Jurnal Ilmiah Informatika (JIF)
| ISSN (Print) 2337-8379 | ISSN (Online) 2615-1049 |
Jurnal online di akses di http://ejournal.upbatam.ac.id/index.php/jif

Analisis Infeksi Malware Pada Perangkat Android Dengan Metode Hybrid Analysis

Asep Solahudin Rusdi, Nur Widiyasono, Heni Sulastri

Universitas Siliwangi, Jl. Siliwangi No. 24, Kota Tasikmalaya, Jawa Barat 46115, Indonesia
Universitas Siliwangi, Jl. Siliwangi No. 24, Kota Tasikmalaya, Jawa Barat 46115, Indonesia
Universitas Siliwangi, Jl. Siliwangi No. 24, Kota Tasikmalaya, Jawa Barat 46115, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 28 Agustus 2019
Revisi Akhir: 10 September 2019
Diterbitkan *Online*: 30 September 2019

KATA KUNCI

Android, Dynamic, Static, Hybrid, Malware

KORESPONDENSI

No HP: 08976971069
E-mail: 147006213@student.unsil.ac.id

A B S T R A C T

Android is the most widely used operating system in the world with a percentage of 76.82% in the Android operating system market share. This fact makes the developers of malicious software (malware) make mobile phone users with the Android operating system as the main target of malware attacks. Attackers can modify application code by entering malicious code, repacking the application and publishing the application in the Android application market. The malware samples used in this study were Judy adware and Marcher banking trojans. This study aims to determine the behavior or characteristics of the malware samples using static and dynamic analysis. The analysis shows that Judy committed ad-fraud by clicking on advertisements without the user's knowledge and Marcher had the ability to collect credential information from the victim's financial account.

1. PENDAHULUAN

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 [1].

Kebutuhan akan mobilitas membuat masyarakat di seluruh belahan dunia marak menggunakan *gadget* seperti *tablet* dan *smartphone* dengan berbagai macam sistem operasi. Sistem operasi untuk perangkat *smartphone* yang paling banyak digunakan adalah Android yakni sebanyak 76,82% dari total *market share* [2].

Malware pada platform Android menyusup lewat layanan distribusi aplikasi (*app store*), baik resmi (Google Play Store) maupun milik pihak ketiga, dengan menyamar menjadi aplikasi sah seperti pemutar video, permainan dan utilitas sistem. Beberapa jenis malware yang banyak beredar di *app store* diantaranya adalah *Adware*, *Banking Trojan* dan *Cryptocurrency-mining malware*. *Adware* merupakan salah satu malware yang paling banyak menyerang pengguna Google Play Store dan terus mengalami peningkatan hingga 36% sejak tahun 2016. Selain itu, serangan Banking Trojan juga meningkat hingga 12% sedangkan *Crypto-Mining* terjadi peningkatan sebesar 5% bersamaan dengan melambungnya harga Bitcoin pada saat itu [3].

Praktisi keamanan teknologi informasi perlu melakukan suatu proses investigasi forensik analisis malware untuk mengidentifikasi, mengamankan, mengamati, dan menyajikan fakta dan opini dari informasi malware. Beberapa metode dan teknik dalam melakukan analisis malware yaitu analisa secara statis (*Static Analysis*) dan analisa secara dinamis (*Dynamic Analysis*), dimana keduanya memiliki kelemahan dan kelebihan masing-masing [4]. *Hybrid Analysis* merupakan gabungan dari *Static* dan *Dynamic Analysis* yang efektif untuk mendeteksi infeksi malware secara akurat [5].

Penelitian ini mencoba menganalisis *malware* Judy, yaitu *adware* yang banyak beredar di Google Play Store, dan Marcher, yaitu salah satu *banking trojan* yang mengambil informasi *e-banking* nasabah, dengan menggunakan metode *hybrid analysis*. Penelitian ini diharapkan dapat memberikan gambaran cara kerja kedua malware tersebut dan dampak yang diakibatkannya terhadap sistem Android.

Berdasarkan latar belakang yang telah dipaparkan, permasalahan yang akan dibahas adalah bagaimana proses

analisis malware dengan menggunakan metode *hybrid analysis*, bagaimana karakteristik dari sampel malware Judy dan Marcher, serta bagaimana langkah pencegahan agar terhindar dari infeksi malware pada perangkat Android.

Batasan masalah dalam penelitian ini adalah analisis hanya dilakukan untuk pengamatan terhadap sampel malware dan dampak serangannya, bukan untuk memperbaiki sistemnya, penelitian dilakukan pada dua sampel malware yaitu Judy dan Marcher, tidak terpasang antivirus pada platform Android, serta proses eksekusi malware dilakukan pada Android Emulator versi Lollipop.

Adapun tujuan dalam penelitian ini adalah mengetahui proses analisis malware dengan menggunakan metode *hybrid analysis, m*engetahui karakteristik dari sampel malware, serta memberikan informasi langkah pencegahan agar terhindar dari infeksi malware pada perangkat Android.

Manfaat dalam penelitian ini adalah menambah wawasan tentang karakteristik malware pada platform Android, khususnya malware Judy dan Marcher, meningkatkan tingkat pendeteksian malware serupa pada platform Android dengan melihat karakteristiknya, serta menjadi rujukan untuk melakukan pengembangan penelitian lebih lanjut dan dapat menjadi dasar untuk pembuatan anti-malware untuk jenis malware serupa.

2. TINJAUAN PUSTAKA

Sistem Operasi Android

Android, sebagai sebuah sistem, adalah sistem operasi berbasis Java yang berjalan pada *kernel* 2.6 *Linux*. Aplikasi Android yang dikembangkan menggunakan Java dan mudah menyesuaikan ke *platform* baru. Sistem operasi Android dapat digambarkan sebagai jembatan antara *smartphone* dan penggunanya, sehingga pengguna dapat mengoperasikan *smartphone* dan dapat menjalankan aplikasi yang terdapat pada *smartphone* tersebut. Android adalah sebuah sistem operasi berbasis linux pada perangkat *mobile* yang mencakup sistem operasi, *middle ware* dan aplikasi-aplikasi didalamnya. Kelas utama perangkat yang didukung oleh sistem operasi Android adalah perangkat *mobile*, namun sekarang ini sudah dikembangkan sehingga Android dapat digunakan sebagai sistem operasi pada *electronic book readers*, *netbooks*, *tablet*, dan *set-top boxes* (STB) [6].

