View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by JEPIN (Jurnal Edukasi dan Penelitian Informatika)

JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 5
No. 1
April 2019

Reverse Engineering untuk Analisis *Malware Remote Access Trojan*

Aldy Putra Aldya [#1], Nur Widiyasono[#2], Tesa Pajar Setia [#3]

[#]*Jurusan Informatika, Fakultas Teknik, Universitas Siliwangi,*
Jln. Siliwangi no 24, Kota Tasikmalaya, 46115, Indonesia
[1]aldy@unsil.ac.id
[2]nur.widiyasono@unsil.ac.id
[3] tesa.paja14@student.unsil.ac.id

***Abstrak*— Para *hacker* menggunakan *malware Remote Access Trojan* untuk merusak sistem kemudian mencuri data para korbannya. Diperlukan analisis mendalam mengenai *malware* baru-baru ini karena *malware* dapat berkamuflase seperti sistem tidak dicurigai. Penggunaan teknik basic analysis sangat tergantung pada perilaku *malware* yang dianalisis, analisis akan sulit ketika ditemukan *malware* baru yang menggunakan suatu teknik baru. *Reverse engineering* merupakan salah satu solusi untuk melakukan analisis *malware* karena menggunakan teknik *reverse engineering* kode pada *malware* dapat diketahui. *Malware Flawed ammyy* ini merupakan *software* yang disalahgunakan dari *Ammyy Admin* versi 3 oleh *hacker* TA505. Penelitian ini bertujuan untuk bagaimana alur untuk melakukkan identikasi *malware* kususnya *malware* RAT dengan teknik reverse engineering dan tools yang bias digunakan. Penelitian ini menggunakan metodologi deskriptif,. Hasil dari penelitian menunjukan bahwa alur untuk melakukan *reverse engineering* dan *tools* yang dapat digunakan.**

Kata kunci*— *Flawed Ammyy, Reverse Engineering, Remote Access Trojan

I. PENDAHULUAN

Internet dapat membantu seseorang memanfaatkan banyak layanan hanya dengan bantuan beberapa klik, dengan internet system yang begitu rumit menjadi lebih praktis, murah dan lain-lain [1]. Meningkatnya pengguna *internet* membuat kejahatan merambah ke dunia maya yang sering disebut sebagai *cybercrime* [2]. *Cybercrime* yang digunakan oleh penyerang semakin beragam dan kompleks. Serangan tersebut diantranya melibatkan *malicious software* atau yang biasa disebut *malware* yang merupakan suatu program jahat [3].

Beragam tujuan yang dimiliki para pelaku ini beberapa diantaranya adalah untuk melakukan aktifitas berbahaya yang berdampak sangat merugikan bagi para korbannya, antara lain seperti penyadapan serta pencurian informasi pribadi [4][5]. Beberapa *malware* berbahaya seperti *Virus, Worm, Trojan Horse*, juga bisa membuat *Back Door* yang dapat melakukan pencurian informasi pribadi atau mengambil kendali sistem yang telah terinfeksi [6]. *Malware* sering masuk ke sistem melalui *file* yang diunduh. *Malware* yang telah memasuki sistem, *malware* akan melakukan aktivitas dan merusak seluruh system [7].

Seorang *investigator* harus memiliki kemampuan untuk melakukan analisa *malware* dalam setiap melakukan investigasi. Meningkatnya sejumlah *malware* yang dapat berevolusi dan mampu beradaptasinya terhadap perangkat analisis yang selama ini digunakan sehingga sulit untuk dilakukan ananlisis [8]. Analisa *malware* dengan menggunakan *Reverse engineering* merupakan salah satu solusi yang bisa digunakan saat ini. *Reverse engineering* dalam analisis *malware* berguna untuk ekstraksi data yang memuat informasi yang ada didalam *malware* [9].

