AUDIT KEAMANAN APLIKASI MOBILE BANKING BERBASIS ANDROID DENGAN METODE ANALISIS STATIS

Taqrim Ibadi[1], Yesi Novaria Kunang[2]
Universitas Bina Darma
Jl. Jend. A. Yani No. 3 Palembang
taqrimibadi@binadarma.ac.id

Universitas Bina Darma
Jl. Jend. A. Yani No.3 Palembang
yesinovariakunang@binadarma.ac.id

ABSTRAK

Saat ini perangkat mobile seperti smartphone dan tablet digunakan untuk berbagai kepentingan seperti untuk pembelanjaan, media sosial bahkan beberapa aktifitas beresiko termasuk di dalamnya melakukan aktifitas perbankan. Hampir seluruh sistem perbankan menyediakan fasilitas mobile banking untuk memudahkan customer melakukan transaksi. Untuk itu pada penelitian ini bertujuan melakukan audit keamanan beberapa aplikasi perbankan yang ada di Indonesia khususnya yang berbasis Android, mengingat Android sebagai Sistem Operasi mobile yang paling banyak penggunanya. Untuk melakukan audit keamanan pada penelitian ini mengacu pada metode analisis statis. Dimana pada metode ini pengujian berlandaskan pada analisa source code program. Hasil penelitian ini akan memperlihatkan apakah aplikasi-aplikasi mobile Banking tersebut memenuhi standar keamanan yang sudah ditentukan dan juga akan melihat kelemahan yang ada pada aplikasi-aplikasi tersebut. Laporan yang dihasilkan bisa menjadi masukan untuk perbaikan dan pengembangan aplikasi.

Kata Kunci: Audit Keamanan, Aplikasi Android, Mobile Banking

A. PENDAHULUAN

Berdasarkan studi September 2015 yang dilakukan oleh KPMG, salah satu perusahaan jasa professional terbesar di dunia, bahwa penggunaan mobile banking potensinya akan meningkat menjadi 1,8 milyar pengguna pada tahun 2019, dan regional Asia Tenggara akan menjadi pelopor utama untuk tren ini. Salahsatu masalah yang berdampak untuk kemajuan teknologi mobile banking ini terutama dalam hal keamanan dan kenyamanan (Goerdeler: 2015).

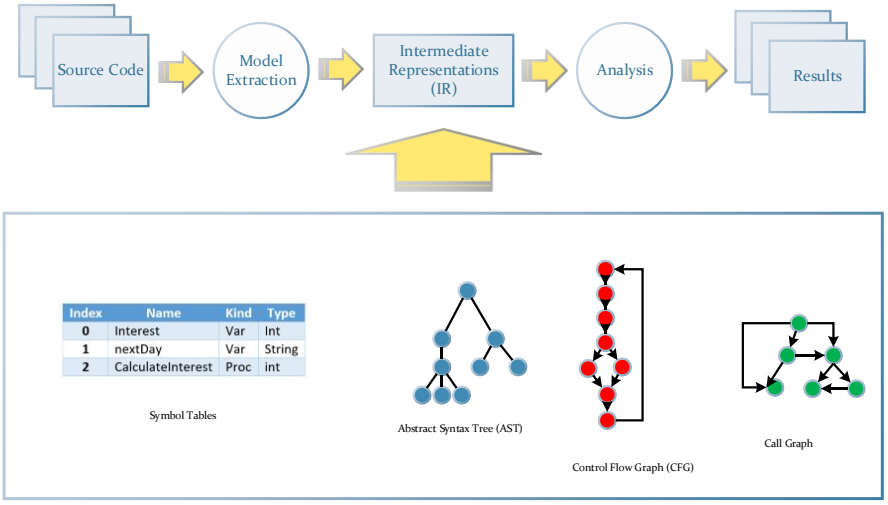
Berdasarkan informasi yang ditulis oleh (Nallayam: 2016) pada databreachtoday.asia tahun 2016, sekitar 85% aplikasi mobile banking dikawasan Asia Pasifik yang tersedia saat ini gagal uji keamanan tingkat dasar, menurut sebuah studi baru-baru ini oleh AppKnox perusahaan keamanan aplikasi berbasis awan. Setengah dari aplikasi yang dipelajari memiliki setidaknya empat celah keamanan, klaim AppKnox. Perusahaannya mengatakan bahwa pihaknya menempatkan 106 aplikasi mobile banking untuk diuji di 14 skenario ancaman yang berbeda dan menemukan bahwa 85% aplikasi mobile banking rentan terhadap celah keamanan tinggi, menengah dan rendah. Studi tersebut juga menemukan bahwa lebih dari 74% aplikasi didiagnosis dengan 5 ancaman teratas pada daftar periksa tim peneliti seperti trust manager yang rusak untuk SSL, izin yang tidak terpakai, eksekusi kode jarak jauh, perlindungan lapisan transport yang tidak memadai dan penggunaan cryptokeys yang membuat mereka rentan terhadap Serangan.

