

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



CHUYÊN ĐỀ KỸ NGHỆ AN TOÀN MẠNG
NGHIÊN CỨU GIẢI PHÁP SIEM && XDR VỚI WAZUH

Sinh viên thực hiện:

Nguyễn Ngọc Anh – MSSV: AT180304

Người hướng dẫn:

TS. Hoàng Đức Thọ

Khoa An toàn thông tin – Học viện Kỹ thuật mật mã

Hà Nội, 2024

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



CHUYÊN ĐỀ KỸ NGHỆ AN TOÀN MẠNG
NGHIÊN CỨU GIẢI PHÁP SIEM && XDR VỚI WAZUH

Sinh viên thực hiện:

Nguyễn Ngọc Anh – MSSV: AT180304

Người hướng dẫn:

TS. Hoàng Đức Thọ

Khoa An toàn thông tin – Học viện Kỹ thuật mật mã

Hà Nội, 2024

MỤC LỤC

MỤC LỤC	1
DANH MỤC CÁC CHỮ VIẾT TẮT	2
DANH MỤC CÁC HÌNH VẼ	3
TÓM TẮT NỘI DUNG	4
LỜI NÓI ĐẦU	5
CHƯƠNG 1: GIỚI THIỆU	6
1.1 Tổng quan về SIEM và XDR	6
1.1.1 <i>SIEM (Security Information and Event Management)</i>	6
1.1.2 <i>XDR (Extended Detection and Response)</i>	7
1.1.3 So sánh giữa SIEM và XDR	9
1.2 Vai trò của Wazuh trong lĩnh vực bảo mật	9
1.2.1 Quản lý sự kiện bảo mật (SIEM)	10
1.2.2 Phát hiện xâm nhập (HIDS/NIDS)	11
1.2.3 Giám sát bảo mật liên tục	11
1.2.4 Tích hợp với Elastic Stack để phân tích dữ liệu bảo mật	11
1.2.5 Đáp ứng các yêu cầu tuân thủ bảo mật	12
1.2.6 Phát hiện và phản ứng mở rộng (XDR)	15
1.3 So sánh giữa các giải pháp SIEM/XDR khác và Wazuh	16
CHƯƠNG 2: TRIỂN KHAI THỰC NGHIỆM	18
2.1 Hướng đối tượng doanh nghiệp	18
2.2 Yêu cầu kỹ thuật	18
2.3 Triển khai Wazuh Server và Agents	19
2.3.1 Cài đặt Wazuh	20
2.3.2 Triển khai giám sát các Agent	30
2.4 Cấu hình Wazuh phát hiện cuộc tấn công Brute-Force	33
2.5 Cấu hình Wazuh phát hiện các cuộc tấn công SQL Injection	36
2.6 Cấu hình Wazuh chặn địa chỉ IP độc hại truy cập đến Web Server	40
2.7 Tích hợp VirusTotal để phát hiện và xóa các phần mềm độc hại	52
CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	79
3.1 Tóm tắt kết quả đạt được	79
3.2 Những khó khăn và thách thức trong quá trình triển khai	79
3.3 Hướng phát triển và cải thiện trong tương lai	80
TÀI LIỆU THAM KHẢO	82

DANH MỤC CÁC CHỮ VIẾT TẮT

AD	Active Directory
API	Application Programming Interface
AWS	Amazon Web Services
FIM	File Integrity Monitoring
GDPR	General Data Protection Regulation
GRC	Governance, risk management and compliance
HIDS	Host-based Intrusion Detection System
IAM	Identity Access Management
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
MFA	Multi-Factor Authentication
NIDS	Network Intrusion Detection System
OAuth	Open standard for Authorization
PCI-DSS	Payment Card Industry Data Security Standard
RBAC	Role-Based Access Control
RDP	Remote Desktop Protocol
SAML	Security Assertion Markup Language
SIEM	Security Information and Event Management
SIM	Security Information Management
SOAR	Security Orchestration, Automation and Response
SSH	Secure Shell
SSO	Single Sign-On
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security/ Secure Sockets Layer
UBA	User Behavior Analytics
XDR	Extended Detection and Response

DANH MỤC CÁC HÌNH VẼ

Hình 1: Các thành phần và khả năng của SIEM	7
Hình 2: Lớp bảo mật XDR	8
Hình 3: Sự khác nhau giữa SIEM và XDR	9
Hình 4: Giao diện web của Wazuh.....	10
Hình 5: Nâng cao phân tích bảo mật tích hợp Wazuh với Elastic Stack	12
Hình 6: Tiêu chuẩn GDPR bảo vệ dữ liệu cá nhân	13
Hình 7: Các đối tượng cần tuân thủ HIPAA	14
Hình 8: Các mục tiêu giám sát PCI-DSS	14
Hình 9: So sánh giữa các giải pháp SIEM/XDR khác và Wazuh	16
Hình 10: Mô hình triển khai thực nghiệm.....	20

TÓM TẮT NỘI DUNG

Đề tài "**Nghiên cứu giải pháp SIEM && XDR với Wazuh**" trình bày quá trình nghiên cứu, triển khai và đánh giá hệ thống bảo mật dựa trên nền tảng mã nguồn mở Wazuh, với mục tiêu xây dựng một giải pháp **SIEM** (Security Information and Event Management) và **XDR** (Extended Detection and Response) tối ưu, tiết kiệm chi phí nhưng vẫn đảm bảo tính hiệu quả trong giám sát và bảo vệ hệ thống mạng.

Nội dung báo cáo bao gồm các phần chính như sau:

1. *Tổng quan về SIEM và XDR*: Giới thiệu khái niệm, vai trò và tầm quan trọng của các giải pháp SIEM và XDR trong an ninh mạng. Báo cáo cũng so sánh các hệ thống SIEM/XDR khác và làm rõ ưu điểm của Wazuh.

2. *Giới thiệu về Wazuh*: Trình bày chi tiết về nền tảng Wazuh, một giải pháp mã nguồn mở tích hợp nhiều tính năng quan trọng như quản lý sự kiện bảo mật, phát hiện các mối đe dọa và khả năng phản ứng tự động.

3. *Yêu cầu kỹ thuật và triển khai hệ thống*: Phân tích các yêu cầu về kỹ thuật, bảo mật, hệ thống và tích hợp cần thiết để triển khai Wazuh trong môi trường doanh nghiệp. Các bước triển khai Wazuh Server và Agents, cũng như cấu hình các chính sách bảo mật, được mô tả chi tiết trong phần này.

4. *Triển khai thực nghiệm*: Mô tả quá trình triển khai thực tế hệ thống Wazuh, bao gồm những khó khăn và thách thức đã gặp phải, từ việc tích hợp hệ thống đến tối ưu hóa khả năng phân tích log và quản lý mối đe dọa.

5. *Kết quả đạt được và hướng phát triển*: Tổng kết các kết quả đạt được sau khi triển khai hệ thống, đánh giá hiệu quả trong việc giám sát, phát hiện và xử lý các mối đe dọa an ninh mạng. Phần này cũng đề xuất các hướng phát triển và cải thiện hệ thống trong tương lai, bao gồm tối ưu hóa hiệu suất, tích hợp thêm các giải pháp bảo mật khác, và cải thiện khả năng phản ứng tự động.

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, an ninh mạng đã trở thành một yếu tố vô cùng quan trọng đối với mọi tổ chức, từ các doanh nghiệp đến các cơ quan chính phủ. Hệ thống mạng của các tổ chức ngày càng phức tạp, đi kèm với sự gia tăng của các cuộc tấn công mạng, đòi hỏi các giải pháp bảo mật toàn diện và hiệu quả hơn để giám sát, phát hiện và phản ứng nhanh chóng với các mối đe dọa.

Với sự xuất hiện của các công nghệ như **SIEM** (Security Information and Event Management) và **XDR** (Extended Detection and Response), các doanh nghiệp có thể theo dõi và quản lý các sự kiện an ninh một cách tập trung, phát hiện sớm những bất thường và bảo vệ tài sản thông tin một cách chủ động. Tuy nhiên, chi phí triển khai các giải pháp thương mại thường khá cao, đòi hỏi đầu tư lớn vào hạ tầng và giấy phép phần mềm.

Trong bối cảnh đó, mã nguồn mở đã trở thành một lựa chọn hấp dẫn nhờ tính linh hoạt và tiết kiệm chi phí. Wazuh – một nền tảng mã nguồn mở nổi bật trong lĩnh vực giám sát an ninh, kết hợp giữa khả năng thu thập, phân tích log và quản lý các mối đe dọa, đã được nhiều tổ chức tin dùng. Đề tài "**Nghiên cứu giải pháp SIEM && XDR với Wazuh**" không chỉ nhằm nghiên cứu và triển khai Wazuh như một công cụ bảo mật mạnh mẽ, mà còn giúp xây dựng một mô hình bảo mật tối ưu dựa trên nhu cầu thực tế của tổ chức, từ đó nâng cao khả năng phát hiện và xử lý sự cố an ninh mạng.

CHƯƠNG 1: GIỚI THIỆU

1.1 Tổng quan về SIEM và XDR

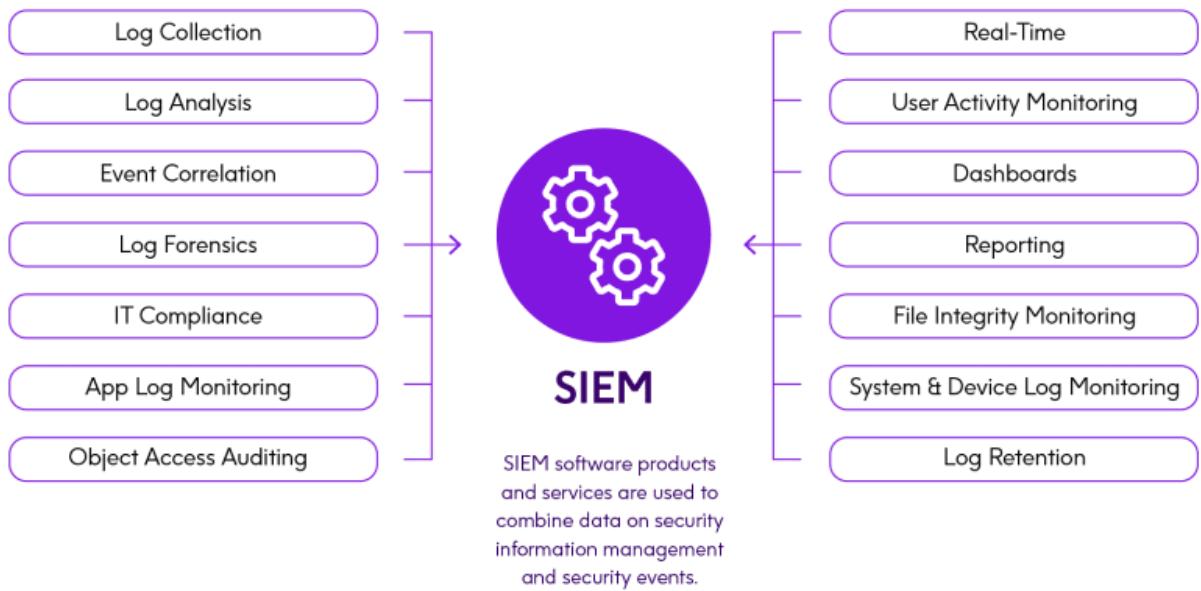
SIEM (Security Information and Event Management) và *XDR (Extended Detection and Response)* là hai giải pháp bảo mật quan trọng trong lĩnh vực quản lý và phát hiện mối đe dọa an ninh mạng. Cả hai đều có mục tiêu chính là giúp các tổ chức phát hiện, phân tích và phản ứng nhanh chóng với các sự cố bảo mật, nhưng cách thức tiếp cận và công nghệ sử dụng lại khác nhau.

1.1.1 SIEM (Security Information and Event Management)

SIEM là một hệ thống quản lý thông tin và sự kiện bảo mật, kết hợp hai yếu tố chính:

- *SIM (Security Information Management)*: Thu thập, lưu trữ và phân tích dữ liệu nhật ký từ nhiều nguồn khác nhau, bao gồm máy chủ, thiết bị mạng, và ứng dụng. Từ đó, nó giúp tổ chức duy trì nhật ký hoạt động và tuân thủ các quy định bảo mật.

- *SEM (Security Event Management)*: Tập trung vào việc phân tích các sự kiện bảo mật thời gian thực, nhằm phát hiện các hành vi bất thường hoặc nguy cơ bảo mật.



Hình 1: Các thành phần và khả năng của SIEM

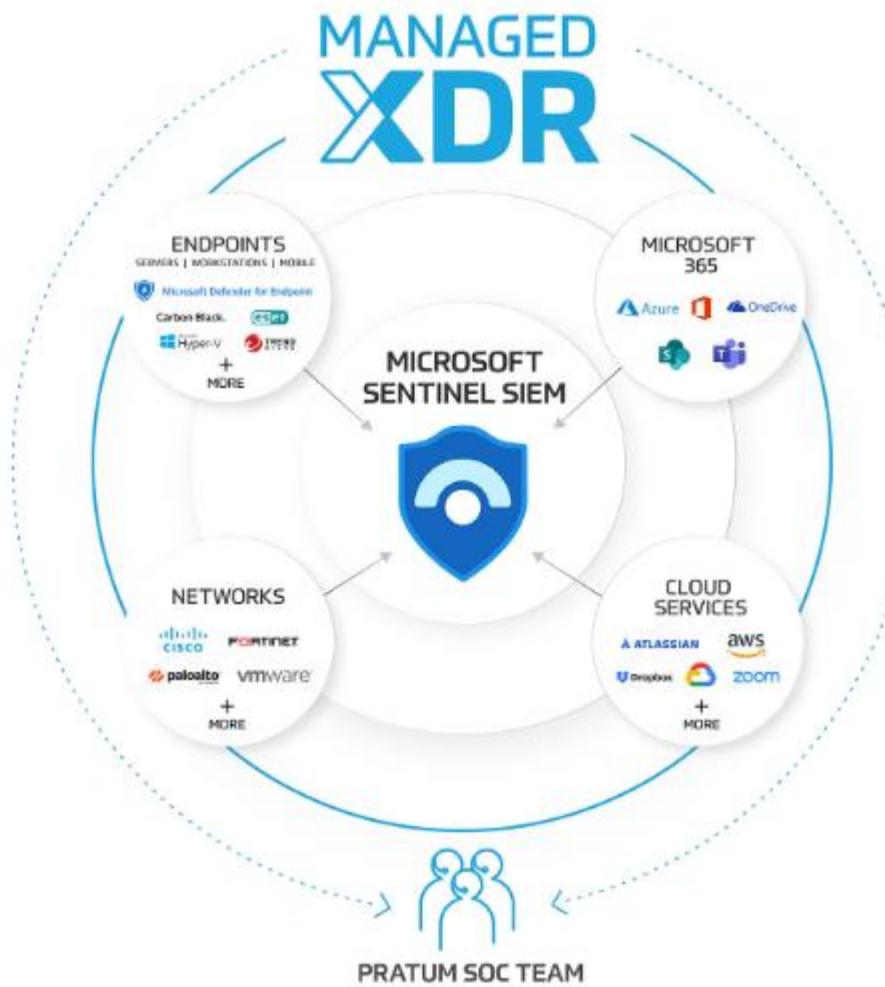
SIEM hoạt động dựa trên việc thu thập dữ liệu từ nhiều nguồn khác nhau trong hệ thống và sau đó phân tích, so sánh dữ liệu để phát hiện các mối đe dọa hoặc sự kiện bất thường. Các tính năng chính của SIEM bao gồm:

- *Thu thập và tập trung dữ liệu* từ nhiều hệ thống khác nhau.
- *Phân tích nhật ký* để tìm kiếm các hành vi bất thường.
- *Phát hiện mối đe dọa* dựa trên các luật và quy tắc đã được thiết lập.
- *Báo cáo và cảnh báo* khi có sự cố bảo mật xảy ra.
- *Quản lý tuân thủ* giúp đảm bảo tuân thủ các tiêu chuẩn và quy định bảo mật như GDPR, HIPAA, PCI-DSS.

1.1.2 XDR (Extended Detection and Response)

XDR là một giải pháp bảo mật hiện đại và mở rộng hơn so với SIEM. Trong khi SIEM chủ yếu tập trung vào việc thu thập và phân tích nhật ký, XDR cung cấp khả năng phản hồi và quản lý mối đe dọa toàn diện trên nhiều lớp bảo mật khác nhau, bao gồm:

- *Endpoint (Thiết bị đầu cuối)*
- *Network (Mạng)*
- *Email*
- *Cloud*
- *Ứng dụng*

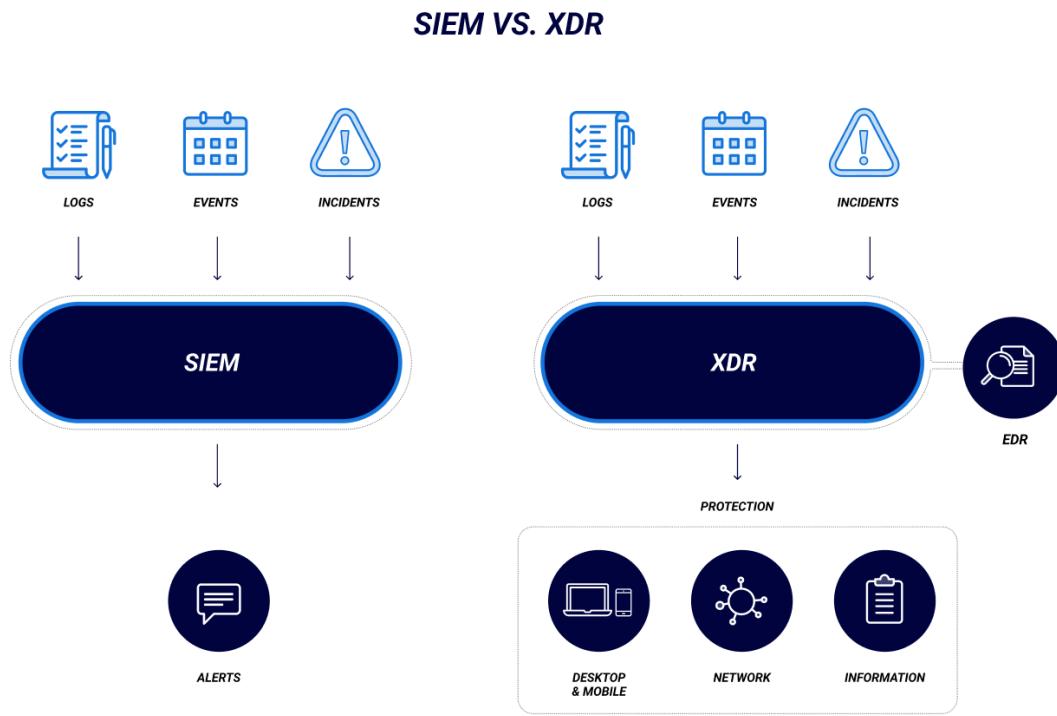


Hình 2: Lớp bảo mật XDR

Các điểm nổi bật của XDR bao gồm:

- *Tự động phát hiện và phản hồi* các mối đe dọa dựa trên trí tuệ nhân tạo (AI) và học máy (Machine Learning).
- *Tích hợp nhiều lớp bảo mật* để cung cấp tầm nhìn toàn diện về môi trường bảo mật của tổ chức.
- *Phân tích sâu các mối đe dọa* từ nhiều nguồn khác nhau, giúp nâng cao khả năng phát hiện và giảm thiểu rủi ro.
- *Phản ứng tự động* đối với các sự cố bảo mật, giảm thời gian phản hồi và xử lý mối đe dọa.

1.1.3 So sánh giữa SIEM và XDR



Hình 3: Sự khác nhau giữa SIEM và XDR

- **SIEM** tập trung vào việc thu thập và phân tích dữ liệu từ nhiều hệ thống và ứng dụng, hỗ trợ tuân thủ và phát hiện mối đe dọa thông qua quy tắc và quy định.

