Perbandingan Keamanan Aplikasi Pesan Instan Android Menggunakan MobSF (*Mobile Security Framework*) Berdasarkan Beberapa Standar

Aldi Cahya Fajar Nugraha[1)], Ray Novita Yasa[2)]
1) Badan Siber dan Sandi Negara, aldi.cahya@bssn.go.id
2) Politeknik Siber dan Sandi Negara, ray.novita@poltekssn.ac.id

Abstrak

Pesan instan berbasis Android secara real-time dengan menggunakan jaringan internet pada smartphone Android untuk berkomunikasi sudah menjadi hal yang lumrah. Hal ini harus dibarengi dengan peningkatan kesadaran akan keamanan informasi oleh pengguna pesan instan untuk meminimalkan potensi pelanggaran privasi dan peretasan akun. Saat ini, rendahnya tingkat kesadaran terhadap keamanan informasi warga dan minimnya informasi terkait tingkat keamanan aplikasi pesan instan berbasis Android di pasaran menjadi tantangan yang harus ditutupi oleh celah tersebut. Untuk menutup permasalahan tersebut, dilakukan analisis perbandingan keamanan aplikasi berbasis Android berdasarkan Standar Verifikasi Keamanan Aplikasi Mobile, NIST SP 800-163 Rev 1 dan Profil Perlindungan NIAP untuk Perangkat Lunak Aplikasi. Analisis yang dilakukan adalah analisis statis terhadap source code aplikasi pesan instan menggunakan MobSF (Mobile Security Framework) terhadap 11 aplikasi pesan instan berbasis Android yaitu Element, Line, Mesibo, Rocket.Chat, Signal, Skype, Snapchat, Speek!, Telegram, Viber dan WhatsApp. Hasil analisis menunjukkan Speek! mendapatkan skor keamanan tertinggi dibandingkan 10 aplikasi lainnya dengan sembilan temuan kerentanan pada Speek!, sedangkan aplikasi pesan instan paling tidak aman adalah Viber dengan 32 kerentanan..

Kata kunci: Aplikasi Pesan Instan Android, Keamanan Aplikasi Android, Mobile Security Framework, Mobile Application Security Verification Standard, NIST SP 800-163 Rev 1, NIAP Protection Profile for Application Software.

Abstract

Real-time Android-based instant messaging using the internet network on Android smartphones to communicate is becoming common. This case must be accompanied by increased awareness of information security by instant messaging users to minimize the potential for privacy violations and account hacking. Currently, the low level of awareness of citizen information security and lack of information related to the level of security of Android-based instant messaging applications on the market are challenges that loopholes must cover. To cover these problems, a comparative analysis of Android-based application security was carried out based on the Mobile Application Security Verification Standard, NIST SP 800-163 Rev 1 and NIAP Protection Profile for Application Software. The analysis is a static analysis of the source code of instant messaging applications using MobSF (Mobile Security Framework) against eleven Android-based instant messaging applications, namely Element, Line, Mesibo, Rocket.Chat, Signal, Skype, Snapchat, Speek!, Telegram, Viber and WhatsApp. The analysis results show Speek! get the highest security score compared to ten other applications with nine vulnerability findings on Speek!, while the least secure instant messaging app was Viber with 32 vulnerabilities.

Keywords: Android Instant Messaging Apps Security, Mobile Security Framework, Mobile Application Security Verification Standard, NIST SP 800-163 Rev 1, NIAP Protection Profile for Application Software.

1. PENDAHULUAN

Pada Januari 2022, terdapat 204,7 juta pengguna internet aktif di Indonesia, meningkat sekitar dua juta pengguna dari tahun sebelumnya [1]. Hal ini berdampak pada aktivitas komunikasi melalui aplikasi pesan instan. Aplikasi pesan instan merupakan aplikasi komunikasi *real-time* berbasis teks melalui perangkat digital antara dua pihak atau lebih. Aplikasi ini telah digunakan oleh berbagai pihak sebagai sarana pertukaran informasi umum dan rahasia. Aplikasi pesan instan umumnya menggunakan jaringan internet publik sebagai sarana layanan komunikasi utama. Kondisi ini berpotensi meningkatkan risiko kerahasiaan dan integritas data dalam proses komunikasi. Oleh karena itu, tingkat kesadaran pengguna terhadap keamanan informasi yang dimilikinya menjadi hal yang perlu diperhatikan.

