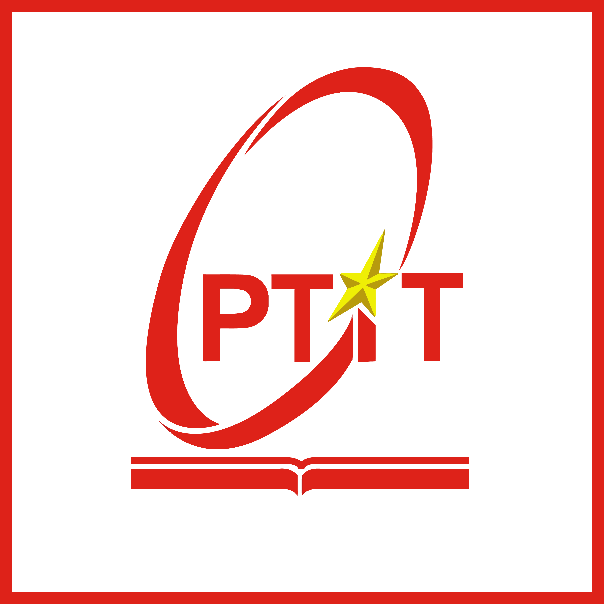
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA AN TOÀN THÔNG TIN**

**--------------**



**BÀI THỰC HÀNH 03**

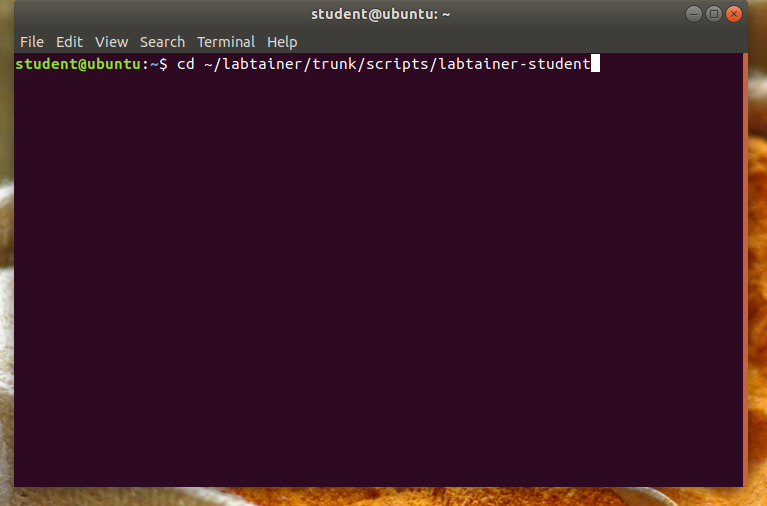
**MÔN HỌC: CƠ SỞ AN TOÀN THÔNG TIN**

|  |  |  |
| --- | --- | --- |
| **Giảng viên** | **:** | **Nguyễn Ngọc Điệp** |
| **Sinh viên** | **:** | **Lê Anh Tuấn** |
| **Lớp** | **:** | **D21CQAT01-B** |
| **Nhóm** | **:** | **03** |
| **Mã sinh viên** | **:** | **B21DCAT205** |
| **Số điện thoại** | **:** | **0369288612** |

Tháng 11/2023

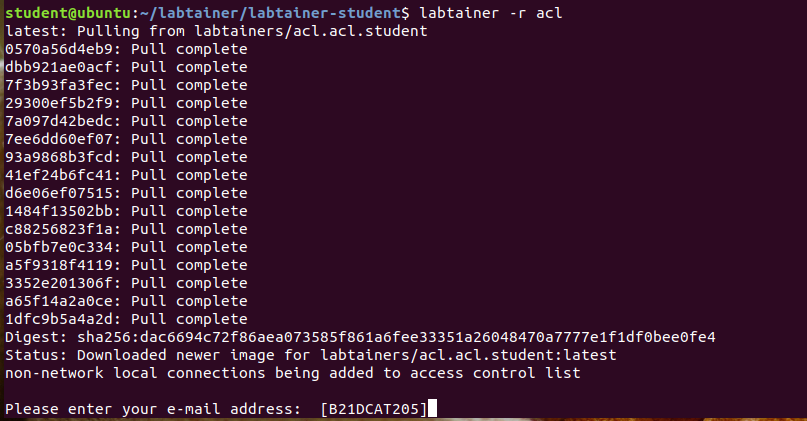
1. **Bài thực hành: Danh sách điều khiển truy cập trên Linux**

* Bật terminal và gõ dòng lệnh cd ~/labtainer/trunk/scripts/labtainer-student để chuyển vào thư mục labtainer-student

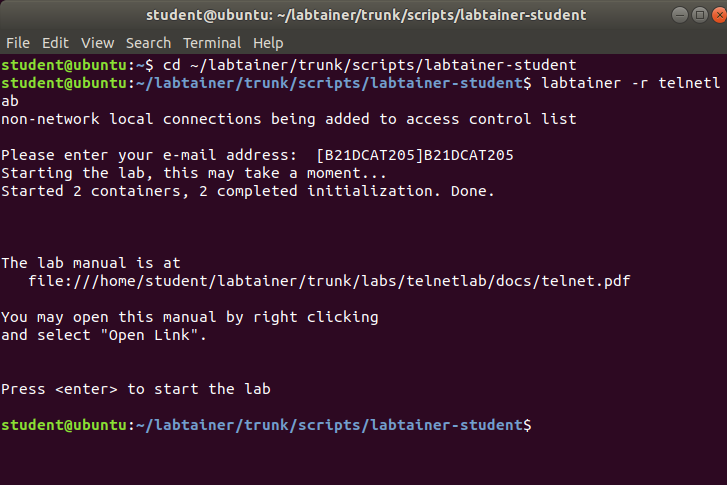


- Trong thư mục labtainer-student, ta tiếp tục gõ câu lệnh *labtainer -r acl*

- Tiếp tục ta được yêu cầu nhập email, ta nhập mã sinh viên

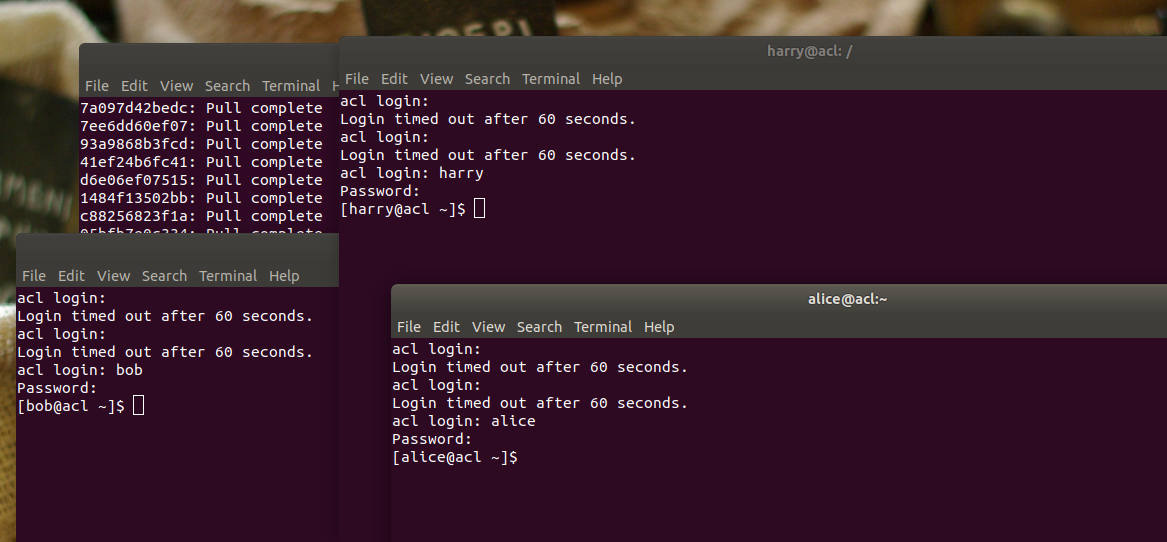


* Ấn enter



- Sau khi khởi động bài lab, 3 thiết bị đầu cuối ảo sẽ được bật chế động login, hãy đăng nhập theo các tài khoản dưới đây:

|  |  |
| --- | --- |
| **User** | **Password** |
| bob | password4bob |
| alice | password4alice |
| harry | password4harry |

