Analisis Dampak *Malware* Berdasarkan Api *Call Network* Dengan Metode *Heuristic Detection*

Impact Analysis Of Malware Based On Call Network Api With Heuristic Detection Method

One Tika Suryati[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]onetika@student.telkomuniversity.ac.id, [2]avonbudi@telkomuniveristy.co.id,
[3]ahmadalmaarif@telkomuniversity.ac.id

Abstrak

***Malware* adalah sebuah program yang memiliki pengaruh negatif pada sistem komputer yang tidak memiliki *user permission*. Tujuan dari dibuatnya *malware* oleh para peretas ialah mendapatkan keuntungan dengan cara yang tidak sah. Oleh karena itu diperlukan suatu *malware analysis*. *Malware analysis* bertujuan untuk mengetahui spesifik dari *malware* sehingga dapat dibangun keamanan untuk melindungi perangkat komputer. Salah satu metode untuk menganalisis *malware* adalah *heuristic detection*. *Heuristic detection* merupakan metode analisis yang memungkinkan untuk menemukan *malware* jenis baru pada suatu *file* atau aplikasi. Banyak *malware* yang dibuat untuk menyerang melalui jaringan internet karena adanya kemajuan teknologi. Berdasarkan kondisi tersebut, maka dilakukanlah *malware analysis* menggunakan API *call network* dengan metode *heuristic detection*. Hal ini bertujuan untuk mengidentifikasi perilaku dari *malware* yang menyerang jaringan. Hasil analisis yang dilakukan adalah kebanyakan *malware* bersifat sebagai spyware, yaitu mengintai aktivitas *user* serta mengambil data *user* tanpa sepengetahuan *user*. Selain itu, terdapat pula *malware* yang bersifat sebagai *adware*, yaitu menampilkan iklan melalui jendela *pop-up* pada perangkat komputer yang mengganggu aktivitas *user*. Sehingga dengan adanya hasil tersebut, dapat diidentifikasi pula tindakan yang dapat dilakukan oleh *user* untuk melindungi perangkat komputernya, seperti dengan memasang *antivirus* atau *antimalware*, tidak mengunduh aplikasi yang tidak sah serta tidak mengakses *website* yang tidak aman.**

Kata kunci : ***Malware*, *malware analysis*, *heuristic detection*,** **API** ***call network*.**

Abstract

Malware is a program that has a negative influence on computer systems that do not have user permissions. The purpose of making malware by hackers is to get profits in an illegal way. Therefore we need a malware analysis. Malware analysis aims to determine the specifics of malware so that security can be built to protect computer devices. One method for analyzing malware is heuristic detection. Heuristic detection is an analytical method that allows finding new types of malware in a file or application. Many malware is made to attack through the internet because of technological advancements. Based on these conditions, the malware analysis is carried out using the API call network with the heuristic detection method. This aims to identify the behavior of malware that attacks the network. The results of the analysis carried out are that most malware is spyware, which is lurking user activity and retrieving user data without the user's knowledge. In addition, there is also malware that is adware, which displays advertisements through pop-up windows on computer devices that interfere with user activity. So that with these results, it can also be identified actions that can be taken by the user to protect his computer device, such as by installing antivirus or antimalware, not downloading unauthorized applications and not accessing unsafe websites.

Keywords: ***Malware*, *malware analysis*, *heuristic detection*,** **API** ***call network*.**

1. Pendahuluan

Internet memiliki peran penting dalam semua bidang pada masyarakat baik dari segi ekonomi hingga pemerintahan. Namun dengan adanya perkembangan internet yang pesat, membuat sistem keamanan pada internet maupun PC *user* harus lebih ditingkatkan. Salah satu ancaman dari sistem keamanan adalah adanya *cyber crime*. *Cyber crime* adalah berbagai macam kejahatan yang ilegal atau terlarang oleh suatu individu atau keompok terhadap perangkat komputer, teknologi informasi jaringan, dan tindakan menargetkan suatu individu pada dunia internet [1]. Semakin berkembangnya *malware* saat ini, maka diperlukannya *malware analysis*. *Malware analysis* berguna untuk melihat bagaimana *malware* bekerja dan melihat sifat dari *malware* tersebut. Pada penelitian ini, *malware analysis* yang digunakan adalah metode statis dengan teknik *heuristic detection*. Kelebihan dari menggunakan metode statis adalah lebih cepat dan aman karena akan mengumpulkan struktur *malware* dari kode

