HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO THỰC TẬP TỐT NGHIỆP

ĐỀ TÀI: *Nghiên cứu, đánh giá và thực nghiệm hệ thống EDR-SIEM Wazuh*

Giảng viên hướng dẫn:	**Th.S Vũ Minh Mạnh**
Họ và tên sinh viên:	**Lê Văn Đức**
Mã sinh viên:	**B19DCAT045**
Lớp:	**D19CQAT01-B**
Đơn vị thực tập	**Trung tâm Internet Việt Nam - VNNIC**

Hà Nội, tháng 8 năm 2023

Lời nói đầu

Cuộc cách Công nghiệp lần thứ tư là một cụm từ phổ biến được nhắc tới hằng ngày trong cuộc sống với việc tiếp cận tập trung vào công nghệ kỹ thuật số - với vai trò trung tâm là các thiết bị điện tử, internet, … Các vấn đề về bảo mật từ đây cũng phát sinh do những kẻ khai thác các lỗ hổng của các hệ thống này để lấy thông tin, đánh cắp tài sàn, danh tính, lửa đảo, … Do đó, các vấn đề về bảo mật mạng, bảo mật hệ thống máy tính, các thiết bị điện tử lại càng được quan tâm hơn bao giờ hết.

Sau 7 tuần thời gian thực tập tại Trung tâm Internet Việt Nam với sự giúp đỡ tận tình của các anh chị trong công ty đặc biệt là anh Nguyễn Huy Bắc, người trực tiếp hướng dẫn, giao việc và giám sát các đầu việc của bọn em; thầy Vũ Minh Mạnh luôn nhiệt tình trả lời các câu hỏi và hướng dẫn bọn em về học phần thực tập này; sản phẩm cuối khoá thực tập là bài báo cáo thực tập tổng hợp lại các kiến thức đã đạt được trong thời gian thực tập. Bài tổng hợp kiến thức có thể còn nhiều sai sót cũng như cần được hoàn thiện thêm, em mong muốn nhận được sự góp ý cũng như hướng dẫn từ các anh, chị.

DANH MỤC HÌNH ẢNH

DANH MỤC CÁC BẢNG

MỤC LỤC

I. MỤC TIÊU THỰC HIỆN TRONG QUÁ TRÌNH THỰC TẬP

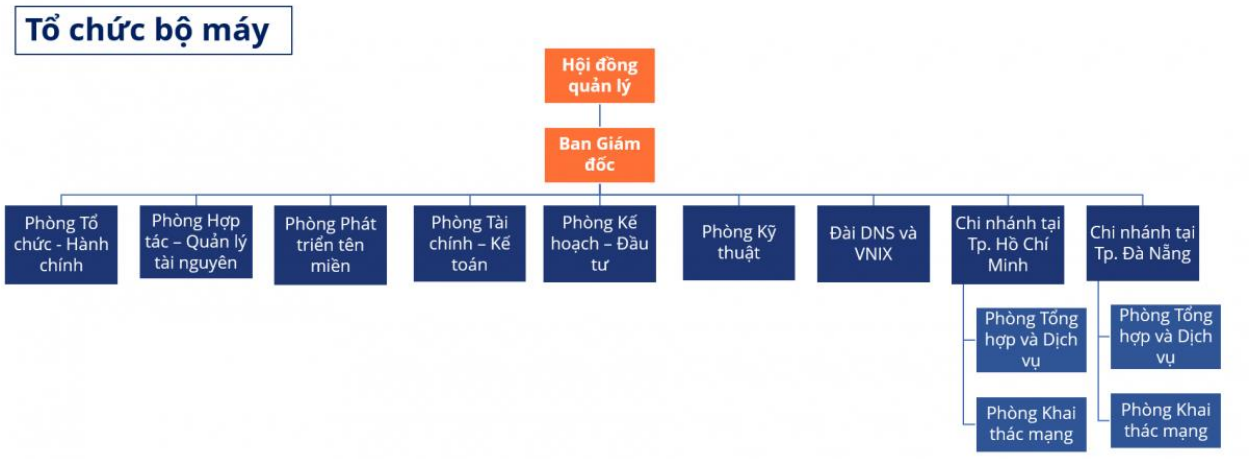
1.1. Mô hình tổ chức

VNNIC - Vietnam Internet Network Information Center hay Trung tâm Internet Việt Nam là đơn bị trực thuộc Bộ thông tin và Truyền thông (tương tự như PTIT) được thành lập ngày 28/04/2000 thực hiện các chức năng quản lý (bao gồm: đăng ký, cấp, phân bổ, thu hồi, ngừng, tạm ngừng) và thúc đẩy việc sử dụng nguồn tai nguyên Internet ở Việt Nam; thiết lập, quản lý và khai thác Hệ thống DNS quốc gia, Trạm trung chuyển Internet quốc gia.

VNNIC là đơn vị thuộc Bộ thông tin và Truyền thông nhưng có tư cách pháp nhân, con dấu và tài khoản riêng với hội đồng quản lý chịu trách nhiệm quản lý trực tiếp đơn vị; có ba trụ sở tại Hà Nội (trụ sở chính), Đà Nẵng và TPHCM.

Các vị trí quản lý đứng đầu gồm: Chủ tịch Hội đồng quản lý, Giám đốc, Phó giám đốc và các vị trí trưởng phòng. Phân thành Hội đồng quản lý, Ban Giám đốc, 7 phòng ban và 2 chi nhánh.

1.2. Chức năng các phòng ban liên quan



Hình 1. Tổ chức bộ máy VNNIC

Mỗi phòng ban trong VNNIC có một chức năng riêng, nhưng cũng có những phòng ban có quan hệ chặt chẽ với nhau như Đài DNS và VNIX với Phòng Kỹ thuật hay hai phòng ban trên với các phòng khai thác mạng, an toàn bảo mật tại chi nhánh Đà Nẵng và TP.HCM. Mối liên hệ này dựa trên yếu tố kỹ thuật, ngoài ra mọi phòng ban đều có liên quan và liên kết với nhau trong công việc.

Tại Đài DNS và VNIX – nơi thực hiện việc thực tập của bản thân, trưởng đài là anh Nguyễn Trường Giang – nhưng cũng đồng thời là phó Giám đốc trung tâm, do vậy người đứng đầu trong Đài là anh Bùi Quang Chiến – phó trưởng Đài.

Đài DNS và VNIX có chức năng chủ yếu liên quan tới tên miền .vn. .

1.3. Nội dung làm việc

1.3.1. Quy định

Đài DNS và VNIX có giờ vào làm việc từ 8:00 AM tới 5:00 PM đã bao gồm giờ nghỉ trưa.

Người hướng dẫn: anh Nguyễn Huy Bắc thuộc Đài DNS và VNIX – trưởng nhóm Hệ thống giám sát với hơn 17 năm công tác tại đơn vị.

1.4. Kỹ năng mềm

Đây là các kỹ năng được người hướng dẫn rèn luyện trong suốt quá trình thực tập, với thời gian 7 tuần thực tập, việc đạt được các mục tiêu về kỹ năng mềm với cá nhân em là quan trọng trong việc định hình được tư duy làm việc như một nhân viên chính thức.

1.4.1. Đặt mục tiêu

Trước khi được giao nhiệm vụ cần xác định trước mục tiêu cần đạt được nằm ở ngưỡng nào một cách cụ thể để công việc không bị lan man và đạt hiệu quả làm việc tốt nhất. Mục tiêu cuối cùng có thể thay đổi linh hoạt theo quá trình thực hiện nhưng cần hiểu rõ mục tiêu kỳ vọng của người phía trên giao việc, đồng thời đề xuất các mục tiêu dự định của bản thân có thể, muốn thực hiện.

1.4.2. Báo cáo công việc

Việc báo cáo được thực hiện theo nhiều mức độ từ cao tới thấp: báo cáo ngày, báo cáo tuần, báo cáo nhóm. Việc báo cáo giúp kiểm soát các mục tiêu đề ra trong phần trên để tránh lan man trong việc thực hiện công việc cũng như thay đổi linh hoạt mục tiêu sao cho phù hợp với thời gian, năng lực của bản thân. Báo cáo ngày được thực hiện thông qua văn bản.

Báo cáo tuần, tổng hợp lại những kiến thức của cả một tuần được thực hiện dưới sự hướng dẫn, giám sát của anh Nguyễn Huy Bắc bằng việc ngồi thảo luận hoặc tổ chức một cuộc họp. Do liên quan tới tổ chức cuộc họp, bản thân học thêm các kỹ năng của

việc tổ chức cuộc họp như chuẩn bị nội dung, slide, các thiết bị, nơi tổ chức, thuyết trình, hỏi và trả lời.

1.4.3. Làm việc nhóm

Thực tập tại VNNIC gồm năm thành viên được người hướng dẫn tổ chức thành một nhóm với việc tạo nên một hệ thống lab hoàn chỉnh. Như vậy, phần việc của mỗi thành viên đều liên quan tới nhau, mỗi tuần sẽ có các báo cáo nhóm giữa các thành viên để báo cáo tiến độ công việc với nhau.

Ngoài ra, do thời gian thực hiện các đầu việc của mỗi người là khác nhau nên cần xác định đầu ra và đầu vào của mỗi người, do đây là một quy trình hoàn thiện nên đầu ra của người này lại là đầu vào của người khác; qua đó các thành viên sẽ không làm ảnh hưởng tới tiến độ của nhau.

1.5. Mục tiêu chính về mặt chuyên môn

Về mặt chuyên môn, nhiệm vụ được giao là hoàn thành, thử nghiệm một hệ thống giám sát, quản lý thông tin sự kiện bảo mật – cụ thể là Wazuh với điểm cần bảo vệ, giám sát, quản lý là một máy chủ Centos 7 chạy DNS server sử dụng gói cài đặt Bind9.

Bộ công cụ Wazuh được cài được trên máy ảo sử dụng tệp .OVA được cung cấp bởi trang chủ, hệ điều hành Centos 7 minimal được tải trên trang chủ và phần mềm bin9 được cài đặt bằng yum.

Phạm vi của bài báo cáo không bao gồm các điểm cuối là các thiết bị mạng như router, firewall, … Các chức năng được sử dụng và thử nghiệm chủ yếu thiên về việc phát hiện và phản hồi, các ứng dụng tới quản lý dữ liệu có được sử và đề cập tuy không phải trọng tâm nhưng sẽ là phần được nghiên cứu bổ sung sau này.

