ANALISIS DAMPAK *MALWARE* BERDASARKAN API *CALL* DENGAN METODE ANOMALI

MALWARE IMPACT ANALYSIS BASED ON API CALL USING ANOMALY METHOD

Naufal Abrian Ismiyushar[1], M. Teguh Kurniawan [2], Adityas Widjajarto [3]

[1,2,3]Prodi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom
[1]**naufalaismiyushar@student.telkomuniversity.ac.id,** [2]**teguh.kurniawan@telkomuniversity.ac.id,**
[3]**adtwjrt@telkomuniversity.ac.id**

Abstrak

Malware telah berkembang sangat pesat dan memiliki berbagai teknik untuk mengelabui *antivirus* saat menginfeksi komputer. Tujuan dari *attacker* atau pembuat *malware* ini adalah dapat memasang *malware* pada *device* tujuan dan mendapat kontrol penuh terhadap *device* tersebut. Dengan adanya ancaman kontrol penuh terhadap *device*, maka korban mengalami kerugian baik dari pencurian informasi, DDoS *attack*, penyalahgunaan komputer korban, *email spam* dan kerugian yang berkaitan lainnya. Dari berbagai ancaman *malware* yang memungkinkan terjadi, harus dilakukan penelitian untuk memahami *signature* dari suatu *malware*. *Malware analysis* sangatlah dibutuhkan untuk melakukan analisis dari segi *impact*, kategori dan ciri-cirinya. Sehingga dari hasil analisis yang dilakukan dapat disimpulkan bagaimana klasifikasi *malware*, deteksi *malware* dan mitigasi *malware*. Dengan melakukan *malware analyis* ada beberapa informasi yang didapat salah satunya pemanggilan API oleh *malware*. Kategorisasi *malware* dilakukan menggunakan *malicious activity data set* berdasarkan API *calls*. Semakin banyak keterkaitan antar *malicious activity* pada suatu *malware*, semakin besar juga dampak yang akan dihasilkan oleh *malware* tersebut. Dan sebaliknya, semakin sedikit keterkaitan *malicious activity* pada suatu *malware*, dampak yang dihasilkan akan kecil. Hasil dari kategorisasi dianalisis berdasarkan metode anomali. Berdasarkan kategorisasi menggunakan metode anomali, terdapat 3 (tiga) *malicious activity* yang tidak memiliki keterkaitan dengan sampel *malware* yang digunakan pada penelitian ini yaitu IAT *Hooking*, *Bind* TCP *Port* dan *Capture Network*. Terdapat juga 2 (dua) *malicious activity* yang hanya memiliki 1 keterkaitan dengan sampel *malware* yang digunakan yaitu *Process Hollowing* dan *Drop Files from* PE *Resource Section*.

Kata kunci : *malware, malware analysis, cyber crime,* anomali, deteksi *malware, malware signature, malicious activity*.

Abstract

Malware has evolved very fast and has various techniques to deceive antivirus when infect computer. The purpose from attacker or malware maker is installing malware on the victim's device and get full control of the device. As is the threat of full control of the device, the victim will suffer a lot of losses from infomation theft, DDoS attack, abuse of victim computer, email spam, and other related losses. From a variety of possible malware threat that will happen, research must be done to understand the signature of a malware. Malware analysis is needed to perform analysis to malware in terms of impact, category, and characteristics. From the result of the analysis can be concluded how is the classification of malware, malware detection and malware mitigation. By performing malware analyis there is some information about malware API calls. Malware categorization is done using malicious activity data set based on API calls. The more links between malicious activity on a malware, the greater the impact that malware will perform. And conversely, the less links between malicious activity on a malware, the impact will be small. The results of categorization then analyzed using anomaly method. Based on the categorization using anomaly method with the sample of malware, there are 3 malicious activity that do not have any links with the malicious activity, that are IAT Hooking, Bind TCP Port and Capture Network. There are also 2 malicious activities that have only 1 link with the malware samples used. That are Process Hollowing and Drop Files from PE Resource Section.

Keywords: malware, malware analysis, cyber crime, anomaly, malware detection, malware signature, malicious activity.

