HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

BÁO CÁO BÀI THỰC HÀNH
HỌC PHẦN: KỸ THUẬT GIẤU TIN
MÃ HỌC PHẦN: INT14102

NHÓM LỚP: D21CQAT03-B

TÊN BÀI: GIẤU VÀ TÁCH TIN TRONG TỆP ÂM THANH BẰNG KỸ THUẬT FHSS

Sinh viên thực hiện:

B21DCAT135 Đặng Quý Nam

Giảng viên: PGS.TS. Đỗ Xuân Chợ

HỌC KỲ 2 NĂM HỌC 2024-2025

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG BIỂU

CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH

1.1 Giới thiệu chung về bài thực hành

Bài thực hành này tập trung vào kỹ thuật tách tin giấu trong âm thanh bằng cách sử dụng thuật toán nhảy tần – FHSS (Frequency Hopping Spread Spectrum).

Giấu tin trong file âm thanh bằng FHSS (Frequency Hopping Spread Spectrum) là một kỹ thuật mã hóa tín hiệu, nơi thông tin được giấu vào tần số của tín hiệu âm thanh. Cách thực hiện và tách thông tin này có thể được mô tả ngắn gọn như sau:

Cách giấu thông tin:

1. **Chia nhỏ thông tin cần giấu**: Thông tin (ví dụ: văn bản, hình ảnh) sẽ được mã hóa thành các bit.
2. **Tần số nhảy**: Sử dụng FHSS, thông tin được "nhảy" qua các tần số khác nhau trong tín hiệu âm thanh theo một dãy tần số ngẫu nhiên. Dãy tần số này được chọn sao cho nó chỉ có thể được giải mã bởi người nhận có thông tin về cách thức nhảy tần số.
3. **Gửi tín hiệu âm thanh**: Tín hiệu âm thanh chứa thông tin được ẩn giấu thông qua các tần số nhảy.

Cách tách thông tin:

1. **Xác định chu kỳ nhảy tần số**: Người nhận phải biết chính xác cách thức nhảy tần số (giống như người gửi).
2. **Lọc tần số**: Lọc và tái tạo tín hiệu âm thanh bằng cách sử dụng tần số nhảy đã biết, sau đó giải mã thông tin giấu bên trong.
3. **Giải mã**: Sau khi tách ra các phần tín hiệu, thông tin được giải mã từ các bit đã ẩn.

Kỹ thuật này mang lại sự bảo mật cao vì tần số nhảy thay đổi liên tục, khiến việc nhận diện và giải mã trở nên khó khăn nếu không biết chính xác cách thức nhảy tần số.

1.2 Nội dung và hướng dẫn bài thực hành

1.2.1 Mục đích

Giúp sinh viên nắm bắt cách thức tách tin bằng FHSS và cách sử dụng SCP để gửi tệp.

1.2.2 Yêu cầu đối với sinh viên

- Hiểu rõ khái niệm giấu tin và vai trò của giấu tin trong âm thanh.
- Nắm được nguyên lý hoạt động của thuật toán giấu tin Frequency Hopping Spread Spectrum (FHSS).

1.2.3 Nội dung thực hành

Khởi động bài lab, tải bài thực hành bằng imodule:

Imodule imodule.tar

Hoặc giải nén file imodule.tar vào foder trunk/labs
Vào terminal, gõ:

Labtainer -r stego-code-fhss-random-freq

(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)

Sau khi khởi động xong, màn hình sẽ xuất hiện 2 terminal để làm bài thực hành, một terminal đại diện cho người dùng bob (người gửi giấu tin), một terminal đại diện cho người dùng alice(người nhận tách tin)

Nhiệm vụ cần phải làm:

- Kiểm tra địa chỉ IP ở 2 máy.
- Ở máy bob, sinh viên tiến hành ghi mã sinh viên của mình vào file message.txt để làm thông điệp giấu tin.
- Sử dụng file hide.py để tiến hành giấu tin: ***python3 hide.py.***
- Trong file hide.py, chương trình sẽ chọn ngẫu nhiên một trong các khoảng tần số: [3000, 11000]; [4000, 12000]; [5000, 13000]; [6000, 14000]; [7000, 15000]; [8000, 16000] để giấu thông điệp. Nhiệm vụ của sinh viên khi nhận được file âm thanh đã giấu phải tìm được khoảng tần số đó để tách tin.
- Ở máy alice, sinh viên cài đặt cấu hình SSH như sau:
    - Ở file /etc/ssh/sshd_config, đảm bảo cấu hình mở cổng 22, cho phép truy cập bằng mật khẩu vào đăng nhập bằng root:
        - Port 22
        - PasswordAuthentication yes
        - PermitRootLogin yes
    - Khởi động lại SSH với cấu hình trên: ***sudo systemctl restart ssh.***

- Ở máy bob, gửi file âm thanh bằng lệnh ***scp stego_audio.wav ubuntu@172.33.0.4:/home/ubuntu*** với password là 1.
- Ở máy alice, sinh viên đọc file âm thanh nhận được bằng audacity.
- Audacity sẽ hiển thị file âm thanh dưới dạng biểu đồ ***waveform***, nhấn vào dấu 3 chấm ở mỗi file, chuyển biểu đồ sang dạng ***spectrogram.***
- Ở phần đầu của file âm thanh, có những đốm sáng là phần giấu tin. Kiểm tra plot spectrum trong Analyze để xem những đốm sáng này nằm ở vùng tần số nào.
- Sau khi biết được vùng tần số, chỉnh sửa phần freq_start trong file ***extract.py*** thành điểm bắt đầu của khoảng tần số.

