Journal of Software Engineering Ampera
Vol. 1, No. 3, October 2020 e-ISSN: 2775-2488
https://journal-computing.org/index.php/journal-sea/index

Analisis Perilaku Malware Dengan Metode Surface Analysis Dan Runtime Analysis

Ahmad Siddiq[1], Helda Yudiastuti[2], Febriyanti Pajaitan*[3]

[1,3]Informatics Departement , Bina Darma University, Palembang, Indonesia
[2]Vocational Departement, Bina Darma University, Palembang, Indonesia
Email: ahmadsiddiq96.as@gmail.com[1], helda.yudiastuti@binadarma.ac.id[2], febriyanti_panjaitan@binadarma.ac.id[3]

Abstrak

Saat ini penggunaan komputer sudah menjadi hal yang biasa baik oleh pegawai, mahasiswa dan anak-anak sekolah. Umumnya sistem operasi yang digunakan adalah Windows, oleh karena kemudahan, namun hal ini juga memberikan peluang bagi orang-orang yang memiliki niat buruk untuk mengambil keuntungan secara ilegal dengan berbagai macam cara, salah satunya dengan membuat sebuah Malware yang disusupkan dalam sebuah program atau dengan cara-cara lainnya. Berdasarkan hal tersebut penulis ingin meneliti perilaku apa saja yang dilakukan oleh Malware ketika menyerang sebuah system operasi khususnya Windows menggunakan metode Surface Analysis dan Runtime Analysis untuk. Dengan menggunkan metode ini, informasi yang didapat berupa jenis file asli yang di periksa, ukuran file sebenarnya dan melihat apakah ada file lain di dalam file yang di periksa. Dari pemeriksaan menggunakan metode ini juga didapatkan hash function atau MD5 yang merupakan identitas unik dari file yang di periksa serta dapat memeriksa secara keseluruhan seperti proses yang berjalan di komputer, dan peristiwa janggal lainnya yang biasa terjadi saat komputer terinfeksi Malware. Sehingga nantinya dapat diklasifikasikan kedalam Malware jenis apa.

Kata Kunci: Windows, Malware, Surface Analysis, Runtime Analysis

1. PENDAHULUAN

Di era digitalisasi saat ini, komputer seolah sudah menjadi suatu hal yang wajib ada dalam berbagai instansi, baik pemerintah maupun swasta, akademik dan juga industri telah menjadikannya sebagai media *input* dan *output* data dan informasi, sehingga komputerisasi memegang peranan penting dalam pengolahan data. Komputerisasi memberikan efisiensi waktu dan efektivitas keamanan data yang lebih baik dibandingkan dengan cara

manual. Namun ini juga memberikan kesempatan bagi orang-orang yang berniat buruk untuk memanipulasi data dan informasi tersebut, seperti mencuri data, membajak akun dan lain sebagainya. Hampir dari setiap kejahatan tersebut melibatkan sebuah *Malware (Malicius Software)* atau bisa juga disebut dengan Program jahat (Adenansi dan Novarina, 2017) [1]. Semakin hari perkembangan *Malware* semakin mutakhir dan kompleks, pembuat *Malware* selalu melakukan perubahan guna mencari-cari celah keamanan dengan membuat banyak varian pada pola serangannya (Perdhana, 2011) [5]. Dari itu pengguna sistem komputer harus lebih waspada dan berhati-hati, terlebih pada pengguna platfom Windows dan MAC.

Masuknya sebuah *malware* ke dalam system dapat melalui berbagai macam cara, seperti disisipkan pada sebuah *file* atau aplikasi tertentu, sehingga korban tidak menyadari bahwa komputernya telah disusupi sebuah *malware*. Hal inilah yang melatar belakangi saya sebagai penulis untuk meneliti perilaku dari *Malware* tersebut, agar dapat mengetahui pola serangan dan aktivitasnya sehingga nantinya dapat dikelompokkan berdasarkan pola-pola sebuah *Malware*. Untuk dapat menganalisis apakah sebuah *file* terdapat *Malware* didalamnya atau *file* itu sendiri merupakan sebuah *Malware* dan juga mengetahui apakah sebuah komputer telah terserang *Malware*, dapat menggunakan beberapa macam metode, di antaranya metode *Surface analysis* dan *Runtime analysis*. Dengan menggunakan kedua metode ini, kita mampu mendapatkan informasi dari sebuah *Malware*, seperti jenis *file* dan tipe serangannya.