1. Pendahuluan

Cyber-crime adalah tindak kriminal yang dilakukan dalam lingkup teknologi, terutama dalam perkembangan komputer dan *Internet. Cyber-crime* sangatlah luas jika dijelaskan, salah satu contohnya adalah serangan *malicious software* atau yang biasa disebut dengan *malware*. *Malware* adalah *software* atau program komputer yang digunakan untuk melakukan aktivitas yang sifatnya merusak dan mengganggu (Zeltser, 2014). Tujuan dari *attacker* atau pembuat *malware* ini adalah dapat memasang *malware* pada target tujuan dan mendapatkan kontrol penuh terhadap *device* tersebut (Zeltser, 2014).

Pada umumnya *malware* disebarkan menggunakan berbagai cara, berikut contohnya adalah *social engineering attack*, *email phishing* dan *file download fraud* untuk membuat korban tertipu dan sistemnya terinfeksi *malware.* Target dari penyebaran *malware* ini adalah pencurian data rahasia, mencari *username* dan *password*, DDoS, dan *spam email* (Zeltser, 2014). *Impact* dari suatu *malware* terhadap korban atau organisasi adalah gangguan pada *service* yang menyebabkan hilangnya produktivitas dan dapat menyebabkan hilangnya pendapatan (MS-ISAC, 2005).

Malware analysis adalah salah satu cara untuk melakukan pencegahan dari tersebarnya sebuah *malware*. Analisis *malware* mempermudah untuk mengenali *signature* suatu *malware* dan dibantu dengan kategorisasi dengan analisis *anomaly* yang akan membuat lebih mudah untuk melakukan kategorisasi. Keuntungan dari penerapan metode anomali adalah deteksi suatu pola akan lebih mudah dan lebih cepat ditemukan. Metode anomali banyak digunakan sebagai penanganan serangan secara preventif dari suatu *malware* (Matt Bromiley, 2016). Dari sampel yang digunakan dapat ditentukan apakah suatu sampel tersebut tidak termasuk *malware* atau termasuk *malware* yang berbahaya dengan menggunakan bantuan metode anomali. Selain itu, tujuan dilakukannya *malware analysis* adalah untuk memahami bagaimana suatu *malware* bekerja (Sikorsi & Honig, 2012). Dalam analisis *malware*, dapat diperoleh informasi mengenai *behaviour* suatu *malware*, serangan yang dilakukan *malware*, hingga API *call* apa saja yang digunakan oleh suatu *malware*. Ketiga informasi tersebut dapat digunakan sebagai parameter untuk melakukan kategorisasi *malware.*

Solusi untuk melakukan kategorisasi *malware* pada penelitian ini adalah dengan menggunakan API *call* yang digunakan oleh *malware*. Dengan mengumpulkan informasi API *call* dari beberapa sampel *malware*, dapat dilakukan kategorisasi menggunakan *data set* yang menggunakan parameter API *call*. Setelah dilakukan kategorisasi, dapat disimpulkan *impact* sampel *malware* terhadap performansi suatu komputer yang terinfeksi. Penentuan *impact* tersebut juga dapat dilakukan dengan menggunakan sampel *malware* di luar penelitian ini. Sehingga ketika terdapat beberapa *malware* yang melakukan infeksi pada suatu komputer, penelitian ini dapat digunakan untuk mengetahui aktivitas yang dilakukan oleh *malware* tersebut.Keuntungan dari penerapan metode anomali adalah deteksi suatu pola akan lebih mudah dan lebih cepat ditemukan. Metode anomali banyak digunakan sebagai penanganan serangan secara preventif dari suatu *malware*.

2. Dasar Teori

2.1 *Malicious Software (Malware)*

Malware adalah perangkat lunak yang dibuat dan digunakan untuk melakukan perusakan sistem, pencurian atau pengumpulan informasi, hingga mendapatkan akses terhadap suatu komputer. *Malware* adalah perangkat lunak yang berupa *script, active content* dan *binary* (Sikorski & Honig, 2012).

2.2 *Malware Analysis*

Malware analysis adalah sebuah seni untuk membedah atau menganalisis *malware* (Sikorski & Honig, 2012). *Malware analysis* mempelajari bagaimana cara *malware* tersebut bekerja, fungsionalitas *malware* dan bagaimana cara pendeteksian serta pencegahannya yang paling efektif terhadap suatu *malware* (Sikorski & Honig, 2012). Untuk melakukan *malware analysis* terdapat dua metode proses analisis yaitu *static analysis* dan *dynamic analysis* (Zalavadita & Sharma, 2007).