Lưu ý: khoảng tần số này là một trong các khoảng tần số đã cho.

- Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

  stoplab

  Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

  Khởi động lại bài lab:

  Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

  labtainer –r stego-code-fhss-random-freq

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU BÀI THỰC HÀNH

2.1 Thiết kế bài thực hành

Bài lab gồm 2 container là bob và alice.

Cấu hình Docker

- Bài lab chạy với image là .network2
- Cần cài thêm thư viện numpy, scipy trong dockerfiles của 2 container
- Docs lưu lại hướng dẫn thực hành cho sinh viên

Các nhiệm vụ cần thực hiện để thành công:

- Khởi động ssh.
- Giấu tin vào file âm thanh.
- Gửi file đến cho alice.
- Sử dụng audacity để xác định khoảng tần số.
- Tách tin từ file âm thanh.
- Kết thúc bài lab và đóng gói kết quả

Để đánh giá được sinh viên đã hoàn thành bài thực hành hay chưa, cần chia bài thực hành thành các nhiệm vụ nhỏ, mỗi nhiệm vụ cần phải chỉ rõ kết quả để có thể dựa vào đó đánh giá, chấm điểm. Do vậy, trong bài thực hành này hệ thống cần ghi nhận các thao tác, sự kiện được mô tả và cấu hình như bảng dưới đây:

```
start_ssh = alice:.bash_history : CONTAINS : systemctl restart ssh
hide_message = bob:.bash_history : CONTAINS : python3 hide.py
send_file = bob:.bash_history : CONTAINS : ubuntu@172.33.0.4:/home/ubuntu
open_audacity = alice:.bash_history : CONTAINS : audacity
extracted_message = alice:*.stdout : CONTAINS : DCAT
```

Hình 1 File result.config

start_ssh : Khởi động ssh ở máy alice

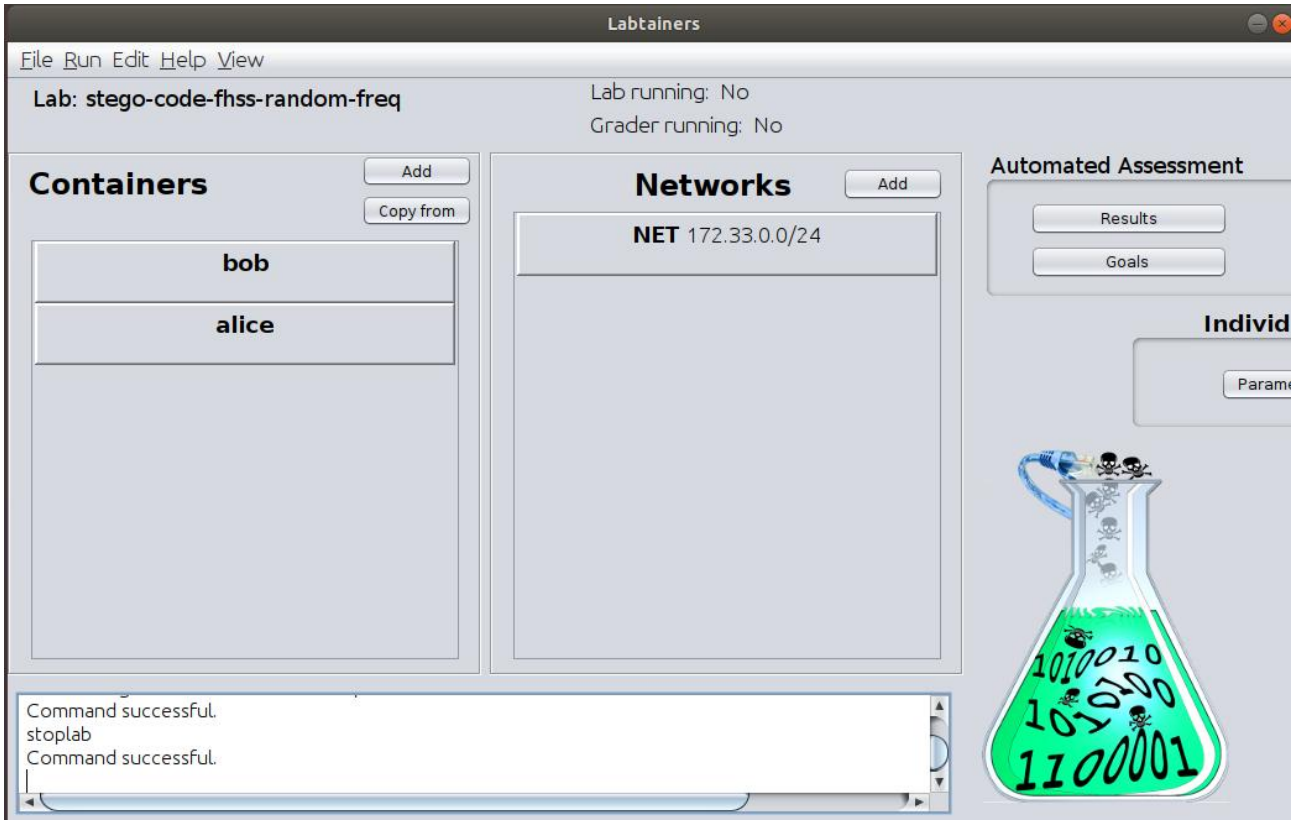
hide_message: Giấu tin vào file âm thanh hay chưa

