ANALISIS DETEKSI *MALWARE REMOTE ACCESS TROJAN* MENGGUNAKAN *DYNAMIC MALWARE ANALYSIS DETECTION TOOLS* BERBASIS *BEHAVIOUR*

MALWARE DETECTION ANALYSIS OF REMOTE ACCESS TROJAN WITH BEHAVIOUR-BASED DYNAMIC MALWARE ANALYSIS DETECTION TOOLS

Epifanio Juang Victorius[1], Avon Budiyono, 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]**juangv@student.telkomuniversity.ac.id** , [2]**avonbudi@telkomuniversity.ac.id**, [3]**ahmadalmaarif@telkomuniversity.ac.id**

Abstrak

Semakin berkembangnya suatu teknologi, semakin besar pula peluang terjadinya *cybercrime* melalui penyerangan *malware*. *Malicious software* (*malware*) merupakan sebuah *software* berbahaya sengaja dirancang untuk menjalankan muatan asing yang merugikan atau merusak sistem korban tanpa sepengetahuannya. Dengan banyak kategori *malware* yang tersebar, membuat semua sistem rentan terhadap serangan *malware*. Salah satu kategori *malware* yang paling berbahaya adalah *Remote Access Trojan* (RAT) yang dapat mengendalikan sistem secara menyeluruh untuk mencuri informasi pribadi, menghapus *file*, memodifikasi *file*, mengganggu kinerja *user*, dan memasang *malware* atau *backdoor* di dalam sistem. Terbukti dengan adanya 557 serangan *malware* RAT yang terjadi atau terdeteksi antara 1 September 2017 hingga 31 Agustus 2018 di beberapa instansi atau individu di United Kingdom. Oleh karena itu, diperlukan *malware analysis* berbasis *behaviour* untuk mengetahui dan menganalisis *malware behaviour* yang unik berupa Windows API dan *Registry* dari *malware* RAT. Penelitian ini menggunakan 3 dari 10 sampel *malware* RAT yang telah didapatkan yaitu DarkComet-RAT, njRAT, dan QuassarRAT untuk diuji dan dianalisis *malware behaviour*-nya *Malware behaviour* yang dianalisis adalah Windows API dan Windows *Registry* ketika *malware* RAT diinisiasi, dan mengeksekusi *Keylogger, File* Transfer dan *Remote.* Desktop menggunakan *dynamic malware analysis detection tools* berbasis *behaviour*. Penelitian ini juga membandingkan *behaviour* inisiasi antara *remote access software* yaitu AeroAdmin dan *malware* RAT untuk mengetahui perbedaan Windows API dan Windows *Registry* yang digunakan. Hasil *malware behaviour* yang didapatkan menjelaskan bahwa *malware* RAT akan menggunakan Windows API dan Registry yang berkaitan dengan RPC dan OLE untuk membuat koneksi dengan sistem yang ditarget, lalu menggunakan Windows API dan Windows *Registry* yang berhubungan dengan *Keyboard Input, Data Access and Storage, Graphic and Gaming* ketika beberapa fitur dieksekusi. *Malware* RAT tidak akan memvalidasi segala aktivitas yang dilakukan dan segala fitur *malware* RAT dapat dijalankan secara manual oleh *attacker*-nya.

Kata kunci : ***malware, malware rat, malware analysis, dynamic malware analysis, malware behaviour.***

Abstract

The more the development of a technology, the greater the chance of cybercrime through malware attacks. Malicious software (malware) is a malicious software intentionally designed to run unfamiliar loads that harm or damage the victim's system without his knowledge. With many malware categories spread, making all systems vulnerable to malware attacks. One of the most dangerous categories of malware is Remote Access Trojan (RAT) which can control the system as a whole to steal personal information, delete files, modify files, disrupt user performance, and install malware or backdoor in the system. Evidenced by the existence of 557 RAT malware attacks that occur or are detected between September 1, 2017 to August 31, 2018 in several agencies or individuals in the United Kingdom. Therefore, malware analysis based behavior is needed to find out and analyze the unique malware behavior in the form of Windows API and Registry from malware RAT. This study uses 3 out of 10 RAT malware samples that have been obtained, namely DarkComet-RAT, njRAT, and QuassarRAT to be tested and analyzed for malware behavior. The malware behavior analyzed is the Windows API and Windows Registry when RAT malware is initiated, and executes Keylogger, File Transfer and The remote. Desktop uses dynamic malware analysis detection tools based on behavior. This study also compares the behavior of initiation between remote access software, namely AeroAdmin and malware RAT to find out the differences between the Windows API and the Windows Registry used. The malware behavior found explains that malware RAT will use the Windows API and Registry related to RPC and OLE to establish connections with targeted systems, then use the Windows API and Windows Registry related to Keyboard Input, Data Access and Storage, Graphic and Gaming when several features are executed. Malware RAT will not validate all activities carried out and all malware RAT features can be run manually by the attacker.

Keywords*: *malware, malware rat, malware analysis, dynamic malware analysis, malware behaviour.