Penelitian ini bertujuan diantaranya: Melakukan auditing untuk melacak semua kejadian, kesalahan dan akses resource yang membahayakan keamanan data pengguna aplikasi mobile banking serta emberikan rekomendasi kelemahan-kelemahan dari aplikasi mobile banking yang diaudit. Berbeda dengan Penelitian yang dilakukan oleh (Su, dkk: 2016), mereka fokus mendeteksi Malware pada Android dan tidak fokus pada kelemahan dari sisi akses resource. Dengan kata lain, penelitina ini pelengkap dari penelitian-penelitian sebelumnya.

B. METODE PENELITIAN

Penelitian ini menggunakan metode pengujian secara statis. Dimana pada pengujian ini, program tidak dijalankan pada platform android melainkan pada program tertentu untuk menganalisa kode program. Hal ini biasanya dilakukan pada tahap awal dari pembangunan aplikasi untuk mengidentifikasi kode yang bermasalah. Keuntungannya adalah teknik ini bisa digunakan meskipun aplikasi belum selesai.

Analisis statis adalah metode analisis yang paling mudah dan mudah diterapkan dalam audit source code. Statis menurut definisi berarti sesuatu yang konstan. Analisis statis dilakukan pada kode statis, yaitu kode sumber mentah atau dekompilasi atau pada kode yang dikompilasi (objek), tetapi analisis dilakukan tanpa menjalankan aplikasi pada platform default. Jika di jalankan, maka tekniknya menjadi Live Forensic Analysis, seperti yang dilakukan pada penelitian (Rahman: 2015). Dalam kebanyakan kasus, analisis statis menjadi analisis kode melalui pencarian string statis. Skenario yang sangat umum adalah mencari pola kode yang rentan atau tidak aman dan menemukan pola yang sama di seluruh kode aplikasi.



Gambar 1 Flow Analisis Statis

Skenario yang sangat umum adalah mencari pola kode yang rentan atau tidak aman dan menemukan pola yang sama di seluruh kode aplikasi. Teknik pengumpulan data menggunakan tiga cara yaitu mengambil data APK, analisis dengan AndroBugs dan analisis dengan SUPER Android Analyzer. Tempat penelitian ini di Laboratorium Komputer Universitas Bina Darma.

C. HASIL DAN PEMBAHASAN

Berdasarkan hasil penelitian menggunakan dua program analisis yaitu AndroBugs dan Super Android Analyzer, di dapatlah data sebagai berikut:

1. Analisis dengan AndroBugs

Tabel 1. Hasil Analisis Statis untuk aplikasi mobile banking pertama

Item	*#*	*Vulnerability*
Critical	*C.1*	Aplikasi menyimpan informasi sensitif di dalam data storage.
	C2	Aplikasi menggunakan *implicit intent* untuk menjalankan layanan, sehingga pengguna tidak menyadari layanan apa yang dijalankan aplikasi.
	C3	Aplikasi tidak memeriksa validasi Sertifikat SSL. Hal ini bisa diakibatkan karena sertifikat CN yang ditandatangani sendiri, kadaluwarsa atau tidak cocok untuk SSL
Warning	*W1*	Aplikasi memuat kode dari luar aplikasi apk, hal ini sangat tidak disarankan karena memungkinkan *code injection* atau *tampering.*
	W2	Ditemukan akses penyimpanan eksternal (sebaiknya pengguna jangan menulis file penting ke penyimpanan eksternal).
	W3	Aplikasi bisa mengambil keluar komponen dan dianalisa oleh aplikasi lain. Sebaiknya harus menambahkan atau memodifikasi atribut ke *[exported="false"]*