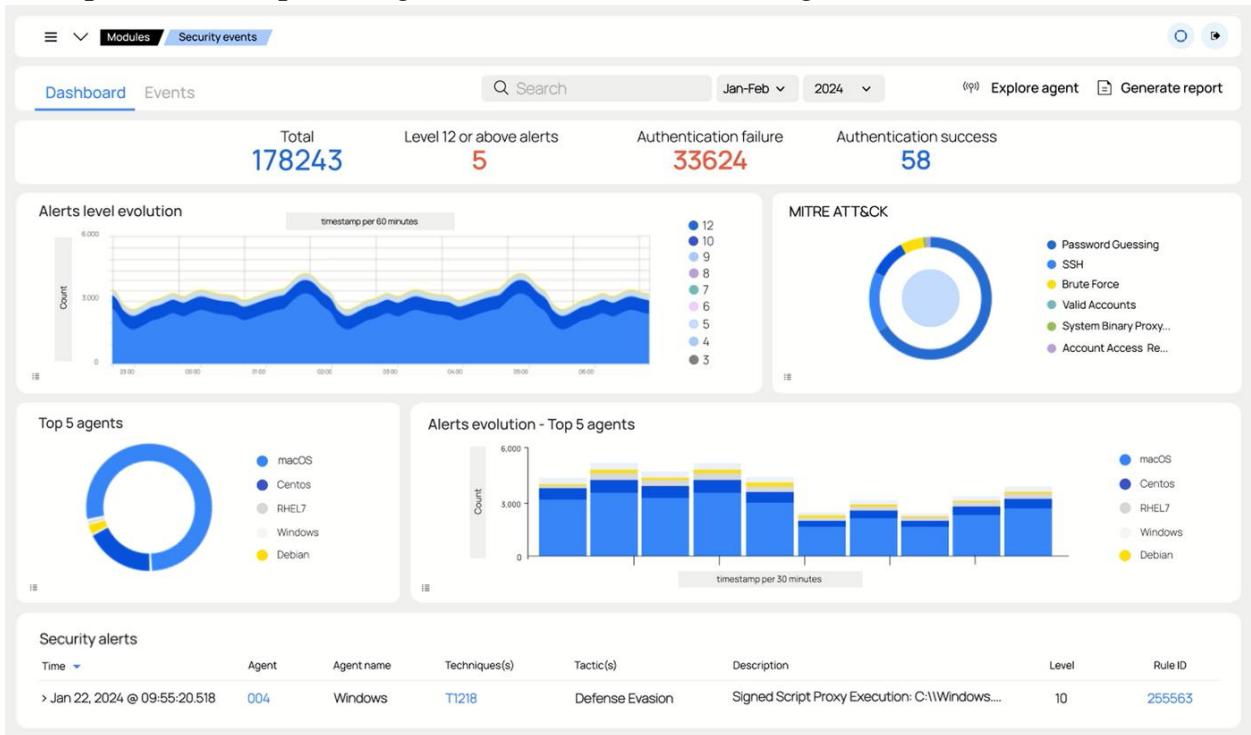
- **XDR** là giải pháp toàn diện hơn, tích hợp các chức năng của SIEM và mở rộng với khả năng phản ứng tự động, kết hợp nhiều lớp bảo mật để tối ưu hóa khả năng phát hiện và phản ứng trước các mối đe dọa.

SIEM và XDR đều có giá trị riêng trong môi trường bảo mật hiện đại. SIEM chủ yếu phục vụ việc tuân thủ và báo cáo sự kiện, trong khi XDR giúp tối ưu hóa quy trình phát hiện và phản ứng trước các mối đe dọa phức tạp.

1.2 Vai trò của Wazuh trong lĩnh vực bảo mật

Wazuh là một nền tảng mã nguồn mở mạnh mẽ, được thiết kế để cung cấp các giải pháp bảo mật toàn diện, chủ yếu tập trung vào quản lý sự kiện và phát hiện các

mỗi đe dọa bảo mật. Wazuh đóng vai trò quan trọng trong việc giúp các tổ chức xây dựng và duy trì hệ thống bảo mật chủ động thông qua khả năng giám sát, phát hiện, phân tích, và phản ứng với các sự cố an ninh mạng.



Hình 4: Giao diện web của Wazuh

1.2.1 Quản lý sự kiện bảo mật (SIEM)

Wazuh hoạt động như một giải pháp *SIEM (Security Information and Event Management)*, giúp thu thập, phân tích và quản lý dữ liệu bảo mật từ nhiều nguồn khác nhau, bao gồm máy chủ, thiết bị mạng, ứng dụng, và các dịch vụ đám mây.

Một số vai trò chính của Wazuh trong quản lý sự kiện bảo mật bao gồm:

- *Thu thập dữ liệu bảo mật* từ nhiều nguồn thông qua các agent hoặc tích hợp với các hệ thống khác như tường lửa, hệ điều hành, cơ sở dữ liệu, và ứng dụng.
- *Phân tích và tương quan sự kiện* nhằm phát hiện các hành vi bất thường hoặc các mối đe dọa tiềm ẩn dựa trên các quy tắc đã được thiết lập trước.
- *Cảnh báo theo thời gian thực* khi phát hiện các mối đe dọa hoặc hành vi bất thường, từ đó cung cấp thông tin chi tiết cho đội ngũ bảo mật để đưa ra phản ứng kịp thời.

- *Tạo báo cáo tuân thủ* cho các tiêu chuẩn bảo mật như GDPR, HIPAA, PCI-DSS, giúp các tổ chức duy trì và chứng minh việc tuân thủ các quy định bảo mật quan trọng.

1.2.2 Phát hiện xâm nhập (HIDS/NIDS)

Wazuh tích hợp khả năng *HIDS* (*Host-based Intrusion Detection System*), cung cấp khả năng phát hiện xâm nhập ở cấp độ máy chủ thông qua việc giám sát các thay đổi trên hệ thống và phát hiện các hành vi bất thường.

Nó giúp theo dõi:

- *Nhật ký hệ thống*: Wazuh theo dõi và phân tích các nhật ký của hệ điều hành, thiết bị mạng, và ứng dụng để phát hiện các hoạt động đáng ngờ như đăng nhập thất bại, cố gắng truy cập trái phép, và các thay đổi không hợp lệ trong cấu hình.
- *Tệp tin và cấu hình hệ thống*: Wazuh có thể giám sát tính toàn vẹn của tệp tin và hệ thống (FIM - File Integrity Monitoring), cảnh báo khi có bất kỳ thay đổi trái phép nào xảy ra.
- *Phát hiện phần mềm độc hại*: Wazuh có thể so sánh các tệp tin hệ thống với cơ sở dữ liệu về phần mềm độc hại để phát hiện và cảnh báo về các tệp tin hoặc quy trình có dấu hiệu nhiễm mã độc.

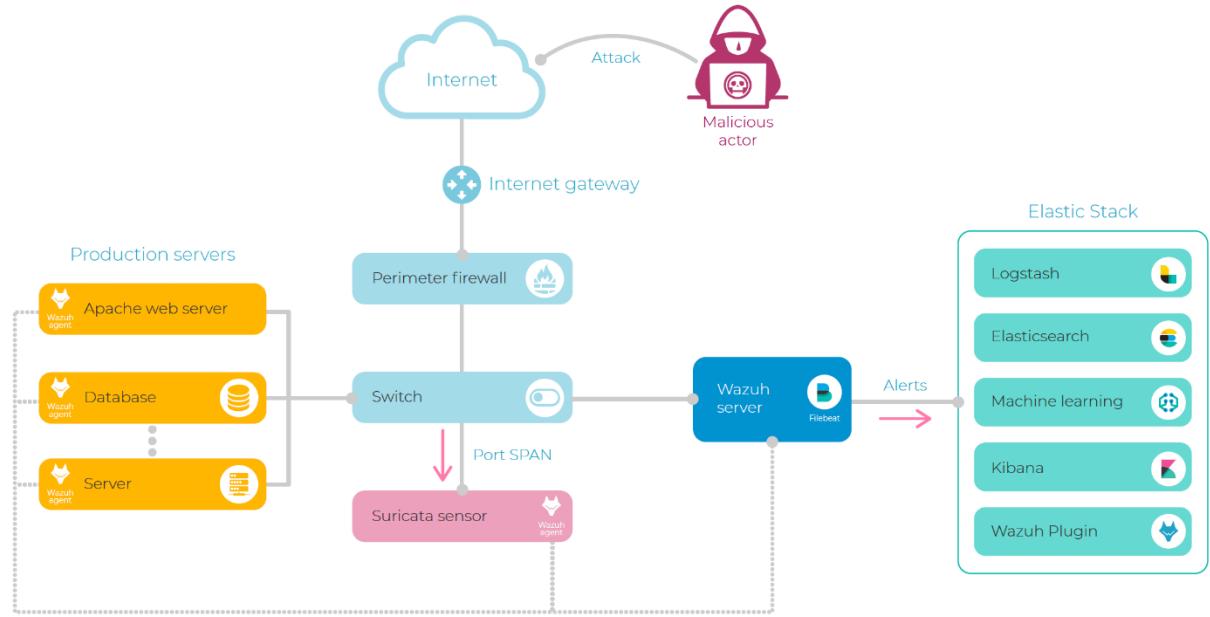
1.2.3 Giám sát bảo mật liên tục

Wazuh đóng vai trò như một nền tảng *giám sát bảo mật liên tục* bằng cách thu thập và phân tích thông tin bảo mật trong thời gian thực từ các máy chủ, ứng dụng và mạng lưới. Nó giúp:

- *Phát hiện mối đe dọa kịp thời*: Với khả năng phân tích dựa trên các mẫu tấn công và hành vi bất thường, Wazuh giúp phát hiện sớm các cuộc tấn công hoặc hành vi xâm phạm an ninh, từ đó giảm thiểu thiệt hại.
- *Tăng cường khả năng phản ứng*: Wazuh cung cấp khả năng tích hợp với các hệ thống khác để tự động hóa việc phản hồi khi phát hiện mối đe dọa, giảm thời gian phản ứng.
- *Tối ưu hóa hiệu suất bảo mật*: Wazuh cung cấp các bảng điều khiển (dashboard) trực quan giúp quản trị viên dễ dàng theo dõi và đánh giá trạng thái an ninh của hệ thống một cách nhanh chóng và toàn diện.

1.2.4 Tích hợp với Elastic Stack để phân tích dữ liệu bảo mật

Wazuh tích hợp chặt chẽ với *Elastic Stack* (*Elasticsearch*, *Logstash*, và *Kibana*) để cung cấp khả năng lưu trữ, phân tích, và trực quan hóa dữ liệu bảo mật.



Hình 5: Nâng cao phân tích bảo mật tích hợp Wazuh với Elastic Stack

Sự kết hợp này mang lại lợi ích vượt trội trong việc:

- *Lưu trữ dữ liệu an toàn và hiệu quả*: Elasticsearch giúp lưu trữ khối lượng lớn dữ liệu bảo mật, đồng thời cung cấp các công cụ tìm kiếm và phân tích mạnh mẽ.
- *Trực quan hóa dữ liệu bảo mật*: Kibana cung cấp các giao diện đồ họa trực quan để hiển thị dữ liệu bảo mật, giúp quản trị viên dễ dàng theo dõi và phân tích tình hình bảo mật trong thời gian thực.
- *Tự động hóa quy trình phân tích*: Logstash xử lý và định dạng dữ liệu từ nhiều nguồn khác nhau, đảm bảo tính toàn vẹn và đầy đủ của thông tin được gửi tới Wazuh để phân tích.

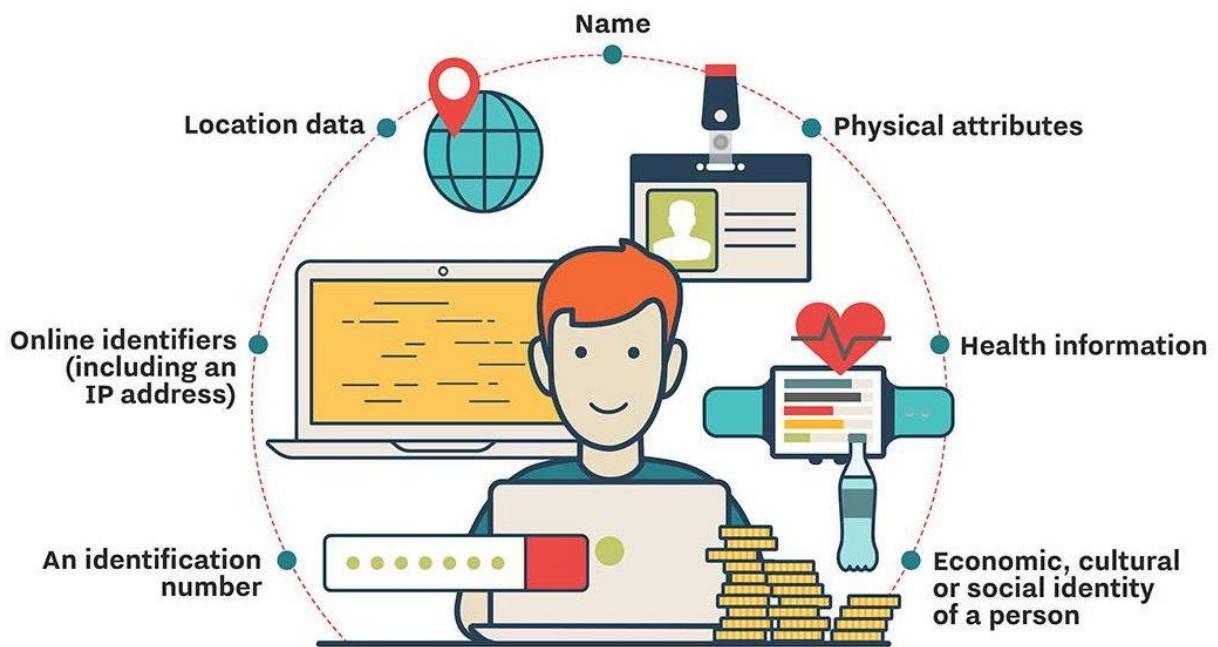
1.2.5 Đáp ứng các yêu cầu tuân thủ bảo mật

Wazuh đóng vai trò quan trọng trong việc giúp các tổ chức *tuân thủ các tiêu chuẩn và quy định bảo mật*. Nó cung cấp các báo cáo chi tiết và tự động về các sự kiện bảo mật, giúp tổ chức chứng minh việc tuân thủ các quy định về bảo mật thông tin như:

- **GDPR:** Đảm bảo rằng tổ chức xử lý và bảo vệ dữ liệu cá nhân theo quy định của Liên minh Châu Âu.

GDPR PERSONAL DATA

The EU's General Data Protection Regulation defines personal data as any information related to a person that can be used to directly or indirectly identify them, including:



Hình 6: Tiêu chuẩn GDPR bảo vệ dữ liệu cá nhân

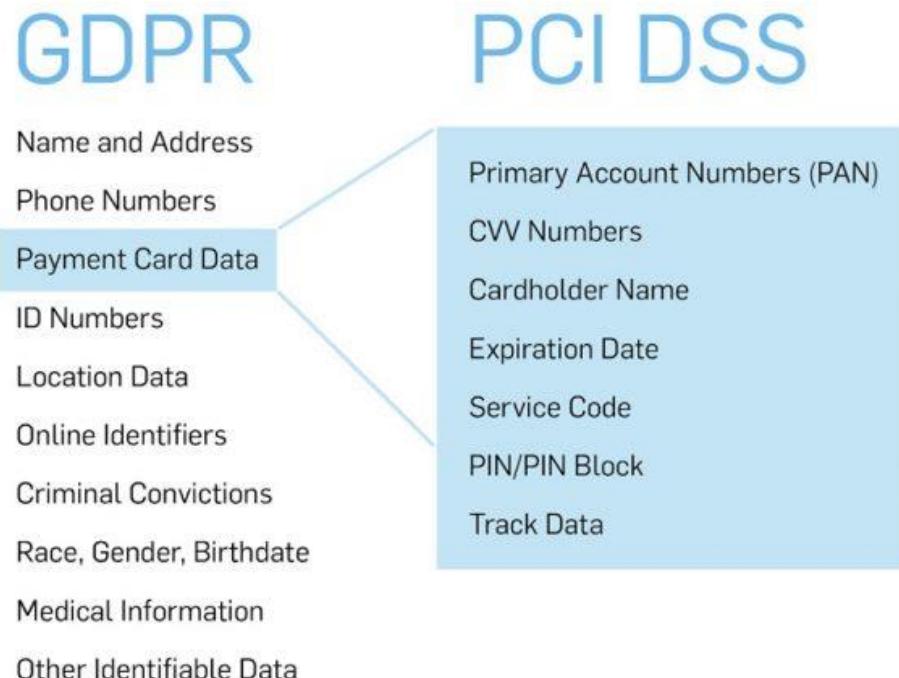
- **HIPAA:** Hỗ trợ tổ chức y tế tuân thủ các tiêu chuẩn bảo mật dữ liệu sức khỏe.

ĐỐI TƯỢNG CẦN TUÂN THỦ LUẬT HIPAA



Hình 7: Các đối tượng cần tuân thủ HIPAA

- **PCI-DSS:** Giúp các tổ chức quản lý thông tin thẻ tín dụng theo yêu cầu của Hội đồng Chuẩn thanh toán.



Hình 8: Các mục tiêu giám sát PCI-DSS

1.2.6 Phát hiện và phản ứng mở rộng (XDR)

Wazuh cung cấp khả năng *mở rộng phát hiện và phản ứng (XDR)*, cho phép theo dõi và xử lý sự cố trên nhiều lớp bảo mật khác nhau, bao gồm mạng, thiết bị đầu cuối, và các ứng dụng. Điều này giúp tổ chức có tầm nhìn toàn diện và khả năng phản ứng nhanh chóng với các mối đe dọa phức tạp trong môi trường bảo mật hiện đại.

