# Pengujian Keamanan Aplikasi Mobile

Studi Kasus: Android

Zaki Akhmad | za at indocisc.co.id | 0xFD57BE80

## Abstrak

Pada paper ini akan dijelaskan metodologi pengujian keamanan aplikasi Android yang disertai dengan contoh implementasinya. Pengujian aplikasi secara teori terbagi menjadi dua, vaitu dinamis dan statis. Pengujian dinamis adalah pengujian yang dilakukan saat aplikasi dijalankan. Kebalikannya, pengujian statis adalah dilakukan dengan melihat kode sumber (source pengujian yang dilakukan tanpa menjalankan aplikasi. Contoh implementasi aplikasi yang diuji Berikut ini adalah garis besar sistematika adalah aplikasi Wordpress untuk Android (sebagai contoh aplikasi open source) dan aplikasi Twitter untuk Android (sebagai contoh aplikasi close source). Metodologi yang dijelaskan ini bersifat agnostik, sehingga bisa diimplementasikan pada beragam platform aplikasi mobile lainnya.

# 1 Pengantar

Aplikasi mobile saat ini semakin banyak dikembangkan seiring semakin terjangkaunya harga ponsel cerdas yang berarti semakin banyak juga penggunanya. Aplikasi mobile yang ada saat ini sangat beragam. Mulai dari aplikasi game sederhana yang tidak membutuhkan koneksi jaringan, aplikasi jejaring sosial hingga aplikasi mobile banking yang mampu melakukan transaksi keuangan.

Pengguna akhir pada umumnya tidak memiliki kesadaran yang tinggi akan keamanan aplikasi. Misalnya, terdapat aplikasi game yang diunduh bisa jadi mengandung *malware* yang mengakses data di ponsel. Atau terdapat aplikasi yang menyimpan kredensial di dalam ponsel dalam keadaan tak terenkripsi. Contoh lainnya, aplikasi yang tidak menggunakan jalur terenkripsi saat melakukan otentikasi atau mengirim informasi sensitif.

# 2 Teori

Berikut ini adalah dasar teori pengujian aplikasi mobile. Pengujian diklasifikasikan menjadi dua yaitu dinamis dan statis. Pengujian dinamis dilakukan dengan menjalankan aplikasi, untuk selanjutnya dilakukan analisis. Pengujian statis dilakukan tanpa menjalankan aplikasi, analisis code) aplikasi.

pengujian aplikasi mobile:

- 1. Dinamis
  - 1. Analisis network traffic
  - 2. Analisis remote services (HTTP/SOAP/dll)
  - 3. Debug aplikasi
- 2. Statis
  - 1. Dapatkan aplikasi
    - 1. Ekstrak aplikasi dari device
    - 2. Dapatkan berkas apk dari pengembang
  - 2. Lakukan reverse engineering
  - Lakukan source code review
    - 1. Menggunakan static analysis tool
    - Manual
  - 4. Lakukan disassembly
  - 5. Lakukan patch

## 3 Konfigurasi Lab

Pada bagian ini akan dijelaskan konfigurasi lab yang digunakan dalam melakukan pengujian. Lab pengujian dapat dilakukan langsung dari perangkat mobile atau bisa juga menggunakan emulator. Berikut detail kedua konfigurasi lab tersebut:

### 3.1 Langsung dari Device

Berikut ini adalah daftar kebutuhan untuk konfigurasi lab pengujian langsung dari device:

- Mobile device (MD)
- Kabel data
- Access point (AP)
- Hub
- Kabel LAN
- Komputer (PC)

Untuk akses langsung ke device dari komputer digunakan kabel data.

PC --- kabel data --- MD

Gambar 1: Skema Akses Langsung

Access point, Hub dan kabel LAN diperlukan untuk pengujian pada sisi keamanan jaringan. Berikut ini adalah diagram skema topologi pengujian keamanan jaringan aplikasi.

Internet --- Hub --- AP – MD

Komputer

Gambar 2: Skema Jaringan

### 3.2 Menggunakan Emulator

Apabila tidak memiliki *mobile device* bukan berarti tidak bisa melakukan pengujian. Pengujian aplikasi *mobile* dapat memanfaatkan emulator yang ada.

Platform mobile pada paper ini dibatasi pada Android. Pada paper ini tidak akan dijelaskan bagaimana menyiapkan: instalasi & konfiguras SDK Android. Untuk petunjuk bagaimana menyiapkan SDK Android dapat dilihat pada situs pengembang Android. Usai SDK Android siap, dapat dipilih versi Android mana yang emulatornya ingin disiapkan. Silakan pilih sesuai kebutuhan. Emulator pada Android disebut dengan Android Virtual Device (AVD). Versi terakhir emulator Android terbaru saat paper ini ditulis adalah Android 4.03 dengan API Level 15.

