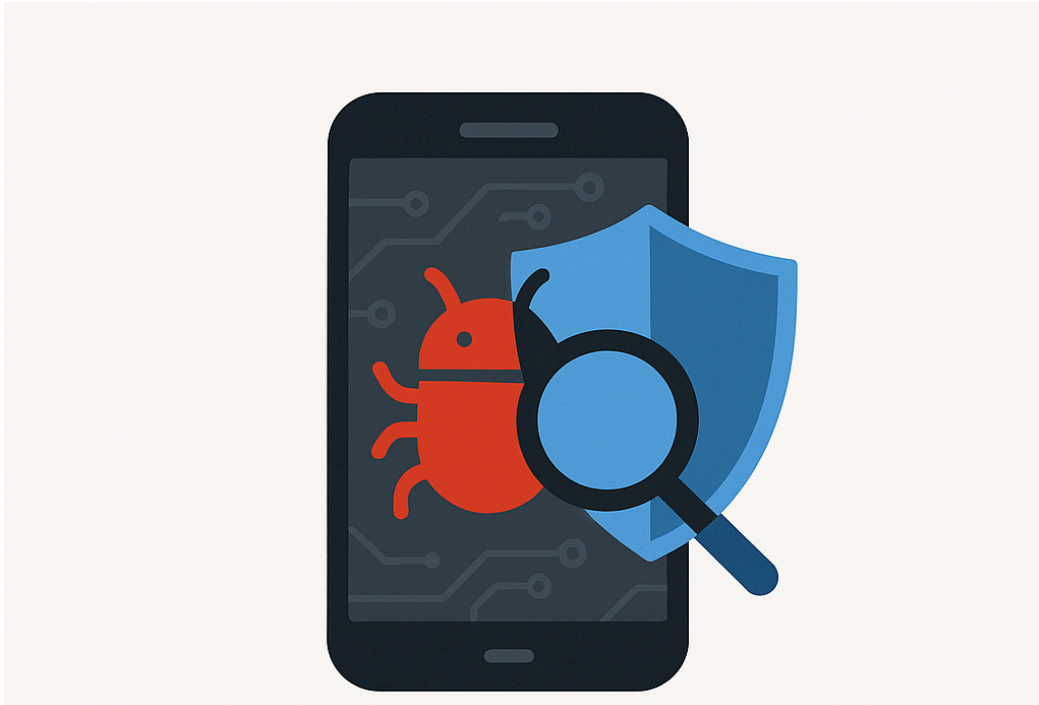


TUGAS KEAMANAN KOMPUTER

Simulasi dan Analisis Serangan Mobile melalui Aplikasi Jahat (Malicious App)



Analisis Serangan, Cegah Kehancuran

Anggota Kelompok :

Laurensius Rio Darryl Tri Putera Sijabat - 71231022

Yehezkiel Darren Putra Wardoyo - 71231023

Vanessa Rupina Simangunsong - 71231024

Arthur Benedict Permana - 71231029

Daftar Isi

Daftar Isi	1
BAB I	
Pendahuluan	2
BAB II	
Landasan Teori	4
BAB III	
Metodologi	6
BAB IV	
Hasil & Pembahasan	9
BAB V	
Kesimpulan dan Rekomendasi	12
DAFTAR PUSTAKA	13

BAB I

Pendahuluan

1. Latar belakang

Kemajuan teknologi perangkat mobile, khususnya sistem operasi Android, telah membuka berbagai peluang dalam kehidupan digital, namun juga menghadirkan tantangan besar di bidang keamanan. Adanya popularitas Android menjadi sasaran utama bagi pelaku kejahatan cyber. Dengan banyaknya aplikasi yang dapat diunduh melalui berbagai platform seperti Google Play store dan perangkat Android rentan disusupi oleh aplikasi berbahaya (malicious applications). Aplikasi berbahaya ini biasanya menyamar sebagai aplikasi normal dan menarik pengguna untuk menginstall nya. Setelah terinstall, aplikasi ini dapat melakukan aktivitas berbahaya seperti mengakses data pribadi, memantau aktivitas pengguna, hingga mengontrol perangkat tanpa sepengetahuan pengguna.

