ANALISIS STATIS MENGGUNAKAN *MOBILE SECURITY* FRAMEWORK UNTUK PENGUJIAN KEAMANAN APLIKASI *MOBILE E-COMMERCE* BERBASIS ANDROID

Cholis Hanifurohman[1)] dan **Deanna Durbin Hutagalung[2)]**

[1,2]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Pamulang
[1,2] Jl. Surya Kencana No. 1 Pamulang, Tangerang Selatan, 15417
E-mail : dosen01825@unpam.ac.id[1)], dosen01677@unpam.ac.id [2)]

ABSTRAK

Pengguna internet di Indonesia setiap tahunnya mengalami peningkatan yang terus naik. Peningkatan yang pesat ini diiukuti juga dengan penggunaan internat menggunakan perangkat *mobile*. Hal ini memberikan dampak positif ke beberapa sektor bisnis seperti jual beli *online* dan juga memicu munculnya beragam aplikasi *mobile* khususnya pada *platform android*. Oleh karena itu perlu dilakukan analisis keamanan terhadap aplikasi dengan melakukan pengujian/pengukuran terhadap tingkat keamanan aplikasi. Tujuan dari penelitian ini adalah untuk meningkatkan pemahaman kepada pengguna aplikasi *mobile e-commerce* terhadap celah-celah keamanan aplikasi *mobile e-commerce* dan memberikan metode dalam melakukan analisis statis menggunakan *Mobile Security* Framework (MobSF) untuk melakukan pengujian keamanan terhadap aplikasi *mobile e-commerce* khususnya yang berbasis android. Analisis statis dilakukan dengan melakukan anailis terhadap kelemahan kriptografi (*weak crypto*), SSL *bypass*, penggunaan *dangerous permission*, *hardcode secret*, *root detection* dan *domain malware check*. Metode yang digunakan dalam melakukan anailis adalah *Mobile Security* Framework (MobSF). Sistem ini mempunyai tiga fase, yaitu kebutuhan perencanaan, proses desain RAD dan fase implementasi. Hasil analisis keamanan keamanan yang dilakukan pada aplikasi *mobile e-commerce* yaitu SP, TP, LZ, BL dan SR yang merupakan lima besar *mobile e-commerce* berbasis android paling populer di Indonesia menunjukkan bahwa beberapa celah keamanan masih terdapat dari di kelima aplikasi hasil tersebut yang perlu diketahui baik oleh pengguna maupun pengembang aplikasi.

Kata Kunci: *Security, Smartphone, Android, E-commerce, MobSF*

1. PENDAHULUAN

Perkembangan internet di Indonesia mengalami perkembangan yang cukup pesat. Perkembangan ini diikuti juga dengan penggunaan *smartphone* yang memberikan dampak positif ke sektor perdagangan elektronik atau *e-commerce*, sehingga sektor ini masih menjadi primadona para investor di 2018. Hasil penelitian yang dilakukan oleh Google Search pada tahun 2018 menobatkan Indonesia menjadi pasar *e-commerce* terbesar dan paling cepat berkembang di Asia Tenggara. Nilai ekonomi internet di Indonesia pada 2018 diperkirakan mencapai US$27 miliar. Dengan rata-rata pertumbuhan majemuk belanja konsumen pertahun naik 49 persen (Google, 2019) (search, 2018).

Perubahan pola perilaku belanja ini juga ditunjukkan dengan volume transaksi *e-commerce* yang meningkat. Laporan tahunan yang dikeluarkan *We Are Social* menunjukkan, prosentase masyarakat Indonesia yang membeli barang dan jasa secara *online* dalam kurun waktu sebulan di 2017 mencapai 41% dari total populasi, meningkat 15% dibanding tahun 2016 yang hanya 26% (Inc, 2019). Pesatnya pertumbuhan *e-commerce* ini juga mendorong aplikasi-aplikasi *mobile e-commerce* berkembang sejalan dengan kebutuhan masyarakat akan layanan berbasis aplikasi *mobile* yang berjalan pada *smartphone*. Kementerian Komunikasi dan Informatika memprediksi pada 2018 jumlah pengguna *smartphone* aktif di Indonesia mencapai 100 juta orang. Dengan jumlah tersebut, Indonesia akan menjadi negara pengguna *smartphone* terbesar keempat di dunia setelah Tiongkok, India, dan Amerika. Meskipun transaksi *e-commerce* pada 2016 masih didominasi PC, tapi pada 2017 kondisi akan berubah. Jumlah transaksi melalui *smartphone* diprediksi mengalahkan transaksi dari PC.