send_file : Kiểm tra đã gửi file âm thanh từ máy bob đến máy alice hay chưa

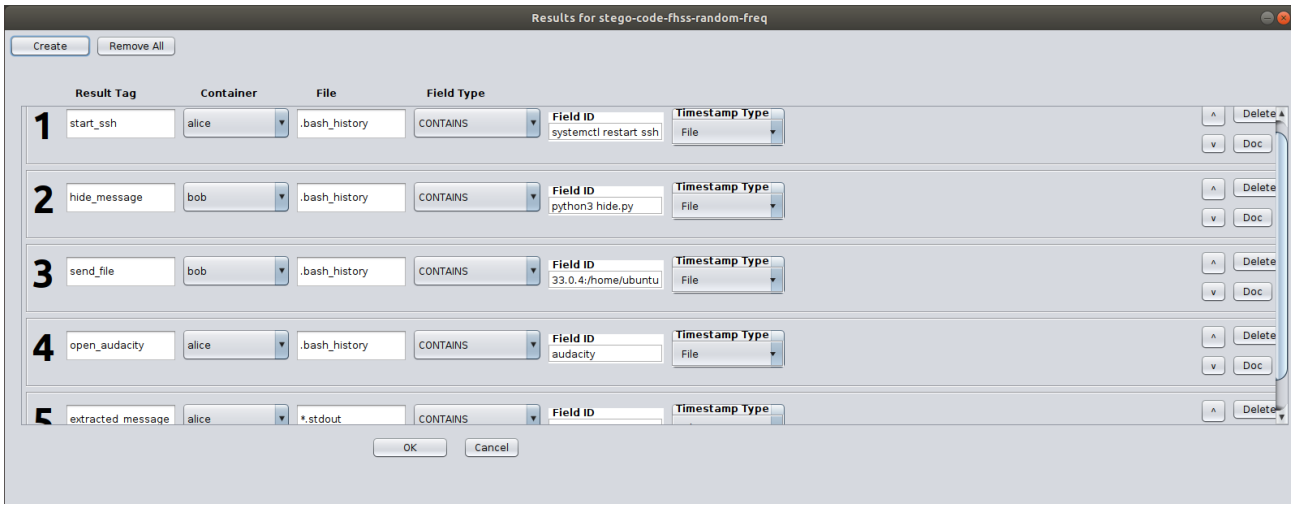
open_audacity: Kiểm tra đã sử dụng audacity hay chưa

extracted_message: Kiểm tra xem đã tách tin hay chưa

2.2 Cài đặt và cấu hình máy ảo



Hình 2 Giao diện labedit



Hình 3 Giao diện result.config

2.3 Tích hợp và triển khai

Bài thực hành được tiển khai như sau:

Docker

Đường dẫn: Docker Hub

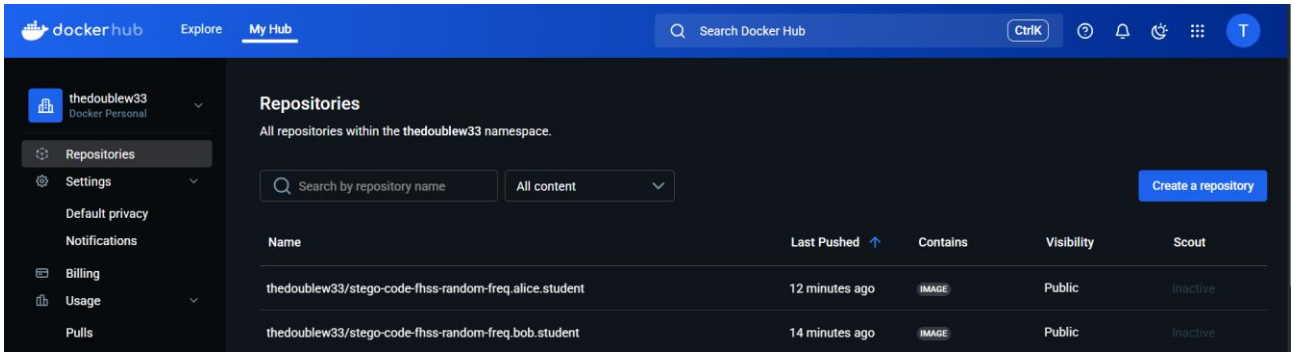
Thêm registry cho bài thực hành

Truy cập vào thư mục trunk/distrib gõ lệnh: *docker login* đăng nhập tài khoản DockerHub

