Buletin Ilmiah Sarjana Teknik Elektro
Vol. 3, No. 2, Agustus 2021, pp. 163-175
ISSN: 2685-9572, DOI: 10.12928/biste.v3i2.4884

GandCrab Ransomware Analysis on Windows Using Static Method

Anisa Oktaviani, Melwin Syafrizal
Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Indonesia

INFORMASI ARTIKEL

Riwayat Artikel:

Dikirimkan 20 September 2021
Direvisi 24 Oktober 2021
Diterima 2 November 2021

Kata Kunci:

GandCrab;
Malware;
Ransomware;
Analisis Statis;

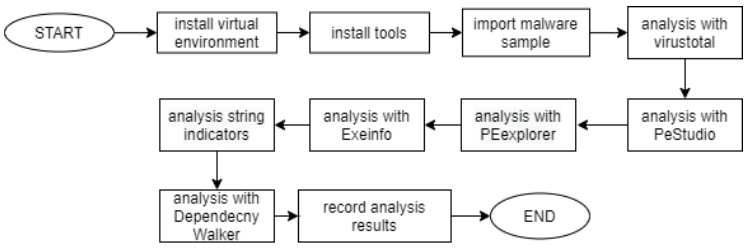
Penulis Korespondensi:

Anisa Oktaviani, Melwin Syafrizal,
Fakultas Ilmu Komputer,
Universitas Amikom Yogyakarta,
Yogyakarta, Indonesia.

Surel:
anisa.oktaviani@students.amikom.ac.id,

Melwin@amikom.ac.id

***ABSTRACT* / ABSTRAK**



Malware-infected operating systems may experience system damage, files or loss of important data. Ransomware is a type of malware that works by attacking the internet network and then encrypting the victim's computer. So that the victim can access his computer again, the victim is asked to redeem (ransom) with some money in the form of Bitcoin. One of them is GandCrab. Gandcrab is a very powerful ransomware and only the creators of Gandcrab know the description of the encrypted files.Static analysis is done by importing malware samples into Virustotal, Dependency walker, PEStudio, Exeinfo PE, and PEexplorer tools to get the strings function, which will then be analyzed to find out how the GandCrab Ransomware works.This study analyzes the gandcrab ransomware malware using a static method. In the Virustotal tool, it was found that the malware sample file was detected as malware with a ratio of 60 out of 70 antimalware. Furthermore, it was found that GandCrab is in PE (portable executable) format, compiled using Microsoft Visual C++ and GndCrab accesses some DLL (dynamic link-library) functions.

Sistem operasi yang terinfeksi *malware* dapat mengalami kerusakan sistem, *file* atau kehilangan data-data penting. *Ransomware* merupakan salah satu jenis *malware* yang bekerja dengan cara menyerang jaringan internet kemudian mengenkripsi komputer korban. Agar korban dapat mengakses komputernya lagi, korban diminta unutk menebus *(ransom)* dengan sejumlah uang dalam bentuk *Bitcoin*. Salah satunya yaitu *GandCrab*. *Gandcrab* merupakan *ransomware* yang sangat kuat dan hanya pembuat *gandcrab* yang mengetahui deskripsi dari *file* yang terenkripsi. Analisis statis dilakukan dengan mengimpor *sample malware* kedalam *tools Virustotal, Dependency walker, PEStudio, Exeinfo PE,* dan *PEexplorer* untuk mendapatkan fungsi *strings* yang kemudian *strings* tersebut akan dianalisa untuk mengetahui cara kerja dari *GandCrab Ransomware.* Penelitian ini melakukan analisis terhadap *malware gandcrab ransomware* dengan menggunakan metode statis. Pada tool *Virustotal,* didapatkan bahwa *file sample malware* terdeteksi sebagai *malware* dengan rasio 60 dari 70 *antimalware.* Selanjutnya ditemukan bahwa *GandCrab* berformat *PE (portable executable),* dikompilasi menggunakan *Microsoft Visual C++* dan *GandCrab* mengakses beberapa fungsi DLL *(dynamic link-library).*

Sitasi Dokumen ini:

A. Oktaviani and M. Syafrizal, "GranCrab Ransomeware Analysis on Windows using Static Method," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 3, no. 2, pp. 163-175, 2021. DOI: 10.12928/biste.v3i2.4884

1. PENDAHULUAN

Malware atau *malicious software* adalah suatu program yang dibuat khusus untuk menginfeksi perangkat lunak atau aplikasi atau dokumen yang tersimpan dalam sebuah sistem. Sistem operasi yang terinfeksi *malware* dapat mengalami kerusakan sistem, *file* atau kehilangan data-data penting [1]. Salah satu jenis *malware* adalah *ransomware*. *Ransomware* bekerja dengan cara menyerang jaringan internet. Selain itu, *ransomware* juga mengenkripsi komputer korban. Untuk dapat mengakses komputernya lagi, korban diminta untuk menebus *(ransom)* dengan sejumlah uang dalam bentuk *Bitcoin*. Berdasarkan data Q3 2020 dari Kaspersky, serangan ransomware mencapai 121.579 korban, di antaranya menyerang pada bidang pendidikan, perawatan kesehatan, tata kelola, dan keuangan [2].

GandCrab merupakan salah satu geng dari *ransomware* yang paling terkenal dan ganas karena mengklaim telah mendapatkan uang tebusan lebih dari US$ 2 miliar dalam waktu 18 bulan. *GandCrab* sempat menyatakan pensiun pada pertengahan tahun 2019 yang lalu, namun virusnya masih banyak tersebar luas. Oleh karena itu, analisis statis pada *GandCrab* merupakan salah satu tindakan yang diperlukan untuk mengetahui karakteristik dari *ransomware* ini. Analisis statis adalah salah satu metode analisis yang dilakukan dengan cara memahami fungsi *source code* dari *malware* untuk mendapatkan informasi yang lengkap dan gambaran detail mekanisme kerja sebuah *malware*. Para peneliti *malware* perlu mengetahui metode ini. Peneliti tertarik melakukan penelitian untuk memahami metode analisis ini lebih lanjut, dan membuat sebuah topik penelitian dengan judul "Analisis *Malware GandCrab Ransomware* Pada Windows Menggunakan Metode Statis".

2. METODE

Dalam penelitian ini dilakukan analisis terhadap *ransomware gandcrab* dengan merancang sebuah *virtual environment* menggunakan Kali Linux. Adapun *virtual environment* digunakan sebagai *emulator hardware* agar *malware* yang di analisis tetap berada di tempat isolasi. Analisis terhadap *malware* menggunakan metode statis yaitu dengan melakukan pengecekan terhadap *strings*. Peneliti akan membongkar *malware* dan kemudian dilakukan pengecekan *strings*, untuk mengetahui struktur, cara kerja, dan fungsi dari *software* tersebut. Pada metode analisis ini, tidak diperlukan eksekusi terhadap *malware.*

