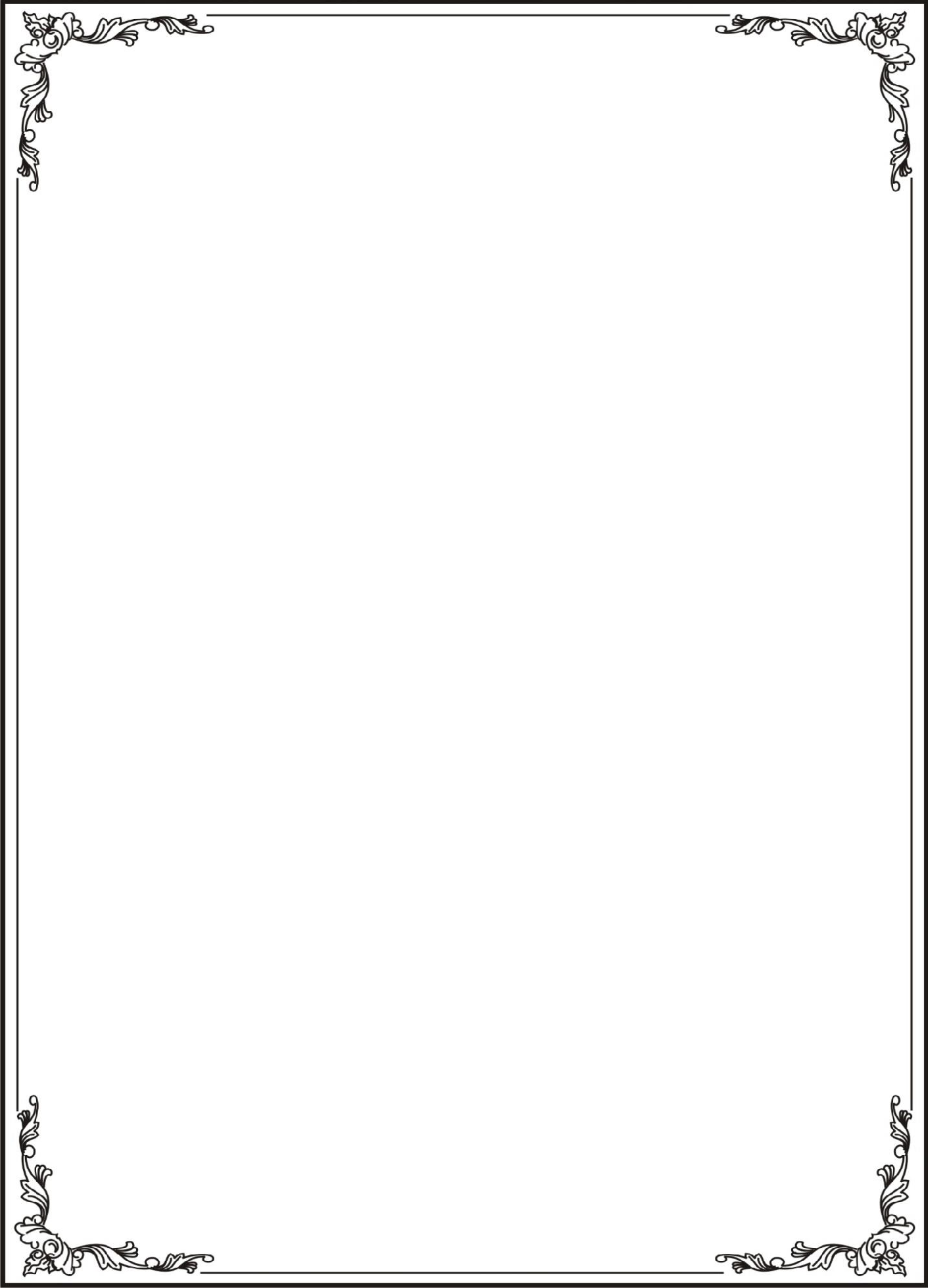
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH** 

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

**NGUYỄN THỊ MINH CHÂU**

**NGUYỄN TRIỆU THIÊN BẢO**

**NGUYỄN PHƯƠNG TRINH**

**NGUYỄN THÀNH ĐẠT**

**TRẦN VĂN THÁI**

A picture containing graphics, logo, clipart, symbol

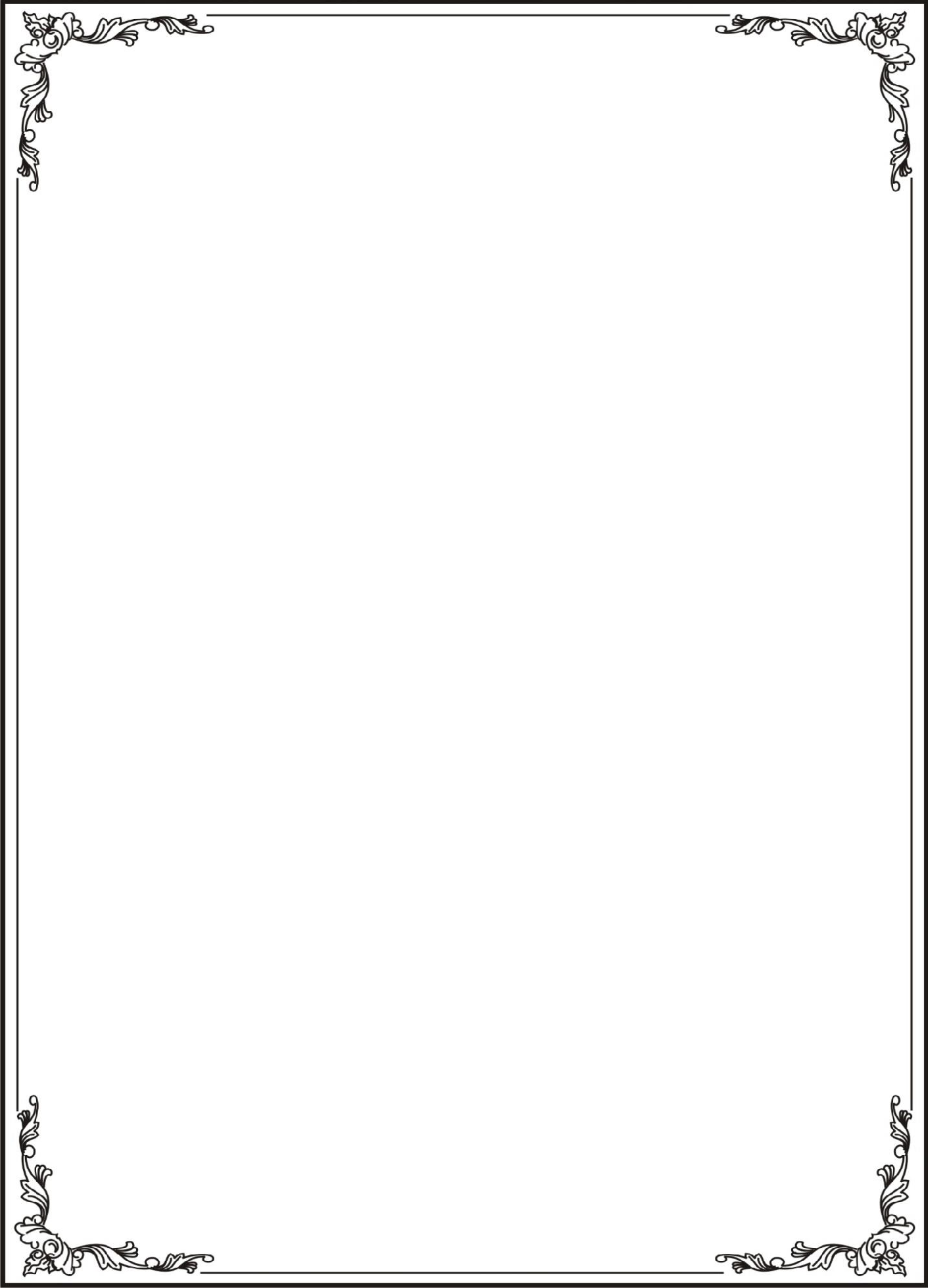
Description automatically generated

**BÁO CÁO ĐỒ ÁN**

HỆ THỐNG TÌM KIẾM, PHÁT HIỆN VÀ NGĂN NGỪA XÂM NHẬP – NT204.O21.ANTT

WAZUH

*Thành phố Hồ Chí Minh, tháng 5 năm 2024*

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH** 

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

**NGUYỄN THỊ MINH CHÂU – MSSV: 21520645**

**NGUYỄN PHƯƠNG TRINH – MSSV: 21521581**

**NGUYỄN TRIỆU THIÊN BẢO – MSSV: 21520155**

**NGUYỄN THÀNH ĐẠT – MSSV: 21521940**

**TRẦN VĂN THÁI – MSSV: 21522581**

A picture containing graphics, logo, clipart, symbol

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

HỆ THỐNG TÌM KIẾM, PHÁT HIỆN VÀ NGĂN NGỪA XÂM NHẬP – NT204.O21.ANTT

WAZUH

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS. ĐỖ HOÀNG HIỂN**

*Thành phố Hồ Chí Minh, tháng 5 năm 2024*

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin phép gửi lời cảm ơn sâu sắc đến tập thể quý thầy cô trường Đại học Công nghệ Thông tin - Đại học Quốc gia TP.HCM, quý thầy cô khoa Mạng máy tính & Truyền thông và Phòng thí nghiệm An toàn thông tin - UIT InSecLab đã tạo điều kiện, giúp chúng em học tập và có được những kiến thức nền tảng cũng như cung cấp các tài nguyên liên quan để hoàn thành được dự án này.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến thầy Đỗ Hoàng Hiển. Với sự tâm huyết, được sự tận tình giảng dạy và hỗ trợ hết lòng của thầy đã cho chúng em nhiều kiến thức bổ ích. Với tình cảm sâu sắc, chân thành, chúng em xin bày tỏ lòng biết ơn đến thầy đã nhiệt tình, hết mình với sinh viên. Đó là động lực rất lớn để chúng em có thể hoàn thành tốt đồ án lần này.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, chúng em đã cố gắng hết mình nhưng đồ án không thể tránh được những thiếu sót. Chúng em rất hy vọng nhận được sự chỉ bảo và đóng góp ý kiến từ thầy để bổ sung, nâng cao kiến thức của mình, phục vụ và hoàn thiện hơn trong những đồ án sau này và khóa luận tốt nghiệp trong lương lai.

Chúng em xin chân thành cảm ơn!

**Nhóm thực hiện**

**MỤC LỤC**

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 5**](#_heading=h.1fob9te)

[***1.1.***](#_heading=h.3znysh7) ***Giới thiệu: 5***

[***1.2.***](#_heading=h.2et92p0) ***Đề tài: Phân tích hiệu quả của Wazuh trong việc phát hiện xâm nhập trên hệ thống. 7***

[***1.3.***](#_heading=h.tyjcwt) ***Mục tiêu 7***

[**CHƯƠNG 2: TỔNG QUAN WAZUH 7**](#_heading=h.3dy6vkm)

[***2.1.***](#_heading=h.1t3h5sf) ***Giới thiệu về Wazuh 7***

[**2.1.1.**](#_heading=h.4d34og8) **Wazuh 7**

[**2.1.2.**](#_heading=h.2s8eyo1) **Cấu trúc của Wazuh 10**

[**2.1.3.**](#_heading=h.17dp8vu) **Một số tính năng chính trong Wazuh 12**

[**2.1.4.**](#_heading=h.3rdcrjn) **Ưu và nhược điểm khi triển khai Wazuh 14**

[**CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG 16**](#_heading=h.26in1rg)

[***3.1.***](#_heading=h.lnxbz9) ***Triển khai môi trường cần thiết 16***

[***3.2.***](#_heading=h.35nkun2) ***Cài đặt Wazuh 18***

[***3.3.***](#_heading=h.1ksv4uv) ***Triển khai Wazuh 21***

[**CHƯƠNG 4: HIỆN THỰC HỆ THỐNG 21**](#_heading=h.44sinio)

[***4.1.***](#_heading=h.2jxsxqh) ***Kịch bản 1: Tấn công Brute Force vào máy chủ SSH trên Web Server 22***

[***4.2.***](#_heading=h.z337ya) ***Kịch bản 2: Phát hiện kết nối mạng dị thường 26***

[***Ta sẽ cài đặt trên server trước: 27***](#_heading=h.3j2qqm3)

[***4.3.***](#_heading=h.1y810tw) ***Kịch bản 3: Phát hiện các thay đổi trái phép trên hệ thống tập tin 31***

[***4.4.***](#_heading=h.4i7ojhp) ***Kịch bản 4: Phát hiện lỗ hổng SQL Injection trên Web Server DVWA 33***

[***4.5.***](#_heading=h.2xcytpi) ***Kịch bản 5: Phát hiện tấn công Shellshock trên Web Server DVWA (CVE-2014-6271) 36***

[***4.6.***](#_heading=h.1ci93xb) ***Kịch bản 6: Phát hiện rootkit Diamorphine trên hệ thống 38***

[***4.7.***](#_heading=h.3whwml4) ***Wazuh phát hiện và loại bỏ phần mềm độc hại bằng cách tích hợp VirusTotal 40***

[***4.8.***](#_heading=h.2bn6wsx) ***Kịch bản 8: Wazuh phát hiện tiến trình chưa được chứng thực 40***

[***4.9.***](#_heading=h.qsh70q) ***Kịch bản 9: Wazuh phát hiện khai thác lỗ hổng XZ Utils (CVE- 2024-3094) 40***

[**CHƯƠNG 5: THỰC NGHIỆM VÀ ĐÁNH GIÁ 49**](#_heading=h.3as4poj)

[***5.1.***](#_heading=h.1pxezwc) ***Kịch bản thực nghiệm 49***

[**5.1.1.**](#_heading=h.49x2ik5) **Kịch bản 1: Tấn công Brute Force vào máy chủ SSH trên Web Server 49**

[**5.1.2.**](#_heading=h.2p2csry) **Kịch bản 2: Phát hiện kết nối mạng dị thường 50**

[**5.1.3.**](#_heading=h.147n2zr) **Kịch bản 3: Phát hiện các thay đổi trái phép trên hệ thống tập tin 50**

[**5.1.4.**](#_heading=h.3o7alnk) **Kịch bản 4: Phát hiện lỗ hổng SQL Injection trên Web Server DVWA 50**

[**5.1.5.**](#_heading=h.23ckvvd) **Kịch bản 5: Phát hiện tấn công Shellshock trên Web Server DVWA (CVE-2014-6271) 51**

[**5.1.6.**](#_heading=h.ihv636) **Kịch bản 6: Phát hiện rootkit Diamorphine trên hệ thống 51**

[**5.1.7.**](#_heading=h.32hioqz) **Kịch bản 7: Wazuh phát hiện và loại bỏ phần mềm độc hại bằng cách tích hợp VirusTotal 51**

[**5.1.8.**](#_heading=h.1hmsyys) **Kịch bản 8: Wazuh phát hiện tiến trình chưa được chứng thực 52**

[**5.1.9.**](#_heading=h.41mghml) **Kịch bản 9: Wazuh phát hiện khai thác lỗ hổng XZ Utils (CVE- 2024-3094) 52**

[***5.2.***](#_heading=h.2grqrue) ***Đánh giá 53***

[**CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 53**](#_heading=h.vx1227)

[***6.1.***](#_heading=h.3fwokq0) ***Kết luận: 53***

[***6.2.***](#_heading=h.1v1yuxt) ***Hướng phát triển: 53***

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu:

Trong bối cảnh hiện đại, khi công nghệ thông tin ngày càng phát triển và trở thành nền tảng của mọi hoạt động kinh tế, xã hội, thì việc bảo vệ thông tin và đảm bảo an ninh mạng trở nên quan trọng hơn bao giờ hết. Các tổ chức và doanh nghiệp không chỉ đối mặt với những cơ hội mới từ sự bùng nổ của công nghệ mà còn phải đối diện với những thách thức và rủi ro an ninh mạng phức tạp hơn. Các cuộc tấn công mạng ngày càng tinh vi, đa dạng và khó phát hiện, từ đó đặt ra yêu cầu cấp thiết về việc phát triển và triển khai các giải pháp bảo mật hiệu quả, toàn diện.