II. TỔNG QUAN VỀ CÔNG NGHỆ

2.1. Các vấn đề liên quan tới an toàn, an ninh

2.1.1. Social Engineering – Kỹ thuật tấn công xã hội

Kỹ thuật tấn công xã hội là hình thức tấn công nhắm vào người dùng và thường liên qaun tới một số hình thức tương tác xã hội. Điểm yếu để kẻ tấn công khai thác là bản chất của các mối quan hệ xã hội như long đồng cảm, muốn giúp đỡ người khác hay bảo vệ người yếu thế. Kẻ tấn công không nhất thiết phải có kiến thức phức tạp về phần mềm hoặc phần cứng để thành công nên đây là hình thức lừa đảo phổ biến và dễ thực hiện nhất.

Có nhiều kỹ thuật con trong tấn công xã hội phổ biến như: phising, vishing, smishing, spam, spear phishing, dumpster diving, shoulder surfing , pharming, tailgating, …

Tất cả các kỹ thuật trên đều nhắm mục tiêu vào yếu tố con người, biện pháp khắc phục hiệu quả nhất là đào tạo và huấn luyện kiến thức bảo mật cho nhân viên, người dùng. Tuy vậy cũng có những biện pháp đối phó hữu ích như: sử dụng thẻ nhân viên, huy hiệu, giám sát người ra vào, sử dụng cửa xoay chỉ cho 1 người đi qua 1 thời điểm, sử dụng các ứng dụng chống spam, chặn các dns độc hại trước khi truy cập, …

2.1.2. Malware – Mã độc

Malicious software hay mã độc là một đoạn mã được tạo ra nhằm đánh cắp thông tin xác thực người dùng, ăn cắp dữ liệu, chiếm quyền kiểm soát hay phá hoại hệ thống máy tính, …

Có nhiều loại malware với các tên gọi, chức năng cũng như cách thức hoạt động khác nhau đã được biết tới như: virus, trojan, rootkit, worm, rasomeware, spyware, …

Việc phát hiện mã độc hiện nay chủ yếu được thực hiện bằng việc sử dụng các phần mềm diệt malware dựa trên chữ ký. Tuy nhiên, với việc các kỹ thuật khai thác máy tính liên tục thay đổi, những lỗ hổng zero-day được phát hiện và khai thác bởi các hacker mũ đen khi mã chưa có chữ ký thì việc phát hiện các hành vi bất thường trong hệ thống là cực kỳ quan trọng bên cạnh việc đào tạo ý thức cảnh giác của người dùng, nhân viên trước những mối nguy hại có khả năng tải mã độc về máy tính.

2.1.3. Unauthorized access – Truy cập trái phép

Truy cập trái phép là hành vi xảy ra khi một ai đó truy cập thành công hoặc cố gắng truy cập vào tài nguyên máy tính và nguồn dữ liệu mà không được phép của chủ sở

hữu. Truy cập trái phép có thể thực hiện ở mức vật lý như việc đột nhập vào văn phòng công ty để đánh cắp dữ liệu hay ở mức hệ thống thông tin là việc cố gắng đăng nhập vào máy tính, leo thang đặc quyền, …

2.1.4. System failure

Lỗi hệ thống là một vấn đề khi một máy tính hoặc một ứng dụng gặp lỗi. Điều này có thể được gây ra bởi các vấn đề về phần cứng như: thiết kế hệ thống, chọn lựa các thành phần xung đột, các phần cứng hoạt động với cường độ cao mà không được thiết kế để làm vậy.

Đây có thể là một vấn đề nghiêm trọng với các hệ thống cần tính sẵn dùng cao như các hệ thống máy chủ web, DNS, … Do vậy việc giám sát hoạt động của thiết bị phần cứng và các ứng dụng là cực kỳ quan trọng trong việc dự đoán, hành động trước khi việc lỗi xảy ra hoặc khắc phục kịp thời nếu lỗi đã xảy ra.

2.1.5. Quản lý tập trung, tìm kiếm, lưu trữ, trực quan hoá dữ liệu nhật ký

Với bốn vấn đề lớn trên, có thể sử dụng nhiều cách để cải thiện, khắc phục, một trong số này là việc giám sát dựa trên việc theo dõi dữ liệu nhật ký – log của mỗi thiết bị.

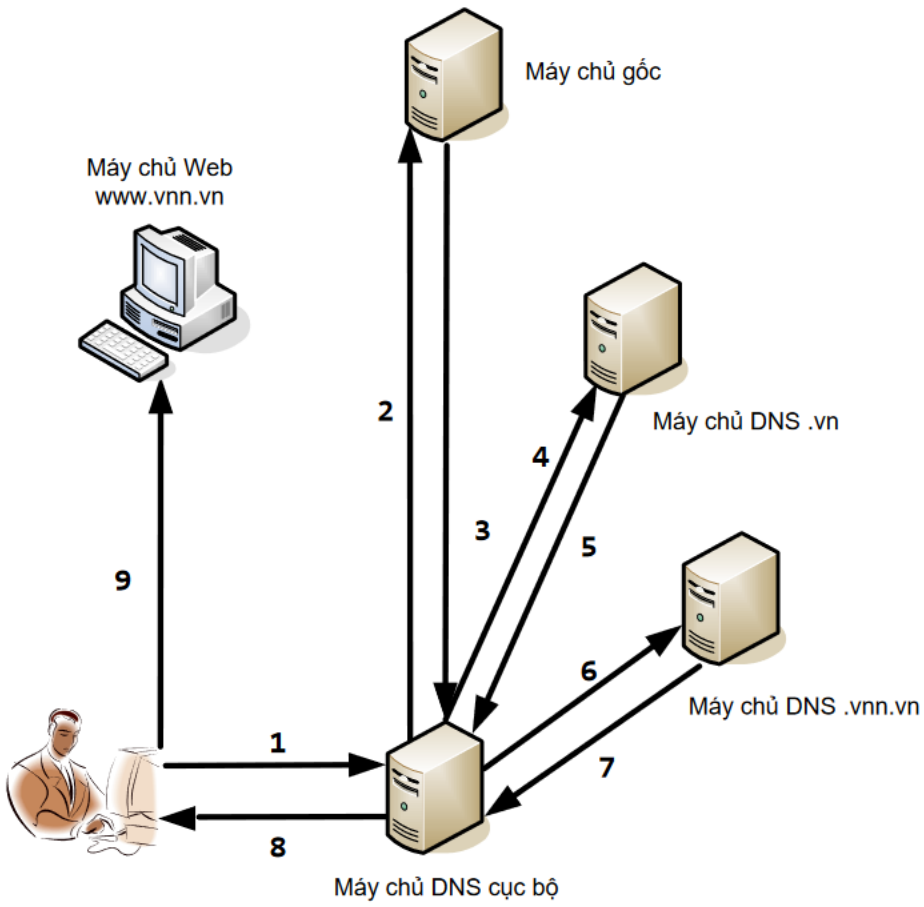
Tuy vậy, việc theo dõi, tìm kiếm và quản lý các dữ liệu này trên mỗi thiết bị thôi đã là không dễ dàng vì: mỗi hệ điều hành như Linux đã có nhiều loại log với các chức năng khác nhau như: messages, secure(auth.log), boot.log, kern.log, faillog, cron, yum.log, …; mỗi ứng dụng lại có một hay nhiều log như apache có access log, error log; … Việc đọc cái tệp log này cũng cần một phần mềm hoặc gói cài đặt khác.

Do đó có thể thấy việc quản lý, tìm kiếm, trực quan hoá và lưu trữ log từ nhiều nguồn khác nhau là rất quan trọng trong việc giám sát, phát hiện, phân tích các hành vi bất thường của hệ thống.

2.2. Hệ thống phân giải tên miền – DNS

DNS hay Domain Name System là hệ thống phân giải tên miền, có chức năng phân giải tên miền sang địa chỉ IP và ngược lại từ địa chỉ IP sang tên miền.

2.2.1. Một truy vấn và các thành phần trong một truy vấn DNS



Hình 2. Truy vấn trong DNS

Client: Là bên muốn phân giải tên miền hoặc đọc thông tin về tên miền -> gửi truy vấn tới tới resolver để tìm kiếm trong local host hoặc truy vấn tiếp.

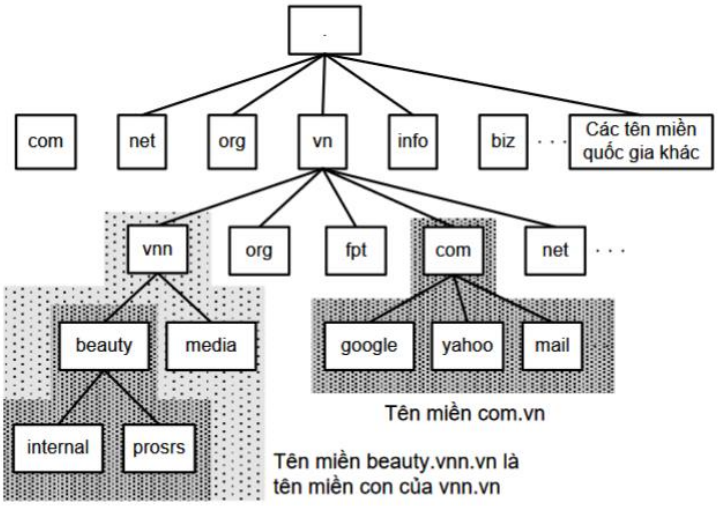
Máy chủ tên miền (name server): Máy chủ tên miền hoạt động trên cơ chế client-server. Máy chủ tên miền lưu giữ các thông tin về một phần của toàn bộ dữ liệu tên miền và có nhiệm vụ trả lời các máy tính (resolver) về các thông tin liên quan đến tên miền.

Resolver: là chư ơ n g trình client dùng để truy cập đến máy chủ. Các chương trình chạy trên máy tính muốn tìm kiếm thông tin về tên miền đều dùng chương trình Resolver để nối đến máy chủ tên miền và lấy thông tin. Nhiệm vụ của Resolver là: truy vấn dns, nhận câu trả lời từ dns server và trả về thông tin cho client.

Có 2 kiểu truy vấn trong DNS là truy vấn đệ quy và truy vấn không đệ quy. Mô tả ngắn gọn như sau. Truy vấn đệ quy: Resolver gửi yêu cầu truy vấn tới một máy chủ tên miền xác định để tìm thông tin về tên miền. Máy chủ này tìm kiếm thông tin và gửi trả lại cho resolver thông tin tên miền hoặc thông báo không tìm thấy thông tin. Lúc náy

resolver chỉ có thể tìm thông tin của tên miền bằng cách truy vấn tời một máy chủ tên miền khác. Truy vấn không đệ quy: được mô tả rõ ràng như hình minh hoạ phía trên.

2.2.2. Không gian tên miền



Hình 3. Cấu trúc không gian tên miền

Không gian tên miền có cấu trúc cây tạo nên tính phân tán một cách tự nhiên với tên miền root viết tắt là "." ở vị trí đứng đầu. Tên miền root hay máy chủ root bao gồm 13 máy chủ trên toàn thế giới sử dụng phương thức truyền tải anycast chạy trên layer 3 mô hình OSI.

