

**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: D21CQAT01-B
TÊN BÀI: TÁCH TIN ẨN TRONG ÂM THANH
BẰNG KỸ THUẬT FHSS**

Sinh viên thực hiện:

B21DCAT105 Đặng Thị Thanh Huyền

Giảng viên: PGS.TS. Đỗ Xuân Chơ

HỌC KỲ 2 NĂM HỌC 2024-2025

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ.....	3
DANH MỤC CÁC BẢNG BIỂU.....	3
DANH MỤC CÁC TỪ VIẾT TẮT	4
CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH.....	5
1.1 Giới thiệu chung về bài thực hành	5
1.2 Nội dung và hướng dẫn bài thực hành	5
CHƯƠNG 2: PHÂN TÍCH YÊU CẦU BÀI THỰC HÀNH.....	8
2.1 Thiết kế bài thực hành	8
2.2 Cài đặt và cấu hình máy ảo	8
2.3 Tích hợp và triển khai	10
CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	13
TÀI LIỆU THAM KHẢO	18

DANH MỤC CÁC HÌNH VẼ

Hình 1: Yêu cầu checkwork.....	8
Hình 2: Giao diện Labedit.....	9
Hình 3: Cài đặt phần result.....	9
Hình 4: Nội dung dockerfiles.....	10
Hình 5: Đẩy image bài thực hành lên docker hub.....	11
Hình 6: Trạng thái bài thực hành trên Docker Hub.....	11
Hình 7: tạo thủ công ở thư mục ~/labs bằng lệnh	12
Hình 8: thực hiện đẩy lên github bằng git hoặc thủ công.....	12
Hình 9: tải bài thực hành bằng imodule	13
Hình 10: Mở bài lab	13
Hình 11: kiểm tra ip máy sender	14
Hình 12: kiểm tra ip máy receiver.....	14
Hình 13: kiểm tra các thư mục có sẵn trong container sender và receiver.....	14
Hình 14: in ra màn hình nội dung file message.txt từ sender.....	15
Hình 15: Tìm và thay đổi các dòng config vào file /etc/ssh/sshd_config	15
Hình 16: restart lại dịch vụ ssh-server.....	16
Hình 17: tiến hành gửi file audio chứa thông điệp đến người nhận.....	16
Hình 18: tiến hành tách thông điệp	16
Hình 19: Kết quả checkwork.....	17

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Thuật ngữ tiếng Anh/Giải thích	Thuật ngữ tiếng Việt/Giải thích
SSH	Secure Shell	Giao thức kết nối mạng an toàn
IP	Internet Protocol	Giao thức liên mạng
LAN	Local Area Network	Mạng cục bộ
FHSS	Frequency Hopping Spread Spectrum	Phổ trải nhảy tần
SCP	Secure Copy Protocol	Giao thức sao chép bảo mật
GUI	Graphical User Interface	Giao diện đồ họa
venv	Virtual Environment	Môi trường ảo
PyCryptodome	Python Cryptographic Library	Thư viện mã hóa trong Python
GitHub	Git Repository Hosting Service	Dịch vụ lưu trữ mã nguồn sử dụng Git
Docker	Containerization Platform	Nền tảng đóng gói và triển khai ứng dụng bằng container

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). Đây là một phương pháp mã hóa và truyền tin có tính bảo mật cao, thường được sử dụng trong các hệ thống truyền thông quân sự hoặc không dây để tránh bị phát hiện và gây nhiễu.

FHSS (Frequency Hopping Spread Spectrum) là kỹ thuật truyền tín hiệu bằng cách thay đổi tần số mang theo một chuỗi định sẵn gọi là *hop pattern* (mẫu nhảy tần). Trong kỹ thuật giấu tin, thuật toán FHSS được ứng dụng để phân tán dữ liệu bí mật vào các vị trí khác nhau trong tệp âm thanh, theo một dãy nhảy tần giả lập (pseudo-random), giúp tăng tính bảo mật và làm cho dữ liệu khó bị phát hiện.