1.3 So sánh giữa các giải pháp SIEM/XDR khác và Wazuh

Tiêu chí	Wazuh	Giải pháp SIEM/XDR khác
Mã nguồn	Mã nguồn mở, miễn phí	Phần lớn là mã nguồn đóng, có phí cao
Chi phí	Không có chi phí bản quyền, chỉ tốn chi phí triển khai	Thường có phí bản quyền cao, bao gồm cả phí bảo trì
Khả năng tích hợp	Tích hợp tốt với Elasticsearch, Kibana, Suricata, AWS, Azure, GCP	Tùy thuộc vào nhà cung cấp, thường bị hạn chế bởi công nghệ độc quyền
Khả năng mở rộng	Linh hoạt, có thể giám sát hàng ngàn thiết bị	Tùy thuộc vào giải pháp, thường yêu cầu nâng cấp bản quyền để mở rộng
Phát hiện mối đe dọa	Cung cấp khả năng phát hiện mối đe dọa theo thời gian thực dựa trên quy tắc và hành vi	Tương tự, nhưng một số giải pháp có khả năng AI và machine learning nâng cao hơn
Quản lý nhật ký	Phân tích nhật ký từ nhiều nguồn khác nhau	Tương tự, nhưng một số giải pháp cung cấp thêm tính năng giám sát nâng cao
Tuân thủ bảo mật	Hỗ trợ tuân thủ các tiêu chuẩn như PCI-DSS, HIPAA, GDPR	Tương tự, thường có hỗ trợ tuân thủ tốt hơn nhưng phụ thuộc vào gói dịch vụ
Tự động hóa phản ứng	Có khả năng tự động hóa phản ứng sự cố	Nhiều giải pháp khác có tự động hóa tốt hơn, đặc biệt là XDR hiện đại
Hỗ trợ giám sát đám mây	Hỗ trợ giám sát trên AWS, Azure, GCP	Tương tự, nhưng các giải pháp thương mại có thể hỗ trợ nhiều dịch vụ đám mây hơn
Tính dễ sử dụng	Yêu cầu kiến thức về thiết lập và tích hợp hệ thống	Thường có giao diện người dùng trực quan hơn và hỗ trợ tốt hơn
Tùy biến	Có khả năng tùy biến cao với các quy tắc và cấu hình	Tùy biến nhưng bị hạn chế bởi cấu trúc hệ thống độc quyền
Hỗ trợ cộng đồng	Cộng đồng mã nguồn mở mạnh, tài liệu phong phú	Thường có hỗ trợ chuyên nghiệp, nhưng cộng đồng người dùng nhỏ hơn
Bảo mật hệ thống	Phát hiện và ngăn chặn mối đe dọa trên nhiều lớp, từ thiết bị đầu cuối đến đám mây	Một số giải pháp XDR hiện đại có khả năng bảo mật tiên tiến hơn trên các nền tảng đa dạng
Thời gian triển khai	Triển khai nhanh chóng nhưng cần thời gian để tinh chỉnh	Thường triển khai nhanh hơn nhờ có các mô hình được thiết lập trước

Hình 9: So sánh giữa các giải pháp SIEM/XDR khác và Wazuh

CHƯƠNG 2: TRIỂN KHAI THỰC NGHIỆM

2.1 Hướng đối tượng doanh nghiệp

Wazuh hướng đến đối tượng doanh nghiệp với các yêu cầu bảo mật cao. Cụ thể, giải pháp này phù hợp với các tổ chức và doanh nghiệp có nhu cầu:

- Quản lý an ninh hiệu quả: quản lý an ninh như giám sát log, giám sát hệ thống, phát hiện xâm nhập, quản lý sự kiện, quản lý cấu hình, và quản lý báo cáo.
- Phát hiện và phòng ngừa xâm nhập: phát hiện và phản ứng đối với các cuộc tấn công mạng, từ những tấn công đơn giản như quét cổng đến các cuộc tấn công phức tạp như tấn công từ chối dịch vụ (DoS) hoặc tấn công dò mật khẩu. Điều này giúp ngăn chặn những mối đe dọa tiềm ẩn tổ chức và bảo vệ thông tin quan trọng.
- Tăng cường khả năng phản ứng và ứng phó: Wazuh cung cấp cơ chế cảnh báo và thông báo sớm về các mối đe dọa bảo mật, giúp tổ chức phản ứng kịp thời và ứng phó hiệu quả. Thông qua việc tổ chức cung cấp cấu hình và quản lý an ninh, Wazuh giúp đẩy lùi khả năng mất dữ liệu và thiệt hại tiềm ẩn từ các cuộc tấn công mạng.

2.2 Yêu cầu kỹ thuật

- Phần cứng

Yêu cầu phần cứng phụ thuộc vào số lượng điểm cuối được bảo vệ và khối lượng công việc trên đám mây. Con số này có thể giúp ước tính lượng dữ liệu sẽ được phân tích và bao nhiêu cảnh báo bảo mật sẽ lưu trữ và lập chỉ mục.

Theo bước bắt đầu nhanh này có nghĩa là triển khai máy chủ Wazuh, bộ chỉ mục Wazuh và bảng điều khiển Wazuh trên cùng một máy chủ. Điều này thường đủ để theo dõi tối đa 100 điểm cuối và trong 90 ngày dữ liệu cảnh báo có thể truy vấn/lập chỉ mục. Bảng bên dưới hiển thị phần cứng được đề xuất để triển khai nhanh:

Agent	CPU	RAM	Storage (90 days)
1 - 25	4 vCPU	8 GiB	50 GB
25 - 50	8 vCPU	8 GiB	100 GB
50 - 100	8 vCPU	8 GiB	200 GB

- Hệ điều hành

Các thành phần trung tâm của Wazuh có thể được cài đặt trên hệ điều hành Linux 64-bit. Wazuh đề xuất bất kỳ phiên bản hệ điều hành nào sau đây:

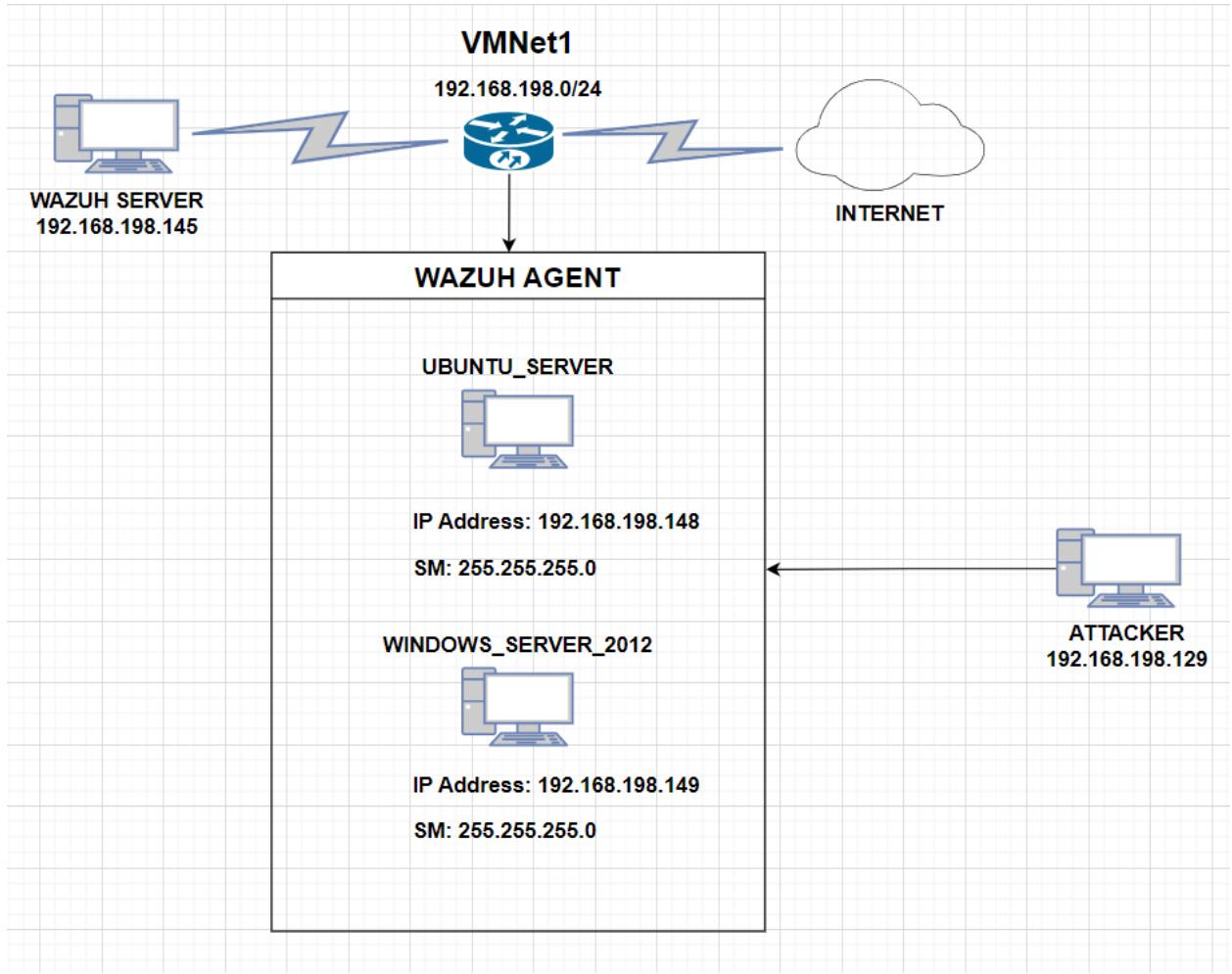
Amazon Linux 2	CentOS 7, 8
Red Hat Enterprise Linux 7, 8, 9	Ubuntu 16.04, 18.04, 20.04, 22.04

- Tính tương thích của trình duyệt Web

- Chrome 95 trở lên
- Firefox 93 trở lên
- Safari 13.7 trở lên

2.3 Triển khai Wazuh Server và Agents

Mô hình triển khai



Hình 10: Mô hình triển khai thực nghiệm

2.3.1 Cài đặt Wazuh

Bước 1: Cập nhật hệ thống

Trước khi bắt đầu, hãy đảm bảo rằng hệ thống được cập nhật với các bản vá và phần mềm mới nhất.

```
sudo apt update
```

```
sudo apt upgrade -y
```

Bước 2: Tải các gói cần thiết cho máy

`apt install apt-transport-https zip unzip lsb-release curl gnupg net-tools` (cả 2 máy Wazuh Server và Agent Ubuntu)

Bước 3: Cài đặt khóa GPG (GNU Privacy Guard: là một loại khóa mã hóa được sử dụng trong hệ thống mã hóa **GPG**, một phần mềm mã nguồn mở dùng để mã hóa và xác thực dữ liệu. GPG hỗ trợ cả mã hóa **symmetric** (mã hóa đối xứng) và **asymmetric** (mã hóa bất đối xứng), trong đó mã hóa bất đối xứng là cơ chế thường dùng cho các khóa GPG.)

```
curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/elasticsearch.gpg --import && chmod 644 /usr/share/keyrings/elasticsearch.gpg
```

```
root@ubuntu-server:~# curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/elasticsearch.gpg --import && chmod 644 /usr/share/keyrings/elasticsearch.gpg
gpg: keyring '/usr/share/keyrings/elasticsearch.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key D27D666CD88E42B4: public key "Elasticsearch (Elasticsearch Signing Key) <dev_@ops@elasticsearch.org>" imported
gpg: Total number processed: 1
gpg:          imported: 1
```

Bước 4: Thêm kho lưu trữ Wazuh

```
echo "deb [signed-by=/usr/share/keyrings/elasticsearch.gpg]
https://artifacts.elastic.co/packages/7.x/apt stable main" | tee
/etc/apt/sources.list.d/elastic-7.x.list
```

```
root@ubuntu-server:~# echo "deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | tee /etc/apt/sources.list.d/elastic-7.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main
```

Bước 5: Cài đặt gói ElasticSearch

Cập nhật lại danh sách gói và cài đặt gói **ElasticSearch**:

```
sudo apt update
```

```
sudo apt install elasticsearch=7.17.9
```

Bước 6: Download file cấu hình /etc/elasticsearch/elasticsearch.yml

```
curl -s https://packages.wazuh.com/4.4/tpl/elastic-basic/elasticsearch_all_in_one.yml
```

Bước 7: Tạo các chứng chỉ (cert)

```
curl -so /usr/share/elasticsearch/instances.yml  
https://packages.wazuh.com/4.4/tpl/elastic-basic/instances_aio.yml  
/usr/share/elasticsearch/bin/elasticsearch-certutil cert ca --pem --in  
instances.yml --keep-ca-key --out ~/certs.zip
```

Bước 8: Giải nén tệp *certs.zip*

```
unzip ~/certs.zip -d ~/certs
```

```
root@ubuntu-server:~# unzip ~/certs.zip -d ~/certs  
Archive: /root/certs.zip  
  creating: /root/certs/ca/  
    inflating: /root/certs/ca/ca.crt  
    inflating: /root/certs/ca/ca.key  
  creating: /root/certs/elasticsearch/  
    inflating: /root/certs/elasticsearch/elasticsearch.crt  
    inflating: /root/certs/elasticsearch/elasticsearch.key  
root@ubuntu-server:~#
```

Bước 9: Tạo thư mục /etc/elasticsearch/certs

```
mkdir /etc/elasticsearch/certs/ca -p  
cp -R ~/certs/ca/ ~/certs/elasticsearch/* /etc/elasticsearch/certs/  
chown -R elasticsearch: /etc/elasticsearch/certs  
chmod -R 500 /etc/elasticsearch/certs  
chmod 400 /etc/elasticsearch/certs/ca/ca.*  
/etc/elasticsearch/certs/elasticsearch.*  
rm -rf ~/certs/ ~/certs.zip
```

Bước 10: Kích hoạt và bắt đầu dịch vụ Elasticsearch

```
systemctl daemon-reload  
systemctl enable elasticsearch.service  
systemctl start elasticsearch.service
```

```
root@ubuntu-server:~# systemctl daemon-reload
```

```
root@ubuntu-server:~# systemctl enable elasticsearch.service
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
root@ubuntu-server:~# systemctl start elasticsearch.service
root@ubuntu-server:~#
```

Bước 11: Tạo thông tin xác thực ngẫu nhiên

```
/usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
```

```
root@ubuntu-server:~# /usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beats_system,remote_monitoring_user.
The passwords will be randomly generated and printed to the console.
Please confirm that you would like to continue [y/N]y

Changed password for user apm_system
PASSWORD apm_system = 2WXhg6qo2eWWj7uIvrLX

Changed password for user kibana_system
PASSWORD kibana_system = zuWMu43uauloNS0HnUlb

Changed password for user kibana
PASSWORD kibana = zuWMu43uauloNS0HnUlb

Changed password for user logstash_system
PASSWORD logstash_system = gCZjDK02ACG4Ucu7e3zf

Changed password for user beats_system
PASSWORD beats_system = 5vCtUrkP2scilZHLwE3t

Changed password for user remote_monitoring_user
PASSWORD remote_monitoring_user = v1ZZYUbExYuXvQiozdRU

Changed password for user elastic
PASSWORD elastic = ybGj7NqxjETxQWklmpaV
```

Lưu lại password đã được tạo ngẫu nhiên ở trên

Bước 12: Kiểm tra quá trình cài đặt

```
curl -XGET https://localhost:9200 -u elastic:ybGj7NqxjETxQWklmpaV -k
```

ybGj7NqxjETxQWklmpaV: đây là mật khẩu của elastic

```
root@ubuntu-server:~# curl -XGET https://localhost:9200 -u elastic:ybGj7NqxjETxQWklmpaV -k
{
  "name" : "elasticsearch",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "I0VN41S2RDa1Ad9t1VBSRA",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Bước 13: Cài đặt khóa GPG

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
```

Bước 14: Thêm kho lưu trữ Wazuh

```
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" | tee -a
/etc/apt/sources.list.d/wazuh.list
```

Bước 15: Cập nhật thông tin gói

```
sudo apt update
```

Bước 16: Cài đặt gói quản lý Wazuh

```
apt install wazuh-manager
```

Bước 17: Kích hoạt và bắt đầu dịch vụ quản lý Wazuh

```
systemctl daemon-reload
```

```
systemctl enable wazuh-manager.service
```

```
systemctl start wazuh-manager.service
```

```
root@ubuntu-server:~# systemctl daemon-reload
root@ubuntu-server:~# systemctl enable wazuh-manager.service
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-manager.service → /lib/systemd/system/wazuh-manager.service.
root@ubuntu-server:~# systemctl start wazuh-manager.service
```

Bước 18: Kiểm tra trạng thái của Wazuh

systemctl status wazuh-manager

```
root@ubuntu-server:~# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
  Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor>
  Active: active (running) since Thu 2024-09-12 23:56:00 +07; 28s ago
    Process: 93339 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (c>
      Tasks: 146 (limit: 2204)
     Memory: 750.4M
       CGroup: /system.slice/wazuh-manager.service
               ├─93413 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
               ├─93453 /var/ossec/bin/wazuh-authd
               ├─93469 /var/ossec/bin/wazuh-db
               ├─93484 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
               ├─93487 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
               ├─93490 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
               ├─93503 /var/ossec/bin/wazuh-execd
               ├─93517 /var/ossec/bin/wazuh-analysisd
               ├─93560 /var/ossec/bin/wazuh-syscheckd
               ├─93575 /var/ossec/bin/wazuh-remoted
               ├─93585 /var/ossec/bin/wazuh-logcollector
               ├─93626 /var/ossec/bin/wazuh-monitord
               └─93636 /var/ossec/bin/wazuh-modulesd
```

Bước 19: Cài đặt gói Filebeat

apt install filebeat=7.17.9

Bước 20: Tải cấu hình Filebeat

```
curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.4/tpl/elastic-
basic/filebeat_all_in_one.yml
```

Bước 21: Tải mẫu cảnh báo cho Elasticsearch và cấp quyền go+r cho /etc/filebeat/wazuh-template.json

```
curl -so /etc/filebeat/wazuh-template.json
```

```
https://raw.githubusercontent.com/wazuh/wazuh/4.4/extensions/elasticsearch/7.x/wazuh-template.json
```

```
chmod go+r /etc/filebeat/wazuh-template.json
```

```
root@ubuntu-server:~# curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/4.4/extensions/elasticsearch/7.x/wazuh-template.json
root@ubuntu-server:~# chmod go+r /etc/filebeat/wazuh-template.json
```

Bước 22: Tải modun Wazuh cho filebeat

```
curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.2.tar.gz | tar -xvz -C /usr/share/filebeat/module
```

Bước 23: Chỉnh sửa tệp /etc/filebeat/filebeat.yml

```
nano /etc/filebeat/filebeat.yml
```

```
GNU nano 4.8          /etc/filebeat/filebeat.yml
# Wazuh - Filebeat configuration file
output.elasticsearch.hosts: ["127.0.0.1:9200"]
output.elasticsearch.password: <elasticsearch_password>

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: false

setup.template.json.enabled: true
setup.template.json.path: /etc/filebeat/wazuh-template.json
setup.template.json.name: wazuh
setup.template.overwrite: true
setup.ilm.enabled: false

output.elasticsearch.protocol: https
output.elasticsearch.ssl.certificate: /etc/elasticsearch/certs/elasticsearch.crt
output.elasticsearch.ssl.key: /etc/elasticsearch/certs/elasticsearch.key
[ Read 32 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit  ^R Read File  ^L Replace  ^U Paste Text  ^T To Spell  ^_ Go To Line
```

Thay thế <elasticsearch_password> bằng mật khẩu của elastic đã tạo và lưu trước đó

Bước 24: Sao chép các chứng chỉ vào /etc/filebeat/certs/

```
cp -r /etc/elasticsearch/certs/ca/ /etc/filebeat/certs/
cp /etc/elasticsearch/certs/elasticsearch.crt /etc/filebeat/certs/filebeat.crt
cp /etc/elasticsearch/certs/elasticsearch.key /etc/filebeat/certs/filebeat.key
```

Bước 25: Kích hoạt và bắt đầu dịch vụ filebeat

```
systemctl daemon-reload
systemctl enable filebeat.service
systemctl start filebeat.service
```

Để đảm bảo dịch vụ filebeat đã được cài đặt thành công chúng ta chạy lệnh sau:
filebeat test output

```
root@ubuntu-server:~# filebeat test output
elasticsearch: https://127.0.0.1:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 127.0.0.1
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.17.9
```

Bước 26: Cài đặt gói Kibana

```
apt install kibana=7.17.9
```

Bước 27: Sao chép các chứng chỉ Elasticsearch vào thư mục cấu hình kibana

```
mkdir /etc/kibana/certs/ca -p
cp -R /etc/elasticsearch/certs/ca/ /etc/kibana/certs/
cp /etc/elasticsearch/certs/elasticsearch.key /etc/kibana/certs/kibana.key
cp /etc/elasticsearch/certs/elasticsearch.crt /etc/kibana/certs/kibana.crt
chown -R kibana:kibana /etc/kibana/
chmod -R 500 /etc/kibana/certs/
chmod 440 /etc/kibana/certs/ca/ca.* /etc/kibana/certs/kibana.*
```

Bước 28: Tải tệp cấu hình kibana

```
curl -so /etc/kibana/kibana.yml https://packages.wazuh.com/4.4/tpl/elasticsearch/kibana_all_in_one.yml
```

