Analisis Dampak *Malware* Terhadap Trafik Jaringan dengan Teknik *Deteksi Behavior-based*

Analysis of The Impact of Malware on Network Traffic With Behavior-based Detection Techniques

Adib Fakhri Muhtadi[1], Avon Budiono, S.T., M.T.[2], Ahmad Almaarif, S.Kom., M.T.[3]

[1,2,3]Prodi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom
[1]adibfakhrimuhtadi@student.telkomuniversity.ac.id, [2]avonbudi@telkomuniveristy.co.id,
[3]ahmadalmaarif@telkomuniversity.ac.id

Abstrak

***Malware* merupakan sebuah perangkat lunak atau program komputer yang digunakan untuk menjalankan *malicious activity*. *Malware* dibuat dengan tujuan merugikan *users* karena dapat melakukan pengubahan data *users*, menghabiskan *bandwidth* dan juga sumber daya lain tanpa ijin *users*. Beberapa penelitian sudah dilakukan sebelumnya untuk mengidentifikasi jenis *malware* dan efeknya. Tetapi tidak mengatasi secara kompleks karena penelitian yang dilakukan sebelumnya hanya terfokus pada pengelompokkan jenis-jenis *malware* yang menyerang melalui trafik jaringan. Oleh karena itu dibutuhkan penelitian analisis dampak malware terhadap trafik jaringan dengan teknik deteksi behavior-based. Teknik ini menganalisis malware dengan menjalankan sampel malware ke dalam sebuah environment dan memantau aktivitas yang ditimbulkan oleh sampel malware. Untuk memperoleh hasil yang akurat, analisis dilakukan dengan mengambil informasi API call network dan aktivitas trafik jaringannya. Dari analisis API call network malware, akan dihasilkan informasi mengenai urutan API call network yang digunakan oleh malware. Kemudian dari trafik jaringannya, diperoleh aktivitas-aktivitas malware dengan menganalisis behavior trafik jaringan malware, payload, dan bandwidth trafik yang terinfeksi. Selanjutnya dari hasil urutan API call network yang digunakan malware dan hasil analisis trafik jaringannya, akan dianalisis sehingga dapat ditentukan dampak malware terhadap trafik jaringan.**

Kata kunci: *malware, dynamic analysis, behavior-based,* trafik jaringan, API *call network*

Abstract

Malware is a software or computer program that is used to run malicious activities. Malware is created with the aim of harming users because it can make changes to data users, spend bandwidth and also other resources without users permission. Several studies have been done before to identify the types of malware and their effects. But the research was not carried out in more depth because the previous research was only on grouping the types of malware that attack through network traffic. Therefore research is needed to analyze the impact of malware on network traffic with behavior-based detection techniques. This technique analyzes malware by running malware samples into an environment and monitoring activities caused by malware samples. To obtain accurate results, the analysis is done by retrieving API call network information and network traffic activities. From the analysis of the malware call network API, the information will be generated about the order of the call network API used by malware. Then from the network traffic, malware activities are obtained by analyzing the network traffic behavior of the infected malware, payload, and traffic bandwidth. Furthermore, from the results of the API call network sequence used by malware and the results of its network traffic analysis, it is analyzed so that the impact of malware on network traffic can be determined.

Keywords: malware, dynamic analysis, behavior-based, network traffic, API call network.

1. Pendahuluan

Di era IoT (Internet of Things) salah satu ancaman terbesar di internet saat ini adalah malicious software, biasanya disebut malware karena hampir semua penyebab utama masalah keamanan internet adalah malware. Malware merupakan program yang diciptakan dengan tujuan merusak dengan menyusup ke sistem komputer. Malware memiliki berbagai macam jenis, yaitu virus, worm, spyware, adware, trojan, keylogger, rootkit, botnet dan phising [1]. Contohnya, botnet biasanya digunakan untuk mengirimkan spam dan host phising website yang menyulitkan untuk dilacak dan diblacklist [2]. Selain botnet, malware yang sering digunakan untuk menyusupi suatu sistem adalah spyware. Menurut data statistik, 70-80% spyware berasal dari situs website yang dianggap aman oleh pengguna internet [3].

Berdasarkan laporan dari ShadowServer (sebuah organisasi yang bergerak dibidang cybercrime), ada ribuan sampel malware baru yang diterima setiap harinya [2]. Setiap sampel malware di analisis satu persatu dengan

tujuan untuk mengetahui jenis malware apa, seberapa besar ancamannya dan bagaimana cara penanganannya. Untuk mendapatkan informasi lengkap mengenai satu sampel malware dibutuhkan analisis static dikarenakan analisis ini dilakukan dengan meneliti source code malware tersebut [4]. Namun, analisis ini menjadi tidak efektif dan efisien karena pembuat malware saat ini sudah mulai menggunakan teknik executable packing dan obfuscation, teknik ini dapat menciptakan beberapa varian malware baru dari satu malware [5]. Untuk mengatasi masalah ini, diperlukan Dynamic Analysis.