1. Pendahuluan

Perkembangan teknologi saat ini sangat cepat dan maju sehingga mempermudah manusia untuk saling terhubung menggunakan internet atau perantara komunikasi lainnya seperti sistem komputer dan *smartphone*. Dengan majunya teknologi pasti memberikan dampak positif bagi penggunanya. Namun tidak semua perkembangan teknologi memberikan dampak positif. Banyak juga dampak negatif yang diberikan oleh perkembangan teknologi khususnya pada dunia internet yaitu *cybercrime.* Segala kegiatan kriminal yang dilakukan melalui perangkat atau media komunikasi untuk mendapatkan informasi dan menggunakannya demi mendapatkan keuntungan atau tidak dari tindak kriminal merupakan istilah dari *cybercrime* [19]. *Malicious software* atau *malware* merupakan sebuah *software* berbahaya yang dirancang untuk merusak sebuah sistem tanpa sepengetahuan korban [7]. Setiap *software* yang mengeksekusi sebuah muatan yang mencurigakan dikatakan sebagai *malware* seperti virus, *worm*, *trojan*, *ransomware*, dan lainnya [4]. *Malware* biasanya mengganggu operasi dari sistem yang bekerja dengan mengambil informasi sensitif atau membuat sebuah akses tanpa izin ke dalam sistem korban [6]. *Malware* memiliki banyak cara penyebaran seperti S*ocial Engineering, File Download Fraud, Email Phising, Malvertising* untuk membuat korban tertipu dan menginfeksi komputer korban dengan *malware*. Ketika komputer korban sudah terinfeksi, *malware* akan bekerja sesuai *behaviour*-nya. Dari *behaviour* tersebut, akan membantu dalam menentukan jenis dari *malware* tersebut. Namun, sebelum menentukan jenis *malware* yang menginfeksi komputer perlu adanya tahapan analisis *malware* terlebih dahulu [21]. Mulai dari tahun 2016 hingga tahun 2017, tercatat 14 juta lebih *malware* terbaru yang berhasil terdeteksi. Pada tahun 2018, sudah terdeteksi 5 juta lebih *malware* baru dan terus meningkat dengan kecepatan 13.000 sampel *malware* baru per harinya [13].

Di dalam perkembangannya, *Remote Access Trojan* (RAT) merupakan salah satu *malware* yang paling berbahaya yang pernah dikembangkan. *Malware* RAT dapat mencuri informasi pribadi, menghapus *file*, memodifikasi *file*, mengganggu kinerja *user*, mengendalikan sistem korban bahkan memasang sejumlah *malware* atau *backdoor* di dalam sistem korban. *Malware* RAT sanggatlah sulit untuk dideteksi karena dapat bersembunyi atau menyamarkan dirinya sebagai *software* atau proses yang biasanya tidak dijelajahi oleh antivirus *software* [8]. *Malware* RAT sering digunakan oleh individual atau kelompok kriminal untuk melakukan *advaced persistent threat* (APTs) karena sangat mudah untuk digunakan, dikonfigurasi, dan dimodifikasi untuk tujuan tertentu. Bahaya dari *malware* RAT dapat dilihat pada serangan yang terjadi pada negara seperti United Kingdom, mulai dari 1 September 2017 hingga 31 Agustus 2018 terjadi sekitar 557 kasus penyerangan *malware* RAT yang berhasil diatasi atau terdeteksi di beberapa instansi atau individu [15]. Sehingga perlu dilakukannya *malware analysis* berbasis *behaviour* untuk mendeteksi serta menganalisis *behaviour* yang unik dari *malware* RAT yang dapat mengelabui deteksi dari antivirus atau *tools* deteksi lainnya [11].

Malware analysis merupakan proses untuk menentukan fungsionalitas dari sampel *malware* berdasarkan hasil pengujiannya baik menggunakan *static analysis* maupun *dynamic analysis* [4]. Static analysis dilakukan secara manual untuk melihat *binary*, struktur data, fungsi lainnya dari sebuah *malware* tanpa mengeksekusinya. *Dynamic analysis* dilakukan dengan mengeksekusi *malware* pada *environment* yang terisolasi dan mengawasi *malware behaviour* untuk dianalisis. Proses *malware analysis* penting untuk dilakukan mengembangkan teknik deteksi dari sebuah *malware* dan menentukan karakteristik dari *malware* tersebut [7]. Memahami tujuan dan *malware behaviour* sangat membantu untuk mendesain dan mengimplementasi mekanisme pencegahan dari serangan *malware*. *Malware behaviour analysis* dikenal sebagai proses memahami tipe dan karakteristik *malware* dari *behaviour* yang dimilikinya. Biasanya *malware behaviour analysis* dilakukan dengan *dynamic analysis* karena analisis dilakukan dengan mengeksekusi *malware* dan mengawasi *behaviour*-nya [25]. Teknik ini memiliki kelebihan untuk menganalisis dan menentukan *malware* dari apa yang dilakukan oleh *malware* dan bukan dari apa yang *malware* perlihatkan [17].

Sebuah atau beberapa *Application Programming Interface* (API) dan *Registry* dapat diidentifikasi sebagai *malware behaviour* ketika dieksekusi khususnya pada *operating system* Windows. API merupakan kumpulan perintah, fungsi, dan protokol yang digunakan sebuah *software* untuk berinteraksi dengan *operating system*. Interaksi tersebut merupakan *behaviour* dari *software* atau *malware* [1]. *Registry* Windows merupakan *database* yang *mendefiniskan operating system* Windows berisikan informasi-informasi penting bagi pengaturannya [12]. *Registry* Windows digunakan untuk melihat *behaviour* yang diakses oleh *software* ketika dieksekusi dan melihat korelasi antar tiap aktivitas yang terjadi pada sebuah sistem [16].

