Jurnal Elektronik Ilmu Komputer Udayana
Volume 10, No 3. February 2022

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment

I Kadek Aldy Oka Ardita[a1], I Gusti Ngurah Anom Cahyadi Putra[a2], Mohammad Rizky Kustiadie [a3], I Gusti Ngurah Made Dika Varuna [a4], Made Yayang Eka Prananda [a5]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Unud, Jimbaran, Bali, 80361, Indonesia
[1]aldy.ardita@gmail.com
[2]anom.cp@unud.ac.id
[3]rizkytegal24@gmail.com
[4]ngurahdika22@gmail.com
[5]yayangp32@gmail.com

Abstract

Seiring berkembangnya beragam aplikasi maka sistem Android haruslah tahan terhadap bebagai serangan malware dengan mengamati izin akses yang diberikan oleh pengguna. Penyerang dapat menggunakan kerentanan dalam aplikasi untuk mencuri berbagai informasi penting. Informasi merupakan aset penting dan berharga berupa rekaman suara, rekaman video, catatan, dll. Oleh karena itu, diperlukan suatu analisis keamanan. dari aplikasi yang digunakan dengan tes / tindakan pada tingkat keamanan aplikasi. Dalam melakukan Vulnerability test atau proses identifikasi celah keamanan pada aplikasi android dilakukan dua teknik yaitu dengan MobSF dan dengan frida. Hasil dari Analisis MobSF sangat terlihat perbedaannya antara mendownload aplikasi melalui pihak ketiga dengan mendownload aplikasi melalui Play Store. Dimana nilai hash yang didapat sangat berbeda baik dari md5, sha1, atau sha256, dari hasil tersebut dapat diketahui bahwa ada perubahan pada file yang disediakan oleh penyedia pihak ketiga. Pada security score didapatkan bahwa aplikasi yang di download melalui pihak ketiga terdapat banyak server dan aktivitas mencurigakan, sedangkan aplikasi yang terdapat di playstore terdapat 2 server yang asli. Pada size, ukuran file yang disediakan oleh pihak ketiga, ukuran file asli hanya 20.37MB sedangkan file yang di sediakan oleh pihak ketiga berukuran 61.33MB. Pada analisis menggunakan frida dilakukan proses penyerangan yaitu bypass login. Dimana pada aplikasi pihak ketiga sudah memiliki email yang telah diinputkan oleh penyedia aplikasi.
Dari hasil analisis yang dilakukan maka lebih baik untuk mendownload aplikasi melalui playstore agar lebih aman. Karena sebagai pengguna awam tidak akan tahu perubahan file apa yang dilakukan dan beresiko atau tidaknya perubahan tersebut terhadap perangkat tersebut

Keywords: *Mobile app, Keamanan, Vulnerability*

1. Pendahuluan

Saat berbagai aplikasi dikembangkan, sistem Android harus tahan terhadap berbagai serangan malware dengan memperhatikan izin akses yang diberikan oleh pengguna. Penyerang dapat menggunakan pelanggaran keamanan dalam aplikasi untuk mencuri informasi[1]. salah satu aset penting dan berharga berupa rekaman suara, rekaman video, catatan, dll. Oleh karena itu, perlu adanya analisis keamanan terhadap aplikasi yang digunakan dengan cara menguji/mengukur tingkat keamanan suatu aplikasi.
Selain menjadi sistem operasi ponsel yang paling populer, Android merupakan sistem operasi yang paling rentan. Banyaknya peretas yang memanfaatkan celah dalam sistem dan aplikasi pihak ketiga [2]. Menurut laporan para peneliti dari TheBestVPN menghitung jumlah kerentanan yang ada pada platform Linux, Windows, dan Android. Hasilnya, sistem keamanan yang dimiliki oleh Google yaitu Android menjadi sistem operasi yang menduduki peringkat pertama dalam jumlah kerentanan pada 2019, dengan total 414 kerentanan. Namun, jumlah kerentanan tersebut mengalami penurunan dari tahun ke tahun. Google memiliki keterbukaan pada sistem operasinya sehingga dapat mengetahui