Dynamic Analysis biasanya menganalisis informasi behavioral seperti aktivitas jaringan, API call, file operation dan catatan modifikasi registri dengan mengeksekusi sampel dalam environment virtual. Kekurangan dari Dynamic Analysis adalah metode ini memerlukan waktu dan sumber daya yang besar untuk eksekusi malware [6]. Penggunaan metode Dynamic Analysis dengan teknik behavioral akan menghasilkan informasi API call.

Pada Windows, setiap program yang dapat dieksekusi perlu membuat satu set API call. Contohnya, untuk file management ada beberapa API calls yaitu, OpenFile, DeleteFile, FindClose, FindFirstFile, GetFileSize [7]. Jadi informasi API call yang diperoleh setelah melakukan Dynamic Analysis dengan teknik behavior-based berguna untuk mengetahui aktivitas apa yang dilakukan dan perilaku apa yang dimiliki oleh malware tersebut sehingga dapat diketahui jenis malware tersebut dan seberapa besar ancamannya terhadap sistem berdasarkan API call yang dijalankan.

Analisis malware secara Dynamic Analysis terhadap network traffic perlu dilakukan karena sedikitnya penelitian yang dilakukan sebelumnya bahkan belum adanya penelitian Dynamic Analysis terhadap trafik jaringan secara kompleks. Untuk memperoleh hasil analisis ini secara kompleks, diperlukan API call network pada setiap malicious activity yang dijalankan oleh malware menggunakan tool Cuckoo Sandbox. API network yang diperoleh setelah mengeksekusi malware pada environment virtual akan dianalisis urutan pemanggilannya dan API network apa saja yang digunakan oleh malware tersebut sehingga dapat diketahui aktivitas-aktivitas yang dilakukan malware yang memanfaatkan Windows API khususnya API network. Setelah semua informasi mengenai API network diperoleh, dilakukan analisis trafik jaringan yang direkam selama menjalankan malware pada environment virtual menggunakan Wireshark. Analisis yang dilakukan pada hasil capture trafik jaringannya adalah dengan melihat behavior malware pada trafik jaringan, payload yang dibawa oleh malware, dan pengukuran bandwidth antara trafik normal dengan trafik yang sudah terinfeksi malware.

Berdasarkan hasil analisis urutan API network dan trafik jaringan tersebut maka hasil akhir dari penelitian ini berupa penjelasan bagaimana dampak setiap sampel malware yang digunakan dalam penelitian ini terhadap trafik jaringan

2. Dasar Teori dan Sistematika Penelitian

2.1 Malware

Malware atau Malicious Software adalah suatu perangkat lunak yang dibuat untuk menyerang dengan merusak, mengganggu, mengambil informasi-informasi penting yang melibatkan confidentiality, integrity dan availability data suatu sistem ataupun aplikasi [8].

Malware merupakan perangkat lunak yang secara eksplisit didesain untuk melakukan malicious activity seperti Trojan, Virus, Spyware dan Exploit [9]. Cara kerja malware adalah dengan masuk ke suatu sistem seperti sistem komputer yang ada melalui berbagai aplikasi yang ada pada sistem tersebut atau bisa juga melalui pengiriman data dari perangkat yang telah terkena virus [10].

1. Jenis-jenis Malware

Malware dapat dibedakan menjadi beberapa macam sesuai dengan metode operasinya dan karakteristiknya [11], berikut jenis-jenis malware.

a. Logic Bomb: malware yang memiliki dua bagian inti yaitu, isi dan pemicu. Malware jenis ini akan aktif apabila pemicu dijalankan. Sama halnya dengan bom waktu, Logic Bomb dapat aktif dengan rentang waktu tertentu.
b. Trojan Horse: sebuah program yang terlihat tidak berbahaya tetapi menjalankan malicious task secara diam-diam. Contohnya ketika user ingin melakukan login ke suatu website dengan memasukkan username dan password tetapi website tersebut sudah dipasang program password-grabbing, peran dari Trojan Horse adalah menampilkan pesan error yang menyatakan bahwa user salah memasukkan username dan password tetapi dibalik proses itu Trojan Horse sudah mengambil informasi autentikasi user.
c. Back Door: sebuah mekanisme yang melewati proses pemeriksaan standar keamanan misalnya authentikasi.
d. Virus: sebuah malware yang akan menggandakan dirinya dengan menginfeksi program lain yang sedang dijalankan.
e. Worm: malware yang mirip dengan virus yaitu dapat menggandakan diri, perbedaannya adalah Worm dapat menggandakan diri dari suatu sistem ke sistem lain melalui jaringan dan tidak bergantung pada executable program.
f. Rabbit: sebuah program yang menggunakan semua resource pada suatu sistem. Contohnya adalah Fork Bomb yang selalu membuat proses baru secara berulang dengan jumlah loop tak terhingga sehingga membuat sistem tersebut menjadi lambat.
g. Spyware: sebuah program yang mengambil informasi sebuah komputer dan mengirimnya ke orang lain.
h. Adware: sebuah program yang mirip dengan Spyware tetapi fokusnya adalah marketing.
i. Zombies: sebuah istilah untuk komputer yang sudah diserang/hack tanpa sepengetahuan pengguna. Biasanya aktivitas yang dijalankan adalah menyebarkan email spam.