program yang dimlikinya [2]. *Malware analysis* dengan *heuristic detection* menggunakan informasi dari API *call*. API *call* adalah suatu prosedur, protokol dan alat untuk membangun suatu aplikasi. Informasi dari API *call* akan digunakan untuk mengetahui aktivitas dari *malware*, sehingga informasi tersebut akan digunakan untuk mengidentifikasikan *malware* menggunakan teknik *heuristic detection*. *Heuristic detection* merupakan teknik yang mencari atau mendeteksi *malware* dengan mencari perintah atau instruksi yang tidak ada pada aplikasi yang dimana ini akan lebih mudah dalam mendeteksi jenis *malware* yang belum ditemukan atau diketahui sebelumnya [2]. Tujuan dari *malware analysis* diantaranya juga untuk mengetahui karakteristik dari *malware* dan target yang akan diserang oleh *malware* tersebut [3].

Berdasarkan data tersebut, untuk mengklasifikasikan jenis *malware* penulis melakukan penelitian terkait analisis *malware* dan pengklasifikasiannya dengan melakukan simulasi pada *virtual machine* dengan menggunakan teknik *heuristic detection*. Maka dari itu hasil dari penelitian ini berupa hasil analisa dan klasifikasi *malware* dengan teknik *heuristic detection*.

2. Dasar Teori dan Sistematika Penelitian

2.1 Penjelasan Malware

Malware merupakan sebuah program yang memiliki pengaruh negatif pada suatu sistem komputer yang tidak memiliki *user permission* merujuk pada *malware* [4]. *Malware* biasanya dikembangkan oleh orang-orang yang tidak beratanggung jawab seperti, penipu, pemeras, pengacau ataupun penjahat lain yang memiliki tujuan utama untuk mendapatkan uang secara tidak sah [5]. Tindakan yang biasanya dilakukan oleh *malware* ketika telah terinstall atau masuk ke dalam suatu sistem diantaranya adalah [5]:

1. Membanjiri suatu sistem komputer atau *web browser* dengan iklan.
2. Membelah diri dan menyerang *file* ataupun sistem lain.
3. Melakukan *install* aplikasi yang memicu *malware* untuk bekerja tanpa sepengetahuan *user* yang berdampak pada kinerja komputer.
4. Mengunci *file* atau sistem operasi dari komputer sehingga tidak dapat digunakan dan memaksa *user* untuk melakukan pembayaran agar dapat mengakses *file* atau sistem operasi tersebut kembali.

Berbeda jenis *malware*, maka berbeda pula langkah atau tindakan yang harus dilakukan untuk menghapus *malware* tersebut. Menghindari tautan yang mencurigakan, mengunjungi situs *website* yang tidak aman, merupakan salah satu cara untuk mencegahnya suatu komputer terinfeksi *malware*. Berikut adalah langkah dasar dalam menghapus *malware* [5]:

1. Sebelum menghapus *malware*, lakukan *backup* pada seluruh *file*.
2. Matikan koneksi internet.
3. Keluarkan CD atau DVD dari *disk drive* dan cabut USB dari komputer.
4. *Scanning* komputer dengan mode normal dan mode aman.
5. *Restart* komputer dan tahan F8 sebelum sistem operasi masuk ke BIOS.
6. Pilih *Advanced Option* > *Startup Settings* > *Safe Mode with Networking* > *Enter*.

2.2 Klasifikasi Malware

Menurut[2], malware dapat diklasifikasikan menjadi 3, yaitu:

