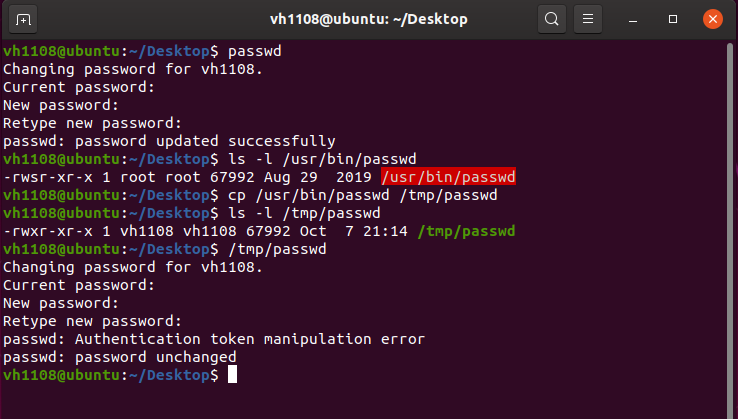
Lab 4

* Các lệnh “passwd”, “chsh”, “su” và “sudo” cần chương trình Set-UID vì như vậy có thể cho phép user tạm có quyền thực thi file bằng chính quyền của chủ sở hữu. Chương trình Set-UID sẽ cấp quyền tạm thời cho user đang đăng nhập chạy file quyền của user tạo ra file đó (owner), user chạy sẽ có UID và cả GID của người tạo ra file đó.
* Nếu không có chương trình Set-UID các user sẽ không được cấp quyền của owner để chạy mội file thực thi không phải mình tạo ra hay command một lệnh nào đó ngoài quyền của user.

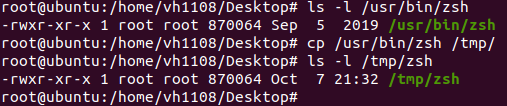


* Ban đầu có thể sử dụng lệnh “passwd” để thay đổi mật khẩu, kiểm tra thư mục /usr/bin/passwd thì thấy có Set-UID ( -rwsr-xr-x ).
* Copy thư mục /usr/bin/passwd vào /tmp/ và kiểm tra quyền thì không thấy set-UID, lúc này ta không thể thực thi /tmp/passwd

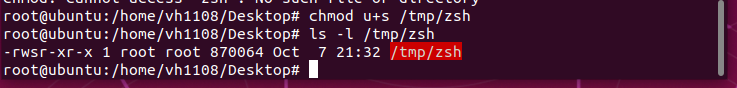


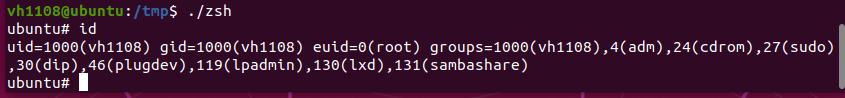
a/

* Đăng nhập vào root và copy /usr/bin/zsh vào /tmp/



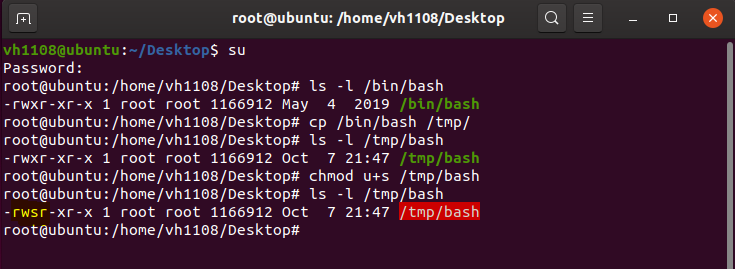
* Đặt /tmp/zsh thành chương trình set-root-UID với quyền 4755



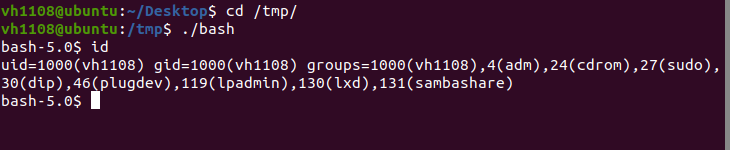
* Đăng nhập với một user bình thường và chạy /tmp/zsh
* 
* Lúc này, người dùng bình thường có được đặc quyền root

b/

* Đăng nhập vào root, sau đó sao chép /bin/bash vào /tmp/. Lúc này /tmp/bash chưa có set-UID, ta đặt chương trình set-UID cho nó và kiểm tra thì thấy đã có set-UID