2.2 Dynamic Analysis

Malware Analysis adalah sekumpulan proses yang dilakukan dengan membedah malware untuk memahami cara kerjanya, cara mengidentifikasi dan cara mengalahkan atau menghilangkannya. Proses ini dibutuhkan untuk menyesuaikan pengembangan teknik pendeteksi malware terhadap pertumbuhan malware baru. Malware Analysis terbagi menjadi tiga metode yaitu Static Analysis, Dynamic Analysis dan Hybrid Analysis [12].

Dynamic Analysis adalah metode deteksi malware dengan menjalankan malware tersebut dalam suatu environment virtual. Metode ini memantau perilaku malware, interaksinya dengan sistem dan efeknya terhadap sistem. Analisis dengan metode ini diharuskan menyediakan environment yang terisolasi agar tidak menyebabkan program lain terkena efek karena menjalankan malware secara langsung [13].

2.3 Teknik Behavior-based

Behavior-based adalah teknik dari metode Dynamic Analysis yang tujuannya untuk mengetahui perilaku malware. Teknik ini dilakukan dengan mengeksekusi malware dan memantau perilakunya dalam aktivitas jaringan, API call, file operation dan catatan modifikasi registry. Kelemahan dari teknik ini adalah memerlukan resources dan waktu yang banyak, sedangkan kelebihan dari teknik ini adalah dapat melakukan identifikasi atau pengenalan terhadap malware-malware baru [14].

Teknik Behavior-based memiliki beberapa kelebihan dan kekurangan [15]. Berikut kelebihan dan kekurangannya.

Kelebihan:

1. Deteksi perilaku malware yang kompleks karena teknik ini mendeteksi malware mulai dari perilaku sampai tujuan akhir malware.
2. Dapat mengidentifikasi malware-malware baru.
3. Dapat mengidentifikasi banyak malware dengan waktu yang lebih sedikit dibandingkan dengan teknik lain.
4. Mempermudah pengklasifikasian malware karena menggunakan beberapa tools untuk mengotomatisasi pendeteksian.

Kekurangan:

1. Tidak mendeteksi secara detail seperti memeriksa signature maupun bit-bit malware.
2. Tidak dapat menemukan solusi malware.
3. Memakan resources yang banyak karena deteksi ini harus menjalankan malware secara langsung di suatu environment.

2.4 API Network Windows

API atau Application Programming Interfaces merupakan antar muka pemrograman aplikasi yang dibuat oleh Windows agar dapat berinteraksi dengan sistem kernel. Misalnya jika aplikasi ingin menjalankan File Management, aktivitas tersebut akan melibatkan beberapa API call seperti OpenFile, DeleteFile, FindClose, FindFirstFile, GetFileSize. Terdapat dua API call yang akan digunakan dalam penelitian ini yaitu API memory dan API network. Windows API adalah dynamic-link libraries (DLL) yang merupakan bagian dari sistem operasi Windows. Keuntungan menggunakan Windows API adalah dapat menghemat waktu pengembangan karena berisi lusinan fungsi yang sudah ditulis dan menunggu untuk digunakan.

API network merupakan API di sistem operasi Windows yang digunakan dalam melakukan proses transfer/request data dalam jaringan, selain itu juga memungkinkan komunikasi antar aplikasi melalui jaringan. API network biasanya digunakan oleh malware dengan menyembunyikan malicous code ke dalam lalu lintas pengiriman ataupun permintaan data pada sebuah jaringan [16].

