ANALISIS MALWARE PADA SISTEM OPERASI ANDROID MENGGUNAKAN METODE *NETWORK TRAFFIC ANALYSIS*

MALWARE ANALYSIS IN ANDROID OPERATING SYSTEM USING NETWORK TRAFFIC ANALYSIS METHOD

Achmad Farhan Febrianto[1], Avon Budiono, S.T., M.T.[2], Ahmad Almaarif., S.Kom., M.T.[3]

[1,2,3]Program Studi Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom
[1]achmadfarhan@student.telkomuniversity.ac.id , [2]avonbudi@telkomuniversity.ac.id,
[3]ahmadalmaarif@telkomuniversity.ac.id

Abstrak

Malware* bisa disebut juga *malicious software* merupakan perangkat lunak yang diciptakan untuk menyusup atau merusak sistem komputer dan juga dapat mempengaruhi *network traffic*. Dengan meningkatnya penyebaran *malware* pada sistem operasi android saat ini. Untuk itu dilakukanya *malware analysis* pada android, *Malware* analysis pada sistem operasi android ini ada dua cara yaitu Static Analysis dan Dynamic Analysis. Static analysis dilakukan tanpa menjalankan *malware* tersebut dan hanya melihat segmen code pada aplikasi. Dynamic Analysis mengeksekusi program dan mengamati hasilnya Metode statik yang digunakan pada penelitian ini adalah *reverse engineering. Reverse engineering* digunakan untuk ekstraksi aplikasi kedalam *source code,* data tersebut akan dihasilkan informasi bahwa malware tersebut mengakses jaringan, Kemudian metode selanjutnya untuk mendekteksi *malware*-nya adalah *network traffic analysis*. Kelebihan *network traffic analysis* ini yaitu dapat mengetahui *malware* yang terhubung langsung dengan website dan dengan analisis ini juga dapat mengetahui aktivitas malware terhadap *network traffic* dari melihat *bandwitdh, time since request* dan payload yang dimiliki malware. Dari hasil analisis, informasi yang didapat adalah *malicious activity* yang mempengaruhi *network traffic.

Kata kunci : ***malware, malware analysis, static analysis, dynamic analysis, network traffic analysis***

Abstract

Malware can also be called malicious software which is software that is created to infiltrate or damage computer systems and can also affect network traffic. With the increasing spread of malware on the current Android operating system. For this reason, malware analysis is done on android, Malware analysis on the Android operating system has two ways, namely Static Analysis and Dynamic Analysis. Static analysis is done without running the malware and only looks at the segment code in the application. Dynamic Analysis executes the program and observes the results Static method used in this study is reverse engineering. Reverse engineering is used for extracting applications into the source code, the data will be generated information that the malware is accessing the network, then the next method to detect malware is network traffic analysis. The advantage of this network traffic analysis is that it can find out malware that is directly connected to the website and with this analysis can also find out malware activity against network traffic from seeing bandwidth, time since request and payload owned by malware. From the results of the analysis, the information obtained is a malicious activity that affects network traffic.

Keywords: ***malware, malware analysis, static analysis, dynamic analysis, network traffic analysis***

1. Pendahuluan

Saat ini perkembangan teknologi semakin cepat berkembang, Sehingga dapat mempermudah manusia untuk mencari informasi yang ada diseluruh dunia dengan menggunakan teknologi yang tersambung *internet*. Dan untuk saat ini *smartphone* merupakan teknologi yang perkembanganya cukup cepat dalam segi jumlah pengguna. Android sendiri merupakan sistem operasi yang sangat digemari oleh pengguna teknologi saat ini. Dengan banyaknya pengguna sistem operasi Android tersebut semakin banyak pula kejahatan yang berada dalam lingkup teknologi *smartphone*.

Cyber Crime adalah kejahatan di mana saluran komunikasi dan perangkat komunikasi telah digunakan secara langsung atau tidak langsung tanpa sepengetahuan korban apakah itu Laptop, Desktop, PDA, Ponsel (Gunjan, 2013). *Malware* bisa disebut juga *malicious software* merupakan perangkat lunak yang diciptakan untuk

menyusup atau merusak sistem komputer. Penyebaran *malware* saat ini begitu mudah baik melalui usb flashdisk, iklan-iklan tertentu pada website, dan media lainnya. (Cahyanto, 2017).