2.3 *Static Analysis*

Static analysis dapat diartikan dengan analisis sebuah perangkat lunak sebelum dijalankan yaitu dengan kata lain *static analysis* dijalankan saat *pre-execution time*. Dengan adanya *static analysis* ini, penganalisis dapat memahami *source code* dari sebuah perangkat lunak. *Static analysis* memiliki teknik analisis yaitu *signature based* dan *heuristic* (Zalavadita & Sharma, 2007).

2.4 *Dynamic Analysis*

Dynamic analysis adalah proses menganalisis *malware* saat sedang dijalankan. Saat dilakukan proses ini, sebuah *malware sample* akan dijalankan pada sebuah *virtual machine* untuk dianalisis lebih dalam dari segi fungsionalitasnya dan *bug* saat aplikasi berjalan (Zalavadita & Sharma, 2007).

2.5 API *Calls*

API (*Application Programming Interface)* adalah kumpulan beberapa perintah yang digunakan untuk memprogram aplikasi. Pemanggilan API *calls* pada *malware* merefleksikan fungsionalitas *malware* dan dapat digunakan untuk memahami *behavior* dari suatu *malware.* Analisis API *call* adalah salah satu cara yang efektif untuk melakukan analisis *malware* (Alquraishi & Batarfi, 2017).

2.6 Kategorisasi Berdasarkan Metode Anomali

Anomaly adalah sebuah pola dari kumpulan data yang tidak memiliki pola yang normal yaitu berbeda dengan data yang lainnya. Pola yang berbeda ini yang disebut dengan anomali atau *outliers* (Chandola, Banerjee & Kumar, 2009). Metode anomali dapat menemukan jenis *malware* baru karena *malware* tersebut memiliki *behavior* yang terdeteksi sebagai hal yang unik atau berbeda dengan *malware* yang lainnya.

2.7 Kategorisasi Berdasarkan Metode Anomali

Metode anomali memiliki kelebihan dan kekurangan dalam proses pengelompokannya (Bromiley, 2016).

Kelebihan metode anomali adalah :

- Memajukan pemahaman mengenai *environment* yang digunakan.
- Memberikan kesempatan yang lebih baik dalam menangkap penyerang sebelum melakukan serangan lebih jauh.
- Mempersiapkan cara untuk menangani hal yang tidak terduga.

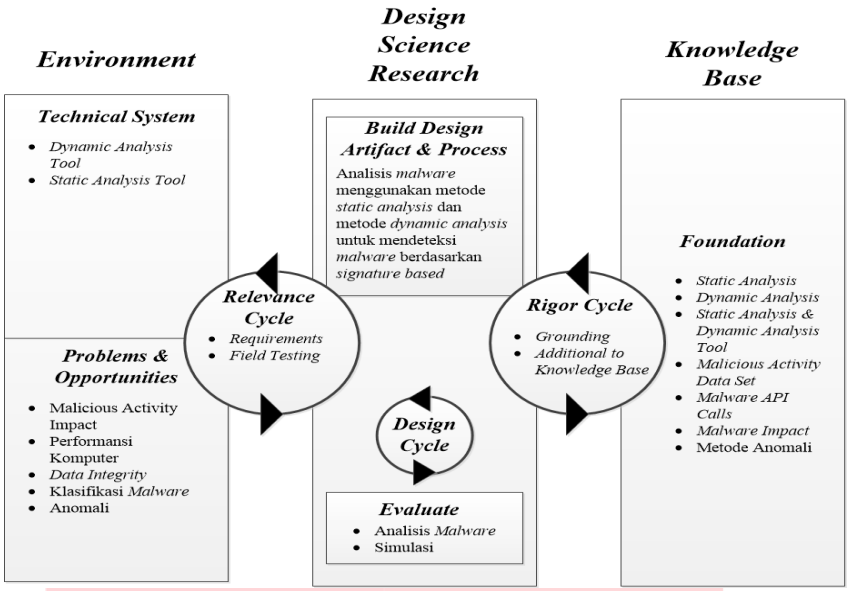
Kekurangan metode anomali adalah :

- Dapat terhambat dalam urusan sumber daya intensif.
- Proses manual yang dilakukan pada metode anomali dapat memperlambat proses yang dilakukan oleh sumber daya yang belum terlatih.
- Membutuhkan ruang lingkup yang luas.

