Jurnal Jaringan Komputer dan Keamanan
JJKK, Vol. 01, No. 01, Februari 2020: 14-24
Received: 11 Des 2019; Revised: 5 Jan 2019; Accepted: 3 Feb 2020

Studi Literatur Analisis Malware Menggunakan Metode Analisis Dinamis dan Statis

Rezki Syaputra, Syaifudin

Teknik Informatika, STMIK GI MDP, Palembang
Teknologi Informasi, Universitas Muhammadiyah Palembang
email : riski.putra_46@mdp.ac.id, Syapuddin@gmail.com
Jl. Rajawali No. 14, Palembang 30113, Indonesia

Abstract

Malware is software or software created to infiltrate or damage computer systems. The spread of malware today is so easy either through a USB flash drive, certain advertisements on websites, and other media. Everything is very closely related to crime such as theft of files, credit cards, internet banking and so forth. In this regard, there is a field that deals with crime namely digital forensics. One of the stages in digital forensics is analyzing digital evidence, in this case malware. To prove a piece of software, it is said that malware is knowing how the program works on a computer system. The Dynamic and Static Malware Analysis Method is a combination of methods that are suitable for analyzing how malware works. Based on an analysis of the workings of malware (poison ivy), it can be concluded that there are several signatures, filenames, and strings that have been investigated in fact can be able to log in remotely without being noticed by the computer owner.

Kata kunci: *Digital Forensics, Malware Analysis, Dynamic Analysis, Static Analysis*

Abstrak

Malware merupakan perangkat lunak atau software yang diciptakan untuk menyusup atau merusak sistem komputer. Penyebaran malware saat ini begitu mudah baik melalui usb flashdisk, iklan-iklan tertentu pada website, dan media lainnya. Semuanya sangat erat kaitannya dengan tindak kejahatan seperti pencurian file, kartu kredit, internet banking dan lain sebagainya. Berkaitan dengan hal itu, ada suatu bidang yang menangani tindak kejahatan yaitu forensik digital. Salah satu tahapan dalam forensik digital yaitu melakukan analisis terhadap barang bukti digital, dalam hal ini adalah malware. Untuk membuktikan suatu software dikatakan malware adalah dengan mengetahui cara kerja program tersebut pada sistem komputer. Metode Malware Analisis Dinamis dan Statis merupakan kombinasi metode yang sesuai untuk menganalisa cara kerja malware. Berdasarkan analisa tentang cara kerja malware (poison ivy), dapat disimpulkan bahwa terdapat beberapa signature, filename, dan string yang sudah diteliti ternyata dapat melakukan proses login secara remote tanpa diketahui oleh pemilik komputer.

Kata kunci: *Forensik Digital, Malware Analysis, Dynamic Analysis, Static Analysis*

1 PENDAHULUAN

Dalam era teknologi yang semakin berkembang pesat saat ini, komputer digunakan untuk memudahkan pekerjaan manusia, dalam pengoperasiannya ada software yang berjalan diatas sistem operasi, dan 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, ada pula 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 & Bradfield, 2010). 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) yangdapat melakukan monitoring dan 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 (T. A. Cahyanto & Prayudi, 2014). Salah satu tahapan utama dalam menginvestigasi tindak kejahatan yaitu mengumpulkan barang bukti digital (Firmansyah, R., Akbar, M., & Negara, E. S. 2019). 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 (T. Cahyanto, 2015; T. A. Cahyanto, Oktavianto, & Royan, 2013). Ada dua tipe analisis dalam melakukan analisis pada malware yaitu dengan analisis statis (analisa kode) dan analisis dinamis (Gandotra, Bansal, & Sofat, 2014; Sikorski & Honig, 2013; Tzermias, Sykiotakis, Polychronakis, & Markatos, 2011). 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 dilakukan dengan membongkar terhadap source code dari malware lalu mempelajari dan memahami melalui kode tersebut atau dengan kata lain proses analisis tidak

memerlukan eksekusi terhadap malware (Moser, Kruegel, & Kirda, 2007; Tzermias et al., 2011). 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 (Bayer, Kirda, & Kruegel, 2010; Bayer, Moser, Kruegel, & Kirda, 2006; Education, Science, Sujyothi, & Acharya, 2017; Egele, Scholte, Kirda, & Kruegel, 2012). Harapan setelah proses eksplorasi dilakukan semoga bisa memberikan pembelajaran tentang efek yang ditimbulkan oleh malware dan membantu praktisi dalam menemukan barang bukti digital.