Oleh karena itu, untuk melakukan *malware analysis* penulis menggunakan *dynamic malware analysis detection tools* berbasis *behaviour* kepada *malware* RAT. Hasil akhir dari penelitian ini berupa *malware analysis* berdasarkan *malware behaviour* ketika menggunakan dan menggunakan API dan mengakses *Registry* Windows.

2. Dasar Teori

2.1 Definisi *Malware*

Malware adalah sebuah *software* yang sengaja dibuat dan bertujuan untuk melakukan perusakan, pencurian data pribadi, serta mendapatkan akses ke dalam sebuah sistem. Setiap perangkat lunak yang merugikan *user*, komputer, atau jaringan dapat dikategorikan sebagai sebuah *malware*. Virus, *Trojan*, *Worm*, *Rootkit*, *Scareware*, *Spyware*, dan *Ransomware* merupakan contoh dari *malware* [17].

Malware merupakan sebuah *software* berbahaya yang sengaja dirancang untuk merusak sebuah sistem tanpa sepengetahuan korban [6]. Setiap *software* yang dapat mengeksekusi sebuah muatan mencurigakan pada sistem dapat dikategorikan sebagai *malware* [3]. *Malware* biasanya mengganggu operasi dari sistem yang bekerja dengan mengambil informasi sensitif atau membuat sebuah akses tanpa izin ke dalam sistem korban [5].

2.2 Kategori *Malware*

Menurut [9], berikut adalah beberapa tipe *malware* berdasarkan cara menginfeksi sebuah sistem:

1. Virus, sebuah *malware* yang bersembunyi pada suatu program yang dianggap tidak berbahaya dengan tujuan untuk memperbanyak diri, mengubah atau merusak sebuah *file*, atau tindakan yang membahayakan sebuah sistem
2. *Worm*, sebuah *malware* yang menginfeksi sistem melalui jaringan yang rentan dengan tujuan untuk memperbanyak dirinya. *Worm* tidak bersembunyi pada sebuah program lainnya untuk memperbanyak diri seperti virus
3. *Trojan*, sebuah *malware* yang menyamarkan dirinya sebagai program atau *file* yang asli agar korban memasangnya pada suatu sistem. *Trojan* membawa fungsi atau program lainnya yang berbahaya ketika dieksekusi pada suatu sistem. Biasanya *trojan* berfungsi sebagai *backdoor* untuk memberikan izin memasuki sistem korban tanpa sepengetahuannya agar *attacker* dapat mengendalikan sistem tersebut
4. *Spyware*, sebuah *malware* yang berfungsi untuk mengawasi aktivitas yang terjadi pada sebuah sistem tanpa sepengetahuan korbannya. *Spyware* dapat mengirimkan informasi yang didapatkannya kepada *attacker* untuk tujuan tertentu
5. *Ransomware*, sebuah *malware* yang dapat mengunci layar komputer atau *smartphone* yang bertujuan menakut-nakuti korbannya agar membayar suatu biaya untuk memulihkan perangkat yang terinfeksi.
6. *Rootkit*, sebuah *malware* yang menyembunyikan dirinya dengan mengubah pengaturan sebuah sistem yang terinfeksi agar dapat selalu aktif. *Rootkit* bertujuan untuk memberikan segala akses untuk mengendalikan sistem korban kepada *attacker*
7. *Backdoor*, sebuah *malware* yang bertujuan untuk mem-*bypass* proses autentikasi agar *attacker* dapat menerima akses masuk ke dalam sistem korban. *Backdoor* biasanya akan memasang beberapa *backdoor* lainnya untuk menjamin akses lainnya ketika *backdoor* terhapus.

2.3 *Malware Analysis*

Malware analysis adalah sebuah kegiatan untuk membedah dan menganalisis sebuah *malware* dengan atau tanpa mengeksekusinya. *Malware analysis* bertujuan untuk mempelajari cara kerja *malware*, cara mengidentifikasi *malware*, fungsionalitas *malware*, *malware behaviour* serta cara pencegahan yang terbaik terhadap suatu malware [17]. Menurut [3], *malware analysis* merupakan proses untuk menentukan fungsionalitas dari sampel *malware* berdasarkan hasil pengujiannya baik menggunakan *static analysis* maupun *dynamic analysis*.

2.3.1 *Static Analysis*

Static analysis adalah aktivitas menganalisis sebuah *software* yang dicurigai sebagai *malware* tanpa atau sebelum mengeksekusinya. *Static analysis* digunakan untuk melihat *source code* dari sebuah *malware*. *Static analysis* membedah sebuah *malware* menggunakan *reverse engineering*, *debugger*, atau *disassembler tools untuk* menemukan pola, *string*, atau aktivitas aneh yang menandakan sebuah *malware* [20].