3 Metodologi Penelitian

3.1 Model Konseptual

Model konseptual adalah gambaran untuk memahami, melaksanakan dan mengevaluasi penelitian sistem informasi. Model konseptual juga biasa disebut dengan kumpulan dari strategi yang terbaik dari berbagai strategi yang akan digunakan (Oren Ole). Tujuan dari model konseptual adalah sebagai kerangka secara terstruktur yang digunakan untuk memahami tujuan dari penelitian. Model konseptual yang digunakan dalam penelitian ini adalah sebagai berikut :



Gambar 1 Model Konseptual

4 Perancangan *software* dan *hardware*

4.1 Skenario Pengujian

Klasifikasi yang dilakukan dalam penelitian ini menggunakan *malicious activity data set*. Berikut adalah tabel acuan yang digunakan (Pektas & Acarman, 2017).

Tabel 1 Malicious Activity Data Set

Nomor	*Malicious Activity*	*Malicious* API *Call*
1	*Process Hollowing*	*CreateProcessInternalW* *GetModuleHandle* *GetProcAddress* *VirtualAllocEx* *WriteProcessMemory* *SetThreadContext* *NtResumeThread*
2	*Create Remote Thread*	*NtOpenProcess* *GetModuleHandle* *GetProcAddress* *VirtualAllocEx* *WriteProcessMemory* *CreateRemoteThread*
3	*Enumerating All Processes*	*CreateToolhelp32Snapshot* *Process32FirstW* *Process32NextW*

Nomor	*Malicious Activity*	*Malicious* API *Call*
4	*Drop Files from* PE *Resource Section*	*GetModuleHandle* *FindResource* *LoadResource* *NtCreateFile*
5	IAT *Hooking*	*GetModuleHandle* *Strcmp* *VirtualProtect*
6	*Delete Itself*	*GetModuleFileName* *ExitProcess* *DeleteFileW*
7	*Download & Execute* PE *Files*	*URLDownloadToFile* *ShellExecuteExW*
8	*Bind* TCP *Port*	*WSAStartup* *Socket*
9	*Capture Network*	*Socket* *Bind* *WSAIoctl* *recvfrom*

5 Pengujian Sistem dan Analisis

5.1 Kategorisasi Berdasarkan *Malicious Activity Data Set*

Berikut adalah hasil *filter* semua API *call* menggunakan *malicious activity data set*.

Tabel 2 Kategorisasi Malware Berdasarkan Malicious Activity

Malicious Activity	*Total*	*List Malware*
Process Hollowing	62 *Malware*	*Malware* 151, *Malware* 153, *Malware* 122, *Malware* 113, *Malware* 127, *Malware* 115, *Malware* 62, *Malware* 116, *Malware* 64, *Malware* 65, *Malware* 66, *Malware* 112, *Malware* 69, *Malware* 138, *Malware* 119, *Malware* 118, *Malware* 165, *Malware* 24, *Malware* 25, *Malware* 21, *Malware* 22, *Malware* 49, *Malware* 47, *Malware* 45, *Malware* 137, *Malware* 130, *Malware* 3, *Malware* 2, *Malware* 5, *Malware* 4, *Malware* 7, *Malware* 6, *Malware* 8, *Malware* 144, *Malware* 143, *Malware* 140, *Malware* 149, *Malware* 120, *Malware* 121, *Malware* 108, *Malware* 109, *Malware* 124, *Malware* 72, *Malware* 126, *Malware* 70, *Malware* 129, *Malware* 166, *Malware* 167, *Malware* 125,