Sử dụng lệnh *./publish.py -d -l stego-code-fhss-random-freq* để đẩy images của bài thực hành lên DockerHub

```
student@ubuntu:~/labtainer/trunk/distrib$ ./publish.py -d -l stego-code-fhss-random-freq
adding [nmaplab]
adding [httplab]
adding [liveforensics]
adding [bind-shell]
adding [tlab]
adding [metasploitable-test]
adding [kali-test]
adding [my-remote-dns]
adding [remote-dns2]
adding [remote-dns]
adding [backups]
adding [centos-log]
adding [dhcp-test]
adding [xlab]
adding [softplc]
adding [iptables]
adding [grfics]
adding [usbtest]
adding [ida]
adding [centossix]
adding [routing-basics2]
adding [shellbasics]
adding [ldaptst]
adding [mariadbtst]
fb4bffaf7f32
5bdb038241e7
b12455680023
Untagged: stego-code-fhss-random-freq.alice.student:latest
Deleted: sha256:20d6d0ada5df9d6dd304840f8ac6a5df1f4b815f0781e2e8f8b8c63b9e025886
Deleted: sha256:46e3f9309abcc90a6c005577216ebf2cfb2d75697f6604a4bf7cc53a73d1dccd
Untagged: stego-code-fhss-random-freq.alice.student.tmp:latest
Deleted: sha256:64ffc51b011a0a68317d82f415923b96305d466a0ae45c82f7dd1192f164d39a
Deleted: sha256:047f331f81708c2445c301b5ed43fff17435562c8c21186bfbae505356354a1d
Deleted: sha256:588b80194aded6322adb15022f70a7588e1b8b5a8073df32be15cd8f5250768e
Deleted: sha256:b2be8b37c3456213d9cea31e502c2007e2e1dc9aeab6235c4081ee92c20e3971
Deleted: sha256:0b17fc1f6e14aa73a9b15d6b620f04868eeb75cf0d2aee9de1b95703fbdd8337
Deleted: sha256:0fbfe22bd0b36bda3e29636c6b31b7bb1a5e38e161488657dd6f8c294dcda2d5
Deleted: sha256:a9decaf65a7df0b08ec458cded6072d107eef73b8bf469acddaea82be01e67c4
Deleted: sha256:0e511b1d4b0910eb7445d6b9e92a64248237a214886cc7b93ecebdc1a3f600f6
Deleted: sha256:222080d70cce7a33b0cc1c6653ea7a1d15b58a52e5d94e40d63609d7f0c02758
```

Hình 4 Đẩy image bài thực hành lên docker hub



Hình 5 Image trên dockerhub

Github

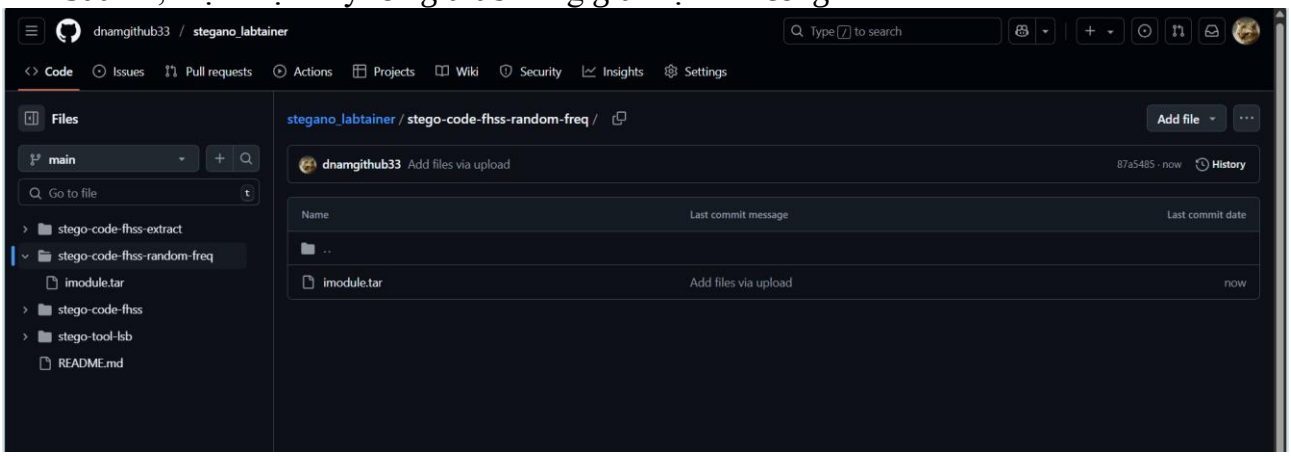
Đường dẫn: tại đây

Ở đường dẫn $LABTAINER_DIR/distrib, tạo file tar bằng create-imodules.sh

```
student@ubuntu:~/labtainer/trunk/scripts/designer/bin$ ./create-imodules.sh
lab is stego-code-fhss-random-freq
Do docs
************************************************************
**  Post /home/student/labtainer/trunk/imodule.tar to your web server **
************************************************************
```