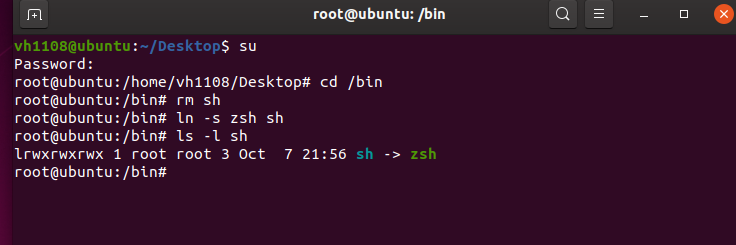
* Đăng nhập vào một người dùng bình thường và chạy /tmp/bash



* Lúc này, người dùng bình thường KHÔNG có đặc quyền root. Ta làm tương tự như với /bin/zsh nhưng với thư mục bin/bash thì không được đặc quyền root.

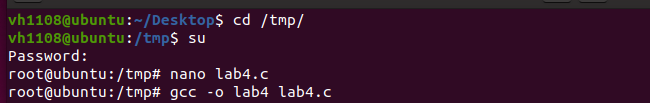
1. Sd

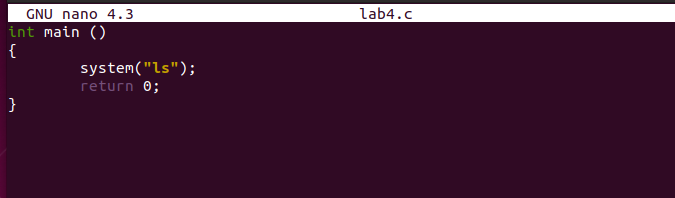
* Đăng nhập vào root, chuyển đến thư mục /bin sau đó xoá tập tin sh và tạo symbolic link giữa zsh và sh bằng lệnh “ln” với option “-s”



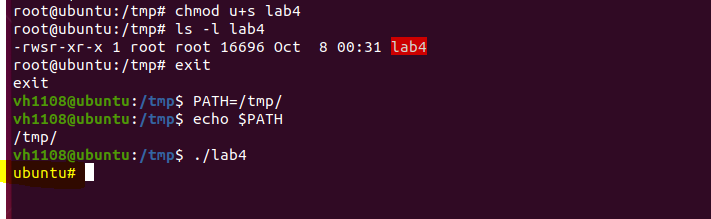
a/

* Đăng nhập với quyền root, soạn file lab4.c với nội dung như hình và tiến hành biên dịch bằng gcc





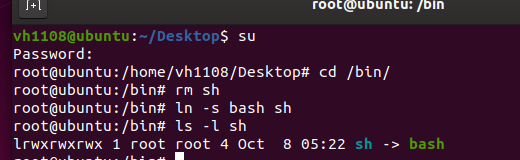
* Đặt set-UID cho file lab4 được dịch ở trên, kiểm tra lại quyền thì thấy đã thành công. Sau đó đăng nhập vào người dùng bình thường và cài đặt PATH ở /tmp/
* Cuối cùng ta tiến hành chạy file lab4 đã dịch ở trên



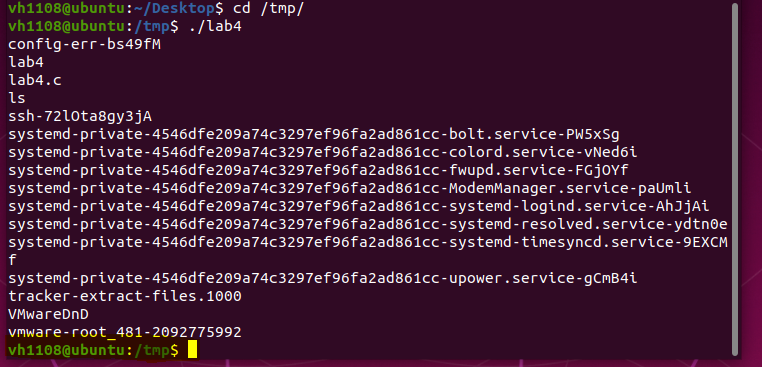
* Sau khi đã sửa đổi biếng môi trường, ta đã nhận được đặc quyền root