	Create new Android Virtual Device (AVD)
Name:	
Target:	~
CPU/ABI:	Android 2.2 - API Level 8
SD Card:	Android 3.1 - API Level 12
	Android 4.0 - API Level 14
	Android 4.0.5 - API Level 15
Snapshot:	Enabled
Skin:	Built-in:
	O Resolution: x
Hardware:	Property Value New
	Delete
Overri	de the existing AVD with the same name

Gambar 3: Android Virtual Device

# 4 Pengujian Dinamis

## 4.1 Aplikasi Wordpress

Wordpress merupakan aplikasi F/OSS Content Management System (CMS) blog. Selain dikembangkan aplikasi blog engine Wordpress itu sendiri, dikembangkan juga aplikasi Wordpress untuk Android. Aplikasi Wordpress untuk Android ini merupakan aplikasi klien yang digunakan untuk mengakses blog Wordpress.

Pada lab pengujian ini, diinstall aplikasi Wordpress pada server, dan aplikasi Wordpress untuk Android pada ponsel Android. Skema jaringan mengikuti Gambar 3 hanya saja tidak perlu tersambung ke Internet karena aplikasi Wordpress terinstall pada server sendiri *(self hosted Wordpress blog)*. Alasan menggunakan *self hosted Wordpress blog* adalah agar analisis dapat dilakukan lebih mendalam dan detail sekaligus lebih sederhana karena tidak memerlukan akses Internet.

Berikut detail versi aplikasi yang digunakan:

- Wordpress: 3.3.1
- Wordpress untuk Android: 2.0.7

### 4.1.1 Analisis Network Traffic

Pertama akan dilakukan analisis network traffic. Tujuan dari aktivitas ini adalah mencari informasi sensitif yang tidak dienkripsi.

Aplikasi Wordpress untuk Android akan digunakan untuk mengakses *self-hosted Wordpress blog* dan melakukan aktivitas sebagai berikut:

- Akses sebagai publik (tanpa otentikasi)
- Melakukan otentikasi, masuk sebagai authorized user.
- Menulis tulisan baru
- Akses menu konfigurasi Wordpress
- Mengubah password

Semua aktivitas ini akan dianalisis dari hasil packet capture-nya. Berikut ini merupakan hasil analisis berkas pcapnya.

Time	Sturre	Destination	Protocal	Lanoth	10%
	7212001011120	1061100161110	100		council a concolinative noral confecto incuento intreviato nerre la concolinative norale en concernante en concernate en concernante en concernat
36 4.840742	192.168.2.170	72.233.61.123	102		66 354/5 > http://www.seq=13/ Ack=254 Win=6912 Len=0 Tsval=//9253 Tsecr=44068685
37 4,936991	192.168.2.170	216, 155, 195, 253	HETP		879 GET /v1/pushchanne1/zak1akhmad7cm001w3d.8338a1d+L85gJ.Ls38e07581yxJ8H18YoPp9Rue34k
38 4,999016	Attiedre_41:11:09	sansunge_23tbat32	APP		60 WHO NAS 192,108,2,1707 Tell 192,108,2,1
39 5.005363	54H5UF9E_23:54:32	ACCIEGITE_41:11:09	100		SU 192.100.2.170 19 at 00171110123104132
41.0.000700	100,100,0,170	70,000,41,100	700		TA DOGT - NAME (COL) CARDON MARTINE MARTINE CON CARDON TO A TANK TO A TANK TO A TANK TANK TANK TANK TANK TANK TANK TA
42 9 220421	70 200 61 122	102 169 2 170	109		74 35007 9 Http://statu. 560-0 Mth-3600 Del-0 H56-1400 SKOL-PEN-1 15V8C-77572 15801-0 W
42 0 220160	192 168 2 170	192 168 2 107	TOP		74 49976 a bitte [SYN] Secto Min-5840 Lan-0 MSS-1460 SACK DEBN-1 TOval -779907 TSector M
44.8.338220	192, 168, 2, 107	192, 168, 2, 170	TOP		74 http > 49976 [SYN, 40X] Secup Ackul Wing14490 Lenup MSSg1490 S4CK PEPMg1 TSva1u5977
45.8.307847	192, 168, 2, 170	72, 233, 61, 123	TOP		66 30007 a http://doc.j.Senal.Ackal.Mine/S840 Lane0 TSval=770609 TSerr=44/06/548
46 8,412150	192,168,2,170	192,169,2,107	TOP		66 49676 > http [ACK] Sect1 Acks1 Wins5840 Lenco TSvals779609 TSecrs587725
47 8,469136	192,168,2,170	72,233,61,123	HTTP		201 GET /blavatar/f8058ab70af1c8b2e3f18ae036628d1d?s=606d=404 HTTP/1,1
48 8.534556	192.168.2.170	192,168,2,107	TOP		298 (TCP segment of a reassembled PDU)
49 8.534619	192.168.2.107	192.169.2.170	TOP		66 http > 46676 [ACK] Seq=1 Ack=171 Win=15616 Len=0 TSval=587774 TSecr=779623
51 8,563289	192.168.2.107	192.168.2.170	TOP		66 http > 46676 (ACK) Seq=1 Ack=552 Win=16640 Len=0 TSval=587781 TSecr=779625
52 8.715250	72.233.61.123	192.169.2.170	TOP		66 http > 39007 [ACK] Seq=1 Ack=135 Win=7168 Len=0 TSval=44059657 TSecr=779616
53 8,721480	72.233.61.123	192.168.2.170	HTTP		328 HTTP/1.1 404 Not Found (text/html)
54 8.725715	192,168,2,170	72.233.61.123	TOP		68 39007 > http [ACK] Seq=136 Ack=263 Win=6912 Len=0 TSval=779642 TSecr=44069658
rane 50: 447	bytes on wire (3576	bits), 447 bytes c	ptured (3576 bi	ts)	
thernet II, S	rct sansungE_2315a1	32 (801/11181231581)	2), Ust: Quanta	0_001801	A (68194181106180194)
Internet Proto	col Warsion 4, Src:	192.168.2.170 (192	168.2.170), Dst	192.168	.2.107 (192.168.2.107)
	antrol Protocol, Sr	C POPT: 400/0 (400/	(col)	cp (au),	seq: 1/1, ACK: 1, Len: Jai
Pansaission C		oycesi: #HOLIYOI, #:	0130171		
[2 Reassembled	To beginnes toos				
12 Reassembled 00 e8 9a 8f 6	5 8c 94 8c 71 f8 2	3 5a 32 08 00 45 00	fq .#Z2.	.E.	