Bước 29: Chỉnh sửa tệp /etc/kibana/kibana.yml

```
nano /etc/kibana/kibana.yml
```

```
GNU nano 4.8                               /etc/kibana/kibana.yml
server.host: 0.0.0.0
server.port: 443
elasticsearch.hosts: https://localhost:9200
elasticsearch.password: <elasticsearch_password>

# Elasticsearch from/to Kibana

elasticsearch.ssl.certificateAuthorities: /etc/kibana/certs/ca/ca.crt
elasticsearch.ssl.certificate: /etc/kibana/certs/kibana.crt
elasticsearch.ssl.key: /etc/kibana/certs/kibana.key

# Browser from/to Kibana
server.ssl.enabled: true
server.ssl.certificate: /etc/kibana/certs/kibana.crt
server.ssl.key: /etc/kibana/certs/kibana.key

# Elasticsearch authentication
xpack.security.enabled: true
elasticsearch.username: elastic
uiSettings.overrides.defaultRoute: "/app/wazuh"
                                         [ Read 22 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^I Replace   ^I Paste Text^T To Snell  ^  Go To Line
```

Tương tự, thay thế <elasticsearch_password> bằng mật khẩu của elastic đã tạo ngẫu nhiên và lưu trước đó

Bước 30: Tạo thư mục /usr/share/kibana/data

```
mkdir /usr/share/kibana/data
```

```
chown -R kibana:kibana /usr/share/kibana/
```

Bước 31: Cài đặt plugin Wazuh kibana

```
cd /usr/share/kibana/
```

```
sudo -u kibana /usr/share/kibana/bin/kibana-plugin install
```

```
https://packages.wazuh.com/4.x/ui/kibana/wazuh\_kibana-4.4.5\_7.17.9-1.zip
```

```
root@ubuntu-server:~# cd /usr/share/kibana/
root@ubuntu-server:/usr/share/kibana# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.4.5_7.17.9-1.zip
Attempting to transfer from https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.4.5_7.17.9-1.zip
Transferring 36505918 bytes.....
Transfer complete
Retrieving metadata from plugin archive
Extracting plugin archive
Extraction complete
Plugin installation complete
```

Bước 32: Liên kết socket của kibana vào cổng đặc quyền 443

```
setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node
```

```
root@ubuntu-server:/usr/share/kibana# setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node
root@ubuntu-server:/usr/share/kibana#
```

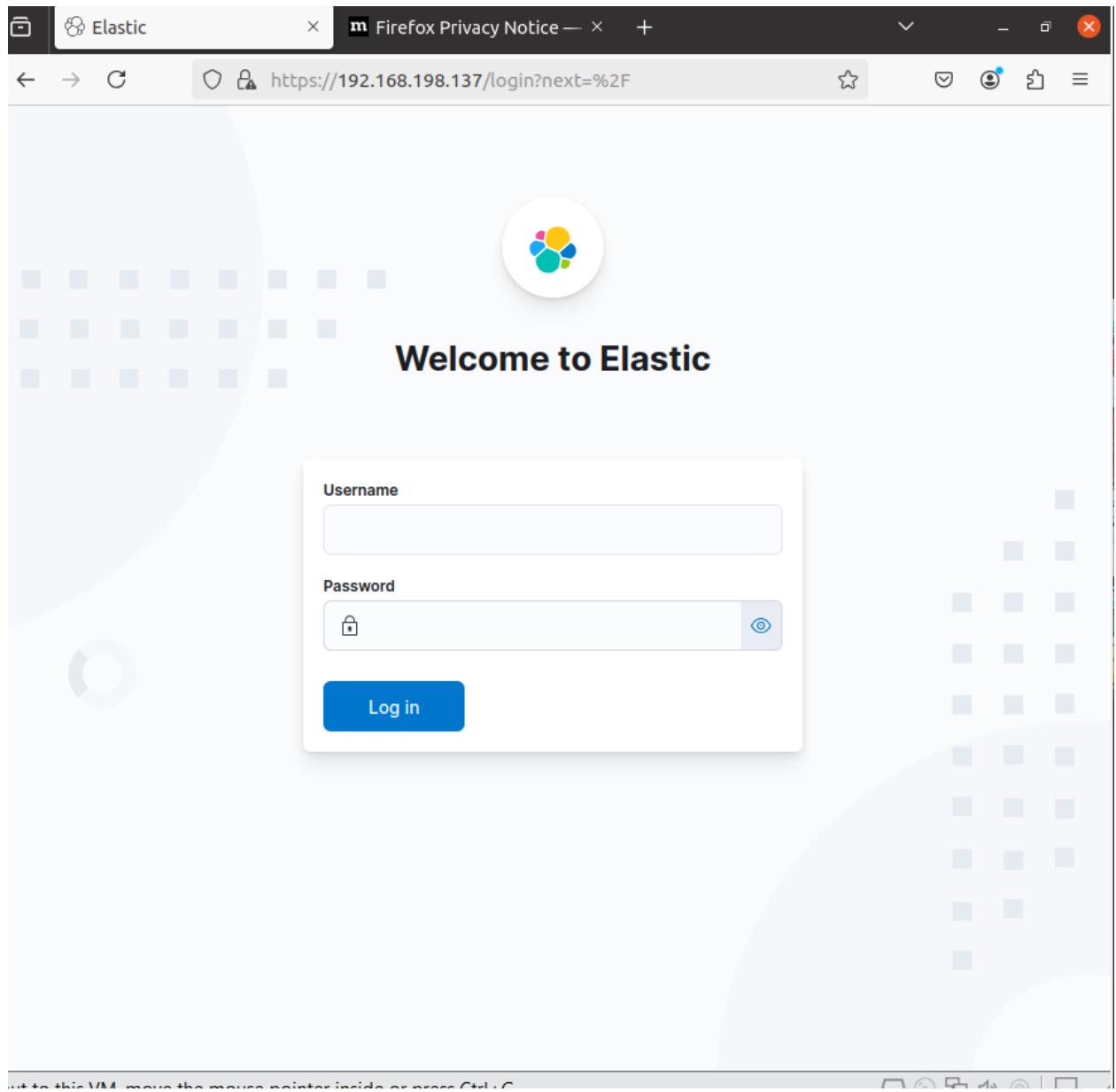
Bước 33: Kích hoạt và bắt đầu dịch vụ kibana

```
systemctl daemon-reload
```

```
systemctl enable kibana.service
```

```
systemctl start kibana.service
```

Bước 34: Truy cập vào giao diện web của Wazuh



Username: elastic

Password: đã tạo và lưu trước đó

2.3.2 Triển khai giám sát các Agent

- Đầu tiên, đối với Ubuntu Agent:

Bước 1: Import khóa GPG

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
```

Bước 2: Thêm kho lưu trữ

```
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
```

Bước 3: Cập nhật các thông tin gói

```
apt update
```

Bước 4: Triển khai wazuh agent

```
WAZUH_MANAGER=192.168.198.145 apt install wazuh-agent
```

192.168.198.145: Địa chỉ của wazuh server

Bước 5: Kích hoạt và bắt đầu dịch vụ wazuh agent

```
systemctl daemon-reload
```

```
systemctl enable wazuh-agent.service
```

```
systemctl start wazuh-agent.service
```

Kiểm tra: Ta đã thành công kết nối giám sát với máy ubuntu-agent

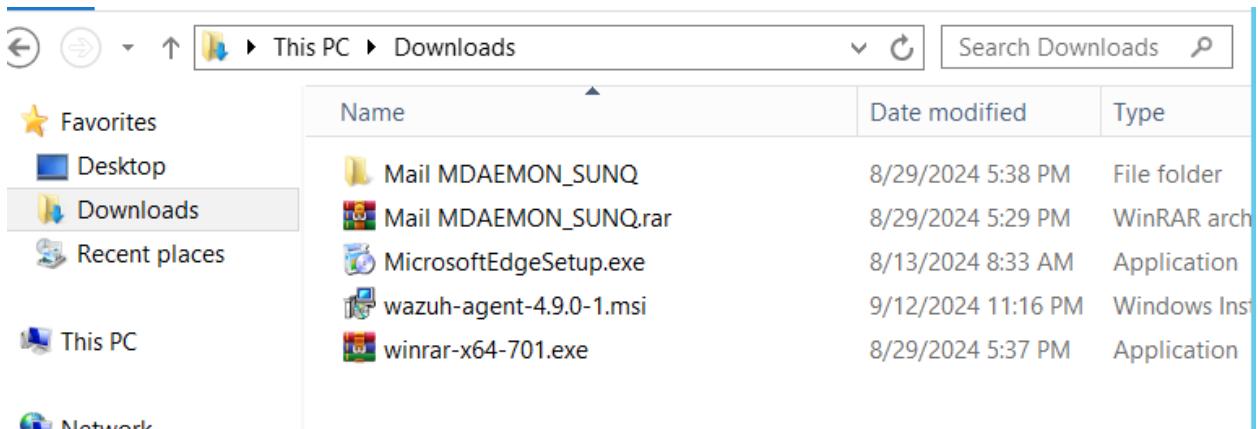
ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	ubuntu-agent	192.168.198.148	default	Ubuntu 22.04.5 LTS	node01	v4.9.0	active	

- Tiếp theo, với máy Agent Windows Server 2012:

Ta sẽ vào trang chủ của Wazuh để tải về phần mềm quản lý Wazuh Agent

Link: <https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-windows.html>

Bước 1: Cài đặt Wazuh Agent lưu ở thư mục download



Bước 2: Triển khai Wazuh Agent

Mở CMD

cd Downloads

wazuh-agent-4.9.0-1.msi /q WAZUH_MANAGER=192.168.198.145

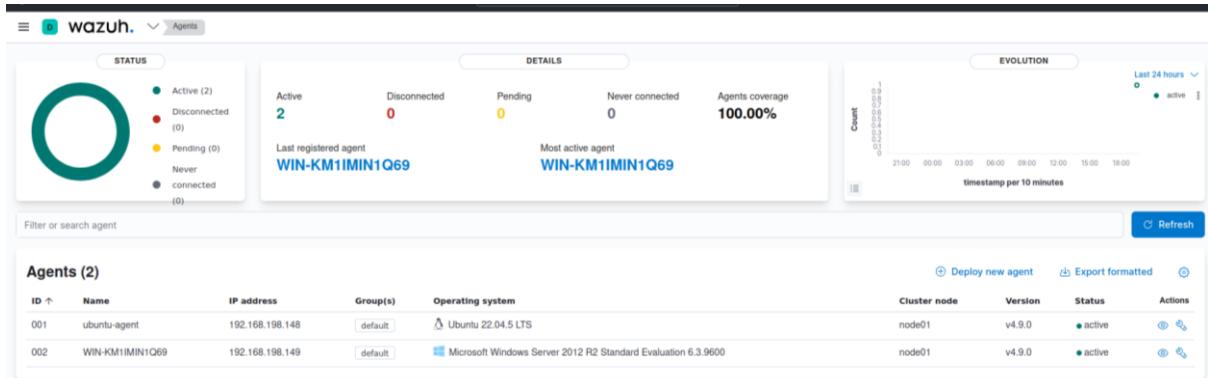
192.168.198.145: Địa chỉ của wazuh server

Bước 3: Bắt đầu chạy Wazuh Agent

NET START Wazuh

```
C:\Users\Administrator\Downloads>NET START Wazuh
The Wazuh service is starting.
The Wazuh service was started successfully.
```

Kiểm tra: thành công kết nối giám sát máy Windows Server Agent



2.4 Cấu hình Wazuh phát hiện cuộc tấn công Brute-Force

Tấn công **brute-force** (hay còn gọi là tấn công dò tìm mật khẩu bằng phương pháp vét cạn) là một phương pháp tấn công bảo mật, trong đó kẻ tấn công thử mọi tổ hợp có thể của các ký tự, số, hoặc ký hiệu cho đến khi tìm ra thông tin đăng nhập đúng hoặc khóa mã hóa.

Kịch bản: Sử dụng máy tấn công là kali linux với IP: 192.168.198.129 tấn công lên máy Agent Ubuntu bằng giao thức SSH và tấn công máy Agent Windows Server bằng giao thức RDP

- Máy tấn công ta sử dụng một công cụ có tên là Hydra
- Chạy Hydra trong vòng khoảng 2p khi cuộc tấn công diễn ra, Wazuh Server sẽ thu thập thông tin log và phân tích sau đó hiển thị cảnh báo đây là cuộc tấn công brute-force

- **Tấn công máy Agent Ubuntu**

Lưu ý: Hãy đảm bảo rằng máy Agent Ubuntu có cài đặt dịch vụ SSH và mở default port: 22

B1: Tạo file wordlist có tên là pass.txt với 10 mật khẩu khác nhau

sudo nano pass.txt

```

GNU nano 8.1                               pass.txt *
1
2
3
4
5
6
7
8
9
abc123

File Name to Write: pass.txt
^G Help          M-D DOS Format      M-A Append      M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend    ^T Browse

```

B2: Chạy hydra

hydra -l ubuntu -P pass.txt 192.168.198.148 ssh

```

[✓] (root㉿kali)-[~/Desktop]
# hydra -l abc -P pass.txt 192.168.198.148 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-29 09:
40:52
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:1/p:10)
, ~1 try per task
[DATA] attacking ssh://192.168.198.148:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-29 09:
40:57

```

Kết quả: Vào Wazuh Server để kiểm tra kết quả giám sát máy chủ Ubuntu
Với máy Agent Ubuntu có các cảnh báo sau:

- Rule ID 5710: là đang có đăng nhập tài khoản người dùng không tồn tại. Quy tắc ở cấp độ 5. Chúng ta có thể thấy được địa chỉ nguồn của máy attacker là: 192.168.198.129.

> Sep 29, 2024 @ 20:40:57.050	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Sep 29, 2024 @ 20:40:57.047	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Sep 29, 2024 @ 20:40:57.047	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Sep 29, 2024 @ 20:40:57.047	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Sep 29, 2024 @ 20:40:57.045	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Sep 29, 2024 @ 20:40:57.040	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710

Table JSON Rule

@timestamp	2024-09-29T13:40:57.050Z
_id	OxsDPplBinS_hBkiWWFL
agent.id	001
agent.ip	192.168.198.148
agent.name	ubuntu-agent
data.srcip	192.168.198.129
data.srcuser	abc
decoder.name	sshd
decoder.parent	sshd
full_log	Sep 29 13:40:56 ubuntu-agent sshd[5612]: Failed password for invalid user abc from 192.168.198.129 port 59264 ssh2
id	1727617257.1691882
input.type	log
location	journald

- Rule ID 5503: người dùng đăng nhập thất bại. Cấp độ 5
- Rule ID 5551: nhiều lần đăng nhập không thành công trong 1 khoảng thời gian ngắn. Cấp độ 10

> Sep 29, 2024 @ 20:40:55.093	T1110	Credential Access	PAM: Multiple failed logins in a small period of time.	10	5551
> Sep 29, 2024 @ 20:40:55.083	T1110.001	Credential Access	PAM: User login failed.	5	5503

- **Tấn công máy Agent Windows Server**

Lưu ý: Hãy đảm bảo rằng đã cài Remote Desktop trên Agent Windows Server và cấu hình user kết nối RDP

Chạy hydra

```
hydra -l abc -P pass.txt rdp://192.168.198.149
```

```
53:18
└─[root@kali]─[~/Desktop]
# hydra -l abc -P pass.txt rdp://192.168.198.149
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
  military or secret service organizations, or for illegal purposes (this is n
  on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-29 11:
32:15
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to
reduce the number of parallel connections and -W 1 or -W 3 to wait between co
nnection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connecti
ons)
[WARNING] the rdp module is experimental. Please test, report - and if possib
le, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 10 login tries (l:1/p:10),
~3 tries per task
[DATA] attacking rdp://192.168.198.149:3389/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-29 11:
32:19
```

Vào Wazuh để kiểm tra kết quả giám sát Agent Windows Server

Với máy Agent Windows Server có cảnh báo sau:

Rule ID 60122: Lỗi người dùng đăng nhập hoặc mật khẩu sai. Cấp độ 5

T1531	Impact	Logon Failure - Unknown user or bad password	5	60122
-------	--------	--	---	-------

2.5 Cấu hình Wazuh phát hiện các cuộc tấn công SQL Injection

SQL Injection là một hình thức tấn công bảo mật trên ứng dụng web, trong đó kẻ tấn công chèn các câu lệnh SQL độc hại vào đầu vào (input) của ứng dụng để thực hiện

các truy vấn không mong muốn tới cơ sở dữ liệu. Mục tiêu của cuộc tấn công này là truy cập, thay đổi hoặc xóa dữ liệu mà không được phép.

Kịch bản: Tấn công vào database. Kẻ tấn công sẽ sử dụng kỹ thuật SQL Injection để tấn công vào database của máy chủ Web:

- Wazuh Server: sẽ chịu trách nhiệm thu thập log phân tích và sau đó hiển thị cảnh báo
- Wazuh Agent: sẽ là máy victim bị tấn công SQL Injection
- Kali linux: là máy attacker

Lưu ý: Tường lửa có thể đang chặn cổng 80. Đảm bảo rằng tường lửa đã mở cổng 80 cho HTTP. Có thể sử dụng lệnh sau để kiểm tra cấu hình tường lửa:

- Trên Ubuntu với UFW (Uncomplicated Firewall):

```
sudo ufw status
```

- Nếu cổng 80 không được mở, có thể mở bằng cách:

```
sudo ufw allow 80/tcp
```

B1: Ở máy Agent Ubuntu cập nhật và cài đặt máy chủ Web Apache

```
apt install apache2
```

B2: Kiểm tra trạng thái dịch vụ Apache đã chạy hay chưa

```
systemctl status apache2.service
```

```
root@ubuntu-agent:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Mon 2024-09-30 00:12:41 +07; 1min 23s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 4543 (apache2)
    Tasks: 55 (limit: 2135)
   Memory: 5.5M
      CPU: 175ms
     CGroup: /system.slice/apache2.service
             ├─4543 /usr/sbin/apache2 -k start
             ├─4544 /usr/sbin/apache2 -k start
             └─4545 /usr/sbin/apache2 -k start

Oct 30 00:12:41 ubuntu-agent systemd[1]: Starting The Apache HTTP Server...
Oct 30 00:12:41 ubuntu-agent apachectl[4542]: AH00558: apache2: Could not reli>
Oct 30 00:12:41 ubuntu-agent systemd[1]: Started The Apache HTTP Server.
```

B3: Cập nhật cấu hình tệp ossec.conf

Điều này cho phép máy Agent Ubuntu giám sát log truy cập của máy chủ Web Apache

Thêm đoạn script sau và lưu lại:

```
#sql_injection
<ossec_config>
  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/access.log</location>
  </localfile>
</ossec_config>
```

```

GNU nano 6.2                               /var/ossec/etc/ossec.conf *

<localfile>
  <log_format>syslog</log_format>
  <location>/var/ossec/logs/active-responses.log</location>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/dpkg.log</location>
</localfile>

</ossec_config>

#sql_injection
<ossec_config>
  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/access.log</location>
  </localfile>
</ossec_config>

^G Help          ^O Write Out  ^W Where Is   ^K Cut           ^T Execute      ^C Location
^X Exit         ^R Read File  ^\ Replace    ^U Paste        ^J Justify      ^/ Go To Line

```

B4: Khởi động lại dịch vụ wazuh agent để áp dụng các thay đổi cấu hình
systemctl restart wazuh-agent.service

B5: Ở máy attacker nhập lệnh tấn công

curl -XGET "http://192.168.198.148/users/?id=SELECT+*+FROM+users";