2.3.2 *Dynamic Analysis*

Dynamic analysis adalah aktivitas menganalisis *malware* bersamaan ketika mengeksekusinya. *Malware* akan dieksekusi pada sebuah *environment* terisolasi biasanya sebuah virtual *machine* untuk dianalisis lebih dalam lagi dari segi fungsionalitas, *behaviour*, atau *bug* saat *malware* berjalan [20].

Dynamic analysis digunakan untuk mendeteksi *malware* yang belum diketahui atau yang biasa disebut sebagai *zero-day malware*. *Dynamic anlysis* dapat dilakukan dengan mengawasi dan melacak *function calls* dan *malware behaviour* menggunakan aplikasi seperti *Sandbox* dan *Process* Explorer [18]. Namun, deteksi dan analisis yang dilakukan membutuhkan waktu yang cukup lama dan tidak aman jika tidak dilakukan pada *environment* yang tepat. *Dynamic analysis* juga kurang tepat digunakan pada *malware* yang ketik dieksekusi dapat mengubah *behaviour*-nya dalam satu kondisi tertentu [19].

2.4 *Malware Behaviour*

Malware Behaviour merupakan sebuah tindakan atau perilaku suatu program atau *file* terhadap *environment*-nya yang mencurigakan dan tidak normal dilakukan sebuah program. *Behaviour* tersebut terjadi ketika adanya sebuah *input* atau aktivitas yang memicu *behaviour* tersebut untuk muncul [8]. *Malware Behaviour* dideskripsikan

sebagai sebuah *file written, process created, system call patern* atau perubahan status dari suatu sistem ketika *malware* dieksekusi [4].

Malware analyzer atau *detection tools* digunakan untuk mendapatkan *Malware Behaviour* ketika dieksekusi. *Malware Behaviour* berbeda-beda untuk setiap sampel *malware* dan setiap *tools* yang digunakan [21].

2.5 *Application Programming Interface* (API) Windows

Application Programming Interface (API) merupakan sebuah kumpulan perintah, fungsi, dan protokol yang digunakan sebuah *software* untuk berinteraksi dengan *operating system*. API menjadi *behaviour* sebuah *software* ketika menggunakan *operating system* dalam hal ini *operating system* Windows sepenuhnya untuk tujuan tertentu. API Windows melingkupi beberapa tingkatan fungsi untuk berinteraksi dengan *system services, security, graphical interface, user interface, networking, windows shell*, dan lainnya [1]. API Windows digunakan untuk melihat pola *behaviour* dan mengklasifikasikan *malware* ketika berinteraksi dengan API itu sendiri khususnya pada *operating system* Windows [13]

2.6 *Registry* Windows

Registry Windows merupakan *database* yang mendefinisikan sebuah sistem berisikan informasi-informasi penting bagi pengaturan *operating system* Windows. *Registry* Windows menyimpan segala informasi mulai dari pengaturan dasar sistem hingga data sensitif sistem sehingga disarankan untuk tidak mengubah isi dari *Registry* [11].*Registry* Windows digunakan untuk melihat *behaviour* yang diakses dengan atau tanpa izin sistem oleh *software* ketika dieksekusi. *Registry* dapat digunakan untuk mendapatkan korelasi antar tiap aktivitas yang terjadi pada sebuah sistem ketika terinfeksi *malware* [14].

2.7 Metodologi Penelitian

Metode yang membantu dalam pengerjaan penelitian ini menggunakan model konseptual. Model konseptual memberikan hubungan antara dasar teori/literatur dan masalah dari penelitian yang dilakukan. Lalu, model konseptual membantu untuk mendeskripsikan masalah dan domain-domain yang terkait untuk mencapai tujuan penelitian dan merumuskan masalah tersebut [2]. Sehingga peneliti dapat menjelaskan tujuan penelitian tugas akhir ini yaitu bagaimana melakukan *malware analysis* dari *malware* RAT menggunakan *dynamic malware detection tools* berbasis *behaviour*. Permasalahan dari penelitian ini adalah semakin maraknya *malware* yang menyerang komputer menimbulkan *malware-malware* baru yang menyerang. Malware RAT merupakan salah satu yang paling berbahaya karena dapat mengendalikan penuh komputer korban ketika terinfeksi. *Malware* RAT sanggatlah sulit dideteksi karena bersembunyi atau menyamarkan diri sebagai *software* atau proses yang normal [7]. *Malware analysis* menggunakan metode *dynamic analysis* dapat membantu untuk menganalisis *behaviour malware* RAT .

Dengan permasalahan dan peluang yang ada pada penelitian ini, dihasilkan sebuah artefak yaitu *malware analysis* menggunakan *dynamic analysis malware* terhadap *behaviour malware* RAT. Analisis *malware behaviour* yang dihasilkan didapatkan dengan menggunakan *dynamic malware detection tools* berbasis *behaviour*. Adapun teori-teori yang digunakan adalah *Dynamic Analysis*, *Malware Detection Tools*, API Windows, *Registry* Windows, dan *Malware Behaviour*. Analisis dilakukan kepada *malware behaviour* ketika diinisiasi, mengeksekusi *Keylogger*, *File* Transfer dan *Remote* Desktop didapatkan. *Malware behaviour* berupa *behaviour* ketika *malware RAT* menggunakan API dan mengakses *Registry* Windows.