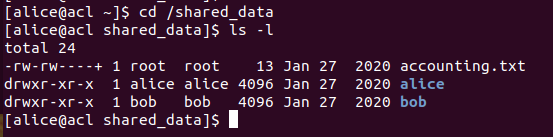


**Nhiệm vụ 1:** Xem lại các quyền trên các file hiện có

* Trên terminal “Alice”,đến thư mục /shared data và liệt kê các quyền trên file, thư mục:

cd /shared\_data

ls -l



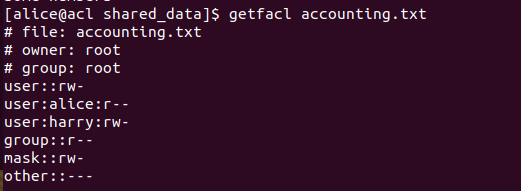
* Chúng ta sẽ thấy các quyền trên file accounting.txt và 2 thư mục. Kiểm tra xem “Alice” có thể xem nội dung file accounting.txt hay không ,ta sử dụng lệnh *cat* để đọc file.Ta sẽ thấy Alide có quyền xem nội dung file.

*cat accounting.txt*



* Ta kiểm tra quyền ACL của file này sử dụng lệnh:

 getfacl accounting.txt



* Ta thấy rằng harry có quyền thực thi nên ta chuyển tới terminal của harry và thực hiện câu lệnh sau để kiểm tra điều đó.

 echo "more stuff" >> /shared\_data/accounting.txt



* Quay trở lại terminal “alice”, thực hiện lệnh sửa đổi file ở trên để xác nhận rằng “alice” không có quyền sửa đổi file này.



**Nhiệm vụ 2:** Cài đặt ACL trên một file

* Với tư cách là người dùng Bob, sử dụng lệnh setfacl để cho phép Alice đọc file /shared\_data/bob/bobstuff.txt.

setfacl –m “u:alice:r” /shared\_data/bob/bobstuff.txt



* Xác nhận khả năng đọc file của alice sử dụng câu lệnh

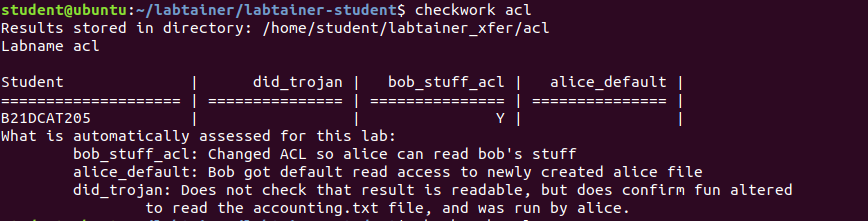
*cat bob/bobstuff.txt*



* Harry không có khả năng đọc file này

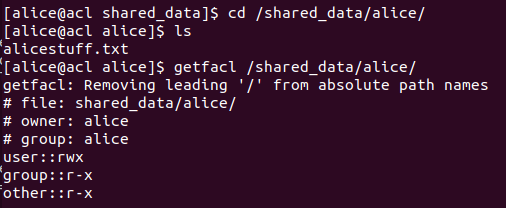


* Checkwork sau khi hoàn thành nhiệm vụ 1 và nhiệm vụ 2.



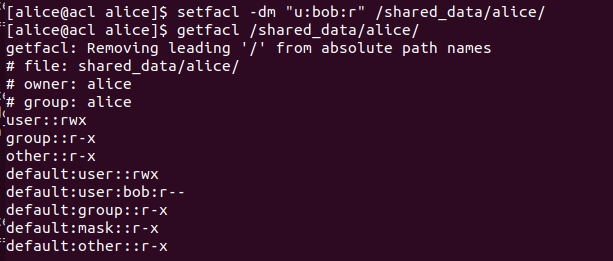
**Nhiệm vụ 3:** Cài đặt ACL mặc định cho 1 thư mục

* Ta nhận thấy bob không có một quyền nào trong đây



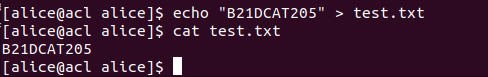
* Ta cấp quyền cho bob có quyền đọc sử dụng câu lệnh

setfacl –dm “u:bob:r” /shared\_data/alice/



* Ta tạo 1 file mới và alice có quyền đọc file đó sử dụng câu lệnh

*echo “B21DCAT205” > test.txt*



* Bob cũng có quyền đọc file



* Vì *default:other::r-x* nên harry cũng có quyền đọc file này

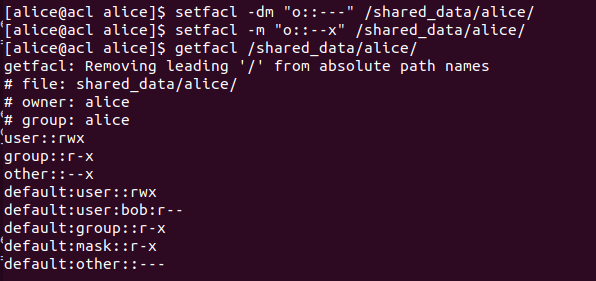


* Để xóa quyền của harry ta sử dụng câu lệnh

setfacl –dm “u:bob:r” /shared\_data/alice/

* Sau đó sử dụng câu lệnh để thiết lập other có quyền được thực thi(x).

setfacl –m “o::--x” /shared\_data/alice/

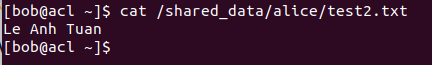


* Kiểm tra lại bằng cách tạo một file mới test2.txt.

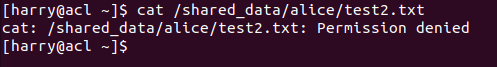
*echo “Le Anh Tuan” > test2.txt*



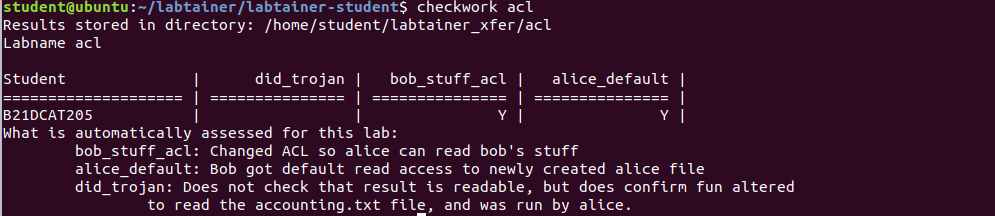
* Ta thấy bob có quyền đọc file này.