Wazuh, một giải pháp an ninh mã nguồn mở, đã nổi lên như một công cụ mạnh mẽ trong việc giám sát và quản lý an ninh mạng. Được xây dựng trên nền tảng của hệ thống phát hiện xâm nhập (IDS) và quản lý sự kiện bảo mật (SIEM), Wazuh không chỉ cung cấp các khả năng giám sát bảo mật mà còn tích hợp nhiều tính năng ưu việt giúp các tổ chức duy trì và nâng cao mức độ an ninh thông tin. Với khả năng thu thập, phân tích và phản hồi các sự kiện an ninh theo thời gian thực, Wazuh giúp phát hiện và ngăn chặn các mối đe dọa tiềm ẩn, đồng thời hỗ trợ tuân thủ các tiêu chuẩn và quy định về bảo mật.

Đồ án nghiên cứu này sẽ tập trung phân tích và đánh giá Wazuh từ nhiều khía cạnh khác nhau, nhằm cung cấp một cái nhìn toàn diện và sâu sắc về giải pháp bảo mật này. Nhóm sẽ đi sâu vào việc tìm hiểu các tính năng chính của Wazuh, bao gồm quản lý sự kiện bảo mật, phát hiện xâm nhập, quản lý tuân thủ và khả năng tích hợp với các công cụ và hệ thống khác. Đặc biệt, đồ án sẽ tập trung phân tích cách Wazuh thu thập và xử lý dữ liệu từ nhiều nguồn khác nhau, cách nó phát hiện và phản hồi các mối đe dọa, và cách nó hỗ trợ các tổ chức trong việc duy trì tuân thủ các quy định về an ninh mạng.

Một phần quan trọng của đồ án là thực hiện triển khai Wazuh trong một môi trường thực tế. Thông qua việc thiết lập một hệ thống thử nghiệm, nhóm chúng em sẽ kiểm tra tính khả thi và hiệu quả của Wazuh trong việc phát hiện và ngăn chặn các cuộc tấn công mạng. Nhóm cũng sẽ đánh giá khả năng mở rộng và tùy chỉnh của Wazuh để phù hợp với các nhu cầu cụ thể của từng tổ chức. Bằng cách này, đồ án sẽ cung cấp các hướng dẫn chi tiết về cách triển khai và tối ưu hóa Wazuh, từ đó giúp các tổ chức áp dụng hệ thống này một cách hiệu quả nhất.

Ngoài ra, đồ án cũng sẽ so sánh Wazuh với các giải pháp bảo mật khác trên thị trường. Việc so sánh này không chỉ nhằm đánh giá hiệu quả và tính năng của Wazuh mà còn giúp nhận diện các ưu và nhược điểm của nó so với các công cụ bảo mật khác. Nhóm sẽ xem xét các yếu tố như khả năng phát hiện mối đe dọa, tính dễ sử dụng, chi phí và khả năng hỗ trợ kỹ thuật, từ đó đưa ra các đánh giá toàn diện và khách quan.

Cuối cùng, dựa trên các kết quả nghiên cứu và thử nghiệm, nhóm sẽ đề xuất các biện pháp cải tiến và tối ưu hóa việc sử dụng Wazuh. Những khuyến nghị này không chỉ nhằm nâng cao hiệu quả bảo mật mà còn giúp tối ưu hóa chi phí và tài nguyên cho các tổ chức. Đồng thời, chúng tôi cũng sẽ đề xuất các tính năng mới hoặc các cải tiến có thể được thêm vào Wazuh để nâng cao hơn nữa khả năng bảo vệ và quản lý an ninh mạng.

Như vậy, đồ án nghiên cứu về Wazuh không chỉ nhằm mục tiêu phân tích và đánh giá một giải pháp bảo mật cụ thể mà còn hướng đến việc cung cấp các kiến thức và công cụ thực tiễn để các tổ chức và doanh nghiệp có thể áp dụng hiệu quả trong việc bảo vệ hệ thống thông tin của mình. Chúng em hy vọng rằng, thông qua nghiên cứu này, Wazuh sẽ được nhìn nhận như một giải pháp bảo mật toàn diện và đáng tin cậy, góp phần vào việc nâng cao an ninh mạng cho các tổ chức và doanh nghiệp trong bối cảnh ngày càng nhiều nguy cơ và thách thức hiện nay.

## Đề tài: Phân tích hiệu quả của Wazuh trong việc phát hiện xâm nhập trên hệ thống.

## Mục tiêu

* Đánh giá hiệu quả của Wazuh trong quản lý bảo mật hệ thống
* Tìm hiểu và triển khai các tính năng cơ bản của Wazuh
* So sánh Wazuh với các giải pháp bảo mật khác
* Đề xuất cải tiến và tối ưu hoá việc sử dụng Wazuh

# TỔNG QUAN WAZUH

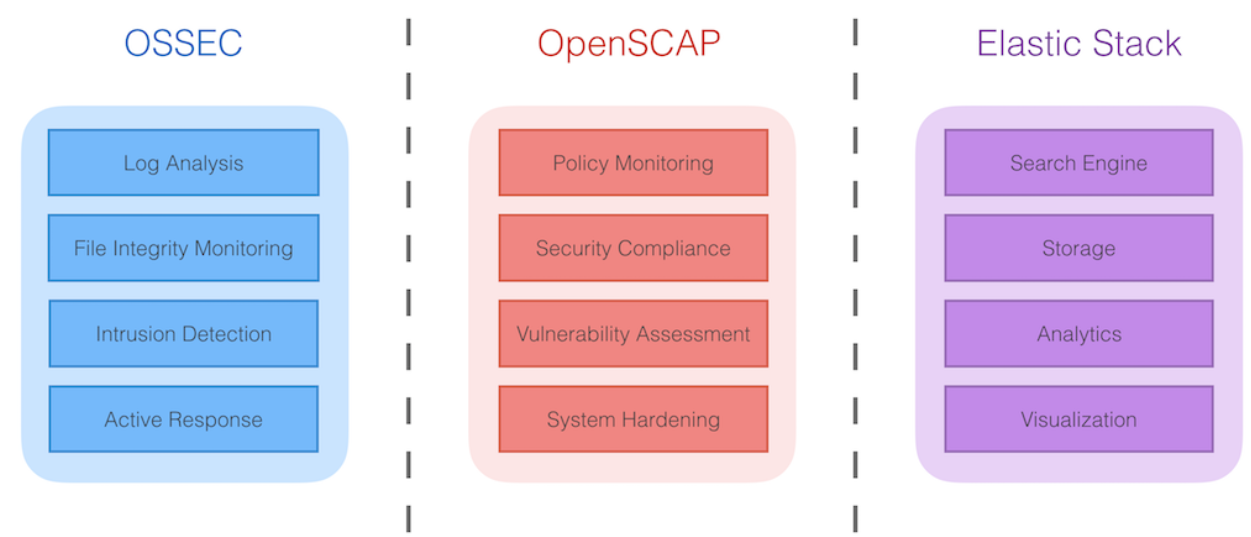
## Giới thiệu về Wazuh

### Wazuh

WAZUH là một dự án mã nguồn mở với các chức năng security detection,

visibility, và compliance monitoring. Wazuh ban đầu được phát triển dựa trên OSSEC HIDS và sau đó được tích hợp thêm Elastic Stack cùng với OpenSCAP

để trở thành một giải pháp an ninh toàn diện.



Wazuh cung cấp các tính năng để phát hiện các mối đe dọa phổ biến như tấn công từ chối dịch vụ (DoS), tấn công dò tìm xung đột (port scan), tấn công tìm lỗ hổng (vulnerability scan) và tấn công tràn đổ bộ nhớ đệm (buffer overflow). Bên cạnh đó còn giám sát toàn vẹn, bảo vệ an ninh và ứng phó sự cố.



Wazuh tự động thu thập và tổng hợp dữ liệu bảo mật từ các hệ thống chạy Linux, Windows, macOS, Solaris, AIX và các hệ điều hành khác trong miền được giám sát, làm cho nó trở thành một giải pháp SIEM cực kỳ toàn diện.

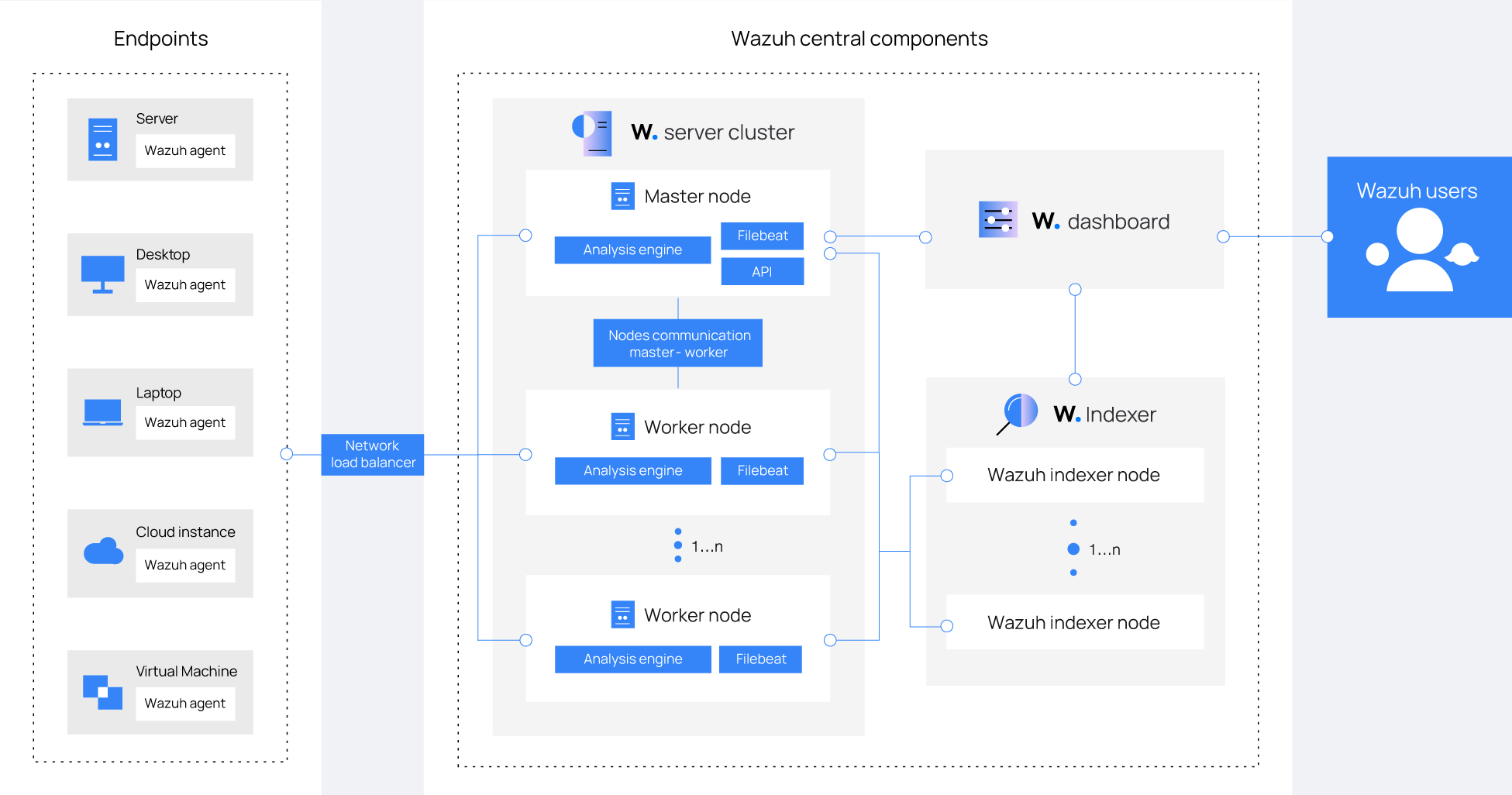
Nhưng quan trọng hơn, Wazuh còn có khả năng phân tích và đối chiếu dữ liệu để phát hiện các điểm bất thường và xâm nhập. Nhờ khả năng thông minh này, Wazuh có thể phát hiện sớm các mối đe dọa trong nhiều môi trường khác nhau.

Phản ứng sự cố là một tính năng cực kỳ hữu ích của Wazuh trong việc đối phó với các mối đe dọa đang diễn ra. Wazuh cung cấp các phản ứng tự động "out-of-the-box", nghĩa là người dùng không cần phải thực hiện bất kỳ cấu hình bổ sung nào để kích hoạt chúng. Khi hệ thống phát hiện mối đe dọa đang hoạt động, các biện pháp đối phó sẽ tự động được triển khai ngay lập tức.

Ví dụ, nhiều tin tặc sử dụng các cuộc tấn công brute-force để đoán kết hợp tên

người dùng và mật khẩu. Wazuh sẽ lưu ý mỗi lần xác thực không thành công. Với đủ lỗi, hệ thống sẽ nhận ra chúng là một phần của cuộc tấn công vũ phu. Vì một tiêu chí nhất định được đáp ứng (ví dụ: năm lần đăng nhập không thành công), nó sẽ chặn địa chỉ IP đó khỏi các lần thử tiếp theo. Điều này có nghĩa là Wazuh không chỉ có thể hứng chịu các cuộc tấn công bạo lực mà còn có thể tắt chúng.

Kiến trúc của Wazuh dựa trên các agent, chạy trên các điểm cuối được giám sát sau đó chuyển tiếp dữ liệu bảo mật đến máy chủ trung tâm .Các thiết bị không có agent như tường lửa, bộ chuyển mạch, bộ định tuyến và điểm truy cập được hỗ trợ và có thể chủ động gửi dữ liệu nhật ký qua Syslog, SSH hoặc sử dụng API của chúng. Máy chủ trung tâm giải mã và phân tích thông tin đến và chuyển kết quả đến Wazuh Indexer để lập chỉ mục và lưu trữ.



#### OSSEC

OSSEC là một HIDS (Host-based Intrusion Detection System) với kiến trúc gồm các agent vệ tinh đa nền tảng và một hệ thống quản lý trung tâm (central manager). Các agent này sẽ forward dữ liệu hệ thống (chẳng hạn log messages, file hashes, và các hoạt động bất thường) đến cho trung tâm quản lý để được phân tích và xử lý và đưa ra các alert thích hợp. Các dữ liệu này sẽ được gửi thông qua các kênh truyền an toàn và được xác thực.

