

Hướng Dẫn Lab Echo Hiding 1k - Mã Hóa

Giới Thiệu

Lab Echo Hiding 1k encode 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 1 nhân. Trong phần này, bạn sẽ sử dụng script Python để mã hóa thông điệp vào 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, tạo thông điệp, chạy script mã hóa, 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 script mã hóa. 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 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 Hiding 1k (phần mã hóa):
`labtainer echo_hiding_1k_encode`
 - 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: `echo_encode.py`, `input.wav`).
 - Liệt kê file bằng `ls` để xác nhận có các file như `echo_integr.py`, `input.wav`, và các file liên quan khác.

Bước 2: Kiểm Tra Script Mã Hóa

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

- `nano echo_encode.py`
 - Script này được thiết kế để nhúng một thông điệp vào file âm thanh bằng cách thêm các echo nhỏ. Xem qua code để hiểu logic (ví dụ: cách nó thêm echo với độ trễ khác nhau để biểu diễn dữ liệu nhị phân).
 - Thoát `nano` bằng cách nhấn `Ctrl+X`.
- 2. Hiểu Script:**
- Script `echo_encode.py` lấy một thông điệp từ file văn bản (ví dụ: `secret.txt`) và nhúng nó vào file âm thanh (ví dụ: `input.wav`) bằng cách thêm echo. Kết quả là một file âm thanh mới (ví dụ: `encoded_audio_file.wav`) chứa thông điệp ẩn.
 - Bạn không cần chỉnh sửa script trong lab này, nhưng việc hiểu cách nó hoạt động sẽ hữu ích.

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`. Bạn có thể kiểm tra nội dung file bằng:
 - `cat secret.txt`
- 2. Chạy Script Mã Hóa:**
- Chạy script `echo_encode.py` để nhúng thông điệp từ `secret.txt` vào file âm thanh:
 - `python3 echo_encode.py`
 - Script này xử lý file âm thanh đầu vào (ví dụ: `input.wav`), thêm echo để mã hóa thông điệp, và tạo file âm thanh đầu ra (ví dụ: `encoded_audio_file.wav`).
- 3. Kiểm Tra File Âm Thanh:**
- Sử dụng công cụ SoX để kiểm tra thông kê của file âm thanh gốc và file đã mã hóa, nhằm phát hiện sự khác biệt (nếu có):
 - `sox input.wav -n stat`
 - `sox encoded_audio_file.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 mã hóa 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.

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_encode`
 - Lệnh này kiểm tra tiến độ lab, xác nhận xem file âm thanh đã mã hóa 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 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_encode`
- Lệnh này đặt lại container lab về trạng thái ban đầu, xóa mọi thay đổi.
- Bắt đầu lại từ Bước 1 để lắp 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 mã hóa tạo các echo này để nhúng 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:**
 - `echo_integr.py`: Nhúng thông điệp từ file văn bản vào file âm thanh bằng cách thêm echo, tạo file âm thanh mới chứa thông điệp ẩn.
- **Công Cụ SoX:** SoX được sử dụng để phân tích đặc tính âm thanh, giúp xác nhận rằng file mã hóa không thay đổi đáng kể so với file gốc, nhưng vẫn chứa thông điệp ẩn.