Máy chủ root sẽ phân quyền – chuyển giao zone cho các máy chủ đứng hàng sau nó, còn được gọi là Top-level Domain (TLD). Các TLD này bao gồm các tên miền phổ biến như .com, .net, … và các Country-code TLD (ccTLD) hay các tên miền cấp quốc gia với tối đa 2 ký tự theo tiêu chuẩn ISO 3166; ví dụ như tên miền .vn, .kr, .jp, … Mỗi node trong cây tên miền này gọi là một nhãn. Các miền TLD lại chuyển giao cho các tên miền cấp thấp hơn nó với tối đa 127 mức.

Mỗi node như đã nói, được gán một label nhất định, mỗi label này có quy tắc đặt tên như sau: 63 ký tự, A-Z, a-z, 0-9 và ký tự "-". Mỗi một nhãn được phân tách bằng một dấu chấm và được viết tuần tự từ trái qua phải với thứ tự các node tăng dần; ví dụ như: ptit.edu.vn. là một tên miền hoàn chỉnh.

Zone là nới lưu trữ thông tin về tên miền và dữ liệu tương ứng với tên miền. Máy chủ tên miền nếu quản lý thông tin về tên miền trong một zone thì coi là có chủ quyền (authority), có thể kiểm tra bằng câu lệnh nslookup; ngược lại nếu thông tin về tên miền được kéo từ một zone khác về thì là non-authority. Mô hình DNS có thể hoạt động theo

kiểu chính và thức cấp hay còn được gọi là Master-Slave với máy chủ master là máy chủ lấy dữ liệu cho các zone của nó từ các file có sẵn trên máy, còn máy salve sẽ lấy dữ liệu từ máy chủ master. Việc máy chủ master này có thể là slave của master khác và ngược lại là hoàn toàn bình thường và có thể cấu hình được. Qua đó, đảm bảo tính sẵn dùng cao cho hệ thống.

Các bản ghi trong DNS đóng vai trò quan trọng như dữ liệu cần được khai báo với các bản ghi như SOA, NS, A, CNAME, MS, PTR, TXT, AAAA, …

2.2.3. Cài đặt DNS Server

Việc cài đặt DNS Server được thực hiện trên máy ảo VMWare, Centos 7 với gói cài đặt Bind9. Việc thực hiện cài đặt được sử dụng câu lệnh cài đặt yum. Có hai zone được khai báo là zone thuận và zone nghịch.

```
BIND 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.13 (Extended Support Version) <
[ducthiago@localhost ~]$
```

```
[root@localhost etc]# named-checkzone vanduc.com /var/named/db.v
anduc.com
zone vanduc.com/IN: loaded serial 0
OK
```

```
[root@localhost etc]# named-checkzone 206.168.192.in-addr.arpa /
var/named/db.206.168.192
zone 206.168.192.in-addr.arpa/IN: loaded serial 2
OK
```

```
[root@localhost etc]# dig www.vanduc.local

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.13 <<>> www.vandu
c.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is
leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31437
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONA
L: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.vanduc.local.              IN      A

;; ANSWER SECTION:
www.vanduc.local.       10800   IN      A       192.168.206.129

;; AUTHORITY SECTION:
vanduc.local.           10800   IN      NS      primary.vanduc.l
ocal.

;; ADDITIONAL SECTION:
primary.vanduc.local.   10800   IN      A       192.168.206.129
```

Hình 4. Cài đặt DNS Server

2.3. Giới thiệu chung về hệ thống giám sát giám sát mạng – IDS

2.3.1. Khái niệm

Hệ thống phát hiện xâm nhập (IDS) và Hệ thống ngăn chặn xâm nhập (IPS) lần đầu tiên được giới thiệu vào năm 1986 dưới dạng một bài báo học thuật được viết bởi Dorothy E. Denning; bài báo có tiêu đề "Mô hình phát hiện xâm nhập".

Từ 2000 đến 2005; phát hiện xâm nhập (IDS) được ưu tiên hơn ngăn chặn xâm nhập (IPS). Điều này là do vào đầu những năm 2000, các mối đe dọa mới như SQL injection và Cross site scripting attack (XSS) đã trở nên phổ biến và những cuộc tấn công này đã vượt qua được tường lửa. Tường lửa rất hiệu quả để ngăn chặn mối đe dọa trong những năm 1990, nhưng với các mối đe dọa mới, IDS trở thành phương pháp an toàn hơn cả.

Vào thời điểm đó, thị trường IPS rất thấp vì hầu hết các tổ chức đều lo lắng rằng IPS có thể ngăn chặn lưu lượng truy cập bất thường vô hại từ các khách hàng tiềm năng. Lý do chính đằng sau việc IDS được ưa thích hơn IPS là vì IDS sẽ phát cảnh báo hoặc cảnh báo cho tổ chức khi phát hiện ra hoạt động nguy hiểm, sau đó tổ chức, cá nhân sẽ thực hiện các biện pháp phù hợp để loại bỏ mối đe dọa.

Việc áp dụng IPS đã không bắt đầu phát triển cho đến cuối năm 2005 khi nhiều người bắt đầu ủng hộ nó. Vào thời điểm đó, chữ ký được viết để phát hiện các khai thác và lỗ hổng. Ý tưởng là đối với mỗi lỗ hổng, có thể có hàng trăm cách để khai thác nó.

Khi tội phạm phát hiện ra một lỗ hổng, chúng có thể tạo ra hơn 3 trăm cách khác nhau để khai thác lỗ hổng đó, khiến các nhà phát triển IDS phải viết hàng trăm chữ ký khai thác khác nhau trở lên. Vì vậy, IPS phù hợp hơn do người dùng lo ngại rằng tất cả các chữ ký sẽ làm chậm mạng vì mỗi kết nối sẽ phải được kiểm tra. Vì lo ngại đó, các nhà phát triển IPS bắt đầu chỉ tạo một chữ ký có thể hoạt động với từng lỗ hổng bất kể có bao nhiêu lần khai thác được kết nối với nó. Nhà phát triển nhận thấy rằng IPS hoặc IDS có hơn 3500 chữ ký có khả năng cản trở hiệu năng của hệ thống. Cho đến ngày nay, cả IDS và IPS vẫn tiếp tục thay đổi và phát triển khi những kẻ tấn công thay đổi các kỹ thuật được sử dụng để xâm nhập vào hệ thống mạng.

Thuật ngữ xâm nhập đề cập đến việc làm gián đoạn ai đó mà không được phép. Trong máy tính, xâm nhập đề cập đến nỗ lực truy cập tài nguyên hệ thống máy tính mà không có bất kỳ sự cho phép nào với ý định gây ra thiệt hại nào đó.

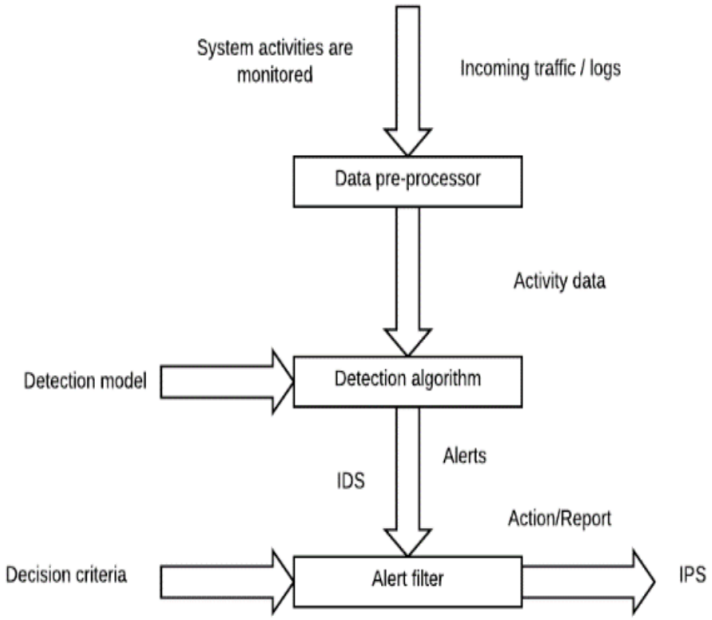
Về cơ bản, Phát hiện xâm nhập đề cập đến bất kỳ loại cơ chế phát hiện hành vi xâm nhập nào đó và hệ thống phát hiện xâm nhập (IDS) đề cập đến một hệ thống thực hiện quá trình phát hiện xâm nhập một cách tự động.

IDS chịu trách nhiệm giám sát lưu lượng dữ liệu trong mạng và bất kỳ hoạt động đáng ngờ nào đối với an ninh mạng. Quản trị viên hệ thống hoặc quản trị viên mạng được cảnh báo hoặc report nếu phát hiện bất kỳ mối đe dọa hoặc hoạt động nguy hại nào trong mạng. Do đó, mục đích chính của IDS là phát hiện và báo cáo các nỗ lực xâm nhập cho các bên liên quan.

IDS sử dụng các loại công cụ và kỹ thuật khác nhau để phát hiện các hoạt động đáng ngờ ở cả cấp độ máy chủ và cấp độ mạng. IDS có thể được chia thành hai loại chính như Hệ thống phát hiện xâm nhập dựa trên máy chủ (HIDS) và Hệ thống phát hiện xâm nhập dựa trên mạng (NIDS).