Nguyên lý hoạt động của giấu và tách tin bằng FHSS

- Bước 1 – Giấu tin:

Trước đó, một thông điệp (thường là dạng văn bản) được mã hóa nhị phân. Sau đó, mỗi bit sẽ được nhúng vào một vị trí cụ thể trong tệp âm thanh dựa trên một chuỗi *hop sequence*. Chuỗi này được tạo từ một khóa bí mật (secret key) và đóng vai trò quyết định vị trí của các bit trong toàn bộ tệp âm thanh.

- Bước 2 – Tách tin:

Để tách được thông điệp, cần biết chính xác chuỗi *hop pattern* đã được sử dụng khi giấu tin. Khi đã có khóa bí mật, chương trình sẽ tạo lại đúng chuỗi nhảy tần để định vị lại chính xác các vị trí chứa dữ liệu trong tệp âm thanh. Sau đó, các bit được trích xuất tuần tự và ghép lại thành thông điệp gốc.

Đặc điểm của kỹ thuật tách tin bằng FHSS

- Tính phân tán cao: dữ liệu được rải đều trên toàn bộ âm thanh nên khó bị phát hiện.
- Chống nhiễu tốt: do việc nhảy tần giúp tránh các đoạn âm bị biến đổi nhẹ.
- Cần có khóa để trích xuất chính xác: người không có khóa sẽ không thể tái tạo được chuỗi nhảy tần, do đó không thể tách được dữ liệu.

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 cách sử dụng các đặc tính của tín hiệu âm thanh và kỹ thuật nhảy tần. Đồng thời làm quen với môi trường thực hành an toàn thông tin qua các lệnh cấu hình và phân tích.

1.2.2 Yêu cầu đối với sinh viên

Hiểu rõ khái niệm steganography và vai trò của tách tin trong âm thanh.

Nắm được nguyên lý hoạt động của thuật toán Frequency Hopping Spread Spectrum (FHSS).

Hiểu cách sử dụng giao thức SSH, SCP để thực hiện kết nối và truyền dữ liệu giữa các máy trong mạng LAN.

1.2.3 Nội dung thực hành

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

imodule <https://github.com/DTHuyn/steg-basic-fhss-extract/raw/master/steg-basic-fhss-extract.tar>

Vào terminal, gõ:

Labtainer -r steg-basic-fhss-extract

(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, một cái là đại diện cho máy gửi: **sender**, một cái là đại diện cho máy nhận: **receiver**. Biết rằng 2 máy nằm cùng mạng LAN 172.20.0.0/24.

- Tiến hành kiểm tra ip của 2 máy
- Thực hiện lệnh kiểm tra các thư mục có sẵn trong container sender và receiver.
- Sinh viên thực hiện in ra màn hình nội dung file message.txt từ sender.
- Tìm và thay đổi các dòng config vào cuối file /etc/ssh/sshd_config trên receiver.

Port 22

PasswordAuthentication yes

PermitRootLogin yes

- Cần restart lại dịch vụ ssh-server: *sudo systemctl restart ssh*
- Tại terminal sender, tiến hành gửi file audio chứa thông điệp đến người nhận.

scp stego_audio.wav ubuntu@<ip_máy_receiver>:/home/ubuntu/

Nhập password là *ubuntu*

- Sau khi terminal receiver nhận được file audio, tiến hành tách thông điệp với cấu trúc lệnh *python3 extract_message.py*
- So sánh thông điệp được tách ra với nội dung file gửi xem có trùng khớp không
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:

labyrinth -r steg-basic-fhss-extract

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU BÀI THỰC HÀNH

2.1 Thiết kế bài thực hành

Bài lab gồm 2 container là sender và receiver. Cấu hình mạng 2 máy nằm cùng mạng LAN 172.20.0.0/24. Setup ip cho sender là 172.20.0.12, máy receiver là 172.20.0.11

