**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**



**PHÁP CHỨNG KỸ THUẬT SỐ**

**ĐỀ TÀI: TÌM HIỂU VÀ TRIỂN KHAI KỸ THUẬT PHÁP CHỨNG SỐ TRÊN HĐH LINUX**

**Giảng viên hướng dẫn: VŨ ĐỨC THỊNH**

**Sinh viên thực hiện:**

**2033216489 – Trần Ngọc Minh**

**2033210445 – Nguyễn Ngọc Lan Anh**

**2033216427 – Trần Đỗ Quốc Hùng**

**TP. Hồ Chí Minh, ngày 18 tháng 10 năm 2024**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**



**PHÁP CHỨNG KỸ THUẬT SỐ**

**ĐỀ TÀI: TÌM HIỂU VÀ TRIỂN KHAI KỸ THUẬT PHÁP CHỨNG SỐ TRÊN HĐH LINUX**

Giảng viên hướng dẫn: VŨ ĐỨC THỊNH

Sinh viên thực hiện:

2033216489 – Trần Ngọc Minh (nhóm trưởng)

2033210445 – Nguyễn Ngọc Lan Anh

2033216427 – Trần Đỗ Quốc Hùng

TP. Hồ Chí Minh, ngày 18 tháng 10 năm 2024

**BẢNG PHÂN CÔNG**

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Nhiệm vụ** | **Tiến độ** |
| 2033216489 | Trần Ngọc Minh (nhóm trưởng) | Tìm hiểu tài liệu và làm word | 100% |
| 2033210445 | Nguyễn Ngọc Lan Anh | Tìm hiểu tài liệu và trình bày demo | 100% |
| 2033216427 | Trần Đỗ Quốc Hùng | Tìm hiểu tài liệu | 30% |

**LỜI CAM ĐOAN**

Chúng em xin cam đoan rằng tiểu luận "Tìm hiểu và triển khai kỹ thuật pháp chứng số trên hệ điều hành Linux" này là kết quả nghiên cứu và làm việc của của nhóm, dưới sự hướng dẫn của thầy Vũ Đức Thịnh.

Kết quả bài làm của đề tài **“**Tìm hiểu và triển khai kỹ thuật pháp chứng số trên hệ điều hành Linux**”** là trung thực và không sao chép từ bất kỳ bài tập của nhóm khác.

Chúng em đã tuân thủ nghiêm ngặt các quy định về trích dẫn nguồn tài liệu và không sao chép bất kỳ phần nào từ các tài liệu khác mà không ghi rõ nguồn gốc. Nếu không đúng sự thật, chúng em xin chịu mọi trách nhiệm.

.

**LỜI CẢM ƠN**

Để hoàn thành bài báo cáo này, chúng em xin chân thành cảm ơn thầy Vũ Đức Thịnh đã tận tình hướng dẫn cũng như nhận xét, góp ý trong suốt quá trình học tập. Với lòng biết ơn sâu sắc nhất, chúng em xin phép gửi đến thầy. Cảm ơn thầy đã truyền đạt vốn kiến thức và những thông tin quý giá cần có cho chúng em trong suốt quá trình học tập tại trường. Nhờ sự hướng dẫn, chỉ bảo tận tình của thầy đã giúp chúng em hoàn thiện đề tài một cách tốt nhất.

Chúng em xin gửi lời cảm ơn chân thành đến thầy, người đã giúp đỡ và hướng dẫn trong quá trình hoàn thành bài báo cáo này. Vì kiến thức và thời gian hạn hẹp nên bài báo cáo không tránh khỏi những sai sót.

Chúng em mong nhận được sự đóng góp ý kiến từ thầy để giúp hoàn thiện kiến thức và bổ sung thêm thông tin cần thiết cho báo cáo của nhóm em.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 1](#_Toc183181534)

[**CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI** 2](#_Toc183181535)

[**1. Bối cảnh và tầm quan trọng** 2](#_Toc183181536)

[**2. Mục tiêu và yêu cầu đề tài** 2](#_Toc183181537)

[**3. Mục đích nghiên cứu** 3](#_Toc183181538)

[**4. Phạm vi đề tài** 3](#_Toc183181539)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 4](#_Toc183181540)

[**1. Tổng quan về pháp chứng số** 4](#_Toc183181541)

[**1.1. Tổng quan về pháp chứng số?** 4](#_Toc183181542)

[**1.2. Quy trình pháp chứng số** 6](#_Toc183181543)

[**1.3. Các tiêu chuẩn trong pháp chứng số** 9](#_Toc183181544)

[**2. Tổng quan về hệ điều hành Linux** 11](#_Toc183181545)

[**2.1. Hệ điều hành Linux là gì?** 11](#_Toc183181546)

[**2.2. Ưu, nhược điểm của hệ điều hành Linux** 14](#_Toc183181547)

[**2.3** **Tổng quan về hệ thống Linux trong phân tích pháp chứng số.** 17](#_Toc183181548)

[**3. Các công cụ và kỹ thuật pháp chứng số** 47](#_Toc183181549)

[**3.1. Thu thập dữ liệu** 47](#_Toc183181550)

[**3.2. Bảo quản dữ liệu** 48](#_Toc183181551)

[**3.3. Phân tích dữ liệu và hệ thống** 50](#_Toc183181552)

[**3.4. Dịch ngược** 51](#_Toc183181553)

[**CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM** 53](#_Toc183181554)

[**1. Triển khai điều tra thử thách SCAN24 của Honeynet Project** 53](#_Toc183181555)

[**1.1. SCAN24 là gì?** 53](#_Toc183181556)

[**1.2. Báo cáo của cảnh sát** 53](#_Toc183181557)

[**1.3. Các câu hỏi cần tìm hiểu và điều tra** 54](#_Toc183181558)

[**2. Triển khai điều tra** 55](#_Toc183181559)

[**2.1. Pháp chứng với Autopsy** 55](#_Toc183181560)

[**2.2. Pháp chứng với The Sleuth Kit (TSK)** 71](#_Toc183181561)

[**KẾT LUẬN** 82](#_Toc183181562)

[**TÀI LIỆU THAM KHẢO** 83](#_Toc183181563)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Pháp chứng số. 5](#_Toc182844468)

[Hình 2. Quy trình pháp chứng số. 7](#_Toc182844469)

[Hình 3. Mô hình pháp y. 8](#_Toc182844470)

[Hình 4. Mô hình pháp y kỹ thuật số bốn giai đoạn. 8](#_Toc182844471)

[Hình 5. Hệ điều hành Linux. 12](#_Toc182844472)

[Hình 6. Khác biệt về cấu trúc của Windows và Linux 18](#_Toc182844473)

[Hình 7. Cấu trúc hệ thống phân vùng Linux. 18](#_Toc182844474)

[Hình 8. Hệ thống log. 31](#_Toc182844475)

[Hình 9. Lệnh xem một phần cuối của một file log. 31](#_Toc182844476)

[Hình 10. Xem tiến trình bộ nhớ. 34](#_Toc182844477)

[Hình 11. Xem các tiến trình đang mở kết nối mạng. 35](#_Toc182844478)

[Hình 12. Xem ACLs của một tệp cụ thể. 38](#_Toc182844479)

[Hình 13. Sử dụng netstat để liệt kê kết nối. 39](#_Toc182844480)

[Hình 14. Liệt kê giao diện mạng. 39](#_Toc182844481)

[Hình 15. Bắt lưu lượng mạng bằng công cụ. 40](#_Toc182844482)

[Hình 16. Bắt lưu lượng bằng wireshark. 40](#_Toc182844483)

[Hình 17. Giao diện trang Honeynet. 56](#_Toc182844484)

[Hình 18. Kiểm tra tính toàn vẹn bằng MD5. 57](#_Toc182844485)

[Hình 19. Unzip file image vừa tải. 57](#_Toc182844486)

[Hình 20. Kiểm tra định dạng thật của file. 58](#_Toc182844487)

[Hình 21. Unzip file. 58](#_Toc182844488)

[Hình 22. Mở Autopsy. 59](#_Toc182844489)

[Hình 23. Giao diện của Autopsy. 59](#_Toc182844490)

[Hình 24. Giao diện lúc tạo New Case. 60](#_Toc182844491)

[Hình 25. Giao diện lúc tạo Case xong. 60](#_Toc182844492)

[Hình 26. Điền thông tin và Add Host cho Case. 61](#_Toc182844493)

[Hình 27. Giao diện lúc Add host xong. 61](#_Toc182844494)

[Hình 28. Add image. 61](#_Toc182844495)

[Hình 29. Đường dẫn của file image. 62](#_Toc182844496)

[Hình 30. Điền đường dẫn vào mục Location. 62](#_Toc182844497)

[Hình 31. Cảnh báo Autopsy không thể xác định loại hệ thống ổ đĩa cho ảnh đĩa. 63](#_Toc182844498)

[Hình 32. Tiến hành phân tích dữ liệu ổ đĩa. 64](#_Toc182844499)

[Hình 33. Thông tin của các file trong ổ đĩa. 64](#_Toc182844500)

[Hình 34. Thông tin của từng file. 64](#_Toc182844501)

[Hình 35. Check các dạng hiển thị của ASCII. 65](#_Toc182844502)

[Hình 36. Sử dụng hexdump để xem tất cả dữ liệu của đĩa. 67](#_Toc182844503)

[Hình 37. Output có xuất hiện cả những file ta đã được xem trên Autopsy. 68](#_Toc182844504)

[Hình 38. Tìm thấy password. 68](#_Toc182844505)

[Hình 39. Trích xuất đĩa. 69](#_Toc182844506)

[Hình 40. Unzip file trích xuất. 69](#_Toc182844507)

[Hình 41. File .xls được tìm thấy sau khi unzip. 70](#_Toc182844508)

[Hình 42. Lịch trình của Joe được hiển thị sau khi mở file .xls. 70](#_Toc182844509)

[Hình 43. Dùng dd để thực hiện khôi phục ảnh đã bị hỏng. 71](#_Toc182844510)

[Hình 44. Ảnh sau khi khôi phục. 71](#_Toc182844511)

[Hình 45. Xem thông tin file sau khi đã tải về bằng lệnh exiftool. 73](#_Toc182844512)

[Hình 46. File sau khi được giải nén. 73](#_Toc182844513)

[Hình 47. Dùng lệnh mmls để hiển thị các phân vùng của image. 74](#_Toc182844514)

[Hình 48. Dùng lệnh fsstat để xem thông tin thống kê về hệ thống file. 75](#_Toc182844515)

[Hình 49. Dùng lệnh fls để liệt kê các tệp và thư mục. 77](#_Toc182844516)

[Hình 50. Có được bức thư sau khi trích xuất file ra để xem. 78](#_Toc182844517)

[Hình 51. Tìm password bằng hexdump kết hợp với grep. 78](#_Toc182844518)

[Hình 52. Tìm thấy vị trí sector bằng lệnh istat. 79](#_Toc182844519)

[Hình 53. Mount image. 79](#_Toc182844520)

[Hình 54. Tìm thấy 2 file đã bị xóa. 80](#_Toc182844521)

[Hình 55. Khôi phục file chứa lịch trình của Joe nằm bắt đầu từ sector 104. 80](#_Toc182844522)

[Hình 56. Giải nén file Shedule và thu được bảng excel. 80](#_Toc182844523)

[Hình 57. File excel sau khi mở lên. 81](#_Toc182844524)

[Hình 58. Khôi phục ảnh. 81](#_Toc182844525)

[Hình 59. Ảnh sau khi đã được khôi phục. 82](#_Toc182844526)

# **LỜI NÓI ĐẦU**

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, dữ liệu số trở thành tài sản quý giá đối với các cá nhân, tổ chức và doanh nghiệp. Cùng với đó, các hành vi vi phạm an ninh mạng như tấn công mạng, đánh cắp dữ liệu, và gian lận trực tuyến cũng ngày càng gia tăng, gây thiệt hại nghiêm trọng. Để đối phó với tình trạng này, pháp chứng số đã ra đời như một giải pháp hỗ trợ trong việc điều tra và thu thập bằng chứng số phục vụ cho các hoạt động điều tra pháp lý.

Pháp chứng số (Digital Forensics) là quá trình thu thập, phân tích và bảo mật các bằng chứng điện tử, từ đó sử dụng các thông tin thu được để làm cơ sở cho các hoạt động điều tra và truy tố tội phạm. Trong bối cảnh các hệ điều hành mã nguồn mở như Linux ngày càng trở nên phổ biến trong môi trường máy chủ và doanh nghiệp, việc triển khai các kỹ thuật pháp chứng số trên nền tảng này trở nên vô cùng quan trọng.

Đề tài "Tìm hiểu và triển khai Kỹ thuật Pháp chứng số trên HĐH Linux" nhằm mục đích cung cấp cái nhìn tổng quan về các phương pháp và công cụ phục vụ cho việc thu thập, bảo mật, và phân tích dữ liệu số trên hệ điều hành Linux. Ngoài ra, đề tài cũng tập trung vào việc triển khai thực tiễn các kỹ thuật này, giúp người đọc có cái nhìn rõ ràng và hiểu biết sâu hơn về quy trình pháp chứng số trong môi trường thực tế.

# **CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI**

## **1. Bối cảnh và tầm quan trọng**

Pháp chứng số có nguồn gốc từ sự phát triển của công nghệ máy tính và mạng, cùng với nhu cầu xử lý và điều tra các tội phạm liên quan đến công nghệ thông tin. Vào những năm 1970 và 1980, khi máy tính và các hệ thống thông tin bắt đầu phổ biến hơn trong các tổ chức, chính phủ và doanh nghiệp, các vụ án liên quan đến gian lận máy tính và tấn công mạng cũng bắt đầu xuất hiện. Pháp chứng số đã phát triển từ những ngày đầu khi nó chỉ là một phần của các cuộc điều tra gian lận máy tính và đã trở thành một lĩnh vực khoa học chuyên nghiệp với các quy trình và phương pháp nghiên cứu rõ ràng.

Trong bối cảnh hiện đại, công nghệ thông tin và internet đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của con người và hoạt động của các tổ chức, doanh nghiệp. Tuy nhiên, điều này cũng đồng nghĩa với việc tội phạm mạng và các mối đe dọa liên quan đến công nghệ kỹ thuật số ngày càng gia tăng và phức tạp hơn. Các cuộc tấn công mạng có thể gây ra những hậu quả nghiêm trọng ảnh hưởng trực tiếp đến cá nhân, tổ chức doanh nghiệp. Pháp chứng số không chỉ quan trọng trong việc giải quyết các vấn đề pháp lý liên quan đến tội phạm mạng, mà còn là một phần thiết yếu trong chiến lược bảo vệ an ninh thông tin của các tổ chức và quốc gia. Với bối cảnh các mối đe dọa mạng ngày càng gia tăng, pháp chứng số ngày càng trở thành một lĩnh vực không thể thiếu trong an ninh mạng hiện đại.

## **2. Mục tiêu và yêu cầu đề tài**

Mục tiêu:

* Hiểu rõ khái niệm và nguyên tắc của pháp chứng số.
* Nắm được quy trình pháp chứng số từ thu thập đến phân tích bằng chứng.
* Khảo sát các công cụ pháp chứng số phổ biến có thể chạy trên hệ điều hành Linux.

Yêu cầu:

* Nắm được kiến thức về lĩnh vực pháp chứng số.
* Tìm hiểu các công cụ và phần mềm pháp chứng số trên Linux.
* Thực hiện được kịch bản pháp chứng số trên Linux.

## **3. Mục đích nghiên cứu**

Mục đích của việc nghiên cứu đề tài này là nhằm hiểu sâu hơn về các phương pháp và kỹ thuật pháp chứng số (digital forensics) trên hệ điều hành Linux, đồng thời cung cấp các phương pháp triển khai thực tế cho các cuộc điều tra pháp chứng liên quan đến bảo mật thông tin và tội phạm mạng. Không chỉ giới hạn trong việc tìm hiểu lý thuyết, mà còn áp dụng thực tế các kiến thức và kỹ thuật pháp chứng số trên Linux, nhằm xây dựng một nền tảng vững chắc cho bản thân trong việc đối phó với các thách thức và mối đe dọa an ninh mạng hiện đại.

## **4. Phạm vi đề tài**

Đề tài chủ yếu tập trung vào các phần sau:

* Tìm hiểu về cơ sở lý thuyết của pháp chứng số, đặc biệt trong môi trường Linux.
* Tập trung vào hệ điều hành Linux, đặc biệt là các phiên bản phổ biến như Ubuntu, Debian, CentOS, hoặc Kali Linux.
* Sử dụng một số công cụ pháp chứng số phổ biến trên Linux.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

## **1. Tổng quan về pháp chứng số**

### **1.1. Tổng quan về pháp chứng số?**

Tại Hội thảo nghiên cứu pháp chứng kỹ thuật số đầu tiên (Digital Forensics Research Workshop - DFRWS) năm 2001, pháp chứng kỹ thuật số được định nghĩa là:

*Việc sử dụng các phương pháp được nghiên cứu và chứng minh khoa học nhằm bảo tồn, thu thập, xác thực, nhận dạng, phân tích, diễn giải, lập hồ sơ và trình bày các bằng chứng kỹ thuật số được trích xuất từ các nguồn kỹ thuật số, với mục đích tạo điều kiện hoặc thúc đẩy việc tái hiện các sự kiện được xác định là tội phạm, hoặc giúp dự đoán các hành động trái phép được chứng minh là gây rối loạn đối với các hoạt động đã lên kế hoạch.*

Khoa học pháp chứng áp dụng khoa học tự nhiên, vật lý và xã hội vào các vấn đề pháp luật. Một trong những lĩnh vực chuyên sâu của khoa học pháp chứng là khoa học pháp chứng số (thường được gọi tắt là pháp chứng số). Pháp chứng số là nhánh của khoa học pháp chứng chuyên về việc áp dụng luật tố tụng hình sự và xử lý bằng chứng liên quan đến máy tính và các thiết bị công nghệ. Phạm vi của nó bao gồm các thiết bị di động như điện thoại thông minh, máy chơi game, cũng như các thiết bị kết nối Internet khác như thiết bị theo dõi sức khỏe, thiết bị thể dục và thiết bị y tế. Cụ thể hơn, pháp chứng số là quá trình toàn diện bao gồm việc thu thập, tiếp nhận, bảo quản, phân tích và trình bày bằng chứng điện tử (còn được gọi là bằng chứng số). Quá trình này phục vụ cho mục đích tình báo, hỗ trợ điều tra và truy tố các loại tội phạm đa dạng, trong đó có cả tội phạm mạng. Pháp chứng số đóng vai trò quan trọng trong việc cung cấp các bằng chứng khoa học, đáng tin cậy cho quá trình tố tụng pháp lý trong thời đại số.

Khoa học pháp chứng kỹ thuật số được hỗ trợ bởi các nguyên tắc pháp y, chẳng hạn như nguyên tắc trao đổi của Edmond Locard. Edmond Locard (1877 – 1966) được mệnh danh là Sherlock Holmes của nước Pháp. Ông là một chuyên gia điều tra pháp y, sáng lập Viện Hình sự học của trường Đại học Tổng hợp Lyon. Locard phát biểu một nguyên tắc mà sau này trở thành kim chỉ nam ngành khoa học điều tra. Ông ta cho rằng bất cứ khi nào hai người tiếp xúc với nhau, một thứ gì đó từ một người sẽ được trao đổi với người khác và ngược lại. Có thể là bụi, tế bào da, bùn đất, sợi, mạt kim loại. Những việc trao đổi này có xảy ra - vì thế chúng ta có thể bắt được nghi phạm. Với Computer Forensics, nguyên tắc này cũng hoàn toàn đúng. Khi bạn làm việc với máy tính hay một hệ thống thông tin, tất cả hành động của bạn đều bị ghi vết lại (mặc dù việc tìm ra thủ phạm trong trường hợp này khó khăn và mất nhiều thời gian hơn rất nhiều)

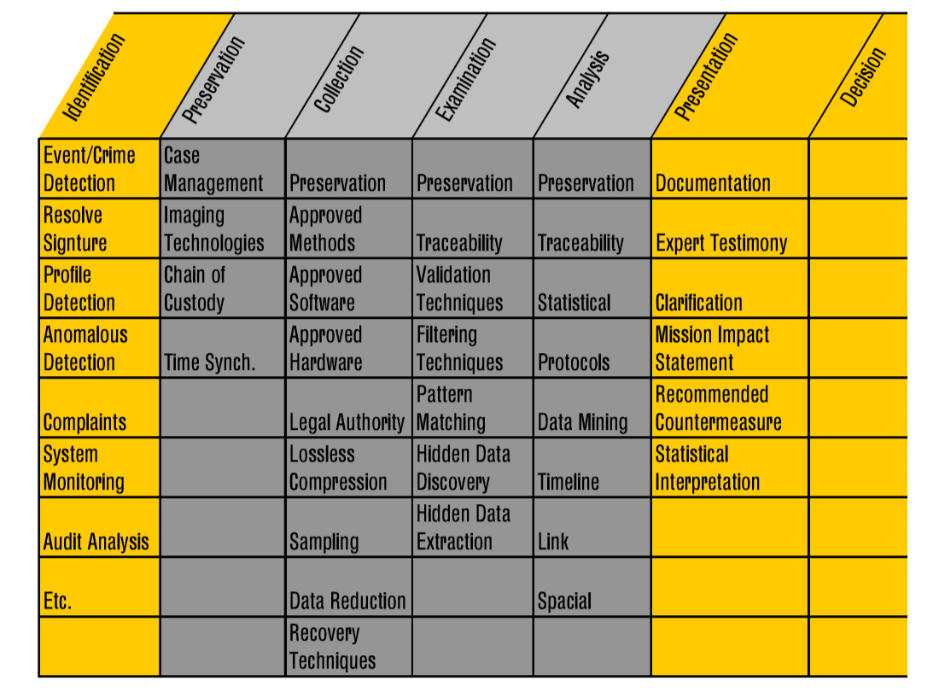


Hình 1. Pháp chứng số.

Các dấu vết kỹ thuật số được để lại do kết quả của việc cá nhân sử dụng công nghệ thông tin và truyền thông (ICT). Đặc biệt, một người sử dụng ICT có thể để lại dấu chân kỹ thuật số, tức là dữ liệu do người dùng ICT để lại có thể tiết lộ thông tin về họ, bao gồm tuổi, giới tính, chủng tộc, dân tộc, quốc tịch, khuynh hướng tình dục, suy nghĩ, sở thích, thói quen, sở thích, tiền sử bệnh án và mối quan tâm, rối loạn tâm lý, tình trạng việc làm, mối quan hệ, vị trí địa lý, thói quen và các hoạt động khác. Dấu chân kỹ thuật số (Digital Footprint) này có thể chủ động hoặc thụ động. Dấu chân kỹ thuật số chủ động được tạo ra bởi dữ liệu do người dùng cung cấp, chẳng hạn như thông tin cá nhân, video, hình ảnh và bình luận được đăng trên các ứng dụng, trang web, bảng tin, phương tiện truyền thông xã hội và các diễn đàn trực tuyến khác. Dấu vết kỹ thuật số thụ động là dữ liệu được người dùng Internet và công nghệ kỹ thuật số thu thập và vô tình để lại (ví dụ: lịch sử trình duyệt Internet). Dữ liệu là một phần của dấu vết kỹ thuật số chủ động và thụ động có thể được sử dụng làm bằng chứng về tội phạm, bao gồm cả tội phạm mạng (tức là bằng chứng kỹ thuật số - Digital Evidence). Dữ liệu này cũng có thể được sử dụng để chứng minh hoặc bác bỏ một vấn đề đang được khẳng định; bác bỏ hoặc hỗ trợ lời khai của nạn nhân, nhân chứng hoặc nghi phạm; và/hoặc buộc tội hoặc minh oan cho nghi phạm về một tội phạm. Bằng chứng số có thể bị sao chép, chỉnh sửa nên chúng ta cần bảo đảm tính nguyên vẹn của bằng chứng, vì nếu giám sát viên không thể kiểm soát toàn bộ hệ thống máy tính thì bằng chứng của họ sẽ không được công nhận, vì thế bằng chứng kỹ thuật số đặt ra những thách thức lớn do khối lượng dữ liệu có sẵn, tốc độ của dữ liệu (tức là tốc độ tạo và chuyển dữ liệu), tính không ổn định của dữ liệu (tức là dữ liệu có thể nhanh chóng biến mất do bị ghi đè hoặc xóa) và tính dễ vỡ của dữ liệu (tức là dữ liệu có thể dễ dàng bị thao túng, thay đổi hoặc làm hỏng). Ngoài ra, việc đánh giá tính xác thực của bằng chứng kỹ thuật số cũng bao gồm việc kiểm tra các quy trình, phương pháp và công cụ được sử dụng để thu thập, thu thập, bảo quản và phân tích bằng chứng kỹ thuật số để đảm bảo rằng dữ liệu không bị sửa đổi theo bất kỳ cách nào.

