

# Hướng Dẫn Lab Echo Hiding 1k decode

## Giới Thiệu

Lab Echo Hiding 1k decode là một bài tập về giấu tin (steganography), trong đó bạn sẽ học cách giấu và khôi phục thông điệp bí mật trong file âm thanh bằng kỹ thuật echo 1 nhân. Trong lab này, bạn sẽ sử dụng script Python để giải mã một thông điệp ẩn trong file âm thanh và kiểm tra kết quả. Lab được thực hiện trong môi trường Labtainer, một nền tảng dựa trên Docker dành cho các bài tập an ninh mạng.

Hướng dẫn này cung cấp các bước chi tiết để khởi động lab, chạy script, giải mã thông điệp, và xác minh kết quả.

## Điều Kiện Tiên Quyết

- **Môi Trường Labtainer:** Cần cài đặt Labtainer trên hệ thống (Linux, macOS, hoặc Windows với Docker). Xem hướng dẫn cài đặt tại [Labtainers GitHub](#) nếu chưa cài.
- **Lệnh Linux Cơ Bản:** Hiểu cách sử dụng các lệnh như `nano`, `cat`, `diff`, và `python3`.
- **Python 3:** Lab sử dụng Python 3 để chạy script giải mã. Labtainer thường đã cài Python, nhưng hãy kiểm tra bằng lệnh `python3 --version`.
- **Truy Cập Terminal:** Sử dụng terminal để tương tác với môi trường Labtainer.

## Hướng Dẫn Từng Bước

### Bước 1: Khởi Động Lab

1. **Chạy Môi Trường Labtainer:**
  - Mở terminal và chạy lệnh sau để khởi động lab Echo Hiding 1k:  
`labtainer echo_hiding_1k_decode`
  - Lệnh này tạo một container Docker chứa môi trường lab với các file và công cụ cần thiết.
  - Đợi container khởi tạo. Bạn sẽ được đưa vào terminal bên trong container.
2. **Kiểm Tra Thiết Lập Lab:**
  - Kiểm tra thư mục hiện tại bằng lệnh `pwd`. Bạn nên ở trong thư mục chứa các file lab (ví dụ: `decode.py`, `encode.py`, `message.txt`).
  - Liệt kê file bằng `ls` để xác nhận có các file như `decode.py`, `encode.py`, và các file liên quan khác.

### Bước 2: Kiểm Tra Script Giải Mã

1. **Mở Script Giải Mã:**
  - Sử dụng trình chỉnh sửa `nano` để xem script `decode.py`:  
`nano decode.py`

- Script này được thiết kế để trích xuất thông điệp ẩn từ file âm thanh bằng kỹ thuật echo. Xem qua code để hiểu logic (ví dụ: cách nó xử lý mẫu âm thanh để phát hiện echo).
  - Thoát nano bằng cách nhấn `Ctrl+X`.
- 2. Hiểu Script:**
- Script `decode.py` phân tích file âm thanh (thường là file WAV) để tìm các echo nhỏ được thêm vào trong quá trình mã hóa. Các echo này đại diện cho dữ liệu nhị phân (0 và 1) tạo thành thông điệp ẩn.
  - Bạn không cần chỉnh sửa script trong lab này, nhưng việc hiểu cách nó hoạt động sẽ hữu ích.

## Bước 3: Giải Mã Thông Điệp Ẩn

- 1. Chạy Script Giải Mã:**
- Chạy script `decode.py` để trích xuất thông điệp ẩn:
  - `python3 decode.py`
  - Lệnh này xử lý file âm thanh (ví dụ: file WAV chứa thông điệp ẩn) và lưu thông điệp giải mã vào file `decode.txt`.
  - Lưu ý: Tài liệu gốc đề cập đến chạy `echo_encode.py`, nhưng đây có thể là lỗi. Nếu lab yêu cầu chạy `encode.py` để mã hóa trước, hãy kiểm tra hướng dẫn hoặc chú thích trong file.
- 2. Xem Thông Điệp Giải Mã:**
- Hiển thị nội dung file giải mã:
  - `cat decode.txt`
  - File này chứa thông điệp ẩn đã được trích xuất. Kiểm tra xem nội dung có dễ đọc không (ví dụ: văn bản như một câu hoặc flag).
- 3. So Sánh Với Thông Điệp Gốc:**
- So sánh thông điệp giải mã với thông điệp gốc (nếu có) để đảm bảo tính chính xác:
  - `diff message.txt decode.txt`
  - Lệnh `diff` hiển thị sự khác biệt giữa `message.txt` (thông điệp gốc) và `decode.txt` (thông điệp giải mã).
  - Nếu không có khác biệt, quá trình giải mã thành công. Nếu có khác biệt, kiểm tra lại quá trình hoặc lỗi trong script.

## Bước 4: Kiểm Tra Kết Quả

- 1. Xác Minh Công Việc:**
- Chạy lệnh kiểm tra của Labtainer để đánh giá kết quả:
  - `checkwork echo_hiding_1k_decode`
  - Lệnh này kiểm tra tiến độ lab, xác nhận xem thông điệp giải mã có khớp với kết quả mong đợi hay không.
  - Xem kết quả để biết phản hồi (ví dụ: “Lab hoàn thành thành công” hoặc lỗi chỉ ra vấn đề).

## Bước 5: Khởi Động Lại Lab (Tùy Chọn)

### 1. Đặt Lại Môi Trường Lab:

- Nếu cần khởi động lại lab (ví dụ: để thử lại hoặc xóa công việc trước), sử dụng:
- `labtainer -r echo_hiding_1k_decode`
- Lệnh này đặt lại container lab về trạng thái ban đầu, xóa mọi thay đổi.
- Bắt đầu lại từ Bước 1 để lặp lại lab.

## Giải Thích

- **Kỹ Thuật Echo Hiding:** Phương pháp giấu tin này nhúng dữ liệu vào âm thanh bằng cách thêm các echo nhỏ với độ trễ cụ thể. Ví dụ, độ trễ ngắn có thể đại diện cho bit 1, và độ trễ dài hơn đại diện cho bit 0. Script giải mã phát hiện các độ trễ này để khôi phục thông điệp.
- **Labtainer:** Một nền tảng cho các bài tập an ninh mạng, cung cấp môi trường cách ly với công cụ và file được cấu hình sẵn. Nó tự động hóa thiết lập và chấm điểm, phù hợp cho mục đích giáo dục.
- **Script Python:**
  - `decode.py`: Phân tích file âm thanh để trích xuất thông điệp ẩn dựa trên mẫu echo.
  - `encode.py` (nếu sử dụng): Nhúng thông điệp vào file âm thanh bằng cách thêm echo.
- **So Sánh File:** Lệnh `diff` đảm bảo thông điệp giải mã khớp với thông điệp gốc, xác nhận quá trình giấu tin hoạt động chính xác.