ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

-----□&□-----

BÁO CÁO ĐỒ ÁN

File fingerprinting of the ZIP format for identifying and tracking provenance

GVHD: Lê Đức Thịnh

Thành viên nhóm

Lâm Hải Đăng – 21520682

Trương Long Hưng – 21520903

Nguyễn Văn Anh Tú – 21520514

Nguyễn Đình Kha - 21520948

Mục lục

Danh sách hình ảnh

I. GIỚI THIỆU

1. Giới thiệu về bài báo

Bài báo "File Fingerprinting of the ZIP Format for Identifying and Tracking Provenance" của các tác giả Minji Um, Jaehyeok Han, và Sangjin Lee tập trung vào phân tích pháp chứng kỹ thuật số của tệp ZIP nhằm xác định nguồn gốc và theo dõi dấu vết tệp. Nghiên cứu này đề xuất rằng cấu trúc tệp ZIP, dù tuân theo tiêu chuẩn chung, vẫn tồn tại những khác biệt chi tiết phụ thuộc vào hệ điều hành và ứng dụng tạo tệp. Những khác biệt này tạo ra dấu vân tay tệp (file fingerprint) độc nhất, giúp các nhà pháp chứng xác định môi trường tạo tệp, hành vi người dùng và phát hiện khả năng chỉnh sửa trái phép.

Nghiên cứu đã thực hiện các thí nghiệm trên nhiều hệ điều hành như Windows, macOS, và Ubuntu với các công cụ nén phổ biến như WinRAR, 7-zip, WinZip, và Compress. Thông qua phân tích các trường dữ liệu bổ sung (Extra Field), các phương pháp mã hóa, và cấu trúc tiêu đề (Header), bài báo đã chứng minh cách phân tích pháp chứng tệp ZIP có thể giúp xây dựng quy trình tự động để xác định nguồn gốc tệp và phát hiện sự thay đổi trong cấu trúc tệp

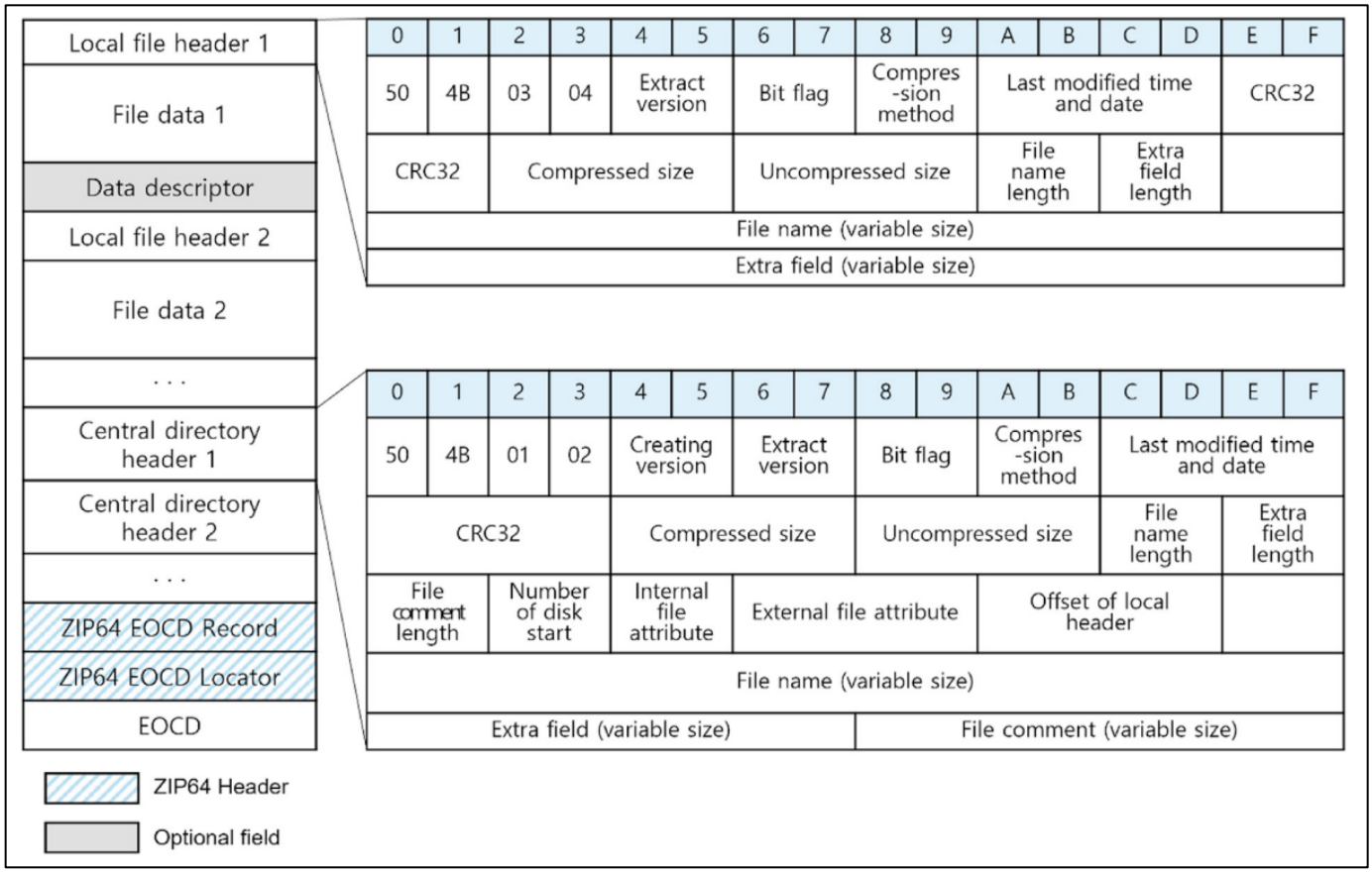
2. Giới thiệu chung

Trong kỷ nguyên số hóa, việc lưu trữ và truyền tải dữ liệu một cách an toàn và hiệu quả là một thách thức quan trọng. Định dạng tệp ZIP, với khả năng nén dữ liệu vượt trội và tính linh hoạt cao, đã trở thành công cụ phổ biến cho nhiều mục đích sử dụng, từ lưu trữ dữ liệu cá nhân đến quản lý dữ liệu doanh nghiệp. với định dạng có cấu trúc bao gồm nhiều tiêu đề (header) và trường siêu dữ liệu (metadata). Mặc dù các cấu trúc này được tiêu chuẩn hóa, chúng có thể khác biệt đáng kể tùy thuộc vào hệ thống và phần mềm được sử dụng để nén. Sự khác biệt này tạo cơ sở cho lĩnh vực pháp chứng kỹ thuật số nhằm truy vết nguồn gốc của tệp, xác định hành vi người dùng và phân tích khả năng bị chỉnh sửa.

Phân tích pháp chứng tệp ZIP dựa trên ý tưởng rằng các cấu trúc bên trong, bao gồm **Local File Header**, **Central Directory Header** và **End of Central Directory (EOCD)**, có thể cung cấp các manh mối pháp chứng quan trọng. Các nghiên cứu trước đây đã chỉ ra rằng siêu dữ liệu của tài liệu (ví dụ: **RSID** của MS Office) và cấu trúc tệp đa phương tiện có thể tiết lộ nguồn gốc tệp và lịch sử chỉnh sửa. Phân tích pháp chứng tệp ZIP mở rộng các khái niệm này bằng cách khám phá cách các khác biệt về cấu trúc xuất hiện dựa trên môi trường hệ điều hành và công cụ nén.