Malware

Malware (*malicious software*) adalah perangkat lunak yang dapat mengganggun kinerja sistem operasi komputer seperti mencuri informasi data sensitif dan melakukan *remote* pada komputer korban tanpa seizin pemilik [2]. *Malware* ada dalam berbagai bentuk seperti *script*, *code*, *activecontent*, dan perangkat lunak. M*alware* adalah sebuah perangkat lunak yang dapat merusak perangkat lunak lain atau membuat perangkat keras bekerja lebih keras karena harus memfasilitasi eksekusi program di luar konteks yang seharusnya [7].

Analisis Malware

1. Analisis Statis

Metode analisis statis dilakukan untuk menyelidiki aplikasi Android tanpa perlu dijalankan. File aplikasi Android (*.apk)* berisi kode sumber yang terkompilasi (*classes.dex*), string dan definisi konstan, *resource* gambar, dan file manifes aplikasi (*AndroidManifest.xml*) yang mendefinisikan metadata tentang aplikasi seperti izin yang diminta, nama *package*, versi, referensi *library*, dan komponen aplikasi (misalnya *Activity*). [4]

2. Analisis Dinamis

Analisis malware secara Dinamis dilakukan dengan menjalankan malware pada lingkungan *virtual sandbox* untuk mengetahui perilaku dari malware terhadap sistem. Beberapa informasi yang bisa didapat dari analisis dinamis yaitu pemanggilan API, pemanggilan sistem, penulisan memori, *data and network capture*, dan sebagainya [8].

3. *Hybrid Analysis*

Metode *Hybrid Analysis* merupakan penggabungan dari analisis statis dan analisis dinamis dengan memeriksa *signature* yang terdapat pada malware kemudian melihat perilaku (*behaviour*) malware terhadap sistem yang terinfeksi [5]. Analisa secara hybrid menggabungkan kedua metode tersebut dengan mempertimbangkan kelebihan dan kekurangan yang dimiliki oleh metode analisa statis maupun dinamis sehingga dapat melengkapi kelemahan antara satu dengan yang lain [9].

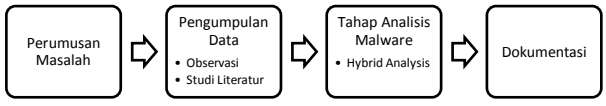
Kajian Penelitian Terdahulu

Penelitian yang dilakukan oleh [8] mempelajari bagaimana suatu *malware* dapat menginfeksi perangkat Android, memasang dan menjalankan *malware* tersebut pada perangkat Android yang menjadi target. Proses penelitian tersebut dilakukan menggunakan berbagai *tool* yang membantu dalam proses analisis statis dan analisis dinamis untuk mempelajari fungsionalitas dan cara kerja dari *malware* yang diteliti.

Penelitian terdahulu ini memiliki konteks atau jenis yang sama dengan penelitian yang akan dilakukan. Perbedaannya terletak pada *software* yang digunakan dan objek penelitian berdasarkan laporan ancaman keamanan terkini. Kontribusi dari penelitian ini adalah mengetahui karakteristik dari sampel malware jenis *Adware* dan *Banking Trojan* serta dampak yang ditimbulkan terhadap sistem yang terinfeksi dengan menggabungkan analisis statis dan dinamis (*Hybrid Analysis*).

3. METODOLOGI

Metodologi yang digunakan pada penelitian ini adalah metode eksperimental yaitu menemukan hubungan sebab-akibat dan pengaruh faktor-faktor pada kondisi tertentu, khususnya infeksi *malware* terhadap perangkat Android, dan dilakukan dalam lingkungan yang terkendali berupa *virtual lab*. Alur dari metodologi penelitian dapat dilihat pada gambar 1.



Gambar 1. Tahapan Penelitian

Proses perumusan masalah didapat dari salah satu laporan keamanan *smartphone* yang dirilis oleh McAfee, Inc. [3] yang menunjukkan bahwa serangan *malware* jenis *Adware* dan *Banking Trojan* terus mengalami peningkatan masing-masing hingga 36% dan 12% sejak tahun 2016.

Metode pengumpulan data dilakukan dengan mengidentifikasi dan membuat alur penelitian yang akan dilaksanakan, agar proses pencarian data tidak terjadi penyimpangan dalam mengemukakan suatu tujuan yang ingin dicapai.

Metode analisis *malware* yang digunakan pada penelitian ini adalah *Hybrid Analysis*, yaitu dengan menggabungkan teknik analisis statis dengan dinamis. *Hybrid Analysis* mengintegrasikan data yang didapat pada analisis statis dan dinamis untuk mendeteksi perilaku dan fungsionalitas yang mencurigakan, sehingga dapat meningkatkan tingkat akurasi pendeteksian *malware.*

Dokumentasi menyimpan hasil keluaran data dan informasi proses s*can* sample *malware* yang didapat dari *tools* analisis, untuk kemudian disajikan dalam laporan penelitian.