2.5 Network Analysis

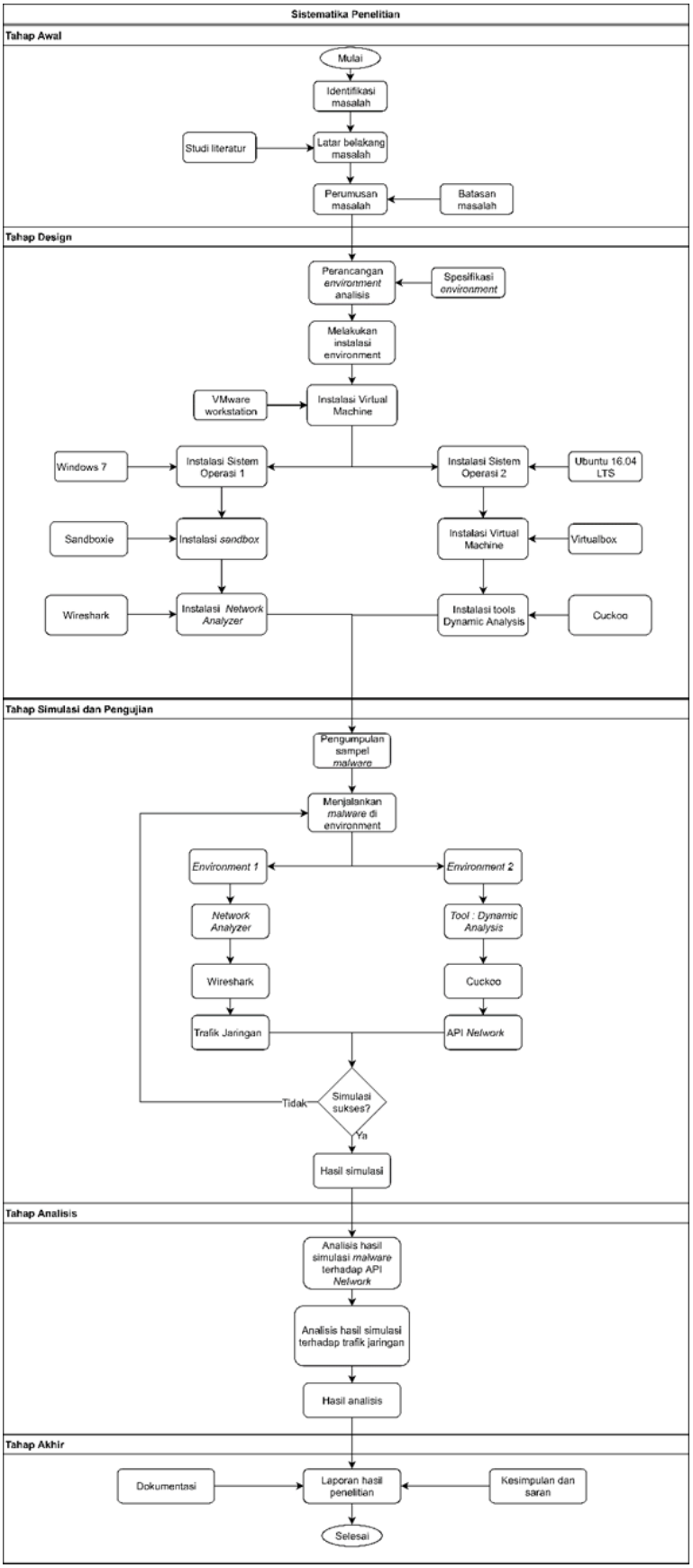
Network analysis biasanya meliputi traffic analysis, protocol analysis, packet analysis, eavesdropping, dsb. Network analysis adalah proses menangkap suatu lalu lintas jaringan dan memeriksanya secara rinci untuk mengetahui apa yang sedang terjadi pada jaringan tersebut. Sebuah penganalisa jaringan adalah sebuah kombinasi antara hardware dan software. Pada penelitian ini, network analyzer yang digunakan adalah Wireshark.

Wireshark adalah salah satu aplikasi penganalisa jaringan terbaik yang tersedia gratis. Wireshark memiliki banyak kelebihan seperti graphical user interface (GUI) yang bagus, mendukung 400 protokol, dan aktif dalam pengembangan dan maintenance. Wireshark dapat dijalankan pada berbagai sistem operasi seperti UNIX-based systems, Mac OS X, dan Windows [17].

2.6 Sistematika Penelitian

Sistematikan penelian merupakan bagian yang mendeskripsikan atau menjelaskan langkah-langkah yang dilakukan dalam penelitian. Langkah pertama ialah identifikasi dari masalah hingga terakhir yaitu laporan atau kesimpulan dari penelitian.