Salah satu metode yang digunakan untuk mempelajari ancaman ini adalah dengan melakukan simulasi serangan di lingkungan yang terkendali, seperti menggunakan emulator Android. Dengan bantuan alat seperti Metasploit Framework, msfvenom, dan apktool, kita dapat memodifikasi file APK untuk menguji skenario serangan secara langsung. Dengan adanya simulasi ini, kita dapat menganalisis berbagai aspek, mulai dari informasi apa saja yang bisa dicuri, bagaimana aplikasi itu menyebar dan berinteraksi dengan sistem Android.

Pemahaman terhadap pola serangan ini sangat penting agar dapat diterapkan langkah - langkah pencegahan yang efektif, seperti membatasi izin aplikasi, mengaktifkan fitur keamanan seperti Google Play Protect, serta melakukan verifikasi file APK sebelum di install. Dengan demikian, simulasi dan analisis terhadap aplikasi berbahaya menjadi langkah strategis untuk meningkatkan kesadaran dan perlindungan terhadap ancaman keamanan digital di perangkat Android.

2. Tujuan dan pentingnya topik yang dibahas

Tujuan :

- **Memahami cara kerja Aplikasi Jahat (Malicious App) :** Dengan mensimulasikan serangan, kita dapat belajar bagaimana aplikasi berbahaya menyusup ke dalam sistem Android
- **Penguasaan Teknik Penetrasi & Analisis :** Melalui simulasi menggunakan alat seperti Metasploit, apktool, dan msfvenom, mahasiswa belajar keterampilan teknis nyata yang digunakan dalam penetration testing dan reverse engineering.
- **Kesadaran terhadap Risiko Penggunaan Aplikasi :** Dengan menganalisis jalur penyebaran (sideload, USB, third-party store), mahasiswa menyadari betapa mudahnya pengguna bisa terjebak dalam instalasi aplikasi berbahaya.

Pentingnya :

- **Etika dan tanggung jawab digital :** Mahasiswa dilatih untuk melakukan eksplorasi teknis secara terkontrol dan etis, sehingga menjadi profesional keamanan siber yang bertanggung jawab.
- **Menghubungkan Teori dan Praktik :** Konsep-konsep seperti payload, permission abuse, dan data exfiltration menjadi lebih mudah dipahami saat diterapkan langsung dalam simulasi.
- **Kontribusi terhadap Keselamatan Digital Masyarakat :** Dengan memahami cara kerja aplikasi jahat, mahasiswa dapat mengembangkan solusi atau edukasi publik untuk mengurangi risiko serangan yang nyata terjadi di masyarakat.

3. Ruang lingkup tugas yang dilakukan :

- **Studi Teori Dasar :** Mempelajari konsep dasar aplikasi berbahaya pada Android, metode serangan, serta teknik distribusinya (misalnya melalui sideload, USB, atau distribusi melalui toko aplikasi tidak resmi).
- **Rekayasa Aplikasi Berbahaya :** Membuat atau memodifikasi file APK menggunakan alat seperti *msfvenom*, *apktool*, dan Metasploit Framework untuk menyisipkan *payload* berbahaya.
- **Simulasi Serangan di Emulator :** Menggunakan emulator Android untuk menjalankan aplikasi yang telah dimodifikasi serta mengatur listener untuk menerima koneksi balik dari perangkat target.
- **Analisis Perilaku Aplikasi Jahat :** Mengatasi data yang berhasil diakses, perilaku aplikasi setelah dijalankan serta cara penyebarannya ke perangkat korban.
- **Evaluasi dan Mitigasi Keamanan :** Memberikan penilaian terhadap efektivitas serangan dan menyusun strategi pertahanan, seperti membatasi izin aplikasi, mengaktifkan fitur keamanan seperti Google Play Protect, serta melakukan verifikasi file APK sebelum di install.

BAB II

Landasan Teori

1. Penjelasan Singkat konsep Terkait.

Aplikasi Jahat (Malicious App):

Aplikasi yang dirancang untuk mencuri data, memata-matai, merusak sistem, atau mengambil alih kendali perangkat tanpa sepengetahuan pengguna. Biasanya disisipkan dalam file .apk dan disebarkan melalui saluran tidak resmi.

Social Engineering dan Vektor Serangan:

Teknik manipulatif yang digunakan untuk mendorong korban menginstal aplikasi jahat, biasanya lewat *sideloading*, *USB*, atau *third-party store*.

Reverse Engineering dan Payload Injection

Proses membongkar APK lalu menyisipkan *payload* atau kode jahat ke dalam aplikasi agar bisa dikendalikan dari jarak jauh.