2.1. Tujuan Penelitian

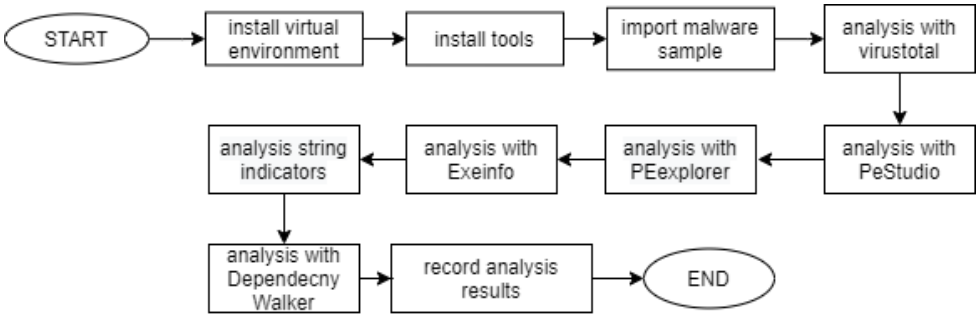
Tujuan yang ingin diraih dalam penelitian ini adalah:

a. Untuk mengetahui karakteristik dan sistem kerja dari *ransomware gandcrab*
b. Membuktikan metode statis dapat digunakan untuk mendeteksi perilaku *malware gandcrab ransomware.*

2.2. Alur Penelitian

Analisis dimulai dengan melakukan pendeteksian *malware* melalui virustotal. Akan muncul seberapa banyak antivirus yang menganggap *malware* ini berbahaya. Selain itu, virustotal juga akan memberikan detail informasi mengenai *malware* yang dianalisis seperti *checksum* SHA256 atau hash MD5 yang berfungsi untuk mengidentifikasi *file malware* tersebut.

Setelah melakukan analisis di virustotal, penelitian ini dilanjutkan dengan analisis di beberapa *tools*. Adapun *tools* tersebut antara lain: Dependency walker, PEStudio, *Exe*info PE, dan PEexplorer. Proses analisis pada virustotal dan PeStudio bertujuan unutk mendapatkan nilai *checksum* dari *malware* yang dianalisis. Kemudian, analisis pada PEexplorer dan *Exe*info dilakukan untuk mengetahui format dari *malware* yang dianalisis dan tipe dari *malware* yang diuji. Dilanjutkan dengan analisis *strings* indicators. Analisis ini bertujuan untuk mendapatkan *strings* dari *malware* yang dianalisis. Terakhir, analisis dilanjutkan dengan *tools* dependency walker. Analisis ini bertujuan untuk mengurutkan dan mengkategorikan *strings* indicator agar lebih mudah untuk di bedah.



Gambar 1. Alur Penelitian

2.2. Alat dan Bahan Penelitian

Dalam proses penelitian ini dibutuhkan perangkat keras dan perangkat lunak pendukung agar analisis berjalan dengan baik. Adapun perangkat keras yang digunakan ialah laptop Toshiba dengan *processor* Intel® Core (TM) i5-2450M CPU @ 2.50GHz, 2501 Mhz, 2 Core(s), 4 Logical Processor(s), model sistem *Satelite C40-A, BIOS* InsydeH20 version 03.72.301.00, memori 4096MB RAM dan *hardisk* sebesar 500gb. Perangkat lunak yang digunakan ialah *virtual box* sebagai *virtual environtment* yang berfungsi untuk menjaga *malware* ttap terisolasi sehingga tidak membahayakan ketika analisis berlangsung. *Operating system* yang digunakan adalah Kali Linux (64 bit) dan Windows 7 (64 bit). Masing-masing menggunakan memori 2048 MB dengan *processor single core*. *Graphics controller* VMSVGA dan *storage* sebesar 80gb.

2.3. Rancangan Sistem

Agar analisis pada penelitian ini berjalan dengan baik, maka diperlukan proses instalasi yang runtut. Dimulai dari instalasi virtual machine, instalasi *tools*, dan pengujian terhadap *malware* yang akan dianalisis. Untuk melakukan analisis *malware*, sangat diharuskan untuk melakukan instalasi virtual machine. Virtual machine berguna untuk menjaga *sample malware* yang akan dianalisis tetap berada di tempat isolasi. Pada penelitian ini, peneliti menggunakan virtual box sebagai virtual *machine environment* dengan sistem operasi kali linux 64-bit dan windows 7 64-bit. Proses instalasi dilakukan dengan menggunakan *file* berekstensi OVA *(Open Virtualization Appliance)*. Dimana, virtual *machine* akan di ekstrak dan diimpor ke dalam perangkat lunak virtualisasi yang telah diinstal pada komputer. Pada penelitian ini, penulis menggunakan software/*tools* yaitu: Dependency walker, PEStudio, *Exe*info PE, dan PEexplorer. Penjelasan setiap *tools* yang digunakan akan dijabarkan pada Tabel 1.

Tabel 1. Penjelasan Tools

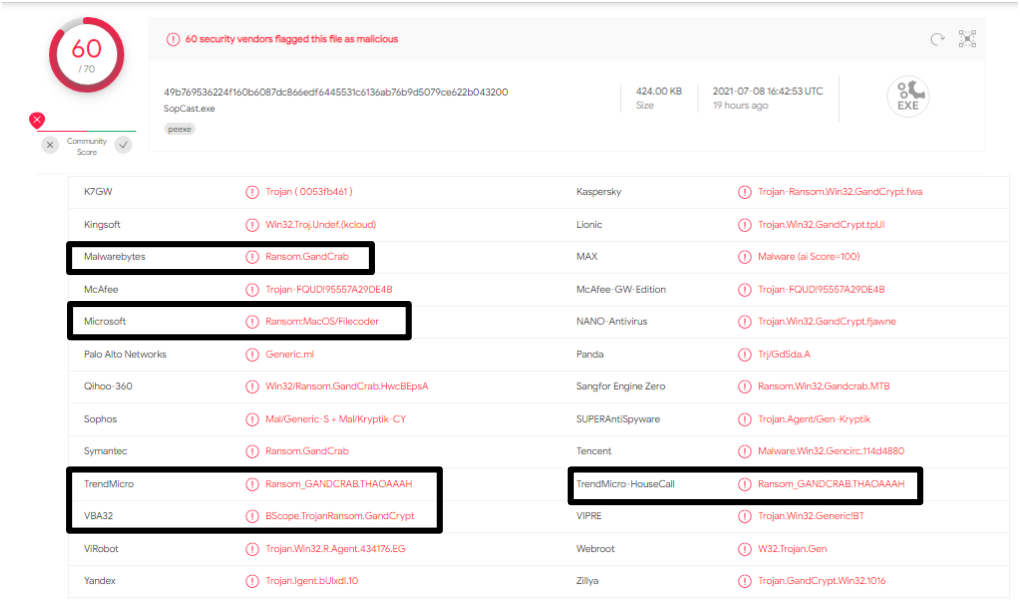
Tools	**Fungsi**
VirusTotal	Menganalisa dan mendeteksi *malware*
*Exe*info PE	Memeriksa *properties* dari *sample malware*
PEexplorer	Memastikan bahwa *sample malware* berformat PE
Dependency walker	Untuk mengetahui daftar yang di impor dari *sample malware*
PEStudio	Menganalisis *strings malware*

3. HASIL DAN PEMBAHASAN

Pada bagian pembahasan, dilakukan pengujian menggunakan perangkat lunak yang sudah disiapkan.