Gambar 1 Sistematika Pemecahan Masalah



3. Pengujian dan Analisis

3.1. Pengujian

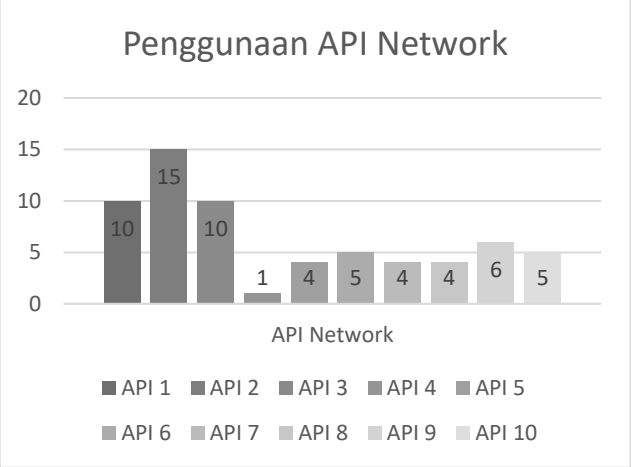
Pengujian dilakukan melalui analisis API *network* dan Trafik Jaringan. Pengujian pertama dilakukan pada Cuckoo Sandbox yang sudah diinstal pada *localhost* Ubuntu yang ada di Vmware. Dari pengujian dengan Cuckoo, diperoleh informasi mengenai semua API *network* yang digunakan oleh malware. Pengujian selanjutnya adalah pengujian menggunakan Wireshark. Wireshark adalah tool penganalisa trafik jaringan yang ditujukan untuk memantau lalu lintas sebuah jaringan. Pengujian ini bertujuan untuk memperoleh aktivitas yang dilakukan malware pada trafik jaringan sehingga dapat diketahui bagaimana perilaku *malware*, *payload*, dan dampaknya terhadap trafik jaringan. Pada pengujian ini analisis yang dilakukan dibagi menjadi tiga bagian, yaitu analisis lalu lintas jaringannya untuk mengetahui perilaku *malware*, analisis *payload* untuk mengetahui aktivitas yang dilakukan, dan analisis pengukuran bandwidth untuk mengetahui dampaknya terhadap trafik jaringan.

1. Cuckoo Sandbox (API *network*)

Setelah melakukan analisis pada Cuckoo untuk semua sampel *malware*, dapat dilakukan pengelompokkan urutan-urutan API *network*. Pengelompokkan dilakukan berdasarkan TID (*task* ID) setiap API *network*. Berikut adalah pengelompokkan urutan-urutan API *network* setelah melakukan analisis terhadap 30 sampel *malware*.

Tabel 1 Kelompok Urutan API *Network*

Kelompok	Urutan API *network*
API 1	*InternetCrackUrlA-InternetOpenA-InternetConnectA-HttpOpenRequestA-HttpSendRequestA-InternetReadFile-InternetCloseHandle*
API 2	*WSAStartup-socket-setsockopt-closesocket-GetAddrInfoW*
API 3	*getaddrinfo-socket-connect-send-recv-closesocket*
API 4	*WSAStartup-InternetOpenA-InternetConnectA-socket-closesocket-NSPStartup-socket-closesocket-WSASocketA-bind-setsockopt-ConnectEx-shutdown-closesocket-InternetCloseHandle-URLDownloadToFileW*
API 5	*WSAStartup-socket-setsockopt-closesocket-WSASend-WSARecv-shutdown-closesocket-HttpQueryInfoA*
API 6	*WSAStartup-WSASocketW-setsockopt-closesocket-gethostname-gethostbyname-GetAdaptersAddresses-GetAdaptersInfo*
API 7	*WSASocketW-WSAConnect-ioctlsocket-closesocket*
API 8	*WSAStarup-InternetCrackUrlA-ObtainUserAgentString-socket-gethostbyname-connect-closesocket*
API 9	*Socket-ioctlsocket-gethostbyname-connect-select-closesocket*
API 10	*InternetOpenA-InternetOpenUrlA-InternetReadFile-InternetCloseHandle*



Gambar 2 Grafik Penggunaan kelompok API *network*

Berdasarkan grafik di atas, dapat disimpulkan bahwa kelompok API *network* yang paling banyak digunakan adalah kelompok urutan API *network* yang ke-2. Berarti setengah dari 30 sampel *malware* memiliki fungsi yang dapat menggunakan *Windows Socket* API agar dapat menyediakan sarana komunikasi dua arah berorientasi koneksi antara *client* dan *server* malware. Dengan *Windows Socket* API, *server malware* dapat mempengaruhi jaringan Windows seperti *Quality of Service* (QoS) sehingga jaringan yang telah terinfeksi dapat terjadi penurunan QoS pada trafik jaringannya.