4. HASIL DAN PEMBAHASAN

Pra-Analisis

Malware yang akan digunakan sebagai objek penelitian, yakni *Judy* dan *Marcher*, masing-masing dapat dilihat pada tabel 1 dan 2. Informasi *malware* yang akan digunakan yaitu sebagai berikut:

Tabel 1. Informasi *Malware Judy* (sumber: *virustotal*)

Nama Malware	*a.expense.Judy*
SHA256	4d1503ef789d31047d39efe28e7abae3104e0b7d0ded9bf899fd92f814246718
Rasio Deteksi	25 / 60
Waktu Analisis	2019-02-23 09:34:38 UTC
Ukuran file	44.6 MB
Tipe File	APK

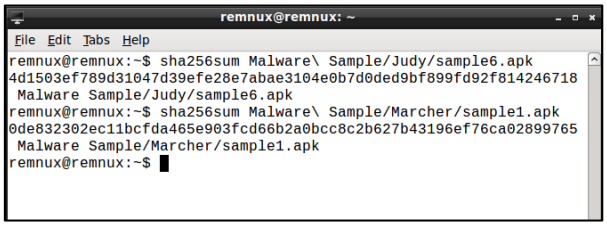
Informasi yang diperoleh dari *virustotal* yaitu sampel dari *malware Judy* memiliki nilai *checksum* SHA256 (*Secure Hash Algorithm*) 4d1503ef789d31047d39efe28e7abae3104e0b7d0ded9bf899fd92f814246718. Rasio pendeteksian dari 60 *anti-malware* hanya 25 *anti-malware* yang dapat mendeteksi *malware Judy*. File sampel berukuran 44.6 MB dengan tipe file APK yang merupakan format berkas untuk mendistribusikan dan memasang perangkat lunak pada ponsel dengan sistem operasi Android.

Tabel 2. Informasi Malware *Marcher* (sumber: *virustotal*)

Nama Malware	*Android-Trojan/Marcher.18614*
SHA256	0de832302ec11bcfda465e903fcd66b2a0bcc8c2b627b43196ef76ca02899765
Rasio Deteksi	31 / 58
Waktu Analisis:	2019-02-23 09:37:50 UTC
Ukuran file:	560.73 KB
Tipe File	APK

Informasi yang didapat dari *virustotal* yaitu sampel *malware Marcher* memiliki nilai *checksum* SHA256 (*Secure Hash Algorithm*) 0de832302ec11bcfda465e903fcd66b2a0bcc8c2b627b43196ef76ca02899765. Rasio pendeteksian dari 58 *anti-malware* hanya 31 *anti-malware* yang dapat mendeteksi *malware Marcher*. File sampel berukuran 560.73 KB dengan tipe file APK yang merupakan format berkas untuk mendistribusikan dan memasang perangkat lunak pada ponsel dengan sistem operasi Android.

Data Integrity atau keaslian dari file sampel dapat diperiksa dengan membandingkan informasi *checksum* yang didapat dari *virustotal* dan nilai *checksum* file sampel yang dilihat menggunakan *tool SHA265Sum*. Hasil pemeriksaan nilai *checksum* menggunakan *tool SHA265Sum* dapat dilihat pada gambar 2.



Gambar 2. Pemeriksaan Nilai *Checksum* dengan *SHA256Sum*

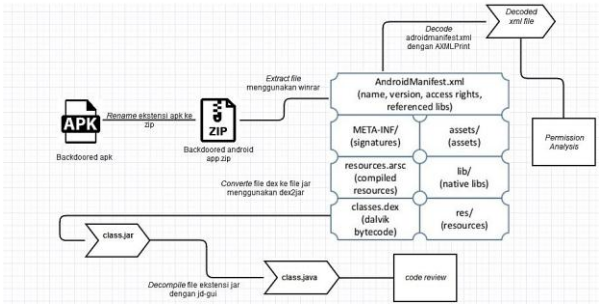
Hasil pemeriksaan nilai *checksum* SHA256 menunjukkan bahwa nilai yang dihasilkan oleh *SHA265Sum* untuk masing-masing sampel *malware* sama dengan informasi yang didapat dari *virustotal*, artinya file tersebut merupakan file asli (identik) dan belum mengalami modifikasi. Perbandingan nilai *checksum* dapat dilihat pada tabel 3.

Tabel 3. Perbandingan Nilai *Checksum VirusTotal* dengan *SHA256Sum*

No	Nama *Malware*	Nilai *Checksum*		Keterangan
		VirusTotal	SHA256Sum	
1	*Judy*	4d1503ef789d31047d39efe28e7abae3104e0b7d0ded9bf899fd92f814246718	4d1503ef789d31047d39efe28e7abae3104e0b7d0ded9bf899fd92f814246718	Identik
2	*Marcher*	0de832302ec11bcfda465e903fcd66b2a0bcc8c2b627b43196ef76ca02899765	0de832302ec11bcfda465e903fcd66b2a0bcc8c2b627b43196ef76ca02899765	Identik

Static Analysis

Analisis statis yang dilakukan pada penelitian ini menggunakan teknik *reverse engineering* untuk mendapatkan Java *source code* dari aplikasi Android yang telah terinfeksi oleh *malware*, lalu dilakukan analisis untuk mengetahui kode yang bersifat merusak. Alur *reverse engineering* dapat dilihat pada gambar 3.



Gambar 3. Alur *reverse engineering* file APK

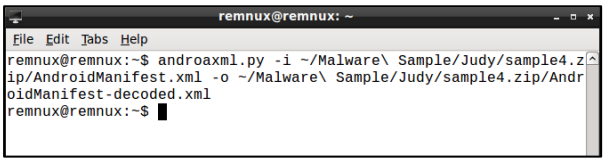
Hal pertama yang dilakukan pada analisis statis dengan teknik *reverse engineering* sampel *malware* adalah mengubah ekstensi file yang mulanya adalah **.apk* menjadi **.zip* agar

konten dari sampel dapat diekstrak menggunakan *software file archiver*. Konten yang telah diekstrak memiliki struktur seperti pada gambar 3. Analisis dilakukan khususnya pada file *AndroidManifest.xml* yang memuat izin apa saja yang diberikan oleh sistem terhadap aplikasi dan *classes.dex* yang merupakan *source code* dari aplikasi yang telah diubah ke dalam *Dalvik Bytecode*.