Metode *Surface analysis* merupakan sebuah metode yang digunakan untuk menganalisis bagian luar dari sebuah *file*, dari pemeriksaan tersebut memberikan informasi berupa jenis *file* asli yang di periksa, ukuran *file* sebenarnya dan melihat apakah ada *file* lain di dalam *file* yang di periksa. Metode ini juga mampu memberikan informasi yang cukup akurat dalam mengetahui *Malware* yang menyamar atau bersembunyi di dalam *file* lain. Dari pemeriksaan menggunakan metode ini juga didapatkan *hash function* atau *fingerprint* (*MD5*, *SHA*-1) yang merupakan identitas unik dari *file* yang di periksa. Metode *Runtime analysis*, sesuai dengan namanya metode ini akan menjalankan atau mengaktifkan *file* yang di periksa untuk mendapatkan informasi mengenai kegiatan apa saja yang di lakukan *Malware* saat berhasil menginfeksi sebuah komputer. Pada tahap ini, komputer akan di periksa secara keseluruhan seperti proses yang berjalan di komputer, perubahan

registry, aktivitas komunikasi internet dan peristiwa janggal lainnya yang biasa terjadi saat komputer telah terinfeksi *Malware* (Agung, 2011) [2].

2. METHODS

2. 1. Metode Penelitian

Metode penelitian yang digunakan adalah metode kualitatif yaitu data yang didapat dari membaca informasi, mengamati objek, dan wawancara (Sugiyono, 2009:19) [6]. Sumber data yang digunakan terdiri dari dua bagian yaitu primer dan sekunder. Data primer didapat dengan melakukan kegiatan membaca informasi, mengamati aktivitas objek, dan bertanya kepada bagian IT. Data sekunder diperoleh dengan membaca pustaka yang berhubungan dengan segala hal yang akan diteliti serta mendokumentasikan kebutuhan yang ada pada obejk penelitian.

2. 2. Metode Analisis

Metode yang di gunakan dalam penelitian ini adalah *Surface analysis* dan *Runtime analysis*. Menurut Richardus Eko Indrajit selaku ketua dari Lembaga *Indonesia Security Incident Response Team On Internet Infrastructure* (IDSIRTII) periode 2010, mengatakan bahwa kedua metode ini merupakan pendekatan yang umum digunakan untuk mendeteksi apakah program tersebut merupakan klasifikasi jenis *malware* atau bukan. Dengan perpaduan kedua metode tersebut, peneliti juga dapat mengetahui dengan baik bagaimana aktivitas *malware* dalam sebuah sistem.

Menurut Indrajit (2012) [3], *Surface Analysis* adalah suatu kajian pendeteksian *malware* dengan mengamati sekilas ciri-ciri khas sebuah *file* program tanpa harus mengeksekusinya. Untuk melihat ciri khas tersebut dapat dilakukan dengan menggunakan bantuan *software* atau perangkat aplikasi pendukung. Analisa ini memiliki ciri-ciri sebagai berikut :

1) Program yang dikaji tidak akan dijalankan, hanya akan dilihat “bagian luarnya” saja (sebagai analogi selayaknya orang yang ingin membelibuah-buahan, untuk mengetahui apakah buah yang bersangkutan masih mentah atau sudah busuk cukup dengan melihat permukaan kulitnya, membauhinya, dan meraba-raba teksturnya atau struktur kulitnya). Dari sini akan dicoba ditemukan hal-hal yang patut

untuk dicurigai karena berbeda dengan cirikhas program kebanyakan yang serupa dengannya; dan

2) Sang pengkaji tidak mencoba untuk mempelajari "*source code*" program yang bersangkutan untuk mempelajari algoritma maupun struktur datanya (sebagaimana layaknya melihat sebuah kotak hitam atau "*black box*".

Runtime analysis pada dasarnya ada kesamaan antara *runtime analysis* dan *surface analysis*, yaitu keduanya sama-sama berada dalam ranah mempelajari ciri-ciri khas yang selayaknya ada pada sebuah program yang normal. Bedanya adalah bahwa *runtime analysis*, dipersiapkan sebuah prosedur dan lingkungan untuk mengeksekusi atau menjalankan programnya yang dicurigai mengandung atau sebagai *malware* tersebut. Model analisa ini menghasilkan kajian yang lebih mendalam karena selain dihilangkannya proses "menduga-duga", dengan mengeksekusi *malware* dimaksud akan dapat dilihat "perilaku" dari program dalam menjalankan "skenario jahatnya" sehingga selanjutnya dapat dilakukan analisa dampak terhadap sistem yang ada.

Menurut (Nurdiyanto, 2013) [4] dalam buku "*practical malware analysis*", *malware* dibagi menjadi 9 kelompok sedangkan berdasarkan sasaran serangannya, *malware* dibagi menjadi dua kelompok.