Di urutan kedua, penggunaan kelompok urutan API *network* yang paling banyak adalah kelompok API 1 dan 3 dengan total penggunaan masing-masing sebanyak 10 sampel *malware*. Hal ini dapat dikatakan bahwa sebanyak 10 *malware* dengan kelompok API *network* ke-1 memiliki aktivitas untuk mengakses sebuah *url/ip* lalu menyediakan layanan *file transfer* antara *client* dan *server* sehingga tanpa disadari, server dapat mengirim dan mengambil data pada komputer *client*. Sedangkan 10 *malware* dengan kelompok API *network* ke-3 memiliki aktivitas untuk mengirim dan menerima data melalui *socket* yang sudah ditentukan. API ke-3 akan mempengaruhi *bandwidth* pada trafik jaringan

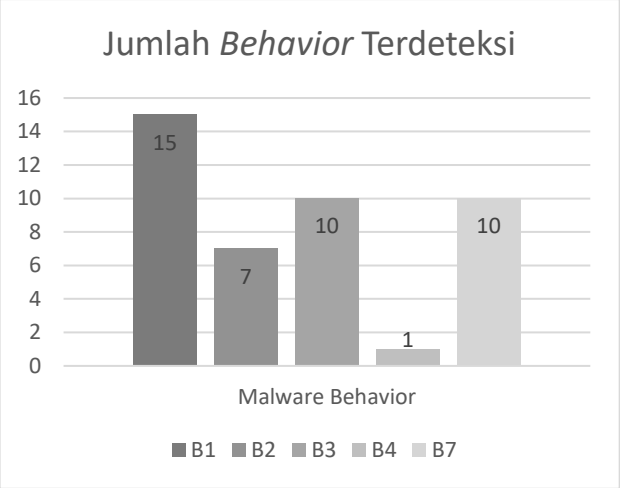
karena proses mengirim dan menerima data yang terjadi.

2. Wireshark

a. *Behavior Malware* pada Trafik Jaringan

Pada pengujian bagian ini, bertujuan untuk mengetahui perilaku-perilaku *malware* pada trafik jaringan. Berikut adalah *behavior malware* pada trafik jaringan yang dikelompokkan menjadi empat bagian, yaitu DNS and NetBIOS *request*, DNS and rDNS, ICMP *echo*, *suspicious activities* [19].

- B1: melakukan DNS *request* / rDNS / NetBIOS *name request*.
- B2 : upaya koneksi ke alamat IP gagal sedangkan permintaan DNS berhasil.
- B3 : upaya koneksi ke alamat IP dengan tidak adanya rDNS.
- B4 : upaya koneksi ke alamat IP dengan rDNS yang gagal.
- B5 : ICMP *echo* ke alamat IP tanpa balasan atau pesan *error*.
- B6 : mecoba koneksi TCP ke alamat IP yang memiliki pesan berhasil atas permintaan ICMP *echo*.
- B7 : upaya koneksi TCP ke alamat IP yang tidak pernah digunakan dalam DNS, NetBIOS, ICMP.

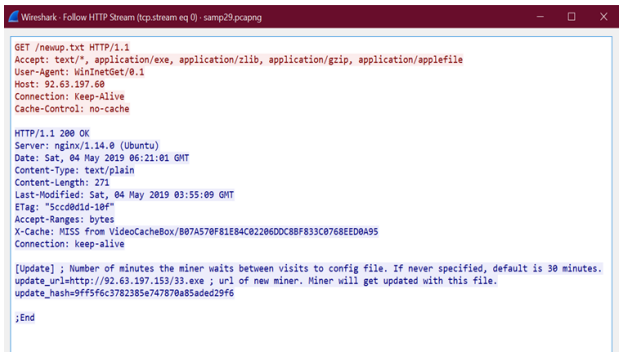


Gambar 3 Grafik *Malware Behavior*

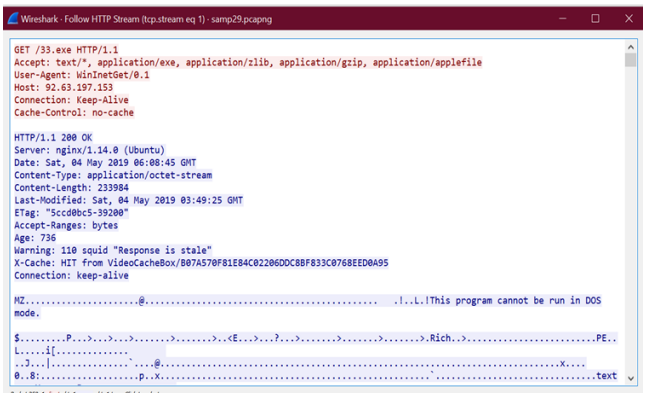
Dari gambar di atas, perilaku *malware* yang paling banyak terdeteksi adalah B1 atau perilaku pertama. Sebanyak 15 sampel *malware* terdeteksi pada B1. Hal ini dikarenakan setengah dari 30 sampel *malware* melakukan DNS *request* untuk memperoleh IP dari masing-masing *domain* yang ingin dihubungkan oleh *malware*.

b. *Payload*

Pada pengujian bagian ini, bertujuan untuk mengetahui aktivitas yang dilakukan oleh *malware* pada trafik jaringan. Berikut adalah pengujian dari salah satu sampel.