* Nhưng Harry không có quyền đọc file này.

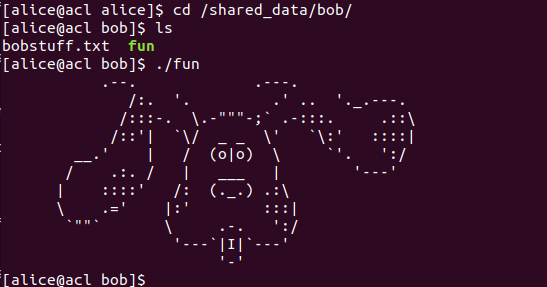


* Checkwork sau khi hoàn thành nhiệm vụ 3



**Nhiệm vụ 4:** Trojan Horses

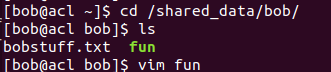
* Đọc thử file fun của bob ở máy alice



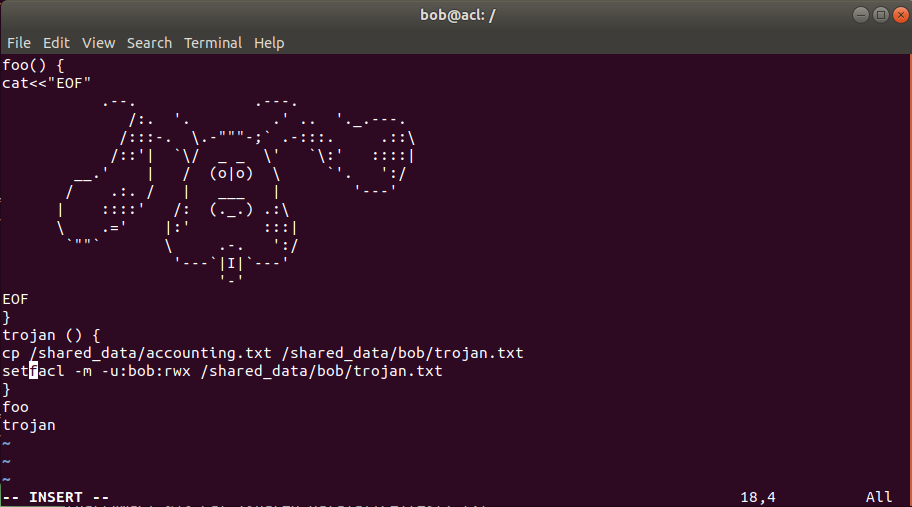
* Kiểm tra lại Bob không có quyền đọc file accounting.txt



* Sử dụng câu lệnh *vim fun* để tiến hành sửa file script

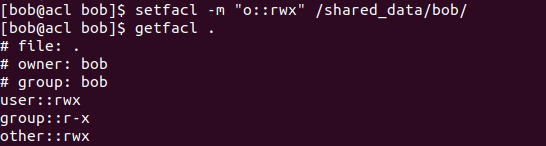


* Thêm vào một hàm có tên là trojan, sau đó định nghĩa nó như hình dưới với việc khi mà alice mở file thì nội dung của accounting.txt sẽ được copy sang tệp có tên trojan.txt và cấp quyền bob có thể đọc được tệp đó



* Cấp quyền cho alice được tạo file trong thư mục bob

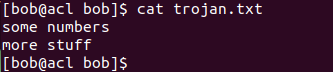
*setfacl -m “0::rwx” /shared\_data/bob/*



* Thực hiện đọc file từ alice.



* Bob có thể đọc file trojan.txt chính là nội dung được sao chép từ accounting.txt

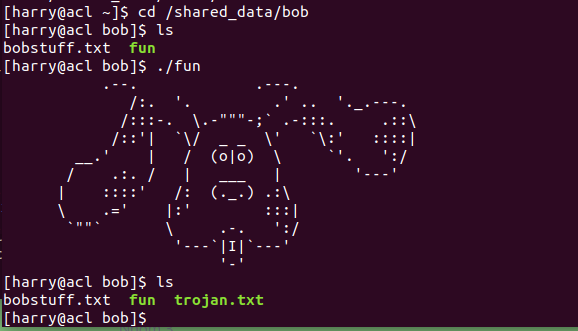


* Xóa đi file để tiến hành thử nghiệm với harry vì harry cũng có quyền đọc file accounting.txt

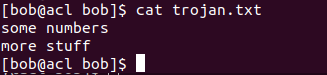
*rm trojan.txt*



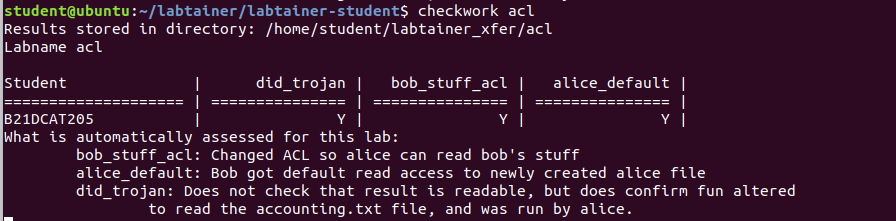
* Thử lại với harry.



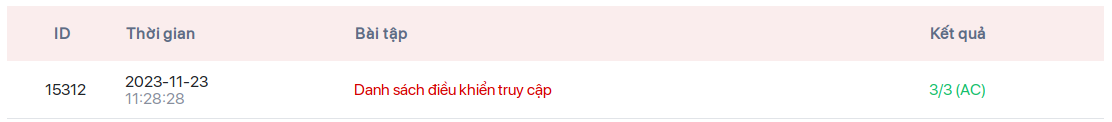
* Vì harry có quyền đọc file accounting nên bob có thể đọc được file trojan.txt tương tự.