		jika tidak diperlukan, atau bisa juga melindunginya dengan izin khusus dengan *"signature"* atau proteksi level yang lebih tinggi pada atribut *"android: permission"*.
	W4	Aplikasi mengambil *"device id(IMEI)"* melalui *"TelephonyManager.getDeviceId()"*.
	W5	Aplikasi mengambil 64-bit number *"Settings.Secure.ANDROID_ID"* sebagai *unique device identifier*. Akan tetapi kelemahannya beberapa perangkat dari manufacture tertentu memiliki *bug Android_ID* yang sama).
	W6	Aplikasi bisa mengirim sms *(sendDataMessage, sendMultipartTextMessage or sendTextMessage)*
	W7	Ditemukan *"setAllowFileAccess(true)"* tidak diset *(enabled by default)* pada WebView. Penyerang bisa menginjeks *malicious script* ke dalam WebView dan mengksploitnya untuk mengakses *local resources*. Pencegahannya dengan mematikan akses ke lokal file system (yang secara default di-enable)
	W8	*D*itemukan *"setJavaScriptEnabled(true)"* pada WebView, yang bisa diekspose untuk XSS attacks.
Notice	*N1*	ADB Backup *di ENABLED (default: ENABLED):* Resikonya data sensitif bisa diambil seperti *lifetime access token, username* atau *password*, dll.
	N2	Aplikasi menggunakan Android SQLite databases. Untuk Android < 4 vulnerable
	N3	Apapun yang didelete bisa di recovered oleh user atau attacker, khususnya untuk perangkat yang diroot.
	N4	Aplikasi memiliki code checking APK installer sources (misal dari Google Play store atau Amazone) untuk memastikan file tidak dihack
	N5	Aplikasi memiliki code checking package signature pada code. Untuk memastikan file apk tidak dirubah.
	N6	Ditemukan komponen *"exported"* (kecuali untuk Launcher) untuk menerima Google's "Android" *actions (AndroidManifest.xml):*

Tabel 2. Hasil Analisis Statis untuk aplikasi mobile banking kedua

Item	*#*	*Vulnerability*
Critical	*C.1*	*A*plikasi menyimpan informasi sensitif di dalam data storage
	C2	Beberapa URL tidak menggunakan SSL (Total:8)
Warning	*W1*	Ditemukan akses penyimpanan eksternal (sebaiknya pengguna jangan menulis file penting ke penyimpanan eksternal)
	W2	Aplikasi mengambil *"device id(IMEI)"* melalui *"TelephonyManager.getDeviceId()"*.
	W3	Aplikasi bisa mengirim *sms (sendDataMessage, sendMultipartTextMessage or sendTextMessage)*
	W4	Ditemukan *"setAllowFileAccess(true)"* tidak diset *(enabled by default)* pada WebView. Penyerang bisa menginjeks *malicious script* ke dalam WebView dan mengksploitnya untuk mengakses *local resources*. Pencegahannya dengan disabling akses ke lokal file system (yang secara degault di-enable)
	W5	*D*itemukan *"setJavaScriptEnabled(true)"* pada WebView, yang bisa diekspos untuk XSS attacks.
Notice	*N1*	ADB Backup di *ENABLED (default: ENABLED):* Resikonya data sensitif bisa diambil seperti *lifetime access token, username* atau *password*, dll.
	N2	Aplikasi menjalankan "root" atau *System Privilege Checking*
	N3	Apapun yang didelete bisa di recovered oleh user atau attacker, khususnya untuk perangkat yang diroot.
	N4	*BKS Keystore file assets/mbank.bks*
	N5	Keystores diproteksi dengan password dan menggunakan SSL-pinning (Total: 2).
	N6	*Native library loading codes(System.loadLibrary(...)) found.*

