Analisis *Ransomware* secara Statis dan Dinamis untuk Pemetaan Evolusi *Ransomware*

Disusun Oleh:

N a m a : Cholid Fitra Fahriza

NIM : 15523095

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

Analisis *Ransomware* secara Statis dan Dinamis untuk Pemetaan Evolusi *Ransomware*

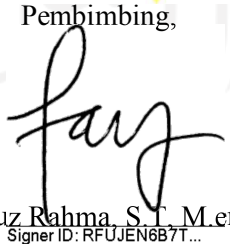
TUGAS AKHIR

Disusun Oleh:

N a m a : Cholid Fitra Fahriza
NIM : 15523095

Yogyakarta, 17 Januari 2022

Pembimbing,



(Fayruz Rahma, S.T., M.eng)

Signer ID: RFUJEN6B7T...

HALAMAN PENGESAHAN DOSEN PENGUJI

Analisis *Ransomware* secara Statis dan Dinamis untuk Pemetaan Evolusi *Ransomware*

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 1 Nopember 2017

Tim Penguji

Fayruz Rahma, S.T, M.eng

Anggota 1

Signer ID: RFUJEN6B7T...

Ahmad Luthfi, S.Kom., M.Kom.

Anggota 2

Moh. Idris, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Cholid Fitra Fahriza

NIM : 15523095

Tugas akhir dengan judul:

Analisis *Ransomware* secara Statis dan Dinamis untuk Pemetaan Evolusi *Ransomware*

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 17 Januari 2022

(Cholid Fitra Fahriza)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan karunia yang tiada hentinya kepada saya, sehingga saya masih dapat merasakan nikmat Iman, Islam, Rezeki serta Kesehatan setiap harinya tanpa kurang suatu apapun.

Shalawat serta salam senantiasa tercurahkan untuk junjungan kita Nabi Muhammad Shalallahu 'Alaihi Wasalam yang telah membawa kita dari zaman yang gelap menuju kepada zaman yang terang benderang. Semoga kita termasuk orang-orang yang mendapat syafaat dari Nabi Muhammad Shalallahu 'Alaihi Wasalam kelak pada hari akhir. Tugas akhir ini saya persembahkan kepada:

1. Kedua orang tua saya yang selalu memberikan dukungan serta doa untuk saya
2. Kepada abang-abang dan adik saya yang selalu ada untuk memberikan semangat,nasihat serta dukungan untuk saya.
3. Kepada sahabat dan teman-teman saya yang ada pada saat suka maupun duka selama masa perkuliahan
4. Ibu Fayruz Rahma, S.T., M.Eng., Selaku pembimbing tugas akhir saya, yang selalu sabar dan menasehati sehingga skripsi ini menjadi lebih baik
5. Bapak Fietyata Yudha S.Kom., M.Kom., Selaku pembimbing tugas akhir saya, yang selalu sabar dan menasehati sehingga skripsis ini menjadi lebih baik

HALAMAN MOTO

Selalu ada solusi untuk setiap masalah

***

Selalu bersyukur untuk semua yang terjadi dalam hidup

KATA PENGANTAR

Assalamu'alaikum Warohmatullahi Wabarokatuh

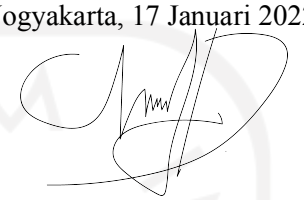
Alhamdulillahirobilalamin, puji syukur kami panjatkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Analisi *Ransomware* Secara Statis dan Dinamis Untuk Memetakan Evolusi *Ransomware*". Laporan Tugas Akhir ini dibuat sebagai syarat untuk memperoleh gelar sarjana Teknik Informatika Universitas Islam Indonesia. Penulis menyadari bahwa dalam pelaksanaan Tugas Akhir dan penyusunan laporan ini tidak dapat lepas dari bimbingan, dukungan dan bantuan dari berbagai pihak. Oleh karena itu perkenankanlah penulis untuk mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Allah SWT karena atas karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik dan semoga Tugas Akhir ini dapat memberikan manfaat dikemudian hari.
2. Orang Tua dan keluarga penulis atas segala doa dan dukungan selama penulis melaksanakan Tugas Akhir
3. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia
4. Bapak R. Teduh Dirgahayu, S.T., M.Sc., Ph.D., selaku kepala jurusan teknik indormatika fakultas teknologi industri Universitas Islam Indonesia
5. Bapak Galang Prihadi Mahardika, S.Kom., M.Kom. selaku dosen pembimbing akademik jurusan teknik informatika fakultas teknologi industri universitas islam Indonesia
6. Ibu Fayruz Rahma S.T., M.Eng., dan Bapak Fietyata Yudha S.Kom., M.Kom., selaku dosen pembimbing tugas akhir di jurusan teknik informatika teknologi industri universitas islam Indonesia
7. Sahabat-sahabat SMA dan Informatika UII terimakasih telah membantu, memberi nasihat dan semangat sehingga laporan ini berjalan dengan baik
8. Semua pihak yang telah banyak membantu dalam pelaksanaan tugas akhir yang tidak dapat peneliti sebutkan satu per satu

Penulis menyadari bahwa laporan Tugas Akhir ini masih belum sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun agar Tugas Akhir ini menjadi lebih baik. Akhir kata semoga laporan ini dapat bermanfaat dan atas segala bantuan yang telah diberikan semoga mendapat imbalan yang setimpal dari Allah SWT, Amin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 17 Januari 2022



(Cholid Fitra Fahriza)

SARI

Malware adalah program berbahaya yang dapat membuat komputer melakukan perintah diluar kendali pengguna sehingga perangkat keras tidak dapat berkerja sebagaimana mestinya. *Ransomware* adalah *malware* yang mengenkripsi *file* pengguna, sehingga *file* tidak dapat diakses sampai pengguna membayarkan jumlah tertentu agar data-data pengguna dapat kembali diakses. *Ransomware* tidak hanya menginfeksi komputer individu namun organisasi seperti rumah sakit dan perusahaan, dengan adanya penelitian ini diharapkan dapat memprediksi evolusi dari *ransomware* dan memahami cara kerjanya sehingga dapat mengantisipasi dan meminimalisir dampak dari *ransomware*.

Metode yang digunakan pada penelitian ini ada dua yaitu analisis dengan metode statis dan dinamis. Metode statis adalah metode yang dilakukan tanpa harus mengeksekusi *malware*, sedangkan metode dinamis adalah metode yang dilakukan dengan menjalankan *malware* di dalam lingkungan yang aman, sehingga tidak dapat mengkontaminasi perangkat sekitar. Metode statis menggunakan sistem operasi Windows 10 dan metode dinamis menggunakan sistem operasi Ubuntu 18.04.

Hasil dari analisis statis adalah data berupa struktur *file ransomware*, kapan pertama kali *ransomware* di-*compile,* library yang digunakan, serta *import* dan *string* yang digunakan tanpa harus menjalankan *ransomware* sehingga dapat mengetahui bagaimana *ransomware* berkerja. Hasil dari analisis dinamis berupa pohon proses, perilaku *ransomware*, dan *signature* dari *ransomware* yang dianalisis, lalu hasil akan digunakan untuk memetakan evolusi dari *ransomware*.

Kata kunci : *Ransomware*, Evolusi, enkripsi, *signature*, analisis, statis, dinamis

GLOSARIUM

Compile	: Proses untuk mengubah berkas kode program dengan berkas lain yang terkait menjadi berkas yang siap untuk dieksekusi oleh sistem operasi secara langsung.
Sandbox:	:Mekanisme keamanan untuk memisahkan program yang sedang berjalan.
Malware	:*Softaware* berbahaya yang dapat merusak berbahaya, panggilan umum untuk perangkat lunak berbahaya
Hash	:Nilai unik pada setiap *file*, yang membuat *file* dapat dikenali
Ransomware	: *Software* berbahaya yang mengambil *file*, menghapus *file* dan mengunci komputer.
Gandcrab	: Nama *ransomware* yang akan dianalisis
GoldenEye	: Nama *ransomware* yang akan dianalisis
Locky	: Nama *ransomware* yang akan dianalisis
Ryuk	: Nama *ransomware* yang akan dianalisis
Registry	: *Database* yang menyimpan pengaturan pada sistem windows,
Signature Cuckoo	: Penjelasan fungsi yang dijalankan oleh ransomware

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Malware adalah program berbahaya yang dapat membuat komputer berperilaku yang membahayakan pengguna komputer ada banyak jenis malware yang dapat membahayakan pengguna komputer seperti Trojan, adware, spyware dan *Ransomware*. Pada penelitian ini akan dilakukan analisis malware yang berjenis *ransomware* dimana *ransomware* adalah *malware* yang bekerja dengan menghadang pengguna untuk mengakses *file* pada komputer .

ransomware akan meminta bayaran agar *file* tersebut dapat diberikan kembali kepada pemilik. Jika tidak maka *file* tersebut akan otomatis terhapus. Serangan *ransomware* sangat sering terjadi. Menurut data yang didapat pada tahun 2018, serangan *ransomware* terjadi sebanyak 812.67 juta kali dan 92% dari jumlah tersebut dikirim melalui email. Dari data tersebut juga diketahui bahwa 7 dari 10 *malware* yang dikirimkan ialah *ransomware* (McMillan, 2018).

Kerugian global karena serangan *ransomware* sendiri mencapai $20 milliar pada tahun 2020 dan *ransomware* merupakan penyumbang utama dan pembuat kerugian terbanyak. Salah satu contohnya adalah *ransomware* "*Ryuk"* yang pada tahun 2018 membuat kerugian dari *ransomware* naik hingga 543% (PurpleSec, 2021). Dengan melihat data dan keadaan zaman yang semakin terfokus pada teknologi, *ransomware* bukanlah sesuatu yang dapat diabaikan. Maka dari itu, penelitian ini akan menganalisis *ransomware* yang datanya dapat digunakan untuk mendeteksi lebih awal ketika ada *ransomware* baru yang memiliki kesamaan proses/perilaku dengan *ransomware* sebelumnya.

Dalam proses pelaksanaan penelitian ini, hal pertama yang perlu dilakukan ialah membuat lingkungan yang aman dengan virtual machine lalu mencari sampel *ransomware* untuk dianalisis. Langkah selanjutnya dengan melakukan analisis statis dengan melakukan pengecekan langsung pada *file ransomware* tanpa menjalankannya *ransomware* untuk melihat struktur *file ransomware* seperti import dan library yang digunakan yang dapat menggambarkan perilaku *ransomware* saat menjalankan *ransomware*. Langkah terakhir yaitu dengan melakukan analisis dinamis sampel *ransomware* akan dijalankan di lingkungan yang terisolasi untuk melihat perilaku *ransomware* saat dijalankan.

Ransomware yang akan dianalisis diantaranya adalah *Gandcrab* (2018), *Locky* (2016), *GoldenEye* (2017), dan *Ryuk* (2018). *Ransomware* yang disebutkan dipilih melalui dampak

yang disebabkan oleh *ransomware* tersebut pada tahun 2016 hingga 2020 (PurpleSec, 2021). *Ransomware* akan dianalisis dengan menggunakan metode analisis statis dan dinamis, untuk melakukan kedua metode tersebut, dibutuhkan peralatan yang disebut tools. Tools yang digunakan pada penelitian ini ialah PeStudio dan Cuckoo sandbox, kedua tools tersebut dipilih karena kemudahan dalam pengoperasian saat analis.

PeStudio merupakan alat yang dapat digunakan untuk melihat struktur dan konten dari suatu *file*. Seperti melihat String, Libary, dan import yang digunakan. PeStuido digunakan untuk melakukan analisis statis. Sementara itu, Cuckoo Sandbox merupakan perangkat lunak untuk melakukan analisis dinamis secara otomatis, dengan memberikan laporan seperti proses yang dilakukan oleh malware dan memonitor koneksi internet pada saat proses infeksi. Kegunaan dari program cuckoo sandbox adalah untuk menganalisis *ransomware* yang dijalankan di dalam *virtual machine* yang sudah terisolasi.

Hasil data analisis dari penelitian ini nantinya dapat dijadikan untuk mendeteksi berbagai *ransomware* baru yang akan muncul di masa mendatang yang memiliki kesamaan dengan *ransomware* yang dianalisis. Dalam penelitian ini, diperlukan informasi penelitian yang lain untuk membantu hasil penelitian. Diharapkan dengan adanya penelitian tentang analisis *ransomware* ini dapat memberikan kontribusi berguna pada bidang keilmuan forensic terhadap analisis *ransomware* di kemudian hari.

1.2 Rumusan Masalah

1. Bagaimana cara menyiapkan tempat pengujian sebagai tempat mengeksekusi dan melihat perilaku *ransomware*?
2. Bagaimana cara melakukan Basic analisis statis dengan tepat?
3. Bagaimana cara melakukan Basic analisis dinamis dengan menggunakan Cuckoo Sandbox dengan tepat?
4. Bagaimana hasil pemetaan evolusi *ransomware* dan perbedaan dari kedua analisis yang dilakukan?

1.3 Batasan Masalah

Berdasarkan rumusan masalah di atas, peneliti membataskan masalah ini pada:

1. Perangkat yang akan digunakan sebagai tempat analisis merupakan *virtual machine* dengan sistem operasi *windows* yang dijalankan dengan *virtualbox*
2. Analisis statis yang dilakukan merupakan Basic analisis statis.

3. Analisis dinamis dilakukan dengan *cuckoo sandbox* secara otomatis dan menghasilkan data *behaviour* analisis dan *process tree*.
4. *Ransomware* yang dianalisis hanya *Gandcrab, Locky, GoldenEye, Ryuk.*

1.4 Tujuan Penelitian

1. Mengetahui cara membuat tempat analisis *ransomware* yang aman.
2. Mengetahui cara melakukan Basic analisis statis.
3. Mengetahui cara melakukan analisis dinamis dengan cuckoo sandbox.
4. Mendapatkan hasil analisis dari kedua metode dan melakukan pemetaan pada evolusi *ransomware*.

1.5 Manfaat Penelitian

Manfaat dari adanya penelitian ini antara lain sebagai berikut:

1. Melalui analisis statis akan mengetahui dan menggambarkan struktur *ransomware*, dan perilaku tersembunyi yang tidak dapat dilihat saat analisis dinamis.
2. Melalui analisis dinamis dapat mengetahui dan menampilkan bagaimana *ransomware* berkerja saat dijalankan pada sistem.
3. Dapat berkontribusi pada bidang forensik *ransomware* sebagai dokumentasi di masa mendatang.

1.6 Metode penelitian

Adapun metode penelitian yang digunakan untuk melakukan penelitian ini di antaranya adalah:

1. **Studi pustaka**, yang merupakan teknik pengumpulan data dengan cara membaca dan mempelajari tentang analisis *ransomware*, Cuckoo Sandbox, teknik forensik, *reverse engineering*, dan hal-hal lain yang berkaitan dengan penelitian ini. Sumber yang diambil dari metode studi pustaka berupa buku, *paper*, jurnal, makalah, dan referensi lainnya.
2. **Persiapan untuk melakukan analisis**, yaitu menyiapkan *ransomware* yang akan dianalisis, tempat untuk menganalisis *ransomware*, seperti VirtualBox dan Cuckoo Sandbox untuk melakukan analisis dinamis, serta aplikasi-aplikasi yang dibutuhkan untuk analisis statis. Aplikasi-aplikasi ini bertujuan untuk membuat tempat yang terisolasi untuk menganalisis *ransomware* sehingga tidak terjadi hal yang tidak diinginkan
3. **Melakukan proses analisis.** Proses analisis dinamis dilakukan dengan cara mengeksekusi *ransomware* sehingga perilaku *ransomware* dapat terlihat.

Sedangkan analisis statis dilakukan tanpa mengeksekusi *ransomware*, melainkan dengan melihat struktur *file* dari *ransomware* tersebut.

4. **Menganalisis hasil analisis statis dan dinamis.** Hal ini dilakukan untuk dapat melihat kelebihan dan kekurangan setiap analisis dan melihat analisis yang paling efektif.
5. **Melakukan pelaporan dan pemetaan** dari evolusi *ransomware* yang didapat selama proses analisis.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, identifikasi masalah, menentukan batasan masalah yang akan dibahas, menjabarkan tujuan dan manfaat dari penelitian ini, asumsi metodologi serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi berbagai teori yang digunakan sebagai landasan untuk menyelesaikan permasalahan yang diangkat pada penelitian ini, bahasan dalam bagian ini mengenai pembahasan teori dasar yang digunakan dalam penelitian, terkait *ransomware* serta aplikasi untuk melaksanakan penelitian.

BAB III METODOLOGI

Bab ini berisi tentang objek dan jenis penelitian, data dan sumber data, teknik mengumpulkan data, dan skema pengerjaan penelitian.

BAB IV IMPLEMENTASI DAN HASIL PENGUJIAN SISTEM

Bab ini berisi hasil dari implementasi, pengujian serta penjelasan sesuai perencanaan yang telah dibuat sebelumnya. Pengujian dilakukan untuk memastikan dan membuktikan bahwa hasil akhir yang didapat sesuai dengan ekspektasi, perkiraan dan fakta yang ada.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang menjelaskan hasil akhir dari penelitian, tercapai atau tidaknya tujuan penelitian, serta menjelaskan kelebihan dan kekurangan yang terdapat pada sistem yang telah dibuat. Sementara itu saran berisi tentang hal-hal yang dapat diperbaiki dan dikembangkan lagi ke depannya, terutama mengenai kekurangan yang masih terdapat pada sistem tersebut.

BAB II
TINJAUAN PUSTAKA

2.1 Landasan Teori

Sebelum melakukan penelitian, peneliti melakukan studi literatur yang berhubungan dengan penelitian ini, didapatkan enam paper dari penelitian sebelumnya didapatkan teori-teori dasar yang berkaitan dengan analisis statis dan analisis dinamis, pemetaan evolusi, serta *tool*s dan teknik untuk mendapatkan hasil yang baik. Dari banyak buku dan jurnal yang telah dibaca berikut beberapa teori yang didapatkan ditampilkan pada tabel (2.1)

Tabel 2.1 Penelitian Sebelumnya

No.	Judul	Penulis	Bulan dan tahun akses	Pencapaian
1	A Complete Dynamic *Ransomware* Anlysis	Navroop Kaur, Dr. Amit Kumar Bindal, 2016.	Maret, 2021	Menejalaskan apa saja yang perlu diperhatikan saat melakukan analisis *ransomware* dengan menggunakan metode analisis dinamis, dan bagaimana cara melakukan analisis dinamis dengan baik dan memberikan beberapa *tools* penjelasan tentang *tools* lain yang dapat digunakan selain Cuckoo Sandbox
2	Implementation of *Ransomware* Analysis using Static and Dynamic Analysis Method	Yudi Prayudi, Imam Riadi, 2015.	Maret, 2021	Menjelaskan bagaimana cara implementasi pada analisis *ransomware* dengan *tool-tool* yang sediakan, dan memberikan contoh langkah demi lankah dalam

				melakukan analisis statis dan dinamis.
3	An Emerging *Ransomware* Analysis Techniques and Tools: A Comparative Analysis	Arkajit Datta, Aju D, Kakelli Anil Kumar, 2021.	Maret, 2021	Menjelaskan metode yang dapat digunakan untuk menganalisis *ransomware* dan beberapa pendekatan lainnya untuk dapat mengerti *ransomware* lebih mendalam pada satu masalah, dan mejelaskan *tool-tool* yang dapat digunakan untuk melakukan setiap analisis.
4	Forensic Analysis of *Ransomware* Families Using Static and Dynamic Analysis	Kul Prasad Subedi, Daya Ram Budhatoki, Dipangkar Dasgupta, 2018.	Maret, 2021	Menjelaskan metode yang digunakan untuk menganalisis *ransomware* dengan tujuan untuk mengetahui keluarga *ransomware* menggunakan metode statis dan dinamis, dengan mencoba banyak *sample* untuk mengetahui apakah memiliki kesamaan antar *sample* satu dan sample lainnya serta apakah dari setiap generasi *ransomware* memiliki fitur yang berkembang atau tidak dari *import* dan *string* yang digunakan.
5	Understanding the Evolution of *Ransomware*:	Aaron Zimba, Mumbi	Maret, 2021	Menjelaskan evolusi *ransomware* dari bermacam sisi seperti metode enkripsi

	Paradigm Shifts in Attack Structures	Chishimba, 2019.		yang digunakan, metode yang mencegah untuk melakukan penyembuhan pada sistem, sasaran dalam melakukan penyerangan acak atau tidak, metode penyebaran, sasaran penyerangan pada saat infeksi, kerugian, serta sistem operasi yang diserang. Informasi-informasi tersebut dapat membantu memberikan petunjuk bagaimana menentukan evolusi dari *ransomware*.
6	PE*File* Analysis: A Static Approach To *Ransomware* Analysis	Subash Poudyal, Kishor Datta Gupta, Sajib Sen, 2019.	Maret, 2021	Menjelaskan tentang struktur PE *file*, misalnya PE *file* terdiri dari dua bagian yaitu *header* dan *section*. S*ection* memiliki format yang berbeda-beda tergantung dari *file* yang dianalis. Dijelaskan juga cara untuk melakukan analisis *ransomware* dengan metode analisis statis.

2.1.1 Malware

Malware adalah singkatan dari *"malicious software"* program berbahaya yang di kembangkan oleh cybercriminal/hacker yang bertujuan untuk mencuri data dan menghapus atau merusak komputer yang terinfeksi, malware merupakan sebutan umum untuk program berbahaya, malware dapat di kategorikan lagi menjadi bagian-bagian sesuai dengan perilaku dari malware.(Kurt Baker, 2021)

Berikut adalah beberapa jenis malware:

1. Virus
2. Worms
3. Trojan
4. Spyware
5. Adware
6. *Ransomware*

Namun pada penelitian ini peneliti akan berfokus pada malware berjenis *ransomware*.

2.1.2 *Ransomware*

Ransomware adalah salah satu bentuk dari perangkat lunak berbahaya yang mengenkripsi *file* korban. Komputer yang terinfeksi oleh *ransomware* membuat *file* di dalam komputer tidak dapat diakses, dan pelaku meminta uang kepada korban agar *file* tersebut dapat kembali diakses setelah pembayaran. Biasanya, ketika terkena *ransomware* halaman depan komputer akan menampilkan sebuah instruksi yang telah disediakan oleh penyerang agar korban mengerti cara membayar tebusan *file* tersebut.(Josh Fruhlinger, 2020)

Setiap kali perangkat keras terkena perangkat lunak berbahaya ini biasanya akan ada hitung mundur yang akan menghapus semua *file* yang terenkripsi jika tidak adanya pembayaran. Jika pembayaran terjadi penyerang akan mengirimkan kunci untuk mengenkripsi perangkat yang terkena *ransomware*, berikut adalah *ransomware* yang akan dianalisis:

a. Gandcrab

Gandcrab adalah perangkat lunak berbahaya yang dibuat untuk manfaatkan monetisasi enkripsi data dari organisasi, pertama kali muncul pada akhir Januari tahun 2018. Perangkat lunak berbahaya ini menyesuaikan catatan tebusan dengan kondisi target dan data yang terenkripsi karena itu, Gandcrab mempunyai permintaan tebusan

dari US$600 sampai dengan US$700,000, Gandcrab menjadi perangkat berbahaya terpopuler pada tahun 2018.

Gambar 2. 1 Tampilan *ransomware* gandcrab

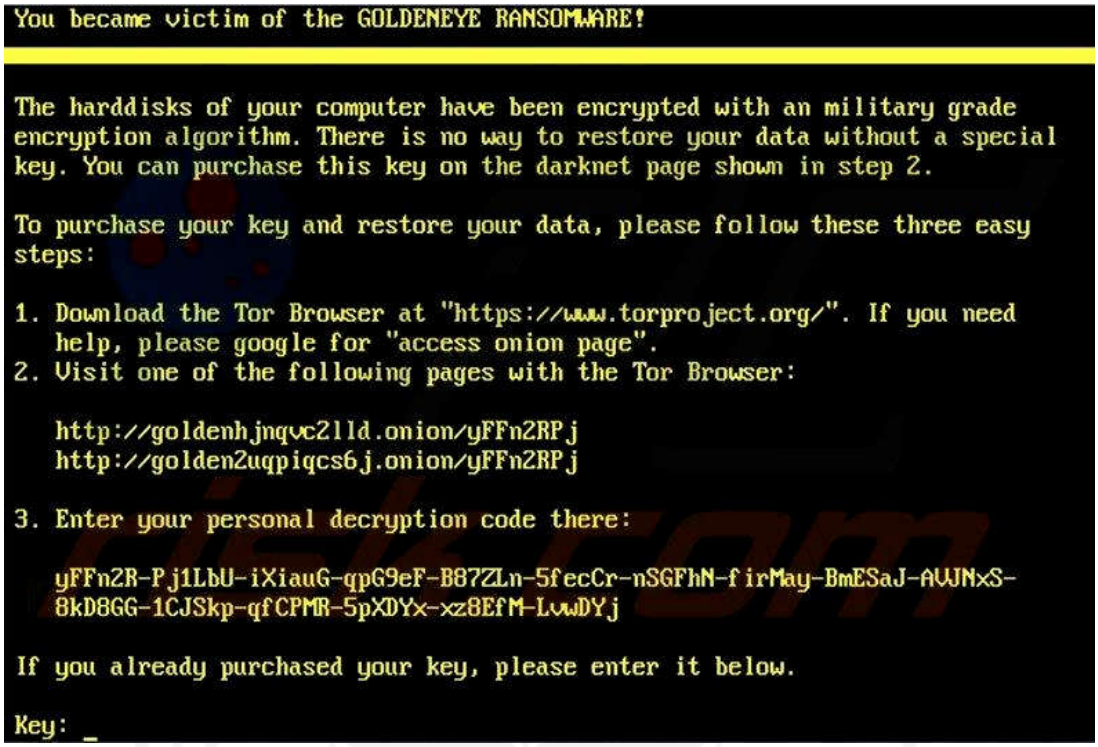
Perangkat lunak berbahaya ini memiliki 5 versi, versi 4 dan 5 diperkirakan menginfeksi sekitar 500,000 komputer seluruh dunia sejak Juli 2018. Menariknya Gandcrab mengadopsi pembayaran via DASH dan BITCOIN, DASH adalah percabangan dari protokol bitcoin yang dapat menghasilkan transaksi lebih cepat dan tidak dapat dilacak.(MalwareBytes, 2020a)

Fitur yang lebih menariknya dari perangkat berbahaya ini adalah ketika tereksekusi akan melakukan pengintaian kepada perangkat keras korban sebelum mengenkripsi *file* utntuk mengecek apakah korban memakai papan ketik Rusia sehingga dapat menghindari korban yang berasal dari Rusia.

Setelah pengecekan papan ketik, Gandcrab akan menutup semua aplikasi berjalan termasuk gim, lalu memulai proses enkripsi semua *file* sehingga tidak ada *file* yang terlewatkan dan *file* yang sangat penting terjamin terenkripsi.

b. GoldenEye

Pada bulan Maret 2016 didapatkan observasi yang menarik dari evolusi l*ow-level ransomware*, yaitu Petya yang digabungkan dengan *ransomware* lain yaitu Mischa dan lahirlah *ransomware* baru bernama GoldenEye. GoldenEye disebarkan melalui email. Perangkat lunak berbahaya ini dapat memblok akses penuh suatu komputer.(MalwareBytes, 2020b)

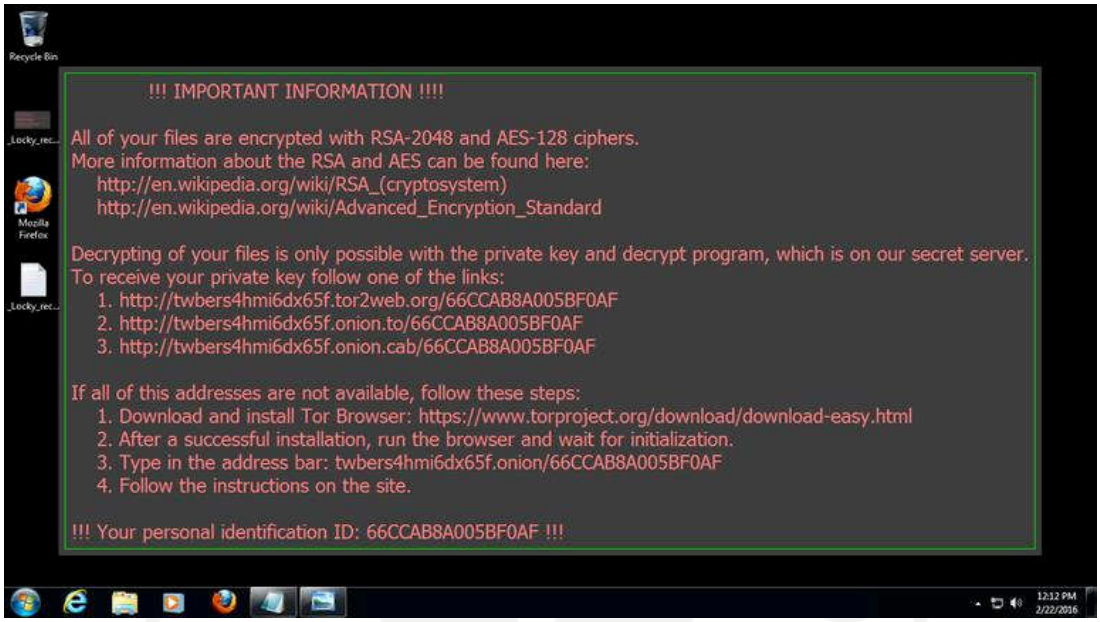


Gambar 2. 2 Tampilan *ransomware* Goldeneye

Perangkat lunak berbahaya bekerja dengan cara mengenkripsi sistem *drive* dan jika mendapatkan *admin permission*, GoldenEye akan mengenkripsi sistem *sektor boot* yang hasilnya akan menghadang pengguna untuk memakai komputer hingga pengguna mendapatkan kunci yang dapat mendekripsi ketika memasuki sistem.

c. Locky

Locky dirilis pada tahun 2016, ketika ahli keamanan komputer menemukan bahwa perangkat berbahaya ini dikirimkan melalui email dan meminta pembayaran dengan faktur yang terdapat di dokumen *word* yang dapat menjalankan program berbahaya jika dibuka. Ketika dibuka *file word* akan menampilkan pesan eror mengatakan bawah data *encoding* salah.



Gambar 2. 3 Tampilan *ransomware* locky

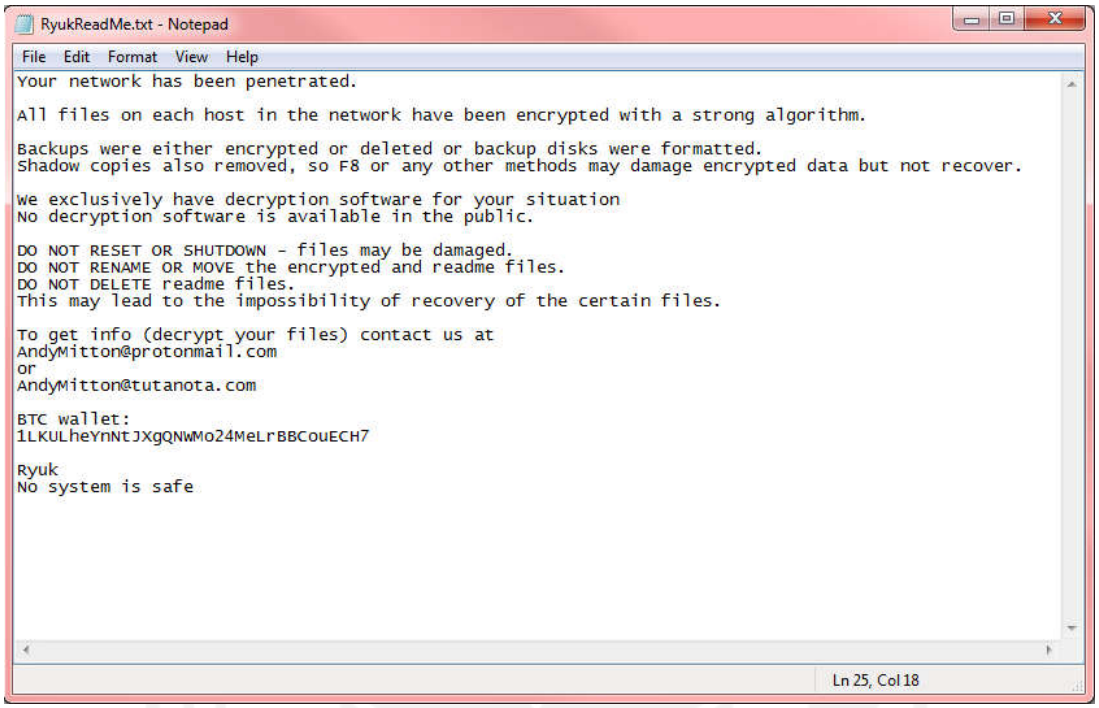
Dengan sedikit bantuan *social engineering* perangkat berbahaya ini dapat disebarkan dengan mudah dengan mengirimkan *file* bertipe .doc, .xls ataupun .zip. Locky biasanya mentarget bisnis-bisnis kecil. Penyerang yang membuat Locky berasal dari Rusia dan juga membuat perangkat berbahaya bernama Dridex.(MalwareBytes, 2020c)

Locky menyerang seluruh dunia kecuali Rusia, negara teratas yang terserang perangkat berbahaya ini yaitu Spanyol, Jerman, USA, Perancis, Italia, Kanada, dan Polandia.

d. Ryuk

Ditemukan pada Agustus 2018, Ryuk adalah karakter fiksi yang ada di komik dan kartun Jepang. Saat ini Ryuk menjadi perangkat normal berbahaya yang menginfeksi banyak komputer di dunia. Seperti perangkat lunak berbahaya lainnya, Ryuk mengenkripsi *file* lalu meminta bayaran, namun Ryuk meminta bayaran yang sangat besar.

Ryuk menargetkan organisasi besar yang biasanya menjamin penyerang mendapatkan dibayar, korban termasuk EMCOR, rumah sakit UHS, dan beberapa outlet koran. Ryuk diestimasikan mendapatkan US$61 juta dalam jangka waktu 1 tahun 8 bulan (Februari 2018 – Oktober 2019).(MalwareBytes, 2020d)



Gambar 2. 4 Tampilan *ransomware* locky

Ryuk berkerja dengan cara mengidentifikasi dan mengenkripsi jaringan, dan menghapus *shadow copies* hingga habis yang menyebabkan penyerang dapat menonaktifkan pemulihan *Windows*, sehingga membuat sangat sulit untuk memulihkan data tanpa mumpunyai data cadangan di luar *server*.

2.1.3 Mesin Virtual

Mesin virtual adalah lingkungan yang dibuat secara virtual didalam komputer, yang memiliki CPU, memori, dan jaringan tersendiri berbeda dengan komputer yang menjalankan aplikasi mesin virtual, dan jika hardware komputer mencukupi dalam satu komputer dapat menjalankan lebih dari 1 mesin virtual dengan beragam sistem operasi.(RedHat, 2019)

Kegunaan dari mesin virtual sendiri bermacam-macam seperti untuk mengetes sistem operasi yang ingin dicoba, atau pun membuat *Workstation* sehingga sistem utama tidak bercampur, namun pada penelitian ini mesin virtual dibangun untuk menganalisis *ransomware* berbahaya, mesin virtual berguna untuk mengisolasi *file* berbahaya sehingga tidak menginfeksi sistem utama.

2.1.4 Sandbox

Sandbox adalah sebuah mekanisme keamanan untuk memisahkan program yang sedang berjalan. Sandbox sering digunakan untuk mengeksekusi kode yang belum teruji atau program yang tidak dipercaya dari pihak ketiga yang tidak diverifikasi, pemasok, pengguna yang tidak dipercaya dan situs yang tidak dipercaya (Bremer, 2019). Konsep ini juga berlaku untuk melakukan analisis dinamis yang bekerja dengan cara menjalankan *ransomware* agar dapat dianalisis di dalam lingkungan yang terisolasi dan mendapatkan informasi tentang apa yang dilakukan oleh *ransomware*.

Teknik *Sandboxing* tentunya memiliki kelebihan dan kekurangan, tetapi teknik ini adalah teknik yang baik untuk melihat perilaku *ransomware*, jaringan, dan *file* yang diunduh oleh *ransomware*. Analisis dinamis akan sempurna jika analisis statis juga dilakukan untuk mendapatkan analisis yang lebih mendalam.

2.1.5 Analisis statis

Berbeda dengan analisis dinamis, metode analisis statis tidak mengaktifkan *ransomware* namun menelusuri dan menganalisis kode sumber yang ada pada *ransomware* dengan membedah *ransomware* tersebut. Hal ini mengakibatkan informasi dan gambaran tentang mekanisme kerja *ransomware* yang didapatkan sangat lengkap dan mendetail. Pemahaman bahasa mesin terutama arsitektur program diperlukan dalam analisis statis untuk membantu analisis kode-kode program *ransomware* dengan cara mengumpulkan informasi dari perilaku yang ditimbulkan oleh *ransomware* tersebut (Cahyanto *et al*., 2017).

Analisis Statis memerlukan alat yang berbeda-beda dalam melakukan setiap langkah analisis, berikut alat-alat yang akan digunakan untuk melakukan analisis statis berdasarkan alur proses analisis yang penulis lakukan.

A. HxD Editor

Hexa dan Disk *editor* adalah *tools* yang dapat melihat dan mengedit nilai nex namun pada analisis ini *tool*s tersebut hanya digunakan untuk melihat nilai hex untuk mengetahui apakah *file* termasuk Portable Executeable/PE (K. A., 2018)

B. EXEinfoPE

EXEinfope adalah tool yang digunakan untuk mengetahui apakah *file ransomware* berbentuk paket atau tidak. Jika berbentuk paket tool ini dapat membuka paket *ransomware* karena jika pada proses analisis *ransomware* masih dalam bentuk paket akan menganggu hasil analisis sehingga membuatnya tidak akurat.(Datta and Anil Kumar, 2021)

C. PeStudio

PeStudio adalah *tool*s yang digunakan untuk melihat struktur *file* yang akan dianalisis seperti nilai *string*, *import*, *export*, dan kapan *file* pertama kali di-*compile.* PeStudio adalah *tool* yang telah terkoneksi oleh database VirusTotal sehingga *tool* ini sangat memudahkan proses analisis.(Datta and Anil Kumar, 2021)

D. CFFexplorer

CFFexplorer adalah *tool* yang juga dapat melihat struktur *file* yang akan dianalisis sama seperti PeStudio. Namun bedanya CFFexplorer tidak tersambung pada database VirusTotal. CFFexplorer bergunasebagai perbandingan dari hasil analisis PeStudio. (Datta and Anil Kumar, 2021)

E. VirusTotal

VirusTotal adalah *website* yang memiliki database setiap virus yang pernah terdeteksi oleh antivirus lainnya. Kegunaan *website* ini adalah untuk mengecek apakah *file* yang akan dianalisis ini adalah *ransomware* atau tidak. VirusTotal juga memiliki banyak fitur lainnya yang dapat membantu menganalisis *ransomware*. (K. A., 2018)

2.1.6 Analisis Dinamis

Metode analisis dinamis memeriksa *file* dalam suatu lingkungan aman berupa mesin fisik ataupun mesin virtual. *File* akan dijalankan dalam lingkungan aman ini agar dapat dianalisis untuk mengetahui dampaknya pada komputer. Proses ini berguna untuk mengetahui kegiatan yang dilakukan suatu *ransomware* ketika menyerang komputer. Hal-hal yang diamati dalam analisis dinamis termasuk proses-proses yang berjalan, perubahan *registry*, komunikasi internet, dan peristiwa janggal lainnya yang dapat terjadi ketika *ransomware* menyerang suatu komputer (Cahyanto *et al*., 2017).

Untuk menjelaskan lebih lanjut ini adalah contoh dari analisis dinamis yang telah dilakukan dengan menggunakan website yang memberikan fasilitas untuk melakukan analisis *ransomware* secara gratis ataupun berbayar ditunjukkan pada gambar 2.5 dengan menggunakan website anyrun.com.

Gambar 2. 5 Tampilan analisis dinamis oleh tool lain

dengan tool yang disediakan oleh web anyrun.com, sebagai benchmark atau gambaran hasil dari analisis dinamis yang akan dilakukan. Pada gambar dijelaskan beberapa hasil yang dapat dihasilkan dari hasil analisis dinamis, seperti:

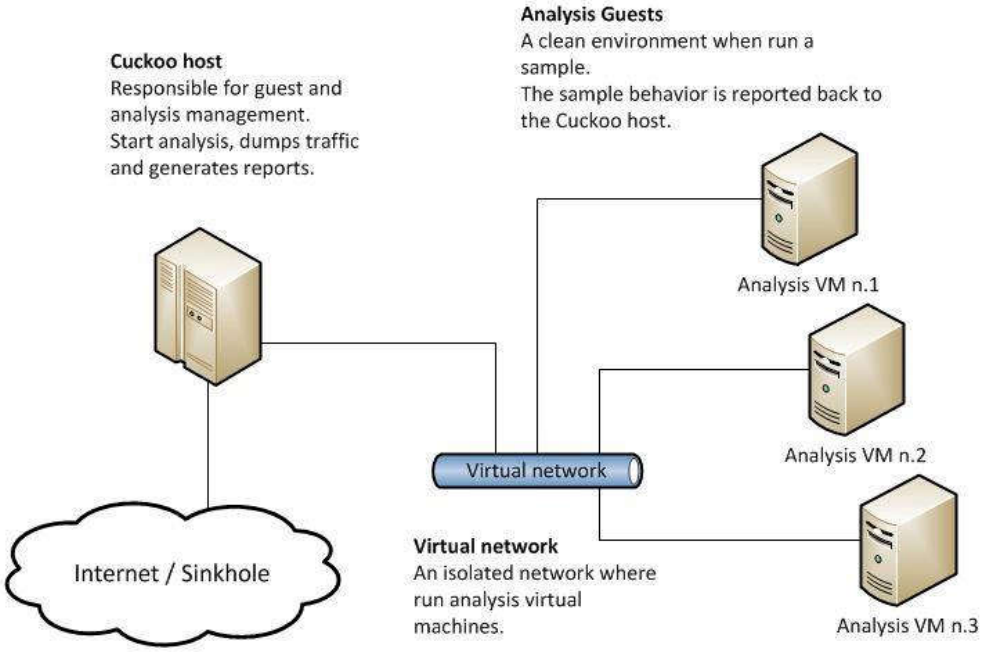
1. Penggunaan CPU
2. Proses Tree
3. Jaringan Komputer

Namun pada penelitian ini dilakukan analisis dinamis dengan tool Cuckoo sandbox, analisis dengan menggunakan cuckoo sanbox dilakukan secara otomatis, volatitility adalah salah satu tool yang mendukung tool cuckoo sandbox agar dapat melakukan analisis dengan baik, lalu yang terakhir menggunakan tool Virtualbox digunakan untuk membuat mesin virtual. Berikut adalah penjelesan tool yang digunakan pada anaalisis dinamis:

A. Cuckoo Sandbox

Cuckoo Sandbox adalah alat untuk melakukan analisis *ransomware* secara otomatis yang digunakan untuk menjalankan dan menganalisis *file* secara otomatis lalu menguraikan aktivitas-aktivitas yang dilakukan *ransomware* saat berjalan pada sistem operasi. Pada penelitian ini digunakan Cuckoo Sandbox. Beberapa hasil yang didapatkan oleh Cuckoo Sandbox (Cuckoo sandbox, 2020) adalah:

- Jejak panggilan API Win32 yang dihasilkan oleh semua proses yang dilakukan oleh *ransomware*
- Proses memori yang diakses oleh *ransomware*
- *File* yang diunduh oleh *ransomware*
- Lalu lintas jaringan *ransomware*
- Screenshot dari *ransomware* yang dijalankan di dalam *VirtualBox*



Gambar 2. 6 Penjelasan tentang cara kerja Cuckoo sandbox

Setiap analisis dilakukan pada mesin virtual yang terisolasi, Infrastruktur Cuckoo terdiri dari *fhost* dan *guest*, di mana *fhost* adalah tempat Cuckoo terpasang dan *guest* sebagai tempat *ransomware* dijalankan.

B. Volatility

Volatility Framework adalah kumpulan *tool*s yang diimplementasikan dengan Python di bawah lisensi public-GNU untuk mengekstrak artefak digital dari sampel *volatile* memori (RAM). Framework ini dimaksudkan untuk memperkenalkan orang kepada teknik dan kompleksitas yang terkait dengan dari artefak digital dari sampel memori untuk penelitian di masa depan.(Volatility Foundation, 2020)

Volatility mendukung berbagai format *file* sampel dan kemampuan untuk mengonversi format antara lain adalah:

- Raw linear sample
- Hibernation *file*
- Crash dump *file*
- LiME format
- VirtualBox core dump

C. VirtualBox

VirtualBox adalah perangkat lunak virtualisasi yang dapat digunakan untuk mengeksekusi sistem operasi, yang berarti dapat menjalankan sistem operasi lain bersamaan dengan sistem operasi yang sedang dijalankan.

Kegunaan dari VirtualBoxsangat bermacam-macam namun pada umumnya VirtualBox digunakan untuk ujicoba dan simulasi instalasi suatu sistem operasi tanpa harus kehilangan sistem yang telah ada.

Dalam kasus ini VirtualBox akan digunakan sebagai lingkungan yang aman untuk menjalankan *ransomware*, agar proses analisis tidak merugikan diri sendiri dan perangkat keras lainnya yang berhubungan dengan satu jaringan. (Oracle, 2021)

2.1.7 Perbedaan Kedua Analisis

Analisis statis adalah proses analisis biner pada *ransomware* tanpa menajalankan program. Analisis statis pada umumnya dilakukan dengan menentukan *signature* dari *file* yang memiliki identifikasi unik pada *file* lalu melihat program agar dapat memahami bagaimana program berkerja dan apa yang dilakukan program dibalik tampilan layar saat infeksi *ransomware*.

Analisis dinamis adalah analisis yang menjalankan program dan melakukan observasi pada perilaku dengan tujuan dapat memberhentikan program untuk menginfeksi komputer lain dalam satu jaringan. Analisis *ransomware* dilakukan di dalam lingkungan tertutup agar tidak menginfeksi sistem utama.

Dengan dua analisis dijelaskan secara umum berikut adalah tabel yang menjelaskan perbedaan pada secara lebih rinci. (FEA Fundamentals, 2019)

Tabel 2.2 Perbedaan Dari Dua Analisis

Analisis Statis	Analisis Dinamis
Analisis statis melakukan analisis dengan melihat program *ransomware* biner tanpa harus menjalankankan program *ransomware*.	Analisis dinamis membutuhkan program untuk dijalankan pada lingkungan tertutup dan harus dimonitor dengan teliti dalam lingkungan yang tertutup.
Menggunakan pendekatan *signature-based* untuk analisis *ransomware*.	Menggunakan pendekatan *behavior-based* untuk analisis *ransomware*.
Melibatkan *fingerprint* dari *file*, memindai virus, mengambil nilai *string*, *import*, deteksi *packer*, dan *debugging*. Untuk teknik yang lebih unggul dapat melakukan *reverse-engineering* pada *ransomware*.	Melibatkan pemanggilan API, intruksi yang diberikan, perubahan *registry*, jaringan yang melakukan komunikasi pada komputer yang terinfeksi, memori yang diubah, dan masih banyak lagi.
Kurang efektif pada program *ransomware* yang memiliki kode yang rumit.	Efektif pada semua tipe *ransomware* karena analisis melakukan eksekusi untuk menjalankan program.

Namun walaupun kedua analisis memiliki perbedaan bukan berarti hasil dari kedua analisis tersebut bertolak belakang satu sama lain. Hasil analisis dari setiap metode menghasilkan data yang membantu satu sama lain untuk mengerti lebih dalam program *ransomware* yang dianalisis.

2.1.8 Penelitian Sebelumnya

Telah dilakukan reviu penelitian sebelumnya yang berkaitan dengan analisis *ransomware* dan mengklasifikasi *ransomware* untuk memetakan evolusi *ransomware*. Dan juga telah menemukan buku, paper, dan jurnal yang dapat menunjang penelitian ini. Salah satu dari paper yang ditulis oleh Kul Prasad Subedi, Daya Ram Budhathoki, dan Dipangkar Dasgupta dengan judul "*Forensic Analysis of Ransomware Families Using Static and Dynamic Analysis*".(Subedi, Budhathoki, and Dasgupta, 2018a)

Paper tersebut membahas tentang proses menganalisis *ransomware* yang bertujuan untuk mengidentifikasi keluarga *ransomware*, yang dilakukan dengan metode analisis dinamis dan statis untuk dapat mengetahui lebih banyak perilaku tersembunyi *ransomware*.

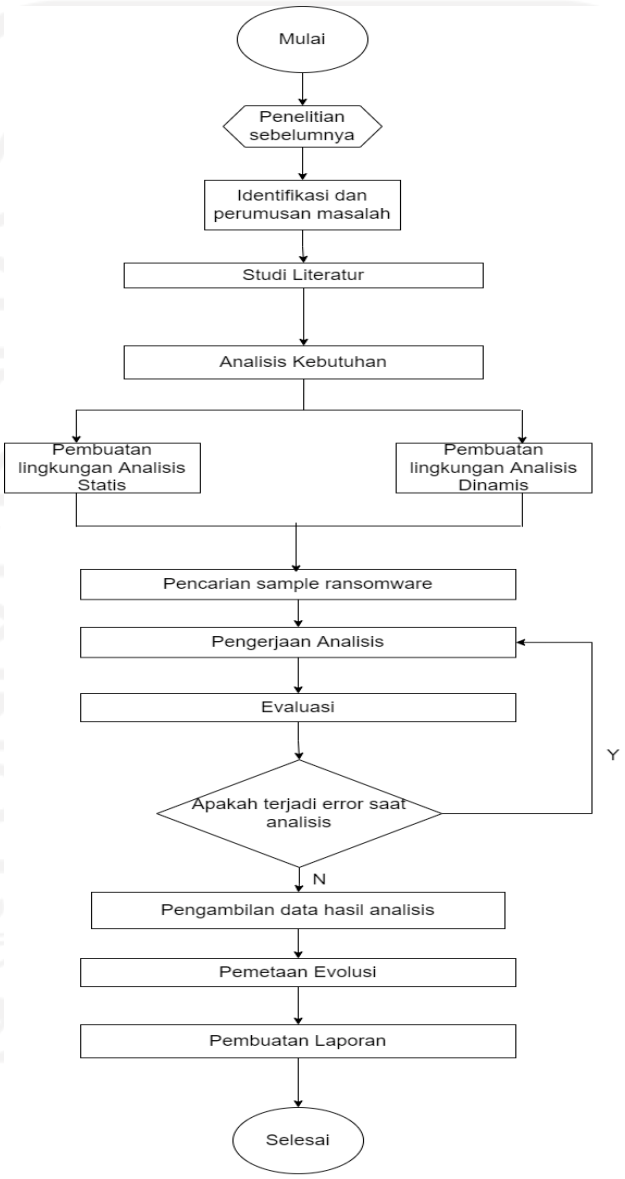
Telah dilakukan juga reviu pada paper yang ditulis oleh Mamoona Humayun, NZ Jhanjhi, Ahmed Alsayat,Vasaki Ponnusamy yang berjudul "*Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures*" (Zimba and Chishimba, 2019) dalam penelitian tersebut dibahas tentang evolusi *ransomware* dari tahun ke tahun seperti kerugian dan cara penyebaran suatu *ransomware*.

Untuk paper terakhir yang dilakukan adalah untuk mereferensi *tools* apa saja yang baik untuk dipakai dalam melakukan analisis statis dan analisis dinamis paper ini ditulis oleh Syarif Yusirwan, Yudi Prayudi, Imam Riadi yang berjudul "*Implementation of Ransomware Analysis using Static and Dynamic Analysis Method*" (YusirwanS, Prayudi, and Riadi, 2015) dalam paper ini dijelaskan *tools* yang digunakan untuk melakukan analisis *ransomware*, namun untuk *tools* yang disebutkan pada paper tidak semuanya dipakai untuk melakukan analisis ini.

BAB III
METODE PENELITIAN

3.1 Langkah Penelitian

Langkah penelitian yang dilakukan akan dijelaskan dengan *flowchart* pada gambar 3.1 yang memberikan gambaran penelitian secara umum.

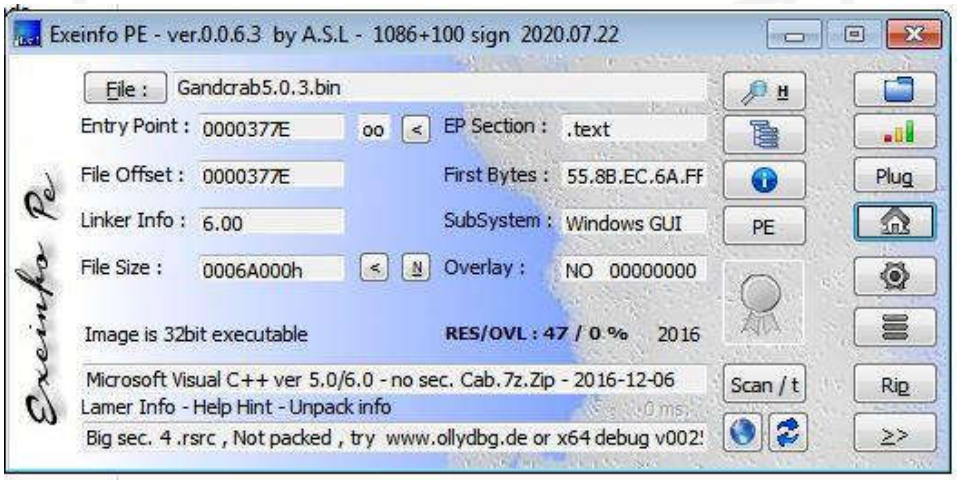


Gambar 3. 1 Diagram alir langkah-langkah penelitian

3.1.1 Alur Penelitian Metode Statis

A. Mengetahui paket dan tipe *file*

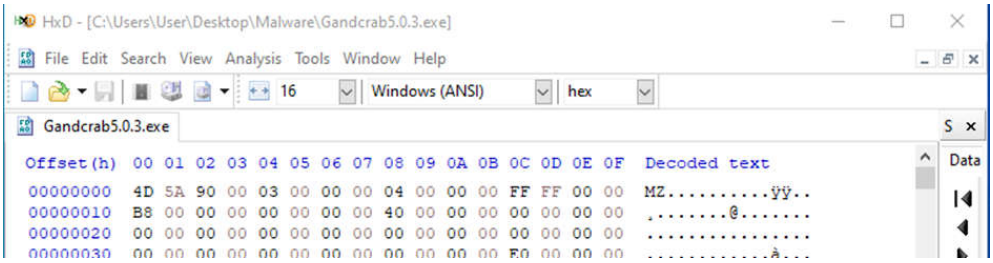
Ransomware biasanya dikirimkan dalam bentuk paket. Paket ini dapat mengompres *file* original yang menyembunyikan konten *ransomware* sehingga membuat *file* terlihat seperti *software* yang aman. Jika *ransomware* masih berbentuk paket maka *file* tersebut tidak dapat dianalisis secara penuh dan jika *ransomware* dianalisis dalam bentuk paket hasil *string* dan hasil analisis lainya terlihat tidak normal dan tidak dapat dibaca. Ketika *ransomware* dalam bentuk paket *file* dieksekusi maka *file* akan mendekompresi dan mengeksekusi *file* yang sesungguhnya.



Gambar 3. 2 Tampilan Exeinfo PE

Gambar 3.2 di atas adalah hasil pengecekan EXEinfope dimana hasil menunjukkan bahwa *file* ini tidak dalam bentuk paket.

Mengidentifikasi tipe *file* ini berguna untuk mengetahui *ransomware* menyerang tipe sistem operasi (Windows, Linux, dll) dan arsitektur (32-bit atau 64-bit). Jika *ransomware* berbentuk tipe *file* PE (p*ortable executable*), maka *ransomware* memiliki informasi tipe *file* format yang dapat dieksekusi oleh windows (.exe, .dll, .sys, .drv, dll) dari infromasi ini dapat diketahui bahwa *ransomware* tersebut menyerang sistem Windows.



Gambar 3. 3 Tampilan tool HxD

Dapat dilihat pada gambar 3.3 adalah metode manual untuk mengetahui tipe *file*. Ada banyak cara yang dapat digunakan untuk mengetahui tipe *file,* misalnya menggunakan CFFexplorer, PE32 dan Python.

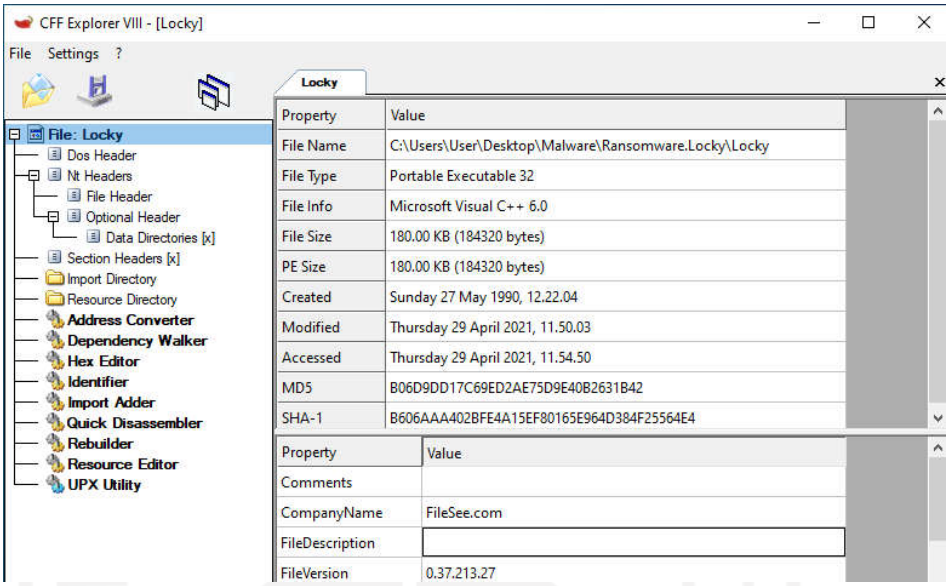
Dari hasil manual dengan tool HxD adalah contoh agar mengerti dasar dari mengetahui tipe *file*, dengan melihat 2 bit awal pada nilai hexadecimal yaitu 4D 5A jika diartikan dalam bentuk ASCII maka akan menghasilkan MZ dan dari data yang didapatkan dapat dicari lagi arti dari 2 bit tersebut.

Penggunaan metode manual memiliki kesulitan yang tinggi, namun penting untuk dilakukan untuk mendapatkan pengertian lebih tentang bagaimana setiap *file* memiliki bit yang berbeda sehingga komputer mengetahui tipe *file* untuk menjalankan *file* yang ingin dieksekusi.

B. Sidik jari *ransomware*

Sidik jari menghasilkan nilai *hash cryptographic* untuk membuat nilai biner berdasarkan isi di dalam *file*. Nilai *hash* yang dimaksud ialah MD5, SHA1, atau SHA256. Kegunaan nilai *hash* dalam menganalisis *ransomware* adalah untuk:

- mengidentifikasi *Ransomware*, walaupun memiliki nama yang berbeda jika isi *file* memiliki kesamaan nilai hash maka akan tetap terdeteksi sebagai *ransomware* yang sama.
- dalam analisis dinamis ketika *ransomware* dieksekusi akan memperbanyak diri dan menyebar. *Ransomware* yang tersebar akan tetap memiliki nilai hash yang sama.
- nilai *hash* dapat membantu mengidentifikasi *file* dan *ransomware*.

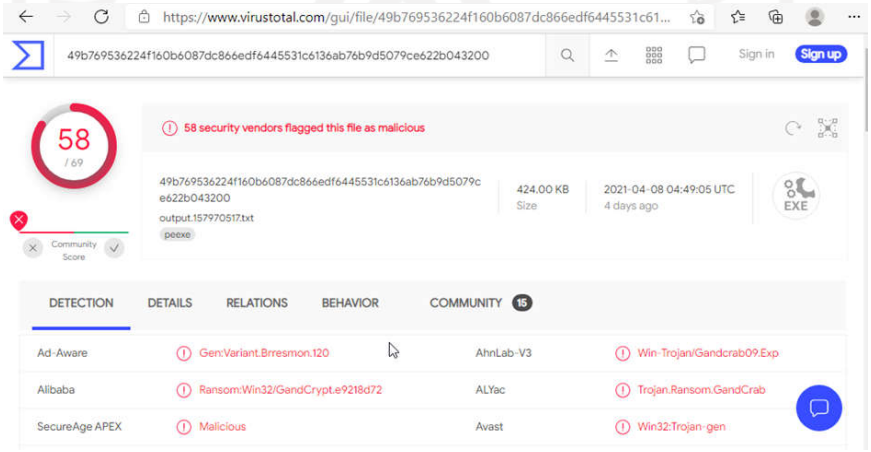


Gambar 3. 4 Tampilan tool CFF explorer

Gambar 3.4 mengambil nilai *hash* dapat memakai *tool* CFFexplorer ataupun PeStudio, yang nantinya nilai *hash* ini akan dimasukkan ke dalam *website* VirusTotal untuk pemindaian informasi sampel.

C. Pemindaian Informasi Sampel *Ransomware*

Pemindaian dapat menentukan apakah *file* memiliki kode berbahaya yang dapat merusak sistem. Hasil dari pemindaian dengan banyak antivirus akan dapat membantu proses analisis lebih mendetail karena setiap anti-virus akan memiliki nilai *scan* yang berbeda dalam penamaan dan tingkat bahaya suatu *ransomware*

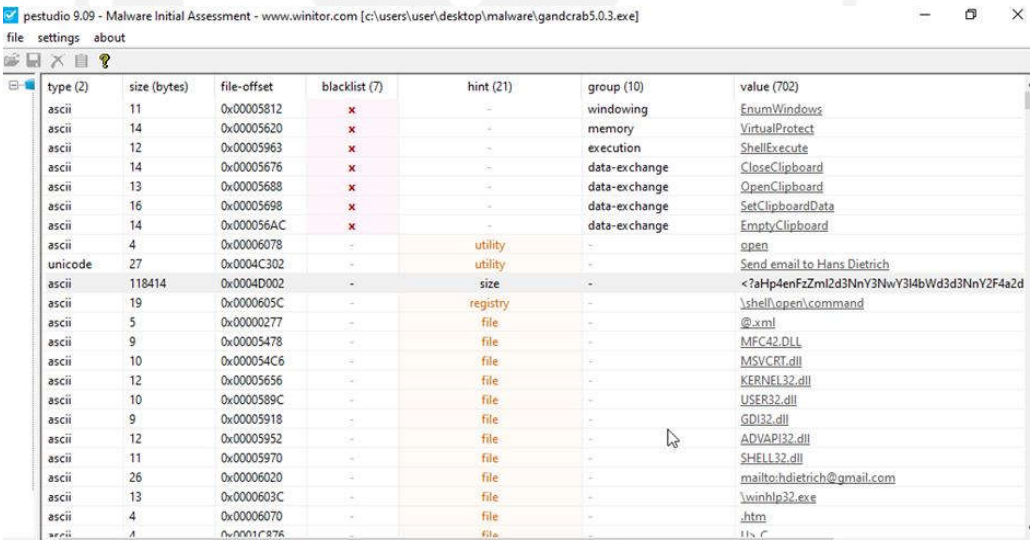


Gambar 3. 5 Tampilan tool Virustotal

Pada gambar 3.5 ditampilkan hasil pemindaian dengan VirusTotal yang dilakukan pada *website* VirusTotal. Data ini dapat dicapai dengan memasukkan nilai *hash* yang didapatkan pada gambar 3.4. Dengan terkonfirmasinya *file* sebagai *file* yang berbahaya maka analisis akan dilanjutkan, jika *file* yang dipindai tidak menghasilkan hasil positif yang menyatakan bahwa *file* tersebut berbahaya analisis tidak perlu dilanjutkan.

D. Mengambil nilai *string*

String adalah nilai ASCII dan Unicode yang tertanam di dalam *file*. Mengambil nilai *string* dapat memberikan petunjuk bagaimana cara program bekerja. Misalnya jika *ransomware* dapat membuat *file* maka nama *file* akan disimpan sebagai *string* atau jika *ransomware* memiliki nama domain yang dikontrol oleh penyerang maka nama domain akan disimpan dalam bentuk *string* serta informasi lain yang dapat memberikan petunjuk kemampuan *ransomware*.



Gambar 3. 6 Tampilan tool PeStuido

Gambar 3.6 Menunjukkan hasil dari nilai *string*, nilai ASCII, dan UNICODE. *String* dapat menunjukkan bagaimana *ransomware* berkerja saat proses infeksi namun untuk memastikan perlu dilakukan analisis dinamis agar dapat mengerti dengan baik bagaimana *ransomware* bekerja. Tidak semua *string* yang terekstrak memiliki nilai yang berguna untuk analisis, contoh string yang perlu diperhatikan termasuk *filename*, URL, IP *address*, *Registry Key*.

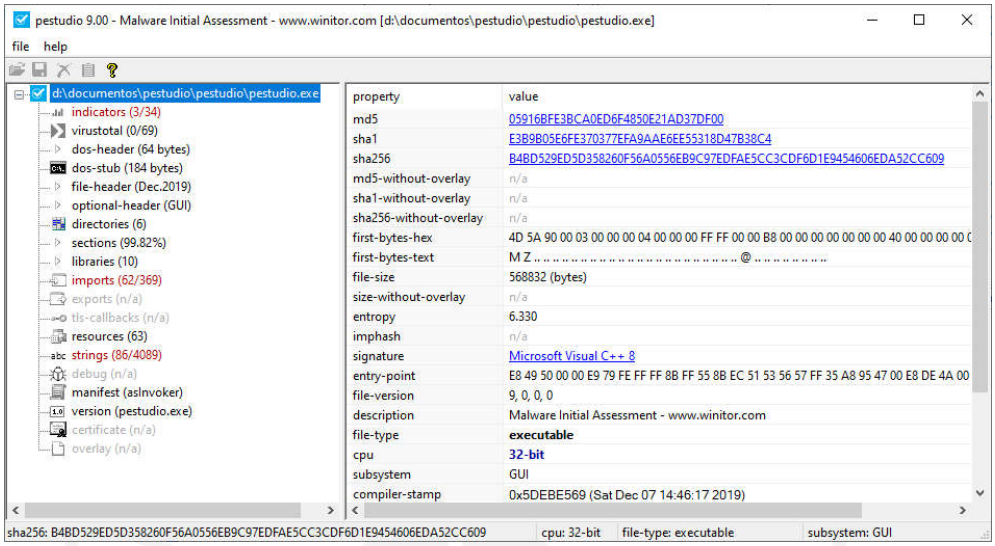
E. Inspeksi dari PE *header*

PE *header* merupakan bagian terpenting dalam melakukan analisis statis karena PE *header* memiliki banyak informasi yang berguna, PE *header* memiliki banyak informasi yang membantu analisis seperti informasi bagaimana sistem operasi menjalankan *executable file*, bagaimana *ransomware* berinteraksi dengan sistem operasi, informasi yang menspesifikasi di mana *executable file* harus dimuat ke dalam memori dan dapat menspesifikasi *file* di mana eksekusi dimulai. Selain itu PE header juga menyimpan informasi tentang *library* yang digunakan oleh *ransomware*.

Tabel 3. 1 Penjelasan PE

Variabel	Penjelasan
MZ Header/ DOS Header	Mendefenisikan *file* sebagai executable binary
DOS Stub (Program cannot be run in DOS mode)	Untuk menampilkan pesan jika dijalankan dalam DOS (untuk pengecekan kompabilitas)
PE *File* Header (*signature*)	Mendefinisikan executable sebagai PE
Image Optional Header	Menyimpan informasi tentang executable : Seperti subsystem dan Entry point
Section Tabel	Intruksi bagaimana memuat executable kedalam memori
Section	Komponen dari code executable dan data yang digunakan oleh executable

Tabel 3.1 di atas menjelaskan bagian-bagian dari PE header dimana setiap bagian header menjelaskan informasi yang berbeda dan pada gambar 3.7 dijelaskan tampilan *file header* seperti yang dijelaskan dalam tabel 3.1 *file header* adalah tempat informasi yang mendefinisikan bahwa *file* ini adalah *file* executable, serta informasi kapan *file* pertama kali di-*compile*.



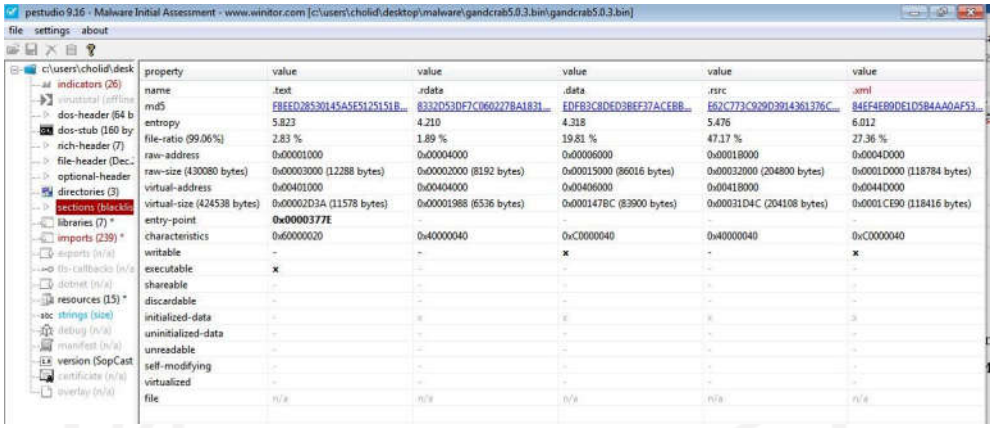
Gambar 3. 7 tampilan tool PeStudio

Dapat dilihat dalam tabel 3.1 di atas penjelasan bagian-bagian PE *header* dan pada bagian *section* disebutkan tentang data yang akan dijelaskan pada tabel 3.2

Tabel 3. 2 Penjelasan Section

Nama Header	Fungsi
.code / .text	Kode executable
.data	Menyimpan data (R/W)
.rdata	Menyimpan data (read only)
.idata	Menyimpan data import
.edata	Menyimpan data export
.rsrc	Menyimpan Resources data (string dan icon)

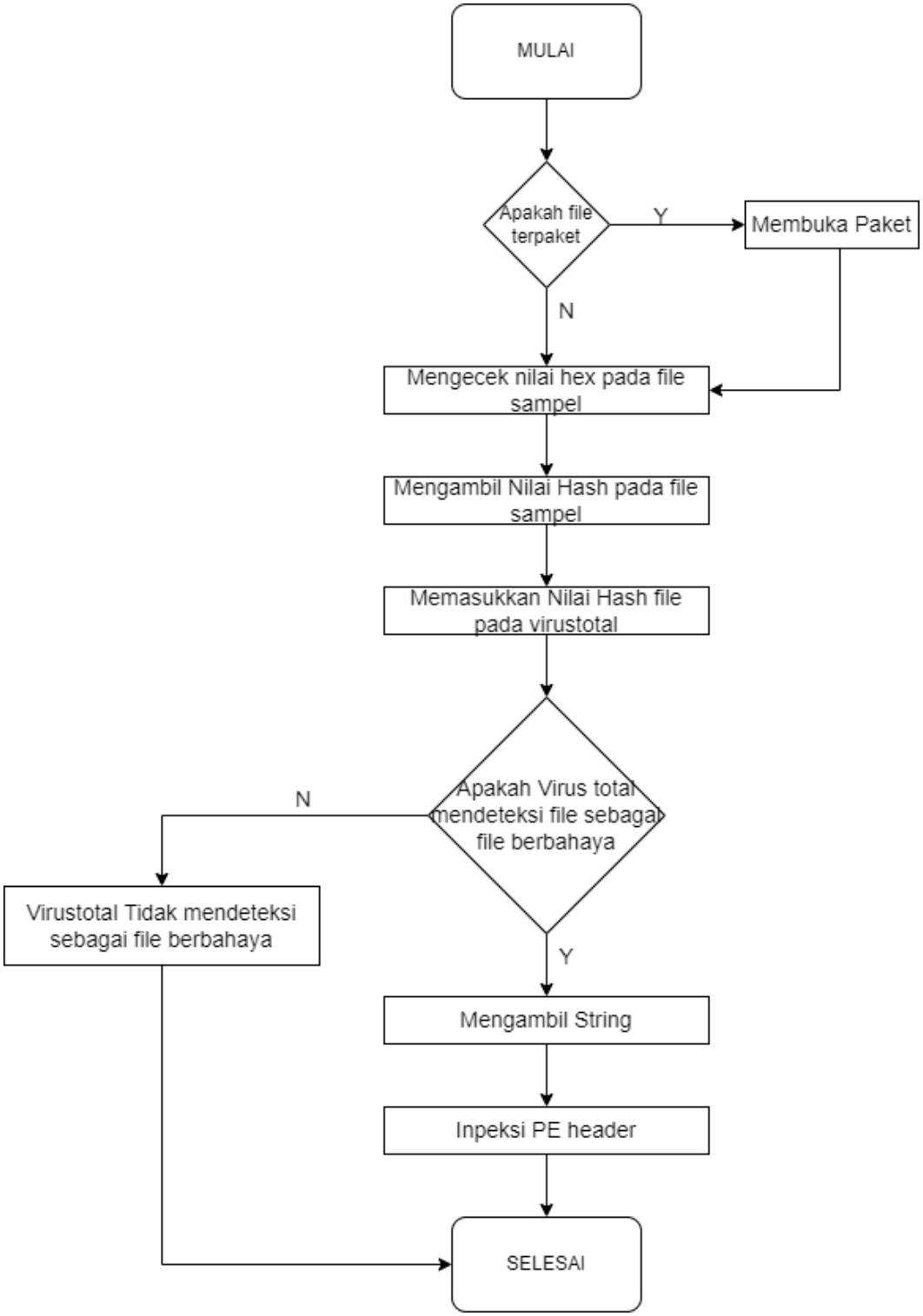
Untuk dimengerti PE atau Portabel executable bukan berarti hanya dimiliki oleh *ransomware* namun semua *file* memiliki PE, dan apa yang membedakan dari tabel 3.1 dan tabel 3.2, tabel 3.1 memiliki informasi detail tentang PE dan tabel 3.2 menyimpan data dan kode untuk executable.



Gambar 3. 8 Tampilan tool PeStuido

Gambar 3.8 adalah bagian dari section yang menampilkan apa yang dijelaskan pada tabel 3.1 dan banyak informasi lainya seperti e*ntry-point, writeable, executable , virtual-size*, dan *raw-size*. Namun pada inspeksi PE *header* dapat ditelusuri lagi lebih dalam dengan melihat informasi *import*, *library* dan *resource* yang akan dijelaskan lebih detail pada BAB IV.

Berikut adalah *flowchart* pengerjaan analisis statis yang dilakukan oleh penulis seperti yang dapat dilihat pada gambar 3.9

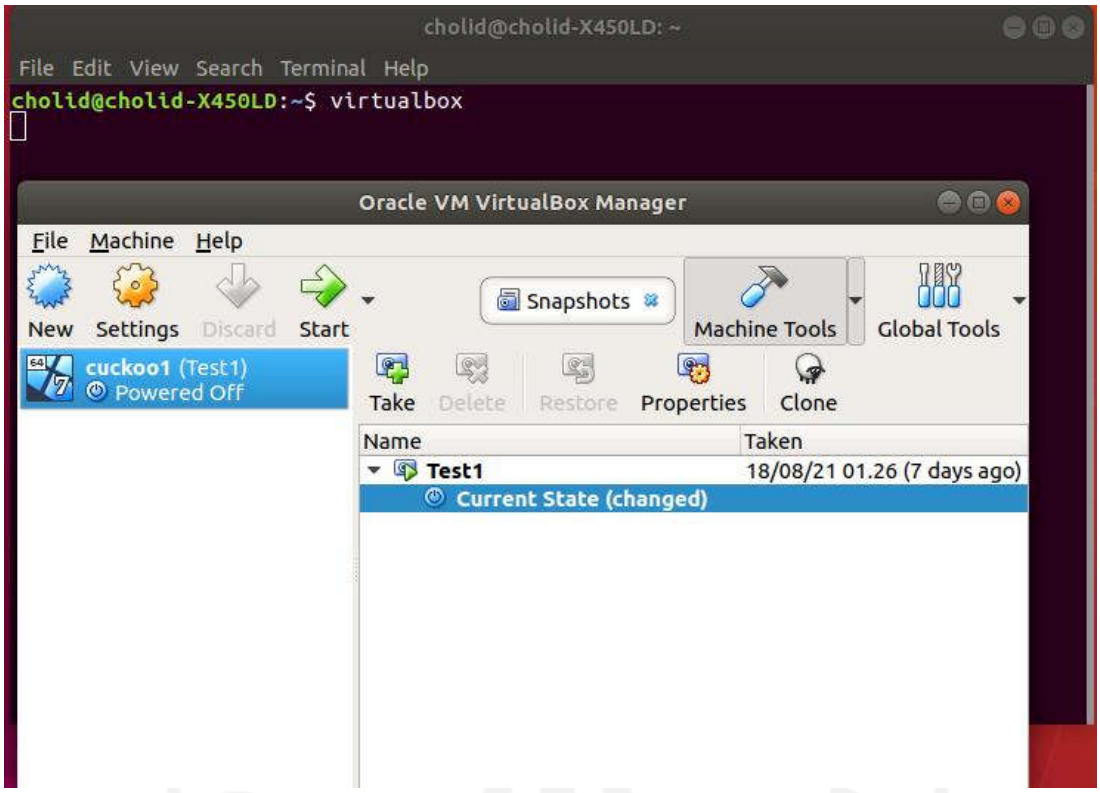


Gambar 3. 9 Alur pengerjaan analisis statis

3.1.2 Alur Penelitian Metode Dinamis

A. Menjalankan Virtualbox

Sebelum menjalankan cuckoo, cuckoo akan mengecek apakah virtualbox sudah siap, maka dari itu langkah pertama untuk menjalankan analisis dinamis adalah untuk menyiapkan virtualbox.

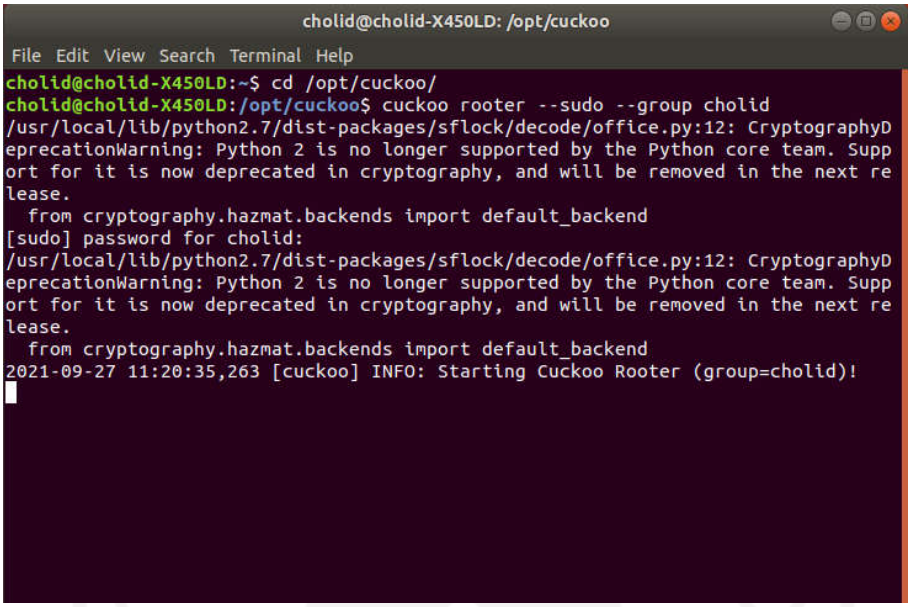


Gambar 3. 10 Menjalankan Virtualbox

Dengan command pada gambar 3.10 akan langsung menampilkan VirtualBox, dan tidak perlu untuk menjalankan mesin virtual cukup menjalankan VirtualBoxnya saja, lalu setelah dijalankan bisa langsung untuk menjalankan Cuckoo, grup, dan Cuckoo web.

B. Menjalankan Cuckoo

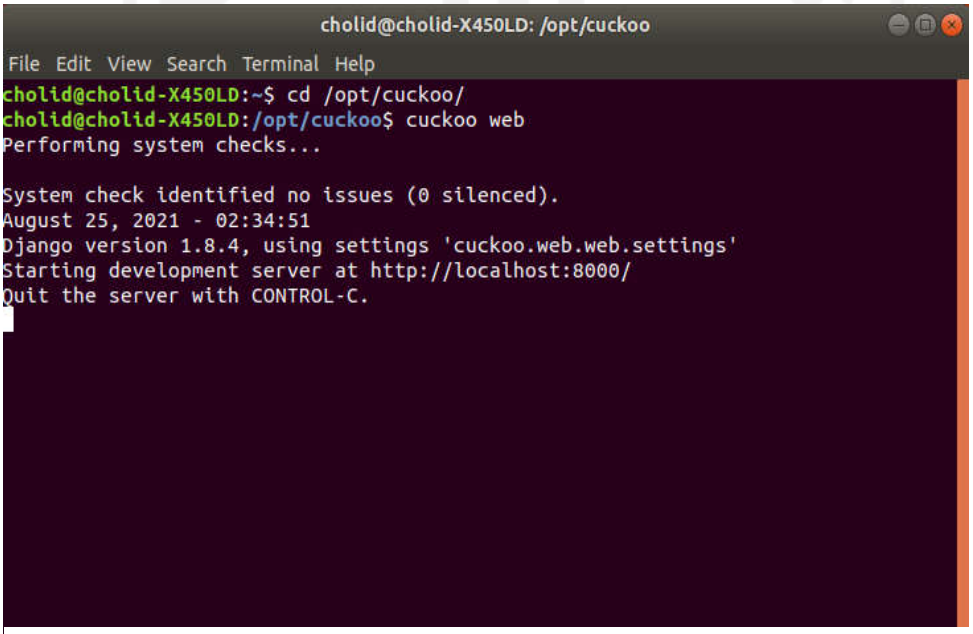
untuk langkah selanjutnya buka tiga terminal dan gunakan *command* cd /opt/Cuckoo/ pada tiga terminal lalu pada terminal jalankan *command* untuk menjalankan cuckoo root, web dan yang terakhir menjalankan Cuckoo.



Gambar 3. 11 Menajalankan Cuckoo rooter

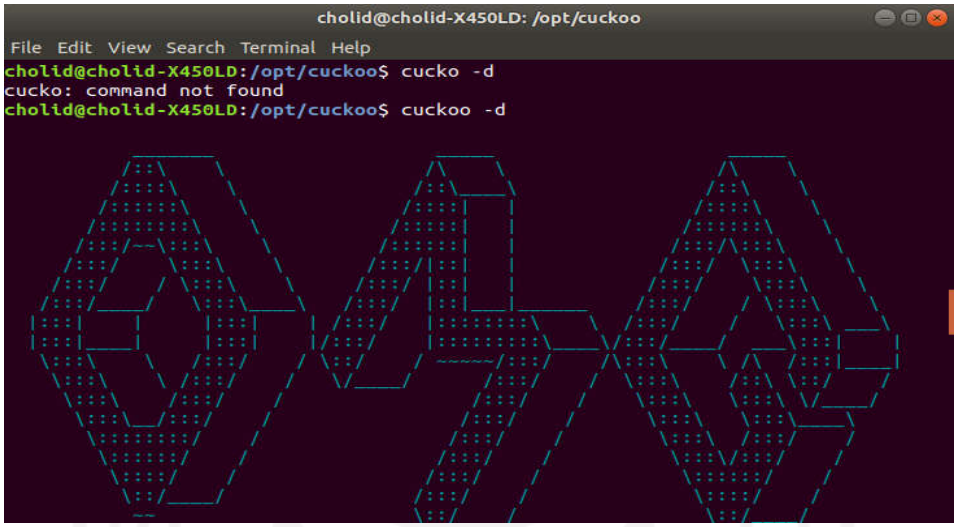
Gambar 3.11 menjalan kan *command* Cuckoo *rooter*. Cuckoo tidak disarankan dijalankan menggunakan *root*, maka *command* yang dijalankan menyediakan *root* akses kepada *user* yang dimasukkan pada grup Cuckoo tanpa harus menjalankan Cuckoo sebagai *root*.

Lalu setelah menjalankan cuckoo rooter, langkah selanjutnya menjalankan cuckoo web, cuckoo web pada gambar 3.12 berguna untuk menampilkan cuckoo interface, sehingga memudahkan dalam mengoperasikan cuckoo sandbox.



Gambar 3. 12 Menjalankan Cuckoo web

Setelah menjalankan Cuckoo web, command selanjutnya yaitu untuk menjalankan cuckoo sandbox, sehingga dapat memulai analisis.

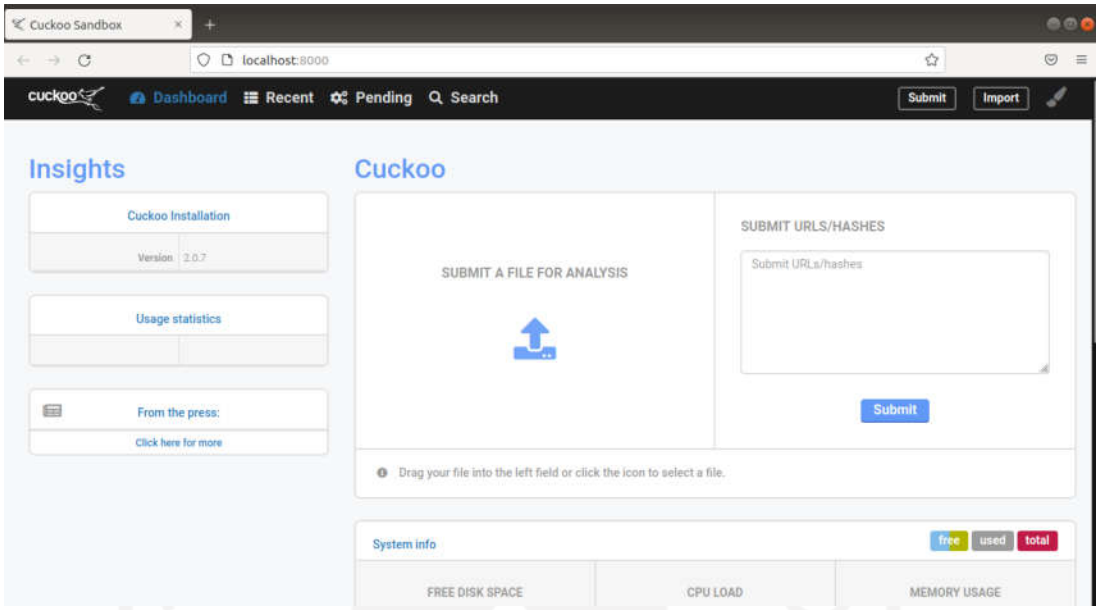


Gambar 3. 13 Menjalankan cuckoo sandbox

Command di atas untuk menjalankan Cuckoo agar dapat memulai dan melihat proses analisis Cuckoo. Setiap proses yang dilakukan oleh Cuckoo akan dilaporkan pada terminal yang ditunjukkan pada gambar 3.13

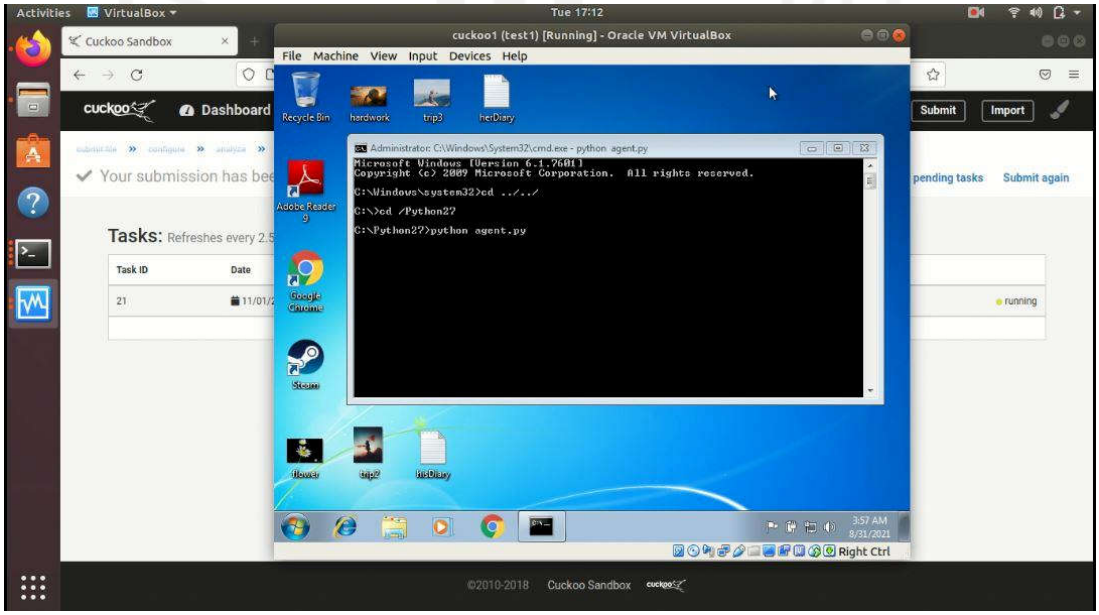
C. Memulai Analisis

Setelah semua terminal sudah dijalankan langkah selanjutnya yaitu untuk memasukkan sampel *ransomware* ke dalam Cuckoo web. Cuckoo akan menganalisis secara otomatis dan akan menampilkan laporan yang langsung dapat diakses pada *website* Cuckoo gambar 3.12.



Gambar 3. 14 Tampilan Cuckoo web

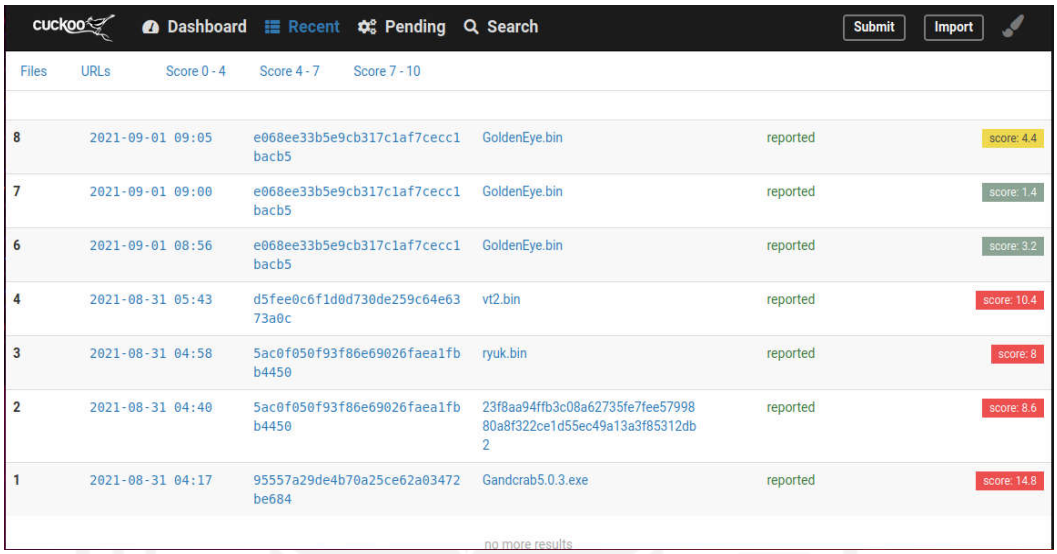
Dengan memasukkan *ransomware* kedalam cuckoo, cuckoo akan memulai analisis secara automatis, namun sebelum memulai analisis, cuckoo akan mengecek sekali lagi apakah virtualbox sudah siap, jika semua kondisi terpenuhi, cuckoo akan memulai analisis seperti pada gambar 3.15



Gambar 3. 15 Tampilan saat menjalankan Analisis

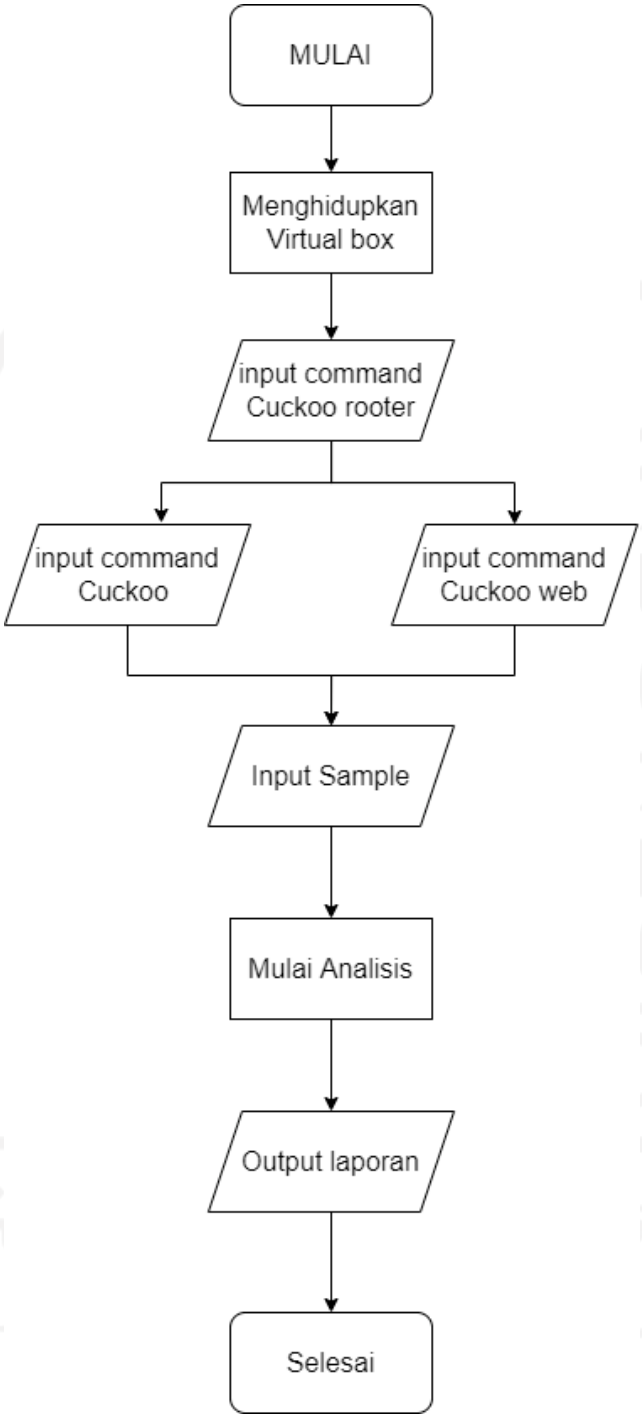
D. Laporan

Ketika mensubmit sampel *ransomware*, Cuckoo akan menyediakan hasil akhir seperti gambar () yang dapat diakses kapan saja pada Cuckoo *website*. Laporan dapat di import, dan disubmit pada komunitas cuckoo untuk dokumentasi.



Gambar 3. 16 Tampilan halaman laporan pada cuckoo web

Berikut adalah flowchart pengerjaan analisis statis yang dilakukan oleh penulis seperti yang dapat dilihat pada gambar 3.17



Gambar 3. 17 Alur penjelasan analisis dinamis

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan dalam penyelesaian penelitian analisis *ransomware* ini didasarkan pada kebutuhan sistem untuk menjalankan *tool* dan menganalisis *ransomware*, hasil dari analisis kebutuhan ini adalah:

3.2.1 Analisis Kebutuhan Perangkat Keras

a. Laptop

Laptop yang digunakan pada penelitian ini adalah laptop Asus A450L dengan spesifikasi prosesor Intel core i5-4200U dengan RAM 4 GB, pada analisis statis laptop menggunakan Windows 10 ketika sudah selesai, laptop diganti lagi dengan OS Ubuntu 18.04.

3.2.2 Analisis Kebutuhan Perangkat Lunak

A. Analisis Statis

Analisis statis memerlukan beberapa tool agar dapat menganalisis struktur *file*, dari mengecek struktur *file*, nilai hex, *file* hash, dan pengecekan *file* sampel, berikut adalah perangkat lunak yang digunakan.

- PeStudio

PeStudio adalah program untuk melihat struktur *file* yang terhubung langsung dengan *database* VirusTotal, dan PeStudio dikenal sebagai *software* untuk emergency response (CERT).

- CFF explorer

CFF explorer adalah alat yang didesain untuk mempermudah mengubah PE, tanpa harus mengorbankan fitur-fitur untuk melihat struktur *file*. Aplikasi ini digunakan untuk membandingkan hasil dengan aplikasi PeStudio agar mendapatkan informasi yang lebih akurat.

- VirusTotal

VirusTotal adalah *website* yang memiliki database setiap virus yang pernah terdeteksi oleh anti-virus lainnya sehingga berguna untuk mengecek apakah *file* yang akan dianalisis ini *ransomware* atau bukan. VirusTotal juga memiliki banyak fitur lainnya yang dapat membantu menganalisis *ransomware*.

- EXEinfope

EXEinfope adalah *tool* yang digunakan untuk mengetahui apakah *file ransomware* dalam berbentuk paket atau tidak, jika berbentuk paket maka *tool* ini dapat membuka

paket *ransomware*, karena jika pada saat analisis *ransomware* masih dalam berbentuk paket akan mengganggu hasil dari analisis membuat analisis menjadi tidak akurat.

- HxD

HxD adalah hex editor, memory editor, dapat mengubah atau melihat nilai hex sebuah *file*, namun untuk analisis statis tidak perlu untuk mengedit, hanya untuk melihat nilai hex pada *file*.

B. Analisis dinamis

Analisis dinamis memerlukan perangkat-perangkat lunak lainnya untuk mendukung kerja cuckoo sandbox dan perangkat-perangkat lunak tersebut adalah:

- Python 2.7

Python adalah bahasa pemograman yang dapat melakukan eksekusi sejumlah instruksi multiguna secara langsung (interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantic dinamis untuk memberikan tingkat keterbatasan (Advernesia,2018). Python 2.7 adalah Python yang perlu di install pada mesin *fhost* dan guest.

- VirtualBox

VirtualBox adalah software gratis yang mendukung Cuckoo dengan baik, virtual box juga software gratis oleh oracle yang fungsi utamanya adalah untuk menjalankan satu atau banyak sistem operasi (OS) di dalam Sistem Operasi utama

- CuckooSandbox

Cuckoo Sandbox adalah *tool* yang digunakan utuk menganalisis *ransomware* yang akan dieksekusi pada sistem operasi yang telah disiapkan

- MongoDB

MongoDB adalah salah satu produk database noSQL open source yang menggunakan struktur data JSON untuk menyimpan datanya. MongoDB adalah merupakan database noSQL yang paling populer di internet, MongoDB sering dipakai untuk aplikasi berbasis cloud, grid computing, atau big data, MongoDB diperlukan agar Cuckoo dapat ditampilkan dalam bentuk website sebagai GUI.

- Volatility

Volatility Framework adalah kumpulan *tool*s yanng diimplementasikan dengan Python di bawah lisensi public GNU, untuk ekstraksi artefak digital pada sampe volatile memori (RAM), Volatility dapat membantu Cuckoo untuk menganalisis lebih baik.

3.2.3 Analisis Kebutuhan Lainnya

A. Analisis Kebutuhan Masukan / Input

Kebutuhan input dalam proses analisis ini adalah *ransomware*. Sampel *ransomware* banyak tersedia baik di github maupun *website* yang berhubungan dengan *ransomware* yang ingin menggunakan sampel untuk tujuan akademis. Selanjutnya *ransomware* akan dianalisis dengan analisis dinamis dan analisis statis. *File ransomware* yang dibutuhkan yaitu *file* GoldenEye, Locky, Ryuk, dan Gandcrab.

B. Analisis Kebutuhan Proses

Proses analisis dinamis akan diproses langsung oleh Cuckoo, sedangkan analisis statis perlu untuk memahami PE *file* dan *string-string* yang digunakan oleh *ransomware*. Proses pemetaan evolusi dan klarikasi dari *ransomware* menggunakan diagram untuk mengklasifikasi karakteristik *ransomware*.

C. Analisis kebutuhan Keluaran / Output

Proses analisis statis mengeluarkan output tulisan yang perlu diproses dan dilihat secara teliti untuk mengetahui dan mengerti bagaimana *ransomware* berkerja tanpa dijalankan, sedangkan analisis dinamis akan menghasilkan laporan yang sudah terproses secara otomatis oleh Cuckoo Sandbox, lalu pada pembuatan diagram akan menampilkan hasil yang telah didapatkann dari kedua analisis.

3.2.4 Analisis Kebutuhan Proses

Pada saat melakukan penelitian analisis *ransomware* dengan metode statis dan dinamis, terdapat beberapa proses yang terjadi, proses berawal dari membuat lingkungan untuk melakukan analisis *ransomware*, lalu setelah lingkungan sudah dibuat langkah selanjutnya adalah untuk mengunduh sampel *ransomware*, yang diunduh dari repository Github, dan website analisis.

Setelah *ransomware* didapatkan, akan dilakukan proses analisis statis dan dinamis, setelah melakukan analisis dengan kedua metode dengan data analisis didapatkan dilakukan pemetaan evolusi pada *ransomware*.

3.2.5 Sampel *Ransomware*

Pencarian sampel *ransomware* dari setiap sampel yang didapatkan dapat dicari menggunakan google, github, dan biasanya website analisis seperti any.run dan hybrid anlysis memberikan sampel untuk dianalisis sendiri, setiap sampel memiliki format yang sama yaitu

dalam bentuk .zip dan memiliki password “infected”, dengan begitu berikut sampel yang didapatkan dari setiap website:

Gandcrab 5.03 : Github

Goldeneye : Anyrun.com

Locky : bazaar.abuse.ch/github

Ryuk: Anyrun.com

3.3 Instalasi Sistem

3.3.1 Kebutuhan Cuckoo Sandbox

Sebagai *fhost* sebelum menginstall Cuckoo ada beberapa yang harus disiapkan agar Cuckoo berjalan dengan baik, sebelum menginstall kebutuhan ini perlu dicek apakah Ubuntu yang terpakai sudah memiliki Python 2.7 karena Cuckoo hanya mensupport Python 2.7. berikut adalah beberapa kebutuhan yang perlu disintal dengan Linux command:

```
$ sudo apt-get install Python Python-pip Python-dev libffi-def
libssl-dev -y
$ sudo apt-get install Python-virtualenv Python-setuptools
$ sudo apt-get install libjpeg-dev zliblg-dev swig
```

Untuk menggunakan Cuckoo berbasis *web interface* diperlukan penginstalan dengan *command*:

```
$ sudo apt-get install mongodb
```

Penggunaan PostgreSQL sebagai *database* juga harus di-install dengan *command*:

```
$ sudo apt-get install postgresql libpq-dev
```

Setelah meng-install kebutuhan di atas, selanjutnya install VirtualBox di mana aplikasi ini nantinya akan menjadi tempat *ransomware* dijalankan sebab penggunaan OS Ubuntu 18.04 tidak memerlukan untuk pembaharuan repositori dan cukup langsung menggunakan *command*:

```
$ sudo apt-get update #untuk berjaga jaga jika repositori belum
terupdate
$ sudo apt-get install VirtualBox
```

Tidak lupa untuk memasukkan *user* di grup vbox*user*, grup vbox*user* akan terbuat secara otomatis, tidak disarankan untuk menjalankan Cuckoo dalam keadaan *root* dengan begitu lakukan dengan *command*:

```
$ sudo usermod -a -G vboxuser cholid
```

Setelah melakukan pemindahan *user*, selanjutnya install aplikasi yang dapat membantu Cuckoo untuk menganalisis dengan maksimal seperti Tcpdump, Volatility, m2crypto, dan dstrom3. Mulai instalasi dari tcpdump dikarenakan tidak disarankan untuk menjalankan Cuckoo dengan *root* akses memerlukan pembuatan grup dan pemberian akses pada *user*.

```
$ sudo apt-get install tcpdump
$ sudo grupadd pcap sduo
$ sudo usermod -a -G pcap cholid
$ sudo chgrp pcap /usr/sbin/tcpdump
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

Lalu untuk mengecek apakah dapat mengakses tcpdump tanpa menggunakan *root* gunakan *command*:

```
$ getcap /usr/sbin/tcpdump
#akan mengeluarkan hasil seperti ini
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip
```

Setelah meng-install tcpdump, selanjutnya akan meng-install *volatility* dengan menggunakan repositori github yang telah disediakan dengan *command*:

```
$ git clone https//github.com/volatilityfoundation/volatility
#setelah diclone silahkan pindah directory dengan menggunakan
$ cd ./volatility
#lalu install dengan command
$ sudo Python install setup.py install
```

Untuk *tool* terakhir cukup dengan menggunakan *command*:

```
$ sudo pip install m2crypto
#dan
$ sudo apt-get install distorm3
```

Dengan semua kebutuhan Cuckoo terinstall dengan begitu dapat langsung meng-install Cuckoo.

3.3.2 Instalasi dan Konfigurasi Cuckoo

Setelah menyiapkan kebutuhan Cuckoo, saatnya untuk menginstall Cuckoo dan mengonfigurasinya. Cuckoo dapat di-instal dengan *command*:

```
$ sudo pip install -U pip setuptool
$ sudo pip install -U Cuckoo
```

```
#setelah terinstall harus memindahkan Cuckoo working directory
$ sudo mkdir /opt/Cuckoo
$ sudo chown cholid:cholid /opt/Cuckoo
$ Cuckoo –cwd /opt/Cuckoo
```

Cuckoo telah terinstall lalu akan dilakukan konfigurasi pada beberapa *file* Cuckoo yaitu: Cuckoo.conf, VirtualBox.conf, memort.conf, reporting.conf

Konfigurasi Cuckoo.conf

```
# Specify the name of the machinery module to use, this module
will
# define the interaction between Cuckoo and your virtualization
software
# of choice.
machinery = VirtualBox

# Enable creation of memory dump of the analysis machine before
shutting
# down. Even if turned off, this functionality can also be enabled
at
# submission. Currently available for: VirtualBox and libvirt
modules (KVM).
memory_dump = yes
```

Untuk konfigurasi ini cukup mengganti pilihan virtual machine dengan VirtualBox dan tidak lupa untuk mengaktifkan memory dump.

Konfigurasi VirtualBox.conf

```
# Default network interface.
interface = vboxnet0

# Specify a comma-separated list of available machines to be used.
For each
# specified ID you have to define a dedicated section containing
the details
# on the respective machine. (E.g. Cuckoo1,Cuckoo2,Cuckoo3)
machines = Cuckoo1

[Cuckoo1]
# Specify the label name of the current machine as specified in
your
# VirtualBox configuration.
label = Cuckoo1

# Specify the operating system platform used by current machine
# [Windows/darwin/Linux].
```

```
platform = Windows

# Specify the IP address of the current virtual machine. Make sure
that the
# IP address is valid and that the fhost machine is able to reach
it. If not,
# the analysis will fail.
ip = 192.168.56.101

# (Optional) Specify the snapshot name to use. If you do not
specify a snapshot
# name, the VirtualBox MachineManager will use the current
snapshot.
# Example (Snapshot1 is the snapshot name):
snapshot = Test1
```

Pertama untuk konfigurasi VirtualBox adalah menentukan jaringan yang akan dipakai dan IP *address* pada mesin virtual, lalu mengganti nama mesin, platform yang dipakai Windows, dan mengganti nama snapshot dengan nama yang telah ditentukan.

Konfigurasi memory.conf

```
[Volatility]
enabled = yes
filter = yes
```

Di konfigurasi memory.conf cukup untuk mengaktifkan volatilitas agar pada laporan analisis dapat menampilkan laporan volatilitas.

Konfigurasi reporting.conf

```
[mongodb]
enabled = yes
fhost = 127.0.0.1
port = 27017
db = Cuckoo
store_memdump = yes
```

Di konfigurasi ini cukup mengaktifkan mongodb agar dapat membuka tampilan web untuk melakukan analisis.

3.3.3 Konfigurasi VirtualBox

Hal pertama yang harus dilakukan ialah meng-instal sistem operasi dengan konfigurasi 2GB RAM, 2 core CPU dan *hard disk* 40GB untuk menghindari *ransomware* mendeteksi

bahwa mesin ini adalah mesin virtual. Ketika mesin sudah terpasang maka selanjutnya ialah membuat *fhost-only network* yang dapat dibuat dengan *command*:

```
$ vboxmanage fhostonlyif create

#command ini akan membuat Vboxnet0 yang nantinya akan dipakai dengan
command selanjutnya

$ vboxmanage fhostonlyif ipconfig vboxnet0 --ip 192.168.56.1
```

Setelah itu ubah konfigurasi *network* virtual dengan *fhost-only* lalu hidupkan mesin virtual untuk mengonfigurasi *network* di dalam mesin, di mana harus mematikan IPv6 dan mengonfigurasi IPv4 dengan IP yang telah disiapkan.

```
Static IP – 192.168.56.101
Default Gateway – 192.168.56.1
DNS – 8.8.8.8
#DNS dapat dikonfigurasi dengan DNS apa saja
```

Setelah mengonfigurasi mesin virtual tidak langsung mendapatkan koneksi internet, agar mendapatkan koneksi internet harus melakukan *routing* dengan *command* Linux:

```
$ sudo iptabels -A FORWARD -o ens32 -i vboxnet0 -s 192.168.56.0/24
-m conntrack –ctstate NEW -j ACCEPT
$ sudo iptabels -A FORWARD -m conntrack –ctstate ESTABLISHED,RELATED
-j ACCEPT
$ sudo iptabels -A POSTROUTING -t nat -j MASQUERADE

#lalu memforward IPv4 dengan
$ sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

Setelah mesin virtual mendapatkan internet langkah selanjutnya ialah mengunduh Python 2.7 32.bit dan Python Pillow, Python 2.7 untuk menjalankan Agent.py yang nanti akan dikirimkan kedalam mesin dan Python Pillow untuk mengaktifkan fitur *screenshot* pada mesin virtual saat analisis.

Anti-virus Windows , *firewall*, dan UAC (*User Account Control*) harus dinonaktifkan agar tidak mengganggu proses analisis. Kirimkan Agent.py yang disediakan oleh Cuckoo lalu jalankan CMD dengan Administrator *permission* dan jalankan Agent.py melalui *command* cmd:

```
$ Python Agent.py
```

Untuk langkah terakhir save *Snapshot* dari mesin dengan keadaan CMD masih berjalan, dengan begitu mesin virtual sudah siap untuk melakukan analisis.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil Analisis Statis

Hasil dari analisis statis telah dilakukan dan didapatkan hasil dari analisis statis pertama adalah struktur dari *ransomware* seperti:

- Nilai Hash
- Compiler-Stamp
- Section
- Tipe *file*

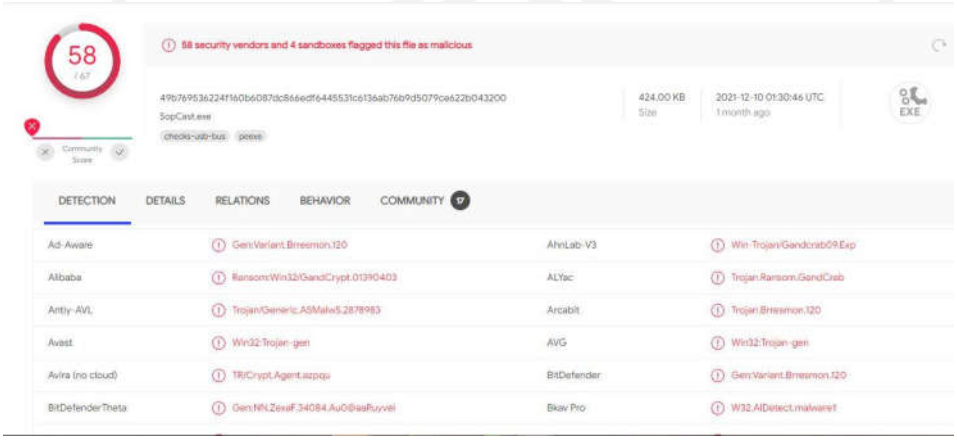
Yang telah didapatkan dengan tool seperti yang dijelaskan pada table. Lalu setelah tabel ringkasan, akan ditunjukkan String, import dan library yang digunakan oleh *ransomware* untuk menginfeksi komputer, string, import, dan library didapatkan dengan tool Pestudio.

Hasil analisis statis dilakukan untuk mengerti apa saja yang dapat dilakukan oleh malware sebelum menganalisis *ransomware* dengann metode dinamis. Dengan didapatkan gambaran bagaimana *ransomware* akan berperilaku pada saat proses infeksi.

Sebelum melakukan analisis statis diperlukan pengecekan pada setiap sampel, untuk memastikan bahwa sampel yang dianalisis adalah *file* berbahaya dengan begitu berikut adalah gambar-gambar yang menunjukkan bahwa *file* yang dinalisi adalah *file* berbahaya:

A. Gandcrab

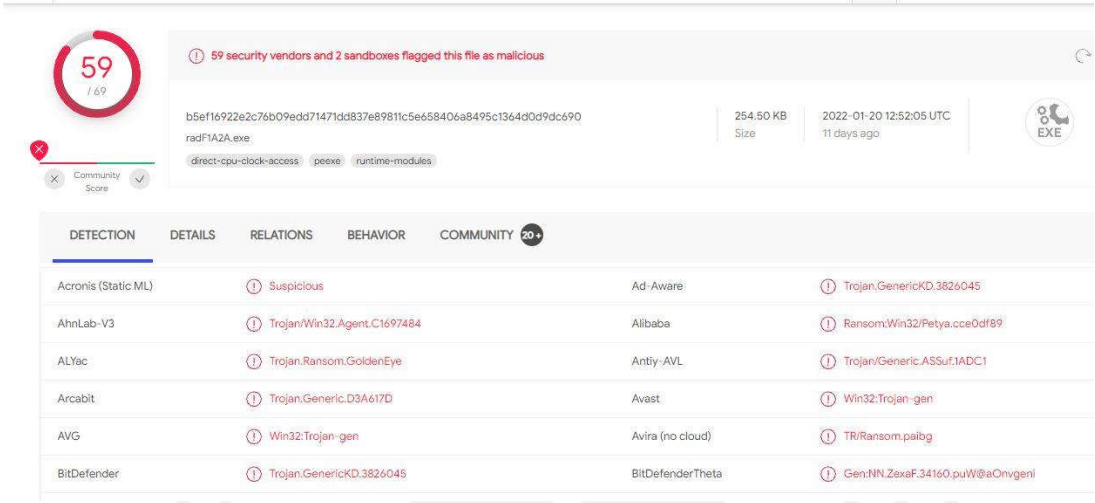
Dengan nilai MD5 Hash : 95557a29de4b70a25ce62a03472be684



Gambar 4. 1 Pengecekan sampel pada *file* gandcrab

B. GoldenEye

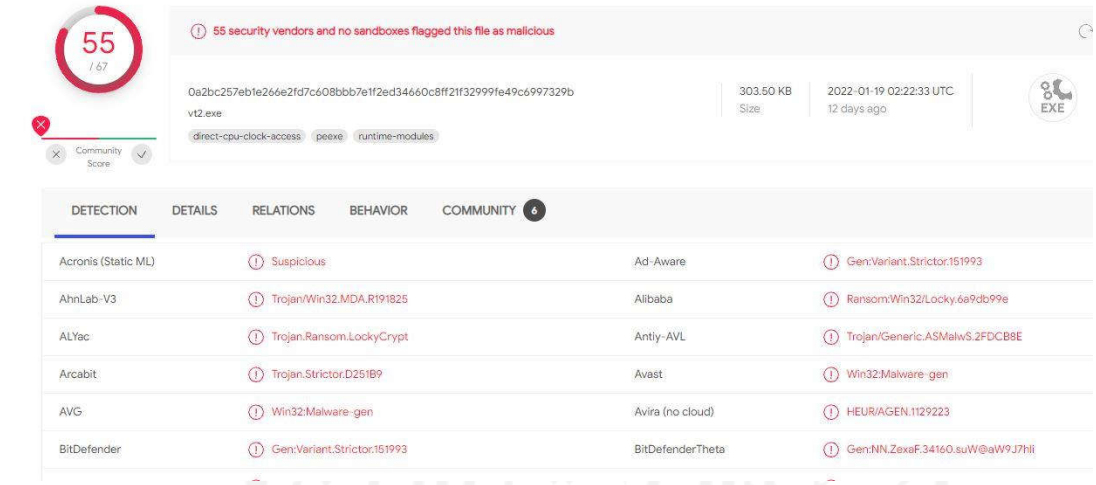
Dengan nilai MD5 Hash : e068ee33b5e9cb317c1af7cecc1bacb5



Gambar 4. 2 Pengecekan sampel pada *file* goldeneye

C. Locky

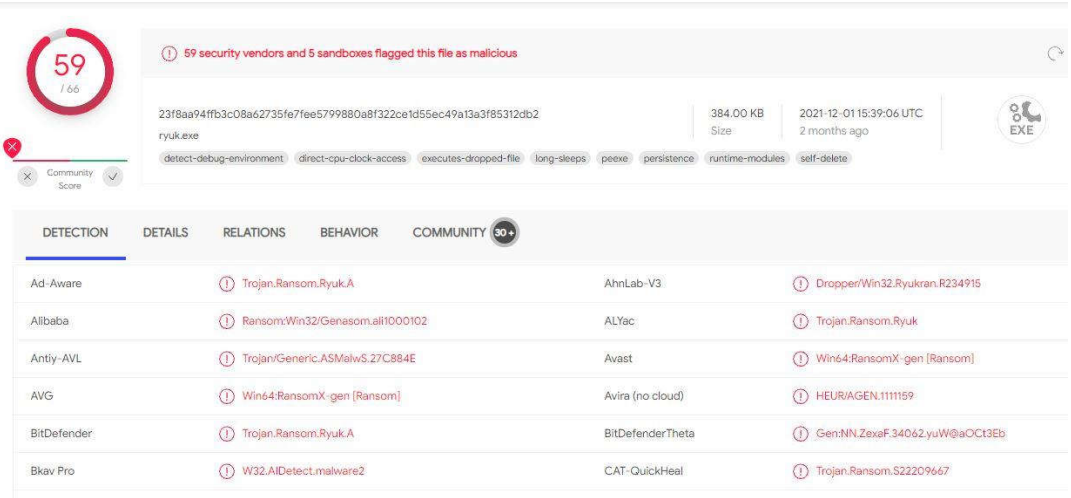
Dengan nilai MD5 Hash : d5fee0c6f1d0d730de259c64e6373a0c



Gambar 4. 3 Pengecekan sampel pada *file* locky

D. Ryuk

Dengan nilai MD5 Hash : 5AC0F050F93F86E69026FAEA1FBB4450

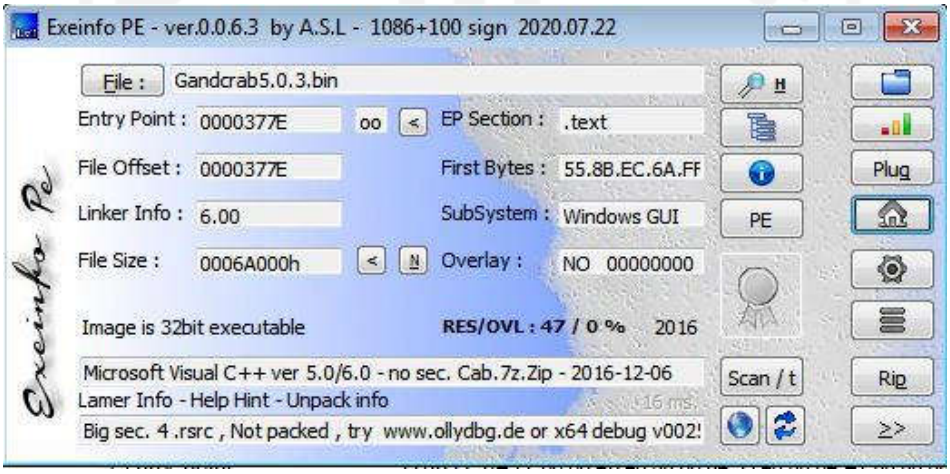


Gambar 4. 4 Pengecekan sampel *file* ryuk

Dengan terkonfirmasi sampel yang akan dianalis adalah *file* berbahaya, dengan menggunakan tool Virustotal, maka analisis dapat dilanjutkan.

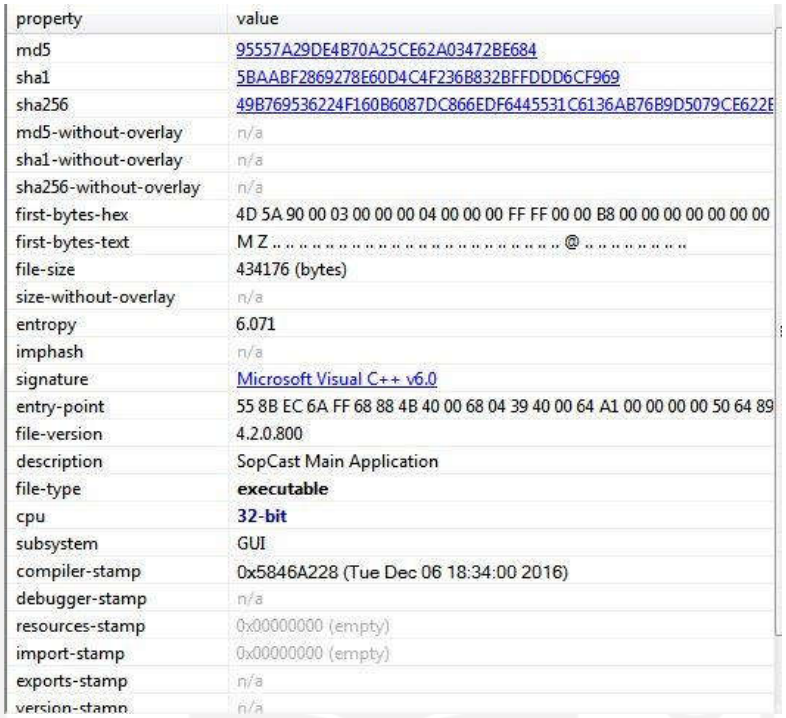
4.1.1 Gandcrab

Gambar 4.5 adalah hasil dari tool ExeinfoPe sebagai pengecek apakah sampel berbentuk paket atau tidak setelah dipastikan bahwa sampel tidak dalam bentuk paket maka dengan begitu analisis statis dapat dilakukan.



Gambar 4. 5 Pengecekan paket pada *file* gandcrab

Gambar 4.6 adalah gambar dari hasil tool PeStudio penjelasan singkat tentang struktur dari *file ransomware*. Seperti nilai hash, compiler-stamp, tipe *file* dan seperti yang dapat dilihat dari gambar, data ini nantinya akan dipersingkat lagi dalam bentuk tabel, yaitu tabel 4.1.



property	value
md5	95557A29DE4B70A25CE62A03472BE684
sha1	5BAABF2869278E60D4C4F236B832BFFDDD6CF969
sha256	49B769536224F160B6087DC866EDF6445531C6136AB76B9D5079CE622E
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	434176 (bytes)
size-without-overlay	n/a
entropy	6.071
imphash	n/a
signature	Microsoft Visual C++ v6.0
entry-point	55 8B EC 6A FF 68 88 4B 40 00 68 04 39 40 00 64 A1 00 00 00 00 50 64 89
file-version	4.2.0.800
description	SopCast Main Application
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5846A228 (Tue Dec 06 18:34:00 2016)
debugger-stamp	n/a
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	n/a
version-stamp	n/a

Gambar 4. 6 Ringkasan struktur *file* gandcrab

Tabel 4.1 adalah ringkasan dari *ransomware* Gandcrab, dari informasi ini dapat dicari lagi informasi-informasi lainnya yang berhubungan dengan *ransomware* ini dengan menggunakan nilai *hash* yang tertera pada tabel.

Tabel 4. 1 Ringkasan *Ransomware* Gandcrab

Tool	**Variabel**	**Value**
PeStudio	Nama	Gandcrab5.0.3.bin
	MD5	95557a29de4b70a25ce62a03472be684
	SHA-1	5baabf2869278e60d4c4f236b832bffddd6cf969
	SHA256	49b769536224f160b6087dc866edf6445531c6136ab76b9d5079ce622b043200
	Compiler-Stamp	06 Desember 2016 11:34:00
	Section	5
	Processor-32bit	True
	Executable	True

EXEinfoPE	Paket	False
VirusTotal	Skor	58/67

Informasi yang tertera pada tabel didapatkan dengan menggunakan tiga *tools*, yaitu PeStudio untuk menampilkan spesifikasi umum pada *ransomware* yang dianalisis seperti kapan *ransomware* di-*compile,* EXEinfoPE untuk mengetahui apakah *ransomware* dalam bentuk paket atau tidak dan VirusTotal untuk menunjukkan hasil kecocokan antara *ransomware* dengan *database* yang ada pada VirusTotal.

Berikut adalah tangkapan layar pada gambar () dari string yang digunakan oleh *ransomware* gandcrab. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah string yang dicurigai oleh PeStudio saja.

encoding (2)	size (bytes)	file-offset	blacklist (7)	hint (70)	group (10)	value (771)
ascii	11	0x00005812	x	import	windowing	EnumWindows
ascii	14	0x00005620	x	import	memory	VirtualProtect
ascii	14	0x00005676	x	import	data-exchange	CloseClipboard
ascii	13	0x00005688	x	import	data-exchange	OpenClipboard
ascii	16	0x00005698	x	import	data-exchange	SetClipboardData
ascii	14	0x000056AC	x	import	data-exchange	EmptyClipboard
ascii	12	0x00005962	x	-	execution	ShellExecute
ascii	4	0x00006078	-	utility	-	open
unicode	27	0x0004C2E6	-	utility	-	Send email to Hans I
unicode	15	0x0004CA98	-	url-pattern	-	www.sopcast.com
unicode	15	0x0004CAD8	-	url-pattern	-	www.sopcast.com
ascii	118414	0x0004D002	-	size	-	<?aHp4enFzZml2d3I
ascii	19	0x0000605C	-	registry	-	\shell\open\comma
ascii	8	0x00005718	-	import	windowing	IsWindow
ascii	12	0x0000579C	-	import	windowing	RedrawWindow
ascii	12	0x00005876	-	import	windowing	UpdateWindow
ascii	8	0x00005782	-	import	resource	CopyIcon
ascii	11	0x00005924	-	import	registry	RegCloseKey
ascii	16	0x000057D4	-	import	reckoning	GetSystemMetrics
ascii	12	0x000055A6	-	import	memory	GlobalUnlock
ascii	10	0x000055B6	-	import	memory	GlobalLock
ascii	11	0x000055C4	-	import	memory	GlobalAlloc
ascii	12	0x00005666	-	import	keyboard-and-mouse	EnableWindow
ascii	11	0x000055D2	-	import	dynamic-library	FreeLibrary
ascii	12	0x000057B8	-	import	desktop	GetCursorPos
ascii	17	0x00005484	-	import	-	__CxxFrameHandler
ascii	11	0x000054D4	-	import	-	__dllonexit
ascii	11	0x000054F4	-	import	-	_XcptFilter
ascii	13	0x00005514	-	import	-	__getmainargs
ascii	9	0x00005524	-	import	-	_initterm
ascii	16	0x00005530	-	import	-	__setusermatherr
ascii	12	0x00005544	-	import	-	_adjust_fdiv
ascii	12	0x00005554	-	import	-	__p__commode
ascii	10	0x00005564	-	import	-	__p__fmode
ascii	14	0x00005572	-	import	-	__set_app_type
ascii	16	0x00005584	-	import	-	_except_handler3
ascii	10	0x00005598	-	import	-	_controlfp

Gambar 4. 7 String dari *file* gandcrab

Dari gambar 4.7 diambil data yang menurut peneliti dan Pestudio, berbahaya dan akan dijelaskan lagi dalam bentuk tabel, yang dapat dilihat pada tabel 4.2.

Tabel 4. 2 String Gandcrab

Encoding	Value	Penjelasan
ascii	EnumWindows	Membuat aplikasi *ransomware* selalu di atas program lain
ascii	VirtualProtect	*Ransomware* dapat mengubah bagian read-only memory menjadi executable
ascii	CloseClipboard	Menutup Clipboard
ascii	OpenClipboard	Membuka clipboard dan mencegah aplikasi lain untuk memodifikasi konten clipboard
ascii	SetClipboardData	Menaruh data pada clipboard
ascii	EmptyClipBoard	Mengkosongkan clipboard lalu fungsi akan memberikan kepemilikan clipboard kepada window yang sedang membuka clipboard
ascii	Shellexecute	Untuk menjalankan program
ascii	IsWindow	Menentukan OS yang diinfeksi

Tabel 4.2 adalah nilai *string* yang didapatkan dari *ransomware* Gandcrab, *ransomware* Gandcrab memiliki lebih dari 8 *string*, namun isi dari tabel 4.2 adalah beberapa *string* yang di-*blacklist* oleh aplikasi PeStudio, dari informasi diatas dapat diketahui bahwa Gandcrab dapat mengubah memori (VirtualProtect), mendeteksi OS yang akan diinfeksi (IsWindow), dan memanipulasi tampilan Windows (EnumWindows).

Berikut adalah tangkapan layar pada gambar 4.8 dari library yang digunakan oleh *ransomware* gandcrab. Seluruh library akan dijelaskan lebih rinci dalam tabel 4.3.

library (7)	blacklist (0)	type (1)	imports (239)	description
mfc42.dll	-	implicit	158	MFCDLL Shared Library - Retail Version
msvcrt.dll	-	implicit	22	Windows NT CRT DLL
kernel32.dll	-	implicit	11	Windows NT BASE API Client DLL
user32.dll	-	implicit	38	Multi-User Windows USER API Client DLL
gdi32.dll	-	implicit	6	GDI Client DLL
advapi32.dll	-	implicit	3	Advanced Windows 32 Base API
shell32.dll	-	implicit	1	Windows Shell Common Dll

Gambar 4. 8 Libary yand digunakan oleh gandcrab

Tabel 4.3 adalah data yang diambil dari gambar 4.8 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* gandcrab.

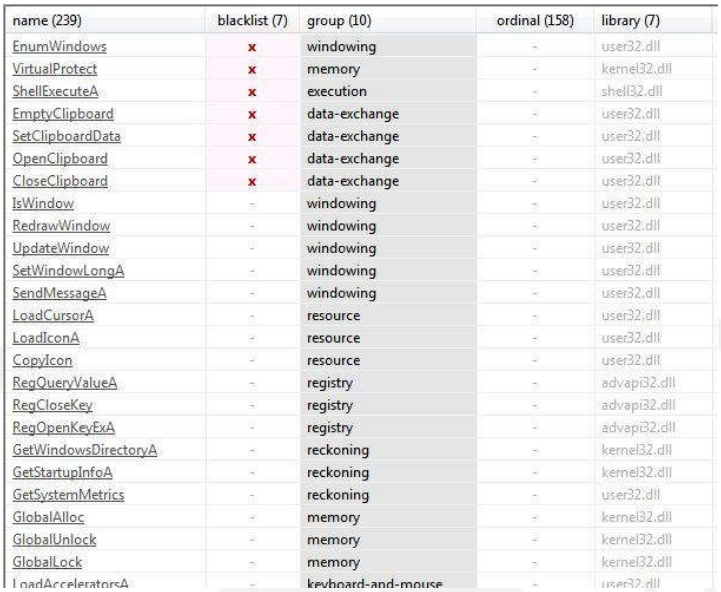
Tabel 4. 3 *Library* Yang Digunakan Oleh Gandcrab

Nama	**Jumlah fungsi**	**Penjelasan**
Kernel32.dll	11	Kernel32 sangat umum digunakan karena memiliki fungsi penting seperti akses, manipulasi memori. *File* dan *hardware*
Shell32.dll	1	*Library* yang berisi Windows Shell API yang digunakan untuk membuka web dan *file*
Advapi32.dll	3	Menyediakan akses ke komponen Windows seperti *service manager* dan *registry*
Mfc32.dll	158	Digunakan dengan visual studio C++ dapat mengontrol banyak fungsi dan komponen didalam aplikasi Windows
Msvcrt.dll	22	Membuat program yang ditulis dengan C++ berkerja dengan baik
*User*32.dll	38	Membuat program untuk menampilkan GUI
Gdi32.dll	6	Membuat program dapat mengekspor GDI

Tabel 4.3 menyediakan informasi *library* yang digunakan oleh Gandcrab, dari tabel ini juga dapat melihat perilaku *ransomware* melalui *library* yang digunakan, seperti mengakses *registry* serta mengakses dan memanipulasi memori *file* dan *hardware*. Namun *ransomware* Gandcrab menggunakan *import* terbanyak dengan *library* Mfc32.dll sebanyak 158.

Library ini menggunakan bahasa C++ sehingga dapat mengontrol banyak fungsi yang digunakan pada aplikasi Windows. Untuk mengurangi eror pada Mfc32.dll Gandcrab menggunakan *library* msvcrt.dll, untuk mensupport *library* mfc32.dll.

Berikut adalah tangkapan layar pada gambar 4.9 dari import yang digunakan oleh *ransomware* gandcrab. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah import yang dicurigai oleh PeStudio saja.



name (239)	blacklist (7)	group (10)	ordinal (158)	library (7)
EnumWindows	x	windowing	-	user32.dll
VirtualProtect	x	memory	-	kernel32.dll
ShellExecuteA	x	execution	-	shell32.dll
EmptyClipboard	x	data-exchange	-	user32.dll
SetClipboardData	x	data-exchange	-	user32.dll
OpenClipboard	x	data-exchange	-	user32.dll
CloseClipboard	x	data-exchange	-	user32.dll
IsWindow	-	windowing	-	user32.dll
RedrawWindow	-	windowing	-	user32.dll
UpdateWindow	-	windowing	-	user32.dll
SetWindowLongA	-	windowing	-	user32.dll
SendMessageA	-	windowing	-	user32.dll
LoadCursorA	-	resource	-	user32.dll
LoadIconA	-	resource	-	user32.dll
CopyIcon	-	resource	-	user32.dll
RegQueryValueA	-	registry	-	advapi32.dll
RegCloseKey	-	registry	-	advapi32.dll
RegOpenKeyExA	-	registry	-	advapi32.dll
GetWindowsDirectoryA	-	reckoning	-	kernel32.dll
GetStartupInfoA	-	reckoning	-	kernel32.dll
GetSystemMetrics	-	reckoning	-	user32.dll
GlobalAlloc	-	memory	-	kernel32.dll
GlobalUnlock	-	memory	-	kernel32.dll
GlobalLock	-	memory	-	kernel32.dll
LoadAcceleratorsA	-	keyboard-and-mouse	-	user32.dll

Gambar 4. 9 Tampilan import yang digunakan oleh gandcrab

Tabel 4.4 adalah data yang diambil dari gambar 4.9 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* gandcrab.

Tabel 4. 4 *Import* Yang Digunakan Oleh Gandcrab

Nama	**Kategori**	**library**	**Penjelasan**
EnumWindows	Windowing	*User*32.dll	Membuat aplikasi *ransomware* selalu di atas program lain
VirtualProtect	Memory	Kernel32.dll	*Ransomware* dapat mengubah bagian read-only memory menjadi executable
ShellExecuteA	Execution	Shell32.dll	Untuk menjalankan program
EmptyClipboard	Data-Exchange	*User*32.dll	Mengkosongkan clipboard lalu fungsi akan memberikan kepemilikan clipboard kepada window yang sedang membuka clipboard
OpenClipBoard	Data-Exchange	*User*32.dll	Membuka clipboard dan mencegah aplikasi lain untuk memodifikasi konten clipboard
SetClipboardData	Data-Exchange	*User*32.dll	Menaruh data pada clipboard
CloseClipboard	Data-Exchange	*User*32.dll	Menutup clipboard

Pada tabel 4.4 disediakan informasi fungsi yang diimpor dari *library*, biasanya yang tunjukkan pada string dan *import* tabel memiliki perbedaan namun pada *ransomware* Gandcrab memiliki kesamaan yang telah diblacklist oleh *tool* PeStudio.

4.1.2 GoldenEye

Gambar 4.10 adalah hasil dari tool ExeinfoPe sebagai pengecek apakah sampel berbentuk paket atau tidak setelah dipastikan bahwa sampel tidak dalam bentuk paket maka dengan begitu analisis statis dapat dilakukan.

Gambar 4. 10 Pengecekan paket pada *file* goldeneye

Gambar 4.11 adalah gambar dari hasil tool PeStudio penjelasan singkat tentang struktur dari *file ransomware*. Seperti nilai hash, compiler-stamp, tipe *file* dan seperti yang dapat dilihat dari gambar, data ini nantinya akan dipersingkat lagi dalam bentuk tabel, yaitu tabel 4.1.

property	value
md5	E068EE33B5E9CB317C1AF7CECC1BACB5
sha1	EF3D2563FA3E29C1BE76A149FF91398AB9987775
sha256	B5EF16922E2C76B09EDD71471DD837E89811C5E658406A8495C1364D0D9DC690
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	260608 (bytes)
size-without-overlay	n/a
entropy	7.015
imphash	n/a
signature	n/a
entry-point	E8 6E A4 00 00 E9 00 00 00 00 6A 14 68 D0 9F 43 00 E8 C6 47 00 00 E8 8A A7 00 00 0F B7 F0 6A
file-version	n/a
description	n/a
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x51C0CABD (Wed Jun 19 04:01:49 2013)
debugger-stamp	0x51C0CABD (Wed Jun 19 04:01:49 2013)
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	n/a
version-stamp	n/a
certificate-stamp	n/a

Gambar 4. 11 Ringkasan struktur *file* goldeneye

Tabel 4.5 adalah ringkasan dari *ransomware* GoldenEye, dari informasi ini dapat dicari lagi informasi-informasi lainnya yang berhubungan dengan *ransomware* ini menggunakan nilai hash yang tertera pada tabel.

Tabel 4. 5 Ringkasan *Ransomware* GoldenEye

Tool	**Variabel**	**Value**
PeStudio	Nama	GoldenEye.bin
	MD5	e068ee33b5e9cb317c1af7cecc1bacb5
	SHA-1	ef3d2563fa3e29c1be76a149ff91398ab9987775
	SHA256	b5ef16922e2c76b09edd71471dd837e89811c5e658406a8495c1364d0d9dc690
	Compiler-Stamp	Rabu, 19 Juni 2013 19:04:49
	Section	5

	Processor-32bit	True
	executable	True
EXEinfoPE	Paket	False
VirusTotal	Skor	59/69

Informasi yang tertera pada tabel didapatka dengan menggunakan tiga *tool*, yaitu PeStudio untuk menampilkan spesifikasi umum pada *ransomware* yang dianalisis seperti kapan *ransomware* di-*compile,* EXEinfoPE untuk mengetahui apakah *ransomware* dalam bentuk paket atau tidak dan VirusTotal untuk menunjukkan hasil kecocokan antara *ransomware* dengan database yang ada pada VirusTotal.

Berikut adalah tangkapan layar pada gambar 4.12 dari string yang digunakan oleh *ransomware* GoldenEye. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah string yang dicurigai oleh PeStudio saja.

encoding (2)	size (bytes)	file-offset	blacklist (42)	hint (225)	group (17)	value (2739)
ascii	19	0x00039C34	x	import	windowing	GetMessageExtraInfo
ascii	19	0x00039EA2	x	import	windowing	GetForegroundWind
ascii	16	0x00039FFC	x	import	windowing	GetDesktopWindow
ascii	16	0x00039C20	x	import	keyboard-and-mouse	UnregisterHotKey
ascii	11	0x00039DBE	x	import	keyboard-and-mouse	GetKeyState
ascii	16	0x00039DCC	x	import	keyboard-and-mouse	GetAsyncKeyState
ascii	9	0x0003A4B0	x	import	file	WriteFile
ascii	18	0x000399AC	x	import	execution	GetExitCodeProcess
ascii	16	0x000399C2	x	import	execution	GetCurrentThread
ascii	18	0x0003A442	x	import	execution	GetCurrentThreadId
ascii	19	0x0003A5CC	x	import	execution	GetCurrentProcessId
ascii	16	0x0003A66A	x	import	execution	TerminateProcess
ascii	14	0x0003A892	x	import	exception	RaiseException
ascii	14	0x00039C0E	x	import	diagnostic	RegisterHotKey
ascii	13	0x00039D6A	x	import	data-exchange	OpenClipboard
ascii	14	0x00039D7A	x	import	data-exchange	CloseClipboard
ascii	16	0x00039D8C	x	import	data-exchange	SetClipboardData
ascii	14	0x00039DA0	x	import	data-exchange	EmptyClipboard
ascii	21	0x0003A7C4	x	import	console	SetConsoleCtrlHand
ascii	16	0x00032E64	x	-	windowing	MonitorFromPoint
ascii	14	0x00032E78	x	-	windowing	GetMonitorInfo
ascii	12	0x0003A2A4	x	-	registry	RegCreateKey
ascii	13	0x0003A2C8	x	-	registry	RegSetValueEx
ascii	14	0x0003A2EC	x	-	registry	RegDeleteValue
ascii	18	0x00034354	x	-	file	CreateSymbolicLink
ascii	10	0x00039B14	x	-	file	DeleteFile
ascii	23	0x00034220	x	-	execution	SetThreadStackGuar
ascii	21	0x00034238	x	-	execution	CreateThreadpoolTi
ascii	31	0x00034264	x	-	execution	WaitForThreadpoolT
ascii	20	0x00034284	x	-	execution	CloseThreadpoolTim
ascii	20	0x0003429C	x	-	execution	CreateThreadpoolW
ascii	25	0x00034318	x	-	execution	GetCurrentProcesso
ascii	13	0x00039AAE	x	-	execution	CreateProcess
ascii	12	0x0003A31C	x	-	execution	ShellExecute
ascii	21	0x0003A5FC	x	-	execution	GetEnvironmentStrin
ascii	24	0x00034368	x	-	dynamic-library	SetDefaultDllDirecto

Gambar 4. 12 String yang digunakan oleh goldeneye

Dari gambar 4.12 diambil data yang menurut peneliti dan Pestudio, berbahaya dan akan dijelaskan lagi dalam bentuk tabel, yang dapat dilihat pada tabel 4.6.

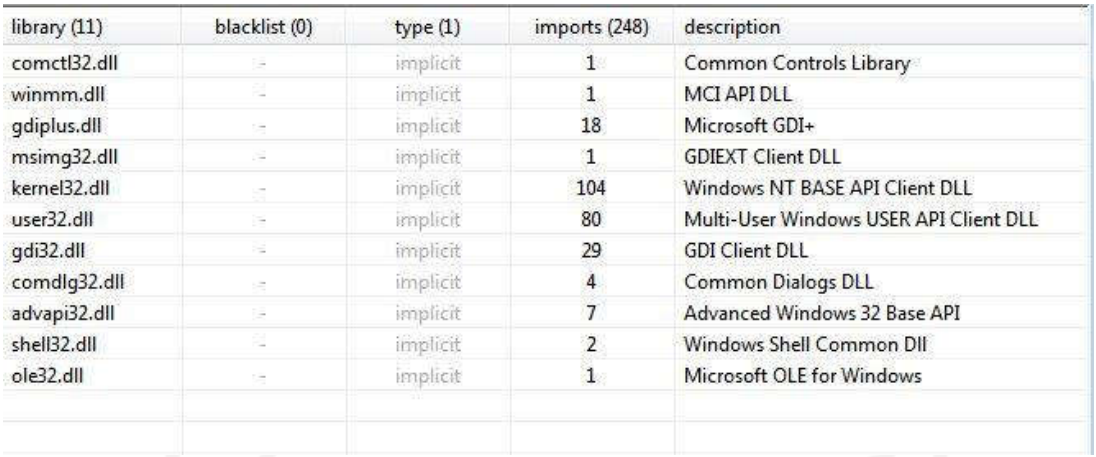
Tabel 4. 6 String Yang Digunakan GoldenEye

Encoding	Value	Penjelasan
ascii	MonitorFromPoint	Dapat menypesifikasikan aplikasi-aplikasi yang berada di dalam monitor secara virtual
ascii	GetMonitorInfo	Mengambil informasi tentang monitor
ascii	RegCreateKey	Membuat spesifik *registry* key jika sudah ada maka fungsi akan membukanya
ascii	RegSetValueEx	Menetapkan data dan tipe yang spesifik di dalam *registry* key
ascii	RegDeleteValue	Menghapus nilai dari *registry* key
ascii	CreateSymbolicLink	Membuat hubungan antara *file*
ascii	Delete*File*	Menghapus *file*
ascii	SetThreadStackGuarantee	Menetapkan size minimum, agar tidak terjadi eror
ascii	CreateTreadPoolTimer	Membuat timer baru pada objek
ascii	GetEnvironmentString	Mengambil variable untuk proses yang sedang berjalan
ascii	GetKeyState	Mengambil data dari spesifikasi virtual keyboard

Tabel 4.6 adalah nilai *string* yang didapatkan dari *ransomware* GoldenEye. *Ransomware* GoldenEye memiliki lebih dari 11 *string*, namun isi dari tabel 4.6 adalah beberapa string yang di-*blacklist* oleh aplikasi PeStudio.

Berdasarkan informasi pada tabel dapat diketahui bahwa GoldenEye dapat mempengaruhi tampilan komputer sehingga pengguna tidak dapat mengakses komputer. GoldenEye juga dapat mengubah *registry* komputer (RegSetValue), menghapus *file* (Delete*File*), membuat *timer* pada objek (CreateThreadPoolTimer), dan mendeteksi bahasa yang digunakan komputer (GetKeyState).

Berikut adalah tangkapan layar pada gambar 4.13 dari library yang digunakan oleh *ransomware* goldeneye. Seluruh library akan dijelaskan lebih rinci dalam tabel 4.7.



library (11)	blacklist (0)	type (1)	imports (248)	description
comctl32.dll	-	implicit	1	Common Controls Library
winmm.dll	-	implicit	1	MCI API DLL
gdiplus.dll	-	implicit	18	Microsoft GDI+
msimg32.dll	-	implicit	1	GDIEXT Client DLL
kernel32.dll	-	implicit	104	Windows NT BASE API Client DLL
user32.dll	-	implicit	80	Multi-User Windows USER API Client DLL
gdi32.dll	-	implicit	29	GDI Client DLL
comdlg32.dll	-	implicit	4	Common Dialogs DLL
advapi32.dll	-	implicit	7	Advanced Windows 32 Base API
shell32.dll	-	implicit	2	Windows Shell Common Dll
ole32.dll	-	implicit	1	Microsoft OLE for Windows

Gambar 4. 13 Library yang digunakan oleh goldeneye

Tabel 4.7 adalah data yang diambil dari gambar 4.13 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* goldeneye.

Tabel 4. 7 *Library* Yang digunakan GoldenEye

Library	***Import***	**Penjelasan**
Kernel32.dll	104	Kernel32 sangat umum digunakan karena memiliki fungsi penting seperti akses, manipulasi memory. *File* dan hardware
Shell32.dll	2	*Library* yang berisi Windows Shell API yang digunakan untuk membuka web dan *file*
Advapi32.dll	7	Menyediakan akses ke komponen Windows seperti service manager dan *registry*
Winmm.dll	1	Windows multimedia API yang mempunyai fungsi low-level audio dan joystick
Comctl32.dll	1	Menampilkan error pada *file* yang menggunakan tipe *file* Windows
*User*32.dll	80	Membuat program untuk menampilkan GUI
Gdi32.dll	29	Membuat program dapat menampilkan GDI
Gdiplus.dll	18	Memberikan fungsi lebih yang mendukung Gdi32.dll
Msimg32.dll	1	Untuk membuat gambar transparan dan gradien transparan
Ole32.dll	1	Untuk menjalankan object linking dan embedding
Comdlg32.dll	4	Berisi modul untuk menampilkan eror dan sukses

Tabel 4.7 adalah *library* yang digunakan oleh GoldenEye. Dari tabel ini juga dapat melihat perilaku *ransomware* melalui *library* yang digunakan seperti mengakses *registry* serta mengakses dan memanipulasi memory *file* dan hardware. Namun *ransomware* GoldenEye

menggunakan *import* terbanyak dengan *library* kernel32.ddl dan *user*32.dll, jika ditotalkan berjumlah 184 *import* yang digunakan.

Ransomware GoldenEye berfokus untuk mencegah komputer yang terinfeksi dapat diakses oleh pengguna, sehingga tidak ada pilihan lain selain membayar *ransomware*.

Berikut adalah tangkapan layar pada gambar 4.14 dari import yang digunakan oleh *ransomware* goldeneye. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah import yang dicurigai oleh PeStudio saja.

name (248)	blacklist (29)	group (15)	ordinal (1)	library (11)
GetDesktopWindow	x	windowing	-	user32.dll
GetForegroundWindow	x	windowing	-	user32.dll
GetMessageExtraInfo	x	windowing	-	user32.dll
RegCreateKeyA	x	registry	-	advapi32.dll
RegDeleteValueA	x	registry	-	advapi32.dll
RegSetValueExA	x	registry	-	advapi32.dll
GetAsyncKeyState	x	keyboard-and-mouse	-	user32.dll
GetKeyState	x	keyboard-and-mouse	-	user32.dll
UnregisterHotKey	x	keyboard-and-mouse	-	user32.dll
DeleteFileA	x	file	-	kernel32.dll
WriteFile	x	file	-	kernel32.dll
CreateProcessA	x	execution	-	kernel32.dll
TerminateProcess	x	execution	-	kernel32.dll
GetEnvironmentStringsW	x	execution	-	kernel32.dll
GetCurrentProcessId	x	execution	-	kernel32.dll
GetCurrentThreadId	x	execution	-	kernel32.dll
GetCurrentThread	x	execution	-	kernel32.dll
GetExitCodeProcess	x	execution	-	kernel32.dll
ShellExecuteA	x	execution	-	shell32.dll
RaiseException	x	exception	-	kernel32.dll
GetModuleHandleExW	x	dynamic-library	-	kernel32.dll
RegisterHotKey	x	diagnostic	-	user32.dll
ChangeDisplaySettingsExA	x	desktop	-	user32.dll
EmptyClipboard	x	data-exchange	-	user32.dll
SetClipboardData	x	data-exchange	-	user32.dll
CloseClipboard	x	data-exchange	-	user32.dll
OpenClipboard	x	data-exchange	-	user32.dll
SetConsoleCtrlHandler	x	console	-	kernel32.dll

Gambar 4. 14 Impor yang digunakan oleh goldeneye

Tabel 4.8 adalah data yang diambil dari gambar 4.14 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* goldeneye.

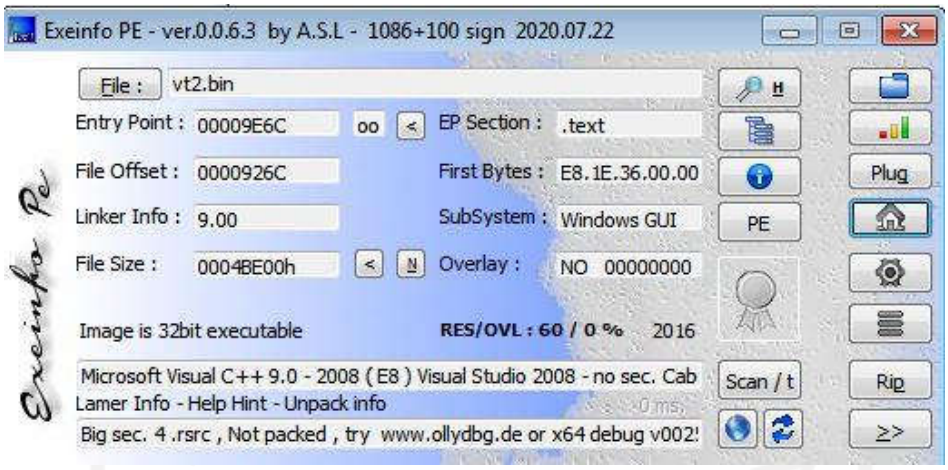
Tabel 4. 8 *Import* Yang Digunakan GoldenEye

Nama	grup	library	Penjelasan
GetDekstopWindow	Windowing	*User*32.dll	Dapat menampilkan program sesuai dengan spesikasi monitor
GetForegroundWindow	Windowing	*User*32.dll	Membuat *ransomware* sebagai aplikasi utama yang harus di tampilkan tanpa gangguan
GetMessageExtraInfo	Windowing	*User*32.dll	Menampilkan informasi ekstra
RegCreateKeyA	*Registry*	Advapi32.dll	Membuat spesifik *registry* key jika sudah ada maka fungsi akan membukanya
RegDeleteValueA	*Registry*	Advapi32.dll	Menghapus nilai dari *registry* key
RegSetValueExA	*Registry*	Advapi32.dll	Menetapkan data dan tipe yang spesifik di dalam *registry* key
GetAsyncKeyState	Keyboard-and-mouse	*User*32.dll	Mengambil informasi keadaan keyboard pada saat fungsi ini dipanggil
GetKeyState	Keyboard-and-mouse	*User*32.dll	Mengambil data dari spesifikasi virtual keyboard
UnregisterHotKey	Keyboard-and-mouse	*User*32.dll	Mematikan fungsi hotkey pada keyboard
Delete*File*A	*File*	Kernel32.dll	Menghapus *file*
Write*File*	*File*	Kernel32.dll	Memodifikasi *file*

Import yang digunakan oleh *ransomware* GoldenEye berfokus pada tabel 4.8 tampilan *ransomware* pada komputer yang ditunjukkan pada fungsi GetDesktopWindow dimana *ransomware* akan menampilkan notifikasi sesuai dengan monitor yang digunakan lalu GoldenEye berfokus untuk mengubah dan menghapus *registry*, mengubah dan menghapus *file*, serta mengecek bahasa yang digunakan oleh pengguna.

4.1.3 Locky

Gambar 4.15 adalah hasil dari tool ExeinfoPe sebagai pengecek apakah sampel berbentuk paket atau tidak setelah dipastikan bahwa sampel tidak dalam bentuk paket maka dengan begitu analisis statis dapat dilakukan.



Gambar 4. 15 Pengecekan paket pada *file* locky

Gambar 4.16 adalah gambar dari hasil tool PeStudio penjelasan singkat tentang struktur dari *file ransomware*. Seperti nilai hash, compiler-stamp, tipe *file* dan seperti yang dapat dilihat dari gambar, data ini nantinya akan dipersingkat lagi dalam bentuk tabel, yaitu tabel 4.9.

property	value
md5	D5FEE0C6F1D0D730DE259C64E6373A0C
sha1	894F45F50454001BD21AD2713FEFC15EB25B2B8B
sha256	0A2BC257EB1E266E2FD7C608BBB7E1F2ED34660C8FF21F32999FE49C6997329B
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	310784 (bytes)
size-without-overlay	n/a
entropy	7.360
imphash	n/a
signature	Microsoft Visual C++ 8
entry-point	E8 1E 36 00 00 E9 78 FE FF FF CC CC CC CC CC CC CC CC CC CC 8B 4C 24 04 F7 C1 03 00 00 00
file-version	n/a
description	n/a
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x584E1766 (Mon Dec 12 10:20:06 2016)
debugger-stamp	n/a
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	n/a
version-stamp	n/a

Gambar 4. 16 Ringkasan struktur *file* locky

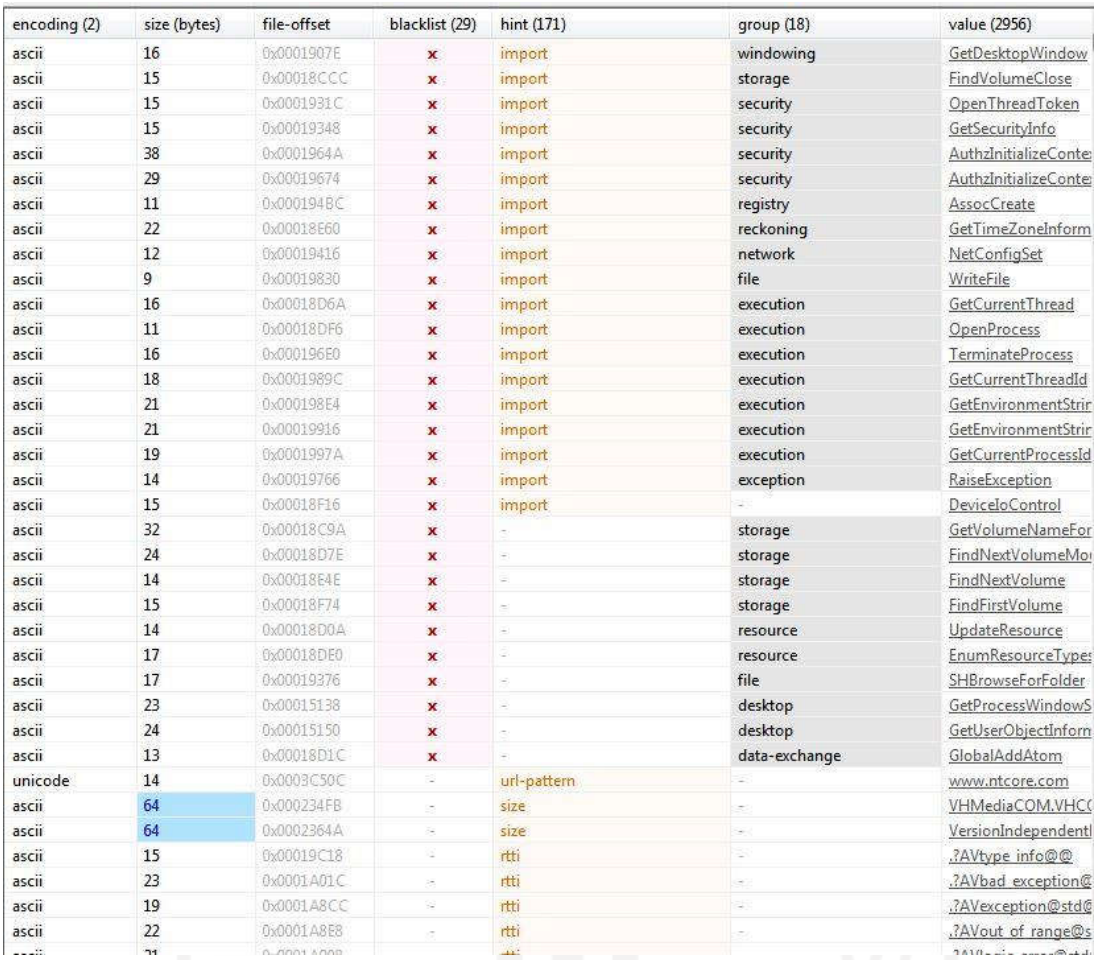
Tabel 4.9 adalah ringkasan dari *ransomware* Locky, dari informasi ini dapat dicari lagi informasi-informasi lainnya yang berhubungan dengan *ransomware* ini menggunakan nilai *hash* yang tertera pada tabel.

Tabel 4. 9 Ringkasan *Ransomware* Locky

tool	**Variabel**	**Value**
PeStudio	Nama	Vt2.bin
	MD5	d5fee0c6f1d0d730de259c64e6373a0c
	SHA-1	894f45f50454001bd21ad2713fefc15eb25b2b8b
	SHA256	0a2bc257eb1e266e2fd7c608bbb7e1f2ed34660c8ff21f32999fe49c6997329b
	Compiler-Stamp	Senin, 12 Desember 2016 10:20:06 2016
	Section	5
	Processor-32bit	True
	executable	True
EXEinfoPE	Paket	False
VirusTotal	Skor	55/67

Informasi yang tertera pada tabel didapatka dengan menggunakan tiga *tool*, yaitu PeStudio untuk menampilkan spesifikasi umum pada *ransomware* yang dianalisis seperti kapan *ransomware* di-*compile,* EXEinfoPE untuk mengetahui apakah *ransomware* dalam bentuk paket atau tidak dan VirusTotal untuk menunjukkan hasil kecocokan antara *ransomware* dengan database yang ada pada VirusTotal.

Berikut adalah tangkapan layar pada gambar 4.17 dari string yang digunakan oleh *ransomware* locky. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah string yang dicurigai oleh PeStudio saja.



encoding (2)	size (bytes)	file-offset	blacklist (29)	hint (171)	group (18)	value (2956)
ascii	16	0x0001907E	x	import	windowing	GetDesktopWindow
ascii	15	0x00018CCC	x	import	storage	FindVolumeClose
ascii	15	0x0001931C	x	import	security	OpenThreadToken
ascii	15	0x00019348	x	import	security	GetSecurityInfo
ascii	38	0x0001964A	x	import	security	AuthzInitializeConte
ascii	29	0x00019674	x	import	security	AuthzInitializeConte
ascii	11	0x000194BC	x	import	registry	AssocCreate
ascii	22	0x00018E60	x	import	reckoning	GetTimeZoneInform
ascii	12	0x00019416	x	import	network	NetConfigSet
ascii	9	0x00019830	x	import	file	WriteFile
ascii	16	0x00018D6A	x	import	execution	GetCurrentThread
ascii	11	0x00018DF6	x	import	execution	OpenProcess
ascii	16	0x000196E0	x	import	execution	TerminateProcess
ascii	18	0x0001989C	x	import	execution	GetCurrentThreadId
ascii	21	0x000198E4	x	import	execution	GetEnvironmentStrir
ascii	21	0x00019916	x	import	execution	GetEnvironmentStrir
ascii	19	0x0001997A	x	import	execution	GetCurrentProcessId
ascii	14	0x00019766	x	import	exception	RaiseException
ascii	15	0x00018F16	x	import	-	DeviceIoControl
ascii	32	0x00018C9A	x	-	storage	GetVolumeNameFor
ascii	24	0x00018D7E	x	-	storage	FindNextVolumeMo
ascii	14	0x00018E4E	x	-	storage	FindNextVolume
ascii	15	0x00018F74	x	-	storage	FindFirstVolume
ascii	14	0x00018D0A	x	-	resource	UpdateResource
ascii	17	0x00018DE0	x	-	resource	EnumResourceTypes
ascii	17	0x00019376	x	-	file	SHBrowseForFolder
ascii	23	0x00015138	x	-	desktop	GetProcessWindowS
ascii	24	0x00015150	x	-	desktop	GetUserObjectInforn
ascii	13	0x00018D1C	x	-	data-exchange	GlobalAddAtom
unicode	14	0x0003C50C	-	url-pattern	-	www.ntcore.com
ascii	64	0x000234FB	-	size	-	VHMediaCOM.VHC(
ascii	64	0x0002364A	-	size	-	VersionIndependentI
ascii	15	0x00019C18	-	rtti	-	.?AVtype_info@@
ascii	23	0x0001A01C	-	rtti	-	.?AVbad_exception@
ascii	19	0x0001A8CC	-	rtti	-	.?AVexception@std@
ascii	22	0x0001A8E8	-	rtti	-	.?AVout_of_range@s

Gambar 4. 17 String yang digunakan oleh locky

Dari gambar 4.17 diambil data yang menurut peneliti dan Pestudio, berbahaya dan akan dijelaskan lagi dalam bentuk tabel, yang dapat dilihat pada tabel 4.10.

Tabel 4. 10 String yang Digunakan Locky

Encoding	**Value**	**Penjelasan**
Ascii	GetDesktopWindow	Dapat menampilkan program sesuai dengan spesikasi monitor
ascii	OpenThreadToken	Membuka akses token
ascii	GetSecurityInfo	Mendapatkan info keamanan komputer
ascii	AuthzInitializeContextFromSid	Membuat *user* sebagai *user* Sid
ascii	AssocCreate	Mengembalikan pointer ke objek
ascii	GetTimeZoneInformation	Mendapatkan informasi tentang timezone
ascii	NetConfigSet	Untuk menkofigurasi jaringan

ascii	Write*File*	Memodifikasi *file*
ascii	GetCurrentThread	Mengambil pseudo handle
ascii	OpenProcess	Membuka lokal proses objek
ascii	TerminateProcess	Memberhentikan spesifik proses
ascii	GetEnvirontmenStrings	Mengambil variable untuk proses yang sedang berjalan
ascii	RaiseException	Membuat pengecualian dalam proses
ascii	DeviceIoControl	Untuk mengirimkan kode pada driver, membuat perangkat keras melakukan proses
ascii	GlobalAddAtom	Menambah karakter string

Tabel 4.10 adalah nilai *string* yang didapatkan dari *ransomware* Locky, *ransomware* Locky memiliki lebih dari 15 string, namun isi dari tabel 4.10 adalah beberapa *string* yang di-*blacklist* oleh aplikasi PeStudio.

Dari informasi tabel 4.10 dapat dilihat bahwa Locky ingin mengambil akses penuh pada komputer dan jaringan dengan membuat akses token, mendapat akses *user*, dan mengontrol jaringan.

Berikut adalah tangkapan layar pada gambar 4.18 dari library yang digunakan oleh *ransomware* locky. Seluruh library akan dijelaskan lebih rinci dalam tabel 4.11.

library (19)	blacklist (4)	type (1)	imports (192)	description
netapi32.dll	x	implicit	2	Net Win32 API DLL
winscard.dll	x	implicit	1	Microsoft Smart Card API
authz.dll	x	implicit	2	Authorization Framework
wlanapi.dll	x	implicit	2	Windows WLAN AutoConfig Client Side API DLL
kernel32.dll	-	implicit	92	Windows NT BASE API Client DLL
user32.dll	-	implicit	36	Multi-User Windows USER API Client DLL
gdi32.dll	-	implicit	14	GDI Client DLL
comdlg32.dll	-	implicit	2	Common Dialogs DLL
advapi32.dll	-	implicit	4	Advanced Windows 32 Base API
shell32.dll	-	implicit	2	Windows Shell Common Dll
ole32.dll	-	implicit	3	Microsoft OLE for Windows
oleaut32.dll	-	implicit	4	© Microsoft Corporation. All rights reserved.
odbc32.dll	-	implicit	3	ODBC Driver Manager
avifil32.dll	-	implicit	1	Microsoft AVI File support library
version.dll	-	implicit	1	Version Checking and File Installation Libraries
shlwapi.dll	-	implicit	3	Shell Light-weight Utility Library
gdiplus.dll	-	implicit	9	Microsoft GDI+
opengl32.dll	-	implicit	9	OpenGL Client DLL
glu32.dll	-	implicit	2	OpenGL Utility Library DLL

Gambar 4. 18 Library yang digunakan oleh locky

Tabel 4.11 adalah data yang diambil dari gambar 4.18 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* locky.

Tabel 4. 11 *Library* yang Digunakan Locky

Library	***Import***	**Penjelasan**
Kernel32.dll	92	Kernel32 sangat umum digunakan karena memiliki fungsi penting seperti akses, manipulasi memory. *File* dan hardware
Shell32.dll	2	*Library* yang berisi Windows Shell API yang digunakan untuk membuka web dan *file*
Advapi32.dll	4	Menyediakan akses ke komponen Windows seperti service manager dan *Registry*
Avifil32.dll	1	Untuk messuport AVI *file library*
Comdlg32.dll	6	Berisi modul untuk menampilkan eror dan sukses
*User*32.dll	36	Membuat program untuk menampilkan GUI
Gdi32.dll	14	Membuat program dapat menampilkan GDI
Gdiplus.dll	9	Memberikan fungsi lebih yang mendukung Gdi32.dll
Ole32.dll	3	Untuk menjalankan object linking dan embedding
Obdc32.dll	3	Untuk mengakses OBDC (open database connectivity)
Oleaut32.dll	4	Menghandle informasi yang di buat oleh aplikasi lain untuk diproses
Opengl32.dll	9	Memiliki fungsi untuk berkomunikasi dengan GPU
Shlwapi.dll	3	Mensimplifikasi proses dan mengambil informasi yang ada di *registry*
Version.dll	1	Untuk mengecek versi *library* dan *file* yang dapat diakses
Authz.dll	2	Untuk mengecek framework yang digunakan
Netapi32.dll	2	Untuk mengecek dan mengatur network interface
Winscard.dll	1	Untuk dapat mengaktifkan fungsi smart card dan smart card reader
Wlanapi.dll	2	Untuk dapat mengoneksi network atau komputer secara wireless

Tabel 4.11 adalah tabel dari *library* yang digunakan oleh Locky, dari tabel ini juga dapat melihat perilaku *ransomware* melalui *library* yang digunakan seperti mengakses *registry* serta mengakses dan memanipulasi memory *file* dan hardware. *Ransomware* Locky menggunakan *library* terbanyak dibandingkan *ransomware* lainnya yang telah dianalisis.

Yang membedakan *ransomware* Locky dengan *ransomware* lain yang telah dianalisis ialah cara *ransomware* Locky selalu melakukan pengecekan sistem saat menginfeksi, seperti menggunakan *library* version.dll, authz.dll, dan netapi32.dll.

Ransomware Locky juga memiliki *library* yang bertujuan untuk dapat mengaktifkan fitur *smartcard* (Winscard.dll) dan *library* untuk dapat berkomunikasi *secara* wireless antar komputer (Wlanapi.dll).

Berikut adalah tangkapan layar pada gambar 4.19 dari import yang digunakan oleh *ransomware* locky. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah import yang dicurigai oleh PeStudio saja.

name (192)	blacklist (28)	group (18)	ordinal (7)	library (19)
GetDesktopWindow	x	windowing	-	user32.dll
FindFirstVolumeW	x	storage	-	kernel32.dll
FindNextVolumeW	x	storage	-	kernel32.dll
FindNextVolumeMountPointA	x	storage	-	kernel32.dll
FindVolumeClose	x	storage	-	kernel32.dll
GetVolumeNameForVolume...	x	storage	-	kernel32.dll
GetSecurityInfo	x	security	-	advapi32.dll
OpenThreadToken	x	security	-	advapi32.dll
AuthzInitializeContextFromSid	x	security	-	authz.dll
AuthzInitializeContextFromA...	x	security	-	authz.dll
EnumResourceTypesA	x	resource	-	kernel32.dll
UpdateResourceA	x	resource	-	kernel32.dll
AssocCreate	x	registry	-	shlwapi.dll
GetTimeZoneInformation	x	reckoning	-	kernel32.dll
NetConfigSet	x	network	-	netapi32.dll
WriteFile	x	file	-	kernel32.dll
SHBrowseForFolderA	x	file	-	shell32.dll
GetCurrentProcessId	x	execution	-	kernel32.dll
GetEnvironmentStrings	x	execution	-	kernel32.dll
GetCurrentThreadId	x	execution	-	kernel32.dll
OpenProcess	x	execution	-	kernel32.dll
GetCurrentThread	x	execution	-	kernel32.dll
TerminateProcess	x	execution	-	kernel32.dll
GetEnvironmentStringsW	x	execution	-	kernel32.dll
RaiseException	x	exception	-	kernel32.dll

Gambar 4. 19 Impor yang digunakan oleh locky

Tabel 4.12 adalah data yang diambil dari gambar 4.19 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* locky.

Tabel 4. 12 *Import* yang Digunakan Locky

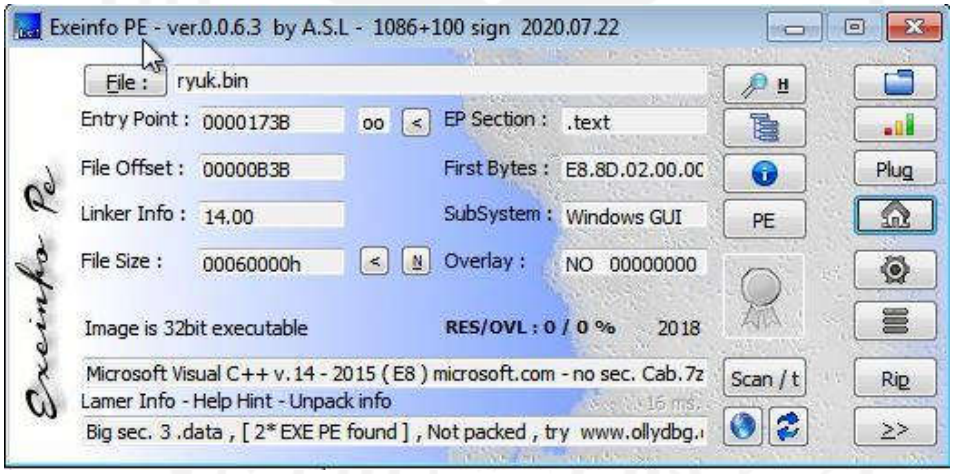
Nama	**grup**	**library**	**Penjelasan**
GetDekstopWindow	Windowing	*User*32.dll	Dapat menampilkan program sesuai dengan spesifikasi monitor
FindFirstVolumeW	Storage	Kernel32.dll	Melakukan pemindaian pada komputer
FindNextVolumeW	Storage	Kernel32.dll	Untuk melakukan pencarian pada driver selanjutnya

FindNextVolumeMountPointA	Storage	Kernel32.dll	Melakukan pencarian selanjutnya pada mounted folder
FindVolumeClose	Storage	Kernel32.dll	Menutup spesifik driver yang telah dilakukan pemidaian
GetSecurityInfo	Security	Advapi32.dll	Mendapatkan info keamanan komputer
AssocCreate	*Registry*	Shlwapi.dll	Mengembalikan pointer ke objek
NetConfigSet	Network	Netapi32.dll	Untuk menkofigurasi jaringan
GlobalAddAtom	Data-exchange	Kernel32.dll	Menambah karakter string
DeviceIoControl	-	Kernel32.dll	Mengrimkan kode pada spesifik driver

Import yang digunakan oleh Locky berfokus pada driver komputer, *hard disk*, jaringan, dan mengontrol *hardware*. Locky juga menjaga tampilan *ransomware* pada monitor dan dapat memberikan perintah pada IO untuk melakukan perintah tertentu.

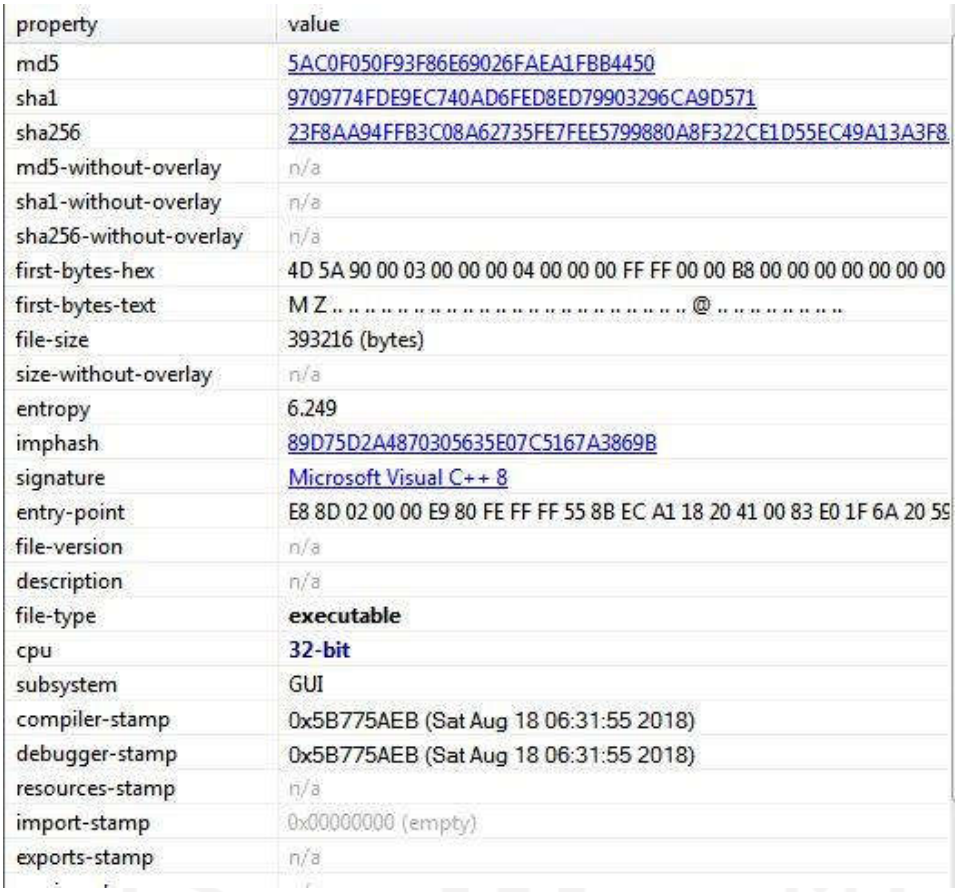
4.1.4 Ryuk

Gambar 4.20 adalah hasil dari tool ExeinfoPe sebagai pengecek apakah sampel berbentuk paket atau tidak setelah dipastikan bahwa sampel tidak dalam bentuk paket maka dengan begitu analisis statis dapat dilakukan.



Gambar 4. 20 Pengecekan paket pada *file* ryuk

Gambar 4.21 adalah gambar dari hasil tool PeStudio penjelasan singkat tentang struktur dari *file ransomware*. Seperti nilai hash, compiler-stamp, tipe *file* dan seperti yang dapat dilihat dari gambar, data ini nantinya akan dipersingkat lagi dalam bentuk tabel, yaitu tabel 4.13.



property	value
md5	5AC0F050F93F86E69026FAEA1FBB4450
sha1	9709774FDE9EC740AD6FED8ED79903296CA9D571
sha256	23F8AA94FFB3C08A62735FE7FEE5799880A8F322CE1D55EC49A13A3F8
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	393216 (bytes)
size-without-overlay	n/a
entropy	6.249
imphash	89D75D2A4870305635E07C5167A3869B
signature	Microsoft Visual C++ 8
entry-point	E8 8D 02 00 00 E9 80 FE FF FF 55 8B EC A1 18 20 41 00 83 E0 1F 6A 20 59
file-version	n/a
description	n/a
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5B775AEB (Sat Aug 18 06:31:55 2018)
debugger-stamp	0x5B775AEB (Sat Aug 18 06:31:55 2018)
resources-stamp	n/a
import-stamp	0x00000000 (empty)
exports-stamp	n/a

Gambar 4. 21 Ringkasan struktur *file* ryuk

Tabel 4.13 adalah ringkasan dari *ransomware* Ryuk, dari informasi ini dapat dicari lagi informasi-informasi lainnya yang berhubungan dengan *ransomware* ini dengan menggunakan nilai *hash* yang tertera pada tabel 4.13.

Tabel 4. 13 Ringkasan *Ransomware* Ryuk

Tool	**Variabel**	**Value**
PeStudio	Nama	Ryuk.bin
	MD5	5AC0F050F93F86E69026FAEA1FBB4450
	SHA-1	9709774FDE9EC740AD6FED8ED79903296CA9D571
	SHA256	23F8AA94FFB3C08A62735FE7FEE5799880A8F322CE1D55EC 49A13A3F85312DB2
	Compiler-Stamp	Jumat, 17 Agustus 2018 16:31:55
	Section	5
	Processor-32bit	True
	executable	True

EXEinfoPE	Paket	False
VirusTotal	Skor	62/69

Informasi yang tertera pada tabel didapatka dengan menggunakan tiga *tools*, yaitu PeStudio untuk menampilkan spesifikasi umum pada *ransomware* yang dianalisis seperti kapan *ransomware* di-*compile,* EXEinfoPE untuk mengetahui apakah *ransomware* dalam bentuk paket atau tidak dan VirusTotal untuk menunjukkan hasil kecocokan antara *ransomware* dengan database yang ada pada VirusTotal.

Berikut adalah tangkapan layar pada gambar () dari string yang digunakan oleh *ransomware* ryuk. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah string yang dicurigai oleh PeStudio saja.

Berikut adalah tangkapan layar pada gambar () dari string yang digunakan oleh *ransomware* ryuk. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah string yang dicurigai oleh PeStudio saja.

encoding (2)	size (bytes)	file-offset	blacklist (62)	hint (649)	group (12)	value (5152)
ascii	9	0x0001062E	x	import	file	WriteFile
ascii	9	0x0003301A	x	import	file	WriteFile
ascii	9	0x00057E0E	x	import	file	WriteFile
ascii	19	0x00010718	x	import	execution	GetCurrentProcessId
ascii	18	0x0001072E	x	import	execution	GetCurrentThreadId
ascii	16	0x00010804	x	import	execution	TerminateProcess
ascii	16	0x00032E32	x	import	execution	TerminateProcess
ascii	19	0x00032E7C	x	import	execution	GetCurrentProcessId
ascii	18	0x00032E92	x	import	execution	GetCurrentThreadId
ascii	19	0x00057C08	x	import	execution	GetCurrentProcessId
ascii	18	0x00057C1E	x	import	execution	GetCurrentThreadId
ascii	16	0x00057CF4	x	import	execution	TerminateProcess
ascii	14	0x00010B18	x	import	exception	RaiseException
ascii	14	0x00032F20	x	import	exception	RaiseException
ascii	14	0x00057D08	x	import	exception	RaiseException
ascii	17	0x00010B2A	x	import	cryptography	SystemFunction036
ascii	17	0x000331C6	x	import	cryptography	SystemFunction036
ascii	17	0x00057FCA	x	import	cryptography	SystemFunction036
ascii	16	0x00032CFA	x	-	security	LookupAccountSid
ascii	15	0x00032D0E	x	-	security	OpenThreadToken
ascii	16	0x00032D20	x	-	security	OpenProcessToken
ascii	21	0x00032D46	x	-	security	AdjustTokenPrivileg
ascii	20	0x00032D5E	x	-	security	LookupPrivilegeVal
ascii	16	0x00057B32	x	-	security	LookupAccountSid
ascii	15	0x00057B46	x	-	security	OpenThreadToken
ascii	16	0x00057B58	x	-	security	OpenProcessToken
ascii	21	0x00057B7E	x	-	security	AdjustTokenPrivileg
ascii	20	0x00057B96	x	-	security	LookupPrivilegeVal
ascii	18	0x00032ADA	x	-	memory	WriteProcessMemo
ascii	18	0x00057912	x	-	memory	WriteProcessMemo
ascii	15	0x0001099C	x	-	file	FindFirstFileEx
ascii	12	0x000109B0	x	-	file	FindNextFile
ascii	10	0x0002EEB8	x	-	file	MoveFileEx
ascii	10	0x00032C00	x	-	file	DeleteFile
ascii	15	0x0003309C	x	-	file	FindFirstFileEx
ascii	12	0x000330B0	x	-	file	FindNextFile

Gambar 4. 22 String yang digunakan oleh ryuk

Dari gambar 4.22 diambil data yang menurut peneliti dan Pestudio, berbahaya dan akan dijelaskan lagi dalam bentuk tabel, yang dapat dilihat pada tabel 4.14.

Tabel 4. 14 String yang Digunakan Ryuk

Encoding	Value	Penjelasan
ascii	Write*File*	Untuk memodifikasi konten dari *file*
ascii	GetCurrentProcessId	Mendapatkan informasi pada proses id untuk memanggil proses
ascii	Sleep	Untuk menunda proses eksekusi agar mengganggu proses analisis
ascii	Free*Library*	Untuk menjalankan *Library* yang telah dimuat
ascii	Load*Library*	Memuat modul spesifik kedalam memori yang tersedia untuk memanggil proses
ascii	GetSystemDefaultLangID	Mengecek bahasa yang digunakan sistem
ascii	CreateRemoteThread	Untuk membuat remote proses untuk meninjeksi kode
ascii	WriteProcessMemory	Memodifikasi data ke remote proses untuk melakukan proses injeksi

Tabel 4.14 adalah nilai *string* yang didapatkan dari *ransomware* Ryuk. *Ransomware* Locky memiliki lebih dari 8 string, namun isi dari tabel 4.14 adalah beberapa *string* yang diblacklist oleh aplikasi PeStudio.

Berdasarkan data diatas Ryuk dapat mengunduh dan membuka *library* baru agar dapat berjalan pada komputer. Sementara itu *string sleep* berguna untuk mengganggu proses analisis sehingga memunculkan data yang tidak akurat.

Berikut adalah tangkapan layar pada gambar 4.23 dari library yang digunakan oleh *ransomware* ryuk. Seluruh library akan dijelaskan lebih rinci dalam tabel 4.15.

library (3)	blacklist (0)	type (1)	imports (70)	description
kernel32.dll	-	implicit	68	Windows NT BASE API
shell32.dll	-	implicit	1	Windows Shell Comm
advapi32.dll	-	implicit	1	Advanced Windows 32

Gambar 4. 23 Library yang digunakan oleh ryuk

Tabel 4.15 adalah data yang diambil dari gambar 4.23 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* ryuk.

Tabel 4. 15 *Import* yang Digunakan Ryuk

Library	***Import***	**Penjelasan**
Kernel32.dll	68	Kernel32 sangat umum digunakan karena memiliki fungsi penting seperti akses, manipulasi memori. *File* dan hardware
Shell32.dll	1	*Library* yang berisi Windows Shell API yang digunakan untuk membuka web dan *file*
Advapi32.dll	1	Menyediakan akses ke komponen Windows seperti service manager dan *Registry*

Tabel 4.15 adalah tabel dari *library* yang digunakan oleh Ryuk. Dari tabel ini dapat dilihat perilaku *ransomware* melalui *library* yang digunakan, seperti mengakses *registry*, mengakses dan memanipulasi memori *file* dan hardware.

Berikut adalah tangkapan layar pada gambar 4.24 dari import yang digunakan oleh *ransomware* ryuk. Nantinya data ini akan dijadikan dalam bentuk tabel, namun yang dimasukkan kedalam tabel adalah import yang dicurigai oleh PeStudio saja.

name (70)	blacklist (11)	group (10)	ordinal (0)	library (3)
WriteFile	x	file	-	kernel32.dll
FindFirstFileExA	x	file	-	kernel32.dll
FindNextFileA	x	file	-	kernel32.dll
GetCurrentProcessId				
GetCurrentThreadId	x	execution	-	kernel32.dll
TerminateProcess	x	execution	-	kernel32.dll
GetEnvironmentStringsW	x	execution	-	kernel32.dll
ShellExecuteW	x	execution	-	shell32.dll
RaiseException	x	exception	-	kernel32.dll
GetModuleHandleExW	x	dynamic-library	-	kernel32.dll
SystemFunction036	x	cryptography	-	advapi32.dll
InitializeSListHead	-	synchronization	-	kernel32.dll
InitializeCriticalSectionAndS...	-	synchronization	-	kernel32.dll
EnterCriticalSection	-	synchronization	-	kernel32.dll
LeaveCriticalSection	-	synchronization	-	kernel32.dll
DeleteCriticalSection	-	synchronization	-	kernel32.dll
GetVersionExW	-	reckoning	-	kernel32.dll
GetWindowsDirectoryW	-	reckoning	-	kernel32.dll
GetTickCount	-	reckoning	-	kernel32.dll
QueryPerformanceCounter	-	reckoning	-	kernel32.dll
IsDebuggerPresent	-	reckoning	-	kernel32.dll
GetStartupInfoW	-	reckoning	-	kernel32.dll
IsProcessorFeaturePresent	-	reckoning	-	kernel32.dll
HeapFree	-	memory	-	kernel32.dll
HeapAlloc	-	memory	-	kernel32.dll
GetStringTypeW	-	memory	-	kernel32.dll
GetProcessHeap	-	memory	-	kernel32.dll
HeapSize	-	memory	-	kernel32.dll
HeapReAlloc	-	memory	-	kernel32.dll
CreateFileW	-	file	-	kernel32.dll
GetSystemTimeAsFileTime	-	file	-	kernel32.dll

Gambar 4. 24 Import yang digunakan oleh ryuk

Tabel 4.16 adalah data yang diambil dari gambar 4.24 dan dijelaskan kembali dalam bentuk tabel untuk setiap library yang digunakan oleh *ransomware* ryuk.

Tabel 4. 16 *Import* Yang Digunakan Ryuk

Nama	**grup**	**library**	**Penjelasan**
Write*File*	*File*	Kernel32.dll	Memodifikasi *file*
FindFirst*File*	*File*	Kernel32.dll	Memindai *file* pertama
FindNext*File*	*File*	Kernel32.dll	Fungsi untuk melanjutkan pemindaian
GetCurrentProcessId	Execution	Kernel32.dll	Mendapatkan informasi pada proses id untuk memanggil proses
GetCurrentThreadId	Execution	Kernel32.dll	Mendapatkan informasi pada thread id untuk memanggi thread
TerminateProcess	Execution	Kernel32.dll	Melakukan pemberhentian pada applikasi
GetEnvironmentString	Execution	Kernel32.dll	Mengambil variabel untuk proses yang sedang berjalan
ShellExecute	Execution	Shell32.dll	Untuk menjalankan program
RaiseException	Execution	Kernel32.dll	Membuat pengecualian dalam proses
GetModuleHandleEx	Dynamic-*library*	Kernel32.dll	Mengambil nilai modul pada proses yang dijalankan
SystemFunction036	Cryptography	Advapi32.dll	Untuk mengenerasi angka acak pseudo

Pada tabel 4.16 *import* Ryuk dapat memodifikasi, memindai dan mencari *file* selanjutnya. *Import* yang digunakan pada Ryuk adalah untuk memerintah komputer agar Ryuk mendapatkan akses-akses pada komputer yang dapat membantu menginfeksi komputer.

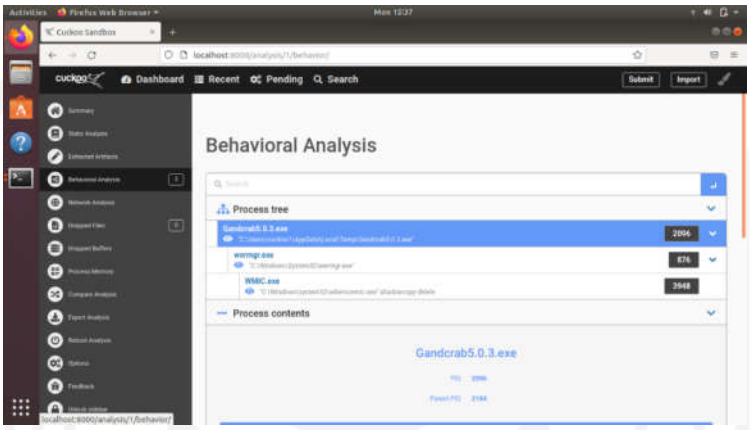
4.2 Analisis Dinamis

Setelah proses membangun lingkungan untuk proses analisis dan mengonfigurasi Cuckoo dan network yang dibutuhkan Cuckoo agar dapat menjalankan analisis dengan baik, maka *ransomware* sudah siap dianalisis. Telah disiapkan beberapa sampel *ransomware* yang sudah siap untuk dianalisis di antara lain ialah:

1. Gandcrab: Gandcrab5.03.bin
2. Locky: vt2.bin
3. Ryuk: Ryuk.bin
4. GoldenEye: GoldenEye.bin

4.2.1 Gandcrab

Pada gambar 4.25 adalah hasil analisis *ransomware* Gandcrab. Gandcrab memiliki 3 pohon proses dimana proses dilakukan satu persatu mulai dari Gandcrab 5.03 dan lalu dilanjutkan wermgr.exe dan wmic.exe. Setiap pohon proses memiliki proses konten masing-masing dalam melakukan infeksi.



Gambar 4. 25 Hasil analisis dinamis gandcrab

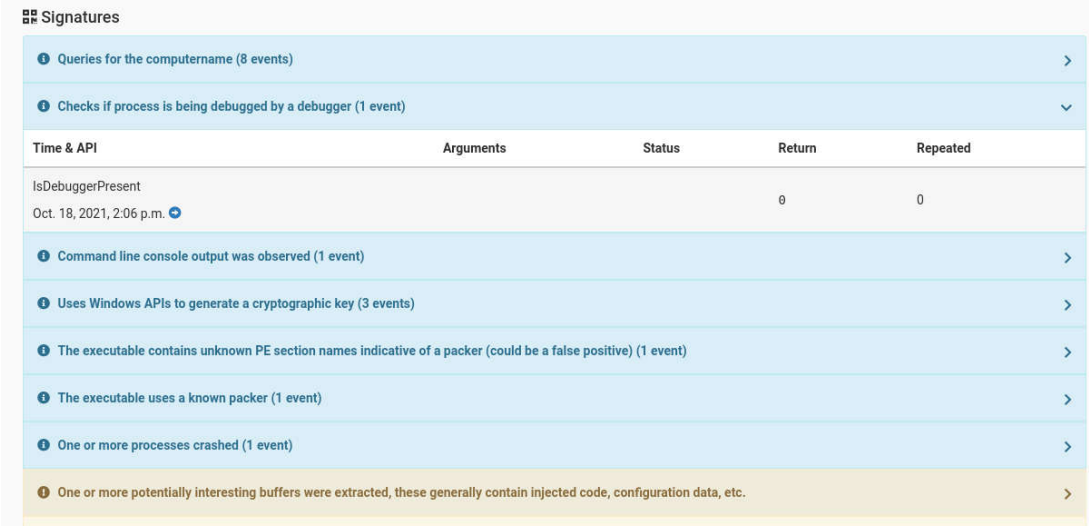
Berikut adalah isi dari proses *tree* pada Gandcrab. Proses *tree* terbagi lagi menjadi 9 bagian untuk memecah banyak proses yang dilakukan oleh program yang dianalisis. Berikut adalah laporan akhir pada hasil analisis dinamis.

Gambar 4. 26 Isi proses dari gandcrab

Setiap proses mengisi bagian yang berbeda dikarenakan setiap proses memiliki tugas masing-masing dalam melakukan infeksi. Misalnya isi proses dari Gandcrab.exe hanya

memiliki isi pada bagian proses yang memiliki perintah protectvirtuallmemory dan CreateProcessInternal.

Dapat dilihat dari gambar 4.27, banyak yang terjadi saat melakukan analisis dinamis salah satunya adalah pengecekan apakah proses yang berjalan sedang di-*debug,* dan pengecekan nama pada komputer, lalu gandcrab membuat API untuk menggenerasi kunci cypto.

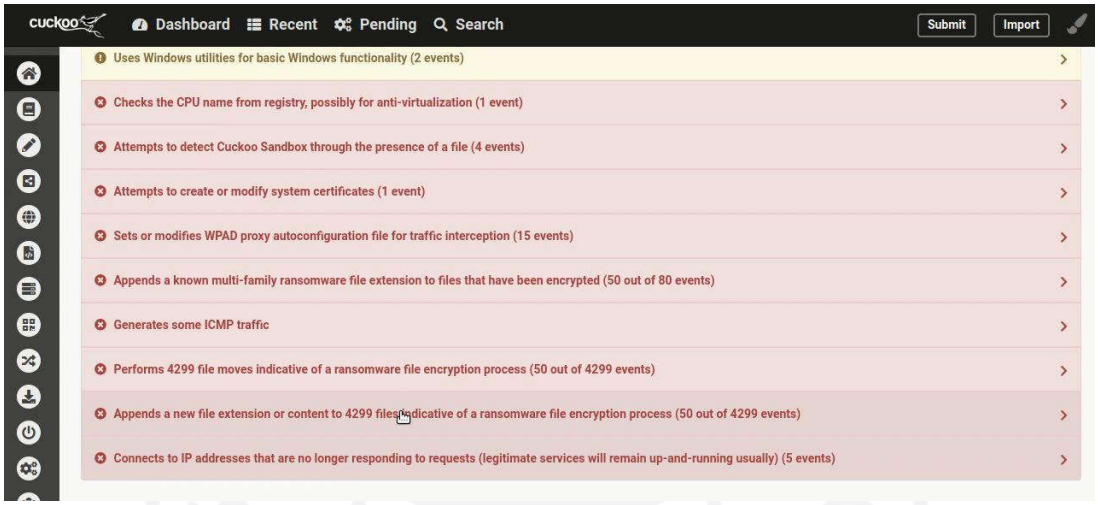


Gambar 4. 27 *Signature* gandcrab

Pada proses selanjutnya gandcrab melakukan injeksi kode, dan mengecek jumlah memori pada perangkat keras, untuk mengecek apakah ransomware berjalan pada mesin virtual, gandcrab juga melakukan permintaan pada HTTP dan masih banyak lagi terlihat seperti pada gambar 4.28.

Gambar 4. 28 *Signature* gandcrab

Signature terakhir adalah *signature* yang dianggap berbahaya, dapat dilihat pada gambar 4.29 gandcrab melakukan pengecekan CPU melalui *registry*, pengecekan cuckoo sandbox, membuat ICMP, dan mengubah *extension* file sehingga file tidak dapat lagi diakses.

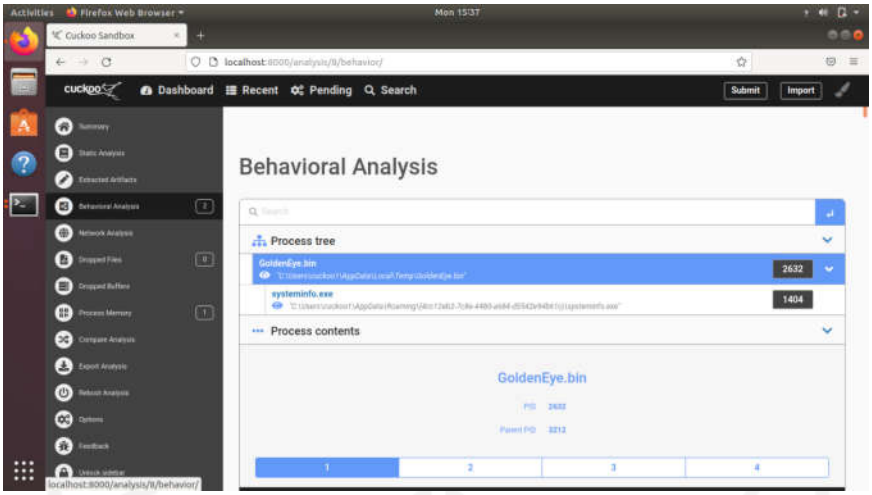


Gambar 4. 29 *Signature* gandcrab

Gambar 4.27, 4.28, dan 4.29 adalah *signature* dari *ransomware* gandcrab yang diberikan oleh cuckoo sehingga dapat mengerti apa saja yang dikerjakan oleh *ransomware*, tidak semua API yang dijalankan oleh *ransomware* akan ditampilkan sebagai *signature*, namun *signature* memudahkan untuk mengerti apa saja yang dilakukan oleh *ransomware*.

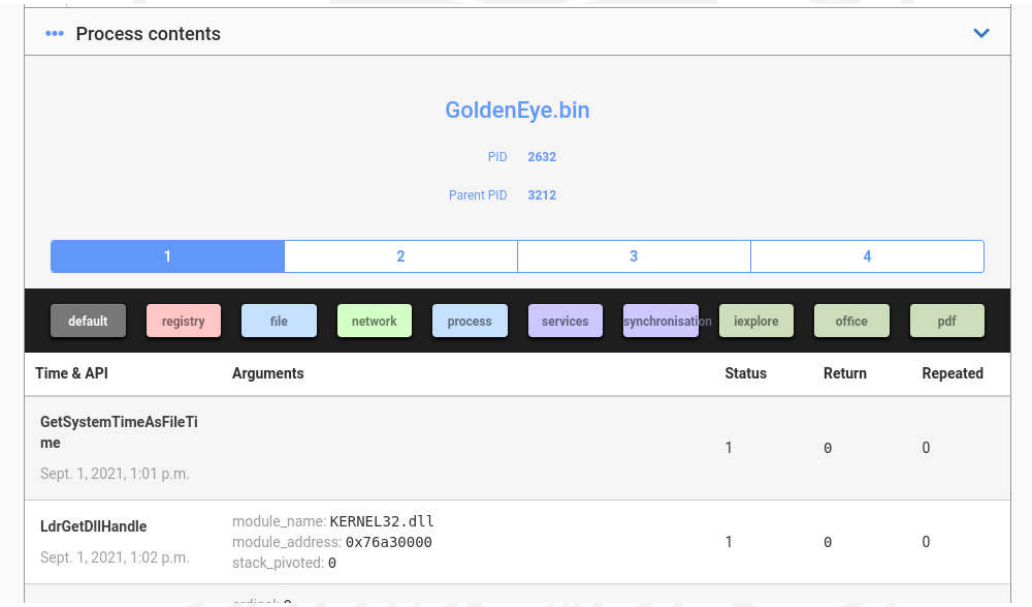
4.2.2 GoldenEye

Pada gambar 4.30 adalah hasil analisis *ransomware* GoldenEye, GoldenEye memiliki 2 pohon proses yang di mana proses dilakukan satu persatu mulai dari GoldenEye.bin dan lalu dilanjutkan systeminfo.exe. Setiap pohon proses memiliki proses konten masing-masing dalam melakukan infeksi.



Gambar 4. 30 Hasil analisis dinamis goldeneye

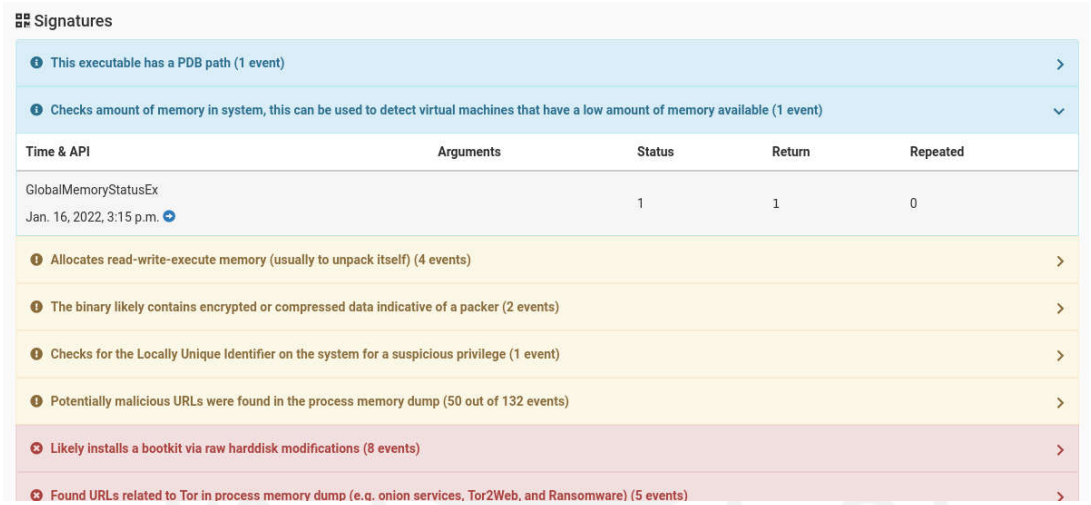
Berikut adalah isi dari proses *tree* pada GoldenEye dan proses *tree* tersebut terbagi lagi menjadi 9 bagian untuk memecah banyak proses yang dilakukan oleh program yang dianalisis. Berikut adalah laporan akhir pada hasil analisis dinamis.



Gambar 4. 31 Isi proses goldeneye

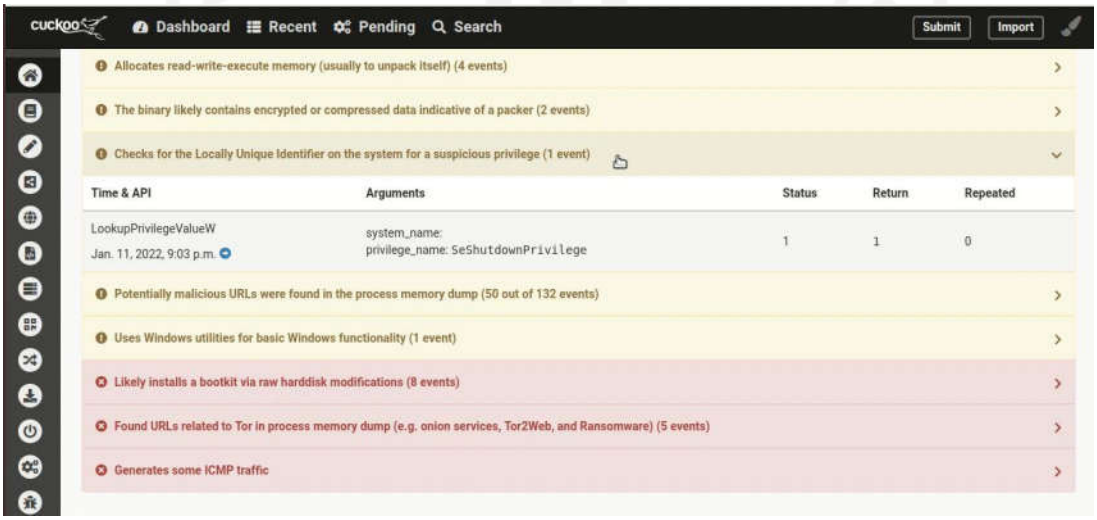
Setiap proses mengisi bagian yang berbeda dikarenakan setiap proses memiliki tugas masing-masing dalam melakukan infeksi. Isi proses dari GoldenEye.bin memiliki 4 bagian yaitu pada bagian *registry*, *file*, proses, dan sinkronisasi. Hal ini berarti proses GoldenEye.bin tidak hanya untuk memulai proses infeksi namun juga untuk mencari *file* dan mengubah *registry*.

Dapat dilihat dari gambar 4.32, banyak yang terjadi saat melakukan analisis dinamis salah satunya adalah pengecekan apakah memori untuk mengetahui apakah ransomware berjalan pada mesin virtual.



Gambar 4. 32 *Signature* Goldeneye

Pada gambar 4.33 goldeneye melakukan pengecekan pada sistem untuk melihat privilege pada komputer, mengambil alih windows untuk menjalankan proses pada komputer, lalu goldeneye membuat bootkit pada hard disk sehingga ransomware dan membuat jaringan ICMP.



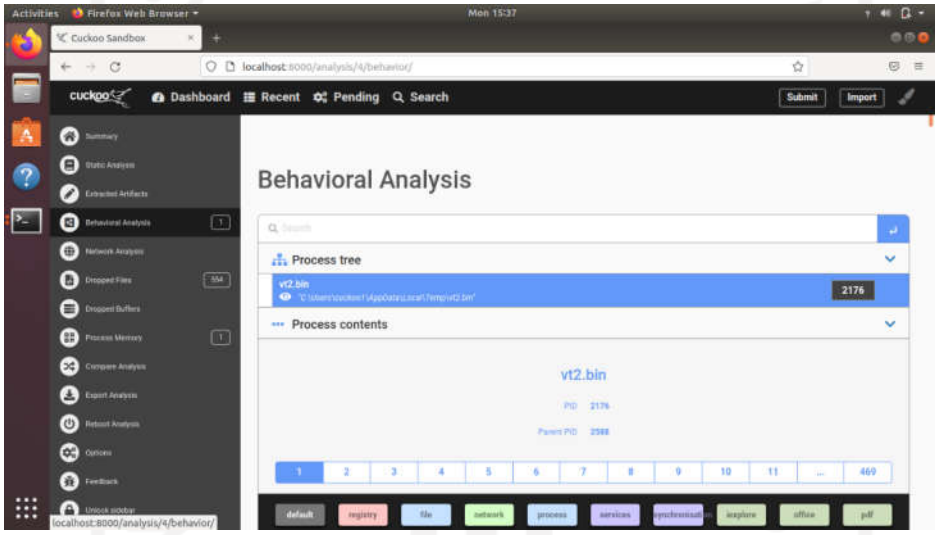
Gambar 4. 33 *Signature* goldeneye

Gambar 4.32 dan 4,33 adalah *signature* dari *ransomware* goldeneye yang diberikan oleh cuckoo sehingga dapat mengerti apa saja yang dikerjakan oleh *ransomware*, tidak semua API

yang dijalankan oleh *ransomware* akan ditampilkan sebagai *signature*, namun *signature* memudahkan untuk mengerti apa saja yang dilakukan oleh *ransomware*.

4.2.3 Locky

Pada gambar 4.34 adalah hasil analisis *ransomware* Locky, Locky hanya memiliki 1 pohon proses yang dimana proses dimulai oleh vt2.bin, tetapi *ransomware* Locky mengunduh sebanyak 554 *file* pada komputer. Hal ini kemungkinan merupakan tujuan dari mengonfigurasi jaringan yang didapat dari analisis statis.



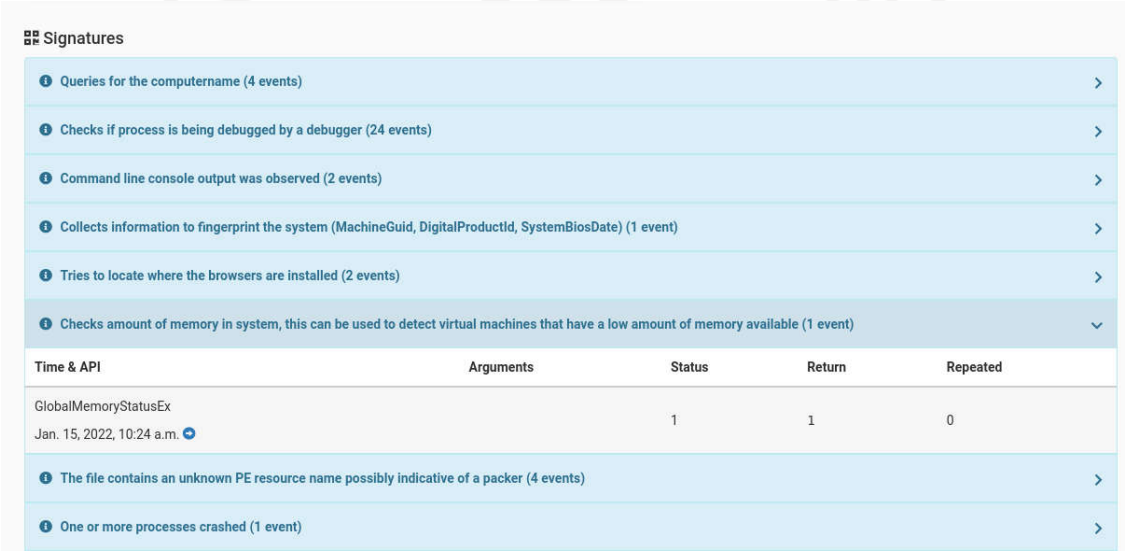
Gambar 4. 34 Hasil analisis dinamis locky

Berikut adalah isi dari proses *tree* pada Locky, dan proses tree tersebut terbagi lagi menjadi 9 bagian untuk memecah banyak proses yang dilakukan oleh program yang dianalisis, ini adalah laporan akhir pada hasil analisis dinamis.

Gambar 4. 35 Isi proses locky

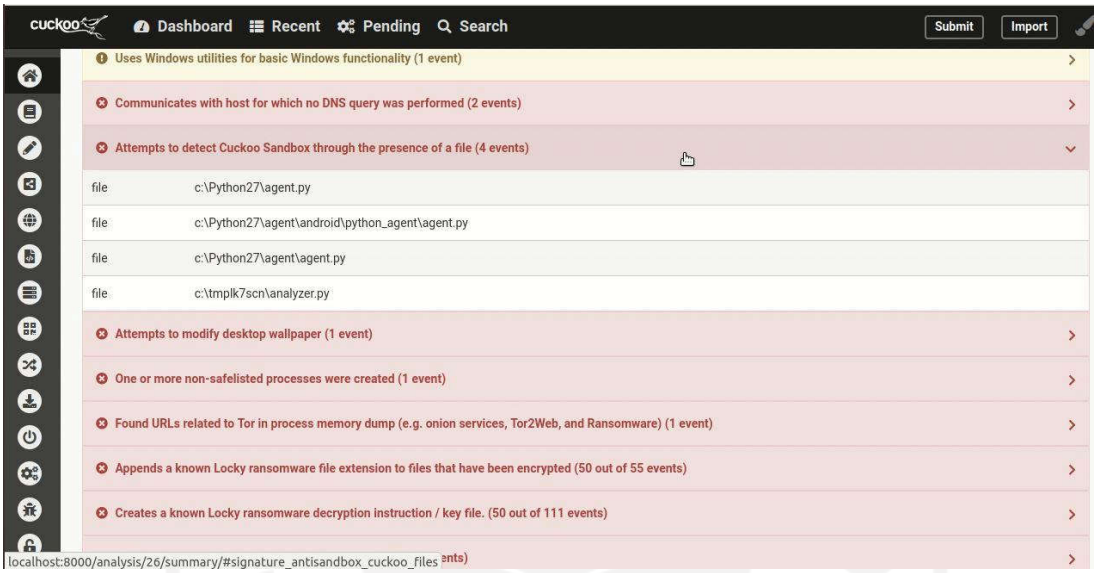
Namun vt.bin pada program Locky mencakup semua kebutuhan untuk menginfeksi komputer, seperti *registry*, *file*, jaringan, proses, servis, dan sinkronisasi.

Dapat dilihat dari gambar 4.36, banyak yang terjadi saat melakukan analisis dinamis salah satunya adalah pengecekan apakah proses yang berjalan sedang di-*debug,* pengecekan memori komputer untuk menghindari jika ransomware berjalan pada mesin virtual.



Gambar 4. 36 *Signature* locky

Pada proses selanjunya locky melakukan pengecekan pada cuckoo sandbox, berhubungan dengan DNS host, mengubah *wallpaper*, dan mengubah file yang ada pada komputer sehingga tidak dapat diakses.

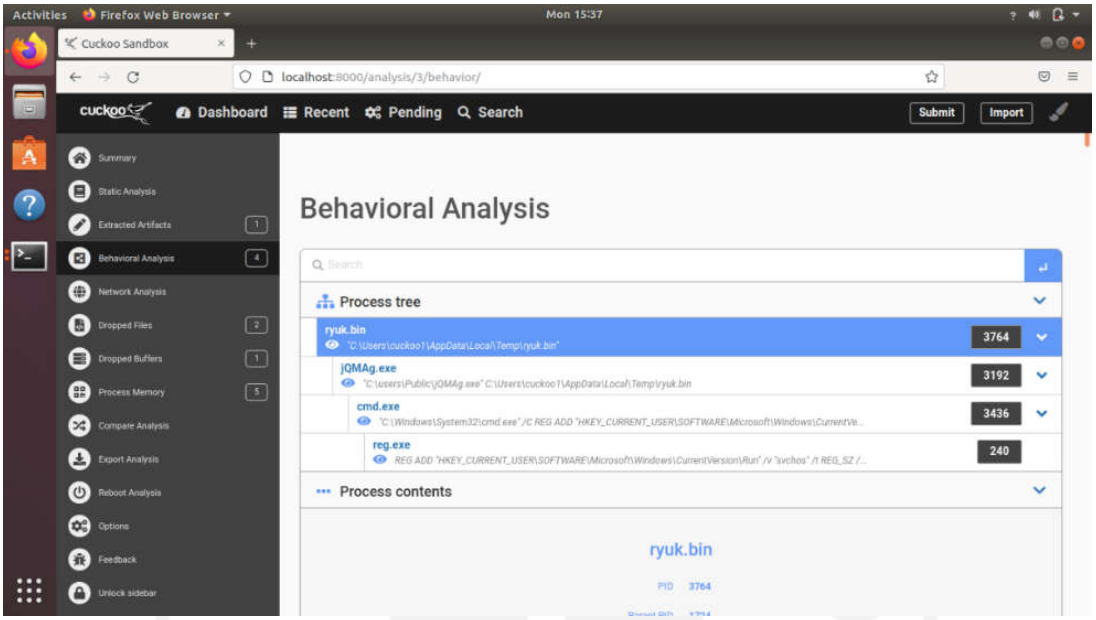


Gambar 4. 37 *Signature* locky

Gambar 4.36 dan 4.37 adalah *signature* dari *ransomware* locky yang diberikan oleh cuckoo sehingga dapat mengerti apa saja yang dikerjakan oleh *ransomware*, tidak semua API yang dijalankan oleh *ransomware* akan ditampilkan sebagai *signature*, namun *signature* memudahkan untuk mengerti apa saja yang dilakukan oleh *ransomware*.

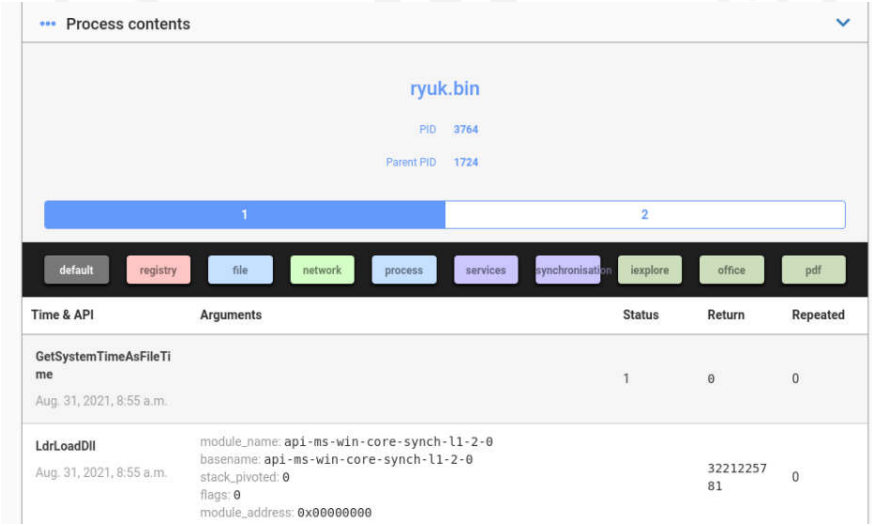
4.2.4 Ryuk

Pada gambar 4.38 adalah hasil analisis *ransomware* Ryuk, Ryuk memiliki 4 pohon proses yang dimana proses dilakukan satu persatu mulai dari Ryuk.bin dan lalu dilanjutkan jqmag.exe, cmd.exe, dan yang terakhir reg.exe.Setiap pohon proses memiliki proses konten masing-masing dalam melakukan infeksi.



Gambar 4. 38 Hasil analisis ryuk

Berikut adalah isi dari proses *tree* pada Ryuk dan proses *tree* tersebut terbagi lagi menjadi 9 bagian untuk memecah banyak proses yang dilakukan oleh program yang dianalisis, ini adalah laporan akhir pada hasil analisis dinamis.

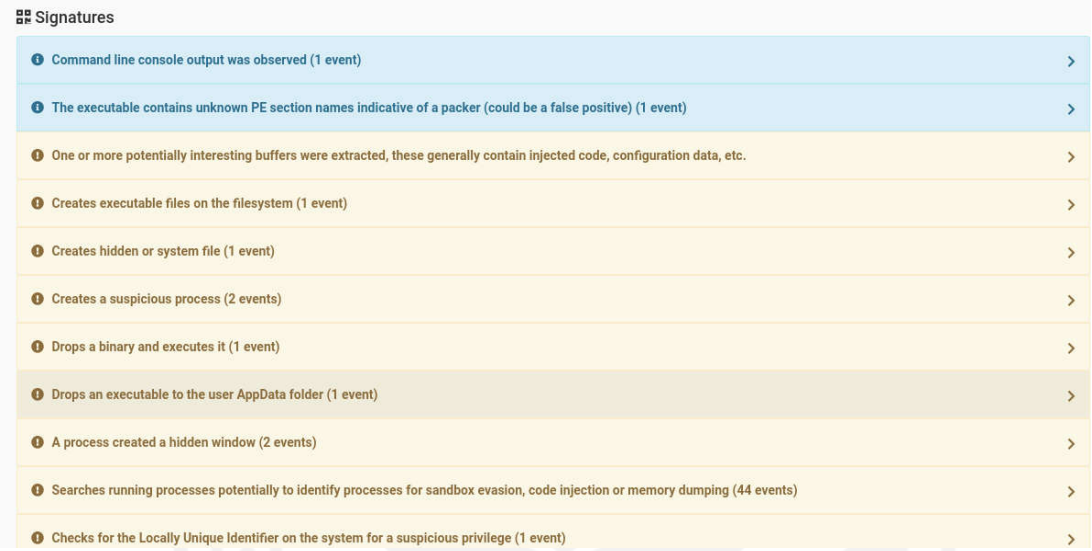


Gambar 4. 39 Isi proses ryuk

Setiap proses mengisi bagian yang berbeda dikarenakan setiap proses memiliki tugas masing-masing dalam melakukan infeksi. Isi proses dari Ryuk.bin memiliki *registry*, *file*,

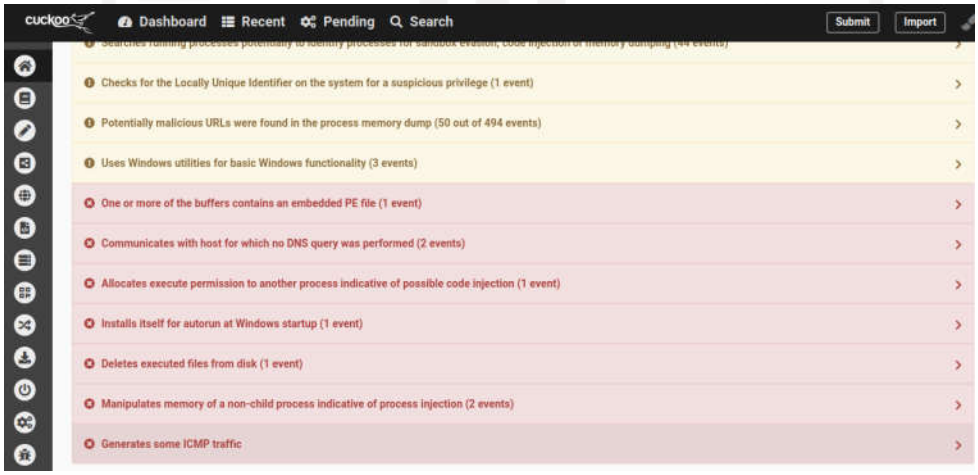
proses, dan sinkronisasi. Hal ini berarti selain memulai proses infeksi, Ryuk.bin juga melakukan perubahan *registry*, pemindaian *file*, dan sinkronisasi pada komputer.

Dapat dilihat dari gambar 4.40, banyak yang terjadi saat melakukan analisis dinamis salah satunya adalah pengecekan apakah sedang dilakukannya pembuangan memori, injeksi kode, dan sandbox, dan mengecek *user* jika ada yang mencurigakan.



Gambar 4. 40 *Signature* ryuk

Pada proses selanjutnya pada gambar 4.41 terjadi porses penghapusan file executeable pada komputer, melakukan komunikasi pada host, memanipulasi memori, memasang auto-run pada komputer, dan memasang jaringan ICMP.

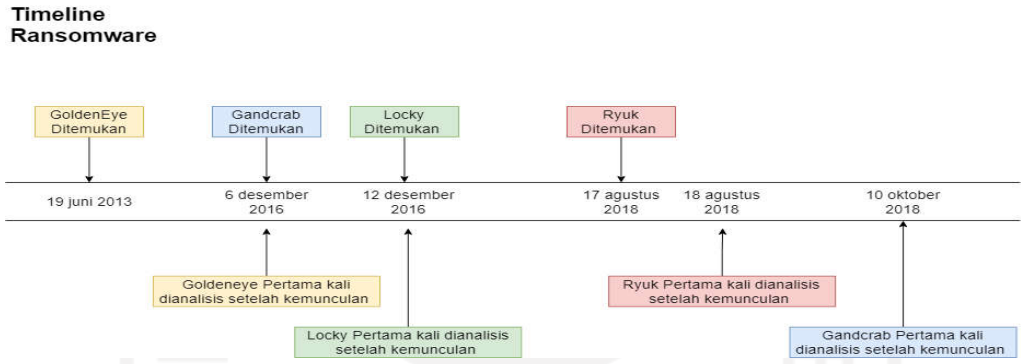


Gambar 4. 41 *Signature* ryuk

Gambar 4.36 adalah *signature* dari *ransomware* ryuk yang diberikan oleh cuckoo sehingga dapat mengerti apa saja yang dikerjakan oleh *ransomware*, tidak semua API yang dijalankan oleh *ransomware* akan ditampilkan sebagai *signature*, namun *signature* memudahkan untuk mengerti apa saja yang dilakukan oleh *ransomware*.

4.3 Evolusi *Ransomware*

Berdasarkan hasil analisis yang telah dilakukan diketahui bahwa aspek-aspek *ransomware* yang berevolusi adalah metode pengenkripsian *file*, metode penyerangan, dan mekanisme pertahanan terhadap analisis. Untuk dapat mengerti evolusi dari *ransomware* pertama yang dilakukan ialah mengetahui kapan *ransomware* dibuat dan kapan *ransomware* dianalisis pertama kali setelah kemunculannya.



Gambar 4. 42 Timeline *ransomware*

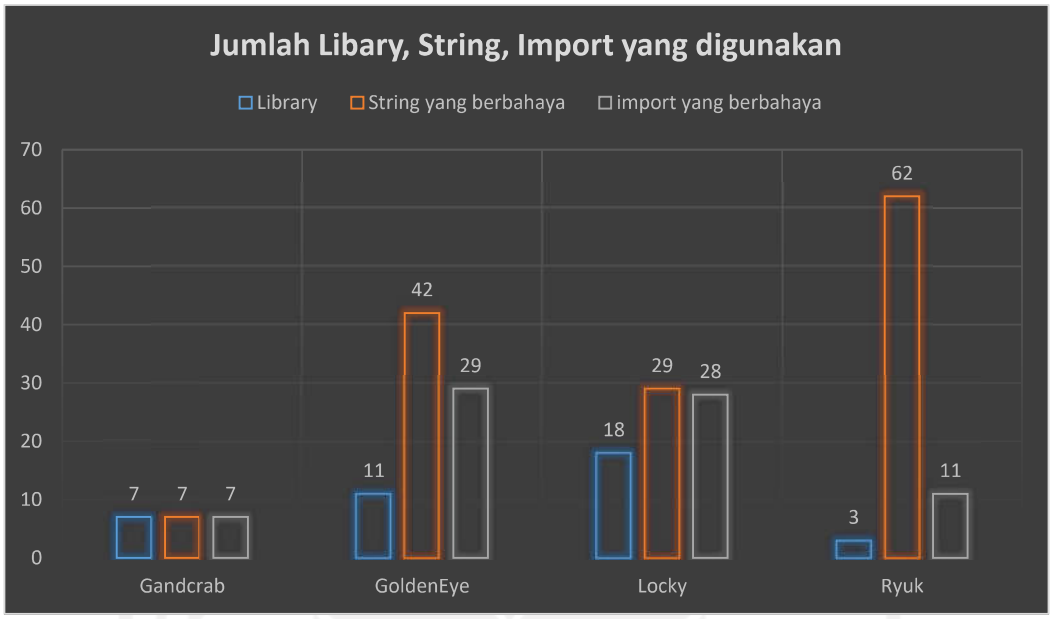
Berikut adalah perbandingan jumlah dari banyaknya *library*, *import* dan *string* yang digunakan pada setiap *ransomware*:

Tabel 4. 17 Jumlah *library*, *import* dan *string* yang digunakan setiap *ransomware*

Nama		**Jumlah Yang Digunakan**			
	Library	***String***		***Import***	
	Digunakan	**BlackList**	**Hint**	**Blacklist**	**Total**
Gandcrab	7	7	70	7	239
GoldenEye	11	42	225	29	248
Locky	18	29	171	28	197
Ryuk	3	62	649	11	70

Pada tabel 4.17 informasi yang didapatkan menggunakan tool PeStudio dengan penjelasan *blacklist* berarti *string* atau *import* telah dinyatakan berbahaya. Sedangkan *hint* adalah *string* yang perlu diperhatikan dan dicari kegunaannya untuk mendapatkan informasi lebih.

Pada gambar 4.43 adalah grafik yang menggambarkan jumlah *library*, *string*, dan *import* yang digunakan oleh setiap *malware*, namun pada *string* dan *import* hanya menggunakan data yang diblacklist oleh PeStudio.



Gambar 4. 43 Jumlah string, library, dan import yang digunakan

Tidak semua *ransomware* menggunakan metode yang sama dalam melakukan penyerangan pada komputer dan mencegah pemakaian komputer hingga korban membayar jumlah yang diminta. Berikut adalah hasil dari analisis dinamis untuk menyerang dan menghindari proses analisis.

Tabel 4. 18 Data yang didapatkan dari metode analisis dinamis dalam penyerangan komputer

Nama	**Metode yang ditemukan pada analisis dinamis**
Gandcrab	Mencoba untuk mengubah atau memodifikasi sertifikasi pada sistem, mengenkripsi *file*, dan membuat ICMP
GoldenEye	Melakukan instalasi *tool*kit pada hard disk dan membuat ICMP
Locky	Membuat komunikasi dengan *fhost* dan menyambungkan untuk mengunduh program berbahaya
Ryuk	Mengalokasi izin eksekusi kepada proses lain untuk melakukan injeksi kode, menginstall autorun pada startup Windows, memanipulasi memori untuk melakukan injeksi proses, dan membuat ICMP

Pada analisis statis hasil yang dikeluarkan adalah beberapa *string* yang perlu diperhatikan dan dilihat perilakunya pada analisis dinamis, karena pada analisis dinamis terkadang tidak medeteksi perilaku *ransomware* seperti yang diperlihatkan pada tabel 4.18.

Tabel 4. 19 Data yang didapatkan dari metode analisis statis dalam penyerangan komputer

Nama	**Metode yang ditemukan pada analisis statis**	
	String	**Penjelasan**
Gandcrab	Clipboard modify, *registry* modify.	Gandcrab dapat menggunakan clipboard sebagai proses penyerang dan mengubah *registry*
GoldenEye	Clipboard modify, register modify, Thread modify, *File*s scanning, detect monitor.	GoldenEye dapat menggunakan clipboard sebagai proses penyerang , mengubah *registry*, mengubah thread, melakukan pencarian pada *file* dan medeteksi resolusi dari monitor
Locky	Storage modify, security info, networkconfig, device IO control, memory modify.	Dapat memodifikasi penyimpanan, mendapatkan sekuriti yang digunakan pada komputer, mengubah konfigurasi jaringan, mengontrol hardware, dan memodifikasi memori.
Ryuk	*File* modify , thread modify, memory modify.	Ryuk dapat Memodifikasi *file*, memodifikasi thread dan memodifikasi memori

Dari hasil analisis yang didapatkan dengan metode statis dan dinamis setiap *ransomware* memiliki cara tersendiri dalam menginfeksi komputer namun *ransomware* memiliki kesamaan dari import dan library yang digunakan, dari kesamaan dapat dilakukan perbandingan apakah import yang digunakan melalui library yang sama meningkat untuk membantu proses infeksi kepada komputer, seperti penggunaan fitur "clipboard" yang digunakan untuk mengambil data yang digunakan oleh *ransomware* GoldenEye dan Gandcrab, dan membuat fungsi ICMP digunakan sebagai pengiriman data.

Setiap *ransomware* menggunakan metode berbeda beda dalam menghindari analisis ataupun mengganggu proses analisis. Hal ini dilakukan untuk memperpanjang viabilitas *ransomware* agar *ransomware* dapat terus memberikan keuntungan bagi penyebar *ransomware*, berikut adalah hasil dari analisis dinamis:

Tabel 4. 20 Data yang didapatkan dari metode analisis dinamis untuk menghindari proses analisis

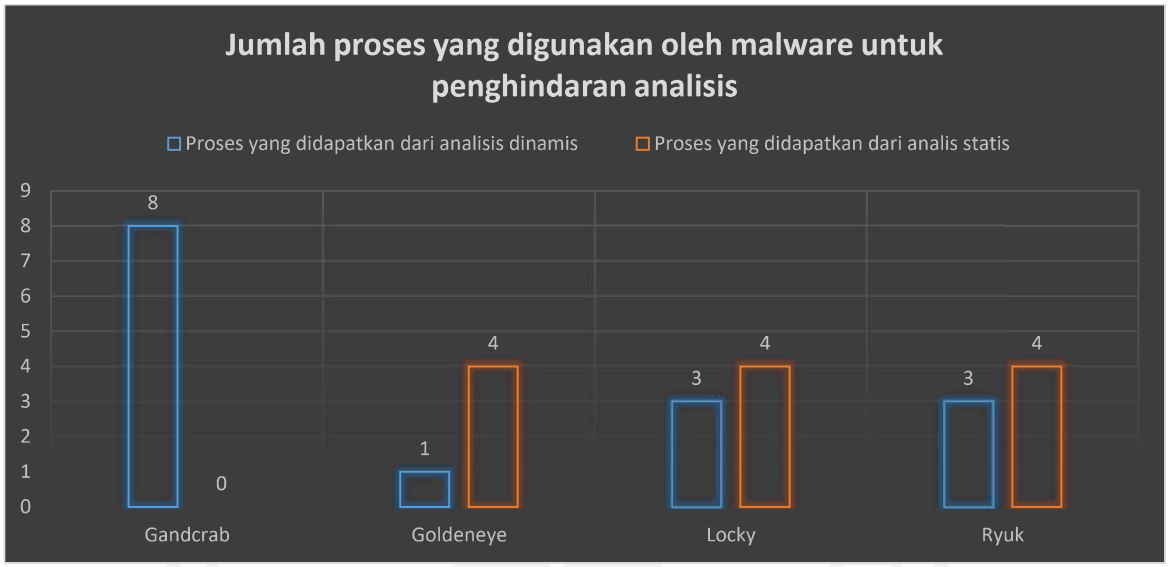
Nama	Metode yang ditemukan dengan metode analisis dinamis							
	Debugger	**Sandbox evasion**	**Code injection**	**Memory dumping**	**Virtual network**	**Anti-virtual**	**Detect Cuckoo**	**Disk check**
Gandcrab	✓	✓	✓	✓	✓	✓	✓	✓
GoldenEye	x	x	x	x	x	✓	x	x
Locky	✓	x	x	x	x	✓	✓	x
Ryuk	x	✓	✓	✓	x	x	x	x

Pada beberapa *ransomware* analisis dinamis tidak menemukan fungsi untuk medeteksi program *debugger*, namun pada analisis statis ditemukan fungsi yang berguna untuk mendeteksi program *debugger*.

Tabel 4. 21 Data yang didapatkan dari metode analisis statis untuk menghindari proses analisis

Nama	**Metode yang ditemukan pada analisis statis**	
	String	**Penjelasan**
Gandcrab	-	Pada *ransomware* Gandcrab peneliti tidak menemukan fungsi tersembunyi yang tidak terdeteksi pada analisis dinamis untuk menghindari analisis.
GoldenEye	IsDebuggerPresent, TerminateProcess, GetCurrentProcessId, sleep.	Pengecekan apakah debugger ada di dalam sistem yang diinfeksi dan memperlambat eksekusi ataupun tidak menjalankan program sama sekali jika beberapa kondisi terpenuhi.
Locky	ExitProcess, TerminateProcess, GetCurrentProcessId, sleep	Memperlambat eksekusi ataupun tidak menjalankan program sama sekali jika beberapa kondisi terpenuhi.
Ryuk	IsDebuggerPresent, TerminateProcess, GetCurrentProcessId, ExitProcess	Pengecekan apakah debugger ada di dalam sistem yang diinfeksi dan memperlambat eksekusi ataupun tidak menjalankan program sama sekali jika beberapa kondisi terpenuhi.

Berikut adalah tampilan grafik dari banyaknya metode yang digunakan untuk menghindari proses analisis dan mengganggu proses analisis:



Gambar 4. 44 Jumlah proses penghindaran analisis

Dari hasil analisis yang didapatkan dengan metode statis dan dinamis *ransomware* dalam menghindari proses analisis memiliki banyak cara, seperti menggunakan fungsi "sleep" dan membuat *ransomware* dalam bentuk paket dimana fungsi ini dapat menunda proses pengeksekusian agar analisis dapat terganggu, namun dengan berkembangnya tool untuk menganalisis *ransomware* memiliki banyak metode untuk menghindari analisi, seperti pendeteksi mesin virtual tidak hanya mengecek jumlah hardisk, *ransomware* juga mengecek CPU,RAM, dan jaringan.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah berhasil membangun lingkungan untuk melakukan analisis statis yang dilakukan dengan menginstall Windows 10 secara virtual agar dapat melakukan analisis secara aman dan berhasil membangun lingkungan untuk analisis dinamis dalam menginstall Cuckoo dan dapat terintegrasi dengan baik kepada mesin virtual Windows sebagai wadah eksekusi. Objek yang dianalisis secara statis dan dinamis berupa program berbahaya *ransomware* dengan nama yang dikenal sebagai Gandcrab, GoldenEye, Locky, dan Ryuk.

Dengan melakukan analisis statis dan dinamis dapat mengetahui informasi yang lebih lengkap untuk mengerti cara kerja *ransomware*, walaupun metode dari dua analisis memiliki perbedaan dalam melakukan analisis, dua proses analisis tersebut memberikan hasil yang membantu satu sama lain dan menutupi kekurangan dari setiap analisis.

Dari analisis yang telah dilakukan *ransomware* berevolusi yang diambil dari cara *ransomware* melakukan penyerangan, *ransomware* semakin banyak menggunakan fungsi yang dapat mengambil data lebih, tidak hanya mengunci *file ransomware* juga mengambil data lainya. Kemampuan untuk menghindari analisis pada *ransomware* juga berevolusi, dengan banyak aplikasi dan metode untuk menganalisis *ransomware*, *ransomware* berevolusi untuk menghindari metode-metode yang ada.

Berdasarkan hasil dan analisis keempat *ransomware*, dapat ditarik kesimpulan sebagai berikut:

1. Evolusi *ransomware* meningkat setiap tahunnya, dilihat dari metode penyerangan dan kemampuan *ransomware* untuk menghindari dan mengganggu proses analisis.
2. Untuk mendapatkan hasil yang optimal diperlukan dua metode analisis yang berbeda yaitu metode statis dan dinamis

5.2 Saran

Penelitian ini masih terdapat banyak kekurangan karena analisis yang dilakukan termasuk analisis dasar dalam melakukan analisis pada *ransomware* sehingga membutuhkan penambahan yang lebih baik lagi untuk menghasilkan laporan analisis yang lebih baik dan dapat dimengerti oleh orang awam. Oleh karena itu, untuk penelitian ke depan disarankan:

1. Menggunakan sampel *ransomware* dengan famili yang sama untuk memudahkan pemetaan evolusi.
2. Mempelajari lebih dalam tentang *ransomware* dan teknik analisis *ransomware*.

Saran yang peneliti berikan dikarenakan hasil analisis menunjukkan bahwa penelitian yang dilakukan hanya menyentuh permukaan dari *ransomware*. Sedangkan terdapat metode analisis lainnya yang dapat mengetahui struktur dan cara kerja *ransomware* dengan lebih dalam.

DAFTAR PUSTAKA

Anon. 2021. "White Paper / Oracle VM VirtualBox Overview / Version 2.0 Oracle VM VirtualBox Overview."

Cuckoo sandbox. 2020. "Cuckoo Sandbox Book Release 2.0.7 Cuckoo Sandbox."

Datta, Arkajit, and Kakelli Anil Kumar. 2021. *An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis Medical Image Fusion View Project Wireless Sensor Networks View Project An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis*.

FEA Fundamentals. 2019. "The Difference between Static and Dynamic Analysis - Enterfea." Retrieved January 28, 2022 (https://enterfea.com/difference-between-static-and-dynamic-analysis/).

Josh Fruhlinger. 2020. "*Ransomware* Explained: How It Works and How to Remove It | CSO Online." Retrieved January 24, 2022 (https://www.csoonline.com/article/3236183/what-is-*ransomware*-how-it-works-and-how-to-remove-it.html).

K. A, Monnappa. 2018. *Learning Malware Analysis : Explore the Concepts, Tools, and Techniques to Analyze and Investigate Windows Malware.* Packt Publishing Ltd.

Kurt Baker. 2021. "11 Types of Malware + Examples That You Should Know." Retrieved January 20, 2022 (https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/).

MalwareBytes. 2020a. "GandCrab *Ransomware* - Removal and Prevention Guide | Malwarebytes." Retrieved January 24, 2022 (https://www.malwarebytes.com/gandcrab).

MalwareBytes. 2020b. "Goldeneye *Ransomware* - the Petya/Mischa Combo Rebranded | Malwarebytes Labs." Retrieved January 24, 2022 (https://blog.malwarebytes.com/threat-analysis/2016/12/goldeneye-*ransomware*-the-petyamischa-combo-rebranded/).

MalwareBytes. 2020c. "Locky: Ransom.Locky | Malwarebytes Labs | Detections." Retrieved January 24, 2022 (https://blog.malwarebytes.com/detections/ransom-locky/).

MalwareBytes. 2020d. "Ryuk - What Is Ryuk *Ransomware*?" Retrieved January 24, 2022 (https://www.malwarebytes.com/ryuk-*ransomware*).

RedHat. 2019. "What Is a Virtual Machine (VM)?" Retrieved January 24, 2022 (https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine).

Subedi, Kul Prasad, Daya Ram Budhathoki, and Dipankar Dasgupta. 2018. "Forensic Analysis of *Ransomware* Families Using Static and Dynamic Analysis." Pp. 180–85 in

Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018. Institute of Electrical and Electronics Engineers Inc.

Volatility Foundation. 2020. "The Volatility Foundation - Open Source Memory Forensics." Retrieved January 27, 2022 (https://www.volatilityfoundation.org/).

YusirwanS, Syarif, Yudi Prayudi, and Imam Riadi. 2015. "Implementation of Malware Analysis Using Static and Dynamic Analysis Method." *International Journal of Computer Applications* 117(6):11–15. doi: 10.5120/20557-2943.

Zimba, Aaron, and Mumbi Chishimba. 2019. "Understanding the Evolution of *Ransomware*: Paradigm Shifts in Attack Structures." *International Journal of Computer Network and Information Security* 11(1):26–39. doi: 10.5815/ijcnis.2019.01.03.

LAMPIRAN