Android sebagai *Sistem operasi* terbuka banyak diincar oleh attacker pembuat *malware*. Pada tahun 2017 sudah ditemukan 750.000 *malware* pada *Sistem operasi* android baru. Diperkirakan sebanyak 3,5 juta *malware* android baru akan muncul pada 2017. Jumlah tersebut lebih banyak dibandingkan total *malware* android pada 2016 yang tercatat di kisaran 3,2 juta (G Data, 2017)

Meningkatnya penyebaran *malware* pada sistem operasi android saat ini maka penelitian ini dilakukan, karena masih minimnya pengetahuan user terhadap *android malware*. Oleh sebab itu perlu dilakukanya *malware analysis. Malware analysis* pada sistem operasi android ini ada dua cara yaitu Static Analysis dan Dynamic Analysis. Static analysis dilakukan tanpa menjalankan *malware* tersebut dan hanya melihat segmen code pada aplikasi. Dynamic Analysis mengeksekusi program dan mengamati hasilnya (Tam, 2017). Salah satu metode untuk mendekteksi *malware*-nya adalah *network traffic analysis*.

Network Traffic Analysis adalah metode analisis pada *malware* dengan memeriksa paket yang dikirimkan *malware* pada remote server dan membandingkan network traffic aplikasi terindikasi *malware* dan network traffic normal untuk mendapatkan karateristik pada *malware* tersebut (Arora, Garg, Peddoju, 2014). Kelebihan *network traffic analysis* ini yaitu dapat mengetahui *malware* yang terhubung langsung dengan website dan engan analisis ini juga dapat mengetahui aktivitas malware terhadap *network traffic* dengan melihat bandwitdh, *time since request* dan payload yang dimiliki malware.

Berdasarkan data yang telah dianalisis tersebut maka hasil dari penelitian ini adalah dampak dari *sample malware* terhadap *network traffic* dan pengaruh *malware* tersebut terhadap *smartphone* pengguna. Data tersebut berupa website yang diakses *malware* dan kecepatan akses internet berupa *time since request* dan *bandwidth*.

2. Dasar Teori

2.1 *Malicious Software* (*Malware*)

Malicious software (malware) adalah perangkat lunak yang dirancang untuk menyerang,merusak, menonaktifkan, atau mengganggu komputer, sistem komputer, atau Jaringan (Kurniawan, 2015). Malware mengacu pada program yang disisipkan ke dalam sebuah sistem, biasanya secara terselubung dengan maksud untuk melihat kerahasiaan korban baik berbentuk data, aplikasi, ataupun sistem oprasi (Zalavadiya, Sharma 2007).

2.2 Android Malware

Android Malware adalah perangkat lunak berbahaya yang dibuat untuk menyerang sistem operasi android pada smartphone. Malware bergantung pada eksploitasi sistem operasi (OS) tertentu dan software pada ponsel. Android malware merupakan perangkat lunak berbahaya yang menyerang smartphone yang dapat membuat hilangnya atau kebocoran informasi rahasia (Saxena, Shrivasta, Mourya, 2016). Android malware dapat mudah menyerang jaringan internet. Dengan semakin banyaknya Wi-Fi hotspot yang tersedia maka akan mempermudah malware menyerang melalui jaringan tersebut dan malware ini berbahaya karena tidak mudah terdeteksi keberadaanya (Jun Li, Zhang, 2014).