Ngoài ra, OSSEC cũng có thể được triển khai với cấu hình một syslog server trung tâm cùng hệ thống giám sát phi tác nhân (agentless configuration monitoring system). Ở cấu hình này, OSSEC sẽ đưa ra các security insight từ các sự kiện và thay đổi diễn ra trên các thiết bị agentless như firewall, switch, routers, access point, network appliance, …

#### OpenSCAP

OpenSCAP là trình thông dịch cho chuẩn ngôn ngữ thẩm định OVAL (Open Vulnerability Assess - ment Language) và định dạng cấu hình an ninh hệ thống ECCDF (Extensible Configuration Checklist Description Format).

OpenSCAP có được sử dụng để kiểm tra các cấu hình của hệ thống và phát hiện các ứng dụng có lỗ hổng. Đây là một công cụ được thiết kế với khả năng thẩm định và gia cố của các hệ thống áp dụng các tiêu chuẩn an ninh công nghiệp cho môi trường doanh nghiệp.

#### Elastic Stack

Elastic Stack là một bộ phần mềm (Wazuh sử dụng Filebeat, Elasticsearch, Kibana) được dùng để thu thập, phân giải, index, lưu trữ, tìm kiếm và biểu diễn log. Elastic Stack cung cấp một high-level dashboard ở phía front-end để phục vụ các thao tác phân tích nâng cao cũng như data mining.

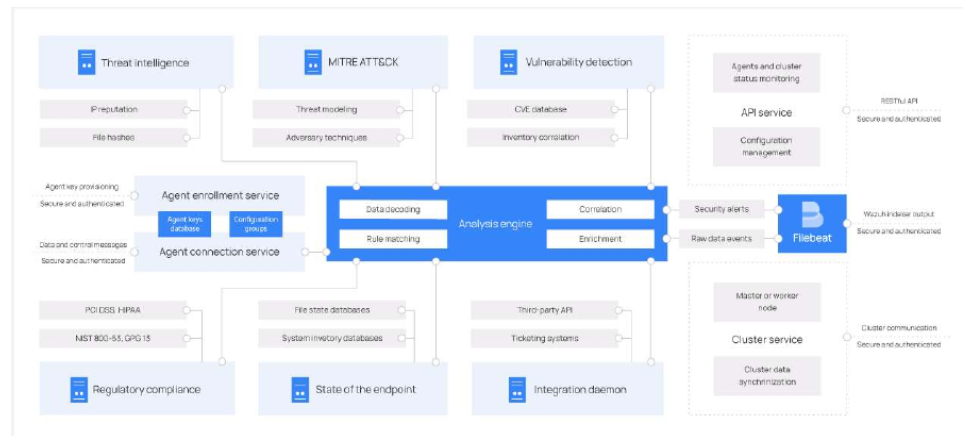
### Cấu trúc của Wazuh

#### Wazuh Server

Wazuh server chịu trách nhiệm phân tích dữ liệu gửi từ agent và trigger

alert khi có một sự kiện khớp với rule (chẳng hạn phát hiện xâm nhập, file bị

thay đổi, cấu hình không tuân thủ chính sách, rootkit,...).

 Wazuh server thường sẽ được cài đặt trên một máy chủ vật lý hoặc máy ảo riêng biệt, hay cũng có thể là một máy trên cloud và chạy các agent để tự giám sát chính nó. Các thành phần chính của Wazuh Server:

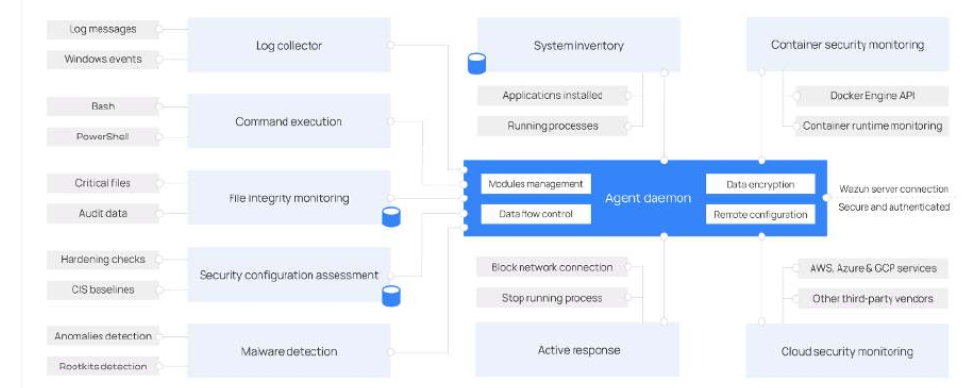
* + *Agents registration service*: Được sử dụng để đăng ký Agent mới bằng cách cung cấp và phân phối khóa xác thực chia sẻ trước dành riêng cho từng Agent. Quá trình này chạy như một dịch vụ mạng và hỗ trợ xác thực thông qua chứng chỉ TLS / SSL hoặc bằng cách cung cấp mật khẩu cố định.
  + *Agents registration service*: Đây là dịch vụ nhận dữ liệu từ các Agent. Nó sử dụng các khóa được chia sẻ trước để xác thực danh tính từng tác nhân và mã hóa thông tin liên lạc giữa tác nhân và Wazuh Serever. Ngoài ra, dịch vụ này được sử dụng để cung cấp quản lý cấu hình tập trung, có thể đẩy các cài đặt tác nhân mới từ xa.
  + *Analysis engine*: Đây là quá trình thực hiện phân tích dữ liệu. Nó sử dụng bộ giải mã để xác định loại thông tin đang được xử lý (ví dụ: Windows events, SSHD logs, web server logs, v.v.) và trích xuất các phần tử dữ liệu có liên quan từ thông báo nhật ký (ví dụ: ource IP address, event ID, username, v.v.) . Tiếp theo, bằng cách sử dụng các quy tắc (Rules), nó xác định các mẫu cụ thể trong các sự kiện được giải mã có thể kích hoạt cảnh báo và thậm chí có thể gọi các biện pháp đối phó tự động (ví dụ: lệnh cấm IP trên tường lửa).
  + *Wazuh RESTful API*: Dịch vụ này cung cấp giao diện để tương tác với cơ sở hạ tầng Wazuh. Nó được sử dụng để quản lý các tác nhân và cài đặt cấu hình máy chủ, để theo dõi trạng thái cơ sở hạ tầng và tình trạng tổng thể, quản lý và chỉnh sửa các quy tắc (Rule) và bộ giải mã Wazuh cũng như truy vấn về trạng thái của các điểm cuối được giám sát. Nó cũng được sử dụng bởi giao diện người dùng web Wazuh, đó là ứng dụng Kibana.
  + *Wazuh cluster daemon*: Dịch vụ này được sử dụng để mở rộng các Wazuh Server theo chiều ngang, triển khai chúng dưới dạng một cụm. Loại cấu hình này, kết hợp với bộ cân bằng tải mạng, cung cấp tính khả dụng cao và cân bằng tải. Daemon cụm Wazuh là thứ mà các máy chủ Wazuh sử dụng để giao tiếp với nhau và giữ đồng bộ hóa.
  + *Filebeat*: Nó được sử dụng để gửi các sự kiện và cảnh báo tới Elasticsearch. Nó đọc kết quả của công cụ phân tích Wazuh và gửi các sự kiện trong thời gian thực. Nó cũng cung cấp khả năng cân bằng tải khi được kết nối với một cụm Elasticsearch nhiều nút.

#### Wazuh Agent

Wazuh agent có thể được cài đặt trên các hệ điều hành Windows, Linux, Solaris, BSD, và Mac . Các agent sẽ thu thập dữ liệu từ hệ thống và ứng dụng và sau đó gửi chúng đến Wazuh Server qua một kênh truyền đã được mã hóa và xác thực. Kênh truyền an toàn này sẽ được khởi tạo bởi một tiến trình đăng ký sử dụng các khóa duy nhất với mỗi agent.

Các agent có thể được sử dụng để giám sát các máy chủ vật lý, các máy ảo và cloud instance (Amazon AWS, Azure, Google Cloud). Các gói cài đặt agent khả dụng hiện tại bao gồm các gói cài đặt cho Linux, HP-UX, AIX, Solaris, Windows, và Darwin (Mac OS X).

Trên các hệ điều hành Unix-based, Wazuh agent sẽ chạy nhiều tiến trình riêng và giao tiếp với nhau thông qua local Unix domain socket. Một trong số các tiến trình này sẽ chịu trách nhiệm giao tiếp và gửi dữ liệu đến Wazuh server. Đối với các hệ điều hành Windows, chỉ có một tiến trình agent duy nhất sử dụng mutex để chạy đa nhiệm các tác vụ. Các tác vụ hoặc tiến trình khác nhau của agent sẽ chịu trách nhiệm giám sát khác nhau (chẳng hạn file kiểm tra tính toàn vẹn của file, đọc log, quét các cấu hình hệ thống).



### Một số tính năng chính trong Wazuh

#### **Phát** **hiện xâm nhập (IDS/IPS)**

Wazuh cung cấp khả năng phát hiện và ngăn chặn các hoạt động không mong muốn hoặc độc hại trong môi trường mạng. Các tính năng cụ thể bao gồm:

Phát hiện dựa trên chữ ký (Signature-based Detection):

* Sử dụng một tập hợp các chữ ký (signatures) đã được xác định trước để phát hiện các mẫu tấn công phổ biến. Đây là phương pháp phát hiện nhanh chóng và hiệu quả đối với các mối đe dọa đã biết.
* Phát hiện dựa trên hành vi (Behavior-based Detection): Phân tích hành vi của hệ thống và người dùng để phát hiện các hoạt động bất thường. Phương pháp này hiệu quả đối với các mối đe dọa mới và chưa biết.
* Tích hợp với Suricata: Wazuh tích hợp với Suricata, một công cụ IDS/IPS mạnh mẽ, để tăng cường khả năng phát hiện và ngăn chặn xâm nhập.

#### Giám sát, báo cáo và phân tích nhật ký (Log monitoring, reporting and analysis)

Wazuh cho phép giám sát, báo cáo và phân tích các nhật ký từ nhiều nguồn khác nhau, bao gồm hệ thống, ứng dụng và thiết bị mạng. Các tính năng chính bao gồm:

* Thu thập nhật ký đa nguồn: Hỗ trợ thu thập nhật ký từ nhiều nguồn khác nhau như hệ điều hành (Windows, Linux, macOS), ứng dụng (web server, database), và thiết bị mạng (firewall, router).
* Phân tích nhật ký thời gian thực: Wazuh cung cấp khả năng phân tích nhật ký trong thời gian thực để phát hiện các sự kiện bảo mật và các hoạt động bất thường.
* Báo cáo tùy chỉnh: Người dùng có thể tạo các báo cáo tùy chỉnh dựa trên các tiêu chí cụ thể, giúp dễ dàng theo dõi và phân tích các sự kiện bảo mật.

#### Hệ thống xác thực và quản lý (Authentication and Management)

Wazuh cung cấp các công cụ để quản lý người dùng, xác thực, và quản lý quyền truy cập vào hệ thống. Các tính năng chính bao gồm:

* Quản lý người dùng: Cho phép tạo, xóa và quản lý người dùng với các quyền hạn khác nhau.
* Xác thực đa yếu tố (Multi-Factor Authentication - MFA): Hỗ trợ xác thực đa yếu tố để tăng cường bảo mật cho việc truy cập vào hệ thống.
* Quản lý quyền truy cập: Cung cấp các công cụ để quản lý quyền truy cập dựa trên vai trò (Role-Based Access Control - RBAC), đảm bảo chỉ những người dùng có thẩm quyền mới có thể truy cập vào các phần nhạy cảm của hệ thống.

#### Hệ thống cảnh báo (Alerting System)

Hệ thống cảnh báo của Wazuh cho phép tạo ra các cảnh báo khi phát hiện các hoạt động đáng ngờ hoặc vi phạm các quy tắc được định nghĩa trước. Các tính năng chính bao gồm:

* Cảnh báo thời gian thực: Hệ thống có khả năng phát hiện và cảnh báo các sự kiện bảo mật trong thời gian thực, giúp quản trị viên phản ứng kịp thời.
* Tùy chỉnh cảnh báo: Người dùng có thể thiết lập các quy tắc cảnh báo tùy chỉnh dựa trên các tiêu chí cụ thể như loại sự kiện, mức độ nghiêm trọng, và nguồn gốc sự kiện.
* Thông báo đa dạng: Hỗ trợ gửi thông báo qua nhiều kênh khác nhau như email, SMS, hoặc tích hợp với các hệ thống thông báo khác như Slack, PagerDuty.

#### Quản lý sự kiện và đưa ra phản hồi (Event management and response)

Wazuh cung cấp khả năng quản lý sự kiện và đưa ra phản ứng nhanh chóng đối với các sự kiện bảo mật. Các tính năng chính bao gồm:

* Quản lý sự kiện: Hệ thống tập trung các sự kiện bảo mật từ nhiều nguồn khác nhau, cung cấp một cái nhìn toàn diện về tình trạng bảo mật của hệ thống.
* Phản ứng tự động: Có khả năng thực hiện các hành động tự động khi phát hiện các sự kiện bảo mật cụ thể, như chặn địa chỉ IP, tắt tài khoản người dùng hoặc gửi cảnh báo đến quản trị viên.
* Tích hợp với công cụ phản ứng: Wazuh có thể tích hợp với các công cụ phản ứng bảo mật khác như Ansible, Puppet để thực hiện các hành động phản ứng tự động và phức tạp.

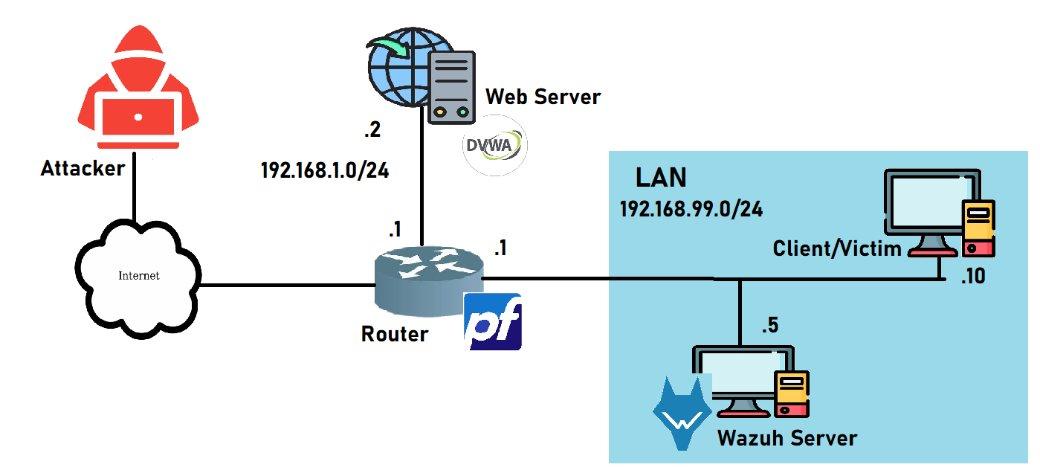
### Ưu và nhược điểm khi triển khai Wazuh

* Ưu điểm:
  + Mã nguồn mở: miễn phí, có thể tuỳ chỉnh theo nhu cầu
  + Tính năng toàn diện: kết hợp nhiều tính năng bảo mật trong một nền tảng
  + Tích hợp dễ dàng: hỗ trợ tích hợp với nhiều công cụ và nền tảng khác
* Nhược điểm:
* Yêu cầu cấu hình ban đầu cao: cần thời gian để thiết lập và cấu hình ban đầu
* Độ phức tạp: có thể phức tạp với người mới bắt đầu hoặc không chuyên về bảo mật mạng

# TRIỂN KHAI HỆ THỐNG

## Triển khai môi trường cần thiết

Mô hình hệ thống mà nhóm em triển khai như sau:



Trong hệ thống này gồm có:

* Máy Attacker: Một máy ảo chạy hệ điều hành Kali Linux được xem như một kẻ tấn công. Ta sẽ thực hiện các hành vi tấn công khác nhau vào hệ thống rồi sử dụng hệ thống Wazuh để phát hiện và ngăn chặn các hành vi này. Attacker nằm ngoài hệ thống mạng nội bộ, được kết nối vào hệ thống mạng thông qua Internet
* Router: Một máy ảo FreeBSD được sử dụng để cài đặt pfSense, một phần mềm mã nguồn mở dùng để quản lí và bảo mật mạng, cung cấp chức năng định tuyến. Ta xem máy này như một router quản lý các kết nối mạng từ mạng nội bộ ra bên ngoài Internet và ngược lại.
* Web Server: Máy ảo có cài đặt DVWA (Damn Vulnerable Web Application), một ứng dụng web được cố tình thiết kế với các lỗ hổng bảo mật để ta có thể thực hiện việc khai thác các lỗ hổng đó.
* LAN (Mạng nội bộ): chứa hai máy chính:

1. Máy Client/Victim: Đây là máy của người dùng cuối (endpoint) trong mạng nội bộ, sẽ bị Attacker thực hiện các hành vi tấn công xâm nhập trái phép vào máy. Ở đây chúng em sử dụng hệ điều hành Ubuntu.
2. Máy Wazuh Server: Máy ảo Ubuntu cài đặt và chạy Wazuh Server dùng để giám sát các sự kiện bảo mật trong mạng nội bộ, phát hiện và ngăn chặn kịp thời các sự kiện đáng ngờ xảy ra.