3.1. *Analysis With Virustotal*

Checksum merupakan item data paling kecil yang dihitung dari struktur data untuk digunakan sebagai verifikasi integritasnya. Yaitu untuk memverifikasi bahwa struktur data mengalami perubahan atau tidak. *Checksum* juga dikenal sebagai hash. Analisis diawali dengan uji coba *malware* yang dilakukan secara online menggunakan virustotal. Tujuan dari dilakukannya uji coba di virustotal yaitu untuk mengetahui hash dari *file* yang akan dianalisis.



Gambar 2. Informasi Gandcrab menggunakan virustotal

Terlihat bahwa hasil analisis *file* gandcrab terdeteksi sebagai *malware* dengan rasio 60 *antimalware* dari 70 *antimalware*. Salah satunya menyebutkan bahwa *file sample malware* tersebut termasuk kedalam jenis *ransomware* yaitu Ransom.GandCrab. Adapun nilai *checksum* MD5 "95557a29de4b70a25ce62a03472be684", nilai *checksum* SHA-1 "5baabf2869278e60d4c4f236b832bffddd6cf969", dan nilai *checksum* SHA-256 "49b769536224f160b6087dc866edf6445531c6136ab76b9d5079ce622b043200". *File sample malware* berukuran 424 kb dengan tipe *file EXE*. *Exe* merupakan singkatan dari *executable* yang berisi program atau *file* tertentu yang dapat dieksekusi atau dijalankan sebagai program di komputer.

MD5	95557a29de4b70a25ce62a03472be684
SHA-1	5baabf2869278e60d4c4f236b832bffddd6cf969
SHA-256	49b769536224f160b6087dc866edf6445531c6136ab76b9d5079ce622b043200
Vhash	045056551d155d6035z61zb16fz12z26fz
Authentihash	b72476902ca99321c3a91c9fd2d1b86e0e7c8261eefe44709f2975f29f5d0d73
Imphash	754f05425de4ad06169098be9bbe56cb
Rich PE header hash	d78f68593665aab949adae802adb0792
SSDEEP	6144:/UGV83D35bJrqV2L/E0tA+j16kUef5Nj1mB9WjEw0tzMV:qvmVe9h1qEtkBzw0tQ
TLSH	T13694E1192922D0D8FE75233173AAC47102707EF35DC62A7710CE7A4A79F6A4CA71F968
File type	Win32 EXE
Magic	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
TrID	Microsoft Visual C++ compiled executable (generic) (42.7%)
TrID	Win32 Dynamic Link Library (generic) (17%)
TrID	Win16 NE executable (generic) (13%)
TrID	Win32 Executable (generic) (11.6%)
TrID	OS/2 Executable (generic) (5.2%)
File size	424.00 KB (434176 bytes)
PEiD packer	Microsoft Visual C++

Gambar 3. checksum file malware

3.2. *Analysis With PeStudio*

Analisis ini dilakukan untuk untuk mengetahui keaslian atau data *integrity* dari *sample malware* dengan membandingkan nilai *checksum* yang diperoleh menggunakan virustotal dan nilai *checksum* yang didapatkan dari PeStudio.

property	value
md5	95557A29DE4B70A25CE62A03472BE684
sha1	5BAABF2869278E60D4C4F236B832BFFDDD6CF969
sha256	49B769536224F160B6087DC866EDF6445531C6136AB76B9D5079CE622B043200

Gambar 4. Informasi Gandcrab dengan PeStudio

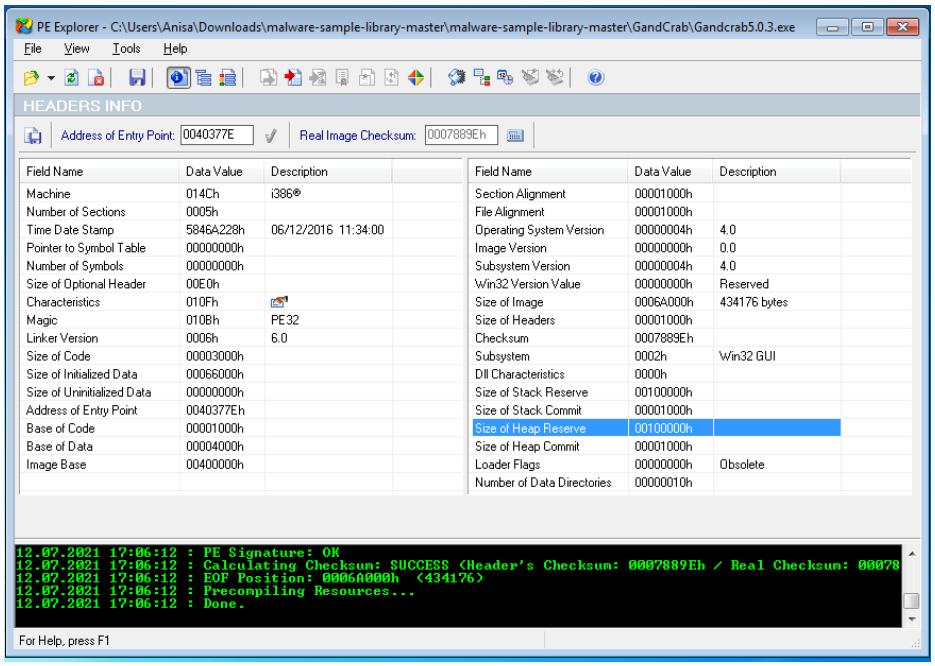
Hasil informasi nilai *checksum* MD5, SHA1, dan SHA26 pada *sample malware gandcrab* yang didapatkan dari PeStudio. Setelah nilai *checksum* didapatkan, selanjutnya ialah proses membandingkan dari kedua nilai *checksum.*

Tabel 2. Perbandingan nilai checksum gandcrab antara virustotal dan PeStudio

	Nilai *Checksum*		**Definisi**
	Virustotal	**PeStudio**	
MD5	95557a29de4b7 0a25ce62a03472be684	95557a29de4b7 0a25ce62a03472be684	Identik
SHA1	5baabf286927 8e60d4c4f236b 832bffddd6cf969	5baabf286927 8e60d4c4f236b 832bffddd6cf969	Identik
SHA26	49b769536224 f160b6087dc866edf6445531c6 136ab76b9d5079ce622b043200	49b769536224 f160b6087dc866edf6445531c6 136ab76b9d5079ce622b043200	Identik

3.3. *Analysis With PEexplorer*

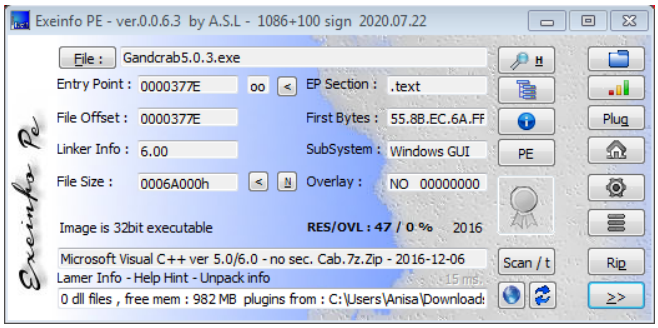
Untuk mengetahui format dari *sample malware* yang akan dianalisis, peneliti menggunakan *tools PE explorer.*