Untuk itu, dengan tren perilaku konsumen saat ini, pelaku *e-commerce* harus bisa memaksimalkan aplikasi *mobile* yang dimiliki. Proses tersebut dapat dimulai dengan optimalisasi UI dan UX, termasuk promo eksklusif bagi pengguna aplikasi.

E-commerce telah membuat hidup lebih mudah bagi banyak orang di seluruh dunia sehingga melakukan transaksi harian yang dilakukan secara nirkabel dengan nyaman tetapi juga menimbulkan beberapa ancaman keamanan (Ogundiya, 2014).

Celah-celah keamanan yang terdapat di aplikasi dapat digunakan penyerang untuk mencuri informasi penting di dalam *smartphone*, dimana informasi merupakan salah

suatu aset penting dan sangat berharga disajikan dalam berbagai format berupa : catatan, lisan, elektronik, pos, dan audio visual.

Dalam beberapa penelitian yang sudah dilakukan, belum terdapat penelitian yang melakukan analisis terhadap celah-celah keamanan yang terdapat dalam aplikasi *e-commerce* yang banyak digunakan di Indonesia.

Oleh karena itu perlu dilakukan analisa keamanan terhadap aplikasi dengan melakukan pengujian/pengukuran terhadap tingkat keamanan aplikasi. *Mobile security framework* (MobSF) adalah salah satu metode yang dapat digunakan untuk melakukan pengukuran terhadap keamanan aplikasi. Hasil analisa keamanan menggunakan MobSF diharapkan dapat memberikan kesadaran terhadap pengguna aplikasi dan memberikan masukan kepada pihak pengembang aplikasi untuk terus meningkatkan aspek keamanan dan dari sisi pengguna aplikasi menyadari akan risiko keamanan terhadap setiap aplikasi *mobile e-commerce* yang mereka gunakan.

2. RUANG LINGKUP

Dalam penelitian ini permasalahan mencakup:

1. Bagaimana melakukan analisis statis menggunakan *Mobile Security* Framwork (MobSF)?
2. Apakah terdapat celah-celah keamanan pada aplikasi *mobile e-commerce* ?
3. Laporan tingkat keamanan dari aplikasi *mobile e-commerce*.

3. BAHAN DAN METODE

Berikut adalah kajian teori dan metologi dari penelitian ini.

3.1 *E-commerce*

Bahan *Mobile e-commerce*, didefinisikan sebagai penyampaian *e-commerce* secara langsung ke tangan pelanggan, di mana pun melalui teknologi nirkabel yang pada awalnya diciptakan oleh Kevin Duffey pada tahun 1997. Perdagangan *mobile* adhoc berlangsung antara beberapa *node* yang dekat satu sama lain tanpa mengandalkan pada layanan infrastruktur apa pun (Alexander-Brown, 2013).

Dalam seluruh proses transaksi *mobile e-commerce* sistem transaksi, ada tiga faktor utama yang tidak aman yang berasal dari *mobile terminals*, *mobile radio interface* dan *network-side* (Ali Mirarab, 2014).