Gambar 4 Payload 1



Gambar 5 Payload 2

Gambar di atas merupakan HTTP *request* yang dilakukan oleh samp29.exe. Sampel tersebut mengakses dua buah *file* bernama *newup.txt* dan *33.exe*. Dari gambar 3, *payload* pertama adalah milik *newup.txt*. Di dalam payload tersebut terdapat perintah yang mengarahkan *client* agar mengakses *file 33.exe* sehingga pada aktivitas HTTP *request* terhadap *33.exe* pun terjadi. Gambar 4 merupakan *payload* yang dimiliki oleh *33.exe*. Pada *payload* tersebut, terdapat kode MZ pada awal *payload*. MZ merupakan kode pada Wireshark dengan fungsi untuk mengunduh sebuah *file* dengan format *executable*. Hal ini dapat dibuktikan dengan mengeksport file tersebut melalui menu *file – Export Object* – HTTP pada Wireshark.

c. *Bandwidth* Trafik Jaringan

Pada pengujian bagian ini, bertujuan untuk mengetahui dampak *malware* terhadap trafik jaringan terutama *bandwidth* pada trafik tersebut setelah terinfeksi *malware*. Untuk mengukur *bandwidth* dalam analisis ini, dilakukan uji coba *download* sebuah *file* dengan format *executable*. *File* ini berukuran 15.17 MB.

Untuk melakukan pengukuran *bandwidth*, data yang diambil adalah jumlah *Average bits/s*. Data ini menunjukan besarnya *bandwidth* yang digunakan pada saat mengunduh *file* yang telah disediakan. Selain itu, data pada *Measurement Bytes* menunjukkan besar ukuran data yang masuk pada trafik tersebut. Berikut adalah hasil pengujian *bandwidth*.

Tabel 2 Hasil Pengukuran *Bandwidth*

Sample	*Bytes*	*Kbits/s*
Normal Traffic	**17339072**	**4379**
Malware Traffic	17360366 - 22023405	1138 - 3716

Dapat dilihat, *bandwidth* trafik normal mencapai 4379000 *bits/s* atau 4,3 *Mbps*. Sedangkan *malware traffic* memiliki *bandwidth* antara 1138-3716 *Kbits/s*. Selain itu, data yang berhasil terekam pada trafik normal sebanyak 17339071 *Bytes*. Sedangkan data pada *malware traffic* memiliki ukuran yang lebih besar.

3. Dampak Terhadap Trafik Jaringan

Berdasarkan hasil analisis *malware* dengan API *network* dan trafik jaringannya, dapat diketahui bagaimana dampak *malware* terhadap trafik jaringan. Dampak yang dimuat pada tabel di bawah adalah dampak yang signifikan dari sampel *malware* dengan jenis yang berbeda

Tabel 3 Dampak *Malware* Terhadap Trafik Jaringan

Sampel	Dampak
Samp30.exe	Sampel ini bertujuan untuk membuat atau mencari kelemahan pada sistem kemudian mengakses *webpage* yang sudah terinfeksi *malware*. Akibatnya *bandwidth* mengalami penurunan kecepatan.
Samp29.exe	Sampel ini melakukan aktivitas jahatnya dengan melakukan pengunduhan sebuah *malware* yang berjenis *Trojan Chapak*. *Malware* jenis ini memiliki fungsi-fungsi Windows API lain yang menyerang sistem termasuk trafik jaringannya sehingga mengakibatkan penurunan *bandwidth*.
Samp28.exe	Sampel ini melakukan pengunduhan *bit-bit* yang diduga merupakan *malicious code.* Penurunan *bandwidth* dapat diakibatkan karena proses pengunduhan atau *malicious code* yang sudah menyerang sistem.
Samp27.exe	Sampel ini menciptakan sebuah *backdoor* yang terhubung dengan IP pada *payload* yang diduga sebagai pintu masuk *malicious file*. Dengan memanfaatkan *Windows Socket* API, sampel ini dapat mempengaruhi QoS trafik jaringan sehingga menyebabkan penurunan *bandwidth*.
Samp23.exe	Sampel ini dikategorikan sebagai sampel yang melakukan *mining* dengan *socket minelitecoin.com:8336* sebagai jalur koneksinya. Proses ini memakan *bandwidth* yang cukup banyak tetapi ukuran data yang terekam selama perekaman trafik tidak dapat dideteksi.

.