Judy

a. *Permission Analysis*

File *AndroidManifest.xml* mengandung informasi yang dibutuhkan oleh sistem Android, seperti *version number*, *metadata*, *package name*, fungsi dan perizinan yang dibutuhkan oleh aplikasi ketika proses pemasangan atau pada saat *runtime*. Proses *decode* dilakukan menggunakan *tool androaxml.py* seperti yang tertera pada gambar 4.



Gambar 4. Proses *decode AndroidManifest.xml malware Judy*

Program *androaxml.py* dieksekusi lewat terminal dengan menggunakan parameter *-i* diikuti dengan alamat file *input* (.../*AndroidManifest.xml*) dan *-o* diikuti dengan alamat file *output* (.../*AndrodManifest-decoded.xml*). File *output* yang telah ter-*decode* dibuka menggunakan aplikasi pengolah teks dan menampilkan *source code* berupa informasi perizinan yang digunakan oleh sampel malware.

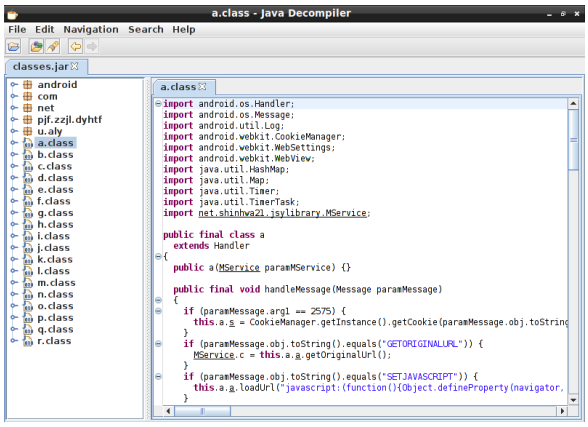
Perizinan yang didapat dari hasil *decode* dikelompokkan berdasarkan tingkat keamanan tersebut untuk mengetahui potensi kerusakan yang akan ditimbulkan oleh malware *Judy*.

Tabel 4. Tingkat keamanan perizinan *malware Judy*

No	Permission	Status
1	android.permission. GET_ACCOUNTS	*Dangerous Permission*
2	android.permission. WAKE_LOCK	*Normal Permission*
3	android.permission. INTERNET	*Normal Permission*
4	android.permission. ACCESS_NETWORK_STATE	*Normal Permission*
5	android.permission. ACCESS_WIFI_STATE	*Normal Permission*
6	android.permission. READ_EXTERNAL_STORAGE	*Dangerous Permission*
7	android.permission. WRITE_EXTERNAL_STORAGE	*Dangerous Permission*
8	android.permission. CHANGE_WIFI_STATE	*Normal Permission*
9	android.permission. GET_TASKS	*Normal Permission*
10	android.permission. RECEIVE_BOOT_COMPLETED	*Normal Permission*
11	android.permission. READ_PHONE_STATE	*Dangerous Permission*
12	android.permission. SYSTEM_ALERT_WINDOW	*Dangerous Permission*
13	com.android.vending. BILLING	*Normal Permission*
14	android.permission. ACCESS_BACKGROUND_ SERVICE	*Normal Permission*

b. *Code Review*

Classes.dex merupakan file *executable* yang dapat berjalan pada *Dalvik Virtual Machine*, dalam file ini terdapat *class* yang dibutuhkan oleh aplikasi pada saat *runtime*. Langkah yang dilakukan pada tahap ini adalah mengubah file *classses.dex* ke dalam format **.jar* menggunakan *tool Dex2Jar*, kemudian melakukan *decompile* dengan *tool JD-GUI* agar struktur *class* dan *source code* dalam bahasa *Java* dapat dianalisis.

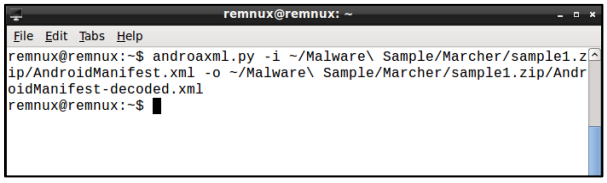


Gambar 5. Struktur *class* dan *source code Java malware Judy*

Marcher

a. *Permission Analysis*

Proses *permission analysis* yang dilakukan pada sampel *malware Marcher* sama seperti sebelumnya, yaitu dengan melakukan *decode* terhadap file *AndroidManifest.xml* menggunakan *tool androaxml.py* yang dapat dilihat pada gambar 4.6.



Gambar 6. Proses *decode AndroidManifest.xml malware Marcher*

Perizinan yang didapat dari hasil *decode* dikelompokkan berdasarkan tingkat keamanan tersebut untuk mengetahui potensi kerusakan yang akan ditimbulkan oleh malware *Marcher*.

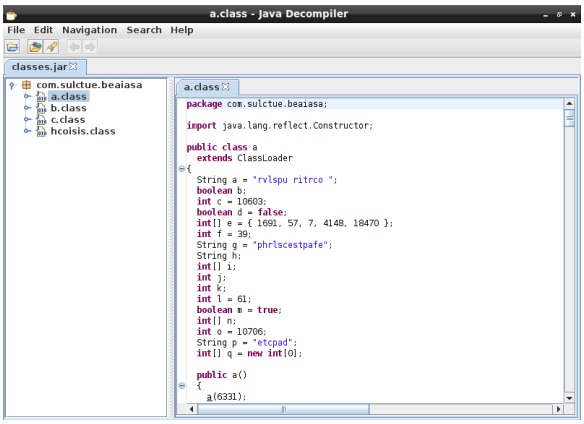
Tabel 5. Tingkat Keamanan Perizinan *malware Marcher*

No	Permission	Status
1	android.permission. RECEIVE_BOOT_COMPLETED	*Normal Permission*
2	android.permission. WAKE_LOCK	*Normal Permission*
3	android.permission. RECEIVE_SMS	*Dangerous Permission*
4	android.permission. SEND_SMS	*Dangerous Permission*
5	android.permission. READ_SMS	*Dangerous Permission*
6	android.permission. WRITE_SMS	*Dangerous Permission*