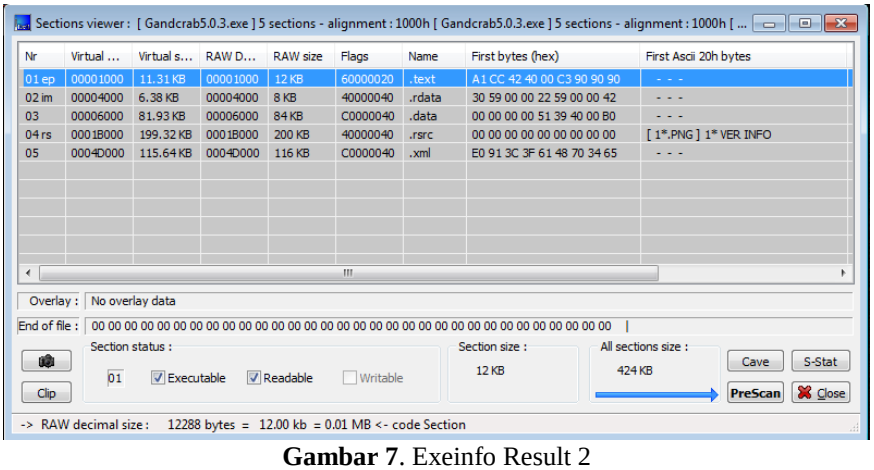
Gambar 5. PE Explorer result

3.4. *Analysis With Exeinfo*

Hasil dari PE explorer dapat diperjelas dengan melanjutkan analisis menggunakan *tools exe*info PE. Setelah dilakukan analisis, terlihat bahwa *malware* tersebut dikompilasi (compiled) menggunakan Microsoft visual C++ 5.0/6.0 dengan tidak ada data overlay dan packer protection Cab.7z.Zip.



Gambar 6. Exeinfo Result 1

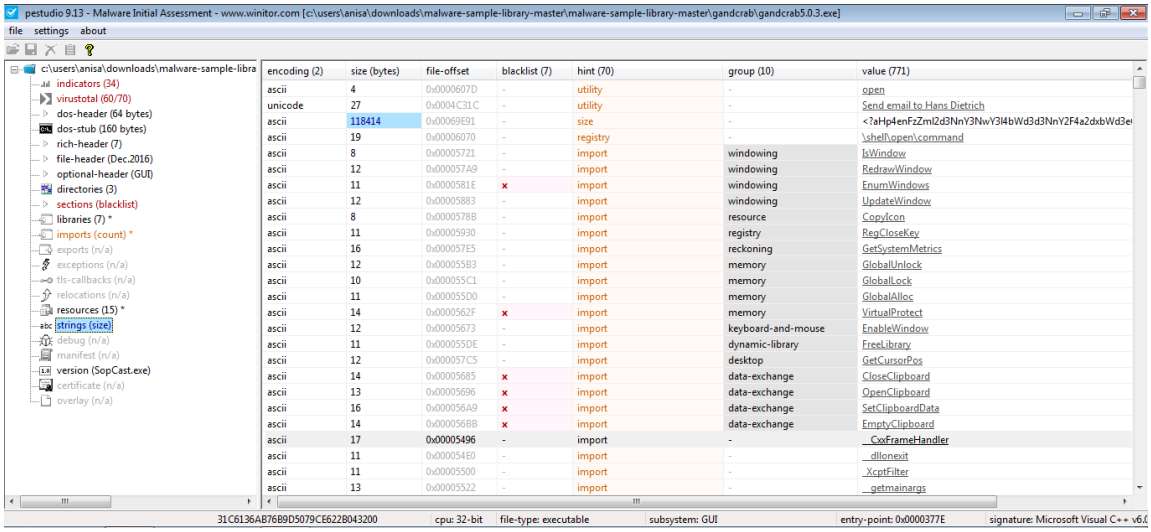


Gambar 7. Exeinfo Result 2

Diketahui bahwa program ini *executable* dan *readable* yang berarti *file* tersebut dapat dieksekusi dan dibaca namun tidak bisa ditulis.

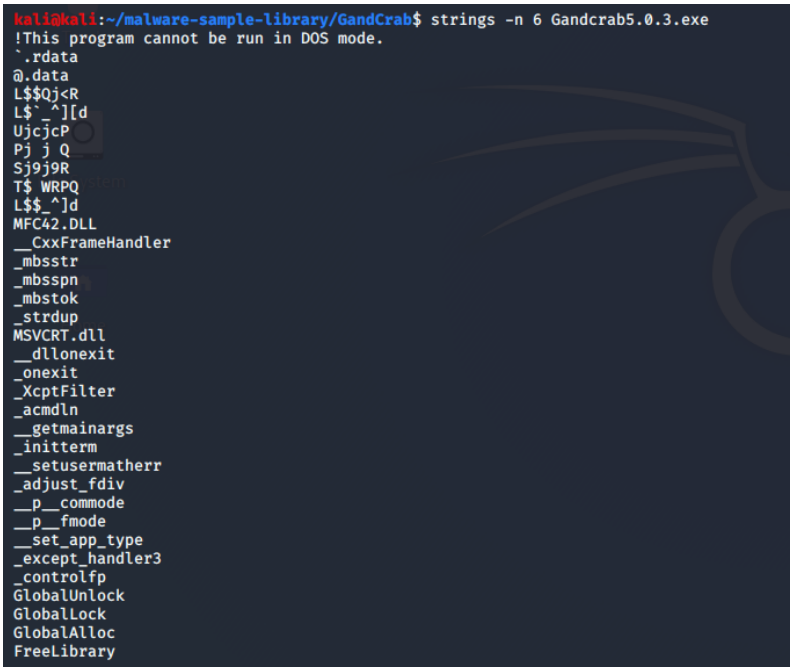
3.5. *Analysis String Indicators*

Untuk mengetahui cara kerja *malware*, langkah awal yaitu mencari *strings* dari *malware* tersebut. *Strings* merupakan uraian karakter pada sebuah program. *Strings* biasanya berupa karakter kode ASCII atau Unicode.



Gambar 8. String Indicators

Pada kali linux, untuk pencarian *strings*, masukkan perintah "*strings* -n 6 gandcrab5.0.3.*exe*" pada *virtual machine environment* kali linux. Hasil pencarian dapat dilihat pada Gambar 9.

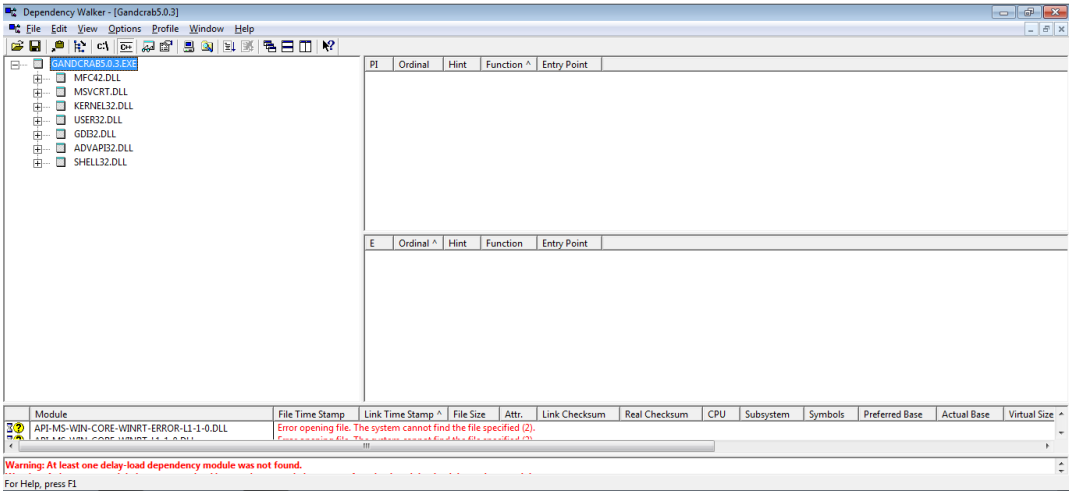


Gambar 9. Strings Malware Gandcrab Ransomware

Setelah mengetahui *strings* dari *malware* tersebut, akan ditemukan fungsi DLL yang disusupi oleh *malware*. DLL merupakan singkatan dari *Dynamic-Link Library*. Yaitu *library* yang menyimpan data-data windows yang berisi arahan yang dibutuhkan oleh suatu aplikasi. Biasanya DLL berisi *resource* seperti *image, icon, files, functions, variables* dsb.