2.3 Jenis *Android Malware*

Pada umumnya Android Malware dikategorikan sebagai berikut (Arshad , Dll 2016):

a. Trojan

Trojan merupakan malware yang terlihat bersifat benign. Bahkan, trojan benar benar mencuri informasi rahasia pengguna tanpa sepengetahuan pengguna. Perangkat semacam ini dapat dengan mudah mendapatkan akses ke riwayat penelusuran, pesan , kontak dan nomor IMEI perangkat. Perangkat mencuri informasi ini tanpa persetujuan pengguna.

b. Backdoors

Backdoor merupakan malware yang menggunakan eksploitasi root untuk memberikan akses root ke malware dan memfasilitasi mereka untuk bersembunyi dari antivirus. Exploid, Rageagainsttecage (RATC) dan Zimperlich adalah tiga eksploitasi root teratas yang mendapatkan kontrol dari perangkat.

c. Worms

Worms merupakan malware yang menyerang melalui jaringan. Misalnya, worm bluetooth menyebarkan malware melalui jaringan bluetooth dengan mengirimkan salinan malware keperangkat yang terhubung.

d. Spyware

Spyware merupakan malware yang muncul sebagai aplikasi jinak, tetapi sebenarnya malware tersebut memonitori informasi rahasia pengguna seperti pesan, kontak, lokasi, dll. Spywares pribadi dapat menginstal muatan berbahaya tanpa sepengetahuan korban. Malware tersebut mengirimkan informasi korban seperti pesan teks, kontak ,dll kepada penyerang yang menginstal software

tersebut pada perangkat korban.

e. Botnet
Botnet merupakan malware yang ada pada jaringan perangkat android yang disusupi. Botmaster merupakan salah satu botnet yang menggunakan server jarak jauh untuk mengambil data korban dan penyerang mengontrol botnet melalui jaringan C & C.

f. Ransomwares
Ransomwares merupakan malware yang mencegah pengguna mengakses data mereka diperangkatnya dengan mengunci perangkat tersebut. Hingga jumlah tebusan yang diajukan penyerang dibayarkan.

g. Riskwares
Riskwares merupakan malware yang dapat mengurangi kinerja perangkat atau dapat juga merusak data misalnya menghapus, menyalin data, memodifikasi data, dll.

2.4 Static Analisis

Teknik static analysis dilakukan tanpa menjalankan aplikasi dalam emulator atau perangkat android. Analisi statis adalah teknik untuk mengamati perilaku malware dengan menganalisa segmen code. Namun ada beberapa kelamahan *static analysis* yaitu ada beberapa code yang sulit dipecahkan dengan teknik ini. Keuntungan dari analisis ini adalah biayanya yang terhitung murah dan tidak memakan waktu banyak. (Saxena, Shrivasta, Mourya, 2016). Bentuk analisis ini akan menguraikan, membongkar, dan mencari pola dalam file APK. Teknik analisis ini cepat dan tidak menghasilkan beban pemrosesan yang tinggi (Sawle, 2014).

2.5 Reverse Engineering

Teknik reverse engineering adalah teknik analisis statik dengan melakukan ekstraksi data informasi yang ada didalam malware sehingga dapat diketahui bagaimana malware tersebut bekerja dan membuat celah dalam melakukan serangan kedalam sistem (Nugroho, Prayudi, 2015). Dengan teknik reverse engineering pada malware keuntunganya adalah dapat mengetahui permission yang akan dijalankan oleh malware.

2.6 Dynamic Analysis

Teknik Dynamic Analysis menjalankan aplikasi yang terdapat malware dan mengamati perilakunya pada sistem serta mampu mengumpulkan informasi mengenai dampak terhadap smartphone ketika malware menjalankan prosesnya. Sehingga nantinya dapat diketahui apa saja yang dilakukan malware saat berhasil menginfeksi sebuah smartphone. Tahapan dalam analisis dinamis ini akan memeriksa smartphone degan menyeluruh dan melihat kemungkinan yang terjadi ketika sebuah smartphone yang telah terinfeksi malware. (Sikorski, Hoing 2012). Dengan cara ini mungkin ada beberapa bagian code yang tidak dijalankan akan tetapi dengan mudah mengidentifikasi perilaku jahat malware yang tidak terdeteksi oleh teknik analisis statis. Meskipun metode static analysis lebih cepat untuk mendeteksi malware tetapi metode tersebut gagal menganalisis obfuscation kode dan malware yang terenkripsi (Arshad , Dll 2016) Salah satu teknik dynamic analysis untuk menganalisi malware pada android adalah network traffic analysis.

