BAB 2
DASAR TEORI DAN TINJAUAN PUSTAKA

Pada bab 2 akan dibahas tentang dasar teori dan tinjauan pustaka yang digunakan dalam pembuatan Proyek Akhir ini.

2.1 Dasar Teori

2.1.1 Definisi Malicious Activity (*Malware*)

Malware merupakan sebuah perangkat lunak yang dirancang untuk merusak sistem dan memanipulasi data tanpa sepengetahuan pemilik dari device yang terinfeksi, malware dapat berbentuk *script, active content* dan *binary*. Malware pada umumnya digunakan oleh hacker atau kriminal lainya untuk merusak jalanya suatu sistem operasi, mencuri data privat seseorang, melakukan *bypass* terhadap *access control* dan merusak sistem pada host.

2.1.2 Jenis – jenis *Malware*

Malware dapat dikategorikan menjadi beberapa jenis dan malware mempunyai banyak variasi juga antara lain:

a. *Adware*

Adware mempunyai kepanjangan *advertising-supported software* adalah jenis malware yang secara otomatis mengerimkan iklan-iklan yang menarik perhatian kepada user, seperti contohnya pop-up iklan pada website tertentu, kadang adware juga menawarkan aplikasi yang gratis namun saat di unduh berisikan malware lain atau adware itu sendiri.

b. *Botnet*

Bot atau botnet adalah sebuah *software* yang diprogram dan dirancang untung melakukan sebuah aktivitas atau operasi secara otomatis, seperti misalnya pada serangan jaringan *DdoS* (*Distributed Denial of Service*) *botnet* digunakan untuk melakukan *ping* kepada *victim* yang sudah di tentukan, *botnet* dapat beroperasi secara otomatis dan dapat juga di kontrol oleh pihak ketiga.

c. *Bug*

Bug merupakan sebuah kejanggalan atau kesalahan pada suatu program yang biasanya tercipta karena kesalahan pembuatnya sendiri dalam memasukkan barisan kode saat pembuatan sebuah program, namun *bug* dapat menjadi alasan kenapa sebuah program berjalan tidak sesuai yang diinginkan bahkan saat tidak teridentifkasi pada jangka waktu yang lama juga dapat menyebabkan kelemahan atau celah yang dapat di eksploitasi.

d. *Ransomeware*

Ransomware adalah jenis *malware* yang saat diaktifkan akan mengunci sistem operasi dan data *user* sebagai tawanan dan data tidak akan diberikan kembali sampai tebusan yang sudah ditentukan sudah dibayar. *Malware* ini membatasi hak akses *user* dan mengenkripsi *file* yang ada pada *hard drive*, dan menampilkan pesan yang bertujuan untuk memaksa user membayar tebusan.

e. *Rootkit*

Rootkit adalah sebuah malware yang memungkinkan hacker melakukan remote access control tanpa terdeteksi user aslinya. Ketika rootkit diaktifkan akan memungkinkan hacker mengeksekusi file, mengakses dan mencuri informasi secara diam-diam.

f. *Spyware*

Spyware adalah jenis *malware* yang berfungsi untuk memata-matai aktivitas *user* tanpa *user* itu sendiri mengetahui, *malware* ini bertujuan untuk mengetahui aktivitas *user*, dan memanen informasi terkait finansial seperti misalnya transaksi bank yang sensitif untuk diketahui.

g. *Trojan Horse*

Trojan horse atau biasa disebut *trojan* adalah sebuah *malware* yang mempunyai metode menyembunyikan *malware* nya di dalam file biasa guna menipu *user*, ketika *file* telah ter-*install* maka *malware* yang berada dalam *file* tersebut akan diaktifkan, *malware* ini dapat memberi akses *remote* pada hacker dan memungkinkan pencurian informasi bahkan sampai pencurian uang elektronik.

h. *Virus*

Virus adalah *malware* yang dapat menduplikat dirinya sendiri dan menyebar ke komputer lain ketika telah diaktifkan, *virus* dapat menyebar ke komputer lain dengan menempelkan dirinya ke beberapa program tertentu yang nantinya diaktifkan di komputer lain. *Virus* dapat digunakan untuk mencuri informasi, mencuri uang elektronik, membuat botnet, merusak host dan juga jaringan.

i. *Worm*

Worm adalah satu diantara *malware* yang paling umum, *worm* menyebar melalui jaringan komputer dengan melakukan eksploitasi terhadap celah yang ada pada sistem operasi. *Worm* pada umumnya menyebabkan rusak nya *host* dengan mengkonsumsi banyak *bandwidth* dan mengisi *web server* sampai *overload*.

j. *Laucher*

Launcher adalah *malware* yang berfungsi untuk menjalankan *malware* lainnya. Pada umumnya, *launcher* menggunakan teknik non tradisional untuk menjalankan *malware* yang lain untuk memastikan akses tersembunyi atau akses yang lebih baik pada sistem.

k. *Information-Stealing Malware*

Information-stealing malware adalah *malware* yang mencuri informasi dari komputer korban dan dengan mengirim informasi tersebut langsung kepada *hacker*. Contohnya adalah *sniffer* yang berfungsi untuk melakukan *sniffing* jaringan dan menangkap semua paket yang berlalu-lintas dan *keylogger* yang berfungsi untuk merekam aktivitas *keyboard*.

l. *Downloader*

Downloader adalah sebuah *malware* yang hanya beroperasi untuk melakukan *download* tehadap *malware* lainnya. *Downloader* biasanya di-*install* oleh *hacker* untuk mendapatkan *akses* pada suatu sistem.

m. *Backdoor*

Backdoor adalah sebuah *malware* yang dapat menginstalasi sendiri *script*nya secara otomatis pada suatu komputer. Tujuannya adalah untuk memberi akses pada *hacker* atau pembuat *malware*. *Backdoor* memungkinkan *hacker* untuk melakukan koneksi pada komputer dengan sedikit bahkan tidak ada otentikasi dan mengeksekusi perintah pada suatu sistem komputer.

n. *Spam-Sending*

Spam-sending malware merupakan sebuah malware yang menginfeksi computer lalu menggunakan komputer tersebut untuk mengirim spam.

o. *Scareware*

Scareware adalah *malware* yang didesain untuk mengintimidasi *user* dan memaksa korban untuk membeli *scareware*. *Scareware* memiliki antarmuka yang mirip dengan *antivirus*. Contohnya adalah menginformasikan *user* bahwa ada virus yang menyerang komputer *user* dan satu-satunya cara untuk melakukan penanggulangan hanya dengan membeli *scareware* tersebut.

2.1.3 *Malicious Activity* pada *Malware*

Berikut ini merupakan rincian proses ketika malware telah berhasil diaktifkan dan apa saja aktivitas- aktvitas yang tidak sesuai dengan malware.

a. *Process Hollowing*

Process hollowing merupakan aktivitas yang diakibatkan oleh *malware.* Saat *malware* melakukan injeksi proses pada memori dan menggantikan dengan kode dari *malware* itu sendiri yang disimpan.

b. *Created Remote Thread*

Create Remote Thread merupakan aktivitas dimana suatu *malware* melakukan *injeksi DLL* dengan menggunakan *API call CreateRemoteThread.*

c. *Enumerating All Processes*

Enumerating All Processes adalah aktivitas dimana suatu *malware* melakukan perhitungan dan mencari tahu proses apa saja yang berjalan pada suatu komputer yang terinfeksi.

d. *Drop Files from PE Resource Section*

Drop Files from PE Resource Section ini adalah aktivitas dari suatu *malware* yang melakukan drop file pada komputer yang terinfeksi.

e. *IAT Hooking*

IAT Hooking adalah singkatan dari *import address table hooking* merupakan aktivitas yang digunakan untuk mencari informasi mengenai fungsi yang berhubungan dengan network yang digunakan oleh suatu aplikasi. *IAT hooking* digunakan *malware* untuk mengambil fungsi data secara bersamaan dari suatu aplikasi menggunakan *API call GetModuleHandle*.

f. *Deleted itself*

Delete Itself merupakan *malicious activity* dimana suatu *malware* dapat menghapus diri sendiri setelah menjalankan suatu script tertentu.

g. *Download & Execute PE Files*

Download & Execute PE Files merupakan *malicious activity* dimana suatu malware melakukan pengunduhan external file untuk selanjutnya dijalankan pada komputer yang terinfeksi.

h. *Bind TCP Port*

Bind TCP Port adalah aktivitas yang dilakukan oleh *malware* untuk membuka suatu *port* tertentu yang dapat diakses dari pihak yang mengetahui. Pada umumnya, *port* yang dibuka merupakan *backdoor* untuk mengakses komputer yang terinfeksi.

i. *Capture Network*

Capture Network merupakan aktivitas dimana suatu *malware* melakukan *sniffing* pada komputer yang terinfeksi. *Sniffing* digunakan guna mencari informasi mengenai aktivitas jaringan pada jaringan yang terinfeksi.

2.1.4 *Malware* Analisis

Malware analisis adalah sebuah aktivitas untuk membedah atau menganalisis. *Malware* analisis mengidentifikasi bagaimana cara *malware* tersebut bekerja, fungsionalitas *malware* dan bagaimana cara pendeteksian serta pencegahanya yang paling efektif terhadap suatu *malware*. Untuk melakukan *malware* analisis terdapat dua metode analisis yaitu *static* analisis dan *dynamic* analisis.

a. *Static* Analisis

Static analisis merupakan model kajian yang paling sulit dilakukan karena sifat anlisisnya yang "*white box*" alias pengkajian melibatkan proses melihat dan mempelajari isi serta algoritma program *malware* dimaksud sambil mengamati sekaligus menjalankan/mengeksekusinya.

Berikut ini merupakan kelebihan dari *static analysis* adalah:

- Dapat menemukan kelemahan dalam kode di lokasi yang tepat.
- Kode tidak benar – benar dijalankan sehingga mengurangi resiko komputer terinfeksi virus.
- Mengurangi waktu penelitian.
- Dapat memberikan rekomendari mitigasi secra otomatis.

Sedangkan kelemahan dari static analysis adalah:

- Alat tidak mendudukung semua bahasa pemrograman.
- Alat menghasilkan deteksi *false positives* atau *false negative.*

b. *Dynamic* Analisis

Merupakan metode mengamati perilaku *malware* selama proses eksekusi dan infeksi *malware* tersebut, guna menentukan perilaku sistem, panggilan fungsi, fungsi parameter yang ada dan juga arus informasi. Pada analisis *dynamic* menggunakan *environment* buatan seperti *virtual machine* dan juga *sandbox*. Dalam konsep identifikasi *malware* yang belum diketahui *dynamic* analisis metode terbaik.

Berikut ini merupakan kelebihan dari *dynamic analysis* adalah:

- Dapat mengidentifikasi kerentanan didalam lingkungan *runtime.*
- *Tools* akan memberikan fleksibelitas tentang apa yang harus dipindai.
- Memungkingkan analisis aplikasi dimana anda tidak memiliki akses menuju kode aktual.
- Memungkinkan anda untuk memvalidasitemuan kode statis.
- Dapat dilakukan dalam aplikasi apapun.

Sedangkan kelemahan dari *dynamic analysis* adalah:

- *Tools* dapat memberikan rasa aman palsu.
- Lebih sulit melacak kerentanan dan butuh waktu lama untuk memperbaikinya.

2.1.5 Penggunaan Metode

Pada umumnya analisis *malware* terbagi menjadi 2 cara yaitu secara dinamis dan secara statis, metode dinamis merupakan metode yang melakukan eksekusi sampel *malware* pada *virtual machine* sementara metode statis membongkar *source code* sampel *malware* tersebut. Diantara kedua metode tersebut analisis malware dynamic merupakan metode paling efektif dilihat dari API (Application Programming Interface) /interface yang dapat menghubungkan satu aplikasi dengan aplikasi lainya yang dipanggil, aktivitas dan perubahan regristry oleh sampel malware. Dengan dilandasi hal itu maka penelitian ini akan menggunakan metode analisis dynamic sebagai metode utama dalam melakukan analisis serta deteksi malware.

2.2 Tinjauan Pustaka

Untuk mendapatkan hasil penelitian yang optimal, dilakukan kajian dari penelitian - penelitian terdahulu, sehingga dapat dijadikan referensi dalam penelitian dengan tujuan agar diperoleh perbandingan kelebihan dan kekurangan.

Pada penelitian sebelumnya, yang dilakukan oleh (Sulhaedir dan Febri Nova Lenti, 2016) yang berjudul "ANALISIS MALWAE TROJAN" adapun permasalahan pada penelitian ini yaitu, saat ini sangat banyak perangkat lunak yang ditawarkan secara gratis (*freeware*). Perangkat lunak *freeware* biasanya dapat didownload secara gratis memiliki banyak resiko seperti perangkat lunak sengaja dibuat lalu diberikan secara gratis tetapi perangkat lunak tersebut adalah sebuah *malware* dengan jenis perangkat lunak yang dikembangkan dengan tujuan kejahatan.

Pada penelitian diatas, dapat disampaikan bahwa penelitian sebelumnya dilakukan analisis dengan metode dinamis didalam jenis *malware* trojan. Dalam hal ini saya akan membuat diimplementasikan pada analisis *malware* jenis *Trojan Horse* yang mencuri data user (*stealer*).