Para peneliti Proofpoint telah menemukan *malware Trojan* akses jarak jauh yang sebelumnya tidak terdokumentasi yang disebut *Flawed ammyy rat* [10]. *Malware Flawed ammyy rat* dibuat dari kode *Ammyy admin* versi versi 3 yang disalah gunakan. *Ammyy admin* merupakan perangkat lunak desktop jarak jauh yang digunakan jutaan konsumen dan bisnis untuk menangani *remote control* dan *diagnosis* pada platfom *Windows* [11]. Penyerang yang melakukan penyebaran *malware Flawed ammyy rat* merupakan kelompok peretas TA505 yang terkenal karena menyebarkan *spam malware* seperti Trojan Dridex perbankan, Locky ransomware, dan Jaff ransomware. [12]

Penelitian yang telah dilakukan sebagai dasar penelitian ini diantaranya Peneltian [1] telah dilakukan analisis statis dan analisis dinamis pada *malware* DrakComet. Hasil dari penelitian tersebut menguraikan metodologi yang efektif dan efisien yang dapat meningkatkan kinerja deteksi dan penghapusan *malware* yang dikumpulkan. Analisis dinamis merupakan cara terbaik untuk melakkan analisis *sample* sebuah *malware*.

Penelitian [13] menguraikan hasil komparasi terhadap metode analisis *malware* statis. Peneliti telah melakukan ekstraksi 11 vektor kelompok kecil untuk 600 *malware*, dan berhasil melakukan klasifikasi lebih dari setengah kode kedalam kelompok yang sesuai menggunakan vektor. Pemeriksaan yang cermat pada kode biner juga

Submitted 26-09-2018; Revised 21-03-2019; Accepted 01-04-2019

menegaskan bahwa vektor bit telah mengklasifikasikannya dengan cara yang benar. Hasil dari eksperimen menunjukkan bahwa bit vektor dapat digunakan secara efektif untuk melakukan analisis *malware* statis, dan vektor bit grup dapat membantu diklasifikasikan *malware*s kedalam kelompok yang sesuai.

Penelitian [9] telah melakukan proses *reverse engineering* pada *malware* Biscuit. Cara kerja *malware* tersebut adalah adanya auto request untuk koneksi ke ip *address* tertentu yaitu ip *address* 114.101.115.115. Selanjutnya proses *reverse engineering* melalui penulusuran perintah: ***bdkzt, ckzjqk, download, exe, exit*** dan ***lists*** telah dapat memetakan bagaimana cara kerja dari *malware* Biscuit.

Peneliti [14] telah melakukan *reverse engineering* untuk membongkar kode ransomware kemudian analisis lebih lanjut. Hasil dari penelitian tersebut menunjukkan meskipun penggunaan enkripsi tangguh, seperti ransomware lain yang menggunakan serangan dengan struktur dan kriptografi primitif. Analisis ini telah menuntun pada kesimpulan bahwa strain ransomware tidak serumit dilaporkan sebelumnya. Analisis praktis dan terperinci ini mencoba memberi kesadaran kepada komunitas bisnis tentang realitas dan Pentingnya keamanan TI, pencegahan, pemulihan dan keterbatasannya.

Peniliti [15] telah melaukan *reverse engineering* pada *malware botnet*. Tujuan utama dari penelitian ini meupakan untuk menentukan pendekatan yang paling memadai sebagai pencegahan dari insiden *botnet*. Gangguan jaringan pada proses ini merupakan aktivitas online ilegal dan pencurian data organisasi, ini dapat dicegah dan bahkan bot sistem Intrusion Prevention spesifik dapat dikembangkan. penelitian menjamin aliran data yang dikonfirmasi dalam ruang digital oleh komunikasi e-governance yang diasuransikan untuk setiap negara dari *terorisme* cyber.

