SKRIPSI

APLIKASI PENDETEKSI KEBERADAAN ROOTKIT MENGGUNAKAN TEKNIK *FILE INTEGRITY MONITORING*

Janice Sella Gracia

NPM: 2013730071

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2017

UNDERGRADUATE THESIS

APPLICATION FOR ROOTKIT EXISTENCE DETECTION USING FILE INTEGRITY MONITORING

Janice Sella Gracia

NPM: 2013730071

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2017

LEMBAR PENGESAHAN

APLIKASI PENDETEKSI KEBERADAAN ROOTKIT MENGGUNAKAN TEKNIK *FILE INTEGRITY MONITORING*

Janice Sella Gracia

NPM: 2013730071

Bandung, 15 Desember 2017

Menyetujui,

Pembimbing 8-1-18

Mariskha Tri Adithia, P.D.Eng

Ketua Tim Penguji

Aditya Bagoes Saputra, M.T.

Anggota Tim Penguji

Luciana Abednego, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

APLIKASI PENDETEKSI KEBERADAAN ROOTKIT MENGGUNAKAN TEKNIK *FILE INTEGRITY MONITORING*

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 15 Desember 2017

Janice Sella Gracia
NPM: 2013730071

ABSTRAK

Rootkit adalah sebuah *malware* yang sangat berbahaya bagi suatu komputer. *Rootkit* mampu memonitor segala aktivitas suatu komputer, berbahaya bagi komputer tanpa disadari oleh pengguna. Salah satu sifat *rootkit* adalah mengubah sistem, sehingga tidak dapat dideteksi. Untuk mengatasi permasalahan ini, diperlukan suatu cara untuk mendeteksi keberadaan *rootkit*.

Penelitian ini akan mencoba mendeteksi keberadaan *rootkit* melalui perubahan yang dilakukan *rootkit*. Isi dari *file-file* sistem komputer mengalami perubahan. Salah satu teknik yang dapat digunakan adalah teknik *file integrity monitoring*, sebuah teknik yang memastikan agar setiap *file* memiliki integritas yang sama. Integritas sebuah file dianggap sama memiliki arti bahwa isi dari file tidak berubah. Setiap *file* memiliki sebuah nilai *checksum*. *Checksum* adalah sebuah teknik untuk mendeteksi eror pada sebuah data yang direpresentasikan dalam bentuk angka. Apabila nilai *checksum* sebuah *file* berubah, maka integritas dari *file* tersebut berubah. Penerapan teknik *file integrity monitoring* akan dibantu dengan algoritma *Fletcher* untuk proses pembuatan *checksum*.

Hasil dari penelitian adalah penulis berhasil membuat sebuah aplikasi pendeteksi keberadaan *rootkit*. Aplikasi dibangun menggunakan *javax.swing library* dan *Scheduler Cron*. Aplikasi yang dibangun telah membuktikan bahwa keberadaan *rootkit* dapat dideteksi melalui pengecekan perubahan isi *file* pada komputer, meskipun cara ini tidak berlaku untuk jenis *rootkit* yang lain, seperti jenis *kernel mode rootkit*. *Kernel mode rootkit* memiliki cara kerja berbeda yang berkaitan dengan kerja *kernel* dalam sistem operasi komputer.

Kata-kata kunci: *rootkit*, *malware*, *file integrity monitoring*, Algoritma Fletcher, *checksum*, *javax.swing* , *library*, *Scheduler Cron*, *kernel*, *kernel mode rootkit*

ABSTRACT

Rootkit is a dangerous malware for a computer. Rootkit can monitor all activities which are working in a computer even without being noticed by a user. One of characteristics that rootkit has is able to make changes in some applications used in a computer, so it cannot be detected. Looking at the problem, there has to be a way needed to detect rootkit presence.

This research is aimed to detect rootkit presence by monitoring all the changes rootkit has made. Some content from files in a computer system get changed. One of techniques which will be used is file integrity monitoring, a technique that ensure all files should have the same integrity. The integrity of a file stay the same when the content of a file doesnt change. Every file will have a checksum value. *Checksum* is a technique in detecting error for a data which is being represented in numberic value. If a file get a modification, then the checksum value will be different. An application of file integrity monitoring technique will be helped by Fletcher's Algorithm in case making checksum for a file.

The result of this research proves that the writer of this research succeeded making an application which is able to detect rootkit presence. The application is build using javax.swing library dan Scheduler Cron. The application has proved rootkit presence can be detected from the changes happend in a content of a file, although not all kinds of rootkit can be detected by the same technique, such as kernel mode rootkit. Kernel mode rootkit has different way in invading an operating system which is related with kernel in the invaded operating system.

Keywords: rootkit, malware, file integrity monitoring, Fletcher's Algorithm, checksum, javax.swing, Scheduler Cron, kernel, kernel mode rootkit

Dipersembahkan untuk Tuhan dan orang tua tercinta ...

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa karena melalui berkat dan rahmat-Nya penulis dapat menyelesaikan skripsi yang berjudul "Aplikasi Pendeteksi Keberadaan Rootkit Menggunakan Teknik *File Integrity Monitoring*". Penulis menyadari penulisan skripsi ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Maka dari itu, penulis ingin mengucapkan terima kasih kepada:

- Kedua orang tua yang selalu memberi motivasi dan semangat ketika menghadapi kesulitan dalam penulisan skripsi ini.
- Joyireh, selaku adik dari penulis yang selalu memberikan semangat dan mengingatkan *deadline* pengumpulan skripsi.
- Pa Chandra, sebagai dosen pembimbing penulis yang memberikan bimbingan, wawasan, dan keyakinan bahwa penulis mampu menyelesaikan skripsi ini dengan baik.
- Cheria, selaku sahabat yang selalu mendukung dan memberikan berbagai candaan.
- Alvin, Dimas, dan Maudy, teman-teman terdekat penulis yang selalu siap sedia dan membantu penulis.
- WannaOne, selaku idola yang memberi hiburan ketika penulis sedang merasa bosan dengan berbagai revisi.
- Staff dan teman-teman magang pada Biro Kepegawaian yang telah memberikan kesempatan bekerja dan memberi semangat kepada penulis.

Semoga seluruh pihak yang telah membantu penulis selalu diberikan berkat dan anugerah dari Tuhan yang Maha Esa. Akhir kata, penulis memohon maaf bila terdapat kesalahan dan kekurangan dalam penyusunan skripsi ini. Semoga skripsi ini dapat berguna bagi semua pihak yang membutuhkan.

Bandung, Desember 2017

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini, dunia sangat bergantung pada keberadaan teknologi. Banyak kegiatan sehari-hari yang dibantu dengan kecanggihan perangkat teknologi, terutama komputer. Tanpa sadar data-data yang tersimpan dalam komputer menjadi berharga, seperti data pribadi sang pengguna komputer. *Hacker* merupakan orang-orang ahli yang mampu menyusup ke dalam suatu sistem komputer tanpa disadari [4]. Sering kali yang menjadi tujuan para hacker adalah untuk mendapatkan data yang penting, mengawasi semua pekerjaan yang dilakukan oleh sistem komputer, bahkan untuk menyebarkan *malware* dan merusak sistem komputer tersebut. *Malware* adalah *software* yang diciptakan dengan tujuan untuk mengubah, merusak, bahkan mencuri data-data yang tersimpan dalam sebuah komputer. Keberadaan *malware* cukup sulit untuk dideteksi [5]. Perlu ada peningkatan lebih lanjut dalam hal menjaga keamanan komputer terhadap *malware* yang saat ini banyak disebarkan.

Sistem operasi merupakan sebuah perangkat lunak yang berguna untuk melakukan kontrol dan manajemen operasi-operasi yang berjalan pada sebuah komputer. Sistem operasi menjadi penghubung antara pengguna dengan perangkat keras komputer. Tanpa ada sistem operasi, maka *software* yang meliputi aplikasi pada komputer tidak dapat berjalan. Beberapa fungsi dari sebuah sistem operasi adalah menyediakan tempat untuk sebuah aplikasi saat berjalan pada memori komputer, menyimpan data-data dari semua operasi yang dilakukan oleh komputer, dan melakukan penjadwalan untuk setiap operasi yang akan berjalan. Ada beberapa contoh sistem operasi yang telah dikenal oleh masyarakat, yaitu Linux, Windows, dan Mac OS [6].

Rootkit merupakah salah satu dari jenis *malware* yang mengancam keamanan sebuah komputer. *Rootkit* mampu memonitor segala hal yang sedang dilakukan pada suatu sistem operasi dari sebuah komputer dalam jarak jauh. Pada kasus Sony BMG yang terjadi di tahun 2005, sebuah *rootkit* ditanamkan pada setiap CD yang dihasilkan perusahaan Sony BMG dengan maksud untuk mencegah terjadi tindakan pembajakan CD. Dengan perkembangan lebih lanjut, *rootkit* dapat digunakan untuk melakukan segala sesuatu pada sebuah sistem operasi komputer, seperti memodifikasi sistem *file*, aplikasi, atau *library*. Selain itu, *rookit* dapat membawa sebuah *virus* ke dalam sistem operasi komputer dan *virus* dapat melumpuhkan kinerja sebuah komputer [4].

Rootkit memiliki beberapa jenis dengan cara kerja yang berbeda untuk menyamarkan keberadaan *rootkit* dalam sebuah sistem operasi komputer. *Rootkit* menanamkan diri sebelum operasi komputer dilakukan, disebut dengan *pre-execution*, sehingga keberadaan mereka tidak terlihat atau terdeteksi. Teknik untuk mendeteksi *rootkit* ada berbagai macam, salah satu yang teknik yang digunakan pada skripsi ini adalah teknik *integrity checking*. Pada teknik *integrity checking* dilakukan perbandingan nilai-nilai biner yang dimiliki oleh sebuah *file* [7]. Perangkat lunak yang dibuat bekerja pada sistem operasi Linux. Hal ini disebabkan oleh kemampuan untuk mengakses isi dari sistem operasi Linux lebih mudah dan jelas dibandingkan sistem operasi lain yang membutuhkan hak akses lebih spesifik, seperti Windows. Pengecekan keberadaan *rootkit* akan dilakukan pada 4 buah jenis direktori dalam sistem operasi Linux, yaitu "*/bin*", "*/sbin*", "*/usr/bin*", dan "*/usr/sbin*". Keempat buah direktori tersebut menampung file-file utama dari sistem yang mendukung jalan sistem operasi Linux, sehingga dibutuhkan pengecekan akan keberadaan *rootkit* [1].

Perangkat lunak yang dibangun berfokus pada pendeteksian untuk direktori sistem operasi dan direktori basis data. Pengecekan pada direktori yang digunakan basis data perlu dilakukan untuk memastikan *rootkit* tidak melakukan perubahan pada hasil deteksi [7]. Setiap hasil deteksi disimpan pada basis data menggunakan MySQL, agar pengecekan dapat dilakukan lebih cepat dan praktis. Perangkat lunak dibangun menggunakan *java* dan *Scheduler Cron* untuk membantu penjadwalan setiap pengecekan *rootkit* yang dilakukan. Perangkat lunak ini diharapkan dapat menjadi pembelajaran dalam pendeteksian sebuah *rootkit*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas, maka dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana cara kerja *rootkit* ?
2. Bagaimana teknik untuk mendeteksi keberadaan suatu *rootkit*?
3. Bagaimana cara pembuatan aplikasi untuk mendeteksi keberadaan sebuah *rootkit* pada suatu sistem operasi dalam sebuah komputer?

1.3 Tujuan

Tujuan dari skripsi topik ini adalah sebagai berikut:

1. Mempelajari *malware* berjenis *rootkit*.
2. Mempelajari teknik mendeteksi keberadaan *rootkit*, yaitu *file integrity monitoring*.
3. Membuat sebuah aplikasi yang mampu mendeteksi keberadaan sebuah *rootkit* dengan menggunakan teknik pendeteksi *rootkit* yang telah dipelajari.

1.4 Batasan Masalah

Batasan-batasan masalah yang terkait dengan topik ini adalah sebagai berikut:

1. Perangkat lunak ini akan melakukan pengecekan semua *file* yang terdapat dalam beberapa direktori sistem operasi komputer, yaitu direktori "*/bin*", direktori "*/sbin*", direktori "*/usr/bin*" dan direktori "*/usr/sbin*". Pengecekan pada direktori basis data juga dapat dilakukan.
2. Perangkat lunak ini akan mampu mendeteksi keberadaan *rootkit* menggunakan teknik yang dipelajari, yaitu *file integrity monitoring*.
3. Perangkat lunak ini akan mampu melakukan perhitungan *checksum* untuk *file*.
4. Perangkat lunak ini menghasilkan kumpulan *file* yang mengalami perubahan nilai *checksum*.
5. Perangkat lunak ini akan menggunakan lingkungan sistem operasi Linux.
6. Perangkat lunak ini menggunakan algoritma bernama *Fletcher Checksum*.
7. Perangkat lunak ini akan melakukan pendeteksian untuk *rootkit* berjenis *user mode rootkit*.

1.5 Metodologi

Langkah-langkah yang diambil untuk melakukan penelitian berdasarkan topik yang dipilih adalah sebagai berikut:

1. Melakukan studi pustaka untuk mempelajari beberapa teori pendukung penelitian berdasarkan topik yang terpilih, antara lain :
   - *malware* berjenis *rootkit*
   - teknik untuk mendeteksi keberadaan sebuah *rootkit*
   - teknik *checksum*
   - sistem operasi berbasis Linux
2. Mengidentifikasi dan menganalisa kebutuhan perangkat lunak yang mampu mendeteksi *rootkit*.
3. Melakukan perancangan perangkat lunak yang mampu mendeteksi keberadaan *rootkit* sesuai dengan kebutuhan yang telah dianalisa.
4. Mengimplementasikan seluruh perancangan perangkat lunak sesuai dengan perancangan yang ada.
5. Melakukan pengujian terhadap perangkat lunak yang telah dibangun.
6. Melakukan penulisan dokumentasi untuk penelitian yang dilakukan.

1.6 Sistematika Pembahasan

Sistematika pembahasan yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Bab 1 Pendahuluan : berisi mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan terkait topik yang terpilih.
2. Bab 2 Landasan Teori: berisi pembahasan teori-teori dasar yang terkait dengan topik yang terpilih, seperti teori mengenai *Rootkit*, teknik pendeteksian *rootkit*, *checksum*, sistem operasi Linux.
3. Bab 3 Analisis : berisi analisis mengenai kebutuhan perangkat lunak yang dibangun, seperti diagram *use-case*, diagram skenario, contoh studi kasus penerapan *checksum* pada sistem operasi Linux, diagram kelas, diagram aktivitas, dan perbandingan algoritma *Adler Checksum* dengan *Fletcher Checksum*.
4. Bab 4 Perancangan : berisi mengenai perancangan dari perangkat lunak yang dibangun, yaitu penjelasan secara rinci mengenai diagram aktivitas, perancangan basis data yang digunakan, diagram kelas yang diterapkan, dan perancangan antarmuka aplikasi.
5. Bab 5 Implementasi dan Pengujian : berisi mengenai implementasi seluruh perancangan perangkat lunak yang telah dilakukan. Dilanjutkan dengan tahapan uji coba terhadap fungsi dari perangkat lunak.
6. Bab 6 Kesimpulan dan Saran : berisi keseluruhan kesimpulan mengenai penelitian dari topik ini dan terdapat beberapa saran-saran untuk pengembangan penelitian ini lebih lanjut.