Dalam dua tahun terakhir, peretasan aplikasi pesan instan masih sering terjadi. Pada bulan April 2021, Telegram disusupi oleh *ToxicEye RAT* (Random Access Trojan) yang mampu mengambil alih sistem, menginstal *Ransomware* hingga membocorkan data perangkat korban [2]. Pada bulan Mei tahun yang sama, akun *WhatsApp* dan *Telegram* milik mantan juru bicara Komisi Pemberantasan Korupsi (KPK) Febri Diansyah dan mantan penyidik KPK Novel Baswedan diretas pada waktu yang hampir bersamaan [3]. Selain itu, pada Agustus 2022, hampir 2000 akun pengguna *Signal* terkena dampak serangan *phishing Twilio*, yang mengungkap nomor telepon dan kode registrasi *Short Message Service* (SMS) korban. Namun, *Signal* mengklaim peristiwa ini tidak berdampak pada data pribadi penggunanya .

Pada September 2022, 24 anggota kru redaksi Narasi menjadi korban peretasan akun *WhatsApp* dan *Instagram* mereka [4]. Kasus-kasus tersebut didukung oleh laporan yang dikeluarkan oleh Zimperium mengenai tren keamanan aplikasi seluler pada tahun 2021. Berdasarkan kasus dan laporan yang telah diuraikan, dapat disimpulkan bahwa masih banyak kerentanan pada aplikasi Android yang dapat dieksploitasi untuk melanggar privasi pengguna.

Beberapa kasus tersebut sejalan dengan beberapa penelitian yang menjadikan aplikasi pesan instan Android sebagai objek penelitian, seperti penelitian Wirara, dkk [5] yang melakukan forensik digital pada *WhatsApp*, Syahib, dkk [6] yang melakukan analisis bukti pada aplikasi *Viber*, Nafila & Prayudi [7] pada aplikasi *Signal Messenger*, Yudhana, dkk [8] pada aplikasi *Facebook Messenger*, Aushaf, dkk [9] pada aplikasi *Telegram* dan Ridwan & Hidayanto [10] yang membandingkan hasil analisis bukti digital dari *Line*, *WhatsApp* dan *Telegram* pada sistem operasi Android, iOS dan Windows Phone. Analisis bukti digital mempunyai korelasi terhadap keamanan informasi, namun lebih ditujukan sebagai bukti sah yang akan diproses di pengadilan [5] sehingga tidak dapat memberikan informasi kepada pengguna terkait aplikasi pesan instan yang aman dari ancaman keamanan data dan privasi. Oleh karena itu, perlu adanya penelitian yang memberikan informasi kepada pengguna berupa perbandingan nilai keamanan beberapa aplikasi pesan instan berdasarkan hasil analisis.

Untuk menjawab keperluan perlu adanya penelitian yang memberikan informasi kepada pengguna berupa perbandingan nilai keamanan beberapa aplikasi pesan instan, penulis mengusulkan analisis perbandingan keamanan beberapa aplikasi pesan instan. Aplikasi pesan instan yang akan dianalisis merupakan aplikasi dengan fitur serupa. Aplikasi pesan instan yang akan dianalisis adalah *Element, Line, Mesibo, Rocket.Chat, Signal, Skype, Snapchat, Speek!, Telegram, Viber* dan *WhatsApp*. Aplikasi ini dinilai menggunakan alat MobSF yang kemudian dianalisis hasilnya menggunakan OWASP MASVS versi 1.4.2, NIST SP 800-163 Rev 1 dan NIAP *Protection Profile for Application Software versi* 1.3. Ketiga standar tersebut dipilih karena mencakup keamanan aplikasi seluler dan memiliki parameter kontrol dalam pengaturan keamanan.