Bài nghiên cứu này không chỉ có ý nghĩa thực tiễn trong lĩnh vực pháp chứng kỹ thuật số mà còn đóng góp vào việc xây dựng các công cụ tự động để hỗ trợ các chuyên gia trong việc xử lý dữ liệu pháp chứng. Với tầm quan trọng của tệp ZIP trong lưu trữ và chia sẻ dữ liệu, phân tích pháp chứng chi tiết loại tệp này hứa hẹn mở ra nhiều hướng nghiên cứu mới, đặc biệt là trong việc chống lại các kỹ thuật giả mạo dữ liệu ngày càng tinh vi.

II. CẤU TRÚC CỦA FILE ZIP



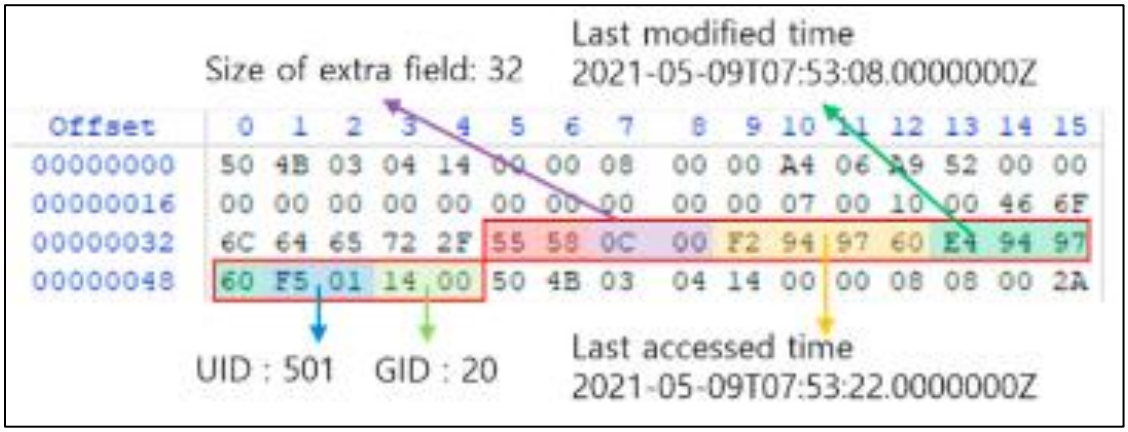
Hình 1. Cấu trúc tổng thể của một file ZIP

Một file ZIP có chữ kí bắt đầu (‘0x504B0304’) và cấu trúc tổng thể như hinh 1, gồm:

- **Local File Header**: Lưu trữ thông tin về từng tệp, bao gồm
  - **CRC32 checksum**: Được sử dụng để kiểm tra tính toàn vẹn của từng tệp
  - **Compression method**: Chỉ ra cách dữ liệu tệp được nén (ví dụ: lưu trữ, deflated).
  - **File name and modification time**: Siêu dữ liệu cơ bản cho từng mục tệp.
- **Central Directory Header**: Một thư mục ở cuối tệp ZIP liệt kê tất cả các tệp trong kho lưu trữ bao gôm:
  - **Metadata references**: Con trỏ đến các Local File Headers và thông tin bổ sung như thuộc tính tệp và trường chú thích.
  - **Offset positions**: Xác định vị trí bắt đầu của Local File Header, giúp truy cập nội dung tệp nhanh chóng.
- **EOCD (End of Central Directory)**: Đánh dấu kết thúc của kho lưu trữ và chứa:
  - **Total number of entries**: Chỉ ra số lượng tệp có trong ZIP.
  - **Location of the Central Directory**: Cần thiết để xác định vị trí bắt đầu của Tiêu đề Thư Mục Trung Tâm..
  - **Comments**: Ghi chú tùy chọn do người dùng thêm vào có thể được bao gồm trong tệp ZIP.

Cấu trúc tệp ZIP cũng hỗ trợ các thành phần tuy chọn:

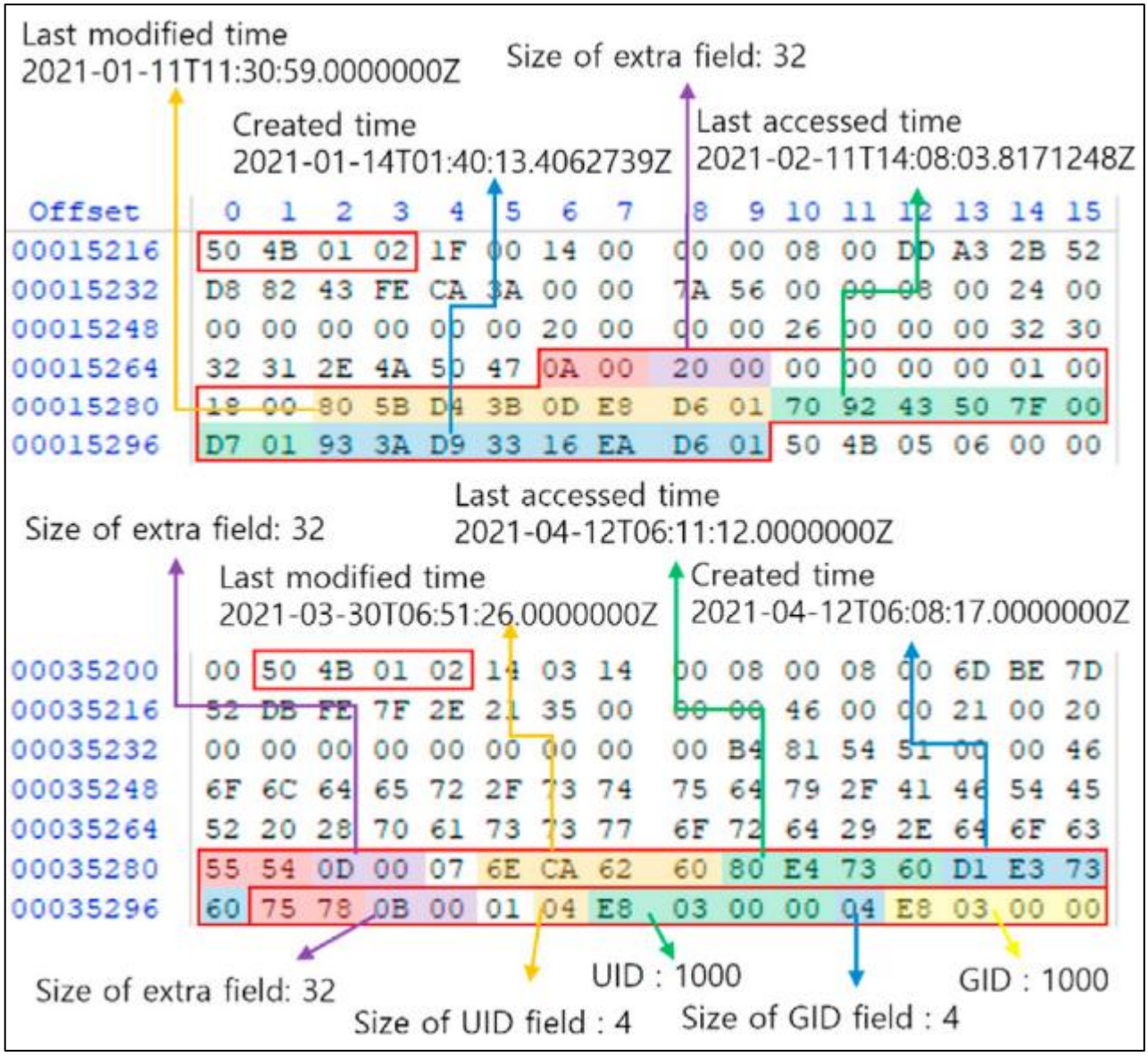
- **Extra Fields**: Dữ liệu bổ sung có thể được thêm vào các tiêu đề cho các mục đích đặc biệt. Các trường này có thể chứa dấu thời gian (ví dụ: dấu thời gian mở rộng cho các bản ghi thời gian chính xác cao), dữ liệu cụ thể cho hệ điều hành và các định danh khác.
- **Data Descriptor**: Một phần được sử dụng khi tệp được ghi ở chế độ truyền phát mà không có kiến thức trước về giá trị băm CRC32 hoặc kích thước, được đánh dấu bởi chữ ký '0x504B0708'.
- **ZIP64 Headers**: Được giới thiệu để khắc phục các hạn chế về kích thước của thông số kỹ thuật ZIP ban đầu, cho phép các tệp và kho lưu trữ lớn hơn.