2.3.2. Các thành phần chính

Các thành phần phổ biến của một hệ thống IDS:



Hình 5.1 Các thành phần của hệ thống IDS

- Data pre-processor: bộ tiền xử lý dữ liệu chịu trách nhiệm thu thập và định dạng dữ liệu sẽ được phân tích bằng thuật toán phát hiện xâm nhập.
- Detection algorithm: phát hiện sự khác biệt giữa lưu lượng truy cập mạng "bình thường" hoặc "hợp pháp" và xâm nhập trên cơ sở mô hình phát hiện.
- Alert filter: ước tính mức độ nghiêm trọng của sự xâm nhập dựa trên các tiêu chí quyết định và các hoạt động độc hại được phát hiện. Sau đó, bộ lọc cảnh báo sẽ cảnh báo cho quản trị viên mạng hoặc hệ thống và thực hiện các hành động phản hồi

2.3.3. Chức năng

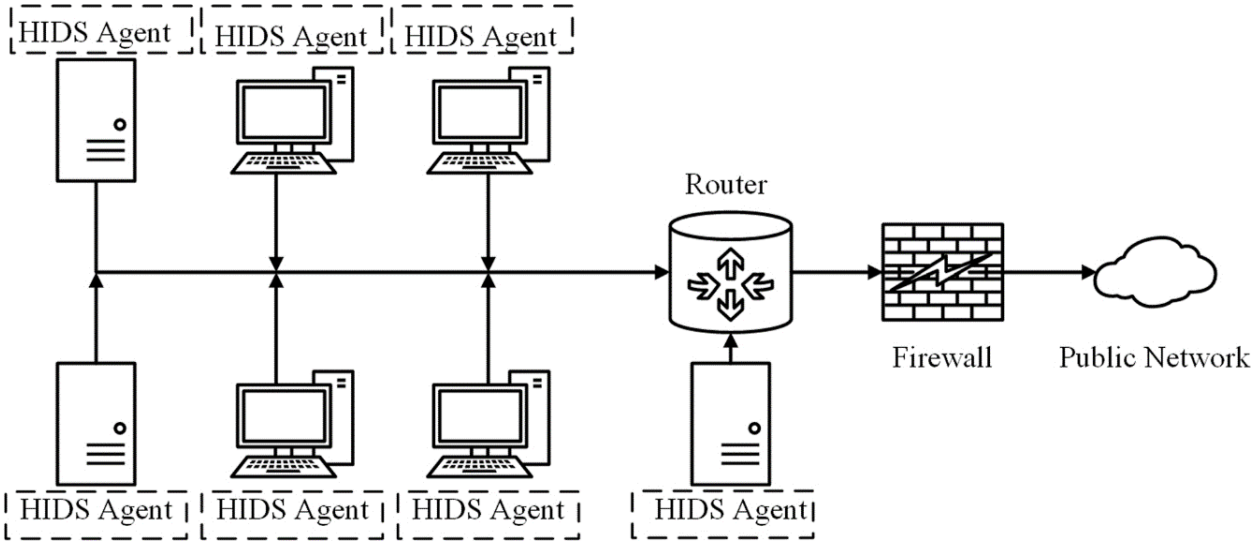
- Data collection: từ module này, IDS nhận dữ liệu đầu vào, lưu vào file và phân tích chúng. IDS dựa trên mạng thu thập và thay đổi các gói dữ liệu và IDS dựa trên máy chủ thu thập các chi tiết như việc sử dụng đĩa và các quy trình của hệ thống.
- Feature Selection: để chọn một tính năng cụ thể, dữ liệu lớn đã có sẵn trong mạng và chúng thường được đánh giá để xâm nhập.
- Analysis: Các dữ liệu được phân tích để tìm ra sự đúng đắn. IDS dựa trên quy tắc phân tích dữ liệu nơi dữ liệu đến lưu lượng được kiểm tra đối với chữ ký hoặc mẫu được xác định trước. Một phương pháp khác là IDS dựa trên sự bất thường trong đó hành vi của hệ thống được nghiên cứu và các mô hình toán học được sử dụng cho nó.
- Action: xác định về cuộc tấn công và phản ứng của hệ thống. Có thể thông báo cho quản trị viên hệ thống tất cả dữ liệu cần thiết thông qua các email/alert hoặc

có thể đóng vai trò tích cực trong hệ thống bằng cách loại bỏ các gói để nó không xâm nhập vào hệ thống hoặc đóng các cổng:

- Thông tin tài liệu liên quan đến hoạt động được quan sát. Thông tin có thể được ghi cục bộ hoặc gửi đến máy chủ quản lý nhật ký tập trung (SIEM).
- Cung cấp thông báo về hoạt động được quan sát cho quản trị viên. Thông báo này được gọi là Alert, có thể ở dạng email, tin nhắn, pages, tập lệnh, v.v. Thông báo bao gồm các chi tiết nhỏ về các sự kiện được bắt được. Thông tin bổ sung về event, yêu cầu quản trị viên truy cập IDS
- Tạo báo cáo. Tất cả các hoạt động hoặc sự kiện được giám sát được tóm tắt trong báo cáo

2.3.4. Phân loại

- Host-based IDS

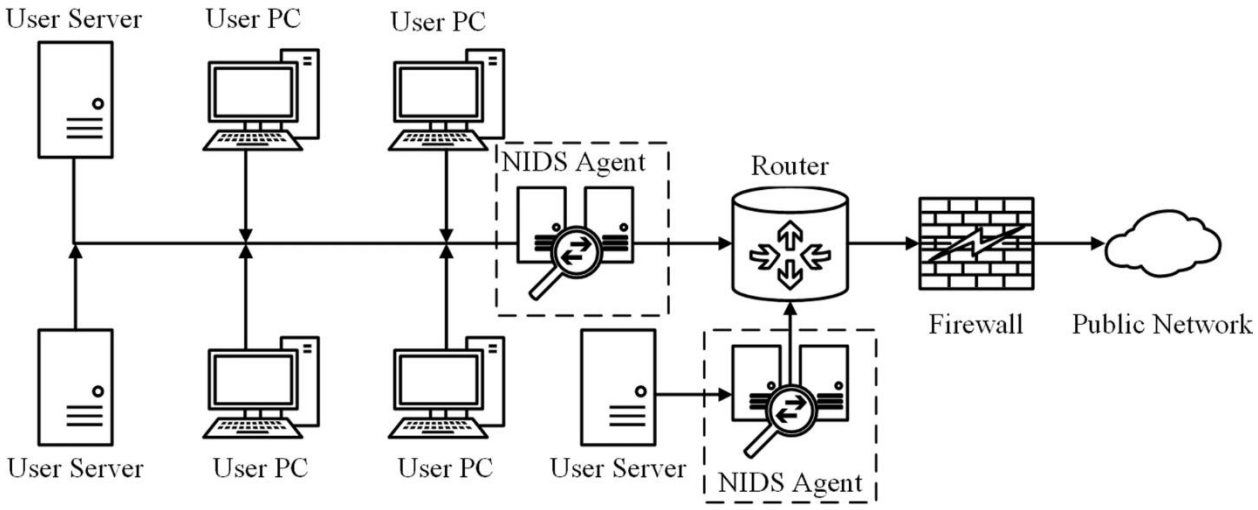


Hình 6. Hệ thống Host-based IDS

Hệ thống phát hiện xâm nhập dựa trên máy chủ (HIDS) là một loại IDS được cài đặt trên các máy tính hoặc thiết bị riêng lẻ trong mạng để phát hiện các hoạt động độc hại có thể xảy ra trên các thiết bị đó. Chức năng chính của HIDS là giám sát hoạt động trên thiết bị chủ và so sánh nó với một bộ quy tắc định trước để xác định mọi hoạt động đáng ngờ hoặc các mối đe dọa tiềm ẩn.

HIDS đặc biệt hữu ích để phát hiện các mối đe dọa bắt nguồn từ bên trong mạng, chẳng hạn như phần mềm độc hại, trojan hoặc các mối đe dọa nội bộ. Chúng cũng có thể hiệu quả trong việc phát hiện các mối đe dọa bắt nguồn từ bên ngoài mạng, chẳng hạn như tin tặc đang cố giành quyền truy cập vào thiết bị thông qua một lỗ hổng.

- Network-based IDS



Hình 7. Hệ thống Network-based IDS

Hệ thống phát hiện xâm nhập dựa trên mạng (NIDS) là một loại IDS được cài đặt trên chính mạng để giám sát lưu lượng và phát hiện mọi hoạt động đáng ngờ. NIDS được thiết kế để phát hiện nhiều mối đe dọa, bao gồm các cuộc tấn công dựa trên mạng như tấn công từ chối dịch vụ (DoS), quét cổng và phân phối phần mềm độc hại.

NIDS có thể giám sát nhiều thiết bị cùng một lúc, làm cho nó trở thành một cách hiệu quả để phát hiện các cuộc tấn công xảy ra trên toàn bộ mạng. Tuy nhiên, nó có thể không hiệu quả trong việc phát hiện các mối đe dọa xảy ra trên các thiết bị riêng lẻ trong mạng.

2.4. Giới thiệu chung về EDR, XDR, SIEM

2.4.1. Endpoint detection and response – EDR và Extended detection and response – XDR

- Endpoint detection and response – EDR: là một giải pháp được thiết kế để phát hiện và điều tra các mối đe dọa mạng trên các thiết bị điểm cuối như PC, máy tính xách tay hoặc máy chủ. Trái ngược với phần mềm chống vi-rút, EDR không chỉ phát hiện các mối đe dọa trên mạng bằng cách quét các tệp để tìm chữ ký vi-rút mà còn bằng cách xem xét hành vi của các thiết bị đầu cuối. Khi phát hiện hành vi đáng ngờ, công cụ này sẽ thông báo cho nhóm bảo mật CNTT và đề xuất các hành động khắc phục. Công cụ EDR cũng có thể cung cấp các phản hồi giảm thiểu tự động, chẳng hạn như cách ly điểm cuối.
- Extended detection and response (XDR) là sự phát triển của giải pháp EDR. XDR mở rộng phạm vi phát hiện ra ngoài các điểm cuối – cung cấp khả năng phát hiện,

phân tích và phản hồi trên nhiều nguồn dữ liệu. XDR thu thập dữ liệu và phân tích hành vi của tất cả các lớp và ứng dụng CNTT. Ngoài các điểm cuối, điều này bao gồm các thành phần mạng và dịch vụ đám mây. Bằng cách này, XDR tạo ra một cái nhìn toàn diện về bảo mật CNTT và các mối đe dọa mạng có thể xảy ra, giúp đơn giản hóa các hoạt động điều tra và ứng phó.

2.4.2. Security information and event management - SIEM

Gartner vào năm 2005 đã giới thiệu SIEM (quản lý sự kiện và thông tin bảo mật) để giám sát endpoint. SIEM là sự kết hợp giữa SIM (quản lý thông tin bảo mật) và SEM (quản lý sự kiện bảo mật), là hai hệ thống riêng biệt được dùng để lưu trữ, phân tích và báo cáo sự kiện (SIM) và thu thập sự kiện theo thời gian thực (SEM). Nhìn chung, SIEM là một công cụ bảo mật hỗ trợ các công ty xác định các lỗ hổng và mối đe dọa bảo mật tiềm ẩn trước khi chúng có thể can thiệp và làm hỏng hoạt động kinh doanh.

SIEM đã trở thành trụ cột trong các trung tâm điều hành an ninh (SOC) hiện nay. Các công cụ quản lý sự kiện và thông tin bảo mật (SIEM) thường được sử dụng để bảo mật hệ thống mạng-vật lý. Chúng thu thập log/event từ các thiết bị và thực hiện nhiều hành động khác nhau để đảm bảo tính bảo mật của các hệ thống được kết nối. Các chức năng chính của SIEM là thu thập, tích hợp, phân tích, chuẩn hóa, tổng hợp, làm phong phú dữ liệu, lập chỉ mục (indexing) và trực quan hoá.