Tabel 3. Hasil Analisis Statis untuk aplikasi mobile banking ketiga

Item	*#*	*Vulnerability*
Critical	*C.1*	*The Keystores* menggunakan *"byte array"* atau *"hard-coded cert info"* dengan *SSL pinning (Total: 2).*
	C2	Beberapa URL tidak menggunakan SSL (Total:6)
	C3	Aplikasi tidak memeriksa validasi Sertifikat SSL. Hal ini bisa diakibatkan karena sertifikat CN yang ditandatangani sendiri, kedaluwarsa atau tidak cocok untuk SSL

Warning	*W1*	Aplikasi mengambil *"device id(IMEI)"* melalui *"TelephonyManager.getDeviceId()"*.
	W2	Aplikasi mengambil 64-bit number *"Settings.Secure.ANDROID_ID"* sebagai *unique device identifier*. Akan tetapi kelemahannya beberapa perangkat dar manufacture tertentu memiliki bug memiliki *Android_ID* yang sama).
	W3	Ditemukan *"setAllowFileAccess(true)"* tidak diset *(enabled by default)* pada WebView. Penyerang bisa menginjeks malicious script ke dalam WebView dan mengksploitnya untuk mengakses *local resources*. Pencegahannya dengan *disabling* akses ke lokal file system (yang secara degault di-enable)
	W4	*D*itemukan *"setJavaScriptEnabled(true)"* pada WebView, yang bisa diekspos untuk XSS attacks.
Notice	*N1*	ADB Backup di *ENABLED (default: ENABLED):* Resikonya data sensitif bisa diambil seperti *lifetime access token*, username atau password, dll.
	N2	Aplikasi menggunakan Android SQLite databases. Untuk Android < 4 vulnerable
	N3	Apapun yang didelete bisa di recovered oleh user atau attacker, khususnya untuk perangkat yang diroot.
	N4	Aplikasi memiliki *code checking package signature* pada code. Untuk memastikan file apk tidak dirubah.

2. Analisis dengan Super Android Analyzer

Tabel 4. Hasil Analisis Super pada mobile banking pertama

Level	Jumlah	Keterangan
Critical	0	-
High	5	• Algorithm lemah, (ada 2 code yang menggunakan algoritma SHA-1 dan MD5, yang mudah dipecahkan untuk mendapatkan plain text) • Terdapat 3 aktifitas membaca dan menulis di eksternal storage yang bisa dibaca aplikasi lain.
Medium	0	-
Low	223	• *Generic Exception in catch (177 code) Exception catching* harus spesifik. Bug ini bisa mengakibatkan error. • Math Random method (2 code): Method ini sebenarnya tidak murni acak, tapi di aplikasi digunakan untuk mengenerate code One Time Password. • *Unchecked output in Logs (41 code)*: Informasi sensitif seharusnya tidak boleh dicatat karena dapat menyebabkan informasi itu diungkapkan. • *Unknown permission* (3 code): Ada 3 permission aplikasi yang kurang jelas.
Warnings	21	• *Base64 decode (9 code):* Aplikasi menggunakan Base64decode • *Exported activity(3 code):* Bisa dimanfaatkan aplikasi lain. • *Exported receiver (3 code):* Bisa dimanfaatkan aplikasi lain. • *Exported service (4 code*): Bisa dimanfaatkan aplikasi lain. • *Get SIM OperatorName (1 code):* Aplikasi mencatat nama operator jaringan. Hal ini bisa saja tanpa sepngetahuan pengguna. • *Sending sms-mms (1 code):* Aplikasi bisa mengirim sms atau sms yang bisa saja tanpa sepengetahuan pengguna.

Tabel 5. Hasil Analisis SUPER pada mobile banking kedua

Level	Jumlah	Keterangan
Critical	2	• *WebView XSS (1 code):* Implementasi WebView yang tidak aman. Masalah ini dapat memungkinkan penyerang untuk melakukan eksekusi kode di WebView dan melakukan serangan *Cross Site Scripting* • *WebView ignores SSL errors(1ctivity)*: WebView mengabaikan error SSL dan menerima sertifikat SSL apa pun. Aplikasi ini bisa dimanfaatkan oleh *Man in the Middle attacks*.
High	5	• Aplikasi ini melakukan pemeriksaan perangkat yang di-rooting. Hal tersebut memperlihatkan dengan mengeksekusi kode tertentu jika perangkat di-root maka aplikasi bisa diambil alih. • Terdapat 3 aktifitas membaca dan menulis di eksternal storage yang bisa dibaca aplikasi lain.
Medium	1	• Aplikasi memungkinkan backup data melalui adb. Dengan akses fisik dengan adb untuk mendapatkan data pribadi aplikasi ke PC.
Low	223	• *Generic Exception in catch (101 code) Exception catching* harus spesifik. Bug ini bisa mengakibatkan error. • Math Random method (1 code): Method ini sebenarnya tidak murni acak, tapi di aplikasi digunakan untuk menggenerate code One Time Password.

