Sistem Operasi (INF-201)

Pertemuan 14: Keamanan & Proteksi Kernel

Tim Dosen Sistem Operasi

Program Studi Informatika – Universitas Bengkulu

Semester Ganjil 2026/2027

Capaian Pembelajaran (Sub-CPMK 14)

- Mahasiswa mampu menganalisis mekanisme proteksi (ACL, *Capability Lists*).
- Mahasiswa memahami serangan *Buffer Overflow* (*stack smashing*) dan teknik mitigasinya (ASLR, DEP/NX, Stack Canaries).
- Mahasiswa mampu mengevaluasi model keamanan spesifik seperti OpenBSD (`pledge`, `unveil`, W^X) vs SELinux (MAC).
- Mahasiswa memahami mekanisme *Unix file permissions* (`chmod`, `setuid`).

Mekanisme Proteksi Domain

Access Control List (ACL) vs Capability List

- **ACL:** Daftar hak akses yang disimpan **pada objek** (misal: File). Mendata subjek mana yang bisa mengakses.
- **Capability List:** Daftar hak akses yang dipegang oleh **subjek/proses**. Seperti tiket/kunci untuk mengakses berbagai objek.

Unix File Permissions (DAC - Discretionary Access Control)

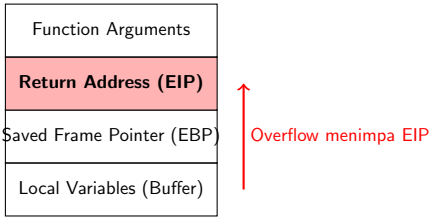
- `rwx` untuk User, Group, dan Others (`chmod`).
- **SetUID (`suid`):** Memungkinkan proses berjalan dengan hak istimewa pemilik file (misal: perintah `passwd` berjalan sebagai `root`).

Eksploitasi Keamanan: Buffer Overflow

Stack Smashing / Buffer Overflow

- Terjadi ketika program menulis data melebihi batas buffer di memori (sering karena fungsi tak aman seperti `strcpy()`).
- Penyerang menimpa *Return Address* di *Stack Frame* sehingga menunjuk ke *malicious code* / *shellcode*.
- Varian lain: *Return-to-libc* (jika *shellcode* dilarang).

Ilustrasi Memori (Stack):



Teknik Mitigasi Keamanan Modern

- **ASLR (Address Space Layout Randomization):** Mengacak alamat awal Heap, Stack, dan Libraries sehingga penyerang sulit menebak alamat memori eksploit.
- **DEP / NX Bit (Data Execution Prevention / No-Execute):** Menandai area memori data (seperti Stack/Heap) sebagai tidak dapat dieksekusi (*non-executable*). Mencegah eksekusi *shellcode*.
- **Stack Canaries:** Menyisipkan nilai acak ("canary") sebelum *Return Address*. Jika buffer overflow terjadi, nilai canary akan berubah, dan program akan terminasi (*abort*) sebelum *Return Address* dipanggil.

Model Keamanan Proaktif: OpenBSD (pledge & unveil)

OpenBSD Security Model (W^X - Write XOR Execute)

- Prinsip: Halaman memori boleh ditulisi (Write) **ATAU** dieksekusi (Execute), tapi **TIDAK** keduanya secara bersamaan.

Pembatasan Akses: pledge() & unveil()

```
#include <unistd.h>
int main() {
    // Membatasi akses direktori hanya /var/www (read-only)
    unveil("/var/www", "r");
    unveil(NULL, NULL); // Kunci unveil

    // Membatasi kemampuan proses hanya untuk baca/tulis dasar
    // System call lain (misal: exec, network) akan memicu SIGABRT
    pledge("stdio rpath", NULL);

    // Aplikasi berjalan...
}
```

Tugas Membaca & Proyek Akhir

Persiapan Pertemuan 15: Virtualisasi & Container

Sebelum kuliah terakhir minggu depan, baca:

- **[OSTEP]** Chapter 50: *Virtual Machines & Container Isolation*
- **[TLCL]** Chapter 36: *Networking & Security Basics*

Capstone Mini-Project

Persiapkan presentasi *Capstone Mini-Project* Anda (Deploy Microservice dengan Docker Container) untuk minggu depan (Minggu 15).