3. Pengujian dan Analisis

3.1. Pengujian *Behaviour* Inisiasi *Malware* RAT

Backdoor RAT dieksekusi pada Komputer *Victim*, akan dilakukan pengumpulan data *behaviour* ketika *malware* RAT dieksekusi pertama kali. Aktivitas ini dikatakan sebagai inisiasi RAT. *Behaviour* inisiasi yang diambil merupakan API dengan kategori *networking* menggunakan aplikasi API Monitor dan *Registry* menggunakan aplikasi *Process* Monitor. Berikut adalah Tabel 1 berisikan *malware behaviour* inisiasi RAT yang didapatkan dari pengujian:

Tabel 1 Hasil Pengujian *Behaviour* Inisiasi *Malware* RAT

RAT	*Behaviour*	
	API Monitor	*Process* Monitor
DarkComet-RAT	• MapVirtualKeyA • GetKeyState • GetKeyboardState • ToAscii	-
njRAT	• EnumWindows • GetWindowThreadProcessID • GetWindow • GetWindowTextLengthW • GetWindowTextW	-
QuasarRAT	• GetKeyState • GetKeyboardState • GetWindowTextW • MultiByteToWideChar • GettKeyboardLayout	-

Tabel 1 merupakah hasil *malware behaviour* dari DarkComet-RAT, njRAT, dan QuasarRAT ketika diinisiasi. Di mana terlihat beberapa API Windows kategori *networking* yang digunakan dan *Registry* Windows yang diakses yang berhubungan dengan *Remote Procedure Call* (RPC) dan membuat koneksi antara sistem yang

terinfeksi dengan aplikasi server RAT. Contohnya, RpcBindingBind untuk membuat hubungan RPC server dari sistem penyerang dan sistem yang terinfeksi. HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL untuk menjalankan *Object Linking and Embedding* (OLE) ketika RPC digunakan oleh *malware* RAT.

3.2. Pengujian *Behaviour Keylogger Malware* RAT

Pada fitur *Keylogger malware* RAT, data *behaviour* didapatkan ketika Komputer *Attacker* mengeksekusi fitur *Keylogger* dari *malware* RAT kepada Komputer *Victim* yang sedang melakukan aktivitas pengetikan atau menekan salah satu tombol *keyboard*. Berikut adalah Tabel 2 *malware behaviour Keylogger* RAT yang didapatkan ketika melakukan pengujian:

Tabel 2 Hasil Pengujian *Behaviour Keylogger Malware* RAT

RAT	*Behaviour*	
	API Monitor	*Process* Monitor
DarkComet-RAT	• WriteFile • CreateFileA	-
njRAT	• WriteFile	-
QuasarRAT	• GetQueuedCompletionStatus • GetDriveTypeW • GetVolumeInformationW • FindFirstFileW • FindNextFileW	-

Tabel 2 merupakah hasil *malware behaviour* dari DarkComet-RAT, njRAT, dan QuasarRAT ketika mengeksekusi *Keylogger*. Di mana terlihat hanya API Windows kategori Windows *Application* UI *Development* yang digunakan untuk mendapatkan aktivitas *keyboard* sistem yang terinfeksi dan mengirimkan hasil aktivitas tersebut ke aplikasi server RAT. Contohnya, GetKeyState untuk mendapatkan status virtual *keys* setiap tombol *keyboard* yang ditekan pada sistem yang terinfeksi. GetWindowTextW untuk menyalin setiap karakter dari suatu aktivitas pada sistem yang terinfeksi.

3.3. Pengujian *Behaviour File* Transfer *Malware* RAT

Pada fitur *File* Transfer *malware* RAT data *behaviour* didapatkan ketika Komputer *Attacker* mengeksekusi fitur *File* Transfer dari *malware* RAT kepada Komputer *Victim* lalu melakukan pengiriman sebuah *file* ke dalam Komputer *Victim* . Berikut adalah data *behaviour File* Transfer RAT yang didapatkan ketika melakukan pengujian:

Tabel 3 Hasil Pengujian Behaviour File Transfer Malware RAT

RAT	*Behaviour*	
	API Monitor	*Process* Monitor
DarkComet-RAT	• WSAStartup • RpcBindingCreateW • RpcBindingBind	• HKLM\System\CurrentControlSet \Control\SESSION MANAGER • HKLM\SOFTWARE\Microsoft\OLE • HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL
njRAT	• RpcStringBindingComposeW • RpcBindingFromStringBindingW • NdrClientCall2	• HKLM\System\CurrentControlSet \Control\SESSION MANAGER • HKLM\SOFTWARE\Microsoft\OLE • HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL
QuasarRAT	• NdrClientCall2	• HKLM\System\CurrentControlSet \Control\SESSION MANAGER • HKLM\SOFTWARE\Microsoft\OLE • HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL

Tabel 3 merupakah hasil *malware behaviour* dari DarkComet-RAT, njRAT, dan QuasarRAT ketika mengeksekusi *File* Transfer. Di mana terlihat hanya API Windows kategori Data *Access and Storage* yang digunakan untuk mendapatkan informasi *disk drive* sistem yang terinfeksi dan mengirimkan *file* yang ditargetkan. Contohnya GetVolumeInformationW untuk mendapatkan informasi tentang sistem *file*, *volume*, dan direktori *root* sistem yang terinfeksi. CreateFileA untuk membuat dan membuka *file* atau perangkat I/O yang telah dibagikan dan diberikan akses melalui jaringan.

