REPOSITOR, Vol. 2, No. 10, Oktober 2020, Pp. 1331-1337
ISSN : 2714-7975
E-ISSN : 2716-1382

Analisis Malware Android Menggunakan Metode Reverse Engineering

Bagus Aji Saputro[1], Lisan Iqbal Alfitra[2], Raykhan Bima Oktaviaji[3]
[1]Universitas Amikom Yogyakarta, Universitas Amikom Yogyakarta [2,]
Universitas Amikom Yogyakarta[3]
e-mail: bagus.11@students.amikom.ac.id[1], lisan.27@students.amikom.ac.id [2],
raykhan.1999@students.amikom.ac.id[3]

Abstrak

Beberapa ancaman yang ada pada system operasi pada perangkat lunak android yang mengancam pengguna dari berbagai aspek yang dapat merusak bagian-bagian pada system operasi yang berhubungan dengan perangkat lunak jahat (malware) dengan menggunakan user untuk menggunakan system operasi android target untuk melakukan dengan serangan malware. Dengan melakukan perubahan kode aplikasi atau paket paket dengan memasukan kode yang berbahaya, dan mengemas ulang kode tersebut dengan script yang seperti yang asli dan mengupload aplikasi yang dibuat ke situs internet ataupun Play Store Android. Penelitian ini bertujuan untuk melihat kode pada sebuah aplikasi yang diduga tersisipi malware dengan metode reverse engineering, agar setiap pengguna untuk dapat sadar untuk dampak apa yang terjadi jika malware yang ada pada aplikasi yang ada di android dengan mengetahui isi code pada percobaan malware dengan menggunakan Analisa static. Android, sebagai system operasi berbasis Java yang jalan pada *kernel* 2.6 *Linux*. Aplikasi Sistem Android dikembangkan dengan Java mudah menyelelaraskan pada *platform* baru. Sistem Android dapat digambarkan seperti jembatan antara *smartphone* dan *user*, sehingga *user* dapat menggunakan *smartphone* dan bisa menjalankan aplikasi yang ada pada *smartphone* tersebut. Android merupakan sebuah sistem operasi dengan dasar linux pada *smartphone* yang meliputi sistem operasi, *middle ware* dan aplikasi yang terdapat didalamnya

Kata kunci: malware, reverse engineering, denteksi malware, android

Abstract

Some of the defenses that exist in the operating system on Android software that opposes users from various aspects that can damage parts of the operating system associated with malicious software (malware) by using users to use the target Android operating system to do with malware attacks. By changing the application code or package package by entering the dangerous code, and repacking the code with a script that looks like the original and uploading the application made to the internet site or the Android Play Store. This research tries to look at the code in applications that challenge malware with the reverse engineering method, so that each user can understand what is happening with malware in an application that is on Android by finding the content code in a malware experiment using static analysis. Android, as a Java-based operating system that runs on the Linux 2.6 kernel. The Android System application developed with Java easily synchronizes on a new platform. The Android system can create a bridge between the user and the smartphone, so that the user can use a smartphone and can install applications on the smartphone. Android is an operating system based on Linux on a smartphone that includes the operating system, middle device and the applications that are in it

Keywords*:* malware, reverse engineering, malware detection, android

1. Pendahuluan

Malware yaitu perangkat lunak berbahaya yang disebut juga Malicious Software merupakan salah satu serangan yang melibatkan serangan terhadap system[1]. Tujuan utama dari papaer ini adalah untuk mengidentifikasi aplikasi code4hk berbahaya atau tidak. Jadi untuk paper ini membahas sistem malware berdasarkan analisis penggunaan izin. Dalam metode ini, penulis

mengidentifikasi izin paling signifikan yang efektif dalam membedakan antara aplikasi berbahaya atau tidak. Sebagai hasil peneliti menggunakan metode reverse engineering untuk mendeteksi malware Android berdasarkan penggunaan izin[2]. Dengan beragam jenis malware berdasarkan tingkat serangan dan kerusakan yang di dapat oleh *smartphone* untuk melakukan tindakan pencurian data pribadi. Ada beberapa kesimpulan tentang malware "Malicious Software" perangkat lunak mencurigakan yang di isipkan di system computer[3].

a. Malware merupakan media yang merusak celah keamanan dengan menggunakan perangkat lunak.
b. Malware merupakan software atau script yang diolah pada system dengan tidak tau pemilik atau user dan bekerja di latar belakang.
c. Malware merupakan media software yang berbahaya bagi pengguna perangkat lunak atau software, akses jarinagn yang berada pada computer [4].