2 TINJAUAN PUSTAKA

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 (FireEye, 2014). 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 pengontrolan (perangkat intruder). Jika sebuah komputer korban telah terinfeksi oleh program Poison Ivy RAT ini maka seorang intruder 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 Analisis Malware Dinamis

Pada metode ini sebuah file yang diperiksa akan diaktifkan dalam sebuah lingkungan yang safe baik pada sebuah mesin fisik yang telah disediakan sebagai laboratorium 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 proses yang berjalan di komputer, perubahan registry, komunikasi internet dan peristiwa janggal lainnya yang memungkinkan terjadi ketika sebuah komputer telah terinfeksi oleh malware.

Teknik analisis dinamic:

1. Monitoring Function Call

   Monitoring ini merupakan panggilan yang dikontrol oleh subroutine, setelah dieksekusi melewati control eksekusi kemudian kembali ke instruksi pada program utama. Seluruh proses akan dipantau oleh program yang membantu untuk menganalisis perilaku. Fungsi hook ini bertanggung jawab melaksanakan fungsi analisis seperti menganalisis parameter input.

2. Analysis of Function Parameter

   Teknik ini sangat penting dalam analisis dinamis karena analisis ini memantau nilai yang sebenarnya. Output dari sistem call CreateFile digunakan sebagai input ke WriteFile. Fungsi ini menggelompokkan menjadi set logis yang menyediakan informasi rinci tentang perilaku program.

3. Information Flow Tracking

   Pendekatan utama fungsi panggilan pemantauan pelaksanaan program adalah menganalisis tentang bagaimana program bekerja pada data. Teknik ini merupakan metodologi inti yang digunakan oleh tools analisis dinamis yang bekerja pada berbagai tingkatan sistem operasi.

2.3 Malware Analisis Statis

Tidak seperti pada metode malware analisis dinamis, dalam metode analisis statis ini file malware tidak akan diaktifkan secara langsung melainkan ditelusuri dan diteliti serta dianalisis terhadap kode sumber yang dituliskan didalam program malware dengan melakukan tahapan pembedahan terhadap program malware.

Proses memeriksa biner yang diberikan tanpa mengeksekusi sebagian besar dilakukan secara manual. Sebagai contoh, jika kode sumber tersedia beberapa informasi menarik, seperti struktur data, fungsi yang digunakan dan grafik panggilan dapat diekstraksi. Informasi ini hilang setelah kode sumber telah dikompilasi menjadi biner yang dapat dieksekusi dan dengan demikian menghambat analisis lebih lanjut. Dalam domain malware biasanya yang terakhir adalah kasusnya, karena kode sumber dari biner malware saat ini biasanya tidak tersedia. Berbagai teknik digunakan untuk analisis malware statis. Beberapa dari mereka dijelaskan di bawah ini :

1. File fingerprinting: Di samping memeriksa fitur eksternal yang jelas dari biner ini termasuk operasi pada tingkat file seperti perhitungan hash kriptografi (misalnya, md5) dari biner untuk membedakannya dari orang lain dan untuk memverifikasi bahwa itu belum dimodifikasi .

2. Ekstraksi string berkode keras: Perangkat lunak biasanya mencetak output (misalnya, pesan status atau pesan kesalahan), yang akhirnya disematkan dalam biner yang dikompilasi sebagai teks yang dapat dibaca. Memeriksa string tertanam ini sering memungkinkan kesimpulan untuk ditarik tentang internal dari biner yang diinspeksi.

3. Format file: Dengan memanfaatkan metadata dari format file yang diberikan tambahan, informasi yang berguna dapat dikumpulkan. Ini termasuk nomor ajaib pada sistem UNIX untuk menentukan jenis file serta membedah informasi dari format file itu sendiri. Misalnya dari biner Windows, yang biasanya dalam format PE (dapat dieksekusi portabel) banyak informasi yang dapat diekstraksi, seperti waktu kompilasi, fungsi yang diimpor dan diekspor serta string, menu, dan ikon.

4. Pemindaian AV: Jika biner yang diperiksa merupakan malware yang terkenal, sangat mungkin terdeteksi oleh satu atau lebih pemindai AV. Untuk menggunakan satu atau lebih AV scanner memakan waktu tetapi kadang-kadang menjadi kebutuhan.

5. Deteksi Packer: Saat ini malware sebagian besar didistribusikan dalam bentuk yang tidak jelas misalnya, dienkripsi atau dikompresi. Ini dicapai dengan menggunakan pengemas, sedangkan algoritma arbitrer dapat digunakan untuk modifikasi. Setelah pengemasan program terlihat sangat berbeda dari perspektif analisis statis dan logikanya serta metadata lainnya dengan demikian sulit untuk dipulihkan. Meskipun ada pembongkar tertentu, seperti PEiD2, tidak ada pembongkar generik, yang menjadikan ini sebagai tantangan utama dari analisis malware statis.

6. Disassembly: Bagian utama dari analisis statis biasanya adalah pembongkaran dari biner yang diberikan. Ini dilakukan dengan menggunakan alat, yang mampu membalikkan kode mesin ke bahasa assembly, seperti IDA Pro. Berdasarkan kode assembly direkonstruksi seorang analis kemudian dapat memeriksa logika program dan dengan demikian memeriksa niatnya. Biasanya proses ini didukung oleh alat debug seperti OllyDbg.

7. Keuntungan utama dari analisis malware statis adalah memungkinkan analisis komprehensif dari biner yang diberikan. Artinya, ini dapat mencakup semua kemungkinan jalur eksekusi sampel malware. Selain itu, analisis statis umumnya lebih aman daripada analisis dinamis karena kode sumbernya tidak benar-benar dijalankan. Namun, itu bisa sangat memakan waktu, rumit dan membutuhkan keahlian.

3 METODOLOGI PENELITIAN

3.1 Tahapan Metode Malware Analisis Dinamis

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 harddisk dan lainnya. Sistem Operasi yang akan digunakan dalam penelitian ini adalah Windows XP karena sangat mudah untukterinfeksi 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.

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. 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.

   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 dilakukanoleh malware terhadap perubahan sistem registry. Sedangkan 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.

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, dan trafik jaringan yang diakses malware.

3.2 Tahapan Metode Malware Analisis Statis

Tahapan metode malware analisis statis dalam penelitian ini sebagai berikut :

1. Ekstraksi File Malware

   Pada tahap ini dilakukan ekstraksi terhadap file malware kedalam bentuk kode String menggunakan bantuan program strings kali linux (Official Kali Linux Documentation, 2013) untuk kemudian dapat dilakukan analisis terhadap kode-kode tersebut.

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 ataubukan, 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 poison ivy (Start, 2015).

3. Disassembler

   Disassembler adalah program komputer yang dapat melakukan konversi terhadap bahasa mesin menjadi bahasa yang lebih mudah dipahami oleh manusia (Popa, 2012). Dengan disassemble, pada penelitian ini akan dilakukan analisis terhadap malware dan mencoba untuk memahami malware dengan menganalisis bahasa assembly dan mengumpulkan informasi dari program malware yang dapat digunakan untuk mengidentifikasi komponen maupun karakteristik malware.

3.3 Tahapan Hasil Analisa dan Pengujian

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 HASIL DAN 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 Analisis Malware Dinamis

Melaksanakan sampel malware tertentu dalam lingkungan yang terkontrol dan memantau tindakannya untuk menganalisis perilaku jahat disebut analisis malware dinamis. Karena Analisis Malware Dinamis dilakukan selama waktu proses dan malware membongkar sendiri, analisis perangkat lunak perusak dinamis menghindari pembatasan analisis statis (yaitu, masalah membongkar dan membingungkan). Dengan demikian mudah untuk melihat perilaku sebenarnya dari suatu program. Keuntungan utama lainnya adalah dapat diotomatisasi sehingga memungkinkan analisis dalam skala besar. Namun, kelemahan utama adalah apa yang disebut kode tidak aktif: Yaitu, tidak seperti analisis statis, analisis dinamis biasanya memantau hanya satu jalur eksekusi dan dengan demikian menderita cakupan kode yang tidak lengkap. Selain itu ada bahaya merugikan sistem pihak ketiga, jika lingkungan analisis tidak diisolasi atau dibatasi secara tepat. Selanjutnya, sampel malware dapat mengubah perilaku mereka atau berhenti mengeksekusi sama sekali setelah mereka mendeteksi untuk dieksekusi dalam lingkungan analisis yang terkontrol.

Terutama dua pendekatan dasar untuk analisis malware dinamis dapat dibedakan:

- Menganalisis perbedaan antara titik-titik yang ditentukan: Sampel malware yang diberikan dijalankan untuk jangka waktu tertentu dan setelah itu modifikasi yang dilakukan pada sistem dianalisis dengan perbandingan ke status sistem awal. Dalam pendekatan ini, laporan Perbandingan menyatakan perilaku malware.

- Mengamati runtime-behavior: Dalam pendekatan ini, aktivitas berbahaya yang dilun-

curkan oleh aplikasi berbahaya dimonitor selama waktu proses menggunakan alat khusus.

4.2 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 (SourceForge, 2015).

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.

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.

4.3 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 (Wireshark Org, 2016). 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) 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.

4.4 Analisis Malware Statis

Teknik statis dasar seperti melihat bagian luar tubuh selama autopsi. Anda dapat menggunakan analisis statis untuk menarik beberapa kesimpulan awal, tetapi lebih banyak analisis mendalam diperlukan untuk mendapatkan keseluruhan cerita. Sebagai contoh, Anda mungkin menemukan bahwa fungsi tertentu diimpor, tetapi Anda tidak akan tahu bagaimana itu digunakan atau apakah itu digunakan sama sekali. Teknik dasar yang dinamis juga memiliki kekurangan. Misalnya, dasar analisis dinamis dapat memberi tahu Anda bagaimana malware subjek Anda merespons saat itu menerima paket yang dirancang khusus, tetapi Anda dapat

mempelajari formatnya paket hanya dengan menggali lebih dalam. Di situlah pembongkaran masuk.

Disassembly adalah keterampilan khusus yang dapat membingungkan bagi mereka yang baru pemrograman. Tetapi jangan berkecil hati; bab ini akan memberi Anda dasar pemahaman tentang pembongkaran untuk membuat Anda lepas dengan kaki kanan.

Dalam arsitektur komputer tradisional, sistem komputer dapat diwakili sebagai beberapa tingkat abstraksi yang menciptakan cara menyembunyikan implementasi rincian. Misalnya, Anda dapat menjalankan OS Windows pada berbagai jenis perangkat keras, karena perangkat keras yang mendasarinya diabstraksikan dari OS. Menunjukkan tiga level pengkodean yang terlibat dalam analisis malware. Pembuat malware membuat program di tingkat bahasa tingkat tinggi dan menggunakan kompilator untuk menghasilkan kode mesin untuk dijalankan oleh CPU. Sebaliknya, malware analis dan insinyur terbalik beroperasi pada tingkat bahasa tingkat rendah kami menggunakan disassembler untuk membuat kode assembly yang bisa kita baca dan analisis untuk mencari tahu bagaimana suatu program beroperasi.

Umumnya, kode sumber sampel malware tidak tersedia. Itu mengurangi teknik analisis statis yang berlaku untuk analisis malware bagi mereka yang mengambil informasi dari representasi biner dari malware. Menganalisis binari membawa tantangan yang rumit. Pertimbangkan, misalnya, bahwa sebagian besar serangan malware host mengeksekusi instruksi dalam set instruksi IA32. Pembongkaran program-program tersebut dapat menghasilkan hasil yang ambigu jika biner menggunakan teknik memodifikasi kode diri. Selain itu, malware yang mengandalkan nilai yang tidak dapat ditentukan secara statis (mis., Tanggal sistem saat ini, instruksi lompat tidak langsung) memperburuk penerapan teknik analisis statis. Yang lain adalah bahwa pembuat malware mengetahui keterbatasan metode analisis statis dan dengan demikian, kemungkinan akan menciptakan contoh malware yang menggunakan teknik ini untuk menggagalkan analisis statis. Oleh karena itu, perlu untuk mengembangkan teknik analisis yang tahan terhadap modifikasi tersebut, dan mampu menganalisis perangkat lunak berbahaya dengan andal.

5 KESIMPULAN

5.1 Kesimpulan

Berdasarkan analisis terhadap program poison ivy yang telah dilakukan maka dapat disimpulkan 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 (WindowsRegistry dan file prefetch) sebagaimana ditemukan seperti kode string “secret agent” serta perilaku ketika program poison ivy diaktifkan tidak memberikan informasi maupun aktifitas secara kasat mata melainkan dalam perilakunya program poison ivy berupaya untuk menghubungkan diri pada program induknya yang dilakukan pada proses background (tidak kasat mata). 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. Cara kerja program poison ivy dapat dianalisis 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. 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.