- Collection (thu thập)
  Agent Based:
  - Agent/Collector trên mỗi thiết bị thu thập, phân tích và chuyển tiếp logs.
  - Windows Servers, Web Servers, Other file-based logs Sysmon, NXLog, OSSEC, etc.

  Agentless:

  - Các thiết bị gửi logs đến máy chủ.
  - Windows Hosts WMI, Cloud Environments (APls), Firewalls, Switches.
- Tích hợp (Aggregation): Tích hợp là quá trình thu thập logs từ nhiều hệ thống, phân tích chúng và trích xuất dữ liệu có cấu trúc, và kết hợp ở định dạng có thể tìm kiếm. Các phương pháp gồm có:
  - Push: Logs được đẩy từ nguồn đến máy chủ.
  - Pull: Logs được kéo về bởi máy chủ từ nguồn.
- Parsing: Thành phần phần mềm có thể truy cập vào một định dạng logs cụ thể và chuyển đổi nó thành dữ liệu có cấu trúc. Nhiều parsers được sử dụng cho các hệ thống khác nhau. Ví dụ log:

Sep 28 16.39.03 app_server sshd[8677] Failed password for invalid user icecast2 from 10.72.109.227 port 57238 ssh2

Sau khi parse:

host = app_server

process = sshd

source_user = icecast2 source_lip = 10.72.109.227

source_port = 57238

- Chuẩn hóa (Normalization): Chuẩn hóa hợp nhất các sự kiện với dữ liệu khác nhau thành một định dạng giảm thiểu chứa các thuộc tính sự kiện chung. Tuân theo một tiêu chuẩn để giảm hồ sơ thành các thuộc tính sự kiện chung, tức là, các tên trường và giá trị chung.
- Phân loại (Categorization): Phân loại liên quan đến việc thêm ý nghĩa cho các sự kiện bằng cách xác định dữ liệu log liên quan đến các sự kiện hệ thống, xác thực, hoạt động cục bộ/từ xa, v.v.
- Làm phong phú (Enrichment): Làm phong phú log đề cập đến việc thêm thông tin quan trọng có thể làm cho dữ liệu hữu ích hơn.

host = app_server

process = sshd

source_user =icecast2 →(Administrator)

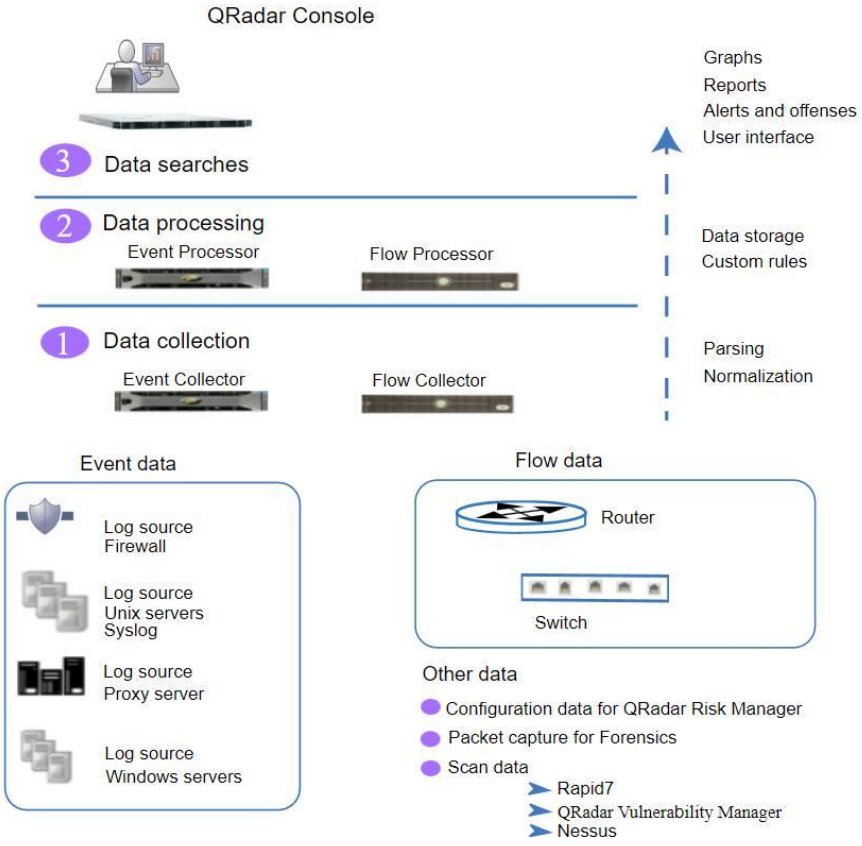
source_ip = 10.72.109.227 →(Internal IP)

source_port =57238

- Correlation Rules & Alerts: Quy tắc tương quan (Correlation Rules) là biểu thức logic khiến hệ thống thực hiện một hành động cụ thể nếu một sự kiện cụ thể xảy ra. Ví dụ:

"Đăng nhập ẩn danh, cảnh báo người dùng". Nói cách khác, quy tắc tương quan là một điều kiện (hoặc tập hợp các điều kiện) hoạt động như một trình kích hoạt cho cảnh báo.

- Lập chỉ mục (Indexing): Index là cơ sở dữ liệu của dữ liệu, được tổ chức theo cách giúp dễ dàng tìm thấy các phần dữ liệu cụ thể.



Hình 8. Mô hình khái quát hoạt động của hệ thống SIEM

III. Wazuh (phiên bản 4.4)

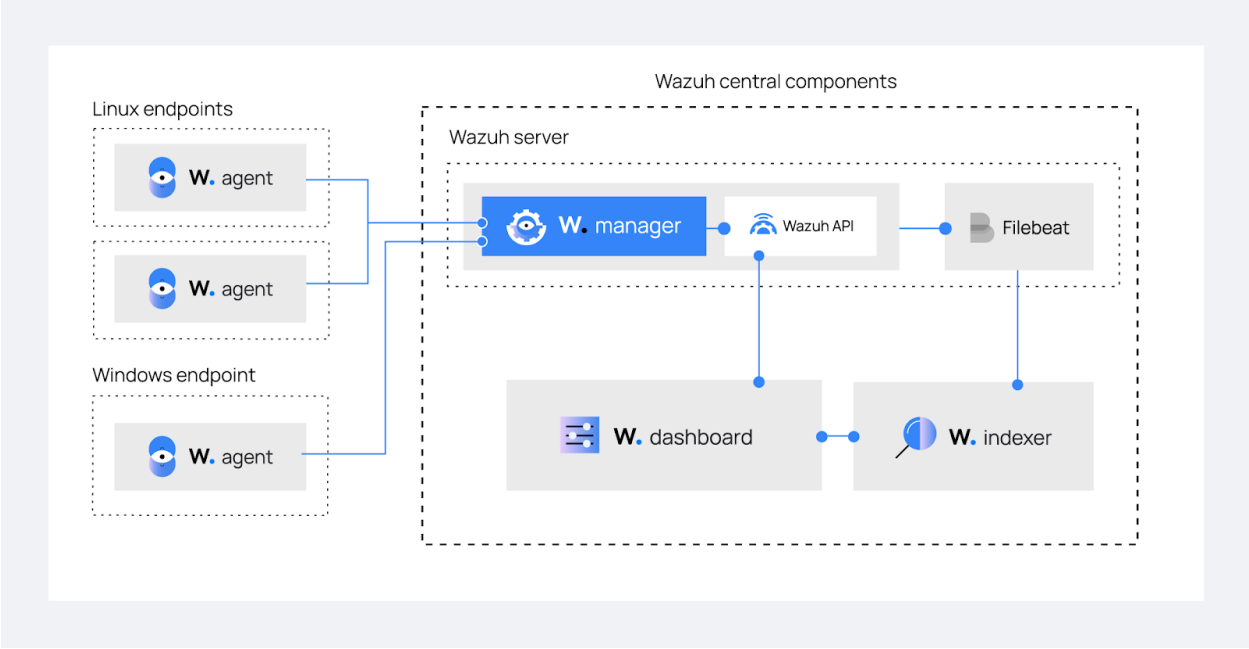
Wazuh là một bộ công cụ mã nguồn mở với các chức năng của một hệ thống SIEM kết hợp với EDR ra đời vào năm 2015. Wazuh bao gồm 3 thành phần chính cốt lõi là OSSEC HIDS – một HIDS mã nguồn mở miễn phí phổ biến nhất thế giới, đây cũng là thành phần tạo nên chức năng EDR cho bộ công cụ này.

Hai thành phần quan trọng còn lại là Wazuh Indexer và Wazuh Dashboard – đây là hai thành phần tiêu chuẩn được thêm vào từ phiên bản 4.3 nhằm thay thế cho bộ công cụ Elasticsearch/Opensearch, hai thành phần này tạo nên chức năng SIEM cho bộ công cụ này nhờ việc thu thập, quản lý, tìm kiếm và lưu trữ thông tin một cách hiệu quả và chủ động đa nguồn.

Wazuh có thể được cài đặt trên đa nền tảng như Linux, máy ảo, các máy chủ đám mây và các điểm cuối có thể được cài đặt trên hầu hết mọi hệ điều hành phổ biến hiện nay như Windows, MacOS, Linux, FreeBSD, … Với các thiết bị mạng không thể cài đặt, Wazuh có thể giám sát bằng việc sử dụng hệ thống Syslog trên các thiết bị này mặc định gửi thông tin về máy chủ thông qua cổng 514/UDP.

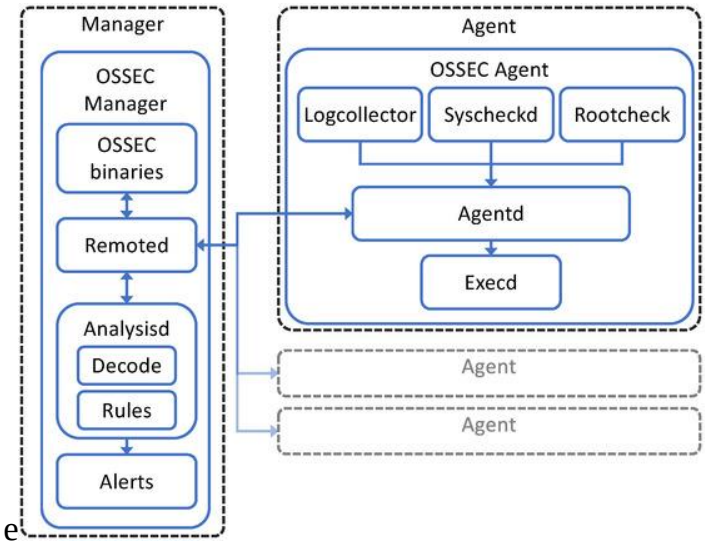
Wazuh với việc tích hợp các công cụ với nhau một cách hiệu quả đã trở thành một giải pháp, quy trình cực kỳ hoàn thiện đặc biệt với việc mã nguồn mở, hoàn toàn miễn phí đặc biệt phù hợp với các tổ chức, các nhân và với mục đích nghiên cứu trong bài đánh giá này. Dưới đây là chi tiết về bộ công cụ này.