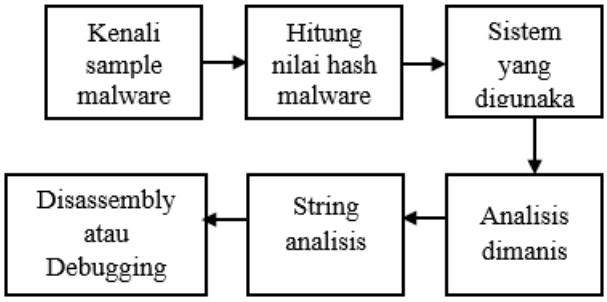
Penelitian [16] melakukan *reverse engineering* pada *malware Flawed ammyy rat*. Hasil dari penelitian dengan analisis dinamis dan *reverse engineering* dengan menggunakan tahap *disassembly* menunjukkan pergerakan dari *malware* dimana *malware* ini tidak dapat berjalan pada sistem dalam kedaan mode DOS. *Malware* ini memanipulasi system dengan cara menjalankan aplikasi Ammyy Admin 3, karena Amyy Admin merupakan aplikasi yang aman maka memberi akses pada *malware* tersebut, kemudian *malware* menyimpan data korban dan melakukan singkron dengan ip address 103.208.86.69.

Malware RAT ini tidak cukup hanya dilakukan dengan teknik basic analysis *malware* karena tidak semua *malware* rat ini berjalan pada system. *Malware* rat ini harus terkoneksi dengan attacker sebagai tuan yang melakukan perintah selanjutnya, seperti mengaktifkan keyloger, menggerakkan pointer korban, bahkan dapat mengaktifkan webcam korban tanpa sepengtahuan. Melakukan hal tersebut system mesti terhubung dengan suatu jaringan, namun untuk melakukan basic analysis *malware* diperlukan system yang steril, tidak terhubung dengan internet karena dapat menyebabkan penyebaran *malware* melalui jaringan. Beda halnya dengan *malware* bonet virus worm ransomware dan lain-lain yang dapat melakukan eksekusi sesuai dengan arsitektur yang telah dibuat.

Tujuan dari penelitian ini untuk melengkapi penelitian [16] dimana penelitian tersebut hanya menggunakan 1 tahap *reverse engineering disassembly*, untuk penelitian kali ini akan menggunakaan tahapan lainnyan seperti identifikasi *sample malware* untuk mengetahui *type file malware*, hitung nilai *hash* untuk megetahui nilai *hash* pada *malware*, dan *string analysis* untuk mengetahui apakah *function* yang didapat dari teknik *disassembly* itu sesuai atau ada *function* lain yang tidak terdefinisi dengan teknik *disassembly*. Kemudian menguraikan alur untuk melakukan *reverse engineering* khususnya *malware* Remote Access Trojan dan rekomendasi tools yang dapat digunakan untuk melakukann *reverse engineering malware* RAT.

II. METODOLOGI

Metodologi yang digunakan pada penelitian ini menggunakan metodologi deskriptif untuk menjelaskan bagaimana proses penelitian. Melakukan reverse engineering dilakukan beberapa tahapan seperti pada gambar 1:



Gambar. 1 Alur *reverse engineering*

A. Kenali *sample malware*

Tahap setelah mendapatkan *sample malware* identifikasi *type file* seperti apa *malware* tersebut. Ini diperlukan untuk menetukan sistem yang bagaimana digunakan nanti dalam melakukan analisis dinamis.

B. Hitung nilai *hash sample malware*

Hash dari sebuah program memiliki identitas seperti halnya sidik jari pada manusia, maka perlu dihitung sebagai bukti digital

C. Sistem yang digunakan

Tahap setelah mengetahui *type file* system yang akan dibangun disesuaikan dengan *type file*. Seperti *type file sample malware executable* sistem yang akan digunakan adalah *Windows*.

D. Analisis dinamis

Tahap analisis dinamis diperlukan untuk melihat pergerakan *malware* ketika dijalankan pada sistem. Penetian kali ini menggunkan teknik *sandbox* unuk melakukan analisis dinamis

E. *String analysis*

Tahap *string analisis* menampilkan nilai yang akan dilakukan proses *load* oleh *sample malware* ketika dieksekusi. Hal ini yang menjadikan dalam proses *reverse engineering* harus dilakukan *string analisis* untuk mendapatkan bukti kuat dari *sample malware*.

F. *Disassembly atau debugging*

Tahap ini *malware* akan dibuka *source kode* yang terdapat pada *malware* tersebut. Teknik *disassembly* merupakan teknik *reverse engineering* untuk menerjemahkan dari bahasa mesin ke bahasa yang mudah dimengerti manusia. *Debugging* digunakan untuk melakukan pengujian dari setiap proses inti yang ada didalam *malware*. Penelitian kali ini cukup menggunakan *disassembly*.