Cách thức hoạt động của mô hình:

* Attacker sử dụng Internet để gửi các yêu cầu tấn công vào máy Client hoặc Web Server chạy DVWA. Attacker sẽ khai thác những lỗ hổng mà nó phát hiện được để xâm nhập vào hệ thống.
* Router đóng vai trò là cầu nối giữa mạng ngoài và mạng nội bộ, kiểm soát các luồng dữ liệu ra vào trên hệ thống, pfSense trên Router sẽ cung cấp các tính năng bảo mật để ngăn chặn các tấn công mạng.
* Wazuh Server sẽ giám sát toàn bộ hoạt động trong mạng nội bộ, phân tích log và phát hiện các hành vi bất thường. Khi phát hiện một cuộc tấn công hoặc hoạt động khả nghi, Wazuh sẽ gửi cảnh báo để quản trị viên mạng có thể kịp thời phản ứng đồng thời có thể ngăn chặn luôn cuộc tấn công.

Bảng địa chỉ IP

| Hostname | IP Address | Netmask | Gateway | DNS |
| --- | --- | --- | --- | --- |
| Router | 192.168.1.1 | 255.255.255.255.0 | 162.168.1.1 | 8.8.8.8 |
| Web Server | 192.168.1.2 | 255.255.255.255.0 | 162.168.1.1 | 8.8.8.8 |
| Wazuh Server | 192.168.99.5 | 2525.255.255.255.0 | 192.168.99.1 | 8.8.8.8 |
| Client | 192.168.99.10 | 255.255.255.255.0 | 192.168.99.1 | 8.8.8.8 |
| Attacker |  | 255.255.255.255.0 |  |  |

Cấu hình phần cứng tối thiểu để có thể cài đặt môi trường:

| Tên máy | RAM (GB) | DISK (GB) |
| --- | --- | --- |
| Router | 1 | 10 |
| Web Server | 2 | 10 |
| Wazuh Server | 2 | 10 |
| Client | 2 | 10 |
| Attacker | 4 | 15 |

Máy Attacker ở đây nhóm triển khai máy ảo Kali Linux. Sau khi lập bảng, chúng em dựa vào đó để tiến hành cài đặt các máy. Sau khi cài đặt cơ bản xong thì thực hiện set up card mạng theo bảng để đảm bảo các máy trong mô hình có thể giao tiếp được với nhau.

## Cài đặt Wazuh

* Cài đặt Wazuh Server

Nhóm tiến hành cài đặt và cấu hình Wazuh Server theo single-node cluster. Wazuh Server là thành phần chính, trong đó chúng em sẽ cài đặt thêm Wazuh manager và Filebeat. Wazuh manager sẽ thu thập và phân tích dữ liệu đã được deploy trên Wazuh agent, kích hoạt các cảnh bảo khi phát hiện các mối đe doạ hoặc bất thường. Filebeat sẽ chuyển tiếp các cảnh báo một cách an toàn và lưu trữ những sự kiện diễn ra.

Cấu hình tối thiểu: thường được cài đặt trên hệ điều hành Linux 64-bit với ít nhất 2GB RAM, 2 core CPU và được khuyến khích 4GB RAM, 8 core CPU. Ngoài ra cần phải trống sẵn tầm 50GB dung lượng ổ đĩa để lưu trữ dữ liệu thu thập được.

Các bước chính:

1. Thêm kho lưu trữ cho Wazuh: bằng cách sử dụng khoá GPG để thêm kho lưu trữ trực tiếp từ Wazuh
2. Cài đặt Wazuh manager: tiến hành cài đặt theo trình tự được hướng dẫn
3. Cài đặt Filebeat: tiến hành cài đặt theo trình tự được hướng dẫn
4. Cấu hình Filebeat: tải file cấu hình sẵn từ web Wazuh rồi chỉnh sửa file config sao cho phù hợp với cấu hình của máy
5. Triển khai các chứng chỉ: folder.rar chứa các chứng chỉ sẽ được tạo tự động trong quá trình cài đặt. Chỉnh sửa các thông số phù hợp với máy rồi chuyển chứng chỉ đến vị trí tương ứng của nó.

* Cài đặt Wazuh Dashboard:

Đây là một thành phần chính của Wazuh. Dashboad có sẵn một giao diện web linh hoạt và trực quan để ta có thể dễ dàng khai thác, phân tích và trực quan hoá dữ liệu về bảo mật. Người dùng có thể dễ dàng dùng được ngay mà không cần tốn quá nhiều thời gian để tìm hiểu, cho phép điều hướng liền mạch thông qua giao diện của người dùng. Với Wazuh Dashboard, người dùng có thể trực quan hoá các sự kiện bảo mật, các lỗ hổng được phát hiện, dữ liệu giám sát tính toàn vẹn của tập tin, các sự kiện giám sát cơ sở hạ tầng đám mây và các tiêu chuẩn tuân thủ theo quy định.

Cấu hình tối thiểu: thường được cài đặt trên hệ điều hành Linux 64-bit với ít nhất 4GB RAM, 2 core CPU và được khuyến khích 8GB RAM, 4 core CPU. Về phần trình duyệt thì Wazuh Dashboard hỗ trợ các trình duyệt Chrome, Firefox và Safari. Nhóm chúng em thực hiện cài đặt Wazuh Dashboard trên trình duyệt Firefox.

Các bước chính:

1. Tải về các gói phụ thuộc (dependencies) cần thiết: “debhelper” từ version 9 trở về sau để hỗ trợ quá trình xây dựng và đóng gói các thành phần của Wazuh Dashboard từ mã nguồn của các gói Debian cài đặt được; “curl” để tải về các tệp cài đặt cần thiết cho Wazuh Dashboard; “tar” dùng cho việc nén hoặc giải nén các tệp cấu hình dữ liệu của Wazuh Dashboard; “libcap2-bin” thì được sử dụng để quản lí quyền của các tiến trình hoặc dịch vụ mà Wazuh Dashboard chạy nhằm đảo bảo nó có quyền cần thiết mà không cần phải chạy trong root.
2. Thêm kho lưu trữ: vì Wazuh Dashboard tụi em đã cài trong cùng host với Wazuh Server nên không cần phải cài đặt thêm gì vì nó đã có sẵn khi cài đặt xong Wazuh Server
3. Cài đặt gói Wazuh Dashboard
4. Cấu hình cho Wazuh Dashboard: chỉnh sửa các giá trị trong file config sao cho phù hợp với cấu hình của hệ thống đang sử dụng
5. Triển khai các chứng chỉ: folder .rar chứa các chứng chỉ sẽ được tạo tự động trong quá trình cài đặt. Chỉnh sửa các thông số phù hợp với máy rồi chuyển chứng chỉ đến vị trí tương ứng của nó.
6. Bảo mật cho hệ thống Wazuh: sau khi cài đặt xong thì tiến hành thay đổi các thông tin xác thực mặc định có sẵn khi cài đặt để tránh khỏi việc bị tấn công một cách dễ dàng.

* Cài đặt Wazuh Agent:

Wazuh Agent là một ứng dụng đa nền tảng và chạy trên các endpoint mà người dùng muốn giám sát nó. Wazuh Agent sẽ giao tiếp với Wazuh Server, gửi dữ liệu gần như trong thời gian thực thông qua một kênh truyền đã được mã hoá và xác thực. Nó được phát triển dựa trên nhu cầu giám sát nhiều loại endpoint khác nhau mà không làm ảnh hưởng đến hiệu suất của chúng. Wazuh Agent hỗ trợ trên các hệ điều hành phổ biến và chỉ yêu cầu trung bình từ 35MB RAM đổ lên.

Wazuh Agent cung cấp các tính năng chính để tăng cường bảo mật cho hệ thống như Log Collector (Trình thu thập nhật kí), Command Execution (Thực thi lệnh), File integrity monitoring (Giám sát tính toàn vẹn của tệp), Security configuration assessment (Đánh giá cấu hình bảo mật), System inventory (Kiểm kê hệ thống), Malware Detection (Phát hiện phần mềm độc hại), Active Response (Phản hồi tích cực), Container Security (Bảo mật container), Cloud security (Bảo mật đám mây)

Các bước chính:

1. Thêm kho lưu trữ Wazuh: tương tự như các thành phần Wazuh đã cài đặt ở trên
2. Triển khai Wazuh Agent: Ta vào Wazuh Dashboard rồi tiến hành đăng kí Agent cho nó, sau đó màn hình sẽ xuất hiện một command prompt để ta gõ vào máy Agent.
3. Tắt cập nhật Wazuh Agent: để hệ thống được đảm bảo hoạt động ổn định thì Wazuh Server luôn chạy phiên bản mới hơn hoặc bằng với phiên bản của Wazuh Agent.

## Triển khai Wazuh

Sau khi đã cài đặt xong môi trường cần thiết thì tiến hành thực hiện các kịch bản mà nhóm đã đề ra trước đó, theo thứ tự từ cơ bản đến nâng cao. Tất cả các máy đã được cài đặt theo như mô hình đề ra đều được sử dụng để phục vụ các kịch bản khác nhau, giúp ta nhìn thấy rõ hơn cách hoạt động, chức năng cảnh báo bảo mật và ngăn chặn các cuộc tấn công xâm nhập bất hợp pháp của Wazuh. Nhóm sẽ thiết lập các chính sách giám sát và quy tắc phát hiện xâm nhập (IDS) trên Wazuh Server. Các quy tắc này sẽ được tùy chỉnh để phù hợp với các yêu cầu bảo mật cụ thể của hệ thống, bao gồm việc giám sát các hoạt động đáng ngờ, kiểm tra tính toàn vẹn của tệp tin và phân tích hành vi mạng.

Cuối cùng, nhóm sẽ thực hiện kiểm tra và tinh chỉnh hệ thống Wazuh để đảm bảo rằng nó hoạt động hiệu quả và có thể phát hiện kịp thời các mối đe dọa. Điều này bao gồm việc kiểm tra các log thu thập được, hiệu chỉnh các quy tắc giám sát, và thực hiện các bài kiểm tra xâm nhập giả định để xác minh khả năng phản ứng của hệ thống. Bằng cách này, nhóm đảm bảo rằng hệ thống Wazuh sẽ cung cấp một lớp bảo mật vững chắc cho mạng nội bộ, giúp phát hiện và ngăn chặn các cuộc tấn công từ bên ngoài cũng như các hoạt động bất thường bên trong mạng.

# HIỆN THỰC HỆ THỐNG

Chương này sẽ trình bày chi tiết về cách thức nhóm đã hiện thực lần lượt các kịch bản đã đề ra.

## Kịch bản 1: Tấn công Brute Force vào máy chủ SSH trên Web Server

1. Trên Wazuh Server:

Wazuh đi kèm với một tập hợp các script mặc định được sử dụng trong active response (phản ứng chủ động). Các script này được đặt trong thư mục /var/ossec/active-response/bin/ trên các điểm cuối Linux/Unix. Script phản ứng chủ động firewall-drop hoạt động trên các hệ điều hành Linux/Unix, sử dụng iptables để chặn các địa chỉ IP độc hại.

Đầu tiên ta sẽ config lại command firewall-drop (/var/ossec/etc/ossec.conf).

Trong khối <command> bao gồm các thống tin về những hành động được thực hiện trên Wazuh Agent:

<name>: Đặt tên cho lệnh giúp dễ dàng nhận diện và quản lý các lệnh phản ứng chủ động, trong trường hợp này ta đặt là “firewall-drop”

<executable>:Chỉ định script hoặc chương trình thực thi sẽ được kích hoạt. Trong trường hợp này, firewall-drop là một script được sử dụng để tương tác với iptables nhằm chặn địa chỉ IP độc hại.

<timeout\_allowed>: cho phép lệnh có thể có thời gian chờ (timeout) sau một khoảng thời gian nhất định. Khi được đặt là “yes”, điều này cho thấy active response có thể duy trì trạng thái trong một khoảng thời gian và sau đó sẽ tự động dừng hoặc thay đổi trạng thái khi hết thời gian chờ.

A screenshot of a computer

Description automatically generated

Tiếp đến ta sẽ config khối active-response (/var/ossec/etc/ossec.conf) với các giá trị cụ thể như sau:

<command>: tên lệnh cần cấu hình. Ở đây là “firewall-drop” đã được định nghĩa ở trên. Lệnh này sẽ được gọi khi một sự kiện cụ thể được kích hoạt.

<location>: chỉ định nơi lệnh sẽ được thực thi. Khi sử dụng giá trị “local”, lệnh sẽ được thực thi trên điểm cuối (endpoint) đang được giám sát, nơi sự kiện kích hoạt xảy ra. Điều này có nghĩa là nếu có một cuộc tấn công brute-force SSH trên máy chủ được giám sát, lệnh “firewall-drop” sẽ được thực thi ngay trên máy chủ đó để chặn địa chỉ IP của Attacker

<rules\_id>: mô-đun active response sẽ thực thi lệnh nếu quy tắc ID 5763 - "SSHD brute force trying to get access to the system" được kích hoạt. Điều này có nghĩa là khi Wazuh phát hiện hành vi tấn công brute-force SSH thông qua quy tắc này, nó sẽ thực thi lệnh để chặn địa chỉ IP của attacker.

<timeout>: Chỉ định thời gian hành động phản ứng chủ động phải kéo dài bao lâu. Trong trường hợp sử dụng này, mô-đun sẽ chặn địa chỉ IP của attacker trong 180 giây. Điều này có nghĩa là địa chỉ IP sẽ bị chặn trong khoảng thời gian đó, sau đó có thể được mở chặn nếu không có hành vi tấn công tiếp theo.