* Checkwork sau khi hoàn thành nhiệm vụ 4



**Kiểm tra kết quả**



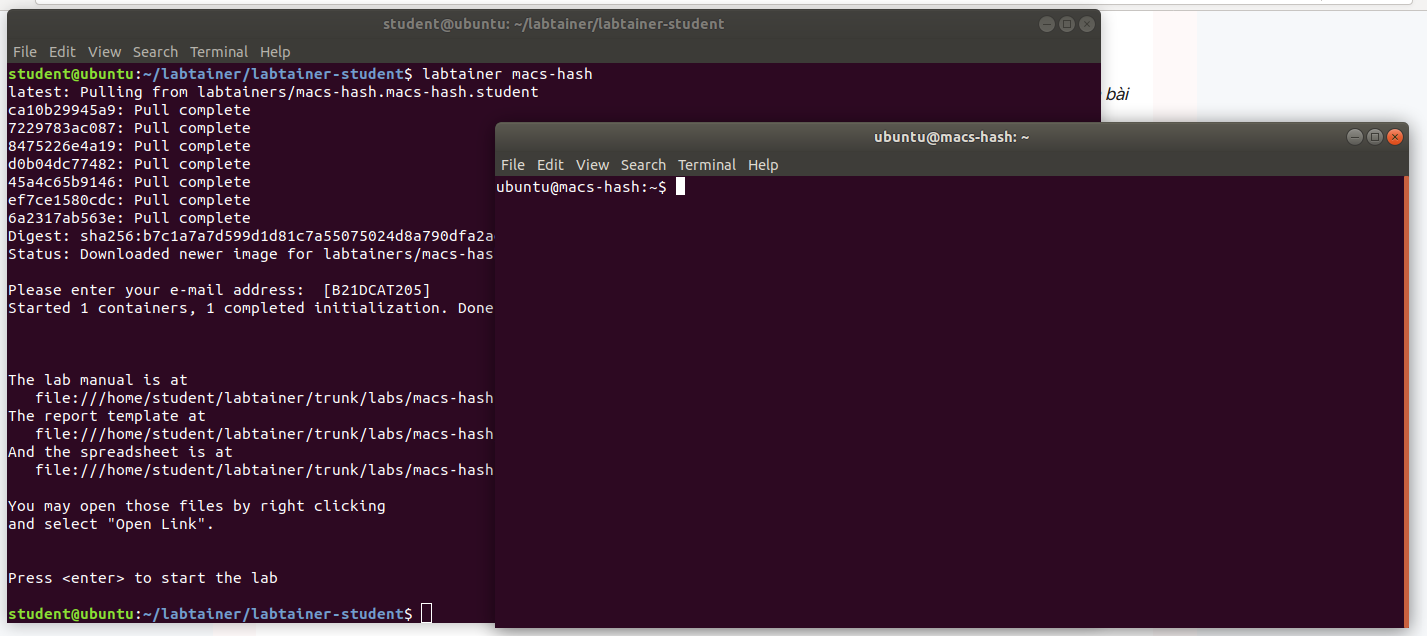
**Kết luận**

* Ta đã biết cách xem các quyền ACL trên file và thư mục
* Cài đặt thành công ACL trên một file
* Cài đặt mặc định ACL trên thư mục
* Biết cách sử dụng ACL để cấp quyền,xóa bỏ quyền cho file, thư mục
* Cần lưu ý một số tình huống đọc file lạ có thể sẽ bị rò rỉ thông tin.

1. **Bài thực hành: Tìm hiểu về hàm băm và mã xác thực thông điệp MACs**

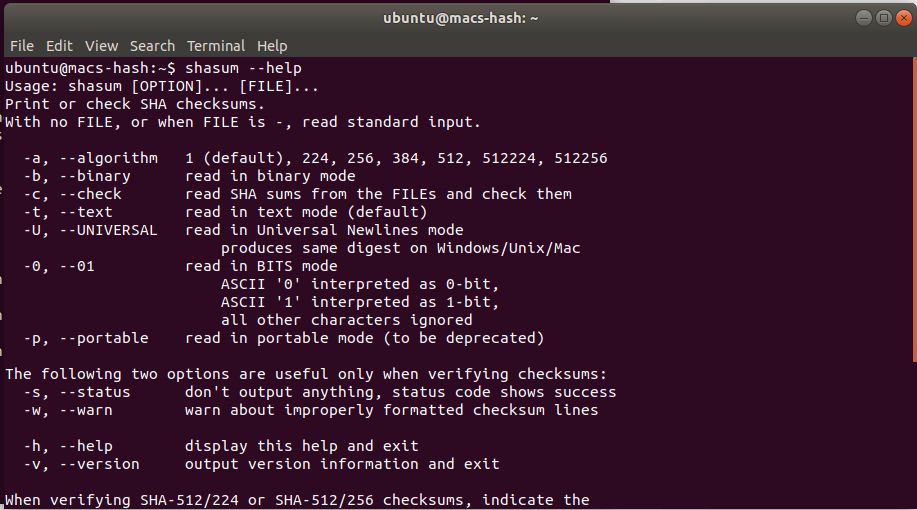
**Khởi động bài lab:**

labtainer -r macs-hash



**Nhiệm vụ 1:** Trong nhiệm vụ này, ta sẽ không sử dụng OpenSSL để tạo bản tóm lược (digest) vì có nhiều cách tạo dễ dàng hơn trên Unix. Thay vào đó, ta sử dụng lệnh Shasum mặc định hỗ trợ SHA-1 với đầu ra 160 bit, cũng như các đầu ra SHA-2 dưới các tùy chọn khác. Ta có thể xem các tùy chọn bằng lệnh dưới:

shasum --help | less

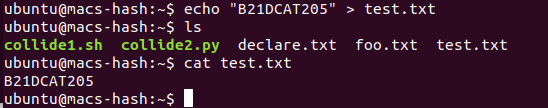


* Tạo ra một bản tóm lược 160 bit SHA-1 với tên foo.txt

shasum –a 1 foo.txt



* Tạo một tệp văn bản test.txt trên hệ thống với nội dung bất kỳ sử dụng câu lệnh *echo “B21DCAT205” > test.txt*



* Sử dụng Shasum để thử 7 thuật toán mã hóa với tệp đã tạo ở trên lần lượt với các câu lệnh:

*shasum -a 1 test.txt*

*shasum -a 224 test.txt*

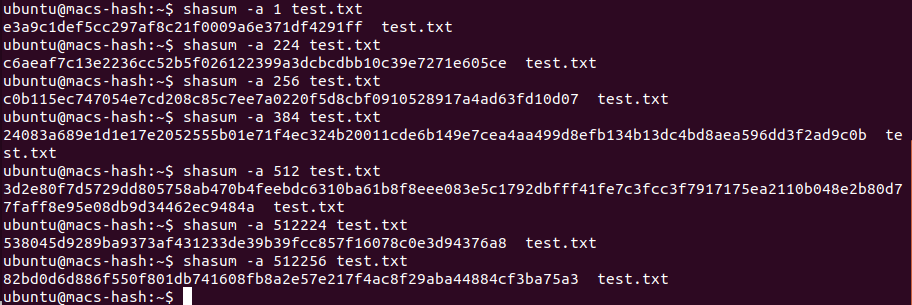
*shasum -a 256 test.txt*

*shasum -a 384 test.txt*

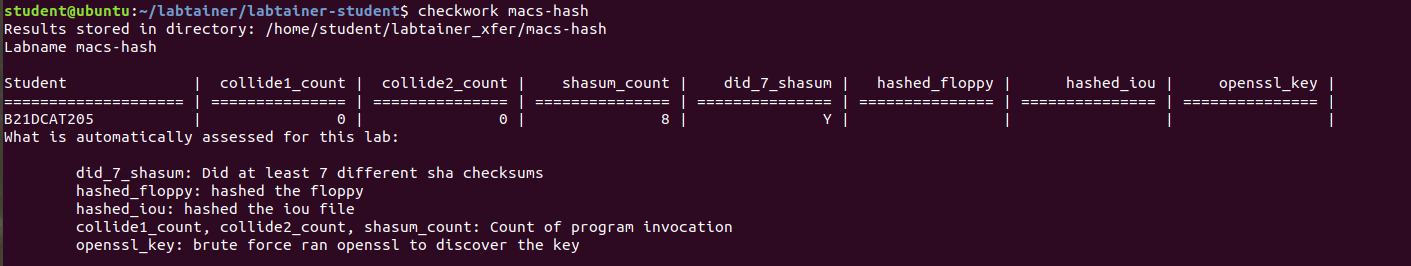
*shasum -a 512 test.txt*

*shasum -a 512224 test.txt*

*shasum -a 512256 test.txt*



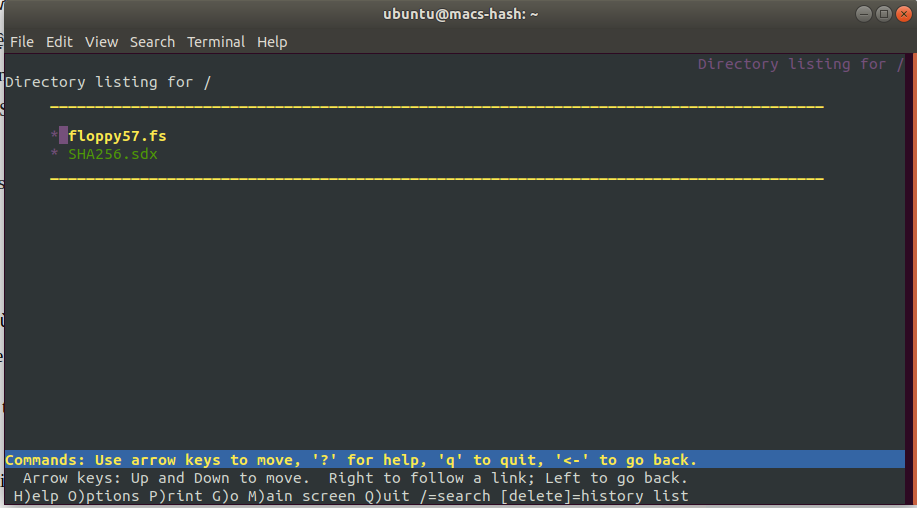
* Checkwork sau khi hoàn thành nhiệm vụ 1



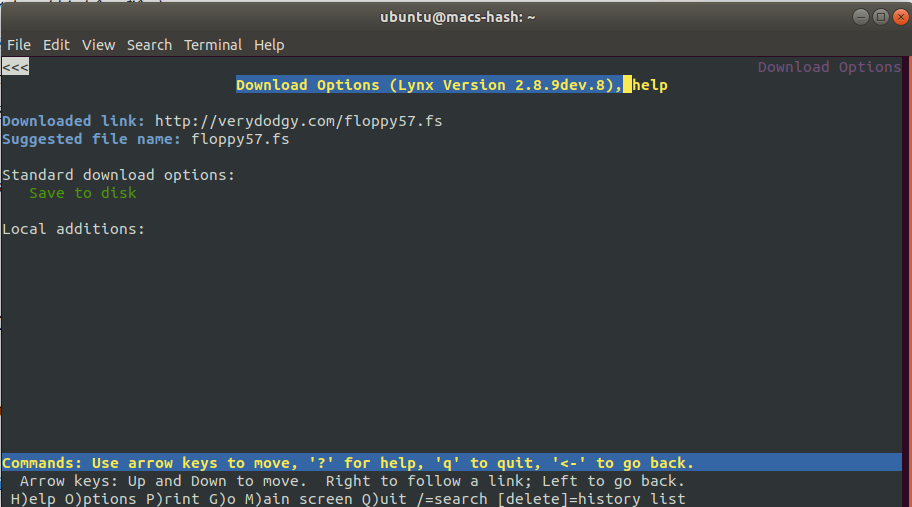
**Nhiệm vụ 2:** Kiểm tra bản tóm lược

Trong nhiệm vụ này, chúng ta sẽ học cách kiểm tra tính toàn vẹn của file tải xuống bằng cách sử dụng hàm băm. Sử dụng trình duyệt lynx để tải xuống 2 file.

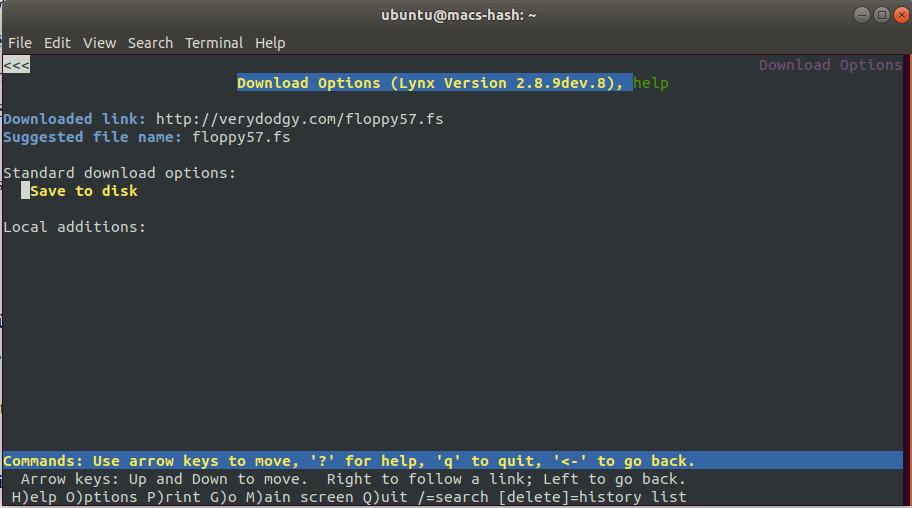
* Gõ *"lynx verydodgy.com"*
* Chọn enter + d để tải file



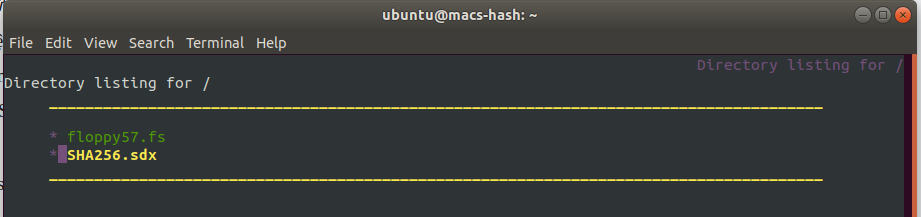
* Tiếp tục ấn mũi tên xuống, sau đó ấn enter 2 lần

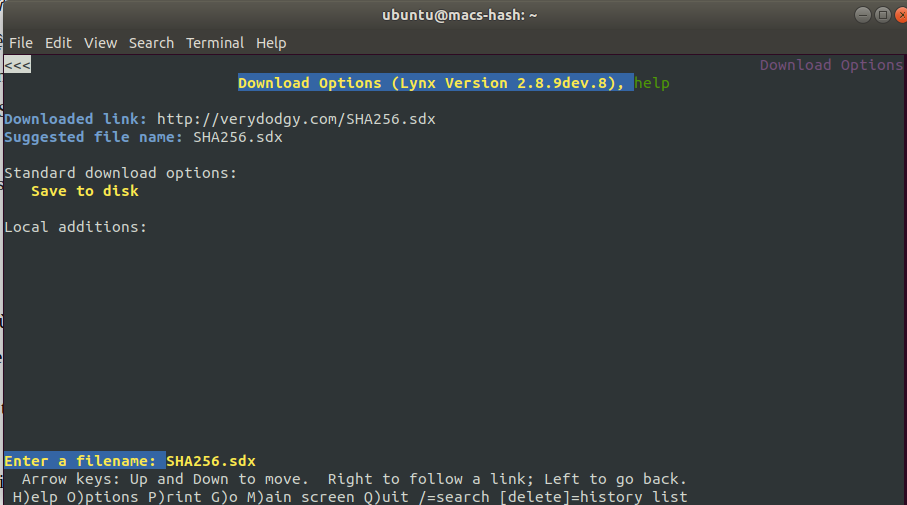


* Sau đó ấn q và mũi tên quay lại để trở về.