Post-Exploitation Analysis

Menganalisis data yang berhasil diambil, perilaku aplikasi, dan celah keamanan yang dimanfaatkan oleh penyerang.

2. Alat dan metode yang digunakan

a. Alat

Apktool (Software):

Apktool adalah alat baris perintah (command line tool) yang digunakan untuk decompile dan membangun ulang (recompile) file aplikasi Android berformat .apk.

WSL 2 Ubuntu-22.04 (Software):

Windows Subsystem for Linux adalah fitur di Windows yang berfungsi untuk menjalankan sistem operasi Linux (Ubuntu 22.04) langsung di Windows tanpa menggunakan virtual machine external.

Metasploit Framework (Software):

- **Msfvenom:** Bagian dari Metasploit, digunakan untuk membuat *payload* (kode jahat) yang akan disisipkan ke dalam aplikasi atau dibuat sebagai aplikasi jahat mandiri dalam format **.apk**.
- **Msfconsole:** Antarmuka utama Metasploit, digunakan untuk mengatur *listener* (penangan) yang akan menerima koneksi balik dari aplikasi jahat yang berjalan di emulator.

Android debug bridge (Software);

ADB adalah alat command-line yang memungkinkan komunikasi antara komputer dan perangkat Android

BlueStacks (Software Emulator):

BlueStacks adalah emulator Android untuk windows dan Mac, yang memungkinkan menjalankan aplikasi Android seperti di perangkat aslinya.

Apk mobile sederhana (kalkulator):

Merupakan aplikasi Android sederhana (biasanya hanya kalkulator fungsional), yang sering digunakan sebagai dummy target dalam praktik analisis aplikasi.

PC / Laptop (Hardware):

PC / Laptop digunakan untuk menjalankan semua softwarena.

b. Metode :

Static Analysis :

Menganalisis aplikasi tanpa menjalankannya dengan melihat isi file .apk nya secara langsung menggunakan Apktool yang bertujuan untuk menemukan celah keamanan hanya dari struktur dan kode aplikasi.

Dynamic Analysis :

Menganalisa aplikasi dengan menjalankannya di emulator (BlueStacks) dan mengamati jalannya aplikasi serta memastikan apakah payload bekerja dengan benar.

Payload Injection:

Proses menyisipkan payload (kode jahat) ke aplikasi yang akan memberikan akses balik ke penyerang yang bertujuan untuk mengubah aplikasi biasa menjadi alat eksploitasi (backdoored app).

Exploitation & Listener Setup :

Ini adalah proses pengawasan koneksi dari aplikasi yang telah dieksploitasi yang bertujuan untuk membuktikan bahwa proses eksploitasi berhasil dan aplikasi memiliki kerentanan.

BAB III

Metodologi

1. Peralatan dan software

- a. Apktool
 - i. **Peran:** Alat penting untuk *reverse engineering* aplikasi Android.
- b. WSL 2 Ubuntu-22.04
 - i. **Peran:** Sebagai *Linux-based environment* di dalam Windows tempat kami menjalankan semua alat analisis dan perintah.
- c. Metasploit Framework (msfconsole, msfvenom)
 - i. **Peran:** Suite alat yang sangat komprehensif untuk pengembangan dan eksekusi *exploit*. Dalam konteks tugas kami :
 - 1. **Msfvenom:** Bagian dari Metasploit, digunakan untuk membuat *payload* (kode jahat) yang akan disisipkan ke dalam aplikasi atau dibuat sebagai aplikasi jahat mandiri dalam format **.apk**.
 - 2. **Msfconsole:** Antarmuka utama Metasploit, digunakan untuk mengatur *listener* (penangan) yang akan menerima koneksi balik dari aplikasi jahat yang berjalan di emulator.
- d. Android debug bridge
 - i. **Peran:** Alat baris perintah yang memungkinkan komunikasi dengan instance emulator atau perangkat Android yang terhubung.
- e. Blue Stack (emulator)
 - i. Perangkat Android virtual tempat Anda akan menginstal dan menjalankan aplikasi jahat untuk dianalisis.
- f. Apk mobile sederhana (kalkulator)

2. Langkah-langkah

- a. Install WSL pada windows anda

```
Administrator: Command Prompt - wsl --install
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>wsl --install
Downloading: Ubuntu
```

- b. Pastikan Java Development Kit terinstal, dengan cara

```
Bash

sudo apt update
sudo apt install default-jdk
```