seberapa rentan sistem operasi yang dimilikinya. Sehingga membuat Google terus memperbaiki sistem operasinya yaitu Android agar tidak mudah di retas .
Beberapa cara telah dilakukan oleh peneliti untuk memperbaiki sebuah sistem keamananan yang ada pada sistem operasi Android. Salah satunya yaitu dilakukannya penetration test [3]. Penetration test adalah mensimulasikan sebuah serangan yang dilakukan karena adanya kerentanan dan menganalisis kerentanan tersebut [4].
Dikarenakan kerentanan yang ada pada sistem operasi Android maka, pada penelitian kali ini akan melakukan analisa keamanan aplikasi Android dengan menggunakan penetration test yang akan menguji ketahanan dari sebuah aplikasi Android. Hasil analisa keamanan aplikasi Android dengan penetration test diharapkan dapat memberi kesadaran bagi pengguna aplikasi dan juga memberikan saran kepada pihak pengembang aplikasi untuk selalu meningkatkan keamanan serta dapat menyadarkan pengguna akan risiko keamanan yang ada pada setiap aplikasi, terdapat dua metode analisis kerentanan aplikasi android diantaranya adalah analisis statis dan analisis dinamis [5], dalam penelitian ini penulis menggunakan metode statis dan dinamis untuk melakukan uji, uji kerentanan sistem secara statis akan menggunakan *tools* bawaan kali linux yaitu MobSf, *tolls* ini merupakan Mobile Security Framework (MobSF) adalah tool otomatis scanning yang biasa digunakan untuk analisis malware, dan security assessment framework yang mampu melakukan analisis statis dan dinamis [6], tools ini digunakan karena memiliki tingkat akses yang sangat mudah. Analisis dinamik akan menggunakan *tools* bawaan kali linux yang bernama frida, tools ini digunakan kerana dimungkinkan menggunakan emulator android yang terkoneksi menggunakan *virtual* usb.

2. Metode

Meote yang akan digunakan pada penelitian ini adalah metode kualitatif yang akan berfokus pada analisis keamanan suatu aplikasi mobile, penelitian ini akan dilaksanakan dengan memanfaatkan *tools* bawaan dari sistem oprasi kali linux.

2.1. Analisis kebutuhan

Kebutuhan non-fungsional:

a. Hardware (perangkat keras)
   1. SSD 512 GB
   2. Ram 8 GB
   3. Intel Core i5
   4. Geforce GTX 1650Ti

b. Software (perangkat lunak)
   1. Kali Linux
   2. Mobile Security Framework (MobSF).
   3. Frida
   4. VMWare

Kebutuhan fungsional:

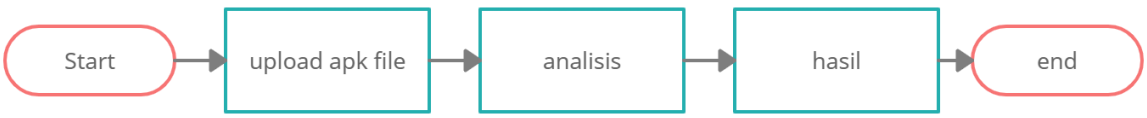
a. Kemampuan ekstrak .apk file
b. Kemampuan analisis celah keamanan

2.2. Rancangan analisis sistem

Dalam penelitian ini, peneliti menerapakan dua metode dalam melakukan *Vulnerability test* atau proses identifikasi celah keamanan pada aplikasi android, yaitu:

a. Analisis statis menggunakan MobSF

   Dalam melakukan analisis statik akan digunakan software Mobile Security Framework (MobSF), tahapan yang dilakukan dalam uji statik dapat dilihat pada flowchart pada gambar 1.