2. LANDASAN TEORI

2.1 Aplikasi Pesan Instan

Aplikasi pesan instan adalah aplikasi yang menyediakan layanan pesan *real-time* antar perangkat digital melalui media internet. Aplikasi ini sudah lama menggantikan *Short Message Service* (SMS) sebagai sarana komunikasi teks jarak jauh karena lebih fleksibel dan berbiaya rendah. Fitur keamanan privasi pengguna merupakan fitur utama yang ditawarkan oleh berbagai pengembang aplikasi pesan instan [11]. Selain itu, fitur pada aplikasi pesan instan tidak hanya bisa mengirim pesan teks saja. Dengan aplikasi pesan instan pengguna dapat saling berbagi *file*, mengirim pesan suara, hingga melakukan panggilan suara dan video. Menurut penelitian yang dilakukan oleh Simanjuntak, dkk menunjukkan bahwa penggunaan aplikasi pesan instan mendapat respon positif dan sangat direkomendasikan untuk kegiatan yang melibatkan banyak peserta, seperti perkuliahan [12]. Dalam penelitian ini aplikasi pesan instan yang digunakan antara lain *Element, Line, Mesibo, Rocket.Chat, Signal, Skype, Snapchat, Speek!, Telegram, Viber dan WhatsApp*.

2.2 Kelemahan Pada Aplikasi Pesan Instan Berbasis Android

Aplikasi Android terus berkembang dari waktu ke waktu, termasuk aplikasi pesan instan. Pengembang aplikasi pesan instan bersaing untuk memberikan fitur-fitur terobosan agar aplikasi yang dibangun dapat digunakan oleh lebih banyak pengguna. Penambahan fitur pada suatu aplikasi sejalan dengan temuan kelemahan baru pada aplikasi tersebut. Terdapat beberapa kelemahan yang sering ditemukan di berbagai aplikasi Android termasuk di dalam aplikasi pesan instan [13], yakni sebagai berikut:

a. Pembangkitan Angka Acak yang Tidak Aman
   Dalam kriptografi, keacakan angka merupakan bagian penting dalam proses pembangkitan kunci. Tingkat keacakan tersebut sulit diimplementasikan pada algoritma deterministik. Maka dari itu, sebagian pengembang memanfaatkan *Psuedo Random Number Generator* (PRNG). Bagian ini berpotensi membuka celah keamanan jika pengembang aplikasi menggunakan pembangkit angka acak yang sederhana dan mudah diprediksi [13].

b. Penggunaan Algoritma MD5 dan SHA1
   MD5 dan SHA1 merupakan algoritma *hash* yang telah lama dipublikasikan. MD5 menghasilkan nilai *hash* berukuran 128-bit sedangkan SHA1 menghasilkan nilai *hash* berukuran 160-bit. Saat ini keduanya dinyatakan tidak aman karena MD5 telah terpecahkan pada tahun 2005 sedangkan SHA1 terpecahkan pada tahun 2017 [14].

c. Pengungkapan Alamat *Internet Protocol* (IP)
   Pengungkapan alamat IP pengguna dapat menyebabkan serangan lebih lanjut [13]. Salah satu serangan yang dapat dilakukan dengan memanfaatkan alamat IP korban yakni pemindaian *port* untuk mengetahui *port* yang terbuka dan dimanfaatkan untuk eksploitasi lanjutan.

d. *Hardcode* Berisi Data Sensitif
   Beberapa aplikasi menyimpan data sensitif dengan cara yang rentan terhadap serangan keamanan dengan menyimpan data tersebut secara langsung

di dalam kode. Data sensitif ini bisa berupa kunci enkripsi, token otentikasi, atau *Application Programming Interface* (API) yang penting untuk keamanan aplikasi. Selain itu, beberapa aplikasi juga menyimpan data sensitif dalam bentuk teks biasa, artinya data tersebut tidak dienkripsi atau diubah menjadi bentuk yang sulit dibaca [13].

e. Penyimpanan *File* Sementara
File sementara umumnya digunakan sangat singkat dan akan dihapus ketika selesai digunakan. Sering ditemukan jenis *file* ini tidak dienkripsi atau di-*encode* karena dianggap sebagai *file* yang tidak memerlukan perlindungan khusus. Hal ini berpotensi menimbulkan celah keamanan dalam konteks kerahasiaan data [13].

f. Implementasi *Webview* yang Tidak Aman
WebView berpotensi membuka beberapa celah keamanan yang mungkin terjadi jika konten dimuat dalam bentuk teks biasa. Hal tersebut menyebabkan data sensitif yang dimuat dapat dengan mudah dibaca oleh penyerang [13].

