

## Bài Thực Hành: Tấn công Giấu Tin Trong Video bằng Phương Pháp LSB (code)

### 1. Mục đích

Một tổ chức tội phạm đang sử dụng kỹ thuật LSB (Least Significant Bit) để truyền tin bí mật trong các file video. Cơ quan an ninh mạng giao bạn nhiệm vụ không chỉ phát hiện và trích xuất thông tin ẩn, mà còn chủ động vô hiệu hóa hoặc làm hỏng dữ liệu ẩn mà không làm hỏng nội dung chính của video. Mục tiêu là khiến dữ liệu ẩn không thể khôi phục được, đồng thời giữ cho video vẫn phát bình thường để tránh gây nghi ngờ.

### 2. Yêu cầu đối với sinh viên:

- Có kiến thức cơ bản về lập trình Python.
- Biết cách sử dụng terminal và các lệnh cơ bản trên hệ điều hành Linux.
- Hiểu khái niệm giấu tin và kỹ thuật LSB.

### 3. Nội dung thực hành

#### 3.1. Chuẩn bị lab

- Khởi động lab
- Chạy lệnh:

```
labtainer -r video-stego-attack-lsb-1
```

Tùy chọn -r đảm bảo môi trường được làm mới (reset) nếu đã chạy trước đó, cung cấp một môi trường sạch để thực hành.

(Chú ý: Sinh viên sử dụng <TÊN\_TÀI\_KHOẢN\_HỆ\_THỐNG> của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)

Môi trường lab được khởi động. Để minh họa nguyên tắc giấu tin trong video HEVC, lab sử dụng một video mẫu và các công cụ như FFmpeg và Python.


#### 3.2. task1: Phát Hiện Giấu Tin

Mục tiêu: Tách audio từ video và kiểm tra xem có thông tin ẩn bằng LSB không.

Tách audio từ video video\_with\_secret.mkv:

```
ffmpeg -i video_with_secret.mkv -vn -acodec pcm_s16le -ar 44100 -ac 1  
secret_audio.wav
```

 -vn: Loại bỏ video, chỉ lấy audio.

 `-acodec pcm_s16le -ar 44100 -ac 1`: Đảm bảo audio là PCM 16-bit, 44.1 kHz, mono.

Kiểm tra thông tin audio:

```
soxi secret_audio.wav
```

Chạy script shell để phát hiện giấu tin:

```
./check_flag.sh
```

Kết quả sẽ cho biết có thông tin ẩn trong audio không.

### **3.3. Task2: Trích Xuất Nội Dung Giấu Ban Đầu**

**Mục tiêu:** Trích xuất thông tin ẩn từ audio gốc để xác nhận nội dung ban đầu.

Giải mã audio để lấy thông tin ẩn:

```
python3 extract_lsb_audio.py secret_audio.wav > extracted_original.txt
```

Xem nội dung đã trích xuất:

```
cat extracted_original.txt
```

Kiểm tra kết quả:

```
bash check_flag.sh extracted_original.txt
```

### **3.4. Task3: Tấn Công Phá Hoại (Destruction Attack)**

**Mục tiêu:** Thêm tiếng ồn để phá hỏng thông tin ẩn mà không ảnh hưởng nhiều đến nội dung audio chính.

Lấy độ dài của `secret_audio.wav`:

```
soxi -D secret_audio.wav
```

- Giả sử kết quả là 6.037188 giây (thay bằng giá trị thực tế).

Tạo tệp tiếng ồn trắng:

```
sox -n -t wav whitenoise.wav synth 6.037188 whitenoise vol 0.001
```

Trộn tiếng ồn với audio gốc:

```
sox -m secret_audio.wav whitenoise.wav damaged.wav
```

Ghép audio bị phá hoại vào video:

```
ffmpeg -i video_with_secret.mkv -i damaged.wav -c:v copy -c:a aac -map 0:v:0 -map  
1:a:0 -shortest video_damaged.mp4
```

Kiểm tra kết quả tấn công:

```
./check_damage.sh damaged.wav
```

### 3.5. Task4: Tấn Công Nén Audio

**Mục tiêu:** Nén audio để phá hỏng thông tin ẩn bằng cách mất mát dữ liệu LSB.

Nén audio với bitrate thấp:

```
ffmpeg -i secret_audio.wav -b:a 16k compressed_audio.mp3
```

Chuyển MP3 về WAV để kiểm tra:

```
ffmpeg -i compressed_audio.mp3 -acodec pcm_s16le -ar 44100 -ac 1  
compressed_audio.wav
```

Ghép audio nén vào video:

```
ffmpeg -i video_with_secret.mkv -i compressed_audio.mp3 -c:v copy -c:a copy -map  
0:v:0 -map 1:a:0 -shortest video_compressed.mp4
```

(Tùy chọn) Kiểm tra khả năng khôi phục:

```
python3 extract_lsb_audio.py compressed_audio.wav
```

- Kết quả nên là chuỗi vô nghĩa.

### 3.65. Task5: Tấn Công Giả Mạo (Injection Attack)

**Mục tiêu:** Thay thế thông tin ẩn bằng nội dung giả mạo.

Tiêm thông điệp giả vào audio:

```
python3 inject_lsb_audio.py secret_audio.wav fake_flag.txt fake_audio.wav
```

Ghép audio giả mạo vào video:

```
ffmpeg -i video_with_secret.mkv -i fake_audio.wav -c:v copy -c:a copy -map 0:v:0 -map 1:a:0 -shortest video_fake.mkv
```

Tách audio để kiểm tra:

```
ffmpeg -i video_fake.mkv -vn -acodec pcm_s16le -ar 44100 -ac 1 fake_audio_extracted.wav
```

(Tùy chọn) Xác nhận nội dung giả:

```
python3 extract_lsb_audio.py fake_audio_extracted.wav
```

Kết quả nên khớp với fake\_flag.txt.

### 3.76. Task6: Kiểm Tra Khả Năng Khôi Phục

Mục tiêu: Xác nhận rằng thông tin ẩn trong audio bị phá hoại không thể khôi phục.

Giải mã audio bị phá hoại:

```
python3 extract_lsb_audio.py damaged.wav > output.txt
```

Xem kết quả:

```
cat output.txt
```

Kiểm tra:

```
bash check_damage.sh damaged.wav
```

o Đảm bảo thông tin gốc không còn khôi phục được.

}

#### 4. Kết quả cần đạt được

- Chạy được tất cả các bước như yêu cầu.
- Cần nộp 1 file: trong thư mục: /home/student/labtainer\_xfer/TÊN\_BÀI\_LAB (tên tài khoản.TÊN\_BÀI\_LAB.lab)
- Kết thúc bài lab:
  - o Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoplab video-stego-attack-lsblsb-1*

- Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.
- Sinh viên cần nộp file .lab để chấm điểm.
- Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh: *checkwork <tên bài thực hành>*
- Khởi động lại bài lab: Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:  
*labtainer -r video-stego-attack-lsblsb-1*