

Bài thực hành: Giấu tin trong văn bản bằng phương pháp sử dụng dấu câu một cách không rõ ràng

1. Mục đích

- Giúp sinh viên hiểu về phương pháp giấu tin trong văn bản bằng phương pháp sử dụng dấu câu một cách không rõ ràng.
- Thực hành viết chương trình Python để giấu tin trong văn bản bằng phương pháp sử dụng dấu câu một cách không rõ ràng.

2. Yêu cầu đối với sinh viên

- Sinh viên có kiến thức xử lý chuỗi, file, và bitwise trong Python

3. Nội dung thực hành

- Tải bài lab: Sinh viên tải bài lab từ link github sau:

<https://github.com/Mbr-clown-lord/stego-syntax-punctuation.git>

- Khởi động bài lab:

labtainer -r stego-syntax-punctuation

(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)

Ý tưởng của việc sử dụng các dấu chấm câu không rõ ràng trong giấu tin trong văn bản khá đơn giản nhưng dễ bị phát hiện bởi vì việc sử dụng những dấu chấm câu không nhất quán sẽ gây ra sự chú ý. Một ví dụ về ý tưởng sử dụng dấu chấm câu nhập nhằng là việc sử dụng cụm từ “Bánh mì, bơ, và sữa” và sử dụng cụm từ “Bánh mì, bơ và sữa.”. Cả hai cụm từ trên đều đúng về mặt cú pháp, thay đổi luân phiên giữa hai hình thức có thể ẩn giấu dữ liệu nhị phân và có thể quy ước thành cứ mỗi lần xuất hiện cấu trúc đầu tiên (dấu “,” xuất hiện ở trước chữ “và”) sẽ ẩn giấu bit “1”, khi xuất hiện cấu trúc thứ hai (không có dấu “,” trước chữ “và”) sẽ ẩn dấu bit “0”.

Trong bài lab này chúng ta sẽ sử dụng văn bản tiếng Anh làm vật chứa tin. Quy ước cứ mỗi lần xuất hiện cấu trúc đầu tiên (dấu “,” xuất hiện ở trước chữ “and”) sẽ ẩn giấu bit “1”, khi xuất hiện cấu trúc thứ hai (không có dấu “,” trước chữ “and”) sẽ ẩn dấu bit “0”. Với quy ước trên hãy sử dụng ngôn ngữ Python để viết mã và hoàn thành các nhiệm vụ dưới đây.

- Sau khi khởi động xong một terminal ảo có tên stego-syntax-punctuation xuất hiện.
- Khám phá: dùng lệnh ls để kiểm tra tệp message.txt(chứa thông điệp cần giấu) và tệp cover_text.txt(tệp văn bản chứa tin)

Xem nội dung tệp message.txt:

```
cat message.txt
```

Xem nội dung tệp cover_text.txt:

```
cat cover_text.txt
```

- 3.1. Nhiệm vụ 1: Chuyển đổi thông điệp bí mật sang dạng nhị phân
 - Yêu cầu: Viết mã Python đặt trong file text_to_binary.py đọc thông điệp bí mật từ tệp message.txt chuyển sang dạng nhị phân sau đó in kết quả ra màn hình
Định dạng đầu ra: Ví dụ: “AB” → “01000001 01000010”
- 3.2. Nhiệm vụ 2: Giấu thông điệp
 - Yêu cầu: Viết mã Python đặt trong file hide_message.py để giấu các bit nhị phân của thông điệp vừa thu được ở nhiệm vụ trước in kết quả ra màn hình đồng thời lưu kết quả vào file stego_text.txt. Dùng lệnh sau để thấy được sự khác nhau giữa văn bản trước và sau khi giấu tin:

```
diff cover_text.txt stego_text.txt | tee diff.txt
```
- 3.3. Nhiệm vụ 3: Tách thông điệp bí mật dạng nhị phân từ văn bản sau khi giấu
 - Yêu cầu: Viết mã Python đặt trong file extract_binary.py để tách các bit thông điệp bí mật vừa giấu trong tệp stego_text.txt
Định dạng đầu ra: Nếu thông điệp ban đầu là “AB” → “0100000101000010”
- 3.4. Nhiệm vụ 4: Chuyển thông điệp bí mật dạng nhị phân về thông điệp ban đầu
 - Yêu cầu: Viết mã Python đặt trong file binary_to_text.py để chuyển các bit thông điệp bí mật vừa tách được về thông điệp bí mật ban đầu
- Kết thúc bài lab:
 - o Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

```
stoplab stego-syntax-punctuation
```
 - o 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:

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

```
labtainer -r stego-syntax-punctuation
```