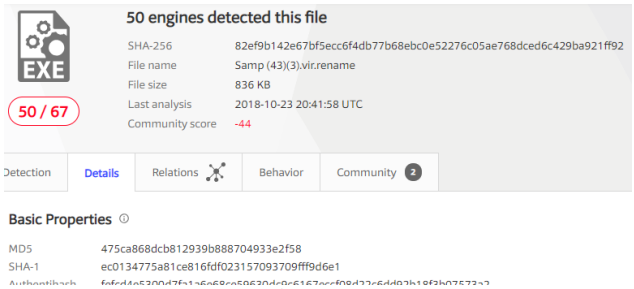
TABEL I TOOLS UNTUK REVERSE ENGINEERING

Tools	Keterangan
Virustotal	Digunakan untuk mengtahui informasi mengenai *sample malware*
Hashmyfile	Digunakan untuk menghitung nilai *hash*
Hybrird analysis	Digunakan untuk *sandbox online*
Bintext	Digunakan untuk *string analysis*
Ida pro	Digunakan untuk *disassembly* atau *debugging*

III. HASIL DAN PEMBAHASAN

A. Kenali *sample malware*

Tahap ini menggunakan *tools virustotal* untuk mendaptakan informasi mengenai *sample malware* yang didapat. *Sample malware* yang didapat diunggah pada *website virustotal*. Hasil unggahan *malware* pada *website* virustotal dapat dilihat pada gambar 2.

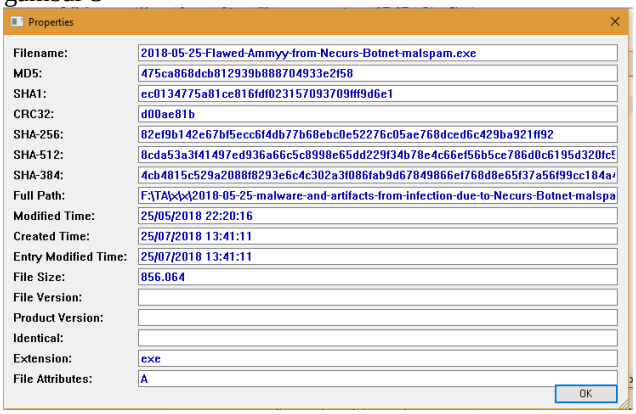


Gambar 2 informasi dari virustotal

Informasi yang didapat mengenai *sample malware* adalah *type file malware* merupakan *file executable* dengan *size file* 836kb kemudian dari 67 anti virus 50 diantaranya menaytakan bahwa *sample malware* berbahaya dengan kategori *Trojan*.

B. Hitung nilai *hash sample malware*

Melakukan penghitungan hash *sample malware* yang didapat gunakan tools *hashmyfile*. Buka aplikasi *hashmyfile* klik *File*, kemudian klik *Add File*, lalu pilih *file sample malware*, setelah itu *open file*. Hasil dari perhitungan hash *sample malware* dapat dilihat pada gambar 3



Gambar 3 Nilai *hash* hasil dari *hashmyfile*

Gambar 3 menunjukan nilai *hash* dari *sample malware* sama dengan nilai hash md5 *malware flawed ammy rat* adalah 475ca868dcb812939b888704933e2f58 dengan demikian *sample malware* yang didapat merupakan *file* asli.

C. Sistem yang digunakan

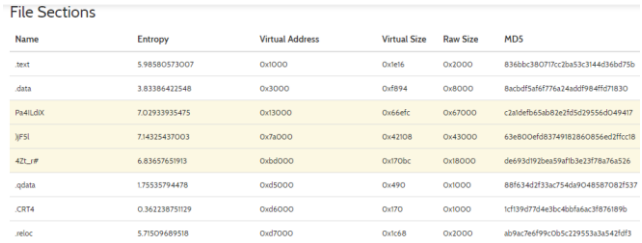
Type file sample malware merupakan *executable* untuk melakukan analisis dinamis menggunakan system operasi *windows*, untuk memudahkan pergerakan *malware* dipilih *windows* 7 32bit dengan jaringan *TOR* ini digunakan untuk menghindari celah perekaman sidik jari pada jaringan eksternal.