3.4. Pengujian *Behaviour Remote* Desktop *Malware* RAT

Pada fitur *Remote* Desktop RAT data *behaviour* didapatkan ketika Komputer *Attacker* mengeksekusi fitur *Remote* Desktop dari *malware* RAT kepada Komputer *Victim* sampai dengan Komputer *Attacker* mendapatkan gambaran dan kendali terhadap layar Komputer *Victim*. Berikut adalah data *behaviour Remote* Desktop RAT yang didapatkan ketika melakukan pengujian:

Tabel 4 Hasil Pengujian *Behaviour Remote* Desktop Malware RAT

RAT	*Behaviour*	
	API Monitor	***Process* Monitor**
DarkComet-RAT	• CreateDCA • SelectObject • GdipCreateBitmapFromHBITMAP • GdipGetImagePixelFormat • GdipCreateBitmapFromScan0 • GdipGetImageGraphicsContext	• HKLM\HARDWARE\DEVICEMAP \VIDEO\Device\Video
njRAT	• GetDC • SelectObject • GdipCreateBitmapFromHBITMAP • GdipBitmapLockBits • GdipBitmapUnlockBits	-
QuasarRAT	• EnumDisplayMonitors • CreateICA • GetStockObject • GdipCreateBitmapFromScan0	-

Gambar 4 merupakah hasil *malware behaviour* dari DarkComet-RAT, njRAT, dan QuasarRAT ketika mengeksekusi *Remote* Desktop. Di mana terlihat beberapa API Windows kategori *Graphic and Gaming* yang digunakan dan *Registry* Windows yang diakses yang berhubungan dengan informasi *display* dan penggunaan grafik atau objek pada sistem yang terinfeksi. Contohnya, SelectObject untuk memilih objek dari DC yang dibuat dan berada pada sistem yang terinfeksi. HKLM\HARDWARE\DEVICEMAP\VIDEO\Device\Video untuk memberikan daftar *display* yang ada pada sistem yang terinfeksi.

3.5. Pengujian *Behaviour Remote* Desktop *Malware* RAT

Perbandingan yang dilakukan berfokus pada *behaviour* inisiasi antara *remote access software* dan *malware* RAT. Data *behaviour* inisiasi yang dibandingkan adalah API dengan kategori *networking* yang didapatkan menggunakan aplikasi API Monitor dan *Registry* menggunakan aplikasi *Process* Monitor. Berikut adalah tabel yang menjelaskan perbandingan *behaviour* antara *behaviour* inisiasi antara *remote access software* dan *malware* RAT:

Tabel 5 Hasil Pengujian *Behaviour Remote Access Software* dan *Malware* RAT

Software/ *Malware*	**API Windows**	***Registry* Windows**
AeroAdmin	• accept • recv • WSAGetLastError • setsockopt • send	• HKLM\System\CurrentControlSet\Control\Lsa • HKLM\Hardware\DeviceMap\Video
DarkComet-RAT	• WSAStartup • RpcBindingCreateW • RpcBindingBind	• HKLM\System\CurrentControlSet\Control\SESSION MANAGER • HKLM\SOFTWARE\Microsoft\OLE • HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL
njRAT	• RpcStringBindingComposeW • RpcBindingFromStringBindingW • NdrClientCall2	• HKLM\System\CurrentControlSet\Control\SESSION MANAGER • HKLM\SOFTWARE\Microsoft\OLE • HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL
QuasarRAT	• NdrClientCall2	• HKLM\System\CurrentControlSet\Control\SESSION MANAGER • HKLM\SOFTWARE\Microsoft\OLE • HKLM\SOFTWARE\Microsoft \Rpc\Extensions\NdrOleExtDLL

Tabel 5 merupakah hasil *behaviour* inisiasi dari *remote access software* yaitu AeroAdmin dan *malware* RAT. Dapat terlihat secara garis besar ada perbedaan dari API Windows yang digunakan dan *Registry* Windows yang diakses. *Remote access software* menggunakan API untuk mengatur Winsock seperti accept, recv, WSAGetLastError, setsockopt dan send sebagai bentuk komunikasi antar sistem yang menggunakan AeroAdmin sebagai *remote access software. Remote access software* melakukan validasi sebelum berhubungan dengan sistem tujuannya, terbukti pada *registry* Windows penggunaan HKLM\System\CurrentControlSet\Control\Lsa untuk melakukan proses *Local Secutiry Authority* (LSA). *Remote access software* juga langsung mengeksekusi *remote* desktop ketika sudah terhubung dengan sistem tujuannya terbukti pada penggunaan *registry* Windows HKLM\Hardware\DeviceMap\Video untuk melihat daftar *display* komputer tujuan.