* Tiếp tục ấn mũi tên xuống và thực hiện tương tự





* Kiểm tra các file vừa cài đặt sử dụng câu lệnh *ls*



* Tạo bản tóm lược SHA256 với file vừa tải xuống. Kiểm tra bản tóm lược tạo ra và file *SHA256.sdx* ta thấy khớp nhau.

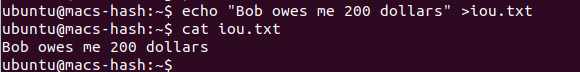




**Nhiệm vụ 3:** Tìm hiểu về “Avalanche Eﬀect”

“Avalanche Eﬀect” mô tả hiện tượng một thay đổi nhỏ ở file đầu vào cũng sẽ làm thay đổi hoàn toàn đến bản tóm lược đầu ra.

* Tạo một file với tên *iou.txt* có nội dung*“Bob owes me 200 dollars”.*



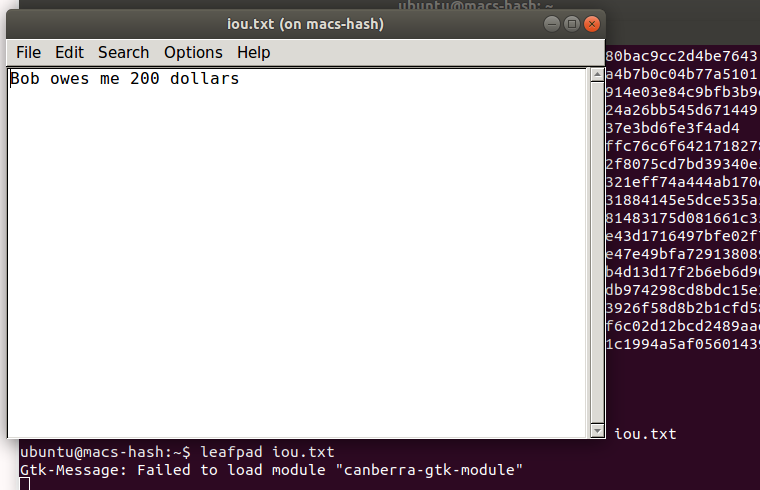
* Tạo bản tóm lược SHA256 của file *iou.txt*

*shasum -a 256 iou.txt*

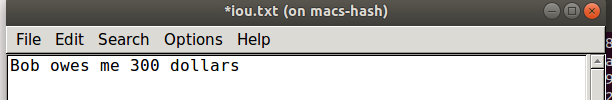


* Mở file *iou.txt* bằng chương trình chỉnh sửa, ví dụ leafpad

*leafpad iou.txt*



* Thay đổi số “2” thành “3”. Điều này sẽ làm 1 bit bị thay đổi và lưu các thay đổi và thoát khỏi chương trình chỉnh sửa.



* Kiểm tra lại file vừa sửa



* Tạo bản tóm lược SHA256 khác đối với tệp*iou.txt* đã sửa đổi.

*shasum -a 256 iou.txt*

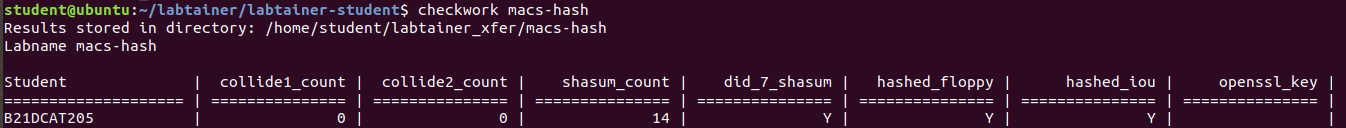


* Tương tự đối với file floppy



Ta thấy rằng khi chỉ thay đổi 1 ký tự, bản tóm lược sẽ thay đổi hoàn toàn, để bản tóm lược trùng khớp với bản gốc thì ta chỉ cần thay đổi đoạn văn bản trở lại như ban đầu sẽ khôi phục được bản tóm lược.

* Checkwork sau khi hoàn thành nhiệm vụ 3



**Nhiệm vụ 4:** Tìm hiểu về Second Pre-Image Resistance

Trong phần này, chúng ta sẽ tìm hiểu các thuộc tính Second Pre-Image Resistant của hàm băm SHA256.

* Đầu tiên, tạo bản tóm lược SHA256  của file *declare.txt*
* Bốn chữ số thập lục phân cuối cùng của bản tóm lược được hiển thị là “7bc8”



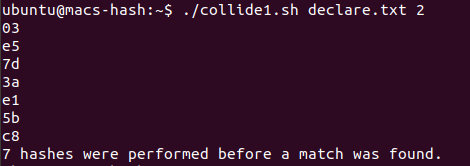
* Thực hiện lệnh sau để tìm dữ liệu ngẫu nhiên mà khi băm, 6 số cuối dạng hex của bản tóm lược sẽ trùng với 6 số cuối dạng hex của bản tóm lược file declare.txt

./collide1.sh declare.txt 1



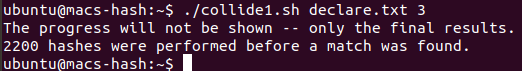
* Bây giờ, hãy thực hiện lệnh sau để so khớp hai chữ số hex cuối cùng của bản tóm lược file *declare.txt.*

./collide1.sh declare.txt 2



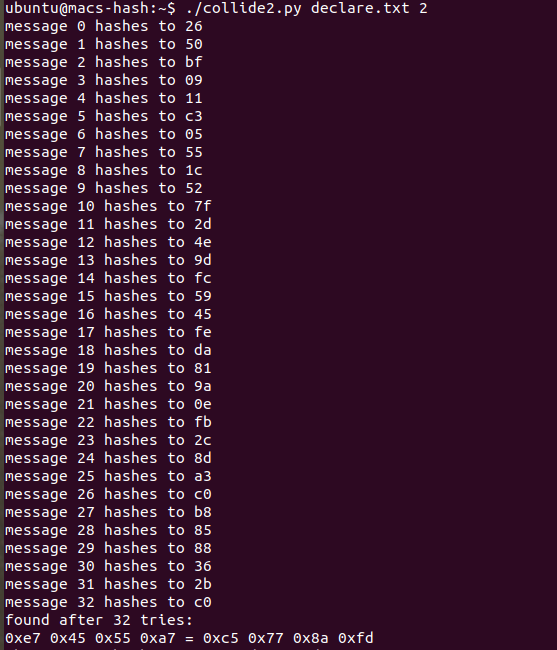
* Thực hiện lệnh sau để so khớp 3 chữ số hex cuối cùng của bản tóm lược file *declare.txt.*

./collide1.sh declare.txt 3



* collide2.py để giải quyết vấn đề xung đột. Nó tạo dữ liệu ngẫu nhiên và băm nó, sau đó lưu mỗi digest trong một bảng. Nó tiếp tục làm điều này cho đến khi nó tìm thấy hai xâu dữ liệu ngẫu nhiên có digest khớp nhau ở byte cuối cùng (tức là hai chữ số hex cuối cùng).