TABEL II SPESIFIKASI *VIRTUAL*

Aplikasi VM	*Sandbox online*
Sistem operasi	*Windows7*
Jaringan	Jaringan *TOR*

D. Analisis dinamis

Virtual mesin *sandbox online* digunakan untuk melakukan analisis dinamis *malware*. *Tools* yang digunakan adalah *hybrid analysis*. *Sample malware* diunggah ke *website hybrid analysis*, kemudian pilih sistem yang ingin digunakan setelah itu buka hasil report *hybrid analysis*.

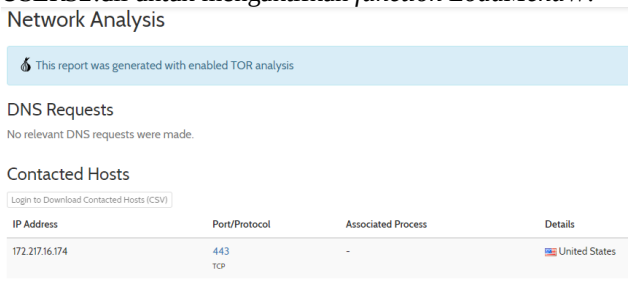


Gambar 4 *file sections malware flawed ammyy rat*

Merujuk gambar 4 terdapat 8 *file sections* setelah *malware flawed ammyy rat* dijalankan pada *hybrid analysis* dintaranya `.text`, `.data`, `Pa4Ldix`, `)F5l`, `4Zt_r#`, `.qdata`, `.CRT4`, dan `.reloc`

Gambar. 5 *File* import *malware flawed ammyy rat*

Merujuk pada gambar 5 *malware* melakukan beberapa import beberapa *file* pada sistem yang telah terinfeksi. *File GDI32.dll* untuk mengaktifkan *function GetBrushOrgEX*. *File KERNEL32.dll* untuk mengaktifkan *function AreFileApisaANSI, GetBinaryTypeA, GetCompressedFileSizeA, GetModuleFileNameA, GetModuleHandlaA, SetFileApisToANSI. File ntdll.dll* untuk mengaktifkan *memset. File RASAPI32.dll* untuk mengaktifkan *function RasRenameEntryA. File SHLWAPI.dll* tidak ada *function* yang diaktifkan. *File USER32*.dll untuk mengaktifkan *function LoadMenuW*.



Gambar. 6 *Network analysis pada hybrid analysis*

Merujuk gambar 6 hasil analisis jaringan dengan mengaktifkan *TOR analysis* menunjukkan bahwa malware melakukan koneksi dengan jaringan eksternal dengan ip *address* 172.217.16.174 dengan menggunakan *port* 443 dimana *server* tersebut berada di Amerika Serikat.

E. *String analysis*

Proses string analisis dilakukan pada karakter ASCII yang ada didalam program *malware*. Untuk melakukan string analisis digunakan program Bintext, seperti pada gambar 7

File pos	Mem pos	ID	Text
0000000CD39C	0000004D539C	0	GetCompressedFileSizeA
0000000CD3B6	0000004D53B6	0	AreFileApisANSI
0000000CD3C8	0000004D53C8	0	SetFileApisToANSI
0000000CD3DC	0000004D53DC	0	SetFileApisToOEM
0000000CD3F0	0000004D53F0	0	GetModuleHandleA
0000000CD404	0000004D5404	0	GetModuleFileNameA
0000000CD41A	0000004D541A	0	GetBinaryTypeA
0000000CD42A	0000004D542A	0	KERNEL32.dll
0000000CD43A	0000004D543A	0	memset
0000000CD442	0000004D5442	0	ntdll.dll
0000000CD44E	0000004D544E	0	RasRenameEntryA
0000000CD45E	0000004D545E	0	RASAPI32.dll
0000000CD46E	0000004D546E	0	LoadMenuW
0000000CD478	0000004D5478	0	USER32.dll