4. Kesimpulan

Berikut adalah kesimpulan dari penelitian ini berdasarkan pengujian dan analisis yang dilakukan:

1. *Dynamic malware analysis* dari *malware* RAT dilakukan dengan mengeksekusi sebuah *backdoor* dari aplikasi server RAT pada *environment* yang terisolasi agar *malware* RAT tidak menginfeksi sistem lain. *Dynamic malware detection tools* berbasis *behaviour* digunakan ketika *backdoor* membuat koneksi antara sistem yang menyerang dan yang terinfeksi. *Malware behaviour* ketika menggunakan API dan mengakses *Registry Windows* didapatkan saat aplikasi server RAT mengeksekusi fitur-fitur *malware* RAT
2. *Malware* RAT *behaviour* ketika diinisiasi menggunakan API dan mengakses *Registry* Windows berhubungan dengan RPC dan membuat koneksi antara sistem terinfeksi dengan aplikasi server RAT. *Malware* RAT *behaviour* saat *Keylogger* dieksekusi menggunakan API Windows dengan kategori Windows *Application* UI *Development* untuk mendapatkan aktivitas *keyboard* dan mengirim hasil aktivitas tersebut ke aplikasi server RAT. *Malware* RAT *behaviour* saat *File* Transfer dieksekusi menggunakan API dengan kategori Data *Access and Storage* untuk mendapatkan informasi *disk drive* sistem terinfeksi dan mengirimkan *file* yang ditargetkan. *Malware* RAT *behaviour* saat *Remote* Desktop

dieksekusi menggunakan API dan mengakses *Registry* Windows berhubungan dengan *Graphic and Gaming* dan mendapatkan informasi *display* dan penggunaan grafik atau objek pada sistem yang terinfeksi

3. Perbedaan terdapat pada *behaviour inisasi* API dan *Registry* Windows yang digunakan oleh *remote access software* dan juga cara penggunaannya. *Remote access software* akan langsung mengeksekusi fitur *Remote* Desktop sedangkan *malware* RAT tidak langsung mengeksekusi fitur *Remote* Desktop.

5. Saran

Berikut adalah saran yang dapat membantu untuk pengembangan penelitian selanjutnya adalah:

1. *Dynamic malware analysis detection tools* yang digunakan dapat ditambahkan lagi untuk mengembangkan hasil analisis dengan data yang lebih lengkap
2. Skenario pengujian *malware RAT behaviour* yang digunakan dapat dikembangkan lagi untuk menghasilkan *malware RAT behaviour* yang baru
3. *Malware* RAT *behaviour* yang didapatkan dapat dikembangkan lagi menggunakan *environment* dengan *hardware* yang asli. Sehingga, *malware behaviour* yang didapatkan menjadi lebih *real-time* dengan sistem yang terinfeksi
4. Untuk penelitian *malware behaviour* selanjutnya, dapat menggunakan *malware* yang lain seperti *worm*, *spyware*, *rootkit*, *ransomware*, dan masih banyak lagi.

Daftar Pustaka:

[1] Alazab, M. (2015). Profiling and classifying the behavior of malicious codes. *Journal of Systems and Software, 100*, 91-102. Diakses pada 15 Juni, 2019, diambil dari https://www.sciencedirect.com/science/article/pii/S0164121214002283.

[2] Aman, W. (2014). A Framework for Analysis and Comparison of Dynamic Malware Analysis Tools. *International Journal of Network Security & Its Applications,6*(5), 63-74. Diakses pada July 2, 2019, diambil dari https://arxiv.org/abs/1410.2131.

[3] Arbez, G., & Birta, L. G. (2016). A tutorial on ABCmod: An Activity Based discrete event Conceptual modelling framework. *2016 Winter Simulation Conference (WSC)*. Diakses pada 1 Oktober, 2018, diambil dari https://ieeexplore.ieee.org/document/7822082.

[4] Aslan, O., & Samet, R. (2017). Investigation of Possibilities to Detect Malware Using Existing Tools. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. Diakses pada 10 Juni, 2019, diambil dari https://ieeexplore.ieee.org/document/8308437.

[5] Bailey, M., Oberheide, J., Andersen, J., Mao, Z. M., Jahanian, F., & Nazario, J. (2007, September 05). Automated Classification and Analysis of Internet Malware. Diakses pada 9 Juni, 2019, diambil dari https://link.springer.com/chapter/10.1007/978-3-540-74320-0_10.

[6] Bazrafshan, Z., Hashemi, H., Fard, S. M., & Hamzeh, A. (2013). A survey on heuristic malware detection techniques. *The 5th Conference on Information and Knowledge Technology*. Diakses pada 8 Juni, 2019, diambil dari https://www.researchgate.net/publication/260729684_A_survey_on_heuristic_malware_detection_techniques.

[7] Bhojani, N. (2014, October). Malware Analysis. Diakses pada 5 Juni, 2019, diambil dari https://www.researchgate.net/publication/267777154_Malware_Analysis.

[8] Budiman, J. (2016). Analysis on Remote Access Trojan Role in Advance Persistent Threat. Diakses pada 5 Juni, 2019, diambil dari https://figshare.com/articles/Analysis_on_Remote_Access_Trojan_Role_in_Advance_Persistent_Threat_A_Concern_for_Cyber_Criminal_Investigations/3510224/1.