3.6. *Analysis With Dependency Walker*

Pada dependency walker, akan terlihat lebih detail fungsi *strings* dari *malware* yang sedang dianalisis.

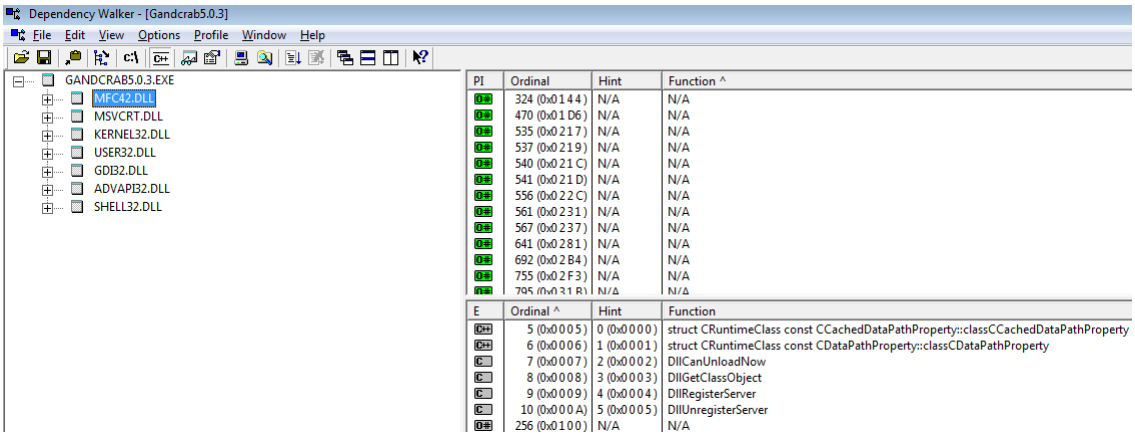


Gambar 10. Tampilan awal dependency walker

Dependency walker menampilkan secara lengkap *strings* dari *sample malware* yang akan dianalisis dan dimana letak fungsi *strings* tersebut pada *windows system*. Setelah *sample malware* di *import* kedalam *tools* dependency walker, maka akan diketahui fungsi DLL yang diserang oleh *malware* yang di analisis.

a. MFC42.DLL

MFC42.DLL (Microsoft Foundation Class) merupakan salah satu *file* DLL yang berisi sekumpulan fungsi driver dan memastikan program windows beroperasi dengan benar. Jika MFC42.DLL disusupi oleh *malware*, kemungkinan data di hard disk pengguna dan registri sistem akan mengalami akumulasi entri yang tidak valid dan mempengaruhi kinerja komputer pengguna. Pada Gambar 11 dapat dilihat *sample malware* gandcrab mengakses *file* MFC42.DLL.



Gambar 11. MFC24.DLL

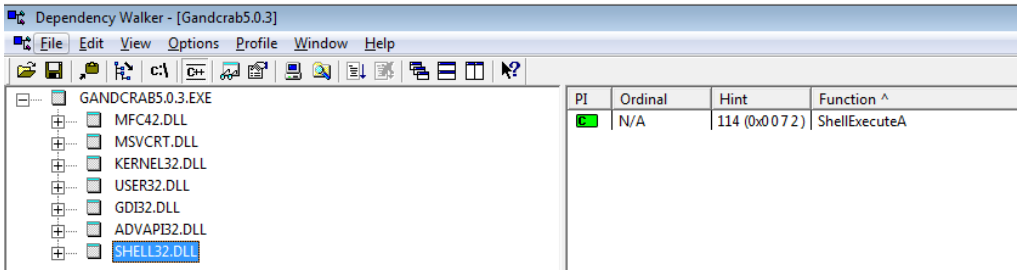
Pada *file* MFC42.DLL memiliki *strings* yang terdiri dari:

- **DllCanUnloadNow** yang berfungsi untuk menentukan apakah DLL pada fungsi ini sedang digunakan atau tidak. Jika tidak, maka akan dilanjutkan ke perintah selanjutnya yang dapat masuk kedalam DLL memori untuk dibongkar.
- **DllGetClassObject** yang berfungsi untuk mengambil ojek kelas dari pengendali objek DLL atau aplikasi objek
- **DllRegisterServer** berfungsi untuk menginstruksikan server pada proses untuk membuat entri registri untuk semua kelas yang ada pada server tersebut.
- **DllUnregisterServer** berfungsi untuk menginstruksikan server pada proses untuk menghapus entri tertentu yang dibuat pada DllRegisterServer.

b. SHELL32.DLL

Library yang berisi fungsi Windows shell API yang digunakan untuk membuka suatu halaman web atau *file*. Jika *file* SHELL32.DLL disusupi oleh *malware*, ini memungkinkan *malware* akan mengakses *file* tertentu

terkait dengan halaman web atau *file*. Pada gambar 12 dapat dilihat *sample malware* gandcrab mengkases *file* SHELL32.DLL.

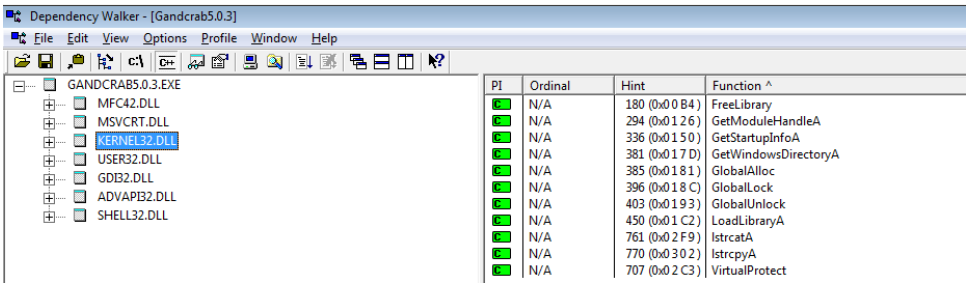


Gambar 12. SHELL32.DLL

Pada *file* SHELL32.DLL hanya memiliki *strings* **Shell*Exe*cuteA** yang berfungsi untuk melakukan operasi pada *file* tertentu.

c. KERNEL32.DLL

Library yang berguna untuk menangani manajemen memori, operasi input/output, interupsi, sinkronasi, dan pembuatan proses. Ketika komputer dioperasikan, KERNEL32.DLL dimuat kedalam ruang memori yang dilindungi sehingga tidak dapat diambil alih oleh program lain. Pada gambar 13 dapat dilihat *sample malware* gandcrab mengakses *file* KERNEL32.DLL.