1. *The Contagious Threat*
   a. *Virus*
      Virus memiliki peran dalam mengambil kontrol yang tidak sah dan dapat menyebab kerusakan tanpa sepengetahuan user. *Virus* biasanya menempel pada sebuah *file* yang dapat menyebarkan infeksi dari satu komputer ke komputer lain yang menyebabkan penurunan kinerja pada komputer.
   b. *Worms*
      Worms merupakan *malware* yang berdiri sendiri dan tidak dapat menyebar luas. *Worms* menggunakan sumber daya jaringan dari komputer dengan mengkonsumsi sumber daya memorinya.
2. *The Masked Threat*
   a. *Trojan*
      Trojan akan menyamar sebagai program yang sah dan melakukan pengunduhan *file* dari internet serta mengambil rincian aktivitas *user*.
   b. *Rootkit*
      Rootkit merupakan *tools* yang berfungsi untuk menyembunyikan *malware* lain. *Rootkit* dapat terunduh secara bersamaan dari *software* lain maupun *trojan* dan akan mengambil *password* serta meng-*install keyloggers*.
3. *The Financial Threat*
   a. *Spyware*
      Spyware merupakan *malware* yang melakukan pengintaian terhadap aktivitas *user* tanpa sepengetahuan *user* serta mengambil informasi sensitif dari *user*.
   b. *Keyloggers*
      Keyloggers bekerja dengan memantau aktivitas *user* seperti *cookies*,*file* pada *drive*, serta merekam *keystrokes*. *Keyloggers* biasanya ter-*install* melalui *malware* lain ataupun saat *user* mengunjungi *website*.

2.3 Malware Analysis

Malware analysis merupakan penyelidikan terhadap *malware* yang bertujuan untuk mengetahui *malware* secara spesifik yang dapat membangun keamanan untuk melindungi perangkat [6].

1. *Heuristic Detection*

   Merupakan teknik *malware analysis* yang bekerja dengan mencari perintah atau instruksi yang dapat memungkinkan ditemukannya *malware* jenis baru [7]. Berikut merupakan kelebihan dan kekurangan dari *heuristic detection* [8]:

   - ✓ Kelebihan *heuristic detection* adalah:
     1. Dapat melihat perilaku *malware* yang akan dieksekusi.
     2. Berpotensi untuk menemukan *malware* yang tidak dikenal pada sistem.
     3. Memberi pemahaman terhadap hal yang tidak terduga kedepannya.
     4. Dapat digunakan secara bersamaan dengan teknik analisis lainnya.
   - ✓ Kekurangan *heuristic detection* adalah:
     1. Dapat memberikan peringatan palsu (*false positive*) pada sistem karena adanya deteksi yang lebih detail.
     2. Membutuhkan pengetahuan yang cukup dalam menganalisis
     3. Proses analisis dilakukan secara manual sehingga memerlukan waktu yang lebih lama.

2.4 Application Program Interface (API)

Menurut Vangie Beal, API merupakan suatu prosedur, protokol serta alat untuk membangun suatu aplikasi yang akan menentukan bagaimana suatu *software* berinteraksi. Keuntungan dari menggunakan Windows API adalah dapat menghemat waktu dalam analisis namun kekurangannya adalah tidak adanya toleransi pada kesalahan [9].

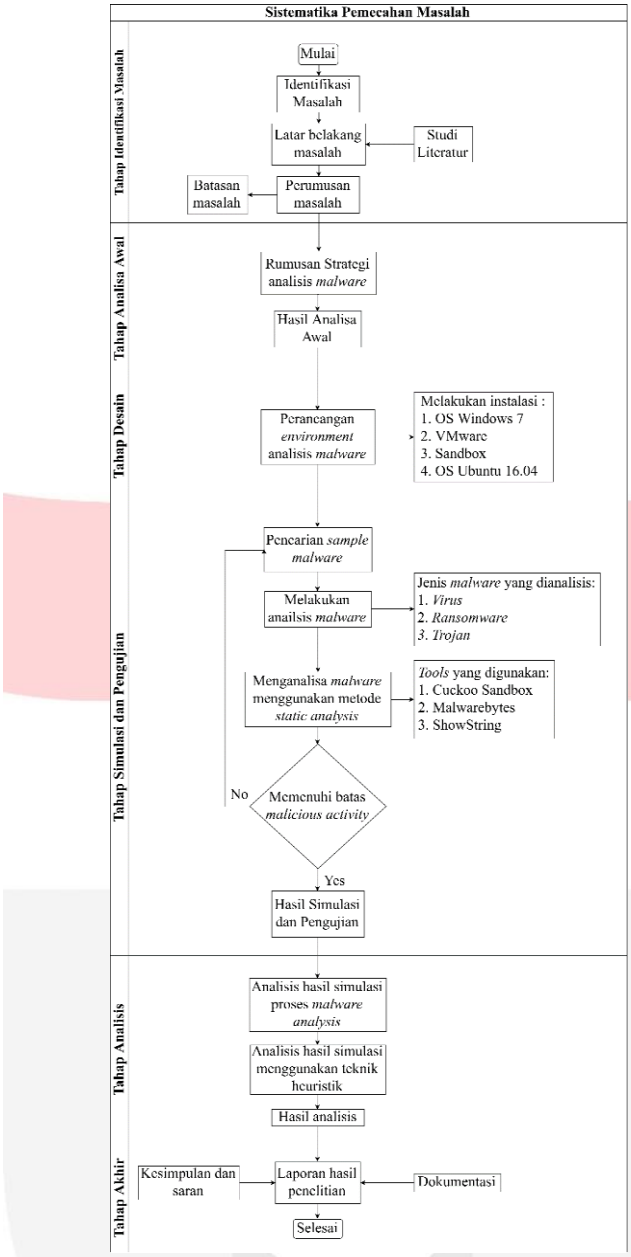
Umumnya *malware* menyerang fungsi jaringan dalam menjalankan programnya, karena pada Windows API komunikasi yang paling sering terjadi pada suatu aplikasi adalah melalui jaringan [10]. Berdasarkan pernyataan tersebut, maka penelitian ini akan berfokus pada API *network* untuk dianalisis.