Gambar 1. flowchart pengujian statis

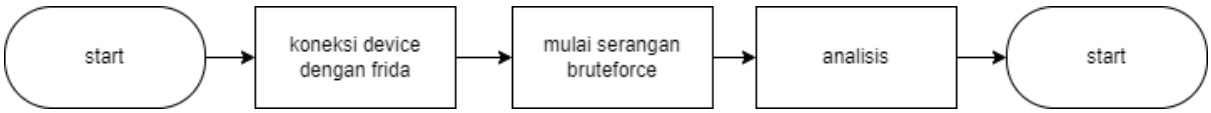
   Tahapan awal adalah melakukan persiapan seperti menentukan aplikasi yang akan di uji, pada penelitian ini akan digunakan aplikasi android yang cukup populer yaitu spotify, pada tahapan

p-ISSN: 2301-5373
e-ISSN: 2654-5101

persiapan penulis mendownload aplikasi spotify dari *play store* dan juga aplikasi spotify yang ada pada link berikut: https://www.goapkmods.com/apps/spotify-premium/. Setelah aplikasi berhasil didapatkan proses akan dilanjutkan dengan proses upload. Setelah proses upload file selesai akan didapatkan beberapa hasil diantaranya adalah informasi mengenai hash sum aplikasi, meta data aplikasi, dan file source code yang dapat digunakan untuk melakukan analisis manual terhadap code.

b. Analisis dinamis menggunakan frida

Frida adalah salah satu tools yang dijalankan menggunakan linux dengan tujuan melakukan analisis dinamik terhadap aplikasi android, berbeda dengan MobSF yang dimana MobSF adalah tools yang digunakan untuk melakukan analisis statik pada sebuah aplikasi android. Langkah - langkah yang dilakukan dalam menggunakan tools ini adalah langkah pertama adalah mengkoneksikan perangkat dengan tools, selanjutnya melakukan beberapa serangan terhadap aplikasi yang telah ditargetkan, terdapat beberapa serangan yang dapat dilakukan salah satunya adalah brute force, flowchar dapat dilihat pada gambar 2.



Gambar 2. flowchart pengujian dinamis

3. Hasil dan pembahasa

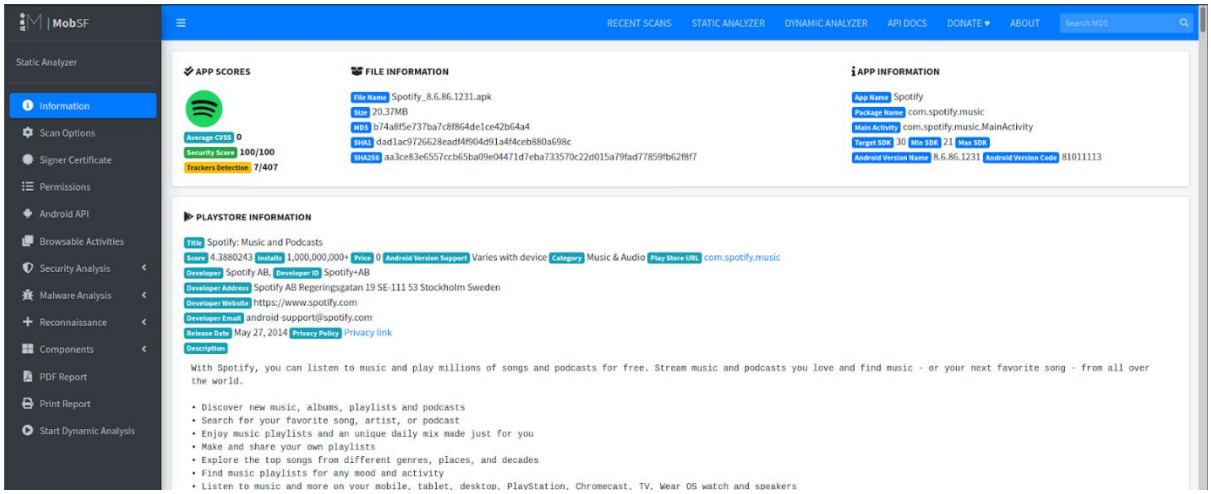
Pada tahapan analisi akan dijelaskan hasil analisis statiks dan dinamis yang telah dilaksanakan.

