EKSPLORASI MALWARE POISON IVY MENGGUNAKAN METODE MALWARE ANALISIS DINAMIS DAN MALWARE ANALISIS STATIS

[1] *Darmawan Ramadana (12 1065 1040),*
[2] *Triawan Adi Cahyanto.,M.Kom,* [3] *Victor Wahanggara. M.Kom,*
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Jember
Email: idar@mail.com

ABSTRAK

Tidak dapat dipungkiri pada era teknologi yang semakin berkembang pesat seperti sekarang ini, peranan komputer sangat penting sebagai alat bantu dalam memudahkan pekerjaan manusia sehingga setiap pekerjaan dapat dilakukan secara cepat. Namun, disisi lain pemanfaatan teknologi komputer juga dapat berdampak buruk seperti melakukan tindakan yang dapat merugikan orang lain dengan melibatkan komputer sebagai media dalam melakukan aksinya, dalam hal ini media yang dimaksud adalah salah satu komponen utama dari komputer yaitu perangkat lunak atau software yang bersifat jahat (merugikan) yang sering disebut sebagai malicious software (Malware).

Celakanya penyebaran malware saat ini sangat erat kaitannya dengan tindak kejahatan seperti kegiatan pencurian file serta kemampuan untuk mengendalikan komputer korban dari jarak jauh yang biasanya dilakukan oleh seorang intruder. Berhubungan dengan tindak kejahatan, ada suatu bidang yang menangani tindak kejahatan yang melibatkan komputer, nama bidang tersebut adalah forensik digital. Salah satu tahapan dalam forensik digital yaitu dengan melakukan analisis terhadap barang bukti digital yang dalam hal ini adalah malware. Malware dapat ditangani dengan mengetahui cara kerja ketika melakukan serangan kedalam sistem komputer. Dengan kata lain malware dapat ditangani ketika berhasil dilakukan analisis dan mengetahui informasi yang ditimbulkan oleh malware tersebut.

Kata kunci *: Poison Ivy, RAT, Malware,* Digital Forensik,, *Malware* Analisis Dinamis, *Malware* Analisis Statis

ABSTRACT

It is inevitable in the era of technology which is growing rapidly, as now, the role of computers is very important as a tool in facilitating the human work. so that every job can be done quickly. However, on the other side the use of computer technology can also impact adversely, such as an action that harm others people by involving the computer as a means of crime in the act. In this case the media refers to one of the main components of the computer i.e. software. software that is evil (harm), often called as malicious software (malware).

Unfortunately spread of malware are currently very closely related to acts of crime , such as theft of files and the ability to control the Victim's computer From Remote Computer which is usually done by an intruder. if we talking about crime , there is a field that handles about crime. which utilizing computer as a means. The field name is digital forensics. one of the step in digital forensics is to perform an analysis of digital evidences, which in this case is malware . Malware can be handled by knowing , how does it work when attacking the computer systems. In other words, malware can be handled when successfully carried out the analysis and find out the information generated by the malware.

Keywords *: Poison Ivy, RAT, Malware, Forensic Digital, Dynamic Malware Analysis, Static Malware Analysis*

1. PENDAHULUAN

Dalam era teknologi yang semakin berkembang pesat sekarang ini, komputer digunakan dalam memudahkan pekerjaan manusia, dalam pengoprasiannya ada *software* yang berjalan diatas sistem operasi, dan ini sangat berperan penting dalam melakukan tugas-tugas yang dikerjakan oleh pengguna karena melalui *software* inilah suatu komputer dapat menjalankan perintah sehingga membantu pengguna dalam menyelesaikan pekerjaannya. Namun tidak semua *software* dapat membantu dan memudahkan manusia dalam melakukan pekerjaannya, adapula jenis *software* yang diciptakan untuk melakukan perusakan atau tindak kejahatan yang dapat merugikan orang lain, software tersebut dikategorikan sebagai *Malicous Software*.

Malicous Software atau yang lebih dikenal sebagai *Malware* merupakan perangkat lunak yang secara eksplisit didesain untuk melakukan aktifitas berbahaya atau perusak perangkat lunak lainnya seperti *Trojan, Virus, Spyware* dan *Exploit* (Kramer, S., dan Bradfield, J. C, 2009). *Malware* diciptakan dengan maksud tertentu yaitu melakukan aktifitas berbahaya yang berdampak sangat merugikan bagi para korbannya, antara lain seperti penyadapan serta pencurian informasi pribadi, hingga kasus perusakan sistem yang dilakukan oleh penyusup (*Intruder*) terhadap perangkat korban dengan berbagai alasan. Salah satu media yang digunakan oleh *intruder* untuk mengendalikan komputer pengguna secara diam-diam dari jarak jauh adalah *malware poison ivy*, dikenal sebagai "*trojan access remote*" karena dapat memberikan kontrol penuh kepada *intruder* melalui pintu belakang (*backdoor)*, kemampuan *malware poison ivy* mengadopsi dari software *Remote Administration Tool* (RAT), yaitu termasuk kategori software yang baik (legal) yang dapat melakukan monitoring & pengontrolan secara penuh, contoh penggunaan software RAT ini biasa digunakan

oleh seorang pimpinan perusahaan untuk mengontrol perangkat kerja (komputer) karyawannya melalui jaringan jarak jauh, dengan fitur tersebut tidak jarang *malware poison ivy* dikatakan juga sebagai Software RAT yang ilegal (RAT *Malware*) dikarenakan tidak memberikan informasi berupa *notifikasi* saat proses remote terhubung (terhubung secara diam-diam), dengan *malware* sebagai medianya maka dalam hal ini merupakan sebuah bukti tindak kejahatan digital yang dilakukan oleh seorang *intruder*.