Berdasarkan Gambar 4, dapat terlihat bahwa IP mobile device yang digunakan adalah 192.168.2.170. Hal ini penting untuk diketahui dalam melakukan analisis lebih dalam dan menyusun formula filter packet pada Wireshark yang digunakan sebagai tool untuk menganalisis packet capture.

Lalu kotak merah pada Gambar 4 yang kedua menunjukkan request POST yang dilakukan oleh aplikasi Wordpress untuk Android. Request ini menarik untuk ditelusuri lebih lanjut. Untuk itu dilakukan follow TCP stream pada paket ini.

<b>X</b> ©	Follow TCP Stream			
	Stream Content			
PATT All gran (mr. 24) at TFPL1.1 part and the second of the TFPL1.1 part of th	Inngarats-parato-values altravioletta in rindra ber-rented s-manaport ber-rented s-manaport istation handle s-related v/booleans-/values-/para	vei do la y de-yvaluee-y para et le popular de y anno 1990 et le la popular de la construction et la construction de la construction ta la construction de la construction ta la construction de la construction de la constru	m-param-salue-string- vetring-restential, ever tring-string-factor (pre-val respective) tring-salution of the val etter/1.1 200 04	2a-/string=/ gress/string=/ user/ tuer/
				¢
Entire conversation (1406 bytes)				×
ASCII	C EBCDIC	U Hex Dump	C Arrays	• Rew
E Belp			M Filter Out This Stream	€ Close

Gambar 5: String Username & Password

Dari hasil analisis paket terlihat saat user akan menulis tulisan baru, username dan password dikirimkan dalam bentuk string tanpa melalui kanal terenkripsi.

Hasil penelusuran lebih lanjut menunjukkan bahwa username dan password **selalu** disertakan dalam setiap request aplikasi saat akan mengakses server.



Gambar 6: Ubah Password

Saat password user Wordpress diubah bukan melalui aplikasi Android, maka aplikasi Android tidak lagi bisa mengakses blog. Hal ini dilakukan untuk mengetahui bagaimana aplikasi menyimpan password.



Gambar 7: Login Usai Password Diganti

Ternyata aplikasi menyimpan password pada semacam berkas konfigurasi. Jadi, perlu dilakukan konfigurasi ulang pada aplikasi dengan mengubah nilai password untuk bisa kembali login.

Dari keseluruhan pengujian, terlihat bahwa desa aplikasi wordpress untuk Android tidak aman. Username dan password dikirim dalam bentuk string dalam setiap request melalui kanal tak terenkripsi.

Cara sederhana meningkatkan keamanan aplikasi Wordpress untuk Android dapat dilakukan dengan menggunakan kanal terenkripsi.Pengguna Android hanya bisa mengakses self-hosted Wordpress blog menggunakan protokol HTTPS. Selanjutnya perlu dipikirkan desain aplikasi yang lebih baik yang **tidak harus selalu** mengirimkan username dan password dalam setiap requestnya. Persisnya pada bagian mekanisme otentikasi aplikasi.