Gambar 7 String analysis menggunakan bintext

Gambar 7 menunjukkan bahwa hasil dari String analysis menggunakan bintext. *Sring* yang digunakan oleh *malware flawed ammyy rat* diantaranya *GetBrushOrgEx, GetCompressedFileSizeA, AreFileApisAnsi, SetFileApisToANSI, SetFileApisToOEM, GetModuleHandleA, GetModuleFileNameA, GetBinaryType, memset, RasRenameEntryA, LoadMenuW*

F. *Disassembly atau debugging*

Proses *disassembly* ini menggunakan tools *IDAPro* Setelah *malware* terbuka selanjutnya adalah melakukan analisa *command* yang ada didalam *malware* yang telah dilist di atas hasil dari *dissaembly* di tunjukan pada Gambar 8.

Address	Ordinal	Name	Library
0000000...		GetBrushOrgEx	GDI32
0000000...		GetCompressedFileSizeA	KERNEL32
0000000...		AreFileApisANSI	KERNEL32
0000000...		SetFileApisToANSI	KERNEL32
0000000...		SetFileApisToOEM	KERNEL32
0000000...		GetModuleHandleA	KERNEL32
0000000...		GetModuleFileNameA	KERNEL32
0000000...		GetBinaryTypeA	KERNEL32
0000000...		RasRenameEntryA	RASAPI32
0000000...	152	StrCmpNCW	SHLWAPI
0000000...		LoadMenuW	USER32
0000000...		memset	ntdll

Gambar. 8 Hasil *dissaembly* menggunkan ida pro

```
BOOL _stdcall Get GetBrushOrgEX(HDC hdc,
LPPOINT lppt)
extrn GetBrushOrgEX:dword
```

Malware memannggil *function* tersebut untuk mengambil alih fungsi pointer agar dapat mempermudah pencurian data.

```
DWORD _stdcall GetCompressedFileSizeA
(LPCSTR lpFileName, LPDWORD
lpFileSizeHigh)
extrn GetCompressedFileSizeA:dword
```

Malware memanggil *function* tersebut untuk melakukan *proceses compress file* tertentu kemudian menyimpannya.

```
BOOL _stdcall AreFileApisAnsi ()
extrn AreFileApisAnsi:dword
```

Perintah dengan *function* tersebut *malware* dapat menentukan apakah fungsi *file* I / O menggunakan halaman kode karakter *ANSI* atau *OEM*.

```
Void _stdcall SetFileApisToAnsi()
extrn SetFileApisToAnsi:dword
```

Function tersebut *malware* dapat membuat *file* I / O berfungsi untuk menggunakan halaman kode set karakter *ANSI* untuk proses saat ini. Fungsi ini berguna untuk operasi input dan output masukan 8-bit.

```
Void _stdcall SetFileApisToOEM()
extrn SetFileApisToOEM:dword
```

Function tersebut dapat membuat file I / O berfungsi untuk proses untuk menggunakan halaman kode karakter set *OEM*. Fungsi ini berguna untuk operasi input dan output masukan 8-bit.

```
HMDULE _stdcall GetModuleHandleA(LPCSTR
lpModuleName)
extrn GetModuleHandleA:dword
```

Malware dengan *function* tersebut memilih file yang memenuhi syarat yang mengandung modul tertentu.

```
DOWRD _stdcall
GetModuleFileNameA(HMODULE hModule,
LPSTR lpFilename, DOWRD nSize)
extrn GetModuleFileNameA:dword
```

Malware dengan menggunakan *function* tersebut dapat meneangani modul yang telah ditentukan

```
BOOL _stdcall GetBinaryTypeA(LPCSTR
lpApplictionName, LPDWORD lpBinaryType)
extrn GetBinaryTypeA:dword
```

Function tersebut *malware* dapat menentukan apakah *file* dapat di eksekusi, jika subsistem mana yang menjalankan file yang dapat dieksekusi

```
DWORD _stdcall RasRenameEntryA(LPCSTR,
LPCSTR LPCSTR)
extrn RasRenameEntryA:dword
```