3.1. Kiến trúc và luồng thông tin



Hình 9. Kiến trúc chung của Wazuh

Kiến trúc của Wazuh dựa trên kiến trúc của Ossec – một HIDS mã nguồn mở, theo mô hình Agent-Agentless – Server. Trong hình trên, các endpoint và W.server chính là Ossec, ngoài ra được kết hợp thêm các thành phần là Wazuh Dashboard và Wazuh Indexer được rẽ nhánh từ OpenSearch (phiên bản mã nguồn mở của Elasticsearch được phát triển bởi AWS).



Hình 10. Luồng dữ liệu Agent-Server

Luồng thông tin cơ bản của Wazuh sẽ đi từ trái sang phải với khởi đầu là việc giao tiếp giữa endpoint và server, do nguồn thông tin dữ liệu và mục tiêu cần bảo vệ cũng chính là cá endpoint.

Các thiết bị cuối này được cài đặt mềm phần Agent giúp chuyển tiếp cá thông tin tới server, nhận các lệnh từ server gửi xuống. Các thiết bị không thể cài Agent như tường lửa, router, switch và access point được hỗ trợ và có thể chủ động gửi dữ liệu nhật ký qua Syslog, SSH hoặc sử dụng API.

Máy chủ trung tâm mở cổng 1514 (có thể cấu hình) để nhận thông tin truyền tới thông qua giao thức Wazuh message được mã hoá bằng mã hoá AES-256bit (có thể sử dụng mã hoá Blowfish thay thế).

Sau đó, Wazuh server giải mã và kiểm tra luật các sự kiện nhận được xem có khớp không. Các sự kiện khớp luật được tiến hành các hành động như cảnh báo, …

Tiếp tới là việc giao tiếp của server và indexer. Filebeat đọc dữ liệu đầu ra của Wazuh server và chuyển tới indexer theo cổng 9200/TCP và đường truyền được đảm bảo bằng mã hoá TLS.

Dữ liệu sau khi được đánh index bởi indexer sẽ được trực quan hoá bởi Wazuh Dashboard. Dashboard có thể nhận thông tin từ server thông qua truy vấn RESTful API được lắng nghe trên cổng 5500/TCP mã hoá bằng TLS, Dashboard có thể hiển thị cấu hình và thông tin liên quan tới trạng thái của server và các Agent, sửa đổi cấu hình Agent và server. Dashboard sử dụng username và password để xác thực.

Ảnh sau mô tả các giao thức, các cổng và mục đích của chúng:

Thành phần	Cổng	Giao thức	Mục đích
Wazuh server	1514	TCP (default)	Dịch vụ kết nối Agent
	1514	UDP (optional)	Dịch vụ kết nối Agent
	1515	TCP	Dịch vụ đăng ký Agent
	1516	TCP	Dịch vụ đăng ký Agent
	514	UDP (default)	Thu thập syslog (tắt theo mặc định)
	514	TCP (optional)	Thu thập syslog (tắt theo mặc định)
	55000	TCP	Wazuh server RESTful API
Wazuh indexer	9200	TCP	Wazuh indexer RESTful API
	9300-9400	TCP	Giao tiếp giữa các cụm Wazuh indexer
Wazuh dashboard	443	TCP	Giao diện web

Hình 11. Các cổng và giao thức trong Wazuh

3.2. Wazuh Server

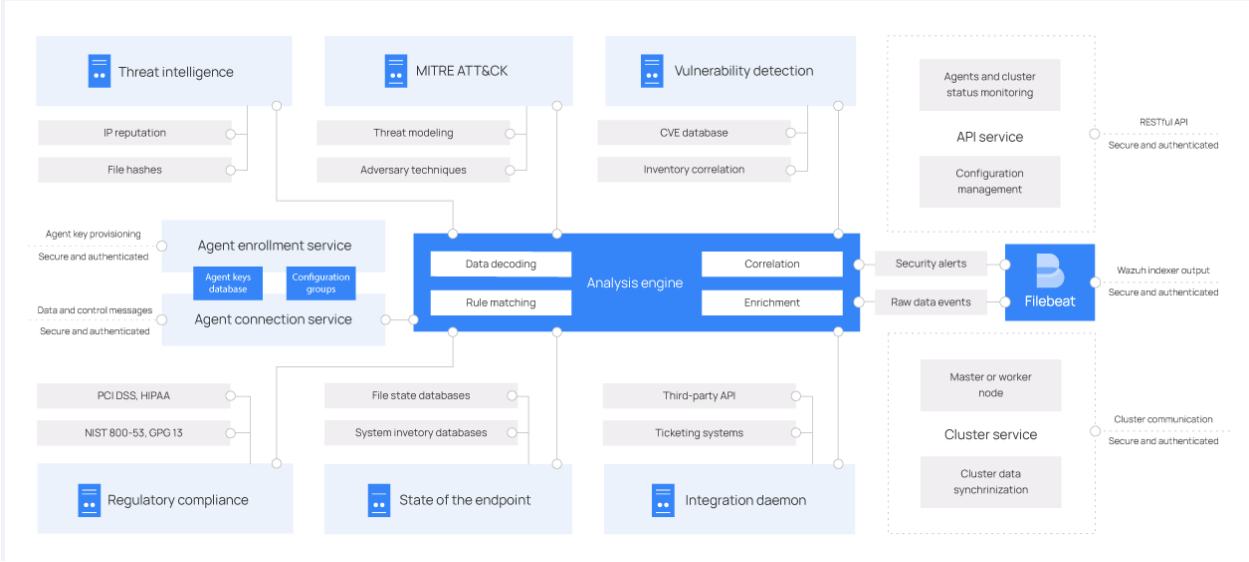
Wazuh Server phân tích dữ liệu nhận được từ các Agent, kích hoạt cảnh báo khi phát hiện thấy các mối đe dọa hoặc sự bất thường. Nó cũng được sử dụng để quản lý cấu hình Agent từ xa và theo dõi trạng thái của chúng.

Wazuh Server sử dụng các nguồn thông tin về mối đe dọa để cải thiện khả năng phát hiện. Đồng thời làm phong phú thêm dữ liệunbằng cách sử dụng các framework như: MITRE ATT&CK và các yêu cầu tuân thủ quy định như PCI DSS, GDPR, HIPAA, CIS và NIST 800-53..

Ngoài ra, Wazuh Server có thể được tích hợp với các phần mềm bên thứ 3.

3.2.1. Kiến trúc và luồng hoạt động

Wazuh chạy công cụ phân tích, API RESTful của Wazuh, dịch vụ đăng ký Agent, dịch vụ kết nối Agent, daemon cụm Wazuh và Filebeat. Máy chủ được cài đặt trên hệ điều hành Linux và thường chạy trên máy vật lý độc lập, máy ảo, bộ chứa docker hoặc phiên bản đám mây.



Hình 12. Kiến trúc và các thành phần của Wazuh Server

3.2.2. Các thành phần

- Agent enrollment service: được sử dụng để đăng ký Agent mới. Dịch vụ này cung cấp và phân phối các khóa xác thực duy nhất cho mỗi Agent. Quá trình này chạy dưới dạng dịch vụ mạng và hỗ trợ xác thực qua chứng chỉ TLS/SSL hoặc bằng cách cung cấp mật khẩu cố định.
- Agent connection service: Dịch vụ này nhận dữ liệu từ các Agent. Sử dụng các khóa được chia sẻ bởi Agent enrollment service để xác thực danh tính của từng Agent và mã hóa thông tin liên lạc giữa Agent và Wazuh Server. Ngoài ra, dịch vụ này cung cấp khả năng quản lý cấu hình tập trung, cho phép cài đặt Agent mới từ xa.
- Analysis engine: Đây là thành phần thực hiện phân tích dữ liệu. Nó sử dụng bộ giải mã để xác định loại thông tin đang được xử lý (sự kiện Windows, nhật ký SSH, nhật ký máy chủ web, v.v.). Các bộ giải mã này cũng trích xuất các thành phần dữ liệu có liên quan từ thông điệp bản ghi, chẳng hạn như địa chỉ IP nguồn, ID sự kiện hoặc tên người dùng. Sau đó, bằng cách sử dụng các quy tắc, công cụ xác định các mẫu cụ thể trong các sự kiện được giải mã có thể kích hoạt cảnh báo và thậm chí có thể yêu cầu các biện pháp phản hồi (ví dụ: cấm địa chỉ IP, dừng quy trình đang chạy hoặc xóa phần mềm phần mềm độc hại).
- Wazuh RESTful API: Dịch vụ này cung cấp giao diện để tương tác với Wazuh. Nó được sử dụng để quản lý cài đặt cấu hình của Agent và Server, theo dõi trạng thái tổng thể, quản lý và chỉnh sửa decoder và rule Wazuh, đồng thời truy vấn về trạng thái của các endpoint được giám sát. Wazuh Dashboard cũng sử dụng thành phần này.
- Wazuh cluster daemon: Dịch vụ này được sử dụng để mở rộng quy mô Wazuh Server theo chiều ngang, triển khai chúng dưới dạng một cluster. Loại cấu hình này,

kết hợp với bộ cân bằng tải mạng, mang lại khả năng sẵn sàng cao và cân bằng tải. Wazuh cluster daemon là thứ mà các Wazuh Server sử dụng để giao tiếp với nhau và để giữ đồng bộ hóa.

- Filebeat: được sử dụng để gửi các sự kiện và cảnh báo đến Wazuh Indexer. Đọc đầu ra 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 cụm bộ chỉ mục Wazuh nhiều nút.