Forensik Digital merupakan disiplin ilmu yang menerapkan investigasi dan identifikasi dalam menindak kejahatan digital (Wahanggara dan Prayudi, 2015). Salah satu tahapan utama dalam menginvestigasi tindak kejahatan yaitu mengumpulkan barang bukti digital. Untuk menemukan barang bukti digital pada *malware*, dibutuhkan analisis lebih mendetail agar dapat mendeteksi aktifitas sebuah *malware* serta mempelajari bagaimana sebuah *malware* menginfeksi dan berkembang dalam sebuah sistem. Ada dua tipe analisis dalam melakukan analisis pada *malware* yaitu dengan analisis statis (analisa kode) dan analisis dinamis (Distler, 2007). Meskipun dari kedua tipe analisis tersebut mempunyai tujuan yang sama yaitu menjelaskan tentang bagaimana sebuah *malware* bekerja namun peralatan, waktu dan kemampuan yang dibutuhkan dalam menganalisa sangatlah berbeda, Analisis Statis melakukan dengan pembongkaran terhadap *source code* dari *malware* lalu mempelajari dan memahami melalui kode tersebut atau dengan kata lain proses analisis tidak memerlukan eksekusi terhadap *malware*, berbeda dengan analisis dinamis yang pada proses analisisnya membutuhkan pengeksekusian terhadap contoh *malware* untuk kemudian dipelajari perilaku yang ditimbulkan oleh *malware* tersebut sehingga dapat diperoleh informasi tentang bagaimana sebuah *malware* tersebut bisa berkembang atau memanipulasi dirinya sendiri, dan pada komponen sistem apa saja *malware* tersebut berkomunikasi. Harapan setelah proses eksplorasi dilakukan semoga bisa memberikan pembelajaran tentang efek yang ditimbulkan oleh *malware* dan membantu praktisi dalam menemukan barang bukti digital.

2. DASAR TEORI

2.1 Poison Ivy RAT (Remote Access Trojan)

Poison Ivy RAT merupakan program yang dapat menghubungkan dan melakukan kontrol secara tersembunyi terhadap satu atau lebih perangkat komputer. Aktifitas *Poison Ivy RAT* dilakukan melalui jaringan, baik itu jaringan *local* maupun jaringan *public* sehingga memungkinkan untuk dilakukan pada jarak yang jauh. *Poison Ivy RAT* menggunakan arsitektur *client server*. Dalam hal ini *server* adalah bagian program yang akan ditanamkan (*backdoor*) dan dijalankan pada perangkat korban yang didalamnya telah diberikan beberapa pengaturan seperti alamat *IP* dan *Port* agar dapat menghubungkan diri pada induk programnya (*calling home*). Induk program yang dimaksud adalah dari sisi *client* yaitu bagian program yang dapat melakukan peggontrolan (perangkat *intruder*). Jika sebuah komputer korban telah terinfeksi oleh program *Poison Ivy RAT* ini maka seorang *inrtruder* dapat melakukan beberapa pengontrolan penuh antara lain seperti, mengakses speaker komputer, mengakses *webcam* untuk merekam audio maupun video, juga dapat digunakan untuk melakukan pencurian password dengan memanfaatkan fitur *Keystroke Logger (KeyLogger)*.

2.2 METODE *MALWARE* ANALISIS

3.2.1.1 Malware Analisis Dinamis

Pada metode ini sebuah file yang diperiksa akan diaktifkan dalam sebuah lingkungan yang *safe* baik pada sebuah mesin fisik yang telah disediakan sebagai lab *malware* maupun yang berupa virtual (mesin *virtual*) untuk selanjutnya mampu dikumpulkan informasi mengenai dampaknya terhadap komputer ketika file *malware* menjalankan prosesnya. Sehingga dapat diketahui kegiatan apa saja yang dilakukan oleh *malware* saat berhasil menginfeksi sebuah komputer. Tahapan dalam analisis dinamis ini akan memeriksa komputer dengan secara keseluruhan seperti porses yang berjalan dikomputer, perubahan *registry*, komunikasi internet dan peristiwa janggal lainnya yang memungkinkan terjadi ketika sebuah komputer telah terinfeksi oleh *malware*.

3.2.1.2 *Malware* Analisis Statis

Tidak seperti pada metode malware analisis dinamis, dalam metode ini file malware tidak akan diaktifkan secara langsung melainkanakan ditelusuri dan diteliti serta dianalisis terhadap kode sumber yang dituliskan didalam program malware dengan melakukan tahapan pembedahan terhadap program malware tersebut, sehingga informasi yang didapatkan sangatlah lengkap dan bisa memberikan gambaran yang sangat detail tentang mekanisme kerja malware tersebut secara keseluruhan. Dalam menggunakan metode malware analisis statis ini dituntut mampu memahami bahasa mesin terutama arsitektur sebuah program karena akan sangat membantu dalam menganalisis susunan kode-kode program malware terkait dengan mengumpulkan informasi dari perilaku yang ditimbulkan oleh malware tersebut.