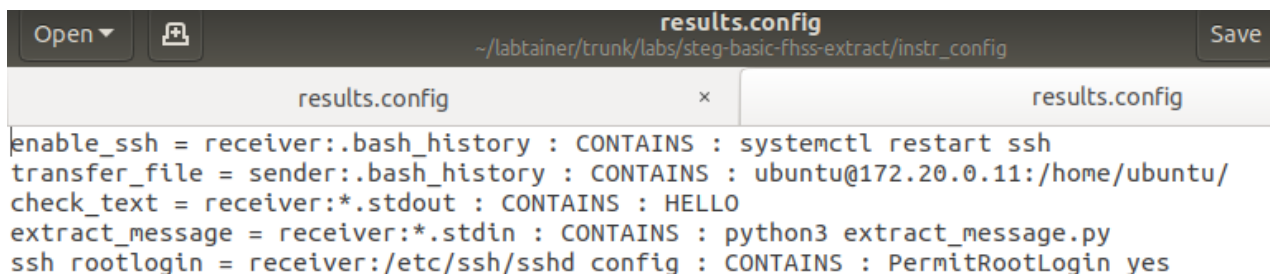
Cấu hình Docker

- Bài lab chạy với image là .network2
- Cần cài thêm môi trường ảo hóa venv và thư viện pycryptodome trong dockerfiles
- Docs lưu lại hướng dẫn thực hành cho sinh viên

Các nhiệm vụ cần thực hiện để thành công:

- Khởi động dịch vụ SSH
- Gửi audio đã giấu tin cho bên nhận
- Tiến hành tách tin thành công và nhận được thông điệp
- 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:



Hình 1: Yêu cầu checkwork

enable_ssh : Tiến hành kiểm tra đã khởi động dịch vụ SSH chưa

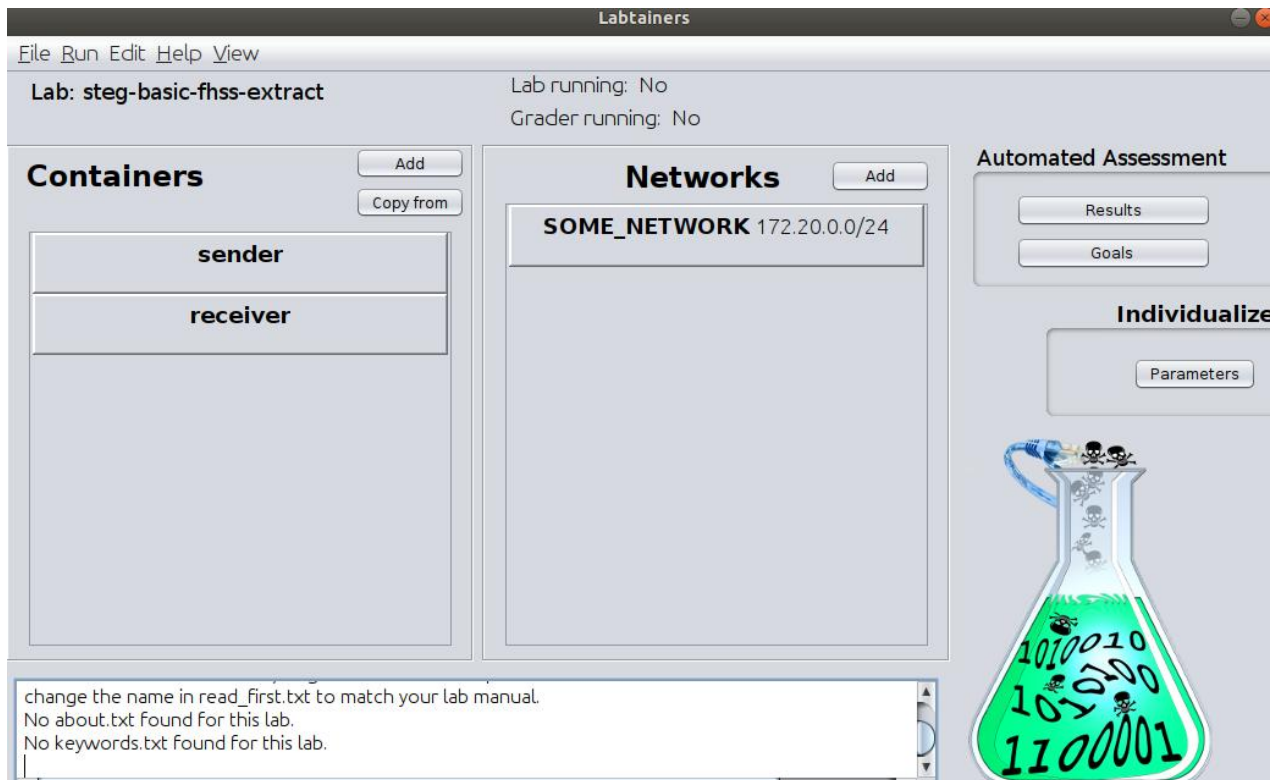
transfer_file : Kiểm tra đã thực hiện đúng câu lệnh gửi file audio cho bên nhận chưa

