BAB 6

KESIMPULAN DAN SARAN

Bab ini akan membahas kesimpulan mengenai perangkat lunak dan pencapaian tujuan. Pada bab ini juga diberikan saran-saran untuk mengembangkan penelitian.

6.1 Kesimpulan

Berdasarkan penelitan dan pengembangan perangkat lunak, beberapa kesimpulan yang didapatkan adalah:

1. Penelitian ini telah mempelajari cara kerja *rootkit*. *Rootkit* bekerja dengan melakukan perubahan pada sistem komputer. Konten dari setiap *file* bawaan sistem telah diubah oleh *rootkit*.
2. Penelitian ini menunjukkan bahwa teknik *file integrity monitoring* dapat digunakan untuk mendeteksi perubahan yang dilakukan oleh *rootkit*. Teknik tersebut akan memeriksa konten dari sebuah *file*. Konten dari *file* direpresentasikan bentuk *checksum*. Apabila konten *file* berubah, maka *checksum* dari *file* mengalami perubahan juga.
3. Penelitian yang dilakukan telah berhasil mengembangkan perangkat lunak yang mampu mendeteksi *rootkit* berjenis *user mode*. Namun, pendeteksian terhadap jenis *rootkit* yang lain tidak dapat digunakan teknik *file integrity monitoring* yang diteliti.
4. Penelitian ini telah berhasil mengimplementasi aplikasi pendeteksi keberadaan *rootkit* dengan memanfaatkan:
   - Teknik mendeteksi *rootkit* yang bernama *file integrity monitoring* yang memastikan bahwa isi dari sebuah *file* harus tetap sama.
   - Algoritma *Fletcher Checksum* untuk mendeteksi perubahan setiap *file* melalui *checksum* yang dibentuk.
   - Penggunaan teknik *whitelisting* untuk memilih *file* yang legal untuk ditambahkan pada sebuah direktori.
5. Penelitian telah berhasil mengimplementasikan penjadwalan eksekusi perangkat lunak untuk memudahkan proses pengecekan pada sistem operasi. Pengimplementasian menggunakan *Scheduler Cron* yang merupakan sebuah penjadwalan untuk sistem operasi berbasis UNIX.

6.2 Saran

Berdasarkan penelitian dan pengembangan perangkat lunak yang dilakukan, berikut adalah beberapa saran untuk pengembangan lebih lanjut:

- Pengembangan perangkat lunak yang dilakukan adalah pendeteksian *rootkit* berjenis *user mode*. Pengembangan yang disarankan adalah pendeteksian terhadap *rootkit* berjenis *kernel mode* karena masih banyak tersebar luas dan terus berkembang.

- Pengembangan perangkat lunak adalah melakukan pengamanan pada direktori basis data. Pengembangan yang disarankan adalah diperlukan penambahan cara lain dalam menjaga direktori basis data dari ancaman *rootkit*, karena direktori basis data berperan penting dalam proses mendeteksi *rootkit*.
- Pengembangan perangkat lunak yang dilakukan adalah penggunaan algoritma *Fletcher Checksum* untuk membantu proses pendeteksian *rootkit*. Pengembangan yang disarankan adalah melakukan perbandingan *Fletcher Checksum* dengan algoritma *checksum* lain yang lebih efisien untuk menyesuaikan perkembangan pada algoritma *checksum*.
- Pengembangan perangkat lunak menggunakan teknik *whitelisting* untuk memilih *file* yang legal agar tidak dianggap *rootkit*. Namun, teknik ini hanya sebatas memilih *file* yang diperbolehkan ditambahkan ke dalam sebuah direktori. Pengembangan yang disarankan adalah pemilihan dilakukan juga untuk *file* yang telah dimodifikasi *rootkit* ataupun terhapus dalam sebuah direktori.

DAFTAR REFERENSI

[1] Bunten, A. (2004) Unix and linux based rootkits techniques and countermeasures. *In Proc of the 16th FIRST Conference on Computer Security Incident Handling*, Hamburg, Germany, 30 April, pp. 2–10. DFN-CERT Services GmbH.

[2] Koopman, P. (2012) Selection of cyclic redundancy code and checksum algorithms to ensure critical data integrity. Technical Report DOT/FAA/TC-14/49. Federal Aviation Administration, United States.

[3] Maxino, T. C. (2006) The effectiveness of checksums for embedded networks. Thesis. Carnegie Mellon University.

[4] Davis, M., Bodmer, S., dan Lemasters, A. (2010) *Hacking Exposed Malware & Rookits : Malware & Rootkits Security Secrets & Solutions*, 2nd edition. The McGraw-Hill, United States.

[5] McGraw, G. dan Morrisett, G. (2000) Attacking malicious code: A report to the infosec research council. Technical Report 0740-7459. Infosec Research Council, IEEE.

[6] Tanenbaum, A. S. (2009) *Modern Operating System*, 3rd edition. Prentice Hall, The Netherlands.

[7] Armstrong, D. (2003) An introduction to file integrity checking on unix systems. Technical Report 104739. SANS Institute, United States.

[8] Alzarooni, K. M. A. Y. (2012) Malware Variant Detection. Disertasi. University College London, United Kingdom.

[9] Smith, S. dan Harrison, J. (2012) Rootkits - symantec. *Symantec Security Response*, **1**, 2–4.

[10] Zovi, D. D. (2001) Kernel rootkits. Technical Report 449. SANS Institute, United States.

[11] THOMAS MARTIN ARNOLD, B. (2011) A comparative analysis of rootkit detection techniques. Thesis. University of Houston Clear Lake.

[12] Stallings, W. (2005) *The Linux Operating System*, 5th edition. Prentice Hall, United States.

[13] Sivathanu, G., Wright, C. P., dan Zadok, E. (2005) Ensuring data integrity in storage: Techniques and applications. *Proceedings of the First ACM Workshop on Storage Security and Survivability*, New York, USA, November, pp. 1–6. ACM, New York.

[14] Garrels, M. (2007) *Introduction to Linux - A Beginer's Guide*, 2nd edition. Fultus Publishing, United States.