g. Penggunaan Algoritma Enkripsi yang Lemah
Algoritma enkripsi DES dan AES merupakan algoritma *block cipher*, yang secara *default* menggunakan mode *Electronic Code Book* (ECB). Dalam mode ECB, input dibagi menjadi blok-blok dengan panjang tertentu. Setiap blok dienkripsi secara terpisah dan tidak saling terkait. Mode ECB terbukti tidak aman, karena ketika dua blok berisi pesan identik, maka hasil enkripsinya juga akan identik [15].

h. Penggunaan Sertifikat yang Tidak Aman
Sertifikat elektronik berisi kunci publik *server* yang digunakan untuk memastikan keaslian *server*. Selain itu, sertifikat elektronik juga melakukan verifikasi bahwa data yang dikirimkan antara *server* dan pengguna tidak dapat diubah atau diakses oleh pihak yang tidak sah. Kunci publik dari pihak ketiga yang terpercaya digunakan untuk mengautentikasi sertifikat *server*, sehingga pengguna dapat memastikan bahwa mereka berkomunikasi dengan *server* yang sah dan dapat dipercaya [13]. Jika sertifikat yang digunakan tidak dipercaya oleh pihak ketiga (*Certificate Authority*) maka akan membuka potensi *man-in-the-middle attack* [16].

i. Permintaan *Privilege* Berlebih
Over privilege merupakan kondisi dimana aplikasi mendapatkan hak akses yang melebihi fungsi utamanya. Sebagian besar aplikasi Android mendapatkan *over privilege* karena tidak terdapat spesifikasi pengembangan yang ketat, pengawasan yang efektif dan rendahnya pengetahuan pengguna terkait tujuan dari izin yang diajukan oleh aplikasi [17].

2.3 OWASP *Mobile Application Security Verification Standard*

OWASP *Mobile Application Security Verification Standard* (MASVS) pertama kali dirilis pada awal 2018. Dokumen ini ditujukan sebagai pedoman bagi arsitek dan pengembang yang ingin mengembangkan aplikasi *mobile* yang aman, maupun penguji keamanan untuk memastikan kelengkapan dan konsistensi hasil pengujian. OWASP MASVS masih dikembangkan sesuai dengan kondisi terkini untuk menjaga relevansi terhadap isu-isu keamanan aplikasi *mobile* yang berkembang. Pada penelitian ini, standar OWASP MASVS yang digunakan adalah versi 1.4.2.

2.4 NIAP *Protection Profile for Application Software*

National Information Assurance Partnership (NIAP) merupakan organisasi yang mengawasi proses evaluasi produk-produk teknologi informasi agar sesuai dengan ketentuan *Common Criteria International* [18]. NIAP menerbitkan dokumen *Protection Profile for Application Software* sebagai media untuk mendeskripsikan fungsionalitas keamanan aplikasi sesuai ketentuan *Common Criteria* serta menentukan syarat fungsional dan jaminan terhadap aplikasi tersebut. Perubahan target utama pada kasus serangan terhadap perangkat lunak juga mendasari diterbitkannya dokumen ini. Pada penelitian ini standar NIAP *Protection Profile for Application Software* yang digunakan adalah versi 1.3.

2.5 NIST SP 800-163 Rev 1

NIST SP (*Special Publication*) berisi rekomendasi dan *best practice* dalam hal keamanan informasi. Lembaga-lembaga federal di Amerika Serikat wajib mematuhi setiap standar NIST SP yang telah diuraikan dalam FIPS (*Federal Information Processing Standard*). Standar NIST dalam bidang keamanan siber dikelompokkan dengan kode NIST SP seri 800. NIST SP 800 berisi rekomendasi untuk keamanan informasi yang meliputi *framework* manajemen risiko, persyaratan keamanan, dan kontrol keamanan [19].

NIST SP 800-163 Rev 1 merupakan pembaruan dokumen panduan NIST dalam hal pemeriksaan dan keamanan aplikasi *mobile*. Dokumen asli NIST SP 800-163 menjelaskan tentang proses organisasi dalam mengevaluasi aplikasi *mobile* terhadap kerentanan di bidang keamanan siber, sedangkan versi Rev 1 mencakup perluasan ruang lingkup dari dokumen asli dengan mengeksplorasi sumber daya yang dapat digunakan untuk menginformasikan persyaratan keamanan informasi pada aplikasi *mobile*. Versi ini juga menjelaskan dan menyempurnakan model pemeriksaan pada dokumen asli dengan mendefinisikan peran dan proses yang mempengaruhi pemeriksaan.