Hình 2. Info-ZIP Unix extra field (0x5855)

Ngoài ra các Extra Field được sử dụng để lưu trữ thông tin bổ sung, chẳng hạn như:

- **Timestamps mở rộng (Extended timestamps):** Lưu trữ thời gian tạo, sửa đổi và truy cập tệp.
- **UID và GID:** Xác định người dùng và nhó người dùng trên các hệ điều hành Unix-based.
- **Language Encoding Flag (EFS):** Quy định phương pháp mã hóa tên tệp (ví dụ: UTF-8)



Hình 3. NTFS (0x000A), Extended timestamp (0x5455), UNIX UID/GID (0x7875)

III. NHẬN DẠNG TỆP ZIP ĐỂ TRUY RA NGUỒN GỐC

Để theo dõi hiêu quả nguồn gốc của tệp ZIP, các nhà phân tích pháp chứng phải kiểm tra các thuộc tính tệp cụ thể và các đặc điểm cấu trúc. Điều này bao gồm việc phân tích siêu dữ liệu được nhúng trong tệp ZIP để khám phá thông tin chi tiết về môi trường tạo tệp và bất kỳ sửa đổi nào mà tệp đã trải qua. Các yếu tố chính trong việc xác định nguồn gốc bao gồm:

1. Extra Fields

Các trường bổ sung đóng vai trò quan trọng trong việc xác định nguồn gốc của một tệp ZIP. Những trường có độ dài biến đổi này có thể lưu trữ thông tin chi tiết vượt ra ngoài dữ liệu tiêu đề cơ bản, cung cấp cái nhìn sâu sắc về công cụ nén và hệ thống đã sử dụng:

- **Extended Timestamps**: Những trường bổ sung này, chẳng hạn như ID Header '0x000A' (NTFS) hoặc '0x5455' (dấu thời gian mở rộng Unix), lưu trữ các giá trị thời gian chính xác cao, bao gồm thời gian tạo, sửa đổi và truy cập. Sự hiện diện và cấu trúc của những dấu thời gian này có thể chỉ ra hệ điều hành và phiên bản ứng dụng đã được sử dụng..
- **OS-Specific Data**: Nội dung của các trường bổ sung có thể bao gồm các định danh độc nhất thay đổi giữa các hệ điều hành. Ví dụ, các trường bổ sung cụ thể cho Windows thường bao gồm dấu thời gian NTFS được định dạng dưới dạng FILETIME 64-bit, trong khi các hệ thống dựa trên Unix sử dụng định dạng thời gian Unix 32-bit.
- **Header Order and Composition**: Thứ tự mà các trường bổ sung này xuất hiện cũng có thể cung cấp manh mối về ứng dụng. Ví dụ, một số ứng dụng có thể liệt kê ID tiêu đề theo một thứ tự cụ thể, có thể so khớp với các chữ ký công cụ đã biết.

2. UID and GID Information

Trên các hệ thống dựa trên Unix như macOS và Linux, các tệp ZIP có thể chứa dữ liệu UID (Định danh Người Dùng) và GID (Định danh Nhóm) trong các trường bổ sung (ví dụ: ID tiêu đề '0x7875'). Thông tin này liên quan đến tài khoản người dùng đã tạo hoặc sửa đổi tệp ZIP:

- **User and Group Identifiers**: UID và GID phản ánh các ngữ cảnh người dùng và nhóm cụ thể dưới đó tệp được tạo ra, giúp liên kết tệp ZIP với một tài khoản hoặc môi trường hệ thống cụ thể.
- **Tracking User Activity**: Bằng cách so sánh dữ liệu UID và GID với hồ sơ hệ thống hoặc cấu hình người dùng đã biết, các nhà phân tích pháp chứng có thể thiết lập một kết nối rõ ràng hơn giữa một tệp ZIP và hệ thống nguồn gốc của nó.

3. Language Encoding Flags (EFS)

Cờ mã hóa ngôn ngữ cung cấp cái nhìn về cài đặt vùng và ngôn ngữ của hệ thống tại thời điểm tạo tệp:

- **Unicode Path Extra Field (‘0x7075’)**: Chỉ ra rằng tên tệp được lưu trữ ở định dạng Unicode, điều này có thể tiết lộ liệu các gói ngôn ngữ cụ thể có đang hoạt động trên hệ thống trong quá trình nén hay không.

- **Encoding Variations**: Các hệ thống khác nhau có thể sử dụng các phương pháp mã hóa khác nhau (ví dụ: UTF-8, CP949 cho tiếng Hàn). Phân tích phương pháp mã hóa giúp thu hẹp hệ điều hành và cấu hình của hệ thống được sử dụng để nén.
- **System Locale Clues**: Sự hiện diện của các cờ mã hóa, kết hợp với phân tích mã hóa tên tệp, có thể phân biệt giữa các phiên bản khu vực của hệ điều hành (ví dụ: một hệ thống Windows tiếng Anh so với phiên bản tiếng Hàn)

4. Ứng dung sử dụng để nén tệp

- Mỗi ứng dụng nén (như WinRAR, 7-zip, hoặc Bandizip) tao header cho cả thư mục gốc và thư mục con.
- **Windows Compressed Folder:** Không tạo Header cho thư mục gốc nhưng lưu thông tin trong đường dẫn của tệp con.
- **Double Zipping (Nén lặp):** Một số ứng dụng, như WinRAR, không nén lại các tệp ZIP có sẵn, trong khi 7-zip hoặc Bandizip có thể nén lại tùy thuộc vào tỷ lệ nén.

5. Trường hợp đặc biệt

- **macOS:** Thư mục hệ thống như __MACOSX thường xuất hiện trong các tệp ZIP tạo từ macOS nhưng không có trong các tệp nén bằng Windows hoặc Linux.
- **Ubuntu:** Sử dụng các giá trị khác biệt trong Extra Field như thời gian chính xác đến nanosecond.

Việc phân tích cấu trúc tệp ZIP không chỉ xác định nguồn gốc tệp mà còn làm sáng tỏ hành vi của người dùng. Kết hợp các đặc điểm từ Header, Extra Field, và thời gian lưu trữ, pháp chứng kỹ thuật số có thể xây dựng quy trình tự động để truy vết nguồn gốc và phát hiện các chỉnh sửa tiềm ẩn. Điều này đặc biệt hữu ích trong các cuộc điều tra pháp chứng liên quan đến giả mạo dữ liệu hoặc tranh chấp thông tin.