Sebagai ponsel yang terus-menerus melintasi domain jaringan, mereka lebih banyak terkena malware daripada smartphone tradisional yang jarang menggunakan internet. Misalnya dengan memanfaatkan sepenuhnya aktivitas korban, malware ponsel mampu menyebar di seluruh domain jaringan dengan lebih mudah. Selain itu dengan lebih dari satu juta aplikasi yang tersedia dan hampir instalasi instan[5].

Malware platform Android menyamar melalui layanan Google Play Store maupun hasil download pada insternet, dengan merubah menjadi aplikasi yang sah seperti aplikasi galeri dan permainan. Beberapa malware yang terdapat di Play Store contohnya adalah *Banking Trojan*, *Adware, RAT* dan *Cryptocurrency-mining malware*. *Adware* merupakan malware yang sangat banyak menyusup pengguna Play Store dan terus melalui peningkatan sampai 36% sejak 2016. Selain itu, malware Banking Trojan juga naik hingga 12% sedang *Crypto-Mining* meningkat sebesar *5%* bersama melambungnya nilai Bitcoin[6].

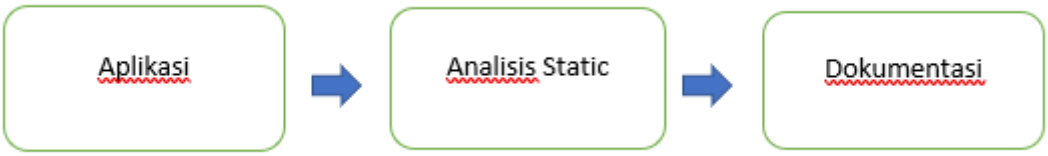
Praktisi keamanan informasi perlu melaksanakan proses investigasi forensik pada analisis malware untuk mengidentifikasi, mengamankan, mengamati, dan menyajikan fakta dan opini dari informasi malware. Beberapa metode dan teknik dalam melakukan analisis malware yaitu analisa secara statis (*Static Analysis*) dan analisa secara dinamis (*Dynamic Analysis*), dimana keduanya memiliki kelemahan dan kelebihan masing[7].

Analisis Statis dilaksanakan dengan melihat *source code* pada *malware* untuk dipelajari dan dipahami melalui *code* tersebut atau dengan kata lain proses analisa tidak perlu melakukan eksekusi terhadap *malware* [8]. Analisis statis dan pelabelan aplikasi tidak bisa mengatasi seringnya penyebaran malware. Selain itu, beberapa metode deteksi (mis., Berbasis *machine learning*), memerlukan sejumlah data yang besar untuk pelatihan, yang menghasilkan analisis statis dan pelabelan infeksi *malware*. Demikian para peneliti beralih ke platform online, seperti Virus Total, yang memberikan hasil pemindaian dari perangkat lunak antivirus komersial yang lengkap[9].

Pengembang Android menyematkan izin dalam aplikasi yang disimpan pada file Androidmanifest.xml untuk mengakses sms, kamera, panggilan, dan lokasi pengguna secara spesifik. Saat memasang aplikasi android, pengguna harus memberi izin untuk memasan aplikasi. Android meminta izin untuk kinerja yang dibutuhkan dan berkomunikasi dengan aplikasi atau server yang memungkinkan untuk melakukan serangan pencurian data tanpa sepengetahuan pengguna[10]. Penelitian ini mencoba menganalisis *permission spyware Code4hk* yaitu RAT yang beredar di Play Store, dan internet, yaitu salah satu *Remote Access Trojan* yang mencuri informasi pengguna.

2. Metode Penelitian

Metodologi yang digunakan pada penelitian ini adalah metode reverse engineering yaitu mengekstrak aplikasi untuk mengetahui source code yang ada didalam aplikasi untuk mengethaui *permission* apa saja yang dibutuhkan oleh aplikasi tersebut dan mungkin jika aplikasi tersebut terhubung pada sebuah server/*IP Address* yang mencurigakan.