Malware dengan *function RasRenameEntry* dapat mengubah nama *entri* di buku telepon.

```
extrn StrCmpNCW:dword
```

Malware dapat membandingkan nomor spesifik sejumlah karakter dari awal dua *string* menggunakan C *run-time* (*ASCII*)

```
HMENU _stdcall LoadMenuW(HINSTANCE
hInstance, LPCWSTR lp MenuName)
extrn RasRenameEntryA:dword
```

Malware dapat memuat sumber daya menu yang spesifik dari file yang dapat *executable* (.exe) yang terkait dengan instance aplikasi.

```
void _cdecl memset(void *Dst, int Val,
size_t Size)
extrn _imp_memset:dword
```

Malware melakukan *setel buffer* ke karakter yang ditentukan

Hasil dari reverse engineering malware flawed ammyy rat menunjuan bagimana pergerkan dari malware tersebut. *Malware* ternyata melakukan koneksi dengan jaringan eksternal dengan ip *address* 172.217.16.174. malware memiliki beberapa file section diantaranya *.text, .data, Pa4Ldix,)F5l, 4Zt_r#, .qdata, .CRT4, dan .reloc* kemdian malware meliki file import *File GDI32.dll* untuk mengaktifkan *function GetBrushOrgEX. File KERNEL32.dll* untuk mengaktifkan *function AreFileApisaANSI, GetBinaryTypeA, GetCompressedFileSizeA, GetModuleFileNameA, GetModuleHandlaA, SetFileApisToANSI. File ntdll.dll* untuk mengaktifkan *memset. File RASAPI32.dll* untuk mengaktifkan *function RasRenameEntryA. File SHLWAPI.dll* tidak ada *function* yang diaktifkan. *File USER32*.dll untuk mengaktifkan *function LoadMenuW*.

IV. KESIMPULAN

Penelitian ini yang berjudul "*Reverse Engineering* Untuk Analisis *Malware Remote Access Trojan*" berdasarkan penelitian sebelumnya yang telah di lakukan, sehingga dapat disimpulkan sebagai berikut:

Tahapan untuk melakukan *reverse engineering* pada suatu *malware* kususnya *malware RAT* adalah sebagai berikut: kenali *sample malware* menggunakan *tools virustotal*, hitung nilai *hash sample malware* menggunakan *tools hashmyfile*, tentukan sistem yang digunakan untuk melakukan analisis dinamis, *monitoring* pergerakan *malware* dengan menggunakan metode analisis dinamis *basic* atau dapat menggunakan *sandbox* untuk *sandbox online* dapat menggunakan tool *hybrid analysis*, *string analysis malware* tersebut untuk memperkuat bukti dengan menggunakan tools *bintext*, terkhir lakukan *disassembly* atau *debugging* pada *sample malware* untuk membuka *source malware* tersebut gunakan *tools ida pro*.

Metode *reverse engineering* dengan teknik *string analysis* berhasil menunjukkan 12 *function* yang terdapat pada *sample malware flawed ammyy rat* ini menunjukkan bahwa penelitain sebelumnya telah menunjukkan hasil *reverse engineering* dengan *disassembly* telah terindentifikasi tidak ada *function* yang terlewat, jadi sebaiknya untuk melakukan *reverse engineering* sebaiknya dilakukan terlebih dahulu *string analysis* untuk mengetahui *function* atau mengetahui proses *load* dari *malware* kemedian dilakukan *disassembly* untuk mengetahui *source kode* yang terdapat pada *malware*. Metode *sandbox online* menunjukkan *file section* dan *file import* dari *malware* tersebut.

UCAPAN TERIMA KASIH

Pertama penulis mengucapkan bersyukur kepada Allah swt, karena berkat rahmat dan barokahnya penulis dapat menyelesaikan penelitian ini. Penulis mengucapkan terimakasih kepada para pembimbing yang sabar mengajarkan dan membimbing penulis, kepada orang tua atas segala dukungan dalam bentuk apapun serta pada pihak-pihak yang terkait yang telah membantu penulis dalam menyelesaikan penelitian ini.