1) Backdoor adalah jenis malware yang menyusup ke sebuah sistem (login) dengan ilegal dan tidak normal tanpa diketahui.
2) *Botnet* dalah jenis yang memberikan akses pada penyerang untuk memberikan instruksi pada setiap komputer yang terkena sesuai keinginannya melalui *server*.
3) *Downloader* jenis ini adalah perilaku *malware* yang dapat *mendownload software* jahat lainnya dan menginstalasinya, sehingga penyerang mampu mendapatkan berbagai informasi dari korban.
4) *Information-stealing Malware* perilaku *Malware* ini mampu mengumpulkan informasi dari korban dan mengirimkannya kepada pelaku.
5) *Launcher* perilaku jenis ini digunakan pelaku untuk menjalankan program lainnya untuk mendapatkan informasi lebih dalam dari korban.
6) *Rootkit* digunakan keberadaan kode lainnya agar pelaku dapat mengakses komputer dari jarak jauh tanpa diketahui.
7) *Scareware* adalah sebuah tipuan dari pelaku, untuk mendapatkan keuntungan dengan menawarkan sesuatu yang memaksa korban membeli produknya, seperti mengatakan bahwa komputer korban

telah terserang virus, dan virus itu hanya dapat diatasi dengan antivirus buatannya, padahal kenyataannya program itu hanya menghapus *Scareware* itu sendiri.

8) *Spam-sending Malware* teknik ini di gunakan pelaku untuk mengirimkan spam pada komputer korbannya dengan cara menginfeksi sistemnya.
9) *Worm* atau *Virus* adalah jenis yang dapat menggandakan dirinya sendiri tanpa perlu bergantung pada program lain, ia dapat menyebar dengan cepat dan masuk ke dalam jaringan komputer melalui *port* layanan yang terbuka.

2. 3. Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini yaitu :

1) Studi Pustaka *(Literature Review)*, Dengan menggunakan metode ini, penulis melakukan pencarian dan pembacaan tentang buku-buku, majalah atau referensi yang berhubungan dengan *malware*, analisa *malware*, *Honeypot* dan lain-lain.
2) Wawancara, Wawancara dilakukan dengan beberapa pihak yang terkait dengan penggunaan Sitem Operasi windows, di antaranya Admin PT. Taysir di kota Palembang dan beberapa pengguna di lingkungan Universitas Binadarma Palembang untuk mendapatkan data mengenai dampak dan perilaku *malware* apa saja yang mereka temukan ketika terserang sebuah *maware*.

3. HASIL DAN PEMBAHASAN

3.1 *Surface Analysis*

Dalam metode ini, peneliti akan melihat dan mengidentifikasi unsur-unsur yang terdapat di dalam sebuah *file* yang dicurigai sebagai *malware*.

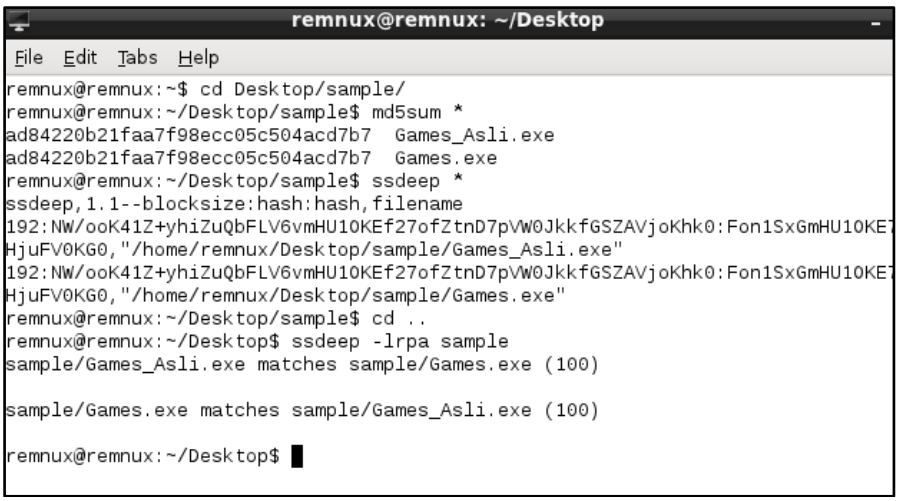
1) *File Atrribute Analysis*

Pada tahapan ini kita berhasil mendapatkan atribut dari sample dengan tipe *executable* pada sistem 32-bit dan "*MD5(AD84220B21FAA7F98ECC05C504ACD7B7)*". Selanjutnya kita dapat melihat indikators yang dihasilkan oleh pestudio dengan menggunakan angka 1 sampai 9 sebagai alamatnya. Angka 1 dan 2 menunjukan bahwa *file* tersebut memiliki tingkat bahaya yang cukup tinggi, karena memiliki kemampuan yang