Hình 6 Tạo file imodule

Sau đó, thực hiện đẩy lên github bằng git hoặc thủ công

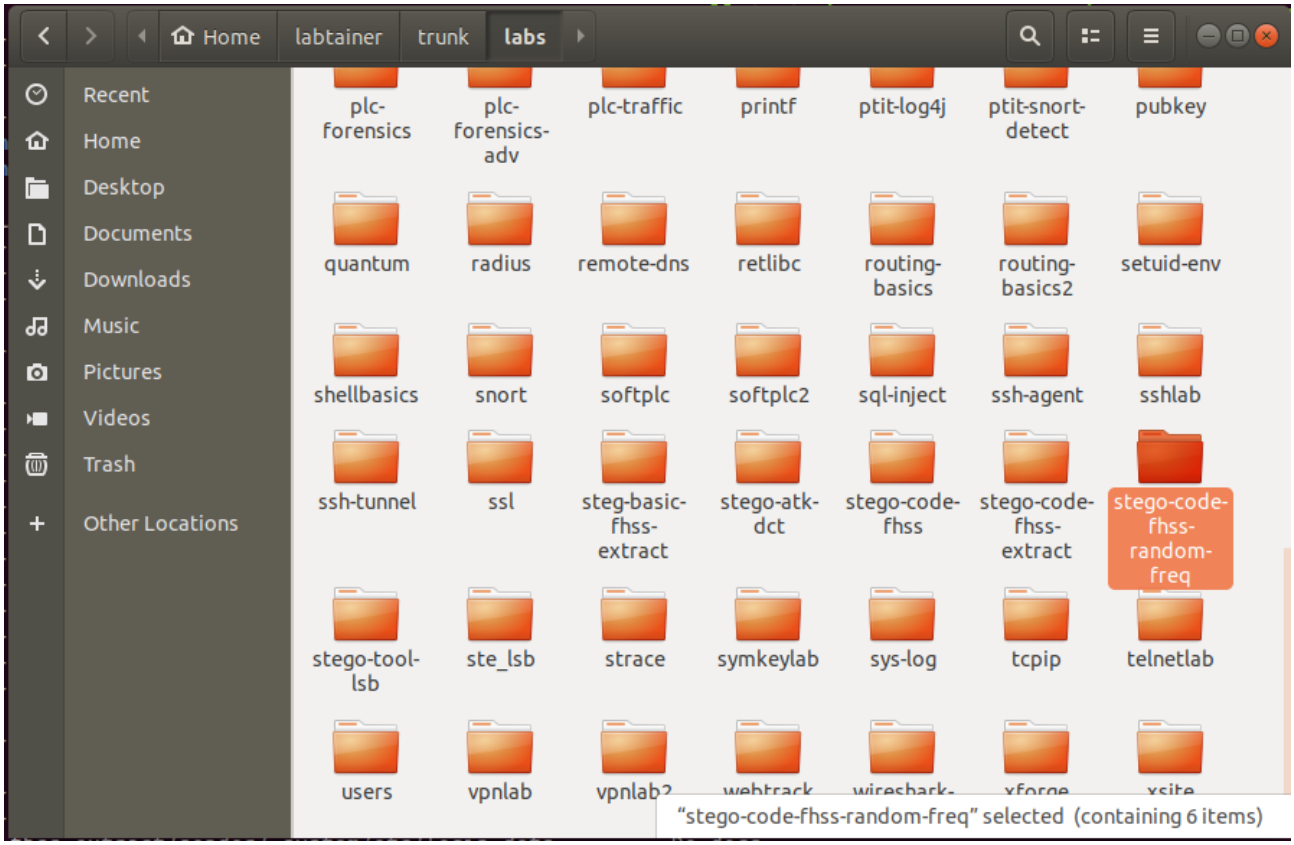


Hình 7 File imodule trên github

CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ.

Bài thực hành đã được xây dựng thành công, dưới đây là hình ảnh minh họa về bài thực hành:

Khởi động bài lab, tải bài thực hành bằng imodule, trong trường hợp không dùng được imodule, giải nén file thực hành trong folder labtainer/trunk/labs:



Hình 8 Cài đặt bài thực hành

Vào terminal, gõ:

Labtainer -r stego-code-fhss-random-freq

(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)

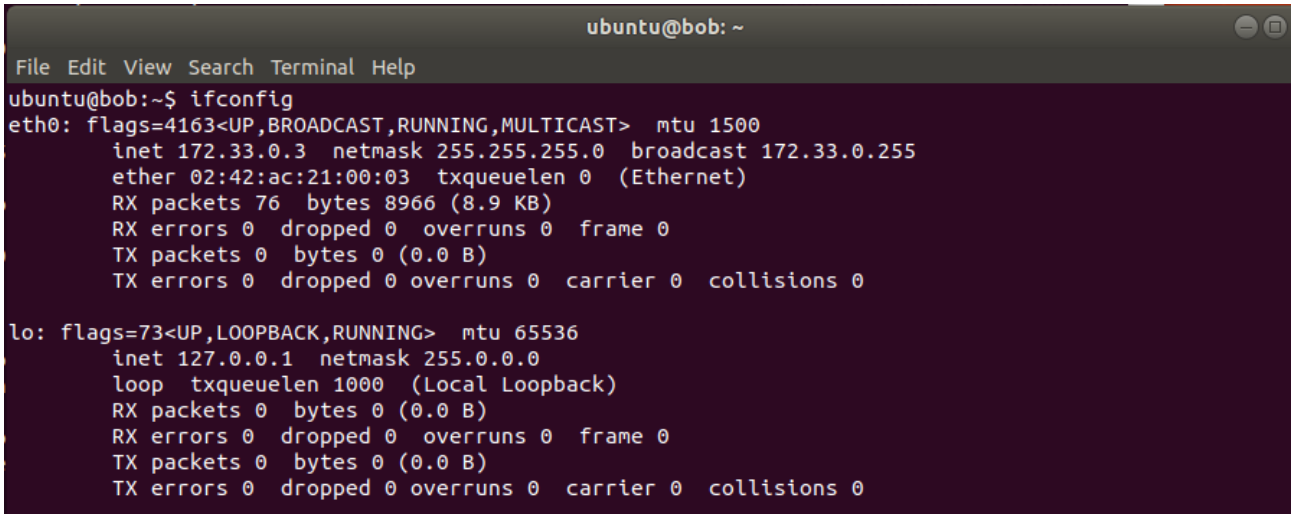
```
student@ubuntu:~/labtainer/labtainer-student$ labtainer stego-code-fhss-random-freq
latest: Pulling from thedoublew33/stego-code-fhss-random-freq.bob.student
51de2797f095: Pull complete
66e67e836d68: Pull complete
8e7e24552aa8: Pull complete
bde42c570a5d: Pull complete
9371b3f58f3d: Pull complete
23c70eb78830: Pull complete
addc7291ae0d: Pull complete
11e253110f34: Pull complete
264c58d22dc2: Pull complete
1326a3fdabad: Pull complete
2f8608921467: Pull complete
6b5a151afeb1: Pull complete
Digest: sha256:ec85fd37020abbd0c208b3f31df8fda1ce1c474b386d934334f88a0cd56e9b55
Status: Downloaded newer image for thedoublew33/stego-code-fhss-random-freq.bob.student:latest
latest: Pulling from thedoublew33/stego-code-fhss-random-freq.alice.student
388ae51d6659: Pull complete
d98eac724bfa: Pull complete
8e7e24552aa8: Pull complete
4e1a8056ad0d: Pull complete
4e115f2d1da7: Pull complete
4e7c4467d2e1: Pull complete
e3a34405aa41: Pull complete
a6699b88faa6: Pull complete
61afff769dd0: Pull complete
9b30cea14208: Pull complete
a817dcd16258: Pull complete
66db6306d1cf: Pull complete
cde9df3d8f73: Pull complete
Digest: sha256:ead610d2bb32fe6341abf37b51ce1fa00795f18bac5b3b949b6c3db20fe978d0
```