2.7 Network Traffic Analysis

Network Traffic Analysis adalah metode analisis pada malware dengan memeriksa paket yang dikirimkan malware pada remote server dan membandingkan network traffic yang terdapat malware dan network traffic yang normal untuk mendapatkan karateristik pada malware tersebut serta apakah dengan adanya malware pada smartphone dapat mempengaruhi kecepatan kinerja smartphone terebut dan kecepatan pengiriman paket data (Arora, Garg, Peddoju 2014). Dengan teknik menganalisis malware pada network traffic keuntunganya adalah dapat mengetahui malware yang mengambil data pengguna smartphone dengan tanpa sepengetahuan pengguna.

2.8 Metodologi Penelitian

Kerangka konseptual dibangun berdasarkan teori yang sudah ada maupun dokumen-dokumen penelitian terdahulu sehingga terintegrasi sebagai satu kesatuan.

Metode yang digunakan dalam penelitian ini menggunakan model konseptual. Model konseptual sangat erat hubungannya dengan teori referensi/literatur yang akan digunakan. Model konseptual memberikan koneksi yang membuatnya lebih mudah untuk memetakan sebuah masalah. Selanjutnya model konseptual akan membantu menyederhanakan masalah dengan mengurangi jumlah *resource* yang akan digunakan. Sehingga peneliti dapat menjelaskan model konseptual pada penelitian tugas akhir ini yang memiliki tujuan untuk bagaimana sebuah *malware* dianalisis dengan menggunakan teknik *signature based detection* untuk

mendapatkan sebuah perintah program yang diduga sebagai *malware*.

Permasalahan dalam penelitian ini berada pada semakin pesatnya jenis-jenis *malware* yang menyerang komputer pengguna tiap tahunnya. Dari permasalahan tersebut didapatkan cara bagaimana cara untuk mengurangi kerentanan sistem pada setiap komputer pengguna. Cara yang dapat digunakan adalah dengan cara melakukan analisis *malware* pada file yang telah dicurigai sebagai *malware*.

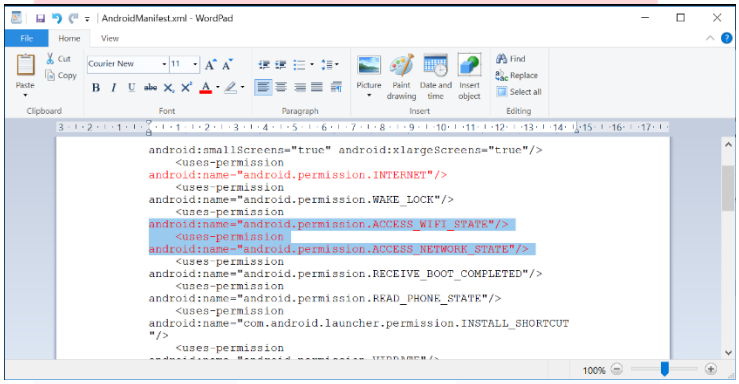
Dengan permasalahan yang ada dan peluang yang terdapat pada penelitian ini, dihasilkan sebuah artefak berupa analisis *malware* menggunakan teknik *signature based detection*. Untuk menghasilkan analisis *malware* tersebut, diperlukan sebuah teknik untuk melakukannya yang dapat membatu dalam analisis *malware* tersebut. Adapun teori yang digunakan adalah dengan menggunakan teori mengenai *static analysis*, teori *signature based detection*, dan teori mengenai API *call*.

Analisis yang akan dihasilkan yaitu berupa hasil API *call memory* yang akan dihubungkan dengan hasil signature yang telah didapat dalam pengujian ini dengan menggunakan metode *signature based detection*

3. Pengujian Sistem dan Analisis

3.1 Pengujian dengan Reverse Engineering

Pengujian yang harus dilakukan pertama kali yaitu melakukan reverse engineering pada apk sebuah *malware* untuk decompile dari apk menjadi barisan *source code.* Dari hasil *reverse engineering* diketahui apakah ada *source code* yang dapat mengancam *network traffic.*



Gambar 3-1 Hasil Reverse Engineering

Gambar 3-1 pada *sample8.apk* terdapat tiga *source code* yang menandakan malware mengakses jaringan internet. Adanya source code android.permission.INTERNET, android.permission.ACCESS_WIFI_STATE, dan android.permission.ACCESS_NETWORK_STATE yang menandakan bahwa malware tersebut mengakses *network traffic.* Dari hasil pengujian ini malware yang terdeteksi adanya *source code* tersebut maka akan dijadikan sample *malware* untuk pengujian lebih lanjut. Penjelasan source code yang mengakses jaringan internet adalah sebagai berikut.