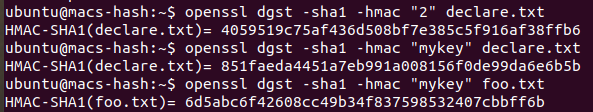
 

* Để tạo một HMAC dựa trên SHA1 bằng OpenSSL, bạn có thể nhập lệnh sau:

*openssl dgst -sha1 -hmac KEY FILENAME*

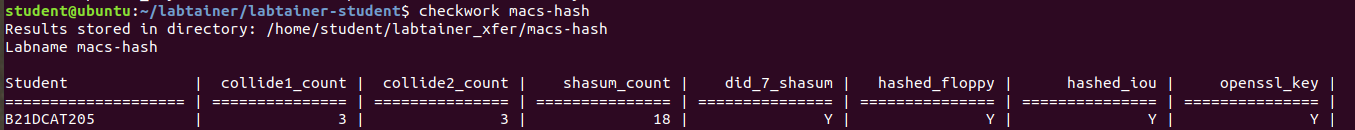
*Với*

* KEY bằng bất kỳ chuỗi nào bạn chọn, miễn là nó không có khoảng trắng.
* FILENAME bằng tên của một tệp trên hệ thống của bạn.





* Checkwork sau khi hoàn thành nhiệm vụ 4



**Kết luận**

* Các nhiệm vụ đã giúp chúng ta hiểu về quá trình tạo bản tóm lược, kiểm tra tính toàn vẹn của file tải xuống, hiệu ứng "Avalanche", và thuộc tính "Second Pre-Image Resistance" của hàm băm SHA256.
* Qua đó, chúng ta có cái nhìn sâu sắc hơn về tính an toàn và tính toàn vẹn của dữ liệu khi sử dụng các thuật toán băm trong bảo mật thông tin.

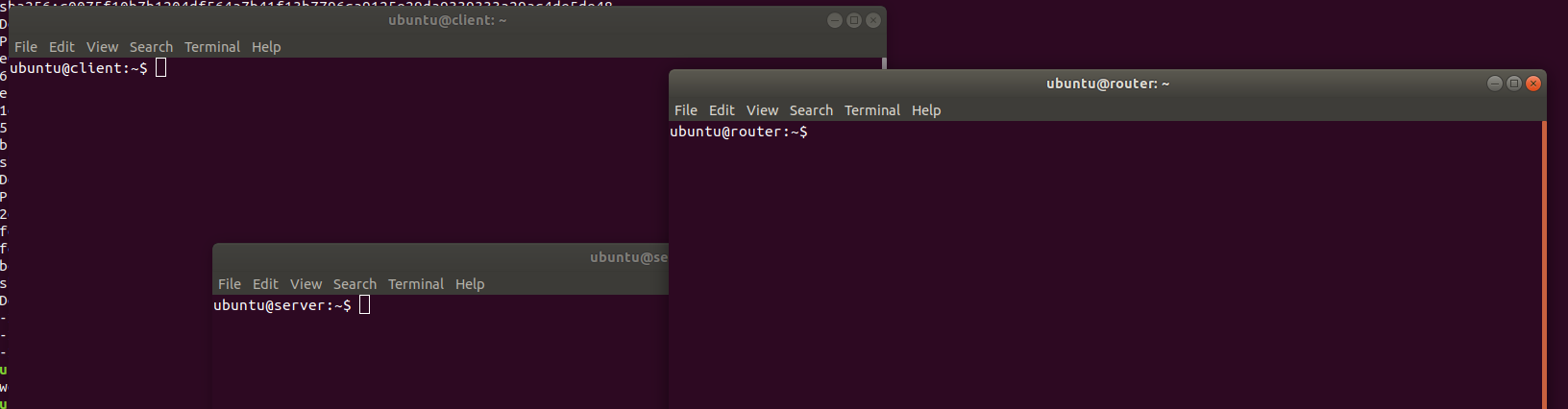
1. **Bài thực hành: VPN host-to-host**

**Khởi động bài lab**:

* Trên terminal, gõ lệnh:

*labtainer -r vpnlab*

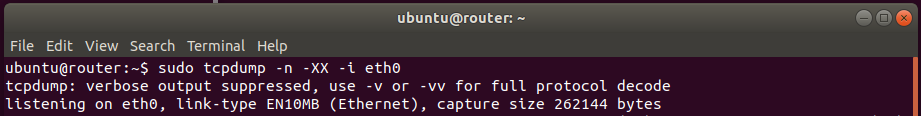
* Sau khi khởi động xong 3 terminal ảo sẽ xuất hiện: trong đó 1 máy ảo đóng vai trò client, 1 máy đóng vai trò server và 1 máy đóng vai trò router.



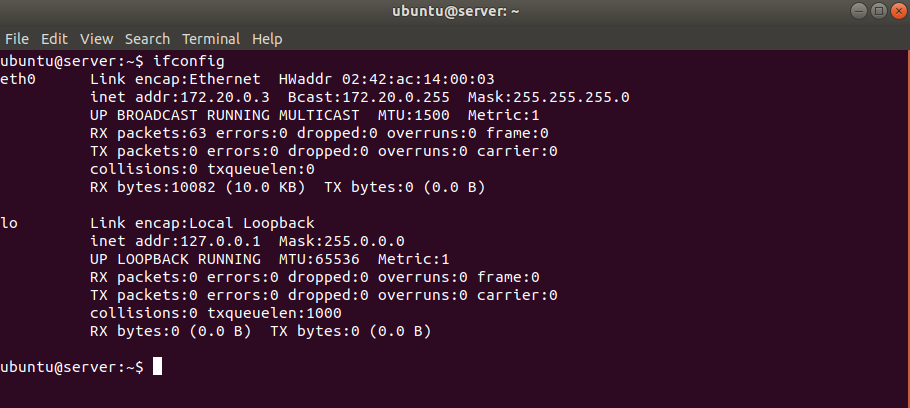
**Các nhiệm vụ cần phải thực hiện:**

* Khởi động tcpdump trên “router” để bắt các gói tin từ interface eth0.

sudo tcpdump -n -XX -i eth0

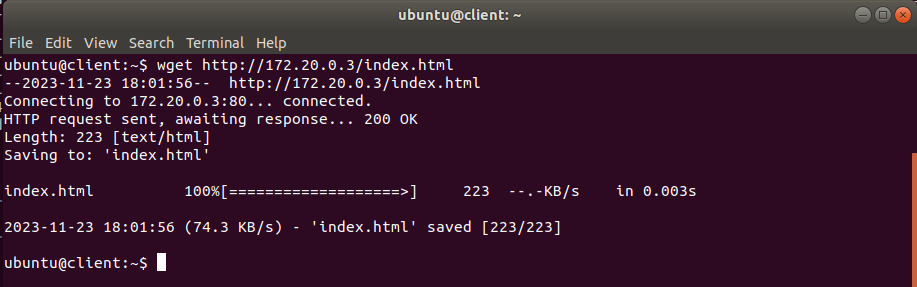


* Trong server sử dụng ifconfig để xem địa chỉ ip, ta thấy địa chỉ ip của server là 172.20.0.3

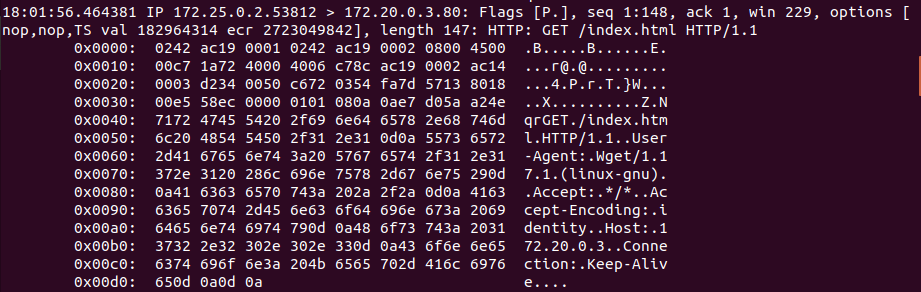


* Sử dụng wget trên máy khách để tìm nạp tệp index.html.