check_text : Kiểm tra thông điệp nhận có chính xác không

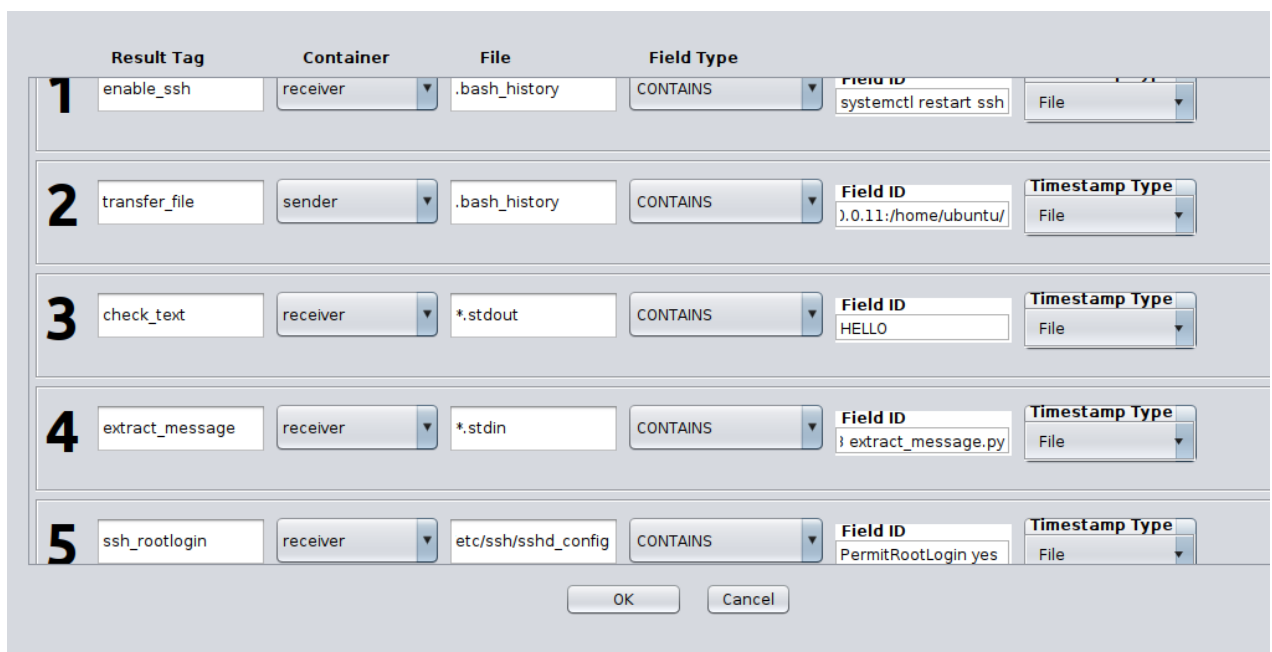
extract_message : Kiểm tra đã thực hiện câu lệnh tách tin chưa

