**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

-----🙞🙜🕮🙞🙜-----

**A blue diamond with a torch and a yellow flame

AI-generated content may be incorrect.**

**NIÊN LUẬN CHUYÊN NGÀNH AN TOÀN THÔNG TIN**

**TRIỂN KHAI VÀ NGHIÊN CỨU HỆ THỐNG SOC (SECURITY OPERATIONS CENTER) SỬ DỤNG NỀN TẢNG WAZUH STACK**

|  |
| --- |
| Giảng viên hướng dẫn:  **TS.GVC NGÔ BÁ HÚNG** |
| Sinh viên thực hiện  **NGUYỄN MINH THƯ**  **MSSV: B2203739**  **KHÓA 48** |

|  |
| --- |
| **CẦN THƠ - 2025** |

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

-----🙞🙜🕮🙞🙜-----

**A blue diamond with a torch and a yellow flame

AI-generated content may be incorrect.**

**NIÊN LUẬN CHUYÊN NGÀNH AN TOÀN THÔNG TIN**

**TRIỂN KHAI VÀ NGHIÊN CỨU HỆ THỐNG SOC (SECURITY OPERATIONS CENTER) SỬ DỤNG NỀN TẢNG WAZUH STACK**

|  |
| --- |
| Giảng viên hướng dẫn:  **TS.GVC NGÔ BÁ HÚNG** |
| Sinh viên thực hiện  **NGUYỄN MINH THƯ**  **MSSV: B2203739**  **KHÓA 48** |

|  |
| --- |
| **CẦN THƠ - 2025** |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

Mục lục

[Chương 1. GIỚI THIỆU 1](#_Toc216207270)

[1.1. Đặt vấn đề và lý do chọn đề tài 1](#_Toc216207271)

[1.2. Mục tiêu đề tài 2](#_Toc216207272)

[1.2.1. Mục tiêu tổng quát 2](#_Toc216207273)

[1.2.2. Mục tiêu cụ thể 2](#_Toc216207274)

[1.3. Đối tượng nghiên cứu 3](#_Toc216207275)

[1.3.1. Hệ thống Trung tâm Điều hành An ninh mạng (SOC) về mặt chức năng cốt lõi: 3](#_Toc216207276)

[1.3.2. Nền tảng SOC mã nguồn mở Wazuh và các thành phần chính: 4](#_Toc216207277)

[1.3.3. Các kỹ thuật bảo mật được tích hợp trong Wazuh để xây dựng lớp phòng thủ: 4](#_Toc216207278)

[1.4. Phạm vi nghiên cứu 5](#_Toc216207279)

[1.4.1. Phạm vi kỹ thuật và triển khai 5](#_Toc216207280)

[1.4.2. Phạm vi tấn công mô phỏng 5](#_Toc216207281)

[1.4.3. Phạm vi dữ liệu và dánh giá 6](#_Toc216207282)

[1.5. Phương pháp nghiên cứu 6](#_Toc216207283)

[1.6. Ý nghĩa khoa học và thực tiễn đề tài 7](#_Toc216207284)

[1.6.1. Ý nghĩa khoa học 7](#_Toc216207285)

[1.6.2. Ý nghĩa thực tiễn 7](#_Toc216207286)

[1.7. Bố cục báo cáo 7](#_Toc216207287)

[Chương 2. CƠ SỞ LÝ THUYẾT VỀ SOC VÀ NỀN TẢNG WAZUH 9](#_Toc216207288)

[2.1. Tổng quan về an ninh mạng và SOC 9](#_Toc216207289)

[2.1.1. Khái niệm an ninh mạng và các mối đe dọa phổ biến 9](#_Toc216207290)

[2.1.2. Khái niệm Security Operations Center (SOC) 10](#_Toc216207291)

[2.1.3. Quy trình vận hành SOC (thu thập, phân tích, phản hồi sự cố) 11](#_Toc216207292)

[2.2. Cơ sở lý thuyết về SIEM 13](#_Toc216207293)

[2.2.1. Khái niệm SIEM và vai trò trong SOC 13](#_Toc216207294)

[2.2.2. Các thành phần cơ bản của SIEM 15](#_Toc216207295)

[2.2.3. Khái niệm correlation và phân cấp mức độ cảnh báo 17](#_Toc216207296)

[2.3. Nền tảng Wazuh Stack 18](#_Toc216207297)

[2.3.1. Giới thiệu chung về Wazuh 18](#_Toc216207298)

[2.3.2. Kiến trúc tổng thể Wazuh Stack 19](#_Toc216207299)

[2.3.3. Thành phần Wazuh Manager – Decoders, Rules, Correlation 20](#_Toc216207300)

[2.3.4. Thành phần Wazuh Indexer – lưu trữ và index log/alert 21](#_Toc216207301)

[2.3.5. Thành phần Wazuh Dashboard – giao diện giám sát và phân tích 21](#_Toc216207302)

[2.3.6. Wazuh Agent – cơ chế thu thập log và gửi sự kiện 22](#_Toc216207303)

[2.4. Các chức năng an ninh chính trong Wazuh theo cấu trúc I-A-I-D 23](#_Toc216207304)

[2.4.1. HIDS – Host Intrusion Detection System 23](#_Toc216207305)

[2.4.2. FIM – File Integrity Monitoring 26](#_Toc216207306)

[2.4.3. Malware Detection dựa trên IOC 30](#_Toc216207307)

[2.4.4. Security Configuration Assessment (SCA) 34](#_Toc216207308)

[2.4.5. MITRE ATT&CK Mapping và Threat Hunting 37](#_Toc216207309)

[2.5. Mô hình phòng thủ theo chiều sâu (Defense in Depth) trong hệ thống SOC Wazuh 40](#_Toc216207310)

[2.5.1. Tổng quan về mô hình phòng thủ theo chiều sâu 40](#_Toc216207311)

[2.5.2. Layer 1 – Network & Perimeter (Router/NAT, Firewall, ACL) 41](#_Toc216207312)

[2.5.3. Layer 2 – Endpoint Security (Wazuh Agent: HIDS, FIM, Malware) 42](#_Toc216207313)

[2.5.4. Layer 3 – Hardening (Configuration Assessment – SCA) 42](#_Toc216207314)

[2.5.5. Layer 4 – Threat Intelligence & Behavior (MITRE, Threat Hunting) 43](#_Toc216207315)

[2.5.6. Layer 5 – SIEM & Correlation (Wazuh Manager, Indexer, Dashboard) 44](#_Toc216207316)

[Chương 3. phương pháp thực hiện 45](#_Toc216207317)

[3.1. Mô hình hệ thống SOC sử dụng nền tảng Wazuh Stack 45](#_Toc216207318)

[3.1.1. Mô tả môi trường thí nghiệm và các thành phần phần cứng/phần mềm 45](#_Toc216207319)

[3.1.2. Sơ đồ cấu hình và kết nối mạng hệ thống SOC Wazuh 47](#_Toc216207320)

[3.1.3. Kiến trúc chức năng theo 5 lớp phòng thủ (Network, Endpoint, Hardening, Threat, SIEM) 50](#_Toc216207321)

[3.1.4. Sơ đồ luồng log và sự kiện trong hệ thống 56](#_Toc216207322)

[3.2. Quy trình triển khai nền tảng Wazuh Stack 59](#_Toc216207323)

[3.2.1. Cài đặt Wazuh Stack trên máy chủ Ubuntu 24.04 59](#_Toc216207324)

[3.2.1. Cài đặt và đăng ký Wazuh Agent trên Windows Victim 61](#_Toc216207325)

[3.3. Thiết kế kịch bản kiểm th chức năng 62](#_Toc216207326)

[3.3.1. Nguyên tắc xây dựng kịch bản kiểm thử 62](#_Toc216207327)

[3.3.2. Mô tả chung 5 kịch bản kiểm thử 63](#_Toc216207328)

[Chương 4. KIỂM THỬ VÀ ĐÁNH GIÁ CÁC TÍNH NĂNG 67](#_Toc216207329)

[4.1. Tổng quan các kịch bản kiểm thử 67](#_Toc216207330)

[4.1.1. Mục tiêu kiểm thử hệ thống SOC Wazuh 67](#_Toc216207331)

[4.1.2. Danh sách và mô tả các tính năng kiểm thử 67](#_Toc216207332)

[4.1.3. Môi trường & Công cụ kiểm thử 68](#_Toc216207333)

[4.2. Kiểm thử tính năng HIDS – Giám sát đăng nhập sai trên Windows (SIEM + HIDS) 70](#_Toc216207334)

[4.2.1. Mô hình thí nghiệm và kịch bản đăng nhập sai liên tiếp trên Windows endpoint 70](#_Toc216207335)

[4.2.2. Quy trình thực nghiệm tạo sự kiện đăng nhập sai (Event ID 4625) và sinh log 71](#_Toc216207336)

[4.2.3. Kết quả ghi nhận log 4625 trên Wazuh Manager và hiển thị trên Dashboard 72](#_Toc216207337)

[4.2.4. Đánh giá khả năng phát hiện và cảnh báo đăng nhập sai/brute-force mô phỏng của HIDS/SIEM 74](#_Toc216207338)

[4.3. Kiểm thử tính năng FIM – Giám sát thay đổi file(FIM) 75](#_Toc216207339)

[4.3.1. Thiết lập thư mục FIM-Test và phạm vi giám sát trên Windows Victim 75](#_Toc216207340)

[4.3.2. 4.3.2. Quy trình tạo/sửa/xóa file để kiểm thử FIM 75](#_Toc216207341)

[4.3.3. Quy trình tạo/sửa/xóa file để kiểm thử FIM 76](#_Toc216207342)

[4.3.4. Kết quả cảnh báo FIM trên Wazuh Dashboard 80](#_Toc216207343)

[4.4. Kiểm thử tính năng Malware Detection dựa trên IOC 81](#_Toc216207344)

[4.4.1. Cấu hình danh sách IOC (malware-hashes, malicious-ip, malicious-domains) 81](#_Toc216207345)

[4.4.2. Quy trình tạo file “giả malware” và/hoặc mô phỏng kết nối đến IOC 83](#_Toc216207346)

[4.4.3. Kết quả cảnh báo Malware Detection trên Wazuh 84](#_Toc216207347)

[4.4.4. Đánh giá khả năng phát hiện, so sánh với cách tiếp cận antivirus truyền thống 86](#_Toc216207348)

[4.5. Kiểm thử tính năng Configuration Assessment 86](#_Toc216207349)

[4.5.1. Cấu hình rule SCA cho Windows Victim và Ubuntu server 86](#_Toc216207350)

[4.5.2. Quy trình chạy kiểm tra SCA và thu thập kết quả PASS/FAIL 87](#_Toc216207351)

[4.5.3. Phân tích một số cấu hình yếu điển hình được phát hiện 90](#_Toc216207352)

[4.6. Phân tích hành vi tấn công với MITRE ATT&CK và Threat Hunting 91](#_Toc216207353)

[4.6.1. Kịch bản tấn công dò mật khẩu SMB từ Kali 91](#_Toc216207354)

[4.6.2. Quan sát trên MITRE ATT&CK Dashboard 92](#_Toc216207355)

[4.6.3. Phân tích trên Threat Hunting Dashboard 93](#_Toc216207356)

[4.6.4. Truy vết chi tiết nguồn tấn công trên Discover 95](#_Toc216207357)

[4.6.5. Nhận xét về vai trò của MITRE ATT&CK và Threat Hunting trong demo 96](#_Toc216207358)

[4.7. Tổng hợp kết quả kiểm thử 97](#_Toc216207359)

[4.7.1. Bảng tổng hợp kiểm thử 97](#_Toc216207360)

[4.7.2. Đánh giá ưu điểm nổi bật của hệ thống 98](#_Toc216207361)

[4.7.3. Thảo luận về khả năng ứng dụng thực tế 99](#_Toc216207362)

[Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 102](#_Toc216207363)

[5.1. Kết quả đạt được 102](#_Toc216207364)

[5.2. Hạn chế 103](#_Toc216207365)

[5.2.1. Phạm vi mô hình và môi trường lab 103](#_Toc216207366)

[5.2.2. Hạn chế về thiết bị, công cụ và nguồn lực 103](#_Toc216207367)

[5.2.3. Hạn chế về kịch bản tấn công và nguồn log 104](#_Toc216207368)

[5.3. Hướng phát triển 104](#_Toc216207369)

[TÀI LIỆU THAM KHẢO 105](#_Toc216207370)

[A. Nguồn số liệu thống kê (Statistical Reports) 105](#_Toc216207371)

[B. Nguồn tài liệu học thuật (Academic References) 106](#_Toc216207372)

[C. Nguồn về kỹ thuật bảo mật (HIDS, FIM, Malware, Vulnerability) 106](#_Toc216207373)

[D. Nguồn về MITRE ATT&CK và Incident Response 106](#_Toc216207374)

[E. Nguồn về SOC, SIEM, Defense-in-Depth, Threat Intelligence 107](#_Toc216207375)

**Danh Mục Ảnh**

[Hinh 1: Mô hình CIA Triad 9](#_Toc216207224)

[Hình 2: cấu thành thệ thống SOC [7] 10](#_Toc216207225)

[Hình 3: Quy trình vận hành SOC 13](#_Toc216207226)

[Hình 4: Vai trò cảu SIEM 14](#_Toc216207227)

[Hình 5: Các thành phần của SIEM 17](#_Toc216207228)

[Hình 7: Nền tảng SIEM + XDR mã nguồn mở 18](#_Toc216207229)

[Hình 8: Thành phần Wazuh Stack 19](#_Toc216207230)

[Hình 9: kiến trúc Wazuh Stack 20](#_Toc216207231)

[Hình 10: Quá trình gom nhóm các Agent theo groups [10] 22](#_Toc216207232)

[Hình 12: So sánh HIDS & NIDS 24](#_Toc216207233)

[Hình 13: Quy trình hoạt động của HIDS trong Wazuh 25](#_Toc216207234)

[Hình 14: FIM trong các khung bảo mật và tiêu chuẩn quy định 27](#_Toc216207235)

[Hình 15: Quy trình hoạt động cảu FIM 29](#_Toc216207236)

[Hình 16. Malware Detection [12] 30](#_Toc216207237)

[Hình 16: Quy trình hoạt độngMalware Detection trên Wazuh 33](#_Toc216207238)

[Hình 17: Quy trình hoạt động SCA trongv Wazuh 36](#_Toc216207239)

[Hình 18: Quy trình hoạt độngMITRE ATT&CK & Threat Hunting 39](#_Toc216207240)

[Hình 19: Sơ đồ phần cứng phần mềm hệ thống 45](#_Toc216207241)

[Hình 20 : sơ đồ kết nối mạng hệ thống SOC 48](#_Toc216207242)

[Hình 21 : Sơ đồ kiến trúc chức năng 5 tầng 51](#_Toc216207243)

[Hình 22 : sơ đồ luồng log và sự kiện 56](#_Toc216207244)

[Hình 23: Trạng thái hoạt động của wazuh-manager 60](#_Toc216207245)

[Hình 24: trạng thái hoạt động của wazuh-indexer 60](#_Toc216207246)

[Hình 25: Trạng thái hoạt động của wazuh-dashboard 61](#_Toc216207247)

[Hình 26: các cổng hoạt động trên wazuh-stack 61](#_Toc216207248)

[Hình 28: trạng thái Wazuh-Agent hoạt động trên Windows 11 62](#_Toc216207249)

[Hình 29: Sơ đồ luồng tổng quát 68](#_Toc216207250)

[Hình 30: trạng thái của dịch vụ Wazuh Agent trên windows 70](#_Toc216207251)

[Hình 31: Kênh log sử dụng 71](#_Toc216207252)

[Hình 32: Xac minh kết nối Agent trên Wazuh manager 71](#_Toc216207253)

[Hình 33: Truy vấn từ hệ thống 73](#_Toc216207254)

[Hình 34: chi tiết sự kiện 78](#_Toc216207255)

[Hình 35: chi tiết log FIM 79](#_Toc216207256)

[Hình 36: kết quả FIM trên Wazuh Dashboard 80](#_Toc216207257)

[Hình 37: Thêm nội dung vào danh sách malware-hashes 82](#_Toc216207258)

[Hình 38: chuỗi SHA256 được sinh 83](#_Toc216207259)

[Hình 39: Xem mức độ cảnh báo của malware trên Overiew 84](#_Toc216207260)

[Hình 40: Lọc cảnh báo theo mức độ 84](#_Toc216207261)

[Hình 41: Lọc cảnh báo theo nhóm malware 85](#_Toc216207262)

[Hình 42: Kiểm tra SCA 88](#_Toc216207263)

[Hình 43: Thay để cấu hình trên máy victim 89](#_Toc216207264)

[Hình 44: Kiểm tra sao khi thay đổi cấu hình từ victim 90](#_Toc216207265)

[Hình 45: Tấn công từ kali 91](#_Toc216207266)

[Hình 46: Quan sát trên MITRE ATT&CK Dashboard 92](#_Toc216207267)

[Hình 47: Quan sát trên Theat Huting Dashboard 94](#_Toc216207268)

[Hình 48: truy vết nguồn tấn công 95](#_Toc216207269)

**Danh Mục Bảng**

[Bảng 1: bảng phân tách chức năng 50](#_Toc216207376)

[Bảng 2: Thông số kỹ thuật 69](#_Toc216207377)

[Bảng 3: Các trường dữ liệu 74](#_Toc216207378)

[Bảng 4: giá trị syscheck.event 80](#_Toc216207379)

[Bảng 6: tổng hợp kết quả kiểm thử 98](#_Toc216207380)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Viết đầy đủ | Nghĩa tiếng Việt |
| SOC | Security Operations Center | Trung tâm điều hành an ninh |
| SIEM | Security Information and Event Management | Hệ thống quản lý và phân tích sự kiện an ninh |
| HIDS | Host-based Intrusion Detection System | Hệ thống phát hiện xâm nhập dựa trên máy chủ/host |
| IDS | Intrusion Detection System | Hệ thống phát hiện xâm nhập |
| IPS | Intrusion Prevention System | Hệ thống ngăn chặn xâm nhập |
| FIM | File Integrity Monitoring | Giám sát toàn vẹn tệp tin |
| SCA | Security Configuration Assessment | Đánh giá cấu hình an ninh |
| IOC | Indicator of Compromise | Chỉ báo xâm nhập (dấu hiệu bị tấn công) |
| DoS | Denial of Service | Tấn công từ chối dịch vụ |
| NAT | Network Address Translation | Biên dịch địa chỉ mạng |
| ACL | Access Control List | Danh sách kiểm soát truy cập |
| LAN | Local Area Network | Mạng cục bộ |
| WAN | Wide Area Network | Mạng diện rộng |
| VM | Virtual Machine | Máy ảo |
| OS | Operating System | Hệ điều hành |
| IP | Internet Protocol | Giao thức liên mạng (địa chỉ IP) |
| MAC | Media Access Control | Địa chỉ vật lý (MAC) |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| RAM | Random Access Memory | Bộ nhớ truy cập ngẫu nhiên |
| GUI | Graphical User Interface | Giao diện đồ họa người dùng |
| CLI | Command Line Interface | Giao diện dòng lệnh |
| HTTP | Hypertext Transfer Protocol | Giao thức truyền siêu văn bản |
| HTTPS | Hypertext Transfer Protocol Secure | Giao thức truyền siêu văn bản bảo mật |
| DNS | Domain Name System | Hệ thống phân giải tên miền |
| MITRE ATT&CK | MITRE Adversarial Tactics, Techniques and Common Knowledge | Khung phân loại chiến thuật, kỹ thuật tấn công MITRE ATT&CK |

**LỜI CẢM ƠN**

Để hoàn thành niên luận này, em xin chân thành cảm ơn quý thầy cô trong khoa đã tận tình hướng dẫn, cung cấp những kiến thức bổ ích và sự hỗ trợ quý báu trong suốt quá trình nghiên cứu và hoàn thiện đề tài.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn Ngô Bá Hùng đã dành thời gian quý báu để định hướng, chỉ dẫn và đưa ra những góp ý thiết thực giúp em vượt qua những khó khăn, thử thách trong quá trình thực hiện.

Cuối cùng, em xin cảm ơn gia đình và bạn bè, những người luôn động viên, khích lệ và tạo mọi điều kiện tốt nhất để em hoàn thành tốt bài niên luận này.

Mặc dù đã rất cố gắng nhưng do thời gian và năng lực còn hạn chế, bài niên luận chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp và chỉ bảo của quý thầy để em có thể hoàn thiện hơn trong những lần nghiên cứu tiếp theo.

Em xin chân thành cảm ơn!

**TÓM TẮT**

Trong bối cảnh các hệ thống thông tin ngày càng đối mặt với nhiều mối đe dọa an ninh mạng, việc xây dựng và vận hành Trung tâm điều hành an ninh (Security Operations Center – SOC) đóng vai trò quan trọng trong giám sát, phát hiện và xử lý sự cố. Niên luận này tập trung vào việc triển khai và nghiên cứu mô hình SOC sử dụng nền tảng Wazuh Stack trong môi trường phòng lab.

Trên cơ sở lý thuyết về SOC, SIEM và các cơ chế giám sát an ninh, báo cáo xây dựng mô hình phòng thủ theo chiều sâu (Defense in Depth) gồm 5 lớp: Network & Perimeter, Endpoint Security, Hardening, Threat Intelligence & Behavior, và SIEM & Correlation. Hệ thống được triển khai với Wazuh Manager, Wazuh Indexer, Wazuh Dashboard trên máy chủ Ubuntu và Wazuh Agent trên máy Windows đóng vai trò “victim”.

Phần thực nghiệm tập trung kiểm thử và đánh giá các tính năng chính của Wazuh: HIDS (giám sát và phát hiện đăng nhập, brute-force), FIM (giám sát thay đổi tệp tin), Malware Detection/Rootcheck dựa trên IOC (hash, IP, domain độc hại), Security Configuration Assessment (đánh giá cấu hình an ninh), cùng khả năng MITRE ATT&CK Mapping và Threat Hunting trên Dashboard. Các kịch bản tấn công và thao tác thử nghiệm được thiết kế trong môi trường lab, log và cảnh báo được thu thập, phân tích để đánh giá khả năng phát hiện, mức độ chi tiết và tính hữu ích cho công tác điều tra sự cố.

Kết quả cho thấy mô hình SOC sử dụng Wazuh Stack đáp ứng tốt mục tiêu giám sát tập trung, phát hiện sự kiện bất thường trên endpoint và hỗ trợ phân tích, truy vết trong phạm vi môi trường thí nghiệm, đồng thời gợi mở khả năng mở rộng áp dụng cho các mô hình đào tạo và tổ chức quy mô nhỏ.

**ABSTRACT**

In the context of increasing cyber threats against information systems, building and operating a Security Operations Center (SOC) plays a crucial role in monitoring, detecting, and responding to security incidents. This thesis focuses on deploying and studying an SOC model based on the Wazuh Stack platform in a laboratory environment.

Grounded on the theoretical background of SOC, SIEM, and security monitoring mechanisms, the report proposes and implements a five-layer Defense in Depth model: Network & Perimeter, Endpoint Security, Hardening, Threat Intelligence & Behavior, and SIEM & Correlation. The system is deployed with Wazuh Manager, Wazuh Indexer, and Wazuh Dashboard on an Ubuntu server, while a Windows machine acts as the victim endpoint running the Wazuh Agent.

The experimental part concentrates on testing and evaluating Wazuh’s key security features: Host-based Intrusion Detection System (HIDS) for monitoring and detecting login and brute-force activities; File Integrity Monitoring (FIM) for tracking critical file changes; Malware Detection/Rootcheck based on Indicators of Compromise (IOC) such as malicious hashes, IP addresses, and domains; Security Configuration Assessment (SCA) for evaluating security configurations; and the capabilities of MITRE ATT&CK Mapping and Threat Hunting on the Wazuh Dashboard. Test scenarios and attack simulations are executed in the lab environment, with logs and alerts collected and analyzed to assess detection capability, level of detail, and usefulness for incident investigation.

The results indicate that the SOC model built on Wazuh Stack can effectively provide centralized monitoring, anomaly detection on endpoints, and support for analysis and investigation within the scope of the experimental environment, while suggesting its potential for training laboratories and small-scale organizations.

# GIỚI THIỆU

## Đặt vấn đề và lý do chọn đề tài

Trong kỷ nguyên chuyển đổi số, an ninh mạng nổi lên như thách thức trọng yếu đối với sự tồn vong của tổ chức. Báo cáo DBIR 2024 của Verizon ghi nhận 30.458 sự cố an ninh và 10.626 vi phạm dữ liệu được xác nhận, trong đó 68% liên quan trực tiếp đến yếu tố con người và khoảng 15% xuất phát từ chuỗi cung ứng/bên thứ ba [1]. Tài chính – Bảo hiểm tiếp tục là ngành mục tiêu hàng đầu.

Về tác động, báo cáo IBM Cost of a Data Breach 2023 cho thấy chi phí trung bình mỗi vụ vi phạm đạt 4,45 triệu USD, với vòng đời sự cố lên đến 277 ngày (204 ngày phát hiện, 73 ngày khắc phục) [2]. Các tổ chức cũng ghi nhận chi phí vi phạm cao hơn đáng kể trong môi trường cloud, đặc biệt là hybrid cloud. Xu hướng tấn công tiếp tục leo thang: theo Unit 42 Ransomware Threat Report 2024, mức tiền chuộc trung bình mà nạn nhân phải trả khoảng 1,1 triệu USD và đang có xu hướng tăng [3]. Báo cáo 2024 của Secureframe cũng chỉ ra rằng trung bình một tổ chức phải đối mặt với gần 1.900 cuộc tấn công mạng mỗi tuần, cho thấy cường độ tấn công ngày càng dày đặc [4].

Tại Việt Nam, tình hình an ninh mạng những năm gần đây diễn biến phức tạp với tần suất và mức độ tinh vi gia tăng. Theo thống kê của Cục An toàn thông tin, trong 8 tháng đầu năm 2024 đã ghi nhận 4.029 sự cố, nâng tổng số lên 4.483 vụ đến hết tháng 10/2024. Mặc dù giảm hơn 57% so với cùng kỳ năm 2023, số sự cố nghiêm trọng lại tăng gần 60% [5]. Bên cạnh đó, theo đánh giá từ các hệ thống giám sát trong nước, năm 2024 ghi nhận hơn 155.600 máy tính bị tấn công ransomware và hơn 90.000 điểm yếu, lỗ hổng tồn tại trong hệ thống của các cơ quan, tổ chức [6].

Những số liệu trên cho thấy một thực tế rõ ràng: các biện pháp phòng thủ rời rạc, thụ động không còn phù hợp trước bức tranh đe dọa đa tầng và tinh vi. Do đó, việc thiết lập Trung tâm Điều hành An ninh Mạng (SOC) cần được xem như một nền tảng chiến lược then chốt, cung cấp khả năng giám sát liên tục 24/7, phân tích phát hiện sớm mối đe dọa, ứng phó nhanh chóng và phối hợp hiệu quả giữa ba yếu tố: con người – quy trình – công nghệ. Mục tiêu trọng yếu của SOC là giảm thiểu thời gian phát hiện (MTTD) và thời gian ứng phó (MTTR), từ đó hạn chế thiệt hại và bảo vệ tài sản số của tổ chức.

Tuy nhiên, đối với nhiều tổ chức – đặc biệt là doanh nghiệp vừa và nhỏ hoặc cơ sở đào tạo – rào cản lớn nhất để triển khai SOC hiệu quả nằm ở chi phí đầu tư cao cho các giải pháp thương mại và yêu cầu đội ngũ vận hành chuyên trách. Trong bối cảnh đó, Wazuh, một nền tảng SOC mã nguồn mở, nổi lên như lựa chọn khả thi với khả năng tích hợp các chức năng quan trọng bao gồm HIDS, FIM, log analysis, vulnerability detection, malware detection và tuân thủ bảo mật, tất cả trong một kiến trúc tập trung duy nhất.

## Mục tiêu đề tài

### Mục tiêu tổng quát

Xây dựng, triển khai và đánh giá một hệ thống Trung tâm Điều hành An ninh mạng (SOC) quy mô phòng thí nghiệm dựa trên nền tảng mã nguồn mở **Wazuh**. Hệ thống được thiết kế theo chiến lược **phòng thủ theo chiều sâu (Defense-in-Depth)**, nhằm chứng minh tính khả thi và đánh giá hiệu quả của một mô hình SOC mã nguồn mở trong việc cung cấp khả năng giám sát tập trung, phát hiện sớm dấu hiệu xâm nhập, phân tích hành vi tấn công và hỗ trợ quy trình phản ứng sự cố trong môi trường thực nghiệm.

### Mục tiêu cụ thể

1. **Nghiên cứu cơ sở lý thuyết**

Tìm hiểu kiến trúc, nguyên lý hoạt động và các quy trình vận hành cốt lõi của một SOC hiện đại.

Phân tích vai trò và chức năng của các thành phần trong hệ sinh thái Wazuh (Manager, Indexer, Dashboard, Agent) và mối liên hệ với các khái niệm:  
SIEM, HIDS, FIM, Malware Detection, Configuration Assessment và Threat Intelligence.

1. **Triển khai hạ tầng giám sát**

Xây dựng Wazuh Stack (Manager – Indexer – Dashboard) trên Ubuntu Server, đóng vai trò trung tâm thu thập, xử lý và trực quan hóa dữ liệu.

Cài đặt và cấu hình Wazuh Agent trên các endpoint Windows để thu thập log, sự kiện bảo mật và giám sát hoạt động tại điểm cuối.

1. **Thiết lập các lớp phòng thủ chuyên sâu**

Cấu hình File Integrity Monitoring (FIM) để phát hiện thay đổi trái phép trên các tệp tin trọng yếu.

Thiết lập Host-based Intrusion Detection System (HIDS) và Malware Detection (Rootcheck) nhằm giám sát hành vi bất thường và dấu hiệu nhiễm mã độc.

Thực hiện Security Configuration Assessment (SCA) theo tiêu chuẩn CIS Benchmark để phát hiện lỗ hổng cấu hình.

Tích hợp Threat Intelligence thông qua ánh xạ cảnh báo vào khung MITRE ATT&CK và kích hoạt khả năng Threat Hunting.

1. **Thực nghiệm và đánh giá**

Mô phỏng các kịch bản tấn công thực tế từ máy Kali Linux (Brute-force, thao tác trái phép với tệp, hành vi nghi vấn, mã độc) để kiểm thử hệ thống.

Phân tích và đánh giá các cảnh báo trên Wazuh Dashboard, kiểm chứng khả năng phát hiện, độ chính xác và thời gian phản hồi.

Đánh giá toàn diện hiệu quả vận hành của mô hình SOC Wazuh, nêu rõ ưu điểm, hạn chế và tính khả thi; đề xuất hướng tối ưu và mở rộng trong tương lai.

## Đối tượng nghiên cứu

### Hệ thống Trung tâm Điều hành An ninh mạng (SOC) về mặt chức năng cốt lõi:

Đối tượng thứ nhất là mô hình hoạt động cơ bản của một Trung tâm Điều hành An ninh mạng (SOC), với các chức năng then chốt:

* **Giám sát an ninh tập trung:**  
  Thu thập và tổng hợp log/sự kiện bảo mật từ nhiều hệ thống (endpoint, hệ điều hành, ứng dụng) về một điểm tập trung.
* **Phát hiện và cảnh báo mối đe dọa:**  
  Phân tích dữ liệu, áp dụng rule phát hiện để nhận diện bất thường, hành vi tấn công và sinh cảnh báo kịp thời.
* **Phân tích sự kiện và điều tra sự cố:**  
  Hỗ trợ tra cứu log, truy vết chuỗi sự kiện, xác định nguyên nhân – phạm vi ảnh hưởng của sự cố.
* **Hỗ trợ phản ứng sự cố:**  
  Cung cấp thông tin đầu vào cho quy trình xử lý sự cố (ngăn chặn, khắc phục, phục hồi) dù trong phạm vi đề tài chỉ dừng lại ở mức **hỗ trợ phân tích**, chưa xây dựng playbook IR hoàn chỉnh.

### Nền tảng SOC mã nguồn mở Wazuh và các thành phần chính:

Đối tượng trọng tâm thứ hai là nền tảng **Wazuh**, được nghiên cứu thông qua kiến trúc và vai trò của từng thành phần:

**Wazuh Manager:** Thành phần xử lý trung tâm, thực thi rule phát hiện, tương quan sự kiện (correlation) và sinh cảnh báo.

**Wazuh Indexer:** Hệ thống lưu trữ và đánh chỉ mục dữ liệu log/alert, phục vụ truy vấn và phân tích.

**Wazuh Dashboard:** Giao diện web dùng để trực quan hóa log, alert, ma trận MITRE ATT&CK và hỗ trợ thao tác phân tích.

**Wazuh Agent:** Được cài đặt trên các endpoint (trong đề tài là máy Windows Victim), thu thập log/sự kiện và thực thi các tác vụ như FIM, SCA, IOC matching theo cấu hình từ Manager.

### Các kỹ thuật bảo mật được tích hợp trong Wazuh để xây dựng lớp phòng thủ:

Đối tượng thứ ba là các cơ chế bảo mật cụ thể mà Wazuh cung cấp để hiện thực hóa mô hình **phòng thủ nhiều lớp (Defense in Depth)** trong môi trường lab:

**HIDS – Host-based Intrusion Detection System:** Giám sát các sự kiện bảo mật trên endpoint (đăng nhập thất bại, thay đổi quyền, tiến trình bất thường…).

**FIM – File Integrity Monitoring:** Theo dõi thay đổi trên file và registry, đặc biệt là thư mục và cấu hình nhạy cảm (trong lab: thư mục FIM-Test trên Windows).

**Malware Detection / Rootcheck dựa trên IOC:** Phát hiện file, đường dẫn, IP, domain đáng ngờ bằng cách so khớp với danh sách IOC do người quản trị định nghĩa; kiểm tra các dấu hiệu rootkit và cấu hình hệ thống bất thường.

**SCA – Security Configuration Assessment:** Tự động kiểm tra và báo cáo mức tuân thủ các thiết lập bảo mật trên Windows so với các khuyến nghị hardening (policy SCA do Wazuh cung cấp).