IV. KỸ THUẬT PHÂN TÍCH PHÁP CHỨNG ĐỂ THEO DÕI NGUỒN GỐC

1. Phát triển bộ phân loại tự động

Đây là một kỹ thuật phân tích forensic để theo dõi nguồn gốc của các tập tin ZIP:

Phát triển một bộ phân loại tự động để xác định môi trường hệ thống mà tập tin ZIP được tạo ra, bằng cách kết hợp đặc điểm của hệ điều hành và ứng dụng thông qua phân tích cấu trúc của tập tin ZIP.

Bộ phân loại tự động sẽ kiểm tra các đặc điểm đáng tin cậy như sự tồn tại của mô tả dữ liệu, loại trường bổ sung, thông tin về hệ điều hành được sử dụng để tạo ra tập tin ZIP. Nếu không thể xác định được nguồn gốc, nó sẽ kiểm tra các đặc điểm khác như double zipping, sự tồn tại của thư mục '__MACOSX' và định dạng của phần nanosecond trong timestamp.

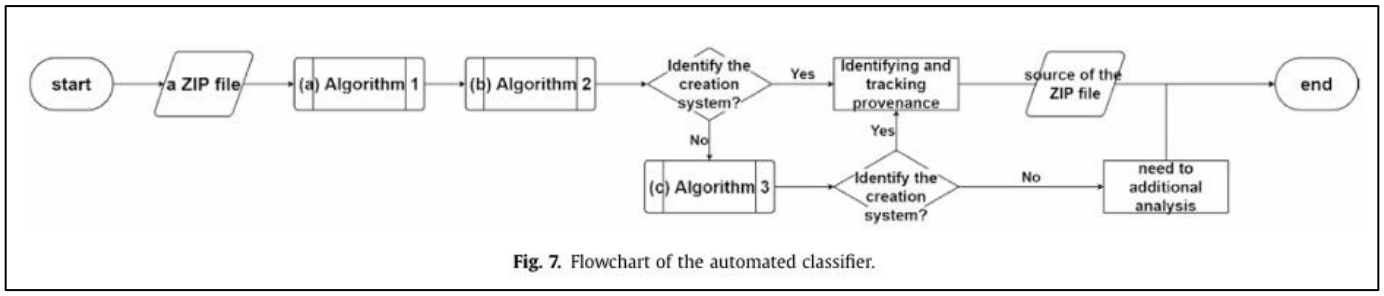


Fig. 7. Flowchart of the automated classifier.

Hình 4. Mô hình phân loại tự động

Một số đặc điểm phụ thuộc vào tập tin nén hơn là môi trường tạo ra, chẳng hạn như giá trị của trường bổ sung 0x000A phụ thuộc vào định dạng nanosecond của tập tin nén. Những đặc điểm này có độ tin cậy thấp hơn trong việc xác định nguồn gốc.

Nguồn gốc của tệp ZIP có thể được xác định bằng cách so sánh các đặc tính của tệp ZIP được lưu trữ trong mỗi PC với hệ điều hành và ứng dụng được cài đặt trong PC đó.

2. Thí nghiệm bổ sung về các thay đổi trong một tệp tin ZIP cụ thể

Các tệp tin ZIP được tạo ra từ một tệp tin hoặc thư mục được người dùng chọn, nhưng trong một số trường hợp, tệp tin được thêm vào một tệp tin ZIP đã có sẵn, hoặc chỉ một phần của một tệp tin ZIP hiện có được xóa hoặc chỉnh sửa. Thực tế, hầu hết các ứng dụng hỗ trợ việc thêm, xóa và chỉnh sửa các tệp tin ZIP hiện có mà không cần giải nén chúng

Phát hiện sửa đổi: Khi tệp ZIP được sửa đổi (thêm, xóa hoặc thay đổi), dấu vân tay tệp sẽ phản ánh các đặc điểm của ứng dụng được sử dụng để sửa đổi.

Nhiều dấu vân tay: Một tệp ZIP có thể chứa dấu vân tay từ nhiều hệ thống nếu tệp đó đã được sửa đổi trong các môi trường khác nhau.

3. Các khác biệt trong kết quả giải nén tệp ZIP

Biến động thông tin thời gian: Thông tin thời gian (thời gian sửa đổi, truy cập và tạo) của các tệp giải nén có thể khác nhau tùy thuộc vào ứng dụng giải nén được sử dụng. Ví dụ: WinRAR đặt thời gian tạo và sửa đổi thành thời gian giải nén, trong khi WinZip sử dụng thời gian được lưu trữ trong trường bổ sung.

Tác động của cấu trúc thư mục: Sự hiện diện của tiêu đề thư mục và cấu trúc thư mục con có thể ảnh hưởng đến thông tin thời gian của các tệp và thư mục được giải nén. Điều này có nghĩa là ngay cả khi cùng một tệp ZIP được giải nén, kết quả có thể khác nhau tùy thuộc vào cấu trúc thư mục trong tệp ZIP.

Hiện tượng đảo ngược thời gian: Có thể xảy ra sự đảo ngược giữa thời gian tạo và sửa đổi của các thư mục được giải nén, cũng như giữa thời gian của thư mục cha và thư mục con. Hiện tượng này làm nổi bật tầm quan trọng của việc phân tích cả tệp ZIP và các hiện vật hệ thống để hiểu chính xác lịch sử của tệp.

Sự khác biệt cụ thể của ứng dụng: Các ứng dụng giải nén khác nhau xử lý các trường bổ sung và cấu trúc bên trong khác nhau, dẫn đến sự khác biệt về thông tin thời gian và cấu trúc của các tệp được giải nén.

Phần này nhấn mạnh nhu cầu phân tích kỹ lưỡng cả tệp ZIP và môi trường hệ thống để giải thích chính xác siêu dữ liệu của tệp đã giải nén.

V. THỰC NGHIỆM

Môi trường: Máy ảo Ubuntu (chạy công cụ zipdetails), Windows (chạy FTK Imager)

