BAB I
PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi yang pesat telah meningkatkan efisiensi dan produktivitas di berbagai sektor, namun di sisi lain juga meningkatkan kerentanan terhadap ancaman siber yang semakin canggih [1], [2]. *Malware* modern kini memanfaatkan berbagai teknik untuk menghindari deteksi dan menembus sistem keamanan tradisional [1]. Dalam konteks ini, keamanan kernel memiliki peran sentral dalam menjaga keandalan sistem operasi modern terhadap ancaman tersebut [3].

Salah satu ancaman siber yang paling sulit dideteksi saat ini adalah *malware* tanpa file (*fileless malware*), yaitu jenis *malware* yang beroperasi langsung di dalam memori utama tanpa menyimpan file eksekusi di penyimpanan permanen [4], [5]. Berbeda dengan *malware* tradisional yang meninggalkan jejak pada hard disk, *fileless malware* sepenuhnya berjalan di memori, sehingga tidak dapat dianalisis oleh antivirus berbasis *signature*. Teknik serangan umum meliputi penyematan kode ke proses sah atau penggunaan fungsi *memfd_create()* dan *fexecve()* untuk menjalankan file anonim di memori tanpa menyentuh sistem file [6]. Akibatnya, deteksi hanya dapat dilakukan melalui pemantauan perilaku atau analisis forensik memori, yang membutuhkan pendekatan lebih kontekstual dan *real-time* [4], [5].

Ancaman *fileless malware* juga meningkat signifikan pada sistem operasi Linux, yang digunakan secara luas di server, perangkat IoT, dan lingkungan *cloud*. Hanchenko [7] melaporkan bahwa tingkat keberhasilan serangan *fileless malware* sepuluh kali lebih tinggi dibandingkan *malware* berbasis file, dengan peningkatan 94% pada tahun 2018. Laporan AT&T Alien Labs dan Aqua Security [7] juga menunjukkan bahwa *malware* jenis ini kini mulai menyerang aplikasi berbasis kontainer, di mana proses deteksinya diperumit oleh keterbatasan alat pemantauan. Wu et al. [8] menemukan bahwa tren ini juga memengaruhi server web berbasis PHP, di mana kode berbahaya dapat dijalankan langsung dari aplikasi web, sedangkan Lima et al. [9] menegaskan bahwa hampir seluruh

malware di server web modern berasal dari kode PHP berbahaya. Bahkan, penelitian Tóth et al. [10] menunjukkan bahwa kode PHP yang dihasilkan oleh *Large Language Models* (LLMs) memiliki tingkat kerentanan sangat tinggi terhadap eksploitasi.

Untuk menghadapi ancaman tersebut, Linux dilengkapi dengan mekanisme keamanan kernel berbasis *Mandatory Access Control* (MAC) seperti Security-Enhanced Linux (SELinux) dan AppArmor [11], [12]. SELinux menerapkan kontrol berbasis label (*label-based policy enforcement*) dengan cakupan luas, sementara AppArmor menggunakan pendekatan berbasis jalur (*pathname-based*) yang lebih sederhana dan mudah dikonfigurasi [13], [14]. Keduanya efektif dalam membatasi akses proses, namun memiliki karakteristik berbeda: SELinux unggul dalam visibilitas audit, sedangkan AppArmor lebih ringan secara performa. Zhang et al. [15] menemukan bahwa *authorization hooks* dalam modul keamanan Linux dapat meningkatkan *overhead* hingga 87% pada operasi sistem file, sedangkan Han dan Lee [12] menyebutkan bahwa kedua modul belum menyediakan pemantauan file secara *real-time*. Hal ini menegaskan perlunya evaluasi empiris untuk menilai keseimbangan antara efektivitas deteksi dan dampak performa sistem dari kedua modul keamanan kernel tersebut.

Berdasarkan kondisi tersebut, penelitian ini dilakukan untuk menganalisis efektivitas deteksi dan pencegahan serangan *fileless malware* serta membandingkan dampak performa sistem (CPU dan RAM) antara SELinux dan AppArmor pada aplikasi web berbasis PHP di sistem operasi Debian. Melalui pendekatan eksperimental di lingkungan virtual, penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem keamanan Linux yang lebih efisien, adaptif, dan tangguh dalam menghadapi ancaman *fileless malware* di masa depan.