Gambar 1 Alur penelitian

3. Hasil Penelitian dan Pembahasan

Malware yang akan digunakan sebagai objek penelitian yakni *Code4HK, Informasi yang diperoleh dari virus total adalah sebagai berikut***.**

Tabel 1 Contoh keterangan tabel

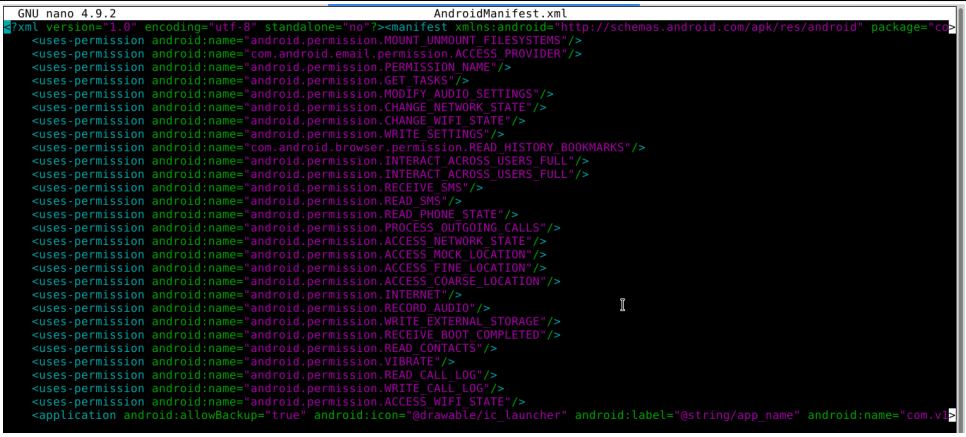
Nama Malware	**code4hk.apk**
SHA256	fe1df17ab903979223e5eb514f fe24f72d540ad26f959201133f 30a1346870df
Ukuran file	*400kb*
Tipe file	*APK*
Rasio Deteksi	*31/63*

3.1. Ananlisis Static Permission Malware

Analisis statis yang dilaksanakan pada penelitian ini memakai teknik *reverse engineering* untuk mendapat Java *source code* dari aplikasi yang telah terinfeksi oleh *malware* tadi, lalu kita melakukan analisis untuk mendapat kode yang bersifat merusak atau mencuri data pengguna. Hal yang pertama dilakukan pada analisis statis sampel *malware* adalah mengesktrak file **.apk* menggunakan tool apktool denagn perintah apktool -d code4hk.apk pada terminal linux. Konten yang diekstrak memiliki struktur seperti pada gambar 2. Analisis khususnya pada file *AndroidManifest.xml* yang memuat izin apa saja yang diberikan oleh sistem terhadap aplikasi dan *classes.dex* yang merupakan *source code* dari aplikasi yang telah diubah ke dalam *Dalvik Bytecode*.

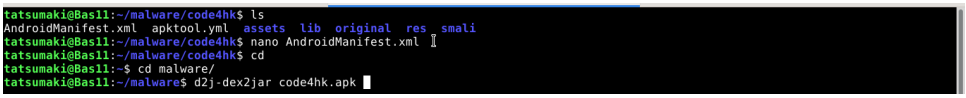
Gambar 2. Isi pada aplikasi Code4hk.apk

File *AndroidManifest.xml* memiliki informasi yang dibutuhkan oleh Android, seperti *metadata*, dan perizinan yang dibutuhkan aplikasi ketika proses pemasangan atau pada saat berjalan pada latar belakang. Proses *decode* dilakukan menggunakan *tool apktool.* Program apktool dieksekusi dengan terminal menggunakan parameter *d* diikuti dengan nama file sample malware. File *output* yang telah ter-*decode* dibuka menggunakan aplikasi pengolah teks dan menampilkan *source code* berupa informasi perizinan yang digunakan oleh sampel malware. seperti yang tertera pada gambar 3.



Gambar 3. Permission yang dibutuhkan oleh aplikasi Code4hk.apk.

Lalu kita ubah apk tadi menjadi file .jar agar bisa mengetahui isi source code dari aplikasi Code4hk tersebut karena dengan mengetahui source code kita bisa menganalisis aplikasi tersebut, cara merubah bisa dilihat seperti pada gambar 4.