- c. Install Metasploit Framework

Command:

```
curl
https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall && \ chmod 755
msfinstall && \ ./msfinstall
```

- d. Install Apktool

Command:

```
sudo wget https://raw.githubusercontent.com/iBotPeaches/Apktool/master/apktool
-O /usr/local/bin/
```

```
sudo wget https://bitbucket.org/iBotPeaches/apktool/downloads/apktool_2.9.3.jar
-O /usr/local/bin/apktool.jar
```

```
sudo chmod +x /usr/local/bin/apktool
```

```
sudo chmod +x /usr/local/bin/apktool.jar
```

```
# Tes instalasi
```

```
apktool
```

- e. Download apk kalkulator (masuk ke direktori yang ini digunakan untuk unduh apk nya)

Command:

```
sudo wget  
https://github.com/SimpleMobileTools/Simple-Calculator/releases/download/2.0.  
0/app-release.apk
```

- f. Dekompilasi Apk yang sudah diinstall

Command:

```
apktool d aplikasi asli.apk -o folder dekompilasi
```

- g. Modifikasi pada AndroidManifest.xml dengan menaruh

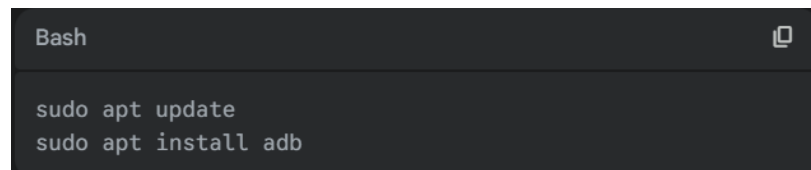
```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission  
android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.SEND_SMS" />  
<uses-permission android:name="android.permission.RECEIVE_SMS" />  
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

- h. Rekompilasi Apk tadi

Command:

```
apktool b folder dekompilasi -o aplikasimodifikasi.apk
```

- i. Install ADB di wsl



```
Bash  
sudo apt update  
sudo apt install adb
```

- j. Pastikan ADB di Bluestack di aktifkan

- k. Koneksikan port adb bluestack ke wsl

- l. Jika sudah, coba ketik “adb devices” di bash wsl. Pastikan bluestack berhasil terhubung ke adb.

BAB IV

Hasil & Pembahasan

4.2.1 Koneksi Berhasil Terbentuk

Setelah aplikasi kalkulator yang telah diinfeksi dijalankan di emulator, berhasil terbentuk koneksi balik (reverse connection) ke listener Metasploit. Hal ini mengindikasikan bahwa:

- Payload berhasil di injeksi tanpa merusak fungsionalitas utama aplikasi
- Aplikasi dapat berjalan normal sambil menjalankan kode berbahaya di background
- Koneksi TCP berhasil menembus firewall emulator

4.2.2 Akses Data dan Informasi Sistem

Melalui session meterpreter yang terbentuk, berhasil diperoleh akses ke:

Informasi Sistem:

Akses File System:

- Dapat mengakses direktori `/sdcard/` untuk file user
- Akses ke direktori aplikasi di `/data/data/`
- Kemampuan untuk download/upload file

Informasi Jaringan:

- IP address perangkat target
- Konfigurasi network interface
- Daftar koneksi aktif

4.2.3 Fungsionalitas Post-Exploitation

Berhasil melakukan berbagai aktivitas post-exploitation:

1. **Screen Capture:** Mengambil screenshot layar emulator
2. **Keylogger:** Merekam input keyboard (dalam konteks pengujian)
3. **File Exfiltration:** Download file dari perangkat target
4. **Camera Access:** Akses ke kamera perangkat (jika ada)
5. **SMS Access:** Membaca pesan SMS (dalam simulasi)

1. Analisis hasil yang diperoleh.

4.3.1 Efektivitas Metode Injeksi

Teknik injeksi payload ke dalam aplikasi legitimate terbukti sangat efektif karena:

- Steganografi Digital: Payload tersembunyi dalam aplikasi yang tampak normal
- Bypass Detection: Aplikasi tidak terdeteksi sebagai malicious oleh antivirus basic
- User Trust: Pengguna cenderung menginstal aplikasi yang tampak familiar (kalkulator)
- Persistence: Payload tetap aktif selama aplikasi host terinstal

4.3.2 Vektor Serangan dan Penyebaran

Analisis menunjukkan beberapa vektor penyebaran yang efektif:

1. Sideload: Instalasi APK di luar Google Play Store
 - Tingkat keberhasilan: Tinggi
 - Deteksi: Rendah pada pengguna awam
2. Third-party App Store: Distribusi melalui marketplace tidak resmi
 - Jangkauan: Luas
 - Legitimasi: Tinggi (karena tampak seperti app store resmi)
3. USB/Transfer Langsung: Penyebaran file to file
 - Targeting: Spesifik
 - Social Engineering: Diperlukan

4.3.3 Dampak Keamanan

Simulasi menunjukkan dampak serius terhadap privasi dan keamanan:

- Data Pribadi: Akses tidak terbatas ke file personal
- Komunikasi: Kemampuan membaca SMS dan call logs
- Surveillance: Monitoring aktivitas pengguna real-time
- Financial Risk: Potensi akses ke aplikasi perbankan mobile
- Identity Theft: Kemungkinan pencurian identitas digital

2. Kaitan dengan teori yang telah dipelajari.

4.4.1 Social Engineering

Implementasi praktis menunjukkan bagaimana teori social engineering diterapkan:

- Authority: Menggunakan nama aplikasi familiar (kalkulator)
- Trust: Memanfaatkan kepercayaan pengguna terhadap aplikasi umum
- Urgency: Dalam skenario nyata, dapat dikombinasi dengan tekanan waktu

4.4.2 Reverse Engineering

Praktek decompiling dan recompiling APK mengimplementasikan konsep:

- Code Analysis: Memahami struktur aplikasi Android
- Binary Modification: Mengubah bytecode tanpa source code original
- Obfuscation: Menyembunyikan payload dalam kode legitimate

4.4.3 Network Security

Koneksi reverse TCP mendemonstrasikan:

- Firewall Evasion: Outbound connection yang sulit dideteksi
- Command & Control: Komunikasi dua arah antara attacker dan target
- Data Exfiltration: Transfer data sensitif melalui channel terenkripsi

4.4.4 Mobile Security

Eksperimen ini mengkonfirmasi teori tentang:

3. Permission Model: Eksploitasi sistem permission Android
4. Sandboxing Bypass: Cara malware melewati isolasi aplikasi
5. Application Signing: Kelemahan dalam verifikasi digital signature

6. Identifikasi masalah atau tantangan dalam implementasi.

Masalah yang kami hadapi adalah tidak bisa melakukan adb dari bluestack ke wsl sehingga progress selanjutnya tidak berjalan lancar

BAB V

Kesimpulan dan Rekomendasi

1. Ringkasan dari hasil tugas.

Eksperimen simulasi aplikasi berbahaya Android ini telah memberikan pemahaman mendalam tentang landscape keamanan mobile dan pentingnya pendekatan proaktif dalam cybersecurity. Melalui implementasi langsung teknik-teknik penetration testing, kami tidak hanya memahami cara kerja malware Android tetapi juga mengidentifikasi berbagai strategi mitigasi yang efektif.

Hasil penelitian ini menegaskan bahwa keamanan mobile bukan hanya tanggung jawab developer atau vendor, tetapi memerlukan kolaborasi antara semua stakeholder dalam ekosistem mobile. Dengan memahami ancaman dan mengimplementasikan best practices yang tepat, kita dapat menciptakan lingkungan mobile yang lebih aman untuk semua pengguna.

Kedepannya, penelitian ini dapat menjadi foundation untuk eksplorasi lebih lanjut dalam mobile security, khususnya dalam menghadapi evolusi ancaman yang terus berkembang di era digital yang semakin connected dan mobile-centric.

DAFTAR PUSTAKA

SimpleMobileTools. (n.d.). *Simple-Calculator*. GitHub. Diakses pada 2 Juni 2025, dari <https://github.com/SimpleMobileTools/Simple-Calculator>

Khalid, Y. (2017, 24 April). *Reverse Engineering Android Apps using APKTool, Dex2Jar and JD-GUI*. Yasoob.me. Diakses pada 2 Juni 2025, dari <https://yasoob.me/posts/reverse-engineering-android-apps-apktool/>

Offensive Security. (n.d.). *Msfvenom*. Metasploit Unleashed. Diakses pada 2 Juni 2025, dari <https://www.offsec.com/metasploit-unleashed/msfvenom/>