7	android.permission. CALL_PHONE	*Dangerous Permission*
8	android.permission. READ_PHONE_STATE	*Dangerous Permission*
9	android.permission. ACCESS_NETWORK_STATE	*Normal Permission*
10	android.permission.INTERNET	*Normal Permission*
11	android.permission. READ_CONTACTS	*Dangerous Permission*
12	android.permission. GET_TASKS	*Dangerous Permission*
13	android.permission. WRITE_SETTINGS	*Signature Permission*
14	android.permission.VIBRATE	*Normal Permission*
15	android.permission. USES_POLICY_FORCE_LOCK	*Dangerous Permission*
16	android.permission. ACCESS_WIFI_STATE	*Normal Permission*
17	android.permission. CHANGE_WIFI_STATE	*Normal Permission*
18	android.permission. CHANGE_NETWORK_STATE	*Normal Permission*
19	com.android.browser.permission. READ_HISTORY_BOOKMARKS	*Normal Permission*

b. *Code Review*

Classes.dex merupakan file *executable* yang dapat berjalan pada *Dalvik Virtual Machine*, dalam file ini terdapat *class* yang dibutuhkan oleh aplikasi pada saat *runtime*. Langkah yang dilakukan pada tahap ini adalah mengubah file *classses.dex* ke dalam format **.jar* menggunakan *tool Dex2Jar*, kemudian melakukan *decompile* dengan *tool JD-GUI* agar struktur *class* dan *source code* dalam bahasa *Java* dapat dianalisis.

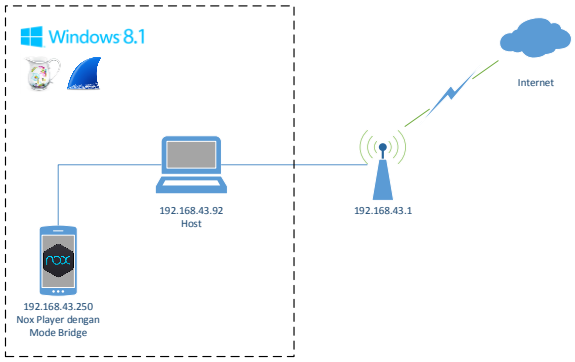


Gambar 7. Struktur *class* dan *source code Java malware Marcher*

Analisis statis dilakukan dengan memeriksa fungsi-fungsi yang terdapat pada kelas *source code* untuk menemukan kode pemrograman yang bersifat *malicious*. Beberapa fungsi *malicous* yang ditemukan pada *Marcher* diantaranya adalah mengumpulkan informasi perangkat smartphone (IMEI, negara, nomor telepon, operator seluler, versi Android, model dan tipe perangkat, aplikasi terpasang), membaca SMS, menghalau antivirus, mendapatkan hak akses Admin, menyadap kredensial akun *mobile banking*, serta mengirimkan informasi yang didapat ke server C&C.

Dynamic Analysis

Proses analisis dinamis dilakukan pada perangkat virtual (*emulator*) Android versi 5.1 (*Lollipop*) dengan menggunakan *tool* Inspeckage untuk tahap *runtime analysis* dan Charles serta Wireshark untuk tahap *network analysis*. Tiap tahap analisis dinamis dilakukan melalui jaringan lokal (LAN) dengan topologi seperti pada gambar 8.



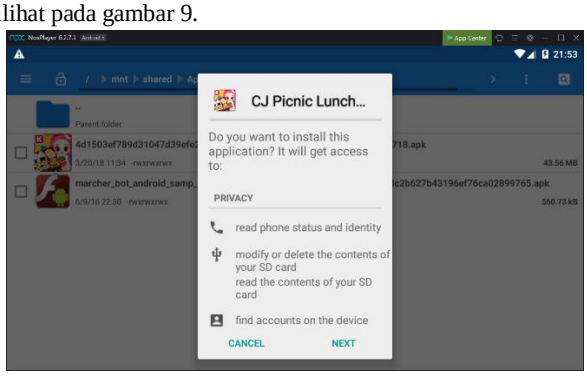
Gambar 8. Topologi Jaringan Analisis Dinamis

Sampel malware dipasang pada perangkat virtual kemudian dijalankan (*install & run*) untuk dilakukan analisis terhadap karakteristik dari sampel malware (*runtime analysis*) beserta aktifitas jaringan yang terekam (*network analysis*).

Judy

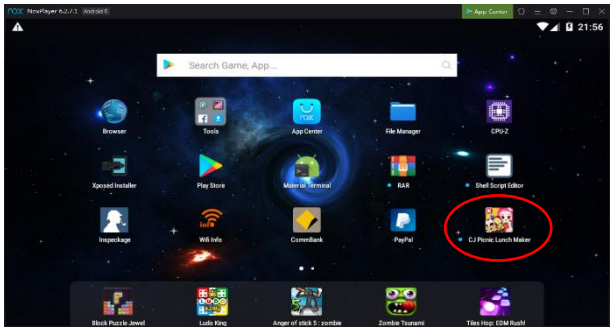
a. *Install & Run*

Pemasangan sampel malware *Judy* dilakukan pada perangkat virtual *Nox Player*. *Judy* menyerupai aplikasi permainan untuk mengelabui korban. Perizinan yang diminta oleh malware *Judy* dapat terlihat secara ringkas pada dialog instalasi. Tampilan pemasangan sampel malware *Judy* dapat dilihat pada gambar 9.