ssh_rootlogin : Kiểm tra đã chỉnh sửa file cấu hình ssh chưa

2.2 Cài đặt và cấu hình máy ảo



Hình 2: Giao diện Labedit



Hình 3: Cài đặt phân result

```
Open [icon] Dockerfile.steg-basic-fhss-extract.receiver.student
~/labtainer/trunk/labs/s...fhss-extract/dockerfiles

#
# Labtainer Dockerfile
#
# This is the default Labtainer Dockerfile template, please choose the appropriate
# base image below.
#
# The labtainer.base image includes the following packages:
#   build-essential expect file gcc-multilib gdb iputils-ping less man manpages-dev
#   net-tools openssh-client python sudo tcl8.6 vim zip hexedit rsyslog
#
# The labtainer.network image adds the following packages:
#   openssl openssh-server openvpn wget tcpdump update-inetd xinetd
#
ARG registry
FROM $registry/labtainer.network2
#FROM $registry/labtainer.network
#FROM $registry/labtainer.centos
#FROM $registry/labtainer.lamp
#
# lab is the fully qualified image name, e.g., mylab.some_container.student
# labdir is the name of the lab, e.g., mylab
# imagedir is the name of the container
# user_name is the USER from the start.config, if other than ubuntu,
#           then that user must be added in this dockerfile
#           before the USER command
#
ARG lab
ARG labdir
ARG imagedir
ARG user_name
ARG password
ARG apt_source
ARG version
LABEL version=$version
ENV APT_SOURCE $apt_source
RUN /usr/bin/apt-source.sh
#
# put package installation here, e.g.,
#   RUN apt-get update && apt-get install -y --no-install-recommends somepackage
#
#
#
```

Hình 4: Nội dung dockerfiles

2.3 Tích hợp và triển khai

Bài thực hành được triển khai như sau:

Docker

Đường dẫn: <https://hub.docker.com/repositories/dthuynd>

Thêm registry cho bài thực hành

Truy cập vào thư mục trunk/distrib gõ lệnh: *docker login* đăng nhập tài khoản DockerHub với *Username: dthuynd* và *password: *****

Sử dụng lệnh *./publish.py -d -l steg-basic-fhss-extract* để đẩy images của bài thực hành lên DockerHub

```

student@ubuntu:~/labtainer/trunk/distrib$ ./publish.py -d -l steg-basic-fhss-extract
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 [mariadbtest]
No images for steg-basic-fhss-extract
Image steg-basic-fhss-extract.sender.student exists nowhere, so force the build
Image steg-basic-fhss-extract.receiver.student exists nowhere, so force the build
Sending build context to Docker daemon 779.3kB
Step 1/4 : FROM steg-basic-fhss-extract.sender.student
--> 9af0af5d89d8
Step 2/4 : ARG version
--> Running in c314dc7bc228

```

Hình 5: Đẩy image bài thực hành lên docker hub

Repositories			
All repositories within the dthuyng namespace.			
<input type="text" value="Search by repository name"/>		All content ▼	
Name	Last Pushed ↑	Contains	Visibility
dthuyng/steg-basic-fhss-extract.receiver.student	3 minutes ago	IMAGE	Public
dthuyng/steg-basic-fhss-extract.sender.student	5 minutes ago	IMAGE	Public