1. Faktor-faktor tidak aman *mobile terminals* terutama dimanifestasikan dalam identitas pengguna, informasi akun, dan kunci otentikasi dan sebagainya. Misalnya, orang lain yang mendapatkan *mobile terminals* pengguna cenderung memalsukan identitas pengguna untuk melakukan beberapa kegiatan ilegal.
2. Faktor-faktor tidak *mobile radio interface*
   Sebagai komunikasi antara terminal seluler dan tetap jaringan dalam transmisi nirkabel bergantung pada antarmuka nirkabel terbuka untuk mengirimkan, setiap orang yang memiliki perangkat nirkabel yang sesuai akan memiliki kesempatan untuk mendapatkan informasi melalui penyadapannya melalui saluran nirkabel, dan bahkan dapat memodifikasi, menghapus atau mengirim kembali informasi, yang menimbulkan ancaman terhadap aktivitas perdagangan.
3. Faktor-faktor jaringan yang tidak aman
   Jaringan terutama mengacu pada jaringan nirkabel, *gateway* dan jalur kabel. Jika informasi tidak dilindungi ketika dikirim dalam jaringan nirkabel, jaringan kabel dan dikonversi oleh *gateway*, kemungkinan akan terekspos menyebabkan ancaman terhadap kegiatan perdagangan.

3.1.1 Aplikasi *Mobile* E-commrec di Indonesia

Peta penggunaan *e-commerce* secara global maupun di Indonesia dapat diketahui melalui laporan berkala yang di keluarkan oleh iPrice (iPrice, 2019).

iPrice Group adalah situs meta-search yang beroperasi di Indonesia dan enam negara lain di Asia Tenggara, yakni; Malaysia, Singapura, Filipina, Thailand, Vietnam dan Hong Kong. iPrice bermitra dengan sejumlah brand terbesar di kawasan ini, seperti Tokopedia, Bukalapak, Lazada, Shopee, Zalora, Gojek, Traveloka, Klook dan banyak lagi. Secara berkala, iPrice Group juga merilis laporan mendalam mengenai *e-commerce*, startup dan topik terkait lainnya (iprice,2019).

Adapun 5 besar aplikasi *mobile e-commerce* yang berbasis android pada semester kedua tahun 2019 (ditampilkan dalam bentuk singkatan nama) seperti pada Tabel 1.

Tabel 1 Aplikasi *Mobile E-commerce*

Peringkat	Aplikasi	Versi
1	SP	v2.41.06
2	TP	v.3.35
3	LZ	v.6.32.0
4	BL	v.4.42.5
5	SR	v.0.3.8

3.2 Ancaman Pada Kemananan Android

Mekanisme berbasis *permission*/izin disediakan untuk keamanan aplikasi android yang mengatur akses aplikasi android pihak ketiga ke sumber daya penting pada perangkat. Mekanisme ini sangat dikritik karena kontrolnya yang kasar terhadap izin aplikasi dan manajemen izin yang tidak efisien, oleh pengembang, dan pengguna. Misalnya, pengguna diizinkan untuk menerima semua permintaan izin dari aplikasi untuk menginstalnya atau menolak instalasi aplikasi. Bagian ini menjelaskan masalah keamanan utama android yang menyebabkan kebocoran informasi pengguna dan menyebabkan hilangnya privasi pengguna (Bansode, 2017).

Setidaknya ada 4 ancaman pada keamanan android :

1. Kebocoran Data
   Aplikasi android yang bocor dapat menempatkan informasi yang sensitif bagi pengguna di lokasi yang

tidak aman di perangkat atau dapat mengirim informasi identifikasi perangkat, misalnya metadata aplikasi seperti detail jaringan. Lokasi perangkat yang tidak aman ini dapat diakses oleh aplikasi jahat lainnya pada perangkat yang sama. Data atau informasi sensitif yang bocor menyebabkan perangkat menjadi kondisi kritis. Eksploitasi kerentanan ini sangat mudah karena penyerang dapat memperoleh akses ke bagian perangkat tempat data sensitif disimpan. Dampak dari kebocoran data perangkat Android sangat parah. Sesuai dengan situs web berita grup peneliti keamanan, 58% perangkat Android memiliki kebocoran privasi dan sekitar 3% memiliki kebocoran PII (*personally identifiable information*).