Gambar 9. Instalasi Sample Malware Judy

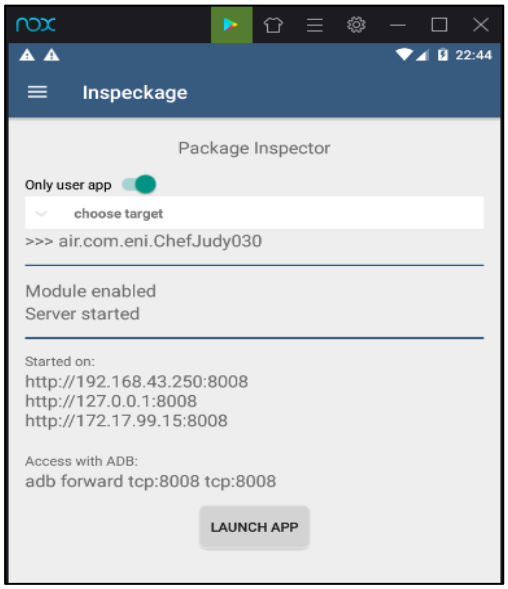
Pintasan malware *Judy* yang terpasang dapat ditemukan pada tampilan beranda *launcher* layaknya aplikasi permainan pada umumnya, seperti yang terlihat pada gambar 10.



Gambar 10. Pintasan Malware *Judy* pada *Launcher*

b. *Runtime Analysis*

Analisis dilakukan dengan menjalankan sampel malware dan melakukan monitoring menggunakan Inspeckage melalui jaringan lokal dengan menjalankan *service* Inspeckage pada perangkat virtual yang telah terinfeksi. *Service* Inspeckage akan menyediakan antarmuka berbasis web yang dapat diakses melalui jaringan, *Androd Debug Bridge* (ADB), atau perangkat virtual (*localhost*) dengan alamat seperti yang tertera pada gambar 11.

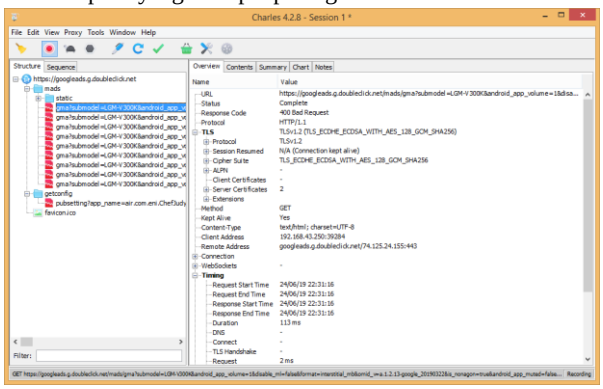


Gambar 11. Tampilan Service Inspeckage - *Judy*

Antarmuka berbasis web dari Inspeckage diakses pada web browser komputer *host* dengan alamat `http://192.168.43.250:8008` untuk melihat hasil monitoring yang dilakukan pada aktivitas *Judy* terhadap sistem pada saat *runtime*. Inspeckage mendeteksi informasi paket seperti *activity, services, requested permission* dan *broadcast receiver* yang terdapat pada *Judy*.

c. *Network Analysis*

Charles dijalankan pada komputer *host* dengan alamat IP `192.168.43.92:8888`. IP dan *port* dari komputer *host* digunakan sebagai *proxy* pada perangkat *virtual* sehingga semua *traffic* HTTP(S) dapat terpantau pada *tool Charles*. Hasil dari monitoring *traffic* pada perangkat virtual yang telah terinfeksi oleh *Marcher* menunjukkan bahwa malware berusaha melakukan komunikasi terhadap *C&C server* dengan metode *POST* seperti yang terdapat pada gambar 12.



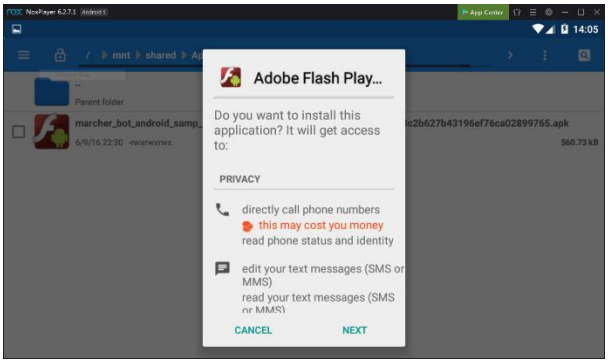
Gambar 12. Monitoring *HTTP(S) Traffic* dengan *Charles*

Gambar 12 menunjukkan *adware* Judy melakukan request ke alamat *https://googleads.g.doubleclick.net/* dengan mengirimkan informasi berupa atribut model smartphone yang digunakan, versi android yang terpasang, dan sebagainya untuk menampilkan iklan. Respon yang diterima adalah *400 Bad Request*, artinya server tidak mengenali *request* yang dikirim oleh *adware Judy*.

Marcher

a. *Install & Run*

Pemasangan sampel malware *Marcher* dilakukan pada perangkat virtual *Nox Player*. *Marcher* menyerupai aplikasi sah *Adobe Flash Player* untuk mengelabui korban. Perizinan yang diminta oleh malware *Marcher* dapat terlihat secara ringkas pada dialog instalasi. Tampilan pemasangan sampel malware *Marcher* dapat dilihat pada gambar 13.

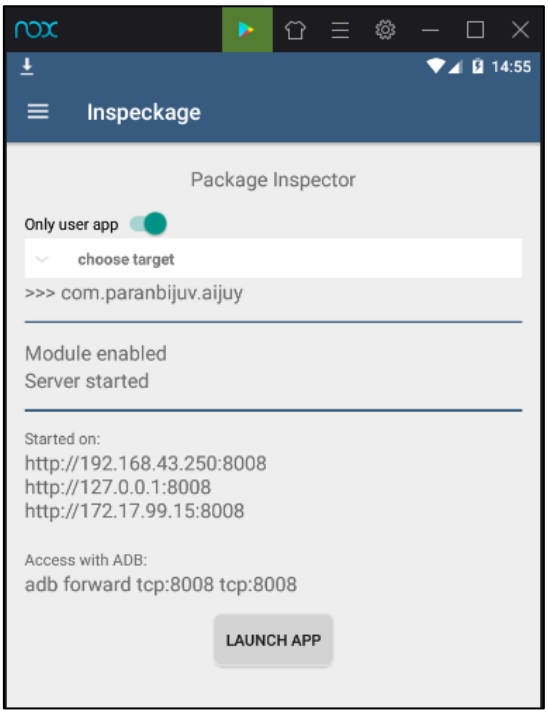


Gambar 13. Instalasi Sampel Malware *Marcher*

Sampel malware *Marcher* yang telah terpasang kemudian dijalankan dan terlihat bahwa *Marcher* berusaha menyembunyikan diri dengan menghapus pintasan pada menu utama agar keberadaannya tidak diketahui oleh *user*.