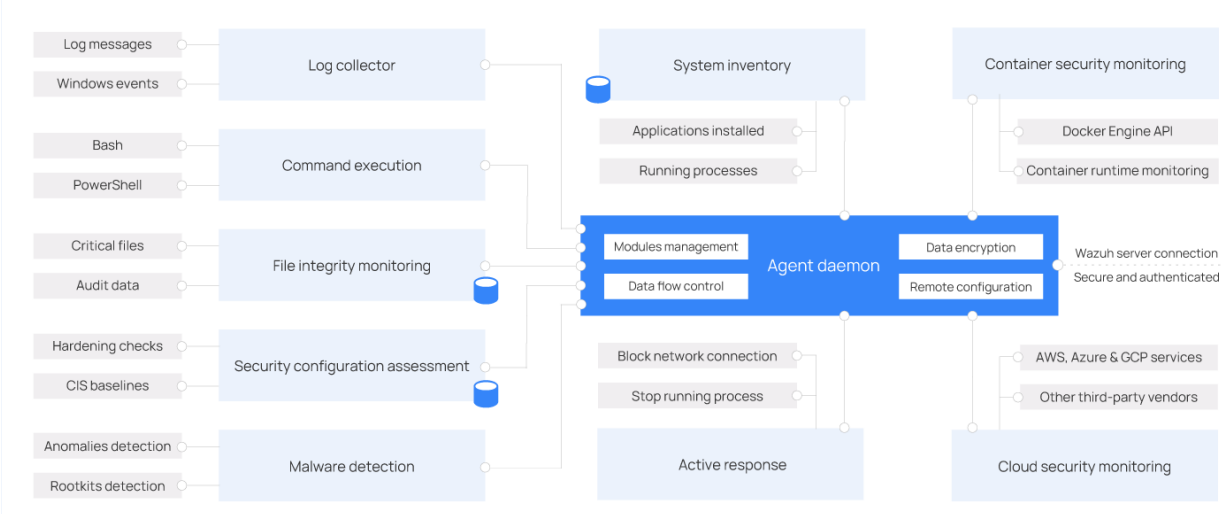
3.3. Wazuh Agent

Wazuh Agent có thể được cài đặt trên các hệ điều hành như Linux, Windows, macOS, Solaris, AIX và các hệ điều hành khác. Có thể được triển khai cho laptop, máy tính để bàn, máy chủ, cloud, container hoặc máy ảo.

Agent giúp bảo vệ hệ thống bằng cách cung cấp khả năng ngăn chặn, phát hiện và phản hồi các mối nguy hại. Agent cũng được sử dụng để thu thập các loại dữ liệu về ứng dụng và hệ thống khác nhau sau đó chuyển tiếp đến Wazuh Server thông qua một kênh được mã hóa và xác thực.

3.3.1. Kiến trúc của Agent

Wazuh Agent có kiến trúc module. Mỗi thành phần phụ trách các nhiệm vụ riêng khác nhau, bao gồm giám sát hệ thống file, đọc log messages, thu thập dữ liệu được kiểm kê, rà quét cấu hình hệ thống và tìm kiếm phần mềm độc hại, báo cáo tình hình của nó cho server. Người dùng có thể quản lý các module Agent thông qua việc cài đặt cấu hình, điều chỉnh cho các trường hợp sử dụng cụ thể.

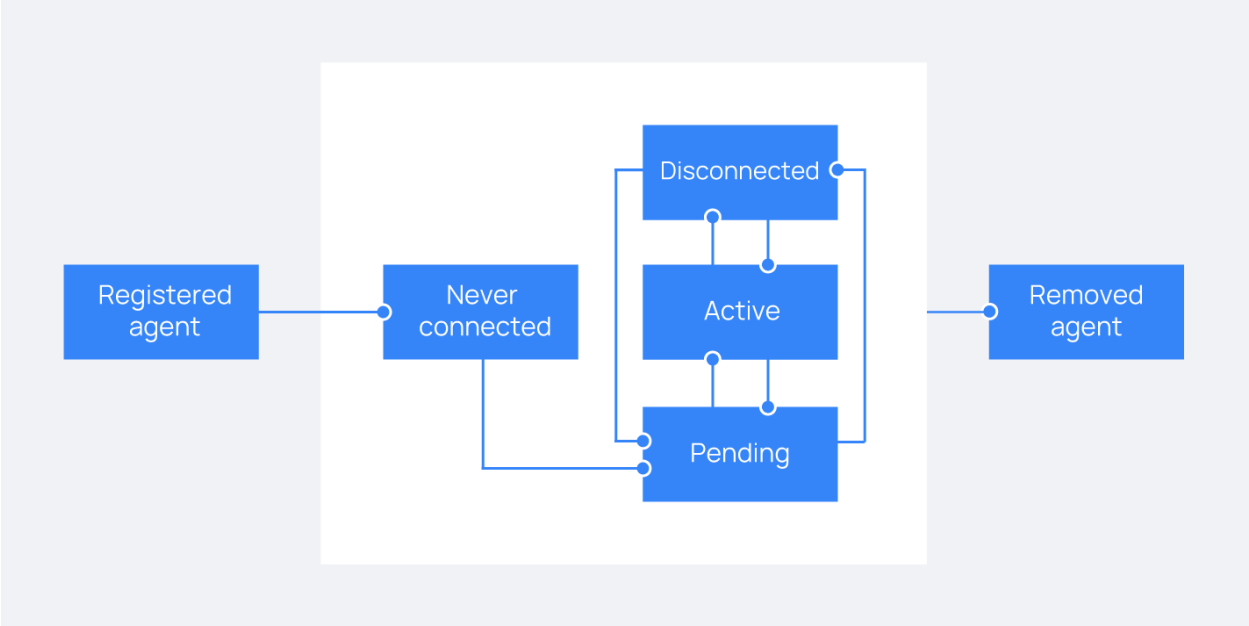


Hình 13. Kiến trúc và các thành phần của Wazuh Agent

Mỗi module trong Agent chạy trên một tiến trình (hoặc task) riêng và được tổng hợp và gửi đi nhờ một tiến trình mang tên Agent daemon tới cổng 1514 của server qua giao thức Wazuh message protocol. Agent daemon đồng thời còn chịu trách nghiệm điều khiển luồng dữ liệu, mã hoá dữ liệu và định cấu hình từ xa cho server.

Hình trên bao gồm tất cả module có thể có trong Agent nhưng không nhất thiết phải là toàn bộ. Giả dụ như module Cloud security monitoring không cần thiết phải có nếu Agent được cài đặt để giám sát một container và ngược lại.

Về vòng đời của một Agent:



Hình 14. Các trạng thái của Agent khi đã được đăng ký

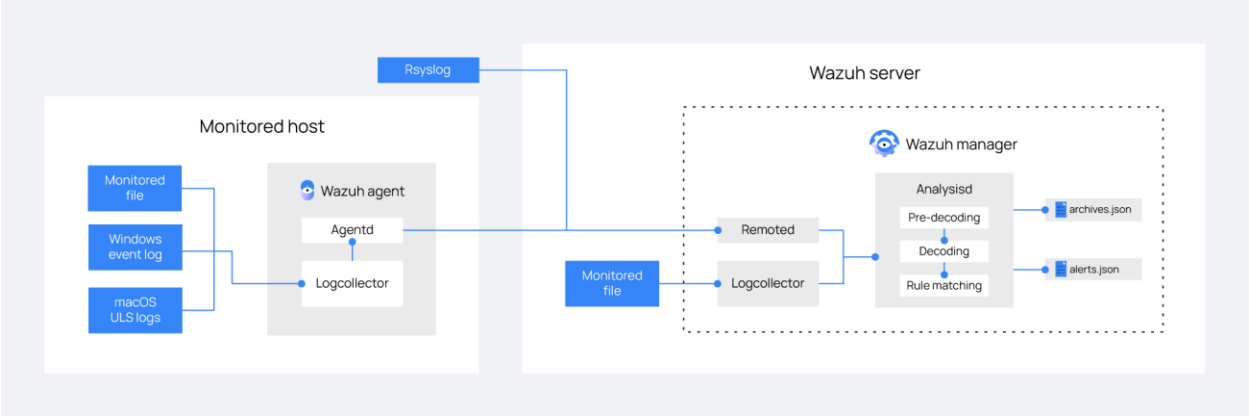
Các trạng thái có thể có của một Agent sau khi được cài đặt và được đăng ký:

- Never connected: được đăng ký nhưng chưa được kết nối
- Pending: Chưa hoàn tất quá trình xác thực do nhận được yêu cầu từ Agent nhưng không nhận được thêm thông tin gì khác. Trạng thái này được lặp lại mỗi lần Agent khởi động và nếu bị dừng ở trạng thái này thì chỉ ra rằng có vấn đề ở tường lửa.
- Active: Kích hoạt là trạng thái kết nối thành công
- Disconnected: Nếu Server không nhận được bất kỳ tin nhắn keep alive nào trong hàm agentS_disconnection_time với mặc định 10 phút một lần.

Các module lại được chia thành các tính năng riêng, có module sẽ lưu giữ và ghi lại dữ liệu như: File integrity monitoring, Security configuration assessment, System Inventory. Sau đây là chi tiết về các module phải có của Agent

3.3.2. Log collector

Thành phần này có cả trên các Agent và Server được dùng để đọc các file log có sẵn và các Windows event, thu thập các log messeage của hệ điều hành và ứng dụng. Log collector hỗ trợ các bộ lọc XPath cho các Windows event và nhận dạng định dạng log nhiều dòng như Linux Audit logs. Đồng thời có thể bổ sung thêm dữ liệu cho các sự kiện dạng .json bằng cách thêm vào meta-data. Mô hình hoạt động của module này được mô tả như sau:



Hình 15. Module Logcollector

- Định cấu hình cho việc nhận các log file hoặc hoặc các event bằng cách cấu hình cho file /var/ossec/etc/ossec.conf trên server:

```
<localfile>
        <location>/var/log/example.log</location>
        <log_format>syslog</log_format>
</localfile>
```

- Rsyslog trên hình thể hiện cho remote syslog hay log được gửi đi từ các Agentless (các thiết bị mạng như firewall, tường lửa) thông qua syslog. Syslog có thể được gửi thông qua một cổng tuỳ chỉnh hoặc lưu trữ syslog dưới dạng bản rõ và giám sát với Wazuh:

```
<ossec_config>
        <remote>
                <connection>syslog</connection>
                <port>Cổng</port>
                <protocol>Giao thức</protocol>
                <allowed-ips>Ip của Wazuh server</allowed-ips>
        </remote>
</ossec_config>
```

- Sau khi log được gửi tới Server thông qua 2 module Agent daemon – Remote dameon thì log tiến tới giai đoạn được phân tích
- Pre-decoding hay tiền giải mã trích xuất các thông tin từ syslog ra như: timestamp, hostname, and program name từ header của log
- Decoding là giai đoạn tìm kiếm bộ giải mã phù hợp với log qua đó trích xuất các trường thông tin có trong log.
- Pha tiếp theo là dò quét liệu thông tin trong log có trùng với rule nào không
- Khi thông tin log trùng với rule, cảnh báo được phát đi
- Dữ liệu nếu được cảnh báo lưu trữ tại: `/var/ossec/logs/alerts/alerts.(json|log)`
- Nếu không được phát cảnh báo thì lưu trữ tại: `/var/ossec/logs/archives/archives.(json|log)`