1. API *Network*

   API *network* memungkinkan sebuah aplikasi untuk berkomunikasi dengan aplikasi lainnya, selain itu juga dapat digunakan sebagai akses ke sebuah *sharing resource* [9]. Cara kerja dari API *network* adalah dengan melakukan *looping* sehingga *resource network* pada sebuah sistem operasi menjadi penuh dan kinerja komputer akan menjadi lebih lambat.

2.5 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.



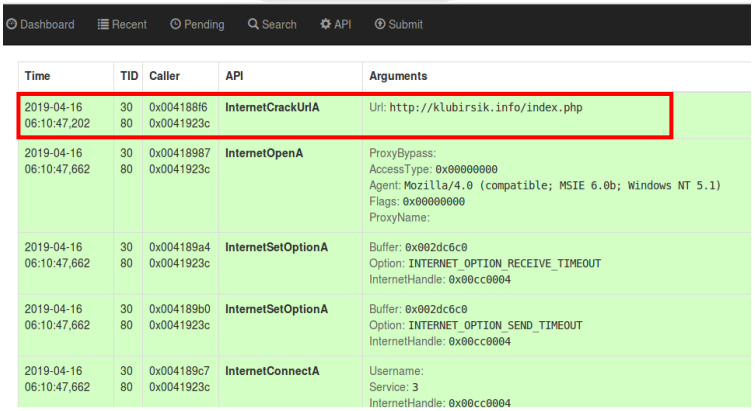
Gambar 1 Sistematika Pemecahan Masalah

3. Pengujian dan Analisis

3.1. Pengujian

1. Pengujian Menggunakan Cuckoo Sandbox

Cuckoo Sandbox adalah *tool* analisis *malware* yang diinstal pada *localhost*. Data yang diambil dari Cuckoo Sandbox adalah berupa API *network*.

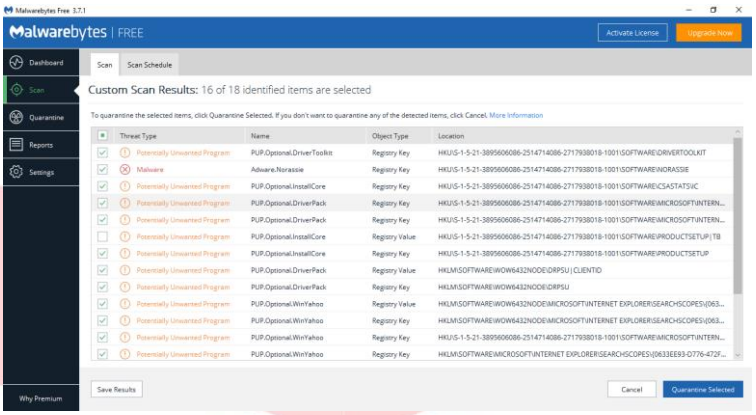


Gambar 2 Hasil Cuckoo Sandbox

Gambar 2 merupakan hasil pengujian salah satu *malware*. Dapat dilihat bahwa terdapat beberapa informasi terkait API *network* yang digunakan. Salah satunya adalah API *network* InternetCrackUrlA, merupakan *link* yang dijadikan target dari *malware*. *Link* tersebut merupakan jebakan agar *user* mengaksesnya sehingga peretas dapat memasuki sistem komputer *user*.