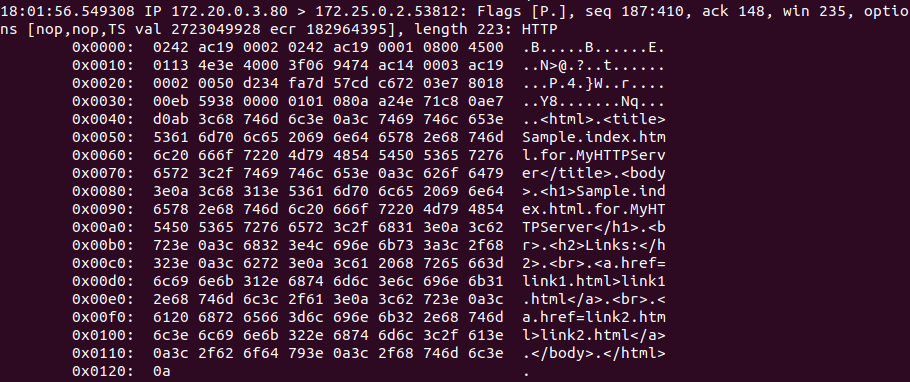
wget [*http://172.20.0.3/index.html*](http://172.20.0.3/index.html)



* Quan sát lưu lượng mạng từ tcpdump.



* Xem html văn bản thuần túy trong luồng dữ liệu.



* Khởi động chương trình openvpn trên máy chủ:

sudo openvpn --config server.conf --daemon

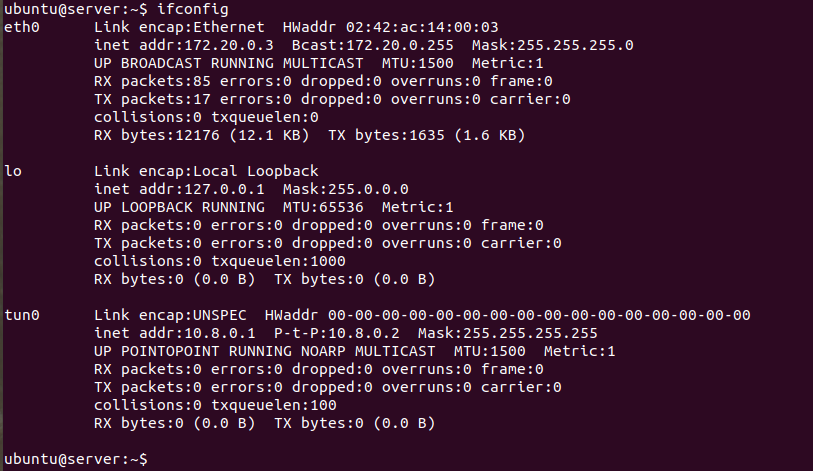


* Sau đó khởi động chương trình openvpn trên máy khách:

*sudo openvpn --config client.conf –daemon*

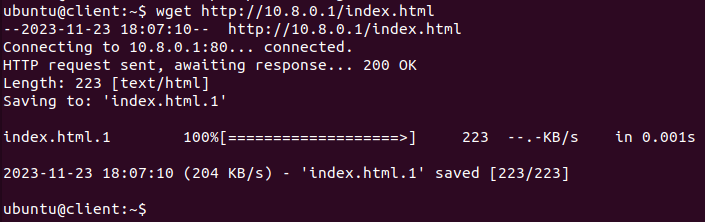


* Sử dụng lại ifconfig từ máy server để kiểm tra rằng có một “đường hầm” giúp đảm bảo an toàn khi truyền dữ liệu là tun0 với địa chỉ ip là 10.8.0.1

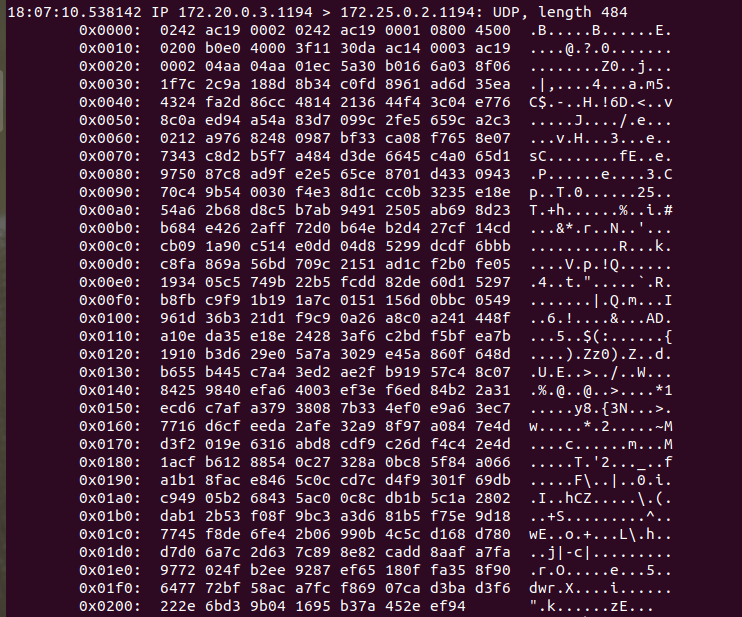


* Sử dụng lại wget, nhưng lần này sử dụng địa chỉ đường hầm của máy chủ(địa chỉ hiển thị ở interface ”tun0” khi chạy lệnh ifconfig trên “server”).

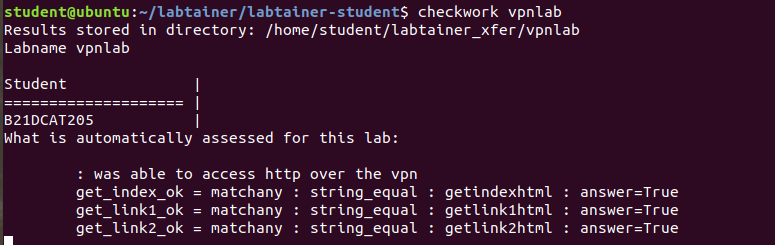
wget <http://10.8.0.1/index.html>



* Quan sát lưu lượng mạng qua tcpdump, ta thấy tất cả thông tin đều được mã hóa và không thể đọc được.



* Checkwork kiểm tra bài làm



**Kết luận**

* + Qua các nhiệm vụ, chúng ta đã thực hành việc bắt gói tin mạng, quan sát nội dung của lưu lượng mạng, triển khai và sử dụng OpenVPN để tạo một kênh kết nối bảo mật.
  + Việc này giúp chúng ta hiểu rõ hơn về cách thức hoạt động của các công cụ mạng cũng như việc thiết lập và sử dụng VPN để bảo vệ dữ liệu truyền qua mạng.