A screenshot of a computer

Description automatically generated

Khởi động lại wazuh server để các thay đổi được lưu lại



Ping xác nhận kết nối giữa hai máy

A screenshot of a computer

Description automatically generated

Chuẩn bị file password.txt gồm 10 mật khẩu ngẫu nhiên. Dùng hydra để thực hiện tấn công brute-force

A screenshot of a computer

Description automatically generated

Sau khi tấn công xong, máy Kali bị chặn tạm thời vì active response đã chặn IP của máy Kali trong thời gian ngắn

A screenshot of a computer

Description automatically generated

Bằng chứng là số icmp\_seq tăng đột ngột từ 12 -> 33

A screenshot of a computer

Description automatically generated

Wazuh Server ghi nhận lại log của sự kiện trên

A screenshot of a computer

Description automatically generated

## Kịch bản 2: Phát hiện kết nối mạng dị thường

Trong kịch bản này, ta sẽ dung CDB list. CDB list là file văn bản dung để lưu danh sách người dung, file hash, địa chỉ IP, domain name. CDB list có dạng key:value hoặc key:. List này có thể được dùng như whitelist hoặc blacklist

## Ta sẽ cài đặt trên server trước:

- Đầu tiên ta tạo common port list gồm cái port thường gặp tại đường dẫn /var/ossec/etc/lists

A computer screen shot of a computer

Description automatically generated

Tiếp đến ta config file ownership và permission để Wazuh Server có thể đọc được list port này



Thêm CDB list vừa tạo vào ruleset: Cấu hình file tại /var/ossec/etc/ossec.conf, ta thêm dòng sau vào tag <ruleset>

<list>etc/lists/common-ports</list>

A screenshot of a computer

Description automatically generated

Trong cùng file này thêm tag command cùng active response. Active response sẽ chứa lệnh để thực thi khi rule ID trong nó được cảnh báo

Lần lượt thêm 2 dòng sau vào 2 tag tương ứng:

<command>

<name>closing-port</name>

<executable><closing-port.sh</executable>

<timeout\_allowed>no</timeout\_allowed>

</command>

<active-response>

<disabled>no</disabled>

<command>closing-port</command>

<location>local</location>

<rules\_id>100370</rules\_id>

</active-response>

A screenshot of a computer

Description automatically generated

Thêm rule bên dưới vào /var/ossec/etc/rules/local\_rules.xml để tạo alert khi không trùng các port trong list

A computer screen shot of a program

Description automatically generated

Khởi động lại máy wazuh-server để nó lưu các thay đổi và thực hiện



**Trên máy Agent:**

Chỉnh lại thông số của Agent ở thẻ interval thành <interval>1m</interval> và <ports all=”yes”>yes</ports>

Như vậy cứ mỗi phút, agent sẽ quét tất cả các port và quét cả những port đang mở

A screenshot of a computer program

Description automatically generated

Tạo shell script closing-port.sh tại /var/ossec/active-response/bin/closing-port.sh. Script này sẽ đóng port nếu port đang mở và không nằm trong file của common-port. Input của script là alert được tạo ra ở server trong bước trên, dưới dạng JSON

A screenshot of a computer

Description automatically generated

Khởi động lại agent để lưu các thay đổi



Tiếp theo bật netcat và lắng nghe ở port 4567 tại máy Client

A screenshot of a computer

Description automatically generated

Sau một lúc ta thấy tiến trình bị terminated

A screenshot of a computer

Description automatically generated

Trên Wazuh dashboard ta nhận được báo cáo port lạ 4567 được mở

A screenshot of a computer

Description automatically generated

## Kịch bản 3: **Phát hiện các thay đổi trái phép trên hệ thống tập tin**

* Trong kịch bản này ta sử dụng module File Integrity Monitoring (FIM) của Wazuh để giám sát các thay đổi trong thư mục /root
* Máy ảo sử dụng: Router, Wazuh server và Client
* Đầu tiên, tại file /var/ossec/etc/ossec.conf trên máy client, ta thêm tag để giám sát thư mục /root

<directoriescheck\_all="yes"report\_changes="yes"realtime="yes">/root</directories> A screenshot of a computer

Description automatically generated

* Khởi động lại wazuh agent để áp dụng các thay đổi



* Ta lần lượt tạo file, thay đổi nội dung file và xóa file tại /root trên máy Client

A screenshot of a computer

Description automatically generated

* Kiểm tra với Wazuh Server ta thấy có 3 cảnh báo được bật lên với các ID 554, 550, 553 tương ứng với 3 hành động ở trên A screenshot of a computer

  Description automatically generated

## Kịch bản 4: Phát hiện lỗ hổng SQL Injection trên Web Server DVWA

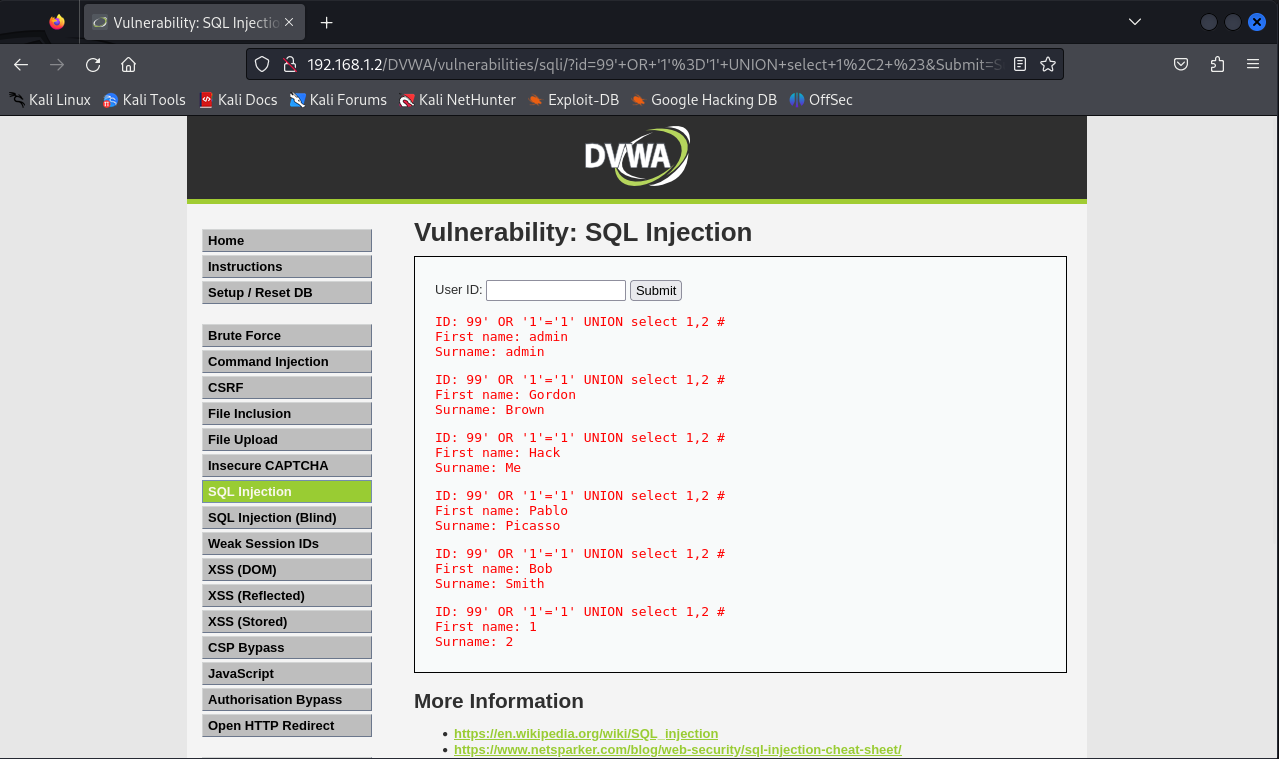
* Ở kịch bản này, sử dụng module module File Integrity Monitoring (FIM) của Wazuh để giám sát các thay đổi trong thư mục access.log của Apache2
* Máy ảo sử dụng: Router, Wazuh server, Web Server, Attacker
* Đầu tiêm ta truy cập vào DVWA được hosting trên Web Server



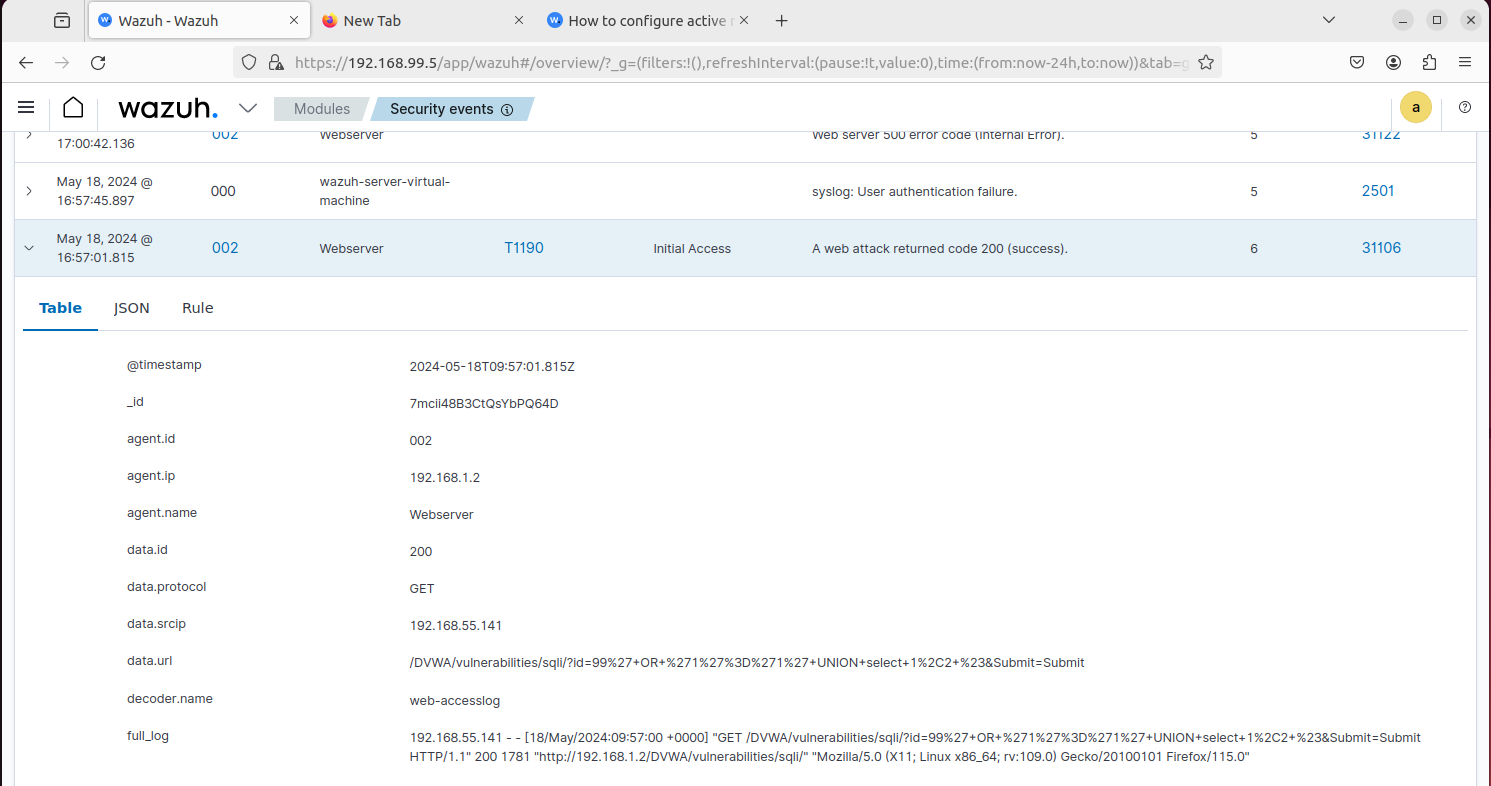
* Thực hiện exploit SQL injection để khai thác thông tinh bằng payload:

99' OR '1'='1' UNION select 1,2 #

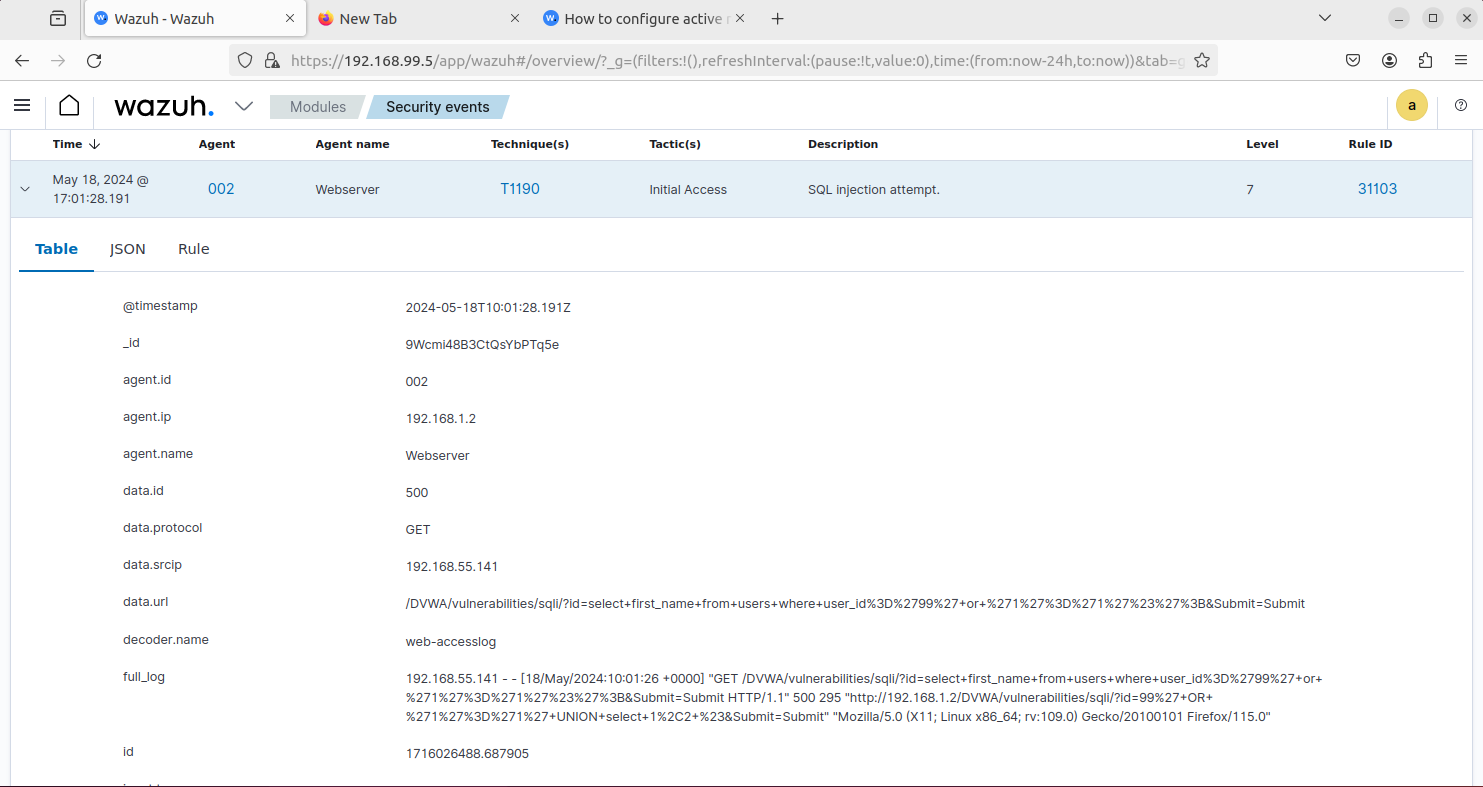
* Sau khi sử dụng payload thì dữ liệu đã bị lộ