Malicious Activity	*Total*	*List Malware*
		Malware 11, *Malware* 133, *Malware* 33, *Malware* 57, *Malware* 50, *Malware* 53, *Malware* 52, *Malware* 55, *Malware* 131, *Malware* 123, *Malware* 162, *Malware* 111, *Malware* 110
Create Remote Thread	27 *Malware*	*Malware* 115, *Malware* 117, *Malware* 116, *Malware* 111, *Malware* 65, *Malware* 113, *Malware* 112, *Malware* 69, *Malware* 119, *Malware* 118, *Malware* 121, *Malware* 122, *Malware* 123, *Malware* 124, *Malware* 125, *Malware* 126, *Malware* 127, *Malware* 129, *Malware* 167, *Malware* 162, *Malware* 38, *Malware* 19, *Malware* 57, *Malware* 55, *Malware* 64, *Malware* 33, *Malware* 110
Enumerating All Processes	24 *Malware*	(*Malware* 151, *Malware* 153, *Malware* 131, *Malware* 130, *Malware* 137, *Malware* 138, *Malware* 49, *Malware* 45, *Malware* 3, *Malware* 2, *Malware* 4, *Malware* 7, *Malware* 6, *Malware* 8, *Malware* 144, *Malware* 140, *Malware* 148, *Malware* 149, *Malware* 72, *Malware* 166, *Malware* 11, *Malware* 33, *Malware* 50, *Malware* 53
Drop Files from PE Resource Section	58 *Malware*	*Malware* 151, *Malware* 122, *Malware* 113, *Malware* 127, *Malware* 50, *Malware* 115, *Malware* 62, *Malware* 116, *Malware* 64, *Malware* 65, *Malware* 66, *Malware* 112, *Malware* 68, *Malware* 69, *Malware* 138, *Malware* 119, *Malware* 118, *Malware* 165, *Malware* 24, *Malware* 25, *Malware* 26, *Malware* 20, *Malware* 49, *Malware* 47, *Malware* 45, *Malware* 5, *Malware* 162, *Malware* 147, *Malware* 142, *Malware* 143, *Malware* 117, *Malware* 148, *Malware* 120, *Malware* 121, *Malware* 108, *Malware* 109, *Malware* 124, *Malware* 72, *Malware* 71, *Malware* 70, *Malware* 129, *Malware* 167, *Malware* 125, *Malware* 38, *Malware* 55, *Malware* 19, *Malware* 18, *Malware* 57, *Malware* 36, *Malware* 53, *Malware* 52, *Malware* 33, *Malware* 131, *Malware* 123, *Malware* 32, *Malware* 111, *Malware* 126, *Malware* 110
IAT Hooking	0 *Malware*	-
Delete Itself	13 *Malware*	*Malware* 151, *Malware* 26, *Malware* 20, *Malware* 49, *Malware* 33, *Malware* 57, *Malware* 45, *Malware*

Malicious Activity	*Total*	*List Malware*
		50, *Malware* 55, *Malware* 64, *Malware* 65, *Malware* 162, *Malware* 72
Download & Execute PE Files	10 *Malware*	*Malware* 55, *Malware* 47, *Malware* 57, *Malware* 120, *Malware* 108, *Malware* 109, *Malware* 65, *Malware* 70, *Malware* 129, *Malware* 167
Bind TCP Port	0 *Malware*	-
Capture Network	0 *Malware*	-

5.2 Keterkaitan Antar *Malicious Activity*

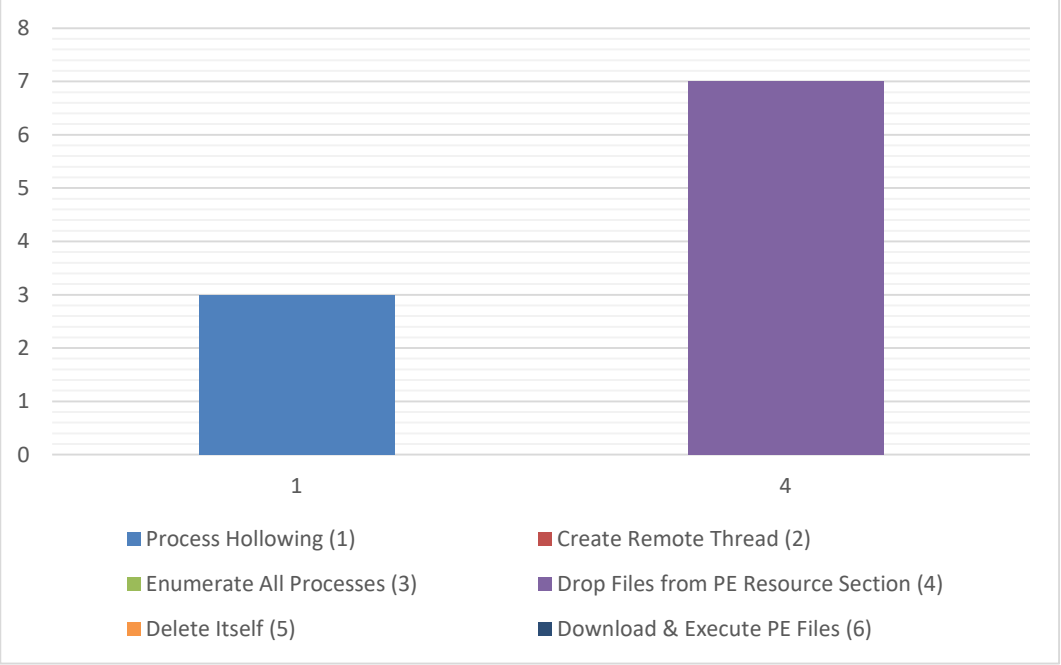
Keterkaitan *malicious activity* dari data yang diperoleh pada kategorisasi mempengaruhi *impact* dari suatu *malware*. *Impact* yang diberikan pada analisis ini adalah *impact* rendah (*low)*, menengah *(medium)* dan tinggi (*high)*. Berikut adalah keterkaitan *malicious activity* dengan parameter jumlah keterkaitannya.