3.1. Analisis statis menggunakan MobSF

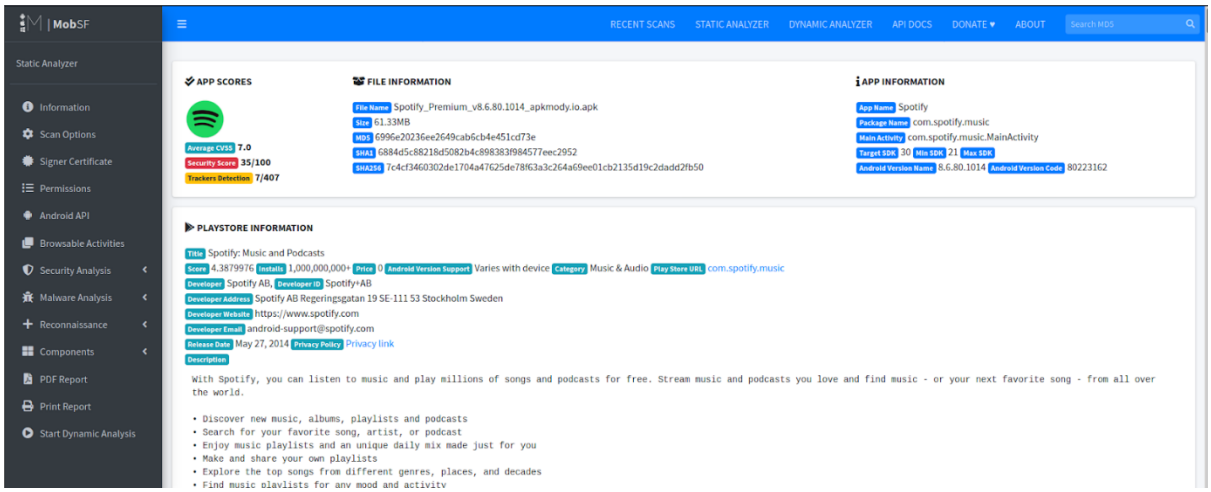
Analisis statis akan dilakukan dengan melakukan upload file apk dan membandingan hasil analisis dari MobSF, terdapat beberapa perbedaan yang mencolok pada hasil kedua aplikasi tersebut perbandingan hasil dapat dilihat pada tabel 1 dan Informasi hasil analisis tersebut dapat dilihat pada gambar 2 dan gambar 3

Table 1. Perbandingan hasil analisis

No	Informasi	Spotify asli	Spotify mod
1	*size*	20.37 MB	Nama: dec12.png
2	*MD5*	b74a8f5e737ba7c8f864de1ce42b64a4	6996e20236ee2649cab6cb4e451vd73e
3	*Security score*	100 / 100	35 / 100
4	*Trackers detection*	7 /407	7 / 407



Gambar 2. Hasil analisis aplikasi original



Gambar 3. Hasil analisis aplikasi pihak ketiga

a. Hasil hash

Hashing merupakan salah satu teknik kriptografi yang sering digunakan dalam melakukan identifikasi perubahan pada file atau suatu pesan, dikarenakan hashing memiliki sifat unik yang akan selalu menghasilkan nilai berbeda walau perubahan pada file atau pesan awal sangat minim.

Pada hasil analisis diatas didapatkan bahwa perbedaan hasil hashing baik itu menggunakan md5, sha1, atau sha256 memunculkan nilai yang sangat berbeda dari hasil tersebut dapat diketahui bahwa ada perubahan pada file yang disediakan oleh penyedia pihak ketiga

b. Security Score

Security score merupakan nilai yang diperoleh dengan melakukan analisis terhadap permission dan aktivitas mencurigakan dari aplikasi itu sendiri, pada sistem yang disediakan oleh pihak ketiga memiliki skor keamanan yang rendah dikarenakan aplikasi pihak ketiga memiliki aktivitas yang mencurigakan saat analisis server yang ada, aplikasi original hanya memiliki dua server yang aktif namun aplikasi yang disediakan oleh pihak ketiga memiliki server 10 server yang aktif dan 1 server bekerja secara anonymous, list server dapat dilihat pada gambar 4 dan 5

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Gambar 5. List server aplikasi pihak ketiga

Gambar 6. List server aplikasi original

c. Size

Selain server dan hasil dari hashing juga terdapat perbedaan yang sangat besar pada ukuran file yang disediakan oleh pihak ketiga, ukuran file asli hanya 20.37MB sedangkan file yang di sediakan oleh pihak ketiga berukuran 61.33MB.