* Kiểm tra trên server xuất hiện cảnh báo web attack thành công:



* Hoặc nếu ta sử dụng payload: select first\_name from users where user\_id='99' or '1'='1'#';
* Server cũng xuất hiện thông báo về SQL injection attempt



## Kịch bản 5: Phát hiện tấn công Shellshock trên Web Server DVWA (CVE-2014-6271)

* Ở kịch bản này, sử dụng module module File Integrity Monitoring (FIM) của Wazuh để giám sát các thay đổi trong thư mục access.log của Apache2
* Máy ảo sử dụng: Router, Wazuh server, Web Server, Attacker
* Đầu tiên , ta tiến hành cài đặt apache2 tại máy Webserver và bật apache2

A screenshot of a computer

Description automatically generated

* Tại /var/ossec/etc/ossec.conf trên máy Web Server, ta thêm dòng dòng giám sát file access.log của apache2



* Khởi động lại agent để áp dụng các thay đổi



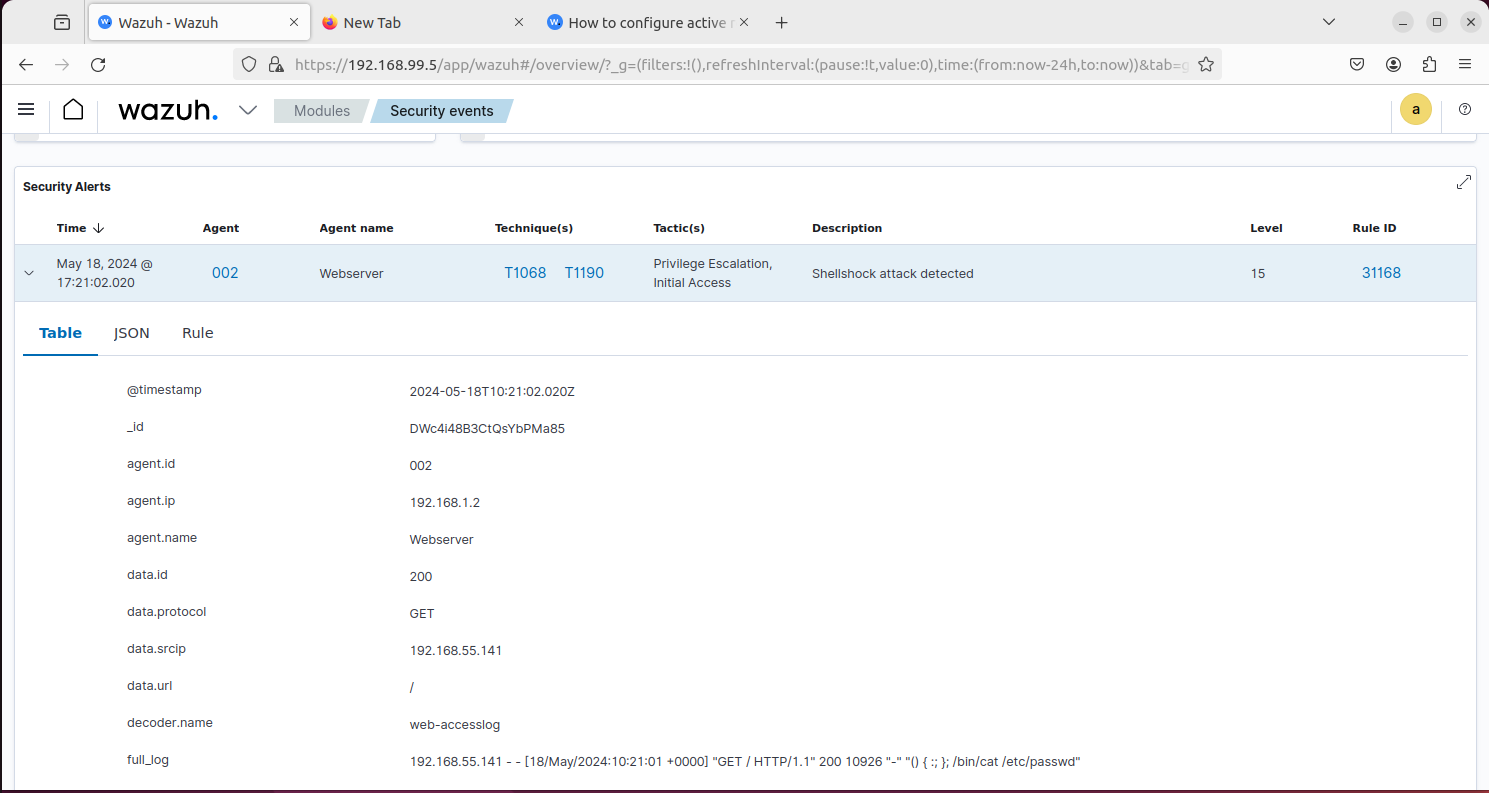
* Trên máy Attacker ta thực hiện lệnh curl để tấn công

sudo curl -H "User-Agent: () { :; }; /bin/cat /etc/passwd" 192.168.1.2

A screenshot of a computer

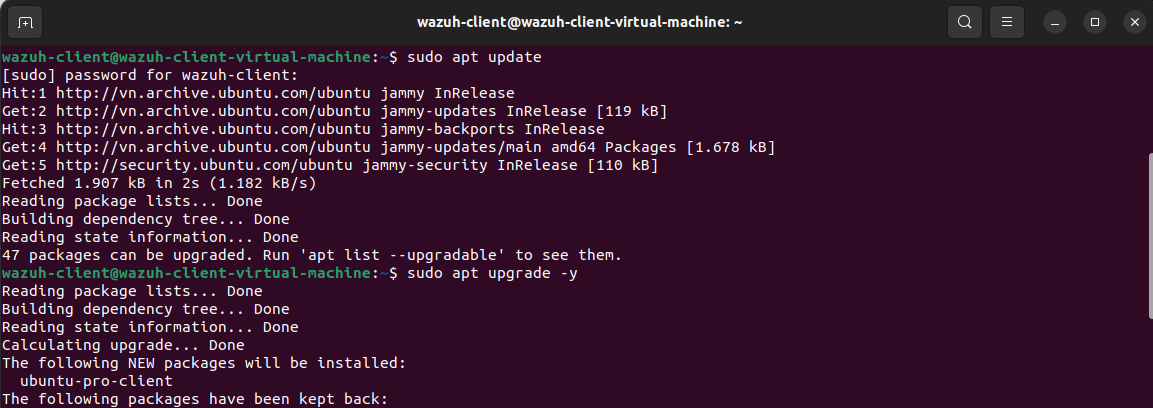
Description automatically generated

* Trên Server lúc này cũng xuất hiện thông báo tấn công shellshock attack

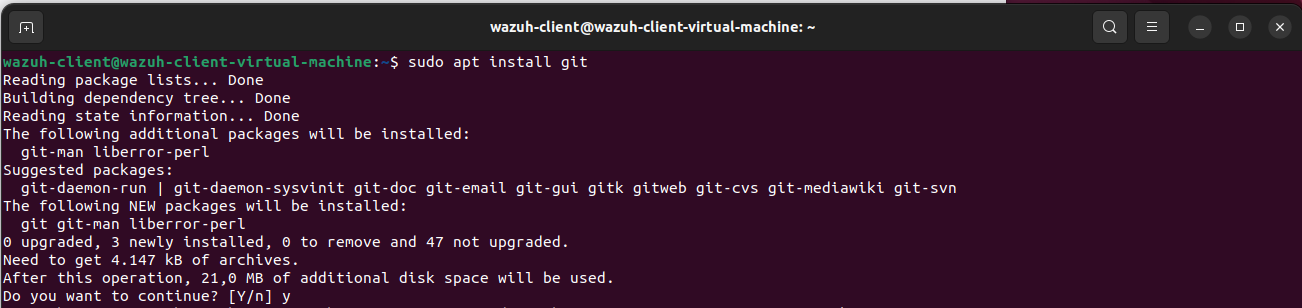


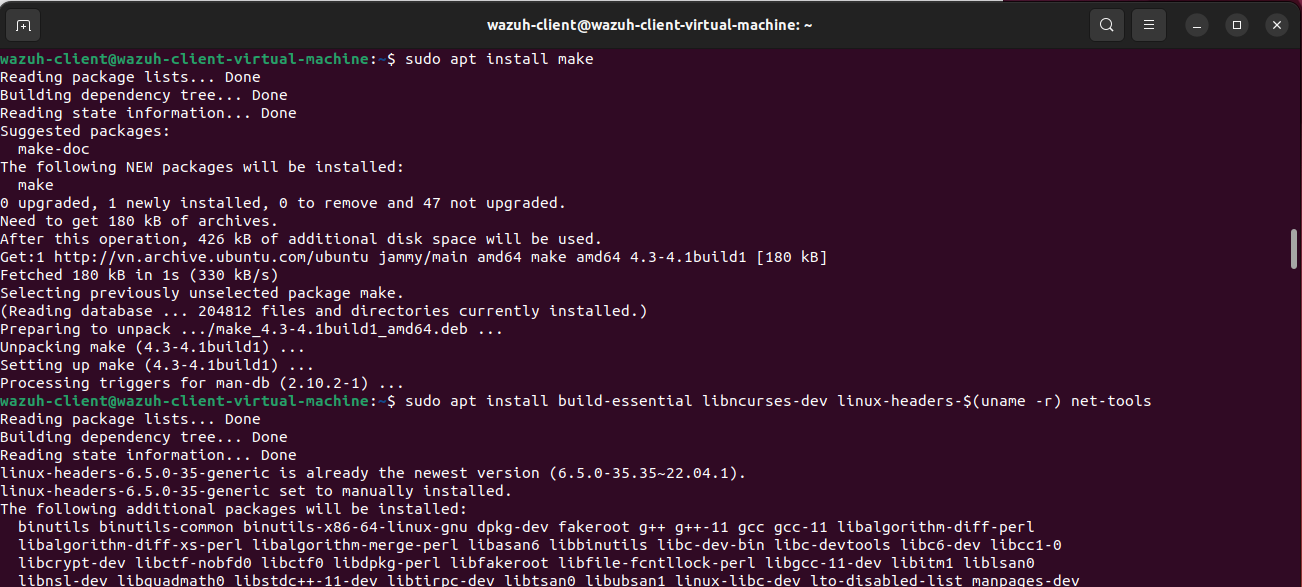
## Kịch bản 6: Phát hiện rootkit Diamorphine trên hệ thống

* Ở kịch bản này, sử dụng Wazuh rootcheck để kiểm tra các điểm bất thường trên máy client
* Máy ảo sử dụng: Router, Wazuh server và Client
* Trên máy Client , tiến hành cập nhật và nâng cấp các packages

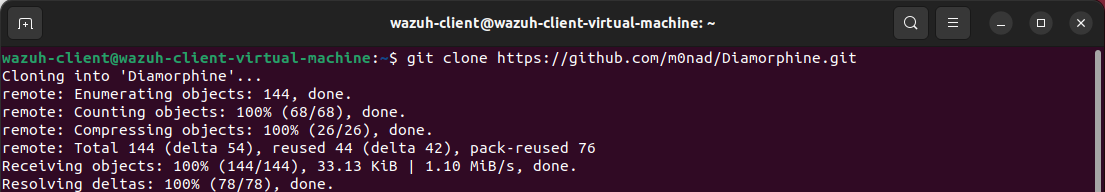


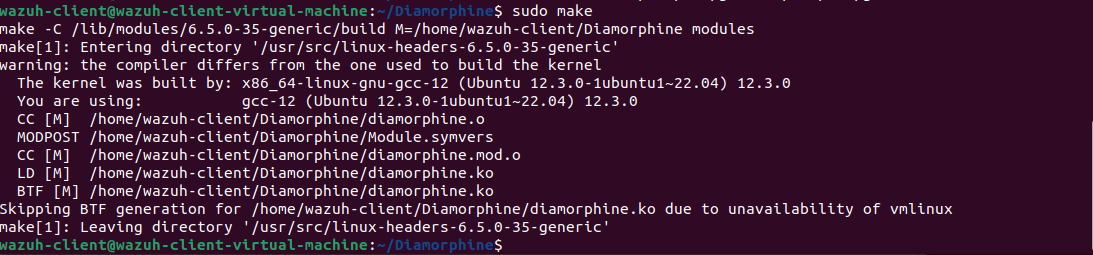
* Cài đặt các package cần thiết





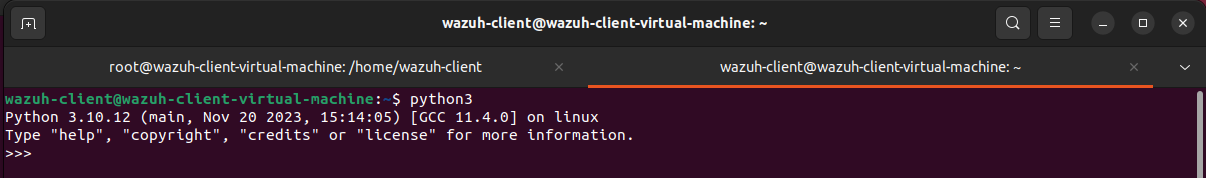
* Sau đó ta tải Diamorphine từ Github và tiến hành cài đặt



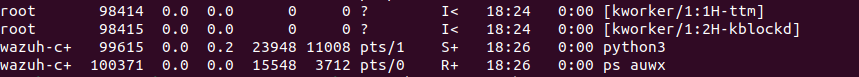


Đang chèn hình ảnh...