**Tích hợp MITRE ATT&CK và hỗ trợ Threat Hunting:** Ánh xạ alert vào tactic/kỹ thuật của MITRE ATT&CK, cung cấp nền tảng để truy vấn, mở rộng điều tra và săn tìm mối đe dọa dựa trên dữ liệu log/alert thu được.

## Phạm vi nghiên cứu

### Phạm vi kỹ thuật và triển khai

**Môi trường thí nghiệm:**  
Hệ thống SOC được triển khai trong môi trường ảo hóa (VirtualBox), quy mô nhỏ, bao gồm:

* 01 máy chủ Ubuntu 24.04 cài đặt Wazuh Manager, Wazuh Indexer và Wazuh Dashboard.
* 01 máy Windows 11 đóng vai trò endpoint/victim, cài đặt Wazuh Agent.
* 01 máy Kali Linux đóng vai trò máy tấn công, cùng mạng với Windows Victim.

**Chức năng trọng tâm:**  
Đề tài tập trung cấu hình, vận hành và đánh giá các module:

* HIDS (phân tích log bảo mật Windows).
* FIM (giám sát thay đổi file).
* Malware Detection / Rootcheck dựa trên IOC nội bộ.
* SCA (đánh giá cấu hình bảo mật).
* Tích hợp MITRE ATT&CK và chức năng SIEM cơ bản trên Wazuh Dashboard.

**Giới hạn:**

* Không triển khai các tính năng SIEM nâng cao, SOAR (Security Orchestration, Automation and Response) hoặc tích hợp với hệ thống bên thứ ba (MISP, OTX, ticketing…).
* Không xây dựng kịch bản phản ứng tự động, chỉ dừng ở mức phát hiện, hiển thị và phân tích thủ công trên Dashboard.

### Phạm vi tấn công mô phỏng

**Các kịch bản tấn công cơ bản trong lab:**  
Nghiên cứu chỉ thực hiện các kịch bản đã được kiểm soát, phục vụ trực tiếp cho việc kiểm thử chức năng của Wazuh, bao gồm:

* Đăng nhập sai nhiều lần (brute-force đơn giản) từ máy Kali vào dịch vụ SMB/Windows để sinh sự kiện **4625 – An account failed to log on**.
* Thao tác tạo/sửa/xóa file trong thư mục được FIM giám sát trên Windows để kiểm tra cảnh báo FIM.
* Tạo file test có hash trùng với IOC đã khai báo trong Wazuh để kích hoạt cảnh báo Malware Detection/Rootcheck.
* Kích hoạt quét SCA trên Windows Victim để sinh báo cáo cấu hình yếu.
* Thực hiện một chuỗi hoạt động từ Kali (scan → đăng nhập sai → thao tác file…) để minh họa mapping MITRE ATT&CK và quy trình threat hunting.

**Giới hạn:**

* Không mô phỏng APT, tấn công zero-day hoặc chiến dịch tấn công quy mô lớn.
* Không triển khai khai thác lỗ hổng thực tế trên dịch vụ hoặc hệ điều hành; toàn bộ tấn công chỉ dừng ở mức tạo sự kiện an toàn trong lab để kiểm thử chức năng phát hiện.

### Phạm vi dữ liệu và dánh giá

* Phân tích log và các cảnh báo thu thập từ Wazuh Agent và hiển thị trên Wazuh Dashboard.
* Đánh giá hệ thống theo các tiêu chí: Khả năng phát hiện (Detection Capability), tốc độ phản hồi, độ chính xác cảnh báo và tính khả thi triển khai.
* Không bao gồm phân tích pháp y số (Digital Forensics) chuyên sâu hoặc quy trình IR đầy đủ theo tiêu chuẩn tổ chức.

## Phương pháp nghiên cứu

* + - * **Nghiên cứu tài liệu:** Thu thập, phân tích tài liệu về kiến trúc SOC, công nghệ Wazuh, các chuẩn CIS, MITRE ATT&CK để xây dựng cơ sở lý thuyết.
      * **Thiết kế hệ thống:** Mô hình hóa kiến trúc SOC Wazuh, sơ đồ mạng và các lớp phòng thủ Defense-in-Depth để định hướng triển khai.
      * **Thực nghiệm:** Cài đặt Wazuh Stack, cấu hình HIDS/FIM/SCA và mô phỏng tấn công từ Kali Linux để kiểm thử chức năng.
      * **Phân tích đánh giá:** Phân tích cảnh báo, ánh xạ MITRE ATT&CK, đánh giá hiệu quả phát hiện và tính khả thi của hệ thống.

## Ý nghĩa khoa học và thực tiễn đề tài

### Ý nghĩa khoa học

Hệ thống hóa cơ sở lý luận: Góp phần củng cố và hệ thống hóa kiến thức về kiến trúc Trung tâm Điều hành An ninh mạng (SOC), cũng như các công nghệ phòng thủ cốt lõi như SIEM, HIDS, FIM và Threat Intelligence.

Minh chứng mô hình thực nghiệm: Cung cấp một mô hình tham chiếu cụ thể cho việc nghiên cứu và ứng dụng các giải pháp mã nguồn mở trong giám sát an ninh mạng, chứng minh tính khả thi của chiến lược phòng thủ chiều sâu (Defense-in-Depth).

Tiền đề cho các nghiên cứu nâng cao: Mở ra hướng nghiên cứu sâu hơn về phân tích hành vi tấn công dựa trên dữ liệu log, kỹ thuật ánh xạ khung MITRE ATT&CK và xu hướng tự động hóa phản ứng trong SOC (SOAR).

### Ý nghĩa thực tiễn

**Giải pháp tối ưu chi phí:** Đề xuất một mô hình SOC hiệu quả với chi phí bản quyền bằng 0 (Open Source), phù hợp để triển khai tại các doanh nghiệp vừa và nhỏ (SMEs) hoặc các cơ sở đào tạo hạn chế về ngân sách.

**Nâng cao năng lực vận hành:** Là tài liệu tham khảo giá trị hỗ trợ quá trình thực hành, giúp người học và kỹ sư an ninh mạng nắm bắt quy trình vận hành SOC, từ cấu hình hệ thống đến phân tích cảnh báo.

**Tăng cường khả năng phòng thủ:** Giúp tổ chức phát hiện sớm các nguy cơ an ninh phổ biến nhưng nguy hiểm như tấn công dò mật khẩu (Brute-force), thay đổi tệp tin hệ thống trái phép và nhiễm mã độc, từ đó giảm thiểu rủi ro và thiệt hại.

**Nâng cao nhận thức bảo mật:** Góp phần khẳng định vai trò không thể thiếu của giám sát tập trung trong bối cảnh an ninh mạng hiện đại.

## Bố cục báo cáo

* **Chương 1 – Tổng quan đề tài**  
  Trình bày bối cảnh, lý do chọn đề tài, mục tiêu, đối tượng – phạm vi nghiên cứu, phương pháp nghiên cứu, ý nghĩa và bố cục báo cáo.
* **Chương 2 – Cơ sở lý thuyết**  
  Trình bày lý thuyết về an ninh mạng, SOC, SIEM, nền tảng Wazuh Stack và các chức năng HIDS, FIM, Malware Detection/Rootcheck, SCA, MITRE ATT&CK, Threat Hunting, cùng mô hình Defense in Depth.
* **Chương 3 – Phương pháp thực hiện**  
  Mô tả mô hình hệ thống SOC Wazuh trong lab, cấu hình phần cứng/phần mềm, sơ đồ mạng, quy trình cài đặt – cấu hình Wazuh Stack và Wazuh Agent, cùng cách thiết kế kịch bản kiểm thử.
* **Chương 4 – Kiểm thử và đánh giá các tính năng**  
  Trình bày quá trình kiểm thử và kết quả đánh giá các tính năng HIDS, FIM, Malware Detection/Rootcheck, SCA, MITRE ATT&CK và Threat Hunting trên hệ thống SOC Wazuh.
* **Chương 5 – Kết luận và hướng phát triển**  
  Tổng hợp kết quả đạt được, nêu các hạn chế của đề tài và đề xuất các hướng phát triển, mở rộng hệ thống trong tương lai.

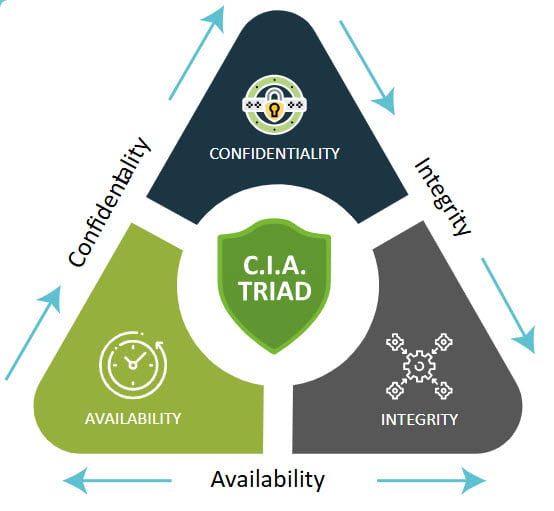
# CƠ SỞ LÝ THUYẾT VỀ SOC VÀ NỀN TẢNG WAZUH

## Tổng quan về an ninh mạng và SOC

### Khái niệm an ninh mạng và các mối đe dọa phổ biến

1. **Khái niệm an ninh mạng**

An ninh mạng (Cybersecurity) là tập hợp các biện pháp kỹ thuật, quy trình và công cụ nhằm bảo vệ hệ thống, dữ liệu và tài nguyên số khỏi truy cập trái phép, phá hoại hoặc tấn công [18].  
Mục tiêu cốt lõi được xác định theo **mô hình CIA Triad: Confidentiality – Integrity – Availability**, một khái niệm được định nghĩa trong NIST SP 800-12 và SP 800-61 [18], [19].



Hinh 1: Mô hình CIA Triad

Nội dung mở rộng về yếu tố con người, quản trị rủi ro và chuỗi cung ứng được ghi nhận trong các báo cáo của ENISA và Verizon DBIR [1], [20].

1. **Các mối đe dọa phổ biến**

Dựa trên **Verizon DBIR 2024** và **ENISA Threat Landscape 2023/2024**, một số mối đe dọa chính bao gồm:

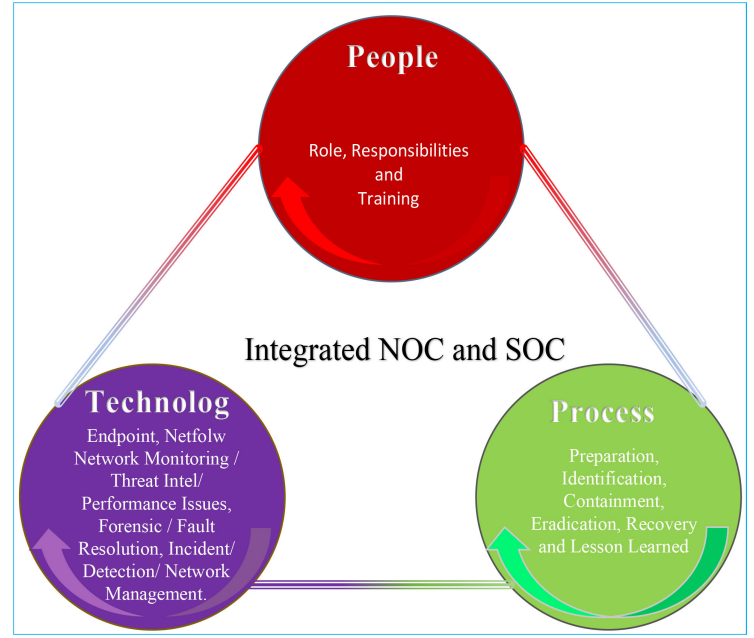
* **Malware/Ransomware**: một trong các kỹ thuật phổ biến nhất gây ra sự cố nghiêm trọng.
* **Khai thác lỗ hổng (Exploitation)**: tấn công dựa vào phần mềm chưa vá, cấu hình lỗi [20].
* **Social Engineering/Phishing** – chiếm **68%** sự cố theo DBIR 2024 [1].
* **Brute-force/Credential Stuffing** – ghi nhận thường xuyên trong thống kê MITRE và DBIR [1], [17].
* **Misconfiguration** – nguyên nhân gia tăng mạnh ở hệ thống cloud theo IBM 2023.
* **Insider Threat** – được nhấn mạnh trong SANS SOC Survey 2023.
* **Supply Chain Attacks** – chiếm ~15% theo DBIR 2024.

Điểm chung của các mối đe dọa này là **đều để lại dấu vết (log, hành vi hệ thống)** – cơ sở để SOC thu thập và phát hiện .

### Khái niệm Security Operations Center (SOC)

1. **Khái niệm chung**

SOC là một đơn vị vận hành an ninh mạng, kết hợp **con người – quy trình – công nghệ**, thực hiện giám sát, phát hiện và ứng phó sự cố 24/7. Khái niệm này được chuẩn hóa trong NIST SP 800-61, ENISA và các nghiên cứu IEEE về SOC [7].



Hình 2: cấu thành thệ thống SOC [7]

1. **Chức năng chính của SOC**

Theo NIST, SANS và các mô hình SOC hiện đại [18], [21], chức năng chính gồm: Centralized Monitoring , Threat Detection & Analysis, Alerting & Triage, Incident Investigation, Incident Response, Reporting & Compliance

Trong đề tài, SOC tập trung vào 3 thành phần cốt lõi:  
**Giám sát – Phát hiện – Cảnh báo.**

1. **Các mô hình SOC**

Phân loại mô hình SOC dựa theo ENISA và các nghiên cứu IEEE:

* SOC nội bộ
* SOC thuê ngoài (MSSP)
* SOC lai (Hybrid SOC)
* SOC đám mây
* SOC phòng thí nghiệm (Lab-scale SOC)

Đề tài xây dựng mô hình **Lab-scale SOC mã nguồn mở dựa trên Wazuh**.

### Quy trình vận hành SOC (thu thập, phân tích, phản hồi sự cố)

Theo **NIST Incident Handling Life Cycle** (SP 800-61r2), quy trình SOC gồm:

1. **Thu thập Dữ liệu (Data Collection)**

Nguồn dữ liệu bao gồm:

* **System Logs** (Windows Event Log, Syslog Linux)
* **Application Logs**
* **Network Logs** (Firewall, IDS/IPS)
* **Endpoint Telemetry** thông qua agent
* **Threat Intelligence** (IOC, MITRE ATT&CK)

Trong Wazuh, phần lớn dữ liệu được thu qua **Wazuh Agent** và các module thu thập của **Wazuh Manager**.

1. **Phân tích & Phát hiện (Analysis & Detection)**

Các phương pháp chính gồm:

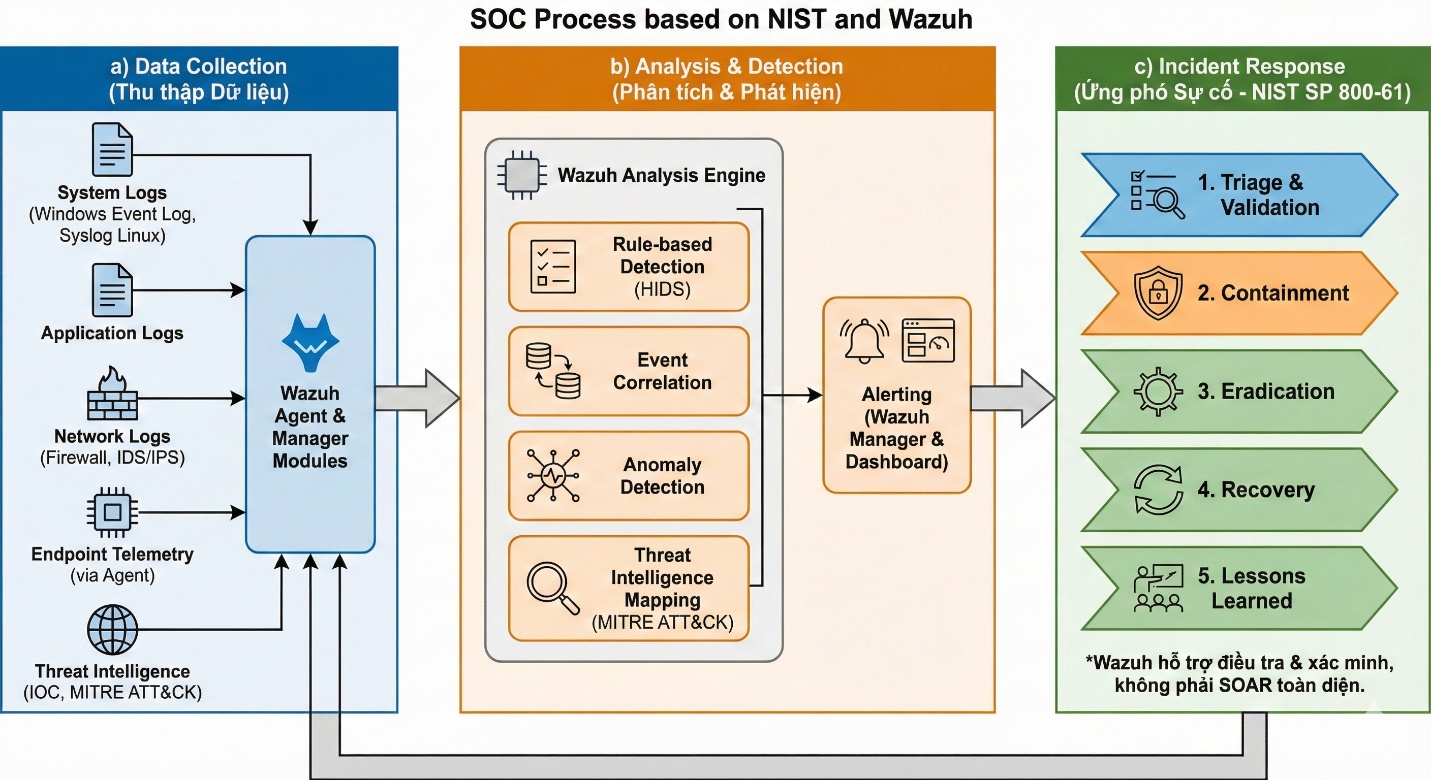
* **Rule-based Detection** – cơ chế chính của Wazuh HIDS.
* **Event Correlation** – tương quan sự kiện theo kiến trúc SOC của NIST và SANS.
* **Anomaly Detection** – mô tả trong HIDS và log analysis của Wazuh.
* **Threat Intelligence Mapping** – Wazuh hỗ trợ MITRE ATT&CK .
* **Alerting** – sinh cảnh báo qua Wazuh Manager và hiển thị trên Dashboard.

**c) Ứng phó Sự cố (Incident Response)**

Bám theo chu trình NIST SP 800-61 [18]:

* **Triage & Validation**
* **Containment**
* **Eradication**
* **Recovery**
* **Lessons Learned**

Wazuh không phải nền tảng SOAR nhưng hỗ trợ phần lớn giai đoạn điều tra và xác minh sự cố.



Hình 3: Quy trình vận hành SOC

## Cơ sở lý thuyết về SIEM

### Khái niệm SIEM và vai trò trong SOC

1. **Khái niệm SIEM**

Security Information and Event Management (SIEM) là giải pháp cho phép thu thập, chuẩn hóa, lưu trữ và phân tích nhật ký (log) nhằm phát hiện hoạt động bất thường và hỗ trợ ứng phó sự cố.

Theo Gartner, SIEM là sự kết hợp của hai nhóm chức năng:

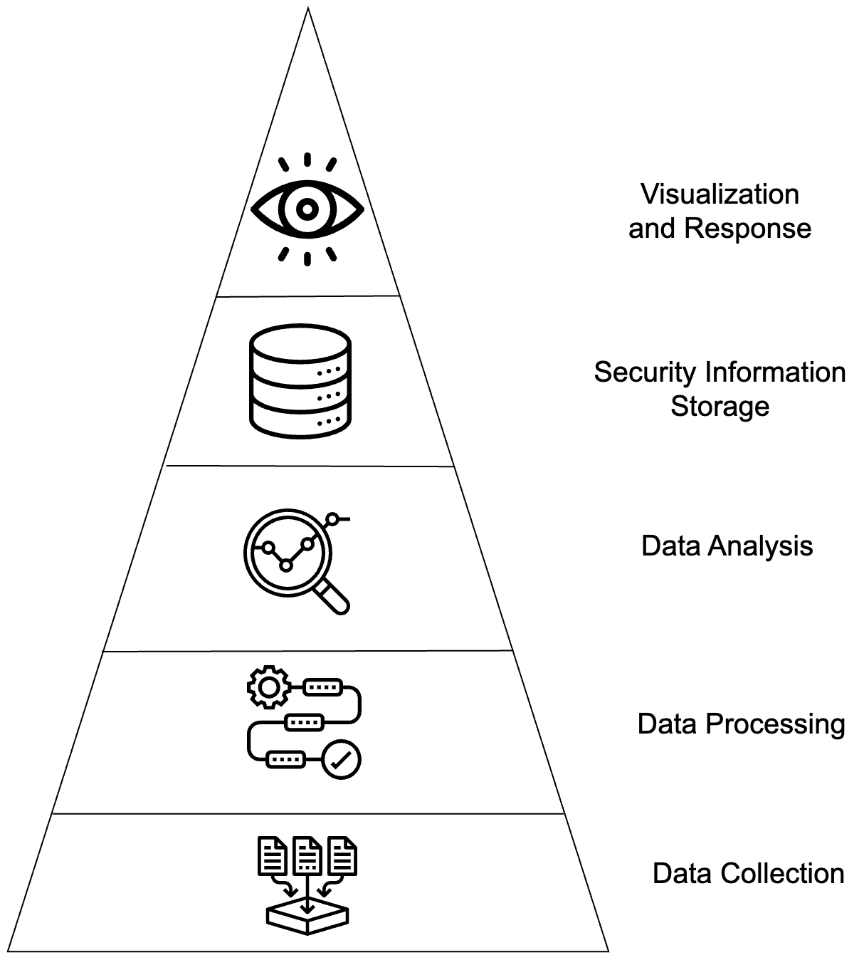
* Security Information Management (SIM): lưu trữ log dài hạn, phân tích và tạo báo cáo tuân thủ.
* Security Event Management (SEM): giám sát thời gian thực, tương quan sự kiện và sinh cảnh báo.

Gartner mô tả SIEM là công nghệ cốt lõi (core technology) trong kiến trúc SOC, cung cấp khả năng thu thập log đa nguồn, phân tích sự kiện, correlation và phát hiện tấn công trong thời gian thực.

1. **Vai trò của SIEM trong SOC**

Trong mô hình SOC hiện đại, SIEM đóng vai trò “nền tảng lõi” bởi vì nó cho phép:

* Tập trung hóa giám sát: thu thập toàn bộ log từ máy chủ, endpoint, ứng dụng và thiết bị mạng vào một nơi duy nhất.
* Phát hiện tấn công theo thời gian thực thông qua rule, correlation và phân tích bất thường.
* Kết hợp Threat Intelligence để đối chiếu log với các IOC và kỹ thuật tấn công (TTPs) như MITRE ATT&CK.
* Hỗ trợ điều tra số và truy vấn sâu trong quá trình Incident Response.
* Đáp ứng yêu cầu tuân thủ và lưu trữ log dài hạn theo ISO 27001 và các tiêu chuẩn quốc tế.



Hình 4: Vai trò cảu SIEM

### Các thành phần cơ bản của SIEM

Theo NIST SP 800-92, Gartner và SANS SOC Survey, một hệ thống SIEM hiện đại bao gồm các thành phần chính sau.

**a) Log Collection – Thu thập dữ liệu**

SIEM tiếp nhận log từ các nguồn trong hệ thống, bao gồm: Windows Event Log trên máy Victim, Log hệ thống từ Wazuh Agent, Log FIM, SCA, IOC, HIDS được sinh tự động

Trong mô hình Wazuh, việc thu thập log được thực hiện bởi **Wazuh Agent** và gửi về **Wazuh Manager** qua kênh bảo mật.

**b) Normalization & Parsing – Chuẩn hóa và phân tích cú pháp**

Log thu thập được chuyển về dạng chuẩn (JSON) và được phân tích bằng hệ thống:

* **Decoders**: trích xuất các trường quan trọng như eventID, user, IP, file path.
* **Rules**: so khớp với mẫu (pattern) đã định nghĩa để nhận diện sự kiện.

Quá trình này giúp SIEM hiểu chính xác ngữ nghĩa của từng bản ghi log.

**c) Storage & Indexing – Lưu trữ và lập chỉ mục**

Các log và cảnh báo được gửi đến **Wazuh Indexer**, nơi chúng được:

* Lưu trữ dài hạn
* Lập chỉ mục để truy vấn nhanh
* Phục vụ tìm kiếm trong quá trình kiểm thử và điều tra

Indexer đảm bảo khả năng tìm kiếm real-time và truy vấn các sự kiện trong toàn hệ thống.

**d) Correlation – Tương quan sự kiện cơ bản**

Wazuh áp dụng các rule sự kiện để phát hiện hành vi bất thường, bao gồm:

* Phát hiện đăng nhập sai liên tiếp (Event ID 4625)
* Phát hiện thay đổi file FIM
* Phát hiện hash thuộc IOC
* Phát hiện cấu hình yếu từ SCA

Trong phạm vi lab, tương quan chủ yếu dựa trên **rule-based detection**, không sử dụng correlation nâng cao đa nguồn.

**e) Threat Intelligence (IOC Matching) – So khớp chỉ báo tấn công**

Hệ thống sử dụng danh sách IOC nội bộ được cấu hình tại Manager:Hash mã độc, Domain/IP độc hại

Wazuh so khớp IOC với hash file hoặc log sự kiện từ Agent để sinh cảnh báo Malware Detection/Rootcheck.

Không sử dụng TI feed nâng cao từ nguồn bên ngoài trong phạm vi lab.

**f) Alerting – Sinh cảnh báo**

Khi rule được kích hoạt, Wazuh sinh cảnh báo (alert) với các thông tin: Mức độ (severity), Loại sự kiện (HIDS, FIM, SCA, IOC), File, user, IP hoặc hành vi liên quan

Alert được hiển thị ngay trên Dashboard và lưu tại Indexer.

**g) Dashboard – Trực quan hóa và truy vấn**

Wazuh Dashboard cung cấp giao diện phục vụ vận hành và kiểm thử: Security Events, FIM EventsSCA Results, MITRE ATT&CK Mapping, Discover (truy vấn log theo trường dữ liệu)

h) **File Integrity Monitoring (FIM) – Giám sát toàn vẹn tệp tin**

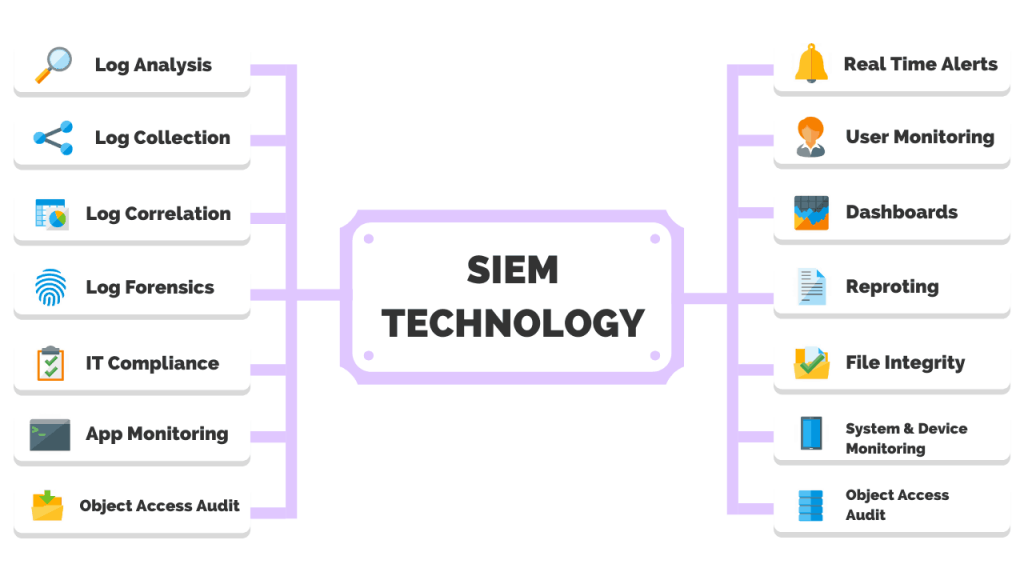
Trong môi trường lab, FIM được cấu hình để giám sát thư mục trên Windows (FIM-Test). Wazuh Agent thực hiện:

* Theo dõi việc **tạo, sửa, đổi tên, xóa** file.
* Tính toán hash (SHA-1, SHA-256) để phát hiện thay đổi nội dung.
* Gửi sự kiện về Manager khi xảy ra bất kỳ biến động nào.

Wazuh Manager áp dụng các rule để phân loại mức độ thay đổi (added, modified, deleted) và sinh cảnh báo.

Đây là chức năng **được bạn triển khai đầy đủ trong thực nghiệm**.

Dashboard là công cụ trung tâm trong quá trình giám sát và phân tích sự kiện trong báo cáo.



Hình 5: Các thành phần của SIEM

### Khái niệm correlation và phân cấp mức độ cảnh báo

* 1. **Khái niệm correlation**

Correlation (tương quan sự kiện) là quá trình kết hợp nhiều sự kiện rải rác để phát hiện hành vi tấn công mà một log đơn lẻ không thể biểu hiện. Theo nghiên cứu SOC Systematic Study [7], correlation là yếu tố biến log thô trở thành thông tin có giá trị trong phát hiện tấn công.

Các dạng correlation phổ biến:

* **Temporal correlation**: nhiều sự kiện trong một khoảng thời gian ngắn.
* **Cross-source correlation:** kết hợp log từ nhiều nguồn (Windows + Firewall).
* **Rule-based correlation:** dựa trên tập luật xác định trước.
* **Threat Intelligence correlation:** đối chiếu IOC từ các nguồn bên ngoài [17].
  1. **Phân cấp mức độ cảnh báo (Severity levels)**

Theo SANS và Wazuh Documentation mức độ cảnh báo thường được phân loại:

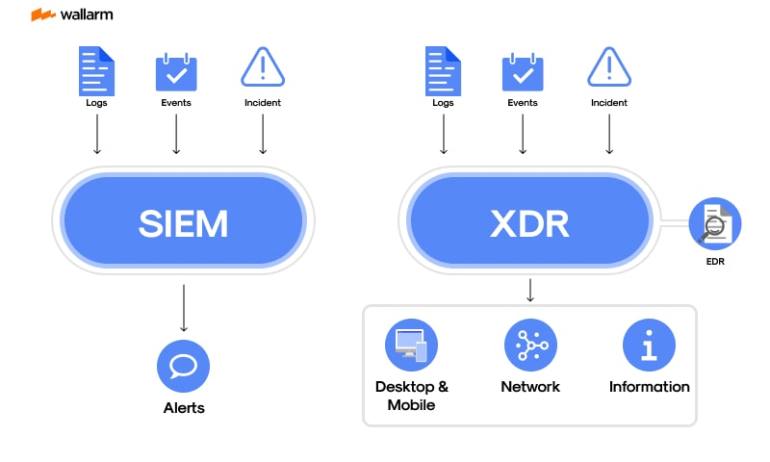
* Low – thông tin hoặc sự kiện ít rủi ro
* Medium – hành vi đáng ngờ, cần theo dõi
* High – có dấu hiệu tấn công rõ ràng
* Critical – tấn công nghiêm trọng, đang diễn ra hoặc đã gây thiệt hại

Trong Wazuh, mức cảnh báo còn được ánh xạ từ rule level 1–15 sang severity.

## Nền tảng Wazuh Stack

### Giới thiệu chung về Wazuh

Wazuh là nền tảng **SIEM + XDR mã nguồn mở**, cung cấp khả năng giám sát bảo mật tập trung, phân tích log, phát hiện xâm nhập và hỗ trợ phản ứng sự cố.



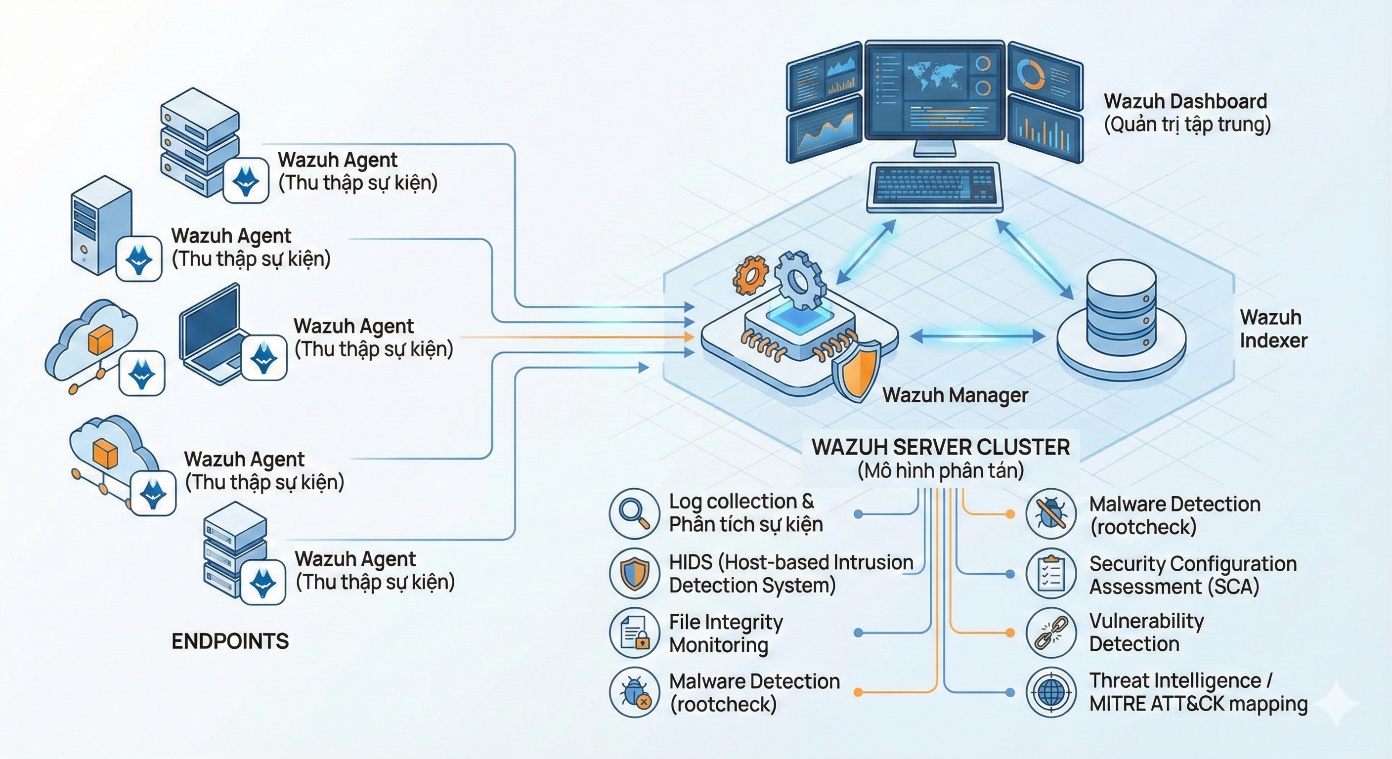
Hình 7: Nền tảng SIEM + XDR mã nguồn mở

Hệ thống được triển khai theo mô hình phân tán gồm ba thành phần chính: **Wazuh Manager**, **Wazuh Indexer** và **Wazuh Dashboard**, kết hợp với **Wazuh Agent** ở phía endpoint để thu thập sự kiện bảo mật [10].