5.2 Saran

Beberapa saran yang diusulkan oleh penyusun untuk penelitian lebih lanjut sebagai berikut :

1. Kedua metode yang digunakan dalam penelitian ini, metode analisis malware statis dan dinamis 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).
2. Malware merupakan topik yang masih sangat terbuka luas maka peneliti, dan juga menyarankan pengembangan teknik analisis program malware dengan memanfaatkan sub-teknik analisis statis yang dikenal dengan nama Reverse Engineering.

Referensi

Bayer, U., Moser, A., Kruegel, C., & Kirda, E. (2006). Dynamic analysis of malicious code. Journal in Computer Virology, 2(1), 67–77. http://doi.org/10.1007/s11416-006-0012-2

Cahyanto, T. (2015). BAUM-WELCH Algorithm Implementation For Knowing Data Characteristics Related Attacks On Web Server Log. PROCEEDING IC-ITECHS 2014. Retrieved from http://jurnal.stiki.ac.id/index.php/IC-ITECHS/article/view/131

Cahyanto, T. A., Oktavianto, H., & Royan, A. W. (2013). Analisis dan Implementasi Honeypot Menggunakan Dionaea Sebagai Penunjang Keamanan Jaringan. JUSTINDO (Jurnal Sistem DanTeknologi Informasi Indonesia), 1(2), 86–92. Retrieved from http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/568

Cahyanto, T. A., & Prayudi, Y. (2014). Investigasi Forensika Pada Log Web Server untuk Menemukan Bukti Digital Terkait dengan Serangan Menggunakan Metode Hidden Markov Models. Snati, 15–19. Retrieved from http://jurnal.uii.ac.id/index.php/Snati/article/view/3280

Cuckoo Sandbox. (2016). Automated Malware Analysis - Cuckoo Sandbox. Retrieved July 31, 2017, from https://cuckoosandbox.org/

Education, I. J. M., Science, C., Sujyothi, A., & Acharya, S. (2017). Dynamic Malware Analysis and Detection in Virtual Environment, (March), 48–55. http://doi.org/10.5815/ijmecs.2017.03.06

Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys, 44(2), 1–42. http://doi.org/10.1145/2089125.2089126

Firmansyah, R., Akbar, M. and Negara, E.S., (2019). Mobile Forensik Pemulihan Data Pada Aplikasi Instant Messenging. In Bina Darma Conference on Computer Science (BDCCS) (Vol. 1, No. 2, pp. 360-371).

FireEye, I. (2014). Poison Ivy: Assessing Damage and Extracting Intelligence, 33. Retrieved from https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-poison-ivy.pdf

Gandotra, E., Bansal, D., & Sofat, S. (2014). Malware Analysis and Classification: A Survey. Journal of Information Security, 5(2), 56–64. http://doi.org/10.4236/jis.2014.52006

Kramer, S., & Bradfield, J. C. (2010). A general definition of malware. Journal in Computer Virology, 6(2), 105–114. http://doi.org/10.1007/s11416-009-0137-1

Moser, A., Kruegel, C., & Kirda, E. (2007). Limits of static analysis for malware detection. Proceedings - Annual Computer Security Applications Conference, ACSAC, 421–430. http://doi.org/10.1109/ACSAC.2007.21

Official Kali Linux Documentation. (2013). Retrieved from https://www.kali.org/kali-linux-documentation/ Popa, M. (2012). Binary Code Disassembly for Reverse Engineering. Journal of Mobile, Embedded and Distributed Systems, IV(4), 233–248. Retrieved from http://jmeds.eu/index.php/jmeds/article/view/81

Sikorski, M., & Honig, A. (2013). Practical Malware Analysis. No Starch, 53(9), 1689–1699. http://doi.org/10.1017/CBO9781107415324.004

SourceForge. (2015). Regshot download. Retrieved July 31, 2017, from https://sourceforge.net/projects/regshot/

tart2015 Start, C. (2015). Project 11 : Poison Ivy Rootkit (15 points) What You Need for This Project. Retrieved July 31, 2017, from https://samsclass.info/123/proj10/p11-PI.htm

Tzermias, Z., Sykiotakis, G., Polychronakis, M., & Markatos, E. P. (2011). Combining static and dynamic analysis for the detection of malicious documents. Proceedings of the Fourth European Workshop on System Security - EUROSEC '11, 1–6. http://doi.org/10.1145/1972551.1972555

U., Kirda, E., & Kruegel, C. (2010). Improving the efficiency of dynamic malware analysis. Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10, 1871. http://doi.org/10.1145/1774088.1774484

Wireshark Org. (2016). Wireshark · Download. Retrieved July 31, 2017, from https://www.wireshark.org/download.html