Dimas Wahyudi – 09011281320004 | Sistem Komputer | Fakultas Ilmu Komputer | Universitas Sriwijaya
mail.dimaswahyudi@gmail.com

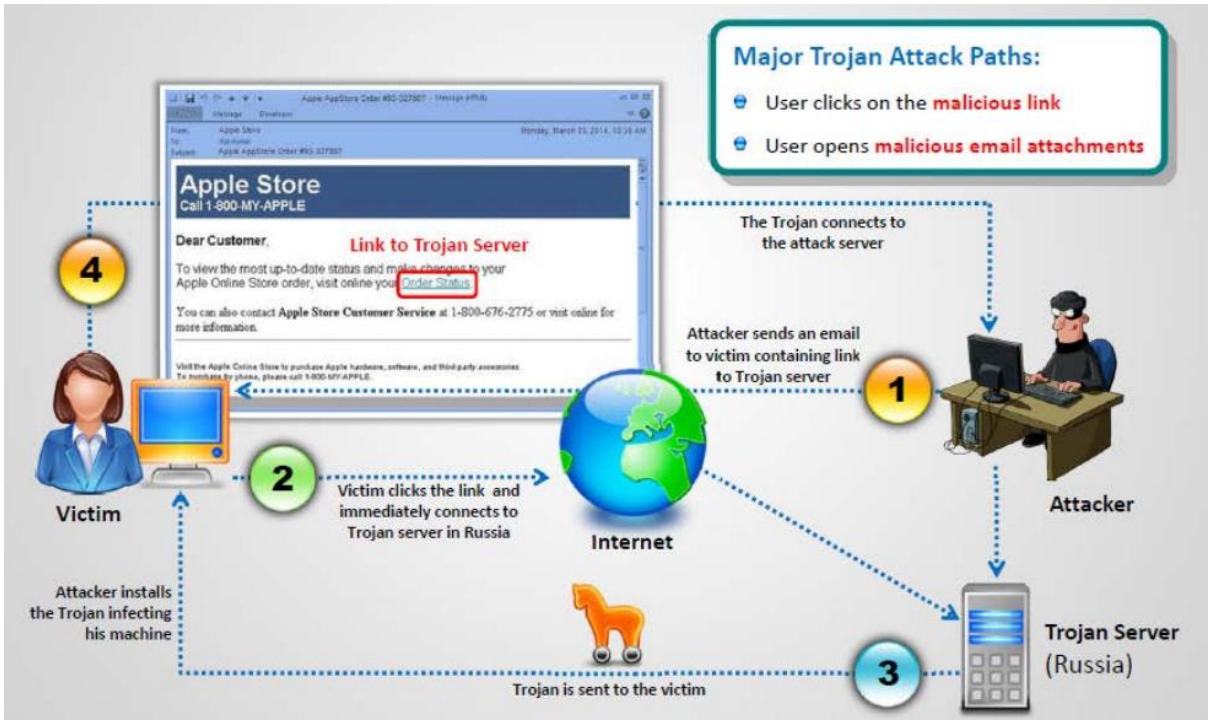
Malware

Dasar Teori : Analisa dan deteksi *malware*

Analisa *malware* adalah suatu aktivitas yang dilakukan oleh sejumlah praktisi keamanan teknologi informasi untuk mendeteksi ada atau tidaknya komponen sub-program atau data yang bertujuan jahat dalam sebuah file elektronik. Analisa atau kajian ini sangat penting untuk dilakukan karena alasan-alasan berikut : (1) *Malware* sering diselundupkan melalui file-file umum dan popular seperti aplikasi (.exe), pengolah kata (.doc), pengolah angka (.xls), gambar (.jpg), dan lain sebagainya – sehingga jika pengguna awam mengakses dan membukanya, akan langsung mejadi korban program jahat seketika; (2) *Malware* sering diselipkan di dalam kumpulan file yang dibutuhkan untuk menginstalasi sebuah program atau aplikasi tertentu. Sehingga jika pengguna melakukan instalasi terhadap aplikasi dimaksud, seketika itu juga malware diaktifkan; (3) *Malware* sering disamarkan dengan menggunakan nama file yang umum dipakai dalam berbagai keperluan, seperti *driver* (.drv), *data* (.dat), *library* (.lib), *temporary* (.tmp), dan lain-lain. Sehingga pengguna tidak sadar akan kehadirannya di dalam komputer yang bersangkutan; (4) *Malware* sering dikembangkan agar dapat menularkan dirinya ke tempat-tempat lain, dengan cara kerja seperti *virus* atau *worms*. Sehingga komputer pengguna dapat menjadi sarang atau sumber program jahat yang berbahaya; (5) *Malware* sering ditanam di dalam sistem komputer tanpa diketahui oleh sang pengguna. Sehingga sewaktu-waktu dapat disalahgunakan oleh pihak yang tidak berwenang untuk melakukan berbagai tindakan kejahatan; dan lain sebagainya.

Gambar 1. *Malware Topology 1*

Pada dasarnya *malware* adalah sebuah program, yang disusun berdasarkan tujuan tertentu dengan menggunakan logika dan algoritma yang relevan dengannya. Oleh karena itulah maka model analisa yang biasa dipergunakan untuk mengkaji malware sangat erat kaitannya dengan ilmu dasar komputer, yaitu: bahasa pemrograman, algoritma, struktur data, dan rekayasa perangkat lunak.



Gambar 2. *Malware Topology 2*

Secara umum, ada 3 (tiga) jenis analisa terhadap sebuah program untuk mendeteksi apakah yang bersangkutan merupakan malware atau bukan. Ketiga pendekatan yang dimaksud akan dijelaskan dalam masing-masing paparan sebagai berikut :

1. *Surface Analysis*
   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 pendudukung. Analisa ini memiliki ciri-ciri sebagai berikut :

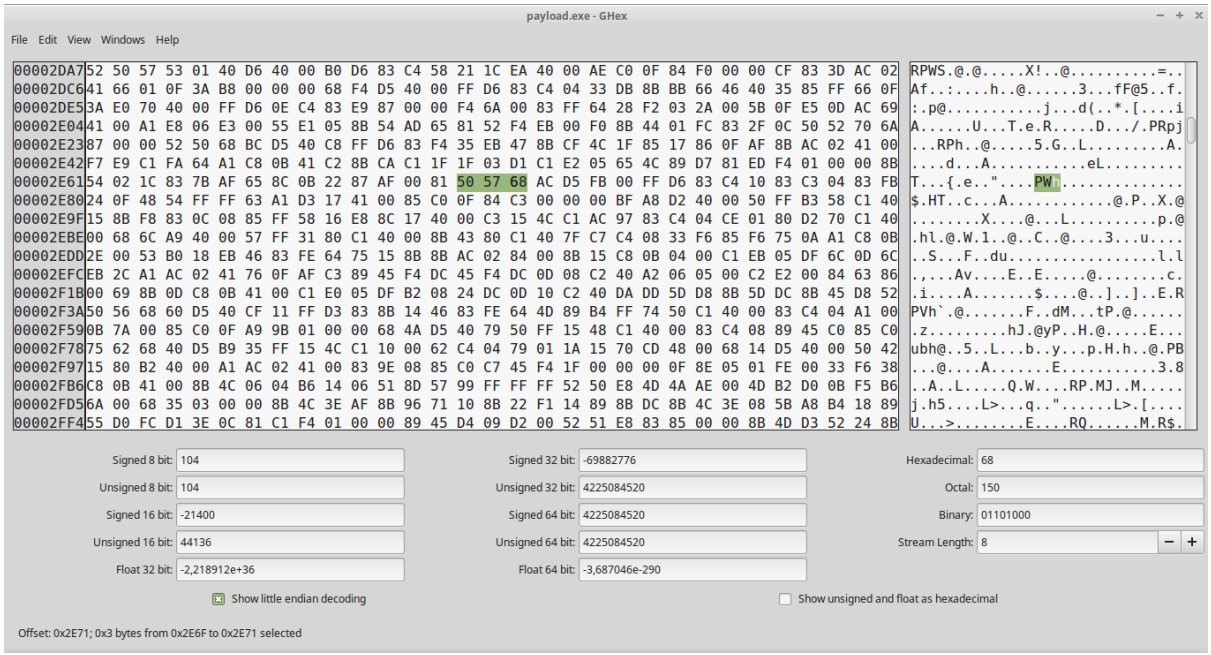
   - Program yang dikaji tidak akan dijalankan, hanya akan dilihat bagian luarnya saja (sebagai analogi selayaknya orang yang ingin membeli buah- buahan, untuk mengetahui apakah buah yang bersangkutan masih mentah atau sudah busuk cukup dengan melihat permukaan kulitnya, membauinya, dan meraba-raba tekstur atau struktur kulitnya). Dari sini akan dicoba ditemukan hal-hal yang patut untuk dicurigai karena berbeda dengan ciri khas program kebanyakan yang serupa dengannya.
   - 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*.

   Saat ini cukup banyak aplikasi yang bebas diunduh untuk membantu melakukan kegiatan *surface analysis* ini, karena cukup banyak prosedur kajian yang perlu dilakukan, seperti misalnya: *HashTab* dan *digest.exe* (*Hash Analysis*), *TrID* (*File Analysis*), *BinText* dan *strings.exe* (*String Analysis*), *HxD* (*Binary Editor*), *CFF Explorer* (*Pack Analysis*), dan *7zip* (*Archiver*).

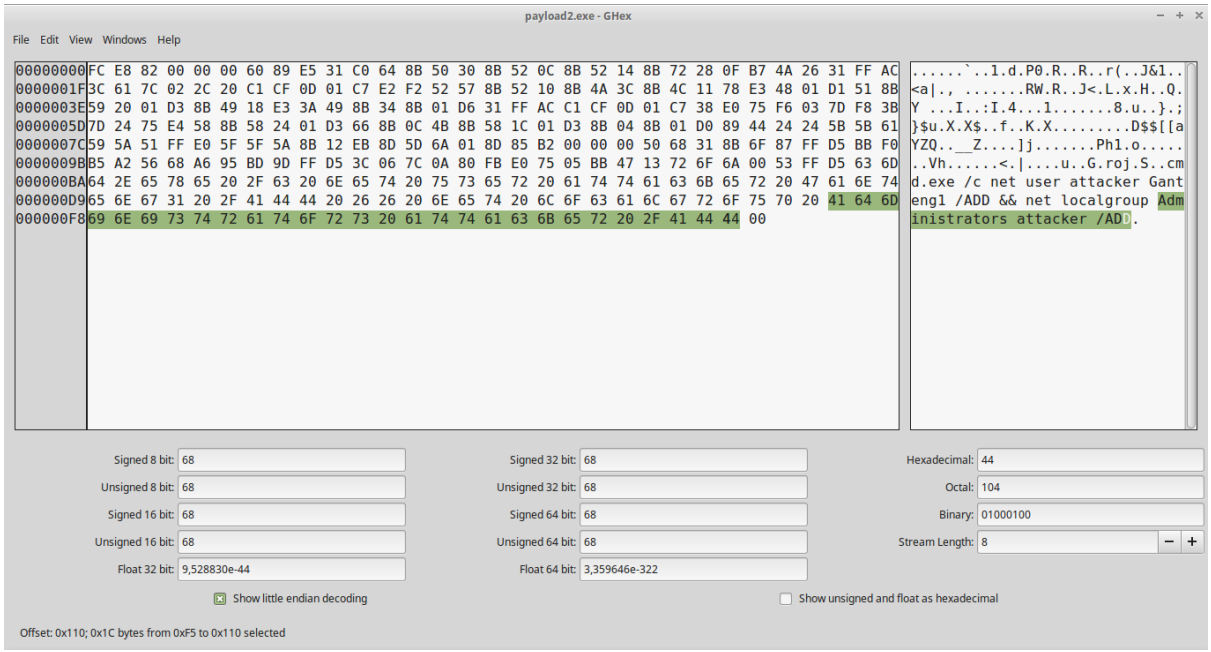
2. *Runtime Analysis*
   Model analisa ini menghasilkan kajian yang lebih mendalam karena selain dihilangkannya proses asumsi, dengan mengeksekusi *malware* dimaksud akan dapat dilihat perilaku dari program dalam menjalankan skenarionya, sehingga selanjutnya dapat dilakukan analisa dampak terhadap sistem yang ada. Oleh karena itulah maka aplikasi pendukung yang dipergunakan harus dapat membantu mensimulasikan kondisi yang diinginkan, yaitu melihat ciri khas dan karakteristik sistem, sebelum dan sesudah sebuah malware dieksekusi. Agar aman, maka program utama yang perlu dimiliki adalah *software* untuk menjalankan *virtual machine*, seperti misalnya: *VMWare, VirtualBox, VirtualPC,* dan lain sebagainya. Sementara itu aplikasi pendukung lainnya yang kerap dipergunakan dalam melakukan kajian ini adalah: *Process Explorer, Regshot, Wireshark, TCPView, Process Monitor, FUNdelete, Autoruns, Streams/ADSSpy*, dan lain-lain. Keseluruhan aplikasi tersebut biasanya dijalankan di sisi klien; sementara di sisi *server*-nya diperlukan *FakeDNS, netcat/ncat, tcpdump/tshark,* dan lain sebagainya.

3. *Static Analysis*
   Static analysis merupakan model kajian yang paling sulit dilakukan karena sifat analisanya yang *white box* alias pengkajian melibatkan proses melihat dan mempelajari isi serta algoritma program *malware* dimaksud, sambil mengamati sekaligus menjalankan / mengeksekusinya. Karena sifat dan ruang lingkupnya yang cukup luas dan mendalam, strategi khusus perlu dipersiapkan untuk melakukan kajian ini. Disamping itu, juga memerlukan sumber daya yang khusus , misalnya adalah SDM yang memiliki pengetahuan dan pengalaman dalam membuat serta membaca program berbahasa mesin atau rakitan (*assembly language*) serta ahli arsitektur dan organisasi piranti komputasi seperti komputer, *PDA, tablet, mobile phone,* dan lain sebagainya. Cukup banyak aplikasi pendukung yang diperlukan, tergantung dari kompleksitas *malware* yang ada. Seperti *IDA Pro* (*Disassembler*); H*ex-Rays, .NET Reflector*, and *VB Decompiler* (*Decompiler*); *MSDN Library, Google (Library); OllyDbg, Immunity Debugger, WinDbg/Syser* (*Debugger*); *HxD, WinHex, 010editor* (*Hex Editor*); *Python, Lunux Shell/Cygwin/MSYS* (*Others*); dan lain-lain.

Tahap : *Ghex tools*



Gambar 3. *Payload*



Gambar 4. *Payload2*

Daftar Pustaka

[1] A. H. Abdullah, "Cyber-Attack Penetration Test and Vulnerability Analysis," vol. 13, no. 1, pp. 125–132.

[2] I. C. of E.-C. C. (EC-Council), "Malware Threat," *Certif. Ethical Hacker V8.00*.

[3] R. E. Indrajit, "Forensik Komputer," *Artikel*, vol. 1, no. C, pp. 1–11, 2011.