel aplikasi yang diteliti memiliki karakteristik dan perilaku yang sama yakni tidak berjalan pada tampilan biasa, namun berjalan pada latar belakang atau *background*. Sehingga pengguna tidak mengetahui proses yang dilakukan. Sampel aplikasi 1 dan sampel aplikasi 2 berjalan pada *background* dengan menghilangkan aplikasi tersebut pada menu dan membuatnya menjadi tidak bisa di*uninstall*. Sampel aplikasi 3 berjalan pada *background* dengan membuat aplikasi ini ketika dijalankan memunculkan notifikasi bahwa aplikasi ini tidak dapat berjalan. Sampel aplikasi 1 dan sampel aplikasi 2 berdasarkan penelitian ditemukan bahwa berasal dari sumber yang sama namun memiliki *package name* yang berbeda dan nilai *hash* yang berbeda. Sehingga membuat terlihat berbeda, namun isi dari *source code* dan yang lainnya sama. Sedangkan sampel aplikasi 3 ini mirip dengan aplikasi yang asli terkait *permission* dan api yang dimiliki sehingga diperlukan analisis mendalam untuk dilihat mengenai keaslian dari sampel aplikasi.
3. Karakteristik yang terlihat sebelum adanya proses instalasi adalah sampel aplikasi 1 dan sampel aplikasi 2 meminta izin atau *permissions* yang sensitif seperti penyimpanan, kontak, sms dan seperti yang dijelaskan pada penelitian. Sampel aplikasi 3 ini memiliki *permission* yang sama dengan aplikasi yang asli sehingga dari segi luarnya tidak bisa terdeteksi biasa.
4. Ketiga sampel aplikasi yang diteliti memiliki dampak yang keseluruhan sama yakni adanya pengiriman paket keluar dan memungkinkan paket yang dikirimkan adalah data pribadi dari pengguna. Sehingga pengguna dapat kehilangan data pribadi mereka.

6. SARAN

Berdasarkan penelitian yang telah dilakukan, penelitian ini dapat memberikan informasi dan hasil yang sesuai berdasarkan pembatasan masalah. Saran pengembangan yang dapat dilakukan selanjutnya adalah sebagai berikut:

1. Menggunakan lebih banyak *tools* untuk membantu dalam proses analisis aplikasi khususnya aplikasi yang diidentifikasi sebagai *malware* agar dapat mendapatkan informasi lebih banyak dan lebih akurat.
2. Menambahkan aplikasi tiruan yang diteliti untuk memperkaya terkait hasil penelitian.
3. Melakukan analisis dinamis pada ponsel non emulator atau pada ponsel asli untuk melihat bagaimana cara kerjanya secara jelas.

REFERENSI

[1] T. H. Project, "*Honeynet Project Overview*," 2020, [Online]. Available: http://old.honeynet.org/speaking/honeynet_project-3.0.1.ppt.zip.

[2] I. A.Saeed, A. Selamat, and A. M. A. Abuagoub, "*A Survey on Malware and Malware Detection Systems*," Int. J. Comput. Appl., vol. 67, no. 16, pp. 25–31, 2013, doi:10.5120/11480-7108.

[3] F. Steinmetz, M. von Meduna, L. Ante, and I. Fiedler, "*Ownership, uses and perceptions of cryptocurrency: Results from a population survey*," Technol. Forecast. Soc. Change, vol. 173, no. January, p. 121073, 2021, doi:10.1016/j.techfore.2021.121073.

[4] Triple-A, "*Crypto Ownership Indonesia*," 2020. https://triple-a.io/crypto-ownershipindonesia/ (accessed Oct. 23, 2021).

[5] L. Abrams, "*Android banking malware infects 300,000 Google Play users*," 2021. https://www.bleepingcomputer.com/news/security/android-banking-malware-infects-300-000-google-play-users/ (accessed Dec. 12, 2021).

[6] J. A. Shaheen, M. A. Asghar, and A. Hussain, "*Android OS with its Architecture and Android Application with Dalvik Virtual Machine Review*," Int. J. Multimed. Ubiquitous Eng., vol. 12, no. 7, pp. 19–30, 2017, doi: 10.14257/ijmue.2017.12.7.03.

[7] W. K. Härdle, C. R. Harvey, R. C. G. Reule, W. K. H, and C. R. Harvey, "*International Research Training Group 1792 Understanding Cryptocurrencies Understanding Cryptocurrencies*," 2018.

[8] "*All Cryptocurrencies*." https://coinmarketcap.com/all/views/all/ (accessed Dec. 13, 2021).

[9] S. Talukder and Z. Talukder, "*A Survey on Malware Detection and Analysis Tools*," Int. J. Netw. Secur. Its Appl., vol. 12, no. 2, pp. 37–57, 2020, doi: 10.5121/ijnsa.2020.12203.

[10] M. Mirza, "*Malicious Software Detection, Protection & Recovery Methods a Survey*" Researchgate, no. September 2014, pp. 1–11, 2014, [Online]. Available: https://www.researchgate.net/publication/272997042_MALICIOUS_SOFTWARE_DETECTION_PROTECTION_RECOVERY_METHODS_A_SURVEY.

[11] S. Gadhiya, K. Bhavsar, and P. D. Student, "*Techniques for Malware Analysis*," Int. J.Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 4, pp. 2277–128, 2013.

[12] D. Uppal, V. Mehra, and V. Verma, "*Basic survey on Malware Analysis, Tools and Techniques*," Int. J. Comput. Sci. Appl., vol. 4, no. 1, pp. 103–112, 2014, doi:10.5121/ijcsa.2014.4110.

[13] C. Edu, "*What is Sandbox Security?*," 6 May, 2021. https://www.forcepoint.com/cyber-edu/sandbox-security (accessed Nov. 07, 2021).

[14] X. Wang, Y. Yang, and Y. Zeng, "*Accurate mobile malware detection and classification in the cloud*," Springerplus, vol. 4, no. 1, pp. 1–23, 2015, doi:10.1186/s40064-015-1356-1.

[15] Prof.Dr.Sugiyono, "Metode penelitian pendidikan (pendekatan kuantitatif, kualitatif dan r & d," p. 456, 2015.

[16] K. Bakour et al., "*The Android Malware Static Analysis: Techniques, Limitations, and Open Challenges*."

[17] K. Dunham, J. A. Morales, S. Hartman, M. Quintans, and T. Strazzerre, *Android Malware and Analysis*, vol. 0, no. 9781461473930. 2015