### 4.1.2 Debug Aplikasi

Untuk debug aplikasi, lab yang akan digunakan adalah dengan menggunakan emulator.

Jalankan emulator, lalu gunakan Android Debug Bridge (adb).

```
$ adb devices
List of devices attached
emulator-5554 device
```

Jika muncul device emulator, maka emulator sukses terdeteksi. Aktivitas selanjutnya akan memanfaatkan utiliti dari adb

```
$ adb shell
#
```

Tanda # menunjukkan akses root. Selanjutnya cari informasi sensitif yang dapat diakses.

Informasi sensitif dapat dicari di database. Android menggunakan sqlite3 sebagai database. Pertama, perlu dicari di direktori mana database aplikasi disimpan.

Hasil penelusuran menghasilkan database Wordpress untuk Android disimpan di direktori

Dari keseluruhan pengujian, terlihat bahwa desain data/data/org.wordpress.android/d ata/data/org.wordpress.android/d

Masuk ke direktori tersebut dan lihat daftar berkas yang ada di dalamnya

```
# ls
webviewCache.db
webview.db
wordpress
```

Gunakan utility sqlite3 untuk mengakses database. Berdasarkan hasil penelusuran, ternyata informasi blog tersimpan di database wordpress # sqlite3 wordpress
SQLite version 3.6.22
Enter ".help" for instructions
Enter SQL statements terminated
with a ";"
sqlite>

#### Lalu jalankan perintah dump.

- Jalankan kali pertama
- Log in
- Twit

Berikut ini adalah hasil analisis network traffic aplikasi Twitter untuk Android.

Analisis pertama kali adalah memfilter paket capture dengan filter "http contains twitter". Dari paket pertama hasil fiter tersebut, terlihat paket capture saat proses instalasi dari Google Play. Dari hasil penelusuran tidak ditemukan informasi sensitif pada paket ini.



Gambar 8: Hasil Perintah Dump

Terlihat bahwa aplikasi Wordpress untuk Android menyimpan username & password dalam database dengan keadaan plain text. Hal ini menunjukkan aplikasi Wordpress untuk Android tidak aman.

#### 4.2 Aplikasi Twitter

Berdasarkan informasi dari situs pengembang twitter, twitter dikembangkan dengan memanfaatkan perangkat lunak *open source*. Begitupun dengan aplikasi Twitter untuk Android. Namun biarpun menggunakan perangkat *lunak open source*, bukan berarti kode sumber aplikasi Twitter untuk Android dibuka.

#### 4.2.1 Analisis Network Traffic

Analisis network traffic aplikasi Twitter untuk Android menggunakan skema jaringan Gambar 2.

Aktivitas yang dilakukan adalah sebagai berikut:

 Install aplikasi Twitter untuk Android dari Google Play

Gambar 9: Akses Twitter di Google Play/Market

Selanjutnya, penelusuran dilanjutkan pada paket berikutnya masih dengan filter yang sama. Hasil TCP Stream menunjukkan response yang mengandung kata MANIFEST. Untuk menemukan paket lain yang mengandung kata MANIFEST, dapat dilakukan dengan mengubah filter menjadi "http contains MANIFEST". Perlu diingat filter ini merupakan *case sensitive*.

~			FORDW TCP Scream			
			Stream Cantent			
ontent-Length: 2364191						
ontent-Type: application/vnd	android.package-archive					
cept-Ranges: bytes						
te: Mon, 09 Apr 2012 08:13:	G OMT					
rver: Blob HTTP Service						
KSS-Protection: 1; mode=blo	k					
rame-Options: SAMEORIGIN						
st-Modified: Mon, 02 Apr 20	2 22:50:13 CMT					
nnection: close						
Content-Type-Options: nosni	1					
ti.a.ztf	NETA-INF/M/NHEESI MF. I	ня		eSV	w	
	W. +.2[.]	:Sd	.Z.TZ. 1		GTPXE	
.J	110.1			f. \$ . 6 nbf. / }		
#62. Pr.c.*')			k.M. 7	.V0.8.0	.U.f)Ak* .NrDv.k 1 1	.M.
eVZ	7.1	Y	40v.^:	5N.H	G	
vb.0#.H.B.c.PC1.		1			1'*9.m. Sb'vA	
* a. a	D.G 2. 1 iX D. ^ S	*d.				
6+<	- 8 - 1	w.m	A.T	.0n5		
a.B	uG0Y./ff.p0sl*	.N7vx3G 1				
f)			d. v70bNe	+.>1Kf=7.1.7		
A1.vc[.e.]8fNb.M <sup>2</sup> /V						
	7.)(:	ed.s\Fl.}				
.t. 125.r. xf.1.sk7.xm.	1d.r.BluS		1mK.h	B		
n.*			W. W. H. WK.			
word, G. tk. Sup. J. at	75.H	1. 15. 80. d	.afx.3[30	6. f. a r 01		
0.M. 0t.wlx. N	**.815.7. x.k.).xt.*	dt. 7.7b8clum	0			
	10B		WB .			
H.Y	v).htwt./	. [.s0.p t	0 7n 10f	br	111 A F. H C C A A	
-						_
tire conversation (2365070 bytes)						
🗱 tind 🖌 🖌 s	e da Beint	ASCI	C EBCDIC	O Hex Dump	C Arrays 💿 Rew	

Gambar 10: Response MANIFEST.

