BAB 6

KESIMPULAN DAN SARAN

Pada bab kesimpulan dan saran ini akan dibahas mengenai kesimpulan yang didapat dari hasil beserta saran yang dapat diberikan berdasarkan implementasi dan pengujian penelitian yang telah dilakukan. Diharapkan dengan adanya bab ini dapat membantu penelitian-penelitian yang berkaitan dengan topik ini. Dengan demikian penelitian-penelitian tersebut dapat dilakukan dengan lebih baik.

6.1 Kesimpulan

Dengan tujuan penelitian yang telah disampaikan pada sub-bab 1.3, penulis dapat mempelajari tentang *SQL Injection*, mempelajari cara untuk mendeteksi serangan *SQL Injection* pada *website*, mempelajari cara untuk memblokir serangan *SQL Injection*, dan membuat aplikasi yang dapat mendeteksi dan memblokir serangan *SQL Injection.* Perangkat lunak monitoring keamanan *website* terhadap serangan *SQL Injection* ini diharapkan dapat menjadi alternatif dalam mendeteksi dan memblokir para *hacker* yang melakukan serangan dengan teknik *SQL Injection* terhadap *server* yang menjalankan *web server* pada port 80 dan *database server* berbasis SQL sebagai media penyimpanan datanya.
Pada proses pembangunan perangkat lunak, penulis mempelajari lebih dalam tentang teknik serangan *SQL Injection* dan berbagai pengetahuan dibidang kemanan informasi serta bagaimana melakukan pembacaan dan pemblokiran paket menggunakan *library Java*, jNetPcap. Dari penilitian ini, penulis dapat menarik beberapa kesimpulan.

1. *SQL Injection* adalah serangan terhadap suatu aplikasi yang menggunakan basis data *SQL* dengan cara memanfaatkan *query* untuk memanipulasi data yang tidak sesuai dengan fungsi aplikasi tersebut. Dengan demikian, *SQL Injection* merupakan salah satu teknik untuk meretas agar mendapatkan data-data yang mungkin saja bersifat sensitif seperti *password* ataupun data kartu kredit.

2. Serangan *SQL Injection* dapat dideteksi dengan cara melakukan pengecekan terhadap masukan *query* dari pengguna aplikasi. Terdapat kata-kata yang sensitif terhadap pemrosesan *query* tersebut misalnya kata "select union all" yang secara terurut dapat mengakibatkan munculnya pesan kesalahan pada *server* aplikasi.

3. Pendeteksian serangan *SQL Injection* didasari pada paket yang berisi *query* dan mengandung *SQL Injection.* Sebelum dapat mendeteksi paket tersebut adalah sebuah *SQL Injection* atau bukan, maka diperlukan suatu cara untuk melakukan pengecekan terhadap setiap paket yang masuk. Setiap paket yang masuk dideteksi apakah mengandung *query* atau tidak. Jika tidak mengandung *query* maka paket tersebut akan diabaikan. Tetapi, jika paket tersebut mengandung *query* maka setiap masukan dari *query* tersebut harus dilakukan pengecekan apakah terdapat *SQL Injection* atau tidak. Caranya dengan mendeteksi apakah pada *query* tersebut memiliki kata-kata yang sensitif terhadap perintah *SQL.*

4. Berdasarkan pengujian fungsional pada Gambar 5.11 dan Gambar 5.15, sistem perangkat lunak berhasil membaca paket sebagai serangan *SQL Injection* dan dicatat pada basis data di tabel data penyerang sebagai penyerang yang diblokir saat pengujian fungsional dengan cara menyerang halaman daftar buku yang dapat dieksploitasi pada *dummy web*. Sistem perangkat lunak memanggil perintah *iptables* untuk memblokir alamat IP penyerang kepada tujuan alamat IP. Pada konfigurasi *server*, peraturan *iptables* yang telah ditambahkan oleh sistem perangkat lunak akan dimasukan kedalam berkas konfigurasi yang akan dibaca oleh *iptables* saat proses *booting* sehingga peraturan-peraturan tersebut tidak akan hilang walaupun *server* mati. Sistem perangkat lunak menambahkan peraturan *iptables* yang baru ke dalam berkas konfigurasi *iptables* yang dimaksud. Berkas konfigurasi dapat dilihat di Lampiran B.

5. Berdasarkan pengujian eksperimental pada Gambar 5.15 dan Gambar 5.14, sistem perangkat lunak tampak berhasil membaca paket yang dikirimkan oleh *SQLMap* sebagai serangan *SQL Injection*. Serangan tersebut dicatat pada basis data di tabel data penyerang sebagai penyerang yang diblokir. Akan tetapi terdapat paket serangan lain yang terlebih dahulu masuk sebelum paket pertama dideteksi memiliki *SQL Injection query* seperti pada Gambar 5.17, Gambar 5.18, dan Gambar 5.19. Hal tersebut dikarenakan *SQLMap* mengirimkan *SQL Injection query* secara *spam* sehingga sebelum paket diproses oleh sistem perangkat lunak, ada paket lain yang terlebih dahulu masuk dan diproses oleh *web server*.

6. Beberapa *tools* pada Linux, Ubuntu yang dapat digunakan sebagai kebutuhan pembangunan sistem perangkat lunak monitoring keamanan *website* terhadap serangan *SQL Injection*. *Tools* tersebut antara lain.

   - *iptables*. Digunakan untuk melakukan seleksi terhadap paket-paket yang melalui/masuk ke dalam jaringan *server*. Pada implementasi dan pengujian digunakan sebagai cara memblokir penyerang yang terdeteksi oleh sistem perangkat lunak. Pengujiannya dapat dilihat pada Gambar 5.10.
   - *gksu*. Digunakan untuk menjalankan aplikasi NetBeans agar dapat mengakses kernel *server* yang membutuhkan *super user permission*/root.

6.2 Saran

Dengan dibangunnya sistem perangkat lunak monitoring kemanan *website* terhadap serangan *SQL Injection* ini diharapkan dapat membantu para administrator *web server* dalam memantau adanya serangan dengan teknik *SQL Injection* terhadap *server*. Agar dapat menjadikan hal tersebut lebih baik lagi, penulis menyadari sistem perangkat lunak yang dibangun ini tentu memerlukan saran dalam pengembangan dan penyempurnaan.

1. Pengembangan fitur pemilihan direktori untuk menyalin dan membaca paket. Fitur ini sangat mungkin untuk ditambahkan dan dibutuhkan untuk mempermudah memilih direktori yang ingin digunakan.

2. Pendeteksian paket dengan sistem perangkat lunak masih sangat lambat karena hal-hal yang tidak diduga. Pengembangan pada bagian ini sangat dibutuhkan agar perangkat lunak dapat berjalan lebih efisien dan cepat.

3. Kegunaan sistem perangkat lunak ini harus ditambahkan menjadi dibeberapa sistem operasi lain seperti Windows maupun UNIX. Walaupun berbeda dalam *tools* yang digunakan, implementasi pada seluruh sistem operasi yang cukup dikenal seperti Windows dan Unix adalah hal penting.

DAFTAR REFERENSI

[1] Halfond, W. G., Viegas, J., dan Orso, A. (2008) A classification of sql-injection attacks and countermeasures. *PREVENTING SQL CODE INJECTION BY COMBINING STATIC AND RUNTIME ANALYSIS*, **1**, 53.

[2] Yadav, S. C. dan Singh, S. K. (2009) *An Introduction to CLIENT/SERVER COMPUTING*. New Age International, New Delhi.

[3] Cobbaut, P. (2007) Linux fundamentals. Netsec BVBA.

[4] Aulds, C. (2001) Linux apache web server administration. SYBEX.

[5] Nugroho, P. A. (2016) Restful web service dengan http dan json. Lecturing Guide Book.

[6] Team, U. D. (2016) Ubuntu server guide. Official Ubuntu Documentation Website.

[7] Groff, J. R. dan Weinberg, P. N. (2011) *Database System Concepts*, 6th edition. McGraw-Hill, New York.

[8] Groff, J. R. dan Weinberg, P. N. (1999) *SQL: The Complete Reference*. McGraw-Hill, Osborne.

[9] Boneh, D. (2009) Sql injection: attacks and defenses. Lecture Presentation.

[10] Purdy, G. N. (2004) *LINUX iptables PICKET REFERENCE*. O'Reilly Media, California.

[11] Weiss, A. (2016) How to prevent sql injection attacks. http://www.esecurityplanet.com/hackers/how-to-prevent-sql-injection-attacks.html. 30 Mei 2017.