Hình 6: Trạng thái bài thực hành trên Docker Hub

Github

Đường dẫn: <https://github.com/DTHuyn/steg-basic-fhss-extract>

Ở đường dẫn \$LABTAINER_DIR/distrib, tạo file tar bằng create-imodules.sh hoặc tạo thủ công ở thư mục ~/labs bằng lệnh:

```
tar -cvf steg-basic-fhss-extract.tar steg-basic-fhss-extract
```

```

student@ubuntu:~/labtainer/trunk/labs$ tar -cvf steg-basic-fhss-extract.tar steg-basic-fhss-extract
steg-basic-fhss-extract/
steg-basic-fhss-extract/sender/
steg-basic-fhss-extract/sender/_bin/
steg-basic-fhss-extract/sender/_bin/fixlocal.sh
steg-basic-fhss-extract/sender/steg-basic-fhss-extract.sender.student.tar.gz
steg-basic-fhss-extract/sender/stego_audio.wav
steg-basic-fhss-extract/sender/home_tar/
steg-basic-fhss-extract/sender/home_tar/home.tar
steg-basic-fhss-extract/sender/message.txt
steg-basic-fhss-extract/sender/sys_tar/
steg-basic-fhss-extract/sender/sys_tar/sys.tar
steg-basic-fhss-extract/sender/sys_steg-basic-fhss-extract.sender.student.tar.gz
steg-basic-fhss-extract/sender/_system/
steg-basic-fhss-extract/sender/_system/etc/
steg-basic-fhss-extract/sender/_system/etc/securetty
steg-basic-fhss-extract/sender/_system/etc/login.defs
steg-basic-fhss-extract/config/
steg-basic-fhss-extract/config/start.config
steg-basic-fhss-extract/config/parameter.config
steg-basic-fhss-extract/config/receiver-home_tar.list
steg-basic-fhss-extract/config/sender-home_tar.list
steg-basic-fhss-extract/receiver/
steg-basic-fhss-extract/receiver/sys_steg-basic-fhss-extract.receiver.student.tar.gz
steg-basic-fhss-extract/receiver/_bin/
steg-basic-fhss-extract/receiver/_bin/fixlocal.sh
steg-basic-fhss-extract/receiver/home_tar/
steg-basic-fhss-extract/receiver/home_tar/home.tar
steg-basic-fhss-extract/receiver/sys_tar/
steg-basic-fhss-extract/receiver/sys_tar/sys.tar
steg-basic-fhss-extract/receiver/steg-basic-fhss-extract.receiver.student.tar.gz
steg-basic-fhss-extract/receiver/extract_message.py

```

Hình 7: tạo thủ công ở thư mục ~/labs bằng lệnh

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

The screenshot shows the GitHub interface for the repository 'steg-basic-fhss-extract'. At the top, it indicates the repository is 'Public'. Below the repository name, there are buttons for 'Pin', 'Unwatch', and a notification bell. The commit history section shows two commits by user DTHuyn: 'Create README.md' (3b9ffdf, 4 hours ago, 2 commits) and 'Add steg-basic-fhss-extract lab .tar file' (4 hours ago). The README section is expanded, showing a link to the raw tar file: <https://github.com/DTHuyn/steg-basic-fhss-extract/raw/master/steg-basic-fhss-extract.tar>.

Hình 8: thực hiện đẩy lên github bằng git hoặc thủ công

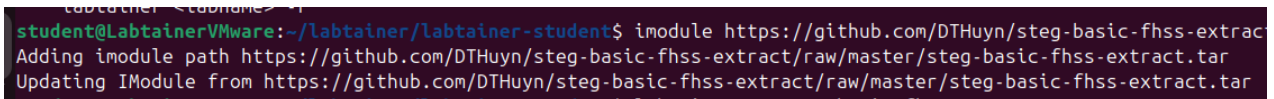
File *steg-basic-fhss-extract.tar* chứa bài thực hành

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:

imodule <https://github.com/DTHuyn/steg-basic-fhss-extract/raw/master/steg-basic-fhss-extract.tar>