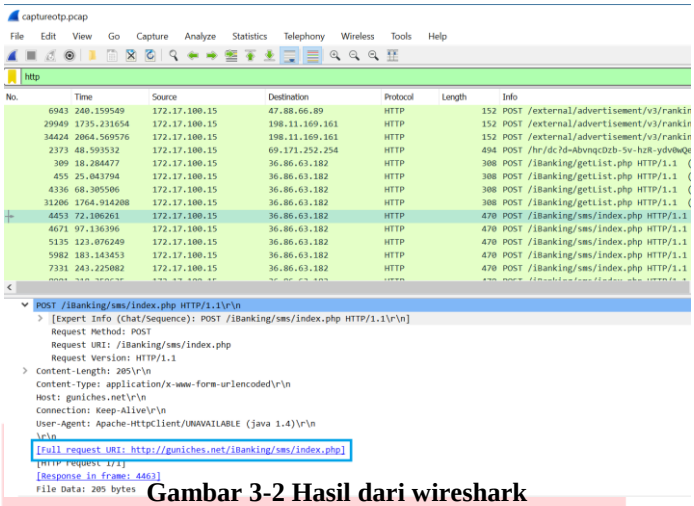
Source code permission	Deskripsi
android.permission.INTERNET	Permission tersebut berfungsi untuk meingizinkan aplikasi menggunakan layanan jaringan dan mengizinkan aplikasi membuka soket jaringan.
android.permission.ACCESS_WIFI_STATE	Persmission tersebut berfungsi untuk mengizinkan aplikasi mengakses informasi tentang jaringan Wi-Fi pada device.
android.permission.ACCESS_NETWORK_STATE	Permission tersebut berfungsi untuk mengizinkan aplikasi mengakses informasi tentang jaringan internet pada device.

Berdasarkan keterangan tabel diatas pengujian pada *sample8.apk* dapat disimpulkan bahwa *sample8.apk* meminta akses jaringan internet pada device pengguna.

3.2 Pengujian dengan Wireshark

Wireshark merupakan tools Network Protocol Analyzer atau biasa disebut sebagai penganalisa protokol jaringan. Pada pengujian ini wireshark digunakan untuk mendapatkan informasi atau aktivitas yang dilakukan

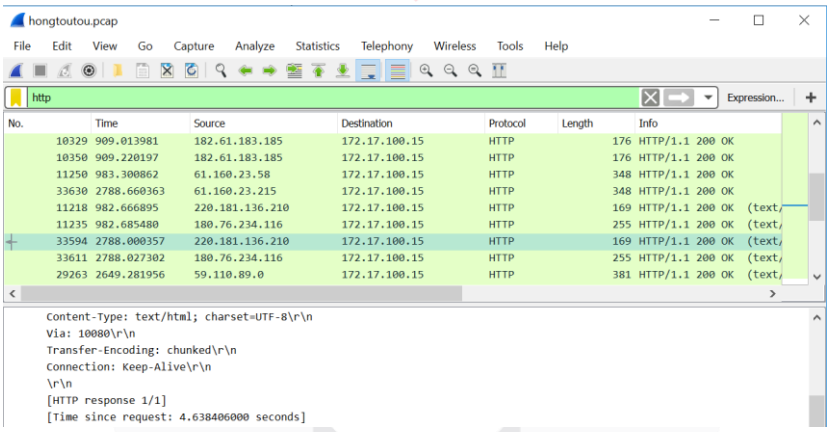
malware didalam network traffic. Pengambilan data yang dilakukan pada wireshark yaitu penggunaan aplikasi selama kurang lebih satu jam dan mendownload file yang berupa .pdf sebesar 9MB.