2. Eskalasi hak istimewa
   Kekurangan keamanan mekanisme izin android dapat menyebabkan peningkatan eskalasi hak istimewa yang disebabkan oleh aplikasi yang dikompromikan. Para penulis menggambarkan peningkatan hak istimewa sebagai: Aplikasi dengan izin yang lebih sedikit (penelepon yang tidak memiliki hak istimewa) tidak dibatasi untuk mengakses komponen aplikasi yang lebih istimewa (hak istimewa). Contoh eskalasi hak istimewa dapat diberikan sebagai - kejahatan lokal dapat mengeksekusi kode arbitrer di kernel tanpa memiliki hak istimewa untuk melakukannya. Hal ini dapat menyebabkan kompromi total pada sistem operasi yang menyebabkan korupsi pada sistem operasi dan menyelesaikan perbaikan perangkat. Sesuai dengan database *Common vulnerabilities and exposures* (CVE), kerentanan eskalasi hak istimewa yang kritis ditemukan di android versi 6 dan di atasnya. Pelanggaran hak istimewa di android membuat jutaan pengguna berisiko dibajak *smartphone*-nya.
3. Pengemasan ulang aplikasi
   Proses pembongkaran atau dekompilasi *file* .apk menggunakan teknik rekayasa balik (*reverse engineering*) dan menambahkan/menyusupkan kode berbahaya ke dalam kode sumber utama dikenal sebagai pengemasan ulang aplikasi android. Untuk pengguna android, menjadi sulit untuk membedakan antara aplikasi jahat yang dipaket ulang dan aplikasi normal karena aplikasi yang dikemas ulang biasanya berfungsi dengan cara yang sama dengan yang sah.
4. Serangan DDos
   Dalam serangan DDos, penyerang berusaha membuat perangkat atau sumber daya tidak tersedia untuk penggunaan yang dimaksudkan dengan mengganggu layanan perangkat host untuk sementara atau tidak terbatas. Sesuai dengan Laporan dari *Symantec Internet Security*, sekitar 7,2% aplikasi android mengalami serangan DDos (Bansode, 2017).

3.3 MobSF

Mobile Security Framework (MobSF) adalah framework yang digunakan untuk pengujian penetreasi terhadap aplikasi seluler (Android / iOS / Windows) otomatis yang mampu melakukan analisis statis, dinamis, dan *malware*. MobSF dapat digunakan untuk analisis keamanan yang efektif dan cepat dari aplikasi seluler Android, iOS dan Windows dan mendukung kedua binari (APK, IPA & APPX) dan kode sumber zip. MobSF dapat melakukan pengujian aplikasi dinamis saat runtime untuk aplikasi Android dan memiliki kemampuan fuzzing API Web yang didukung oleh CapFuzz, pemindai keamanan khusus Web API. MobSF dirancang untuk membuat integrasi CI / CD atau DevSecOps secara mulus (Abraham, 2019).

3.3.1 Analisis Statis

Analisis statis adalah salah satu tahapan pengujian aplikasi seluler. Menurut pentester Hacken (Hacken, 2019), kerangka kerja open source yang paling nyaman adalah MobSF. Secara umum analisis statis menggunakan MobS pada aplikasi *mobile e-commerce* mencakup:

1. SSL *bypass*
2. *Weak Crypto*
3. *Permissions*
4. *Hardcode secrets*
5. *Malware check*

3.4 Metode *Rapid Application Development* (RAD)

Metode yang digunakan dalam membangun aplikasi ini dengan metode pengembangan sistem *Rapid Application Development* (RAD). Sistem ini mempunyai tiga fase, yaitu kebutuhan perencanaan, proses desain RAD dan fase implementasi. (Kendal,2002).

3.4.1 Fase Metode Kebutuhan Perencanaan (*Requirement Planning Phase*)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasikan tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasikan syarat-syarat informasi yang ditimbulkan dari tujuan- tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan.

3.4.2 Proses Desain RAD (*Rapid Aplication Development*)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diuji coba dan kemudian diperkenalkan kepada organisasi.