Wazuh tích hợp các chức năng trọng yếu của SOC như:

* Log collection và phân tích sự kiện
* HIDS (Host-based Intrusion Detection System)
* File Integrity Monitoring
* Malware Detection (rootcheck)
* Security Configuration Assessment (SCA)
* Vulnerability Detection
* Threat Intelligence / MITRE ATT&CK mapping

Tất cả được quản trị tập trung thông qua Dashboard.



Hình 8: Thành phần Wazuh Stack

### Kiến trúc tổng thể Wazuh Stack

Kiến trúc Wazuh Stack dựa trên mô hình ba lớp [10]:

* **Wazuh Agent (Endpoint layer)**  
  Thu thập log, sự kiện hệ thống, thay đổi file, dữ liệu registry, tiến trình, và gửi về Manager.
* **Wazuh Manager (Processing layer)**  
  Thực hiện decoding, parsing, rule matching, correlation, tạo alert và điều phối các module như FIM, SCA, Malware, Vulnerability.
* **Wazuh Indexer & Dashboard (Storage & Visualization layer)**  
  Indexer (dựa trên Elasticsearch OpenSearch) lưu trữ và cho phép truy vấn log/alert.  
  Dashboard trực quan hóa dữ liệu, hiển thị cảnh báo, MITRE mapping, agent status và hỗ trợ điều tra sự cố.

A diagram of a computer process

AI-generated content may be incorrect.

Hình 9: kiến trúc Wazuh Stack

### Thành phần Wazuh Manager – Decoders, Rules, Correlation

**a) Decoders (Phân tích cú pháp dữ liệu)**

* Nhận log từ Agents hoặc Syslog.
* Trích xuất các trường (fields) quan trọng bằng cơ chế pattern matching.
* Chuyển log về dạng chuẩn (JSON).  
  Decoders giúp SIEM hiểu đúng cấu trúc log từ Windows, Linux, firewall, webserver,…

**b) Rules (Bộ luật phát hiện)**

Hệ thống rule đa tầng của Wazuh gồm mức 0–15, dùng để:

* So khớp log với các mẫu sự kiện nguy cơ
* Xác định mức độ severity
* Kích hoạt alert khi có dấu hiệu tấn công

Rules có thể phát hiện: brute-force, privilege escalation, file modification, malware signature, bất thường hệ thống,…

**c) Correlation (Tương quan sự kiện)**

Wazuh Manager kết hợp nhiều sự kiện để tạo alert chính xác hơn, ví dụ:

* Nhiều lần đăng nhập sai → brute-force
* Thay đổi file + tiến trình lạ → nghi ngờ malware
* Registry modification + driver load → khả năng rootkit  
  Cơ chế correlation của Wazuh được triển khai qua rule groups liên kết theo chuỗi điều kiện [10].

### Thành phần Wazuh Indexer – lưu trữ và index log/alert

Wazuh Indexer là hệ thống lưu trữ phân tán dựa trên OpenSearch/Elasticsearch, có chức năng [10]:

* Lưu trữ log và alert do Wazuh Manager gửi đến
* Lập chỉ mục (indexing) để hỗ trợ truy vấn tốc độ cao
* Cung cấp API để Dashboard truy vấn dữ liệu
* Đảm bảo tính toàn vẹn và độ sẵn sàng của dữ liệu giám sát
* Hỗ trợ tìm kiếm nâng cao (full-text search, filter, aggregation)

Indexing là thành phần bắt buộc của SIEM nhằm phục vụ phân tích sự cố và tuân thủ lưu trữ log dài hạn.

### Thành phần Wazuh Dashboard – giao diện giám sát và phân tích

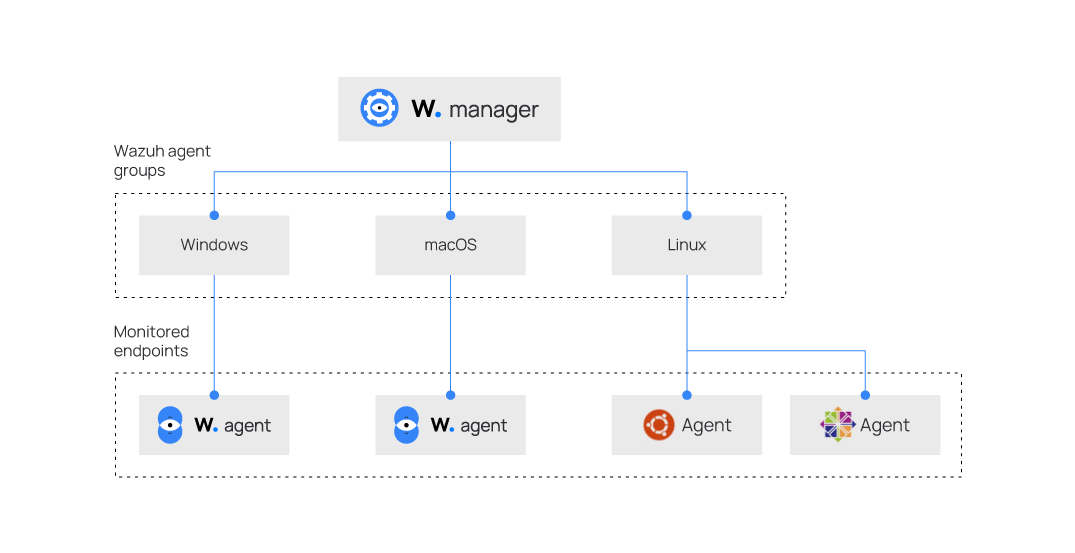
Wazuh Dashboard cung cấp giao diện trực quan hóa toàn bộ hoạt động của SOC, bao gồm [10]:

* Hiển thị alert theo thời gian thực
* Bảng điều khiển MITRE ATT&CK (mapping rule → tactic/technique)
* Quản lý agent (online/offline, cấu hình, trạng thái)
* Truy vấn log/alert theo thời gian, severity hoặc nguồn dữ liệu
* Bộ lọc, biểu đồ, đồ thị timeline phục vụ điều tra sự cố
* Giao diện quản trị hệ thống Wazuh

Dashboard giúp SOC Analyst thực hiện monitoring, triage và incident investigation.

### Wazuh Agent – cơ chế thu thập log và gửi sự kiện

Wazuh Agent là thành phần được cài đặt trên endpoint (Windows, Linux, macOS), thực hiện các nhiệm vụ [10]:



Hình 10: Quá trình gom nhóm các Agent theo groups [10]

**a) Thu thập nhật ký (Log Collection)**

* Windows Event Log
* Sysmon, PowerShell
* Syslog
* Ứng dụng (IIS, Apache, Nginx)

**b) File Integrity Monitoring (FIM)**

* Giám sát thay đổi file: modified, created, deleted
* Hashing (MD5/SHA1/SHA256) để đánh giá integrity

**c) HIDS Events**

* Process creation
* Registry modification (Windows)
* User login/logout
* System calls (Linux)

**d) Gửi dữ liệu về Manager**

Dữ liệu được gửi qua kênh bảo mật (TLS) để Manager xử lý, tạo alert và chuyển vào Indexer.

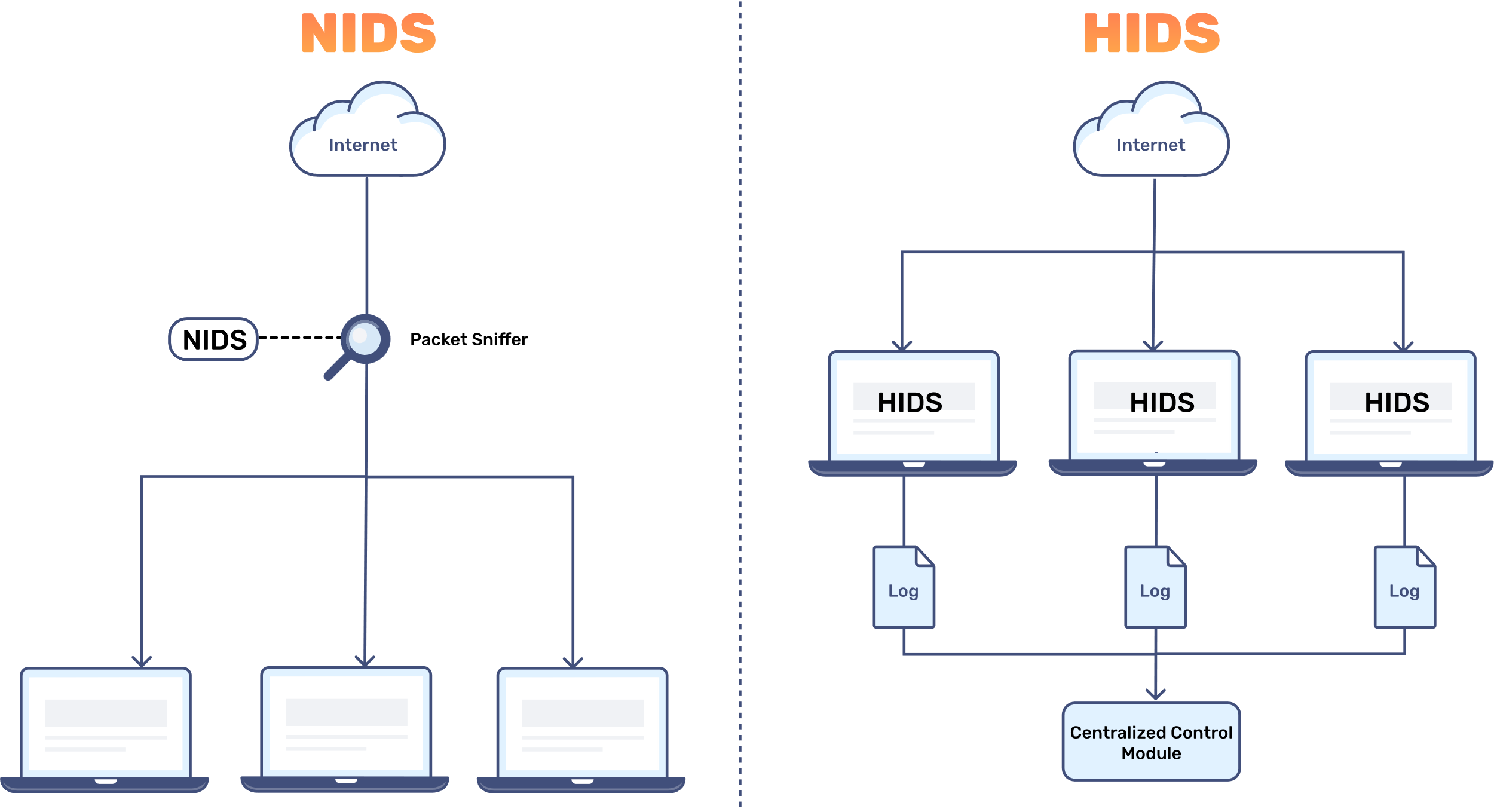
Agent là thành phần trực tiếp đưa dữ liệu từ endpoint vào SIEM, đóng vai trò thiết yếu trong mọi chức năng của Wazuh.

## Các chức năng an ninh chính trong Wazuh theo cấu trúc I-A-I-D

Wazuh cung cấp một tập hợp chức năng an ninh chủ chốt để giám sát, phát hiện và đánh giá rủi ro trên hệ thống, được xây dựng trên mô hình agent–manager–indexer/dashboard và hoạt động theo cách tiếp cận phòng thủ nhiều lớp [10][11][12][14]. Các chức năng này gồm: HIDS (giám sát hoạt động trên host), FIM (giám sát tính toàn vẹn tệp tin), phát hiện mã độc/Rootcheck dựa trên IOC, đánh giá cấu hình bảo mật (SCA), và ánh xạ MITRE ATT&CK phục vụ truy tìm mối đe dọa

### HIDS – Host Intrusion Detection System

1. **Giới thiệu (Introduction)**  
   Hệ thống Phát hiện Xâm nhập Máy chủ (Host Intrusion Detection System – HIDS) trong Wazuh là thành phần giám sát bảo mật tại điểm cuối (endpoint), thu thập và phân tích nhật ký từ hệ điều hành, ứng dụng, dịch vụ, thiết bị mạng và dịch vụ cloud để phát hiện dấu hiệu tấn công hoặc hoạt động bất thường [10]. HIDS cung cấp mức nhìn sâu vào bên trong máy chủ, bổ sung cho các giải pháp dựa trên mạng (NIDS) vốn chủ yếu quan sát lưu lượng mạng.



Hình 12: So sánh HIDS & NIDS

*Nguồn: bunny,What is a Network Intrusion Detection system (NIDS),2025*

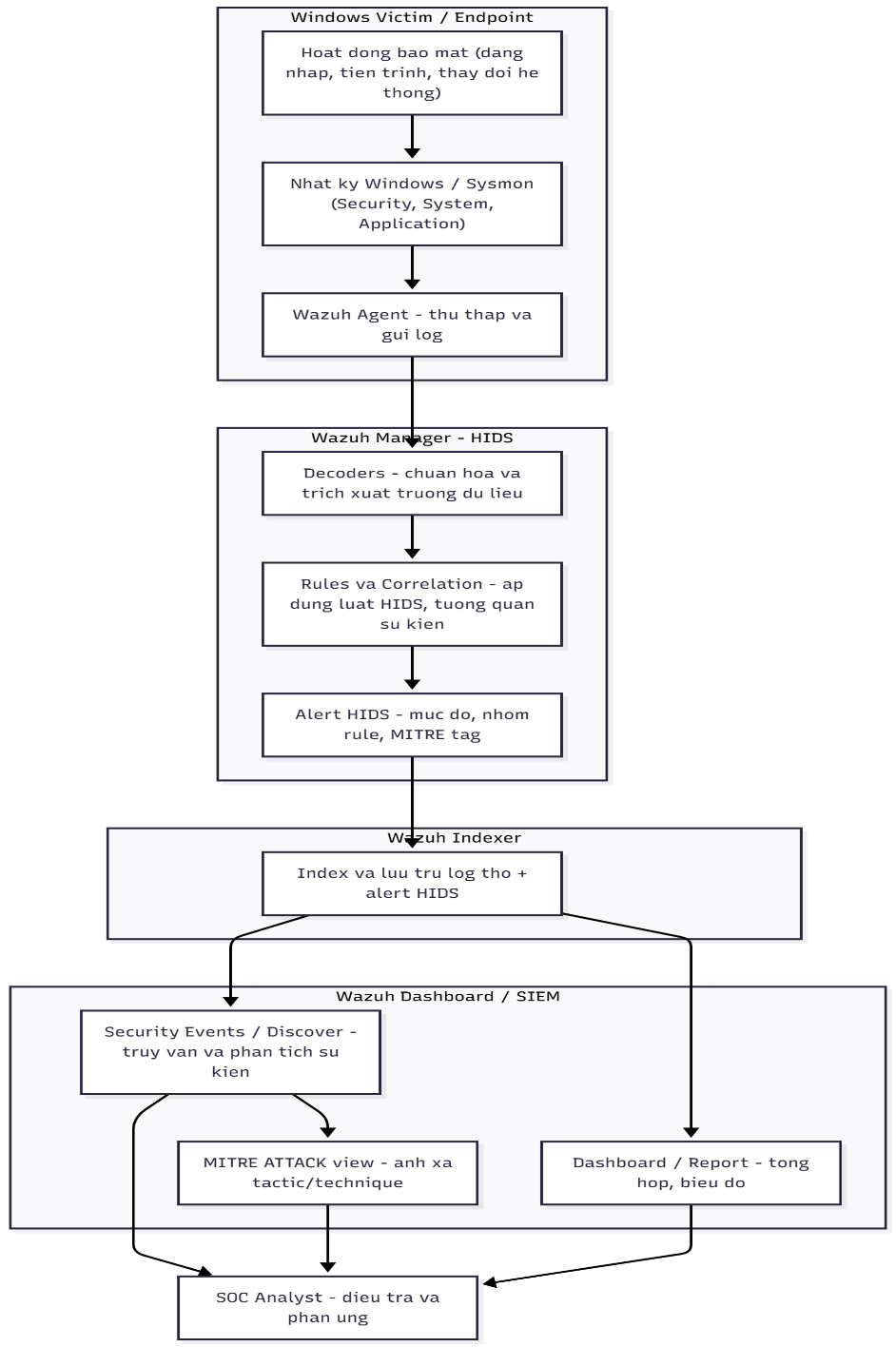
1. **Phân tích (Analysis)**  
   Theo tài liệu Wazuh, HIDS dựa trên một bộ cơ chế chính [10]:

* Phân tích và tương quan nhật ký (log analysis & correlation): log được giải mã, chuẩn hóa và so khớp với rules để phát hiện các mẫu tấn công đã biết (failed logins bất thường, privilege escalation, thực thi lệnh trái phép, truy cập trái phép,…).
* Phát hiện dựa trên chữ ký (signature-based): sử dụng bộ luật được định nghĩa sẵn và có thể mở rộng để phát hiện các TTP đã biết.
* Phát hiện bất thường (anomaly-based): một số luật được thiết kế để phát hiện hành vi vượt ngưỡng bình thường (số lần đăng nhập thất bại, số lượng process lạ, pattern truy cập dịch vụ,…).
* Tương quan đa sự kiện (multi-event correlation): kết hợp nhiều sự kiện rời rạc theo thời gian để nhận diện chuỗi tấn công phức tạp (ví dụ brute-force → login thành công → privilege escalation).

1. **Triển khai (Implementation)**  
   Trong kiến trúc Wazuh, HIDS được triển khai theo mô hình client–server [10]:

* Wazuh Agent: cài trên các máy chủ/endpoint; thu thập log (Windows Event Log, Sysmon, syslog, systemd journal, auditd, log ứng dụng) và gửi về Manager.
* Wazuh Manager: thực hiện decoding, normalization, áp dụng ruleset và sinh alert khi phát hiện sự kiện bất thường [10].
* Wazuh Indexer & Dashboard: lưu trữ, lập chỉ mục log/alert và cung cấp giao diện truy vấn, dashboard, báo cáo, hỗ trợ điều tra sự cố [10].

**Quy trình hoạt động tổng quát của HIDS trong Wazuh** [10]:



Hình 13: Quy trình hoạt động của HIDS trong Wazuh

* **Sự kiện bảo mật xảy ra trên endpoint**  
  Ví dụ: đăng nhập sai nhiều lần, tạo/chạy process lạ, thay đổi quyền trên file hệ thống. Hệ thống ghi nhận vào các log như Windows Security, Sysmon hoặc syslog.
* **Wazuh Agent thu thập và gửi dữ liệu**  
  Agent đọc các log này qua API/kênh sự kiện, chuẩn hóa thành sự kiện dạng JSON và gửi về Wazuh Manager qua kênh giao tiếp (thường cổng 1514/UDP hoặc TCP đã cấu hình).
* **Wazuh Manager phân tích và áp rules**
  + Module **decoders** trích xuất trường quan trọng (user, process, IP nguồn, event ID, logon type,…).
  + Module **rules** so khớp sự kiện với bộ luật HIDS (nhóm authentication\_failed, windows, sysmon,…) để xác định loại hành vi và mức độ nghiêm trọng.
  + Nếu phù hợp điều kiện tương quan (ví dụ nhiều sự kiện 4625 trong thời gian ngắn), Manager có thể nâng mức cảnh báo hoặc gắn thêm ngữ cảnh MITRE ATT&CK.
* **Ghi log và hiển thị cảnh báo**  
  Sự kiện và alert được gửi tới **Wazuh Indexer**, nơi chúng được lưu trữ và lập chỉ mục; **Wazuh Dashboard** truy vấn từ Indexer để hiển thị trong các mục như *Security Events*, *Discover*, *MITRE ATT&CK*.
* **Nhà phân tích SOC điều tra**  
  Dựa trên alert HIDS (mức độ, rule.id, nhóm, MITRE tactic/technique) và log chi tiết, nhà phân tích truy vấn, lọc, pivot sang các trường khác (user, IP, hostname) để xác định nguyên nhân và phạm vi ảnh hưởng.

1. **Thảo luận (Discussion)**  
   HIDS đóng vai trò là "con mắt" quan trọng bên trong hệ thống, cung cấp tầm nhìn (visibility) mà các công cụ bảo mật biên mạng không có được. HIDS cung cấp mức nhìn sâu bên trong host nhưng phụ thuộc mạnh vào việc tinh chỉnh rules, khả năng mở rộng hạ tầng và biện pháp bảo vệ chính Agent/Manager khỏi bị vô hiệu hóa [10].

### FIM – File Integrity Monitoring

1. **Giới thiệu (Introduction)**  
   Giám sát Tính toàn vẹn Tệp tin (File Integrity Monitoring – FIM) trong Wazuh theo dõi các tệp tin, thư mục và khóa Registry quan trọng nhằm phát hiện mọi thay đổi trái phép về nội dung, thuộc tính hoặc trạng thái tồn tại [11]. FIM là một trong các yêu cầu trọng yếu trong nhiều chuẩn tuân thủ như PCI-DSS, HIPAA, GDPR,… [11].

A circular diagram with text

AI-generated content may be incorrect.

Hình 14: FIM trong các khung bảo mật và tiêu chuẩn quy định

1. **Phân tích (Analysis)**  
   Theo tài liệu FIM của Wazuh [11]:

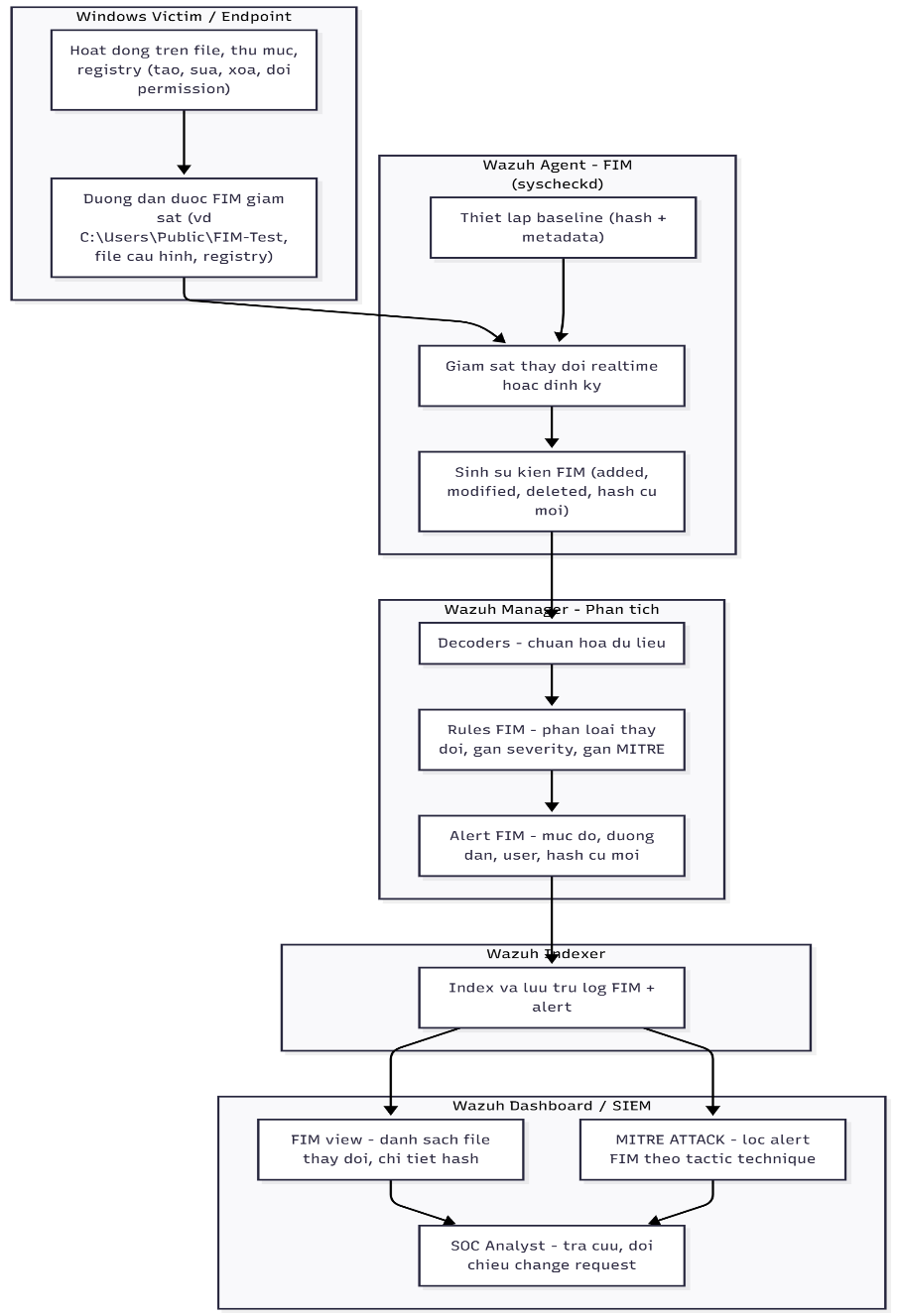
* FIM thiết lập **baseline** cho các đối tượng được giám sát (hash, quyền, owner, group, kích thước, thời gian chỉnh sửa,…).
* Mọi thay đổi sau đó được phát hiện thông qua:
  + Thay đổi nội dung: so sánh hash (MD5, SHA-1, SHA-256) để phát hiện sửa đổi nội dung tệp.
  + Thay đổi thuộc tính: giám sát permission, owner, group và các thuộc tính file.
  + Thay đổi tồn tại: phát hiện tạo mới, đổi tên, xóa file/thư mục.
  + Giám sát Registry trên Windows: theo dõi các khóa registry quan trọng phục vụ phát hiện persistence hoặc thay đổi cấu hình hệ thống [11].  
    FIM đặc biệt hữu ích trong việc nhận diện ransomware (mã hóa hàng loạt tệp), backdoor chỉnh sửa tệp hệ thống và các thay đổi cấu hình trái phép [11].

1. **Triển khai (Implementation)**  
   Cơ chế triển khai FIM trong Wazuh [11]:

* Cấu hình trên Agent (khối syscheck): khai báo đường dẫn, file, registry cần giám sát và chọn chế độ real-time hoặc quét định kỳ.
* Thiết lập baseline: lần quét đầu xác lập trạng thái chuẩn của đối tượng (hash + metadata).
* Giám sát thay đổi: mỗi thay đổi được Agent ghi nhận và gửi sự kiện về Manager.
* Đánh giá tại Manager: Manager áp dụng rules FIM để phân loại mức độ nghiêm trọng (thay đổi file hệ thống thường có severity cao hơn file tạm) và sinh cảnh báo tương ứng.
* Lưu trữ: sự kiện FIM được lưu tại Indexer, cung cấp audit trail đầy đủ cho điều tra và chứng cứ số [11].

Quy trình hoạt động tổng quát của FIM trong Wazuh :

* **Windows Victim / Endpoint**
  + Người dùng hoặc tiến trình thực hiện thao tác trên file/thư mục/registry (tạo mới, sửa, xóa, đổi permission,…).
* **Syscheck/FIM trên Wazuh Agent**
  + Agent (module syscheckd) giám sát các đường dẫn đã cấu hình.
  + Khi phát hiện thay đổi, Agent tính lại hash, so sánh với baseline và ghi sự kiện FIM.
* **Gửi sự kiện về Wazuh Manager**
  + Sự kiện FIM (JSON) được đẩy từ Agent lên Manager qua kênh bảo mật.
* **Wazuh Manager – Phân tích FIM**
  + Decoders chuẩn hóa trường dữ liệu (đường dẫn, loại thay đổi, hash cũ/mới, user,…).
  + Rules FIM phân loại loại hành động (added/modified/deleted, permission change, registry change) và gán **mức độ (severity)**, **nhóm rule**, **MITRE tag** nếu có (ví dụ Impact / Data Destruction).
* **Wazuh Indexer**
  + Lưu và đánh index cả log thô FIM lẫn alert tương ứng, tạo “audit trail” cho điều tra sau này.
* **Wazuh Dashboard / SIEM**
  + Tab **File Integrity Monitoring** cho phép xem chi tiết: file nào, thay đổi gì, hash trước/sau, user thực hiện, thời gian.
  + Tab **MITRE ATT&CK** và **Threat Hunting** dùng các alert FIM này để hiển thị theo tactic/technique (ví dụ T1485 – Data Destruction).
  + Điều tra viên SOC tra cứu, đối chiếu với change ticket để phân biệt thay đổi hợp lệ và thay đổi nghi ngờ.



Hình 15: Quy trình hoạt động cảu FIM

1. **Thảo luận (Discussion)**  
   FIM đem lại khả năng truy vết chi tiết về “ai – cái gì – khi nào – ở đâu” đối với các thay đổi trên hệ thống. Tuy nhiên, giám sát quá rộng (too many paths) có thể gây áp lực lên tài nguyên và sinh nhiều sự kiện không cần thiết. Do đó, cần chọn lọc phạm vi giám sát (thư mục hệ thống, file cấu hình, thư mục dữ liệu nhạy cảm) và phối hợp với quy trình quản lý thay đổi (change management) để phân biệt thay đổi hợp pháp và trái phép [11].

**A magnifying glass with icons

AI-generated content may be incorrect.**

Hình 16. Malware Detection [12]

### Malware Detection dựa trên IOC

1. **Giới thiệu (Introduction)**  
   Chức năng Malware Detection trong Wazuh sử dụng các chỉ báo tấn công (Indicators of Compromise – IOC) như hash tệp tin, địa chỉ IP, tên miền hoặc mẫu đường dẫn để phát hiện sự hiện diện của mã độc, C2 server hoặc các thành phần đáng ngờ trên hệ thống [12].  
   Thay vì phân tích hành vi phức tạp, cơ chế này dựa trên việc so khớp nhanh (matching) giữa dữ liệu thu thập từ endpoint và các IOC đã được định nghĩa trước, giúp phát hiện sớm các mối đe dọa đã biết, bổ trợ cho HIDS, FIM và các giải pháp AV/EDR khác [12].
2. **Phân tích (Analysis)**

**IOC Matching – so khớp IOC (hash/IP/domain)**  
Theo tài liệu Wazuh, IOC được quản lý thông qua các danh sách (text/CDB) trong etc/lists/, được sử dụng bởi rules để so khớp [12]:

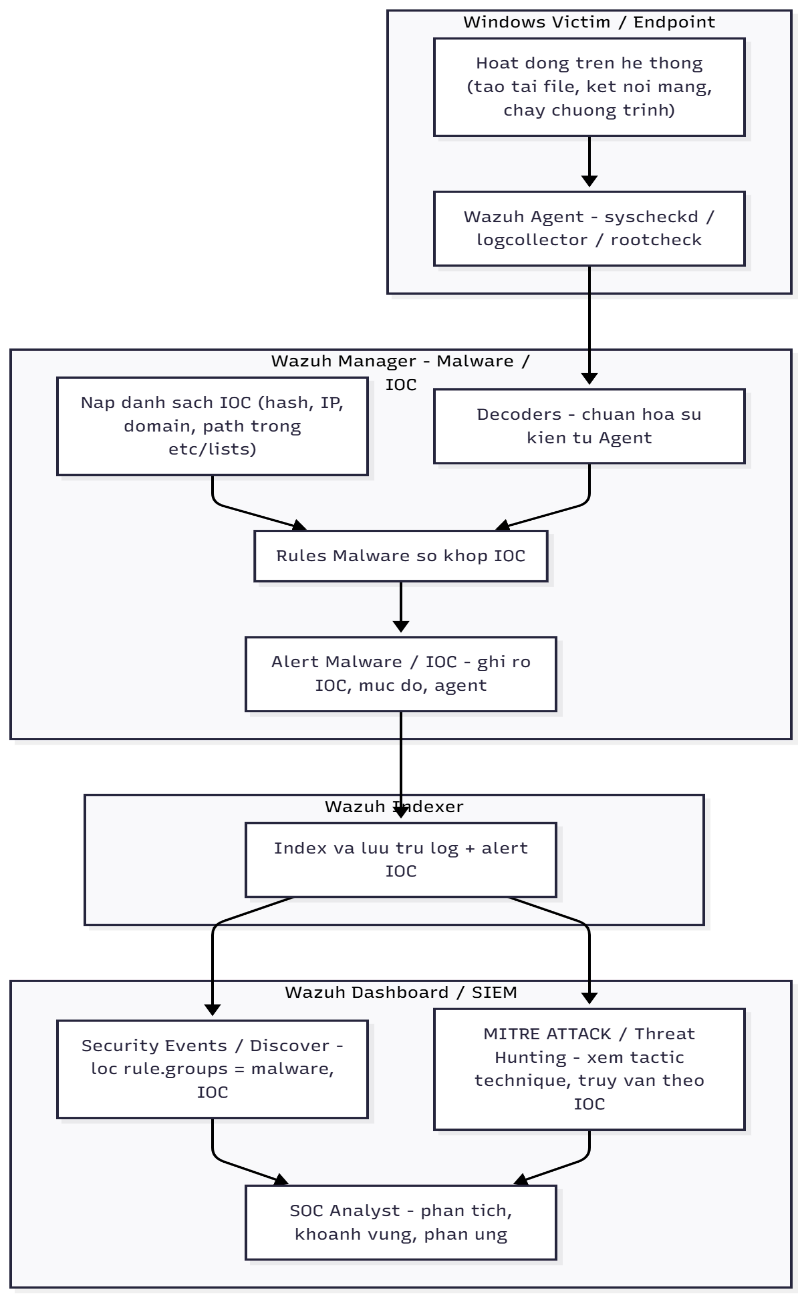
* + Hash IOC: so khớp hash tệp tin với danh sách mã độc đã biết.
  + Network IOC: so khớp IP, domain, URL với IOC C2/phishing/malware hosting.
  + IOC theo đường dẫn: phát hiện sự xuất hiện của file ở những đường dẫn đã được đánh dấu.  
    Khi có match, alert Wazuh sẽ ghi rõ IOC tương ứng, mức độ nghiêm trọng và host bị ảnh hưởng [12].