REFERENCES

[1] N. Zalavadiya and S. Priyanka, "A Methodology of *Malware* Analysis, Tools and Technique for Windows Platform - RAT Analysus," 2017.

[2] S. C. Y. Hutauruk, F. A. Yulianto and G. B. Satrya, "*Malware* Analysis Pada Windows Operating System Untuk Mendeteksi Trojan," *e-Proceding of Enggineering* , vol. III, no. 2, pp. 3590-3595, 2016.

[3] R. Adenansi and L. A. Novarina, "*Malware* Dynamic," *JOEICT (Jurnal of Education and Information Communication Technology),* vol. 1, no. 1, p. 37, 2017.

[4] D. R. Septani, N. Widiyasono and H. Mubarok, "Investigasi Serangan *Malware* Njrat Pada PC," *Jurnal Edukasi dan Penelitian Informatika (JEPIN),* vol. II, no. 2, pp. 123-128, 2016.

[5] T. A. Cahyanto, V. Wahanggara and D. Ramadana, "Analisis dan Deteksi *Malware* Menggunakan Metode Analisis Dinamis," *JUSTINDO, Jurnal Sistem & Teknologi Informasi Indonesia* , vol. II, no. 1, pp. 19-30, 2017.

[6] U. K. Bavishi and B. M. Jain, "*Malware* Analysis," *International Journals of Advanced Research in Computer Science and Software Engineering,* vol. VII, no. 12, pp. 27-33, 2017.

[7] D. Uppal , V. Mehra and V. Verma, "Basic on *Malware* Analysis, Tools, and Technique," *International Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1,* pp. 103-112, 2014.

[8] A. H. Muhammad, B. Sugiantoro and A. Luthfi, "Metode Klasifikasi dan Analisis Karakteristik *Malware* Menggunakan Konsep Ontologi," *Tenomatika,* vol. IX, no. 2, pp. 16-28, 2017.

[9] H. A. Nugroho and Y. Prayudi, "Penggunaan Teknik Reverse Engineering Pada *Malware* Analysis Untuk Identifikasi Serangan *Malware*," *KNSI 2014, 27-28 Februari 2015, STMIK Dipanegara Makasar,* pp. 1-8, 2015.

[10] Proofpoint Staff, "Proofpoint," 7 Maret 2018. [Online]. Available: https://www.proofpoint.com/us/threat-insight/post/leaked-source-code-ammyy-admin-turned-flawedammyy-rat.

[11] K. Sheridan, "Darkreading," 12 Maret 2018. [Online]. Available: https://www.darkreading.com/endpoint/flawedammyy-rat-campaign-puts-new-spin-on-old-threat/d/d-id/1331248.

[12] A. Saraswat, "Hacking, Hacking Tools, Vulnerability," 10 Maret 2018. [Online]. Available: https://professionalhackers.in/beware-of-flawedammyy-rat-that-steals-credentials-and-record-audio-chat/.

[13] K. Ki-Su, S. Hyo-Jeong and K. Hyong-Shik, "A Bit Vector Based Binary Code Comparison Method for Static *Malware* Analysis," *Journal of Computers,* vol. xiii, no. 5, pp. 545-554, 2018.

[14] A. Zimba, L. Simukonda and M. Chishimba, "Demystifying Ransomware Attacks: Reverse Engineering and Dynamic *Malware* Analysis of WannaCry for Network and Information Security," *ZAMBIA INFORMATION COMMUNICATION TECHNOLOGY (ICT) JOURNAL,* vol. i, no. 1, pp. 35-40, 2017.

[15] B. Thakar and C. Parekh, "Reverse Engineering of Bonet (APT)," *Information and Communication Technology for Intelligent Systems,* vol. ii, no. 1, pp. 252-262, 2017.

[16] T. P. Setia, N. Widiyasono and A. P. Aldya, "Analisis *Malware* Falwed Ammyy RAT Dengan Metode Reverser Engineering," *Jurnal Pengembangan IT (JPIT),* pp. 371-380, 2018.