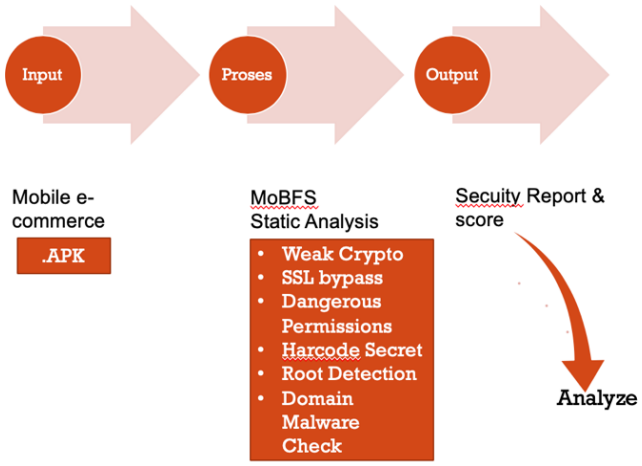
3.4.2 Fase Implementasi (*Implementation Phase*)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna.

4. PEMBAHASAN

Sesuai dengan metode yang dilakukan, terdapat kebutuhan baik *sofware* maupun *hardware* untuk melakukan pengujian. Software yang dibutuhkan adalah Git, Python 3.6+, JDK 8, Virtual Box, dan mAc SDK Header. Adapun kebutuhan *hardwarenya* adalah perangkat android yang sudah *root*.

Dalam penelitian ini peneliti memanfaatkan salah satu aplikasi *opensource* yang biasa digunakan untuk melakukan *pentesting* terhadap aplikasi *mobile*. Oleh karena itu dalam langkah ini dilakukan dengan melakukan instalasi sesuai dengan dokumentasi yang sudah terdapat dalam paket *source code* aplikasi MobSF seperti pada Gambar 1.



Gambar 1. Desain Analisis Statis menggunakan MobSF

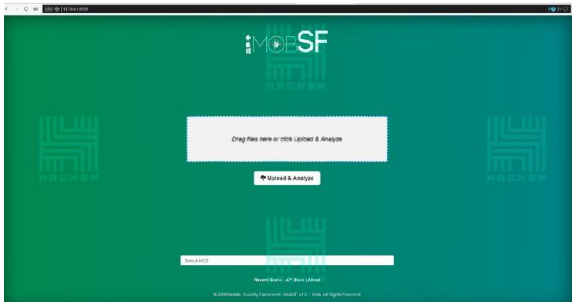
4.1. Tahapan Proses Analisis Statis

Tahapan proses analisis statis menggunakan MobSF setelah berhasil melakukan instalasi MobSF, jalankan script berikut pada direktori MobSF : *$.run.sh*.

Kemudian pada browser diakses melalui alamat http://127.0.0.1:8000 untuk mendapatkan beberapa fitur seperti :

1. *file upload*
2. *view previous scan reports*
3. *transition to* API documentation
4. *transition to* GitHub project

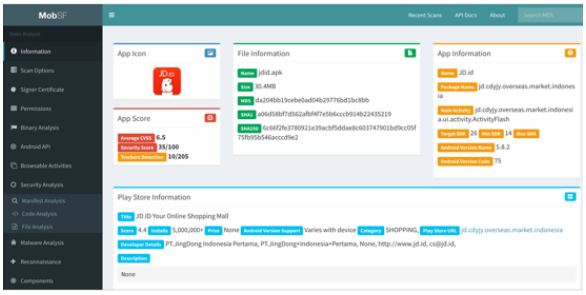
Pada saat MobSF dijalankan pertama kali akan muncul halaman serti pada Gambar 2. Halaman Home pada MobSF.



Gambar 2. Halaman Home pada MobSF

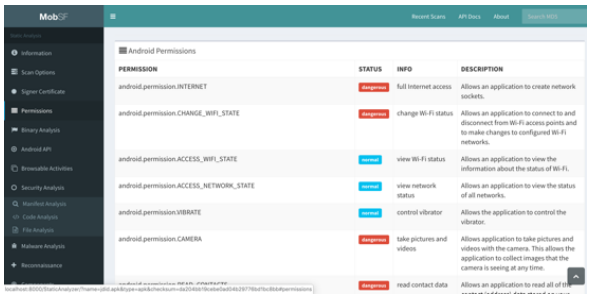
Setelah *file* diunduh dan dianalisis, halaman dengan hasil analisis muncul. Ada menu di sebelah kiri yang memungkinkan untuk menavigasi dengan cepat di seluruh halaman (hasilnya cukup banyak). Berikut ini informasi bermanfaat dalam tangkapan layar seperti pada Gambar 3.