Analisis dari paket ini adalah paket ini merupakan*follow*. Percobaan untuk melakukan re-assemble response dari request paket aplikasi Twitter untuk terhadap berkas gambar, masih gagal dilakukan. Android. Isi MANIFEST tidak bisa terbaca karena tidak berada dalam format clear text.

Skenario selanjutnya adalah melakukan reassemble paket untuk bisa mendapatkan berkas .apk dari aplikasi. Usaha mencari string sensitif (username & password) tidak ditemukan karena jalur komunikasi yang digunakan terenkripsi begitupun isi dari paket tersebut. Bahkan isi twit-pun tidak bisa didapatkan.



Gambar 11: Reassemble Paket Dari hasil reassemble, didapatkan berkas apk

Dari hasil reassemble, didapatkan berkas apk berukuran 2.3 MB

\$ ls -1 2364191 2012-04-23 17:52 twitter.apk

Analisis berkas apk lebih lanjut akan dibahas pada bagian Analisis Statis. Sekarang kembali melakukan analisis *network traffic*.

	Wireshark: 4219 Expert Infos	
mors: 0 (0) Warnings: 4	(501) Nobes: 10 (1594) Chats: 124 (2124) Debalis: 4219	Ins
Greup Protocol	Summary	. â
- Sequence HTTP	GET /profile_images/SS8870659/Z4KI27_Low_normal.jpg HTTP/1.1\/\n	1
- Sequence HTTP	GET /profile_imapes/1773244478/402874_2799252193249_1616871530_2560814_88483765_n_normal.jpg HTTP/1.1	
- Sequence TCP	Connection establish request (SVN): server port imaps	2 2
- Sequence TCP	Connection establish acknowledge (SIN+ACK): server port imags	2
- Sequence HTTP	GET/dete/Feed/epi/user/ze%40owesp.org?imgmac=1024&max=result=1000&thumbsize=144e,1024a&visibility=visibil	2
- Sequence HTTP	HTTP/1.1 304 Nat Modifiedu/in	+ TA7
- Sequence HTTP	GET /prefile_images/1826528511/Untikled_normal.png HTTP/1.1\r\n	, UVV
- Sequence HTTP	GET /profile_images/2098840509/cover-8_normal.jpg HTTP/1.1\/\n	4
- Sequence HTTP	GET /profile_images/2091593835/Toto-twitter-baru_normal.JPG HTTP91.1\/\n	1
- Sequence HTTP	GET /profile_images/1440149453/twitter-logo-kompas-tv_normal.jpg HTTP/1.1\r\n	1
- Sequence HTTP	GET /profile_images/114631381/pboto_normal.pg HTTP/1.1\v/n	
- Sequence TCP	Connection establish request (3/N): server port horroom	, <u> </u>
- Sequence TCP	Connection establish acknowledge (DIN+ACK): server port hpvroom	4
- Sequence HTTP	GET /profile_images/2016998541/6d8944a62fb7e3c04c31328f73041dc25abcdefg_normal.jpg HTTP/1.1\/\n	1 1
- Sequence HTTP	GET /profile_images/1157953544/evetar_normal.jpg HTTP/1.1/r/n	1 + •
- Sequence HTTP	GET /profile_images/451912986/affan-ong_normal.jpg HTTP)1.1\r/n	
- Sequence HTTP	GET /profile_images/1838342918/swampy_normal.png HTTP/1.1\/\n	10
- Sequence HTTP	GET /profile_images/1626928100/gondrong_normal.jpeg HTTP/1.1\r/n	1
- Sequence HTTP	GET/profile_images/2065109964/3bc2d6f080d611e18cf91231380f6290_7_normaljpg HTTP/1.1\r/n	1
- Sequence HTTP	GET /prefile_images/1979165979/rawa_normal.jpg HTTF)1.1/v/n	1
- Sequence HTTP	GET /profile_images/1090051912/lbukika_2_normal.jpg HTTP)1.1\r/n	1
- Sequence HTTP	GET /profile_images/1644271600/ynugrahoLINEART-KUNIS-JENGGOT-TIPIS_normal.grg HTTP/1.1\r/in	1 / 1
- Sequence HTTP	GET / data/Feed/api/user/zakiakhmad7imgmass=1024&mas-results=1000&htmmbsize=144u,1024u&visibil@pvisible&kin	
-Sequence HTTP	GET /utm.giPutmwv=6.8.1me8utmn=8669799388utme=8(1Mpg8cation%20Version*2Distribution%20Source*3ICF	, / u
Sequence HTTP	POST / HTTP/1.1\/\m	1
-Sequence HTTP	HTTP/1.0 200 GK/p/n	1
- Sequence HTTP	GET /profile jmages/1821418671/image_normal.jpg HTTP/1.11/in	1 🔍