2. Pengujian Menggunakan Malwarebytes

Malwarebytes adalah *tool antimalware* untuk melakukan *scanning* terhadap suatu program. Program ini akan menghapus segala program yang berbahaya sebelum program berbahaya tersebut mengganggu aktivitas *user*.

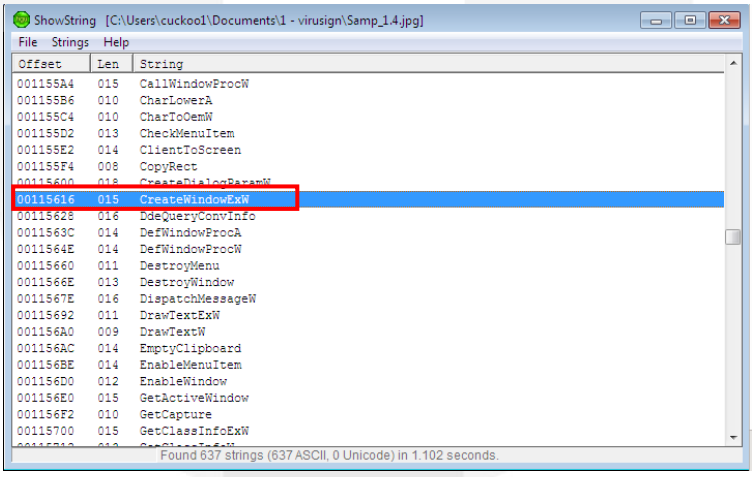


Gambar 3 Hasil Scanning Malwarebytes

Dari hasil *scanning* pada Gambar 3 terdapat beberapa PUP yang terdeteksi, dimana PUP tersebut bisa menimbulkan sebuah *malware* seperti *adware* atau *spyware*.

3. Pengujian Menggunakan ShowString

ShowString merupakan *tool* yang berfungsi untuk melihat *string* yang terdapat pada suatu *file*. *String* tersebut dapat menggambarkan bagaimana suatu *file* menjalankan tugasnya.

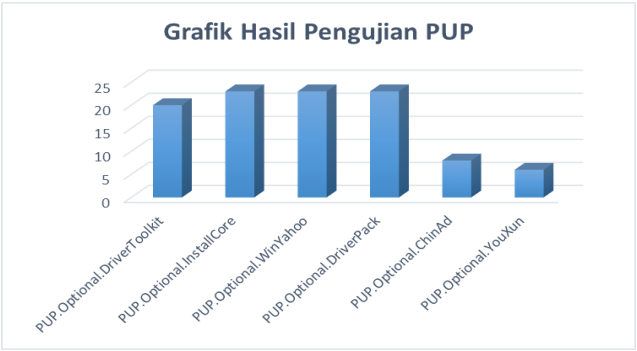


Gambar 4 Hasil Deteksi String Malware

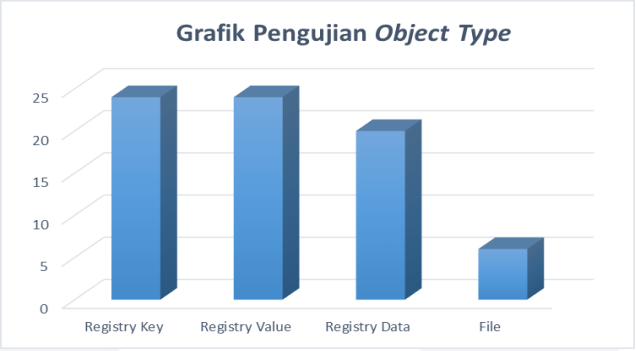
Gambar 4 tersebut terlihat beberapa fungsi yang terdapat pada kedua *file*. Salah satu fungsinya yaitu CreateWindowExW yang dimana *file* tersebut dapat membuat jendela baru yang saling tumpang tindih satu sama lain seperti jendela *pop-up*.