[9] Cloonan, J. (2017, April 11). Advanced Malware Detection - Signatures vs. Behavior Analysis. Diakses pada 23 September, 2018, diambil dari https://www.infosecurity-magazine.com/opinions/malware-detection-signatures/

[10] Eze, A. O., & Chukwunonso, C. (2018). Malware Analysis and Mitigation in Information Preservation. *IOSR Journal of Computer Engineering (IOSR-JCE),20*(4), 1st ser., 53-62. Diakses pada 1 Juni, 2019, diambil dari http://www.iosrjournals.org/iosr-jce/papers/Vol20-issue4/Version-1/H2004015362.pdf

[11] Gardåsen, K. T. (2014). Detecting Remote Administration Trojans through Dynamic Analysis using Finite-State. *MachinesMaster of Science in Information Security 30 ECTS*. Diakses pada 18 Mei, 2019, diambil dari https://brage.bibsys.no/xmlui/bitstream/handle/11250/198379/KTGardasen.pdf.

[12] Gavitt, B. D. (2008). Forensic Analysis of the Windows Registry in Memory. *The Digital Forensic Research Conference DFRWS 2008 USA*. Diakses pada 5 Juni, 2019, diambil dari https://www.sciencedirect.com/science/article/pii/S1742287608000297.

[13] Gierow, H., & Benzmüller, R. (2018, September 07). More attacks are launched from the web. Read more to find out about the most recent malware targeting users. Diakses pada 25 September, 2018, diambil dari https://www.gdatasoftware.com/blog/2018/09/31037-malware-figures-first-half-2018-danger-web

[14] Gupta, S., Sharma, H., & Kaur, S. (2016). Malware Characterization Using Windows API Call Sequences. *Security, Privacy, and Applied Cryptography Engineering Lecture Notes in Computer Science*,271-280. Diakses pada 22 Mei, 2019, diambil dari https://link.springer.com/chapter/10.1007/978-3-319-49445-6_15.

[15] Insikt Group. (2019, March 14). Talking to RATs: Assessing Corporate Risk by Analyzing Remote Access Trojan Infections. Diakses pada 25 Juni, 2019, diambil dari https://www.recordedfuture.com/rat-corporate-risk-assessment/

[16] Meshram, M. G. (2015). Investigating the Artifacts Using Windows Registry and Log Files. *International Journal of Computer Science and Mobile Computing*,*4*(6), 625-631. Diakses pada 4 Juni, 2019, diambil dari https://www.semanticscholar.org/paper/Investigating-the-Artifacts-Using-Windows-Registry-Meshram-Kapgate/fd29b30b1df9917a498e7d28bba7b62fc96c576a.

[17] Mujumdar, A., & Masiwal, G. (2013). Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), 2*(6). Diakses pada 20 Mei, 2019, diambil dari https://www.semanticscholar.org/paper/Analysis-of-Signature-Based-and-Behavior-Based-Mujumdar-Masiwal/418bbd845325f7634f8b99d91fac722bef1c2bb5.

[18] Rao, V., & Hande, K. (2017). A comparative study of static, dynamic and hybrid analysis techniques for android malware detection. *The International Journal of Engineering Development and Research*,*5*(2). Diakses pada 2 Juli, 2019, diambil dari https://www.researchgate.net/publication/288905288_A_comparison_of_static_dynamic_and_hybrid_analysis_for_malware_detection.

[19] Saputra, R. W. (2016). A Survey of Cyber Crime in Indonesia. *2016 International Conference on ICT For Smart Society*. Diakses pada 15 September, 2018, diambil dari https://ieeexplore.ieee.org/document/7792846.

[20] Sihwail, R., Omar, K., & Ariffin, K. A. (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), 1662. Diakses 2 Juli, 2019, diambil dari http://www.insightsociety.org/ojaseit/index.php/ijaseit/article/view/6827

[21] Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. Diakses pada 25 September, 2018, diambil dari https://repo.zenk-security.com/Virus-Infections-Detections-Preventions/Practical_Malware_Analysis.pdf.

[22] Tahir, R. (2018). International Journal of Education and Management Engineering. *A Study on Malware and Malware Detection Techniques*,*8*(2), 20-30. Diakses pada 15 Mei, 2019, diambil dari https://www.researchgate.net/publication/324592375_A_Study_on_Malware_and_Malware_Detection_Techniques.

[23] Uppal, D., Mehra, V., & Verma, V. (2014). International Journal on Computational Science & Applications. *Basic Survey on Malware Analysis, Tools and Techniques*,*4*(1), 103-112. Diakses pada 5 April, 2019, diambil dari https://www.semanticscholar.org/paper/Basic-survey-on-Malware-Analysis,-Tools-and-Uppal-Mehra/b25479a230816e8566df0937d9ff9265754f826d.

[24] Zalavadiya, N. (2017). International Journal of Innovative Research in Computer and Communication Engineering. *A Methodology of Malware Analysis, Tools and Technique for Windows Platform – RAT Analysis*,*5*(3). Diakses pada 25 September, 2018, diambil dari http://www.ijircce.com/upload/2017/march/253_A Methodology.pdf

[25] Zolkipli, M. F., & Jantan, A. (2010). 2010 Second International Conference on Network Applications, Protocols and Services. *Malware Behavior Analysis: Learning and Understanding Current Malware Threats*. Diakses pada 14 Mei, 2019, diambil dari https://ieeexplore.ieee.org/document/5635801.