biasanya ada pada sebuah malware, seperti memodifikasi registry dan terdapat blacklist pada stringnya yaitu melakukan perubahan pada *file*, menghapus serta merusak. (Agung, 2011). Maka, semakin banyak angka 1 dan 2, maka semakin besar pula kemungkinan *file* tersebut merupakan sebuah malware. Didapati pada *file Games.exe* terdapat 5 dari 11 indikators yang menunjukan bahwa *file* tersebut kemungkinan merupakan *file* yang berbahaya.

2) *Fuzzy Hashing*

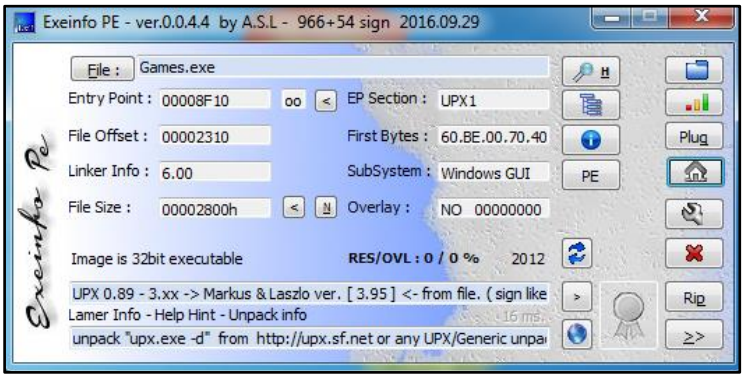
Untuk mengidentifikasi keaslian dari malware yang akan dianalisa, kedua *file* kita letakkan dalam satu folder yang sama untuk memudahkan pencocokan *md5* dan *hash* diantara keduanya. Terlihat pada gambar dibawah menunjukan bahwa *md5* dan *hash* memiliki nilai kesamaan 100%, hal ini menunjukan bahwa *file* yang akan kita analisa nantinya sama persis dengan *file* asli.



Gambar 2. *Fuzzy Hasing file* sempel

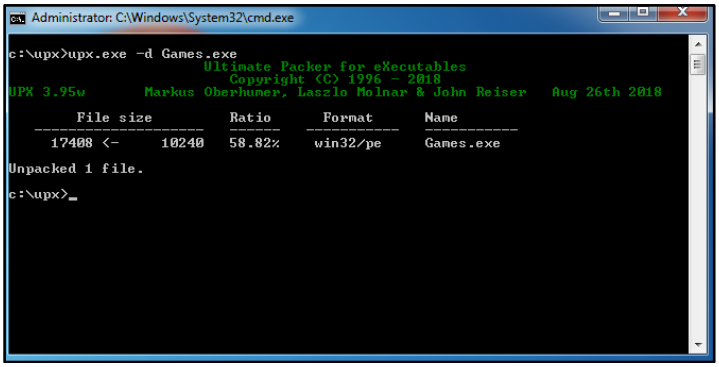
3) *Packer Chack*

Penting untuk kita mengetahui apakah *file* tersebut telah di *pucking* atau tidak oleh pembuatnya karena strings pada *file* yang telah di *packing* akan terenkripsi sehingga dapat mempersulit analisa nantinya. Terlihat pada gambar di bawah ini yang menunjukan bahwa *file* dikompres atau *packing*, maka sebelum mulai melakukan analisa ke tahap selanjutnya *file* harus di *unpacking* untuk dapat melihat srting yang ada pada *file* tersebut.



Gambar 3. Hasil Packers Chack *file Games.exe* dengan Exeinfo PE

Untuk melakukan *unpacking* kita harus mengetahui tool apa yang digunakan untuk *packing* pada *file* tersebut, informasi yang didapat dari Exeinfo bahwa *tool* yang digunakan adalah UPX 0.89 maka kita harus mendownloadnya terlebih dahulu, pada *lamer info* terdapat informasi bahwa tool bisa didapatkan dialamat *http://upx.sf.net*. Setelah *tool* didapat maka kita akan menjalankannya melalui *Commend Prompt (CMD)* dengan menempatkan *file* satu *folder* bersama *tool*.



Gambar 5. Hasil *Unpacking*.