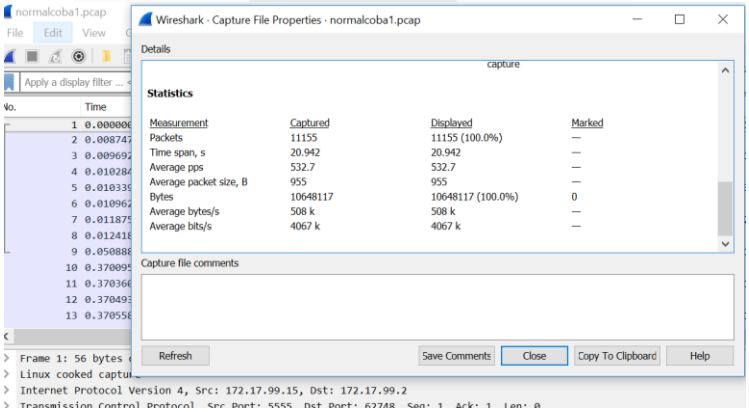
Gambar 3-2 Hasil dari wireshark

Gambar 3-2 diatas merupakan hasil dari penelitian pada dua buah malware *sample1.apk* menggunakan wireshark. Pada gambar tersebut dapat dilihat bahwa kedua *sample malware* mengakses sebuah domain.

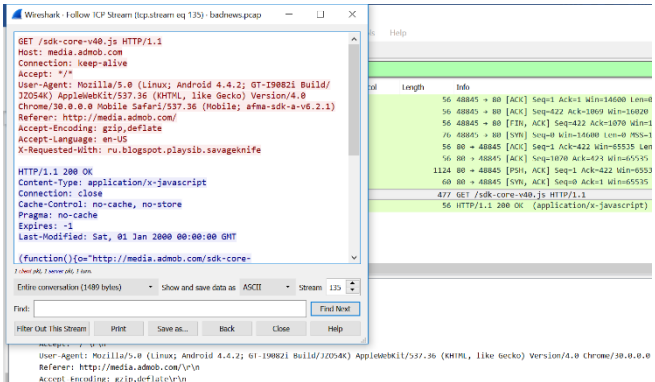
Pengujian terhadap *time since request* untuk mengetahui berapa lama waktu yang dibutuhkan dalam proses *Get and Post* pada sebuah protokol HTTP. Dengan melihat status dari proses *network traffic* pada protokol HTTP menunjukan OK.



Gambar 3-3 Hasil dari wireshark



Gambar 3-4 Hasil dari bandwidth



Gambar 3-5 Hasil dari payload

Gambar 3-3 hasil pengujian terhadap emulator yang sudah terinfeksi oleh malware yang berjalan dalam emulaor, gambar tersebut menunjukan waktu yang dibutuhkan untuk sebuah proses *get and post* sampai OK yaitu 4.6384 sec. Data diatas menunjukan bahwa *malicious activity* dapat mempengaruhi jaringan.

Gamber 3-4 Menunjukan bahwa bandwidth data yang diambil dari *Average bits/s* setelah dijadikan rata – rata menghasilkan 4.068K bits/s. Angka tersebut menjadi acuan sebagai *normal traffic* tidak adanya *malicious activity.*

Gamber 3-5 menunjukan bahwa malware tersebut melakukan malicious activity yaitu berupa adanya request GET terhadap URL pada http://media.admob.com/sdk-core-v40.js dan payload yang dibawa oleh malware berisikan file java script

3.3 Hasil Analisis menggunakan Reverse Engineering

Analisis menggunakan reverse engineering hasil yang diperoleh setelah melakukan pengujian yang dilakukan pada tools apktool dari sample malware yang dianalisis akan diambil informasi yang menandakan bahwa malware tersebut mengakses network traffic.