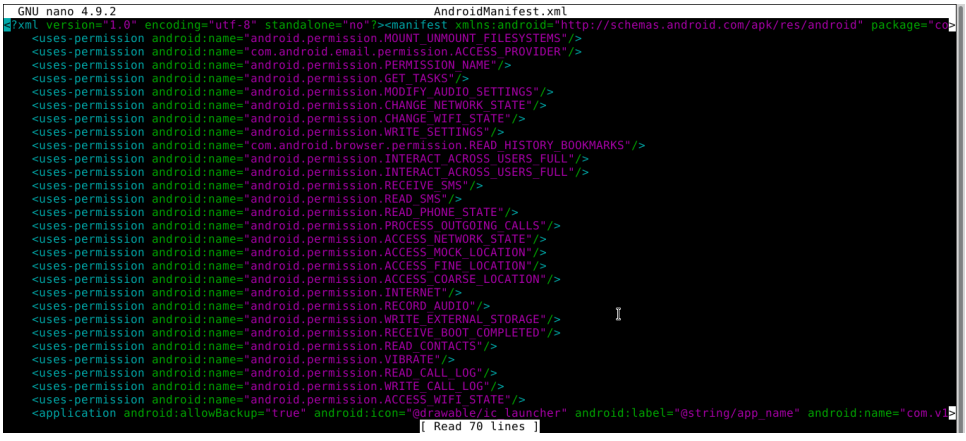


Gambar 4. Mengubah Code4hk.apk menjadi .jar.

Setelah diteliti lebih lanjut terdapat sebuah file .apk pada folder assets yang diinstall tanpa sepengetahuan pengguna yaitu bernama qq.apk.

Gambar 5. file qq.apk

Pada aplikasi qq.apk ini terdapat *permission* yang lebih berbahaya daripada code4hk.apk karena pada qq.apk bisa mengontrol semua aplikasi pada smartphone korban.