File PE yang berisi *Header* dan beberapa bagian yang penting. Di bawah ini merupakan keterangan dari gambar di atas:

a) .text : Pada bagian ini berisikan kode yang dapat di eksekusi
b) .rdata : Bagian ini menampung dan membaca data yang dapat diakses secara global
c) .data : Menyimpan data global yang dapat diakses hanya melalui program saja

4) *String analysis*

Tool yang kita gunakan selanjutnya yaitu Yara untuk menganalisa keberadaan string pada *file*. Dengan menggunakan aturan ini kita akan melihat ada berapa string yang didapatkan oleh Yara *Rule*. Semakin banyak string yang didapatkan maka semakin besar kemungkinan *file* merupakan sebuah malware karena aturan yang ada pada yara *rule* mampu membaca string pada program.

Gambar 5. Hasil Yara *Rule*.

Analisa strings, pada tahap ini penulis mendapatkan hasil dengan menggunakan Pestudio yaitu terdapat 101 string yang terbaca pada *file* tersebut dari 272 yang ada. Terlihat adanya sebuah alamat web yaitu *www.1535ss.com*, ini membuktikan adanya interaksi dari program untuk mengakses ke jaringan internet. Penulis juga menggunakan dua tool yang ada pada sistem operasi remnux yaitu yara dan strings, hasil dari kedua tool ini menampilkan adanya string yang biasa dimiliki oleh sebuah *malware*. Hasil yang didapat dengan menggunakan ketiga tools berupa string yang biasanya terdapat dalam sebuah malware diantaranya :

a) *GetSystemDirectory,* digunakan untuk membuka penyimpanan pada sistem
b) *DeleteFile,* digunakan untuk menghapus *file*
c) *Sleep,* digunakan untuk memberikan perintah masuk ke mode tidur pada sistem, hal ini memungkinkan adanya kontrol jarak jauh dari *file* tersebut
d) *InternetOpenUr,* digunakan untuk dapat terhubung ke beberapa server eksternal untuk mengunduh sesuatu
e) *HTTP,* ini digunakan untuk memanggil sebuah laman web

f) *WININET*.dll, merupakan modul yang didalamnya terdapat instruksi InternetOpenUr yang digunakan untuk dapat terhubung kesebuah layanan server.
g) *OpenSCManager,* untuk membentuk koneksi ke manajer kontrol layanan pada komputer yang ditentukan dan membuka database pengelola kontrol layanan yang ditentukan.

5) *Malware Scan*

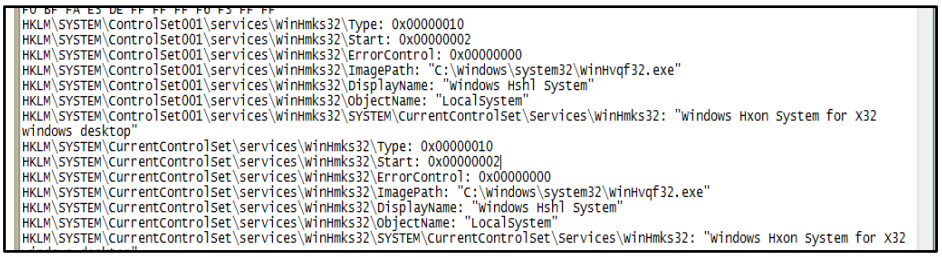
Pada tahap ini terlihat hasil *scan* dari *sample* menunjukan bahwa *file* belum terdeteksi sebagai sebuah *malware* melalui *md5* ketika *file* sudah dibongkar, dari itu kita akan mencari tau lebih banyak pada metode selanjutnya dengan cara mengeksekusi *file*. Namun setelah *file* di *Upload* maka virus total menampilkan informasi bahwa 53 dari 70 anti virus menyatakan *file* tersebut merupakan sebuah *malware* dengan jenis *worm* dan *trojan*. Perilaku dari program ini yaitu mencoba untuk melakukan komunikasi melalui jaringan ke alamat *www.1535ss.com* , membuat *file* baru pada *system32* yang merupakan pusat *file system* pada sistem operasi windows.

3.2 *Runtime Analysis*

Pada tahap ini kita akan menjalankan *file* yang akan kita Analisa untuk melihat aktivitas apa saja yang dilakukannya, sehingga kita dapat menilai apakan *file* tersebut merupakan malware atau bukan.

1) Melihat perubahan Registry dengan Regshot

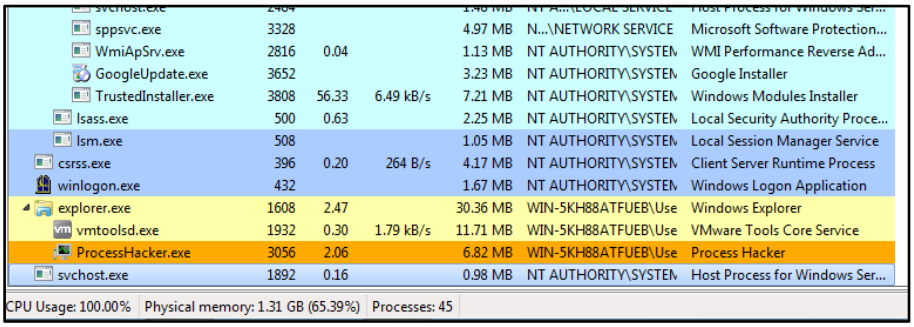
Dengan menggunakan Regshot kita dapat melihat perubahan yang terjadi pada regitry setelah *file* dijalankan. Terlihat *file* melakukan beberpa perubahan dengan menambahkan beberapa *file registry* berupad "*HKLM\SYSTEM\ControlSet001\services\WinHmks32*".



Gambar 6. Hasil *monitoring* Rigshot

2) Melihat semua aktivitas aplikasi pada sistem dengan *Process Hacker*

Aplikasi ini akan memeriksa semua *file* yang sedang berjalan di dalam sistem, mirip seperti *Task Manager* yang ada di windows, namu lebih jelas dan rinci. Terliha pada Proses Hacker adanya proses svchost.exe ketika *file Games.exe* di eksekusi. Ketika *file* svchost.exe kita Terminate, maka terlihat pada wireshark layanan yang meminta akses ke *www.1535ss.com* berhenti.



Gambar 7. Proses yang berjalan di latar belakang sistem

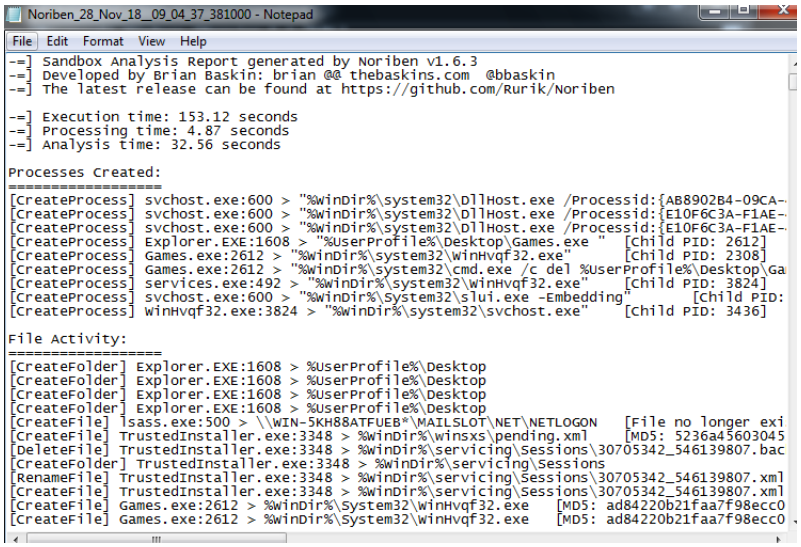
3) Melihat aktivitas malware di dalam sebuah jaringan menggunakan CaptureBAT

Dengan menggunakan CaptureBAT kita akan melihat perilaku dari *file Games.exe* setelah *file* tersebut dieksekusi. Install *file* CaptureBAT pada windows, selanjutnya ia akan meminta untuk restart, setelah itu buaka cmd dan ketikan "*cd C:\Program File\Capture\CaptureBAT.exe*" maka tool akan aktif, selanjutnya kita juga persiapkan Wireshark pada remnux untuk melihat paket data yang berjalan selama proses Analisa berlangsung pada captureBAT. Eksekusi *file Games.exe* dengan cara klik kanan, Run Administrator, dan tunggu beberapa saat sambill melakukan beberapa aktivitas seperti mengetikan sesuatu di notepade untuk melihat kemungkinan adanya keylog yang dilakukan, Selanjutnya matikan captureBAt dengan mengklik enter, maka ia akan menyimpan hasil capture secara otomatis di "*C:\Program File\Capture*" dalam bentuk "*capture_11102018_1056.zip*".