Gambar 12: Twitter Get Image

Hasil analisis berikutnya, didapatkan request image avatar dari pengguna twitter yang di-

# Gambar 13: Penggunaan Protokol TLSv1

## 4.2.2 Debug Aplikasi

 84
 41
 11
 19
 96
 71
 16
 28
 34
 00
 00
 00
 00
 00
 01
 01
 01
 02
 02
 03
 02
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 00
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03
 03<

Debug aplikasi Twitter untuk Android menggunakan konfigurasi lab dengan emulator. Karena pada emulator tidak terdapat akses ke Google Play/Market maka aplikasi twitter akan langsung diinstall dari berkas apk tanpa akses ke Google Play/Market.

Instalasi dilakukan menggunakan tool apk

\$ adb -s emulator-5554 install
twitter.apk
1435 KB/s (2364191 bytes in 1.608s)
pkg: /data/local/tmp/twitter.apk
Success





Gambar 15: Dump Global.db Twitter

Dari hasil dump database global.db tidak didapatkan informasi sensitif. Yang bisa dibaca hanya username namun tidak tersedia informasi password.

Gambar 14: Twitter pada Emulator

Selanjutnya, masih menggunakan tool adb, akan dicoba mencari informasi sensitif yang tersimpan di dalam device (emulator). Masuk ke direktori database twitter yang berada di

data/data/com.twitter.android/dat
abases

List berkas yang ada di direktori tersebut:

# ls 15016157.db 0.db

global.db

Akses database menggunakan sqlite3. Pertama akan didump database global

Selanjutnya dicoba dump database lainnya.

a : bash	twitter : adb		za : eclipse	
sqlite>				÷
COMMIT:	AS C UN U.USEr_Id=c.tag MHb	Re (u.type=9 OR u.t	/pe=10) AND U.Tag=-1 ORDER BY	1. 10;
CREATE VIEW user_recommendations_view	# AS SELECT u.*,c.user_id AS	conn_user_id,c.use	name AS conn_username,c.name.	AS conn_name FROM user_groups_view A
image FROM search_queries LEFT JOIN u	user_groups_view ON search_c		_groups_view.tag WHERE user_g	roups_view.type=6;
query_id,user_groups_view.username AS	S username, user_groups_view.	user_id AS user_id,	iser_groups_view.image_url AS	image_url,user_groups_view_image_AS
(REATE VIEW slug users view AS SELECT	I search queries, id AS id	search queries name	AS name search queries query .	AS query search queries query id AS
status_id, ref_id AS rt_orig_ref_id FR	ROM status_groups_view WHERE	type=0 AND tweet_t	/pe=1 AND sender_id=owner_id G	ROUP BY rt_orig_status_id) AS retwee
CREATE VIEW status_groups_retweets_vi	iew AS SELECT status_groups_	view.*, retweets.*	FROM status_groups_view LEFT J	OIN ( SELECT g_status_id AS rt_orig_
users.image AS image FROM lists,users	s WHERE lists.creator id=use	rs.user id;	nume h5 user nume, user at nume h	s name, users. timige_on c HS timige_on c,
stor id AS croater id lists i follow	AS i follow lists is last	ists subscribers AS	subscribers, Lists.members AS	S page upors image upl AS image upl
CREATE VIEW lists view AS SELECT list	tsid AS _id,lists.owner_ic	AS owner_id,lists.	type AS type,lists.list_id AS	list_id,lists.list_name AS list_name
ers s ON sender_id=s.user_id;				tende and their messages carries and
CREATE VIEW messages conversation AS	SELECT nessages.*.s.usernat	e s username.s.name	s name, s, image url s profile	image url FROM messages LEFT JOIN us
ername, s.name s_name, s.image_url s_pr id_OPDEP_PV_created_ASC)_CPOUP_PV_thr	rorite_image_url FROM messag	es LEFT JOIN users	r uw recipient_id=r.user_id LE	FI JUIN USERS S UN Sender_id=s.user_
CREATE VIEW messages_threaded AS SELE	ECT * FROM (SELECT messages.	*,r.username r_user	name,r.name r_name,r.image_url	r_profile_image_url,s.username s_us
d;				
sername AS username users name AS nam	s sender_id, messages.recipie	url users image AS	mane FROM messages users WHER	messages.is_tast AS is_tast,users.u
CREATE VIEW messages_sent_view AS SEL	LECT messagesid AS _id,mes	sages type AS type,	messages.msg_id AS msg_id,mess	ages.content AS content,messages.cre
1d;				
rs.username AS username,users.name AS	5 name, users.image url AS in	age url,users.image	AS image FROM messages, users	WHERE messages.sender id=users.user
created AS created messages sender 1	id AS conder id messages reg	inient id AS recipi	pe,messages.msg_10_AS_msg_10, ant id maccanes is read AS_is	read messages is last AS is last use
ps LEFT JOIN users AS user ON user_gr	roups.user_id=user.user_id;			
AS statuses, user.geo_enabled AS geo_	enabled,user.image AS image	user.friendship AS		me AS friendship_time FROM user_grou
ied AS verified, user.profile created	AS profile created, user.ima	ige url AS image url	user.followers AS followers,u	ser.friends AS friends,user.statuses
me.user.description_AS_description.us	ser.web_url_AS_web_url_user.	ba color AS ba colo	user location AS location us	er.protected AS protected.user.verif
CREATE VIEW user_groups_view AS SELEC	CT user_groupsid AS _id,us	er_groups.type AS t	/pe,user_groups.tag AS tag,use	r_groups.owner_id_AS_owner_id,user_g
N statuses.author_id=user.user_id LEF	FT JOIN users AS sender ON s	tatus_groups.sender	_id=sender.user_id;	÷

Gambar 16: Dump 15016157.db Twitter

Hasil dump database lainnya juga tidak menemukan username & password dalam keadaan plaintext. Yang dapat ditemukan hanya *direct message* di twitter pada database 15016157.db

## **5** Analisis Statis

Analisis statis pada prinsipnya adalah melakukan code review. Seorang code reviewer harus mengetahui prinsip dan implementasi dari secure coding. Selain itu, seorang code reviewer juga harus tahu bagaimana menerapkan konfigurasi dengan least privilege.

### 5.1 Aplikasi Wordpress

Berdasarkan hasil analisis dinamis, dicoba dicari bagian dari source code yang menangani otentikasi, dan bagaimana mekanisme menyimpan password. Karena aplikasi Wordpress Privilege ini memang dibutuhkan oleh aplikasi. untuk Android adalah aplikasi dengan kode sumber terbuka, jadi tidak diperlukan melakukan menulis tulisan baru, membaca komentar; reverse engineering. Reverse engineering sederhananya adalah kegiatan membalikkan dari eksternal storage untuk menyimpan file hasil binary aplikasi ke source code.

Berikut ini adalah hasil penelusuran kode sumber aplikasi Wordpress untuk Android. Pertama lakukan pencarian string dengan kata kunci "password". Tool yang digunakan adalah ackgrep yang mampu mencari string dalam satu direktori.



- Kamera
- Menulis pada eksternal storage
- Lokasi

Misalnya: Internet untuk mengakses arsip tulisan, Kamera untuk mengambil gambar; Menulis pada kamera; dan Lokasi untuk fitur posting with geolocation. Jadi aplikasi Wordpress untuk Android sudah menerapkan konsep least privilege.

### 5.2 Aplikasi Twitter

Karena aplikasi Twitter untuk Android tidak open source maka pertama-tama perlu dilakukan reverse engineering. Reverse engineering aplikasi Android dapat dilakukan dengan mengacu pada artikel yang ditulis oleh Jack Maninno.

Ekstrak aplikasi twitter.apk

\$ unzip twitter.apk

Didapatkan berkas sebagai berikut:

```
$ ls
```

AndroidManifest.xml assets classes.dex META-INF orq res

Gambar 17: Kata Password dalam Kode Sumber resources.arsc

Dari hasil pencarian menggunakan ack-grep dapat terlihat, password yang digunakan adalah httppassword. Dengan kode sumber aplikasi ini maka dapat terlihat bahwa memang aplikasi mengirimkan password dalam keadaan plain-text melalui protokol http.

Lalu periksa berkas AndroidManifest.xml Berkas AndroidManifest.xml adalah berkas yang memuat informasi mendasar aplikasi. Perhatikan permission yang dimiliki aplikasi. Apakah permission tersebut benar-benar dibutuhkan aplikasi (ingat kembali konsep keamanan: least privilege).

Berdasarkan penelusuran, permission yang dimiliki oleh aplikasi Wordpress untuk Android adalah:



Gambar 18: Informasi Twiter dari Legal.html

Dari berkas legal.html vang terdapat dalam direktori assets diketahui aplikasi Twitter untuk Android dikembangkan menggunakan komponen berikut:

• Internet

- Bouncy Castle Crypto
- Google Data Java Client
- Jackson Java JSON-processor Core

Selanjutnya cari informasi lebih lanjut soal komponen-komponen ini. Apa kegunaanya, bagaimana implementasinya.