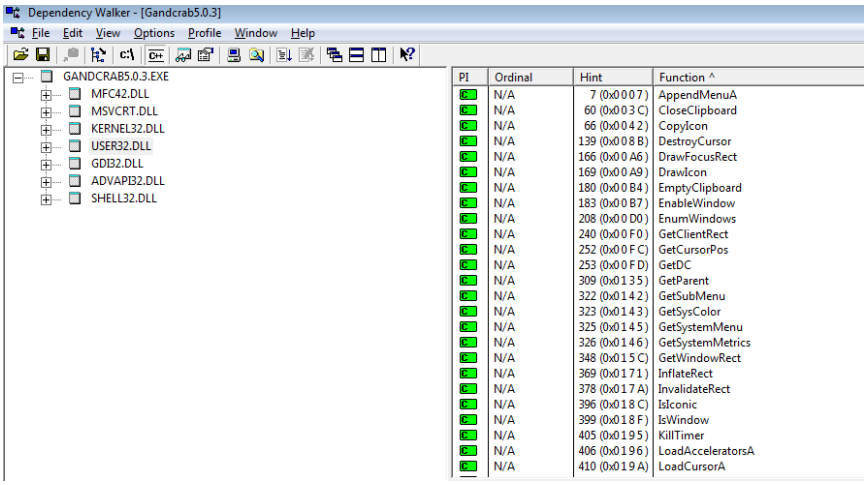
Gambar 13. KERNEL32.DLL

Pada *file* KERNEL32.DLL memiliki *strings* yang terdiri dari:

- **FreeLibrary** yang berfungsi untuk mengosongkan modul DLL yang sedang berjalan jug mengurangi jumlah referensinya. Sehingga, apabila jumlah referensi menjadi nol, maka alamat proses yang berjalan saat itu menjadi tidak valid.
- **GetModuleHandleA** yang berfungsi untuk mengambil alih control fungsi yang sedang berjalan untuk fungsi yang ditentukan.
- **GetStartupInfoA** yang berfungsi untuk mengambil info yang yang ditentukan saat proses pemanggilan dibuat.
- **GetWindowsDirectoryA** yang berfungsi untuk mengambil jalur direktori pada windows.
- **GlobalAlloc** yang berfungsi untuk mengalokasikan jumlah *byte* yang ditentukan oleh *heap*.
- **GlobalUnlock** yang berfungsi untuk mengurangi jumlah kunci yang terkait dengan objek memori.
- **LoadLibraryA** yang berfungsi untuk memuat fungsi yang ditentukan ke dalam ruang alamat dari proses pemanggilan.
- **VirtualProtect** yang berfungsi untuk mengubah proteksi pada halaman diruang alamat virtual dari proses panggilan.

d. USER32.DLL

Mengatur GUI (graphical user interface) yaitu *file* yang berfungsi untuk menyesuaikan tampilan windows, desktop, dan menu dari komputer penggguna. Jika *file* USER32.DLL diserang oleh *malware*, ini memungkinkan *malware* akan mengubah tampilan interface pada komputer pengguna. Pada gambar 14 dapat dilihat *sample malware* gandcrab mengakses *file* USER32.DLL.



Gambar 14. USER32.DLL

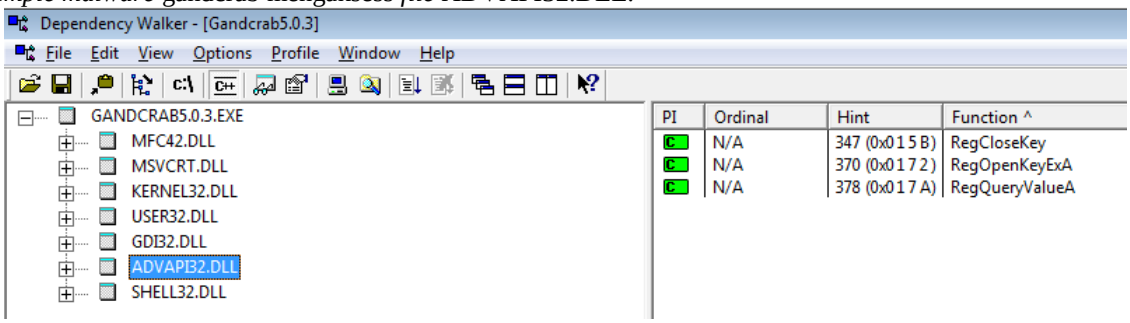
Pada *file* USER32.DLL memiliki *strings* yang terdiri dari:

- **AppendMenuA** yang berfungsi untuk menambahkan *item* baru pada menu tertentu untuk mengatur tampilan, konten maupun item menu.
- **CloseClipboard** yang berfungsi untuk menutup *clipboard.*
- **CopyIcon** yang berfungsi untuk menyalin *icon* tertentu dari fungsi lain dari fungsi saat ini
- **DestroyCursor** yang berfungsi untuk mengambil alih fungsi kursor dan menggunakan memori yang digunakan kursor saat itu.
- **DrawFocusRect** yang berfungsi untuk mengambil alih fungsi gambar.
- **DrawIcon** yang berfungsi untuk menggambar *icon* tertentu.
- **EmptyClipboard** yang berfungsi untuk mengosongkan *clipboard* dan mengambil alih fungsi.
- **EnableWindow** yang berfungsi untuk mengaktifkan dan menonaktifkan fungsi *keyboard* dan *mouse* dan kontrol tertentu.
- **EnumWindows** yang berfungsi untuk menghitung jumlah jendela yang sedang berjalan.
- **GetClientRect** yang berfungsi untuk mengambil kordinat area jendela yang sedang berjalan.
- **GetCursorPos** yang berfungsi untuk mengambil posisi cursor pada jendela layer yang sedang berjalan.
- **GetDC** yang berfungsi untuk mengambil alih fungsi DC *(device context)* pada jendela yang sedang berjalan.
- **GetParent** yang berfungsi untuk mengambil kendali pada jendela layer tertentu.
- **GetSubMenu** yang berfungsi untuk mengambil kendali menu tarik-turun atau submenu tertentu yang sedang aktif.
- **GetSysColor** yang berfungsi untuk mengubah warna tampilan pada layer utama.
- **GetSystemMenu** yang berfungsi untuk memodifikasi menu layar utama.
- **GetSystemMetrics** yang berfungsi untuk mengambil alih sistem metrik tertentu.
- **GetWindowRect** yang berfungsi untuk mengatur dimensi persegi pada jendela layar tertentu.
- **InflateRect** yang berfungsi untuk mengubah dimensi persegi pada layar.
- **InvalidateRect** yang berfungsi untuk menambah persegi panjang pada wilayah pembaruan jendela tertentu.
- **IsIconic** yang berfungsi untuk meminimalkan *icon* pada jendela tertentu.
- **IsWindow** yang berfungsi untuk mengidentifikasi jendela saat itu.
- **KillTimer** yang berfungsi untuk merusak waktu yang ditentukan.
- **LoadAcceleratorsA** yang berfungsi untuk memuat tabel akselerator tertentu.
- **LoadCursorA** yang berfungsi untuk memuat sumber *cursor* yang ditentukan dari *file* yang dapat dieksekusi *(.EXE).*
- **LoadIconA** yang berfungsi untuk memuat sumber *icon* yang ditentukan dari *file* yang dapat dieksekusi *(.EXE).*
- **LoadMenuA** yang berfungsi untuk memuat sumber menu yang ditentukan dari *file* yang dapat dieksekusi *(.EXE).*