3.2. Analisis dinamis menggunakan Frida

Langkah awal melihat proses yang berjalan pada sistem android

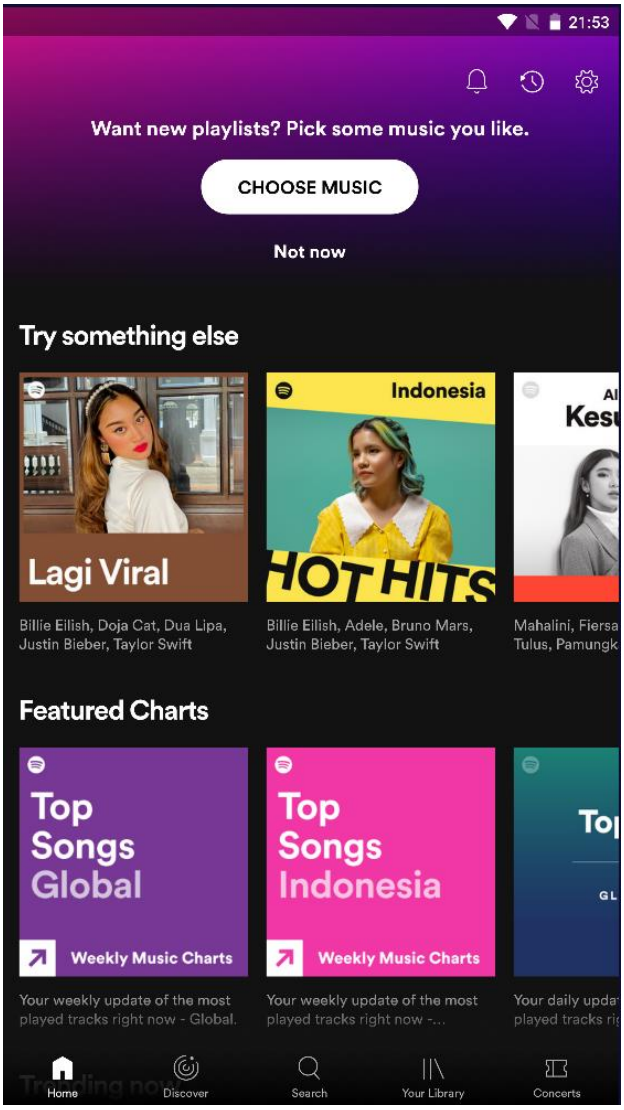
PID	Name	Identifier
672	Android Keyboard (AOSP)	com.android.inputmethod.latin
1056	Android Services Library	android.ext.services
570	Android System	android
1283	Blocked Numbers Storage	com.android.providers.blockednumber
1381	Calendar	com.android.calendar
1418	Calendar Storage	com.android.providers.calendar
570	Call Management	com.android.server.telecom
936	Cell Broadcasts	com.android.cellbroadcastreceiver
1081	Clock	com.android.deskclock
1283	Contacts Storage	com.android.providers.contacts
1403	Email	com.android.email
570	Fused Location	com.android.location.fused
1145	Launcher3	com.android.launcher3
1457	Messaging	com.android.messaging
792	MmsService	com.android.mms.service
792	Phone Services	com.android.phone
792	Phone and Messaging Storage	com.android.providers.telephony
1193	Print Spooler	com.android.printspooler
808	Settings	com.android.settings
570	Settings Storage	com.android.providers.settings
1737	Sieve	com.mwr.example.sieve
687	System UI	com.android.systemui
1283	User Dictionary	com.android.providers.userdictionary
1173	com.android.smspush	com.android.smspush
1117	com.genymotion.genyd.GenydServiceApp	com.genymotion.genyd
1103	com.genymotion.systempatcher.SystemPatcherApp	com.genymotion.systempatcher

Gambar 7. Aplikasi yang berjalan

Langkah selanjutnya adalah melakukan proses penyerangan, serangan yang di uji cobakan pada aplikasi ini adalah uji coba serangan bypass pada proses login. Sistem yang diujikan memiliki

kelemahan terhadap bypass login, pada sistem tidak terjadi begitu banyak perubahan saat aplikasi digunakan tanpa memasukan username dan password hanya sistem masih berjalan normal. Hal ini dikarenakan aplikasi yang disediakan pihak ketiga sudah dimasukan email default oleh penyedia.

