

Hướng Dẫn Lab Echo Watermark

Giới Thiệu

Lab Echo Watermark là một bài tập về giấu tin (steganography), trong đó bạn sẽ học cách nhúng một thông điệp bí mật vào file âm thanh bằng kỹ thuật echo để tạo watermark. Watermark này cho phép ẩn thông tin (ví dụ: một chuỗi văn bản) mà không làm thay đổi đáng kể chất lượng âm thanh. Trong lab này, bạn sẽ sử dụng các script Python để tiền xử lý âm thanh, thêm echo, lưu file đã gắn watermark, 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, xử lý file âm thanh, nhúng thông điệp, kiểm tra file âm thanh, 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`, `echo`, `sox`, và `python3`.
- **Python 3:** Lab sử dụng Python 3 để chạy các script xử lý âm thanh. Labtainer thường đã cài Python, nhưng hãy kiểm tra bằng lệnh `python3 --version`.
- **Công Cụ SoX:** Công cụ SoX (Sound eXchange) được sử dụng để kiểm tra và so sánh file âm thanh. Đảm bảo SoX có sẵn trong môi trường Labtainer (kiểm tra bằng `sox --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 Watermark:
`labtainer echo_watermark`
 - 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: `Tien_xu_ly.py`, `Add_echo.py`, `Save_audio.py`, `input.wav`).
 - Liệt kê file bằng `ls` để xác nhận có các file như `Tien_xu_ly.py`, `Add_echo.py`, `Save_audio.py`, `input.wav`, và các file liên quan khác.

Bước 2: Tiền Xử Lý File Âm Thanh

1. Chạy Script Tiền Xử Lý:

- Chạy script `Tien_xu_ly.py` để chuẩn bị file âm thanh đầu vào:
- `python3 Tien_xu_ly.py`
- Script này có thể thực hiện các tác vụ như chuẩn hóa âm thanh, kiểm tra định dạng file (ví dụ: WAV), hoặc tạo các điều kiện cần thiết để thêm watermark.
- Kiểm tra output của script (nếu có) để đảm bảo không có lỗi.

2. Kiểm Tra File Đầu Vào:

- Đảm bảo file âm thanh đầu vào (ví dụ: `input.wav`) tồn tại trong thư mục hiện tại:
- `ls input.wav`
- Nếu file không tồn tại, kiểm tra lại môi trường lab hoặc liên hệ người hướng dẫn.

Bước 3: Giấu Tin Vào Âm Thanh

1. Tạo File Thông Điệp:

- Tạo một file văn bản tên `secret.txt` chứa thông điệp cần giấu (ví dụ: từ "secret"):
- `echo "secret" > secret.txt`
- Lệnh này ghi nội dung "secret" vào file `secret.txt`. Kiểm tra nội dung file bằng:
- `cat secret.txt`

2. Thêm Echo để Đánh Dấu:

- Chạy script `Add_echo.py` để nhúng thông điệp từ `secret.txt` vào file âm thanh bằng cách thêm echo:
- `python3 Add_echo.py`
- Script này đọc file `secret.txt`, chuyển thông điệp thành dữ liệu nhị phân, và thêm echo với độ trễ cụ thể vào file âm thanh (ví dụ: `input.wav`) để tạo watermark.

3. Lưu File Đã Gắn Watermark:

- Chạy script `Save_audio.py` để lưu file âm thanh đã gắn watermark:
- `python3 Save_audio.py`
- Script này tạo file âm thanh đầu ra (ví dụ: `watermarked.wav`) chứa thông điệp ẩn. Kiểm tra file đầu ra bằng:
- `ls watermarked.wav`

Bước 4: Kiểm Tra File Âm Thanh

1. Kiểm Tra Thống Kê File Âm Thanh:

- Sử dụng công cụ SoX để kiểm tra thống kê của file âm thanh gốc và file đã gắn watermark:
- `sox input.wav -n stat`
- `sox watermarked.wav -n stat`
- Lệnh `sox ... -n stat` hiển thị các thông số âm thanh như độ lớn, tần số, v.v. So sánh kết quả để xác nhận rằng file đã gắn watermark vẫn giữ chất lượng âm thanh gần giống file gốc, nhưng chứa thông điệp ẩn.
- Lưu ý: Sự khác biệt giữa hai file thường rất nhỏ, vì kỹ thuật echo chỉ thêm các thay đổi tinh vi.

2. So Sánh Hai File Âm Thanh:

- Tạo một file âm thanh so sánh bằng cách kết hợp file gốc và file đã gắn watermark:
- `sox --combine mix-power input.wav watermarked.wav comparison.wav`
- Lệnh này tạo file `comparison.wav` biểu thị sự khác biệt giữa hai file âm thanh.
- Kiểm tra thống kê của file so sánh:
- `sox comparison.wav -n stat`
- Kết quả giúp xác nhận rằng watermark đã được thêm mà không làm thay đổi đáng kể âm thanh gốc.

Bước 5: Kiểm Tra Kết Quả

1. Xác Minh Công Việc:

- Chạy lệnh kiểm tra của Labtainer để đánh giá kết quả:
- `checkwork echo_watermark`
- Lệnh này kiểm tra tiến độ lab, xác nhận xem file âm thanh đã gắn watermark có chứa thông điệp đúng hay không.
- Xem kết quả để biết phản hồi (ví dụ: “Lab hoàn thành thành công” hoặc lỗi chỉ ra vấn đề).

Bước 6: 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_watermark`
- 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 Watermark:** Phương pháp này nhúng thông điệp vào âm thanh bằng cách thêm các echo nhỏ với độ trễ cụ thể, tạo ra một watermark ẩn. Echo được thiết kế để không ảnh hưởng đáng kể đến chất lượng âm thanh, nhưng có thể được phát hiện bằng các công cụ giải mã.
- **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:**
 - `Tien_xu_ly.py`: Chuẩn bị file âm thanh đầu vào (ví dụ: chuẩn hóa, kiểm tra định dạng).
 - `Add_echo.py`: Nhúng thông điệp từ `secret.txt` vào file âm thanh bằng cách thêm echo.
 - `Save_audio.py`: Lưu file âm thanh đã gắn watermark (ví dụ: `watermarked.wav`).
- **Công Cụ SoX:** SoX được sử dụng để phân tích và so sánh đặc tính âm thanh, giúp xác nhận rằng watermark đã được thêm thành công mà không làm thay đổi đáng kể file gốc.