		• *Unchecked output in Logs (127 code)*: Informasi sensitif seharusnya tidak boleh dicatat karena dapat menyebabkan informasi itu diungkapkan. • *Unknown permission (1 code)*: Ada 1 permission aplikasi yang kurang jelas.
Warnings	21	• *Base64 decode (11 code):* Aplikasi menggunakan Base64decode • *Certificate or Keystore disclosure (2 code):* Penguraian source kode dapat mengungkap sertifikat atau keystore hardcoded. • *Email disclosure (1 code):* Penguraian source kode dapat mengungkap email informasi. • *Exported activity(25 code):* Bisa dimanfaatkan aplikasi lain. • *Get SIM OperatorName (2 code):* Aplikasi mencatat nama operator jaringan. Hal ini bisa saja tanpa sepngetahuan pengguna. • *Get SIM Serial (1 code):* Aplikasi mencatat Serial SIM. Hal ini bisa saja tanpa sepngetahuan pengguna. • *Large heap (1 code):* Aplikasi ini membutuhkan large heap. • *URL Disclosure (25 code):* Dengan melakukan dekompilisasi source code bisa didapatkan private URL.

Tabel 6. Hasil Analisis SUPER pada mobile banking ketiga

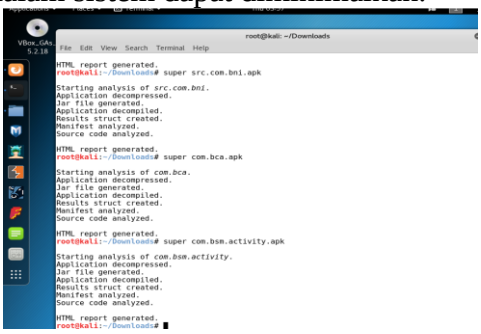
Level	Jumlah	Keterangan
Critical	0	-
High	25	• Algorithm lemah (25 code): Menggunakan algoritma yang lemah, memungkinkan penyerang untuk mendekrip sandi yang diacak. Code beberapa menggunakan MD5, RC4 dan DES
Medium	1	• Aplikasi memungkinkan backup data melalui adb. Dengan akses fisik dengan adb untuk mendapatkan data pribadi aplikasi ke PC.
Low	863	• *Generic Exception in catch (429 code) Exception catching* harus spesifik. Bug ini bisa mengakibatkan error. • *Math Random method (189 code):* Method ini sebenarnya tidak murni acak, tapi di aplikasi digunakan untuk menggenerate code *One Time Password*. • *Sleep Method (1 code):* Metode Sleep digunakan dengan vars sebagai argumen. Jika variabel tersebut dimodifikasi, itu bisa memaksa aplikasi berhenti. • Unchecked output in Logs (243 code): Informasi sensitif seharusnya tidak boleh dicatat karena dapat menyebabkan informasi itu diungkapkan. • Unknown permission (1 code): Ada 1 permission aplikasi yang kurang jelas.
Warnings	403	• *Certificate or Keystore disclosure (2 code):* Penguraian source kode dapat mengungkap sertifikat atau keystore hardcoded. • *Email disclosure (4 code):* Penguraian source kode dapat mengungkap email informasi. • *Exported activity(2 code):* Bisa dimanfaatkan aplikasi lain. • *IP Disclosure (356 code):* Proses Dekompilasi bisa melihat IP Private • URL Disclosure (39 code): Dengan melakukan dekompilisasi source code bisa didapatkan private URL.