* Mở một terminal khác và khởi chạy python3 trong terminal



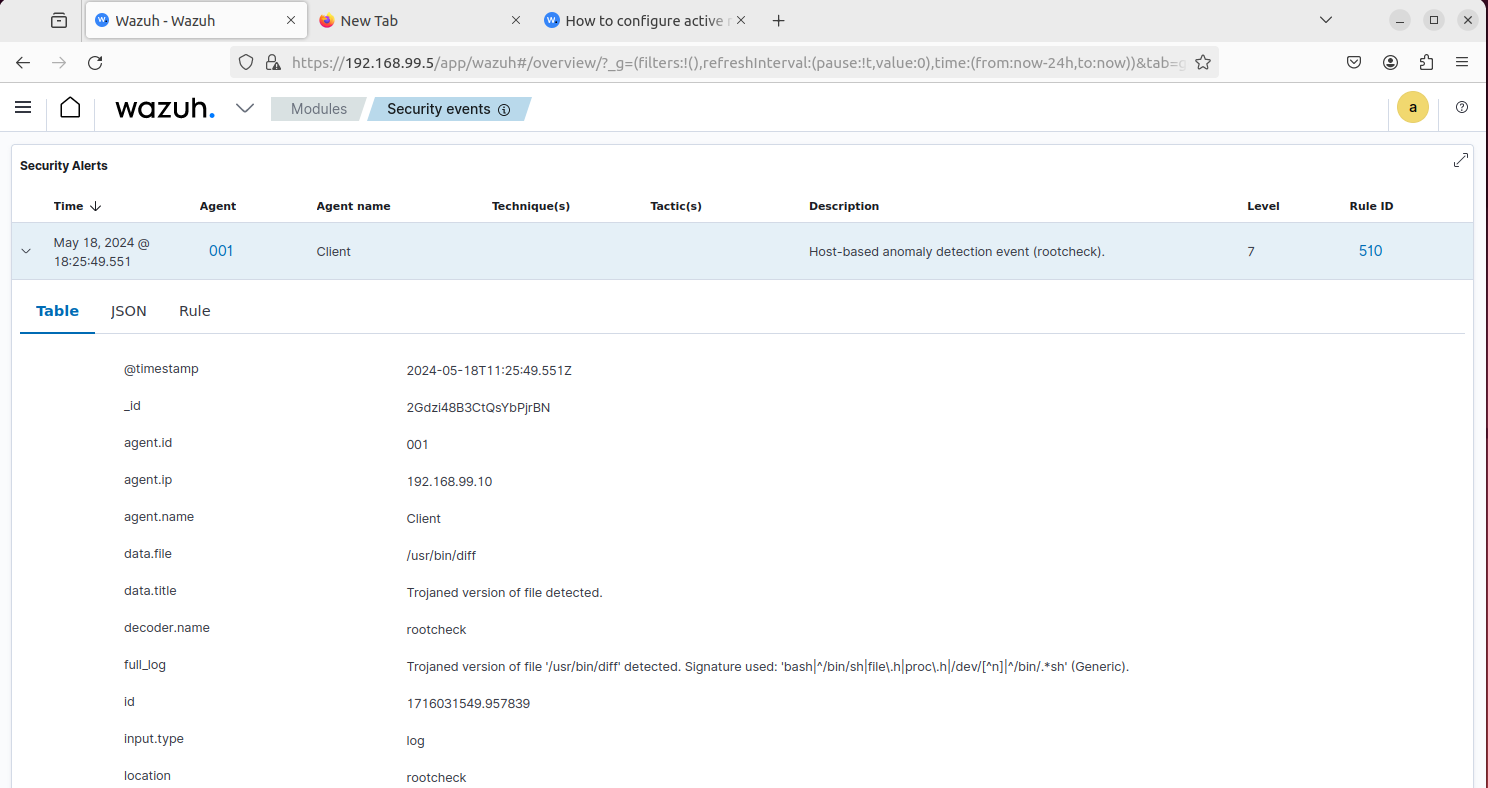
* Chạy ps -auwx để tìm PID của python3. Trong trường hợp này là 99615



* Chạy lệnh kill -63 99615 để ẩn tiến trình này

Đang chèn hình ảnh...

* Trên server lúc này cũng xuất hiện cảnh báo về rootcheck bất thường



## Wazuh phát hiện và loại bỏ phần mềm độc hại bằng cách tích hợp VirusTotal

* Ở kịch bản này, sử dụng module File Integrity Monitoring (FIM) của Wazuh để giám sát các thay đổi trong thư mục và API VirusTotal để quét các tệp trong thư mục.
* Máy ảo sử dụng: Router, Wazuh server và Client
* Đầu tiên, trong file config */var/ossec/etc/ossec.conf* ở trên agent wazuh máy client. Kiểm tra block *<syscheck> → <disable> đã được set về ‘no’*; điều này đảm bảo cho FIM giám sát được các thay đổi trong thư mục
* Sau đó, vẫn trên agent, dùng apt *cài đặt jq*, một tiện ích xử lý đầu vào dạng JSON từ active response script (sudo apt -y install jq)
* Tạo script remove\_threat.sh trên agent với đường dẫn: */var/ossec/active-response/bin/remove-threat.sh* có các luồng hoạt động chính:
* Thiết lập môi trường:
* LOCAL được đặt thành thư mục hiện tại của script.
* Chuyển đến thư mục của script và sau đó chuyển đến thư mục cha.
* PWD được đặt thành đường dẫn hiện tại (từ thư mục cha của script).
* Đọc dữ liệu JSON đầu vào:

Script đọc dữ liệu JSON từ đầu vào chuẩn (standard input) vào biến INPUT\_JSON.

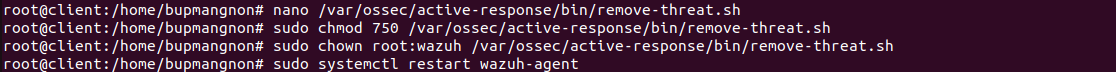
* Trích xuất thông tin từ JSON:
* FILENAME được lấy từ trường .parameters.alert.data.virustotal.source.file trong JSON, chứa tên tệp cần loại bỏ.
* COMMAND được lấy từ trường .command trong JSON, chỉ định lệnh cần thực hiện.
* LOG\_FILE là đường dẫn đến tệp log của phản ứng chủ động.
* Xử lý lệnh:
* Nếu COMMAND là "add", script sẽ gửi một thông báo điều khiển đến execd và chờ phản hồi.
* Nếu phản hồi (COMMAND2) không phải là "continue", script sẽ ghi một thông báo vào tệp log và kết thúc.
* Loại bỏ tệp:
* Script sẽ xóa tệp được chỉ định bởi FILENAME bằng lệnh rm -f.
* Nếu tệp được xóa thành công, script sẽ ghi một thông báo thành công vào tệp log.
* Nếu không, script sẽ ghi một thông báo lỗi vào tệp log.
* Thoát: kết thúc với mã thoát 0.

A computer screen shot of white text

Description automatically generated

Nội dung script remove\_threat.sh trên agent

* Thay đổi quyền sở hữu và quyền truy cập của script trên với owner là root (đọc, ghi, thực thi) và group là wazuh (đọc, thực thi)
* Sau khi chỉnh xong, restart lại wazuh để cập nhật script cho hệ thống



Chỉnh sử quyền sở hữu và truy cập cho script

* Đến máy server wazuh, thêm vào phần được bôi trắng như dưới hình trong file */var/ossec/etc/ossec.conf* với mục đích:
  + Ở block <integration> với mục đích truyền API key của tài khoản VirusTotal, kích hoạt truy vấn đến VirusTotal bất cứ khi nào rule 100200, 100201 xuất hiện
  + Ở block <command> và <active-respond> nhằm kích hoạt active response và trigger script remove\_threat.sh bất cứ khi nào VirusTotal quét và gắn cờ đó là tệp tin độc hại

A screenshot of a computer

Description automatically generated

File */var/ossec/etc/ossec.conf* trên server wazuh

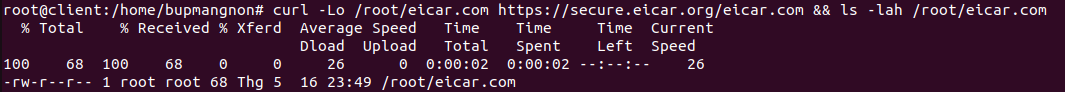
* Thêm các rules vào file */var/ossec/etc/rules/local\_rules.xml* trên server wazuh (phần được bôi trắng như dưới hình) nhằm:
* <group name="syscheck,pci\_dss\_11.5,nist\_800\_53\_SI.7,"> nhằm alert về những thay đổi trong thư mục /root được phát hiện bởi FIM
* <group name="virustotal,"> nhằm alert về kết quả sau khi active response

A screenshot of a computer

Description automatically generated

File */var/ossec/etc/rules/local\_rules.xml* trên server wazuh

* Sau đó restart lại server wazuh (systemctl restart wazuh-manager)
* Demo agent curl 1 website độc hại [https://secure.eicar.org/eicar.com](https://secure.eicar.org/eicar.com%20%20)  và lưu về trong /root/eicar.com.



Curl 1 website độc hại và lưu về local

* Quan sát các alert trên wazuh dashboard, nhận thấy có các rule\_id: 100201 (có chỉnh sửa trong /root); 87105 (VirusTotal scan và gắn flag độc hại); 100092 (script remove\_threat.sh đã loại bỏ file khỏi /root):

A screenshot of a computer

Description automatically generated

Các alert trong wazuh-dashboard sau khi curl tệp tin độc hại

A screenshot of a computer

Description automatically generated

Chi tiết rule\_id 553 phát hiện chỉnh sửa trong /root

A screenshot of a computer

Description automatically generated

Chi tiết rule\_id 100092 loại bỏ tệp tin độc hại khỏi /root

## Kịch bản 8: Wazuh phát hiện tiến trình chưa được chứng thực

* Ở kịch bản này, nhóm sử dụng netcat để demo một tiến trình mới chưa được chứng thực trong hệ thống.
* Máy ảo sử dụng: Router, Wazuh server và Client
* Thêm rules vào */var/ossec/etc/rules/local\_rules.xml* (phần được bôi trắng như hình) với mục đích:
* <rule id="100050" level="0"> lấy danh sách những tiến trình đang chạy
* <rule id="100051" level="7" ignore="900"> phát hiện khi lệnh netcat (nc) được sử dụng để lắng nghe kết nối trên hệ thống

A screenshot of a computer program

Description automatically generated

File */var/ossec/etc/rules/local\_rules.xml* trên server wazuh

* Sau đó, restart lại server wazuh
* Ở trên máy client (agent wazuh), thêm vào file */var/ossec/etc/ossec.conf* block như hình để định kỳ 20s lấy ra danh sách những tiến trình đang chạy:

A screenshot of a computer

Description automatically generated

File */var/ossec/etc/ossec.conf* trên agent client

* Sau đó, restart lại agent
* Trên client, dùng netcat để lắng nghe



Netcat trên client

* Mở wazuh dashboard để xem các alert: rule\_id 100051 phát hiện có tiến trình netcat đang chạy trong hệ thống

A white background with black text

Description automatically generated

Alert phát hiện có tiến trình netcat

## Kịch bản 9: Wazuh phát hiện khai thác lỗ hổng XZ Utils (CVE- 2024-3094)

* XZ Utils là bộ command-line tools được sử dụng rộng rãi cho việc nén, tránh mất mát dữ liệu trên các hệ điều hành giống Unix, bao gồm cả Linux. Nổi bật trong đó là *xz* và *lzma*. Andres Freund đã phát hiện một backdoor trong XZ Utils, *cụ thể là trong thư viện liblzma*, báo cáo lên oss security và được đánh mã **CVE-2024-3094** với điểm CVSS là 10/10. Attacker xâm nhập vào source của project XZ, bí mật cài đặt backdoor vào các tarballs. Các phiên bản bị ảnh hưởng là 5.6.0 và 5.6.1. CVE-2024-3094 có thể cho phép thực thi Remote Code Execution (RCE) thông qua SSH.
* Ở kịch bản này, nhóm sẽ sử dụng PoC (Proof of Concept) [amlweems/xzbot: notes, honeypot, and exploit demo for the xz backdoor (CVE-2024-3094) (github.com)](https://github.com/amlweems/xzbot) cho phép người dùng thiết lập honeypots và máy chủ dễ bị tấn công cũng như thực hiện khai thác CVE này.
* Indicators of Compromise (IoCs) là các tín hiệu hoặc biểu hiện cho thấy một hệ thống hoặc mạng có thể đã bị tấn công hoặc bị xâm nhập. Trong trường hợp lỗ hổng CVE-2024-3094 để thực hiện RCE, các hành động tấn công cho phép attacker thực thi các lệnh sau khi lấy được quyền truy cập trái phép. *Mỗi lần thực thi lệnh sẽ khởi tạo một tiến trình mới.* Các lệnh được thực thi thông qua dịch vụ SSH được dự kiến sẽ *bắt đầu như là các tiến trình con, cháu hoặc các thế hệ con cháu của tiến trình máy chủ SSH (sshd).*
* Để khai thác, trước tiên cần hiểu các đặc điểm của các tiến trình sshd hợp lệ và không hợp lệ. Sau đó, chúng ta phân tích các patterns xuất hiện của các tiến trình được khởi tạo bởi việc khai thác lỗ hổng CVE-2024-3094:
* Các tiến trình con sshd lành tính: hình dưới là cây tiến trình của sshd cha với process id = 765, ở dưới cho thấy có phiên ssh với process id = 1050 đang kết nối đến pts/1 pseudo-terminal. Tóm lại, các tiến trình con sshd lành tính được liên kết với các sshd pseudo-terminals và một phiêm shell (bash, sh,…) thường được tạo dưới dạng một tiến trình con của pseudo-terminals để tương tác với người dùng và quản lý phiên

A close-up of a computer code

Description automatically generated

Hình . Các tiến trình con lành tính

* Các tiến trình con sshd độc hại: khai thác CVE-2024-3094 thông qua SSH với PoC ) [amlweems/xzbot: notes, honeypot, and exploit demo for the xz backdoor (CVE-2024-3094) (github.com)](https://github.com/amlweems/xzbot). Hình dưới là ví dụ về cây tiến trình khi khai thác lỗ hổng để thực thi cmd ‘sleep 60’. Thấy rằng: sleep 120 là một tiến trình con của sshd (process id = 941) và không được liên kết với pseudo-terminal cũng như phiên shell như đã thấy trong các quy trình lành tính ở trên. Ngoài ra, các tác nhân đe dọa có thể chạy cmd thông qua trình thông dịch shell như sh hoặc bash.

A screenshot of a computer screen

Description automatically generated

* Để có thể thực hiện tấn công, khai thác lỗ hổng này, nhóm tạo một agent mới trong hệ thống đã có tên ‘fedora’ trên máy ảo Fedora40 (rpm amd64) IP: 192.168.99.99
* Máy ảo sử dụng: Router, Wazuh server, Fedora40 (agent ‘fedora’) và Kali (attacker)
* Trên agent ‘fedora’, thực hiện thêm vào file */var/ossec/etc/ossec.conf* (phần được bôi trắng)để:
* Thẻ <command>: định cấu hình hiển thị danh sách các quy trình con sshd không được liên kết với pseudo-terminal.
* Thẻ <frequency>: cấu hình agent Wazuh để thực thi các pre-defined định kỳ sau 60 giây.
* Thẻ <alias> thêm chuỗi sshd\_child\_processes vào log được tạo, hữu ích khi kết hợp các rules → hiệu quả hơn.