2.6 *Mobile Security Framework* (MobSF)

MobSF merupakan *tools* automasi untuk

melakukan pengujian keamanan aplikasi secara statis dan dinamis berupa *penetration testing*, analisis malware, dan penyedia *framework* keamanan terhadap aplikasi *mobile* pada berbagai sistem operasi [20]. *Tools* ini tersedia dalam dua tipe, yakni tipe live dan localhost. Perbedaan keduanya yakni tipe live diakses melalui https:///mobsf.live/ dan dapat langsung melakukan pengujian aplikasi, sedangkan tipe *localhost* perlu melakukan konfigurasi *server* MobSF pada komputer sebelum melakukan pengujian [21]. Pada penelitian ini, MobSF yang digunakan adalah versi *localhost* agar hasil uji keamanan terhadap aplikasi tersimpan pada penyimpanan lokal.

3. METODE PENELITIAN

Pada penelitian ini yang menjadi objek penelitian adalah keamanan beberapa aplikasi pesan instan berbasis Android yaitu aplikasi *Element, Line, Mesibo, Rocket.Chat, Signal, Skype, Snapchat, Speek!, Telegram, Viber* dan *WhatsApp*. Kode sumber aplikasi pesan instan pada Tabel 1 diuji secara statis dengan memanfaatkan alat *Mobile Security Framework* (MobSF) dan standar keamanan aplikasi seluler OWASP MASVS, NIST SP 800-163 Rev 1, dan NIAP *Protection Profile for Application Software*. Penelitian ini merupakan penelitian kuantitatif yang akan membandingkan tingkat keamanan aplikasi berdasarkan skor aplikasi pada tiap standar.

Tahapan penelitian beserta metode atau prosedur yang dilakukan, yaitu:

a. Persiapan Sistem
   Pada tahap ini, peneliti melakukan persiapan terhadap sistem *Mobile Security Framework* (MobSF) yang akan digunakan sebagai *tools* untuk pengujian statis terhadap kode sumber aplikasi. MobSF versi localhost merupakan aplikasi berbasis *web*, sehingga memerlukan *web server* untuk dapat beroperasi. *Tools* ini di-*install* pada komputer dengan sistem operasi Windows 10 Pro 64-bit, prosesor Intel Core i7 generasi ke-10 dan kapasitas RAM sebesar 16GB. Setelah proses instalasi selesai, MobSF dapat diakses melalui *browser* dengan alamat http://localhost:8000.

b. Analisis Aplikasi
   Pada tahap ini, aplikasi pesan instan yang akan dianalisis telah disiapkan. Aplikasi pesan instan diunggah ke sistem MobSF localhost untuk melakukan analisis statis. Proses analisis terhadap seluruh aplikasi tidak dapat dilakukan dalam satu waktu yang sama, sehingga aplikasi selanjutnya dapat dianalisis ketika sistem telah selesai menganalisis aplikasi sebelumnya. Pemindaian satu aplikasi dilakukan secara paralel karena standar yang digunakan tidak saling terkait. Hal tersebut dibuktikan oleh perbedaan kategori keamanan aplikasi pada masing-masing standar. Hasil pemindaian MobSF berisi deskripsi singkat terkait kesesuaian keamanan kode sumber aplikasi terhadap OWASP MASVS versi 1.4.2, NIST SP 800-163 Rev 1 dan NIAP *Protection Profile for Application Software* versi 1.3. hal ini karena MobSF sudah sesuai dengan standar ketiga standar yang menjadi acuan.

c. Laporan Hasil Analisis
   Pada tahap ini penulis menarik kesimpulan atas hasil analisis setiap aplikasi pesan instan berbasis Android terhadap parameter keamanan OWASP MASVS versi 1.4.2, NIST SP 800-163 Rev 1 dan NIAP *Protection Profile for Application Software* versi 1.3. Kesimpulan tersebut disusun ke dalam laporan berbentuk deskripsi naratif yang menjelaskan parameter apa saja yang dipenuhi dan tidak dipenuhi oleh suatu aplikasi pesan instan yang disertai bukti analisis. Seluruh aplikasi pesan instan yang dianalisis dibandingkan berdasarkan tingkat keamanannya dalam bentuk tabel.