### **1.2. Quy trình pháp chứng số**

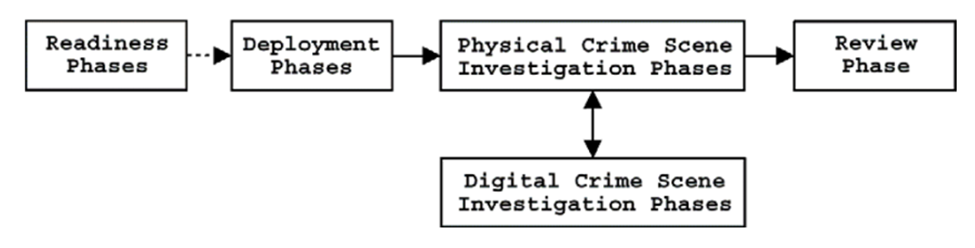
Quy trình khoa học pháp y kỹ thuật số bao gồm: tìm kiếm (Search), thu thập (Acquisition), bảo quản (Preservation) và duy trì bằng chứng kỹ thuật số (Maintenance of digital evidence); Mô tả, giải thích và xác lập nguồn gốc của bằng chứng kỹ thuật số và ý nghĩa của nó; Phân tích bằng chứng và tính hợp lệ, độ tin cậy và sự liên quan của nó đối với vụ án; Và báo cáo bằng chứng có liên quan đến vụ án. Năm 2001, Hội thảo nghiên cứu khoa học pháp chứng kỹ thuật số, một tổ chức tình nguyện phi lợi nhuận, [chuyên] tài trợ cho các nhóm làm việc kỹ thuật, các hội nghị thường niên và các thử thách để giúp thúc đẩy hướng nghiên cứu và phát triển, đã phát triển một mô hình dựa trên giao thức của Cục Điều tra Liên bang Hoa Kỳ về các cuộc khám xét hiện trường vụ án thực tế, bao gồm bảy giai đoạn: nhận dạng, bảo quản, thu thập, kiểm tra, phân tích, trình bày và quyết định.



Hình 2. Quy trình pháp chứng số.

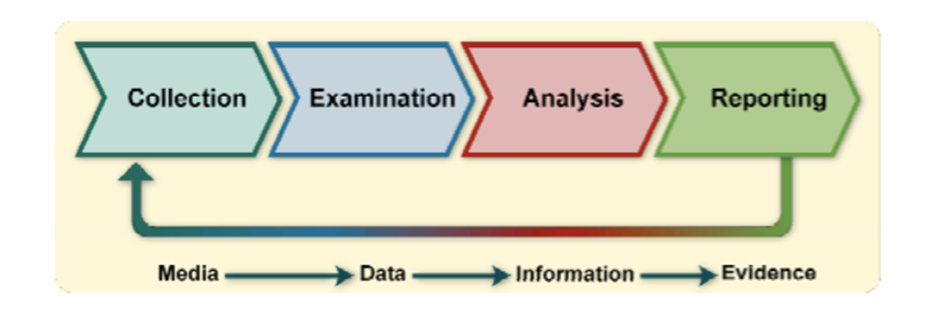
Năm 2002, một mô hình pháp y kỹ thuật số khác đã được đề xuất, dựa trên mô hình Hội thảo nghiên cứu pháp y kỹ thuật số năm 2001 và giao thức tìm kiếm hiện trường vụ án của Cục Điều tra Liên bang Hoa Kỳ (đối với hiện trường vụ án thực tế). Mô hình pháp y kỹ thuật số trừu tượng có chín giai đoạn: nhận dạng, chuẩn bị, chiến lược tiếp cận, bảo quản, thu thập, kiểm tra, phân tích, trình bày và trả lại bằng chứng.

Năm 2003, Mô hình điều tra kỹ thuật số tích hợp đã được đề xuất, đây là phương pháp điều tra toàn diện hơn có năm giai đoạn cơ bản, mỗi giai đoạn có các giai đoạn riêng: sẵn sàng, triển khai, điều tra hiện trường vụ án vật lý, điều tra hiện trường vụ án kỹ thuật số, và xem xét.



Hình 3. Mô hình pháp y.

Năm 2006, Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ đã đề xuất một mô hình pháp y kỹ thuật số bốn giai đoạn trong Hướng dẫn tích hợp các kỹ thuật pháp y vào Phản ứng sự cố (SP 800-86): giai đoạn thu thập, giai đoạn kiểm tra, giai đoạn phân tích, và giai đoạn báo cáo.



Hình 4. Mô hình pháp y kỹ thuật số bốn giai đoạn.

Một mô hình điều tra khác được Viện Tư pháp Quốc gia (NIJ) thuộc Bộ Tư pháp Hoa Kỳ đề xuất vào năm 2001 và được sửa đổi vào năm 2008. Đặc biệt, điều tra hiện trường tội phạm điện tử của NIJ: Hướng dẫn cho người ứng cứu đầu tiên tập trung vào các quy trình tại hiện trường vụ án thực tế, chẳng hạn như bảo vệ và đánh giá hiện trường vụ án (tức là xác định các thiết bị có liên quan có bằng chứng kỹ thuật số tiềm năng), lập hồ sơ hiện trường vụ án, thu thập các thiết bị có liên quan và đóng gói, vận chuyển và cuối cùng là bảo vệ các thiết bị.

Các mô hình nêu trên dựa trên giả định rằng tất cả các giai đoạn đều được hoàn thành cho mỗi cuộc điều tra tội phạm và tội phạm mạng. Tuy nhiên, trên thực tế, điều này không phải lúc nào cũng đúng. Do khối lượng dữ liệu và các thiết bị kỹ thuật số thu thập, lưu trữ và chia sẻ dữ liệu đã tăng theo cấp số nhân, dẫn đến nhiều vụ án hình sự liên quan đến một số loại thiết bị kỹ thuật số hơn, nên việc tiến hành kiểm tra chuyên sâu từng thiết bị kỹ thuật số ngày càng được coi là không thực tế. Theo quan điểm đó, các mô hình quy trình giám định kỹ thuật số đã được phát triển để xem xét điều này. Ví dụ, Rogers và cộng sự al (2006) đã đề xuất Mô hình quy trình phân loại hiện trường pháp y mạng (CFFTPM), một mô hình quy trình pháp y kỹ thuật số tiếp cận tại chỗ hoặc tại hiện trường để cung cấp khả năng nhận dạng, phân tích và diễn giải bằng chứng kỹ thuật số trong một khung thời gian ngắn, mà không cần phải đưa hệ thống/phương tiện trở lại phòng thí nghiệm để kiểm tra chuyên sâu hoặc thu thập hình ảnh pháp y hoàn chỉnh. Dựa trên mô hình này, Casey, Ferraro và Nguyen (2009) đã đề xuất "ba cấp độ giám định pháp y" có thể được sử dụng tại hiện trường hoặc trong phòng thí nghiệm: khảo sát/phân loại giám định pháp y, giám định pháp y sơ bộ, giám định pháp y chuyên sâu.

Tính khả thi và tính liên quan của từng mô hình và các thành phần của nó vẫn tiếp tục được tranh luận cho đến ngày nay. Thực tế là mỗi quốc gia đều tuân theo các tiêu chuẩn, giao thức và quy trình pháp y kỹ thuật số riêng của mình. Tuy nhiên, sự khác biệt trong các quy trình lại là rào cản đối với hợp tác quốc tế trong các cuộc điều tra thực thi pháp luật.

### **1.3. Các tiêu chuẩn trong pháp chứng số**

Tổ chức Tiêu chuẩn hóa Quốc tế (ISO) và Ủy ban Kỹ thuật Điện Quốc tế (IEC) đã phát triển và công bố các tiêu chuẩn quốc tế nhằm hài hòa hóa các thực hành pháp chứng số giữa các quốc gia. Năm 2012, ISO và IEC đã ban hành tiêu chuẩn quốc tế ISO/IEC 27037 về xử lý bằng chứng số, bao gồm các hướng dẫn về nhận dạng, thu thập, trích xuất và bảo quản bằng chứng số. Bốn giai đoạn được đề xuất để xử lý bằng chứng kỹ thuật số như sau:

* Nhận dạng (Identification): Giai đoạn này tập trung vào việc nhận dạng các nguồn dữ liệu tiềm năng có thể chứa bằng chứng số liên quan đến sự cố hoặc vi phạm đang được điều tra. Các nguồn này có thể bao gồm máy tính, máy chủ, thiết bị di động, hệ thống mạng, thiết bị lưu trữ, và các ứng dụng phần mềm. Quá trình này bao gồm xác định và ghi chép các thông tin liên quan như loại thiết bị, vị trí, trạng thái hoạt động, và các chi tiết cụ thể để đảm bảo chuẩn bị tốt cho các giai đoạn tiếp theo.
* Thu thập (Collection): Giai đoạn thu thập là quá trình thu thập dữ liệu từ các nguồn đã được nhận dạng, mà không làm thay đổi, hủy hoại hoặc ảnh hưởng đến tính toàn vẹn của dữ liệu. Điều này có thể bao gồm việc chụp ảnh hệ thống, tạo bản sao đầy đủ của ổ đĩa, và thu thập dữ liệu từ bộ nhớ hệ thống. Việc thu thập phải được thực hiện bằng các công cụ pháp chứng số thích hợp và các quy trình đảm bảo tính hợp lệ và độ tin cậy của bằng chứng số.
* Trích xuất (Acquisition): Trích xuất là quá trình tạo ra các bản sao của dữ liệu đã được thu thập nhằm phân tích và điều tra sâu hơn. Dữ liệu cần phải được trích xuất một cách an toàn, đảm bảo rằng không có thay đổi nào được thực hiện trên dữ liệu gốc. Quá trình này bao gồm việc sao chép dữ liệu ra các thiết bị lưu trữ khác và tạo các bản sao lưu để phân tích tiếp theo mà không làm thay đổi dữ liệu gốc.
* Bảo quản (Preservation): Giai đoạn bảo quản liên quan đến việc duy trì và bảo vệ tính toàn vẹn của bằng chứng số đã được thu thập và trích xuất. Điều này bao gồm bảo vệ dữ liệu khỏi các thay đổi không mong muốn, mất mát hoặc hư hại thông qua các biện pháp bảo mật vật lý và điện tử. Cần ghi lại chi tiết về chuỗi cung ứng bằng chứng (chain of custody), các biện pháp bảo mật đã thực hiện, và các thông tin liên quan khác để đảm bảo tính hợp lệ pháp lý của bằng chứng.

ISO/IEC đã công bố các hướng dẫn khác về quy trình giám định kỹ thuật số bao gồm tính hợp lệ và độ tin cậy của các công cụ và phương pháp giám định kỹ thuật số (ISO/IEC 27041:2015, Hướng dẫn đảm bảo tính phù hợp và đầy đủ của các phương pháp điều tra sự cố) và các giai đoạn kiểm tra (hoặc phân tích) và diễn giải của quy trình giám định kỹ thuật số. ISO/IEC 27041:2015 cung cấp các hướng dẫn để đảm bảo rằng các công cụ và phương pháp điều tra sự cố và pháp chứng số được sử dụng là phù hợp và đầy đủ. Bốn thành phần chính của tiêu chuẩn này bao gồm việc xác định mục tiêu điều tra, đánh giá và kiểm tra các công cụ và phương pháp, cũng như duy trì và nâng cấp chúng. Tiêu chuẩn này giúp đảm bảo rằng các tổ chức sử dụng đúng các công cụ và phương pháp có khả năng thu thập và phân tích bằng chứng số một cách đáng tin cậy và hợp pháp.

ISO/IEC đã phát triển đồng thời ISO/IEC 27042:2015 trong cùng thời gian, nó không chỉ phát triển đồng bộ mà còn bổ sung cho nhau trong quy trình pháp chứng số, đảm bảo các cuộc điều tra được tiến hành đúng chuẩn và có giá trị pháp lý. ISO/IEC 27042:2015 cung cấp hướng dẫn chi tiết cho việc phân tích và diễn giải bằng chứng số trong các cuộc điều tra pháp chứng số. Tiêu chuẩn này đảm bảo rằng quá trình phân tích được thực hiện một cách có hệ thống, chính xác, và đáng tin cậy, giúp các tổ chức xác định rõ ràng các sự kiện xảy ra trong một sự cố và đưa ra các quyết định phù hợp dựa trên các bằng chứng đã phân tích.

ISO/IEC 27043:2015 được ra đời trong bối cảnh các mối đe dọa an ninh mạng ngày càng gia tăng và phức tạp. Các cuộc tấn công mạng và vi phạm dữ liệu đã trở nên phổ biến hơn, và các tổ chức phải đối mặt với áp lực lớn để bảo vệ thông tin nhạy cảm của mình. Tiêu chuẩn này đưa ra các nguyên tắc cơ bản để hướng dẫn quá trình điều tra sự cố an ninh thông tin. Các nguyên tắc bao gồm tính toàn vẹn, tính nhất quán, tính trung lập, và tính chính xác của quá trình điều tra. ISO/IEC 27043:2015 cung cấp một khung làm việc tổng thể cho các hoạt động điều tra sự cố an ninh thông tin. Tiêu chuẩn này đưa ra hướng dẫn từ chuẩn bị, phát hiện sự cố, thu thập và phân tích bằng chứng, đến báo cáo và cải tiến quy trình. Việc tuân thủ tiêu chuẩn này giúp các tổ chức tiến hành điều tra sự cố một cách hiệu quả, đáng tin cậy, và hợp pháp.

## **2. Tổng quan về hệ điều hành Linux**

### **2.1. Hệ điều hành Linux là gì?**

Linux là hệ điều hành mã nguồn mở, hệ điều hành được phát triển dựa trên Unix và được tạo bởi Linus Torvalds vào năm 1991. Tính đến thời điểm hiện tại Linux trở thành một trong những hệ điều hành phổ biến nhất trên thế giới.

Linux là hệ điều hành đa nền tảng, có thể sử dụng trên nhiều nền tảng, nhiều thiết bị khác nhau như: Điện thoại, máy tính, laptop, máy chơi game,... Đây là khả năng khá linh hoạt của hệ điều hành vì có thể phù hợp với nhiều môi trường.

Hệ điều hành Linux có một cộng đồng vô cùng mạnh mẽ và rộng lớn. Đây là cộng đồng có nhiều đóng góp vào việc phát triển nền tảng, hỗ trợ kiểm tra và cung cấp những hỗ trợ cho Linux. Điều này đảm bảo rằng Linux luôn được cải thiện và cập nhật liên tục các phiên bản mới nhất và có thể thực hiện vá lỗi ngay lập tức.



Hình 5. Hệ điều hành Linux.

Cấu trúc quan trọng của hệ điều hành Linux bao gồm 6 phần:

* Nhân hay còn biết đến là (Kernel): Đây là phần vô cùng quan trọng, có thể nói là đây là phần cốt lõi của hệ điều hành Linux. Phần nhân sẽ là nhiệm vụ chính là quản lý những tài nguyên cứng và cung cấp các dịch vụ khác. Cụ thể, Kernel giúp quản lý bộ nhớ, quản lý các tiến trình giúp giao tiếp với phần cứng và hỗ trợ các tính năng cơ bản của hệ thống.
* Shell: Tiếp đến, cấu trúc của hệ điều hành Linux sẽ bao gồm Shell. Có thể hiểu rằng Shell chính là giao diện của người dùng trong dòng lệnh của hệ điều hành. Shell cho phép người dùng được tương tác với các hệ thống bằng cách nhập các dòng lệnh và nhận những kết quả từ hệ thống.
* Hệ thống File (File System): Linux hỗ trợ rất nhiều loại hệ thống tệp, trong đó bao gồm: Ext4, Btrfs, XFS, và NTFS. Hệ thống lúc này sẽ quản lý các dữ liệu được lưu trữ, được tổ chức trên đĩa cứng hoặc các thiết bị lưu trữ khác.
* Tiến trình hay còn được gọi là (Process): Tiếp đến chính là phần tiến trình, Linux sẽ quản lý toàn bộ những tiến trình có sử dụng bảng tiến trình để có thể theo dõi và quản lý sự thay đổi của trạng thái của tiến trình. Cụ thể sẽ quản lý những tài nguyên, các lập lịch thực thi và quản lý bộ nhớ.
* Giao diện người dùng đồ họa (Graphical User Interface – GUI): Linux cung cấp một số môi trường đồ họa như GNOME, KDE, Xfce và LXDE. GUI cung cấp một giao diện đồ họa dễ sử dụng cho người dùng và cho phép họ tương tác với các ứng dụng. Ngoài ra, người dùng còn có thể sử dụng hệ thống thông qua các phần tử đồ họa như cửa sổ, biểu tượng và menu.
* Các tiện ích và ứng dụng: Ngoài ra, hệ điều hành Linux sẽ đi kèm với nhiều các tiện ích và tất cả các ứng dụng như: Trình duyệt web, chương trình trò chơi, bộ văn phòng, trình chơi đa phương tiện, trình chỉnh sửa ảnh,… Những ứng dụng đi kèm sẽ giúp người dùng thực hiện các nhiệm vụ khác nhau trên hệ điều hành Linux.

Về lợi ích của Linux:

* Tính ổn định cao: Linux được biết đến với tính ổn định vượt trội. Hệ điều hành này ít gặp sự cố và khá ít bị treo máy hoặc gặp lỗi so với các hệ điều hành khác.
* Bảo mật mạnh mẽ: Linux có tính bảo mật cao. Với cộng đồng nguồn mở rộng lớn, các lỗ hổng bảo mật thường được phát hiện và khắc phục nhanh chóng. Hơn nữa, hệ điều hành này cung cấp các công cụ và cơ chế bảo mật mạnh mẽ như hệ thống quyền hạn và phân quyền, giúp ngăn chặn các mối đe dọa từ phía bên ngoài.
* Tùy chỉnh và linh hoạt: Linux cho phép người dùng tùy chỉnh và điều chỉnh hệ thống theo nhu cầu riêng. Bạn có thể thay đổi giao diện người dùng, cấu hình hệ thống, và lựa chọn các thành phần phần mềm khác nhau để tạo ra một trải nghiệm độc đáo và phù hợp với yêu cầu cá nhân.
* Hiệu suất cao: Linux thường cho hiệu suất tốt hơn so với các hệ điều hành khác. Nó sử dụng tài nguyên phần cứng một cách hiệu quả và có khả năng tối ưu hóa hiệu năng hệ thống. Điều này đặc biệt hữu ích đối với các máy tính có cấu hình thấp hoặc máy chủ đòi hỏi xử lý tải cao.
* Hỗ trợ phong phú từ cộng đồng: Linux có một cộng đồng nguồn mở lớn và nhiệt tình. Người dùng Linux có thể tận dụng sự hỗ trợ đa dạng từ cộng đồng, gồm các diễn đàn, mailing list, trang web hướng dẫn, và tài liệu phong phú. Điều này giúp người dùng giải quyết vấn đề, tìm kiếm thông tin và chia sẻ kinh nghiệm.
* Chi phí thấp hoặc miễn phí: Linux là một hệ điều hành mã nguồn mở, nghĩa là bạn có thể tải về, cài đặt và sử dụng miễn phí. Điều này giúp tiết kiệm chi phí phần mềm và giúp các tổ chức và cá nhân có thể tiết kiệm tài nguyên và tập trung vào các mục tiêu khác.
* Đa nền tảng: Linux có khả năng chạy trên nhiều nền tảng phần cứng khác nhau, từ máy tính cá nhân, máy chủ đến thiết bị di động và nhúng. Điều này mang lại sự linh hoạt và tương thích rộng rãi cho người dùng.

### **2.2. Ưu, nhược điểm của hệ điều hành Linux**

**Ưu điểm của hệ điều hành Linux**

* Bộ công cụ mạnh mẽ: Linux đi kèm với hàng trăm công cụ chuyên về bảo mật và kiểm thử xâm nhập, giúp người dùng thực hiện các tác vụ liên quan đến an ninh mạng một cách hiệu quả.
* Cập nhật thường xuyên: Các công cụ và tài nguyên trong Kali Linux được cập nhật đều đặn, đảm bảo rằng người dùng luôn có truy cập đến công nghệ mới nhất và các lỗ hổng bảo mật được vá kịp thời.
* Tích hợp công cụ đa dạng: Linux tích hợp nhiều công cụ từ nhiều lĩnh vực khác nhau của an ninh mạng, bao gồm cả quét mạng, tấn công, phục hồi dữ liệu, và nhiều công cụ khác, giúp người dùng tiết kiệm thời gian khi tìm kiếm và cài đặt các công cụ riêng lẻ.
* Hỗ trợ nhiều nền tảng: Linux có thể chạy trên nhiều loại thiết bị và nền tảng, từ máy tính cá nhân đến máy chủ và máy ảo, cung cấp sự linh hoạt cho người dùng.
* Phổ biến trong cộng đồng an ninh mạng: Vì Linux được sử dụng rộng rãi trong cộng đồng chuyên gia an ninh mạng, nó trở thành một công cụ quen thuộc và được chấp nhận trong ngành.
* Hỗ trợ chế độ Live và cài đặt: Linux có thể chạy từ chế độ Live CD mà không cần cài đặt, cho phép người dùng kiểm tra các tính năng mà không làm thay đổi hệ thống của họ. Nó cũng có chế độ cài đặt trực tiếp, giúp người dùng có thể chuyển đổi từ chế độ thử nghiệm sang cài đặt chính thức khi họ quyết định sử dụng Kali Linux một cách liên tục.

**Nhược điểm của hệ điều hành Linux**

* Không dành cho người dùng thông thường: Linux không được thiết kế để sử dụng hàng ngày cho các công việc thông thường. Giao diện và trải nghiệm người dùng không được tối ưu hóa cho việc làm việc thông thường, điều này có thể làm cho người dùng không chuyên sẽ gặp khó khăn khi sử dụng.
* Yêu cầu kiến thức chuyên sâu: Để tận dụng hết các công cụ và chức năng của Linux, người dùng cần có kiến thức sâu rộng về mạng và bảo mật. Điều này đồng nghĩa với việc người dùng không chuyên sẽ cần mất nhiều thời gian và nỗ lực để học và hiểu các công cụ này.
* Pháp lý: Linux chủ yếu được sử dụng cho các hoạt động kiểm thử bảo mật hợp pháp và giáo dục, nhưng cũng có thể bị lạm dụng cho các hoạt động tấn công mạng không đạo đức và bất hợp pháp. Sử dụng Kali Linux một cách không đúng cách có thể dẫn đến các hậu quả pháp lý.
* Bảo mật không chặt chẽ: Linux được thiết kế để thử nghiệm bảo mật và kiểm tra hệ thống, điều này có nghĩa là các thiết lập an toàn không được cấu hình mặc định một cách chặt chẽ. Điều này đòi hỏi người dùng phải tự tìm hiểu và cấu hình bảo mật một cách cẩn thận để tránh lỗ hổng bảo mật không mong muốn.

|  |  |  |
| --- | --- | --- |
|  | Linux | Windows |
| Yêu cầu cấu hình | Không cần cấu hình máy quá mạnh để sử dụng. | Cần cấu hình cao hơn. |
| Giao diện sử dụng | Không thân thiện với người dùng. Đòi hỏi người dùng phải hiểu biết về công nghệ.  Thường xuyên dùng lệnh để tương tác với thiết bị. | Thân thiện với người dùng, có thể học dễ dàng.  Thường sử dụng các phần mềm dưới dạng giao diện đồ họa. |
| Khả năng tùy biến | Có độ tùy biến cao, hỗ trợ nhiều môi trường GUI (Giao diện đồ họa người dùng) | Độ tùy biến không cao bằng. |
| Ứng dụng | Số lượng ứng dụng miễn phí nhiều. | Có nhiều ứng dụng nhưng không có nhiều ứng dụng miễn phí. |
| Bảo mật | Số lượng malware hay virus rất ít. Thế nên nó sẽ an toàn hơn. | Số lượng malware hay virus rất nhiều, thường là mục tiêu nhắm đến của tội phạm. |
| Khả năng vá sửa lỗi | Tốc độ vá lỗi về bảo mật thường nhanh hơn nhờ cộng đồng hỗ trợ lớn. | Thường mất thời gian hơn để tung ra bản vá. |
| Đối tượng sử dụng | Số lượng người sử dụng ít hơn. Phổ biến là các lập trình viên. | Số lượng người dùng đông đảo.  Phổ biến nhất là người dùng thông thường. |

### **2.3** **Tổng quan về hệ thống Linux trong phân tích pháp chứng số.**

Kiến trúc của hệ điều hành Linux thường bao gồm 3 phần chính: Kernel (nhân), Shell, Application. Phần nhân là phần quan trọng nhất, được ví như là trái tim của Linux. Kernel chứa các module, thư viện để quản lý và giao tiếp với các phần cứng và các ứng dụng. Shell là một thành phần cao hơn ở bên trong hệ điều hành, có thể tưởng tượng nó là một chương trình, chương trình này cho phép ta thực thi câu lệnh, viết câu lệnh. Các câu lệnh đó được thực thi trên shell. Bên trong Linux có nhiều dạng shell, phổ biến là bash, zsh, fish, ksh, csh,…Trong đó bash là shell mặc định phổ biến nhất trên các phiên bản Linux hiện tại. Ta vẫn có thể cài đặt thêm các shell khác. Còn về Application là các chương trình ứng dụng được thiết kế để thực hiện nhiệm vụ cụ thể hoặc cung cấp dịch vụ cho người dùng.

**2.3.1. Cấu trúc và hệ thống file trong hệ điều hành Linux.**

File system hiểu đợn giản đó là cách tập tin được tổ chức trên ổ đĩa (ssd, hdd), dùng để quản lý cách mà dữ liệu được đọc và lưu trữ trên thiết bị. File system có thể dùng để chỉ một phân vùng hoặc một ổ đĩa dùng để lưu trữ các tập tin và hệ thống tập tin.

Các loại filesystem được Linux hỗ trợ:

* Filesystem cơ bản: EXT2, EXT3, EXT4, XFS, Btrfs, JFS, NTFS,…
* Filesystem dành cho dạng lưu trữ Flash: thẻ nhớ,…
* Filesystem dành cho hệ cơ sở dữ liệu
* Filesystem mục đích đặc biệt: procfs, sysfs, tmpfs, squashfs, debugfs,…

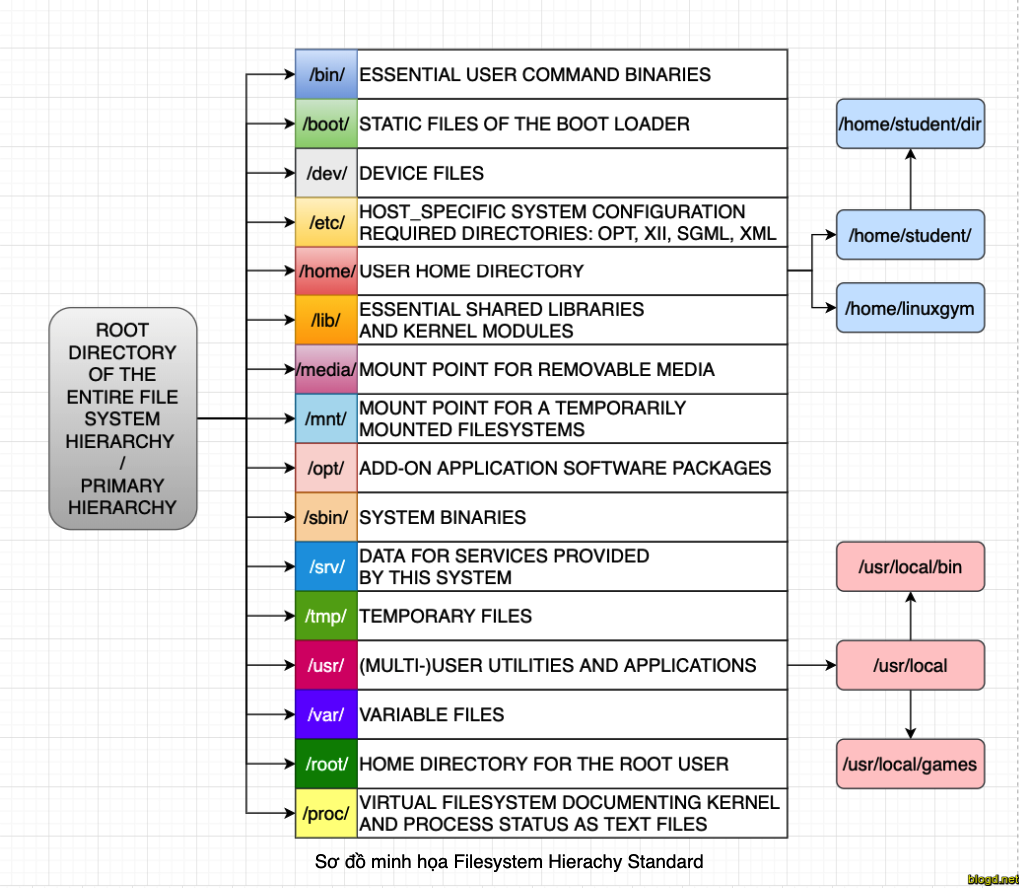
Về hệ thống file trên hệ điều hành Linux có sự khác biệt lớn so với hệ điều hành Windows cách tổ chức, quản lý hệ thống, phân vùng ổ đĩa cũng như hệ thống file được sử dụng và còn nhiều sự khác biệt khác.

A screenshot of a computer program

Description automatically generated

Hình 6. Khác biệt về cấu trúc của Windows và Linux

Cấu trúc hệ thống file trên hệ điều hành Linux có dạng cây và nơi bắt nguồn mọi thứ đều bắt nguồn từ thư mục gốc (/).



Hình 7. Cấu trúc hệ thống phân vùng Linux.

Tất cả file và thư mục đều nằm trong này. Trong thư mục (/) có các thư mục thông thường được nhìn thấy như:

/*bin*: đây là nơi chứa các tập tin thực thi nhị phân. Các chương trình này rất cần thiết cho hoạt động của hệ thống, đặc biệt là trong môi trường khởi động hoặc chế độ sửa lỗi. Tất cả user trên hệ thống nằm tại thư mục này đều có thể sử dụng lệnh.

Ví dụ: *ps, ls, ping, grep, cp.*

/s*bin*: cũng tương tự như /bin, nó cũng chứa tập tin thực thi nhị phân. Lệnh Linux nằm trong thư mục này được sử dụng bởi Admin hệ thống nhằm mục đích duy trì hệ thống, nó yêu cầu quyền root để chạy. Các chương trình trong thư mục này thường được sử dụng để cấu hình hoặc quản lý hệ thống.

Ví dụ: *iptables, reboot, fdisk, ifconfig, swapon.*

/*ect*: chứa cấu hình các tập tin cấu hình của hệ thống, các tập tin lệnh để khởi động các dịch vụ của hệ thống. Ngoài ra nó còn chứa shell scripts  và shutdown, sử dụng để chạy, ngừng các chương trình cá nhân. Các file trong thư mục này không phải là các file thực thi mà là các file văn bản chứa thông tin cấu hình.

Ví dụ: */etc/resolv.conf, /etc/logrotate.conf, /etc/passwd*.

/*dev*: đây là thư mục chứa các file đại diện cho thiết bị phần cứng, nó chứa các tập tin để nhận biết cho các thiết bị của hệ thống. Linux coi mọi thiết bị như là một file, do đó các file trong /dev là các đại diện cho phần cứng như ổ đĩa, thiết bị âm thanh, cổng mạng.

Ví dụ: */dev/tty1, /dev/usbmon0*

/*proc*: đây là hệ thống file giả lập, chứa thông tin về các tiến trình đang chạy và thông tin hệ thống. Các file trong đây không phải là thật mà là giao diện để truy cập thông tin kernel và tiến trình.

Ví dụ: */proc/cpuinfo, /proc/meminfo*

/var: chứa các file có thể thay đổi kích thước và nội dung khi hệ thống hoạt động, như file log, spool (thư chờ), và dữ liệu tạm thời. Nó lưu lại tập tin ghi các số liệu, bao gồm: hệ thống tập tin log (/*var*/*log*), các gói và các file dữ liệu (/*var*/*lib*), email (/*var*/*mail*), print queues (/*var/spool*); lock files (/*var/lock*); các file tạm thời cần khi reboot (/*var*/*tmp*)

/*tmp*: thư mục chứa các tập tin tạm thời được tạo bởi hệ thống và *user*. Các tập tin tạo thư mục này sẽ bị xóa khi hệ thống khởi động lại.

/*usr*: đây là thư mục lớn chứa các chương trình người dùng, các file thư viện, tài liệu, và các mã nguồn cần thiết cho các chương trình này.

/*home*: đây là thư mục chính lưu trữ tất cả các tập tin cá nhân của tất cả *user*.

/*boot*: thư mục này chứa các file cần thiết cho quá trình khởi động hệ thống, bao gồm kernel và bootloader.

/*lib*: chứa các thư viện dùng chung cho các chương trình trong /*bin* và /*sbin*. Thư viện này tương tự như các thư viện DLL trong Windows.

/*opt*: thư mục này chứa các ứng dụng tùy chọn được cài đặt vào hệ thống, thường là các ứng dụng không phải của hệ thống cơ bản. Đây là nơi lưu trữ các phần mềm lớn hoặc các phần mềm cài đặt từ nguồn bên ngoài.

/*mnt* và /*media*: thư mục này được sử dụng để gắn kết (*mount*) các thiết bị lưu trữ ngoài như ổ USB, CD-ROM. Khi bạn cắm thiết bị vào, hệ thống tự động gắn kết nó vào đây.

/*srv*: thư mục này chứa dữ liệu phục vụ cho các dịch vụ mạng đang chạy trên hệ thống, chẳng hạn như máy chủ web, máy chủ FTP.

Linux là một hệ điều hành hỗ trợ nhiều loại file hệ thống nhất hiện nay. Để có thể hiểu rõ hơn về hệ thống file hiện nay, ta cần biết về *journaling*, nó là một tính năng được sử dụng trong tất cả loại file hệ thống ở thời điểm hiện tại. Journaling là quá trình ghi lại sự thay đổi của metadata (siêu dữ liệu) hoặc cả dữ liệu của người dùng vào một khu vực đặc biệt gọi là journal (nhật ký). Journal giúp hệ thống tệp ghi nhận trước khi thực hiện các thao tác trên dữ liệu chính, để trong trường hợp gặp sự cố, các thay đổi có thể được khôi phục dễ dàng từ journal. Nói đơn giản hơn thì journaling giúp tăng tính ổn định và bảo toàn dữ liệu sau khi hệ thống xảy ra sự cố. Cách hoạt động của nó cũng tương đối đơn giản, đầu tiên, khi có sự thay đổi xảy ra (có thể là tạo file, xóa file, di chuyển file hoặc cập nhật các quyền truy cập,...) thì hệ thống tệp sẽ ghi lại các thay đổi này vào journal. Sau khi đã ghi vào journal thì hệ thống tệp mới thực hiện thay đổi trên dữ liệu chính, và sự thay đổi được hoàn thành trên dữ liệu chính thì mục ghi trong journal sẽ được đánh dấu là hoàn thành hoặc bị xóa. Nếu hệ thống xảy sự cố trong lúc đang cập nhật dữ liệu thì journal có thể khôi phục lại dữ liệu chính bằng cách đọc các thay đổi chưa hoàn thành và thực hiện lại chúng. Tuy nhiên tính năng này cũng có nhược điểm:

* Tăng chi phí ghi dữ liệu: Quá trình ghi dữ liệu vào journal và sau đó là dữ liệu chính làm tăng số lượng thao tác ghi, có thể làm giảm hiệu suất của ổ đĩa.
* Tiêu tốn dung lượng lưu trữ: Journal chiếm một phần dung lượng của ổ đĩa.
* Tiêu hao tài nguyên hệ thống: Quá trình journaling đòi hỏi thêm năng lực xử lý và bộ nhớ, đặc biệt đối với chế độ full journal.

Đối với các hệ thống yêu cầu tính ổn định và độ tin cậy cao, khả năng mất dữ liệu phải được giảm xuống mức tối thiểu thì journaling là một tính năng đáng để đầu tư.

Thành phần cơ bản thường có của hệ thống file trong Linux bao gồm: Superblock, Inode, Data Blocks, Directory Entry, Block Group, Free Space Bitmap. Ngoài ra ta có thể cho journaling vào mục này bởi các hệ thống file ngày nay đều có tính năng journaling. Các thành phần liên kết với nhau trên hệ thống tếp như sau:

* Khi ta tạo một file mới, hệ điều hành sẽ tìm một inode trống, gán các thuộc tính cho inode đó, và sau đó tìm các block trống để lưu trữ dữ liệu của file.
* Khi ta truy cập một file, hệ điều hành sẽ tìm inode của file đó dựa trên tên file và đường dẫn, sau đó sử dụng thông tin trong inode để tìm các block chứa dữ liệu của file.
* Khi ta xóa một file, inode của file đó được đánh dấu là "đã xóa" và các block chứa dữ liệu của file được trả về danh sách các block trống.

Superblock là một vùng chứa thông tin quan trọng về hệ thống tệp như số lượng block, inode, thông tin về các block dữ liệu, kích thước, trạng thái, cấu trúc của hệ thống tệp, nếu nó bị hỏng thì nó có thể dẫn đến hỏng hệ thống tệp. Superblock nằm ở đầu hệ thống tệp, mỗi hệ thống tệp có ít nhất một Superblock và có thể có các bảng sao lưu để dự phòng khi Superblock chính bị hỏng.

Inode (Index node) là cấu trúc dữ liệu trong hệ thống tệp dùng để lưu trữ thông tin về một tệp, chẳng hạn như loại tệp, kích thước tệp, thời gian tạo, quyền truy cập, chủ sở hữu, nhóm sở hữu, và vị trí trên đĩa của các block chứa dữ liệu tệp. Mỗi tệp và thư mục trên hệ thống tệp đều  có 1 Inode duy nhất, hệ thống sử dụng nó để tìm ra dữ liệu thật của tệp trên ổ đĩa. Nó không chứa tệp hay dữ liệu của tệp mà chỉ chứa metadata như kích thước, quyền truy cập, … và các chỉ mục trỏ đến vị trí thực sự của tệp trong các block trên ổ đĩa.

Data Blocks chứa dữ liệu thực tế của tệp. Mỗi tệp được phân bổ vào một hoặc nhiều data block. Hệ thống tệp chia ổ đĩa ra thành các block có kích thước cố định (thường là 4K hoặc 8K tùy hệ thống) và dữ liệu được phân bổ vào các block này, tùy thuộc vào kích thước tệp sẽ chiếm số lượng block khác nhau. Cấu trúc của block gồm 2 phần là Direct block và Indirect block. Direct block chứa dự liệu trực tiếp của tệp, entry trong Inode sẽ trỏ trực tiếp đến một block trên đĩa còn Indirect block là khi một file quá lớn để có thể chứa tất cả các địa chỉ block trực tiếp trong Inode, hệ thống sẽ sử dụng Indirect block, mỗi entry trong Inode sẽ trỏ đến một block chứa danh sách các địa chỉ block dữ liệu.

Directory Entry cho phép một hệ điều hành tìm kiếm một file hoặc thư mục trong hệ thống tệp. Mỗi thư mục là một file đặc biệt chứa danh sách các file và thư mục con. Mỗi mục trong danh sách này được gọi là một Directory Entry. Khi ta tạo một tệp trong thư mục, hệ thống tệp sẽ thêm một mục vào thư mục đó, với tên tệp và inode tương ứng. Các thư mục không lưu trữ nội dung tệp mà chỉ lưu trữ danh sách các tệp con (dưới dạng các cặp khóa-giá trị, với khóa là tên tệp và giá trị là Inode trỏ tới thông tin của tệp). Một số hệ thống tệp không dùng directory entry mà sử dụng b-tree, nó là một cách tiếp cận hiệu quả hơn, đặc biệt đối với các hệ thống tệp lớn và cần phải thực hiện các thao tác tìm kiếm, chèn, xóa một cách nhanh chóng, được sử dụng để tổ chức các directory entry.

Block Group giúp phân tán các hoạt động đọc/ghi, cải thiện hiệu suất và độ tin cậy của hệ thống tệp. Trong các hệ thống tệp như ext4, dữ liệu được tổ chức thành các block group để tối ưu hóa hiệu suất truy xuất. Mỗi block group có một superblock riêng (superblock group), bảng inode riêng và bảng bitmap riêng.

Free Space Bitmap giúp hệ điều hành quản lý các block trống, tìm kiếm các block trống để lưu trữ dữ liệu mới, nói đơn giản là nó giúp hệ thống tệp nhanh chóng tìm kiếm và phân bổ không gian trống cho dữ liệu mới. Mỗi bit trong bitmap đại diện cho một block trên ổ đĩa; 0 nghĩa là block đó trống, 1 nghĩa là block đó đã được sử dụng.

Một trong những hệ thống tệp Linux lâu đời và phổ biến nhất là hệ thống tệp mở rộng, hay ext. Mọi bản phân phối Linux hiện đại đều hỗ trợ ext4, và nhiều bản trong số đó chỉ định nó là hệ thống tệp mặc định trong quá trình cài đặt. Do sự phổ biến của ext (2, 3 và 4), nhiều công cụ pháp lý thương mại hỗ trợ ext4. TSK (và Autopsy) hỗ trợ nó, và nhiều công cụ khắc phục sự cố, gỡ lỗi và phục hồi dữ liệu ext4 khác cũng có sẵn. Với ext2 không có khả năng journaling (ghi nhật ký), vì vậy mà nó giúp hệ thống tệp nhanh hơn nhưng đó cũng chính là nhược điểm khi hệ thống gặp sự cố thì có có thể mất dữ liệu. Nó là một hệ thống tệp mặc định của Linux trong một khoảng thời gian dài trước khi ext3 xuất hiện, hiện nay ext2 chỉ sử dụng chủ yếu cho các thiết bị có yêu cầu ghi nhanh, ít khả năng xảy ra lỗi hoặc không cần journaling như ổ USB. Còn ext3 ra mắt sau ext2 8 năm, ext3 thêm tính năng journaling để tăng cường độ ổn định của hệ thống tệp và giảm thiểu rủi ro khi xảy ra mất dữ liệu. Nó cải thiện được độ tin cậy và giảm thời gian khắc phục sau sự cố, tuy nhiên nó lại hạn chế về kích thước file và hệ thống tệp so với ext4. Như đã đề cập thì ext4 hiện tại được hỗ trợ rất nhiều, được nâng cấp hơn và có nhiều ưu điểm hơn so với 2 thế hệ trước nó. Cho phép hệ thống tệp lớn hơn lên tới 1 exabyte và kích thước file tối đa có thể lên tới 16 terabyte, có thêm tính năng chống phân mảnh. Ext4 sử dụng extents thay vì block mapping giúp quản lý không gian hiệu quả hơn khi lưu trữ các file lớn bởi nó giảm thiểu khả năng phân mảnh và tăng tốc độ xử lý, ngoài ra tính năng journaling cũng được cải tiến để bảo vệ dữ liệu tốt hơn và thời gian khôi phục dữ liệu giảm. Nó không phù hợp với các thiết bị có dung lượng lưu trữ hạn chế vì yêu cầu tài nguyên cao hơn ext2 và ext3. EXT4 là hệ thống tệp phổ biến trên Linux với ưu điểm đơn giản, hiệu suất cao, và tương thích ngược với EXT2/EXT3. Nó hỗ trợ kích thước file lên đến 16TB và hệ thống tệp đến 1EB, tích hợp tính năng journaling để đảm bảo tính toàn vẹn dữ liệu khi xảy ra sự cố. Extents thay thế block mapping, cải thiện hiệu quả lưu trữ và hiệu suất truy cập file lớn. EXT4 có cơ chế delayed allocation, giúp giảm phân mảnh và tối ưu hóa ghi dữ liệu. Mặc dù không có các tính năng như snapshot hay RAID tích hợp, EXT4 là lựa chọn ổn định cho máy trạm và máy chủ nhỏ.

XFS được tích hợp vào Linux từ năm 2001, nổi bật nhờ khả năng xử lý tệp lớn, độ tin cậy cao, và hỗ trợ quản lý khối lượng công việc nặng trong các môi trường hiện đại như máy chủ, lưu trữ mạng (NAS), và cơ sở dữ liệu. Cấu trúc dữ liệu có sự khác biệt với ext, thay vì sử dụng directory entry thì XFS sử dụng B-tree. B-tree với cấu trúc cây cân bằng và tối ưu hóa hơn directory entry, chính vì vậy mà tốc độ tìm kiếm nhanh hơn. Tuy nhiên, việc đó chỉ thích hợp đối với các hệ thống phức tạp hoặc thư mục có nhiều mục nhập, nó sẽ duy trì hiệu suất cao khi dữ liệu lớn và cần truy xuất nhanh. Còn đối với các hệ thống đơn giản để triển khai và dữ liệu nhỏ, các thư mục có số lượng tệp hoặc thư mục con nhỏ thì directory entry sẽ phù hợp hơn. Nó sử dụng extent thay cho block mapping, điều này tương tự với ext4 giúp quản lý không gian hiệu quả hơn. XFS có một số hạn chế như không thể thu nhỏ hệ thống tệp (shrink), một khi đã phân bổ rồi thì ta không thể giảm kích thước của nó lại, như đã nói thì nó không phù hợp cho tệp nhỏ, về journaling thì chỉ dùng cho metadata, người dùng sẽ không được ghi vào journal, chính vì vậy mà nó sẽ có rủi ro về mất dữ liệu nếu bị mất điện đột ngột.

BTRFS (B-tree file system) được phát triển từ năm 2007, nó là một hệ thống hiện đại tập trung quản lý dữ liệu tiên tiến, tính linh hoạt và hiệu suất. Đúng với tên của nó thì nó sử dụng cấu trúc dữ liệu B-tree để quản lý metadata giúp nâng cao hiệu suất tìm kiếm và cập nhật dữ liệu nhanh chóng. Nó được thiết kế để quản lý các khối lượng dữ liệu lớn một cách hiệu quả, có thể tự động điều chỉnh kích thước và cấu trúc để đáp ứng nhu cầu thay đổi của dữ liệu, với nhiều tính năng giúp bảo vệ dữ liệu như checksum, snapshot, quản lý lỗi, … Với nhiều tính năng như vậy thì nó đang được sử dụng tăng dần, thường dùng cho các máy chủ, tuy nhiên, về hiệu suất thì nó kém hơn EXT4 và XFS khi xử lý tệp rất lớn hoặc khối lượng công việc I/O nặng, một số tính năng vẫn chưa ổn định hoàn toàn, yêu cầu tài nguyên cũng cao hơn, và đặc biệt là nó phức tạp hơn, yêu cầu người dùng phải có kiến thức sâu về hệ thống tệp để có thể khai thác hết tiềm năng của nó. BTRFS chưa thực sự phổ biến và dễ sử dụng bằng ext4, không phải bản phân phối linux nào cũng tích hợp nó vào mặc định. BTRFS là một hệ thống tệp tiên tiến và linh hoạt, lý tưởng cho các môi trường hiện đại yêu cầu quản lý dữ liệu lớn và phức tạp. Tuy nhiên, để sử dụng hiệu quả, cần cân nhắc các hạn chế về hiệu suất và sự ổn định, đặc biệt khi so sánh với các hệ thống tệp như ext4 hoặc XFS. BTRFS phù hợp với các hệ thống máy chủ lưu trữ, ảo hóa và backup hơn là các ứng dụng máy trạm thông thường. BTRFS là hệ thống tệp hiện đại, linh hoạt, tích hợp tính năng Copy-On-Write (CoW), hỗ trợ snapshot, RAID (0, 1, 10), và quản lý dữ liệu lớn. Nó cung cấp deduplication, compression, và khả năng mở rộng cao, đồng thời đảm bảo tính toàn vẹn dữ liệu với checksum CRC32. Với thiết kế hiện đại, BTRFS phù hợp cho các môi trường yêu cầu quản lý dữ liệu phức tạp hoặc lưu trữ linh hoạt, nhưng hiệu suất của nó có thể thấp hơn EXT4 hoặc XFS trong một số trường hợp.

ZFS là một hệ thống tệp và trình quản lý khối lượng lưu trữ (volume manager) được phát triển bởi Sun Microsystems, ra mắt lần đầu vào năm 2001. Đây là hệ thống tệp tiên tiến, mạnh mẽ, được thiết kế để quản lý khối lượng dữ liệu lớn, tính toàn vẹn cao và tính năng vượt trội so với nhiều hệ thống tệp khác. Nó là một hệ thống tệp tiên tiến với các đặc điểm nổi bật như Copy-On-Write (CoW) giúp bảo vệ dữ liệu gốc và hỗ trợ snapshot/clone hiệu quả. Tích hợp RAIDZ1/2/3 cung cấp khả năng chịu lỗi cao mà không cần công cụ bên ngoài. ZFS đảm bảo tính toàn vẹn dữ liệu với checksum SHA-256 và cơ chế Self-Healing tự động sửa lỗi khi phát hiện hỏng hóc. Hệ thống này hỗ trợ nén dữ liệu (compression) và loại bỏ dữ liệu trùng lặp (deduplication) để tối ưu lưu trữ. Với khả năng mở rộng lên đến zettabyte, dynamic storage pooling linh hoạt, và hiệu suất cao trong quản lý dữ liệu lớn, ZFS phù hợp cho các ứng dụng lưu trữ doanh nghiệp, NAS và backup dữ liệu. Tuy nhiên, nó yêu cầu tài nguyên lớn, đặc biệt là RAM, và có cấu hình phức tạp so với các hệ thống tệp như EXT4 hay BTRFS.

Sự phổ biến của các loại hệ thống tập tin trên như sau:

* EXT4: Phổ biến nhất trên mọi loại hệ thống, đặc biệt máy tính cá nhân và máy chủ nhỏ.
* XFS: Phổ biến thứ hai, tập trung vào máy chủ doanh nghiệp và cơ sở dữ liệu lớn.
* BTRFS: Đang phát triển, phổ biến trong môi trường lưu trữ hiện đại với tính năng snapshot.
* ZFS: Chuyên dụng, phổ biến trong doanh nghiệp và hệ thống NAS, ít thấy trong môi trường thông thường.

**2.3.2. Hệ thống phân vùng trong Linux**

MBR (Master Boot Record) và GPT (GUID Partition Table) là hai chuẩn chính dùng để quản lý phân vùng trên ổ cứng. Cả hai đều cung cấp thông tin cần thiết về cấu trúc phân vùng, nhưng chúng khác nhau về cách tổ chức và khả năng hỗ trợ phần cứng hiện đại.

Được giới thiệu vào năm 1983, MBR chứa thông tin về cách phân vùng logical và các hệ thống tệp tin được sắp xếp trên đĩa và chứa Bootloader (bộ tải khởi động) được cài đặt trên hệ điều hành. MBR hỗ trợ tối đa bốn phân vùng chính nếu muốn nhiều phân vùng hơn bạn phải thực hiện chuyển đổi 1 trong những phân vùng chính thành "extended partition" (phân vùng mở rộng) và tạo phân vùng Logical bên trong phân vùng đó, MBR sử dụng 32 bit để lưu trữ các địa chỉ khối và đối với các ổ đĩa cứng có các sector 512 byte, MBR chỉ xử lý tối đa 2TB. Trên ổ đĩa MBR, dữ liệu phân vùng và dữ liệu khởi động được lưu trữ ở một vị trí. Nếu dữ liệu này bị ghi đè hoặc bị hỏng thì coi như bạn gặp rắc rối rồi đó. MBR là nơi lưu trữ thông tin về cách chia các phân vùng trên ổ cứng, giúp hệ thống xác định cách quản lý các vùng dữ liệu khác nhau. Nó cũng cung cấp thông tin về hệ điều hành để có thể tải vào cho quá trình khởi động hệ thống. MBR chứa các chương trình xác định phân vùng nào trên ổ cứng sẽ được sử dụng để khởi động hệ thống. Nếu không có MBR, hệ thống sẽ không thể khởi động.

**Ưu điểm của MBR:**

* **Khả năng tương thích:**MBR thích khả năng tương thích rộng với các hệ thống, hệ điều hành và tiện ích đĩa cũ hơn. Nó hoạt động tốt với legacy BIOS (Basic Input/Output System) thường thấy trong các máy tính cũ.
* **Làm quen:**MBR đã là sơ đồ phân vùng tiêu chuẩn trong một thời gian dài và nhiều quản trị viên và người dùng hệ thống đã quen thuộc với các quy trình thiết lập và quản lý của nó.
* **Đơn giản và nhẹ:**MBR được coi là đơn giản và nhẹ vì nó chiếm một lượng không gian nhỏ (512 byte) trên ổ lưu trữ và có cấu trúc đơn giản, giúp dễ dàng thao tác và sửa chữa.

**Nhược điểm của MBR:**

* **Kích thước phân vùng hạn chế:**MBR giới hạn đáng kể kích thước phân vùng, hỗ trợ ổ đĩa lên tới 2TB. Nếu có ổ lưu trữ lớn hơn, MBR không thể sử dụng hết dung lượng của nó.
* **Giới hạn phân vùng:**MBR có giới hạn tối đa là bốn chính hoặc ba phân vùng chính và một phân vùng mở rộng. Hạn chế này có thể là một trở ngại nếu yêu cầu nhiều phân vùng hơn cho các mục đích khác nhau.
* **Thiếu dư thừa dữ liệu:**MBR không bao gồm các cơ chế tích hợp để backup dữ liệu hoặc kiểm tra tính toàn vẹn. Nếu bảng MBR hoặc bảng phân vùng bị hỏng, nó có thể dẫn đến mất dữ liệu.

GPT là một chuẩn mới và chuẩn này đang dần thay thế chuẩn MBR. Chuẩn GPT có nhiều ưu điểm và lợi thế hơn chuẩn MBR. Tuy nhiên chuẩn MBR có tính tương thích cao và trong một số trường hợp thì chuẩn này cực kỳ quan trọng và cần thiết.

GPT viết tắt GUID (Globally Unique Identifier ) Partition Table là chuẩn mới đang dần thay thế MBR. GPT liên quan với UEFI - UEFI thay thế cho BIOS, UEFI có giao diện và tính năng hiện đại hơn , và GPT cũng thay thế các hệ thống phân vùng MBR xa xưa bằng các tính năng, giao diện hiện đại hơn.

**Ưu điểm của GPT:**

* **Kích thước phân vùng lớn:**GPT hỗ trợ kích thước phân vùng lớn đáng kể, cho phép các ổ đĩa có kích thước lên tới 9,4 ZB. Thường được sử dụng trong các hệ thống hiện đại.
* **Số phân vùng tăng:**GPT hỗ trợ tới 128 phân vùng chính, số lượng cao hơn nhiều so với MBR. Giới hạn phân vùng hầu như không giới hạn, tùy thuộc vào hệ điều hành và kích thước đĩa. Tính linh hoạt này có lợi cho các hệ thống và kịch bản đa khởi động đòi hỏi nhiều phân vùng.
* **Dự phòng dữ liệu và tính toàn vẹn:**GPT bao gồm các bản sao dự phòng của các cấu trúc dữ liệu quan trọng trải rộng trên ổ lưu trữ. Sự dư thừa này giúp tăng cường tính toàn vẹn dữ liệu và làm cho phục hồi từ các lỗi đĩa mạnh mẽ hơn. Ngoài ra, GPT sử dụng tổng kiểm tra CRC32 để xác minh tính toàn vẹn của các cấu trúc dữ liệu, cung cấp một lớp bảo vệ bổ sung.

**Nhược điểm của GPT**

* **Khả năng tương thích với các hệ thống cũ:**GPT yêu cầu một hệ thống với phần sụn UEFI để sử dụng đầy đủ các tính năng của nó. Mặc dù các hệ thống hiện đại chủ yếu hỗ trợ UEFI, các hệ thống cũ sử dụng BIOS cũ có thể bị hạn chế hoặc không hỗ trợ cho GPT.
* **Hỗ trợ hệ điều hành và phần mềm:**Mặc dù hầu hết các hệ điều hành hiện đại đều hỗ trợ GPT, một số phiên bản cũ hơn, như Windows XP 32-bit hoặc Windows Server 2003, có thể có những hạn chế. Đảm bảo rằng hệ điều hành dự định cài đặt hoặc sử dụng tương thích với GPT là điều cần thiết.
* **Đường cong học tập**: GPT giới thiệu sơ đồ phân vùng mới có thể yêu cầu người dùng tự làm quen với các quy trình thiết lập và quản lý. Đường cong học tập này có thể là một bất tiện cho những người quen với MBR.

Hệ thống phân vùng trong Linux được sử dụng để chia ổ cứng thành các khu vực riêng biệt, giúp dễ dàng quản lý và bảo trì dữ liệu.

Phân vùng chính (Primary Partition): Một ổ cứng có thể chứa tối đa 4 phân vùng chính. Một trong số các phân vùng này có thể là phân vùng mở rộng.

Phân vùng mở rộng (Extended Partition): Phân vùng này không chứa dữ liệu trực tiếp, mà nó hoạt động như một container cho các phân vùng logic.

Phân vùng logic (Logical Partition): Các phân vùng nằm bên trong phân vùng mở rộng. Điều này cho phép người dùng vượt qua giới hạn 4 phân vùng chính trên một ổ cứng.

**2.3.3. Hệ thống log trong Linux.**

Log là một phần rất quan trọng trong việc kiểm soát và quản lý được lưu lượng, tiến trình và các hoạt động của hệ thống. Nếu bạn là một người quản trị hệ thống Linux thì sẽ phải tiếp xúc với nó. Log đóng vai trò quan trọng trong việc khắc phục sự cố. Trên thực tế thì khi có sự cố xảy ra thì các quản trị viên Linux dày dặn kinh nghiệm sẽ nói cho ta biết việc đầu tiên cần làm là đi xem log.

Có rất nhiều loại log: log hệ thống, log cho kernel, log cho trình quản lý gói, log cho quá trình khởi động, cho Apache,… Hầu như tất cả mọi thứ ta có thể nghĩ đến đều có log. Log thường được tìm thấy ở vị trí thuận tiện: /var/log, đây đều là log hệ thống và dịch vụ. Vị trí của log ứng dụng sẽ tùy thuộc vào nhà phát triển và ứng dụng có cho phép cấu hình nhật ký tùy chỉnh không. Ta hoàn toàn có thể xem được nhật ký hệ thống của mình, tất cả đều được thực hiện khá đơn giản từ dòng lệnh.

A screenshot of a computer

Description automatically generated

Hình 8. Hệ thống log.

Ta có thể xem một phần cuối của một file log như sau:

A screen shot of a computer program

Description automatically generated

Hình 9. Lệnh xem một phần cuối của một file log.

File Log Hệ Thống:

* *lastlog*: Lưu lịch sử đăng nhập lần cuối của từng người dùng. Hữu ích để kiểm tra các hoạt động đăng nhập.
* *btmp* và *btmp.1*: Ghi lại các lần đăng nhập thất bại, hỗ trợ phát hiện các cuộc tấn công brute-force.
* *wtmp*: Lưu lịch sử đăng nhập và đăng xuất của người dùng.

File Log Ứng Dụng và Dịch Vụ:

* *apache2*: Chứa log của máy chủ web Apache, gồm access.log (log truy cập) và error.log (log lỗi).
* *nginx*: Log cho máy chủ Nginx, bao gồm thông tin về truy cập và lỗi của Nginx.
* *redis*: Log của dịch vụ Redis. Theo dõi các hoạt động và lỗi của hệ thống Redis.
* *postgresql*: Log cho cơ sở dữ liệu PostgreSQL, hỗ trợ theo dõi kết nối và lỗi trong cơ sở dữ liệu.
* *openvpn*: Log của OpenVPN, ghi lại các kết nối VPN, hữu ích để kiểm tra tính ổn định và bảo mật kết nối.
* *exim4*: Log cho hệ thống gửi nhận email Exim4. Chứa thông tin về email được gửi đi và lỗi nếu có.
* *stunnel4*: Log của Stunnel (công cụ mã hóa lưu lượng TCP). Theo dõi lưu lượng mã hóa.

File Log Màn Hình và Phiên Đăng Nhập:

* *Xorg.0.log, Xorg.0.log.old, Xorg.1.log, Xorg.1.log.old*: Lưu thông tin từ Xorg (hệ thống đồ họa X Window), giúp xác định lỗi khi khởi động màn hình.
* *lightdm và sddm.log*: Log của các trình quản lý đăng nhập đồ họa (LightDM và SDDM), ghi lại các sự kiện đăng nhập.

File Log Hệ Thống và Quản Lý:

* *journal*: File log hệ thống, lưu toàn bộ các sự kiện và thông báo từ kernel và các dịch vụ thông qua systemd.
* *dpkg.log* và *dpkg.log.1*: Log của hệ thống quản lý gói dpkg (Debian), lưu lại các gói được cài đặt, xóa hoặc nâng cấp.
* *alternatives.log* và *alternatives.log.1*: Theo dõi thay đổi các phiên bản thay thế (alternatives) của các ứng dụng.
* *sysstat*: Chứa các thông tin thống kê hệ thống được ghi nhận định kỳ, hữu ích để giám sát hiệu năng.

Log Các Công Cụ và Dịch Vụ Khác:

* *gvm*: Liên quan đến công cụ Greenbone Vulnerability Management, ghi nhận các thông tin quét bảo mật.
* *inetsim*: Log của công cụ mô phỏng các dịch vụ mạng. Hữu ích trong các bài kiểm tra an ninh mạng.
* *mosquitto*: Log của dịch vụ MQTT Mosquitto, ghi lại các thông điệp và lỗi liên quan đến MQTT broker.
* *vmware-\* và vmware-network-\**: Log của các dịch vụ VMware, ghi lại các sự kiện về mạng, máy ảo và các tiến trình liên quan đến VMware.
* *notus-scanner*: Log của công cụ notus-scanner, ghi lại các quét và kết quả phát hiện bảo mật.
* *samba*: Log của dịch vụ Samba, cho phép chia sẻ file giữa Linux và Windows, giúp kiểm tra kết nối và lỗi khi chia sẻ tệp.

Log Khác

* *fontconfig.log*: Log của dịch vụ quản lý font chữ fontconfig, thường ghi nhận các thay đổi liên quan đến cấu hình font.
* *macchanger.log.1.gz*: Log của công cụ thay đổi địa chỉ MAC, thường được dùng trong các thử nghiệm bảo mật.

**2.3.4. Tiến trình và bộ nhớ trong Linux**

Tiến trình là một chương trình đang chạy và có bộ nhớ riêng biệt. Các tiến trình được quản lý bởi nhân Linux. Mỗi tiến trình sẽ được gán một ID (PID) duy nhất để định dạng và nhân Linux sử dụng các PID này để quản lý, theo dõi và giao tiếp giữa các tiến trình. Tiến trình có các trạng thái như running (đang chạy), sleeping (chờ hoặc đang ngủ), stopped (bị dừng), zombie (chờ thu hồi tài nguyên sau khi kết thúc). Để tạo một tiến trình mới, Linux sử dụng hệ thống để gọi fork() (fork() là một tiến trình system call trong Linux dùng để tạo một tiến trình mới) để tạo ra bản sao của tiến trình hiện tại. Tiến trình con có PID khác nhau nhưng nó sẽ chia sẻ không gian bộ nhớ chung với tiến trình cha.

Ta có thể xem các tiến trình như sau:

A screenshot of a computer program

Description automatically generated

Hình 10. Xem tiến trình bộ nhớ.

Hoặc có thể xem các tiến trình đang mở kết nối mạng:

A computer screen shot of white text

Description automatically generated

Hình 11. Xem các tiến trình đang mở kết nối mạng.

Ta có thể quản lý tiến trình với các lệnh khác như kill để dừng một tiến trình nếu muốn.

Trong pháp chứng số, tiến trình là một phần quan trọng ngoài log để ta có thế biết được các hoạt động đã xảy ra trước đó với thời gian cụ thể. Không chỉ thế, tiến trình còn giúp ta phát hiện kỹ thuật tấn công rootkit. Ngoài PID ta có các tham số khác để giúp việc điều tra như tên tiến trình (cho ta thông tin của tiến trình đang chạy, tuy nhiên nó cũng có hạn chế là các tiến trình độc hại có thể ngụy trang dưới tên của các tiến trình hợp pháp để tránh bị phát hiện), trạng thái của tiến trình (để xem các tiến trình đang hoạt động hay kết thúc, ở trạng thái zombie thường có thể là dấu hiệu lỗi phần mềm hoặc các kỹ thuật che dấu mã độc, UID và GID (xác định quyền truy cập của các tiến trình), đường dẫn thực thi (ta có thể biết được vị trí của file thực thi tiến trình trên hệ thống, một tiến trình có tên hợp pháp nhưng nó nằm ở vị trí bất thường thì cũng là một điểm đáng nghi), các kết nối mạng liên quan đến tiến trình, bộ nhớ (tiến trình nào sử dụng bộ nhớ và bộ nhớ khu vực nào), thời gian khởi tạo (tham số này giúp ta biết được tiến trình bắt đầu chạy từ thời điểm nào và kết hợp với các log hệ thống để phát hiện các hoạt động khả nghi), … Linux cung cấp các phương thức giao tiếp giữa các tiến trình với nhau như signal, pipe, socket và bộ nhớ chia sẻ để trao đổi thông tin giữa chúng

Bộ nhớ được quản lý bởi nhân Linux để cung cấp tài nguyên cho các tiến trình một cách hiệu quả, không để tình trạng tràn bộ nhớ. Linux sử dụng mô hình bộ nhớ ảo giúp các tiến trình có không gian địa chỉ riêng biệt, bảo vệ dữ liệu của tiến trình này không bị tiến trình khác truy cập. Khi bộ nhớ RAM bị đầy thì Linux sẽ lưu tạm dữ liệu từ RAM sang vùng swap trên ổ cứng. Các tiến trình có không gian bộ nhớ riêng sẽ chia thành các vùng như

* Text segment: chứa mã thực thi của chương trình.
* Data segment: chứa dữ liệu khởi tạo của biến toàn cục.
* Heap: khu vực cấp pháp bộ nhớ động, mở rộng khi chương trình yêu cầu.
* Stack: khu vực lưu trữ các biến cục bộ và thông tin về hàm, giảm kích thước hàm khi kết thúc.

Bộ nhớ cache và buffer để tối ưu hóa, Linux sử dụng chúng để nhằm lưu tạm các dữ liệu thường truy cập để giảm thời gian truy xuất vào các dữ liệu thường truy cập.

**2.3.5. Phân quyền và bảo mật**

Trong quá trình điều tra và phân tích hệ thống để tìm ra bằng chứng liên quan đến các hành vi vi phạm hoặc xâm nhập, phân quyền và bảo mật đóng vai trò quan trọng để điều tra ra các bằng chứng.

Phân quyền là phương pháp giới hạn truy cập của người dùng và tiến trình chỉ trong phạm vi cần thiết., giúp ngăn chặn người dùng trái phép hoặc tiến trình không đáng tin cậy truy cập hoặc chỉnh sửa dữ liệu nhạy cảm.

Quyền là thuộc tính của tệp và thư mục. Linux áp dụng các quyền truy cập dựa trên người dùng và nhóm (chủ sở hữu - owner, người cùng nhóm - group, người dùng khác - other) có quyền gì (đọc, ghi, thực thi) trên tệp và thư mục giúp ngăn chặn người dùng trái phép hoặc tiến trình không đáng tin cậy truy cập hoặc chỉnh sửa dữ liệu nhạy cảm.. Linux sử dụng 9 bit cho việc này, trong đó 3 bit đầu cho biết quyền đọc, ghi, thực thi của owner, 3 bit tiếp theo cho biết quyền của group, 3 bit cuối cho biết quyền của other. Ví dụ, *111110100 hay rwxrw-r--* cho biết owner có cả ba quyền, group có thể đọc và ghi nhưng không được thực thi, other chỉ được đọc. Trong các lệnh, 3 bit xác định quyền cho một đối tượng người dùng được biểu diễn bằng một số nguyên (có giá trị từ 0 đến 7), quyền được biểu diễn bằng ba số nguyên liên tiếp. Ví dụ, 764 là biểu diễn của *rwxrw-r--* ở trên.

Với các tệp và thu mục được tạo mới, quyền được xác định dựa trên quyền cơ sở (base permission) và user mask.

Base permission được thiết lập sẵn, không thể thay đổi:

* Đối với file thông thường giá trị BS là 666 *(110 110 110) (rw-rw-rw-)*
* Đối với thư mục (file đặc biệt) giá trị BS là 777 *(111 111 111) (rwxrwxrwx)*

User mask sẽ che đi một số bit trong Base Permission để tạo ra quyền truy cập chính thức cho file (tương tự như cơ chế của subnet mask). Cụ thể, quyền truy cập được tính bằng “Base permission” & “biểu diễn bù 1 của user mask”. Ví dụ, Base Permission là 666 (tức *110110110* khi chuyển sang dạng nhị phân), nên nếu giá trị user mask là 022 (có dạng nhị phân là *000010010* => dạng bù 1 của nó thì chuyển *1->0, 0->1* nên ta được *111101101*) thì quyền truy nhập chính thức của file sẽ là:

*110 110 110 & 111 101 101 = 110 100 100 = 644 (rw-r–r–)*

Có thể tính nhanh quyền truy xuất tệp và thư mục theo các công thức sau:

*Permit of file = 666 - umask*

*Permit of directory = 777 - umask*

Giá trị user mask mặc định cho người dùng thông thường là 002. Với mask này thì quyền hạn truy cập mặc định cho thư mục là 775 và file là 664.

Giá trị mask mặc định cho root là 022. Với mask này thì quyền hạn truy cập mặc định cho thư mục là 755 và file là 644.

Sử dụng chương trình umask để thay đổi user mask. Các tệp và thư mục được tạo ra sau lệnh umask sẽ chịu tác động của giá trị mask mới.

Việc phân quyền này giúp chúng ta trong quá trình truy vết xem ai là người đã truy cập hoặc sửa đổi, việc này rất quan trọng khi điều tra về các hành vi truy cập trái phép. Các thông tin về UID, GID giúp ta biết rõ người nào đã thực hiện các thay đổi.

Trong Linux chỉ có tài khoản root mới có toàn quyền trên hệ thống, còn lại thì nó chỉ cấp các quyền tối thiểu cần thiết để thực hiện nhiệm vụ. Điều này giúp thu hẹp phạm vi tìm kiếm lại dựa trên sự phân quyền ở các tiến trình, nếu một người dùng hoặc một tiến trình nào đó được cấp quyền cao hơn mức cần thiết thì ta hoàn toàn có cơ sở để nghi ngờ hành vi lợi dụng quyền cao. Các hoạt động sử dụng quyền root được ghi lại trong log hệ thống (chẳng hạn như /var/log/auth.log) để giúp ta theo dõi các hành vi lạm dụng quyền root.

Danh sách điều khiển truy cập ACLs có sẵn trong nhiều bản phân phối hiện đại và thường được kích hoạt mặc định trong hầu hết các tệp tin phổ biến như ext4, xfs hoặc btrfs.  ACLs có khả năng phân quyền tốt hơn so với quyền mặc định Linux, với ACLs thì ta có thể kiểm tra và xác định các quyền cụ thể của một người dùng hoặc một nhóm có trên từng tệp hoặc thư mục. Mỗi tệp và thư mục đều có quyền cụ thể giúp bảo vệ dữ liệu trước các truy cập trái phép. Phân quyền này giúp ta có thể kiểm tra xem các quyền của tệp có bị thay đổi trái phép không, từ đó xác định ra các hành vi bất thường, và có thể áp dụng nó vào việc chỉ cho phép xem trên các tệp hoặc file thư mục để bảo vệ toàn vẹn bằng chứng thu được trong quá trình điều tra. Ta có thể xem ACLs của một tệp cụ thể như sau:

A screen shot of a computer

Description automatically generated

Hình 12. Xem ACLs của một tệp cụ thể.

Mã hóa là yếu tố quan trọng để bảo mật thông tin nhạy cảm. Tính năng như GPG (GNU Privacy Guard) trong Linux giúp mã hóa tệp để bảo vệ dữ liệu khỏi bị truy cập trái phép, ngoài việc dựa vào phân quyền để bảo toàn tính toàn vẹn của dữ liệu điều tra thì việc mã hóa các tệp bằng chứng hoặc thiết lập quyền truy cập đặc biệt để bảo vệ tính toàn vẹn của dữ liệu. Ví dụ như khi ta nhận một ổ đĩa được trích xuất và được yêu cầu phân tích nó, trước khi vào việc phân tích thì việc mã hóa có thể giúp ích trong quá trình này để có thể so sánh với đoạn mã được mã hóa từ file đó, ta sẽ so sánh hai đoạn mã so sánh với nhau, nếu chúng trùng khớp thì có nghĩa dữ liệu này đúng với dữ liệu gốc ban đầu. Bằng cách mã hóa và xác minh chữ ký, ta có thể phát hiện ra các tệp đã bị chỉnh sửa hoặc chèn mã độc.

**2.3.6. Phân tích lưu lượng mạng**

Như chúng ta đã biết, mỗi ngày đều có các cuộc tấn mạng lớn, nhỏ xảy ra, và quá trình phân tích lưu lượng mạng để điều tra về các cuộc tấn công đã xảy ra là một điều không thể thiếu. Nó giúp ta xác định các hành vi phạm tội khi nào, từ đâu kẻ tấn công xâm nhập vào hệ thống cũng như khôi phục quá trình và hành động của kẻ tấn công. Bất kỳ hoạt động trên mạng nào đều để lại dấu vết và việc của ta là tìm ra các bằng chứng của các hành vi vi phạm trên không gian mạng. Ta hoàn toàn có thể xây dựng lại chuỗi sự kiện đã xảy ra để nắm rõ hơn về hành động của kẻ tấn công, đặc biệt là các sự kiện truyền tải dữ liệu hoặc thực thi mã độc qua mạng.

Network Statistics (netstat) là một công cụ hữu ích để kiểm tra các kết nối mạng, các cổng đang mở, bảng định tuyến và các giao thức mạng. Nó có thể liệt kê tất cả các kết nối TCP và UDP đang mở trên hệ thống:

A screenshot of a computer

Description automatically generated

Hình 13. Sử dụng netstat để liệt kê kết nối.

Liệt kê các giao diện mạng và thống kê lưu lượng qua các giao diện đó.

A screenshot of a computer

Description automatically generated

Hình 14. Liệt kê giao diện mạng.

Ngoài ta còn có các công cụ khác để phân tích lưu lượng mạng như tcpdump dùng để bắt gói dữ liệu trên giao diện eth0, ta có thể lưu file để phân tích sau.

A screen shot of a computer screen

Description automatically generated

Hình 15. Bắt lưu lượng mạng bằng công cụ.

Một công cụ mà ta đã cực kỳ quen thuộc và đã sử dụng nhiều trong các môn học trước là wireshark. Công cụ này với nhiều tính năng, giao diện thân thiện giúp ta phân tích các lưu lượng mạng được trực quan và rõ ràng, dễ hiểu hơn.

A screenshot of a computer

Description automatically generated

Hình 16. Bắt lưu lượng bằng wireshark.

**2.3.7. Sự khác biệt trong pháp chứng giữa máy Linux server và client.**

**Khi Linux đóng vai trò server, các loại chứng cứ tiềm năng chủ yếu xoay quanh:**

* **Log hệ thống và dịch vụ:**
  + Log từ các dịch vụ server (web server, database server) rất quan trọng trong việc xác định: thời gian và phương pháp tấn công, địa chỉ IP của kẻ tấn công.
  + Các log thường gặp:

***/****var/log/auth.log*: Ghi nhận đăng nhập SSH.

*/var/log/apache2/access.log*: Lưu truy cập vào server web.

*/var/log/nginx/error.log*: Log lỗi của Nginx.

*/var/log/syslog*: Log hệ thống chung.

* **Cơ sở dữ liệu và file dịch vụ:** Nếu server chứa database, dữ liệu này có thể bị truy cập trái phép hoặc sửa đổi.
* **Cấu hình và quyền truy cập:** File cấu hình chứa thông tin nhạy cảm, chẳng hạn: file cấu hình của dịch vụ web (/*etc*/*apache2*/, /*etc*/*nginx*/), cấu hình SSH (/*etc/ssh/sshd\_config*).
* **Mã độc hoặc công cụ tấn công:** Một server có thể bị khai thác để chạy mã độc hoặc công cụ khai thác:
* Tìm kiếm mã độc trong /*tmp* hoặc /*var/tmp*.
* Dữ liệu tải lên bất hợp pháp có thể nằm trong */var/www/* hoặc /*srv/.*

Thách thức trong pháp chứng trên server:

* **Dữ liệu lớn:** Một server thường lưu trữ khối lượng dữ liệu lớn (vd: log hàng triệu dòng).
* **Dịch vụ phức tạp:** Một server chạy nhiều dịch vụ khác nhau, mỗi dịch vụ có thể tạo ra loại log riêng.
* **Khôi phục dữ liệu đã bị xóa:** Phải tìm kiếm dữ liệu trong ổ đĩa hoặc phân vùng đã bị xóa, thường là trong các hệ thống file như ext4, xfs.

**Khi Linux đóng vai trò client, việc điều tra tập trung vào:**

* **Dữ liệu người dùng:**
  + Dữ liệu cá nhân, tài liệu hoặc các file tải về từ server.
  + Ví dụ: File trong */home/[user]/Documents, /home/[user]/Downloads.*

File ẩn: *~/.bash\_history* (lịch sử các lệnh đã chạy).

* **Dấu vết kết nối đến server:** Cần kiểm tra các log hoặc file cấu hình liên quan đến hoạt động kết nối:
* */etc/ssh/ssh\_config*: Kiểm tra các kết nối SSH đến server.
* */etc/fstab*: Dữ liệu gắn kết từ server qua NFS hoặc CIFS.
* *~/.ssh/known\_hosts*: Danh sách các server mà client đã từng kết nối.
* **Dấu vết truy cập mạng:** Phân tích các kết nối mạng của client để tìm dấu hiệu bất thường:
  + - Log từ iptables hoặc firewalld.
    - Gói tin mạng: Sử dụng công cụ như Wireshark hoặc tcpdump để phân tích lưu lượng.
* **Mã độc hoặc phần mềm gián điệp:** Client có thể bị nhiễm mã độc, thường được lưu trong:
  + - */tmp, /var/tmp*: Đây là nơi các mã độc tạm thời được lưu trữ.
    - *~/.config/*: Một số mã độc có thể ngụy trang thành cấu hình ứng dụng.

Thách thức trong pháp chứng trên client:

* Người dùng chủ động xóa dấu vết: Dữ liệu lịch sử (vd: *.bash\_history*, lịch sử trình duyệt) có thể bị xóa.
* Phần mềm mã hóa: Người dùng có thể sử dụng công cụ mã hóa để ẩn dữ liệu (vd: VeraCrypt).
* Phát hiện rootkit: Một số mã độc có thể ẩn trong kernel hoặc các thư viện hệ thống.

**Bảng so sánh**

| **Yếu tố** | **Server** | **Client** |
| --- | --- | --- |
| **Dữ liệu chính cần thu thập** | Log dịch vụ, cấu hình, dữ liệu hệ thống, mã độc. | Dữ liệu cá nhân, lịch sử truy cập, kết nối mạng, file cấu hình. |
| **Phân tích log** | Log của các dịch vụ (Apache, SSH, MySQL). | Log truy cập mạng, lịch sử lệnh (*~/.bash\_history*). |
| **Khám phá mã độc** | Mã độc trong thư mục tạm hoặc các thư mục phục vụ dịch vụ (**/***var*, **/***srv*). | Mã độc lưu trong thư mục người dùng (**/***tmp*, **~/.***config***/**). |
| **Phân tích kết nối** | Tìm nguồn gốc truy cập vào server (IP, giao thức). | Phân tích kết nối từ client tới server (SSH, NFS, Samba). |
| **Thách thức đặc thù** | Dữ liệu log lớn, dịch vụ phức tạp, khôi phục dữ liệu đã xóa. | Người dùng xóa dấu vết, dữ liệu mã hóa hoặc rootkit. |

**2.3.8. Các công cụ hỗ trợ**

**The Sleuth Kit**

Có rất nhiều công cụ hỗ trợ trong quá trình điều tra số trên linux, một bộ công cụ nổi tiếng là The Sleuth Kit. The Sleuth Kit (TSK) là một bộ công cụ mã nguồn mở được sử dụng trong pháp chứng số (digital forensics) để phân tích và điều tra các hệ thống file và phân tích các thiết bị lưu trữ. The Sleuth Kit cung cấp các công cụ dòng lệnh mạnh mẽ để phân tích các đĩa cứng, hệ thống file, và các thông tin từ các hệ thống lưu trữ như ổ cứng, ổ USB, và các thiết bị khác. Bộ công cụ này là phần mềm quan trọng trong việc thu thập chứng cứ số trong các vụ án pháp lý, điều tra tội phạm mạng, hoặc trong các tình huống bảo mật. Nó là một bộ công cụ bao gồm nhiều công cụ bên trong nó, mỗi lệnh tương ứng với một công cụ trong TSK và thực hiện nhiệm vụ rất cụ thể.

Các lệnh chính trong TSK:

* **fls**: Liệt kê các file và thư mục trong hệ thống file.
* **istat**: Hiển thị thông tin chi tiết về một inode, chẳng hạn như kích thước, thời gian tạo, và vị trí lưu trữ của file.
* **icat**: Trích xuất dữ liệu từ một inode, tức là bạn có thể lấy nội dung của một file từ một hệ thống file trong image.
* **mmount**: Mount một image ổ đĩa (hoặc phân vùng) vào hệ thống file hiện tại để có thể thao tác dễ dàng.
* **mmls**: Hiển thị thông tin về các phân vùng trên một ổ đĩa hoặc trong một image.
* **fsstat**: Hiển thị thông tin chi tiết về hệ thống file trong image, chẳng hạn như loại hệ thống file, kích thước, và số lượng block.
* **img\_stat**: Hiển thị thông tin về một image đĩa, chẳng hạn như kích thước, kiểu phân vùng, và các đặc điểm khác.
* **dcat**: Trích xuất dữ liệu trực tiếp từ một image hoặc ổ đĩa, thường dùng để lấy các sector nhất định.
* **tag**: Đánh dấu các file hoặc đối tượng trong hệ thống file để phục vụ cho phân tích sâu hơn hoặc báo cáo.
* **tsk\_recover**: Khôi phục các file đã bị xóa trong hệ thống file của image.
* **file\_stat**: Cung cấp thông tin chi tiết về một file hoặc thư mục, chẳng hạn như quyền truy cập, thời gian sửa đổi, và kích thước.
* **tsk\_loaddb**: Tạo hoặc nạp cơ sở dữ liệu từ các kết quả của các lệnh khác để phân tích dữ liệu một cách dễ dàng hơn.

**Autopsy**

Autopsy là một công cụ mã nguồn mở với giao diện đồ họa (GUI) được xây dựng trên nền tảng The Sleuth Kit (TSK), Autopsy được sử dụng để thực hiện các hoạt động pháp y số trên hình ảnh đĩa của bằng chứng. Cuộc điều tra pháp y được thực hiện trên hình ảnh đĩa được hiển thị ở đây. Các kết quả thu được từ Autopsy giúp ích cho việc điều tra và xác định thông tin liên quan. Công cụ Autopsy được sử dụng bởi các cơ quan thực thi pháp luật, cảnh sát địa phương và cũng có thể được sử dụng trong các doanh nghiệp để điều tra bằng chứng được tìm thấy trong một tội phạm máy tính. Nó cũng có thể được sử dụng để phục hồi thông tin đã bị xóa. Autopsy cho phép tạo các báo cáo chi tiết về quá trình phân tích và các phát hiện trong suốt cuộc điều tra. Các báo cáo này có thể được sử dụng làm chứng cứ trong các vụ án pháp lý.

Các tính năng chính trong Autopsy:

* **Phân tích hệ thống file**: Autopsy có thể phân tích và diễn giải cấu trúc hệ thống file của ổ đĩa hoặc phân vùng để tìm ra các file bị xóa, các file ẩn hoặc các dữ liệu đáng ngờ. Công cụ này có khả năng phục hồi các file đã bị xóa từ hệ thống file hoặc ổ đĩa.
* **Khôi phục dữ liệu đã xóa**: Một trong những tính năng quan trọng của Autopsy là khả năng khôi phục các file đã bị xóa trong khi vẫn bảo toàn được cấu trúc và dữ liệu gốc. Điều này cực kỳ hữu ích trong các cuộc điều tra tội phạm mạng, nơi các chứng cứ có thể đã bị cố tình xóa bỏ.
* **Phân tích email và lịch sử web**: Autopsy có khả năng phân tích các file email (ví dụ, PST, MBOX) và các file lịch sử web, giúp điều tra viên tìm kiếm các thông tin liên quan đến cuộc trò chuyện, danh sách liên lạc, hoặc các hoạt động trực tuyến của một cá nhân.
* **Chạy các plugin và phân tích dữ liệu logs**: Autopsy hỗ trợ chạy các plugin để thực hiện phân tích chuyên sâu hơn, chẳng hạn như quét virus, tìm kiếm các dấu vết của phần mềm độc hại, hoặc phân tích các log hệ thống. Điều này giúp cho việc phát hiện các sự kiện đáng ngờ trở nên dễ dàng hơn

**Chkrootkit**

Chkrootkit là một công cụ phát hiện rootkit trong hệ điều hành Linux. Rootkit là phần mềm gián điệp hoặc phần mềm độc hại được thiết kế để ẩn giấu sự hiện diện của mình trong hệ thống. Chkrootkit giúp kiểm tra hệ thống Linux để phát hiện các rootkit và cung cấp các cảnh báo khi phát hiện các dấu hiệu của một cuộc tấn công.

Các tính năng chính Chkrootkit:

* Phát hiện các rootkit đã cài đặt trong hệ thống Linux.
* Quét các file và tiến trình hệ thống để tìm dấu hiệu của rootkit.
* Cảnh báo khi phát hiện các rootkit phổ biến.

**Plaso**

Plaso là công cụ pháp chứng mã nguồn mở dùng để thu thập và phân tích các dữ liệu logs từ nhiều nguồn khác nhau, chẳng hạn như hệ thống file, hệ điều hành, và các ứng dụng. Plaso hỗ trợ việc xây dựng dòng thời gian từ các dữ liệu logs và giúp các nhà điều tra phân tích các sự kiện theo trình tự thời gian.

Các tính năng chính của Plaso:

* **Tạo dòng thời gian từ logs**: Plaso có thể thu thập dữ liệu từ nhiều nguồn logs khác nhau và xây dựng một dòng thời gian giúp nhà điều tra thấy được các sự kiện diễn ra trên hệ thống.
* **Hỗ trợ nhiều định dạng logs**: Công cụ này có thể xử lý nhiều định dạng logs khác nhau, chẳng hạn như các file của hệ điều hành, các logs ứng dụng, và các loại logs khác.
* **Đánh giá và phân tích sự kiện**: Plaso có thể phân tích các sự kiện để xác định hoạt động đáng ngờ, như truy cập vào các file nhạy cảm, thay đổi quyền truy cập, hay cài đặt phần mềm độc hại.

**TestDisk**

TestDisk là công cụ phục hồi dữ liệu mạnh mẽ giúp khôi phục các phân vùng bị mất và dữ liệu bị xóa trong hệ thống file của Linux. TestDisk hỗ trợ nhiều loại hệ thống file và có thể phục hồi các phân vùng, file và thư mục bị mất hoặc hỏng. Nó rất hữu ích trong trường hợp các ổ đĩa bị hư hỏng hoặc các phân vùng bị xóa vô tình.

Các tính năng chính của TestDisk:

* Khôi phục các phân vùng bị mất hoặc bị xóa.
* Phục hồi file bị mất trên các phân vùng.
* Hỗ trợ nhiều loại hệ thống file như EXT, NTFS, FAT, và HFS.

**DD (Disk Dump)**

DD là công cụ mạnh mẽ trên Linux để sao chép ổ đĩa hoặc phân vùng thành ảnh bit-by-bit, giúp bảo toàn toàn bộ dữ liệu khi thu thập chứng cứ từ thiết bị lưu trữ.

Các lệnh chính trong DD:

* **dd if=<source> of=<destination>**: Sao chép toàn bộ dữ liệu từ ổ đĩa nguồn vào file đích.
* **dd if=<source> of=<destination> bs=512**: Sao chép dữ liệu với kích thước block cố định (512 byte), giúp kiểm tra và sao chép các sector cụ thể.
* **dd if=<source> of=<destination> conv=noerror,sync**: Chuyển dữ liệu từ ổ đĩa nguồn sang file đích mà không bỏ qua các lỗi, giúp sao lưu toàn vẹn dữ liệu dù có lỗi xảy ra trong quá trình sao chép.

## **3. Các công cụ và kỹ thuật pháp chứng số**

Kỹ thuật pháp chứng số (digital forensics) là quá trình thu thập, phân tích, và bảo quản dữ liệu số từ các thiết bị điện tử để điều tra các sự cố, tội phạm hoặc vi phạm pháp luật.

### **3.1. Thu thập dữ liệu**

Phần thu thập dữ liệu (Data Acquisition) là bước đầu tiên và rất quan trọng trong quá trình pháp chứng số, đảm bảo rằng dữ liệu được thu thập một cách an toàn và toàn vẹn để có thể sử dụng làm bằng chứng trong các cuộc điều tra pháp lý. Phần này bao gồm hai kỹ thuật chính: Imaging và Live Acquisition.

* Imaging là quá trình tạo bản sao chính xác từng bit của một thiết bị lưu trữ, chẳng hạn như ổ cứng, USB, hoặc thẻ nhớ. Mục tiêu của quá trình này là tạo ra một bản sao không thay đổi (forensically sound) của thiết bị gốc để phân tích mà không ảnh hưởng đến dữ liệu ban đầu. Một số công cụ phổ biến của nó như: dd, dcfldd, Guymager, FTK Imager, …
* *dd:* Đây là một công cụ dòng lệnh cơ bản trong Unix/Linux dùng để sao chép dữ liệu từ thiết bị này sang thiết bị khác theo từng bit. *dd* có thể tạo bản sao của toàn bộ ổ đĩa hoặc phân vùng, đảm bảo bản sao chính xác của dữ liệu gốc.
* *dcfldd:* Một biến thể nâng cao của *dd* với các tính năng bổ sung như tính toán hash trong khi sao chép, tạo nhiều bản sao đồng thời và ghi log chi tiết. Điều này hữu ích cho việc xác minh tính toàn vẹn của dữ liệu.
* *Guymager:* Công cụ đồ họa mã nguồn mở giúp tạo ảnh đĩa một cách dễ dàng và nhanh chóng. Nó hỗ trợ nhiều định dạng ảnh khác nhau và cung cấp tính năng xác minh dữ liệu bằng cách sử dụng các thuật toán hash như MD5, SHA-1, và SHA-256.
* Live Acquisition là quá trình thu thập dữ liệu từ một hệ thống đang hoạt động. Đây là một kỹ thuật quan trọng trong các trường hợp mà hệ thống không thể bị tắt để tránh mất mát dữ liệu hoặc khi cần thu thập thông tin về trạng thái hiện tại của hệ thống.

### **3.2. Bảo quản dữ liệu**

Phần bảo quản dữ liệu (Data Preservation) là một bước rất quan trọng trong quá trình pháp chứng số. Mục tiêu của giai đoạn này là đảm bảo rằng tất cả các dữ liệu thu thập được giữ nguyên vẹn, không bị thay đổi hoặc hư hỏng, từ lúc thu thập đến lúc phân tích và trình bày làm bằng chứng. Việc bảo quản dữ liệu đúng cách là cần thiết để đảm bảo tính hợp lệ của các bằng chứng trong các quá trình pháp lý. Một số kỹ thuật bảo quản dữ liệu phổ biến của nó như:

* Hashing (Băm Dữ Liệu): là quá trình tạo ra một giá trị hash duy nhất từ dữ liệu thu thập được sử dụng để xác minh tính toàn vẹn của dữ liệu trong quá trình thu thập, phân tích, và lưu trữ. Một giá trị hash là một chuỗi ký tự độc nhất được tạo ra bằng cách sử dụng các thuật toán băm như MD5, SHA-1, hoặc SHA-256. Công cụ phổ biến để thực hiện quá trình này như:
* *md5sum:* Tạo giá trị hash MD5 cho một tệp tin hoặc dữ liệu. MD5 cung cấp tốc độ băm nhanh nhưng không còn được coi là an toàn do có khả năng xảy ra va chạm (collision).
* *sha256sum:* Tạo giá trị hash SHA-256, an toàn hơn so với MD5. SHA-256 được sử dụng phổ biến trong pháp chứng số do độ an toàn và độ dài hash lớn hơn.
* *shasum:* Hỗ trợ tạo các hash SHA-1, SHA-256, và các thuật toán SHA khác.
* *Phần mềm pháp chứng:* Các công cụ như *FTK Imager* và *Autopsy* cũng tích hợp tính năng tính toán giá trị hash cho các file hệ thống và ổ đĩa.
* Write Blockers (Chặn Ghi): là thiết bị phần cứng hoặc phần mềm được sử dụng để ngăn chặn việc ghi dữ liệu lên ổ đĩa hoặc thiết bị lưu trữ gốc trong quá trình thu thập và phân tích. Điều này đảm bảo rằng dữ liệu gốc không bị thay đổi, đảm bảo tính toàn vẹn của bằng chứng. Có hai loại write blockers:
* *Write Blockers phần cứng (Hardware Write Blockers):* Các thiết bị vật lý kết nối giữa máy tính và thiết bị lưu trữ để ngăn chặn bất kỳ hoạt động ghi nào. Ví dụ: Tableau T35u, CRU WiebeTech Forensic ComboDock.
* *Write Blockers phần mềm (Software Write Blockers):* Các chương trình phần mềm ngăn chặn ghi dữ liệu bằng cách kiểm soát quyền truy cập vào thiết bị. Tuy nhiên, chúng không an toàn bằng phần cứng vì vẫn có nguy cơ dữ liệu bị thay đổi do sự cố phần mềm hoặc lỗi người dùng.
* Lưu trữ và quản lý bằng chứng (Evidence Storage and Management): Quá trình lưu trữ và quản lý bằng chứng đòi hỏi phải lưu giữ an toàn và chính xác các dữ liệu và bản sao dữ liệu để đảm bảo rằng chúng có thể sử dụng làm bằng chứng hợp pháp.
* *Sử dụng hệ thống quản lý bằng chứng (Evidence Management System):* Các hệ thống này giúp theo dõi và quản lý các tệp bằng chứng, cung cấp khả năng theo dõi chuỗi hành trình dữ liệu (chain of custody).
* *Lưu trữ tại chỗ và ngoài chỗ (On-site and Off-site Storage):* Bằng chứng số thường được lưu trữ tại nhiều địa điểm để bảo vệ chống lại mất mát dữ liệu do thiên tai hoặc sự cố an ninh.
* *Mã hóa và bảo mật:* Sử dụng mã hóa để bảo vệ dữ liệu trong khi lưu trữ và truyền tải, đảm bảo rằng chỉ những người có quyền mới có thể truy cập được.

### **3.3. Phân tích dữ liệu và hệ thống**

Phân tích dữ liệu và hệ thống (Data and System Analysis) là quá trình quan trọng trong pháp chứng số, tập trung vào việc thu thập, trích xuất, và phân tích thông tin từ các thành phần khác nhau của hệ thống máy tính, bao gồm đĩa cứng, bộ nhớ, log file, và lưu lượng mạng. Mục tiêu chính của phân tích này là khám phá các bằng chứng tiềm năng, xác định các hoạt động đáng ngờ hoặc vi phạm, và đưa ra các kết luận hỗ trợ cho các cuộc điều tra pháp lý. Phân tích dữ liệu và hệ thống bao gồm các khía cạnh quan trọng như:

* Phân tích dữ liệu đĩa (Disk Data Analysis): là quá trình trích xuất thông tin từ các thiết bị lưu trữ như ổ cứng, SSD, và các thiết bị lưu trữ di động để phát hiện các bằng chứng tiềm năng. Các công cụ hỗ trợ như: *Autopsy, The Sleuth Kit, TestDisk, Foremost.*
* Phân tích bộ nhớ (Memory Analysis): là quá trình nghiên cứu các ảnh bộ nhớ (RAM) của hệ thống để phát hiện các quy trình độc hại, các kết nối mạng đáng ngờ, và các thông tin chỉ tồn tại trong bộ nhớ. Các công cụ hỗ trợ như: *Rekall, LiME, Memdump.*
* Phân tích file nhật ký (Log File Analysis): là quá trình kiểm tra và phân tích các file nhật ký được tạo ra bởi hệ điều hành, ứng dụng, và các dịch vụ mạng để theo dõi các sự kiện và phát hiện các hoạt động đáng ngờ. Các công cụ hỗ trợ như: *Log2timeline, plaso, Splunk, ELK Stack.*
* Phân tích mạng (Network Forensics): là quá trình thu thập, ghi lại, và phân tích dữ liệu lưu lượng mạng để phát hiện các cuộc tấn công mạng, hành vi không bình thường, hoặc sự vi phạm an ninh. Các công cụ hỗ trợ như: *Wireshark, tcpdump, NetworkMiner, Zeek (Bro).*
* Phân tích ứng dụng và dữ liệu (Application and Data Analysis): là quá trình trích xuất và phân tích các thông tin từ các ứng dụng và file cụ thể như email, tài liệu, và cơ sở dữ liệu. Các công cụ hỗ trợ như: *Xplico, Bulk Extractor.*

### **3.4. Dịch ngược**

Dịch ngược là quá trình nghiên cứu một đối tượng (như phần mềm, phần cứng, hoặc mã nguồn) để hiểu cách thức hoạt động bên trong của nó mà không có quyền truy cập vào mã nguồn gốc hoặc thiết kế ban đầu. Trong bối cảnh pháp chứng số, phân tích ngược thường được sử dụng để phân tích phần mềm độc hại (malware), khám phá các lỗ hổng bảo mật, hoặc để điều tra cách thức một phần mềm hoạt động mà không có tài liệu hoặc mã nguồn ban đầu. Một số kỹ thuật phổ biến của nó như:

* Static Analysis (Phân Tích Tĩnh): Phân tích mã nhị phân hoặc mã máy mà không thực thi chương trình. Tập trung vào phân tích cấu trúc và luồng điều khiển của chương trình.
* Dynamic Analysis (Phân Tích Động): Thực thi chương trình hoặc mã trong môi trường cách ly (sandbox) để quan sát hành vi của nó.
* Hybrid Analysis (Phân Tích Kết Hợp): Kết hợp cả static và dynamic analysis để có một cái nhìn toàn diện về phần mềm.
* Decompilation (Dịch Ngược): Chuyển đổi mã máy hoặc mã nhị phân thành mã nguồn có thể đọc được (như C hoặc Python).
* Signature-Based Analysis (Phân Tích Dựa Trên Chữ Ký): Sử dụng các mẫu chữ ký (signatures) của các mã độc hoặc mã đã biết để phát hiện các đoạn mã hoặc các hành vi tương tự.

Các công cụ được sử dụng phổ biến để hỗ trợ cho quá trình dịch ngược này:

* *IDA Pro (Interactive DisAssembler):* Disassemblermạnh mẽ, hỗ trợ phân tích mã máy (machine code) sang mã assembly có thể đọc được. Hỗ trợ plugin và mở rộng, có chế độ tương tác cho phép phân tích các đường dẫn mã. Ứng dụng cho: Phân tích mã độc, lỗ hổng bảo mật, và các phần mềm đóng gói.
* *Ghidra:* Bộ công cụ phân tích ngược nguồn mở do Cơ quan An ninh Quốc gia Hoa Kỳ (NSA) phát triển. Hỗ trợ disassembly và decompilation đa kiến trúc, có giao diện đồ họa và script Python. Ứng dụng cho: Phân tích phần mềm độc hại, phục hồi mã nguồn, nghiên cứu phần mềm.
* *OllyDbg:* Debugger mạnh mẽ dành cho các chương trình Windows, tập trung vào phân tích mã nhị phân (binary code). Có khả năng theo dõi các lệnh trong thời gian thực, hiển thị các thư viện và biến cục bộ. Ứng dụng cho: Phân tích các ứng dụng Windows, khai thác lỗi và thực hiện nghiên cứu phần mềm.
* *Radare2:* Framework phân tích ngược nguồn mở, cung cấp nhiều công cụ để phân tích mã máy, debug, và sửa đổi mã. Khả năng scripting mạnh mẽ, hỗ trợ nhiều định dạng file, nền tảng. Ứng dụng cho: Phân tích mã độc, nghiên cứu bảo mật và lỗ hổng.
* *Binwalk:* Công cụ phân tích ngược nhắm đến các file nhị phân firmware. Tự động phát hiện và trích xuất các phần tử trong firmware như hệ điều hành nhúng. Ứng dụng cho: Phân tích firmware cho các thiết bị IoT, router, và các thiết bị nhúng khác.
* *Cutter:* Giao diện đồ họa người dùng (GUI) dành cho *Radare2*, giúp dễ dàng phân tích các file nhị phân. Hỗ trợ đa nền tảng, có các công cụ phân tích nâng cao. Ứng dụng cho: Phân tích mã độc, nghiên cứu phần mềm và bảo mật.

# **CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM**

## **1. Triển khai điều tra thử thách SCAN24 của Honeynet Project**

### **1.1. SCAN24 là gì?**

SCAN24 là một thách thức trong chuỗi "Scan of the Month" của Honeynet Project, nhằm kiểm tra khả năng pháp chứng kỹ thuật số của người tham gia. Trong thử thách SCAN24, người tham gia được cung cấp một tình huống giả định và cần phân tích một ảnh đĩa của một đĩa mềm đã được khôi phục để trả lời một loạt câu hỏi pháp y. Thử thách yêu cầu người tham gia đọc một báo cáo của cảnh sát trước khi bắt đầu phân tích để có cái nhìn tổng quan về bối cảnh. Sau đó, người tham gia sẽ tải xuống và kiểm tra ảnh đĩa image.zip (đi kèm mã kiểm tra MD5) để tìm câu trả lời cho các câu hỏi, bao gồm việc xác định nhà cung cấp của Joe Jacobs và các bước mà nghi phạm đã thực hiện để che giấu thông tin trong các tệp.

### **1.2. Báo cáo của cảnh sát**

Kịch bản như sau: Joe Jacobs, 28 tuổi, đã bị bắt hôm qua vì tội bán ma túy bất hợp pháp cho học sinh trung học. Một cảnh sát địa phương đóng giả làm học sinh trung học đã bị Jacobs tiếp cận tại bãi đậu xe của Trường trung học Smith Hill. Jacobs hỏi cảnh sát chìm rằng anh ta có muốn mua cần sa không. Trước khi cảnh sát chìm kịp trả lời, Jacobs đã rút một ít cần sa ra khỏi túi và đưa cho cảnh sát. Jacobs nói với cảnh sát, "Nhìn thứ này này; người Colombia không thể trồng nó tốt hơn! Nhà cung cấp của tôi không chỉ bán trực tiếp cho tôi mà còn tự trồng nó".

Jacobs đã nhiều lần bị phát hiện tụ tập tại các bãi đậu xe của nhiều trường trung học địa phương vào khoảng 2:30 chiều, thời điểm trường học thường kết thúc trong ngày. Các viên chức nhà trường từ nhiều trường trung học đã gọi điện cho cảnh sát về sự hiện diện của Jacobs tại trường của họ và ghi nhận tình trạng sử dụng ma túy gia tăng ở học sinh kể từ khi anh ta đến.

Cảnh sát cần sự giúp đỡ của bạn. Họ muốn cố gắng xác định xem Joe Jacobs có bán ma túy cho học sinh ở các trường khác ngoài Smith Hill hay không. Vấn đề là không có học sinh nào đứng ra giúp cảnh sát. Dựa trên bình luận của Joe về người Colombia, cảnh sát muốn tìm ra nhà cung cấp/sản xuất cần sa của Joe Jacob.

Jacobs đã phủ nhận việc bán ma túy tại bất kỳ trường nào khác ngoài Smith Hill và từ chối cung cấp cho cảnh sát tên nhà cung cấp/sản xuất ma túy của mình. Jacobs cũng từ chối xác thực lời khai mà anh ta đã nói với cảnh sát chìm ngay trước khi bị bắt. Sau khi ban hành lệnh khám xét và khám xét nhà của nghi phạm, cảnh sát đã có thể thu được một lượng nhỏ cần sa. Cảnh sát cũng thu giữ một đĩa mềm, nhưng không có máy tính và/hoặc phương tiện truyền thông nào khác trong nhà.

Cảnh sát đã chụp ảnh đĩa mềm của nghi phạm và cung cấp cho bạn một bản sao. Họ muốn bạn kiểm tra đĩa mềm và cung cấp câu trả lời cho các câu hỏi sau. Cảnh sát muốn bạn đặc biệt chú ý đến bất kỳ thông tin nào có thể chứng minh rằng Joe Jacobs thực sự đã bán ma túy tại các trường trung học khác ngoài Smith Hill. Họ cũng muốn bạn cố gắng xác định, nếu có thể, nhà cung cấp của Joe Jacob là ai.

Jacobs đã đăng lệnh tại ngoại ở mức 10.000,00 đô la. Sợ rằng anh ta có thể bỏ trốn, cảnh sát muốn bắt anh ta lại càng sớm càng tốt. Để làm được như vậy, cảnh sát đã yêu cầu bạn hoàn thành đầy đủ kết quả và nộp trước ngày 25 tháng 10 năm 2002. Vui lòng cung cấp cho cảnh sát một hồ sơ mạnh mẽ bao gồm các phát hiện cụ thể của bạn liên quan đến các câu hỏi, nơi các phát hiện nằm trên đĩa, các quy trình và kỹ thuật được sử dụng và bất kỳ hành động nào mà nghi phạm có thể đã thực hiện để cố ý xóa, ẩn và/hoặc thay đổi dữ liệu trên đĩa mềm. Chúc may mắn!

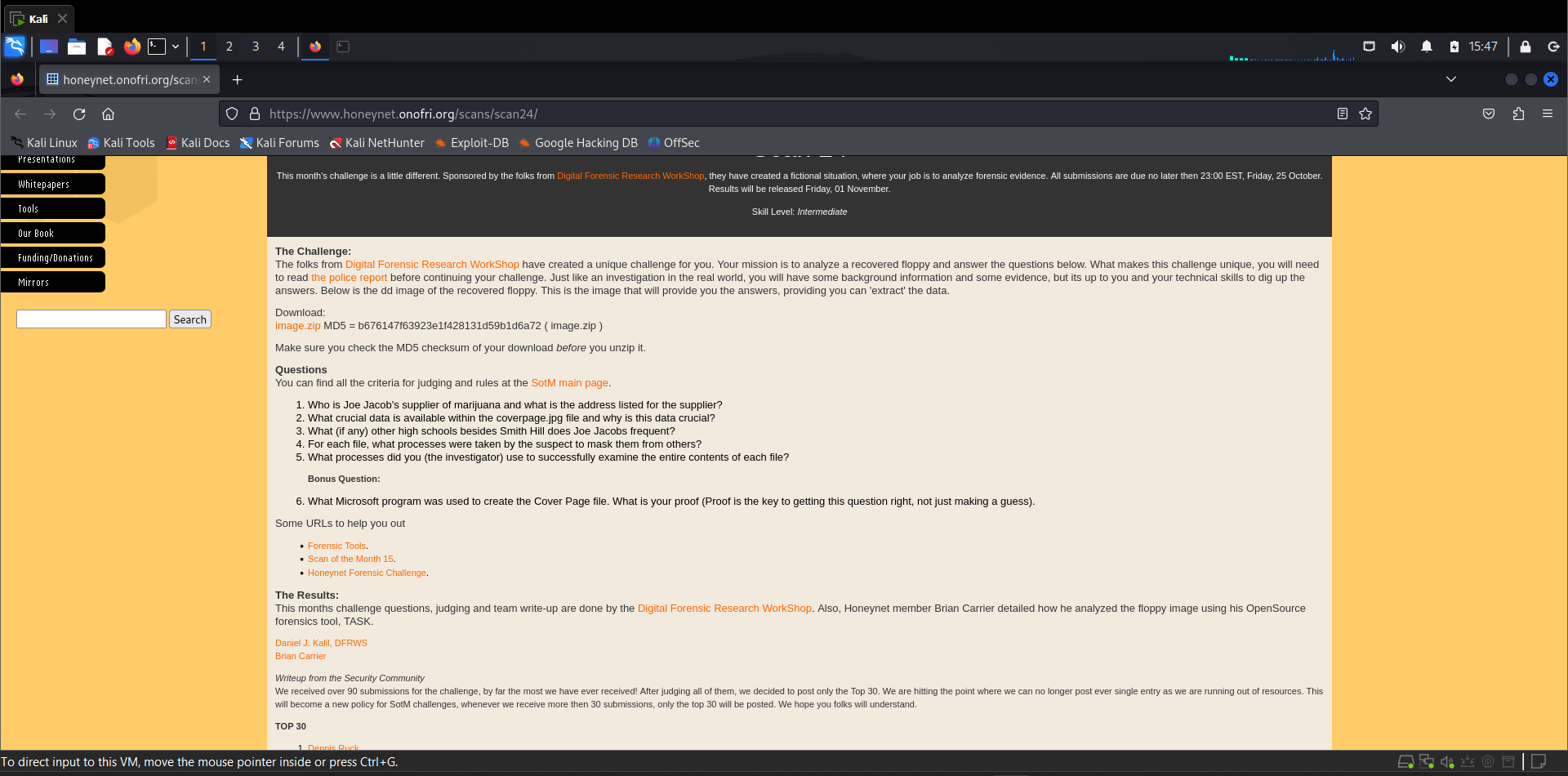
### **1.3. Các câu hỏi cần tìm hiểu và điều tra**

* Ai là nhà cung cấp cần sa của Joe Jacobs và địa chỉ của nhà cung cấp đó?
* Dữ liệu quan trọng nào có trong tệp coverpage.jpg và tại sao nó quan trọng?
* Joe Jacobs còn thường xuyên lui tới trường trung học nào khác ngoài Smith Hill?
* Với mỗi tệp, nghi phạm đã thực hiện những quy trình nào để che giấu chúng khỏi người khác?
* Những quy trình nào bạn (người điều tra) đã sử dụng để kiểm tra thành công toàn bộ nội dung của từng tệp?
* Câu hỏi phụ: Chương trình Microsoft nào đã được sử dụng để tạo ra tệp Cover Page và bằng chứng cho điều đó là gì?

## **2. Triển khai điều tra**

### **2.1. Pháp chứng với Autopsy**

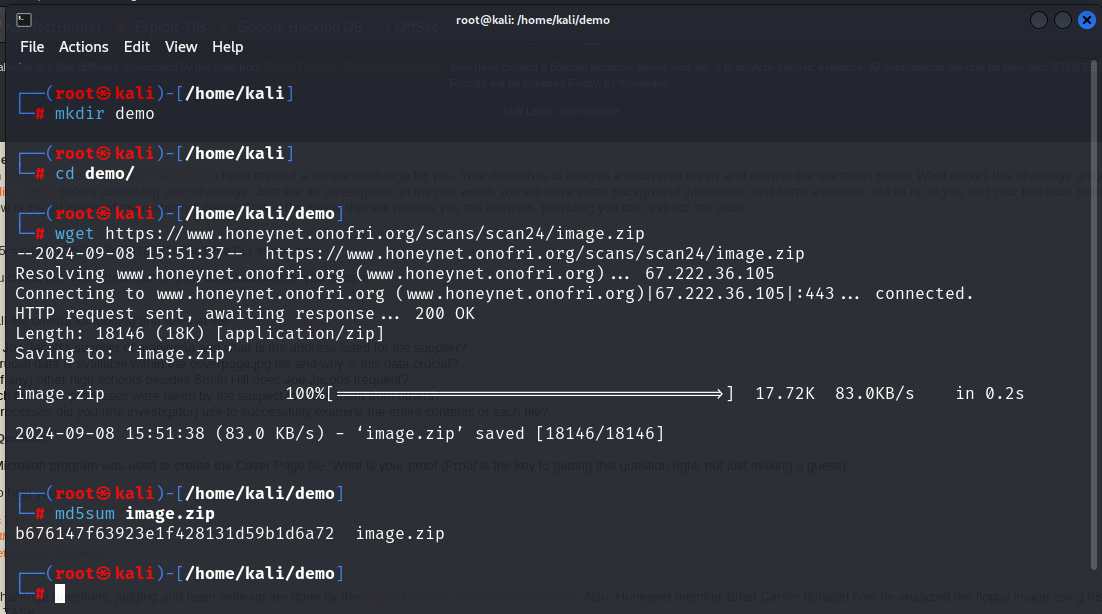
Tạo thư mục để lưu trữ các dữ liệu điều tra với tên demo sau đó ta cần tải file image về trên trang của Honeynet:



Hình 17. Giao diện trang Honeynet.

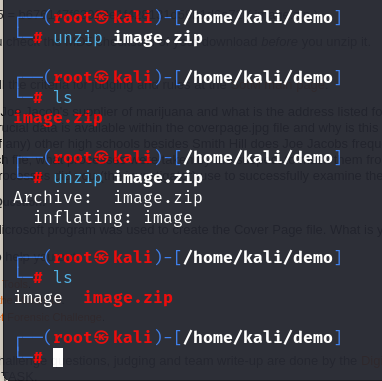
Đoạn mã xác nhận dữ liệu toàn vẹn: MD5 = b676147f63923e1f428131d59b1d6a72 ( image.zip )

Sau khi dùng md5sum ta được đoạn mã giống với mã xác nhận dữ liệu toàn vẹn không mất mát: b676147f63923e1f428131d59b1d6a72 image.zip



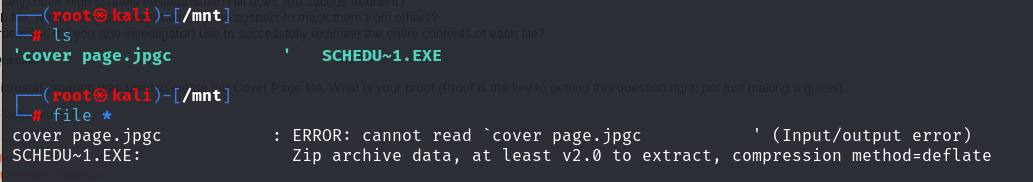
Hình 18. Kiểm tra tính toàn vẹn bằng MD5.

Kế tiếp tiến hành unzip file:



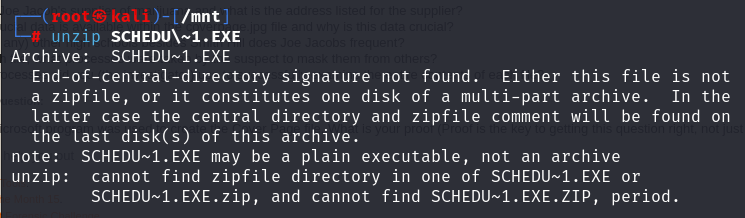
Hình 19. Unzip file image vừa tải.

Sau khi gắn kết đĩa ta vào thư mục gắn kết thì thấy xuất hiện 2 file. Sử dụng lệnh file để kiểm tra xem định dạng thật của 2 file này là gì:



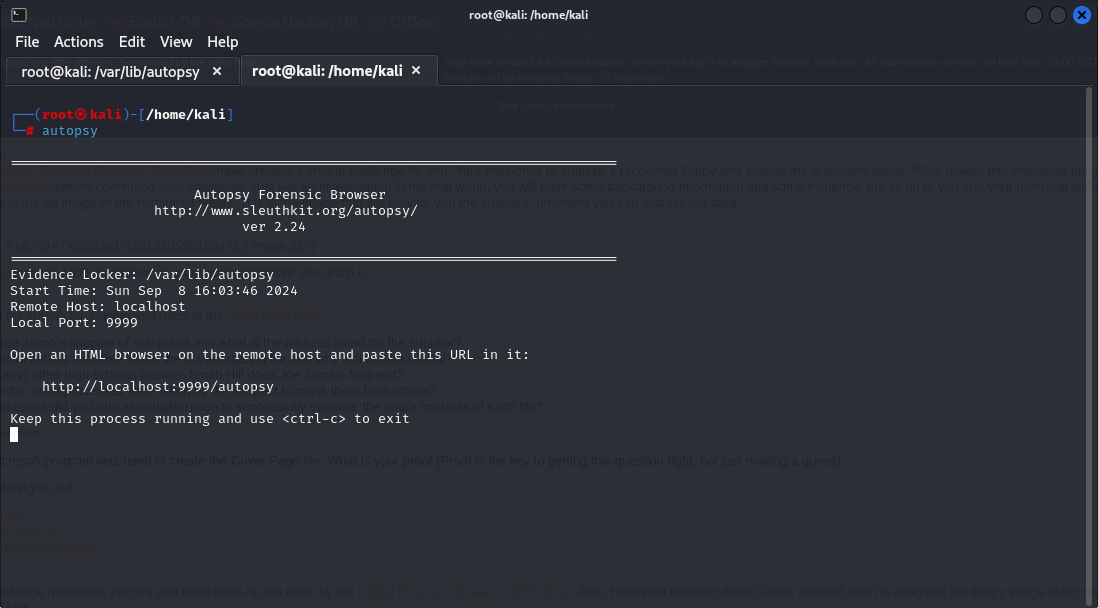
Hình 20. Kiểm tra định dạng thật của file.

Ta thấy có một file ảnh nhưng không thể đọc được và một file zip hiển thị dưới dạng file .exe. Tiến hành unzip thử file còn lại tuy nhiên cũng không thu được kết quả gì



Hình 21. Unzip file.

Điều này có nghĩa file này đã bị cắt, xóa hoặc có tác động gì đó đến nó dẫn đến việc nó bị hỏng. Ta sử dụng công cụ autopsy để tiến hành phân tích file đĩa này. Để mở giao diện của autopsy ta tiến hành truy cập vào đường dẫn hiển thị sau khi mở autopsy ra (một lưu ý là hãy chạy autopsy dưới quyền root: http://localhost:9999/autopsy



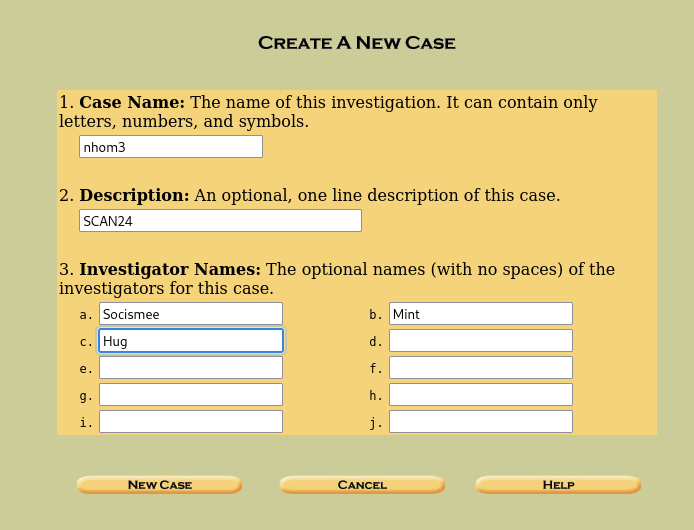
Hình 22. Mở Autopsy.

Ta có giao diện của autopsy như sau, tiến hành chọn New Case để tạo một hồ sơ trường hợp phân tích mới:



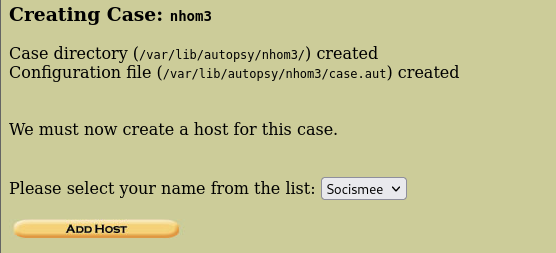
Hình 23. Giao diện của Autopsy.

Điền thông tin của tên hồ sơ (trường hợp), mô tả của nó và tên các điều tra viên sau đó nhấn chọn New Case:



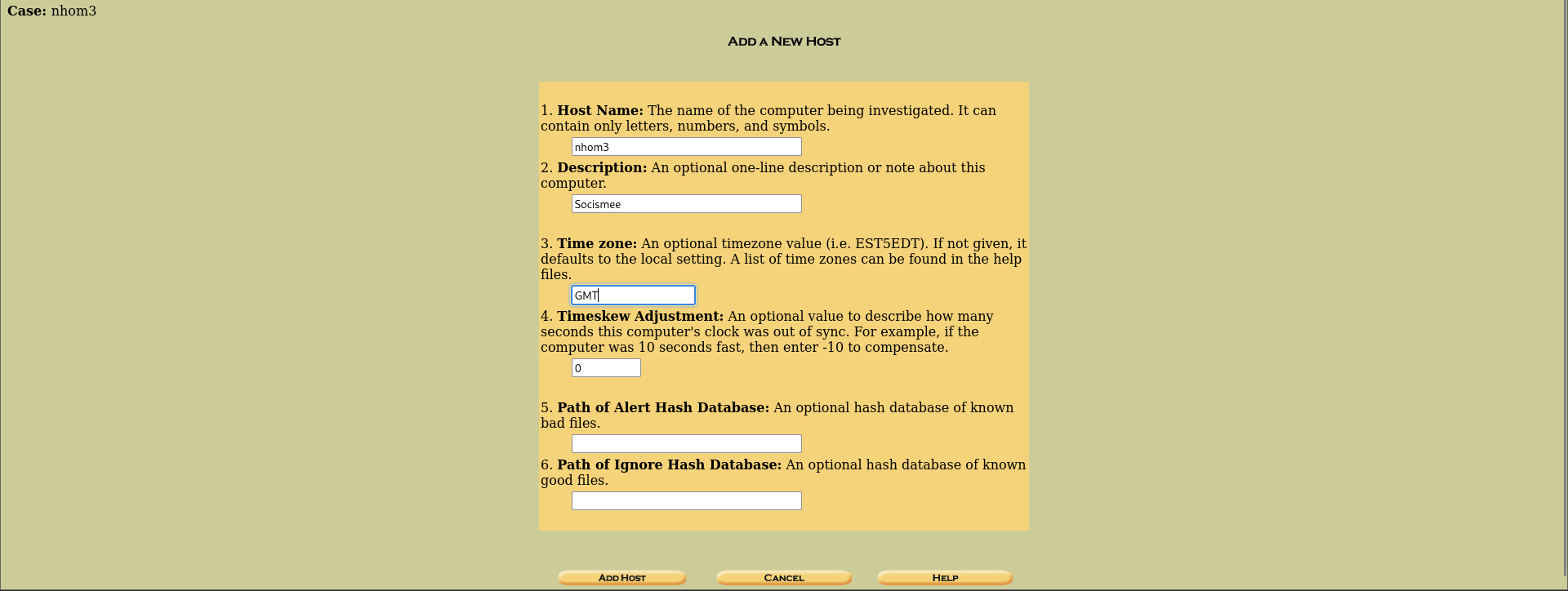
Hình 24. Giao diện lúc tạo New Case.

File được tạo, chọn tên điều tra viên tương ứng và chọn Add host:



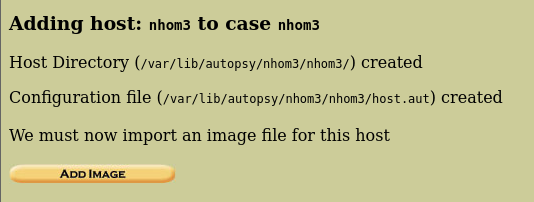
Hình 25. Giao diện lúc tạo Case xong.

Điền các thông tin và chọn Add host:



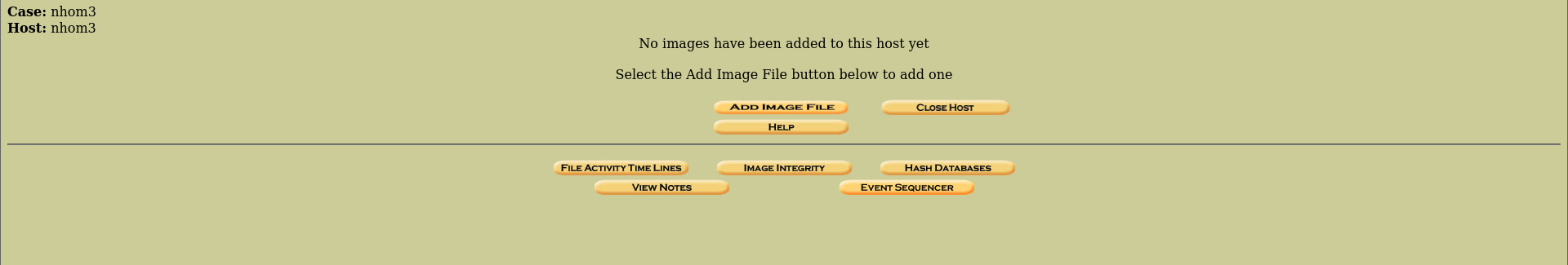
Hình 26. Điền thông tin và Add Host cho Case.

Sau đó chọn Add image:



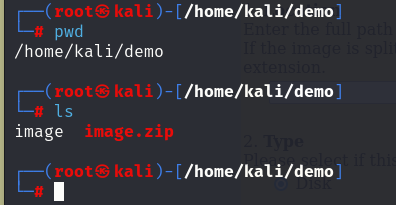
Hình 27. Giao diện lúc Add host xong.

Chọn Add image file:



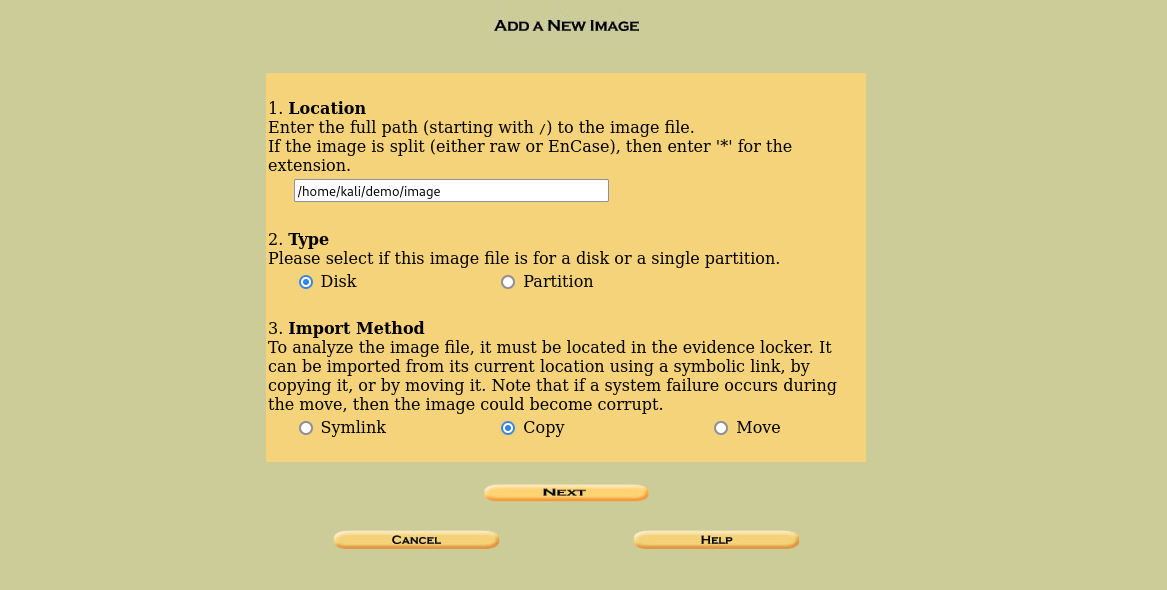
Hình 28. Add image.

Điền đúng thông tin đường dẫn của file image đã tải về và unzip ban đầu:



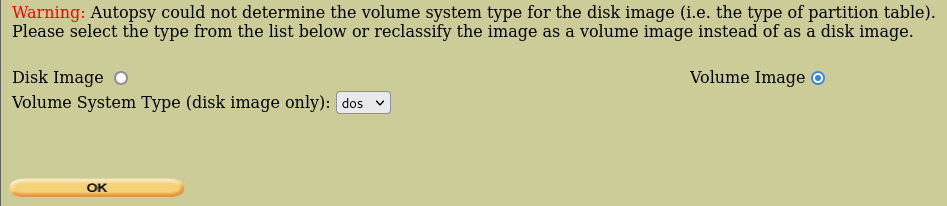
Hình 29. Đường dẫn của file image.

Click chọn Next:



Hình 30. Điền đường dẫn vào mục Location.

Ta thấy có cảnh báo: Autopsy không thể xác định loại hệ thống ổ đĩa cho ảnh đĩa (tức là loại bảng phân vùng). Vui lòng chọn loại từ danh sách bên dưới hoặc phân loại lại ảnh thành ảnh ổ đĩa thay vì ảnh đĩa. Ta click chọn Volume Image và chọn OK:

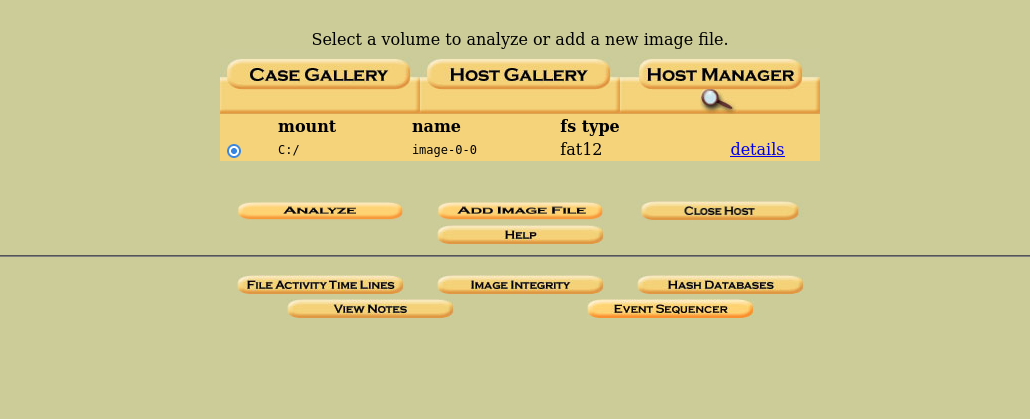


Hình 31. Cảnh báo Autopsy không thể xác định loại hệ thống ổ đĩa cho ảnh đĩa.

Chọn Add:

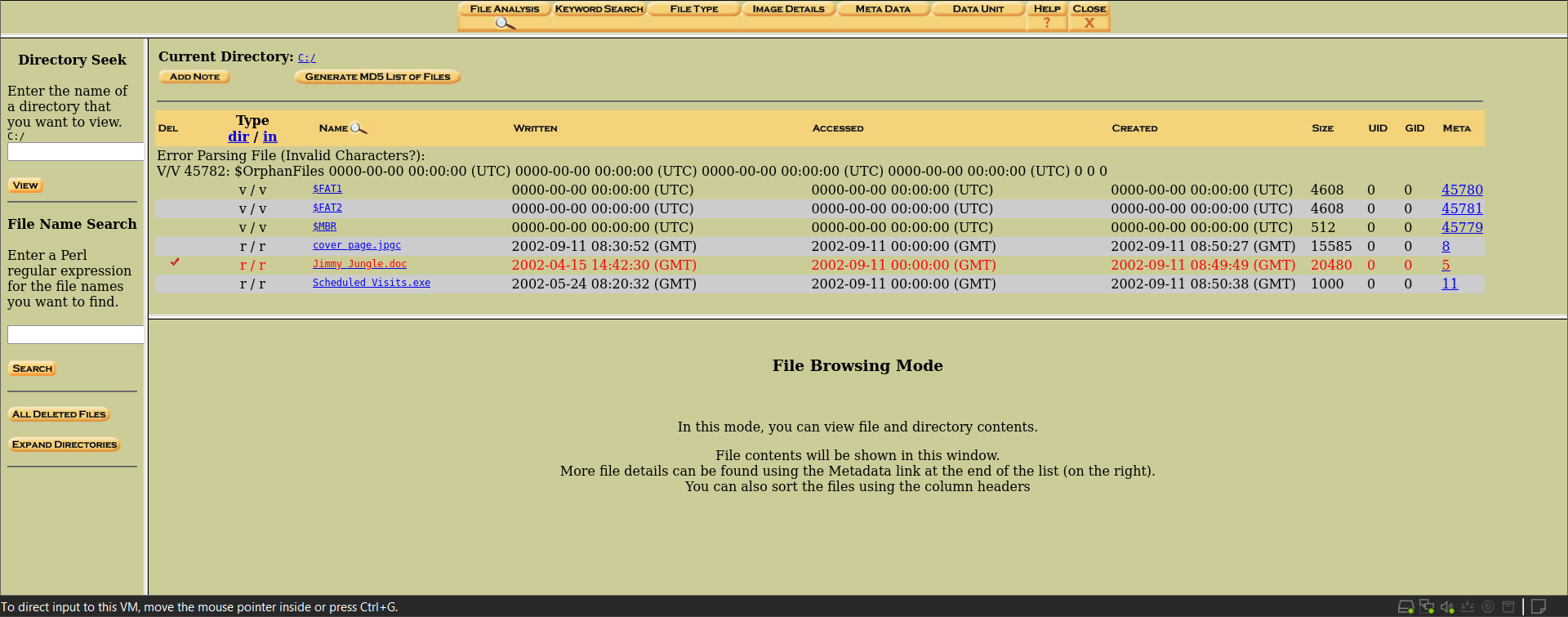


Tiến hành phân tích dữ liệu của ổ đĩa:



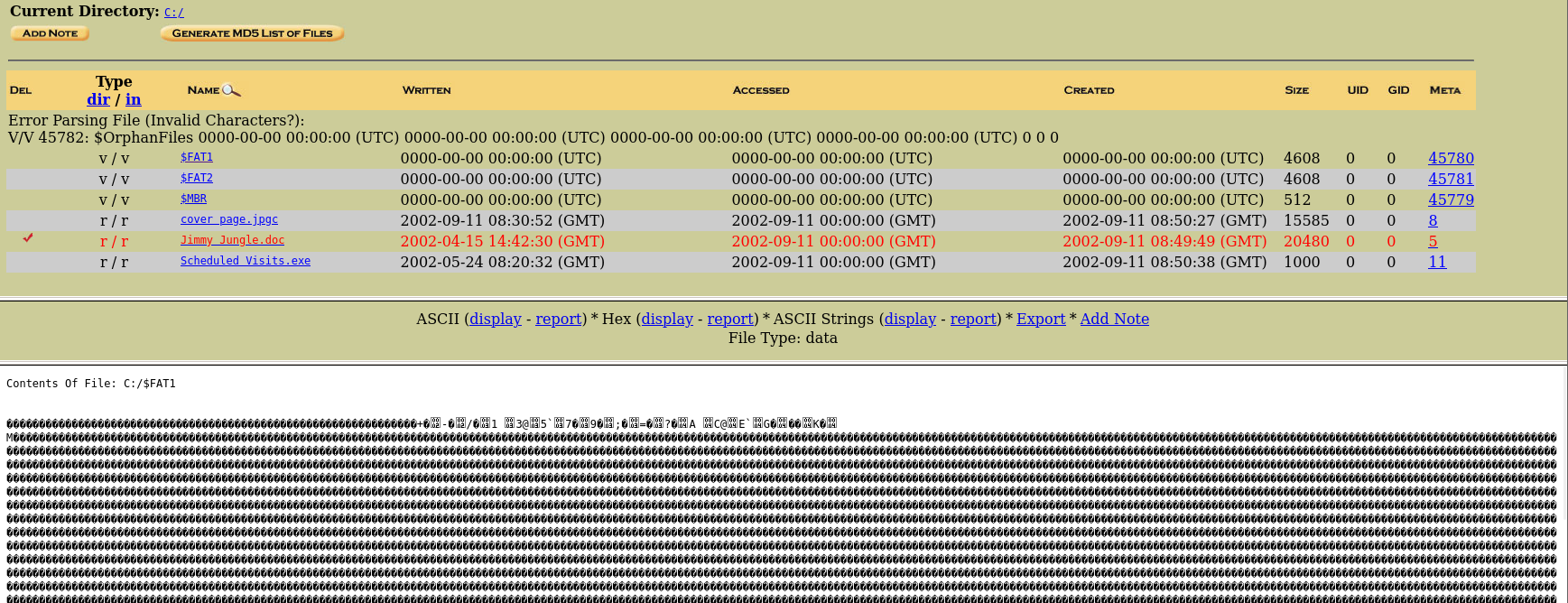
Hình 32. Tiến hành phân tích dữ liệu ổ đĩa.

Ta chọn chế độ File Analyze để xem tất cả thông tin của các file trong ổ đĩa như sau:



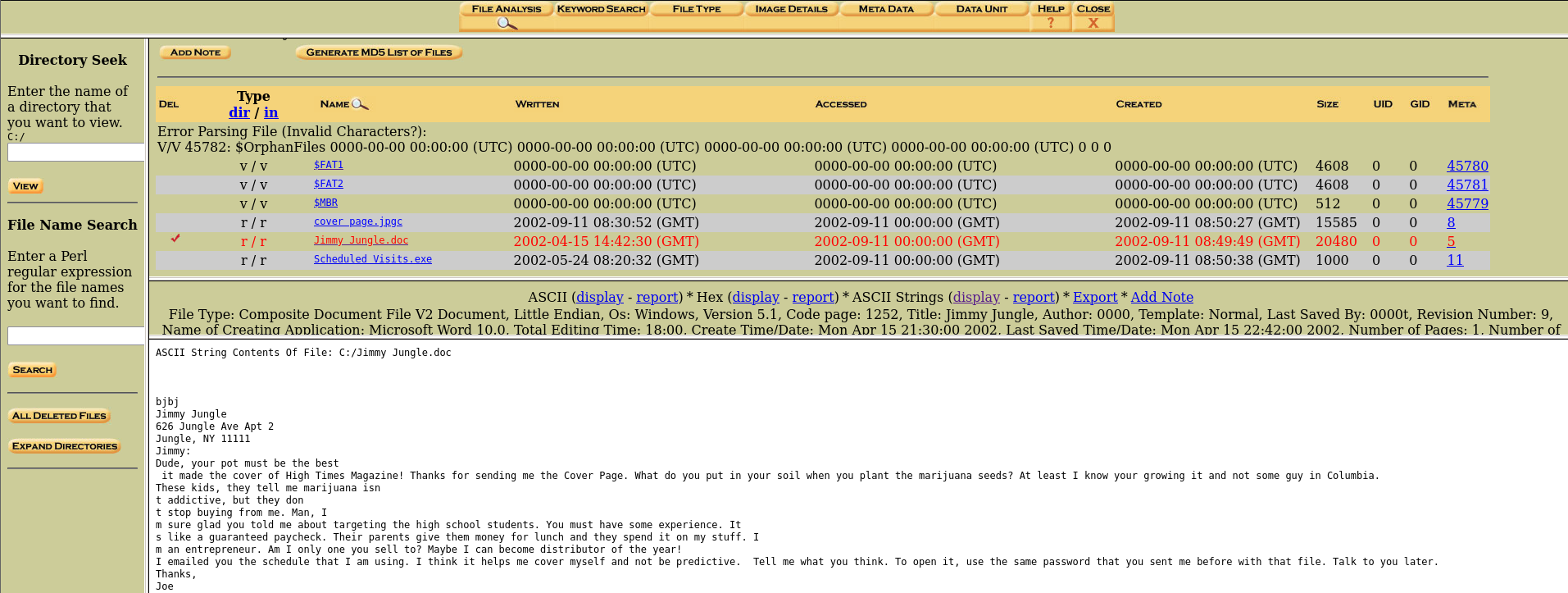
Hình 33. Thông tin của các file trong ổ đĩa.

Nhấn vào từng file để xem thông tin:



Hình 34. Thông tin của từng file.

Ta để ý 3 file dưới cùng, đó chính là file .jpgc, file .doc và file .exe. Tuy nhiên khi check các dạng hiển thị của ASCII, Hex và Strings thì chỉ có file .doc ở dạng hiển thị là ASCII có chứa thông tin của một bức thư.



Hình 35. Check các dạng hiển thị của ASCII.

Bức thư với nội dung như sau:

*Jimmy Jungle*

*626 Jungle Ave Apt 2*

*Jungle, NY 11111*

*Jimmy:*

*Dude, your pot must be the best it made the cover of High Times Magazine! Thanks for sending me the Cover Page. What do you put in your soil when you plant the marijuana seeds? At least I know your growing it and not some guy in Columbia.*

*These kids, they tell me marijuana isnt addictive, but they dont stop buying from me. Man, Im sure glad you told me about targeting the high school students. You must have some experience. Its like a guaranteed paycheck. Their parents give them money for lunch and they spend it on my stuff. Im an entrepreneur. Am I only one you sell to? Maybe I can become distributor of the year!*

*I emailed you the schedule that I am using. I think it helps me cover myself and not be predictive. Tell me what you think. To open it, use the same password that you sent me before with that file. Talk to you later.*

*Thanks,*

*Joe*

Có thể dịch như sau:

*Jimmy Jungle*

*626 Đại lộ Jungle, Căn hộ 2*

*Jungle, NY 11111*

*Jimmy:*

*Ông ơi, cần sa của ông chắc phải là loại tốt nhất vì nó đã lên trang bìa của Tạp chí High Times! Cảm ơn ông đã gửi cho tôi Trang Bìa đó. Ông bỏ gì vào đất khi trồng hạt cần sa vậy? Ít nhất tôi biết là ông tự trồng nó chứ không phải mua từ một gã nào đó ở Colombia.*

*Mấy đứa trẻ này, chúng nói với tôi cần sa không gây nghiện, nhưng chúng không ngừng mua từ tôi. Trời ơi, tôi rất mừng là ông đã nói với tôi về việc nhắm vào học sinh trung học. Ông chắc hẳn có kinh nghiệm rồi. Nó giống như một khoản tiền lương được đảm bảo vậy. Bố mẹ chúng đưa tiền cho chúng ăn trưa và chúng dùng nó để mua hàng của tôi. Tôi là một doanh nhân đấy. Tôi có phải là người duy nhất ông bán cho không? Có lẽ tôi có thể trở thành nhà phân phối của năm!*

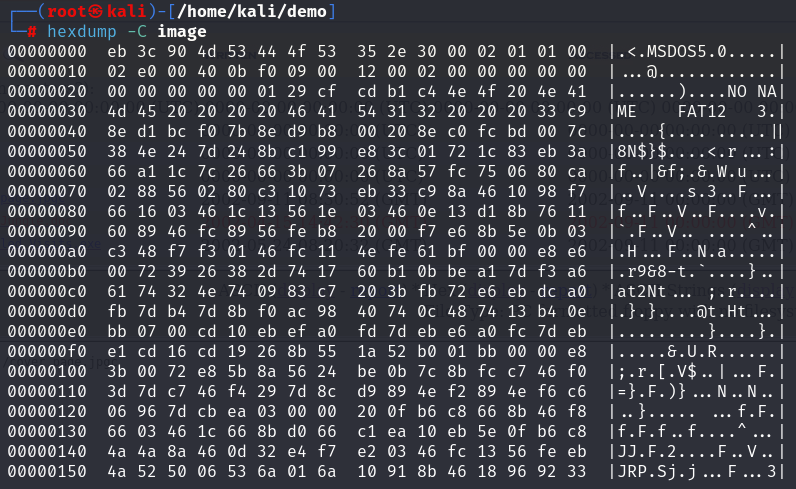
*Tôi đã gửi email cho ông lịch trình mà tôi đang sử dụng. Tôi nghĩ nó giúp tôi che đậy và không bị đoán trước. Hãy cho tôi biết ông nghĩ sao. Để mở nó, hãy sử dụng cùng mật khẩu mà ông đã gửi cho tôi trước đây với tệp đó. Nói chuyện với ông sau nhé.*

*Cảm ơn,*

*Joe*

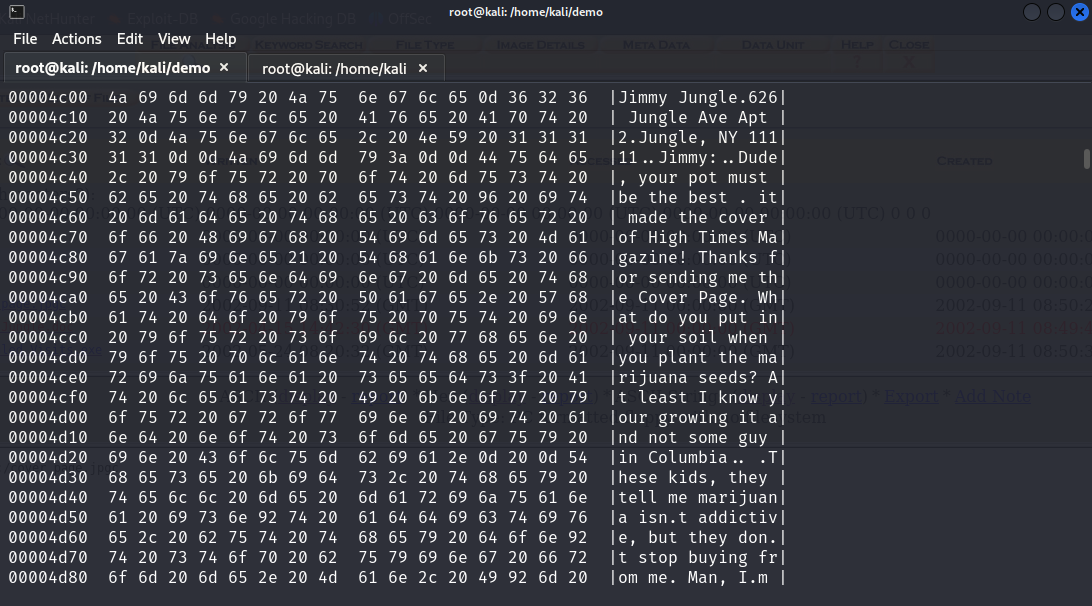
Từ đây ta có thể dễ dàng trả lời được người cung cấp cần sa cho Joe là ***Jimmy Jungle***, có địa chỉ tại ***626 Đại lộ Jungle, Căn hộ 2 Jungle, NY 11111***, và dữ liệu trong tệp coverpage quan trọng vì nó chứa mật khẩu về lịch trình mà Joe đang sử dụng.