- **OpenClipboard** yang berfungsi untuk membuka *clipboard* dan mencegah aplikasi lain memodifikasi konten *clipboard*.
- **PtInRect** yang berfungsi untuk menentukan titik yang diperlukan berada pada titik yang ditentukan.
- **RedrawWindow** yang berfungsi untuk memperbarui area pada layar utama *user*.
- **ReleaseDC** yang berfungsi untuk membebaskan fungsi DC *(device context)* untuk digunakan aplikasi lain.
- **SendMessageA** yang berfungsi untuk mengirim pesan khusus pada layar utama.
- **SetClipboardData** yang berfungsi untuk menempatkan data pada *clipboard* dalam *format clipboard* tertentu.
- **SetCursor** yang berfungsi untuk mengatur kursor.
- **SetTimer** yang berfungsi untuk mengatur waktu dengan batas yang ditentukan.
- **SetWindowLongA** yang berfungsi untuk mengubah atribut tertentu pada layar.
- **TranslateAcceleratorA** yang berfungsi untuk memroses tombol akselerator untuk perintah menu.
- **UpdateWindow** yang berfungsi untukm emperbarui layar utama dengan mengirim pesan tertentu.

e. ADVAPI32.DLL

Mengatur panggilan keamanan dan fungsi manipulasi *windows* registri. Pada gambar 15 dapat dilihat *sample malware* gandcrab mengaksess *file* ADVAPI32.DLL.



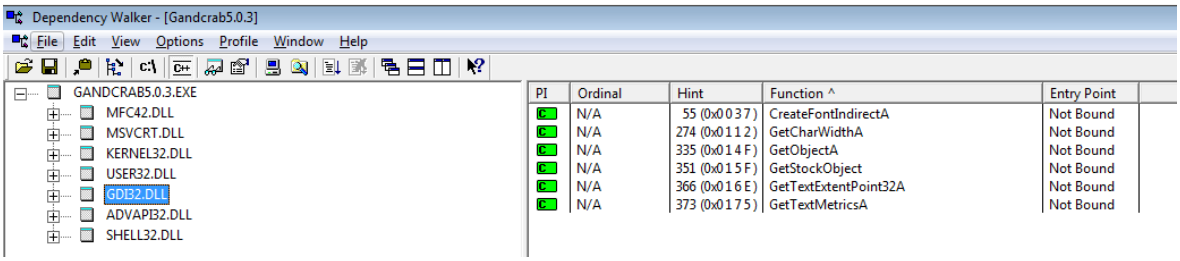
Gambar 15. ADVAPI32.DLL

Pada *file* ADVAPI32.DLL memiliki *strings* yang terdiri dari:

- **RegCloseKey** yang berfungsi untuk menutup akses pada registri tertentu.
- **RegOpenKeyExA** yang berfungsi untuk membuka registri tertentu.
- **RegQueryValueA** yang berfungsi untuk mengambil data terkait nilai *default* dari registri yang ditentukan. Data hanya berupa string yang di akhiri dengan *null*.

f. GDI32.DLL

Mengatur GDI *(graphics device interface)* membantu *windows* untuk melakukan gambar tingkat rendah dua dimensi seperti: teks, manajemen font, dan fungsi serupa pada paint. Jika *file* GDI32.DLL disusupi *malware*, ini memungkin *malware* dapat mengubah font atau membuat teks tertentu. Pada gambar 16 dapat dilihat bahwa *sample malware* gandcrab mengakses *file* GDI32.DLL.



Gambar 16. GDI32.DLL

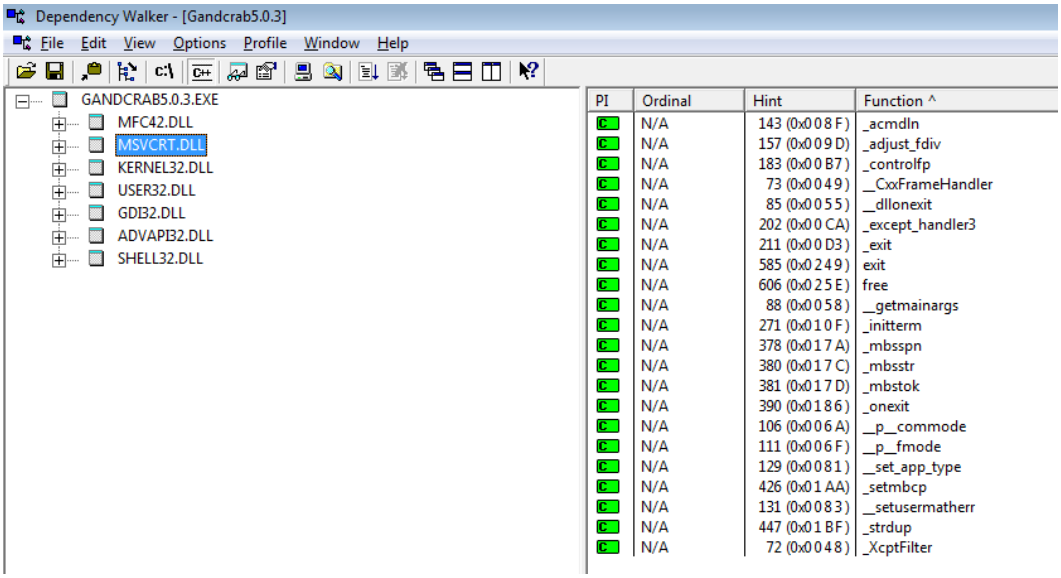
Pada *file* GDI32.DLL memiliki *strings* yang terdiri dari:

- **CreateFontIndirectA** yang berfungsi untuk membuat *font* yang memiliki karakteristik tertentu.
- **GetCharWidthA** yang berfungsi untuk mengatur lebar koordinat karakter menyesuaikan dengan font saat ini.
- **GetObjectA** yang berfungsi untuk mengambil informasi objek grafis yang ditentukan.

- **GetStockObject** yang berfungsi untuk mengambul alih pengaturan *font.*
- **GetTextExtentPoint32A** yang berfungsi untuk menghitung lebar dan tinggi *string* teks yang ditentukan.
- **GetTextMetricsA** yang berfungsi untuk mengisi *buffer* yang ditentukan dengan metrik untuk *font* yang saat ini dipilih.

g. MSVCRT.dll

Berisi sekumpulan fungsi untuk program yang berjalan menggunakan Microsoft visual c++ atau MSVC. Pada gambar 17 dapat dilihat bahwa *sample malware* gandcrab mengakses *file* MSVCRT.DLL.