b. *Runtime Analysis*

Analisis dilakukan dengan menjalankan sampel malware dan melakukan monitoring menggunakan Inspeckage melalui jaringan lokal dengan menjalankan *service* Inspeckage pada perangkat virtual yang telah terinfeksi. *Service* Inspeckage menyediakan antarmuka berbasis web yang dapat diakses melalui jaringan, *Android Debug Bridge* (ADB), atau perangkat virtual (*localhost*) dengan alamat seperti yang tertera pada gambar 14.



Gambar 14.Tampilan Service Inspeckage - Marcher

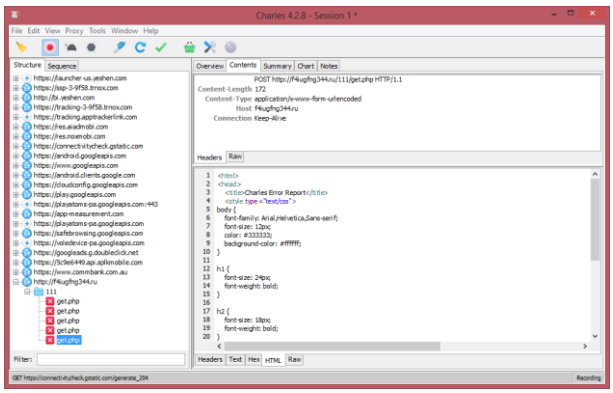
Antarmuka berbasis web dari Inspeckage diakses pada web browser komputer *host* dengan alamat `http://192.168.43.250:8008` untuk melihat hasil monitoring yang dilakukan pada aktivitas *Marcher* terhadap sistem pada saat *runtime*. Inspeckage mendeteksi informasi paket seperti *activity, services, requested permission,* dan *broadcast receiver* yang terdapat pada *Marcher*.

Inspeckage mendeteksi aktifitas pada tab *Shared Preferences* yang mana merupakan pengaturan dari malware *Marcher* yang terdapat pada file `main_prefs.xml`. File tersebut mengandung daftar aplikasi dan situs *mobile banking* yang diduga merupakan target dari penyerang/pembuat malware.

Hasil analisis yang dilakukan pada file `main_prefs.xml` ialah terdapat 3 (tiga) *array* bertipe *string* masing-masing bernama `default_apps_json`, `default_browsers_ apps_json`, dan `server_api_json` yang berpotensi mengandung target serangan berupa aplikasi dan halaman web serta alamat *Command & Control* (*C&C*) *Server* dari malware *Marcher*.

c. *Network Analysis*

Charles dijalankan pada komputer *host* dengan alamat IP `192.168.43.92:8888`. IP dan *port* dari komputer *host* digunakan sebagai *proxy* pada perangkat *virtual* sehingga semua *traffic* HTTP(S) dapat terpantau pada *tool Charles*. Hasil dari monitoring *traffic* pada perangkat virtual yang telah terinfeksi oleh *Marcher* menunjukkan bahwa malware berusaha melakukan komunikasi terhadap *C&C server* dengan metode *POST* seperti yang terdapat pada gambar 4.27.



Gambar 15. Monitoring *HTTP(S) Traffic* dengan *Charles*

Gambar 15 menunjukkan bahwa malware *Marcher* melakukan komunikasi dengan metode *POST* ke alamat `http://f4iugfng344.ru /111/get.php`. Respon yang diterima adalah laman galat pada sisi *Domain Name Server* (*DNS*), artinya *DNS* gagal mentranslasikan nama domain ke alamat IP sehingga kemungkinan besar *C&C server* dari sampel malware *Marcher* telah dimatikan (*down*).

Riwayat dari *C&C server Marcher* dapat dilihat dengan melihat *DNS History* dari alamatnya untuk mengetahui informasi mengenai sumber malware sebelum *C&C server* dimatikan. Hasil penelusuran menunjukkan bahwa domain `f4iugfng344.ru` terdaftar pada daftar hitam (*black list*) yang dirilis oleh *Infoblox* yang dapat dilihat pada tabel 6.

Tabel 6. *Infoblox Threat Protection Rules* [10]

Rule ID	***Rule Type***	***Rule Name***	***Description***
12500 1508	*System*	*DROP UDP MOBILE_MALWARE Trojan-Banker.Android OS.Marcher DNS UDP Lookup (f4iugfng344.ru)*	*This rule drops MOBILE_MALWARE Trojan-Banker.AndroidOS.Mar cher DNS Lookup using UDP (f4iugfng344.ru)*
12500 1509	*System*	*DROP TCP MOBILE_MALWARE Trojan-Banker.Android OS.Marcher DNS TCP Lookup (f4iugfng344.ru)*	*This rule drops MOBILE_MALWARE Trojan-Banker.AndroidOS.Mar cher DNS Lookup using TCP (f4iugfng344.ru)*

Karakteristik

Berikut adalah karateristik yang dimiliki oleh masing-masing malware:

a. *Judy*

1. Menyamar sebagai aplikasi permainan.
2. Mengakses klik iklan yang dilakukan secara *background* tanpa sepengetahuan pengguna.