Tiếp tục giải đố để tìm ra các câu hỏi tiếp theo. Từ lịch trình mà Joe nói tới, ta nghĩ đến mật khẩu được giấu trong coverpage. Ta sử dụng hexdump với file image để xem tất cả dữ liệu của đĩa vì file trích xuất ở autopsy đã bị hỏng dẫn đến việc Autopsy có thể bỏ qua hoặc không hiển thị chính xác những vùng này.



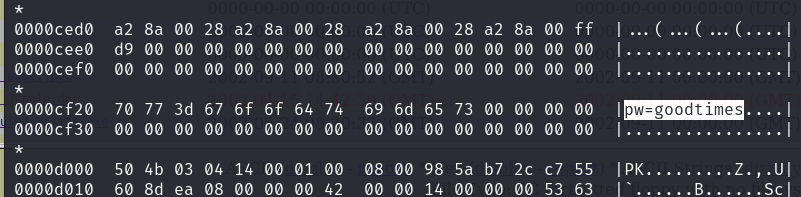
Hình 36. Sử dụng hexdump để xem tất cả dữ liệu của đĩa.

Trong output có xuất hiện cả những file ta đã được xem trên autopsy bởi dữ liệu đó không bị hỏng hoặc nằm ở vị trí không chuẩn nên ta vẫn có thể được dữ liệu hai bên là tương ứng.



Hình 37. Output có xuất hiện cả những file ta đã được xem trên Autopsy.

Trong đoạn output ta tìm thấy được đoạn có thể đọc được như sau và có thể dễ dàng nhận biết được pw ở đây có nghĩa là password. Vậy password ở đây là *goodtimes*



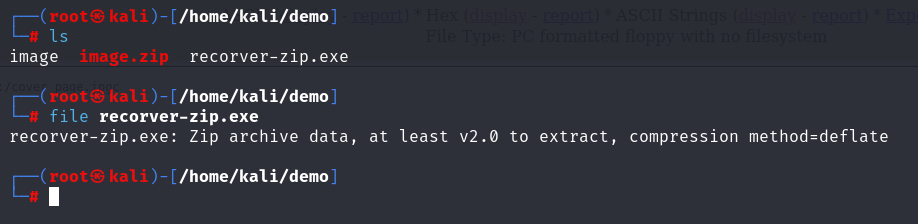
Hình 38. Tìm thấy password.

Ta trích xuất đĩa với *dd*:

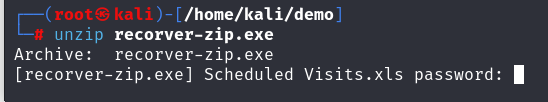


Hình 39. Trích xuất đĩa.

Ta có thể xác nhận file vừa tạo với đuôi .exe thực chất là một file zip với lệnh file:

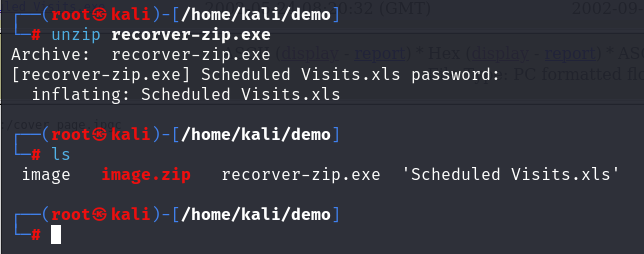


Tiến hành unzip file của trích xuất lại thì ta thấy khi unzip yêu cầu mật khẩu, ta nhập mật khẩu đã tìm được trước đó vào:



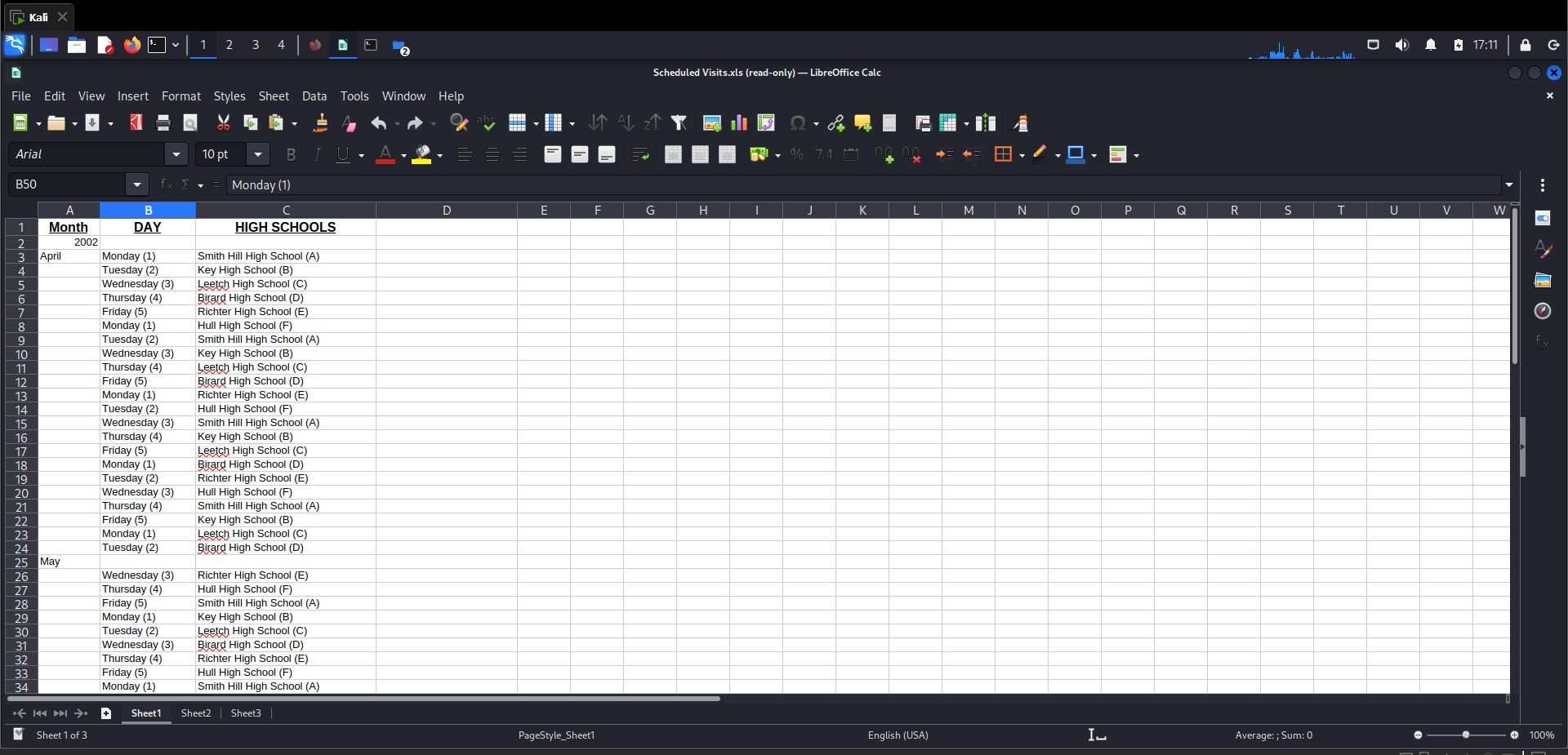
Hình 40. Unzip file trích xuất.

Sau khi unzip ta thấy được một file .xls:



Hình 41. File .xls được tìm thấy sau khi unzip.

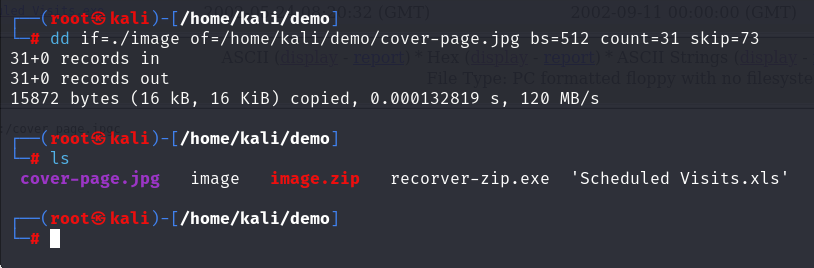
Mở file lên xem thì đây đúng là lịch trình của Joe:



Hình 42. Lịch trình của Joe được hiển thị sau khi mở file .xls.

Từ đây là lại dễ dàng trả lời cho các câu hỏi kế tiếp đó là các trường ngoài Smith Hill là: Key High School, Leetch High School, Birard High School, Richter High School, Hull High School. Các kỹ thuật mà Joe đã sử dụng đó là xóa file JimmyJungle.doc đi và ẩn mật khẩu trong file coverpage.jpgc.

Điều chúng ta phải làm bây giờ là thực hiện khôi phục lại bức ảnh đã bị hỏng. Dựa trên đoạn Hex đã thực hiện, ta cần tìm ra ranh giởi bắt đầu và kết thúc của nó để tiến hành khôi phục lại nó. Thực hiện với công cụ *dd* như đã sử dụng để nén lại ổ đĩa vậy.



Hình 43. Dùng dd để thực hiện khôi phục ảnh đã bị hỏng.

Và đây là hình ảnh được trích xuất:



Hình 44. Ảnh sau khi khôi phục.

Câu trả lời cho câu hỏi phụ rất đơn giản đó chính là Microsoft Word bới khi sử dụng *hexdump* ta đã thấy được phiên bản của nó.

### **2.2. Pháp chứng với The Sleuth Kit (TSK)**

Như đã thực hiện bằng autopsy, trong phần này, thay vì sử dụng Autopsy, nhóm em sẽ thực hiện phân tích với The Sleuth Kit (TSK) – một công cụ nền tảng mà Autopsy được xây dựng dựa trên. Nếu như Autopsy hướng đến sự tiện lợi với giao diện đồ họa trực quan, phù hợp cho những nhà điều tra cần thao tác nhanh mà không đòi hỏi kỹ năng dòng lệnh, thì TSK lại mang đến sự linh hoạt và kiểm soát toàn diện cho các chuyên gia pháp chứng số. Với khả năng làm việc trực tiếp trên dòng lệnh, TSK cho phép đi sâu hơn vào từng chi tiết trong quá trình phân tích.

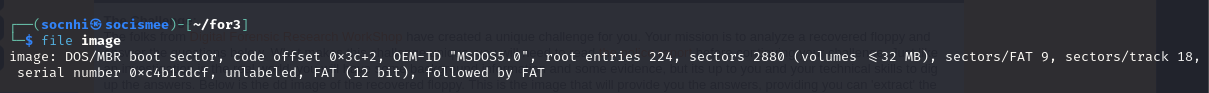
Đầu tiên, ta cũng tải file về tương tự như đã thực hiện với autopsy. Sau đó ta cũng sẽ dùng một số lệnh để xem loại file cũng như một số thông tin về file ta có được. Sau đây là hình ảnh về thông tin của file cần phân tích:

A screenshot of a computer program

Description automatically generated

Hình 45. Xem thông tin file sau khi đã tải về bằng lệnh exiftool.

Ta có thể chắc chắn rằng đây là một file nén, ta sẽ tiến hành giải nén file này ra:



Hình 46. File sau khi được giải nén.

Sau khi giải nén xong ta sẽ thấy có một số thông tin:

* *image*: DOS/MBR boot sector: thông tin này cho biết chúng ta đang phân tích một hình ảnh của sector khởi động (boot sector) của một ổ đĩa có hệ thống file DOS hoặc MBR (Master Boot Record). Boot sector là phần đầu tiên của ổ đĩa, chứa các thông tin quan trọng để khởi động hệ thống.
* *code offset 0x3c+2, OEM-ID "MSDOS5.0":*
* *code offset 0x3c+2*: Đây là vị trí của một giá trị quan trọng trong boot sector, thường chứa địa chỉ bắt đầu của bảng phân vùng (partition table).
* *OEM-ID "MSDOS5.0":* Đây là một chuỗi ký tự nhận dạng hệ điều hành, trong trường hợp này là MSDOS phiên bản 5.0.
* root entries 224: Số lượng mục nhập trong thư mục gốc của ổ đĩa. Mỗi mục nhập đại diện cho một file hoặc thư mục.
* sectors 2880 (volumes <=32 MB): Tổng số sector của ổ đĩa, trong trường hợp này giới hạn ở các ổ đĩa có dung lượng tối đa 32 MB.
* sectors/FAT 9: Số sector được sử dụng cho mỗi bảng phân bổ file (FAT). FAT là một cấu trúc dữ liệu quan trọng để quản lý các file trên ổ đĩa.
* sectors/track 18: Số sector trên mỗi track (vòng quay) của đĩa.
* serial number 0xc4b1cdcf: Số serial của ổ đĩa, một số nhận dạng duy nhất cho mỗi ổ đĩa.
* unlabeled: Ổ đĩa không có nhãn (label).
* FAT (12 bit): Ổ đĩa sử dụng bảng phân bổ file 12 bit, tức là mỗi mục nhập trong FAT chiếm 12 bit.
* followed by FAT: Sau phần thông tin trên là phần chứa bảng phân bổ file thực tế.

Tiếp theo ta dùng lệnh *mmls* để hiển thị các phân vùng của image, tuy nhiên nó không cho thông tin gì cả:

A computer screen with text

Description automatically generated

Hình 47. Dùng lệnh mmls để hiển thị các phân vùng của image.

Ta dùng lệnh fsstat để xem thông tin thống kê về hệ thống file trong image này và ta thấy một số thông tin quan trọng như dưới đây:

A screenshot of a computer

Description automatically generated

Hình 48. Dùng lệnh fsstat để xem thông tin thống kê về hệ thống file.

Đầu tiên là về thông tin của hệ thống file ta có thể thấy:

* File System Type: FAT12 - Đây là hệ thống tệp FAT đời đầu, thường được sử dụng phổ biến trên các thiết bị lưu trữ nhỏ.
* OEM Name: MSDOS5.0 - Phiên bản của hệ điều hành MS-DOS đã tạo hoặc định dạng hệ thống tệp.
* Volume ID: 0x4b1cdcf - Mã nhận dạng duy nhất của phân vùng hoặc ổ đĩa.

Tiếp đến là bố cục của hệ thông file:

* Total Range: 0 – 2879 (Tổng số sector từ đầu đến cuối của hệ thống tệp là 2880 sector).
* Reserved: 0 – 0 (Sector đầu tiên được dành riêng cho Boot Sector)
* Boot Sector: 0 (Đây là nơi chứa thông tin khởi động của hệ thống tệp).
* FAT 0: 1 - 9 (Là bản sao đầu tiên của bảng FAT (File Allocation Table), quản lý việc gán các cluster trong hệ thống tệp)
* FAT 1: 10 – 18 (Cũng giống với FAT 0 nhưng mà là bản sao dự phòng).
* Data Area: 19 – 2879 (Đây là nơi chứa dữ liệu thực tế).
* Root Directory: 19 – 32 (Nơi này chứa danh sách các file/directory cấp gốc).
* Cluster Area: 33 – 2879 (Dùng để lưu trữ dữ liệu thực tế của file).

Tiếp theo là phần thông tin Metadata:

* Range: 2 – 45782 (Đây là dải cluster được sử dụng để lưu trữ và quản lý siêu dữ liệu, bao gồm thông tin cấu trúc hệ thống tệp và vị trí tổ chức dữ liệu. Con số này đại diện cho phạm vi không gian lưu trữ dành riêng cho việc theo dõi và định nghĩa dữ liệu).
* Root Directory: 2 (Root directory bắt đầu tại cluster thứ 2).

Và thêm một phần nữa đáng chú ý là FAT CONTENTS:

* 73-103 (31) → EOF: Cluster 73 đến 103 chứa dữ liệu và kết thúc tại EOF (End of File). Nghĩa là có một file đang được lưu trong khu vực này.
* 104-108 (5) → EOF: Cluster 104 đến 108 cũng có chứa một file khác và nó cũng kết thúc tại EOF.

Ta tiếp tục dùng lệnh fls để liệt kê các tệp và thư mục trong đó:

A screen shot of a computer

Description automatically generated

Hình 49. Dùng lệnh fls để liệt kê các tệp và thư mục.

Sau khi đã liệt kê ra bằng fls chúng ta có thể thấy một số thông tin ta cần phải chú ý là những dòng:

* r/r \* 5: Jimmy Jungle.doc (Dấu \* cho biết rằng file này đã bị chỉnh sửa hoặc thay đổi metadata).
* r/r 8: cover page.jpgc (Đây là file ảnh, có thể chứa hình ảnh quan trọng liên quan đến vụ việc).
* r/r 11: Scheduled Visits.exe (Đây là file thực thi, có thể là chương trình độc hại hoặc công cụ hoặc nó có thể là một dạng file khác).

Sau khi trích xuất các file ra để xem thì ta có được bức thư có được thông tin như ảnh dưới đây, còn 2 file còn lại nội dung không có giá trị, thông tin này cho biết mật khẩu giải nén nằm trong page cover.

A screenshot of a computer

Description automatically generated

Hình 50. Có được bức thư sau khi trích xuất file ra để xem.

Ta sẽ tìm password của nó với hexdump:

A screenshot of a computer code

Description automatically generated

Hình 51. Tìm password bằng hexdump kết hợp với grep.

Xem thông tin của 2 file không có giá trị kia ta thấy được vị trí sector của nó, tuy nhiên file cover đã bị thay đổi, ta có được số lượng sector ban đầu để kiểm tra:

A screenshot of a computer program

Description automatically generated

Hình 52. Tìm thấy vị trí sector bằng lệnh istat.

Ta sẽ mount image này để xem xét kỹ hơn

A computer code with text

Description automatically generated with medium confidence

Hình 53. Mount image.

Đến file /mnt để xem và thấy được 2 file đã bị xóa, như vậy ta cần khôi phục file chứa lịch trình của Joe nằm bắt đầu từ sector 104

A screenshot of a computer

Description automatically generated

Hình 54. Tìm thấy 2 file đã bị xóa.

A screen shot of a computer

Description automatically generated

Hình 55. Khôi phục file chứa lịch trình của Joe nằm bắt đầu từ sector 104.

Giải nén file schedule.exe yêu cầu mật khẩu thì ta nhập mật khẩu đã tìm được ở bước trước đó thu được bảng excel

A screenshot of a computer program

Description automatically generated

Hình 56. Giải nén file Shedule và thu được bảng excel.

Mở file được thông tin sau:

A screenshot of a computer

Description automatically generated

Hình 57. File excel sau khi mở lên.

Khôi phục file ảnh ta làm tương tự, tuy nhiên với sector 451 lại không cho ra kết quả đúng, vì vậy ta dựa theo thông tin ban đầu thu được là từ sector 73

A computer screen shot of a computer code

Description automatically generated

Hình 58. Khôi phục ảnh.

Mở file ảnh để xem:

A screenshot of a computer

Description automatically generated

Hình 59. Ảnh sau khi đã được khôi phục.

Ta có thể thấy kết quả đưa ra giống với phần thực nghiệm bằng công cụ Autopsy.

# **KẾT LUẬN**

Trong bài tiểu luận này, chúng em đã hoàn thành và đạt được một số kết quả nhất định như sau:

*Về mặt lý thuyết:*

* Hiểu được pháp chứng kỹ thuật số là gì, các loại pháp chứng máy tính.
* Hiểu được thêm về hệ điều hành Linux.
* Tìm hiểu về các quy trình, kỹ thuật pháp chứng và các công cụ phổ biến của nó.

*Về mặt thực nghiệm:*

* Tìm hiểu cách sử dụng các công cụ pháp chứng trong Linux.
* Thực hiện triển khai được một thử thách điều tra mang tính thực tế.

# **TÀI LIỆU THAM KHẢO**

* Clound, CMC. 2023. *Hệ điều hành Linux là gì? Tổng hợp kiến thức cơ bản về Linux.* 10 1. Accessed 11 18, 2024. https://cmccloud.vn/tin-tuc/linux-la-gi-143
* Doo-Hyun, Kyr. 2020. *Pháp y kỹ thuật số là gì?* 11 30. Accessed 11 18, 2024. https://quantrimang.com/cong-nghe/phap-y-ky-thuat-so-la-gi-177235
* Kanade, Vijay. 2023. *Master Boot Record (MBR) vs. GUID Partition Table (GPT): What’s Best for You?* 7 11. Accessed 11 18, 2024. https://www.spiceworks.com/tech/hardware/articles/mbr-vs-gpt-the-best-choice-for-your-computer/
* Ly, Hồng. 2023. *Linux là gì | Tổng hợp kiến thức [A-Z] hệ điều hành Linux .* 12 1. Accessed 11 18, 2024. https://vinahost.vn/linux-la-gi/
* Nations, United. n.d. *Teaching Module Series.* Accessed 11 18, 2024. https://sherloc.unodc.org/cld/en/education/tertiary/cybercrime/module-4/index.html
* Nguyễn, Đang. 2019. *Tổng quan về filesystem trên Linux.* 1 5. Accessed 11 18, 2024. https://blogd.net/linux/tong-quan-ve-filesystem-tren-linux/
* Project, The Honeynet. n.d. *Scan 24.* Accessed 11 18, 2024. https://honeynet.onofri.org/scans/scan24/
* Việt, Trịnh Đình. 2020. *Hiểu rõ về quyền truy cập tệp trong linux.* 4 15. Accessed 11 18, 2024. https://viblo.asia/p/hieu-ro-ve-quyen-truy-cap-tep-trong-linux-07LKXJ8elV4