3.2 Hasil Analisis Dengan Metode *Heuristic Detection*

Analisis dengan metode *heuristic detection* ini berdasarkan hasil pengujian yang terdeteksi pada Malwarebytes. Berikut merupakan grafik hasil dari pengujian.



Gambar 5 Grafik Hasil Pengujian PUP



Gambar 6 Grafik Pengujian *Object Type*

Gambar 5 terlihat bahwa PUP yang paling banyak terdeteksi sebagai ancaman pada malware yang menggunakan API network adalah:

1. PUP.Optional.InstallCore, PUP ini merupakan *bundler* yang akan melakukan penginstalan *adware*. Dampak yang dapat ditimbulkan adalah dengan menampilkan *pop-up* iklan yang menggangu *user*.
2. PUP.Optional.WinYahoo akan melakukan perubahan pada halaman *default browser*. Hal ini berdampak pada adanya ekstensi yang terinstal secara otomatis pada *browser* dan dapat mengarahkan *browser* untuk membuka situs yang tidak diinginkan oleh *user*.
3. PUP.Optional.DriverPack merupakan PUP yang akan melakukan penginstalan driver secara otomatis pada sistem komputer. Hal tersebut dapat memicu masuknya spyware.

Berdasarkan data tersebut, *malware* yang menggunakan API *network* memiliki pola perilaku, seperti melakukan perubahan pada *browser*, menginstal *driver* yang tidak sah, serta menampilkan *pop-up* iklan-iklan yang tidak diinginkan. Dari ketiga perilaku tersebut, aktivitas *spyware* merupakan aktivitas yang paling banyak, sehingga *malware* yang menggunakan API *network* memiliki kecendrungan terhadap masuknya *spyware*. Sedangkan PUP yang tidak dominan terdeteksi adalah:

1. PUP.Optional.DriverToolkit melakukan pengunduhan aplikasi Driver Toolkit secara otomatis ketika *user* mengakses suatu *website* yang tidak aman. *User* akan diarahkan untuk melakukan instalasi yang bertujuan untuk memperbaiki sistem operasi perangkat komputer *user*.
2. PUP.Optional.ChinAd merupakan *adware* yang menampilkan situs *marketplace* yang berasal dari China. Dampak dari adanya *malware* ini adalah adanya pembajakan *browser* dan bisa membuat *user* melakukan *misclick* ketika sedang menggunakan *broswer*.
3. PUP.Optional.YouXun, PUP yang mengunduh sebuah *file* yang terinfeksi *malware* sehingga berdampak masuknya *malware* pada perangkat komputer *user* dan berpengaruh pada kinerja perangkat komputer.

PUP.Optional.DriverToolkit dan PUP.Optional.YouXun memiliki pola perilaku yang sama, yaitu mengunduh suatu *file*. Kedua PUP ini melakukan pengunduhan secara otomatis tanpa sepengatahuan *user*, namun untuk instalasi masih dilakukan secara manual sehingga *user* masih dapat mencegah terinstalnya program *malware* pada perangkat komputer. Sedangkan PUP.Optional.ChinAd merupakan PUP yang sedikit terdeteksi karena *adware* yang biasa menyerang perangkat komputer bersifat luas, yaitu dapat menampilkan iklan berupa *pop-up* atau *tab* baru pada *browser* yang dapat berasal dari mana saja. *Adware* yang menyerang perangkat komputer juga dapat berupa situs *game* yang tidak diingkan oleh *user*.

Gambar 6 terlihat bahwa tempat atau lokasi yang banyak digunakan sebagai target *malware* adalah:

1. *Registry key* dan *registry value*, jika *malware* langsung menyerang kedua *registry* tersebut maka akan terjadi perubahan pada sistem operasi, karena *malware* dapat mengubah konfigurasi yang terdapat pada kedua *registry*

tersebut, seperti komputer tidak dapat melakukan *shutdown*, tidak dapat membuka *file* yang telah di *hidden*, serta tidak dapat mengakses beberapa aplikasi yang terdapat dalam perangkat komputer *user*.