Berdasarkan tabel hasil analisis super 4, 5, 6, dari pengujian menggunakan program analisis, program mobile banking yang memiliki tingkat resiko kerentanan lebih tinggi dibandingkan program lainnya terdapat pada program mobile banking kedua. Hal ini dikarenakan adanya celah keamanan berjumlah dua pada level kritis. Beberapa kerentanan memiliki kesamaan pada setiap program, lantas hal ini tidak menjadikan semua program itu sama, karena permission akses dari tiap program jelas berbeda dan memiliki tujuan yang bebeda pula. Informasi ini sudah selayaknya diberitahukan kepada pengguna agar mereka lebih berhati-hati dalam menggunakan program yang dipasang pada smartphone mereka.

Perlu diketahui juga bahwa hasil analisis menggunakan program Super Android Analyzer lebih rinci dibandingkan dengan AndroBugs, dilihat pada kelengkapan dan jumlah item yang ditemukan. Hal ini tentu sangat berpengaruh terhadap laporan rekomendasi kelemahan-kelemahan dari aplikasi mobile banking yang diaudit. Sejalan dengan penelitian

(Rafique:2013) yang mengatakan bahwa metodologi yang efektif dan alat yang tepat akan mendeteksi serangan dengan mudah dan perubahan dalam sistem dapat diminimalkan.

Gambar 2. Proses Analisis dengan AndroBugs pada file apk mobile banking



Gambar 3. Proses Analisis Statis menggunakan tools Super

Pada akhirnya, kedua program analisis tersebut membuktikan bahwa benar adanya jika program mobile banking memiliki beberapa tingkatan sistem keamanan untuk menunjukkan kepada pengguna bahwa mereka layak untuk dipercaya. Hal ini sejurus dengan penelitian yang pernah dilakukan oleh (Kuncoro, dkk: 2017) ada empat tingkatan yaitu umum, bahaya, tanda tangan, dan sistem. Tapi lain hal dengan penelitian oleh (Hintea, dkk: 2016), yang mengatakan sebagian besar aplikasi mentransimisikan informasi sensitif data pribadi pengguna yang tidak terenkripsi. Kita bisa lihat pada tabel 1, 2 ,3 dan 5 ditemukannya beberapa URL tidak menggunakan SSL bahkan mengabaikan error SSL dan menerima sertifikat SSL apa pun.

D. SIMPULAN DAN SARAN

Dari hasil analisis Statis pada ketiga aplikasi mobile Banking tersebut dapat dilihat ketiga aplikasi tersebut memiliki kerentanan dari yang kritis, high, medium yang sangat perlu diperhatikan. Terutama untuk aplikasi mobile Banking kedua yang memiliki 2 kerentanan kritis (hasil analisis statis dengan menggunakan tools SUPER).

Dari celah kerentanan yang ditemukan menjadi masukan perbaikan pengembangan aplikasi ke depannya dengan memperhatikan kelemahan yang ditemukan.

DAFTAR PUSTAKA

Goerdeler, K. P. (2015, September 29). Diakses dari KPMG Web Site: https://home.kpmg.com

Hintea, Diana., Taramonli, Chrysanthi., Bird, Robert., & Yusuf, Rezhna. (2016). Forensic Analysis of Smartphone Applications for. *Annual ADFSL Conference on Digital Forensics, Security and Law*, 167-182.

Kuncoro, Adam. Prayogo., Riadi, Imam., & Luthfi, Ahmad. (2017). Mobile Forensics Development of Mobile Banking Application using Static Forensic. *International Journal of Computer Applications*, 5-10.

Nallayam, Radhika. (2016, Juni 9). Diakses dari http://www.databreachtoday.asia

Rafique, Mamoona., & M.N.A.Khan. (2013). Exploring Static and Live Digital Forensics: Methods, Practices and Tools. *International Journal of Scientific & Engineering Research*, 1048-1056.

Rahman, Shuaibur., & Khan, M.N.A. (2015). Review of Live Forensic Analysis Techniques. *International Journal of Hybrid Information Technology, 8*, 379-388. Retrieved from http://dx.doi.org/10.14257/ijhit.2015.8.2.35

Su, Ming-Yang., Fung, Kek-Tung., Huang, Yu-Hao., Kang, Ming-Zhi., & Chung, Yen-Heng. (2016). *Detection of Android Malware: Combined With Static Analysis and Dynamic Analysis*. IEEE, 1013-1018.