A black and white screen

Description automatically generated

File */var/ossec/etc/ossec.conf* trên agent

* Sau đó restart lại agent để cập nhật
* Trên wazuh server, thêm vào các rules sau trong file */var/ossec/etc/rules/local\_rules.xml* (phần được bôi trắng): rule này sẽ match với các event được tạo bởi module Wazuh Logcollector có chứa chuỗi ‘sshd\_child\_processes’

A screenshot of a computer

Description automatically generated

File */var/ossec/etc/rules/local\_rules.xml* trên wazuh server

* Sau đó restart lại wazuh-manager để cập nhật rules
* Quá trình khai thác lỗ hổng:
* Trong /root trên Fedora40 (victim, agent) tạo 1 script weaken\_ fedora.sh để tạo môi trường chứa lỗ hổng CVE-2024-3094 và khai thác được qua dịch vụ SSH

A screenshot of a computer

Description automatically generated

File weaken\_fedora.sh

* Chạy script dưới quyền root
* Thực hiện tấn công: Trên máy Kali (attacker) chạy câu lệnh *‘docker run -it --rm golang:latest /bin/bash -c "mkdir -p /xzbot && pushd /xzbot/ && git clone https://github.com/awwalquan/xzbot.git && ls -laF && pushd ./xzbot/ && go build -o /xzbot/tmp/; popd && /xzbot/tmp/xzbot -h && /xzbot/tmp/xzbot -addr 192.168.99.99:22 -cmd 'sleep 120'" ‘* như hình dưới, cụ thể:
* *docker run -it --rm golang:latest*: Tạo một container mới từ image golang:latest và khởi chạy một phiên bash tương tác trong container. –rm để xóa container sau khi chạy xong
* *mkdir -p /xzbot*: Tạo thư mục /xzbot trong container.
* pushd /xzbot/: Đẩy thư mục làm thư mục làm việc hiện tại trong container sang /xzbot.
* *git clone https://github.com/awwalquan/xzbot.git*: Sao chép mã nguồn từ kho lưu trữ GitHub tới thư mục /xzbot trong container.
* *ls -laF*: Liệt kê tất cả các tệp và thư mục trong /xzbot để kiểm tra xem mã nguồn đã được sao chép thành công hay không.
* *pushd ./xzbot/*: Đẩy thư mục làm thư mục làm việc hiện tại trong container sang ./xzbot.
* *go build -o /xzbot/tmp/*: Biên dịch mã nguồn Go trong thư mục hiện tại và tạo một tệp thực thi có tên là xzbot trong /xzbot/tmp.
* *popd*: Trở lại thư mục trước đó trong stack thư mục làm việc.
* */xzbot/tmp/xzbot -h*: Chạy tệp thực thi xzbot với tùy chọn -h để hiển thị trợ giúp.
* */xzbot/tmp/xzbot -addr 192.168.99.99:22 -cmd 'sleep 120'*: Chạy tệp thực thi xzbot với tùy chọn -addr để chỉ định địa chỉ và cổng của một endpoint (agent ‘fedora’ với địa chỉ và port tương ứng)

A black screen with orange and yellow dots

Description automatically generated

Thực hiện tấn công trên máy Kali

* Kết quả chạy tấn công:

A screen shot of a computer

Description automatically generated

Kết quả chạy backdoor khai thác trên lỗ hổng

* Tiến trình con sshd độc hại trên agent fedora:

A screen shot of a computer

Description automatically generated

Tiến trình con độc hại trên victim

* Kết quả Wazuh phát hiện được có tiến trình con độc hại trên wazuh dashboard: rule\_id = 100010

A screenshot of a computer

Description automatically generated

Các alert trên wazuh dashboard khi thực hiện tấn công đặt backdoor trên victim

A screenshot of a computer

Description automatically generated

Chi tiết rule ID 100010

# THỰC NGHIỆM VÀ ĐÁNH GIÁ

Chương này sẽ trình bày về các kịch bản thực nghiệm và đánh giá kết quả thực nghiệm

## Kịch bản thực nghiệm

Với hệ thống đã triển khai, nhóm đề xuất 9 kịch bản thực nghiệm khác nhau nhằm đánh giá khả năng hoạt động của Wazuh như sau:

### Kịch bản 1: Tấn công Brute Force vào máy chủ SSH trên Web Server

Kịch bản được mô tả trong tài liệu Wazuh về việc chặn các cuộc tấn công brute force SSH sử dụng module active response(phản ứng chủ động), rất quan trọng đối với hệ thống của chúng ta. Attacker sẽ nhắm mục tiêu vào máy Victim chạy Ubuntu 22.04

Việc triển khai tính năng active response của Wazuh giúp nó phản ứng nhanh hơn. Khi phát hiện một loạt các lần đăng nhập SSH thất bại, nó có thể dùng active response để tạm thời chặn địa chỉ IP của Attacker. Điều này rất quan trọng vì:

* + Phòng thủ tự động: Nó cung cấp một cơ chế phòng thủ tự động ngay lập tức mà không cần sự can thiệp thủ công, giảm thời gian hệ thống bị xâm nhập, đánh cắp thông tin.
  + Bảo vệ tài nguyên: Bằng cách chặn các IP độc hại, nó ngăn chặn kẻ tấn công tiêu thụ tài nguyên mạng và có thể làm quá tải hệ thống của bạn.
  + Phân tích log: Khả năng giám sát liên tục và phân tích log của Wazuh giúp nhận diện các mẫu tấn công, cho phép chuẩn bị những chiến lược phản ứng tốt hơn trong tương lai.

Tóm lại, tích hợp tính năng phản active response của Wazuh để chặn các cuộc tấn công brute-force SSH sẽ giúp bảo vệ hệ thống khỏi các nỗ lực truy cập trái phép, đảm bảo một môi trường mạng an toàn hơn.

### Kịch bản 2: Phát hiện kết nối mạng dị thường

Đây như một hệ thống giám sát mạng tự động để phát hiện và đáp ứng với các kết nối mạng dị thường. Trong trường hợp này, nó sử dụng một danh sách kiểm tra (CBD list) để xác định các hoạt động mạng không mong muốn và chặn cổng (port) tương ứng.

Khi hệ thống phát hiện một kết nối mạng không bình thường dựa trên CBD list, nó sẽ tự động áp dụng các biện pháp bảo mật như chặn cổng (port) tương ứng. Chặn cổng này có thể là một biện pháp bảo vệ đơn giản để ngăn chặn các loại truy cập không mong muốn vào hệ thống, ngăn chặn tiến trình tấn công hoặc lây lan của malware.

### Kịch bản 3: Phát hiện các thay đổi trái phép trên hệ thống tập tin

Kịch bản này giúp kiểm tra và theo dõi các tệp tin nhạy cảm cũng như đáp ứng các yêu cầu tuân thủ quy định. Wazuh có sẵn một mô-đun FIM tích hợp, giúp theo dõi các thay đổi trong hệ thống tệp để phát hiện việc tạo, sửa đổi và xóa tệp.

Cụ thể, kịch bản này có thể sử dụng các công cụ giám sát hệ thống tập tin như để theo dõi sự thay đổi trong các tập tin, thư mục và cấu trúc hệ thống tập tin. Khi có sự thay đổi, hệ thống sẽ tự động so sánh với các tiêu chuẩn đã được xác định trước để xác định xem có sự can thiệp trái phép nào hay không.

### Kịch bản 4: Phát hiện lỗ hổng SQL Injection trên Web Server DVWA

Mục đích chính của kịch bản này là tìm kiếm, phát hiện và cảnh báo về các lỗ hổng SQL Injection trên Web Server DVWA để cải thiện mức độ bảo mật và an toàn của hệ thống web. Ta sẽ phát hiện và xác định các điểm yếu trong mã nguồn của ứng dụng web DVWA liên quan đến lỗ hổng SQL Injection.

Thông qua việc phát hiện và cảnh báo về các lỗ hổng SQL Injection, kịch bản này cung cấp một cơ chế bảo mật quan trọng để ngăn chặn các cuộc tấn công từ attacker, giúp các bảo vệ các dữ liệu quan trọng.

### Kịch bản 5: Phát hiện tấn công Shellshock trên Web Server DVWA (CVE-2014-6271)

Kịch bản phát hiện tấn công Shellshock trên Web Server DVWA (CVE-2014-6271) nhằm mục đích chủ yếu là bảo vệ hệ thống khỏi một trong những lỗ hổng nghiêm trọng nhất từng được phát hiện trong hệ điều hành Unix/Linux. Shellshock, còn được gọi là Bashdoor, là một lỗ hổng trong GNU Bash cho phép kẻ tấn công thực thi các lệnh tùy ý thông qua biến môi trường. Phát hiện này vào năm 2014 đã gây chấn động cộng đồng bảo mật do tính nghiêm trọng và khả năng khai thác rộng rãi của nó.

Kịch bản phát hiện này đóng vai trò then chốt trong việc bảo vệ hệ thống khỏi các mối đe dọa bảo mật nghiêm trọng, đảm bảo tính toàn vẹn và an toàn của dữ liệu.

### Kịch bản 6: Phát hiện rootkit Diamorphine trên hệ thống

Mục đích của kịch bản này là xác định và phát hiện sự tồn tại của rootkit Diamorphine trên hệ thống. Rootkit Diamorphine là một loại rootkit mã nguồn mở phổ biến trên hệ điều hành Linux, có khả năng che giấu các hoạt động của kẻ tấn công và duy trì quyền truy cập không bị phát hiện. Ta cần đảm bảo rằng bất kỳ dấu hiệu nào của rootkit Diamorphine đều được phát hiện và cảnh báo kịp thời, cho phép quản trị viên hệ thống hành động nhanh chóng để xử lý mối đe dọa.

### Kịch bản 7: Wazuh phát hiện và loại bỏ phần mềm độc hại bằng cách tích hợp VirusTotal

VirusTotal là một dịch vụ quét virus trực tuyến với cơ sở dữ liệu phong phú và khả năng phân tích mạnh mẽ. Kết hợp với hệ thống giám sát và phản ứng như Wazuh, việc tích hợp VirusTotal giúp cung cấp một lớp bảo vệ bổ sung cho hệ thống mạng, giúp phát hiện và ngăn chặn các mối đe dọa một cách hiệu quả hơn.

Mục đích của kịch bản này là tăng cường khả năng phát hiện và phản ứng trước các mối đe dọa bảo mật. Thông qua việc sử dụng dữ liệu và khả năng phân tích của VirusTotal, kịch bản này nhằm cung cấp cho tổ chức một công cụ mạnh mẽ để phát hiện các phần mềm độc hại một cách chính xác và toàn diện. Đồng thời, việc tự động hoặc bán tự động loại bỏ các phần mềm độc hại đã được phát hiện giúp giảm thiểu sự can thiệp của con người và tăng cường khả năng phản ứng nhanh chóng trước các mối đe dọa.

### Kịch bản 8: Wazuh phát hiện tiến trình chưa được chứng thực

Mục đích chính của này là tăng cường khả năng phát hiện các tiến trình không mong muốn hoặc độc hại đang chạy trên hệ thống. Bằng cách giám sát và phân tích các tiến trình, kịch bản này giúp ngăn chặn các hoạt động không mong muốn, từ việc cài đặt phần mềm không an toàn đến các cuộc tấn công độc hại.

Một mục tiêu cụ thể của kịch bản là phát hiện và cảnh báo ngay khi có bất kỳ tiến trình nào chưa được chứng thực xuất hiện trên hệ thống. Điều này giúp ta có thể phản ứng kịp thời, kiểm tra và xử lý nguy cơ một cách hiệu quả. Bằng cách cảnh báo ngay khi có sự xuất hiện của các tiến trình không được chứng thực, ta sẽ tăng cường khả năng phản ứng và giảm thiểu thiệt hại từ các cuộc tấn công.

### Kịch bản 9: Wazuh phát hiện khai thác lỗ hổng XZ Utils (CVE- 2024-3094)

Trong môi trường mạng ngày nay, mối đe dọa từ lỗ hổng bảo mật không ngừng gia tăng, đặc biệt là khi các attacker không ngừng tìm cách khai thác các lỗ hổng mới để xâm nhập vào hệ thống. Mục đích chính của kịch bảnlà tăng cường khả năng phát hiện và ngăn chặn các cuộc tấn công sử dụng lỗ hổng này trên hệ thống.

Kịch bản này không chỉ giúp phát hiện sớm các hoạt động khai thác lỗ hổng XZ Utils, mà còn đảm bảo rằng hệ thống được bảo vệ an toàn và dữ liệu quan trọng không bị ảnh hưởng. Bằng cách phát hiện và cảnh báo ngay khi có các hoạt động không đáng tin cậy được phát hiện, kịch bản giúp ngăn chặn các cuộc tấn công trước khi chúng gây ra thiệt hại nghiêm trọng cho hệ thống và dữ liệu.

## Đánh giá

Qua quá trình xây dựng và thực hành các kịch bản ở trên, IDPS dựa trên Wazuh có khả năng phát hiện một số cuộc tân công phổ biến. Hệ thống cũng có khả năng phản hổi nhanh và xử lí tốt các tác nhân tấn công.

Bên cạnh những điểm mạnh, hệ thống của nhóm xây dựng vẫn còn nhiều hạn chế do bị giới hạn tài nguyên bộ nhớ. Cũng như giới hạn trong phạm vi môn học nên nhóm không thể trình bày hết các tính năng mà Wazuh cung cấp

Kết luận, hệ thống Wazuh do nhóm xây dựng tuy còn nhiều khuyết điểm nhưng vẫn hoàn thành tốt các yêu cầu cũng như các kịch bản tấn công đã được đề ra

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận:

Việc triển khai các kịch bản bảo mật bằng hệ thống Wazuh đã đem lại những kết quả đáng kể trong việc bảo vệ và tăng cường an ninh cho hệ thống mạng và ứng dụng web. Mỗi kịch bản đều có mục đích và mục tiêu cụ thể, hướng đến việc phát hiện sớm, cảnh báo kịp thời và ngăn chặn hiệu quả các mối đe dọa bảo mật. Kết quả đạt được là một hệ thống bảo mật mạnh mẽ hơn, với khả năng phản ứng nhanh chóng và hiệu quả trước các mối đe dọa. Các biện pháp bảo vệ này không chỉ bảo vệ tính toàn vẹn và an toàn của dữ liệu mà còn nâng cao nhận thức và kỹ năng của đội ngũ quản trị viên về an ninh mạng. Hơn nữa, hệ thống Wazuh mà nhóm xây dựng đã tuân thủ tốt hơn các tiêu chuẩn và quy định bảo mật quốc tế, tăng cường niềm tin của người dùng. Tóm lại, các kịch bản bảo mật với Wazuh đã đóng góp quan trọng vào việc bảo vệ và duy trì an ninh cho hệ thống, đảm bảo hoạt động ổn định và an toàn trong môi trường mạng ngày càng phức tạp.

## Hướng phát triển:

Hướng phát triển của đề tài này sẽ tập trung vào việc nâng cao hơn nữa khả năng bảo mật và phản ứng trước các mối đe dọa ngày càng tinh vi và phức tạp. Một hướng phát triển tiềm năng là tích hợp trí tuệ nhân tạo và học máy vào hệ thống Wazuh để tự động hóa quá trình phát hiện và phản ứng, giúp cải thiện độ chính xác và giảm thiểu thời gian phản hồi. Ngoài ra, việc mở rộng phạm vi giám sát để bao phủ cả các thiết bị IoT và các dịch vụ đám mây cũng sẽ là một bước quan trọng nhằm đảm bảo an ninh toàn diện trong môi trường công nghệ hiện đại ngày nay. Tăng cường khả năng phân tích dữ liệu bảo mật bằng cách tích hợp với các hệ thống thông tin an ninh (SIEM) khác cũng sẽ giúp tạo ra một cái nhìn tổng quan và chi tiết hơn về tình hình an ninh mạng. Bên cạnh đó, việc xây dựng và thử nghiệm các kịch bản bảo mật mới để đối phó với các lỗ hổng và kỹ thuật tấn công mới nhất sẽ đảm bảo hệ thống luôn được cập nhật và sẵn sàng trước các mối đe dọa.

# TÀI LIỆU THAM KHẢO

| STT | Tài liệu |
| --- | --- |
| 1 | Wazuh Documentation <https://documentation.wazuh.com/> |
| 2 | Wazuh Blog Threat hunting using inventory data collected by Wazuh  [https://wazuh.com/blog/detecting-threats-using-inventory-data](https://wazuh.com/blog/detecting-threats-using-inventory-data/) |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |