Bab I PENDAHULUAN

I.1 Latar Belakang

Di era IoT (*Internet of Things*) salah satu ancaman terbesar di internet saat ini adalah *malicious software*, biasanya disebut *malware* karena hampir semua penyebab utama masalah keamanan internet adalah *malware*. *Malware* merupakan program yang diciptakan dengan tujuan merusak dengan menyusup ke sistem komputer. *Malware* memiliki berbagai macam jenis, yaitu *virus, worm, spyware, adware, trojan, keylogger, rootkit, botnet* dan *phising* (Cahyanto, Wahanggara, & Ramadana, 2017). Contohnya, *botnet* biasanya digunakan untuk mengirimkan *spam* dan *host phising website* yang menyulitkan untuk dilacak dan di-*blacklist* (Bayer, Comparetti, Hlauschek, Kruegel, & Kirda, 2009). Selain *botnet*, *malware* yang sering digunakan untuk menyusupi suatu sistem adalah *spyware*. Menurut data statistik, 70-80% *spyware* berasal dari situs website yang dianggap aman oleh pengguna internet (Jain & Bajaj, 2014).

Berdasarkan laporan dari ShadowServer (sebuah organisasi yang bergerak dibidang *cybercrime*), ada ribuan sampel *malware* baru yang diterima setiap harinya (Bayer et al., 2009). Setiap sampel *malware* di analisis satu persatu dengan tujuan untuk mengetahui jenis *malware* apa, seberapa besar ancamannya dan bagaimana cara penanganannya. Untuk mendapatkan informasi lengkap mengenai satu sampel *malware* dibutuhkan analisis *static* dikarenakan analisis ini dilakukan dengan meneliti *malware source code* tersebut (Ismail, 2016). Namun, analisis ini menjadi tidak efektif dan efisien karena pembuat *malware* saat ini sudah mulai menggunakan teknik *obfuscation*, teknik ini dapat melindungi *malware* dengan melakukan penyamaran sehingga *malware* sulit untuk dideteksi (Perdisci, Roberto and Lee, Wenke and Feamster, 2010). Untuk mengatasi masalah ini, diperlukan *Dynamic Analysis.*

Dynamic Analysis digunakan menganalisis informasi *behavioral* seperti aktivitas jaringan, API *call*, *file operation* dan catatan modifikasi registri dengan mengeksekusi sampel dalam *environment* virtual. Kekurangan dari *Dynamic Analysis* adalah metode ini memerlukan waktu dan sumber daya yang besar untuk mengeksekusi *malware* (Kim, Wang, & Rho, 2001). Penggunaan metode *Dynamic*

Analysis dengan teknik *behavior-based* akan menghasilkan informasi API *call*. Informasi ini berisi urutan-urutan API yang digunakan oleh *malware* selama melakukan *malicious activity*. Pada Windows, setiap program yang dapat dieksekusi perlu membuat satu set API *call*. Contohnya, untuk *file management* ada beberapa API *calls* yaitu, *OpenFile, DeleteFile, FindClose, FindFirstFile, GetFileSize* (Merialdo, 2012).

Informasi API *call* yang diperoleh setelah melakukan *Dynamic Analysis* dengan teknik *behavior-based* berguna untuk mengetahui aktivitas apa yang dilakukan dan perilaku apa yang dimiliki oleh *malware* tersebut sehingga dapat diketahui bagaimana dampaknya terhadap trafik jaringan serta apa yang menyebabkan terjadinya dampak tersebut.

Analisis *malware* secara *Dynamic Analysis* terhadap *network traffic* perlu dilakukan karena sedikitnya penelitian yang dilakukan sebelumnya. Sebanyak 230 sampel *malware* yang diperoleh dari tanggal 21 Maret 2019 sampai 11 April 2019 melalui Virus Sign, hanya 30 sampel *malware* yang terdeteksi memiliki API *network* dan membawa *url* atau *ip*. Untuk memperoleh hasil yang diharapkan agar tujuan penelitian ini tercapai, diperlukan API *call network* pada setiap *malware* yang diperoleh menggunakan *tool* Cuckoo Sandbox. API *network* yang diperoleh setelah mengeksekusi *malware* pada *environment* virtual akan dianalisis urutan pemanggilannya dan API *network* apa saja yang digunakan oleh *malware* tersebut sehingga dapat diketahui aktivitas-aktivitas yang dilakukan *malware* yang memanfaatkan Windows API khususnya API *network*. Setelah semua informasi mengenai API *network* diperoleh, dilakukan analisis trafik jaringan yang direkam selama menjalankan *malware* pada *environment* virtual menggunakan Wireshark. Analisis yang dilakukan pada hasil *capture* trafik jaringannya adalah dengan melihat *behavior malware* pada trafik jaringan, *payload* yang dibawa oleh *malware*, dan pengukuran *bandwidth* antara trafik normal dengan trafik yang sudah terinfeksi *malware*.

Berdasarkan hasil analisis urutan API *network* dan trafik jaringan tersebut maka hasil akhir dari penelitian ini berupa penjelasan bagaimana dampak setiap sampel *malware* yang digunakan dalam penelitian ini terhadap trafik jaringan.

I.2 Perumusan Masalah

Berdasarkan latar belakang, rumusan masalah pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana analisis *malware* dengan teknik deteksi *behavior-based* yang dilakukan pada OS Windows 7.
2. Bagaimana menentukan suatu aktivitas *malware* berdasarkan API *network* yang digunakan oleh *malware.*
3. Bagaimana mengetahui aktivitas *malware* berdasarkan *behavior*, *payload*, dan *bandwidth* pada trafik jaringan.
4. Bagaimana pemilihan sampel *malware* yang akan digunakan dalam penelitian.

I.3 Tujuan Penelitian

Berdasarkan perumusan masalah maka tujuan dari tugas akhir ini adalah:

1. Melakukan analisis *malware* pada OS Windows 7 dengan menganalisis API *network* dan trafik jaringannya.
2. Melakukan analisis *malware* dengan Cuckoo Sandbox untuk memperoleh API *network* yang digunakan oleh *malware*.
3. Melakukan *capture* trafik jaringan dengan tool Wireshark dan menganalisis hasil *capture* trafik jaringannya.
4. Melakukan pengujian sampel dengan pengecekan pada Cuckoo.

I.4 Manfaat Penelitian

Manfaat dari tugas akhir ini, yaitu:

1. Dapat mengetahui teknik deteksi *malware* dengan teknik *behavior-based* untuk melakukan *malware analysis*.
2. Dapat mengetahui pemanfaatan Windows API khususnya API network oleh malware dalam menjalankan aktivitasnya.
3. Dapat menganalisa aktivitas *malware* berdasarkan *behavior*, *payload*, dan *bandwidth* trafik jaringan.

I.5 Batasan Masalah

Batasan masalah untuk tugas akhir ini adalah:

1. Tidak melakukan analisis secara *static*
2. Tidak memberikan saran aplikasi analisis yang mempermudah penelitian
3. Hanya menggunakan satu jenis API *call* yaitu, API *network*.
4. Hanya fokus pada protokol HTTP, TCP, DNS, NetBIOS, dan ICMP.
5. Tidak menganalisis dengan OSI *Layer* sebagai acuan.
6. Hanya fokus pada *bandwidth* untuk mengetahui dampak terhadap trafik jaringan

I.6 Batasan Implementasi

Batasan implementasi dari tugas akhir ini adalah:

1. Menggunakan *virtual machine* berupa VMware dan VirtualBox
2. Menggunakan OS Ubuntu 16.04 LTS sebagai *environment* utama
3. Menggunakan OS Windows 7 (32-bit) sebagai *environment* virtual
4. Menggunakan 30 sampel *malware*
5. Menggunakan teknik deteksi *behavior-based*
6. Hanya mengambil informasi API *network* dan trafik jaringan
7. Menggunakan Wireshark sebagai *network analyzer*

I.7 Sistematika Penulisan

Penulisan ini akan dijabarkan dengan sistematika penulisan sebagi berikut:

BAB I PENDAHULUAN

Bab ini memberikan uraian latar belakang, rumusan masalah, tujuan penelitan, manfaat penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan literature yang relevan dengan permasalahan yang dihadapi. Menjelaskan setiap teori yang digunakan berdasarkan referensi yang telah didapatkan.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang langkah-langkah penelitian secara rinci, mulai dari tahap awal, tahap analisis, tahap desain, tahap pengujian dan tahap akhir dari penelitian.

BAB IV RANCANGAN SISTEM DAN SKENARIO PENGUJIAN

Bab ini akan menjelaskan tentang perancangan environment virtual yang meliputi spesifikasi sistem dan *hardware*

BAB V HASIL DAN ANALISIS

Bab ini berisi hasil analisis yang dilakukan terhadap malware yang sudah dieksekusi pada *environment* virtual.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari penelitian yang dilakukan dan saran yang bermanfaat bagi pembaca dan dapat digunakan untuk penelitian selanjutnya.