4) Analisa menggunakan Noriben Malware Analysis Sandbox

Noriben bekerjasama dengan procmon dalam memonitoring aktivitas yang terjadi dalam system, sehingga kita perlu meletakkan *file* procmon dalam satu

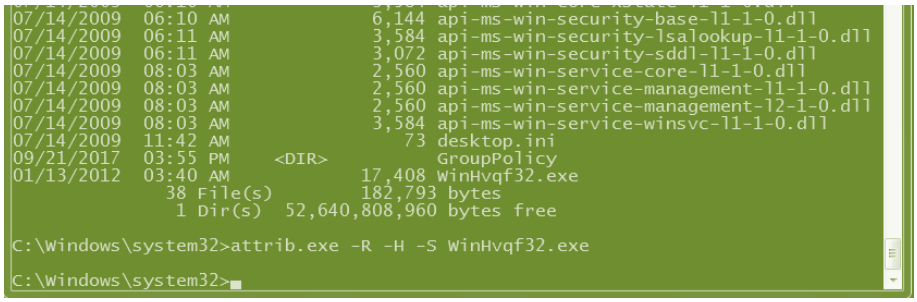
folder. Tempatkan sempel malware di desktop agar mudah ditemukan, dan selanjutnya kita mengecek alamat *IP* yang ada pada Windows, jika semua sudah selesai maka kita beralih ke system operasi Remnux untuk menjalankan wireshark yang nantinya akan memonitoring semua lalulintas jaringan dengan ditambah sebuah perangkat lunak Inetsim untuk memanipulasi jaringan local sehingga seakan-akan seperti berjalan pada jaringan internet. Jika semua telah selesai maka jalankan Noriben dan tunggu sampai dalam keadaan *runtime complete*. Jalankan malware dengan memberikan hak akses administrator.



Gambar 8. Hasil *monitoring* Noriben

Hasil dari Analisa Noriben menunjukan pada Proceses created adanya proses *WinHvqf32.exe* ketika *file Games.exe* dijalankan dan *file* tersebut meminta layanan *svchost.exe* (*Generic Host Process* untuk *Win32 Services*) yang merupakan bagian integral dari OS Windows untuk melalui layanan services yang mana layanan tersebut dapat melakukan *Automatic Update*, hal ini menjadi mungkin bagi *file Games.exe* untuk mengakses jaringan. Pada *file activity* juga menampilkan adanya aktivitas pembuatan *file WinHvqf32.exe* oleh *file Games.exe* yang memiliki md5 sama persis di lokasi *system32*. Ada yang menarik dengan *file* ini, ketika di cari secara biasa kita tidak akan bias menemukannya, karena *file*nya di super hidden atau di sembunyikan dengan sangat baik, sehingga meskipun pengaturan *show hidden folder*, *file*, *and driver* sudah diaktifkan, *file* tetap tidak muncul dan terlihat, dari itu kita

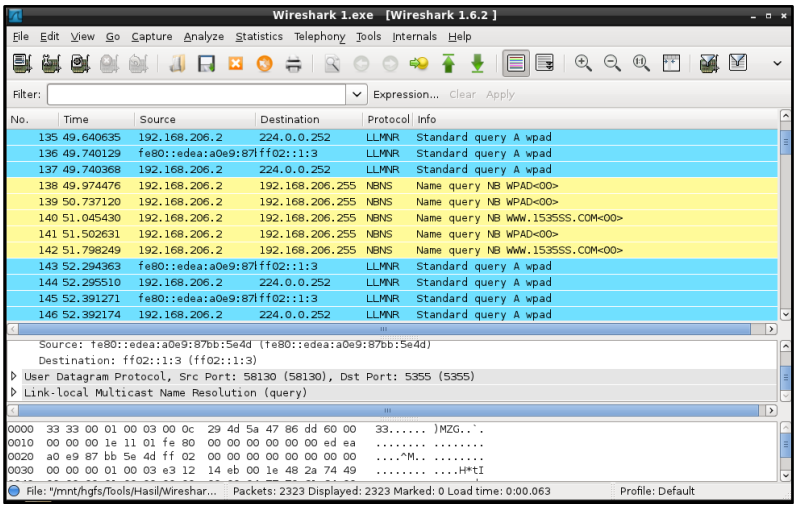
menggunakan trik tambahan untuk menampilkan *file* yang sudah di super *hidden* dengan cara masuk ke *cmd*, ketikan "*dir/ah*" dan enter, seluruh *file* akan tampil semua termasuk yang telah di *hidden*. Selanjutnya kita akan menampilkan *file* tersebut di *windows explorer* karena meskipun sudah tampil di *cmd*, *file* tersebut belum tampil pada *windows explorer*. Cara menampilkannya yaitu dengan menggunakan perintah "*attrib*" dan diiringi dengan nama *file*.