2. *Registry data*, dampak yang ditimbulkan ketika *registry data* terinfeksi *malware* hampir sama dengan dampak dari *registry key* dan *registry value*, dikarenakan *registry data* merupakan *file* konfigurasi aktual yang berada di dalam *registry value*.
3. *File*, ketika suatu *file* terinfeksi oleh malware, maka *file* tersebut bisa saja tidak dapat dibuka atau *file* tersebut dapat berubah menjadi virus lainnya, hilang atau disembunyikan oleh *malware* yang menginfeksinya.

Berdasarkan hasil analisis, PUP yang telah terdeteksi secara keseluruhan dapat menimbulkan malware, namun terdapat PUP yang membutuhkan tindakan *user* terlebih dahulu untuk menjalankan *malware*-nya. PUP.Optional.DriverToolkit dan PUP.Optional.YouXun merupakan PUP yang membutuhkan tindakan *user* terlebih dahulu. Kedua PUP ini hanya akan mengunduh *file* secara otomatis tanpa sepengetahuan *user*, namun untuk penginstalannya membutuhkan persetujuan *user*. Sedangkan keempat PUP lainnya masuk ke dalam sistem operasi secara langsung dan menginfeksi sistem operasi dengan *malware*. Dampak terbesar dari *malware* yang menggunakan API *network* adalah timbulnya spyware.

4. Kesimpulan

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

1. Pengujian *malware* dapat menggunakan beberapa *environment* yang berfungsi sebagai *sandbox*. Pada penelitian ini sistem operasi Windows dijalankan pada VMware yang berfungsi sebagai *sandbox* agar sistem operasi utama tidak terinfeksi. Sistem operasi Windows digunakan sebagai target dalam analisis menggunakan Cuckoo Sandbox.
2. Hasil penelitian berdasarkan hasil pengujian dari Malwarebytes yang dapat mendeteksi program yang menimbulkan *malware* dengan metode *heuristic detection* dan melihat *string malware* pada ShowString.
3. *Malware* dengan API *network* akan menyerang *registry key* sistem operasi dan memiliki program yang dapat menimbulkan *spyware* atau *adware* yang dapat mengganggu aktivitas *user* ketika menggunakan perangkat komputernya.
4. Rekomendasi untuk melindungi sistem komputer seperti menggunakan *antivirus* atau *antimalware*, tidak memasang aplikasi yang tidak sah, tidak mengakses *website* yang tidak aman serta tidak perlu memasang aplikasi tambahan yang tidak dibutuhkan ketika memasang suatu aplikasi.

Daftar Pustaka:

[1] Donalds, C. and Osei-Bryson, K. (2018). Toward a cybercrime classification ontology: A knowledge-based approach. Computers in Human Behavior, 92, pp.403-418.

[2] Uppal, D., Mehra, V. and Verma, V. (2014). Basic survey on Malware Analysis, Tools and Techniques. International Journal on Computational Science & Applications, 4(1), pp.103-112.

[3] Deka, D., Sarma, N. and J. Panicker, N. (2016). Malware Detection Vectors and Analysis Techniques: A Brief Survey. IEEE.

[4] Shijo, P. and Salim, A. (2015). Integrated Static and Dynamic Analysis for Malware Detection. Procedia Computer Science, 46, pp.804-811.

[5] What is Malware and How Can We Prevent It?. (2018). Diakses pada 9 Desember 2018, dari https://antivirus.comodo.com/blog/how-to/what-is-malware/.

[6] More, S. and Gaikwad, P. (2016). Trust-based Voting Method for Efficient Malware Detection. Procedia Computer Science, 79, pp.657-667.

[7] Zalavadiya, N. and Sharma, P. (2017). A Methodology of Malware Analysis, Tools and Technique for windows platform – RAT Analysis. International Journal of Innovative Research in Computer and Communication Engineering(3).

[8] Sihwail, R., Omar, K. and Zainol Ariffin, K. (2018). A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. International Journal on Advanced Science, Engineering and Information Technology, 8(4-2), p.1662.

[9] Docs.microsoft.com. (2018). Walkthrough: Calling Windows APIs (Visual Basic). Diakses pada 9 Desember 2018, dari https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/com-interop/walkthrough-calling-windows-apis.

[10] Gandotra, E., Bansal, D. and Sofat, S. (2014). Malware Analysis and Classification: A Survey. Journal of Information Security, 05(02), pp.56-64.