5.2.1 *Malicious Activity Malware* Dengan 0 Kombinasi

Terdapat 3 *malicious activity* yang tidak termasuk dalam kategorisasi dari sampel *malware* yang digunakan. Dari 170 sampel *malware*, terdapat 0 (nol) *malware* yang termasuk dalam kategori yang tidak digunakan yaitu IAT *hooking*, *bind* TCP *port* dan *capture network*. Hal ini terjadi selaras dengan pengujian yang dilakukan, dimana analisis *malware* ini hanya menggunakan 3 (tiga) API *calls* yaitu API *process*, API *file* dan API *registry*. Sedangkan ketiga *malicious activity* tersebut sudah dipastikan menggunakan API *network*.

5.2.2 *Malicious Activity Malware* Dengan 1 Kombinasi

Terdapat 2 (dua) *malicious activity* yang termasuk dalam kategorisasi dari sampel *malware* yang digunakan yaitu *process hollowing* dan *drop files from* PE *resource section.* Berikut adalah *chart* mengenai jumlah *malware* yang termask dalam kategorisasi diatas.



Gambar 2 Grafik 1 Kombinasi Malicious Activity

Pada gambar 14 dapat dilihat terdapat total 10 (sepuluh) *malware* dengan pembagian pada *process hollowing* terdapat 3 (tiga) *malware* dan pada *drop files from* PE *resource section* terdapat

7 (tujuh) *malware*.

5.3 Analisis Hasil Menggunakan Metode Anomali

Dari hasil analisis *malware* yang diperoleh. Terdapat kategorisasi *malware* dengan jumlah 0 (nol) hingga 5 (lima) *malicious activity*. Metode anomali adalah pola unik atau bisa disebut *outliers* (Chandola, Banerjee & Kumar, 2009). Pada penelitian ini, pola yang berbeda berdasarkan parameter *malicious activity* yang terkait adalah kategorisasi dengan jumlah 0 (nol) dan 1 (satu) *malicious activity*.

Tidak ada *malware* yang termasuk dalam IAT *hooking, bind TCP port* dan *capture network.* Hal ini termasuk anomali yang sudah dapat terjawab mengapa bisa terjadi. Pada *malicious activity* dengan kategori IAT *hooking* tidak ada satupun *malware* yang tergolong pada kategori tersebut. Pada umumnya IAT *hooking* banyak menggunakan API *GetModuleHandle* dimana dari hasil analisis 170 sampel *malware* tidak ada yang menggunakan API tersebut dan juga IAT *hooking* memiliki hubungan dengan *network application*. Sedangkan *malicious activity* dengan kategori *bind TCP port* dan *capture network* merupakan kategori khusus untuk *malware* yang menginfeksi jaringan. API *calls* yang digunakan dalam pengujian *analisis malware* ini adalah API *process,* API *file* dan API *registry* dimana ketiga API tersebut tidak ada hubungannya dengan jaringan. Apabila parameter API ditambah dengan API *network,* tidak menutup kemungkinan ada beberapa *malware* yang termasuk dalam kategori IAT *hooking, bind TCP port* maupun kategori *capture network.*