b/

* Đăng nhập vào root, tạo liên kết giữa bash và sh



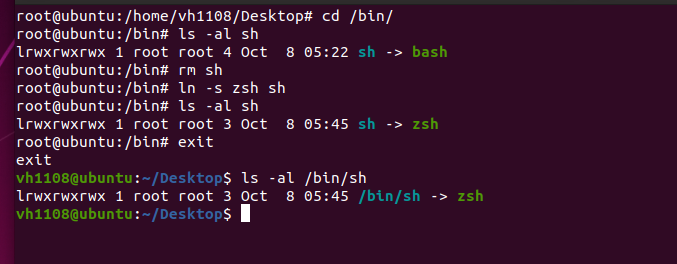
* Thực hiện lại cuộc tấn công như trên



* Việc tạo liên kết giữa bash và sh làm cho người dùng lúc này không có đặc quyền của root như thí nghiệm ở trên.

a/

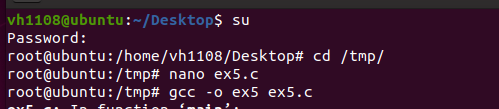
* Đăng nhập vào root, tạo liên kết giữa zsh và sh



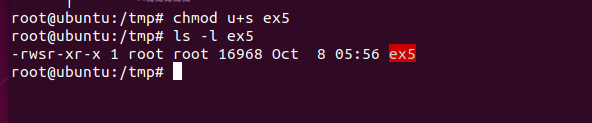
* Soạn file ex5.c với nội dung như hình



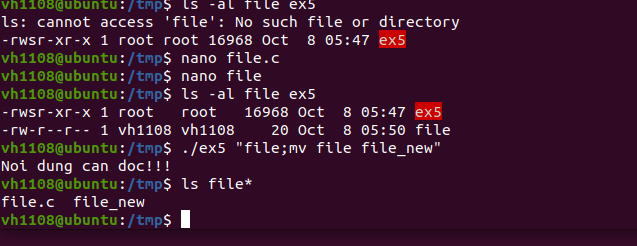
* Sử dụng gcc biên dịch file ex5.c thành ex5



* Đặt set-UID cho ex5



* Tạo tâp tin có tên file để thực hiện việc đọc. Thực thi ex5 với tâp tin file và đổi tên thành file\_new



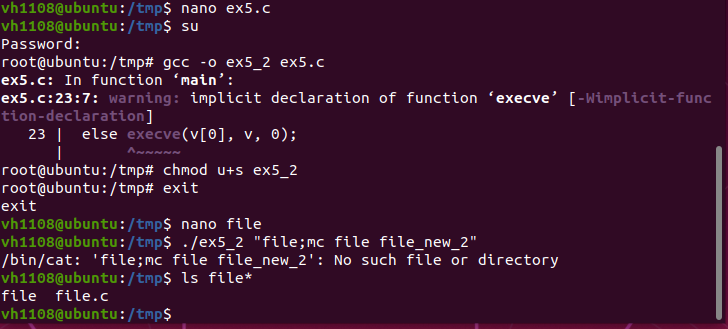
* Các lệnh này không an toàn. Bob có thể tiến hành đọc, ghi và di chuyển các tập tin mà chỉ có người dùng root có quyền.

b/

* Đặt q =1



* Biên dịch lại chương trình thành tập tin ex5\_2, sau đó set-UID cho ex5\_2
* Tạo 1 tập tin tên file để thực thi
* Tiến hành đọc tập tin file và đổi tên thành file\_new\_2



* Khi đặt q=1 thì cuộc tấn công không thành công. Vì khi q =0 thì hệ thống sẽ gọi system() ( bin/sh liên kết với zsh). Sau khi chạy “cat file” với quyền root thì nó tiếp tục chạy “mv file flie\_new”. Nhưng khi đặt q = 1 thì sẽ gọi execve() nó sẽ coi “file;mv file file\_new\_2” như một tên file và tiến hành gọi và kết quả trả về là không có file tên “file;mv file file\_new\_2”.