B. Perumusan Masalah

Berdasarkan latar belakang masalah dan identifikasi celah penelitian, rumusan masalah dalam penelitian ini dirumuskan dalam bentuk pertanyaan-pertanyaan penelitian yang akan dijawab secara spesifik melalui eksperimen:

1. Bagaimana efektivitas SELinux dan AppArmor dalam mendeteksi dan mencegah serangan *fileless malware* pada kerentanan aplikasi web PHP di sistem operasi Debian?
2. Bagaimana perbandingan dampak performa sistem (penggunaan CPU dan RAM) antara SELinux dan AppArmor saat menghadapi serangan *fileless malware* pada aplikasi web PHP?

C. Batasan Penelitian

Dalam penelitian ini, ruang lingkup akan difokuskan pada aspek-aspek tertentu agar proses analisis, implementasi, dan evaluasi dapat dilakukan secara terarah dan mendalam. Mengingat kompleksitas sistem keamanan Linux dan beragamnya bentuk serangan *fileless malware*, maka penulis membatasi cakupan studi hanya pada skenario dan variabel yang relevan langsung dengan tujuan penelitian. Batasan ini ditetapkan untuk memastikan validitas eksperimen, terukur hasil, serta kesesuaian dengan sumber daya dan waktu yang tersedia. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Jenis Serangan yang dianalisis terbatas pada *fileless malware* yang dieksekusi melalui kerentanan aplikasi web PHP, tanpa melibatkan *malware* berbasis file.
2. Aplikasi Web yang digunakan adalah aplikasi web berbasis PHP, yang memiliki kerentanan PHP sebagai titik masuk simulasi serangan.
3. Lingkungan Uji dilakukan secara lokal menggunakan mesin virtual (VM) berbasis sistem operasi Linux (Debian), untuk menjamin keamanan dan kontrol penuh terhadap eksperimen.
4. Teknologi Keamanan yang diuji dibatasi pada dua mekanisme utama: SELinux dan AppArmor, dengan konfigurasi default serta penyesuaian minimal menggunakan kebijakan tambahan (*custom policy*).
5. Penelitian ini menyertakan pengujian pada sistem operasi tanpa modul keamanan aktif (*No Protection*) sebagai data dasar (*baseline*) untuk mengukur seberapa besar peningkatan keamanan dan beban performa yang ditimbulkan oleh pengaktifan SELinux dan AppArmor

6. Parameter yang dievaluasi mencakup kemampuan deteksi, pencegahan eksekusi *malware*, pencatatan log, serta dampak performa sistem (CPU/RAM), tanpa menganalisis aspek kriptografi, *sandboxing*, atau mitigasi jaringan secara mendalam.
7. Sumber *fileless malware* adalah simulasi internal dengan script yang dirancang untuk *payload fileless* berbasis curl dengan berbagai karakteristik.

D. Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Menganalisis efektivitas SELinux dan AppArmor dalam mendeteksi dan mencegah serangan *fileless malware* pada kerentanan aplikasi web PHP di sistem operasi Debian.
2. Membandingkan dampak performa sistem (penggunaan CPU dan RAM) antara SELinux dan AppArmor saat menghadapi serangan *fileless malware* pada aplikasi web PHP.

E. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat baik secara teoritis maupun praktis bagi berbagai pihak yang berkepentingan dalam bidang keamanan sistem dan jaringan. Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Administrator Sistem: Memberikan panduan berbasis bukti bagi administrator sistem dalam memilih modul keamanan (SELinux atau AppArmor) yang paling sesuai untuk aplikasi web berbasis PHP di lingkungan Linux, dengan mempertimbangkan efektivitas deteksi dan pencegahan serangan *fileless malware*, serta dampak performa sistem.
2. Penyedia Layanan Keamanan: Memberikan data empiris yang dapat digunakan untuk solusi keamanan yang ada atau mengembangkan alat deteksi dan mitigasi baru untuk *fileless malware* di lingkungan Linux.
3. Penelitian Keamanan Siber: Menambah literatur penelitian di bidang keamanan sistem operasi Linux dan *fileless malware*, khususnya dalam konteks perbandingan antara SELinux dan AppArmor.

4. Dasar Penelitian Lanjutan: Hasil penelitian ini dapat menjadi dasar untuk penelitian lanjutan terkait optimalisasi konfigurasi modul keamanan, pengembangan deteksi *fileless malware* yang lebih canggih, atau analisis dampak pada jenis serangan siber lainnya.