Terdapat *malware* dengan 1 (satu) *malicious activity* dengan jumlah 3 (tiga) *malware* pada kategori *process hollowing* dan 7 (tujuh) *malware* pada kategori *drop files from PE resource section*. *Malware* dengan kategori *process hollowing* melakukan injeksi sebuah *shellcode* pada suatu proses memori. Hal ini berbahaya jika menginfeksi proses kritikal dan tidak terlalu berbahaya jika hanya menginfeksi proses biasa. Namun, apabila *process hollowing* digabungkan dengan *create remote thread* dan *enumerating all process*. Dimana masing masing fungsi nya adalah untuk melakukan penyebaran dan meningkatkan skalabilitas serangan, dapat sangat berbahaya karena akan menginfeksi semua proses yang diperoleh. Sementara itu, *malware* dengan kategori *drop files from PE resource section* berfungsi untuk melakukan modifikasi *file* dan sejenisnya. Kategori ini tidak begitu berbahaya jika dibandingkan dengan *process hollowing* karena dalam performansi, *file* lebih rendah pengaruhnya jika dibanding dengan proses.

Kategorisasi berdasarkan metode anomali pada penelitian ini memiliki *impact* sedang (*medium*). Karena apabila suatu *malware* hanya tergolong dalam satu kategori saja, serangannya akan tidak maksimal dan tidak menutup kemungkinan *malware* tersebut tidak bisa berjalan dengan semestinya.

5.4 Rekomendasi Hasil Analisis *Malware* Yang Termasuk Kategori Anomali

Dari hasil analisis *malware* dengan metode anomali. Terdapat pengelompokan yang termasuk yaitu 0 (nol) kombinasi dan 1 (satu) kombinasi *malicious activity*.

Rekomendasi ini dituliskan menggunakan pemetaan informasi setiap *malware* dengan bantuan aplikasi *Virustotal*. Fitur yang digunakan adalah pemetaan *behavior* dari suatu *malware* yang di dalamnya terdapat informasi mengenai *network communication, file system actions, registry actions, process and service actions, synchronization mechanisms & signals*, *modules loaded* dan *highlighted actions*. Berikut pada Tabel 29 adalah rekomendasi untuk mencegah *malware* yang termasuk dalam kategori anomaly untuk melakukan *process hollowing* dan *drop files from PE resource section*.

Tabel 3 List Rekomendasi Malicious Activity

No.	*Malicious Activity*	Rekomendasi
1.	*Process Hollowing*	Melakukan pencarian dan menghentikan proses yang terinfeksi oleh suatu *malware* dengan bantuan aplikasi *ProcessExplorer* dan *ProcessMonitor.* Tindakan preventif yang dilakukan dapat menggunakan bantuan *antivirus.*
2.	*Drop Files from PE Resource Section*	Melakukan pencegahan dengan *antivirus* dan memberikan edukasi terhadap *end user* untuk

No.	*Malicious Activity*	Rekomendasi
		tidak melakukan *download* dari situs yang tidak dapat dipercaya. Apabila telah terinfeksi dapat dilakukan penghapusan *file malware.*

6 Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Untuk melakukan analisis *malware* pada sistem operasi Windows, dapat dilakukan dengan menggunakan beberapa *environment* yang dijadikan sebagai *sandbox.* Penelitian ini menggunakan sistem operasi Windows pada Vmware sebagai *sandbox* supaya tidak terjadi kesalahan infeksi *malware* pada sistem operasi utama yang digunakan. Utamanya saat melakukan *dynamic analysis* dimana *malware* dianalisis saat sedang dieksekusi.
2. Penentuan kategori *malware* pada penelitian ini dilakukan dengan mencari *data set* untuk mengkategorikan *malware.* Dalam proses analisis *malware,* penelitian ini mencari informasi mengenai API *process,* API *file* dan API *registry.* Dari data yang diperoleh, kategorisasi dapat dilakukan dengan menggunakan *malicious activity data set* dimana parameter tersebut menggunakan parameter API *calls.*
3. Strategi deteksi *malware* pada penelitian ini, dapat dilakukan dengan melihat keterkaitan antar *malware* pada setiap *malicious activity.* Dari data yang diperoleh, akan menghasilkan gambaran kategori dari 170 sampel *malware* yang digunakan. Dari keterkaitan *malware* pada setiap *malicious activity* juga dapat diperoleh *impact* yang akan terjadi apabila *malware* tersebut menginfeksi suatu komputer.
4. Apabila *malware* telah dideteksi *impactnya,* rekomendasi yang diberikan yaitu mengenali kategori suatu *malware* dari dampak terhadap performansi komputer. Sebagai contoh proses, memori, *file* dan system I/O. Rekomendasi lainnya adalah dengan mencari solusi dari pemetaan serangan *malware* terhadap komputer yang terinfeksi.

