BAB I

PENDAHULUAN

1.1 Latar Belakang

Malicious Software atau *malware* merupakan sekumpulan instruksi atau program yang berjalan pada suatu sistem komputer yang membuat sistem tersebut melakukan sesuatu yang diinginkan penyerang (Skoudis dan Zeltser, 2004). Umumnya, *malware* memiliki ukuran yang relatif kecil, tetapi dengan akibat yang besar, seperti pencurian dan penghapusan data. Pertumbuhan *malware* berlangsung sangat cepat. Menurut laporan "The State of Malware 2013" dari McAfee, ada 100.000 sampel *malware* baru per harinya atau 69 *malware* setiap menitnya (McAfee, 2013).

Di sisi *antivirus*, pertumbuhan *malware* yang sangat cepat ini berdampak pada semakin besarnya ukuran *database signature*. *Database signature* merupakan basis data yang digunakan *antivirus* yang berisi penanda *malware* yang telah diketahui; *database* ini digunakan untuk memeriksa *file* lain dengan dilakukan pembandingan *signature file*-nya apakah sama dengan *signature* yang ada di *database*. *Signature* suatu *file* umumnya berupa nilai *hash* yang merupakan nilai unik yang merepresentasikan data dalam *file* tersebut. Nilai *hash* dihasilkan oleh suatu *Cryptographic Hash Function*, seperti MD5, SHA-1, dan SHA256. Masing-masing *hash function* memiliki algoritma yang berbeda-beda sehingga nilai *hash* dan panjang karakter nilai *hash* yang dihasilkan berbeda-beda untuk *file* yang sama.

Setiap satu *malware* memiliki atau dapat dikenali dengan satu *signature* yang unik (Nurjadi, 2013). Contohnya, untuk mengenali dua juta *malware*

dibutuhkan dua juta *signature* dan jika ada 100.000 sampel *malware* baru per harinya (McAfee, 2013), maka dalam waktu sepuluh hari jumlah *signature* akan mencapai tiga juta. Diasumsikan tiga juta *signature* tersebut menggunakan MD5 (bila dikonversi ke dalam *string*, panjang *hash*-nya 32 byte), maka ukuran yang diperlukan sekitar 32 x 3.000.000 = 92 Mbytes (Nurjadi, 2013).

Gambar 1.1. Laporan Kaspersky Lab tentang pertumbuhan *malware* tahun 2013 (Kaspersky, 2013)

Seperti yang ditunjukkan dalam gambar 1.1, dalam laporan lain tentang pertumbuhan *malware* yang dipublikasi oleh Kaspersky Lab 2013, dipaparkan ada 315.000 *malware* baru setiap harinya (Kaspersky, 2013). Jika mengacu pada laporan Kaspersky Lab untuk perkiraan jumlah *signature malware* yang dibutuhkan, maka untuk mengenali dua juta *malware* dibutuhkan dua juta *signature* dan ada 315.000 sampel *malware* baru per harinya, dalam waktu sepuluh hari jumlah *signature* akan mencapai 5,15 juta. Asumsi *file signature* tidak dikompresi digunakan pada penghitungan sederhana sebelumnya dan *antivirus* yang dimaksud adalah *antivirus* dengan metode deteksi *signature-based detection*. Umumnya, setiap *malware* memiliki banyak sekali varian atau turunan. Bila menggunakan *hash function* seperti SHA256 yang bersifat mendeteksi

kesamaan suatu *file*, *hash malware* induk hanya dapat mendeteksi induknya, tidak bisa mendeteksi *malware* turunannya. Jadi, apabila suatu *signature-based antivirus* yang memiliki *database* SHA256 sebanyak 5,15 juta *signature*, *malware* yang dapat terdeteksi hanya 5,15 juta saja, tidak bisa mendeteksi *malware* yang *signature*-nya belum dibangun ke dalam *database* (Zeltser, 2011). Untuk memperbanyak jumlah *malware* yang dapat dideteksi, pengguna harus menunggu *antivirus* melakukan *update database signature*-nya secara berkala, mulai dari per hari sampai per bulan.

Dalam penelitian ini, keterbatasan *hash* tradisional seperti MD5, SHA-1, dan SHA256, dimana nilai *hash*-nya digunakan untuk membandingkan kesamaan *file* yang menghasilkan *output* apakah *file* identik atau tidak, akan digantikan dengan *fuzzy hashing* sebagai salah satu metode *hash* yang berbeda dari *hash* tradisional karena dengan menggunakan *fuzzy hash* dapat mendeteksi kemiripan *file* dengan *output* berupa rentang nilai dari nol (tidak mirip) sampai dengan satu (sangat mirip) (Kassner, 2011). Jadi, bila *signature-based antivirus* memiliki *database fuzzy hash* sebanyak 5,15 juta *signature*, *malware* yang dapat terdeteksi dapat melebihi 5,15 juta walau *signature malware* sebenarnya belum dibangun ke dalam *database*, terutama *malware* yang masih mirip atau varian dari *malware* yang telah terdapat dalam 5,15 juta *signature* dengan batas kemiripan tertentu. Hal ini dapat memberikan manfaat, yaitu penggunaan *fuzzy hash* dalam *database signature malware* dapat meningkatkan deteksi *malware* untuk jumlah data *hash* yang sama jika dibandingkan SHA256 sehingga *antivirus* dapat mendeteksi suatu *malware* baru yang belum terdapat pada *database signature*-nya dengan batas kemiripan tertentu berdasarkan *hash malware* yang sudah ada pada *database*.

Implementasi *fuzzy hash* dalam penelitian ini dibuat sebagai *signature malware* sederhana dan dilakukan pengujian jumlah tingkat deteksi dan akurasi deteksi.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, berikut rumusan masalah penelitian yang diajukan.

a. Bagaimana cara *database signature* yang mengimplementasi *fuzzy hash* dari suatu *malware* dihasilkan?
b. Berapa banyak jumlah peningkatan deteksi *malware* yang menggunakan *database signature* yang diperoleh dengan penggunaan *fuzzy hashing* jika dibandingkan SHA256 ?
c. Bagaimana tingkat akurasi deteksi *malware* menggunakan *fuzzy hash*?

1.3 Batasan Masalah

Berikut batasan masalah penelitian:

a. *malware* yang diuji terbatas pada *malware scripting* dan *malware executable* berekstensi EXE dan sampel *malware* yang digunakan adalah *malware* yang menyerang sistem operasi Windows;
b. perbandingan aspek tingkat deteksi dan akurasi deteksi *fuzzy hashing* terhadap *hash* lain terbatas pada SHA256;
c. *tools* yang digunakan dalam pengujian *database signature* merupakan aplikasi buatan penulis dan hanya sebagai simulasi membangun *database signature* dan simulasi deteksi *malware* menggunakan *hash* yang terdapat di dalam *database signature*. Tidak dibahas cara pembersihannya dan tidak dibuat aplikasi *removal*-nya.

1.4 Tujuan Penelitian

Tujuan dari penelitian adalah mengimplementasi *fuzzy hashing* untuk meningkatkan jumlah deteksi *malware* terhadap *hash* SHA256 dengan metode *signature-based detection*.

1.5 Manfaat Penelitian

Berikut manfaat yang didapat penelitian:

a. penggunaan *fuzzy hash* dalam *database signature malware* dapat meningkatkan deteksi *malware* untuk jumlah data *hash* yang telah diketahui yang sama jika dibandingkan SHA256;
b. *antivirus* dapat mendeteksi atau menduga suatu *malware* baru yang belum terdapat pada *database signature*-nya dengan batas kemiripan tertentu berdasarkan *hash malware* yang sudah ada pada *database* sehingga untuk mendeteksi suatu *malware* baru tidak terlalu bergantung dan menunggu pada *update signature* baru oleh *developer antivirus*.

1.6 Sistematika Penulisan

Guna memahami lebih jelas laporan skripsi ini, dilakukan dengan cara mengelompokkan materi menjadi beberapa sub-bab dengan sistematika penulisan sebagai berikut.

a. Bab I : Pendahuluan

   Bab ini menjelaskan tentang latar belakang penelitian, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penulisan.

b. Bab II : Landasan Teori

Bab ini berisikan teori-teori yang berupa definisi, rumus, dan algoritma yang berkaitan dengan penelitian, yaitu penjelasan tentang *malware, file signature, malware signature,* proses *traditional hashing* secara umum, proses *fuzzy hashing, levenshtein distance* untuk menghitung kemiripan *fuzzy hash*, metode deteksi *malware* dengan *signature-based detection*, dan rumus untuk menghitung tingkat akurasi dan deteksi.

c. Bab III : Metode dan Perancangan Sistem

Bab ini berisikan tentang pengembangan dan perancangan *library* dan aplikasi, serta *tools* yang digunakan untuk menunjang pengembangan aplikasi.

d. Bab IV : Implementasi dan Uji Coba

Bab ini berisikan tentang penjelasan dalam pengoperasian secara bertahap aplikasi yang dikembangkan beserta analisis hasil uji coba yang dilakukan dengan delapan skenario.

e. Bab V : Simpulan dan Saran

Bab ini berisikan kesimpulan yang merupakan jawaban atas tujuan penelitian yang dikemukakan pada Bab I, keterbatasan dari penelitian, baik dalam kaitannya dengan kemampuan implementasi dan uji coba, maupun kendala lain yang akan menjadi masukan bagi penelitian berikutnya.