4. HASIL DAN PEMBAHASAN

Analisis dilakukan terhadap 11 aplikasi pesan instan menggunakan bantuan *tools* MobSF. *Tools* ini akan melakukan penilaian terhadap kode sumber setiap aplikasi. Hasil dari MobSF kemudian dikelompokkan berdasarkan poin-poin standar pada OWASP MASVS, NIAP v1.3, dan NIST SP 800-163 Rev 1. Perbandingan penilaian keamanan aplikasi pesan instan menurut OWASP MASVS disajikan pada Tabel 1. Perbandingan penilaian aplikasi pesan instan menurut NIAP v1.3 disajikan pada Tabel 2. Perbandingan penilaian aplikasi pesan instan menurut NIST SP 800-163 Rev 1 disajikan pada Tabel 3. Simbol silang (×) menandakan bahwa aplikasi memiliki kerentanan pada kategori tersebut.

Tabel 1. Hasil penilaian keamanan berdasarkan OWASP MASVS

No	Aplikasi	MSTG-CODE-2	MSTG-CRYPTO-2	MSTG-CRYPTO-3	MSTG-CRYPTO-4	MSTG-CRYPTO-6	MSTG-NETWORK-3	MSTG-PLATFORM-2	MSTG-PLATFORM-7	MSTG-RESILIENCE-1	MSTG-RESILIENCE-2	MSTG-STORAGE-2	MSTG-STORAGE-14
1	Element	-	-	-	×	×	×	-	-	×	-	×	×
2	Line	×	×	×	×	×	-	-	×	-	-	×	×
3	Mesibo	-	-	-	×	-	×	-	-	-	-	×	×
4	Rocket.Chat	-	-	-	×	-	-	-	×	-	-	×	×
5	Signal	×	-	×	×	×	-	-	-	-	-	×	×
6	Skype	×	-	×	×	×	-	-	-	-	-	×	×
7	Snapchat	×	-	×	×	×	-	×	-	-	×	×	×
8	Speek!	-	-	-	-	-	-	-	-	-	-	×	×
9	Telegram	×	-		×	×	-	-	×	-	×	×	×
10	Viber	×	×	×	×	×	×	-	×	-	×	×	×
11	WhatsApp	×	-	×	×	×	×	-	×	-	-	×	×

Tabel 2. Hasil penilaian keamanan berdasarkan NIAP v1.3

No	Aplikasi	FCS_CKM.1.1(1)	FCS_COP.1.1(1)	FCS_COP.1.1(2)
1	Element	×	×	×
2	Line	×	×	×
3	Mesibo	-	-	×
4	Rocket.Chat	×	×	×
5	Signal	×	×	×
6	Skype	-	×	×
7	Snapchat	-	×	×
8	Speek!	-	-	-
9	Telegram	×	-	×
10	Viber	×	-	×
11	WhatsApp	×	×	×

Tabel 3. Hasil Penilaian Keamanan Berdasarkan NIST SP 800-163 Rev 1

No	Aplikasi	*Incorrect Permission*	*Exposed Communication*	*Exposed Data Storage*	*Traditional Software Vulnerabilities*
1	Element	11	1	2	1
2	Line	10	1	2	1
3	Mesibo	8	1	2	1
4	Rocket.Chat	4	0	2	1
5	Signal	15	0	2	2
6	Skype	10	0	2	2
7	Snapchat	10	0	2	2
8	Speek!	5	0	2	0
9	Telegram	14	0	2	1
10	Viber	15	1	2	2
11	WhatsApp	14	0	2	2

Terdapat perbedaan pada jumlah temuan kerentanan menurut OWASP MASVS dan NIAP v1.3 dengan NIST SP 800-163 Rev 1. Hal ini disebabkan oleh perbedaan kategori yang diberikan oleh setiap standar. Kategori keamanan pada OWASP MASVS dan NIAP v1.3 hanya bernilai "terpenuhi" atau "tidak terpenuhi". Sedangkan kategori keamanan pada NIST SP 800- 163 Rev 1 tidak bersifat "terpenuhi" atau "tidak terpenuhi". Kategori keamanan pada NIST SP 800-163 Rev 1 berisi jumlah temuan kerentanan yang sesuai dengan kategori tersebut.

Secara keseluruhan, *Speek!* lebih aman dibandingkan 10 aplikasi lainnya. Hal ini dapat dilihat bahwa jumlah temuan kerentanan pada *Speek!* paling sedikit dibandingkan dengan aplikasi lainnya, yakni sembilan temuan kerentanan. Penilaian menggunakan MobSF juga menunjukkan bahwa *Speek!* merupakan aplikasi berkategori "*Low Risk*" dengan nilai 73/100. Pada urutan kedua terdapat *Rocket.Chat* dengan jumlah 14 temuan kerentanan. *Mesibo* menempati urutan ketiga dengan 17 temuan kerentanan. Urutan selanjutnya terdapat *Element* dan *Snapchat* yang masing-masing memiliki 24 temuan kerentanan. Kemudian terdapat *Line* dengan 25 temuan kerentanan, Telegram dengan 26 kerentanan, *Signal* dengan 28 kerentanan dan *WhatsApp* dengan 29 kerentanan. Posisi terakhir terdapat *Viber* dengan jumlah 32 kerentanan dan menjadi yang paling tidak aman dari seluruh aplikasi yang dianalisis.

5. KESIMPULAN

Speek! dinilai sebagai aplikasi pesan instan berbasis Android yang aman dibandingkan 10 aplikasi lainnya. Jumlah kerentanan yang ditemukan pada *Speek!* paling sedikit jika dibandingkan dengan 10 aplikasi lainnya. *Rocket.Chat* menempati posisi kedua sebagai aplikasi pesan instan yang aman menurut OWASP MASVS, NIAP v1.3 dan NIST SP 800-163 Rev 1 dengan jumlah kerentanan yang ditemukan sebanyak 14 kerentanan. Kemudian terdapat *Meisbo* dengan 17 kerentanan, *Element* dan *Snapchat* dengan 24 kerentanan, *Line* dengan 25 kerentanan, *Telegram* dengan 26 kerentanan, *Signal* dengan 28 kerentanan dan *WhatsApp* dengan 29 kerentanan. Pada urutan terakhir terdapat *Viber* dengan 32 kerentanan dan menjadi aplikasi yang paling tidak aman dari 11 aplikasi yang dianalisis.

REFERENSI

[1] C. M. Annur, "Ada 204,7 Juta Pengguna Internet di Indonesia Awal 2022," Katadata, 23 Maret 2022. [Online]. Available: https://databoks.katadata.co.id/datapublish/2022/03/23/ada-2047-juta-pengguna-internet-di-indonesia-awal-2022. [Accessed 23 Oktober 2022].

[2] E. Montalbano, "Telegram Platform Abused in ToxicEye Malware Campaigns Threatpost," Threatpost, 22 April 2021. [Online]. Available: https://threatpost.com/telegram-toxiceye-malware/165543/. [Accessed 23 Oktober 2022].

[3] A. Suminar, "Whatsapp dan Telegram Aktivis Antikorupsi Diretas, Termasuk Novel dan Febri - Suara Surabaya," suarasurabaya.net, 21 Mei 2021. [Online]. Available: https://www.suarasurabaya.net/kelanakota/2021/whatsapp-dan-telegram-aktivis-antikorupsi-diretas-termasuk-novel-dan-febri/. [Accessed 23 Oktober 2022].

[4] A. Ramadhan, S. Asril, " Akun Media Sosial Milik 24 Karyawan Narasi Diretas" Kompas.com, 26 September 2022. [Online]. Available: https://nasional.kompas.com/read/2022/09/26/15282151/akun-media-sosial-milik-24-karyawan-narasi-diretas. [Accessed 23 Oktober 2022].

[5] A. Wirara, B. Hardiawan and M. Salman, "Identifikasi Bukti Digital pada Akuisisi Perangkat *Mobile* dari Aplikasi Pesan Instan "WhatsApp"," Teknoin, vol. 26, no. 1, pp. 66-74, 2020A. Amiruddin, AAP Ratna, R Harwahyu, RF Sari, Secure multi-protocol gateway for Internet of Things, 2018 Wireless Telecommunications Symposium (WTS), 1-8, 2018.