4. Kesimpulan

Berdasarkan penelitian Analisis Dampak *Malware* Berdasarkan API *call Network* Dengan Metode *Heuristic Detection*, dapat disimpulkan bahwa:

1. Analisis dampak *malware* terhadap trafik jaringan menggunakan teknik deteksi *behavior-based* dapat dilakukan dengan melihat perilaku *malware* melalui aktivitasnya. Aktivitas *malware* tersebut dapat diketahui dengan menyusun urutan-urutan API *network* yang digunakan *malware* dan menganalisis trafik jaringan yang sudah terinfeksi *malware*.
2. Aktivitas *malware* dapat diidentifikasikan dengan melihat urutan-urutan API *network* yang digunakan oleh *malware*. Urutan ini dapat dikelompokkan dengan melihat *Task* ID setiap proses yang sedang berjalan. Dalam satu proses membutuhkan beberapa fungsi API *network* sehingga aktivitas tidak dapat ditentukan dengan hanya mengidentifikasi masing-masing API *network.*
3. Aktivitas *malware* juga dapat ditentukan melalui analisis terhadap trafik jaringannya seperti analisis *payload, bandwidth,* dan *behavior* jaringan *malware*. Analisis *payload* bertujuan untuk melihat aktivitas suatu *malware* dengan mengecek fungsi-fungsi atau perilaku yang ditunjukkan selama pengujian. Analisis *bandwidth* bertujuan untuk memeriksa dampak *malware* terhadap trafik jaringan. Sedangkan analisis *behavior* jaringan *malware* bertujuan untuk mendeteksi apakah sampel tersebut dapat diketegorikan *malware* atau tidak.

Daftar Pustaka:

[1] I. A.Saeed, A. Selamat, and A. M. A. Abuagoub, "A Survey on Malware and Malware Detection Systems," *Int. J. Comput. Appl.*, 2013.

[2] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "2009__Scalable, Behavior-Based Malware Clustering, 2009.pdf," *Secur. Syst. Lab*, 2009.

[3] M. Jain and P. Bajaj, "Techniques in Detection and Analyzing Malware Executables: A Review," *Int. J. Comput. Sci. Mob. Comput.*, vol. 35, no. 5, pp. 930–935, 2014.

[4] J. Ismail, "Analisa Malware Metode Statik | Jul Ismail," 2016. [Online]. Available: https://julismail.staff.telkomuniversity.ac.id/analisa-malware-metode-statik/. [Accessed: 21-May-2019].

[5] N. Perdisci, Roberto and Lee, Wenke and Feamster, "Behavioral clustering of HTTP-based malware and signature generation using malicious network traces," *7th USENIX Conf. Networked Syst. Des. Implement.*, pp. 26--26, 2010.

[6] Y. S. Kim, E. Wang, and H. M. Rho, "Geometry-based machining precedence reasoning for feature-based process planning," *Int. J. Prod. Res.*, vol. 39, no. 10, pp. 2077–2103, 2001.

[7] G. Merialdo, "Medusa," *Rev. Medica Homeopat.*, vol. 5, no. 2, pp. 61–62, 2012.

[8] P. Mell, K. Kent, and J. Nusbaum, "Guide to malware incident prevention and handling recommendations of the national institute of standards and technology," *Nist Spec. Publ. 800-83*, p. 101, 2005.

[9] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Justindo*, vol. 2, no. 1, pp. 19–30, 2017.

[10] W. Utama, "Apa Itu Malware, Pengertian, Penjelasan dan Jenis Malware yang Perlu Diwaspadai," 2017. [Online]. Available: https://www.klikmania.net/apa-itu-malware. [Accessed: 21-May-2019].

[11] J. Aycock, *Computer Viruses and Malware*. Canada: Springer, 2006.

[12] M. Sikorski and A. Honig, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software," *Comput. Secur.*, 2012.

[13] P. V. Shijo and A. Salim, "Integrated static and dynamic analysis for malware detection," in *Procedia Computer Science*, 2015.

[14] W. Liu, P. Ren, K. Liu, and H. X. Duan, "Behavior-based malware analysis and detection," in *Proceedings - 2011 1st International Workshop on Complexity and Data Mining, IWCDM 2011*, 2011.

[15] G. Jacob, H. Debar, and E. Filiol, "Behavioral detection of malware: From a survey towards an established taxonomy," *J. Comput. Virol.*, vol. 4, no. 3, pp. 251–266, 2008.

[16] Webi, "Windows API Index - Windows applications | Microsoft Docs," 2018. [Online]. Available: https://docs.microsoft.com/en-us/windows/desktop/apiindex/windows-api-list#networking-and-internet. [Accessed: 21-May-2019].

[17] A. Orebaugh, G. Ramirez, J. Burke, L. Pesce, J. Wright, and G. Morris, *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. 2006.

[18] J. Jonker and B. Pennink, *The Essence of Research Methodology*. 2009.

[19] J. A. Morales, A. Al-Bataineh, S. Xu, and R. Sandhu, "Analyzing and exploiting network behaviors of malware," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng.*, vol. 50 LNICST, pp. 20–34, 2010.