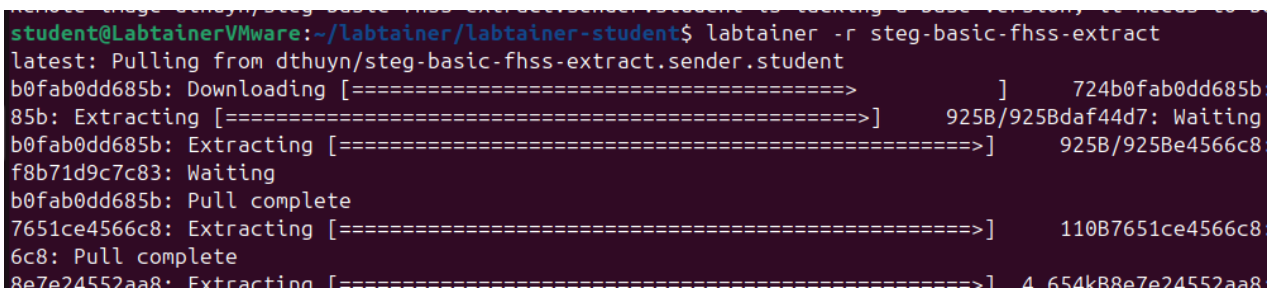
```
student@LabtainerVMware:~/labtainer/labtainer-student$ imodule https://github.com/DTHuyn/steg-basic-fhss-extract/raw/master/steg-basic-fhss-extract.tar
Adding imodule path https://github.com/DTHuyn/steg-basic-fhss-extract/raw/master/steg-basic-fhss-extract.tar
Updating IModule from https://github.com/DTHuyn/steg-basic-fhss-extract/raw/master/steg-basic-fhss-extract.tar
```

Hình 9: tải bài thực hành bằng imodule

Vào terminal, gõ:

Labtainer -r steg-basic-fhss-extract

(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@LabtainerVMware:~/labtainer/labtainer-student$ labtainer -r steg-basic-fhss-extract
latest: Pulling from dthuhn/steg-basic-fhss-extract.sender.student
b0fab0dd685b: Downloading [=====>] 724b0fab0dd685b
85b: Extracting [=====>] 925B/925Bdaf44d7: Waiting
b0fab0dd685b: Extracting [=====>] 925B/925Be4566c8
f8b71d9c7c83: Waiting
b0fab0dd685b: Pull complete
7651ce4566c8: Extracting [=====>] 110B7651ce4566c8
6c8: Pull complete
8e7e24552aa8: Extracting [=====>] 4 654kB8e7e24552aa8
```

Hình 10: Mở bài lab

Sau khi khởi động xong, màn hình sẽ xuất hiện 2 terminal, một cái là đại diện cho máy gửi: **sender**, một cái là đại diện cho máy nhận: **receiver**. Biết rằng 2 máy nằm cùng mạng LAN 172.20.0.0/24.

- Tiến hành kiểm tra ip của 2 máy

```

ubuntu@sender:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.0.12 netmask 255.255.255.0 broadcast 172.20.0.255
    ether 62:a6:1c:e4:f6:fc txqueuelen 0 (Ethernet)
    RX packets 56 bytes 7278 (7.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3 bytes 126 (126.0 B)
    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
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Hình 11: kiểm tra ip máy sender

```

ubuntu@receiver:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.0.11 netmask 255.255.255.0 broadcast 172.20.0.255
    ether 66:79:9f:8a:f4:9b txqueuelen 0 (Ethernet)
    RX packets 56 bytes 7278 (7.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3 bytes 126 (126.0 B)
    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
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Hình 12: kiểm tra ip máy receiver

- Thực hiện lệnh kiểm tra các thư mục có sẵn trong container sender và receiver.

```

ubuntu@sender:~$ ls
message.txt  stego_audio.wav

```

Hình 13: kiểm tra các thư mục có sẵn trong container sender và receiver

- Sinh viên thực hiện in ra màn hình nội dung file message.txt từ sender.

```
ubuntu@sender:~$ cat message.txt  
HELLO  
ubuntu@sender:~$
```

Hình 14: in ra màn hình nội dung file message.txt từ sender

- Tìm và thay đổi các dòng config vào file /etc/ssh/sshd_config trên receiver.

Port 22

PasswordAuthentication yes

PermitRootLogin yes

```
# This is the sshd server system-wide configuration file. See  
# sshd_config(5) for more information.  
  