[6] M. I. Syahib, I. Riadi and R. Umar, "Analisis Bukti Digital Aplikasi Viber Menggunakan Metode National Institute of Standards Technology (NIST)," Jurnal Sains Komputer & Informatika (J-SAKTI), vol. 4, no. 1, pp. 170-178, 2020.

[7] F. L. Nafila and Y. Prayudi, "Analisis Digital Artifak Aplikasi Signal Messenger Pada Sistem Operasi Android dengan Metode NIST," Jurnal Sains Komputer & Informatika (J-SAKTI), vol. 6, no. 1, pp. 532-543, 2022.

[8] A. Yudhana, I. Riadi and I. Anshori, "Analisis Bukti Digital Facebook Messenger Menggunakan Metode NIST," IT Journal Research and Development, vol. 3, no. 1, pp. 13-21, 2018.

[9] R. F. Aushaf, S. J. I. Ismail and G. B. Satyra, "Implementasi Forensik Digital di Telegram pada Sistem Operasi Android," e-Proceeding for Applied Science, vol. 7, no. 6, pp. 2767-2778, 2021.

[10] K. Ridwan and B. C. Hidayanto, "Analisis Keamanan Aplikasi Pesan Instan LINE, WhatsApp dan Telegram dalam Sistem Android, iOS dan Windows Phone," in Tugas Akhir Fakultas Teknolgi Informasi dan Komunikasi - Institut Teknologi Sepuluh Nopember, Surabaya, 2018.

[11] M. S. Asyaky, N. Wdiyasono and R. Gunawan, "Analisis dan Perbandingan Bukti Digital Aplikasi Instant Messenger Pada Android," SinkrOn - Jurnal & Penelitian Teknik Informatika, vol. 3, no. 1, pp. 220-231, 2018.

[12] M. B. Simanjuntak, N. Lustyantie and I. Iskandar, "Pembelajaran Berbasis Telegram Group dan Microsoft Team di Kelas Bahasa Inggris (Penilaian berbasis Persepsi Siswa)", Jurnal Pendidikan Tambusai, vol. 6, no. 2, pp. 11114-11119, 2022.

[13] A. Yang , "Cryptocurrency Security Study based on Static Taint Analysis," Highlights in Science, Engineering and Technology, vol. 39, pp. 962-970, 2023.

[14] Z. A. Al-Odat and S. U. Khan, "The Sponge Structure Modulation Application to Overcome the Security Breaches for the MD5 and SHA-1 *Hash* Functions," in 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, 2019.

[15] M. Tyagi, M. Manoria and B. Mishra, "Analysis and Implementation of AES and RSA for Cloud," International Journal of Applied Engineering Research, vol. 14, no. 20, pp. 3918-3923, 2019.

[16] Common Weakness Enumeration, "CWE-295: Improper Certificate Validation (4.12)", The MITRE Corporation, 29 June 2023. [Online]. Available: https://cwe.mitre.org/data/definitions/295.html. [Accessed 23 July 2023].

[17] S. Wu and J. Liu, "Overprivileged Permission Detection for Android Applications," in 2019 IEEE International Conference on Communications (ICC), Shanghai, 2019.

[18] National Information Assurance Partnership, "NIAP: What is NIAP/CCEVS?," National Information Assurance Partnership, [Online]. Available: https://www.niapccevs.org/Ref/What_is_NIAP.CCEVS.cfm. [Accessed 18 December 2022].

[19] M. Ogata, J. Franklin, J. Voas, V. Sritapan and S. Quirolgico, Vetting the Security of *Mobile* Applications, Gaithersburg, MD: National Institute of Standards and Technology, 2019.

[20] C. Hanifurohman and D. D. Hutagalung, "Analisis Statis Menggunakan *Mobile* Security Framework Untuk Pengujian Keamanan Aplikasi *Mobile* E-Commerce Berbasis Android," Sebatik, vol. 24, no. 1, pp. 22-28, 2020.

[21] A. R. Tambunan, T. Yuniati and Y. A. Setyoko, "Implementasi Static Analysis Dan Background Process Untuk Mendeteksi Malware Pada Aplikasi Android Dengan *Mobile* Security Framework," LEDGER: Journal Informatic and Information Technology, vol. 1, no. 2, pp. 60-74, 2022.