1. **Triển khai (Implementation)**

* IOC được quản lý tập trung tại Wazuh Manager dưới dạng file list/CDB trong etc/lists/, cập nhật thủ công hoặc từ nguồn Threat Intelligence bên ngoài [12].
* Kết quả quét/so khớp được gửi về Manager, áp dụng ruleset để sinh cảnh báo; các alert được lưu tại Indexer và hiển thị qua Dashboard, phục vụ phân tích và điều tra [12].

Quy trình qua trình hoạt động trong Wazuh:

* **Windows Victim / Endpoint**
  + Người dùng hoặc tiến trình tạo / tải về / thực thi file, kết nối tới IP/domain, ghi log hệ thống, v.v.
  + Wazuh Agent (modules: syscheckd, logcollector, rootcheck) thu thập:
  + hash file, đường dẫn file;
  + log mạng, DNS, firewall;
  + các dấu hiệu rootkit, file/registry bất thường.
* **So khớp IOC tại Wazuh Manager**
  + Manager nạp các danh sách IOC từ thư mục etc/lists/ (hash, IP, domain, đường dẫn) [12].
  + Khi nhận sự kiện từ Agent, Manager dùng decoder + rules để so khớp với IOC tương ứng:
    - Nếu hash file nằm trong list hash độc hại → sinh alert Malware.
    - Nếu IP/domain khớp IOC C2/phishing → sinh alert Network IOC.
    - Nếu đường dẫn xuất hiện file “đã đánh dấu” → sinh alert Suspicious path.
* **Sinh cảnh báo và lưu trữ**
  + Rule ghi rõ IOC trùng khớp (ví dụ: tên mẫu malware, IP C2, domain), mức độ nghiêm trọng (thường high/critical) và agent bị ảnh hưởng.
  + Alert được gửi sang Wazuh Indexer để lưu trữ và đánh chỉ mục.
* **Hiển thị trên Wazuh Dashboard / Threat Hunting**
  + Tab **Security Events / Discover**: có thể lọc theo rule.groups: malware, data.win.integrity.hash, lists.malware-hashes,…
  + Tab **MITRE ATT&CK**: các alert IOC được ánh xạ vào tactic/technique như *Command and Control*, *Execution* hoặc *Impact* (tùy rule).
  + Tab **Threat Hunting**: điều tra viên lọc theo IOC, agent hoặc khoảng thời gian để truy vết phạm vi ảnh hưởng.



Hình 16: Quy trình hoạt độngMalware Detection trên Wazuh

1. **Thảo luận (Discussion)**  
   Phát hiện mã độc dựa trên IOC là lớp bảo vệ nhẹ, nhanh và hiệu quả với các mẫu mã độc đã biết, rất phù hợp cho môi trường nghiên cứu/demo hoặc nơi không thể triển khai đầy đủ AV/EDR thương mại [12]. Hạn chế chính:

* Tính phản ứng (reactive): chỉ phát hiện khi IOC/chữ ký đã được cập nhật; không đủ để phát hiện zero-day hoặc mẫu chưa được ghi nhận.
* Phụ thuộc chất lượng IOC: IOC lỗi thời hoặc không phù hợp có thể gây nhiều false positives hoặc bỏ sót.  
  Vì vậy, chức năng này nên được sử dụng như một lớp bổ trợ, kết hợp với HIDS, FIM và các giải pháp AV/EDR khác [12].

### Security Configuration Assessment (SCA)

1. **Giới thiệu (Introduction)**  
   Đánh giá Cấu hình Bảo mật (Security Configuration Assessment – SCA) là chức năng chủ động của Wazuh, tự động kiểm tra cấu hình hệ thống so với các tiêu chuẩn bảo mật như CIS Benchmarks, các best practice và chính sách nội bộ [14]. Mục tiêu là phát hiện sớm misconfiguration và hỗ trợ báo cáo tuân thủ.
2. **Phân tích (Analysis)**  
   Theo tài liệu Wazuh, SCA sử dụng các policy kiểm tra (SCA policies) áp dụng cho từng hệ điều hành/ứng dụng [14]:

* Kiểm tra cấu hình hệ điều hành: chính sách mật khẩu, khóa màn hình, auditing, dịch vụ không cần thiết, cấu hình firewall,…
* Kiểm tra cấu hình ứng dụng: web server, database,… nếu có policy tương ứng.
* Mỗi rule trong policy được gán trạng thái Passed/Failed/Not applicable và có thể kèm hướng dẫn khắc phục (remediation).  
  SCA giúp lượng hóa mức độ tuân thủ (compliance score) của từng host và toàn bộ môi trường.

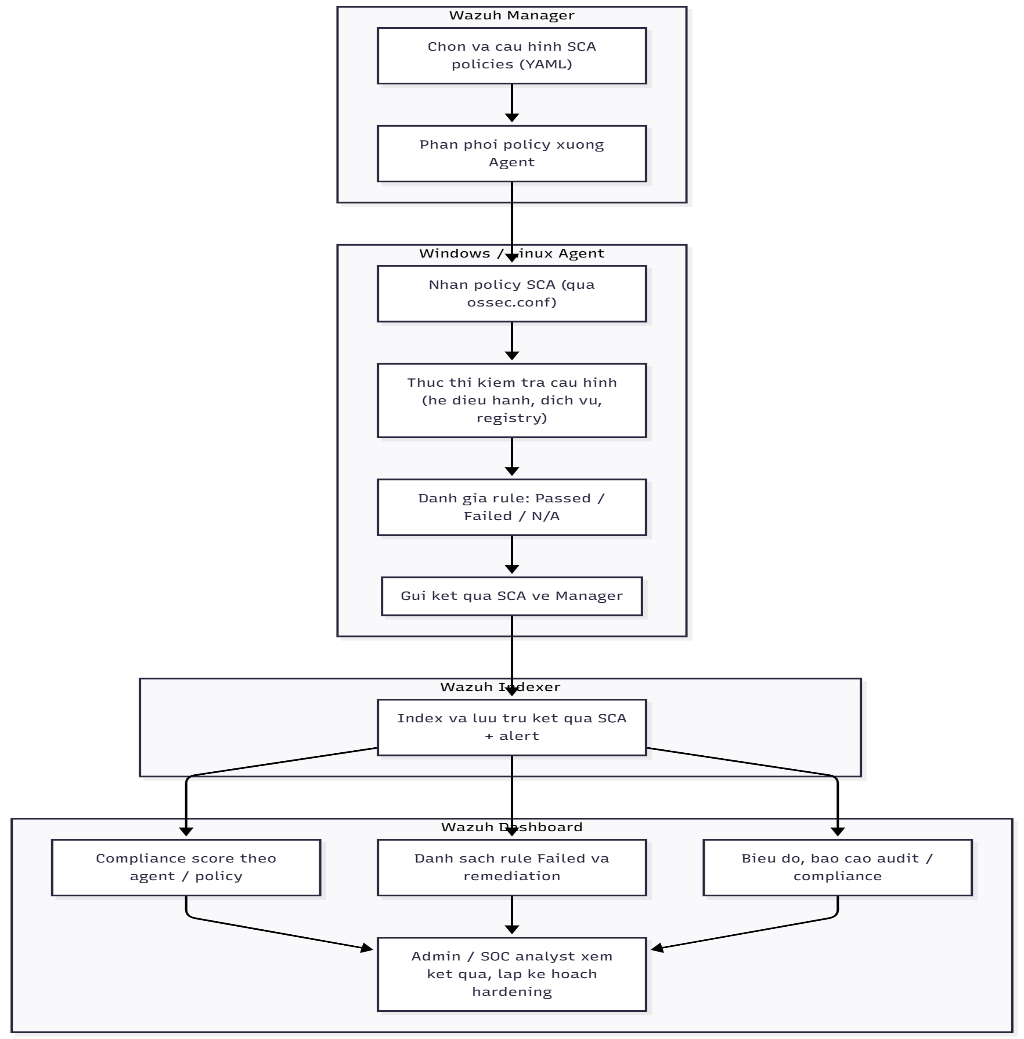
1. **Triển khai (Implementation)**

Quy trình triển khai SCA [14]:

* Lựa chọn policy: chọn các policy phù hợp (ví dụ CIS Linux, Windows) trên Manager và phân phối cho Agent.
* Thực thi trên Agent: Agent chạy policy định kỳ hoặc theo yêu cầu, thu thập thông tin cấu hình và đánh giá theo từng rule.
* Gửi kết quả về Manager: Manager tổng hợp, lưu trữ kết quả tại Indexer.
* Dashboard: hiển thị điểm tuân thủ, danh sách các kiểm tra failed, biểu đồ và báo cáo phục vụ audit/compliance.

Quy trình tổng quát của SCA trong Wazuh:

* **Chuẩn bị policy trên Wazuh Manager**
  + Quản trị viên kích hoạt các file policy SCA (định dạng YAML, ví dụ cis\_win10.yml, cis\_debian.yml) tương ứng với từng loại hệ điều hành/ứng dụng.
  + Mỗi policy gồm nhiều rule: điều kiện kiểm tra, cách đánh giá Passed/Failed, mức độ nghiêm trọng, khuyến nghị khắc phục.
* **Phân phối policy tới Wazuh Agent**
  + Manager phân phối cấu hình SCA xuống agent thông qua kênh quản lý thông thường (ossec.conf).
  + Agent nhận được danh sách policy cần chạy và lịch chạy (interval, scan on start,…).
* **Thực thi kiểm tra trên Agent**
  + Module SCA trên Agent tuần tự thực hiện các rule:
    - Đọc file cấu hình hệ thống, registry, service, quyền user/group…
    - Chạy một số lệnh hệ thống (PowerShell, shell) để thu thông tin.
  + Mỗi rule được gán trạng thái: Passed / Failed / Not applicable và điểm trọng số.
* **Gửi kết quả về Wazuh Manager & Indexer**
  + Agent gửi báo cáo SCA (theo từng policy + host) về Manager.
  + Manager chuẩn hóa dữ liệu, sinh event SCA (đặc biệt cho các rule Failed quan trọng) và chuyển sang Indexer để lưu trữ, đánh chỉ mục.
* **Trực quan trên Wazuh Dashboard**
  + Tab **Security configuration assessment** hiển thị:
    - Điểm tuân thủ tổng thể (compliance score %) cho từng agent.
    - Danh sách các kiểm tra Failed, mức độ severity, gợi ý remediation.
    - Biểu đồ so sánh nhiều host, xu hướng cải thiện sau các lần hardening.
  + Điều tra viên có thể lọc theo: agent, policy, severity, trạng thái rule để ưu tiên xử lý.
* **Vòng lặp cải tiến cấu hình**
  + Dựa trên các mục Failed, quản trị viên điều chỉnh cấu hình hệ thống (hardening).
  + Sau đó chạy lại SCA để kiểm tra lại, tạo thành vòng lặp: Đánh giá → Khắc phục → Đánh giá lại.



Hình 17: Quy trình hoạt động SCA trongv Wazuh

1. **Thảo luận (Discussion)**  
   SCA giúp tự động hóa quá trình hardening và báo cáo tuân thủ, nhưng policy cần được hiệu chỉnh theo bối cảnh để tránh phá vỡ dịch vụ; các lần quét nên được lên lịch hợp lý vì policy lớn có thể tốn tài nguyên, và kết quả “Failed” cần được diễn giải trong ngữ cảnh vai trò máy chủ [14].

### MITRE ATT&CK Mapping và Threat Hunting

1. **Giới thiệu (Introduction)**  
   MITRE ATT&CK® là cơ sở tri thức về chiến thuật, kỹ thuật và quy trình tấn công trong thế giới thực [17]. Wazuh tích hợp khung MITRE ATT&CK bằng cách ánh xạ các rules và alert của mình với các kỹ thuật/kịch bản ATT&CK, qua đó hỗ trợ phân loại, ưu tiên hóa và truy tìm mối đe dọa (threat hunting) có hệ thống [10][17].
2. **Phân tích (Analysis)**  
   Theo tài liệu Wazuh và MITRE [10][17]:

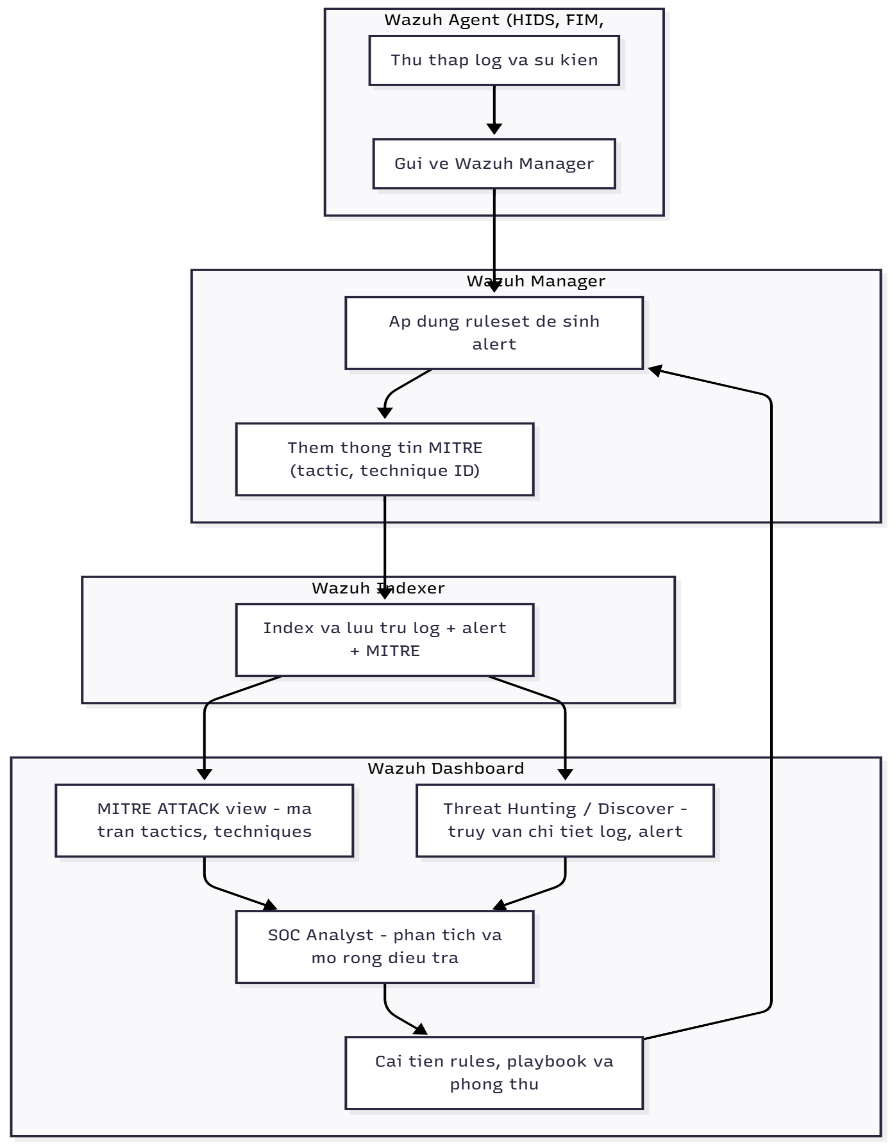
* Mỗi detection rule có thể chứa thông tin MITRE như tactic, technique ID (Txxxx), giúp gán ngữ cảnh ATT&CK cho alert.
* Wazuh Dashboard cung cấp ma trận MITRE ATT&CK hiển thị các kỹ thuật đã xuất hiện trong môi trường, thường dưới dạng heatmap, giúp xác định “điểm nóng” về mặt TTP.
* Trong hoạt động threat hunting, ATT&CK được sử dụng theo hai hướng:
  + Bắt đầu từ một kỹ thuật/kịch bản (hypothesis-driven hunting), sau đó dùng Wazuh để truy vấn log/alert liên quan.
  + Bắt đầu từ một alert/IOC, rồi dùng ATT&CK để suy ra các kỹ thuật liên quan khác trong attack chain để mở rộng điều tra.

1. **Triển khai (Implementation)**

* Ruleset của Wazuh được bổ sung các trường MITRE (tactic, technique ID); người triển khai cần dùng bộ rules cập nhật để tận dụng mapping này [10].
* Module MITRE trên Dashboard tự động tổng hợp thông tin MITRE từ alert và hiển thị trong ma trận ATT&CK; nhà phân tích có thể drill-down vào từng kỹ thuật để xem danh sách alert liên quan [10][17].
* Trong quy trình vận hành SOC, ATT&CK được sử dụng làm ngôn ngữ chung khi mô tả sự cố và khi lập playbook, từ đó gắn kết detection – investigation – response.

Quy trình khai thác MITRE ATT&CK và Threat Hunting trong Wazuh có thể tóm tắt như sau:

* **Sinh alert từ các module HIDS/FIM/IOC/SCA**
  + Wazuh Agent thu thập log và gửi về Manager.
  + Manager áp dụng ruleset, tạo alert (ví dụ: đăng nhập thất bại nhiều lần, sửa registry, xóa file, IOC match,…).
* **Ánh xạ alert sang MITRE ATT&CK**
  + Mỗi alert chứa trường MITRE (tactic, technique ID – Txxxx).
  + Alert được gắn ngữ cảnh: thuộc chiến thuật nào (Credential Access, Defense Evasion, Impact, …) và kỹ thuật cụ thể.
* **Tổng hợp trên MITRE Dashboard**
  + Module **MITRE ATT&CK** trong Dashboard đọc các trường MITRE từ alert.
  + Sinh **ma trận ATT&CK** và các biểu đồ:
    - Top tactics, top techniques.
    - Số lần xuất hiện theo agent/thời gian.
* **Khởi tạo phiên Threat Hunting**
  + Nhà phân tích có thể:
    - **Bắt đầu từ kỹ thuật**: chọn một kỹ thuật (ví dụ Brute Force – T1110), xem toàn bộ alert liên quan và truy ngược lại host/user/phiên tấn công.
    - **Bắt đầu từ alert cụ thể**: từ một cảnh báo HIDS/FIM/IOC, xem MITRE mapping rồi mở rộng truy vấn log (Discover/Threat Hunting) để tìm thêm sự kiện cùng tactic/kỹ thuật.
* **Truy vấn & mở rộng điều tra**
  + Sử dụng các truy vấn trên Wazuh (DQL / KQL) để lọc theo:
    - agent.name, data.win.eventdata.ipAddress, rule.mitre.id, rule.id, …
  + Tìm các bước trước–sau trong chuỗi tấn công (ví dụ: trước brute-force có scan cổng, sau brute-force có chạy lệnh lạ, sửa registry…).
* **Rút ra kết luận & cải thiện detection**
  + Kết quả hunting được dùng để:
    - Xác nhận sự cố, thu thập bằng chứng.
    - Điều chỉnh rules (thêm điều kiện, tăng/giảm mức severity).
    - Cập nhật playbook phản ứng sự cố gắn với tactic/kỹ thuật ATT&CK tương ứng.



Hình 18: Quy trình hoạt độngMITRE ATT&CK & Threat Hunting

1. **Thảo luận (Discussion)**  
   Ánh xạ MITRE ATT&CK giúp SOC có ngôn ngữ chung khi phân tích sự cố, ưu tiên xử lý theo chiến thuật/kỹ thuật trọng yếu, nhưng chất lượng phụ thuộc vào mức độ chính xác và cập nhật của mapping trong ruleset; ma trận ATT&CK là công cụ định hướng, không thay thế được phân tích bối cảnh và tri thức chuyên môn của nhà phân tích [10][17].

## Mô hình phòng thủ theo chiều sâu (Defense in Depth) trong hệ thống SOC Wazuh

### Tổng quan về mô hình phòng thủ theo chiều sâu

1. **Khái niệm và nguồn gốc.**  
   Defense in Depth (DiD), hay "Phòng thủ theo chiều sâu", là chiến lược bảo mật trong đó nhiều lớp kiểm soát an ninh được triển khai đồng thời trên con người, quy trình và công nghệ nhằm tăng tính kiên cường của hệ thống. Khi một lớp phòng thủ thất bại, các lớp tiếp theo vẫn có khả năng phát hiện, làm chậm hoặc ngăn chặn cuộc tấn công [19][20][22].

Chiến lược này bắt nguồn từ tư duy quân sự truyền thống với nhiều tuyến phòng thủ liên tiếp. Trong an ninh mạng, DiD thừa nhận rằng không có giải pháp đơn lẻ nào là hoàn hảo, do đó cần thiết kế hệ thống để tăng chi phí và thời gian tấn công, đồng thời tạo nhiều cơ hội cho đội bảo mật phát hiện và ứng phó.

1. **Nguyên tắc và lợi ích chính**

Các tổ chức như NIST, ENISA và Microsoft xác định một số nguyên tắc cốt lõi của DiD [19][20][22]:

* Đa dạng hóa biện pháp kiểm soát: Kết hợp linh hoạt các biện pháp phòng ngừa, phát hiện và ứng phó, ưu tiên sử dụng nhiều lớp kiểm soát thay vì phụ thuộc vào một công nghệ duy nhất.
* Phân tầng bảo vệ: Triển khai kiểm soát ở nhiều cấp độ – từ chính sách, mạng, điểm cuối, ứng dụng đến dữ liệu và lớp giám sát–phản ứng.
* Giả định vi phạm: Luôn giả định rằng hệ thống có thể bị xâm nhập, từ đó tập trung vào khả năng phát hiện sớm, cô lập và khôi phục từ bên trong.
* Lợi ích chính của DiD:
* Tăng khả năng phòng thủ tổng thể và tính chịu lỗi của hệ thống.
* Kéo dài thời gian tấn công, tạo "thời gian vàng" cho SOC xử lý.
* Giới hạn phạm vi ảnh hưởng khi xảy ra sự cố.
* Hỗ trợ đáp ứng yêu cầu tuân thủ các chuẩn an ninh như NIST CSF, ISO/IEC 27001.

1. **Các lớp phòng thủ điển hình**  
   Mô hình DiD truyền thống thường được mô tả qua 7 lớp phòng thủ logic (từ ngoài vào trong):

* Chính sách & Con người: Chính sách an ninh, quy trình vận hành, đào tạo nhận thức cho người dùng và quản trị viên.
* An ninh vật lý: Bảo vệ cơ sở hạ tầng phần cứng, trung tâm dữ liệu, phòng máy chủ, tủ rack và kiểm soát ra/vào.
* An ninh mạng: Tường lửa, IDS/IPS, VPN, phân đoạn mạng và lọc nội dung.
* An ninh điểm cuối: Antivirus, EDR, HIDS, FIM và kiểm soát ứng dụng.
* An ninh ứng dụng: Secure coding, WAF, cấu hình an toàn và kiểm thử bảo mật ứng dụng.
* An ninh dữ liệu: Mã hóa, kiểm soát truy cập, DLP, sao lưu và khôi phục.
* Giám sát & Ứng phó: SOC, SIEM, threat intelligence, SOAR và quy trình xử lý sự cố.
* Trong mô hình SOC sử dụng Wazuh của đề tài này, các lớp được hiện thực trực tiếp chủ yếu là an ninh điểm cuối (Wazuh Agent với HIDS, FIM, Malware Detection, SCA trên Windows endpoint) và giám sát & ứng phó (Wazuh Manager, Indexer, Dashboard đóng vai trò SIEM/SOC thu thập, phân tích log và sinh cảnh báo), các lớp còn lại được đề cập ở mức khung lý thuyết để hoàn thiện mô hình Defense in Depth tổng thể.

### Layer 1 – Network & Perimeter (Router/NAT, Firewall, ACL)

Lớp này là tuyến phòng thủ đầu tiên, kiểm soát luồng lưu lượng vào/ra hệ thống và giảm bề mặt tấn công (attack surface). Nhiệm vụ chính là chặn hoặc hạn chế kết nối trái phép ngay tại biên giới mạng trước khi chúng tiếp cận tài nguyên nội bộ [19][20].

Trong triển khai thực nghiệm:

* **Router/NAT**: Sử dụng cơ chế NAT trong môi trường ảo hóa để ẩn địa chỉ IP nội bộ của máy chủ Wazuh và các máy trạm, tránh việc truy cập trực tiếp từ Internet, đồng thời chỉ cho phép lưu lượng cần thiết đi qua.
* **Firewall host-based**: Tận dụng tường lửa tích hợp (Windows Firewall trên Windows 11, UFW/iptables trên Ubuntu Server) để chỉ cho phép các kết nối phục vụ vận hành hệ thống (ví dụ: kênh agent–manager trên cổng 1514, truy cập dashboard qua HTTPS 443).

Mặc dù Wazuh tập trung vào giám sát ở tầng endpoint/host, việc duy trì một lớp phòng thủ mạng tối thiểu vẫn là bắt buộc trong chiến lược Defense in Depth, nhằm bảo đảm chỉ lưu lượng hợp lệ mới có thể tiếp cận hệ thống [19][20].

### Layer 2 – Endpoint Security (Wazuh Agent: HIDS, FIM, Malware)

Đây là lớp trọng tâm của giải pháp Wazuh, nơi các chức năng giám sát trực tiếp được triển khai trên từng thiết bị thông qua Wazuh Agent [10].

Các chức năng chính được triển khai:

* **HIDS (Host-based Intrusion Detection)**: Phân tích gần thời gian thực các log hệ thống và ứng dụng để phát hiện dấu hiệu tấn công như brute-force, leo thang đặc quyền hoặc lệnh PowerShell/binary đáng ngờ [10].
* **FIM (File Integrity Monitoring)**: Giám sát thay đổi file và registry, cảnh báo khi phát hiện tạo/sửa/xóa file trong các thư mục được bảo vệ (ví dụ thư mục FIM-Test trên Windows) [11].
* **Malware Detection dựa trên Rootcheck/IOC**: Kết hợp FIM với danh sách IOC (hash, IP, domain) và module Rootcheck để phát hiện file độc hại, dấu hiệu rootkit hoặc cấu hình hệ thống bất thường [12].

Lớp này bảo đảm rằng, ngay cả khi tấn công đã vượt qua firewall mạng, mọi hoạt động bất thường tại điểm cuối vẫn được ghi nhận và gửi về các lớp phía trên để phân tích [10][11][12].

### Layer 3 – Hardening (Configuration Assessment – SCA)

Lớp phòng thủ chủ động này tập trung vào việc loại bỏ lỗ hổng từ gốc thông qua module Security Configuration Assessment (SCA) của Wazuh [14][19].

Các hoạt động chính (khi SCA được bật trên endpoint):

* Thiết lập baseline cấu hình: So sánh cấu hình hiện tại với các policy tham chiếu (ví dụ CIS Benchmarks hoặc policy nội bộ) để xác định trạng thái an toàn mong muốn.
* Đánh giá định kỳ: Tự động kiểm tra các thiết lập bảo mật quan trọng như chính sách mật khẩu, dịch vụ mạng, audit policy, trạng thái các dịch vụ không cần thiết.
* Cảnh báo sai phạm: Ghi nhận và báo cáo chi tiết các cấu hình yếu (ví dụ mật khẩu không hết hạn, dịch vụ dư thừa đang hoạt động), kèm theo gợi ý khắc phục nếu policy hỗ trợ [14].

SCA giúp chủ động “vá” lỗ hổng cấu hình, giảm đáng kể cơ hội thành công của kẻ tấn công trước khi chúng kịp khai thác [14][19].

### Layer 4 – Threat Intelligence & Behavior (MITRE, Threat Hunting)

Lớp này nâng cao giá trị phòng thủ bằng cách bổ sung ngữ cảnh (context) và trí tuệ mối đe dọa (threat intelligence) vào dữ liệu thu thập được [12][17][20][21].

Các khả năng chính:

* **Tích hợp MITRE ATT&CK**: Ánh xạ các cảnh báo và luật phát hiện của Wazuh sang chiến thuật/kỹ thuật ATT&CK (tactic/technique ID Txxxx), hiển thị trên ma trận MITRE ATT&CK trong Dashboard. Điều này giúp xác định cảnh báo tương ứng với giai đoạn nào trong chuỗi tấn công (reconnaissance, credential access, lateral movement,…) [17].
* **Threat Hunting & IOC**:
  + Sử dụng danh sách IOC nội bộ (hash file độc, IP/domain C2, URL độc hại) trong etc/lists/ để so khớp với hoạt động trên endpoint, phát hiện mối đe dọa đã biết [12].
  + Kết hợp (nếu có) với nguồn Threat Intelligence bên ngoài để cập nhật IOC mới, đồng thời dùng khả năng truy vấn log của Wazuh để thực hiện threat hunting theo giả thuyết (hypothesis-driven) hoặc dựa trên IOC (IOC-driven) [20][21].

Lớp này chuyển hóa dữ liệu log/raw events thành thông tin giàu ngữ cảnh, hỗ trợ analyst hiểu “ai – làm gì – theo kỹ thuật ATT&CK nào” thay vì chỉ nhìn thấy từng sự kiện rời rạc.

### Layer 5 – SIEM & Correlation (Wazuh Manager, Indexer, Dashboard)

Đây là “bộ não” của hệ thống SOC Wazuh, nơi hội tụ và phân tích dữ liệu từ tất cả các lớp phía dưới [10][20][21].

Kiến trúc Wazuh ở lớp này:

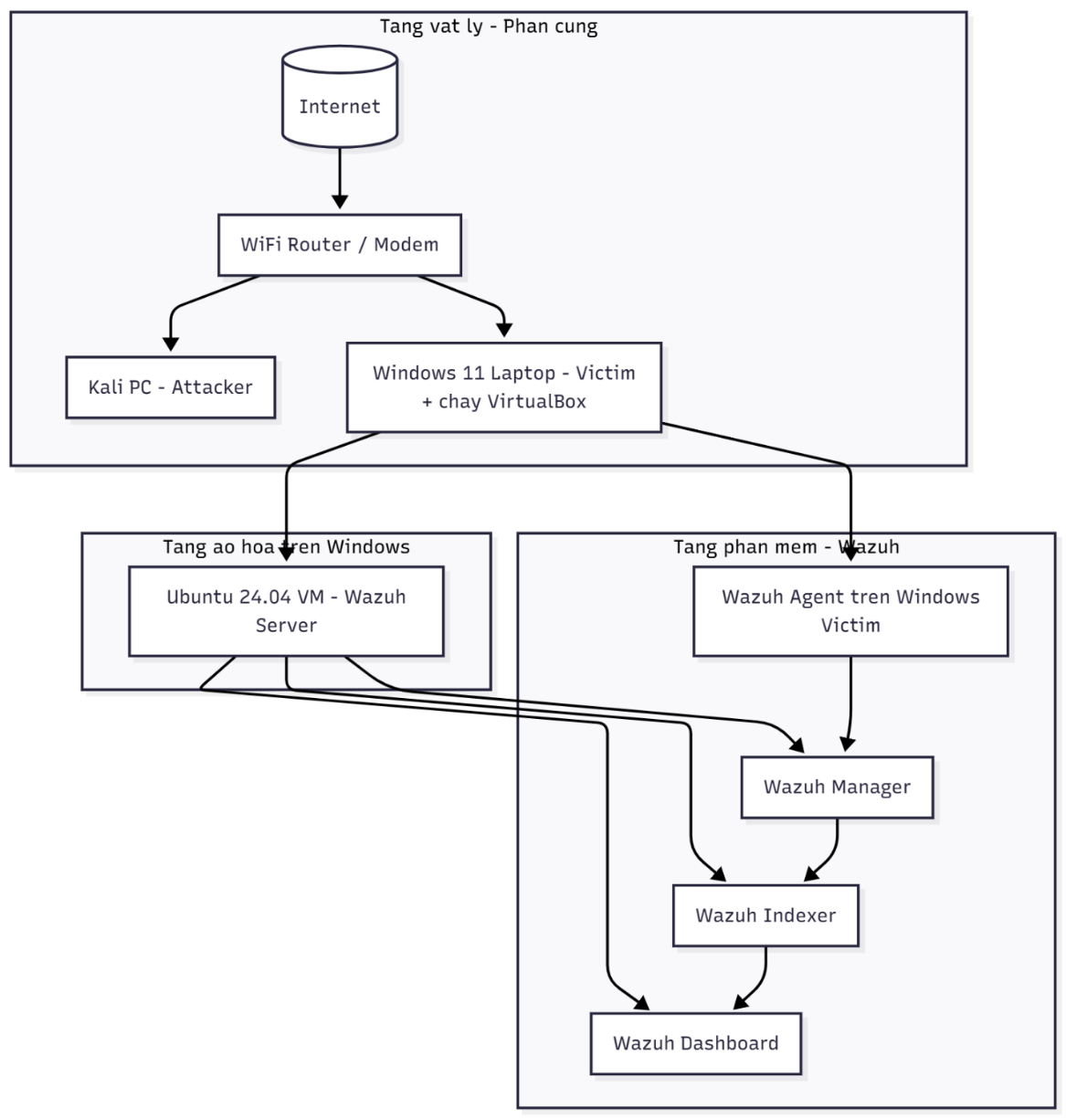
* **Wazuh Manager**: Thành phần xử lý trung tâm, nhận log/sự kiện từ Agent, thực hiện giải mã (decoding), chuẩn hóa (normalization), áp dụng rules và tương quan sự kiện để phát hiện mẫu tấn công phức tạp hoặc chuỗi sự kiện liên quan [10].
* **Wazuh Indexer (OpenSearch)**: Lưu trữ và lập chỉ mục log/alert, cho phép truy vấn nhanh phục vụ điều tra, thống kê và xây dựng dashboard [10][20].
* **Wazuh Dashboard**: Giao diện trực quan hóa thống nhất, hiển thị:
  + Trạng thái agent và module (HIDS, FIM, SCA, Malware Detection).
  + Ma trận MITRE ATT&CK và các dashboard phục vụ threat hunting.
  + Các bảng điều khiển (dashboard) và báo cáo giúp SOC nắm “bức tranh toàn cảnh” về an ninh hệ thống [10][21].

Lớp này cung cấp cái nhìn tổng thể (holistic view) và là nơi ra quyết định: từ một cảnh báo đơn lẻ ở Layer 2–4, SOC có thể truy vấn, tương quan và đánh giá mức độ nghiêm trọng để đưa ra hành động ứng phó phù hợp, hoàn thiện mô hình phòng thủ theo chiều sâu của hệ thống SOC Wazuh.

# phương pháp thực hiện

## Mô hình hệ thống SOC sử dụng nền tảng Wazuh Stack

### Mô tả môi trường thí nghiệm và các thành phần phần cứng/phần mềm



Hình 19: Sơ đồ phần cứng phần mềm hệ thống

* + 1. **Tầng vật lý – Phần cứng**
* **Kết nối Internet**  
  Hệ thống sử dụng đường truyền Internet gia đình, kết nối tới thiết bị Wi-Fi Router/Modem.
* **Wi-Fi Router/Modem**  
  Thiết bị trung tâm của mạng LAN, cấp phát địa chỉ IP nội bộ và kết nối Internet cho các máy trạm.
* **Windows 11 Laptop – Victim + chạy VirtualBox**
  + Vai trò:
    - Máy trạm sử dụng hằng ngày của người dùng.
    - Endpoint **victim** trong các kịch bản tấn công (đăng nhập sai, brute-force SMB, FIM, malware IOC, SCA).
    - Máy **host** cài VirtualBox để chạy máy ảo Ubuntu SOC.
* **Kali PC – Attacker**
  + Vai trò: máy tấn công độc lập, nằm cùng mạng LAN với Windows laptop.
  + Sử dụng để chạy các công cụ và script tấn công (thử mật khẩu SMB, sinh log brute-force, v.v.) hướng vào Windows victim.
    1. **Tầng ảo hóa trên Windows**

**Ubuntu 24.04 VM – Wazuh Server**

* + Được triển khai trên VirtualBox của Windows 11 laptop.
  + Là máy chủ SOC, cài đặt trọn bộ **Wazuh Stack** gồm:
    - Wazuh Manager
    - Wazuh Indexer
    - Wazuh Dashboard
  + Nhận log từ agent Windows, xử lý, lưu trữ và hiển thị trên Dashboard để phân tích.
    1. **Tầng phần mềm – Nền tảng Wazuh**
* **Wazuh Agent trên Windows Victim**
  + Được cài trực tiếp trên Windows 11 laptop.
  + Thu thập:
    - Windows Security Log (HIDS),
    - sự kiện FIM (thay đổi file/registry),
    - kết quả SCA,
    - các sự kiện liên quan IOC/malware.
  + Gửi log về Wazuh Manager trên Ubuntu thông qua kênh mạng Host-Only.
* **Wazuh Manager (trên Ubuntu)**
  + Nhận log từ Wazuh Agent, thực hiện giải mã (decoder), áp dụng rule, tương quan sự kiện và sinh alert.
* **Wazuh Indexer (trên Ubuntu)**
  + Lưu trữ log và alert đã xử lý, cho phép truy vấn và tìm kiếm theo thời gian, trường dữ liệu và rule.
* **Wazuh Dashboard (trên Ubuntu, truy cập từ Windows)**
  + Giao diện web cho quản trị viên SOC.
  + Hiển thị các module: Discover, Security Events, File Integrity Monitoring, Malware Detection, SCA, MITRE ATT&CK, Threat Hunting,…
  + Được truy cập từ Windows laptop, phục vụ quan sát kết quả của các kịch bản tấn công và kiểm thử trong lab.

### Sơ đồ cấu hình và kết nối mạng hệ thống SOC Wazuh

A diagram of a diagram

AI-generated content may be incorrect.

Hình 20 : sơ đồ kết nối mạng hệ thống SOC

* + - 1. **KIẾN TRÚC MẠNG TỔNG THỂ**

Hệ thống SOC Wazuh được triển khai trên mô hình lai giữa mạng vật lý thực tế và mạng ảo hóa, nhằm tách biệt môi trường tấn công, môi trường quản lý SOC và kênh Internet. Toàn bộ hệ thống gồm ba thành phần chính sau.

**1. Hạ tầng mạng thực tế (Physical Network)**

* **Router Wi-Fi / Modem (LAN)**
  + Gateway: **172.16.0.1/16**
  + Quản lý và cấp phát địa chỉ trong dải **172.16.0.0/16** cho các thiết bị nội bộ.
* **Windows 11 Host (MinhThu)**
  + Địa chỉ Wi-Fi: **172.16.107.85/16**
  + Vừa là **workstation** thông thường, vừa là **host chạy VirtualBox** và đồng thời là **Windows victim** trong các bài demo HIDS/FIM/IOC.
* **Kali Linux PC**
  + Nhận địa chỉ động trong dải **172.16.0.0/16** (do router cấp DHCP).
  + Đóng vai trò **máy tấn công (attacker)**, thực hiện brute-force SMB, thử đăng nhập, v.v. vào Windows MinhThu.

**2. Hạ tầng mạng ảo hóa (Virtual Network)**

Được triển khai bên trong VirtualBox trên máy Windows MinhThu.

* **VirtualBox NAT Network**
  + Dải: **10.0.2.0/24**
  + Gateway ảo: **10.0.2.2**
  + Cung cấp kết nối Internet cho máy ảo thông qua NAT, tách máy ảo ra khỏi LAN 172.16.0.0/16.
* **VirtualBox Host-Only Network**
  + Dải: **192.168.56.0/24**
  + Mạng nội bộ chỉ giữa host Windows và các máy ảo, không truy cập trực tiếp ra Internet.
  + Được sử dụng làm **mạng quản lý SOC**, kênh truyền log và truy cập Dashboard.
* **Ubuntu 24.04 SOC Server (soc\_ubuntu)**
  + Máy ảo cài đặt **Wazuh Manager, Wazuh Indexer và Wazuh Dashboard**.
  + Có hai giao diện mạng:
    - **10.0.2.15/24 (NAT)** – dùng để máy ảo ra Internet (cập nhật gói, truy cập tài liệu, v.v.).
    - **192.168.56.103/24 (Host-Only)** – dùng để giao tiếp với Windows host và nhận log từ Wazuh Agent.

**3. Windows Host-Only Adapter**

* **Địa chỉ:** **192.168.56.1/24**
* Là card mạng ảo trên Windows MinhThu, nằm cùng dải **192.168.56.0/24** với Ubuntu SOC Server.
* Đóng vai trò **cầu nối giữa Windows host và mạng ảo Host-Only**, phục vụ:
  + Windows MinhThu gửi log HIDS/FIM tới Wazuh Server (192.168.56.103).
  + Người quản trị truy cập giao diện Wazuh Dashboard từ Windows MinhThu qua địa chỉ 192.168.56.103.
    - 1. **PHÂN TÍCH KIẾN TRÚC BẢO MẬT VÀ TÍNH NĂNG**

**1. Phân tách chức năng theo phân đoạn mạng**

Hệ thống được chia thành ba phân đoạn mạng với chức năng và mục tiêu bảo mật riêng biệt:

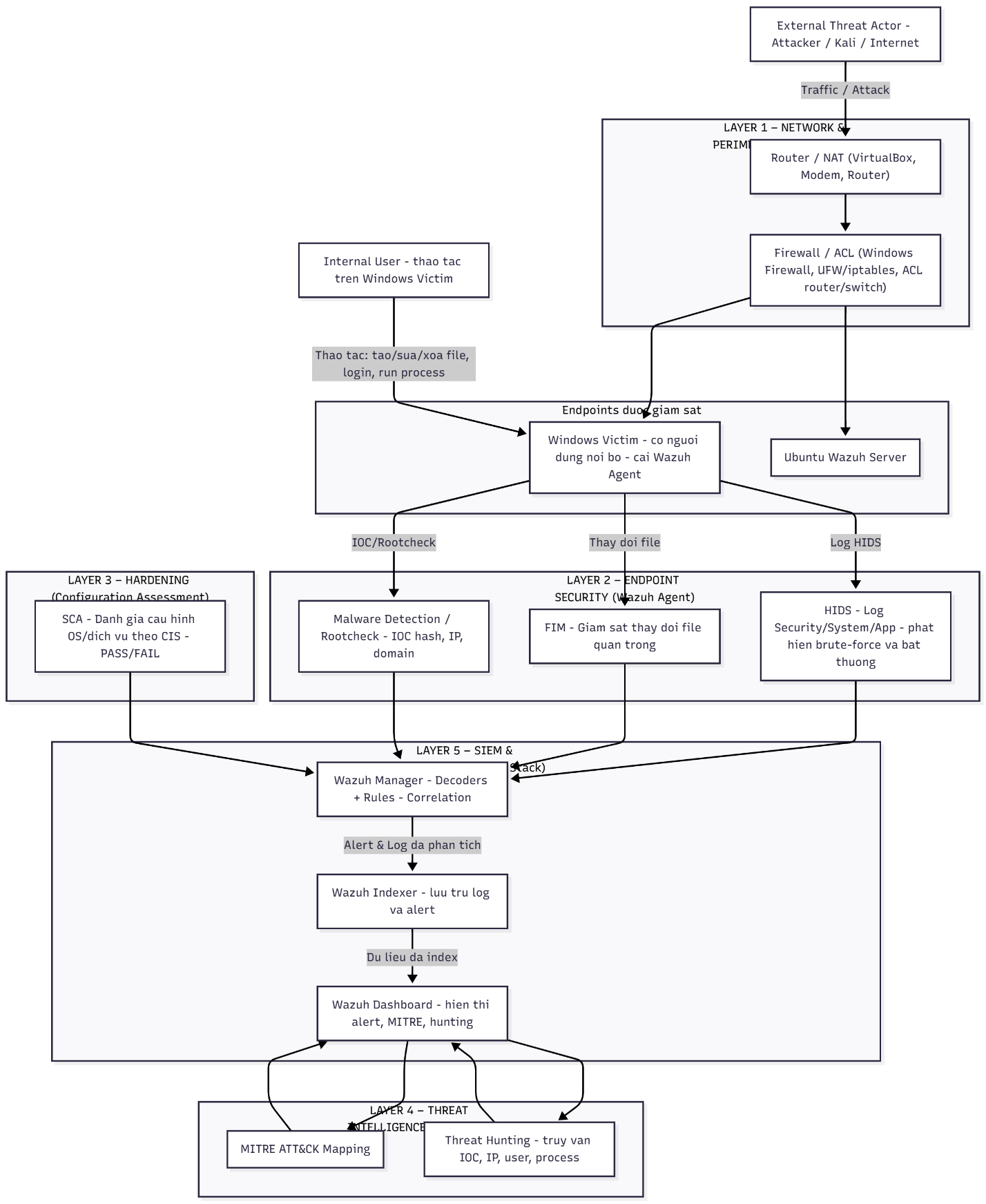
|  |  |  |  |
| --- | --- | --- | --- |
| **Phân đoạn mạng** | **Dải địa chỉ** | **Thành phần chính** | **Mục đích bảo mật** |
| **Mạng tấn công (Attack Segment)** | **172.16.0.0/16** | Router Wi-Fi/Modem, Kali Linux PC, Windows MinhThu (Wi-Fi 172.16.107.85) | Mô phỏng môi trường thực tế: Kali tấn công Windows qua LAN, cô lập lưu lượng tấn công khỏi mạng SOC. |
| **Mạng quản lý SOC (Management Segment)** | **192.168.56.0/24** | Windows Host-Only adapter (192.168.56.1), Ubuntu SOC Server (192.168.56.103) | Bảo vệ kênh thu thập log và truy cập Dashboard; endpoint gửi log về SOC qua kênh riêng, không lộ SOC ra LAN. |
| **Mạng Internet (Internet Access)** | **10.0.2.0/24** | VirtualBox NAT network, Ubuntu SOC Server (10.0.2.15) | Cung cấp Internet cho SOC Server (cập nhật hệ thống, truy cập repo), tách biệt đường đi Internet khỏi mạng tấn công. |

Bảng 1: bảng phân tách chức năng

**Windows MinhThu** là nút giao giữa hai phân đoạn:

* Thuộc **mạng tấn công** với IP Wi-Fi **172.16.107.85/16** (bị Kali tấn công).
* Đồng thời thuộc **mạng quản lý SOC** với IP Host-Only **192.168.56.1/24** (gửi log và truy cập Dashboard).

### Kiến trúc chức năng theo 5 lớp phòng thủ (Network, Endpoint, Hardening, Threat, SIEM)



Hình 21 : Sơ đồ kiến trúc chức năng 5 tầng

* 1. **Lớp Mạng và Biên (Lớp 1 – Network Perimeter)**

**Thành phần chính:**

* Router/Modem/NAT (bao gồm NAT của VirtualBox và router vật lý).
* Hệ thống Firewall/ACL: Windows Firewall trên các điểm cuối, UFW/iptables trên Ubuntu, và ACL trên thiết bị mạng (nếu có).

**Chức năng:**

* Kiểm soát lưu lượng vào/ra giữa môi trường bên ngoài (Internet/Kali) và hệ thống nội bộ.
* Áp dụng các chính sách lọc dựa trên IP, cổng (port), và giao thức để hạn chế truy cập trái phép.

**Mục tiêu bảo mật:**

* Giảm thiểu bề mặt tấn công từ bên ngoài.
* Ngăn chặn hoặc hạn chế lưu lượng tấn công trực tiếp đến các điểm cuối và máy chủ SOC.
  1. **Lớp Bảo mật Điểm cuối (Lớp 2 – Endpoint Security)**

**Thành phần:**

* Máy Windows Victim cài đặt Wazuh Agent.
* Người dùng nội bộ (Internal User) thực hiện các thao tác trên Windows như tạo/sửa/xóa tệp, đăng nhập, và chạy tiến trình.
* Các tấn công từ bên ngoài (Kali/Internet) nhắm vào Windows Victim.

**Chức năng chính của Agent:**

* **HIDS (Host-based Intrusion Detection System):** Thu thập và phân tích nhật ký Windows Security/System/Application. Phát hiện hành vi bất thường như đăng nhập sai nhiều lần, brute-force, hoặc truy cập dịch vụ trái phép.
* **FIM (File Integrity Monitoring):** Giám sát các thay đổi trên tệp và registry quan trọng (thêm, sửa, xóa).

**Ý nghĩa:**

* Cung cấp cái nhìn chi tiết về mọi hoạt động trên điểm cuối, bao gồm cả từ người dùng hợp lệ và kẻ tấn công.
* Là điểm khởi đầu của đa số luồng dữ liệu an ninh trong hệ thống.
  1. **Lớp Cấu hình và Phát hiện Phần mềm độc hại (Lớp 3 – Hardening & Malware Detection)**

**Thành phần (chạy trên Wazuh Agent và Wazuh Server):**

* **SCA (Security Configuration Assessment):** Đánh giá cấu hình hệ điều hành và dịch vụ dựa trên các tiêu chuẩn như CIS Benchmark. Trả về kết quả PASS/FAIL cho từng quy tắc (ví dụ: kiểm tra 26001 – Tuổi thọ tối đa của mật khẩu).
* **Rootcheck và Phát hiện Malware dựa trên IOC (Indicators of Compromise):** So khớp tệp, hash, IP, và domain với danh sách IOC. Phát hiện tệp nghi ngờ malware, cấu trúc thư mục bất thường, hoặc dấu hiệu rootkit.

**Mục tiêu:**

* Cứng hóa cấu hình điểm cuối (hardening) trước và sau khi xảy ra tấn công.
* Phát hiện sớm các dấu hiệu xâm nhập hoặc sự hiện diện của mã độc dựa trên IOC.
  1. **Lớp Tình báo Mối đe dọa (Lớp 4 – Threat Intelligence)**

**Thành phần:**

* Ánh xạ MITRE ATT&CK trên Wazuh Dashboard.
* Dashboard Threat Hunting (truy vấn nhật ký/cảnh báo).

**Chức năng:**

* Ánh xạ cảnh báo từ HIDS/FIM/IOC/SCA sang kỹ thuật và chiến thuật trong ma trận MITRE ATT&CK (ví dụ: Brute Force – Credential Access, Modify Registry – Defense Evasion).
* Cho phép nhà phân tích truy vấn chủ động dựa trên: IOC (hash/IP/domain), người dùng, tiến trình, agent, quy tắc, hoặc MITRE ID để tìm dấu hiệu tấn công tiềm ẩn.

**Ý nghĩa:**

* Chuyển đổi SOC từ trạng thái phòng thủ thụ động (chỉ xem cảnh báo) sang săn tìm mối đe dọa chủ động (điều tra dựa trên dữ liệu đã thu thập).
  1. **Lớp Phân tích và Hiển thị (Lớp 5 – SIEM & Dashboard)**

**Thành phần:**

* **Wazuh Manager:** Thực hiện giải mã, khớp quy tắc, và tương quan trên nhật ký từ agent. Tạo cảnh báo với mức độ nghiêm trọng, gán MITRE ID, và nhóm quy tắc (HIDS, FIM, malware, SCA, v.v.).
* **Wazuh Indexer:** Lưu trữ nhật ký và cảnh báo đã xử lý, hỗ trợ truy vấn thời gian thực và lịch sử.
* **Wazuh Dashboard:** Giao diện hiển thị cảnh báo và các dashboard chuyên biệt (Security Events, FIM, Malware Detection, SCA, MITRE ATT&CK, Threat Hunting, v.v.).

**Vai trò:**

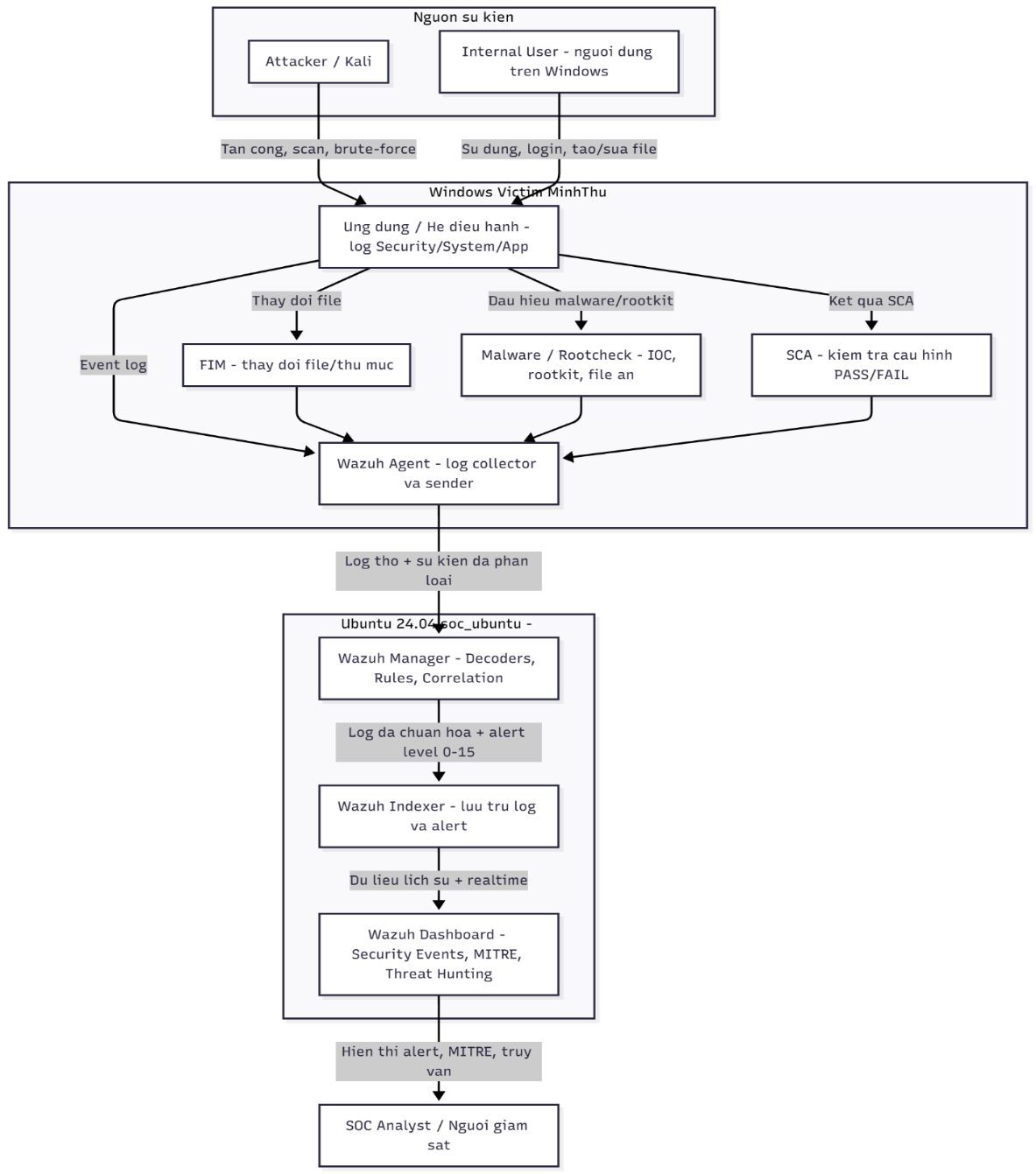
* Tập trung hóa việc thu thập, phân tích và hiển thị dữ liệu an ninh.
* Cung cấp điểm truy cập duy nhất để quản trị viên quan sát, tìm kiếm và phân tích các kịch bản tấn công trong phòng thí nghiệm.
  1. **Luồng Dữ liệu và Tích hợp Đa Lớp**

Luồng dữ liệu trong mô hình này hoạt động như sau:

* Kẻ tấn công bên ngoài (Kali/Internet) gửi lưu lượng tấn công qua Lớp 1 (Network & Perimeter) vào Windows Victim.
* Người dùng nội bộ và kẻ tấn công tạo ra các sự kiện trên Windows (đăng nhập, brute-force, thao tác tệp, chạy tiến trình).
* Wazuh Agent trên Windows ghi nhận: nhật ký HIDS (Security/System/App), sự kiện FIM, kết quả SCA, và sự kiện Rootcheck/IOC.
* Các sự kiện này được gửi về Wazuh Manager → giải mã, áp dụng quy tắc, tương quan → tạo cảnh báo.
* Cảnh báo và nhật ký được lưu tại Wazuh Indexer và hiển thị trên Wazuh Dashboard.
* Từ Dashboard, các module MITRE ATT&CK và Threat Hunting sử dụng dữ liệu đã lập chỉ mục để: tổng hợp theo chiến thuật/kỹ thuật, hỗ trợ nhà phân tích truy vấn IOC, IP, người dùng, tiến trình, và xây dựng bức tranh toàn cảnh của chiến dịch tấn công.

Nhờ cấu trúc nhiều lớp này, cùng một sự kiện (ví dụ: brute-force SMB từ Kali) có thể được quan sát đồng thời từ nhiều góc độ: nhật ký chi tiết ở Lớp 2 (HIDS), dấu hiệu malware/IOC ở Lớp 3, kỹ thuật MITRE ở Lớp 4, và cảnh báo hợp nhất cùng timeline trên Lớp 5. Điều này giúp hệ thống SOC Wazuh trong môi trường thí nghiệm đạt được cơ chế giám sát phòng thủ theo chiều sâu hiệu quả.

### Sơ đồ luồng log và sự kiện trong hệ thống



Hình 22 : sơ đồ luồng log và sự kiện

* 1. **Nguồn sinh sự kiện**
* **Attacker / Kali**
  + Thực hiện tấn công, scan, brute-force… vào Windows Victim qua mạng LAN/Internet.
* **Internal User – người dùng trên Windows**
  + Thao tác hợp lệ: đăng nhập, mở ứng dụng, tạo/sửa/xóa file, chạy tiến trình.

Hai nhóm tác nhân này đều tác động lên **Windows Victim MinhThu** và tạo ra log ở các kênh khác nhau.

* 1. **Thu thập sự kiện trên Windows Victim MinhThu**

Trên Windows Victim, các hoạt động được ghi nhận dưới nhiều dạng:

* **Nhật ký hệ điều hành / ứng dụng**

Security/System/Application log ghi lại đăng nhập, quyền truy cập, lỗi hệ thống, lỗi ứng dụng,…

* **FIM – thay đổi file/thư mục**

Các thao tác tạo/sửa/xóa file tại thư mục được giám sát (FIM-Test) sinh sự kiện FIM.

* **Malware / Rootcheck – IOC, rootkit, file ẩn**

Khi file, hash, IP, domain trùng với IOC hoặc có dấu hiệu rootkit/cấu trúc bất thường, module Malware/Rootcheck sinh sự kiện tương ứng.

* **SCA – kiểm tra cấu hình PASS/FAIL**

Module SCA chạy định kỳ, so sánh cấu hình Windows với bộ rule CIS, cho kết quả từng check ở trạng thái PASS hoặc FAIL (ví dụ check 26001).

Tất cả các dòng log và kết quả kiểm tra này đều được **Wazuh Agent trên Windows** thu thập.

* + - 1. **Vai trò của Wazuh Agent – log collector và sender**
* **Thu thập (collector)**
  + Lấy log thô từ Windows Event Log, sự kiện FIM, kết quả SCA, sự kiện Rootcheck/IOC.
* **Đóng gói và gửi (sender)**
  + Chuẩn hóa tối thiểu và gửi về **Wazuh Manager** trên máy Ubuntu soc\_ubuntu qua kênh đã cấu hình (Host-Only 192.168.56.0/24).

Tại thời điểm này, dữ liệu vẫn ở dạng log thô kèm một số trường đã được agent gắn thêm (agent.id, agent.name,…).

* + - 1. **Xử lý trên Ubuntu soc\_ubuntu – Wazuh Manager và Indexer**

1. **Wazuh Manager – Decoders, Rules, Correlation**
   * Nhận log thô + sự kiện phân loại từ agent.
   * Áp dụng **decoder** để tách trường (IP nguồn, user, eventID, path, hash,…).
   * Áp dụng **ruleset** để xác định mức độ nghiêm trọng, gắn nhãn HIDS/FIM/Malware/SCA, gán **MITRE ID** nếu phù hợp.
   * Thực hiện **correlation** giữa nhiều sự kiện (ví dụ nhiều lần 4625 gần nhau → brute-force).
   * Kết quả là **log đã chuẩn hóa + alert với level 0–15**.
2. **Wazuh Indexer – lưu trữ log và alert**
   * Nhận log/alert đã xử lý từ Manager.
   * Lưu vào index, cho phép truy vấn cả dữ liệu **thời gian thực** và **lịch sử** theo thời gian, field, rule, MITRE ID,…
     + 1. **Hiển thị trên Wazuh Dashboard và phân tích bởi SOC analyst**

* **Wazuh Dashboard – Security Events, MITRE, Threat Hunting**
  + Kết nối tới Indexer để lấy dữ liệu đã lập chỉ mục.
  + Hiển thị theo nhiều góc nhìn:
    - Security Events (tất cả alert/log quan trọng),
    - File Integrity Monitoring, Malware Detection, SCA,
    - MITRE ATT&CK (tactic/technique),
    - Threat Hunting (truy vấn IOC, IP, user, process,…).
* **SOC Analyst / Người giám sát**
  + Truy cập Dashboard từ Windows host, xem và lọc alert theo kịch bản (HIDS, FIM, IOC, SCA).
  + Thực hiện threat hunting dựa trên log đã chuẩn hóa, MITRE mapping và timeline.
    - 1. **Ý nghĩa của sơ đồ luồng**
* Thể hiện rõ chuỗi **Nguồn sự kiện → Agent → Manager → Indexer → Dashboard → SOC analyst**.
* Chứng minh rằng mọi kịch bản demo (đăng nhập sai, brute-force SMB, thay đổi file, file IOC, check SCA) đều đi chung một pipeline xử lý, chỉ khác module sinh sự kiện ban đầu.
* Khẳng định vai trò trung tâm của **Wazuh Agent** ở phía endpoint và **Wazuh Manager/Indexer/Dashboard** ở phía SOC trong việc hiện thực hóa mô hình giám sát phòng thủ theo chiều sâu.

## Quy trình triển khai nền tảng Wazuh Stack

### Cài đặt Wazuh Stack trên máy chủ Ubuntu 24.04

Máy chủ SOC được triển khai trên hệ điều hành Ubuntu 24.04.3 LTS (môi trường máy ảo VirtualBox), cài đặt Wazuh Stack theo hướng dẫn chính thức (phiên bản 4.x, mô hình all-in-one: Manager, Indexer, Dashboard trên cùng một máy)

Quy trình cài đặt được thực hiện tóm tắt như sau:

* Cập nhật hệ điều hành và cài đặt các gói phụ thuộc cần thiết.
* Tải script cài đặt Wazuh từ kho chính thức và phân quyền thực thi.
* Chạy script ở chế độ all-in-one để tự động cài đặt Wazuh Manager, Wazuh Indexer, Wazuh Dashboard và cấu hình cơ bản.

# Cập nhật hệ điều hành

sudo apt update && sudo apt upgrade -y

# Kiểm tra phiên bản Ubuntu

lsb\_release -a

# Tải script cài đặt Wazuh 4.x

curl -sO https://packages.wazuh.com/4.x/wazuh-install.sh

# Phân quyền và chạy script chế độ all-in-one

sudo chmod +x wazuh-install.sh

sudo ./wazuh-install.sh -a

Sau khi cài đặt, trạng thái các dịch vụ chính của Wazuh Stack được kiểm tra bằng systemctl. Kết quả cho thấy cả ba thành phần **wazuh-manager**, **wazuh-indexer** và **wazuh-dashboard** đều đang hoạt động ở trạng thái active (running)

sudo systemctl status wazuh-manager

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 23: Trạng thái hoạt động của wazuh-manager

sudo systemctl status wazuh-indexer

A screenshot of a computer

AI-generated content may be incorrect.

Hình 24: trạng thái hoạt động của wazuh-indexer

sudo systemctl status wazuh-dashboard

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 25: Trạng thái hoạt động của wazuh-dashboard

Tiếp theo, các cổng dịch vụ quan trọng được kiểm tra bằng ss để xác nhận Wazuh Stack đã lắng nghe trên các port mặc định cho agent và Indexer:

sudo ss -tulnp | grep -E '1514|1515|9200|443'

A screenshot of a computer

AI-generated content may be incorrect.

Hình 26: các cổng hoạt động trên wazuh-stack

### Cài đặt và đăng ký Wazuh Agent trên Windows Victim

Máy Windows 11 (MinhThu) được cài đặt Wazuh Agent và đăng ký kết nối về Wazuh Manager theo hướng dẫn chính thức của Wazuh cho hệ điều hành Windows [10]. Quá trình thực hiện gồm: tải bộ cài Agent, khai báo địa chỉ IP/hostname của Wazuh Manager trong quá trình cài đặt, sau đó khởi động dịch vụ Agent và kiểm tra trạng thái từ cả phía server (Ubuntu) và phía client (Windows).

**Kiểm tra từ phía Wazuh Manager (Ubuntu)**

Sau khi hoàn tất cài đặt và đăng ký Agent trên Windows, danh sách agent được kiểm tra bằng công cụ agent\_control trên máy chủ Wazuh:

sudo /var/ossec/bin/agent\_control -l

A screenshot of a computer

AI-generated content may be incorrect.

Hình 27: Danh sách Agent

Kết quả này cho thấy:

* Agent ID **000** là chính máy chủ Wazuh (b2203739-VirtualBox), trạng thái **Active/Local**.
* Agent ID **001** là máy Windows Victim **MinhThu**, trạng thái **Active**, chứng tỏ Agent đã đăng ký và đang kết nối bình thường tới Wazuh Manager.

**Kiểm tra từ phía Windows Victim**

Trên máy Windows 11, trạng thái dịch vụ Wazuh Agent được kiểm tra bằng PowerShell:

Get-Service | Where-Object { $\_.Name -like "Wazuh\*" }

A black screen with text and a black background

AI-generated content may be incorrect.

Hình 28: trạng thái Wazuh-Agent hoạt động trên Windows 11

Kết quả cho thấy dịch vụ **WazuhSvc** đang ở trạng thái **Running**, xác nhận rằng Wazuh Agent trên Windows đã khởi động thành công và sẵn sàng gửi log/sự kiện về Wazuh Manager [10].

## Thiết kế kịch bản kiểm th chức năng

### Nguyên tắc xây dựng kịch bản kiểm thử

Kiểm tra script được thiết kế dựa trên các nguyên tắc sau:

Nguyên tắc 1: Bám sát mục tiêu đề tài – Mỗi kịch bản phải gắn trực tiếp với một hoặc nhiều chức năng cốt lõi của Wazuh trong mô hình SOC (HIDS, FIM, Phát hiện phần mềm độc hại/Rootcheck, SCA, MITER ATT&CK, SIEM/Tương quan)

Nguyên tắc 2: Tái lập môi trường dễ dàng trong phòng thí nghiệm trường – Các bước thực hiện phải rõ ràng, sử dụng các tầng hạ tầng hiện có (máy chủ Ubuntu + Windows Victim), không yêu cầu thêm thiết bị chuyên dụng ngoài phạm vi phòng thí nghiệm.

Nguyên tắc 3: Phản ánh tình huống thực tế – bản mô phỏng các hành vi tấn công/vi phạm điển hình (đăng nhập sai, thay đổi tệp, tệp có IOC, cấu hình yếu) thường gặp trong môi trường doanh nghiệp.

Nguyên tắc 4: Có tiêu chí quan sát và đánh giá rõ ràng – Mỗi kịch bản đều phải sinh ra log/alert cụ thể, có thể quan sát trên Wazuh Dashboard (hoặc qua tệp cảnh báo/log) và đối chiếu với mô-đun liên kết (FIM, SCA, MITER ATT&CK, vv).

Nguyên tắc 5: An toàn trong lab – Chỉ sử dụng dữ liệu/kịch bản “giả lập” (IOC test, file không gây hại, thao tác trên thư mục test), không ảnh hưởng đến bên ngoài hệ thống.

### Mô tả chung 5 kịch bản kiểm thử

**Kịch bản 1 – HIDS/SIEM: Phát hiện đăng nhập sai trên Windows**

**Mục tiêu**  
Kiểm tra khả năng thu thập và phân tích sự kiện bảo mật từ Windows Event Log (đặc biệt là sự kiện đăng nhập sai) của HIDS và kênh SIEM trong Wazuh [10].

**Cách thực hiện tổng quát**

* Thực hiện nhiều lần đăng nhập sai vào máy Windows Victim để sinh sự kiện bảo mật (ví dụ: Event ID 4625 trong Security Log).
* Kiểm tra trên Wazuh Dashboard xem có xuất hiện cảnh báo HIDS tương ứng hay không, nội dung cảnh báo có chứa thông tin user, IP, thời gian, loại lỗi đăng nhập.
* Nếu ruleset đã được ánh xạ MITRE, kiểm tra kỹ thuật tương ứng trên ma trận MITRE ATT&CK (ví dụ kỹ thuật brute force / credential access) [10][17].

**Tiêu chí đánh giá**

**Pass**: Wazuh Dashboard hiển thị ít nhất một cảnh báo liên quan đến sự kiện đăng nhập sai (Event ID 4625), với mức độ nghiêm trọng phù hợp, trong vòng ≤ 5 phút sau khi thực hiện chuỗi đăng nhập sai.

**Fail**: Không có cảnh báo được sinh ra, hoặc cảnh báo không phản ánh đúng loại sự kiện (sai Event ID, thiếu thông tin chính).

**Kịch bản 2 – FIM: Phát hiện thay đổi tệp tin/thư mục nhạy cảm**

**Mục tiêu**  
Xác minh chức năng giám sát tính toàn vẹn tệp tin (File Integrity Monitoring – FIM) trên thư mục được bảo vệ [11].

**Cách thực hiện tổng quát**

* Cấu hình FIM giám sát thư mục thử nghiệm (ví dụ: C:\Users\Public\FIM-Test) trên Windows.
* Thực hiện lần lượt các thao tác: tạo mới file, sửa nội dung file, xóa file trong thư mục này.
* Kiểm tra trên Wazuh Dashboard các sự kiện FIM về: loại thao tác (create/modify/delete), hash trước/sau (nếu có), đường dẫn file và thời gian.

**Tiêu chí đánh giá**

**Pass**: Dashboard ghi nhận đầy đủ sự kiện FIM với:

Đúng loại thao tác (tạo/sửa/xóa),

Đúng đường dẫn tệp,

Có thông tin hash trước/sau đối với thao tác sửa (modify),  
trong vòng ≤ 2 phút kể từ thời điểm thao tác.

**Fail**: Thiếu một trong các thao tác (create/modify/delete), thông tin sai/không đầy đủ, hoặc không có sự kiện FIM tương ứng.

**Kịch bản 3 – Phát hiện phần mềm độc hại dựa trên IOC / Rootcheck**

**Mục tiêu**  
Đánh giá khả năng phát hiện mối đe dọa dựa trên IOC và module Rootcheck của Wazuh [12].

**Cách thực hiện tổng quát**

Cấu hình danh sách IOC (hash file, IP, domain) trong thư mục etc/lists/ trên Wazuh Manager và bật module Rootcheck theo tài liệu [12].

Tạo một tệp “giả lập mã độc” trên Windows với hash trùng một IOC trong danh sách test, hoặc kích hoạt kịch bản làm Rootcheck phát hiện cấu hình/đối tượng bất thường.

Quan sát cảnh báo Malware/Rootcheck trên Dashboard, kiểm tra phần nội dung IOC (hash/IP/domain) được match và mức độ nghiêm trọng (severity).

**Tiêu chí đánh giá**

* + **Pass**: Dashboard hiển thị cảnh báo Malware/Rootcheck với:
    - IOC khớp chính xác (đúng hash/IP/domain test),
    - Mức độ nghiêm trọng ở mức cao (high hoặc tương đương),  
      trong vòng ≤ 10 phút sau khi tạo tệp/kích hoạt Rootcheck.
  + **Fail**: Không có cảnh báo, IOC hiển thị sai hoặc không khớp, mức độ cảnh báo không tương xứng.

**Kịch bản 4 – SCA: Phát hiện cấu hình bảo mật yếu**

**Mục tiêu**  
Kiểm tra module Security Configuration Assessment (SCA) trong việc phát hiện cấu hình sai (misconfiguration) trên endpoint Windows [14][19].

**Cách thực hiện tổng quát**

* Kích hoạt module SCA và gán policy phù hợp cho Windows Victim (ví dụ policy CIS Windows).
* Thực hiện quét cấu hình (scan) theo lịch hoặc kích hoạt thủ công.
* Trên Dashboard, quan sát kết quả SCA: điểm tuân thủ (compliance score) và danh sách các kiểm tra Failed (ví dụ: chính sách mật khẩu, dịch vụ không cần thiết, audit policy chưa bật đầy đủ).

**Tiêu chí đánh giá**

**Pass**: Dashboard hiển thị:

* + - Điểm tuân thủ (compliance score) < 80% (phản ánh có nhiều cấu hình chưa đạt),
    - Ít nhất 03 kiểm tra ở trạng thái Failed được liệt kê chi tiết trong vòng ≤ 15 phút sau khi chạy scan.

**Fail**: Điểm tuân thủ quá cao (gần 100%) trong khi thực tế cấu hình chưa được harden, hoặc không có/ít kiểm tra Failed, hoặc kết quả scan không được hiển thị.

**Kịch bản 5 – MITRE ATT&CK & Threat Hunting: Phân tích chuỗi tấn công và truy vết sự kiện**

**Mục tiêu:**  
Đánh giá khả năng của Wazuh trong việc ánh xạ cảnh báo sang khung **MITRE ATT&CK** và hỗ trợ **săn tìm mối đe dọa (threat hunting)** dựa trên chuỗi sự kiện sinh ra từ một cuộc tấn công thực nghiệm.

**Mô tả chung:**  
Kịch bản giả lập một máy tấn công Kali Linux trong cùng mạng ảo, thực hiện các hoạt động thăm dò và tấn công vào máy Windows Victim (MinhThu), ví dụ như đăng nhập thất bại nhiều lần vào dịch vụ chia sẻ tệp (SMB). Các sự kiện bảo mật trên Windows (nhật ký Security, sự kiện đăng nhập thất bại, cảnh báo HIDS…) được Wazuh Agent thu thập, gửi về Wazuh Manager và sinh cảnh báo trong index wazuh-alerts-\*.  
Trên Dashboard, các cảnh báo này được tự động gán vào **tactic/kỹ thuật MITRE** liên quan (Credential Access / Brute Force, v.v.), đồng thời xuất hiện trên các bảng điều khiển **MITRE ATT&CK** và **Threat Hunting** dưới dạng biểu đồ tactic, kỹ thuật và agent bị ảnh hưởng.

**Kết quả mong đợi:**

Hệ thống hiển thị được các tactic/kỹ thuật MITRE tương ứng với chuỗi hành vi tấn công từ Kali vào Windows.

Người vận hành có thể bắt đầu từ dashboard MITRE ATT&CK hoặc Threat Hunting để nhận diện máy bị tấn công, IP nguồn, tài khoản bị nhắm tới và thời điểm tấn công, qua đó chứng minh khả năng **phân tích chuỗi tấn công và truy vết sự kiện** của SOC Wazuh.

# KIỂM THỬ VÀ ĐÁNH GIÁ CÁC TÍNH NĂNG

## Tổng quan các kịch bản kiểm thử

### Mục tiêu kiểm thử hệ thống SOC Wazuh

Giai đoạn kiểm thử nhằm đánh giá mức độ đáp ứng của mô hình SOC Wazuh so với các yêu cầu chức năng (Chương 2) và kiến trúc phòng thủ theo chiều sâu (Chương 3). Các mục tiêu cụ thể:

* **Kiểm chứng luồng dữ liệu:**  
  Đảm bảo log/sự kiện từ Windows Victim được Wazuh Agent thu thập, gửi về Wazuh Manager và lưu trữ đầy đủ tại Wazuh Indexer; có thể truy vấn và hiển thị trên Dashboard.
* **Đánh giá năng lực phát hiện:**  
  Xác minh độ chính xác và độ nhạy của các module cốt lõi đã triển khai trong đề tài:
  + HIDS (giám sát sự kiện bảo mật Windows, đặc biệt là đăng nhập thất bại).
  + FIM (giám sát thư mục FIM-Test trên Windows).
  + Malware Detection dựa trên IOC + Rootcheck.
  + SCA (Security Configuration Assessment) trên Windows Victim.  
    Đồng thời đánh giá khả năng tương quan, phân loại và cảnh báo sự kiện của Wazuh Manager (vai trò SIEM).
* **Đánh giá hỗ trợ vận hành SOC:**  
  Kiểm tra khả năng trực quan hóa cảnh báo, ánh xạ tactic/kỹ thuật theo **MITRE ATT&CK**, cũng như hỗ trợ quá trình **threat hunting** và truy vết nguồn gốc sự cố trên Wazuh Dashboard.

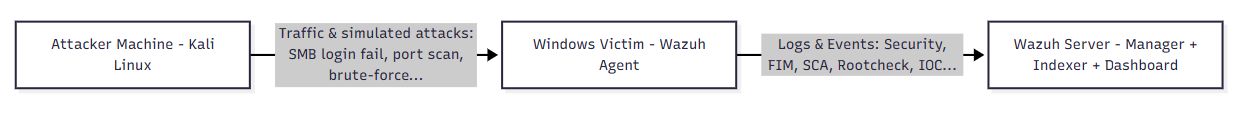
### Danh sách và mô tả các tính năng kiểm thử

* **Thành phần hệ thống được kiểm thử**
  + Wazuh Manager chạy trên Ubuntu Server 24.04 (VM b2203739-VirtualBox).
  + Wazuh Indexer (OpenSearch) tích hợp cùng Manager.
  + Wazuh Dashboard (giao diện web).
  + Wazuh Agent cài trên máy Windows 11 (MinhThu – Victim).
* **Các module Wazuh trong phạm vi kiểm thử**
  + HIDS / SIEM: thu thập và phân tích log Security/Systems trên Windows.
  + FIM: giám sát thư mục thử nghiệm C:\Users\Public\FIM-Test.
  + Malware Detection dựa trên IOC và Rootcheck (sử dụng các file danh sách IOC trong /var/ossec/etc/lists/malicious-ioc/).
  + SCA: đánh giá cấu hình bảo mật Windows (dùng policy SCA do Wazuh cung cấp).
  + MITRE ATT&CK Integration và Threat Hunting: tổng hợp, ánh xạ tactic/kỹ thuật và hỗ trợ truy vấn săn tìm mối đe dọa.
* **Loại kiểm thử**
  + Kiểm thử chức năng (Functional Testing) cho từng module.
  + Kiểm thử hiệu quả phát hiện cơ bản (mức độ có/không, mức độ cảnh báo, ánh xạ ATT&CK).
  + Kiểm thử an ninh ở mức lab (không công bố ra bên ngoài hệ thống).
* **Thời gian kiểm thử**
  + Dự kiến khoảng **40 giờ làm việc** gồm: chuẩn bị môi trường, cấu hình, chạy 5 kịch bản, thu thập và phân tích kết quả.

### Môi trường & Công cụ kiểm thử

**A. Kiến trúc môi trường lab**

* Máy tấn công: **Kali Linux 2022.2** (VirtualBox, chế độ Host-Only 192.168.56.0/24).
* Máy nạn nhân: **Windows 11 – MinhThu**, cài Wazuh Agent, nằm cùng mạng 192.168.56.0/24 với Kali và Wazuh server.
* Máy chủ SOC: **Ubuntu Server 24.04 – b2203739-VirtualBox**, cài Wazuh Manager + Indexer + Dashboard.



Hình 29: Sơ đồ luồng tổng quát

**B. Thông số kỹ thuật (mô tả ở mức tổng quan)**

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Hệ điều hành / Phiên bản** | **Vai trò chính** |
| Wazuh Manager | Ubuntu Server 24.04, Wazuh 4.x | Phân tích, tương quan, sinh alert |
| Wazuh Indexer | Tích hợp cùng Manager (OpenSearch) | Lưu trữ và lập chỉ mục log/alert |
| Wazuh Dashboard | Web UI (Node.js, Wazuh 4.x) | Giao diện quan sát, MITRE, Threat Hunting |
| Windows Victim | Windows 11, Wazuh Agent 4.x | Endpoint được giám sát |
| Kali Linux | Kali-Linux-2022.2 VM | Máy tấn công sinh sự kiện kiểm thử |

Bảng 2: Thông số kỹ thuật

**C. Công cụ và dữ liệu mẫu thực tế sử dụng**

* **Công cụ trên Kali (tấn công mô phỏng):**
  + nmap: scan mạng/cổng để sinh log tường lửa/IDS.
  + smbclient: thử đăng nhập SMB với mật khẩu sai, tạo sự kiện 4625 (failed logon).
  + Một số lệnh mạng cơ bản (ping, nc, …) phục vụ kiểm tra kết nối.
* **Hành vi trên Windows Victim (sinh log):**
  + Đăng nhập sai tên/mật khẩu từ Kali → sinh Security Event ID 4625.
  + Thao tác tạo/sửa/xóa file trong thư mục FIM-Test → sinh log FIM.
  + Chạy SCA task từ Wazuh → sinh kết quả kiểm tra cấu hình.
* **IOC và “malware” giả lập:**
  + File hash độc hại được khai báo thủ công trong:
    - /var/ossec/etc/lists/malicious-ioc/malware-hashes
    - /var/ossec/etc/lists/malicious-ioc/malicious-ip
    - /var/ossec/etc/lists/malicious-ioc/malicious-domains
  + File test trên Windows có hash trùng với IOC để kích hoạt cảnh báo Malware Detection/Rootcheck.
  + Không sử dụng VirusTotal/YARA/THOR; toàn bộ IOC được quản lý nội bộ trong Wazuh Manager.
* **Chính sách tham chiếu cho SCA:**
  + Bộ rule SCA mặc định của Wazuh cho Windows, dựa trên các khuyến nghị hardening tương đương CIS benchmark (cụ thể rule ID và nội dung được trích từ file cấu hình SCA trong phần phụ lục).

## Kiểm thử tính năng HIDS – Giám sát đăng nhập sai trên Windows (SIEM + HIDS)

### Mô hình thí nghiệm và kịch bản đăng nhập sai liên tiếp trên Windows endpoint

**Môi trường thí nghiệm**

Endpoint: máy Windows Laptop (hostname **MinhThu**) đóng vai trò **victim**, đã cài **Wazuh Agent** và hiển thị trạng thái *Online* trên Wazuh Dashboard.

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 30: trạng thái của dịch vụ Wazuh Agent trên windows

Máy chủ SOC: máy ảo Ubuntu cài đặt **Wazuh Manager + Wazuh Indexer + Wazuh Dashboard**, đóng vai trò thu thập, phân tích và lưu trữ log.

Kênh log sử dụng: **Windows Security Log**, tập trung vào sự kiện **Event ID 4625 – An account failed to log on**.

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 31: Kênh log sử dụng

**Kịch bản tấn công mô phỏng**

* Hành vi: Thực hiện 5 lần đăng nhập sai liên tiếp vào tài khoản local MinhThu ngay trên máy Windows (Local Interactive Logon).
* Mục tiêu kỹ thuật: Kích hoạt Event ID 4625 trong Windows Security Log, mô phỏng giai đoạn đầu của cuộc tấn công brute-force password.
* Điểm đặc biệt: Tấn công được thực hiện trực tiếp trên console (IP nguồn 127.0.0.1), không qua mạng, để kiểm tra khả năng giám sát HIDS tại chỗ.

### Quy trình thực nghiệm tạo sự kiện đăng nhập sai (Event ID 4625) và sinh log

**Bước 1: kiểm tra tiền đề và cấu hình**

Xác minh kết nối Agent:

# Trên Wazuh Manager kiểm tra agent

sudo /var/ossec/bin/agent\_control -l | grep MinhThu

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 32: Xac minh kết nối Agent trên Wazuh manager

* Agent hiển thị trạng thái **Active**, phiên bản, và địa chỉ IP.

cấu hình giám sát Windows Log (đã được thiết lập từ trước):

* Agent cấu hình mặc định giám sát Windows Security Log.
* Rule set mặc định của Wazuh đã bao gồm decoding cho Event ID 4625.

**Bước 2: tạo sự kiện đăng nhập sai**

* 1. Trên máy MinhThu:

Nhấn Win+L để khóa màn hình

Nhập sai mật khẩu 3-5 lần liên tiếp

Mỗi lần sai sẽ thấy cảnh báo bảo mật không đúng

* 1. Với mỗi lần nhập sai, Windows sinh một bản ghi trong **Security Log** với **Event ID 4625 – Failed Logon**.

Thu thập và gửi log bởi Wazuh Agent

* Wazuh Agent trên Windows được cấu hình giám sát kênh **Security**.
* Các sự kiện 4625 mới sinh ra được agent đọc, đóng gói và gửi về **Wazuh Manager** qua kênh giao tiếp bảo mật (TLS).
  1. Phân tích, lưu trữ và hiển thị trên Wazuh
* **Wazuh Manager** nhận log, áp dụng bộ giải mã (decoder) và rule HIDS tương ứng với Event ID 4625 để phân loại sự kiện và gán mức độ cảnh báo.
* Sự kiện sau khi được phân tích được gửi sang **Wazuh Indexer** để lưu trữ.
* **Wazuh Dashboard** truy vấn dữ liệu từ Indexer, cho phép hiển thị và phân tích các sự kiện 4625 trên giao diện Discover/Dashboards.

### Kết quả ghi nhận log 4625 trên Wazuh Manager và hiển thị trên Dashboard

Sau khi thực hiện kịch bản đăng nhập sai, kết quả quan sát trên Wazuh Dashboard như sau:

1. **Truy vấn trên Discover**

* Chọn index pattern **wazuh-alerts-\***.
* Áp dụng các bộ lọc:
  + agent.name: "MinhThu" – chỉ lấy log từ máy Windows victim.
  + data.win.system.eventID: 4625 – chỉ lấy các sự kiện đăng nhập thất bại.
* Kết quả: xuất hiện nhiều bản ghi (hits) trong khoảng thời gian tương ứng với lúc thực hiện đăng nhập sai.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 33: Truy vấn từ hệ thống

1. **Các trường dữ liệu quan trọng**

|  |  |  |
| --- | --- | --- |
| **Trường dữ liệu (field)** | **Giá trị minh họa trong demo** | **Ý nghĩa trong ngữ cảnh thí nghiệm** |
| agent.name | MinhThu | Tên agent / hostname của máy Windows endpoint sinh ra log. |
| agent.ip | 192.168.56.1 | Địa chỉ IP của máy Windows endpoint trong mô hình mạng SOC. |
| data.win.system.eventID | 4625 | Mã sự kiện Windows **Failed Logon** (đăng nhập không thành công) trong Security Log. |
| data.wineventdata.targetUserName | MinhThu | Tài khoản người dùng mà kẻ tấn công (hoặc người dùng) cố gắng đăng nhập nhưng nhập sai mật khẩu. |
| data.win.eventdata.ipAddress | 127.0.0.1 | Địa chỉ IP nguồn của phiên đăng nhập; 127.0.0.1 thể hiện việc đăng nhập sai được thực hiện trực tiếp trên máy. |
| data.win.eventdata.subjectLogonId | 0x3e7 | Logon ID của tài khoản/tiến trình ghi log; giá trị này thường tương ứng với tài khoản hệ thống cục bộ (SYSTEM). |
| data.win.eventdata.subjectDomainName | Phụ thuộc cấu hìnhmáy | Domain / tên máy / workgroup gắn với tài khoản ghi log; giá trị cụ thể tùy cấu hình Windows trong thí nghiệm. |
| data.win.system.channel | Security | Cho biết sự kiện thuộc kênh **Windows Security Log**. |
| rule.id (nếu hiển thị) | ID rule tương ứng | Mã rule của Wazuh dùng để nhận diện và phân loại sự kiện 4625. |
| rule.level (nếu hiển thị) | Ví dụ: 5, 7… | Mức độ cảnh báo do Wazuh gán cho sự kiện, phục vụ ưu tiên xử lý trong SOC. |

Bảng 3: Các trường dữ liệu

1. Diễn giải ý nghĩa

* Chuỗi các sự kiện 4625 hiển thị trên Dashboard, với cùng agent.name, targetUserName và ipAddress trong một khoảng thời gian ngắn, phản ánh chính xác hành vi nhập sai mật khẩu liên tiếp đối với tài khoản người dùng trên endpoint.
* Điều này chứng minh pipeline: **Windows Security Log → Wazuh Agent → Wazuh Manager → Wazuh Indexer → Dashboard** hoạt động đầy đủ và nhất quán.

### Đánh giá khả năng phát hiện và cảnh báo đăng nhập sai/brute-force mô phỏng của HIDS/SIEM

Từ kết quả trên, có thể đánh giá khả năng phát hiện và cảnh báo của hệ thống như sau:

1. **Khả năng thu thập và tập trung log**

* Mọi lần đăng nhập sai trên endpoint Windows đều sinh ra Event ID 4625 và được **Wazuh Agent** thu thập đầy đủ.
* Các sự kiện này được tập trung về Wazuh Manager và lưu trong **Wazuh Indexer**, cho phép truy vấn tập trung trên Dashboard.  
  → Hệ thống đáp ứng yêu cầu cơ bản của một SOC: tập trung log bảo mật từ endpoint.

1. **Khả năng phát hiện hành vi bất thường (HIDS)**

* Các rule HIDS của Wazuh đã nhận diện chính xác sự kiện Failed Logon (4625) và gắn nhãn cảnh báo tương ứng.
* Trên Dashboard có thể lọc nhanh theo eventID, agent.name, targetUserName, ipAddress để phân tích các chuỗi đăng nhập thất bại.  
  → Hệ thống chứng minh được khả năng phát hiện hành vi đăng nhập sai và hành vi mô phỏng brute-force ở mức đơn giản trên endpoint.

1. **Khả năng hỗ trợ phân tích của SIEM**

* Giao diện Discover/Dashboards cho phép xem chi tiết từng sự kiện, các trường như targetUserName, ipAddress, subjectLogonId, channel…, từ đó hỗ trợ phân tích nguồn gốc và bối cảnh của sự kiện.
* Dựa trên dữ liệu 4625 đã được ghi nhận, người quản trị có thể xây dựng thêm các rule tương quan (correlation rules) để:
* Phát hiện brute-force thực thụ (ví dụ N lần 4625 từ cùng một IP trong một khoảng thời gian T).
* Kết hợp với các sự kiện 4624 (logon thành công) sau chuỗi 4625 để phát hiện đăng nhập nghi ngờ.

## Kiểm thử tính năng FIM – Giám sát thay đổi file(FIM)

### Thiết lập thư mục FIM-Test và phạm vi giám sát trên Windows Victim

Endpoint sử dụng trong bài kiểm thử là máy Windows Laptop (hostname **MinhThu**), đã cài **Wazuh Agent** và được cấu hình kích hoạt module **File Integrity Monitoring (FIM)**. Phạm vi giám sát được khai báo trên Wazuh Manager trỏ tới thư mục dùng cho thí nghiệm:

* Đường dẫn giám sát: C:\Users\Public\FIM-Test\\*
* Chế độ: giám sát **realtime** (syscheck.mode = realtime), ghi nhận ngay khi có thay đổi thêm/sửa/xóa file trong thư mục.

Thư mục kiểm thử được tạo trực tiếp trên máy Windows bằng lệnh PowerShell:

New-Item -ItemType Directory C:\Users\Public\FIM-Test

Lệnh trên cho phép chứng minh rằng thư mục tồn tại trên endpoint, là nguồn sinh log cho bài kiểm thử FIM.

### 4.3.2. Quy trình tạo/sửa/xóa file để kiểm thử FIM

Để tạo ra các sự kiện FIM tương ứng với ba loại hành động “thêm mới”, “chỉnh sửa” và “xóa” file, quy trình thao tác trên Windows được thực hiện tuần tự như sau:

1. **Tạo file demo.txt trong thư mục FIM-Test**

echo "hello fim" > C:\Users\Public\FIM-Test\demo.txt

Hành động này tạo một file mới demo.txt trong thư mục giám sát, dự kiến sinh sự kiện FIM với trường syscheck.event = added.

1. **Chỉnh sửa nội dung file demo.txt**

echo "update" >> C:\Users\Public\FIM-Test\demo.txt

Câu lệnh trên ghi thêm nội dung vào file đã tồn tại, làm thay đổi kích thước và hàm băm của file. FIM dự kiến ghi nhận sự kiện với syscheck.event = modified.

1. **Xóa file demo.txt**

del C:\Users\Public\FIM-Test\demo.txt

Hành động xóa file trong thư mục được giám sát sẽ sinh sự kiện FIM với syscheck.event = deleted.

Toàn bộ thao tác được thực hiện trên cùng một endpoint **MinhThu**, trong khoảng thời gian ngắn, nhằm kiểm chứng khả năng ghi nhận đủ ba loại hành vi thay đổi file cơ bản của FIM.

### Quy trình tạo/sửa/xóa file để kiểm thử FIM

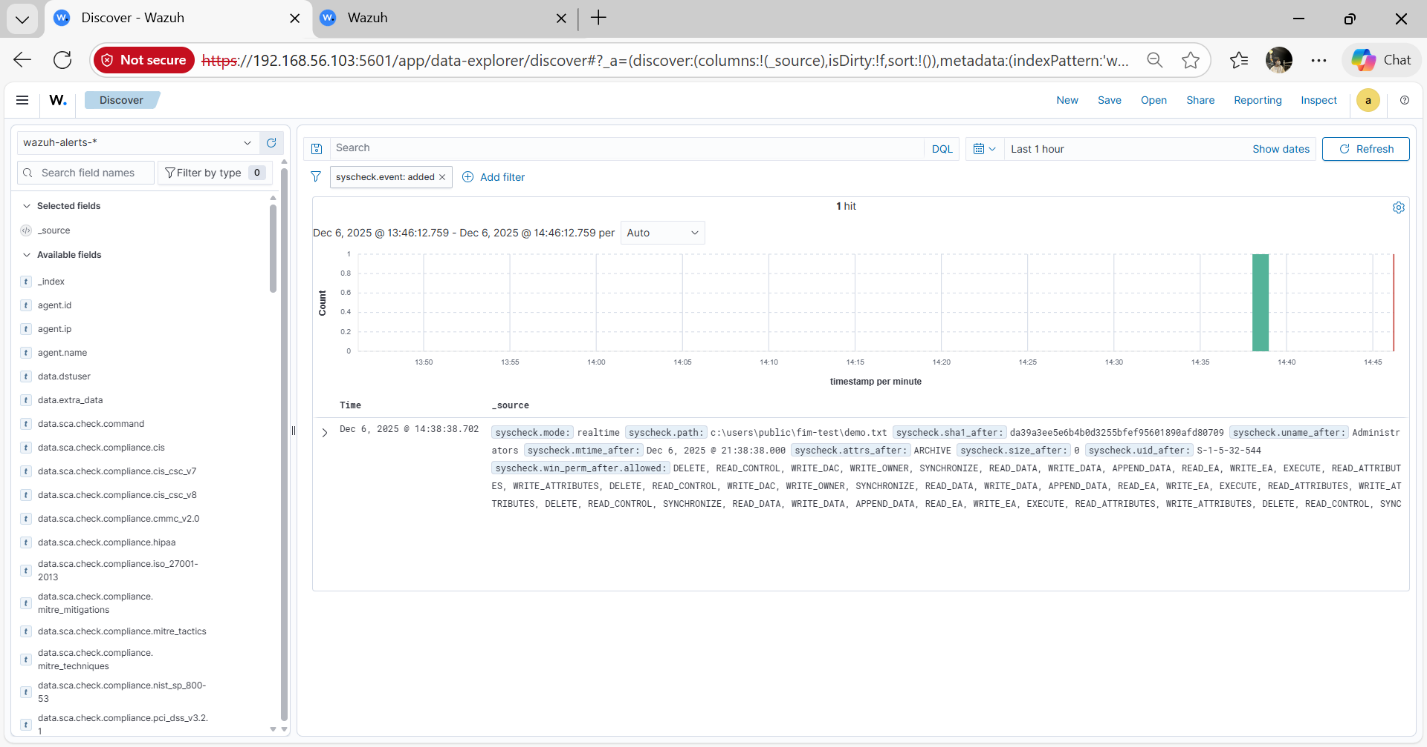
Sau khi thực hiện quy trình trên, kết quả quan sát trên Wazuh Dashboard thể hiện qua hai mức: tổng quan trên giao diện File Integrity Monitoring và chi tiết trên Discover.

1. **Tổng quan trên tab File Integrity Monitoring**

Trên màn hình **File Integrity Monitoring**:

* Biểu đồ **Alerts by action over time** hiển thị các đỉnh tương ứng với số lượng cảnh báo FIM theo từng phút, phân loại theo hành động **added / modified / deleted**.
* Biểu đồ **Top 5 agents** cho thấy toàn bộ cảnh báo FIM trong khoảng thời gian thí nghiệm đều đến từ agent **MinhThu**, khẳng định endpoint Windows đang gửi log FIM về đúng Manager.
* Biểu đồ **Actions** cho thấy phần lớn sự kiện thuộc loại **modified**, kèm theo một tỷ lệ nhỏ các sự kiện **added** và **deleted**. Điều này phù hợp với thực tế: hệ thống FIM đồng thời giám sát cả registry và file, nên số lượng sự kiện modified có thể rất lớn.
* Mục **Top 5 users** liệt kê hai tài khoản sinh thay đổi chính là **Administrators** và **SYSTEM**, với số lượng cảnh báo khác nhau. Điều này phản ánh việc cả tài khoản quản trị và tiến trình hệ thống đều có thể tạo/ghi/xóa file hoặc thay đổi registry trong phạm vi được giám sát.

1. **Chi tiết log FIM trên Discover – sự kiện “added”**



Khi truy vấn chi tiết:

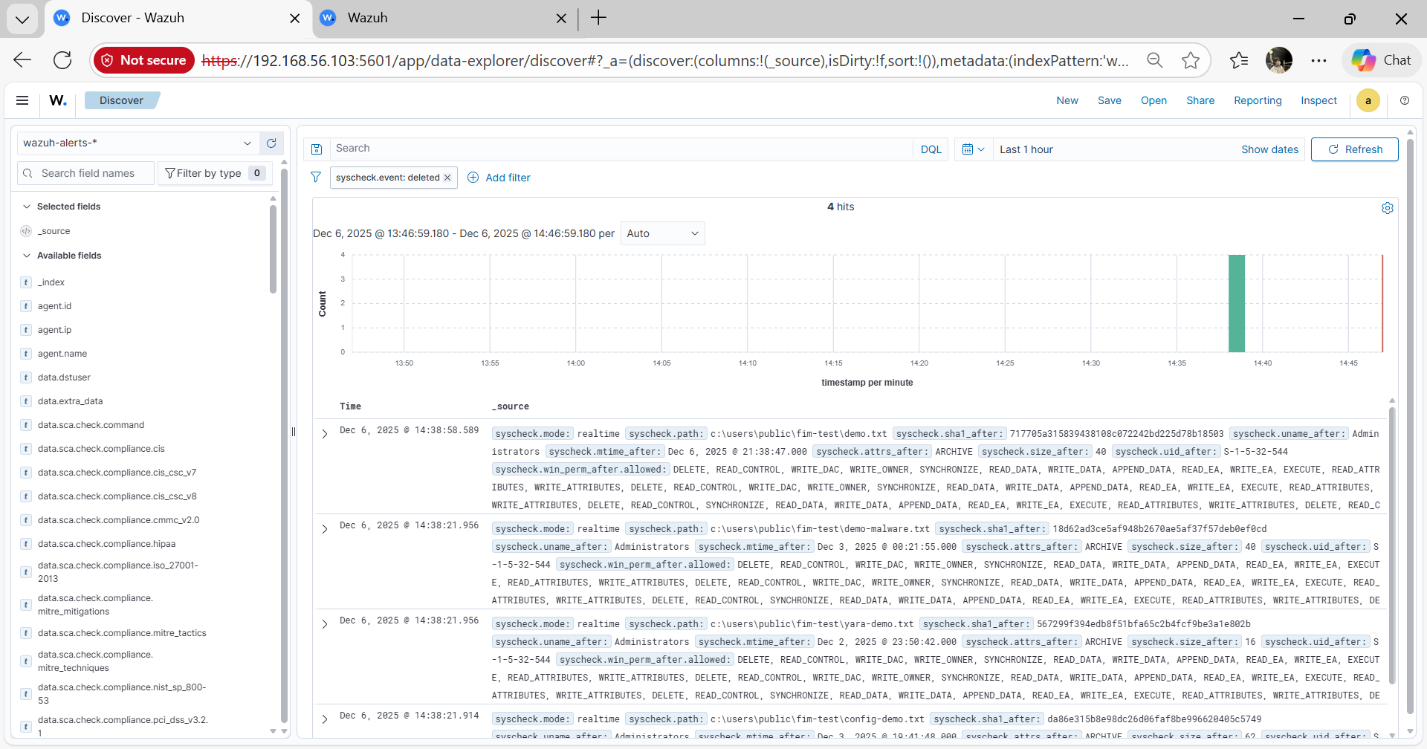
* Chọn index pattern: wazuh-alerts-\*.
* Áp dụng filter:
  + agent.name: "MinhThu"
  + syscheck.event: "added"

Kết quả trả về bản ghi có các trường tiêu biểu:

* syscheck.mode = realtime
* syscheck.path = c:\users\public\fim-test\demo.txt
* syscheck.sha1\_after hiển thị giá trị hàm băm sau khi file được tạo.
* syscheck.uname\_after = Administrators (hoặc tài khoản tương ứng)

Bản ghi này chứng minh FIM đã phát hiện và ghi nhận việc **tạo mới** file demo.txt đúng thời điểm thực nghiệm.