3. METODE PENELITIAN

3.1 Analisis

3.2.1 Tahapan *Malware* Analisis Dinamis

3.2.1.1 Membangun Virtual Lab

Dalam menganalisa *malware* diperlukan sebuah lingkungan yang aman (*Virtual Lab*), dimana peneliti dapat dengan bebas melakukan analisa terhadap *malware*, tanpa harus khawatir *malware* tersebut akan menyebar dan menimbulkan kerusakan terhadap komputer. *Virtual Lab* yang dimaksud dalam penelitian ini adalah sebuah mesin virtual yang didalamnya sudah terinstal berbagai macam *tools* yang diperlukan untuk kegiatan analisa. Program untuk mesin *virtual* yang digunakan dalam penelitian ini adalah *Virtualbox*.

Pengaturan pada mesin virtual untuk kegiatan menganalisis *malware* meliputi sistem operasi yang digunakan serta seluruh konfigurasinya, termasuk pertimbangan untuk mampu terhubung dengan jaringan

serta adanya sambungan dengan perangkat fisik seperti hard disk dan lainnya.
Sistem Operasi yang akan digunakan dalam penelitian ini adalah *Windows XP* karena sangat mudah untuk terinfeksi oleh *malware* sehingga sesuai untuk digunakan dalam kegiatan analisis *malware*. Lingkungan sistem operasi dikonfigurasi sedemikian rupa untuk mengakomodasi kegiatan analisis *malware*. Konfigurasi yang dimaksud adalah pengaturan terhadap sistem operasi yang dilakukan sesuai kebutuhan, dalam hal ini yaitu tidak dipasang program *antivirus* dan juga pertimbangan akan penggunaan *firewall*.
Dengan penggunaan *virtual lab* memungkinkan untuk kegiatan analisis *malware* dilakukan dilingkungan komputer seperti pada keadaan yang nyata namun dengan resiko yang hampir tidak ada karena mesin *virtual* telah diatur untuk tidak memberikan pengaruh terhadap komputer utama.

3.2.1.2 Menjalankan Malware
Dalam tahap ini dilakukan pengujian dengan menjalankan sampel file malware (Poison Ivy) pada virtual lab, sehingga dapat menghasilkan informasi mengenai perilaku apa saja yang dilakukan oleh malware terhadap sistem ketika file tersebut dijalankan.

3.2.1.3 Analisis Perilaku Malware
Dalam proses analisis akan diperiksa secara keseluruhan proses yang berjalan pada komputer seperti, perubahan *registry*, aktivitas komunikasi jaringan dan peristiwa janggal lainnya yang terjadi ketika komputer telah terinfeksi oleh *malware*.
a. Proses analisis terhadap perubahan pada sistem *registry* menggunakan program pendukung *regshot*, yang mana dengan program *regshot* ini peneliti akan melakukan analisis pada sistem *registry* dengan cara membandingkan *snapshot* dari *registry* sebelum *malware* diaktifkan dan *snapshot* dari *registry* setelah program *malware* diaktifkan sehingga akan dapat diketahui perbedaan dan aktifitas apa saja yang telah dilakukan oleh *malware* terhadap perubahan sistem *registry*.
b. Wireshark dalam penelitian ini digunakan untuk menganalisa kinerja jaringan, tujuannya agar didapatkan informasi mengenai kemungkinan adanya indikasi yang ditimbulkan oleh perilaku malware terhadap sistem jaringan.

3.2.1.4 Analisis Malware Otomatis (Cuckoo Sandbox)
Untuk lebih menguatkan hasil dari temuan perilaku *malware* sebelumnya dimana file *malware* dijalankan pada *virtual lab* maka pada tahap ini dilakukan analisis menggunakan program yang dapat melakukan analisis perilaku *malware* secara otomatis yaitu menggunakan *Cuckoo Sandbox,* program tersebut akan menyajikan informasi aktifitas terhadap *malware* yang sedang dianalisis antara lain seperti :
- File apa saja yang dibuat *malware*
- File apa saja yang dihapus malware
- File apa saja yang diunduh malware
- Aktifitas malware pada memori
- Trafik jaringan yang diakses malware.

3.2.2 Tahapan Malware Analisis Statis
3.2.2.1 Ekstraksi File Malware
Pada tahap ini dilakukan ekstraksi terhadap file *malware* kedalam bentuk kode *String* menggunakan bantuan software *program strings kali linux* untuk kemudian dapat dilakukan analisis terhadap kode-kode tersebut.

3.2.2.2 Analisis Perilaku Kode
Tujuan lebih lanjut dalam penelitian ini juga diharapkan dapat memberikan output berupa hasil pengujian apakah dapat dibuktikan bahwa file dari program poison ivy merupakan suatu malware atau bukan, untuk itu dibutuhkan sentuhan teknik Static Malware Analysis (analisis statik) yang difokuskan pada pencarian dan analisis terhadap kode string yang mengandung perilaku ataupun ciri dari program poision ivy.

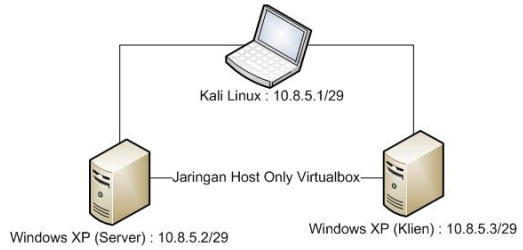
3.2.2.3 *Disassembler*
Disassembler adalah program komputer yang dapat melakukan konversi terhadap bahasa mesin menjadi bahasa yang lebih mudah dipahami oleh manusia. Dengan *disassemble,* pada penelitiian ini akan dilakukan analisis terhadap *malware* dan mencoba utnuk memahami *malware* dengan menganalisis bahasa *assembly* dan mengumpulkan informasi dari program *malware* yang dapat digunakan untuk mengidentifikasi komponen maupun karakteristik *malware*.

3.2.3 Hasil Pengujian dan Analisis
Tahap ini mengumpulkan hasil temuan dari tahapan pengujian dan analisis untuk kemudian dilakukan perbandingan terhadap informasi perilaku *malware*, baik yang didapatkan dengan cara mengeksekusi *malware* secara langsung (Analisis *Malware* Dinamis) maupun yang dilakukan dengan mengamati kode dari file *malware* (Analisis *Malware* Statis). Perbandingan yang dimaksud dalam penelitian ini bukan membandingkan kinerja dari kedua metode yang digunakan, melainkan mencari dan melakukan

pembuktian terhadap kemiripan output yang dihasilkan oleh kedua metode tersebut sehingga dapat dipastikan kebenaran atas perilaku yang telah ditimbulkan oleh *malware*.

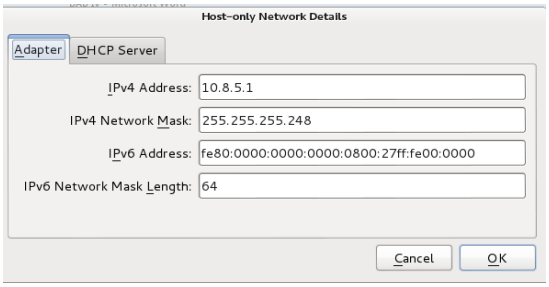
4. PEMBAHASAN
Dalam menganalisis program *malware*, diperlukan tahap pengujian yang dapat digunakan sebagai acuan dalam menentukan karakteristik dan menggali informasi terkait dari perilaku yang akan ditimbulkan oleh program *malware* tersebut.

4.1. Pengujian dan Analisis
4.1.1 Analisis Malware Dinamis
melakukan pengaturan alamat IP jaringan pada *virtual* lab yang akan dibangun. Gambar 4.1 menggambarkan topologi dalam arsitektur jaringan *virtual* lab.

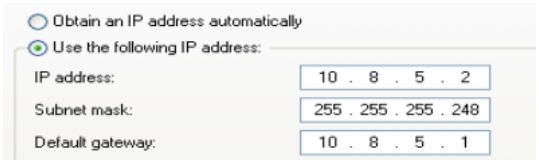


Gambar 4.14 Topologi Arsitektur Jaringan Virtual Lab

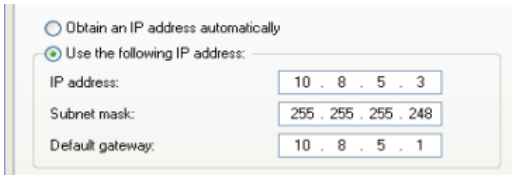
Pengaturan alamat IP pada interface kartu jaringan host only, yang mana dalam kebutuhan komputer utama (Kali Linux) dapat dilakukan pengisian alamat IP pada interface vboxnet0, sedangkan pada tiap virtual lab baik server maupun klien dapat dilakukan pada menu setting, sub menu network dan interface diarahkan pada "Adapter1 - Host only adapter", untuk kemudian dilakukan pengisian alamat IP pada saat virtual lab sudah dijalankan.



Gambar 4.15 Pengaturan Alamat IP *Vboxnet* (*Kali Linux*)



Gambar 4.16 Pengaturan Alamat IP Komputer Server

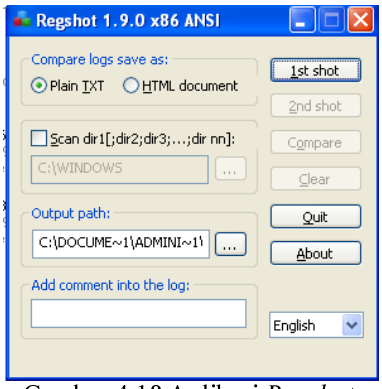


Gambar 4.17 Pengaturan Alamat IP Komputer Klien

a. Analisis Perilaku Menggunakan *Regshot*.

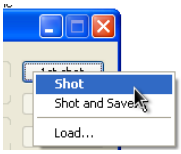
Setelah *Virtual Lab* berhasil dibangun, maka pada tahapan selanjutnya dapat dilakukan analisa langsung terhadap program *malware poison ivy,* pada langkah ini program *malware poison ivy* akan diaktifkan secara langsung pada *virtual lab* sehingga program *malware* akan mencoba untuk menginfeksi sistem. Namun sebagai langkah awal dalam tahap analisis ini diperlukan gambaran dari kondisi sistem pada saat dalam keadaan normal (belum terinfeksi) menggunakan alat pendukung *Regshot*.

Regshot bekerja dengan cara melakukan *snapshot* pada sistem *Windows* sebanyak dua kali. *Snapshot* yang pertama diambil sebelum *malware* diaktifkan pada sistem dan *snapshot* kedua diambil setelah *malware* diaktifkan dan berhasil menginfeksi sistem. Berikut adalah tampilan aplikasi regshot ketika berjalan pada *Windows XP*.



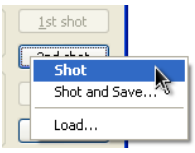
Gambar 4.18 Aplikasi *Regshot*

Sebelum menjalankan malware pada sistem Windows XP, hal yang perlu dilakukan adalah melakukan pengambilan snapshot yang pertama dengan cara menekan tombol 1st shot, maka regshot akan memulai proses snapshot.



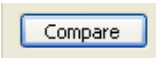
Gambar 4.19 *Menu Regshot* pada 1st shot.

Regshot akan mendata semua file, ketika proses snapshot yang pertama selesai maka untuk selanjutnya dapat dilakukan langkah mengaktifkan program malware sehingga program malware tersebut dapat melakukan beberapa perubahan terhadap sistem. Pada saat program malware telah melakukan perubahan maka diperlukan untuk mengambil snapshot yang kedua untuk mendapatkan informasi sistem apa saja yang telah berubah.



Gambar 4.21 Pengambilan shot kedua

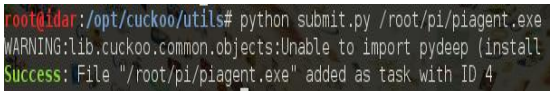
Setelah pengambilan shot pertama dan shot kedua pada *regshot* maka dapat dilakukan komparasi data antara kedua *snapshot* yang telah dilakukan pada langkah sebelumnya dengan memanfaatkan *menu compare*.



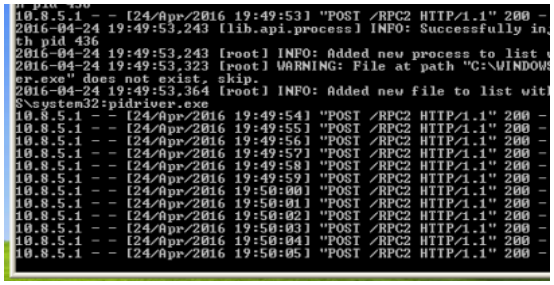
Gambar 4.22 *Menu Compare* pada *Regshot*.

b. Analisis Perilaku Menggunakan *Cuckoo Sandbox*

Dapat dilihat program dari sisi klien (*cuckoo sandbox agent*) mesin *cuckoo sandbox* yang sedang malakukan analisis.



Gambar 4.34 Melakukan eksekusi program *malware* pada mesin *cuckoo sandbox*



Gambar 4.35 Analisis *Malware Cuckoo Sandbox Agent*

Perintah untuk mengaktifkan *service bottle server* pada mesin *cuckoo sandbox* sehingga informasi dari hasil analisis dapat disajikan dalam bentuk *visual* berbasis *web*.

Gambar 4.36 Perintah Mengaktifkan *Bottle Server Cuckoo Sandbox*

Penjelaskan informasi detail *file* dari program *malware* yang telah dianalisis yaitu piagent.exe (*poison ivy*).

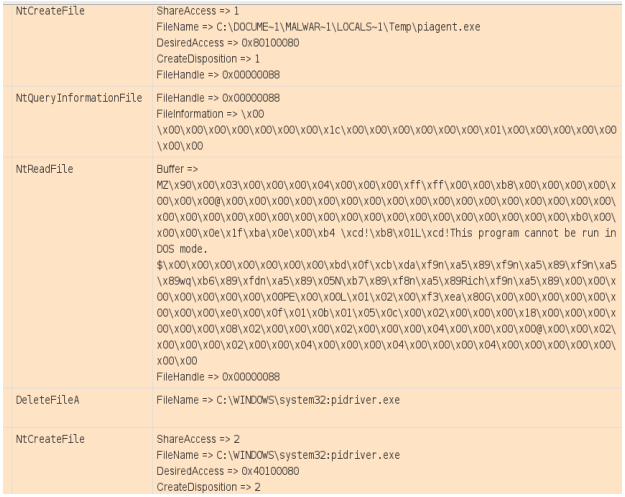
File name	piagent.exe
File size	7168 bytes
File type	PE32 executable (GUI) Intel 80386, for MS Wi
CRC32	2C15AF8B
MD5	e701b565cc14413f36f2a4ece7aae505
SHA1	8e1d35a4e3dec78d13e8059d315343daa01edef3
SHA256	a415c4b5cc25f98599a83fdab4dc276499ce6585401b
SHA512	a4163109b53ecfb0c691924c5537b2037639f5a7894a
Ssdeep	None
PEiD	None matched
Yara	None matched
VirusTotal	Permalink

Gambar 4.37 Informasi Detail *File* Program PIAGENT.EXE

Penjelaskan kesimpulan perilaku yang dilakukan oleh program *malware poison ivy* terhadap mesin *cuckoo sandbox* diantaranya yaitu, membuat 2 file baru dengan nama "piagent.exe" dan sebuah file dengan nama "pidriver.exe", adapula perilaku lainnya program *malware poison ivy* mencoba menandai dirinya ketika aktif dalam memori dengan membuat *mutex* (*Mutual Exclusion*) bernama "**)!VoqA.I4**". Dan perilaku yang lain dari program *malware poison ivy* yaitu melakukan penambahan *value key* pada "*startup*" dengan tujuan agar program dapat aktif pada saat komputer pertama kali melakukan *booting*.

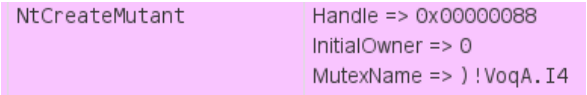
Gambar 4.38 Kesimpulan Perilaku Dari Program *Malware Poison Ivy* Terhadap Mesin *Cuckoo Sandbox*

Pada gambar berikut ini menjelaskan detail informasi yang berhasil dideteksi oleh mesin *cuckoo sandbox* terhadap perilaku program *malware poison ivy* dalam upaya pembuatan *file* "piagent.exe" dan *file* "pidriver.exe"



Gambar 4.39 Upaya pembuatan *File* yang Dilakukan Program *Poison Ivy*

Penjelaskan detail informasi yang berhasil dideteksi oleh mesin *cuckoo sandbox* terhadap perilaku program *malware poison ivy* dalam upaya pembuatan *Mutual Exclusion (Mutex)*.



Gambar 4.40 Upaya Pembuatan *Mutual Exclusion* yang Dilakukan Program *Poison Ivy*

Pada gambar 4.41 menjelaskan detail informasi yang berhasil dideteksi oleh mesin *cuckoo sandbox* terhadap perilaku program *malware poison ivy* dalam upaya penambahan *value key registry*, dimana program *malware* berusaha membuka subkey pada "SOFTWARE\Microsoft\Windows\CurrentVersion\Run" dan upaya memberikan nilai terhadap *value* "secret_agent" dengan nilai "C:\Windows\system23:pidriver.exe", sehingga program *malware poison ivy* akan berjalan dalam *startup* komputer.

RegOpenKeyExA	Handle => 0x00000088 Registry => 0x80000002 SubKey => SOFTWARE\Microsoft\Windows\CurrentVersion\Run
RegSetValueExA	Handle => 0x00000088 Buffer => C:\WINDOWS\system32:pidriver.exe\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 ValueName => secret_agent Type => 1
RegCloseKey	Handle => 0x00000088

Gambar 4.41 Upaya Penambahan dan Perubahan *File Registry*

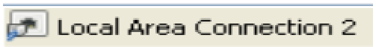
Pada gambar 4.42 menjelaskan detail informasi yang berhasil dideteksi oleh mesin *cuckoo sandbox* terhadap perilaku program *malware poison ivy* dalam upaya melakukan koneksi jaringan. Terlihat dimana program berusaha menyiapkan koneksi dengan memanggil instruksi *socket*.

WSAStartup	VersionRequested => 0x00000101
socket	type => 1 protocol => 0 af => 2
connect	socket => 0x00000080
closesocket	socket => 0x00000080

Gambar 4.42 Upaya Koneksi Jaringan yang Dilakukan Program *Poison Ivy*

c. Analisis Paket Jaringan Program *Malware Poison Ivy* Menggunakan *Wireshark*.
Pada tahap ini akan dilakukan dua kali pengujian langsung dengan mengaktifkan program *malware poison ivy* terhadap dua perangkat komputer virtual *windows* yang telah dirancang sebelumnya, dimana pada sisi komputer *server* akan ditanamkan program *malware* yang dapat menginfeksi sistem serta menjadi pelayan (*service*) terhadap komputer klien yang melakukan *request*, tentunya pada komputer klien ini telah terinstal program *client malware poison ivy*. Kemudian dilakukan beberapa aktifitas sehingga dapat dianalisis paket data yang berjalan dalam jaringan dengan memanfaatkan program *wireshark*, pada percobaan 1 akan dilakukan menggunakan *port default* dari program *malware poison ivy* yaitu *port 3460*.

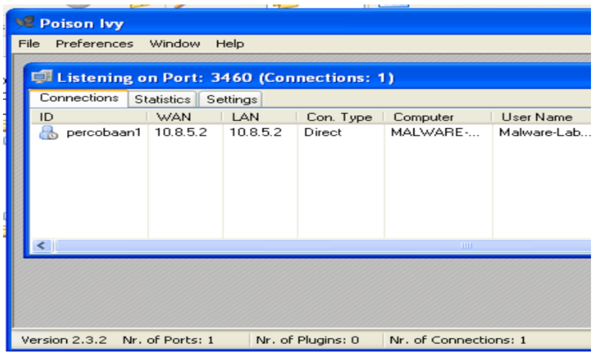
Adapun pengaturan awal pada *wireshark* yaitu pemilihan kartu jaringan pada daftar *interface* program *wireshark*. Kartu jaringan yang akan dianalisis adalah *NIC (Network Interface Card)* dari mesin *virtual* yang terdeteksi dengan nama "*Local Area Connection 2*" seperti digambarkan pada gambar 4.43.



Gambar 4.43 Daftar Kartu Jaringan *Wireshark*.

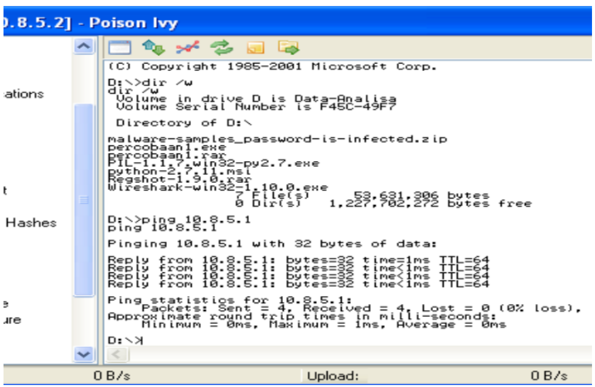
Untuk memulai analisis paket jaringan menggunakan *wireshark* dapat digunakan tombol *start* pada *menu "capture > start"*, maka program *wireshark* akan mulai melakukan *sniffing* (merekam aktifitas jaringan) secara *real time*, dengan ini "percobaan1" dimulai.

Disisi lain pada gambar 4.44 menggambarkan ketika komputer *server* (komputer yang tertanam *poison ivy*) berhasil terkoneksi dengan komputer klien (pengontrol) dengan nama id "percobaan1".



Gambar 4.44 Program *poison ivy* terkoneksi dengan program induk.

Pada gambar 4.44 menggambarkan aktifitas pengontrolan program *poison ivy* (*remote shell*) terhadap komputer *server* dengan melakukan *ping* pada alamat ip komputer utama (komputer *kali linux*) yang dilakukan oleh komputer klien agar, program *wireshark* dapat merekam informasi paket data yang dilalui selama aktifitas tersebut berlangsung seperti yang dapat dilihat pada gambar 4.46.



Gambar 4.45 *Remote Shell* Komputer *Server* Oleh Komputer Klien

Ditemukan pendefinisian kode *string* "secret_agent" pada offset 004016D3 sampai offset 004016DE.

```
.data:004016D3                db   73h
.data:004016D4                db   65h
.data:004016D5                db   63h
.data:004016D6                db   72h
.data:004016D7                db   65h
.data:004016D8                db   74h
.data:004016D9                db   5Fh
.data:004016DA                db   61h
.data:004016DB                db   67h
.data:004016DC                db   65h
.data:004016DD                db   6Eh
.data:004016DE                db   74h
```

Gambar 4.55 Temuan Kode *String* "secret_agent" pada *IDA Pro*

Keterangan konversi dari gambar 4.55 :

- 5F (Hexa) = 95 (Decimal) = _ (ASCII)
- 61 (Hexa = 97 (Decimal) = a (ASCII)
- 63 (Hexa) = 99 (Decimal) = c (ASCII)
- 65 (Hexa) = 101 (Decimal) = e (ASCII)
- 67 (Hexa) = 103 (Decimal) = g (ASCII)
- 6E (Hexa) = 110 (Decimal) = n (ASCII)
- 72 (Hexa) = 114 (Decimal) = r (ASCII)
- 73 (Hexa) = 115 (Decimal) = s (ASCII)
- 74 (Hexa) = 116 (Decimal) = t (ASCII)

Hasil Hexa : 73 65 63 72 65 74 5F 61 67 65 6E 74
Hasil Decimal : 115 101 99 114 101 116 95 97 103 101 110 116
Hasil ASCII : secret_agent

4.2. Hasil Temuan pada Pengujian dan Analisis

Pada tabel 4.2 menampilkan hasil temuan secara keseluruhan dari pengujian dan analisis yang telah dilakukan, baik dengan teknik analisis dinamis maupun teknik analisis statis terhadap program *malware poison ivy*. Penyajian hasil temuan dilakukan bertujuan guna mendapatkan kebenaran informasi yang dihasilkan dari kedua teknik / metode tersebut terkait perilaku program *malware Poison Ivy*.

NO	TEMUAN	ANALISIS DINAMIS			ANALISIS STATIS	
		REGSHOT	CUCKOO SANDBOX	WIRESHARK	STRINGS KALI LINUX	IDA DISASSEMBLER
1.	PENAMBAHAN REGISTRY : HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\SECRET_AGENT	DITEMUKAN	DITEMUKAN	-	DITEMUKAN	DITEMUKAN
2.	PENAMBAHAN FILE PREFETCH : C:\WINDOWS\PREFETCH\PIAGENT.EXE-0AEBFBEE.PF	DITEMUKAN	-	-	-	-
3.	PENAMBAHAN FILE BARU : C:\DOCUME~1\USER\LOCALS~1\TEMP\PIAGENT.EXE	-	DITEMUKAN	-	-	-
4.	PENAMBAHAN FILE BARU : C:\WINDOWS\SYSTEM32:PIDRIVER.EXE	-	DITEMUKAN	-	DITEMUKAN	-
5.	ALAMAT IP PROGRAM INDUK (*CONTROLLER*) : 192.168.56.20	-	-	DITEMUKAN	DITEMUKAN	DITEMUKAN
6.	NOMOR PORT UNTUK JALUR KOMUNIKASI : 3460	-	-	DITEMUKAN	-	-
7.	PROTOKOL YANG DIGUNAKAN DALAM PENGIRIMAN PAKET DATA : TCP (TRANSMISSION CONTROL PROTOCOL)	-	-	DITEMUKAN	-	-

Tabel 4.2 Hasil Temuan dari Pengujian dan Analisis Program *Malware Poison Ivy*

Tabel 4.2 (Lanjutan) Hasil Temuan dari Pengujian dan Analisis Program *Malware Poison Ivy*.

NO	TEMUAN	ANALISIS DINAMIS			ANALISIS STATIS	
		REGSHOT	CUCKOO SANDBOX	WIRESHARK	STRINGS KALI LINUX	IDA DISASSEMBLER
8.	KODE STRING MUTUAL EXCLUSION (PEMBUATAN MUTEX) :)!VOQA.I4	-	DITEMUKAN	-	DITEMUKAN	-
9.	KODE STRING (NAMA IDENTITAS/ ID) : PI_AGENT	-	-	-	DITEMUKAN	DITEMUKAN
10.	KODE STRING PASSWORD AUTENTIKASI : ADMIN	-	-	-	DITEMUKAN	DITEMUKAN

Keterangan : - (lambang *minus*) = Tidak ditemukan.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian dan analisis terhadap program *poison ivy* yang telah dilakukan maka penulis dapat menyimpulkan beberapa hal sebagai berikut :

1. Program *Poison Ivy* jelas dapat dikatakan sebagai *malware* karena mempunyai beberapa karakteristik dari program *malware* pada umumnya yaitu melakukan penambahan dan perubahan terhadap sistem (*Windows Registry* dan *file prefetch*) sebagaimana ditemukan seperti kode *string* **"secret_agent"** dan **"PIAGENT.EXE-0AEBFBEE.pf"** serta perilaku ketika program *poison ivy* diaktifkan tidak memberikan informasi maupun aktifitas secara kasatmata melainkan dalam perilakunya program *poison ivy* berupaya untuk menghubungkan diri pada program induknya yang dilakukan pada proses *background* (tidak kasatmata). Selain itu dari sisi klien (*controller*) program *poison ivy* dapat melakukan pengontrolan penuh terhadap komputer yang terinfeksi melalui komunikasi jaringan tanpa melakukan prosedur autentikasi secara *legal*.
2. Program *poison ivy* dapat dianalisis dengan baik menggunakan dua metode analisis *malware* yaitu metode analisis *malware* dinamis yang dapat memberikan solusi dalam menganalisis program *malware* yang terkendala pada bagian-bagian kode *signature* bersifat polimorfik maupun yang terenkripsi terkait pencarian perilaku dari program *malware* dan metode yang kedua adalah metode analisis *malware* statis dimana metode ini memungkinkan temuan informasi program *malware* melalui kode-kode *hexa* dan *string* ataupun *binary* yang terkandung didalamnya yang tidak dapat ditemukan jika dilakukan dengan metode analisis *malware* dinamis.

Laporan disajikan dalam bentuk runtutan langkah-langkah dimulai dari *pre-testing,* pengujian serta analisis dan diakhiri dengan pengumpulan hasil temuan yang disajikan dalam bentuk tabel dari kedua metode analisis *malware* yang digunakan.

5.2 Saran

Berdasarkan pengalaman dalam proses penelitian ini terdapat beberapa saran yang diusulkan oleh penyusun terkait untuk penelitian lebih lanjut :

Menurut peneliti dari kedua metode yang digunakan dalam penelitian ini, metode analisis *malware* statis merupakan model kajian yang paling sulit dilakukan karena sifatnya yang melibatkan proses melihat dan mempelajari isi program (*white box*) yang sedang dianalisis, untuk itu peneliti menyarankan untuk mempersiapkan strategi yang lebih mendalam pada kajian metode ini khususnya pada sumber daya manusia (SDM) yang harus memiliki pengetahuan dan pengalaman dalam membaca program berbahasa mesin (*assembly language*). Selain itu karena m*alware* merupakan topik yang masih sangat terbuka luas maka peneliti juga menyarankan pengembangan teknik analisis program *malware* dengan memanfaatkan sub-teknik analisis statis yang dikenal dengan nama *Reverse Engineering* (Perdhana, 2011).

REFERENSI

Ari Nugroho, Prayudi, 2015, Penggunaan Teknik Reverse Engineering Pada Malware Analysis Untuk Identifikasi Serangan Malware, Makalah KNSI 2014.

Distler, D. 2007. Malware Analysis: An Introduction. Jurnal of SANS Institute. December, 2007.

Indriantono, N. Supomo B., (1999), Methodologi Penelitian Bisnis untuk Akuntansi & Manajemen, Yogyakarta, BPFE.

(IANA, 2016) http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml (diakses 13 Mei 2016).

Kramer, S., and Bradfield, J. C. (2009). A general definition of malware. Journal in Computer Virology. 6(2), 105-114. Doi:10.1007/s11416-009-0137-1

Kurniawan, 2012, Network Forensics: Panduan analisis & Investigasi Paket Data Jaringan menggunakan Wireshark, ANDI.

R. Perdhana, 2011, Harmless Hacking: Malware Analysis dan Vulnerability Development, GrahaIlmu.

Wahanggara dan Prayudi, 2015, Sistem Deteksi Malicious Software Berbasis System Call untuk Klasifikasi Barang Bukti Digital Menggunakan Metode Support Vector Machine, Seminar Teknologi dan Rekayasa ISSN.

Yusirman S, Prayudi dan Riadi, 2015, Implementation of Malware Analysis using

Static and Dynamic Analysis Method, International Journal of Computer Applications Vol. 117. No.6.

(Yovisto, 2015) http://blog.yovisto.com/fred-cohen-and-the-first-computer-virus/ (diakses 10 Januari 2016).

(_____, 2005) http://www.ascii-code.com/ (diakses 20 Mei 2016).