Gambar 9. Menampilkan *file hidden*

5) Monitoring paket data pada jaringan menggunkan *Wireshark*

Hasil monitoring yang dilakukan Wireshark ketika *file Games.exe* dieksekusi pada system opersi windows telah menagkap adanya komunikasi yang dilakukan oleh *IP 192.168.206.2* kepada *192.168..208.225* menggunakan *protocol NBNS* dan meminta layanan ke alamat *www.1533ss.com.*



Gambar 10. Wireshark

Tabel 1. Hasil Analisa dan pengujian perilaku malware menggunakan metode Surface Analysis

No	Tahapan	*Surface Analysis*		
		Tool	Hasil Temuan	
1	*File Attribute analysis*	Pestudio	*File Name*	*Games.exe*
			MD5	*AD84220B21FAA7F98ECC05C504ACD7B7*
			CPU	*32-bit*
			Type	*executable*
			Indicators	4/14
2	*Phackers Check*	Exeinfo	*UPX 0.89 - 3.xx -> Markus & Laszlo ver. [3.95]*	
			unpack "upx.exe -d" from http://upx.sf.net or any UPX/Generic unpacker	
			.text	
			.rdata	
			.data	
3	*Fuzzy Hashing*	Md5sum	Sama	
		ssdeep	100%	
4	*Analisa Strings*	Pestudio	101/272	
			www.1535ss.com	
		Yara dan Strings	*OpenSCManager*	
			WININET.dll	
			http://	
			HTTP	
			ExitProcess	
			CreateProcess	
			InternetOpenUrl	
			Sleep	
			GetSystemDirectory	
5	*Malware Scan*	*www.virustotal.com*	*Worm* dan *Trojan*	

Tabel 2. Hasil Analisa dan pengujian perilaku malware menggunakan metode Runtime Analysis

No	Temuan	Runtime Analysis				
		Regshot	Process Hacker	CaptureBAT	Noriben Sanbox	Wireshark
	Penambahan registry	✓	-	✓	✓	-
	Penambahan file baru pada C:\Windows\system32	✓	-	✓	✓	-
	File hidden	-	-	✓	✓	-
	Alamat IP/web Program : www.1535ss.com	-	-	-	-	✓
	Nomor Port yang digunakan : 8080	-	-	-	-	✓
	Protocol yang digunakan ; NBNS	-	-	-	-	✓
	File yang berjalan dilatar belakang	-	✓	-	-	-

4. KESIMPULAN

Berdasarkan analisa dan pengujian terhadap sampel Games.exe maka didapatlah kesimpulan diantaranya:

1) *File* Games.exe jelas dapat dikatakan sebuah *malware* dikarenakan memiliki strings yang pada umumnya terdapat pada sebuah *malware*.
2) Melakukan penambahan dan membuat sebuah perubahan pada *registry*

3) *File* dapat menggandakan diri ke folder *system32* dengan menjadi *file WinHvqf32.exe* yang dibuktikan dengan kesamaan *MD5* pada *file* tersebut.
4) *File* dapat menyembunyikan dirinya dengan menghapus *file* utama dan melakukan hidden pada *file* duplikat.
5) *File* juga melakukan interaksi pada jaringan dengan melakukan akses ke website *www.1535ss.com*.
6) Dari kesemua tools yang digunakan, Noriben merupakan tool yang paling baik dalam menganalisa sebuah *file*.
7) Dari hasil *scan* yang dilakukan menggunakan virus total menunjukan bahwa md5 dari *file* Games.exe belum tercatat sebagai *malware*.
8) Dapat disimpulkan bahwa *file* Games.exe merupakan sebuah *malware* jenis *Botnet*, *Trojan* dan *Worm* atau *virus* menurut perilakunya.

DAFTAR PUSTAKA

[1] Adenansi, R., dan Novarina, L. A. (2017). "Malware dynamic". *JoEICT (Journal of Education And ICT), 1*(1).

[2] Agung, M. F. (2011). "Pengertian serta penjelasan metode secara umummengenai Malware Analysis". Konsep Dasar Malware Analysis.

[3] Indrajit, R. E., 2012. “Analisa Malware”. E-Artikel Sistem Dan Teknologi Informasi.

[4] Nurdiyanto, W. (2013). “Skenario Kombinasi Tools yang Efektif dalam Analisis Malware”. http://pusdiklat.bps.go.id/index.php?r=artikel/view&id=248. Diakses 20/09/2018.

[5] Perdhana, M. R. (2011). "Analisa Malware *Harmless Hacking*" (Cetakan pertama ed.). Yoyakarta: Graha Ilmu.

[6] Sugiyono (2012). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta.