



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pembahasan sebelumnya, dapat disimpulkan bahwa metode *Hybrid port knocking* telah berhasil diimplementasikan pada aplikasi *client-server* melalui *virtual machine*. *Hybrid port knocking* dapat dijadikan *layer* keamanan tambahan untuk melindungi *server* dan meminimalkan terjadinya serangan karena menggabungkan beberapa teknik seperti *port knocking*, steganografi, serta *mutual authentication*.

Hasil yang diperoleh dari pengujian konsumsi waktu terhadap *hybrid port knockingserver* dan *client script* adalah *server* dan *client script* berbasis DSA and ElGamal *public key algorithm* mengkonsumsi waktu yang paling lama dibandingkan dengan yang lainnya. Dimana rata-rata waktunya adalah 1,074 s untuk membuka *port*, 1,032 s untuk mengirim *echo message*, 1,03 s untuk me-restart layanan pada *server* misalnya layanan *httpd*, dan 1,078 untuk menutup *port*, sedangkan untuk *client* adalah 2,08 s untuk membuka *port*, 2,058 s untuk mengirim *echo message*, 2,054 untuk me-restart layanan pada *server*, dan 2,082 s untuk menutup *port*.

Hasil yang diperoleh dari pengujian konsumsi memori terhadap *hybrid port knockingserver* dan *client script* adalah *server* dan *client script* berbasis DSA and ElGamal *public key algorithm* mengkonsumsi memori paling banyak dengan rata-rata penggunaan pada *server* sebesar 20,27 MiB setara dengan 21,25 MB untuk membuka *port*, 20,026 MiB setara dengan 20,99 MB untuk mengirimkan *echo message*, 20,034 MiB setara dengan 21,007 MB untuk me-restart layanan pada *server* serta 20,304 MiB setara dengan 21,29 MB untuk menutup kembali *port* yang telah dibuka. Penggunaan memori pada *client* rata-rata sebesar 20,308 MiB setara dengan 21,12 MB untuk

membuka *port*, 20,08 MiB setara dengan 20,88 MB untuk mengirimkan *echo message*, 20,094 MiB setara dengan 20,89 MB untuk me-*restart* layanan yang disediakan oleh *server*, dan 20,332 MiB setara dengan 21,14 MB untuk menutup *port*.

Sehingga disimpulkan bahwa pemilihan jenis algoritma yang digunakan berpengaruh terhadap konsumsi waktu serta konsumsi memori. Semakin lama proses yang dilakukan oleh sebuah algoritma dalam melakukan proses enkripsi dan dekripsi maka akan mempengaruhi konsumsi waktu dan memori. Selain itu request yang dikirimkan oleh *client* juga berpengaruh terhadap konsumsi waktu dan memori. Request untuk membuka dan menutup port memerlukan waktu dan memori yang lebih banyak dibandingkan dengan request lainnya.

5.2 Saran

Penelitian ini masih memiliki berbagai kekurangan. Maka dari itu dibutuhkan saran agar aplikasi ini dapat menjadi lebih baik. Saran yang diusulkan adalah dengan membuat *Graphical User Interface* (GUI) yang terintegrasi dengan aplikasi *hybrid port knocking*, dimana terdapat kolom isian untuk *client's* login lalu *client* dapat berkomunikasi dengan *server*.

U
M
M
N