Sample Malware	Android Permission Internet		
	Permission 1	Permission 2	Permission 3
Sample1.apk	V	V	V
Sample2.apk	V	-	-
Sample3.apk	V	-	-

*Ket 1 = android.permission.INTERNET

2 = android.permission.ACCESS_NETWORK_STATE

3 = android.permission.ACCESS_WIFI_STATE

Dari hasil analisis ini bisa disumpulkan bahwa semua kelompok sample malware setidaknya memiliki permission yang mengakses internet dan terdapat juga ada sample malware yang terdeteksi bahwa permission tersebut mengakses informasi tentang jaringan dan informasi tentang wifi yang digunakan oleh device.

3.4 Hasil Analisis Time since request dan bandwidth

Pada bagian analisis yang menggunakan wireshark ini hasil yang akan didapatkan adalah *malicious activity* pada saat malware dijalankan pada emulator dan direkam oleh wireshark. Informasi yang akan diambil dari malicious activity pada malware adalah *Timesince request, bandwidth* dan isi *payload*

Sample	Time since tertinggi (s)	Bandwidth (Bits/s)
Normal Traffic	0.8627	4.068K
Sample1.apk	9.7989	2.093K
Sample2.apk	1.3251	3.135K

Tabel diatas merupakan hasil pengukuran analisis bandwitdh pada wireshark dan sudah dilakukan pengukuran untuk bandwidth tersebut. Analisis ini dilakukan dengan mengunduh file dengan size 9MB yang berformat .pdf pada email yahoo agar dapat mengukur *bandwidth* yang terpakai.

3.5 Hasil analisis payload dan malicious activity

Sample	Website	Jenis Aplikasi
Sample1.apk	• Guniches.net	Aplikasi facebook seperti facebook lite yang mengirimkan *One time*

	• Utosedi.net	*Password* untuk mengambil data user.

Payload	Malicious Activity
Iccid : 89014103212407959864 os : Android 4.4.2 operator : T-Mobile	Malicious activity yang terdeteksi adalah pengalihan pada website yang seharusnya melakukan get pada guniches.net tetapi dialihkan ke mercusuar.uzone.id ini berdasarkan dengan code 302 Found yang termasuk dalam kategori Redirection kemudian pada post terdapat pengiriman payload pada http server yang berupa data device.

Hasil analisis diatas dapat diambil kesimpulan Jika terdapat payload maka payload tersebut bisa bermacam macam ada file berbentuk HTML, Javascript, png dan bisa juga berisikan data device dari pengguna. Semua sample malware yang memiliki malicious activity dipastikan juga mengakses sebuah website/ip yang malware tersebut miliki.

3.6 Dampak Malware terhadap network traffic

Berdasarkan hasil analisis dapatdiketahui bagaimana dampak malware tersebut terhadap *network traffic.* Berikut ini merupakan tabel dari dampak yang disebabkan oleh malware terhadap *network traffic.*

Sample Malware	Malicious activity	Dampak
Sample1.apk	Malicious activity yang terdeteksi adalah pengalihan pada website yang seharusnya melakukan get pada guniches.net tetapi dialihkan ke mercusuar.uzone.id ini berdasarkan dengan code 302 Found yang termasuk dalam kategori Redirection kemudian pada post terdapat pengiriman payload pada http server yang berupa data device.	Malware ini berjenis Trojan yang bersifat mengambil informasi tanpa sepengetahuan pengguna, telah dibuktikan dengan *sample1.apk* yang hasilnye berupa informasi tentang device dan adanya mengakses website secara diam diam sehingga mengakibatkan penurunan bandwidth yang cukup besar, untuk waktu yang dibutuhkan dalam proses *get and post* pada HTTP terbilang paling tinggi dari semua sample.

4. Kesimpulan

4.1 Teknik statik analisis dibutuhkan untuk menentukan sampel malware yang mengakses jaringan dilihat dari permission yang dimiliki. Permission didapatkan melalui reverse engineering menggunakan apktool. Data yang didapatkan menjadi acuan bahwa malware tersebut bisa dianalisis lebih lanjut menggunakan tools yang sudah ditentukan. menggunakan reverse engineering untuk mencari malware yang akan dijadikan sampel uji.