3. Mengumpulkan informasi perangkat yang terinfeksi lewat *ad tracking.*

b. *Marcher*

1. Memiliki kemampuan untuk menyembunyikan diri setelah malware terpasang.
2. Mengumpulkan informasi perangkat yang terinfeksi seperti IMEI, versi Android, operator seluler, nomor telepon.
3. Membaca isi pesan SMS pada perangkat untuk mendapatkan kode pengaman *Two-Factor Authentication* (2FA).
4. Melakukan *phising* dengan menampilkan *Webview Overlay* untuk menutupi halaman login aplikasi sah.
5. Mengirimkan informasi yang didapat ke server C&C.

Pencegahan

Pencegahan yang dapat dilakukan untuk melindungi perangkat *smartphone* dari *malware* secara umum:

1. Selalu *update* versi sistem operasi Android jika tersedia.
2. Tidak mengunduh aplikasi dari penyedia yang tidak terpercaya.
3. *Disable* fitur *install from unknown resource* yang terdapat di *developer option.*
4. Pasang antivirus khusus perangkat Android dan pastikan *database*-nya selalu *update.*
5. Gunakan fitur *screen lock* pada perangkat Android, untuk memastikan perangkat aman dari pemasangan *malware* secara manual.

5. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil dari analisis malware *Judy* dan *Marcher* menggunakan metode analisis statis dan dinamis pada perangkat dengan sistem operasi Android 5.1 (*Lollipop*), ditemukan hasil diantaranya :

1. Analisis dengan metode statis dan dinamis dapat mendeteksi karakteristik dari malware pada perangkat Android dengan kelebihan dan kekurangan masing-masing. Analisis statis dengan teknik reverse engineering dapat dilakukan untuk memahami fungsi malware dengan melihat perijinan yang digunakan dan kode sumber. Analisis dinamis dapat dilakukan untuk melihat bagaimana perilaku dari malware ketika dipasang dan dijalankan serta aktifitas jaringan.
2. Judy memiliki karakteristik antara lain menyamar sebagai aplikasi permainan, melakukan klik iklan tanpa sepengetahuan pengguna yang dilakukan secara background (*ad-fraud*), serta mendapatkan informasi perangkat lewat *ad-tracking.*
3. Marcher memiliki karakteristik diantaranya menyembunyikan diri setelah malware terpasang, mengumpulkan informasi perangkat yang terinfeksi, membaca pesan SMS untuk melewati *Two-Factor Authentication*, melakukan phising dengan Webview Overlay serta mengirim informasi yang didapat ke server C&C.
4. Langkah pencegahan yang dapat dilakukan untuk mencegah infeksi malware pada perangkat Android yaitu dengan memperbarui versi sistem operasi Android jika tersedia, tidak mengunduh aplikasi dari penyedia yang tidak terpercaya, mematikan fitur *install from unknown resource* yang terdapat pada *developer option*, gunakan fitur *screen* lock, serta pasang antivirus khusus perangkat Android dan pastikan *database*-nya selalu terbaru.

Saran

Saran ini ditujukan untuk pengerjaan penelitian berikutnya, melalui laporan ini diharapkan dapat mengembangkan sistem yang dapat menganalisis secara otomatis menggunakan *machine learning* sehingga analisis terhadap sampel malware yang masif lebih cepat dan efisien. Penelitian serupa juga dapat dilakukan untuk investigasi dan analisis malware pada perangkat *mobile* lainnya untuk mempelajari cara kerja malware pada sistem yang berbeda.

UCAPAN TERIMA KASIH

Penulis menyampaikan banyak terimakasih ke berbagai pihak dalam penyelesaian penelitian ini, diantaranya :

1. Nur Widiyasono, M.Kom., CEH., CHFI., selaku pembimbing I yang telah memberikan bimbingan dan pengarahan kepada penulis.
2. Heni Sulastri, M.T., selaku pembimbing II yang telah memberikan bimbingan dan pengarahan kepada penulis.

DAFTAR PUSTAKA

[1] Y. N. Kunang, "Analisis Forensik Malware pada Platform Android," *Konferensi Nasional Ilmu Komputer (KONIK)*, pp. 141-148, 2014.

[2] StatCounter, "Mobile Operating System Market Share Worldwide," 2018. [Online]. Available: http://gs.statcounter.com/os-market-share/mobile/worldwide.

[3] McAfee Inc., "McAfee Mobile Threat Report Q1, 2018," McAfee Inc., California, 2018.

[4] S. Gadhiya dan K. Bhavsar, "Techniques for Malware Analysis," *International Journal of Advanced Research in Computer Science and Software Engineering*, pp. 972-975, 2013.

[5] N. Zalavadiya dan P. D. Sharma, "A Methodology of Malware Analysis, Tools and Technique for windows platform – RAT Analysis," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 5042-5054, 2017.

[6] S. Herlambang, Deteksi Malware Android Berdasarkan System Call Menggunakan Algoritma Support Vector Machine, Malang: UMM, 2018.

[7] Wandera, "The current state of mobile malware," Wandera, San Francisco, 2017.

[8] R. Alghamdi, K. Alfalqi dan M. Waqdan, "Android Platform Malware Analysis," *(IJACSA) International Journal of Advanced Computer Science and Applications*, pp. 140-146, 2015.

[9] R. S. Wardhana, "Analisa Hybrid untuk Sistem Deteksi Malware Otomatis dengan Support Vector Model Classifier," pp. 1-10, 2015.

[10] Infoblox, "Infoblox Administrator Guide," Infoblox Technical Publication, California, 2018.

BIODATA PENULIS

Asep Solahudin Rusdi
Mahasiswa jurusan Teknik Informatika, Fakultas Teknik, Universitas Siliwangi, Tasikmalaya.

Nur Widiyasono
Dosen jurusan Teknik Informatika, Fakultas Teknik, Universitas Siliwangi, Tasikmalaya.

Heni Sulastri
Dosen jurusan Teknik Informatika, Fakultas Teknik, Universitas Siliwangi, Tasikmalaya.