1. *Application hash sum.*
2. *Supported Android OS Versions.*
3. *the number and type of components (exported or not): it's important, as exported components can lead to critical vulnerabilities.*
4. *The ability to view and download java- and smali-files that can be analyzed either by other tools or manually.*
5. *View manifest file for analysis.*



Gambar 3. Halaman Hasil Analisis pada MobSF

MobSF dapat melihat deskripsi analisis izin, yang terdapat dalam *file* AndroidManifest.xml. MobSF menganalisis izin aplikasi Android, menentukan statusnya terkait kekritisan, dan deskripsi izin. Di sini perlu memahami arsitektur OS Android untuk menilai tingkat kekritisannya yang sebenarnya. Dari informasi izin ini akan dapat terlihat seberapa banyak izin yang diminta aplikasi dan seberapa banyak yang dalam kategori *dangerous permissions* (Oglaza, 2017). Contoh hasil analisis yang ditampilkan oleh MobSF seperti pada Gambar 3.



Gambar 4. Halaman Hasil Analisis izin pada MobSF

Selain analisis izin, pada bagian *security analysis* akan ditampilkan juga hasil dari analisis source code dan analisis *file* yang merupakan aset dari aplikasi serta analisis *malware* dari domain-domain yang terdapat dalam aplikasi tersebut.

Dari hasil analisis statis aplikasi dan kode sumber memberikan pemahaman dasar tentang arsitektur aplikasi android dan beberapa serangan yang potensial. Analisis statis digunakan sebagai awal dari setiap pentesting aplikasi.

Untuk lebih lengkapnya analisis statis dilanjutkan dengan analisis dinamis dimana MobSF juga menyediakan analisis dinamis tersebut.

4.2. Hasil Analisis Statis

Analisis yang dilakukan untuk analisis keamanan aplikasi *mobile e-commerce* adalah analisis statis dengan melihat seluruh aspek analisis statis dalam MobSF. android Berdasarkan hasil analsis statis yang dilakukan didapatkan hasil seperti yang ditampilkan dalam Tabel 2.

Tabel 2 Hasil Analisis Statis

App	Weak Crypto	SSL Bypass	Dangerous Permissions	Hardcode Secret	Root Detection	Domain Malware Check	*Security* score
SP	NO	YES	YES	NO	YES	GOOD	36
TP	NO	YES	YES	YES	YES	GOOD	40
LZ	YES	YES	YES	YES	YES	GOOD	37
BL	YES	YES	YES	NO	YES	GOOD	38
SR	YES	YES	YES	NO	YES	GOOD	35

4.2.1 *Weak Crypto*

Analisis *weak crypto* dilakukan dengan melihat apakah terdapat implementasi algoritma kriptografi yang lemah atau penggunaan algoritma kriptografi yang sudah usang atau sudah dianggap tidak layak. Dari ke-5 aplikasi yang dilakukan analisis, terdapat 2 aplikasi yaitu LZ dan BL yang masih menggunakan algoritma SHA-1 sebagai algoritma yang digunakan dalam sertifikatnya digitalnya untuk *code signing*. Hasil penelitian menunjukkan bahwa penggunaan SHA-1 untuk sertifikat atau untuk otentikasi *handshake* di TLS, SSH atau IKE berbahaya, karena sudah terbukti dapat dilakukan *collision attack* dan bisa disalahgunakan oleh penyerang. SHA-1 telah rusak sejak 2004, tetapi masih digunakan di banyak sistem keamanan; sehingga disarankan kepada pengguna untuk menghapus dukungan SHA-1 untuk menghindari serangan (Peyrin, 2019). Sedangkan SR terdeteksi menggunakan *Random number* yang tidak aman.