Berdasarkan informasi dari situs bouncy castle, bouncy castle merupakan API kriptografi. Dalam implementasinya di Android, yang digunakan adalah yang berbahasa JAVA. Jadi, aplikasi Twitter untuk Android menggunakan bouncy castle sebagai API kriptografi.

Google Data Java Client berguna untuk mengirim query Google Data API (GData) dan melakukan interpretasi responnya. Belum diketahui apa kegunaan library ini pada aplikasi Twitter untuk Android.

Jackson Java JSON-processor Core merupakan *high-performance JSON processor*. Aplikasi Twitter untuk Android menggunakan library ini untuk memproses JSON.

Selanjutnya, lakukan analisis terhadap berkas AndroidManifest.xml. Berkas AndroidManifes.xml berada dalam format biner, jadi informasinya tidak bisa terbaca. Untuk itu perlu digunakan tool apk-tool sebagai tool untuk melakukan reverse engineering berkas apk.

\$ ./apktool d twitter.apk twitter

- I: Baksmaling...
- I: Loading resource table...
- I: Loaded.
- I: Loading resource table from file:

/home/za/apktool/framework/1.apk

- I: Loaded.
- I: Decoding file-resources...
- I: Decoding values\*/\* XMLs...
- I: Done.
- I: Copying assets and libs...

Sekarang sudah didapatkan berkas

AndroidManifest.xml dalam format text. Lakukan analisis terhadap berkas ini.



Gambar 19: Berkas AndroidManifest.xml Twitter

Terlihat permission yang dimiliki aplikasi. Berbeda dengan aplikasi Wordpress untuk Android, aplikasi Twitter untuk Android tidak memiliki privilege untuk akses hardware kamera. Jadi pengguna tidak bisa mengakses kamera langsung dari aplikasi Twitter untuk Android.

### 6 Ringkasan

Berikut ini adalah beberapa ringkasan yang bisa dibuat:

- 1. Pengujian aplikasi mobile dilakukan pada level aplikasi, bukan pada level sistem operasi perangkat mobile
- 2. Pengujian aplikasi mobile dapat dilakukan langsung pada device atau dapat juga menggunakan emulator
- Pengujian aplikasi mobile dibedakan menjadi 2: pengujian dinamis, pengujian yang dilakukan saat aplikasi dijalankan; dan pengujian statis, pengujian tanpa menjalankan aplikasi.
- Pengujian dinamis berusaha mencari informasi sensitif dalam keadaan tidak terenkripsi: baik saat melalui transport layer maupun saat disimpan dalam database di perangkat.
- Pengujian statis dasarnya adalah melakukan code review untuk memahami cara kerja aplikasi. Untuk aplikasi yang tidak bersifat opensource, perlu dilakukan

reverse engineering.

# 7 Kesimpulan & Saran

Telah dijabarkan metodologi pengujian aplikasi mobile dengan studi kasus aplikasi Android: Wordpress untuk Android dan Twitter untuk Android

Berikut ini adalah kesimpulan dari penelitian ini:

- 1. Aplikasi Wordpress untuk Android selalu mengirimkan userid dan password dalam setiap request-nya dalam keadaan tidak terenkripsi.
- 2. Aplikasi Wordpress untuk Android menyimpan userid dan password pada database dalam keadaan tidak terenkripsi. http://jack-
- 3. Aplikasi Twitter untuk Android menggunakan transport layer yang password, dan twit dikirim dalam keadaan Platform, O'Reilly terenkripsi.
- 4. Aplikasi Twitter untuk Android menyimpan password pada database dalam keadaan terenkripsi. Pesan langsung (direct message) masih dapat dibaca karena tersimpan dalam format clear text.

Berikut ini adalah saran untuk mengembangkan

penelitian ini lebih lanjut:

- 1. Analisis dinamis: gunakan proxy untuk meng-intercept traffic antara aplikasi dengan server sehingga dapat melakukan analisis lebih dalam.
- 2. Analisis statis: mempelajari bagaimana mengembangkan aplikasi Android yang aman sehingga dapat melakukan code review lebih baik.

# 8 Referensi

APK-Tool, http://code.google.com/p/androidapktool/

Jack Maninno, Reversing Android Apps 101,

mannino.blogspot.com/2010/09/reversingandroid-apps-101.html

terenkripsi saat mengakses server. Userid, Jeff Six, Application Security for the Android

Marko Gargenta, Learning Android, O'Reilly

**OWASP** Mobile Security Project, https://www.owasp.org/index.php/OWASP\_Mobi le Security Project

Situs Pengembang Android, http://developer.android.com

Situs Pengembang Twitter, https://dev.twitter.com