```

[~]# curl -XGET "http://192.168.198.148/users/?id=SELECT+*+FROM+users";
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.198.148 Port 80</address>
</body></html>

```

Kết quả: Đây là một cảnh báo về cuộc tấn công SQL Injection có:

- Rule ID là 31103 và level là 7

Sep 30,					
> 2024 @					
00:42:23.1	T1190	Initial Access	SQL injection attempt.	7	31103

- Chúng ta có thể thấy được địa chỉ nguồn của cuộc tấn công này là: 192.168.198.129

data.protocol	GET
data.srcip	192.168.198.129
data.url	/users/?id=SELECT+*+FROM+users
decoder.name	web-accesslog
full_log	192.168.198.129 - - [30/Sep/2024:00:42:22 +0700] "GET /users/?id=SELECT+*+FROM+users HTTP/1.1" 404 438 "-" "curl/8.8.0"

2.6 Cấu hình Wazuh chặn địa chỉ IP độc hại truy cập đến Web Server

- *Đối với máy Agent Ubuntu*

Bước 1: Cấu hình Wazuh Agent: cấu hình tệp ossec.conf: theo dõi nhật ký truy cập Apache

nano /var/ossec/etc/ossec.conf

Ctrl+W để search: <localfile>, xong nhấn Enter và thêm đoạn script sau:

```
<localfile>
<log_format>syslog</log_format>
<location>/var/log/apache2/access.log</location>
</localfile>
```

Ctrl+X nhấn y, Enter để lưu

```
GNU nano 4.8          /var/ossec/etc/ossec.conf      Modified
<enabled>yes</enabled>
<interval>5m</interval>
<max_eps>10</max_eps>
</synchronization>
</syscheck>

<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/apache2/access.log</location>
</localfile>

<localfile>
  <log_format>full_command</log_format>
```

Key Bindings:

- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut Text
- ^J Justify
- ^C Cur Pos
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste Text
- ^T To Spell
- ^ Go To Line

Hình 24. Cấu hình tệp ossec.conf

Script này là khôi lệnh theo dõi nhật ký truy cập Apache

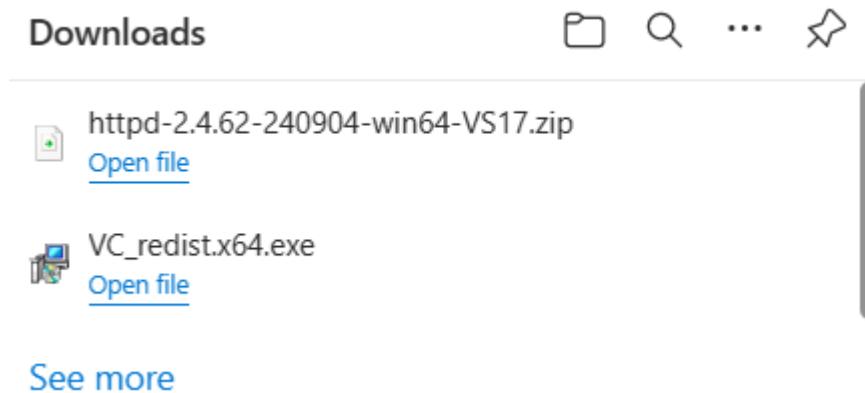
Bước 5: Khởi động lại Wazuh Agent để áp dụng các thay đổi
systemctl restart wazuh-agent.service

- Đối với máy Agent Windows Server

Bước 1: Cài đặt Web Apache

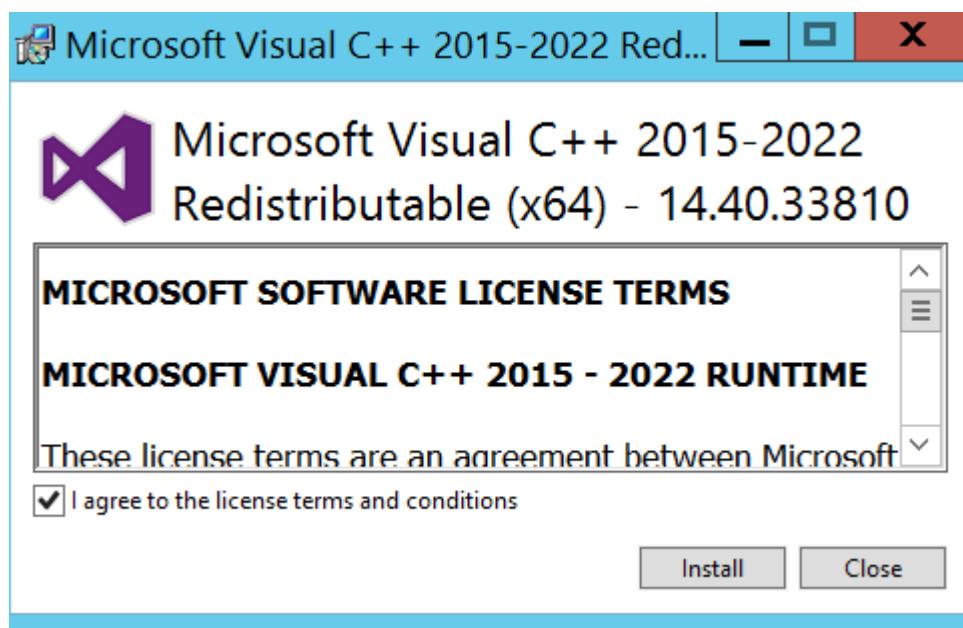
Cài Visual C++ Redistributable Visual Studio và file apache theo link sau:

<https://www.apachelounge.com/download/>



Hình 25. Download apache và visual code redistributable cho Windows Server 2012

Cài đặt vc_redist trước

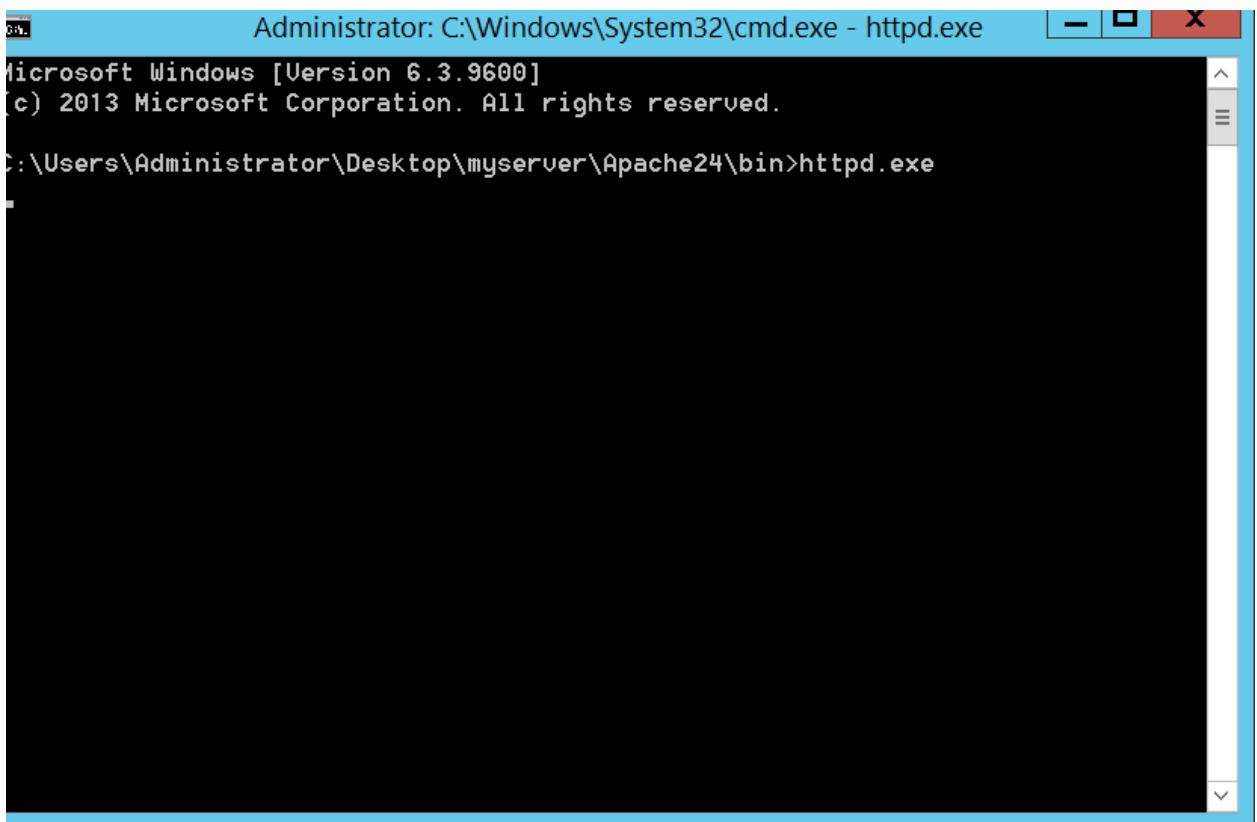


Hình 26. Cài đặt vc_redist

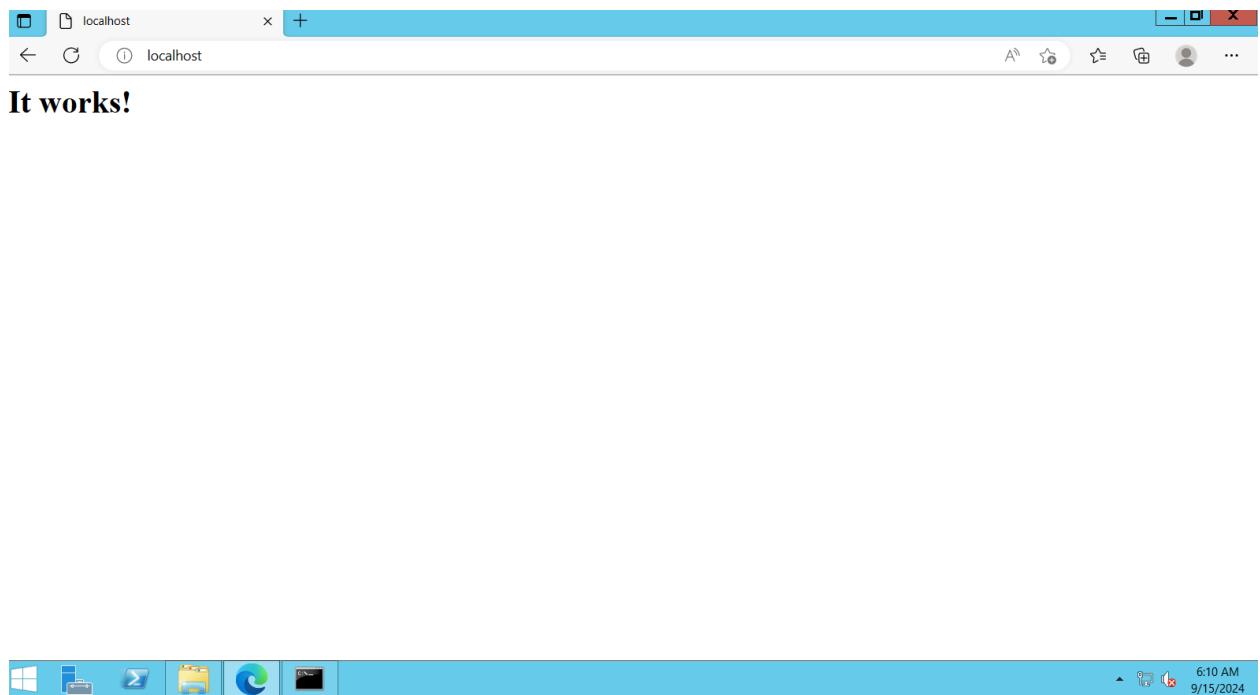
Tiếp theo cài đặt apache theo link hướng dẫn sau:

<https://www.bing.com/videos/riverview/relatedvideo?q=web+apache+download+for+windows+server+2012&mid=929BC09D85BB9ECB2E7C929BC09D85BB9ECB2E7C&FORM=VIRE>

Bước 2: Chạy và kiểm tra trạng thái Apache



Hình 27. Chạy httpd.exe



Hình 28. Web apache đang hoạt động trên Windows Server 2012

Bước 3: Cấu hình Wazuh Agent

The screenshot shows a Windows Notepad window titled "ossec.conf - Notepad". The file contains XML configuration code for an OSSEC agent. A specific section of the code, which defines log analysis for the Apache access log, is highlighted with a blue selection bar. The highlighted code is as follows:

```
<!-- Log analysis -->
<localfile>
  <location>Application</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>C:\Users\Administrator\Desktop\myserver\Apache24\logs\access.log</location>
</localfile>

<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
```

Script này là khôi lệnh theo dõi nhật ký truy cập Apache

Bước 4: Khởi động lại Wazuh agent để áp dụng các thay đổi

Mở PowerShell

Restart-Service -Name WazuhSvc

Hoặc

Restart-Service -Name wazuh

- *Đối với máy Wazuh Server*

Bước 1: Cài đặt tiện ích wget

apt update && apt install -y wget

Bước 2: Tải cơ sở dữ liệu của alienVault IP

```
 wget https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienVault_reputation.ipset -O /var/ossec/etc/lists/alienVault_reputation.ipset
```

Bước 3: Thêm địa chỉ IP attacker vào cơ sở dữ liệu alienVault IP

```
 echo "192.168.198.129" >> /var/ossec/etc/lists/alienVault_reputation.ipset
```

192.168.198.129 là địa chỉ của attacker

Bước 4: Tải script để chuyển đổi định dạng tệp

```
 wget https://wazuh.com/resources/iplist-to-cdblist.py -O /tmp/iplist-to-cdblist.py
```

Bước 5: Chuyển đổi định dạng alienVault .ipset sang định dạng .gpg

```
/var/ossec/framework/python/bin/python3 /tmp/iplist-to-cdblist.py  
/var/ossec/etc/lists/alienVault_reputation.ipset /var/ossec/etc/lists/blacklist-alienVault
```

```
root@ubuntu-server:~# /var/ossec/framework/python/bin/python3 /tmp/iplist-to-cdblist.py /var/ossec/etc/lists/alienVault_reputation.ipset /var/ossec/etc/lists/blacklist-alienVault  
[ /var/ossec/etc/lists/alienVault_reputation.ipset ] -> [ /var/ossec/etc/lists/blacklist-alienVault ]  
root@ubuntu-server:~#
```

Bước 6: Xóa tệp không cần thiết

```
rm -rf /var/ossec/etc/lists/alienVault_reputation.ipset
```

```
rm -rf /tmp/iplist-to-cdblist.py
```

Bước 7: Gán quyền cho tệp /var/ossec/etc/lists/blacklist-alienVault

```
chown wazuh:wazuh /var/ossec/etc/lists/blacklist-alienVault
```

Bước 8: Kích hoạt tập lệnh active response

```
nano /var/ossec/etc/rules/local_rules.xml
```

```

<group name="attack">
    <rule id="100100" level="10">
        <if_group>web|attack|attacks</if_group>
        <list field="srcip" lookup="address_match_key">etc/lists/blacklist-
alienvault</list>
        <description>IP address found in AlientVault reputation
database</description>
    </rule>
</group>

```

```

GNU nano 6.2                               /var/ossec/etc/rules/local_rules.xml *
<!--
Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
-->
<rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
</rule>

</group>

<group name="attack">
    <rule id="100100" level="10">
        <if_group>web|attack|attacks</if_group>
        <list field="srcip" lookup="address_match_key">etc/lists/blacklist-alienvault</list>
        <description>IP address found in AlientVault reputation database</description>
    </rule>
</group>

^G Help          ^O Write Out      ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File      ^Y Replace       ^U Paste          ^J Justify       ^/ Go To Line

```

Bước 9: Cấu hình Wazuh Server

Thêm quy tắc để kích hoạt tập lệnh phản hồi vào tệp bộ quy tắc local rules

sudo nano /var/ossec/etc/ossec.conf

Thêm:

```
<list>etc/lists/blacklist-alienVault</list>
```

```
<ruleset>
  <!-- Default ruleset -->
  <decoder_dir>ruleset/decoders</decoder_dir>
  <rule_dir>ruleset/rules</rule_dir>
  <rule_exclude>0215-policy_rules.xml</rule_exclude>
  <list>etc/lists/audit-keys</list>
  <list>etc/lists/amazon/aws-eventnames</list>
  <list>etc/lists/security-eventchannel</list>
  <list>etc/lists/blacklist-alienVault</list>

  <!-- User-defined ruleset -->
  <decoder_dir>etc/decoders</decoder_dir>
  <rule_dir>etc/rules</rule_dir>
```

G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
X Exit ^R Read File ^A Replace ^U Paste Text ^T To Spell ^L Go To Line

#For Ubuntu endpoint

```
<ossec_config>

  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>120</timeout>
  </active-response>
</ossec_config>
```

#Script này ngăn kết nối mạng đến từ điểm cuối của attacker Ubuntu trong 120s

#For Windows endpoint

```
<ossec_config>
```

```

<active-response>

<command>netsh</command>

<location>local</location>

<rules_id>100100</rules_id>

<timeout>120</timeout>

</active-response>

</ossec_config>

```

#Script này sẽ chặn IP attacker Windows trong 120s

```

GNU nano 6.2                               /var/ossec/etc/ossec.conf
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/dpkg.log</location>
</localfile>

</ossec_config>

#For Ubuntu endpoint
<ossec_config>
  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>120</timeout>
  </active-response>
</ossec_config>

#For Windows endpoint
<ossec_config>
  <active-response>
    <command>netsh</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>120</timeout>
  </active-response>
</ossec_config>

^G Help          ^O Write Out      ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File     ^L Replace       ^U Paste         ^J Justify       ^/ Go To Line

```

Bước 10: Khởi động lại Wazuh Server để áp dụng các thay đổi

systemctl restart wazuh-manager.service

- **Trên máy attacker truy cập vào máy chủ web apache agent ubuntu**
Giả lập tấn công bằng lệnh sau:
curl http://192.168.198.148
192.168.198.148: địa chỉ IP web apache agent ubuntu

Kết quả: Vào wazuh server để xem cảnh báo:

- Phát hiện địa chỉ IP truy cập vào web server

Security Alerts		Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
Sep 30, 2024 @ 22:20:09.772					IP address found in AlienVault reputation database	10	100100
Table JSON Rule							
	@timestamp		2024-09-30T15:20:09.772Z				
	_id		jm-EQ5lB093sm6xshpgT				
	agent.id		001				
	agent.ip		192.168.198.148				
	agent.name		ubuntu-agent				
	data.id		200				
	data.protocol		GET				
	data.srcip		192.168.198.129				
	data.url		/				
	decoder.name		web-accesslog				
	full_log		192.168.198.129 - - [30/Sep/2024:22:20:09 +0700] "GET / HTTP/1.1" 200 10926 "-" "curl/8.8.0"				
	id		1727709609.4213246				
	input.type		log				
	location		/var/log/apache2/access.log				
	manager.name		anh-attt				
	rule.description		IP address found in AlienVault reputation database				

- Sau đó wazuh đã block địa chỉ IP đó trong 120s

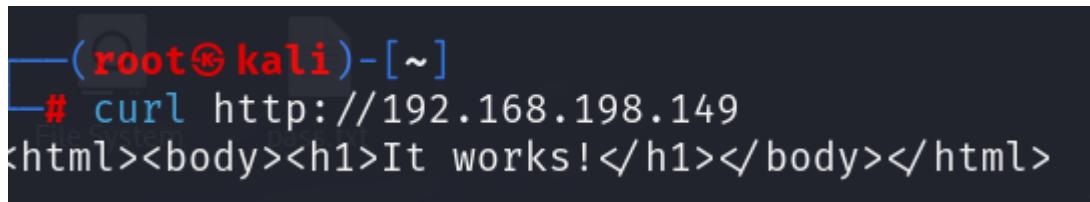
Sep 30, 2024 @ > 22:22:10.7 80	Host Unblocked by firewall-drop Active Response	3	652
Sep 30, 2024 @ > 22:20:10.6 37	Host Blocked by firewall-drop Active Response	3	651

- Trên máy attacker truy cập vào máy chủ web apache agent windows server

Giả lập tấn công bằng lệnh sau:

```
curl http://192.168.198.149
```

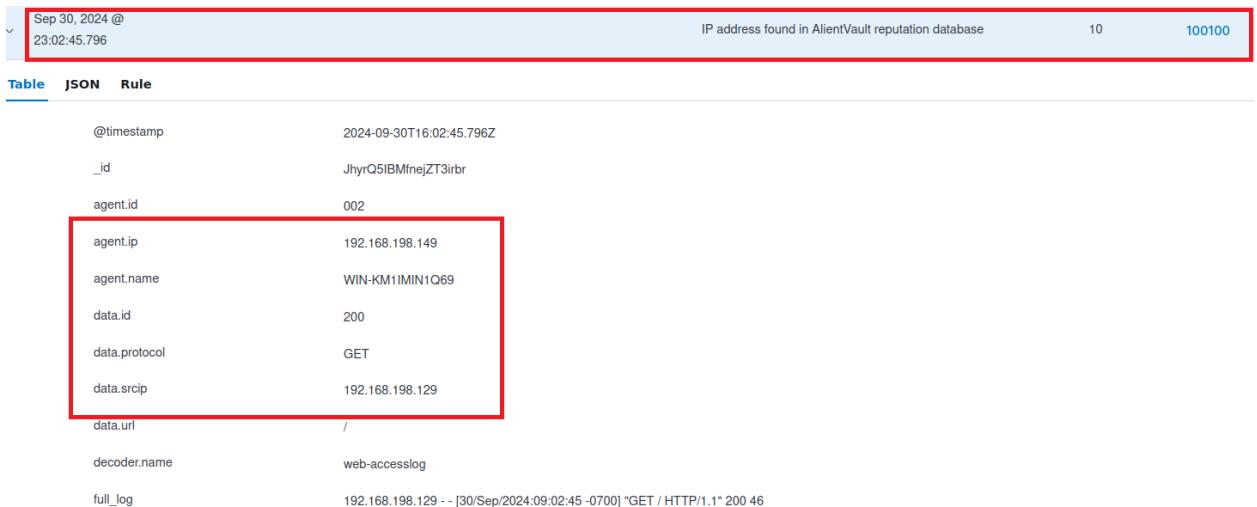
192.168.198.149: địa chỉ ip web apache agent windows server



```
(root㉿kali)-[~]
# curl http://192.168.198.149
<html><body><h1>It works!</h1></body></html>
```

Kết quả: Vào wazuh server để xem cảnh báo:

- Phát hiện địa chỉ IP truy cập vào web server



Sep 30, 2024 @ 23:02:45.796		IP address found in AlienVault reputation database	10	100100
Table	JSON	Rule		
		@timestamp	2024-09-30T16:02:45.796Z	
		_id	JhyrQ5IBMfnejZT3irbr	
		agent.id	002	
		agent.ip	192.168.198.149	
		agent.name	WIN-KM1IMIN1Q69	
		data.id	200	
		data.protocol	GET	
		data.srcip	192.168.198.129	
		data.url	/	
		decoder.name	web-accesslog	
		full_log	192.168.198.129 -- [30/Sep/2024:09:02:45 -0700] "GET / HTTP/1.1" 200 46	

- Sau đó wazuh đã block địa chỉ IP đó trong 120s

Sep 30, 2024 @ 23:02:47.273	Active response: active-response/bin/netsh.exe - add	3	657
Table JSON Rule			
@timestamp	2024-09-30T16:02:47.273Z		
_id	JxyrQ5IBMfnejZT3jrbJ		
agent.id	002		
agent.ip	192.168.198.149		
agent.name	WIN-KM1IMIN1Q69		
data.command	add		
data.origin.module	wazuh-execod		
data.origin.name	node01		
data.parameters.alert.agent.id	002		
data.parameters.alert.agent.ip	192.168.198.149		
data.parameters.alert.agent.name	WIN-KM1IMIN1Q69		
data.parameters.alert.data.id	200		
Sep 30, 2024 @ 23:04:48.323	Active response: active-response/bin/netsh.exe - delete	3	657
Table JSON Rule			
@timestamp	2024-09-30T16:04:48.323Z		
_id	OxytQ5IBMfnejZT3d7bx		
agent.id	002		
agent.ip	192.168.198.149		
agent.name	WIN-KM1IMIN1Q69		
data.command	delete		
data.origin.module	wazuh-execod		
data.origin.name	node01		
data.parameters.alert.agent.id	002		
data.parameters.alert.agent.ip	192.168.198.149		
data.parameters.alert.agent.name	WIN-KM1IMIN1Q69		
data.parameters.alert.data.id	200		
data.parameters.alert.data.protocol	GET		
data.parameters.alert.data.script	192.168.198.129		

2.7 Tích hợp VirusTotal để phát hiện và xóa các phần mềm độc hại

Kịch bản: Không sử dụng máy attacker. Trong trường hợp này chúng ta sẽ giám sát tính toàn vẹn của tệp và dùng API VirusTotal để quét các tệp đó. Sau đó, ta sẽ cấu hình Wazuh Server để kích hoạt lệnh phản hồi và xóa các tệp mà VirusTotal phát hiện là độc hại. Để sử dụng VirusTotal, chúng ta cần khóa API VirusTotal. Trong trường hợp sử dụng này để xác thực Wazuh Server với API VirusTotal.

1.Kích bản tải malware ở máy Agent Ubuntu

- Ở máy Agent Ubuntu

B1: Cấu hình file ossec.conf. Cấu hình trong khối <syscheck> để thay đổi định dạng giám sát thư mục root theo thời gian thực.

```
nano /var/ossec/etc/ossec.conf
```

Thêm:

```
<directories realtime="yes">/root</directories>
```

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
  <directories realtime="yes">/root</directories>

  <!-- Frequency that syscheck is executed default every 12 hours -->
  <frequency>43200</frequency>

  <scan_on_start>yes</scan_on_start>

  <!-- Directories to check (perform all possible verifications) -->
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>
```

B2: Để xử lý đầu vào json từ tập lệnh active response, ta cần cài đặt jq cho máy Agent

```
apt install jq
```

B3: Tạo file [/var/ossec/active-response/bin/remove-threat.sh](#)

Để kích hoạt phản hồi xóa file độc hại từ endpoint

Thêm đoạn script sau:

```
#!/bin/bash
```

```
LOCAL=`dirname $0`;
```

```
cd $LOCAL
```

```
cd ../
```

```
PWD=`pwd`
```

```

read INPUT_JSON
FILENAME=$(echo $INPUT_JSON | jq -r
.parameters.alert.data.virustotal.source.file)
COMMAND=$(echo $INPUT_JSON | jq -r .command)
LOG_FILE="${PWD}/..//logs/active-responses.log"

#----- Analyze command -----
if [ ${COMMAND} = "add" ]
then
# Send control message to execd
printf '{"version":1,"origin":{"name":"remove-threat","module":"active-
response"},"command":"check_keys", "parameters":{"keys":[]}}\n'

read RESPONSE
COMMAND2=$(echo $RESPONSE | jq -r .command)
if [ ${COMMAND2} != "continue" ]
then
echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Remove threat
active response aborted" >> ${LOGFILE}
exit 0;
fi
fi

# Removing file
rm -f $FILENAME
if [ $? -eq 0 ]; then
echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Successfully
removed threat" >> ${LOGFILE}
else

```

```

echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Error removing
threat" >> ${LOG_FILE}
fi

```

```
exit 0;
```



```

GNU nano 6.2          /var/ossec/active-response/bin/remove-threat.sh *
#!/bin/bash

LOCAL=`dirname $0`;
d $LOCAL
d ../

WD=`pwd`

read INPUT_JSON
FILENAME=$(echo $INPUT_JSON | jq -r .parameters.alert.data.virustotal.source.file)
COMMAND=$(echo $INPUT_JSON | jq -r .command)
LOG_FILE="${PWD}/../logs/active-responses.log"

----- Analyze command -----
if [ ${COMMAND} = "add" ]
then
# Send control message to execd
printf '{"version":1,"origin":{"name":"remove-threat","module":"active-response"},"command":"'${COMMAND}'"}'
read RESPONSE
COMMAND2=$(echo $RESPONSE | jq -r .command)
if [ ${COMMAND2} != "continue" ]
then
echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Remove threat active response aborted" >> ${LOG_FILE}
exit 0;
fi
i

: Removing file
rm -f $FILENAME
if [ $? -eq 0 ]; then
echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Successfully removed threat" >> ${LOG_FILE}
else
echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Error removing threat" >> ${LOG_FILE}
fi
exit 0;else

```

B4: Thay đổi quyền sở hữu và quyền của tệp </var/ossec/active-response/bin/remove-threat.sh>

```
sudo chmod 750 /var/ossec/active-response/bin/remove-threat.sh
```

```
sudo chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh
```

B5: Khởi động lại Wazuh Agent để áp dụng các thay đổi
sudo systemctl restart wazuh-agent

- **Ở máy Wazuh Server**

B1: Thêm các quy tắc sau vào tệp local_rules.xml

Các quy tắc này được thêm vào để cảnh báo về những thay đổi trong thư mục root được phát hiện khi FIM quét

```
nano /var/ossec/etc/rules/local_rules.xml
```

Thêm đoạn script:

```
<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,">
    <!-- Rules for Linux systems -->
    <rule id="100200" level="7">
        <if_sid>550</if_sid>
        <field name="file">/root</field>
        <description>File modified in /root directory.</description>
    </rule>
    <rule id="100201" level="7">
        <if_sid>554</if_sid>
        <field name="file">/root</field>
        <description>File added to /root directory.</description>
    </rule>
</group>
```

```

GNU nano 6.2                               /var/ossec/etc/rules/local_rules.xml *

<!-- Example -->
<group name="local,syslog,sshd,">

    <!--
    Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
    -->
    <rule id="100001" level="5">
        <if_sid>5716</if_sid>
        <srcip>1.1.1.1</srcip>
        <description>sshd: authentication failed from IP 1.1.1.1.</description>
        <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
    </rule>

</group>

<group name="attack">
    <rule id="100100" level="10">
        <if_group>web|attack|attacks</if_group>
        <list field="srcip" lookup="address_match_key">etc/lists/blacklist-alienVault</list>
        <description>IP address found in AlienVault reputation database</description>
    </rule>
</group>

<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,">
    <!-- Rules for Linux systems -->
    <rule id="100200" level="7">
        <if_sid>550</if_sid>
        <field name="file">/root</field>
        <description>File modified in /root directory.</description>
    </rule>
    <rule id="100201" level="7">
        <if_sid>554</if_sid>
        <field name="file">/root</field>
        <description>File added to /root directory.</description>
    </rule>
</group>

```

B2: Thêm script quy tắc vào file ossec.conf vào máy wazuh server
 nano /var/ossec/etc/ossec.conf

Để kích hoạt tích hợp VirusTotal, ta cần thay thế VirusTotal API Key mặc định bằng khóa API Key VirusTotal của mình. Điều này kích hoạt truy vấn VirusTotal bất cứ khi nào.

Thêm :

```
#VirusTotal
<ossec_config>
    <integration>
```

```

<name>virustotal</name>
<api_key>7948aa22354fe9532c6b2d1c13e46fa715376c1fba0</api_key> <!--
- Replace with your VirusTotal API key -->
<rule_id>100200,100201</rule_id>
<alert_format>json</alert_format>
</integration>
</ossec_config>
```

```

GNU nano 6.2          /var/ossec/etc/ossec.conf *
#For Windows endpoint
<ossec_config>
  <active-response>
    <command>netsh</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>120</timeout>
  </active-response>
</ossec_config>

#VirusTotal
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>7948aa22354fe9532c6b2d185392c672a5749c13e46fa715376c1fba0</api_key>
    <rule_id>100200,100201</rule_id>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

Tiếp tục thêm:

```
<ossec_config>
  <command>
    <name>remove-threat</name>
```

```

<executable>remove-threat.sh</executable>
<timeout_allowed>no</timeout_allowed>
</command>
<active-response>
<disabled>no</disabled>
<command>remove-threat</command>
<location>local</location>
<rules_id>87105</rules_id>
</active-response>
</ossec_config>
```

```

#VirusTotal
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>7948aa22354fe9532c6b2d185392c672a5</api_key>
    <rule_id>100200,100201</rule_id>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>

<ossec_config>
  <command>
    <name>remove-threat</name>
    <executable>remove-threat.sh</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>
  <active-response>
    <disabled>no</disabled>
    <command>remove-threat</command>
    <location>local</location>
    <rules_id>87105</rules_id>
  </active-response>
</ossec_config>
```

Điều này cho phép trực tiếp phản hồi và kích hoạt tập lệnh remove.sh khi VirusTotal gán cờ một tệp là độc hại.

B3: Thêm quy tắc sau vào tệp local_rules.xml vào máy Wazuh server

nano /var/ossec/etc/rules/local_rules.xml

```
<group name="virustotal,">
    <rule id="100092" level="12">
        <if_sid>657</if_sid>
        <match>Successfully removed threat</match>
        <description>$(parameters.program) removed threat located at
$(parameters.alert.data.virustotal.source.file)</description>
    </rule>
    <rule id="100093" level="12">
        <if_sid>657</if_sid>
        <match>Error removing threat</match>
        <description>Error removing threat located at
$(parameters.alert.data.virustotal.source.file)</description>
    </rule>
</group>
<!-- Rules for Linux systems -->
<rule id="100200" level="7">
    <if_sid>550</if_sid>
    <field name="file">/root</field>
    <description>File modified in /root directory.</description>
</rule>
<rule id="100201" level="7">
    <if_sid>554</if_sid>
    <field name="file">/root</field>
    <description>File added to /root directory.</description>
</rule>
</group>

<group name="virustotal,">
    <rule id="100092" level="12">
        <if_sid>657</if_sid>
        <match>Successfully removed threat</match>
        <description>$(parameters.program) removed threat located at $(parameters.alert.data.vir
    </rule>
    <rule id="100093" level="12">
        <if_sid>657</if_sid>
        <match>Error removing threat</match>
        <description>Error removing threat located at $(parameters.alert.data.virustotal.source.
    </rule>
</group>
```

Khối lệnh này cảnh báo về kết quả phản hồi đang hoạt động

B4: Khởi động lại máy Wazuh server để áp dụng các thay đổi
systemctl restart wazuh-manager.service

Tại máy Agent Ubuntu tải xuống một tệp độc hại thử nghiệm vào thư mục root

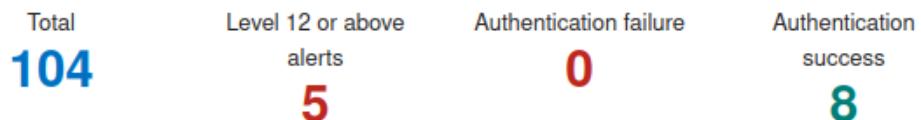
curl -LO <https://secure.eicar.org/eicar.com> && ls -lah eicar.com

```
root@ubuntu-agent:~# curl -LO https://secure.eicar.org/eicar.com && ls -lah eicar.com
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload Total Spent   Spent    Left  Speed
100      68  100      68    0     0       17      0  0:00:04  0:00:03  0:00:01   17
-rw-r--r-- 1 root root 68 0.0. 1 14:46 eicar.com
root@ubuntu-agent:~#
```

Kết quả:

Security Alerts					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
Oct 1, 2024 @ 16:40:10.17 6			active-response/bin/remove-threat.sh removed threat located at /root/eicar.com	12	100092
Oct 1, 2024 @ 16:40:08.99 1	T1070.004 T1485	Defense Evasion, Impact	File deleted.	7	553
Oct 1, 2024 @ 16:40:08.71 0	T1203	Execution	VirusTotal: Alert - /root/eicar.com - 63 engines detected this file	12	87105

- Phát hiện cảnh báo file độc hại với Rule ID là 87105 và level là 12. Cấp cảnh báo 12 này cho ta biết là event rất quan trọng



- Có 63 công cụ đã phát hiện ra file này là file độc hại

location	virustotal
manager.name	anh-attt
rule.description	VirusTotal: Alert - /root/eicar.com - 63 engines detected this file
rule.firedtimes	1

- Và Wazuh đã xóa file độc hại này ngay sau đó

Oct 1, 2024				
@				
16:40:08.99	T1070.004 T1485	Defense Evasion, Impact		File deleted.
1				

Table JSON Rule

@timestamp	2024-10-01T09:40:08.991Z
_id	JupzR5IB9Dlp8o-apYEI
agent.id	001
agent.ip	192.168.198.148
agent.name	ubuntu-agent
decoder.name	syscheck_deleted
full_log	File '/root/eicar.com' deleted Mode: realtime

- Cuối cùng là hiển thị phản hồi xóa threat nằm ở /root

location	/var/ossec/logs/active-responses.log
manager.name	anh-attt
rule.description	active-response/bin/remove-threat.sh removed threat located at /root/eicar.com
rule.firedtimes	1

2. Kích bản tải malware ở máy Agent Windows Server

- Ở máy Agent Windows Server

B1: Vào tệp ossec.conf

Tìm khôi `<syscheck>`: đảm bảo nó được đặt `<disabled>no</disabled>`

The screenshot shows a Notepad window titled "ossec.conf - Notepad". The file contains XML configuration code. A specific line of code is highlighted with a blue rectangle: `<disabled>no</disabled>`. This line is located within the `<syscheck>` section, which is preceded by a multi-line comment and followed by another multi-line comment.

```

<!-- Security Configuration Assessment -->
<sca>
    <enabled>yes</enabled>
    <scan_on_start>yes</scan_on_start>
    <interval>12h</interval>
    <skip_nfs>yes</skip_nfs>
</sca>

<!-- File integrity monitoring -->
<syscheck>

    <disabled>no</disabled>

    <!-- Frequency that syscheck is executed default every 12 hours
    <frequency>43200</frequency>

    <!-- Default files to be monitored. -->
    <directories recursion_level="0" restrict="regedit.exe$|system
    <directories recursion_level="0" restrict="at.exe$|attrib.exe$>
    <directories recursion_level="0" restrict=">%WINDIR%\SysNative\drivers\etc>
    <directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%>

```

⇒ Điều này cho phép module Wazuh FIM giám sát các thay đổi thư mục

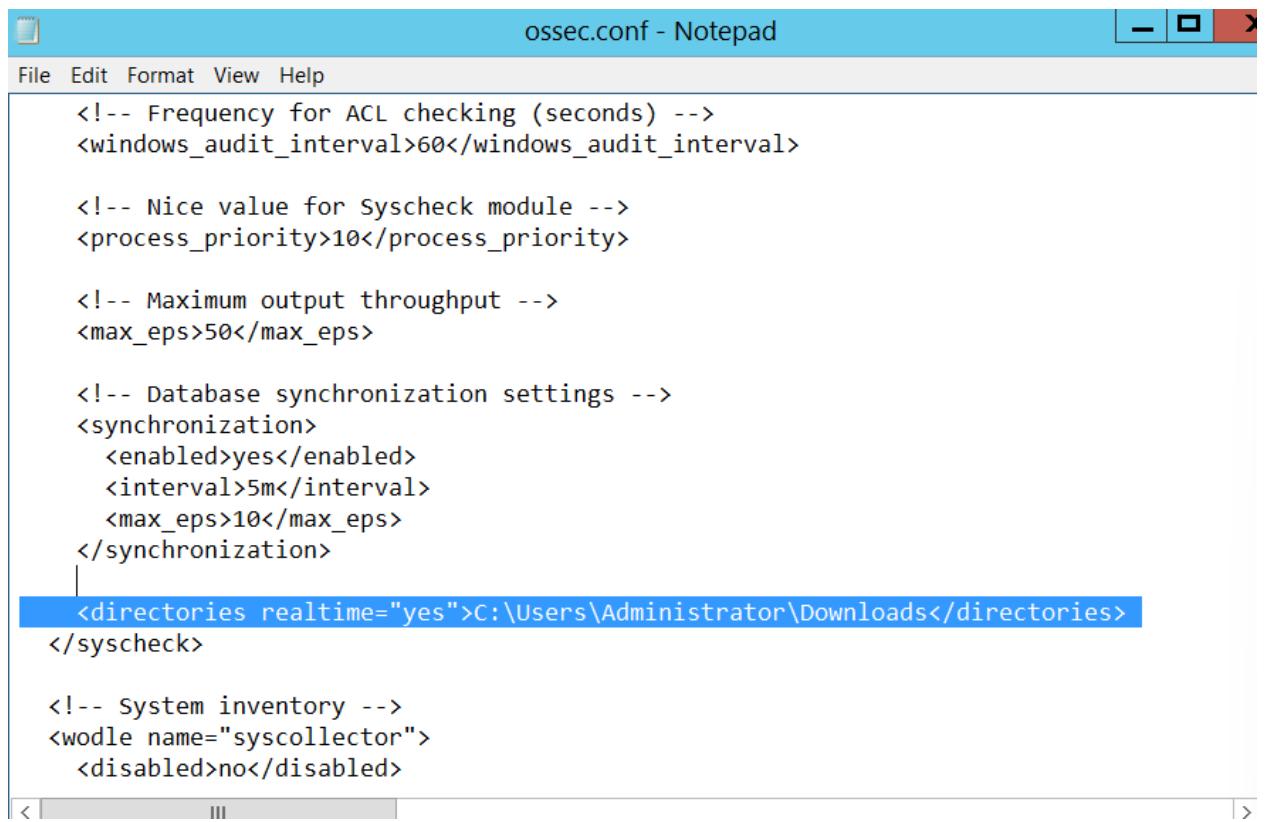
Thêm script trong khôi `<syscheck>` để định cấu hình 1 thư mục để được theo dõi theo thời gian thực realtime.

Trong TH này, định cấu hình Wazuh để giám sát thư mục :

```

<directories
realtime="yes">C:\Users\Administrator\Downloads</directories>

```



The screenshot shows a Windows Notepad window titled "ossec.conf - Notepad". The file contains XML configuration code for the Ossec host agent. The code includes settings for ACL checking frequency, process priority, maximum output throughput, database synchronization, real-time directory monitoring, and system inventory collection. A specific section for monitoring the "C:\Users\Administrator\Downloads" directory in real-time is highlighted with a blue selection bar.