4.2.2. SSL *bypass*

Certificate Pinning adalah lapisan keamanan tambahan untuk melakukan perlindungan terhadap serangan *man-in-the-middle*. Dengan ini memastikan bahwa hanya Otoritas Sertifikat (CA) bersertifikat yang dapat menandatangani sertifikat untuk domain aplikasi, dan bukan CA mana pun di *certificate store*.

Pengembang aplikasi menerapkan *Certificate Pinning* untuk menghindari rekayasa terbalik, memungkinkan pengembang menentukan sertifikat mana yang diizinkan oleh aplikasi. Alih-alih mengandalkan toko sertifikat

Analisis SSL *bypass* dilakukan dengan melakukan *check* apakah terdapat *service* yang melibatkan protokol http yang tidak mewajibkan penggunaan SSL sebagai persyaratan keamanan transaksi dalam protokol http menggunakan SSL seperti mengizinkan http di *manifest*, atau terdapat *string* berkonten http:// yang merupakan *weak implementation*. Ketika mengimplementasikan ke sebuah aplikasi seharusnya menggunakan komunikasi https baik digunakan secara *native* (contoh : *web service*), atau dalam yang digunakan dalam *webview* yang mengakes sebuah webpage melelui *webview* atau mengimplementasikan keduanya. Hasil analisis terhadap SSL *bypass* menunjukkan kelima aplikasi terdapat implementasi yang memungkinkan terjadinya SSL *bypass*.

4.2.3. *Dangerous permissions*

Secara umum istilah izin berarti mengizinkan seseorang untuk melakukan hal tertentu dengan persetujuan atau otorisasi yang diberikan untuk melakukan segala jenis tindakan. Di Android, izin atau *permission* juga mengikuti konsep secara umum dari izin. Aplikasi Android dibangun untuk melakukan serangkaian tindakan, beberapa di antaranya memerlukan izin dari pengguna.

Analisis terhadap *permission* lebih diarahkan pada seberapa banyak *dangerous permissions* yang digunakan oleh aplikasi. Semua aplikasi yang dilakukan analisis menunjukkan bahwa semuanya mengandung *dangerous permissions*. Hal ini menunjukkan bahwa aplikasi

tersebut memungkinkan mengambil data-data pribadi dari perangkat jika pengguna memberikan hak izin.

4.2.4. *Hardcode secret*

Salah satu temuan yang lebih umum yang dilaporkan terkait celah keamanan aplikasi Android adalah masalah yang melibatkan *harcode secret* atau sesyau yang bersifat rahasia (biasanya berupa *password* atau *key*) yang tersimpan pada source Android.

Analisis *hardcode secret* dilakukan dengan melakukan pengecekan apakah terdapat *credential, password, key* atau informasi rahasia lainnya yang tersimpan secara *hardcode* di dalam aplikasi. Dari kelima aplikasi terdapat 2 aplikasi yang kemungkinan terdapat *hardcode secret* yaitu TP dan LZ yang ditemukan dalam bentuk file. *File* yang ditemukan diduga menyimpan data.

4.2.5. *Root Detection*

Root Access adalah proses yang memungkinkan pengguna *smartphone*, tablet, dan perangkat lain yang menjalankan sistem operasi Android untuk mendapatkan kontrol istimewa (dikenal sebagai akses root). "*Rooting*" adalah proses di mana seseorang mendapatkan akses ke perintah admin dan fungsi sistem operasi. Ini memberi kemampuan (atau izin) untuk mengubah atau mengganti aplikasi sistem, file, dan pengaturan, menghapus aplikasi yang sudah diinstal, menjalankan aplikasi khusus yang memerlukan izin tingkat administrator.

Analisis *root detection* dilakukan dengan melakukan check apakah aplikasi mempunyai fungsi untuk melakukan deteksi akses *root* terhadap perangkat android yang digunakan. Dimana akses memungkinkan untuk akses langsung ke dalam sistem termasuk data-data yang dimiliki aplikasi. Hasil analisis *root detection* menunjukkan kelima aplikasi sudah menyediakan mekanisme *root detection* di dalamnya, sehingga aplikasi tersebut tidak akan berjalan pada perangkat yang sudah terdapat akses root (*rooted device*) atau minimal memberikan informasi kepada pengguna bahwa perangkat yang digunakan dalam keadaan akses *root* dan berpotensi membahayakan.

