

BAB V PENUTUP

Bab ini merupakan bagian akhir dari skripsi yang menyajikan intisari dari seluruh hasil pengujian dan analisis data yang telah dijabarkan sebelumnya.

5.1 Kesimpulan

Berdasarkan hasil eksperimen dan analisis komparatif terhadap metode *Fail2Ban* dan *ModSecurity* dalam memitigasi serangan DDoS lapisan aplikasi (*HTTP Flood*), diperoleh kesimpulan sebagai berikut:

1. Keduanya efektif mencegah kelumpuhan server (*downtime*). *ModSecurity* memblokir serangan di lapisan aplikasi (*Layer 7*) dengan mengirimkan status HTTP 403 *Forbidden*, sedangkan *Fail2Ban* memblokir akses secara total di lapisan jaringan (*Layer 3/4*) melalui *firewall* (*iptables*).
2. Kecepatan deteksi, *ModSecurity* bereaksi seketika (0,1 detik) tanpa membiarkan paket masuk ke *backend*, sedangkan *Fail2Ban* memiliki jeda waktu (*delay*) 1 hingga 2 detik karena harus menunggu sistem selesai mencatat *log* kegagalan.
3. Beban Komputasi (*Resource Usage*), *Fail2Ban* jauh lebih hemat sumber daya karena CPU langsung kembali stabil di 0.0% - 2.8% setelah IP diblokir atau di *ban*, berbanding terbalik dengan *ModSecurity* yang membuat CPU bekerja sangat berat hingga 158,1% karena sistem terus-menerus merespons serangan.

5.2 Saran

Berdasarkan hasil penelitian serta keterbatasan yang dialami selama proses pengerjaan skripsi, terdapat beberapa saran yang dapat diberikan untuk perbaikan dan pengembangan lebih lanjut:

1. Penerapan keamanan *hibrida* (Integrasi Sistem) penelitian selanjutnya sangat disarankan untuk mengimplementasikan dan menguji arsitektur keamanan *hibrida* (*Embedded Firewall*) yang menggabungkan *ModSecurity* dan *Fail2Ban* secara bersamaan. Skenario yang disarankan

adalah menggunakan kecepatan *ModSecurity* untuk mencetak *log error* deteksi instan, yang kemudian secara cepat dibaca oleh *Fail2Ban* untuk memblokir IP di tingkat *firewall*. Hal ini akan menutupi kelemahan jeda waktu di *Fail2Ban* sekaligus mengeliminasi beban komputasi jangka panjang dari *ModSecurity*.

2. Peningkatan instrumen akuisisi data kinerja pada penelitian mendatang, disarankan untuk mengintegrasikan alat monitoring *resource* tingkat lanjut berbasis deret waktu (*time-series database*) seperti kombinasi *Prometheus* dan *Grafana*, guna menghasilkan visualisasi grafik CPU dan RAM yang otomatis dan lebih akurat.
3. Pengujian pada vektor serangan tingkat lanjut skripsi ini berfokus pada mitigasi ancaman kehabisan sumber daya menggunakan modul penelitian serangan berbasis volume *request* seperti HULK. Untuk melihat batas ketahanan *server*, peneliti berikutnya dapat menguji sistem keamanan terhadap vektor serangan DDoS *low-and-slow* (seperti *Slowloris* atau *R-U-Dead-Yet*), maupun serangan asimetris terdistribusi (*botnet multi-IP* skala besar) untuk melihat apakah *Fail2Ban* dan *ModSecurity* mampu beradaptasi pada variasi serangan yang lebih kompleks.