**PROJECT 2**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ tên | Phân công | Mức độ hoàn thành |
| 1712358 | Nguyễn Minh Đức | Phần 1 | 100% |
| 1712369 | Phạm Quốc Dũng | Phần 2 open | 100% |
| 1712379 | Đặng Thành Duy | Phần 2 write | 100% |

* Mô tả tổ chức/thiết kế của đồ án:
  + Các hàm chính:
    - Phần 1:
* *static* *int* dev\_open(*struct* inode\* *i*, *struct* file\* *f*)

Mở thiết bị

* *static* *int* dev\_close(*struct* inode\* *i*, *struct* file\* *f*)

Tắt thiết bị

* *static* *ssize\_t* dev\_read(*struct* file\* *f*, *char* \_\_user\* *buf*, *size\_t* *len*, *loff\_t*\* *off*)

Đọc thiết bị

* *static* *int* \_\_init mod\_init(*void*)

Khởi tạo thiết bị

* *static* *void* \_\_exit mod\_exit(*void*)

Hủy thiết bị

* + - Phần 2:
* *void* get\_filename(*const* *char*\* *pathname*)

Lấy tên file từ đường dẫn

* asmlinkage *int* hook\_open(*const* *char*\* *pathname*, *int* *flags*, *mode\_t* *mode*)

Hook syscall open

* asmlinkage *ssize\_t* hook\_write(*int* *fd*, *const* *void*\* *buf*, *size\_t* *count*)

Hook syscall write

* *void* disable\_page\_protection(*void*)

Tắt bảo vệ ghi trên cpu

* *void* enable\_page\_protection(*void*)

Bật lại bảo vệ ghi trên cpu

* *int* \_\_init hook\_init(*void*)

Khởi tạo hook

* *void* \_\_exit hook\_exit(*void*)

Hủy hook

* Sơ đồ biểu diễn sơ lược đồ án:

insmod rand

Đăng ký số hiệu

exit

Xóa số hiệu

Tạo class

Thiết lập thiết bị

Hủy class

Hủy thiết bị

Thêm thiết bị

Mở thiết bị

Đọc /dev/rand

Đọc thiết bị

Tạo số ngẫu nhiên

rmmod rand

Lệnh

copy vào user buffer

Xóa thiết bị ra khỏi hệ thống

Đóng thiết bị

Hủy thiết bị

Hủy class

Xóa số hiệu

exit

insmod hook

Tìm bảng sys\_call\_table

exit

Lưu syscall gốc

Tắt bảo vệ ghi

Thay thế syscall gốc

Bật lại bảo vệ ghi

hook

syscall

Rút trích các thông tin

rmmod hook

Tắt bảo vệ ghi

Trả về syscall gốc

Nếu syscall không bị hook

Khôi phục lại syscall

Bật lại bảo vệ ghi

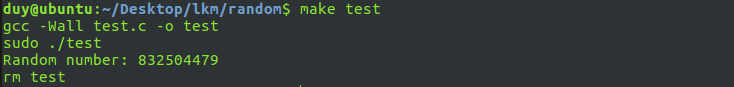
exit

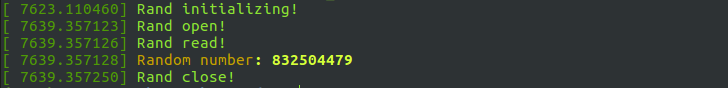
* Test case:

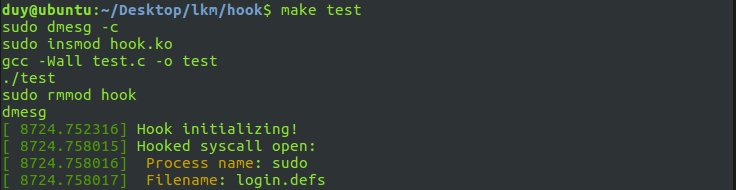
ubuntu: 16.04, linux kernel: 4.15.0-70

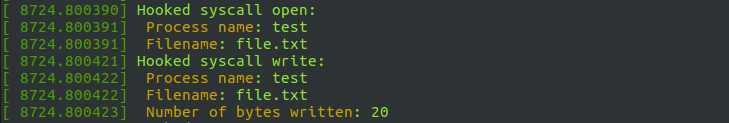
* Phần 1:

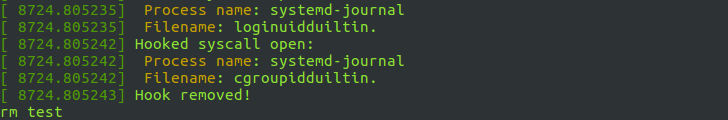






* Phần 2:





* Hướng dẫn sử dụng:

Linux kernel: 4.15.0-70 hoặc thấp hơn.

Cần chạy bằng quyền root(hoặc sudo nếu ubuntu…).

* Phần 1:

1. Chạy lệnh **make** để tạo module.
2. Chạy lệnh **modinfo rand.ko** để xem thông tin của module.
3. Chạy lệnh **dmesg -c** để xóa log cũ.
4. Chạy lệnh **insmod rand.ko** để thêm module vào kernel.
5. Chạy lệnh **lsmod rand | grep rand** để xem module được kích hoạt chưa.
6. Chạy lệnh **gcc -Wall test.c -o** để tạo file test.
7. Chạy lệnh **./test** để test.
8. Chạy lệnh **rmmod rand.ko** để gở bỏ module.
9. Chạy lệnh **dmesg** để xem log.
10. Chạy lệnh **make clean** để dọn dẹp các file được tạo ra.

* Phần 2:

1. Chạy lệnh **make** để tạo module.
2. Chạy lệnh **modinfo hook.ko** để xem thông tin của module.
3. Chạy lệnh **dmesg -c** để xóa log cũ.
4. Chạy lệnh **insmod rand.ko** để thêm module vào kernel.
5. Chạy lệnh **lsmod rand | grep rand** để xem module được kích hoạt chưa.
6. Chạy lệnh **gcc -Wall test.c -o** để tạo file test.
7. Chạy lệnh **./test** để test.
8. Chạy lệnh **rmmod rand.ko** để gở bỏ module.
9. Chạy lệnh **dmesg** để xem log.
10. Chạy lệnh **make clean** để dọn dẹp các file được tạo ra.

* Các nguồn tài liệu tham khảo:

<https://uwnthesis.wordpress.com/2016/12/26/basics-of-making-a-rootkit-from-syscall-to-hook/>

<https://github.com/le4ker/linux-kernel-security-suite>

<https://stackoverflow.com/>