4.2.6. *Domain Malware Check*

Analisis *domain malware check* dilakukan dengan melakukan *check* apakah domain-domain yang terdapat dalam aplikasi terindikasi dalam kategori domain yang mengandung *malware* atau tidak. Hasil analisis menunjukkan bahwa kelima aplikasi tidak menunjukan bahwa terdapat domain yang terindikasi malware.

5. KESIMPULAN

Analisis yang dilakukan terhadap aplikasi android *mobile e-commerce* dengan versi aplikasi pada saat dilakukan analisis, sehingga memungkinkan hasilnya berbeda jika dilakukan analisis dengan waktu yang berbeda seiring dengan update yang secara rutin biasa dilakukan oleh pengembang aplikasi.

Hasil analisis keamanan yang dilakukan pada aplikasi *mobile e-commerce* yaitu *SP, TP, LZ, BL* dan *SR* yang merupakan lima besar *mobile e-commerce* berbasis android paling populer di Indonesia menunjukkan bahwa beberapa celah keamanan masih terdapat dari di kelima aplikasi tersebut dengan tingkat kemanan yang relatif sama Celah-celah kemanan yang ditemukan dapat menjadi *security awareness* untuk pengguna aplikasi tersebut yang jumlahlah cukup banyak di Indonesia. Untuk pengembang aplikasi android khususnya untuk *e-commerce* hasil analisis ini dapat dijadikan catatan keamanan yang harus diperbaiki.

6. SARAN

Analisis yang dilakukan hanya menggunakan analisis statis yang merupakan tahapan awal dalam analisis keamanan sehingga perlu dilanjutkan menggunakan analisis dinamis untuk mendapatkan hasil yang lebih lengkap dari analisis keamanan yang dilakukan dan perlu dibandingkan dengan metode analisis lainnya seperti OWASP *Mobile Application Security Verification Standard* untuk pengujian komprehensif yang mencakup proses, teknik, dan alat yang digunakan untuk keamanan aplikasi seluler.

7. DAFTAR PUSTAKA

Abraham, A. 2019. *Mobile Security Framework MobSF*. Available at: https://github.com/MobSF/*Mobile-Security*-Framework-MobSF [Accessed 14 October 2019].

Alexander-Brown, K. M. &. S., 2013. *Android Security Cookbok.* s.l.:Pack Publishing.

Mirarab, A. 2014. Study of secure m-commerce, challenges and solutions. *ACSIJ Advances in Computer Science: an International Journal,* 3(2).

Bansode, S.M. & Granthi, P. K 2017. Android *Security*: A Survey of *Security* Issues And Defenses. *International Research Journal of Engineering and Technology (IRJET),* 4(07).

Hacken. 2019. *Hacken.* Available at: https://hacken.io/research/industry-news-and-insights/static-*analysis*-of-android-*mobile*-applications-mobsf-manual/ [Accessed 14 October 2019].

Inc, W. A. S., 2019. *We are social.* Available at: ttps://wearesocial.com/special-reports/digital-southeast-asia-2017 [Accessed 14 October 2019].

iPrice, 2019. *iPrice.* Available at: https://iprice.co.id/insights/mapofecommerce/ [Accessed 14 October 2019].

Google, 2019. *Year in search.* Available at: ttps://www.thinkwithgoogle.com/_qs/documents/6818/Year_in_Search__Insights_for_Brands_2018_Indonesia.pdf. [Accessed 14 October 2019].

Oglaza, A. L. R. Z. P. e. a., 2017. A new approach for managing Android permissions: learning users' preferences. *EURASIP J. on Info. Security,* Volume 13.

Ogundiya, U. J. W. &. A. O., 2014. *Mobile* Commerce And *Security* Issues. *International Journal of Scientific Research Engineering & Technology (IJSRET)*, 3(4).

Peyrin, G. L. a. T., 2019. *From Collisions to Chosen-Prefix Collisions Application to Full SHA-1*. Singapore, s.n.