Gambar 8. Proses bypass login



Gambar 9. Hasil bypass login

4. Kesimpulan

Dalam melakukan Vulnerability test atau proses identifikasi celah keamanan pada aplikasi android dilakukan dua teknik yaitu dengan MobSF dan dengan frida. Hasil dari Analisis MobSF sangat terlihat perbedaannya antara mendownload aplikasi melalui pihak ketiga dengan mendownload aplikasi melalui Play Store. Dimana nilai hash yang didapat sangat berbeda baik dari md5, sha1, atau sha256, dari hasil tersebut dapat diketahui bahwa ada perubahan pada file yang disediakan oleh penyedia pihak ketiga. Pada security score didapatkan bahwa aplikasi yang di download melalui pihak ketiga terdapat banyak server dan aktivitas mencurigakan, sedangkan aplikasi yang terdapat di playstore terdapat 2 server yang asli. Pada size, ukuran file yang disediakan oleh pihak ketiga, ukuran file asli hanya 20.37MB sedangkan file yang di sediakan oleh pihak ketiga berukuran 61.33MB.

Pada analisis menggunakan frida dilakukan proses penyerangan yaitu bypass login. Dimana pada aplikasi pihak ketiga sudah memiliki email yang telah diinputkan oleh penyedia aplikasi.

Dari hasil analisis yang dilakukan maka lebih baik untuk mendownload aplikasi melalui playstore agar lebih aman. Karena sebagai pengguna awam tidak akan tahu perubahan file apa yang dilakukan dan beresiko atau tidaknya perubahan tersebut terhadap perangkat tersebut.

References

[1] Anwar, Nuril, et al. "Ekstraksi Logis Forensik Mobile pada Aplikasi E-Commerce Android." *Mobile and Forensics* 2.1 (2020): 1-10.

[2] Alviansyah, Fauzan Awanda, and Erika Ramadhani. "Implementasi Dynamic Application Security Testing pada Aplikasi Berbasis Android." *AUTOMATA* 2.1 (2021).

[3] Hanifurohman, Cholis, and Deanna Durbin Hutagalung. "Analisa Keamanan Aplikasi Mobile E-Commerce Berbasis Android Menggunakan Mobile Security Framework." *PROCEEDINGS UNIVERSITAS PAMULANG* 1.1 (2020).

[4] Hanifurohman, Cholis, and Deanna Durbin Hutagalung. "ANALISIS STATIS MENGGUNAKAN MOBILE SECURITY FRAMEWORK UNTUK PENGUJIAN KEAMANAN APLIKASI MOBILE E-COMMERCE BERBASIS ANDROID." *Sebatik* 24.1 (2020): 22-28.

[5] Kartono, Aan, Anang Sularsa, and Setia Juli Irzal Ismail. "Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan Mobsf." *eProceedings of Applied Science* 5.1 (2019).

[6] Rama, Gilang Aditya, Fauziah Fauziah, and Nurhayati Nurhayati. "Perancangan Sistem Keamanan Brangkas Menggunakan Pengenalan Wajah Berbasis Android." *JURNAL MEDIA INFORMATIKA BUDIDARMA* 4.3 (2020): 635-641.

[7] Merina, Calysta. *Analisis perbandingan kinerja test automation framework untuk functional testing pada aplikasi berbasis android dengan metode the distance to the ideal alternative*. BS thesis. Fakultas Sains Dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.

[8] Yumnun, Luqman Hakim, Ari Kusyanti, and Dany Primanita Kartikasari. "Implementasi OWASP Mobile Security Testing Guide (MSTG) Untuk Pengujian Keamanan Pada Aplikasi Berbasis Android." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN* 2548 (2020): 964X.

This page is intentionally left blank.