Công cụ: zipdetails, FTK Imager

Dataset: https://github.com/pmqs/zipdetails/tree/main/t/files

1. Thực nghiệm 1 số đặc trưng khác nhau của 7zip trên windows và linux.

1.1. Thực nghiệm với file ZIP trên linux

```
danu@ubuntu:~/zipdetails/bin$ ./zipdetails /home/danu/zipdetails/t/files/0002-7z/linux/7z-linux-bzip2/7z-linux-bzip2.zip

0000 LOCAL HEADER #1       04034B50 (67324752)
0004 Extract Zip Spec      2E (46) '4.6'
0005 Extract OS            03 (3) 'Unix'
0006 General Purpose Flag  0000 (0)
0008 Compression Method    000C (12) 'BZIP2'
000A Modification Time     52899E15 (1384750613) 'Fri Apr  9 12:48:42 2021'
000E CRC                   F90EE7FF (4178503679)
0012 Compressed Size       00000134 (308)
0016 Uncompressed Size     000001BE (446)
001A Filename Length       0009 (9)
001C Extra Length          0000 (0)
001E Filename              'lorem.txt'
0027 PAYLOAD

015B CENTRAL HEADER #1     02014B50 (33639248)
015F Created Zip Spec      3F (63) '6.3'
0160 Created OS            03 (3) 'Unix'
0161 Extract Zip Spec      2E (46) '4.6'
0162 Extract OS            03 (3) 'Unix'
0163 General Purpose Flag  0000 (0)
0165 Compression Method    000C (12) 'BZIP2'
0167 Modification Time     52899E15 (1384750613) 'Fri Apr  9 12:48:42 2021'
016B CRC                   F90EE7FF (4178503679)
016F Compressed Size       00000134 (308)
0173 Uncompressed Size     000001BE (446)
0177 Filename Length       0009 (9)
0179 Extra Length          0024 (36)
017B Comment Length        0000 (0)
017D Disk Start            0000 (0)
017F Int File Attributes   0000 (0)
     [Bit 0]               0 'Binary Data'
0181 Ext File Attributes   81ED8020 (2179825696)
     [Bit 5]               Archive
     [Bit 15]              Possible p7zip/7z Unix Flag
     [Bits 16-24]          01ED (493) 'Unix attrib: rwxr-xr-x'
     [Bits 28-31]          08 (8) 'Regular File'
0185 Local Header Offset   00000000 (0)
0189 Filename              'lorem.txt'
0192 Extra ID #1           000A (10) 'NTFS FileTimes'
0194   Length              0020 (32)
0196   Reserved            00000000 (0)
019A   Tag1                0001 (1)
019C   Size1               0018 (24)
019E   Mtime               01D72D70F58D8A80 (132624677210000000) 'Fri Apr  9 11:48:41 2021 0ns'
01A6   Atime               01D72D710938F100 (132624677540000000) 'Fri Apr  9 11:49:14 2021 0ns'
01AE   Ctime               01D72D70F58D8A80 (132624677210000000) 'Fri Apr  9 11:48:41 2021 0ns'

01B6 END CENTRAL HEADER    06054B50 (101010256)
01BA Number of this disk   0000 (0)
01BC Central Dir Disk no   0000 (0)
01BE Entries in this disk  0001 (1)
01C0 Total Entries         0001 (1)
01C2 Size of Central Dir   0000005B (91)
01C6 Offset to Central Dir 0000015B (347)
01CA Comment Length        0000 (0)
#
# Done
```

Hình 5. Điều tra bằng zipdetails với file được zip bằng 7z trên Linux

Phần Local Header:

- **Compression Method**: BZIP2 (phương pháp nén được sử dụng).
- **Modification Time**: Fri Apr 9 12:48:42 2021 (thời điểm file lorem.txt được chỉnh sửa).
- **Filename Length**: 9 bytes, phù hợp với tên file lorem.txt.
- Đây là những đặc điểm giúp xác định môi trường nén hoặc sự can thiệp chỉnh sửa file.

Phần Central Header:

- **Created Zip Spec**: 6.3 cho thấy file được nén bởi phiên bản ZIP 6.3.
- **Create OS**: Unix, thể hiện hệ điều hành đã tạo ra file ZIP.
- **Extract OS**: Unix, cho thấy file ZIP này có thể được giải nén trên hệ thống Unix.

Nhận xét:

Dựa trên thông tin này, chúng ta có thể phân biệt được file ZIP này được tạo bởi công cụ 7-Zip trên Linux, nhờ các dấu hiệu sau:

- Phương pháp nén BZIP2. Hệ thống Unix
- Extra Fields có dữ liệu bổ sung không thường thấy trên Windows hoặc macOS.
- Timestamp và định dạng thông tin cho thấy file này được tạo trên hệ thống Unix-based. Định dạng này thể hiện thời gian chính xác đến mức giây.

1.2. Thực nghiệm với file ZIP trên Windows

```
danu@ubuntu:~/zipdetails/bin$ ./zipdetails /home/danu/zipdetails/t/files/0002-7z/windows/7z-wi
ndows-bzip2/7z-windows-bzip2.zip

0000 LOCAL HEADER #1       04034B50 (67324752)
0004 Extract Zip Spec      2E (46) '4.6'
0005 Extract OS            00 (0) 'MS-DOS'
0006 General Purpose Flag  0000 (0)
0008 Compression Method    000C (12) 'BZIP2'
000A Modification Time     56F34719 (1458784025) 'Wed Jul 19 01:56:50 2023'
000E CRC                   F90EE7FF (4178503679)
0012 Compressed Size       00000134 (308)
0016 Uncompressed Size     000001BE (446)
001A Filename Length       0009 (9)
001C Extra Length          0000 (0)
001E Filename              'lorem.txt'
0027 PAYLOAD

015B CENTRAL HEADER #1     02014B50 (33639248)
015F Created Zip Spec      3F (63) '6.3'
0160 Created OS            00 (0) 'MS-DOS'
0161 Extract Zip Spec      2E (46) '4.6'
0162 Extract OS            00 (0) 'MS-DOS'
0163 General Purpose Flag  0000 (0)
0165 Compression Method    000C (12) 'BZIP2'
0167 Modification Time     56F34719 (1458784025) 'Wed Jul 19 01:56:50 2023'
016B CRC                   F90EE7FF (4178503679)
016F Compressed Size       00000134 (308)
0173 Uncompressed Size     000001BE (446)
0177 Filename Length       0009 (9)
0179 Extra Length          0024 (36)
017B Comment Length        0000 (0)
017D Disk Start            0000 (0)
017F Int File Attributes   0000 (0)
     [Bit 0]               0 'Binary Data'
0181 Ext File Attributes   00000020 (32)
     [Bit 5]               Archive
0185 Local Header Offset   00000000 (0)
0189 Filename              'lorem.txt'
0192 Extra ID #1           000A (10) 'NTFS FileTimes'
0194   Length              0020 (32)
0196   Reserved            00000000 (0)
019A   Tag1                0001 (1)
019C   Size1               0018 (24)
019E   Mtime               01D9BA1EF46C993E (133342306096683326) 'Wed Jul 19 01:56:49 2023 668
332600ns'
01A6   Atime               0000000000000000 (0) 'No Date/Time'
01AE   Ctime               0000000000000000 (0) 'No Date/Time'

01B6 END CENTRAL HEADER    06054B50 (101010256)
01BA Number of this disk   0000 (0)
01BC Central Dir Disk no   0000 (0)
01BE Entries in this disk  0001 (1)
01C0 Total Entries         0001 (1)
01C2 Size of Central Dir   0000005B (91)
01C6 Offset to Central Dir 0000015B (347)
01CA Comment Length        0000 (0)
#
# Done
```

Hình 6. Điều tra bằng zipdetails với file được zip bằng 7z trên Windows

Phần Local Header:

- **Compression Method**: BZIP2, cho thấy phương pháp nén được sử dụng là BZIP2.

- **Modification Time**: Wed Jul 19 01:56:50 2023, thể hiện thời điểm file lorem.txt được chỉnh sửa.
- **Filename Length**: 9 bytes, phù hợp với tên file lorem.txt.

Phần Central Header:

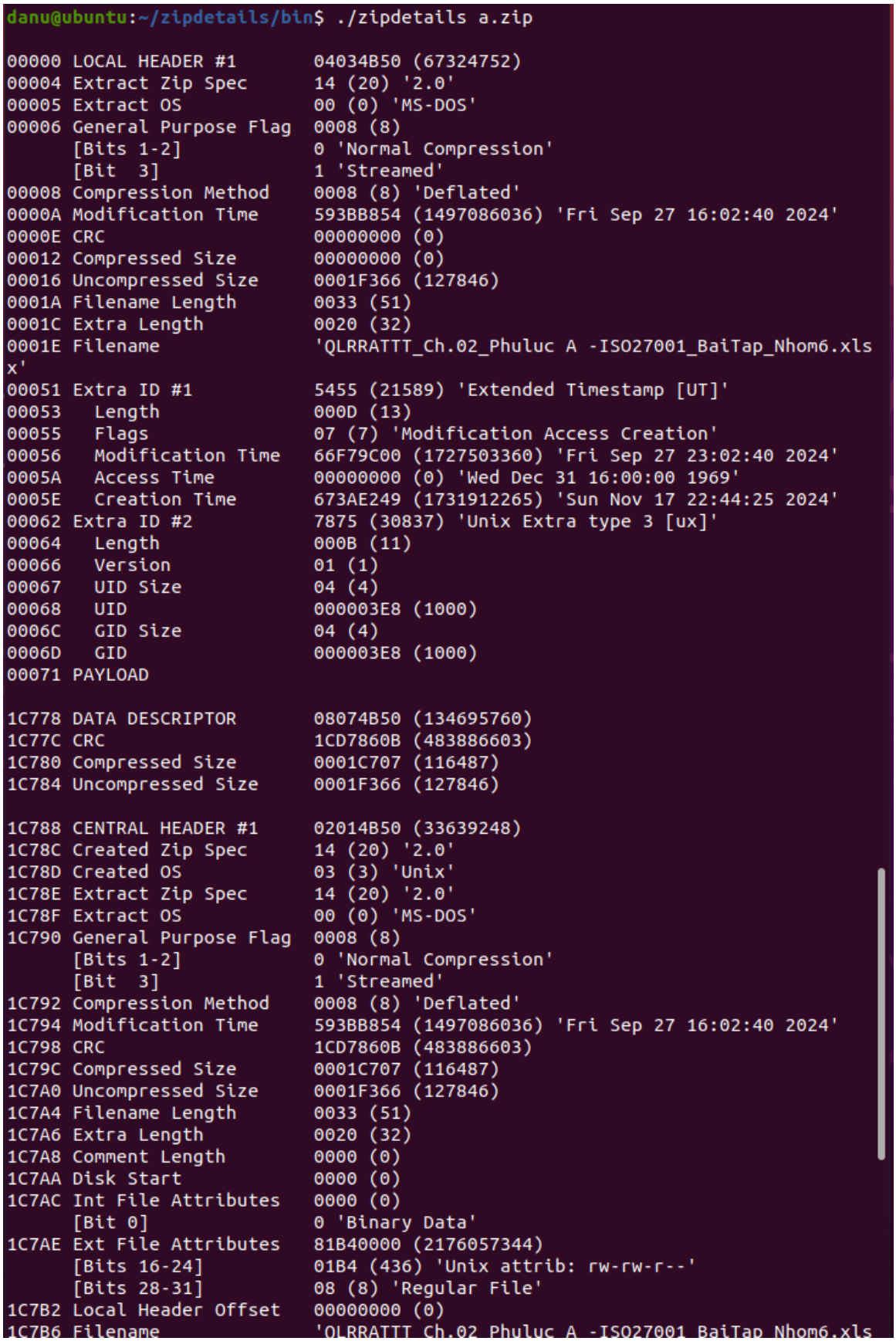
- **Created Zip Spec**: 6.3, cho thấy file được nén bởi phiên bản ZIP 6.3.
- **Create OS**: Thể hiện hệ điều hành đã tạo ra file ZIP. Trong ảnh, giá trị này là MS-DOS, thường liên quan đến hệ thống Windows.
- **Extract OS**: Cũng là MS-DOS, cho thấy file ZIP này có thể được giải nén trên hệ thống Windows.

Nhận xét:

Dựa vào các thông tin này, có thể xác định rằng file ZIP này được tạo bởi công cụ 7-Zip trên Windows, nhờ vào các dấu hiệu sau:

- Phương pháp nén BZIP2 được sử dụng phổ biến trên các hệ thống khác nhau, nhưng kết hợp với các thông tin khác có thể chỉ ra môi trường tạo file.
- MS-DOS => Windows
- Thời gian chỉnh sửa và định dạng thông tin tương thích với hệ thống Windows. Định dạng này thể hiện thời gian chính xác đến mức microseconds.

1.3. Thực nghiệm với file ZIP trên Windows được giải nén và nén lại trên Linux



Hình 7. Điều tra bằng zipdetails với file được zip bằng 7z trên Windows sau đó giải nén và nén lại trên Linux

```
1C784 Uncompressed Size     0001F366 (127846)

1C788 CENTRAL HEADER #1     02014B50 (33639248)
1C78C Created Zip Spec      14 (20) '2.0'
1C78D Created OS            03 (3) 'Unix'
1C78E Extract Zip Spec      14 (20) '2.0'
1C78F Extract OS            00 (0) 'MS-DOS'
1C790 General Purpose Flag  0008 (8)
      [Bits 1-2]            0 'Normal Compression'
      [Bit  3]              1 'Streamed'
1C792 Compression Method    0008 (8) 'Deflated'
1C794 Modification Time     593BB854 (1497086036) 'Fri Sep 27 16:02:40 2024'
1C798 CRC                   1CD7860B (483886603)
1C79C Compressed Size       0001C707 (116487)
1C7A0 Uncompressed Size     0001F366 (127846)
1C7A4 Filename Length       0033 (51)
1C7A6 Extra Length          0020 (32)
1C7A8 Comment Length        0000 (0)
1C7AA Disk Start            0000 (0)
1C7AC Int File Attributes   0000 (0)
      [Bit 0]               0 'Binary Data'
1C7AE Ext File Attributes   81B40000 (2176057344)
      [Bits 16-24]          01B4 (436) 'Unix attrib: rw-rw-r--'
      [Bits 28-31]          08 (8) 'Regular File'
1C7B2 Local Header Offset   00000000 (0)
1C7B6 Filename              'QLRRATTT_Ch.02_Phuluc A -ISO27001_BaiTap_Nhom6.xls
x'
1C7E9 Extra ID #1           5455 (21589) 'Extended Timestamp [UT]'
1C7EB   Length              000D (13)
1C7ED   Flags               07 (7) 'Modification Access Creation'
1C7EE   Modification Time   66F79C00 (1727503360) 'Fri Sep 27 23:02:40 2024'
#
# INFO: Offset 0x1C7F2: Extra Field 'Extended Timestamp [UT]' (ID 0x5455): Unex
pected 'Access Time' present
#
1C7F2   Access Time         00000000 (0) 'Wed Dec 31 16:00:00 1969'
#
# INFO: Offset 0x1C7F6: Extra Field 'Extended Timestamp [UT]' (ID 0x5455): Unex
pected 'Creation Time' present
#
1C7F6   Creation Time       673AE249 (1731912265) 'Sun Nov 17 22:44:25 2024'
1C7FA Extra ID #2           7875 (30837) 'Unix Extra type 3 [ux]'
1C7FC   Length              000B (11)
1C7FE   Version             01 (1)
1C7FF   UID Size            04 (4)
1C800   UID                 000003E8 (1000)
1C804   GID Size            04 (4)
1C805   GID                 000003E8 (1000)

1C809 END CENTRAL HEADER    06054B50 (101010256)
1C80D Number of this disk   0000 (0)
1C80F Central Dir Disk no   0000 (0)
1C811 Entries in this disk  0001 (1)
1C813 Total Entries         0001 (1)
1C815 Size of Central Dir   00000081 (129)
1C819 Offset to Central Dir 0001C788 (116616)
1C81D Comment Length        0000 (0)
#
# Info Count: 2
#
# Done
```

Hình 8. Điều tra bằng zipdetails với file được zip bằng 7z trên Windows sau đó giải nén và nén lại trên Linux

Thông tin cơ bản:

- Compression Method: "Deflated", là phương pháp nén phổ biến trong ZIP.
- General Purpose Flag: Giá trị cho thấy dữ liệu được nén theo kiểu "Streamed" (không lưu metadata khi nén).
- Filename: Tên file được nén là QLRRATTT_Ch.02_Phuluc A - ISO27001_BaiTap_Nhom6.xls.

Extra Field:

- Extra ID #1 (Extended Timestamp): Chứa thông tin liên quan đến thời gian chỉnh sửa, truy cập, và tạo file.
  - Modification Time: Fri Sep 27 23:02:40 2024.
  - Access Time: Có giá trị bất thường là Wed Dec 31 16:00:00 1969.
- Extra ID #2 (Unix Extra type 3): Chứa thông tin về UID và GID của file trên hệ thống Linux, được thiết lập lần lượt là 1000.

Central Header:

- Có sự khác biệt giữa hệ điều hành nén và giải nén:
  - Created OS: Windows (03 - Unix) khi file được nén lần cuối.
  - Extract OS: MS-DOS (00), phản ánh môi trường ban đầu khi file được nén trên Windows.

Nhận xét

- Thời gian "Access Time" bất thường:
  - Giá trị thời gian Wed Dec 31 16:00:00 1969 có thể liên quan đến việc Linux không lưu trữ hoặc không đọc được thời gian truy cập ban đầu từ Windows. Đây có thể là giá trị mặc định hoặc lỗi khi chuyển đổi giữa hai hệ thống.
- Extra Fields khác biệt:
  - Linux thêm các Extra Fields như UID và GID để phản ánh quyền sở hữu trên hệ thống Unix. Điều này không xuất hiện khi nén trên Windows.
- Tính tương thích:
  - File ZIP vẫn giữ được cấu trúc cần thiết để có thể được giải nén mà không gặp vấn đề. Điều này cho thấy sự tương thích tốt giữa hai hệ điều hành.

2. Demo thay đổi Extract OS, Created OS của file ZIP và kiểm thử với tool

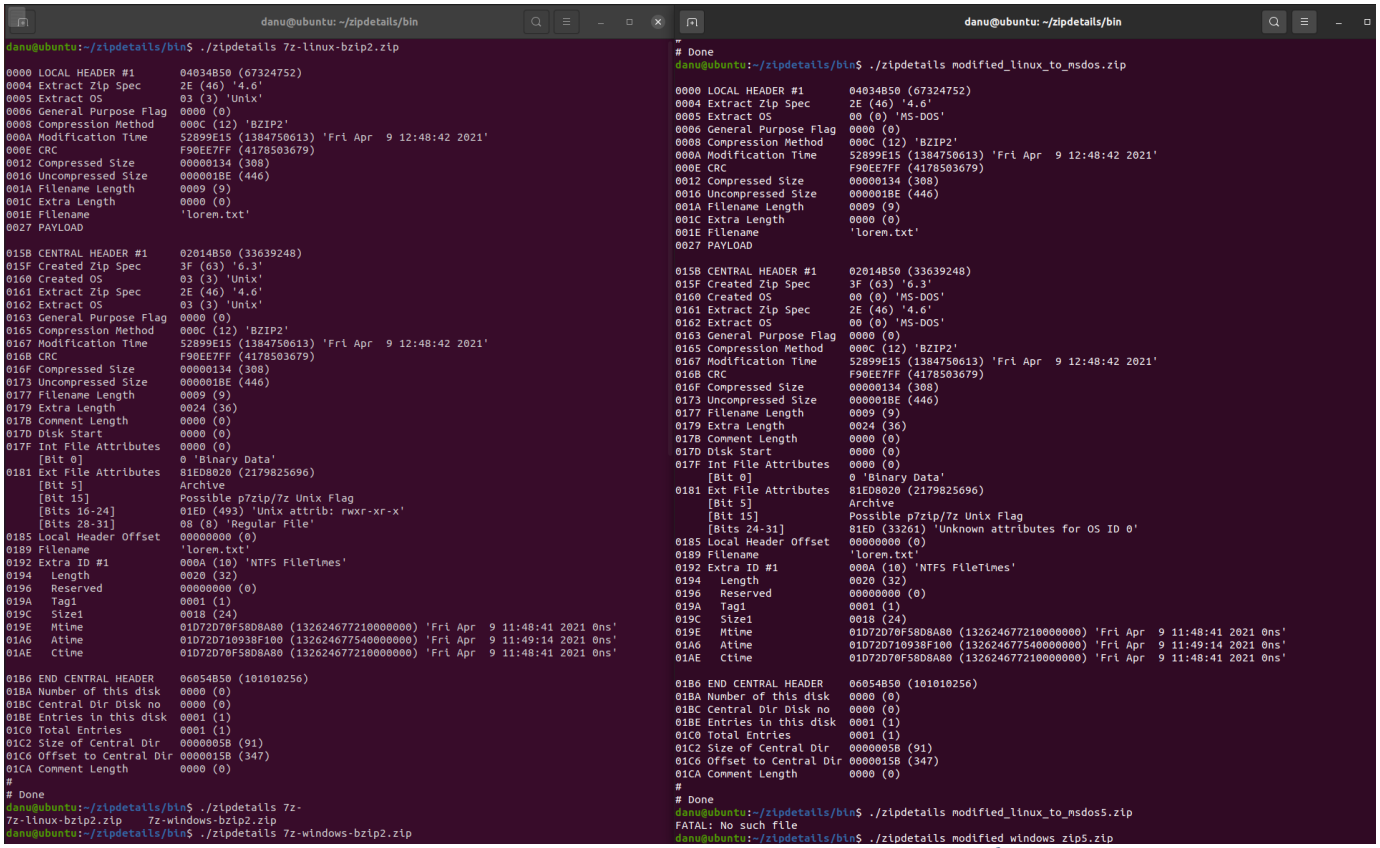
Viết code python dùng để thay đổi Extract OS, Created OS của file ZIP

```
import struct
def replace_unix_with_msdos(zip_path, output_path):
    with open(zip_path, 'rb') as zip_file:
        data = zip_file.read()

    modified_data = bytearray(data)
```

```
    central_directory_signature = b"PK\x01\x02"
    local_file_header_signature = b"PK\x03\x04"

    central_directory_offset = modified_data.find(central_directory_signature)
    while central_directory_offset != -1:
        version_made_by_offset = central_directory_offset + 4
        current_version_made_by = struct.unpack("<H",
modified_data[version_made_by_offset:version_made_by_offset + 2])[0]

        new_version_made_by = (0 << 8) | (current_version_made_by & 0xFF)
        struct.pack_into("<H", modified_data, version_made_by_offset, new_version_made_by)

        version_needed_offset = central_directory_offset + 6
        current_version_needed = struct.unpack("<H",
modified_data[version_needed_offset:version_needed_offset + 2])[0]

        new_version_needed = (0 << 8) | (current_version_needed & 0xFF)
        struct.pack_into("<H", modified_data, version_needed_offset, new_version_needed)

        central_directory_offset = modified_data.find(central_directory_signature,
central_directory_offset + 46)

    local_file_header_offset = modified_data.find(local_file_header_signature)
    while local_file_header_offset != -1:
        version_needed_offset = local_file_header_offset + 4
        current_version = struct.unpack("<H",
modified_data[version_needed_offset:version_needed_offset + 2])[0]

        new_version = (0 << 8) | (current_version & 0xFF)
        struct.pack_into("<H", modified_data, version_needed_offset, new_version)

        local_file_header_offset = modified_data.find(local_file_header_signature,
local_file_header_offset + 30)

    with open(output_path, 'wb') as output_file:
        output_file.write(modified_data)

    print(f"Change Unix to MS-DOS success!")

zip_file_path = "C:\\Users\\Lenovo\\DanuxPeach\\FileHocTap\\Nam4\\Foren\\doan\\sample\\7z-
linux-bzip2.zip"
output_file_path =
"C:\\Users\\Lenovo\\DanuxPeach\\FileHocTap\\Nam4\\Foren\\doan\\output\\modified_linux_to_m
sdos.zip"
replace_unix_with_msdos(zip_file_path, output_file_path)
```

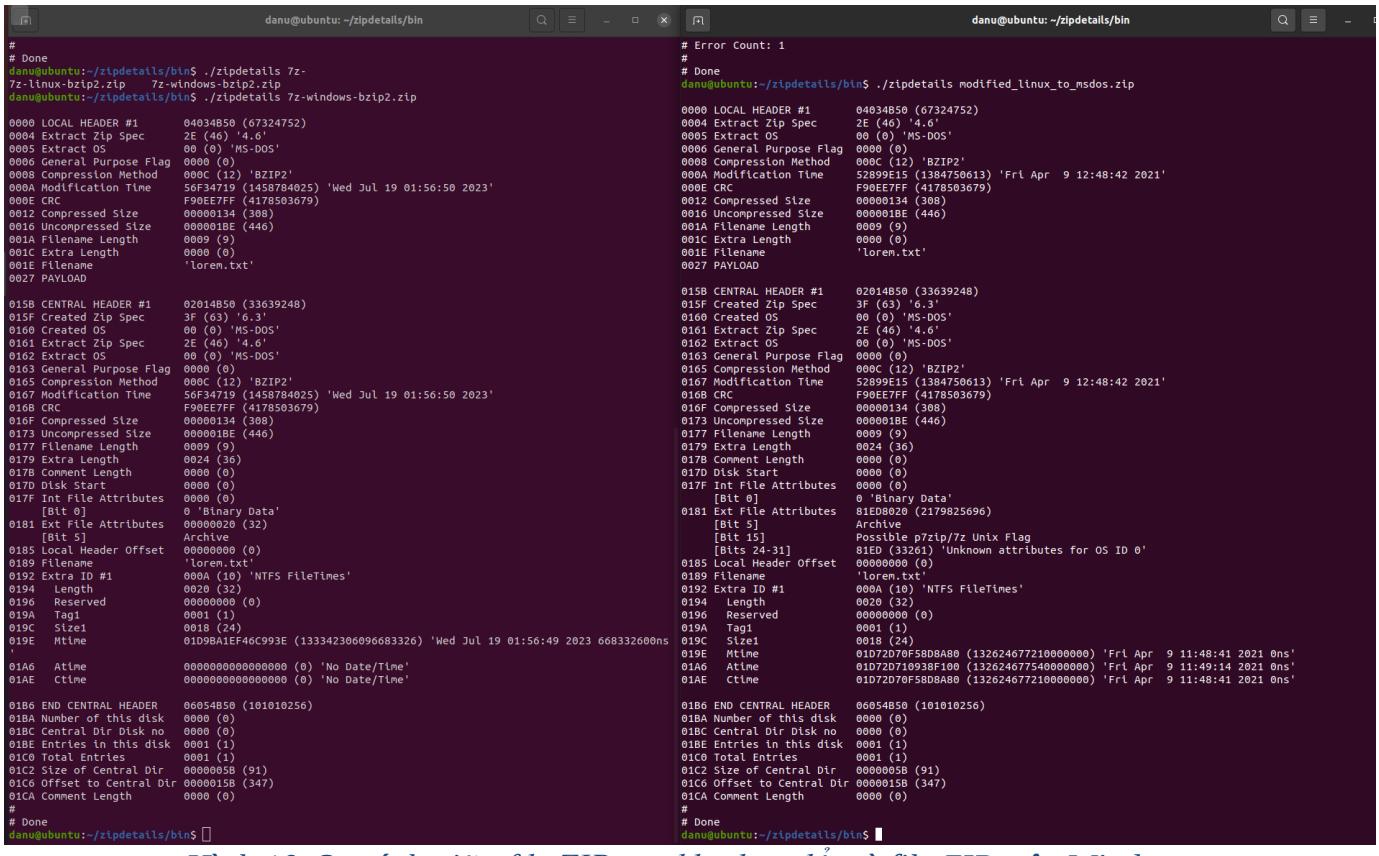
Ở kịch bản này thì nhóm thử thay đổi thông tin của file ZIP ở trên Linux thì thu được kết quả như

Hình 9. Chứng tỏ được là tool này không thể phát hiện thông tin của file ZIP được thay đổi bởi code python. Nếu kỹ hơn thì có thể thay đổi các thông tin khác cho giống với file ZIP trên Windows hơn.



Hình 9. So sánh file ZIP trước và sau thay đổi

Sau khi thay đổi thành công thì nhóm thực hiện so sánh với file ZIP trên Windows thì thấy được giống. Từ đó có thể thay đổi thông tin file ZIP để lẩn tránh công cụ truy vết dấu vân tay của file ZIP.



Hình 10. So sánh giữa file ZIP sau khi thay đổi và file ZIP trên Windows

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Trong nghiên cứu này, nhóm đã thực hiện phân tích các đặc điểm pháp chứng số (fingerprinting) của định dạng file ZIP để xác định nguồn gốc và theo dõi sự thay đổi của file. Những kết quả đạt được bao gồm:

1.1. Phân tích fingerprint của file ZIP

- Nhận diện được các đặc điểm pháp chứng quan trọng trong file ZIP, bao gồm:
  - Metadata (thời gian tạo, chỉnh sửa).
  - Các trường Extra Fields (các giá trị bổ sung khác nhau giữa môi trường).
  - Cách xử lý thư mục và file trong quá trình nén.

1.2. Mô hình hóa môi trường tạo file ZIP

- Từ các fingerprint thu được, nhóm có thể xác định được nguồn gốc của file ZIP, bao gồm: hệ điều hành sử dụng, công cụ nén, và đôi khi cả phiên bản phần mềm.

1.3. Công cụ hỗ trợ phân tích

- Một công cụ bán tự động đã được phát triển, hỗ trợ quá trình phân tích và phân loại các file ZIP dựa trên các fingerprint, giúp tăng tốc độ và độ chính xác trong

quá trình điều tra pháp chứng số.

1.4. Ý nghĩa pháp chứng số

- Các kết quả nghiên cứu có ý nghĩa quan trọng trong việc tái dựng sự kiện từ các file nén, hỗ trợ điều tra các vụ việc liên quan đến an ninh mạng, vi phạm bản quyền, hoặc gian lận thông tin.

Tuy nhiên, nghiên cứu vẫn còn một số hạn chế, chẳng hạn như cần cập nhật fingerprint khi các công cụ và hệ điều hành thay đổi, và chưa xử lý tốt các trường hợp file ZIP bị thay đổi hoặc nén lại nhiều lần.

2. Hướng phát triển

2.1. Phân tích thêm các định dạng nén khác (OOXML, JAR).

- Nghiên cứu các định dang phổ biến hơn như OOXML (Word, Excel, PowerPoint) và JAR (Java).

2.2. Tăng độ chính xác và mở rộng dataset.

- Xây dựng cơ sở dữ liệu dấu vân tay toàn diện hơn để hỗ trợ nhiều hệ điều hành và ứng dụng hơn.
- Khám phá thêm các phương pháp đối phó với các kỹ thuật chống pháp chứng nhằm cải thiện độ chính xác của phân tích.

HẾT