Hình 9 Khởi động badi lab

- Kiểm tra địa chỉ IP ở 2 máy.

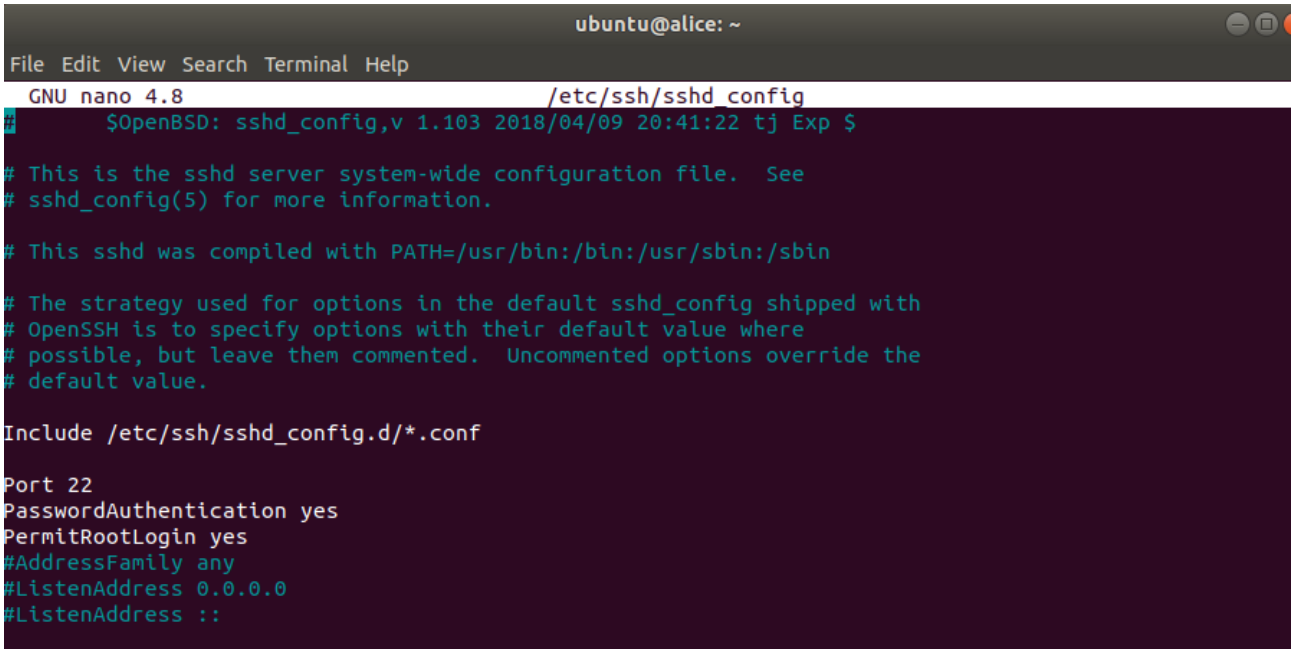
```
ubuntu@alice: ~
File  Edit  View  Search  Terminal  Help
ubuntu@alice:~$ ls
extract.py
ubuntu@alice:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.33.0.4  netmask 255.255.255.0  broadcast 172.33.0.255
        ether 02:42:ac:21:00:04  txqueuelen 0  (Ethernet)
        RX packets 61  bytes 7553 (7.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1862 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1466 (1.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1466 (1.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Hình 10 Địa chỉ IP máy alice



Hình 11 Địa chỉ IP máy bob

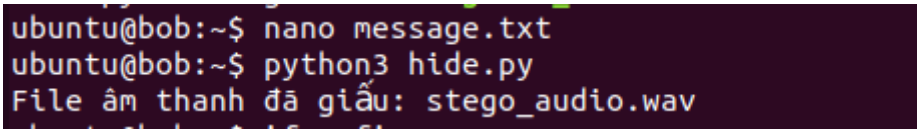
- Ở máy receiver, sinh viên cài đặt cấu hình SSH như sau:
    - Ở file /etc/ssh/sshd_config, đảm bảo cấu hình mở cổng 22, cho phép truy cập bằng mật khẩu vào đăng nhập bằng root:
        - Port 22
        - PasswordAuthentication yes
        - PermitRootLogin yes



Hình 12 Cấu hình SSH máy receiver

- Khởi động lại SSH với cấu hình trên: ***sudo systemctl restart ssh.***

- Sang máy bob, chỉnh sửa thông điệp và giấu tin:



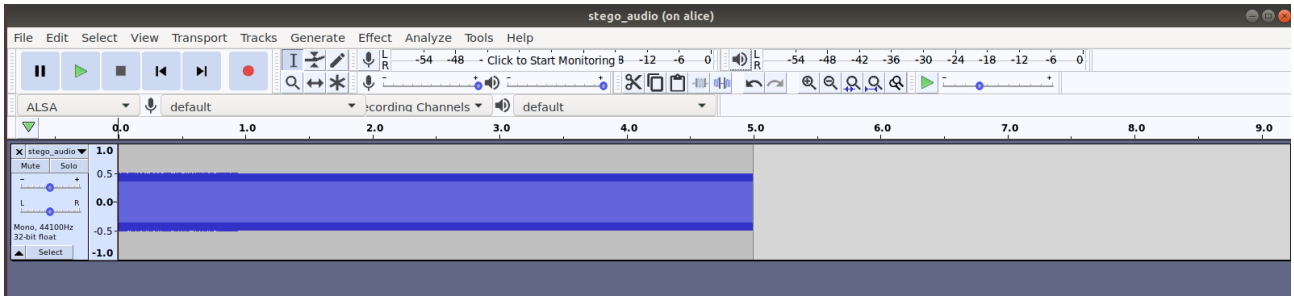
Hình 13 Giấu tin

- Ở máy sender, gửi file âm thanh bằng lệnh ***scp stego_audio.wav ubuntu@172.33.0.4:/home/ubuntu*** với password là 1.

```
ubuntu@bob:~$ scp stego_audio.wav ubuntu@172.33.0.4:/home/ubuntu
The authenticity of host '172.33.0.4 (172.33.0.4)' can't be established.
ECDSA key fingerprint is SHA256:ZtE8xi5Y50aUktZ/XtgjIs1c5jxYQB84Vq5ofmlgGng.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.33.0.4' (ECDSA) to the list of known hosts.
ubuntu@172.33.0.4's password:
stego_audio.wav                                   100%  431KB 113.3MB/s   00:00
ubuntu@bob:~$
```

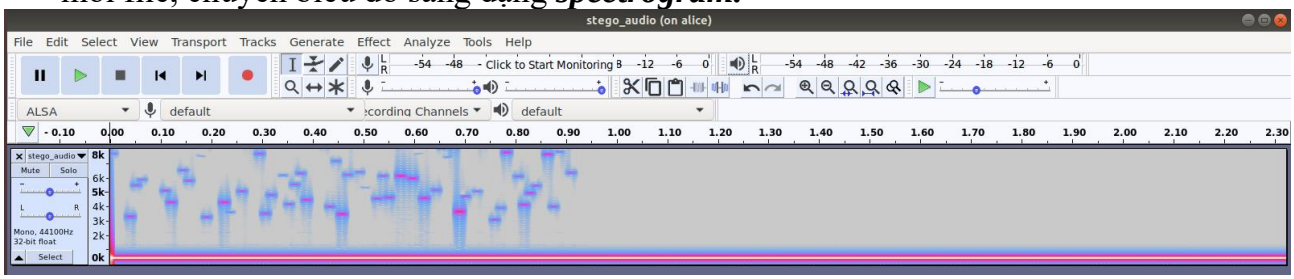
Hình 14 Gửi file sang máy alice

- Ở máy alice, sinh viên đọc file âm thanh nhận được bằng audacity.



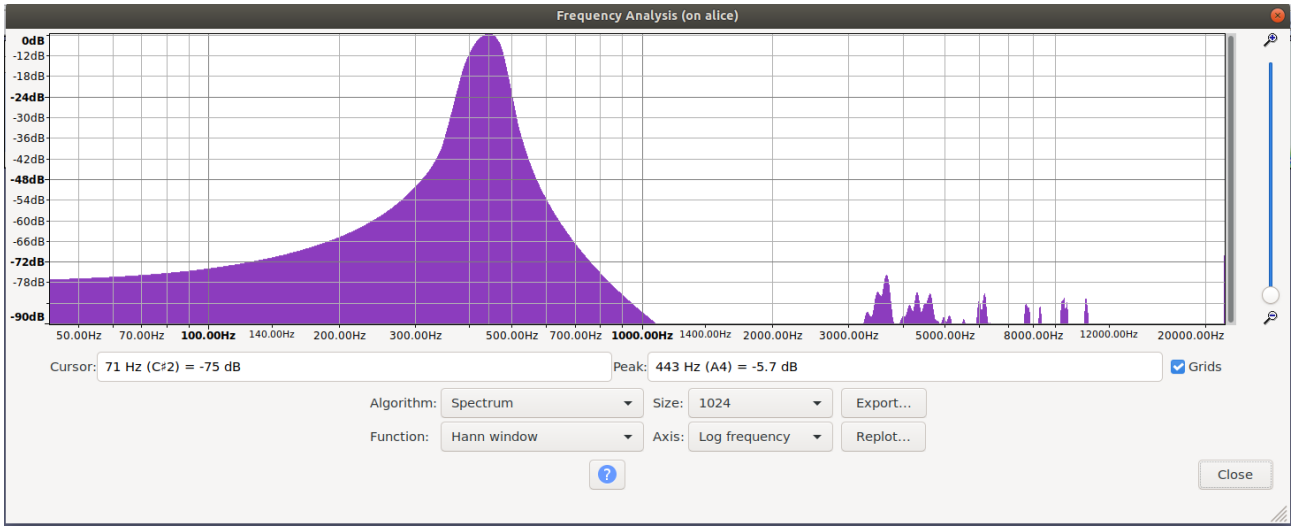
Hình 15 Đọc file âm thanh bằng audacity

- Audacity sẽ hiển thị file âm thanh dưới dạng biểu đồ ***waveform***, nhấn vào dấu 3 chấm ở mỗi file, chuyển biểu đồ sang dạng ***spectrogram.***



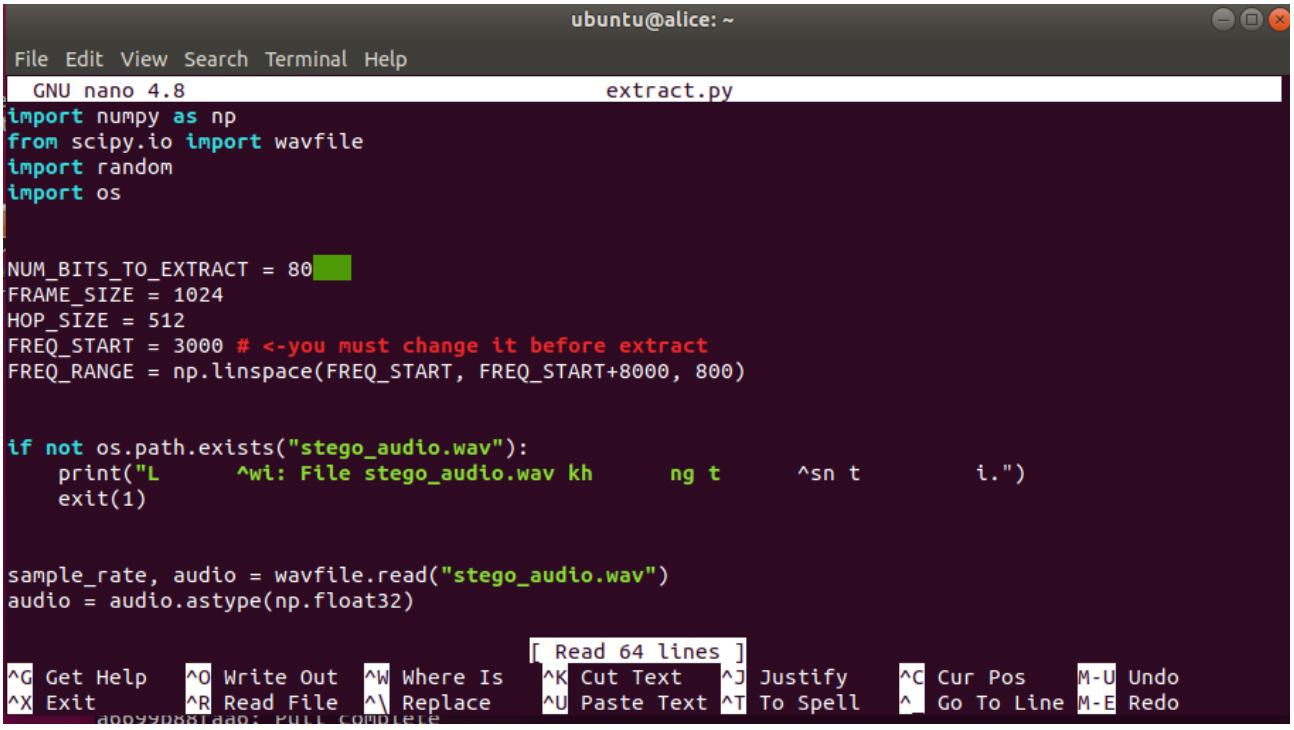
Hình 16 Spectrogram

- Ở phần đầu của file âm thanh, có những đốm sáng là phần giấu tin. Kiểm tra plot spectrum trong Analyze để xem những đốm sáng này nằm ở vùng tần số nào.



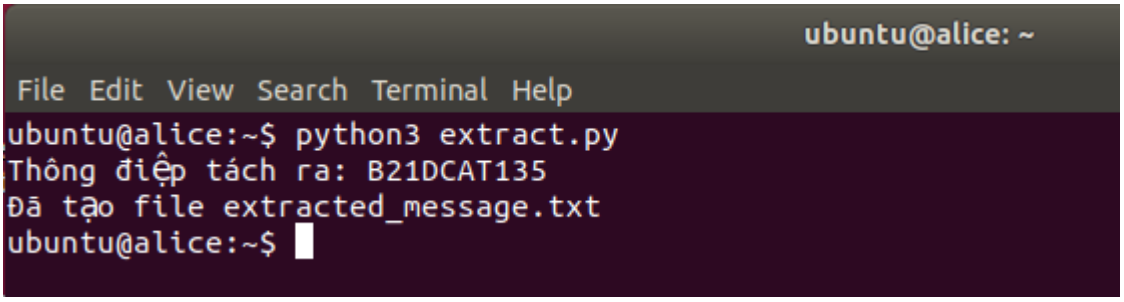
Hình 17 Plot Spectrum

- Sau khi biết được vùng tần số, chỉnh sửa phần freq_start trong file ***extract.py*** thành điểm bắt đầu của khoảng tần số.



Hình 18 sửa extract.py

- Tách tin với file extract vừa sửa.



Hình 19 Tin đã tách

- Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:
  stoplab

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

Kết quả checkwork

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/stego-code-fhss-random-freq
Labname stego-code-fhss-random-freq

Student              |       start_ssh |    hide_message |       send_file |   open_audacity | extracted_messa |
==================== | =============== | =============== | =============== | =============== | =============== |
b21dcat135           |               Y |               Y |               Y |               Y |               Y |
What is automatically assessed for this lab:
```

Hình 20 Checkwork

Khởi động lại bài lab:

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

labtainer –r stego-code-fhss-random-freq

TÀI LIỆU THAM KHẢO

[1] Bài giảng Các kỹ thuật giấu tin, PGS. TS Đỗ Xuân Chợ