Gambar 17. MSVCRT.DLL

Pada *file* MSVCRT.DLL memiliki *strings* yang terdiri dari:

- **_acmdln** berfungsi untuk menyimpan data dalam bentuk *string* karakter.
- **__CxxFrameHandler** merupakan fungsi *internal* CRT yang digunakan untuk menangani pengecualian yang terstruktur.
- **__dllonexit** berfungsi untuk menjadwalkan waktu keluar.
- **_except_handler3** merupakan fungsi *internal* CRT yang diigunakan oleh kerangka kerja untuk menemukan penangan pengecualian yang sesuai untuk memproses pengecualian saat ini
- __getmainargs berfungsi untuk memanggil penguraian baris perintah dan menyalin argumen ke main() kembali.
- **_initterm** merupakan metode *internal* yang menjalankan tabel *pointer* fungsi dan menginisialisasinya.
- **_mbsspn** berfungsi untuk mengembalikan indeks karakter pertama, dalam *string*, yang bukan milik.
- **_mbsstr** berfungsi untuk mengembalikan pointer ke kemunculan pertama dari *string* pencarian dalam sebuah *string*.
- **_mbstok** berfungsi untuk menemukan token berikutnya dalam sebuah *string*, dengan menggunakan lokal saat ini atau lokal tertentu yang diteruskan.
- **__p__commode** berfungsi untuk menunjuk ke variabel global _commode, yang menentukan mode komit *file default* untuk operasi I/O *file.*
- **__p__fmode** berfungsi untuk menunjuk ke variabel global _fmode, yang menentukan mode terjemahan *file default* untuk operasi I/O *file.*
- **__set_app_type** berfungsi untuk mengatur tipe dari aplikasi saat ini.
- **_setmbcp** berfungsi untuk menetapkan halaman kode *multibyte* baru.
- **Strdup** merupakan fungsi yang digunakan untuk peringatan saat mengkompilasi.

4. KESIMPULAN DAN SARAN

Proses analisis ini diakhiri dengan tahap *reporting* yaitu memaparkan hasil temuan dari analisis yang telah selesai dilakukan. Pada virustotal, didapatkan bahwa *file sample malware* terdeteksi sebagai *malware* dengan

rasio 60 dari 70 anti*malware*. Dan didapatkan nilai *checksum* MD5 "95557a29de4b70a25ce62a03472be684". *File sample malware* berukuran 424kb dan memilik tipe *file EXE*. Pada PeStudio, ditemukan bahwa nilai *checksum* dari *sample malware* yang dianalisis sama dengan nilai *checksum* yang ditemukan pada saat proses analisis pada *tools* virustotal. Analisis dilanjutkan pada proses mencari tau format dari *sample malware* yang dianalisis. Dan didapatkan bahwa format *sample malware* tersebut ialah PE (*portable executable)* yang berarti bahwa program tersebut dapat dieksekusi pada OS windows saja. Selanjutnya yaitu analisis pada *tools Exeinfo.* Analisis ini dilakukan untuk mengetahui menggunakan apa *sample malware* yang dianalisis ini dikompilasi atau *compiled.* Dan didapati bahwa *sample malware* dikompilasi menggunakan Microsoft Visual C++. Setelah mengetahui *sample malware* ini dikompilasi menggunakan Microsoft Visual C++, maka analisis dilanjutkan dengan mencari fungsi *strings* dari *sample malware* ini. Analisis *strings indicator* menggunakan *tools* PeStudio dan menggunakan *virtual environment* Kali Linux. Setelah mengetahui *strings* dari *sample malware* yang dianalisis, selanjutnya analisis dilakukan lebih lanjut pada fungsi *strings* tersebut. Untuk mengetahui apa saja yang diakses oleh *sample malware* yang sedang dianalisis. Dari hasil penelitian yang telat dilakukan, penulis menyarankan beberapa hal untuk dapat digunakan pada penelitian berikutnya yaitu untuk penelitian selanjutnya dapat menggunakan *tools* yang lebih lengkap lagi untuk membedah *source code* dari *malware gandcrab ransomware*. Selain itu, hasil penelitian dapat dijadikan bahan untuk membuat sebuah *antimalware* atau sejenisnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada Fakultas ilmu Komputer Universitas Amikom Yogyakarta, terutama KaProdi Teknik Komputer dan Dosen Pembimbing dan juga pihak-pihak yang telah memberikan dukungan sehingga penelitian ini dapat diselesaikan.

REFERENSI

[1] I. Hariman and A. Syams, "Analisis Malware Dengan Teknik Static Analysis," 2015.
[2] Chebyshev Victor, "IT threat evolution Q3 2020 Mobile statistics | Securelist." 2020, [Online]. Available: https://securelist.com/it-threat-evolution-q3-2020-mobile-statistics/99461
[3] S. C. Hsiao and D. Y. Kao, "The static analysis of WannaCry ransomware," *Int. Conf. Adv. Commun. Technol. ICACT*, vol. 2018-Febru, pp. 153–158, 2018. https://doi.org/10.23919/ICACT.2018.8323680
[4] S. Usharani, P. Manju Bala, and M. Martina Jose Mary, "Dynamic Analysis on Crypto-ransomware by using Machine Learning: GandCrab Ransomware," *J. Phys. Conf. Ser.*, vol. 1717, p. 012024, 2021. https://doi.org/10.1088/1742-6596/1717/1/012024
[5] S. Talukder and Z. Talukder, "A Survey on Malware Detection and Analysis Tools," *Int. J. Netw. Secur. Its Appl.*, vol. 12, no. 2, pp. 37–57, 2020. https://doi.org/10.5121/ijnsa.2020.12203
[6] Malwarebytes, "GandCrab Ransomware - Removal and Prevention Guide | Malwarebytes." 2019, [Online]. Available: https://www.malwarebytes.com/gandcrab
[7] S. Gadhiya, K. Bhavsar, and P. D. Student, "Techniques for Malware Analysis," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 4, pp. 2277–128, 2013.
[8] D. Uppal, V. Mehra, and V. Verma, "Basic survey on Malware Analysis, Tools and Techniques," *Int. J. Comput. Sci. Appl.*, vol. 4, no. 1, pp. 103–112, 2014. https://doi.org/10.5121/ijcsa.2014.4110
[9] P. Nunes, I. Medeiros, J. C. Fonseca, N. Neves, M. Correia, and M. Vieira, "Benchmarking Static Analysis Tools for Web Security," *IEEE Trans. Reliab.*, vol. 67, no. 3, pp. 1159–1175, 2018. https://doi.org/10.1109/TR.2018.2839339
[10] A. Braga, R. Dahab, N. Antunes, N. Laranjeiro, and M. Vieira, "Understanding How to Use Static Analysis Tools for Detecting Cryptography Misuse in Software," *IEEE Trans. Reliab.*, vol. 68, no. 4, pp. 1384–1403, 2019. https://doi.org/10.1109/TR.2019.2937214

BIOGRAFI PENULIS

Anisa Oktaviani - Lahir di Padang pada tanggal 10 Oktober 1999. Menyelesaikan Studi pada Universitas Amikom Yogyakarta Program Studi Teknik Komputer Angkatan 2017. Tertarik pada penelitian bidang cybersecurity dan malware.

Melwin Syafrizal - Mahasiswa Program Ph.D di Universiti Teknikal Malaysia (UTeM) Melaka. Beliau juga merupakan dosen di Departemen Teknik Komputer, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Indonesia. Bidang penelitiannya adalah Jaringan Komputer, Analisis Keamanan Jaringan, Keamanan Siber, dan Pertahanan Siber. Email: Melwin@amikom.ac.id