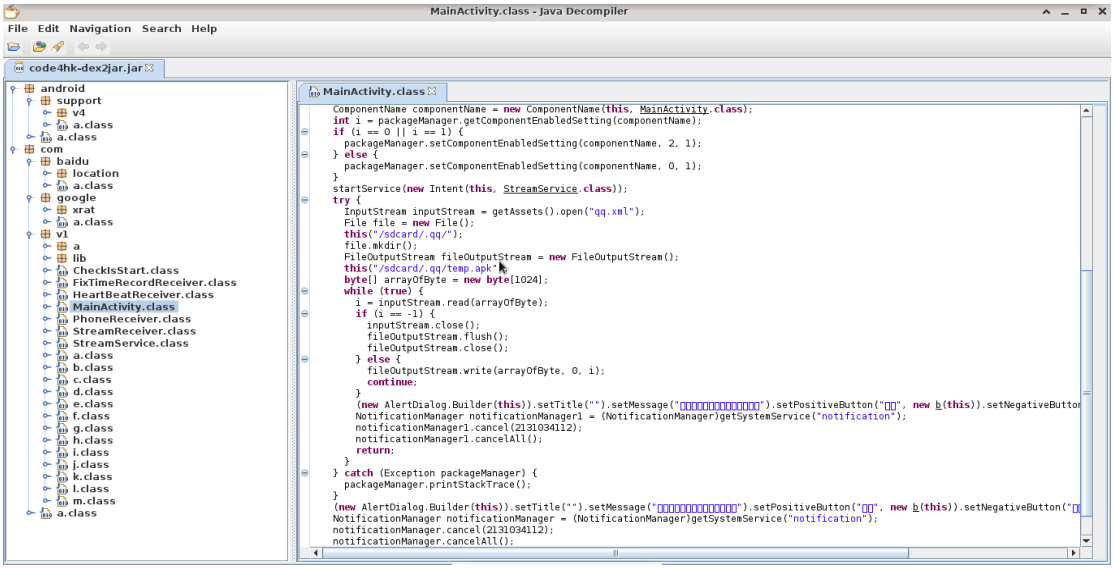


Gambar 6. Isi Permission dari qq.apk

3.2. Code Analisis Malware

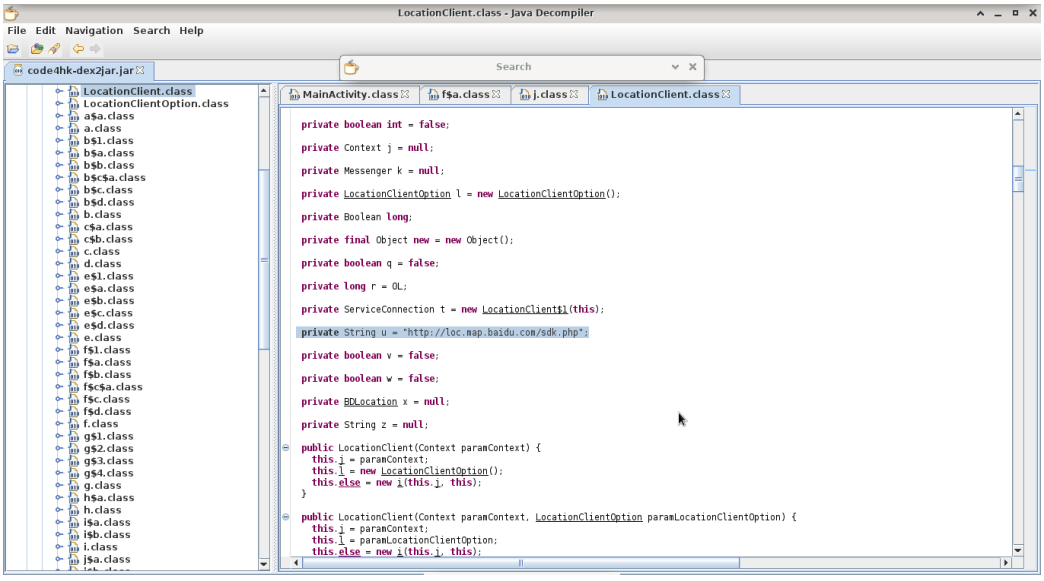
Kita melakukan analisis pada *Source Code* yang teradpat pada aplikasi code4hk untuk melihat kode yang mencurigakan. Pertama kita buka file jar tersebut menggunakan aplikasi *jd-gui* yang terdapat pada sistem operasi linux kita ketikkan peruintah *jd-gui* lalu akan muncul

aplikasi tersebut kemudian kita buka file .jar tadi lalu kita analisis pada folder *.com lalu v3* kemudian *MainActivity.class* seperti pada gambar 7.



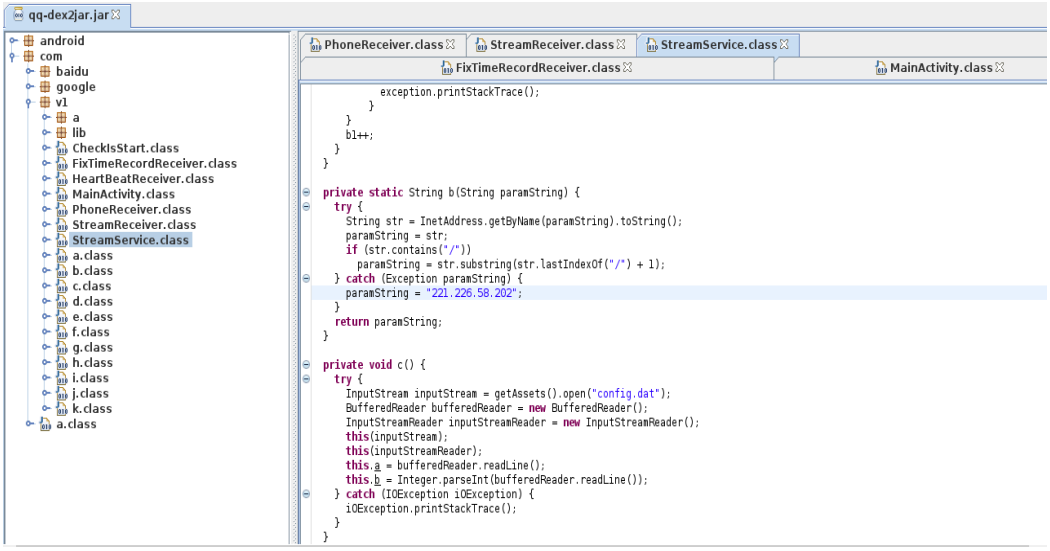
Gambar 7. Struktur code dan class pada Code4hk.apk.

Disini kami mendapatkan sebuat perintah dari aplikasi tersebut untuk membuat folder pada folder */sdcard/qq/temp/qq.apk.* Lalu pada *locationclient.class* aplikasi tersebeut terhubung pada sebuah website yaitu http://loc.map.baidu.com/sdk.php



Gambar 8. class file LocationClient.class

Lalu kita buka file jar tersebut menggunakan aplikasi *jd-gui* yang terdapat pada sistem operasi linux kita ketikkan peruintah *jd-gui* lalu akan muncul aplikasi tersebut kemudian kita buka file .jar tadi lalu kita analisis pada folder *.com lalu v1* kemudian *PhoneReceiver.class* seperti pada gambar 9.



Gambar 9. Isi code dari qq.apk

Analisis dilakukan untuk memeriksa fungsi yang ada pada *source code* sehingga menemukan kode yang bersifat malicious. Sebagian fungsi *malicous* ditemukan pada *qq.apk* diantaranya adalah menyadap panggilan yang diterima oleh kkorban mengumpulkan informasi smartphone, membaca SMS yang masuk pada smartphone korban, mendapatkan akses Admin, serta dapat mengirimkan informasi yang didapat ke server *Attacker* dengan ip 221.226.58.202.

4. Kesimpulan

Dari analisis yang dilakakukan pada aplikasi malware code4hk.apk dapat disimpulkan beberapa hal seperti dibawah ini:

1. Proses analisis malware bisa dilakukan dengan mengekstrak file .apk, lalu dikonversi dengan *tool dex2jar* dan JD-GUI untuk mendapatkan *source code.*
2. Aplikasi code4hk menginstall aplikasi pada *background* tanpa sepengetahuan pemilik *smartphone.*
3. Aplikasi code4hk dapat mengakses seluruh perangkat android tanpa sepengetahuan pemilik.
4. Aplikasi qq.apk dapat me-remote dan mengetahui lokasi smartphone karena terhubung dengan web http://loc.map.baidu.com/sdk.php dan server C&C dengan ip 221.226.58.202.
5. Pengguna android sebaiknya lebih hati-hati dalam menginstall aplikasi dari *Play Store* maupun Internet karena sekarang sudah banyak aplikasi yang disisipi malware terutama aplikasi mod.

Referensi:

[1] D. Septiani, N. Widiyasono, and H. Mubarok, “Investigasi Serangan Malware Njrat Pada PC,” *J. Edukasi Dan Penelit. Inform. JEPIN*, vol. 2, Dec. 2016, doi: 10.26418/jp.v2i2.16736.

[2] S. Jogsan, “Sunali Jogsan, 2020, Survei Deteksi Malware Berbasis Izin dalam Aplikasi Android, JURNAL INTERNASIONAL PENELITIAN TEKNOLOGI & TEKNOLOGI (IJERT) Volume 09, Edisi 04 (April 2020),” *J. Int. Penelit. Teknol. Teknol. IJERT*, vol. 09, Apr. 2020, doi: http://dx.doi.org/10.17577/IJERTV9IS040774.

[3] Adenansi, R., and Novarina, L. A., “Malware dynamic.,” *JoEICT J. Educ. ICT*, 2017.

[4] Ferdiansyah, “Analisis Aktivitas dan Pola Jaringan Terhadap Eternal Blue & Wannacry Ransomware,” vol. Vol.4, No.1, 2018.

[5] K. Tam, A. Feizollah, N. Anuar, R. Salleh, and L. Cavallaro, “The Evolution of Android Malware and Android Analysis Techniques,” *ACM Comput. Surv.*, vol. 49, pp. 1–41, Jan. 2017, doi: 10.1145/3017427.

[6] McAfee Inc., “McAfee Mobile Threat Report Q1, 2019,” McAfee Inc., California, 2019.

[7] S. Gadhiya and K. Bhavsar, “‘Techniques for Malware Analysis,’ International Journal of Advanced Research in Computer Science and Software Engineering,” *Pp 972-975*, vol. 2013.

[8] Moser, A., Kruegel, C., and Kirda, E., “Limits of static analysis for malware detection.,” *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, pp. 421–430, 2007.

[9] A. Salem, S. Banescu, and A. Pretschner, *Maat: Automatically Analyzing VirusTotal for Accurate Labeling and Effective Malware Detection*. 2020.

[10] M. Bhatt, H. Patel, and S. Kariya, “A Survey Permission Based Mobile Malware Detection,” *Int. J. Comput. Technol. Appl.*, vol. 6, p. 2, Oct. 2015.