```
<!-- Frequency for ACL checking (seconds) -->
<windows_audit_interval>60</windows_audit_interval>

<!-- Nice value for Syscheck module -->
<process_priority>10</process_priority>

<!-- Maximum output throughput -->
<max_eps>50</max_eps>

<!-- Database synchronization settings -->
<synchronization>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <max_eps>10</max_eps>
</synchronization>

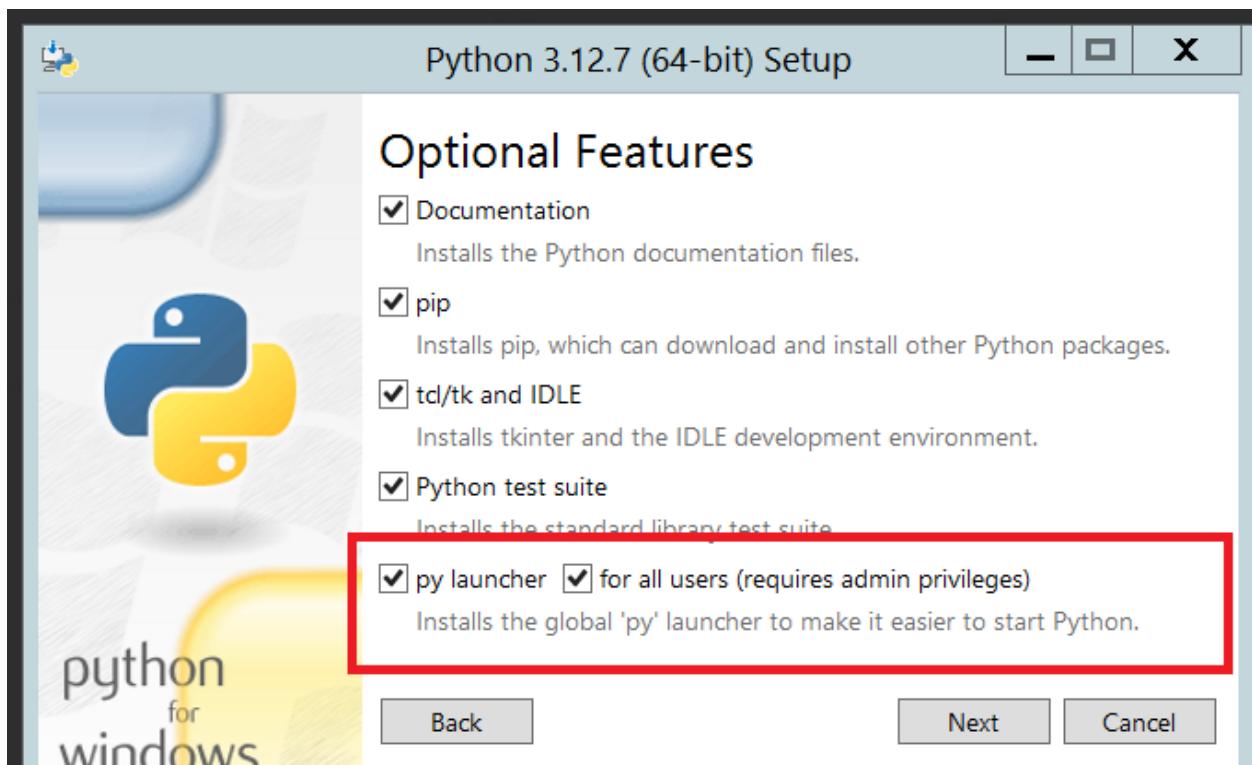
<!-- Directories monitored in real-time -->
<directories realtime="yes">C:\Users\Administrator\Downloads</directories>
</syscheck>

<!-- System inventory -->
<wodle name="syscollector">
    <disabled>no</disabled>
```

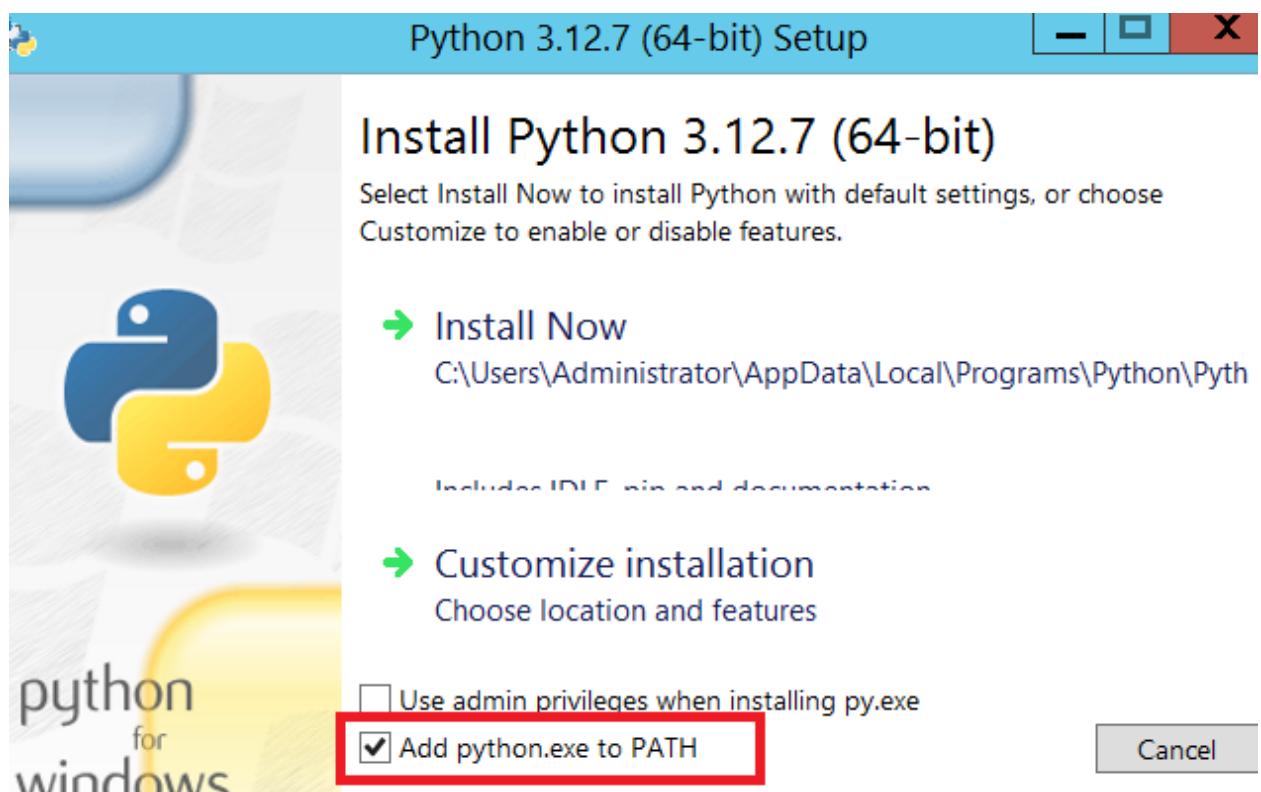
B2: Tải xuống trình cài đặt thực thi Python từ [trang web chính thức của Python](#).

B3: Chạy trình cài đặt Python sau khi tải xuống. Đảm bảo kiểm tra các box sau:

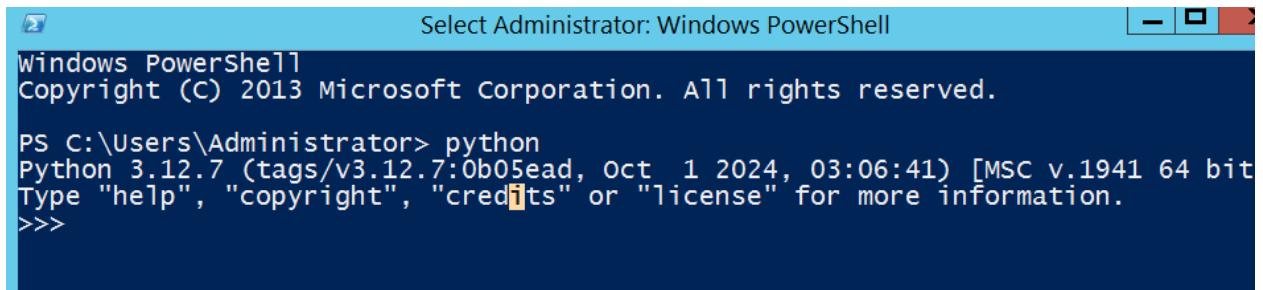
- `Install launcher for all users`



- Add Python 3.X to PATH (Điều này đặt trình thông dịch vào đường dẫn thực thi)



Kiểm tra xem python đã cài thành công chưa

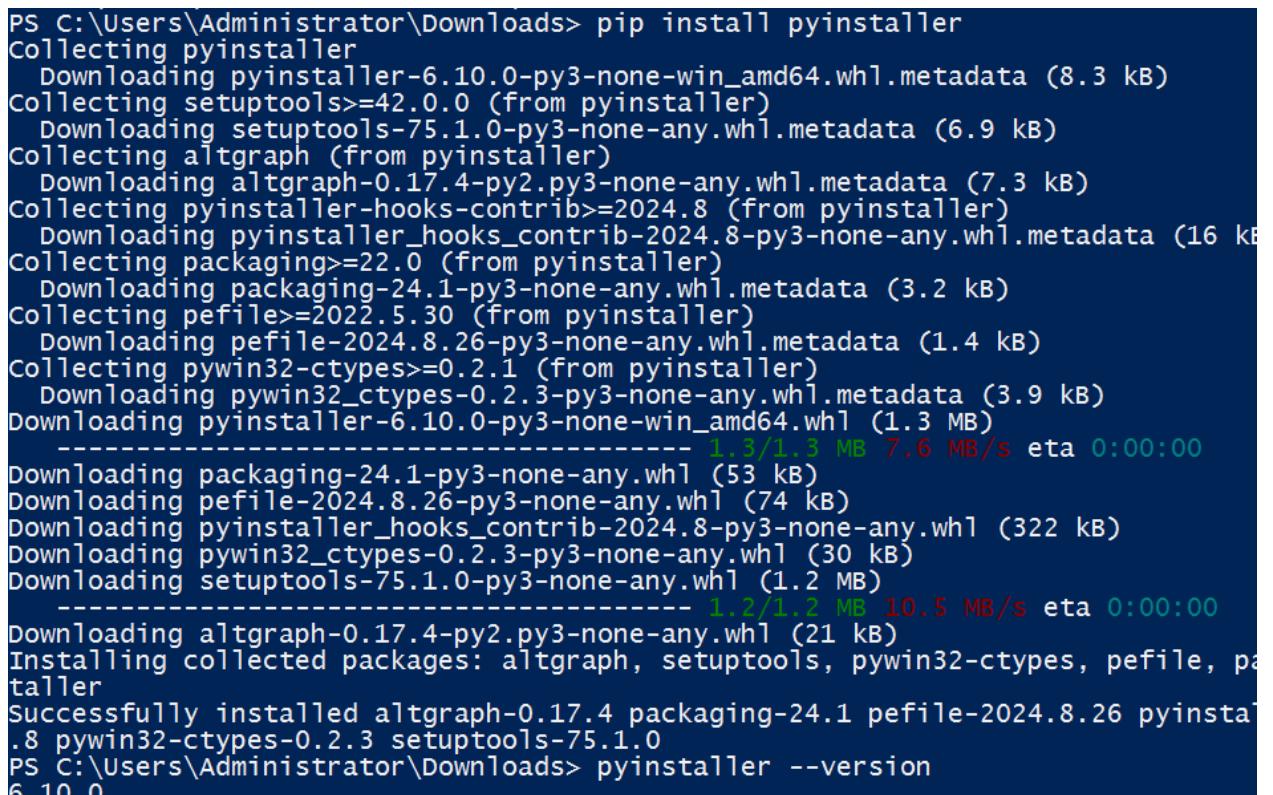


```
Select Administrator: Windows PowerShell  
Windows PowerShell  
Copyright (C) 2013 Microsoft Corporation. All rights reserved.  
PS C:\Users\Administrator> python  
Python 3.12.7 (tags/v3.12.7:0b05ead, Oct 1 2024, 03:06:41) [MSC v.1941 64 bit  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

B4: Sau khi Python hoàn tất quá trình cài đặt, hãy mở terminal PowerShell dành cho quản trị viên và sử dụng pip để cài đặt PyInstaller:

```
pip install pyinstaller
```

```
pyinstaller --version
```



```
PS C:\Users\Administrator\Downloads> pip install pyinstaller  
Collecting pyinstaller  
  Downloading pyinstaller-6.10.0-py3-none-win_amd64.whl.metadata (8.3 kB)  
Collecting setuptools>=42.0.0 (from pyinstaller)  
  Downloading setuptools-75.1.0-py3-none-any.whl.metadata (6.9 kB)  
Collecting altgraph (from pyinstaller)  
  Downloading altgraph-0.17.4-py2.py3-none-any.whl.metadata (7.3 kB)  
Collecting pyinstaller-hooks-contrib>=2024.8 (from pyinstaller)  
  Downloading pyinstaller_hooks_contrib-2024.8-py3-none-any.whl.metadata (16 kB)  
Collecting packaging>=22.0 (from pyinstaller)  
  Downloading packaging-24.1-py3-none-any.whl.metadata (3.2 kB)  
Collecting pefile>=2022.5.30 (from pyinstaller)  
  Downloading pefile-2024.8.26-py3-none-any.whl.metadata (1.4 kB)  
Collecting pywin32-ctypes>=0.2.1 (from pyinstaller)  
  Downloading pywin32_ctypes-0.2.3-py3-none-any.whl.metadata (3.9 kB)  
Downloading pyinstaller-6.10.0-py3-none-win_amd64.whl (1.3 MB)  
----- 1.3/1.3 MB 7.6 MB/s eta 0:00:00  
Downloading packaging-24.1-py3-none-any.whl (53 kB)  
Downloading pefile-2024.8.26-py3-none-any.whl (74 kB)  
Downloading pyinstaller_hooks_contrib-2024.8-py3-none-any.whl (322 kB)  
Downloading pywin32_ctypes-0.2.3-py3-none-any.whl (30 kB)  
Downloading setuptools-75.1.0-py3-none-any.whl (1.2 MB)  
----- 1.2/1.2 MB 10.5 MB/s eta 0:00:00  
Downloading altgraph-0.17.4-py2.py3-none-any.whl (21 kB)  
Installing collected packages: altgraph, setuptools, pywin32-ctypes, pefile, pyinstaller  
Successfully installed altgraph-0.17.4 packaging-24.1 pefile-2024.8.26 pyinstaller-6.10.0  
PS C:\Users\Administrator\Downloads> pyinstaller --version
```

Ở đây, bạn sử dụng PyInstaller để chuyển đổi tập lệnh Python phản hồi đang hoạt động thành ứng dụng thực thi có thể chạy trên điểm cuối endpoint Windows.

B5: Tạo một tập lệnh active response `remove-threat.py` để xóa tệp khỏi điểm cuối Windows

Thêm đoạn script này vào file `remove-threat.py`:

```

#!/usr/bin/python3
# Copyright (C) 2015-2022, Wazuh Inc.
# All rights reserved.

import os
import sys
import json
import datetime

if os.name == 'nt':
    LOG_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-
responses.log"
else:
    LOG_FILE = "/var/ossec/logs/active-responses.log"

ADD_COMMAND = 0
DELETE_COMMAND = 1
CONTINUE_COMMAND = 2
ABORT_COMMAND = 3

OS_SUCCESS = 0
OS_INVALID = -1

class message:
    def __init__(self):
        self.alert = ""
        self.command = 0

def write_debug_file(ar_name, msg):
    with open(LOG_FILE, mode="a") as log_file:

```

```

    log_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d
%H:%M:%S')) + " " + ar_name + ": " + msg +"\n")

def setup_and_check_message(argv):

    # get alert from stdin
    input_str = ""
    for line in sys.stdin:
        input_str = line
        break

    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid input
format')
        message.command = OS_INVALID
        return message

    message.alert = data

    command = data.get("command")

    if command == "add":
        message.command = ADD_COMMAND
    elif command == "delete":
        message.command = DELETE_COMMAND
    else:
        message.command = OS_INVALID
        write_debug_file(argv[0], 'Not valid command: ' + command)

```

```
return message

def send_keys_and_check_message(argv, keys):

    # build and send message with keys
    keys_msg = json.dumps({"version": 1, "origin": {"name": argv[0], "module": "active-response"}, "command": "check_keys", "parameters": {"keys": keys}})

    write_debug_file(argv[0], keys_msg)

    print(keys_msg)
    sys.stdout.flush()

    # read the response of previous message
    input_str = ""
    while True:
        line = sys.stdin.readline()
        if line:
            input_str = line
            break

    # write_debug_file(argv[0], input_str)

    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid input format')
```

```

        return message

action = data.get("command")

if "continue" == action:
    ret = CONTINUE_COMMAND
elif "abort" == action:
    ret = ABORT_COMMAND
else:
    ret = OS_INVALID
    write_debug_file(argv[0], "Invalid value of 'command'")

return ret

def main(argv):

    write_debug_file(argv[0], "Started")

    # validate json and get command
    msg = setup_and_check_message(argv)

    if msg.command < 0:
        sys.exit(OS_INVALID)

    if msg.command == ADD_COMMAND:
        alert = msg.alert["parameters"]["alert"]
        keys = [alert["rule"]["id"]]
        action = send_keys_and_check_message(argv, keys)

        # if necessary, abort execution
        if action != CONTINUE_COMMAND:

```

```

if action == ABORT_COMMAND:
    write_debug_file(argv[0], "Aborted")
    sys.exit(OS_SUCCESS)
else:
    write_debug_file(argv[0], "Invalid command")
    sys.exit(OS_INVALID)

try:
    file_path =
msg.alert["parameters"]["alert"]["data"]["virustotal"]["source"]["file"]
    if os.path.exists(file_path):
        os.remove(file_path)
        write_debug_file(argv[0], json.dumps(msg.alert) + " Successfully
removed threat")
    except OSError as error:
        write_debug_file(argv[0], json.dumps(msg.alert) + "Error removing
threat")

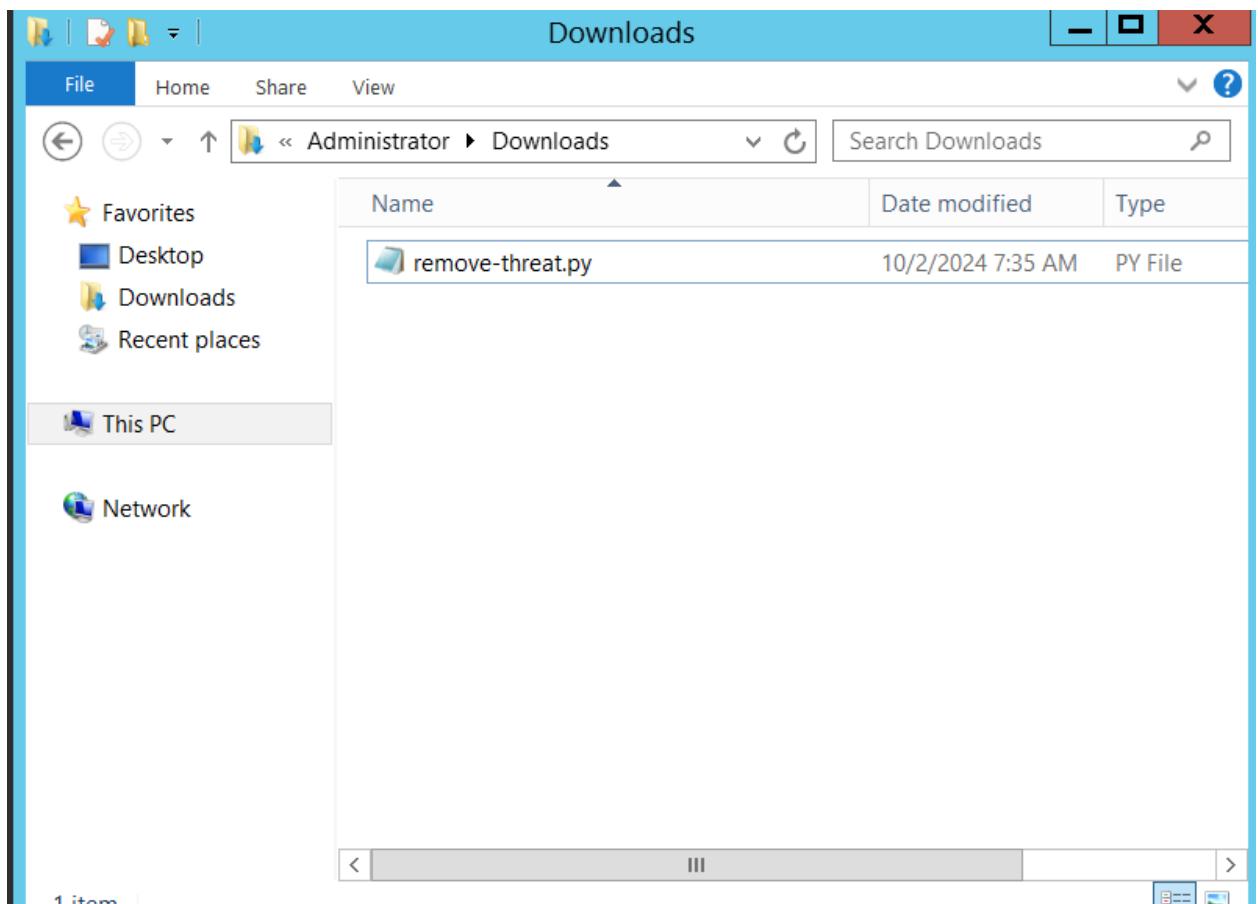
else:
    write_debug_file(argv[0], "Invalid command")

    write_debug_file(argv[0], "Ended")

    sys.exit(OS_SUCCESS)

if __name__ == "__main__":
    main(sys.argv)

```

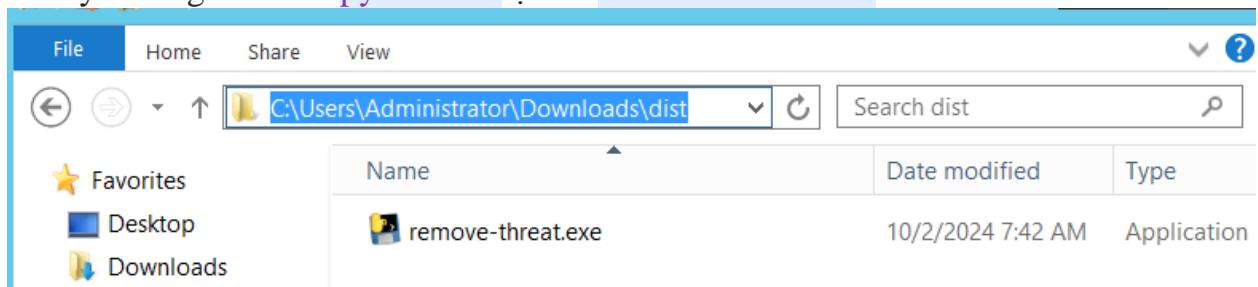


B6: Chuyển đổi tập lệnh Python phản hồi đang hoạt động **remove-threat.py** thành ứng dụng thực thi Windows. Chạy lệnh PowerShell sau với tư cách quản trị viên để tạo tệp thực thi:

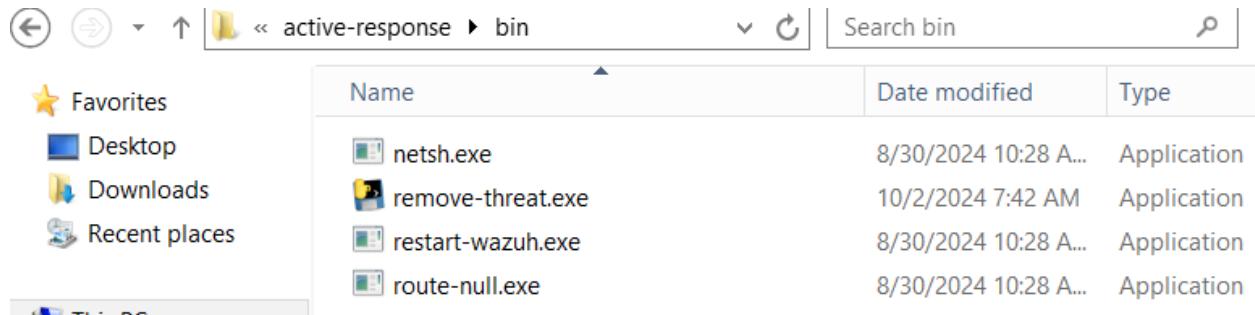
```
> pyinstaller -F \path_to_remove-threat.py
```

```
PS C:\Users\Administrator\Downloads> pyinstaller -F C:\Users\Administrator\Downloads\remove-threat.py
```

Lưu ý đường dẫn nơi **pyinstaller** tạo ra **remove-threat.exe**



B7: Di chuyển tập tin thực thi **remove-threat.exe** vào thư mục **C:\Program Files (x86)\ossec-agent\active-response\bin**



B8: Khởi động lại tác nhân Wazuh để áp dụng các thay đổi. Chạy lệnh PowerShell sau với tư cách quản trị viên:

> Restart-Service -Name wazuh

- Ở máy Wazuh Server

Thực hiện các bước sau trên Wazuh Server để cấu hình tích hợp VirusTotal. Các bước này cũng kích hoạt và kích hoạt tập lệnh phản hồi hoạt động bát cứ khi nào phát hiện tệp đáng ngờ.

B1: Thêm cấu hình sau vào `/var/ossec/etc/ossec.conf` trên máy chủ Wazuh để kích hoạt tích hợp VirusTotal.

Thay thế `<YOUR_VIRUS_TOTAL_API_KEY>` bằng [khóa API VirusTotal](#) của chính mình. Điều này cho phép kích hoạt truy vấn VirusTotal bát cứ khi nào bất kỳ quy tắc nào trong nhóm `syscheck` FIM được kích hoạt:

```

<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key><YOUR_VIRUS_TOTAL_API_KEY></api_key> <!-- Replace with your VirusTotal API key --
  >
    <group>syscheck</group>
    <alert_format>json</alert_format>

```

```
</integration>  
</ossec_config>
```

B2: Thêm các khối sau vào tệp `/var/ossec/etc/ossec.conf` máy chủ Wazuh.

Điều này cho phép phản hồi chủ động và kích hoạt tệp `remove-threat.exe` thực thi khi truy vấn VirusTotal trả về kết quả khớp dương tính với các mối đe dọa:

```
<ossec_config>  
  <command>  
    <name>remove-threat</name>  
    <executable>remove-threat.exe</executable>  
    <timeout_allowed>no</timeout_allowed>  
  </command>  
  
  <active-response>  
    <disabled>no</disabled>  
    <command>remove-threat</command>  
    <location>local</location>  
    <rules_id>87105</rules_id>  
  </active-response>  
</ossec_config>
```

```

#VirusTotal for Agent Windows Server
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>7948aa22354fe9532c6b2d185392c672a5749c13e46fa715376c1fba02ee8938</api_key> <!-- Re>
    <group>syscheck</group>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>

<ossec_config>
  <command>
    <name>remove-threat</name>
    <executable>remove-threat.exe</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <disabled>no</disabled>
    <command>remove-threat</command>
    <location>local</location>
    <rules_id>87105</rules_id>
  </active-response>
</ossec_config>

```

B3: Thêm các quy tắc sau vào tệp [/var/ossec/etc/rules/local_rules.xml](#) máy chủ Wazuh để cảnh báo về kết quả active response.

```

<group name="virustotal,">
  <rule id="100092" level="12">
    <if_sid>657</if_sid>
    <match>Successfully removed threat</match>
    <description>$(parameters.program) removed threat located at
$(parameters.alert.data.virustotal.source.file)</description>
  </rule>

  <rule id="100093" level="12">
    <if_sid>657</if_sid>
    <match>Error removing threat</match>
    <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
  </rule>
</group>

```

```

#For Agent Windows Server
<group name="virustotal,">
  <rule id="100092" level="12">
    <if_sid>657</if_sid>
    <match>Successfully removed threat</match>
    <description>$(parameters.program) removed threat located at $(parameters.alert.data.virusTotal.url)</description>
  </rule>

  <rule id="100093" level="12">
    <if_sid>657</if_sid>
    <match>Error removing threat</match>
    <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
  </rule>
</group>

```

B4: Khởi động lại trình quản lý Wazuh để áp dụng các thay đổi cấu hình:

systemctl restart wazuh-manager.service

- Mô phỏng tấn công:

B1: Thực hiện theo các bước sau để tạm thời tắt tính năng bảo vệ chống vi-rút Microsoft Defender theo thời gian thực trong Windows Security:

- Nhập vào menu **Start** và nhập để tìm kiếm ứng dụng đó. **Windows Security**
- Chọn **Windows Security app** từ kết quả, đi tới **Virus & threat protection** và trong phần **Virus & threat protection settings**, chọn **Manage settings**.
- Tắt tính năng **Real-time protection**.

B2: Tải tệp [EICAR test](#) xuống **C:\Users\<USER_NAME>\Downloads** mục trên điểm cuối Windows.

```

> Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile eicar.txt
> cp .\eicar.txt C:\Users\<USER_NAME>\Downloads

```

```

PS C:\Users\Administrator> Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile eicar.txt
PS C:\Users\Administrator> cp .\eicar.txt C:\Users\Administrator\Downloads
PS C:\Users\Administrator>

```

Nếu báo lỗi này:

```
PS C:\Users\Administrator> Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile eicar.txt
Invoke-WebRequest : The request was aborted: Could not create SSL/TLS secure channel.
At line:1 char:1
+ Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile eicar.txt
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
+ FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
```

Lỗi này là do PowerShell không thể thiết lập kết nối SSL/TLS an toàn với máy chủ. Điều này thường xảy ra khi phiên bản TLS mặc định của hệ thống không tương thích với yêu cầu bảo mật của máy chủ đích. thì ta có thể fix như sau:

1. Mở PowerShell với quyền Administrator:

- Nhấn Start, gõ **PowerShell**.
- Nhấp chuột phải vào **Windows PowerShell** và chọn **Run as administrator**.

2. Chạy lệnh sau để thiết lập TLS 1.2:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

Hoặc

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12 -bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls (Nếu muốn hỗ trợ cả các phiên bản TLS khác (như TLS 1.1 hoặc TLS 1.0), có thể kết hợp chúng bằng cách sử dụng toán tử -bor)

```
PS C:\Users\Administrator> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
PS C:\Users\Administrator> Invoke-WebRequest -Uri https://secure.eicar.org/eicar.com.txt -OutFile eicar.txt
PS C:\Users\Administrator> cp ./eicar.txt C:\Users\Administrator\Downloads
PS C:\Users\Administrator>
```

Thao tác này kích hoạt truy vấn VirusTotal và tạo cảnh báo. Ngoài ra, tập lệnh active response sẽ tự động xóa tệp.

Kết quả: Vào Wazuh Server để xem cảnh báo

Time	rule.description	timestamp per 30 minutes	rule.level	rule.id
> May 1, 2024 @ 23:12:10.563	VirusTotal: Alert - c:\users\thecotilking\downloads\eicar.txt - 64 engines detected this file		12	87105
> May 1, 2024 @ 23:09:06.173	active-response/bin/remove-threat.exe removed threat located at c:\users\thecotilking\downloads\eicar.txt		12	100092
> May 1, 2024 @ 23:07:53.995	File deleted.		7	553
> May 1, 2024 @ 23:07:39.655	VirusTotal: Alert - c:\users\thecotilking\downloads\eicar.txt - 64 engines detected this file		12	87105
> May 1, 2024 @ 23:05:21.518	File added to the system.		5	554

- Có 64 công cụ đã phát hiện ra file này là độc hại.

- File đã bị xóa ngay sau khi được tải xuống.

CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

3.1 Tóm tắt kết quả đạt được

Sau khi triển khai và thực hiện đề tài "*Nghiên cứu giải pháp SIEM && XDR với Wazuh*", các kết quả sau đây đã được đạt được:

1. Cải thiện khả năng giám sát và phát hiện mối đe dọa

- Phát hiện và cảnh báo sớm các mối đe dọa an ninh, bao gồm tấn công brute-force, SQL injection, và các tệp tin độc hại.
- Phát hiện và chặn IP độc hại truy cập đến Web Server.

2. Tích hợp thành công với các hệ thống khác: Wazuh được tích hợp với các hệ thống hiện có như tường lửa, hệ thống phát hiện xâm nhập (IDS/IPS), và các ứng dụng quản lý bảo mật khác như VirusTotal, Elastic Stack để cung cấp một bức tranh bảo mật toàn diện.

3.2 Những khó khăn và thách thức trong quá trình triển khai

Trong quá trình triển khai, cũng đã gặp phải một số khó khăn và thách thức, cụ thể như sau:

1. Khả năng mở rộng hệ thống

- Wazuh cần được triển khai trên một môi trường hạ tầng đủ mạnh để có thể xử lý các tác vụ phức tạp như phân tích log thời gian thực, phát hiện các mối đe dọa và phản ứng kịp thời. Tuy nhiên, việc mở rộng hệ thống có thể gặp khó khăn về mặt tài nguyên hạ tầng, như CPU, RAM, dung lượng lưu trữ.

- Việc đảm bảo hiệu suất hệ thống khi mở rộng số lượng agent và thiết bị giám sát đòi hỏi phải có kế hoạch chi tiết và cài đặt đúng cách các thành phần như Wazuh Cluster, Elasticsearch.

2. Thiếu tài liệu và hỗ trợ kỹ thuật

- Mặc dù Wazuh là một nền tảng mã nguồn mở mạnh mẽ, nhưng đôi khi tài liệu hỗ trợ cho các vấn đề cụ thể chưa đầy đủ, đặc biệt là khi gặp các tình huống cần cấu hình nâng cao.

- Ngoài ra, việc tìm kiếm sự hỗ trợ từ cộng đồng hoặc nhóm phát triển chính thức cũng có thể mất thời gian do không có sự hỗ trợ thương mại trực tiếp (trừ khi sử dụng dịch vụ trả phí).

3.3 Hướng phát triển và cải thiện trong tương lai

Đề tài "*Nghiên cứu giải pháp SIEM && XDR với Wazuh*" là một giải pháp bảo mật tiềm năng, nhưng để tối ưu hóa và mở rộng trong tương lai, cần có những kế hoạch phát triển và cải thiện. Dưới đây là các hướng phát triển và cải thiện chính:

1. Tối ưu hóa hiệu suất hệ thống

- Cải thiện khả năng xử lý log: Khối lượng dữ liệu log mà hệ thống thu thập sẽ tăng lên theo thời gian và số lượng thiết bị. Do đó, cần nghiên cứu và áp dụng các biện pháp tối ưu hóa khả năng xử lý log, giảm thiểu độ trễ trong việc phân tích log và cảnh báo. Điều này có thể bao gồm việc sử dụng Elasticsearch Cluster, tối ưu hóa cấu hình của Wazuh Manager và tăng cường tài nguyên phần cứng.

- Tối ưu bộ nhớ và tài nguyên hệ thống: Đảm bảo hệ thống có thể mở rộng để quản lý số lượng lớn agent mà không làm giảm hiệu suất, thông qua việc áp dụng các phương pháp phân tán, load balancing hoặc tăng cường phần cứng.

2. Mở rộng khả năng tích hợp

- Tích hợp với nhiều nền tảng khác: Wazuh có thể được tích hợp với các hệ thống khác như SIEM doanh nghiệp hoặc các dịch vụ đám mây lớn như AWS, Azure, và Google Cloud Platform để giám sát toàn diện hơn. Trong tương lai, việc mở rộng tích hợp với nhiều hệ thống hơn nữa sẽ giúp cung cấp một giải pháp bảo mật mạnh mẽ hơn.

- Hỗ trợ thêm các công cụ bảo mật khác: Tích hợp với các giải pháp tường lửa nâng cao, IPS/IDS, và các công cụ quản lý lỗ hổng sẽ giúp cải thiện khả năng phát hiện và phản ứng nhanh chóng đối với các mối đe dọa tiềm ẩn.

3. Phát triển tính năng phân tích và dự đoán

- Ứng dụng trí tuệ nhân tạo và machine learning: Sử dụng AI và ML để phân tích và dự đoán các mối đe dọa bảo mật dựa trên các mẫu hành vi và dữ liệu log thu

thập được. Điều này có thể giúp Wazuh không chỉ phản ứng với các mối đe dọa hiện tại mà còn dự đoán trước các mối đe dọa trong tương lai.

- Cải thiện hệ thống cảnh báo thông minh: Tinh chỉnh các rules và thuật toán cảnh báo để giảm bớt cảnh báo sai (false positives) và tăng khả năng phát hiện đúng các mối đe dọa thực sự.

4. Tăng cường giám sát và quản lý tập trung

Phát triển hệ thống giám sát tập trung: Cải thiện hệ thống giám sát tập trung cho nhiều máy chủ và thiết bị, giúp quản lý dễ dàng hơn trong các môi trường doanh nghiệp lớn hoặc hệ thống phức tạp. Điều này giúp tiết kiệm thời gian và nâng cao hiệu quả trong việc phát hiện các mối đe dọa tiềm tàng.

TÀI LIỆU THAM KHẢO

Hiểu về SIEM && XDR:

https://www.researchgate.net/publication/372503637_Battle_of_Defenses_Understanding_SIEM_vs_XDR_in_Modern_Cybersecurity

Cài đặt Wazuh: <https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-linux.html>

Hướng dẫn thực nghiệm: <https://documentation.wazuh.com/current/proof-of-concept-guide/index.html>