Daftar Pustaka:

- Zeltser, L. (2014). *What is Malware*. The SANS Institute.
- Sikorski, M. & Honig, A. (2012). *Practical Malware Analysis*. San Francisco, USA.
- Zalavadita, N. & Dr. Sharma, Priyanka. (2007). *A Methodology of Malware Analysis, Tools and Technique for windows platform – RAT Analysis.* Ahmedabad, India
- Chandola, V., Banerjee, A. & Kumar, V. (2009). *Anomaly Detection: A Survey*. ACM Comput. Surv. 41, 3, *Article* 15(*July* 2009), 58 *pages*. DOI = 10.1145/1541880.1541882
- *Malware Threats and Mitigation Strategies.* (2005). *Multi-State Information Sharing and Analysis Center and United States Computer Emergency Readiness Team*. Albany NY.
- Oren Ole. (2015). *A Method For Optimization Of A Conceptual Model.* Central Institute for Industrial Research Forskningav, Norway.
- Das Malwerk. (2016). Diakses pada 05 Desember 2017, dari http://dasmalwerk.eu/.
- Microsoft. (2018) Windows API *Index*. Diakses pada 25 Januari 2018, dari https://docs.microsoft.com/en-us/windows/desktop/apiindex/windows-api-list.
- Fortuna, Andrea. (2017, Oktober 9). *Understanding Process Hollowing*. Diakses pada 5 Februari 2018, dari https://www.andreafortuna.org/cybersecurity/understanding-process-hollowing/.
- Tigzy. (2014, Oktober 15). *Import Address Table* (IAT). Diakses pada 5 Februari 2018, dari https://www.adlice.com/userland-rootkits-part-1-iat-hooks/.
- Greenberg, Adam. (2015, Maret 11). *Self-deleteing Malware Targets Home Routers to Gather Information.* Diakses pada 5 Februari 2018, dari https://www.scmagazine.com/malware-that-connects-to-home-routers-deletes-itself-without-a-trace/article/536538/.
- Anupama, Bindu. (2007, November 6). *Know Your TCP System call sequences.* Diakses pada 5 Februari 2018, dari https://www.ibm.com/developerworks/aix/library/au-tcpsystemcalls/index.html.

- Pektas, A., Acarman, T. (2017). *Malware Classification Based on API Calls and Behaviour Analysis.* Istanbul, Turki.
- Dolly Upal,Vishakha Mehra & Vinod Verma. (2014). *Basic Survey on Malware Analysis, Tools and Techniques*,India.
- Dipjoyoti Deka, Nityananda Sarma, Nithin J.Panicker. *Malware Detection Vectors and Analysis Tehniques*: *A Brief Survey*,India.
- Ehab M.Alkhateeb, (2017) .*Dynamic Malware Detection using API Similarity*,United Arab Emirates.
- Ismahani Ismail, (2010).*Detecting Worms Using Data Mining Technique Learning in the Presence of Class Noise*,Malaysia.
- Yunan Zhang & Qingjia Huang, (2017). *Based on Multi-Features and Clustering Ensemble Method for Authomatic Malware Categorization*.
- Matt Bromiley, (2016). *Keys to Effective Anomaly Detection*, The SANS Institute, Singapore.
- Alquraishi, S., Batarfi, O. (2017). *A Comparison Between* API *Call Sequences and Opcode Sequences as Reflectors of Malware Behavior.* King Abdulaziz University. Jeddah, Saudi Arabia.
- Shijo, P. V., Salim, A. (2014). *Integrated Static and Dynamic Analysis for Malware Detection.* College of Engineering Trivandrum. India.