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin  
  
# The strategy used for options in the default sshd_config shipped with  
# OpenSSH is to specify options with their default value where  
# possible, but leave them commented. Uncommented options override the  
# default value.  
  
Include /etc/ssh/sshd_config.d/*.conf  
  
Port 22  
#AddressFamily any  
#ListenAddress 0.0.0.0  
#ListenAddress ::  
  
#HostKey /etc/ssh/ssh_host_rsa_key  
#HostKey /etc/ssh/ssh_host_ecdsa_key [ Read 123 lines ]
```

Hình 15: Tìm và thay đổi các dòng config vào file /etc/ssh/sshd_config

- Cần restart lại dịch vụ ssh-server: *sudo systemctl restart ssh*


```

ubuntu@receiver:~$ sudo systemctl restart ssh
ubuntu@receiver:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-04-25 18:19:46 UTC; 6s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 350 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 351 (sshd)
     Tasks: 1 (limit: 4551)
    CGroup: /docker/187fdee88b9d7fcf1ca301f0b5d5813a4cb96394cdb4d2d727a1daeca779e807/system.slice/
            └─351 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Apr 25 18:19:46 receiver systemd[1]: Starting OpenBSD Secure Shell server...
Apr 25 18:19:46 receiver sshd[351]: Server listening on 0.0.0.0 port 22.
Apr 25 18:19:46 receiver sshd[351]: Server listening on :: port 22.
Apr 25 18:19:46 receiver systemd[1]: Started OpenBSD Secure Shell server.

```

Hình 16: restart lại dịch vụ ssh-server

- Tại terminal sender, tiến hành gửi file audio chứa thông điệp đến người nhận.

`scp stego_audio.wav ubuntu@<ip_máy_receiver>:/home/ubuntu/`

Nhập password là *ubuntu*

```

ubuntu@sender:~$ scp stego_audio.wav ubuntu@172.20.0.11:/home/ubuntu/
The authenticity of host '172.20.0.11 (172.20.0.11)' 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 '172.20.0.11' (ECDSA) to the list of known hosts.
ubuntu@172.20.0.11's password:
stego_audio.wav                                100% 431KB 18.5MB/s 00:00

```

Hình 17: tiến hành gửi file audio chứa thông điệp đến người nhận

- Sau khi terminal receiver nhận được file audio, tiến hành tách thông điệp với cấu trúc lệnh `python3 extract_message.py`

```

ubuntu@receiver:~$ ls
extract_message.py  stego_audio.wav
ubuntu@receiver:~$ python3 extract_message.py
Thông điệp tách ra: HELLO
Đã tạo file extracted_message.txt
ubuntu@receiver:~$ cat extracted_message.txt
HELLOubuntu@receiver:~$

```

Hình 18: tiến hành tách thông điệp

- So sánh thông điệp được tách ra với nội dung file gửi xem có trùng khớp không
Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

stoplast

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@LabtainerVMware:~/labtainer/labtainer-student$ checkwork steg-basic-fhss-extract
Results stored in directory: /home/student/labtainer_xfer/steg-basic-fhss-extract
Successfully copied 137kB to steg-basic-fhss-extract-igrader:/home/instructor/B21DCAT105.steg-basic-fhss-extract.lab
Successfully copied 2.05kB to /home/student/labtainer_xfer/steg-basic-fhss-extract
Labname steg-basic-fhss-extract
```

Student	enable_ssh	transfer_file	check_text	extract_message	ssh_rootlogin
=====	=====	=====	=====	=====	=====
B21DCAT105	Y	Y	Y	Y	Y

What is automatically assessed for this lab:

Hình 19: Kết quả checkwork

Khởi động lại bài lab:

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

labtainer -r steg-basic-fhss-extract

TÀI LIỆU THAM KHẢO

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