1. **Chi tiết log FIM trên Discover – sự kiện “deleted”**

****

Hình 34: chi tiết sự kiện

Tiếp tục lọc:

* agent.name: "MinhThu"
* syscheck.event: "deleted"

Kết quả hiển thị nhiều bản ghi, trong đó có các đường dẫn:

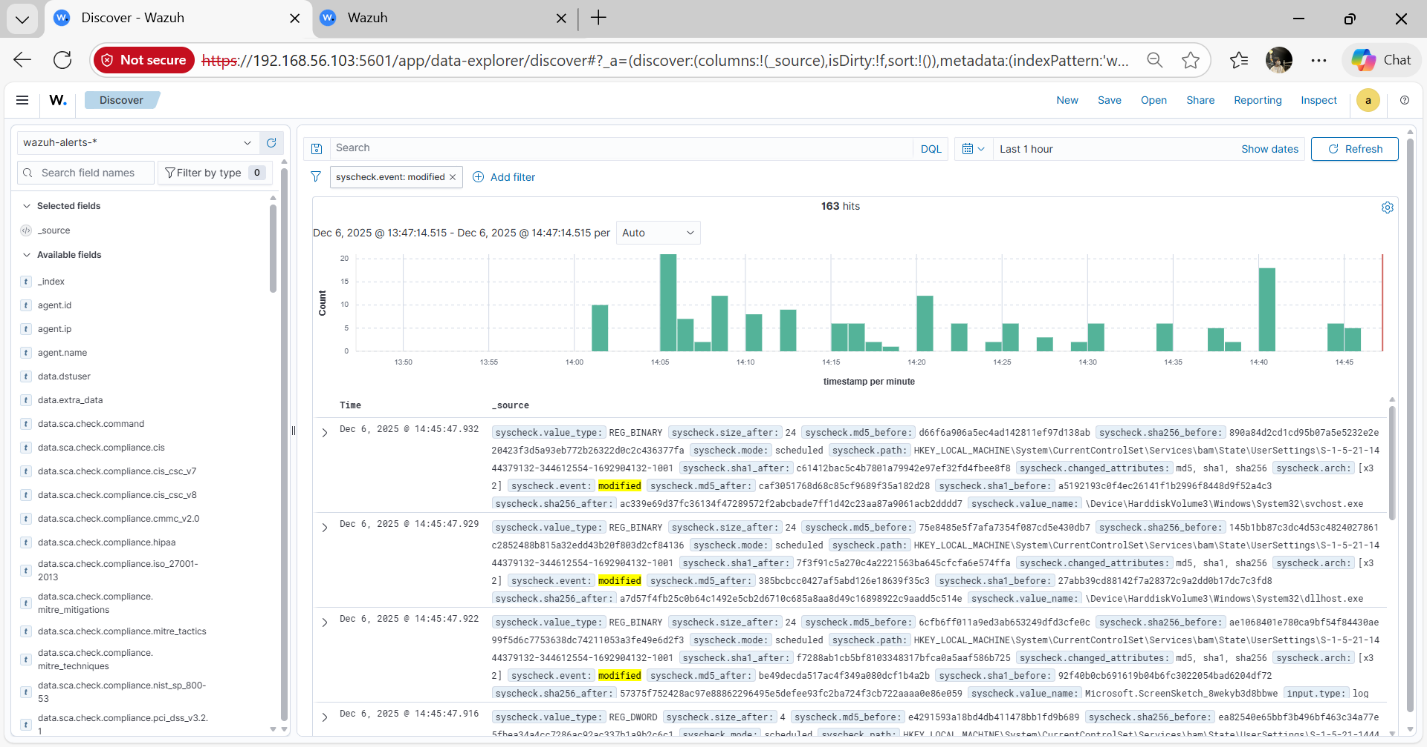
* syscheck.path = c:\users\public\fim-test\demo.txt
* syscheck.path = c:\users\public\fim-test\demo-malware.txt
* syscheck.path = c:\users\public\fim-test\test-yara-demo.txt

Các bản ghi này đều có:

* syscheck.mode = realtime
* syscheck.size\_after = 0 đối với file đã bị xóa
* Thông tin user (ví dụ syscheck.uname\_after = Administrators)

Điều này cho thấy mọi thao tác xóa file trong thư mục FIM-Test (kể cả các file khác bạn dùng cho các demo khác) đều được FIM ghi nhận với syscheck.event = deleted.

1. **Chi tiết log FIM trên Discover – sự kiện “modified”**

****

Hình 35: chi tiết log FIM

Khi lọc:

* agent.name: "MinhThu"
* syscheck.event: "modified"

Hệ thống trả về số lượng lớn bản ghi. Trong số đó có:

* Bản ghi liên quan đến file demo.txt hoặc các file khác trong thư mục FIM-Test.
* Nhiều bản ghi khác liên quan đến **registry** (ví dụ đường dẫn HKEY\_LOCAL\_MACHINE\System\...), do FIM đồng thời giám sát cả **registry key/value**.

Các trường tiêu biểu:

* syscheck.changed\_attributes chứa giá trị md5, sha1, sha256 đối với các thay đổi nội dung file.
* syscheck.sha1\_before và syscheck.sha1\_after cho thấy hàm băm trước và sau khi chỉnh sửa.

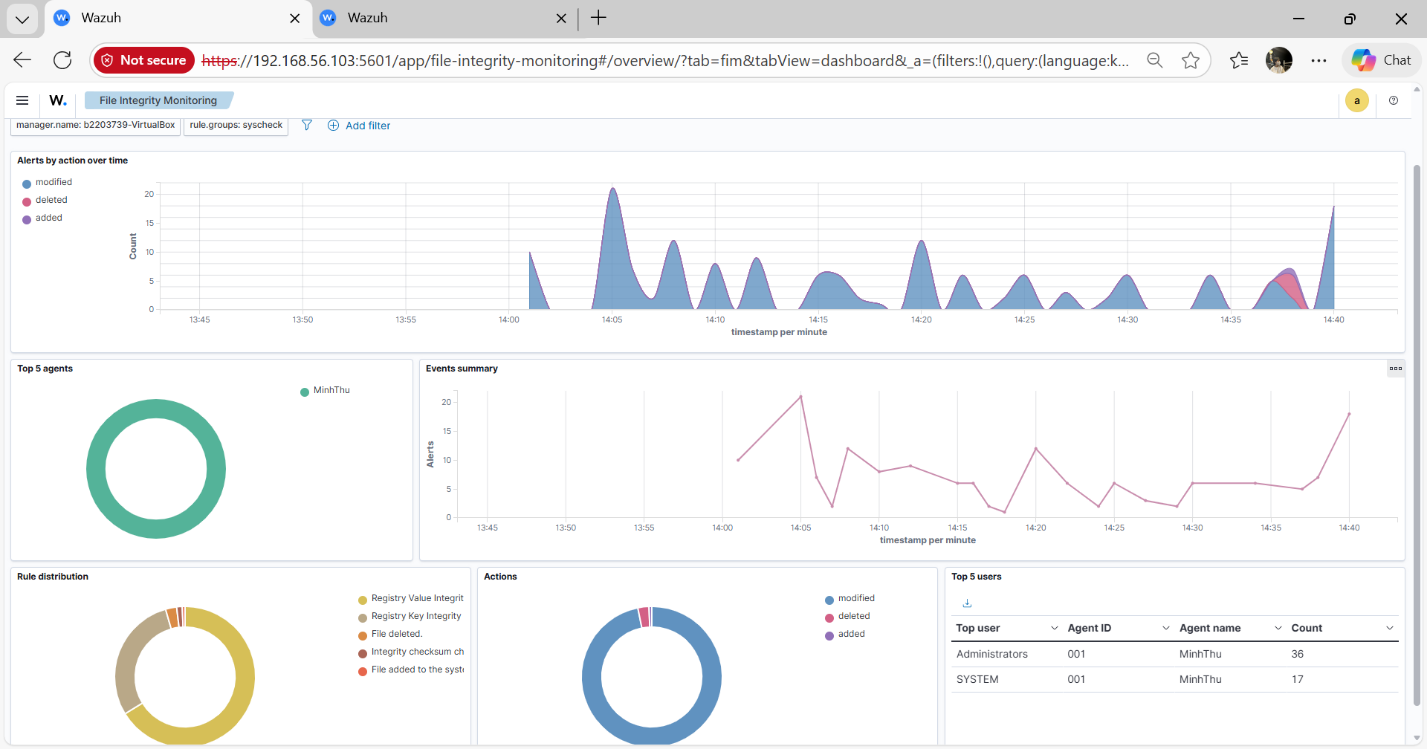
Kết quả này chứng minh FIM không chỉ ghi nhận việc sửa đổi file demo.txt mà còn log mọi thay đổi nội dung và thuộc tính trong phạm vi được cấu hình.

1. Bảng ý nghĩa trường syscheck.event sử dụng trong báo cáo

|  |  |
| --- | --- |
| **Giá trị syscheck.event** | **Ý nghĩa trong kết quả FIM** |
| added | FIM phát hiện file mới được tạo trong đường dẫn giám sát. |
| modified | FIM phát hiện nội dung hoặc thuộc tính file/registry thay đổi. |
| deleted | FIM phát hiện file/registry key bị xóa khỏi hệ thống. |

Bảng 4: giá trị syscheck.event

### Kết quả cảnh báo FIM trên Wazuh Dashboard



Hình 36: kết quả FIM trên Wazuh Dashboard

Hình thể hiện giao diện **File Integrity Monitoring** của Wazuh trong khoảng thời gian thực hiện thí nghiệm FIM trên agent Windows **MinhThu**:

* Biểu đồ **“Alerts by action over time”** ở phía trên hiển thị số lượng cảnh báo FIM theo từng phút, phân loại theo ba hành động **modified**, **deleted** và **added**. Diện tích vùng màu cho thấy đa số cảnh báo thuộc loại *modified*, trong khi *deleted* và *added* xuất hiện với tần suất thấp hơn.
* Biểu đồ **“Top 5 agents”** (góc dưới bên trái) thể hiện duy nhất một agent là **MinhThu**, chứng tỏ toàn bộ cảnh báo FIM trong khoảng thời gian quan sát đều xuất phát từ máy Windows victim này.
* Biểu đồ **“Events summary”** (giữa) mô tả sự thay đổi số lượng cảnh báo FIM theo thời gian. Các đỉnh trên đồ thị trùng với thời điểm thực hiện các thao tác tạo, sửa và xóa file trong thư mục C:\Users\Public\FIM-Test và các thay đổi registry trên endpoint.
* Biểu đồ **“Rule distribution”** cho thấy phân bố các loại rule FIM được kích hoạt. Phần lớn cảnh báo thuộc các rule về **Registry Value Integrity** và **Registry Key Integrity**, bên cạnh đó là các rule liên quan đến **file deleted**, thay đổi **checksum** và **file added**, phản ánh việc hệ thống giám sát đồng thời cả file system và registry.
* Biểu đồ **“Actions”** tổng hợp tỉ lệ ba loại hành động **modified / deleted / added**. Vùng màu tương ứng với **modified** chiếm ưu thế, phù hợp với việc có nhiều thay đổi nội dung file/registry trong suốt quá trình kiểm thử.
* Bảng **“Top 5 users”** (góc dưới bên phải) liệt kê các tài khoản sinh nhiều sự kiện FIM nhất. Kết quả thí nghiệm ghi nhận hai tài khoản chính là **Administrators** và **SYSTEM**, với số lượng cảnh báo khác nhau, cho thấy cả tài khoản quản trị và tiến trình hệ thống đều tham gia các thao tác bị FIM giám sát.

## Kiểm thử tính năng Malware Detection dựa trên IOC

### Cấu hình danh sách IOC (malware-hashes, malicious-ip, malicious-domains)

Trong mô hình SOC, tính năng Malware Detection của Wazuh được cấu hình dựa trên các danh sách IOC (Indicator of Compromise) nội bộ:

* etc/lists/malicious-ioc/malware-hashes – lưu trữ các **hash** của file bị coi là độc hại (SHA256).
* etc/lists/malicious-ioc/malicious-ip – lưu trữ danh sách địa chỉ IP độc hại (không sử dụng trực tiếp trong demo này).
* etc/lists/malicious-ioc/malicious-domains – lưu trữ tên miền độc hại (không sử dụng trực tiếp trong demo này).

Trong cấu hình Wazuh Manager, các file list này được nạp thông qua thẻ <list> trong ossec.conf, cho phép engine rule tham chiếu danh sách IOC khi phân tích log.

Đối với demo, chỉ sử dụng danh sách **malware-hashes**. Nội dung được bổ sung thêm một dòng:

c3596874c280f79951bcb10e6875883b39c7a52c4b0eebbb806fbc79a0c06ac0: DemoTestFile

A screenshot of a computer

AI-generated content may be incorrect.

Hình 37: Thêm nội dung vào danh sách malware-hashes

Trong đó:

* Chuỗi đầu là **SHA256** của file “giả malware” sẽ tạo trên Windows.
* Trường thứ hai DemoTestFile là nhãn (description) để hiển thị trong log.

Sau khi chỉnh sửa, Wazuh Manager được khởi động lại để nạp lại danh sách IOC:

sudo systemctl restart wazuh-manager

### Quy trình tạo file “giả malware” và/hoặc mô phỏng kết nối đến IOC

Quy trình thực nghiệm được thực hiện trên thư mục FIM-Test của endpoint Windows **MinhThu**:

Tạo file “giả malware” mới

"this is demo malware file for IOC test" | Out-File "C:\Users\Public\FIM-Test\demo-malware.txt" -Encoding ASCII

Tính SHA256 của file

Get-FileHash "C:\Users\Public\FIM-Test\demo-malware.txt" -Algorithm SHA256

Kết quả trả về SHA256:

A screen shot of a computer

AI-generated content may be incorrect.

Hình 38: chuỗi SHA256 được sinh

Chuỗi này được chuyển sang dạng chữ thường và thêm vào file malware-hashes như trình bày ở mục 4.4.1.

1. **Tạo lại sự kiện FIM để kích hoạt kiểm tra IOC**

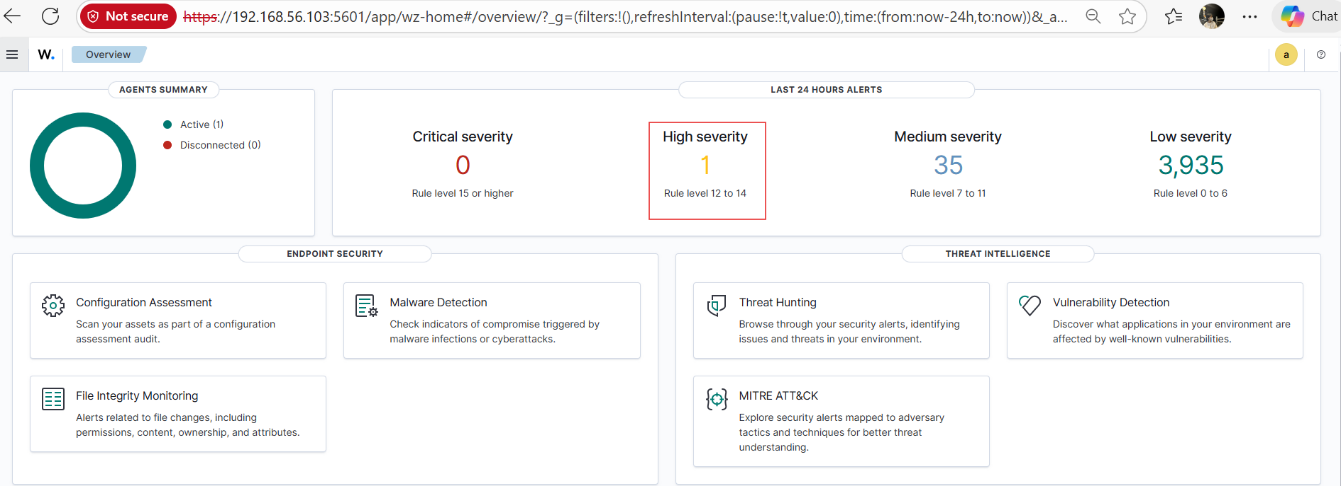
Sau khi Manager nạp IOC mới, cần sinh ra một sự kiện FIM mới liên quan tới file demo-malware.txt để Wazuh so sánh hash của file với danh sách IOC:

"modify content to trigger fim again" | Out-File "C:\Users\Public\FIM-Test\demo-malware.txt" -Encoding ASCII

* + Hành động này thay đổi nội dung file, làm FIM sinh sự kiện syscheck.event = modified. Trong quá trình xử lý, engine rule sẽ tính lại hash và so khớp với danh sách malware-hashes. Khi trùng khớp, rule thuộc nhóm **malware** sẽ được kích hoạt và sinh cảnh báo Malware Detection ở mức độ cao.

### Kết quả cảnh báo Malware Detection trên Wazuh

**Tổng quan mức độ cảnh báo trên trang Overview**



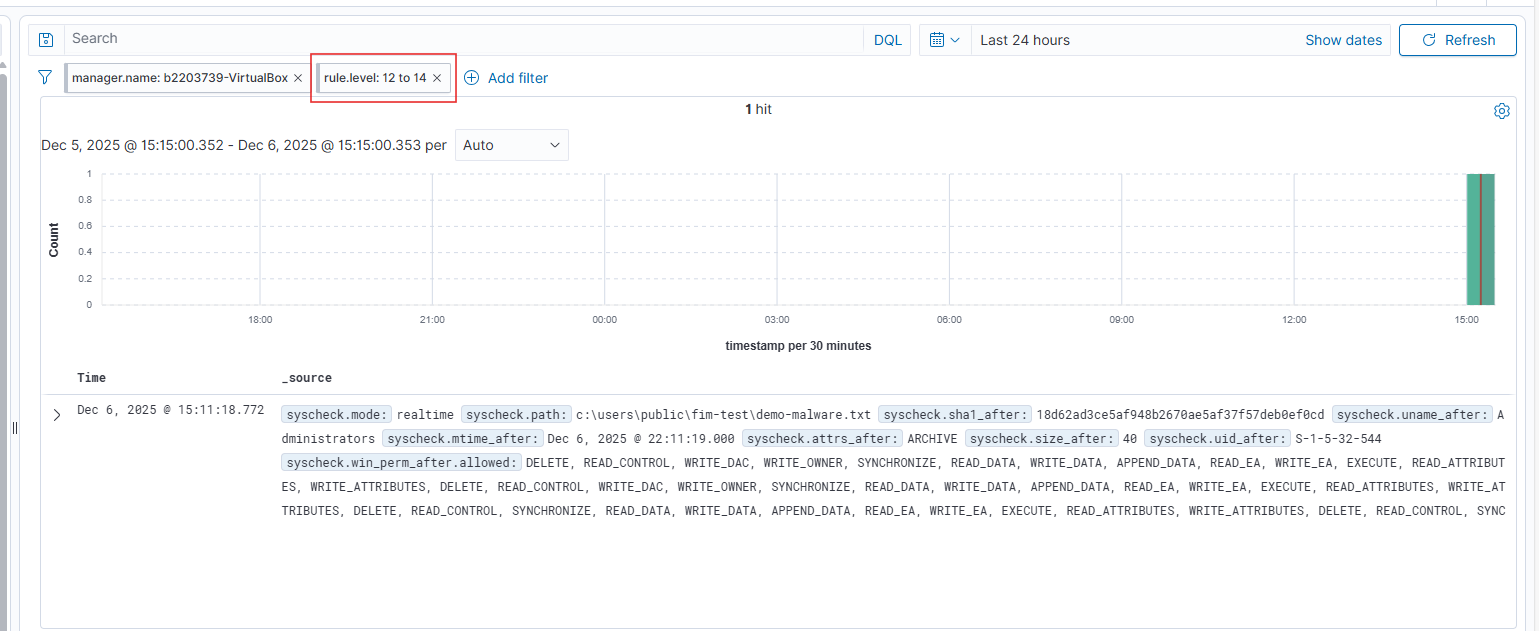
Hình 39: Xem mức độ cảnh báo của malware trên Overiew

Trên màn hình **Overview**, khối **“Last 24 hours alerts”** hiển thị:

* **High severity: 1 (Rule level 12 to 14)**
* Critical = 0, Medium và Low ở các giá trị khác.

Cảnh báo High severity duy nhất này chính là sự kiện Malware Detection phát hiện file demo-malware.txt trùng với IOC hash đã cấu hình.

**Lọc cảnh báo theo mức độ rule.level**



Hình 40: Lọc cảnh báo theo mức độ

Trên Discover, khi lọc:

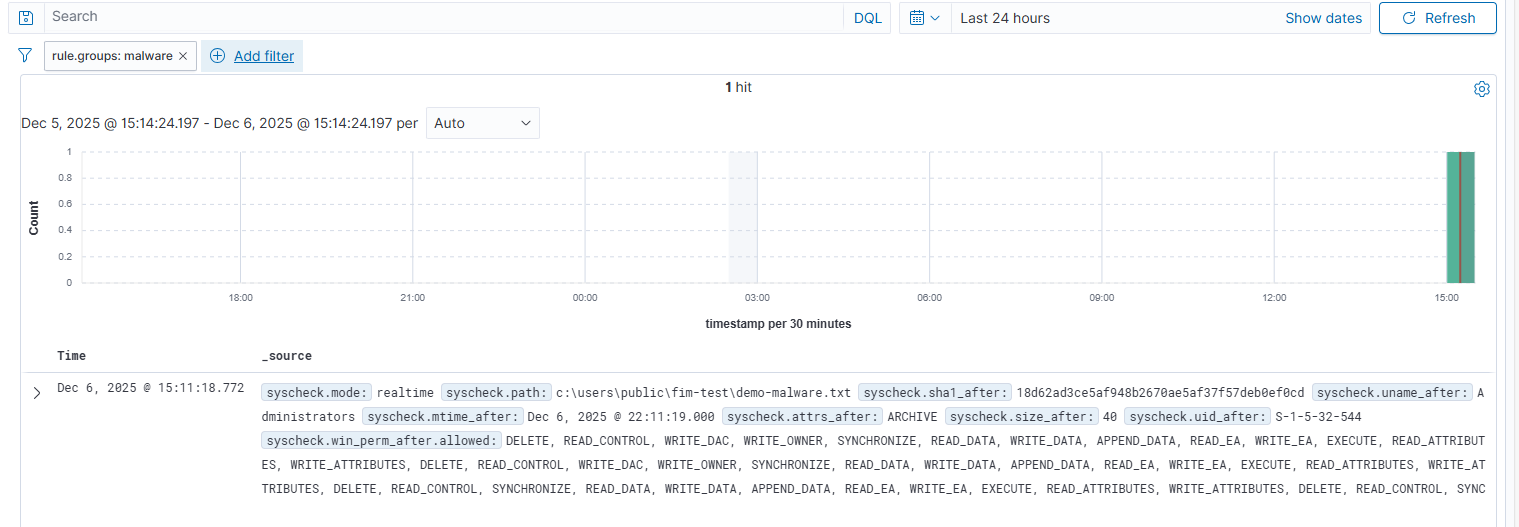
* rule.level: 12 to 14
* manager.name: b2203739-VirtualBox (tên Manager trong mô hình)

Kết quả trả về **1 hit** với các trường chính:

* syscheck.mode = realtime
* syscheck.path = c:\users\public\fim-test\demo-malware.txt
* syscheck.sha1\_after = ... (hash nội dung file sau khi chỉnh sửa)
* syscheck.uname\_after = Administrators
* syscheck.size\_after = 40 (kích thước file sau khi sửa)

Điều này chứng minh sự kiện malware được ghi nhận dưới dạng một cảnh báo FIM có mức rule.level nằm trong khoảng High severity.

**Lọc cảnh báo theo nhóm rule.groups: malware**



Hình 41: Lọc cảnh báo theo nhóm malware

Tiếp tục trên Discover, khi đặt filter:

* rule.groups: malware

Kết quả cũng trả về đúng bản ghi nói trên, với syscheck.path = c:\users\public\fim-test\demo-malware.txt. Việc bản ghi thuộc nhóm **malware** xác nhận rằng:

* Hash của file do FIM thu thập đã trùng với IOC trong malware-hashes.
* Rule phát hiện thuộc nhóm Malware Detection/Rootcheck, không phải chỉ là cảnh báo thay đổi file thông thường.

Như vậy, pipeline đầy đủ đang hoạt động:

FIM (syscheck) → phát hiện thay đổi file → gửi sự kiện lên Manager → Manager so sánh hash với danh sách IOC → rule nhóm malware kích hoạt → sinh alert mức High và hiển thị trên Overview/Discover.

### Đánh giá khả năng phát hiện, so sánh với cách tiếp cận antivirus truyền thống

Từ kết quả thực nghiệm:

* File demo-malware.txt chỉ là một file văn bản bình thường, không chứa mã thực thi, nhưng vì hash của nó được đưa vào danh sách malware-hashes, Wazuh vẫn nhận diện và sinh cảnh báo mức High.
* Điều này chứng minh hệ thống có thể **phát hiện bất kỳ file nào trùng IOC đã định nghĩa**, miễn là có sự kiện FIM liên quan tới file đó (tạo, sửa hoặc xóa).
* Kỹ thuật này phù hợp cho việc theo dõi các file, công cụ, hoặc “dấu vết” tấn công cụ thể mà đội SOC đã biết trước.

1. **So sánh ngắn gọn với antivirus truyền thống**

* **Điểm mạnh của IOC-based Malware Detection trong Wazuh**
  + Linh hoạt: quản trị viên có thể tự bổ sung IOC (hash, IP, domain) dựa trên nguồn Threat Intelligence hoặc kết quả điều tra sự cố.
  + Tích hợp: cảnh báo malware được hiển thị chung với các loại cảnh báo khác (HIDS, FIM, SCA, …) trên cùng một Dashboard SIEM.
  + Thích hợp cho môi trường SOC nơi IOC thường xuyên được cập nhật thủ công hoặc từ feed.
* **Hạn chế so với antivirus truyền thống**
  + Phụ thuộc hoàn toàn vào **danh sách IOC có sẵn**; nếu IOC không được cập nhật, hệ thống sẽ không phát hiện được mẫu mới.
  + Không thực hiện quét sâu, phân tích hành vi hoặc sandbox như nhiều sản phẩm antivirus chuyên dụng; chủ yếu dựa vào so khớp chỉ số (hash, IP, domain) với log FIM và các sự kiện khác.

## Kiểm thử tính năng Configuration Assessment

### Cấu hình rule SCA cho Windows Victim và Ubuntu server

Trong mô hình SOC, chức năng **Security Configuration Assessment (SCA)** của Wazuh được sử dụng để đánh giá mức độ tuân thủ cấu hình bảo mật của các hệ điều hành theo bộ tiêu chuẩn CIS Benchmark.

* Trên **Wazuh Manager (Ubuntu)** đã kích hoạt policy SCA cho:
  + **CIS Microsoft Windows 11** áp dụng cho agent Windows **MinhThu**.
  + Ngoài ra còn có các policy SCA cho hệ điều hành Linux trên server Ubuntu. Không đủ dữ liệu để xác minh chi tiết từng policy trên Ubuntu vì demo tập trung vào Windows.
* Đối với Windows 11, agent sẽ nhận file policy SCA tương ứng (CIS Microsoft Windows 11). Mỗi rule trong policy có:
  + **ID** (ví dụ: 26001),
  + **Title**,
  + **Command** cần chạy (ví dụ: net.exe accounts),
  + **Điều kiện kiểm tra** (chuỗi “Checks (Condition: all)” trong giao diện),
  + Kết quả **Passed/Failed** sau khi so sánh output lệnh với điều kiện.

Demo tập trung vào rule **26001 – “Ensure ‘Maximum password age’ is set to 365 or fewer days, but not 0”** trên agent Windows MinhThu.

### Quy trình chạy kiểm tra SCA và thu thập kết quả PASS/FAIL

Quy trình mô tả việc **phát hiện sẵn cấu hình yếu** và **khắc phục**, không tạo lỗi mới:

1. Chạy kiểm tra SCA và quan sát kết quả ban đầu

A screenshot of a computer

AI-generated content may be incorrect.

Hình 42: Kiểm tra SCA

* Trên Wazuh Dashboard, vào **Configuration Assessment → MinhThu → CIS Microsoft Windows 11**.
* Trong danh sách các check, rule **26001 – “Ensure ‘Maximum password age’ is set to 365 or fewer days, but not 0”** xuất hiện với trạng thái **Failed**.
* Phần “Checks (Condition: all)” cho thấy cơ chế kiểm tra: Wazuh chạy lệnh net.exe accounts, đọc giá trị dòng Maximum password age (days): X, và so sánh với điều kiện:
  + 0 < X <= 365 → Passed
  + Các giá trị khác → Failed

1. Khắc phục cấu hình trên Windows endpoint

* Trên máy Windows MinhThu, đặt lại tuổi tối đa mật khẩu về giá trị tuân thủ CIS, ví dụ 90 ngày:

net accounts /MAXPWAGE:90

NET STOP Wazuh

NET START Wazuh

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 43: Thay để cấu hình trên máy victim

* Khởi động lại dịch vụ agent Wazuh để agent chạy lại bài kiểm tra SCA và gửi kết quả mới lên Manager (tên service thực tế tùy bản cài đặt, trong demo sử dụng lệnh dạng NET STOP / NET START đối với dịch vụ Wazuh).

1. Xem lại kết quả SCA sau khi khắc phục

A screenshot of a computer

AI-generated content may be incorrect.

Hình 44: Kiểm tra sao khi thay đổi cấu hình từ victim

* Quay lại **Configuration Assessment → MinhThu → CIS Microsoft Windows 11**.
* Rule **26001** giờ hiển thị trạng thái **Passed**, chứng tỏ giá trị Maximum password age trên endpoint đã nằm trong dải 0 < X ≤ 365 đúng theo yêu cầu của CIS.

### Phân tích một số cấu hình yếu điển hình được phát hiện

Rule 26001 ban đầu được SCA phát hiện ở trạng thái Failed, chứng minh SCA có khả năng tự động nhận diện cấu hình mật khẩu không tuân thủ (mật khẩu không hết hạn hoặc hết hạn sau hơn 365 ngày).

Sau khi áp dụng lệnh net accounts /MAXPWAGE:90 và agent gửi lại kết quả, trạng thái chuyển sang Passed, cho thấy Wazuh có thể được dùng như một cơ chế xác nhận lại việc hardening cấu hình một cách khách quan.

## Phân tích hành vi tấn công với MITRE ATT&CK và Threat Hunting

### Kịch bản tấn công dò mật khẩu SMB từ Kali

Mục tiêu của phần này là phân tích một kịch bản tấn công dò mật khẩu (brute-force/password spraying) từ máy **Kali** vào máy Windows **MinhThu** qua dịch vụ **SMB**, sau đó quan sát cách Wazuh ánh xạ các cảnh báo sang **MITRE ATT&CK** và hiển thị trên **Threat Hunting** để phục vụ điều tra.

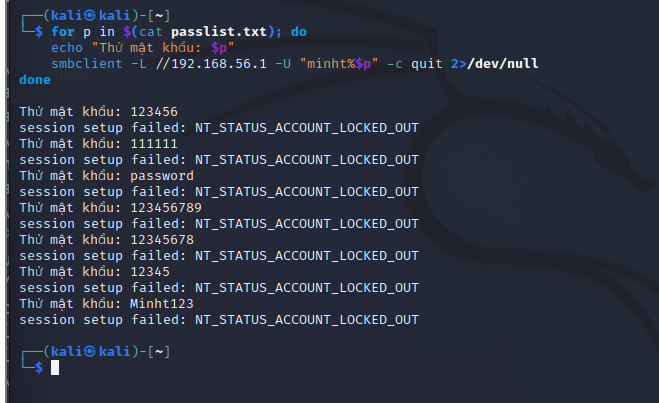
Trên máy Kali, tấn công được thực hiện bằng script shell:

for p in $(cat passlist.txt); do

echo "Thử mật khẩu: $p"

smbclient -L //192.168.56.1 -U "minht%$p" -c quit 2>/dev/null

done



Hình 45: Tấn công từ kali

Ý nghĩa:

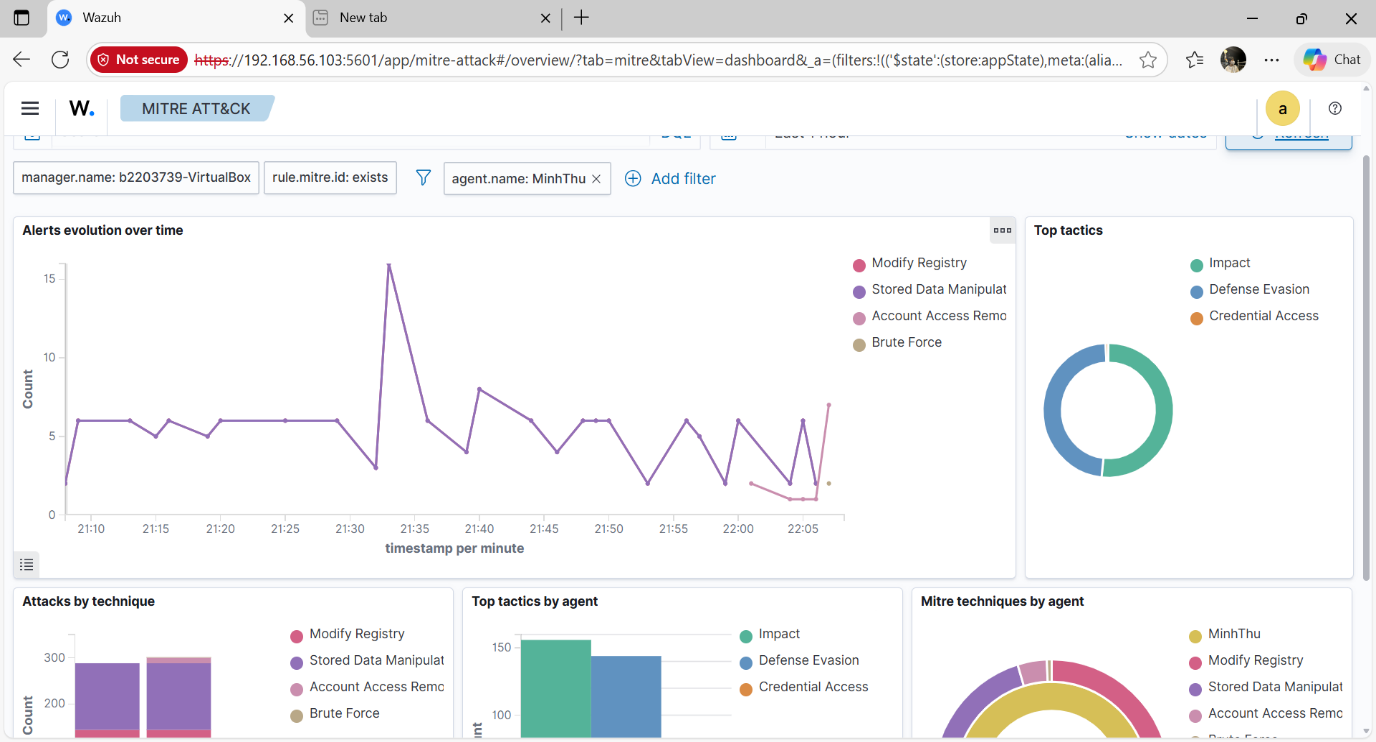
* passlist.txt chứa danh sách nhiều mật khẩu sẽ được thử lần lượt.
* smbclient -L //192.168.56.1 -U "minht%$p":
  + Kết nối tới dịch vụ SMB trên Windows victim (IP 192.168.56.1).
  + Sử dụng tài khoản minht và mật khẩu ứng viên $p.
* Mỗi vòng lặp tương ứng với **một lần thử đăng nhập** từ Kali vào tài khoản minht trên Windows.