4.2 Analisis pada network traffic dilakukan setelah mendapatkan file tcpdump dari emulator. Kemudian dilihat pada tool wireshark dianalisis dengan dua cara. Pertama mendiamkan malware selama kurang lebih satu jam untuk dilihat malicious activity yang dilakukan malware selama dijalankan pada emulator untuk mendapatkan website atau ip, time since request dan payload. Kedua melakukan download pada saat malware dijalankan dengan tiga kali pengujian kemudian dicari rata rata dari Average bits/s pada measurement untuk dijadikan perhitungan bandwidth sebagai tanda bahwa malware tersebut berpengaruh terhadap network traffic.

4.3 Hasil pengujian dan analisis yang telah dilakukan dianalisis kembali untuk dijadikan dampak yang disebabkan oleh malware, Malicious activity terhadap network traffic sangat berpengaruh terhadap kecepatan akses jaringan internet dan ada juga malware yang mengambil data informasi tanpa sepengetahuan user.

Daftar Pustaka:

[1] Gunjan, V. K., Kumar, A., & Avdhanam, S. (2013). A survey of cyber crime in India. 2013 15th International Conference on Advanced Computing Technologies (ICACT)

[2] Cahyanto, T. A. (2017). Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis. Jurnal Sistem & Teknologi Informasi Indonesia, 2, 1st ser., 1-12.

[3] K. T. (2017). The Evolution of Android Malware and Android Analysis Techniques. ACM Computing Surveys, 49, 76th ser., 1-41.

[4] Arora, A., Garg, S., & Peddoju, S. K. (2014). Malware Detection Using Network Traffic Analysis in Android Based Mobile Devices. 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies.

[5] N. Z. (2017). A Methodology of Malware Analysis, Tools and Technique for windows platform – RAT Analysi. International Journal of Innovative Research in Computer and Communication Engineering, 5(3), 1-13.

[6] Kurniawan, H., Rosmansyah, Y., & Dabarsyah, B. (2015). Android anomaly detection system using machine learning classification. 2015 International Conference on Electrical Engineering and Informatics (ICEEI).

[7] V. S. (2016). Behavior Analysis of Android malware detection for Smart phon. International Journal of Engineering Research & Science (IJOER), 2(12), 1-6

[8] Li, J., Zhai, L., Zhang, X., & Quan, D. (2014). Research of android malware detection based on network traffic monitoring. 2014 9th IEEE Conference on Industrial Electronics and Applications.

[9] Arshad, S., Ali, M., Khan, A., & Ahmed, M. (2016). Android Malware Detection & Protection: A Survey. International Journal of Advanced Computer Science and Applications, 7(2).

[10] Zaman, M., Siddiqui, T., Amin, M. R., & Hossain, M. S. (2015). Malware detection in Android by network traffic analysis. 2015 International Conference on Networking Systems and Security (NSysS).

[11] Deka, D., Sarma, N., & Panicker, N. J. (2016). Malware detection vectors and analysis techniques: A brief survey. 2016 International Conference on Accessibility to Digital World (ICADW)

[12] Sikorski, M., & Honig, A. (2012). Practical malware analysis the hands-on guide to dissecting malicious software. San Francisco, CA: No Starch Press.

[13] Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y. (2011). "Andromaly": A behavioral malware detection framework for android devices. Journal of Intelligent Information Systems,38(1)

[14] Edem, E. I., C. B., A. A., & P. W. (2014). Analysis of Malware Behaviour: Using data Mining Clustering Techniques to Support Forensics Investigation. Fifth Cybercrime and Trustworthy Computing Conference, 1-10.

[15] Sawle, P. D., & Gadicha, A. B. (2014). Analysis of Malware Detection Techniques in Android. International Journal of Computer Science and Mobile Computin, 3(3), 176-182.

[16] Lueg, C. (2017, April 27). Blog (EN). Retrieved May 27, 2019, from https://www.gdatasoftware.com/blog/2017/04/29712-8-400-new-android-malware-samples-every-day

[17] Nugroho, H. A., & Y. P. (2015). Penggunaan teknik reverse engineering pada malware analysis untuk identifikasi serangan malware.