Ảnh chụp màn hình trên Kali cho thấy các lần thử mật khẩu đều trả về thông báo:

session setup failed: NT\_STATUS\_ACCOUNT\_LOCKED\_OUT

Điều này chứng tỏ tại thời điểm đó tài khoản minht đã bị khóa (do chính sách lockout), nhưng các request đăng nhập từ Kali vẫn tiếp tục gửi tới Windows và sinh thêm log bảo mật.

### Quan sát trên MITRE ATT&CK Dashboard



Hình 46: Quan sát trên MITRE ATT&CK Dashboard

Trên Wazuh Dashboard, tab **MITRE ATT&CK → Dashboard** được lọc theo:

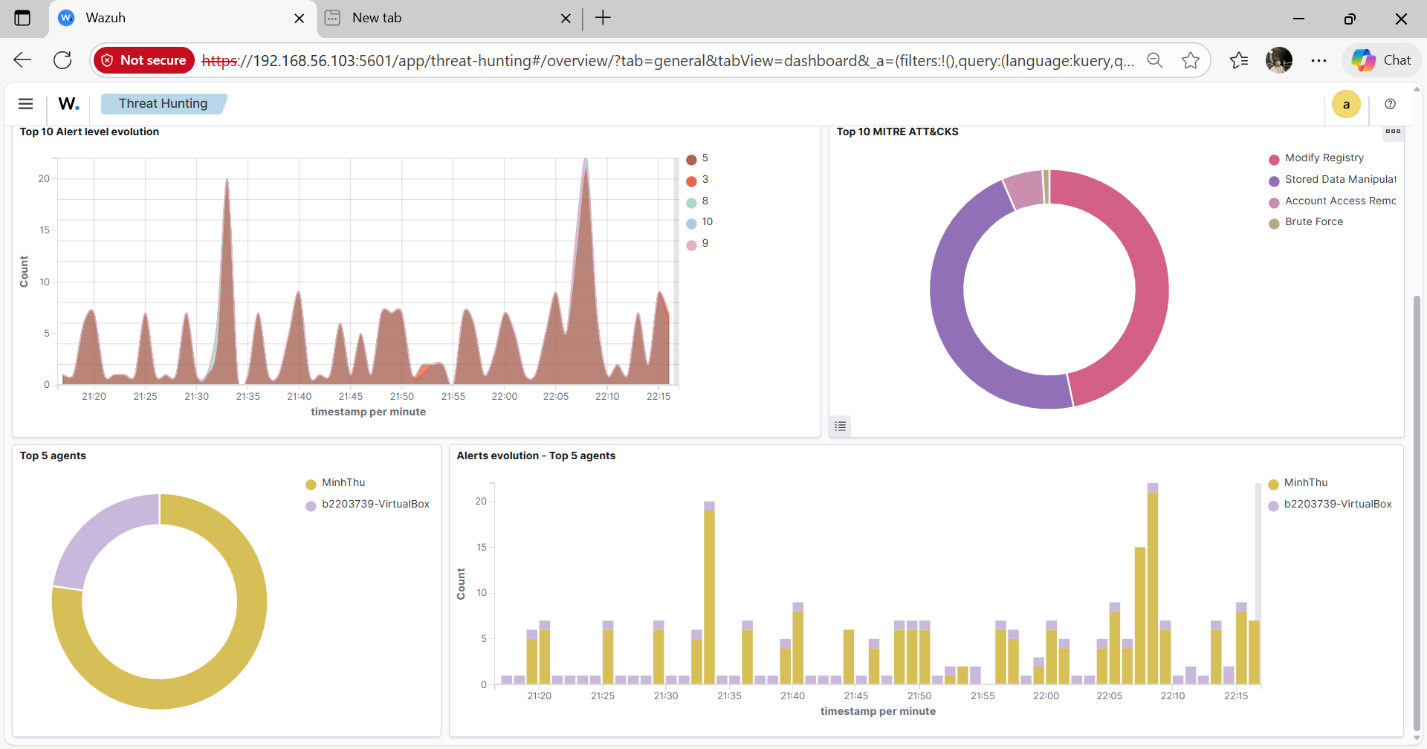
* manager.name: b2203739-VirtualBox (Wazuh Manager trên Ubuntu).
* agent.name: MinhThu (Windows victim).
* rule.mitre.id: exists (chỉ lấy các alert có ánh xạ MITRE).

Kết quả:

* Biểu đồ **“Alerts evolution over time”** xuất hiện các đỉnh cảnh báo trong khoảng thời gian tương ứng với lúc chạy script brute-force từ Kali.
* Biểu đồ **“Attacks by technique”** liệt kê các kỹ thuật MITRE, trong đó có:
  + **Modify Registry**, **Stored Data Manipulation**, **Account Access Removal**, **Brute Force**, …
  + Đối với bài tấn công dò mật khẩu, kỹ thuật liên quan trực tiếp là **Brute Force** và các kỹ thuật liên quan đến thay đổi trạng thái tài khoản (Account Access).
* Vòng tròn **“Top tactics”** hiển thị các tactic chính như:
  + **Credential Access** (liên quan đến brute-force/mật khẩu).
  + **Defense Evasion**, **Impact**, … tùy thuộc tập hợp alert sinh ra trong toàn bộ phiên.

Như vậy, MITRE ATT&CK Dashboard cho phép nhìn kịch bản tấn công không chỉ ở mức **event Windows**, mà còn ở mức **tactic/technique**: chuỗi thử mật khẩu từ Kali được Wazuh gán vào nhóm kỹ thuật “Brute Force” thuộc tactic “Credential Access”.

### Phân tích trên Threat Hunting Dashboard



Hình 47: Quan sát trên Theat Huting Dashboard

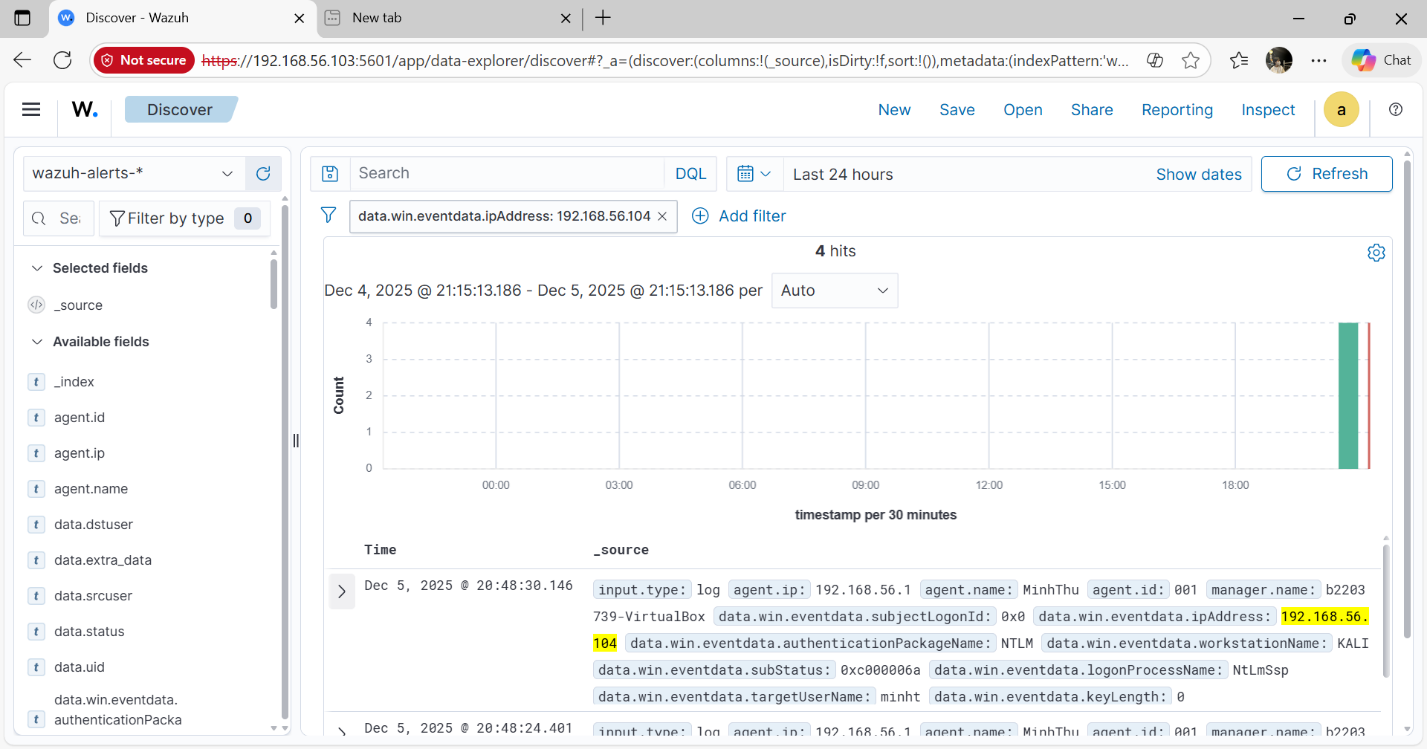
Trên tab **Threat Hunting → Overview**, trong khoảng thời gian thực hiện tấn công:

* Biểu đồ **“Top 10 alert level evolution”** xuất hiện nhiều đỉnh cảnh báo, thể hiện số lượng alert tăng mạnh trong lúc script brute-force đang chạy.
* Vòng tròn **“Top 10 MITRE ATT&CKs”** cho thấy các kỹ thuật chiếm ưu thế, trong đó có:
  + **Brute Force**,
  + **Stored Data Manipulation**,
  + **Account Access Removal**, …
* Biểu đồ **“Top 5 agents”**: phần lớn alert đến từ agent **MinhThu**, một phần nhỏ từ chính Manager (các alert nội bộ).
* Biểu đồ **“Alerts evolution – Top 5 agents”** giúp so sánh timeline cảnh báo giữa MinhThu và Manager; số lượng cảnh báo trên MinhThu chiếm tỷ trọng lớn tại thời điểm brute-force.

Từ Threat Hunting Dashboard, quản trị viên có thể nhanh chóng nhận thấy:

* Một volume lớn các cảnh báo liên quan **Credential Access / Brute Force** xuất phát từ cùng một endpoint (MinhThu) trong một khoảng thời gian ngắn.
* Đây là dấu hiệu mạnh mẽ của hành vi tấn công mật khẩu cần được điều tra chi tiết.

### Truy vết chi tiết nguồn tấn công trên Discover



Hình 48: truy vết nguồn tấn công

Để xác định chính xác nguồn tấn công, Discover được lọc theo:

* Index: wazuh-alerts-\*.
* Bộ lọc:
  + data.win.eventdata.ipAddress: 192.168.56.104 (địa chỉ IP của máy Kali trong mô hình).
  + Có thể thêm agent.name: MinhThu để chắc chắn chỉ lấy log từ Windows victim.

Kết quả:

* Các bản ghi log cho thấy:
  + agent.ip = 192.168.56.1, agent.name = MinhThu → log thuộc về Windows victim.
  + data.win.eventdata.ipAddress = 192.168.56.104 → địa chỉ IP nguồn của phiên đăng nhập là máy Kali.
  + data.win.eventdata.workstationName = KALI (theo ảnh chụp) → xác nhận hostname phía client là **KALI**.
  + data.win.eventdata.targetUserName = minht → tài khoản bị tấn công.
  + Các trường trạng thái đăng nhập (status, subStatus, authenticationPackageName = NTLM, …) phản ánh việc đăng nhập thất bại và/hoặc tài khoản bị khóa.

Từ đây có thể kết luận:

* Hệ thống không chỉ ghi nhận “nhiều lần đăng nhập thất bại” mà còn chỉ rõ **IP và hostname của máy tấn công** (Kali 192.168.56.104), cũng như tài khoản bị nhắm tới (minht).
* Các alert này đồng thời được MITRE và Threat Hunting tổng hợp thành kỹ thuật **Brute Force** dưới tactic **Credential Access**.

### Nhận xét về vai trò của MITRE ATT&CK và Threat Hunting trong demo

Từ kịch bản brute-force SMB từ Kali vào Windows MinhThu, có thể rút ra:

1. **MITRE ATT&CK**
   * Cho phép ánh xạ các alert kỹ thuật (Windows Security Log, rule Wazuh) sang tactic/technique chuẩn hóa, giúp trả lời câu hỏi:
     + “Cuộc tấn công này thuộc loại hành vi gì trong ATT&CK?”
   * Trong demo, chuỗi thử mật khẩu được phân loại vào kỹ thuật Brute Force (Credential Access), đồng thời liên quan đến các kỹ thuật thao tác account, registry và file trong toàn bộ hệ thống.
2. **Threat Hunting**
   * Cung cấp góc nhìn thời gian và phân bố: khi nào số lượng alert tăng đột biến, agent nào bị ảnh hưởng nhiều nhất, kỹ thuật MITRE nào xuất hiện nhiều nhất.
   * Hỗ trợ thợ săn mối đe dọa (analyst) phát hiện sớm chiến dịch brute-force hoặc các hoạt động nghi vấn khác mà không cần chờ tới khi có sự cố rõ ràng.
3. **Kết hợp với HIDS/FIM và SCA**
   * HIDS cung cấp event chi tiết cấp hệ điều hành (thử đăng nhập, khóa tài khoản).
   * MITRE và Threat Hunting tổng hợp các event này thành một bức tranh chiến thuật – kỹ thuật, giúp hiểu được “câu chuyện tấn công” thay vì chỉ nhìn từng log đơn lẻ.

Như vậy, demo 4.6 chứng minh rằng hệ thống SOC sử dụng Wazuh không chỉ phát hiện các sự kiện brute-force trên Windows victim mà còn cho phép phân tích ở mức chiến thuật – kỹ thuật (MITRE ATT&CK) và truy vết theo thời gian, theo tác nhân (Threat Hunting), đáp ứng yêu cầu phân tích hành vi tấn công trong một SOC thực tế.

## Tổng hợp kết quả kiểm thử

### Bảng tổng hợp kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chức năng kiểm thử** | **Kịch bản / Hành vi mô phỏng** | **Bằng chứng chính trên Wazuh** | **Kết luận** |
| 1 | HIDS – Giám sát đăng nhập sai (EventID 4625) | Người dùng nhập sai mật khẩu nhiều lần trên Windows MinhThu (local logon) | Discover với filter agent.name: MinhThu, eventID: 4625, các trường targetUserName, ipAddress = 127.0.0.1 | HIDS + SIEM thu thập và hiển thị chính xác chuỗi đăng nhập thất bại trên endpoint. |
| 2 | HIDS + MITRE/Threat Hunting – Brute-force SMB từ Kali | Kali chạy vòng lặp smbclient với danh sách mật khẩu vào tài khoản minht trên 192.168.56.1 | Discover hiển thị data.win.eventdata.ipAddress = 192.168.56.104, workstationName = KALI; MITRE/Threat Hunting gán kỹ thuật **Brute Force** (Credential Access) | Hệ thống nhận diện được nguồn tấn công (IP/hostname Kali) và ánh xạ vào kỹ thuật MITRE tương ứng. |
| 3 | FIM – File Integrity Monitoring | Tạo/sửa/xóa file demo.txt trong thư mục C:\Users\Public\FIM-Test | Discover với syscheck.path = ...\demo.txt, syscheck.event = added/modified/deleted; dashboard FIM hiển thị thống kê theo action | FIM theo dõi đầy đủ ba loại thay đổi file (add/modify/delete) trên thư mục được cấu hình. |
| 4 | Malware Detection dựa trên IOC hash | Tạo file demo-malware.txt, tính SHA256, thêm hash vào malware-hashes, sau đó sửa file để kích FIM | Discover với rule.groups: malware, syscheck.path = ...\demo-malware.txt; Overview hiển thị 1 alert High severity (rule.level 12–14) | Khi hash file trùng IOC, hệ thống sinh alert malware mức High, chứng minh cơ chế IOC hoạt động. |
| 5 | Configuration Assessment (SCA) – CIS Win11 | Endpoint MinhThu có sẵn cấu hình Maximum password age không đúng chuẩn; chỉnh lại net accounts /MAXPWAGE:90 | Trong CIS Windows 11: check **26001** ban đầu **Failed**, sau khi cấu hình lại chuyển sang **Passed** | SCA phát hiện được cấu hình yếu và xác nhận lại trạng thái tuân thủ sau khi hardening. |
| 6 | MITRE ATT&CK + Threat Hunting (tổng hợp) | Phân tích toàn bộ alert có rule.mitre.id trên agent MinhThu | MITRE dashboard hiển thị các tactic như Credential Access, Defense Evasion, Impact; Threat Hunting thể hiện timeline alert theo MITRE | Hệ thống cung cấp góc nhìn chiến thuật – kỹ thuật và dòng thời gian alert phục vụ phân tích hành vi. |

Bảng 5: tổng hợp kết quả kiểm thử

### Đánh giá ưu điểm nổi bật của hệ thống

1. **Chuỗi thu thập – phân tích – lưu trữ – hiển thị hoạt động ổn định**

* Tất cả kịch bản (HIDS, FIM, IOC, SCA, brute-force từ Kali) đều cho kết quả trên Dashboard đúng với hành vi thực hiện trên endpoint.
* Điều này cho thấy pipeline **Windows/Kali → Agent → Manager → Indexer → Dashboard** đang vận hành đầy đủ.

1. **Bao phủ nhiều lớp chức năng an ninh trên cùng nền tảng**

* Hệ thống hiện giám sát đồng thời:
  + HIDS (Windows Security Log, brute-force SMB),
  + FIM (file + registry),
  + Malware Detection dựa trên IOC,
  + Configuration Assessment (SCA) theo CIS.
* Mỗi module được chứng minh bằng ít nhất một demo cụ thể, có log và ảnh chụp giao diện kèm theo.

1. **Khả năng ánh xạ MITRE ATT&CK và hỗ trợ Threat Hunting**

* Các alert liên quan brute-force và thao tác tài khoản được gán vào kỹ thuật MITRE tương ứng, giúp phân loại theo tactic **Credential Access**, **Defense Evasion**, **Impact**.
* Threat Hunting dashboard cung cấp timeline và phân bố alert theo agent/kỹ thuật, hỗ trợ điều tra sự kiện trên toàn hệ thống.

1. **Khả năng đánh giá và cải thiện cấu hình bảo mật**

* Module SCA không chỉ báo cáo trạng thái hiện tại (Passed/Failed) mà còn phản ánh rõ tác động của việc hardening (ví dụ check 26001 từ Failed → Passed sau khi chỉnh MAXPWAGE).
* Điều này cho phép sử dụng Wazuh như công cụ kiểm tra lại cấu hình sau khi triển khai biện pháp bảo mật.

1. **Tính minh bạch và dễ kiểm chứng**

* Mỗi demo đều gắn với lệnh cụ thể trên Windows/Kali và có log tương ứng trên Wazuh, người đọc có thể tái hiện lại trong môi trường lab tương tự để kiểm chứng.

### Thảo luận về khả năng ứng dụng thực tế

1. **Ứng dụng trong môi trường lab và đào tạo SOC**

* Mô hình hiện tại (01 Manager Ubuntu + 01 agent Windows + 01 máy Kali tấn công) đã đủ để minh họa chu trình:
  + Sinh sự kiện an ninh thực tế,
  + Thu thập – phân tích – cảnh báo trên Wazuh,
  + Mapping MITRE và thực hiện threat hunting cơ bản,
  + Đánh giá cấu hình theo CIS và hardening.
* Cấu trúc và các bước demo có thể dùng làm mẫu cho bài thực hành SOC/Wazuh trong môi trường học thuật hoặc đào tạo nội bộ.

1. **Khả năng mở rộng quy mô nhỏ – vừa**

* Kiến trúc agent–manager cho phép gắn thêm nhiều Windows endpoint khác, áp dụng cùng policy HIDS/FIM/IOC/SCA mà không thay đổi thiết kế lõi.
* Với việc đã kiểm chứng hoạt động trên một agent cụ thể, bước kế tiếp hợp lý là nhân rộng cấu hình này cho nhóm máy cùng loại trong mạng nội bộ (ví dụ phòng thực hành, nhóm máy nhân viên) để giám sát tập trung.
* Không đủ dữ liệu để đánh giá hiệu quả trên môi trường doanh nghiệp lớn (hàng trăm–hàng nghìn agent, nhiều vùng mạng, yêu cầu HA). Nội dung báo cáo chỉ chứng minh tính khả thi ở quy mô lab và tiềm năng mở rộng dần.

1. **Kết hợp với các công cụ/giải pháp khác**

* Trong môi trường thực tế, Wazuh có thể đóng vai trò **nền tảng giám sát endpoint và cấu hình**, trong khi các lớp khác như firewall, IDS mạng, antivirus, backup… tiếp tục đảm nhiệm chức năng chuyên biệt.
* Các kịch bản đã thực hiện (brute-force, thay đổi file, IOC hash, cấu hình mật khẩu yếu) là các tình huống phổ biến mà nhiều tổ chức cần theo dõi, do đó hoàn toàn có thể được chuyển từ lab sang hệ thống thật khi có hạ tầng và quy trình vận hành phù hợp.

1. **Giới hạn hiện tại của hệ thống demo**

* Hệ thống mới chỉ giám sát một endpoint Windows và chưa thu thập log từ thiết bị mạng, server ứng dụng hay dịch vụ cloud.
* Chưa có thử nghiệm về hiệu năng, độ trễ cảnh báo, khả năng chịu tải khi số lượng alert tăng rất lớn.
* Do vậy, báo cáo chỉ khẳng định được **tính đúng đắn chức năng** (functional correctness) của các module đã kiểm thử, không đưa ra kết luận về **hiệu năng** hay **khả năng chịu tải** ở quy mô sản xuất.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

Qua quá trình nghiên cứu và triển khai, đề tài đã đạt được một số kết quả chính sau:

* Xây dựng được mô hình SOC thu nhỏ dựa trên nền tảng Wazuh Stack, bao gồm Wazuh Manager, Wazuh Indexer và Wazuh Dashboard cài đặt trên máy chủ Ubuntu, cùng Wazuh Agent trên máy Windows endpoint.
* Cấu hình thành công kênh thu thập và phân tích log dạng SIEM cho môi trường lab: thu thập Windows Event Log (đặc biệt là sự kiện đăng nhập – ví dụ 4625), log hệ điều hành và các sự kiện bảo mật quan trọng, đồng thời hiển thị, truy vấn và thống kê trên Dashboard.
* Triển khai và kiểm thử thực tế các chức năng an ninh chính của Wazuh ở tầng endpoint:
  + HIDS (Host-based IDS): phát hiện các hành vi đăng nhập bất thường, brute-force, một số hành vi nghi vấn trên hệ thống Windows.
  + FIM (File Integrity Monitoring): giám sát thay đổi tệp tin và thư mục trên Windows, đặc biệt tại thư mục thử nghiệm FIM-Test, ghi nhận chi tiết các hành động tạo/sửa/xóa file.
  + Malware Detection dựa trên IOC/Rootcheck: cấu hình danh sách IOC và sử dụng cơ chế kiểm tra rootcheck để phát hiện file có hash trùng IOC hoặc hành vi hệ thống đáng ngờ.
* Ứng dụng module SCA (Security Configuration Assessment) ở mức cơ bản để đánh giá cấu hình bảo mật của hệ thống theo các policy chuẩn, từ đó nhận diện một số thiết lập cấu hình yếu.
* Tích hợp và minh họa được việc ánh xạ cảnh báo Wazuh với khung MITRE ATT&CK trên Dashboard, giúp thể hiện rõ hơn bối cảnh chiến thuật/kỹ thuật trong các kịch bản demo.
* Xây dựng được mô hình phòng thủ nhiều lớp (Defense in Depth) phù hợp với hệ thống SOC Wazuh: từ lớp Network & Perimeter (NAT, firewall host-based), Endpoint Security (Agent – HIDS, FIM, Malware), Hardening (SCA), đến lớp Threat Intelligence & Context và lớp SIEM & Correlation tại Wazuh Manager/Indexer/Dashboard.
* Hoàn thiện hệ thống sơ đồ kiến trúc, luồng log/alert, sơ đồ các lớp phòng thủ và mô tả chi tiết kịch bản demo, hỗ trợ cho báo cáo và thuyết minh luận văn.

## Hạn chế

### Phạm vi mô hình và môi trường lab

 Mô hình SOC Wazuh được triển khai trong môi trường lab ảo hóa, với quy mô nhỏ: một máy chủ Wazuh trên Ubuntu và một endpoint Windows là chính. Không đủ dữ liệu để xác minh việc triển khai trên số lượng lớn agent hoặc nhiều phân đoạn mạng, do đó chưa phản ánh được đầy đủ các vấn đề về hiệu năng, khả năng mở rộng và độ phức tạp vận hành khi áp dụng ở quy mô doanh nghiệp.

 Hệ thống vận hành trong môi trường NAT, không phơi bày trực tiếp ra Internet và không tích hợp với hạ tầng mạng phức tạp (nhiều VLAN, DMZ, nhiều vùng bảo mật). Vì vậy, các kịch bản liên quan đến tấn công qua nhiều hop hoặc di chuyển ngang trên nhiều subnet chưa được hiện thực.

 Chưa kết nối log từ các thiết bị mạng/chuyên dụng (firewall phần cứng, router doanh nghiệp, IDS/IPS mạng, WAF,…), nên phạm vi giám sát chủ yếu dừng lại ở tầng host và hệ điều hành.

### Hạn chế về thiết bị, công cụ và nguồn lực

 Hệ thống được triển khai trên nền tảng phần cứng phổ thông của sinh viên, tài nguyên CPU/RAM/lưu trữ có hạn, nên không thể mô phỏng được tải log lớn, số lượng agent đông hoặc kịch bản tấn công kéo dài, liên tục như môi trường sản xuất.

 Công cụ được sử dụng tập trung vào Wazuh Stack; chưa có điều kiện tích hợp hoặc so sánh với các giải pháp thương mại khác (SIEM/EDR/AV trả phí), do đó phần đánh giá hiệu năng và so sánh giải pháp còn hạn chế.

 Thời gian và nguồn lực cá nhân có hạn, nên việc tinh chỉnh rules, tối ưu hiệu năng, thử nghiệm đa nền tảng (nhiều phiên bản Windows, Linux server khác nhau) mới được thực hiện ở mức tối thiểu đủ phục vụ mục tiêu demo.

### Hạn chế về kịch bản tấn công và nguồn log

 Các kịch bản tấn công chủ yếu tập trung vào một số nhóm tình huống điển hình: đăng nhập sai (4625), thay đổi file trong thư mục được giám sát, tạo file có IOC đã định trước,… Các chuỗi tấn công phức tạp hơn (lateral movement, privilege escalation nhiều bước, persistence bằng nhiều kỹ thuật) chưa được mô phỏng đầy đủ.

 Nguồn log đưa vào Wazuh còn hạn chế: chủ yếu là log hệ điều hành Windows, log của chính máy chủ Wazuh và các sự kiện FIM/Malware/SCA. Chưa thu thập log từ dịch vụ ứng dụng (web server, database server, AD, mail server, VPN,…) nên phạm vi quan sát của SOC còn hẹp so với thực tế doanh nghiệp.

 IOC và nguồn Threat Intelligence sử dụng trong demo ở mức cơ bản, chủ yếu là danh sách hash/IP/domain tĩnh được cấu hình thủ công; chưa tích hợp với các nền tảng Threat Intelligence ngoài (MISP, OTX,…) nên tính cập nhật và độ phong phú của IOC còn hạn chế.

## Hướng phát triển

Các hướng phát triển tiếp theo cho hệ thống SOC Wazuh có thể bao gồm:

* Mở rộng quy mô mô hình: triển khai thêm nhiều agent trên các hệ điều hành khác (nhiều máy Windows, Linux server, máy chủ dịch vụ), bổ sung nhiều subnet và phân đoạn mạng để đánh giá khả năng mở rộng, hiệu năng và độ ổn định.
* Bổ sung nguồn log đa dạng: tích hợp log từ firewall/router, web server, database, Active Directory, VPN, ứng dụng nội bộ,… để mô hình SOC gần với môi trường doanh nghiệp hơn, đồng thời thử nghiệm tương quan sự kiện đa nguồn.
* Hoàn thiện và kích hoạt đầy đủ các module Wazuh khác (đặc biệt là Vulnerability Detector) với cấu hình provider phù hợp, nhằm bổ sung thêm một lớp phát hiện lỗ hổng và đối chiếu với kết quả SCA.
* Xây dựng thêm các kịch bản tấn công nâng cao:
* Chuỗi tấn công nhiều bước bám theo MITRE ATT&CK (từ initial access đến impact).
* Kịch bản di chuyển ngang (lateral movement) giữa nhiều máy, tấn công dịch vụ cụ thể (web, database, AD).
* Tăng cường sử dụng Threat Intelligence: tự động cập nhật IOC từ các nguồn mở, tích hợp với Wazuh để nâng cao khả năng phát hiện, đồng thời kết hợp với module MITRE ATT&CK để làm giàu ngữ cảnh cảnh báo.
* Nghiên cứu bước đầu tích hợp tự động hóa ứng phó (SOAR ở mức cơ bản), ví dụ: tự động khóa tài khoản khi phát hiện brute-force lặp lại, tự động cô lập host nghi nhiễm, hoặc gửi thông báo đến kênh liên lạc của nhóm vận hành.
* Chuẩn hóa rule set, dashboard và quy trình vận hành (runbook/playbook) để có thể tái sử dụng mô hình SOC Wazuh trong các bài lab nâng cao và trong các môi trường thực tế nhỏ–trung bình.

# TÀI LIỆU THAM KHẢO

## **A. Nguồn số liệu thống kê (Statistical Reports)**

**[1]** Verizon, 2024 Data Breach Investigations Report (DBIR), Verizon Enterprise, 2024.  
(Nguồn cho: 30.458 sự cố, 10.626 vi phạm dữ liệu, 68% human factor, 15% supply-chain).  
Link: [https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf](https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf?utm_source=chatgpt.com)

**[2]** IBM Security and Ponemon Institute, Cost of a Data Breach Report 2023, IBM, 2023.  
Link: <https://www.ibm.com/reports/data-breach>

**[3]** Palo Alto Networks – Unit 42, 2024 Ransomware Threat Report, 2024.  
Link: <https://www.paloaltonetworks.com/unit42>

**[4]** Secureframe, Cybersecurity Statistics 2024, 2024.  
(Nguồn cho: 1.900 cuộc tấn công/tuần).  
Link: <https://secureframe.com/resources/cybersecurity-statistics>

**[5]** Cục An toàn thông tin – Bộ Thông tin và Truyền thông, Bản tin An toàn thông tin Việt Nam 2024, Hà Nội, 2024.  
(Nguồn cho số liệu Việt Nam).  
Nguồn: <https://mic.gov.vn> – <https://attt.vn> – <https://ncsc.gov.vn>

**[6]** BKAV Corporation, Báo cáo An ninh mạng Việt Nam 2024, Hà Nội, 2024.  
(Nguồn cho 155.600 ransomware; 90.000 lỗ hổng).  
Link: <https://www.bkav.com.vn>

# **B. Nguồn tài liệu học thuật (Academic References)**

**[7]** A. K. Sahu et al., Security Operations Center: A Systematic Study and Open Challenges, IEEE Access, 2021.  
**[8]** E. Casey et al., Security Operation Center: Concepts, Implementation, and Best Practices, 2019.  
**[9]** S. Subramanian et al., Integrated Network and Security Operation Center: A Systematic Analysis, IEEE Access, 2022.

# **C. Nguồn về kỹ thuật bảo mật (HIDS, FIM, Malware, Vulnerability)**

**[10]** Wazuh Documentation, Host-based Intrusion Detection System (HIDS), Wazuh 4.8.  
Link: <https://documentation.wazuh.com>

**[11]** Wazuh Documentation, File Integrity Monitoring (FIM), Wazuh 4.8.

**[12]** Wazuh Documentation, Malware Detection / Rootcheck Module, Wazuh 4.8.

**[13]** Introdution to Azure security

Link: <https://learn.microsoft.com/en-us/azure/security/fundamentals/overview>

**[14]** Wazuh Documentation, Security Configuration Assessment (SCA), Wazuh 4.8.

**[15]** Malware Detection Approaches.

**[16]** Malware.pdf

# **D. Nguồn về MITRE ATT&CK và Incident Response**

**[17]** MITRE Corporation, MITRE ATT&CK Framework, 2024.  
Link: <https://attack.mitre.org>

**[18]** NIST SP 800-61r2, Computer Security Incident Handling Guide, 2012.

Link: <https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final>

**[19]** NIST Cybersecurity Framework (NIST CSF) 1.1, 2018.

# **E. Nguồn về SOC, SIEM, Defense-in-Depth, Threat Intelligence**

**[20]** ENISA, Threat Landscape Report 2023/2024.

**[21]** SANS Institute, Building a Modern SOC, 2020.

**[22]** Microsoft, Zero Trust and Defense-in-Depth Security Model, 2024.