3.3.3. File integrity monitoring (FIM)

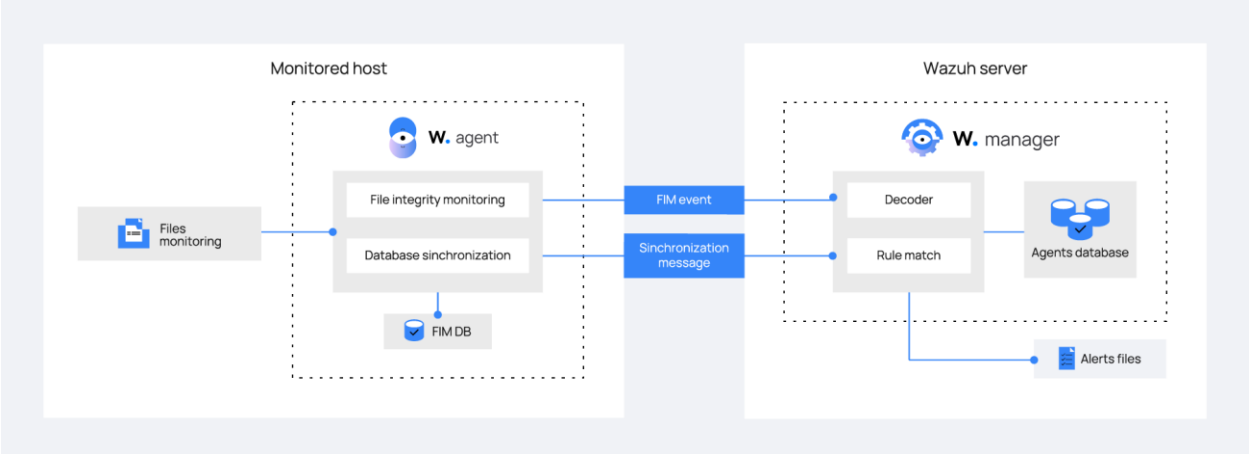
Giám sát tính toàn vẹn của file (FIM) là một quy trình bảo mật được sử dụng để giám sát tính toàn vẹn của file hệ thống và ứng dụng. FIM là một lớp bảo vệ an ninh quan trọng đối với bất kỳ tổ chức nào để giám sát các tài sản nhạy cảm.

Nó cung cấp khả năng bảo vệ cho các file dữ liệu, ứng dụng và thiết bị nhạy cảm bằng cách theo dõi, quét và xác minh tính toàn vẹn của chúng. Nó giúp các tổ chức phát hiện các thay đổi đối với các file quan trọng trên hệ thống, giúp giảm nguy cơ dữ liệu bị đánh cắp hoặc xâm phạm.

Module này cho phép kiểm tra các tệp để xem liệu chúng có bị thay đổi hay không, thay đổi như thế nào và khi nào cũng như ai hoặc cái gì thay đổi chúng. Module Wazuh FIM so sánh thông tin cơ bản với thông tin của phiên bản mới nhất của tệp bằng cách

lưu trữ các mã hash <checksum> bằng thuật toán sha, md5, … ban đầu của tệp và so sánh mã hash của tệp sau khi thay đổi.

Mô hình hoạt động của module như sau:



Hình 16. Module FIM

Như đã đề cập ở phần đầu, FIM là module có cơ sở dữ liệu, cụ thể là sử dụng hai cơ sở dữ liệu. Một là sơ sở dữ liệu được dựng lên dựa trên SQlite trên endpoint được giám sát và được lưu trữ tại: /var/ossec/queue/fim/db trên Linux. Một cơ sở dữ liệu khác về FIM được lưu trữ trên Server được đánh ID theo Agent tại: /var/ossec/queue/db.

Module FIM giúp Agent và Server đồng bộ với nhau bằng cách luôn cập nhật cơ sở dữ liệu có trong Server. Dữ liệu FIM có trong Server này lại được dùng để phục vụ các truy vấn API, giả dụ như Wazuh Dashboard.

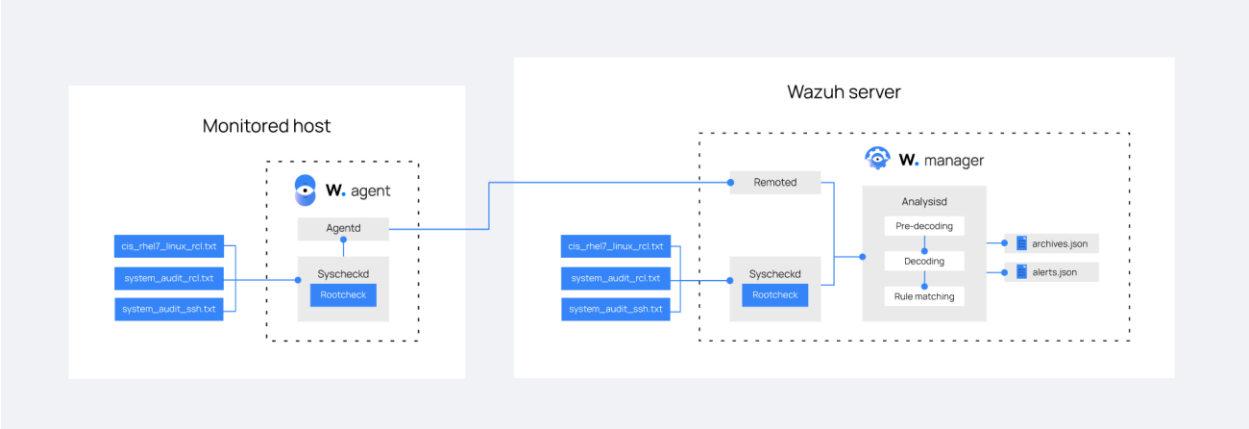
Các thuộc tính cơ bản được sử dụng trong module này:

- Realtime = “yes”: theo dõi thời gian thực
- Ghi lại cái thuộc tính trước khi thay đổi như mã hash, group, size, chủ sở hữu, …
- <frequency>: tần số quét
- <scan_time> và <scan_day>: đặt lịch quét
- report_changes: báo cáo nội dung bị thay đổi trong tệp
- <ignore>: bỏ qua

3.3.4. Security configuration assessment (SCA)

Sử dụng module SCA - Security Configuration Assessment để giám sát các tệp cấu hình để đảm bảo chúng tuân thủ các chính sách, tiêu chuẩn hoặc các yêu cầu nhằm tang cường bảo mật cho hệ thống. Việc quét định kỳ giúp module này có thể phát hiện ra các ứng dụng dễ bị tấn công, chưa được vá lỗi hoặc cấu hình bảo mật sai.

Mô hình hoạt động của module SCA:



Hình 17. Module SCA

*Lưu ý: module rootcheck trong ảnh được thay thế bằng module SCA từ phiên bản wazuh 3.9

Các chính sách, tiêu chuẩn, … trên được đọc dưới dạng .txt, có thể được tự phát triển hoặc dựa theo các tiêu chuẩn cho từng hệ điều hành, dịch vụ được đưa ra bởi Center of Internet Security – CIS. Bao gồm các file hardening của các hệ điều hành họ redhat và Debian; các dịch vụ như SSH, …

Thực hiện cấu hình kiểm tra bằng cách thêm vào file ossec.conf các file hardening cần thiết:

Cấu hình

```
<rootcheck>
  <frequency>Tần số quét</frequency>
  <system_audit>FILE</system_audit>
  <system-audit>FILE</system_audit>
</rootcheck>
```

Các chức năng chính của module này bao gồm:

- Kiểm tra các tiến trình đang chạy. Sử dụng các systemcalls như getsid hay getpgid để kiểm tra các tiến trình đang chạy vì câu lệnh ps có khả năng bị rootkit lẩn tránh.
- Kiểm tra các port ẩn

Các cổng ẩn có thể được mở để giao tiếp với kẻ tấn công. Hàm bind() của ngôn ngữ C được sử dụng để quét các cổng trong hệ thống.

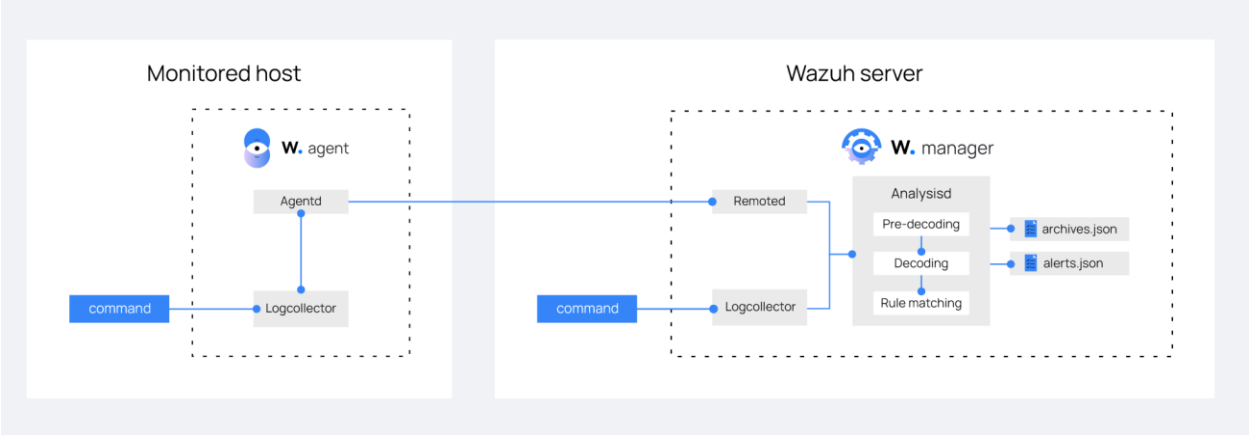
- Kiểm tra các tệp và quyền bất thường
  Module kiểm tra các tệp suid, thư mục ẩn và các tệp khác để kiểm tra các quyền bất thường.
- Kiểm tra các file ẩn sử dụng system calls
  Quét toàn bộ hệ thống, so sánh sự khác biệt giữa kích thước được thống kê và kích thước tệp khi sử dụng system calls fopen và read; so sánh mỗi nút của thư mục với opendir và readdir. Nếu mà không trùng nhau thì có thể malware tồn tại. Các thư mục mặc định mà module này giám sát trong Unix gồm: /bin, /sbin, /usr/bin, /usr/sbin, /dev, /lib, /etc, /root, /var/log, /var/mail, /var/lib, /var/www, /usr/lib, /usr/include, /tmp, /boot, /usr/local, /var/tmp và /sys.
- Kiểm tra thư mục /dev, đây là thư mục có thể chứa các tệp tin đáng ngờ.
- Kiểm tra network interfaces.
  Wazuh quét tìm bất kỳ giao diện mạng nào trong hệ thống có bật chế độ hỗn tạp. Nó kiểm tra đầu ra của lệnh ifconfig. Nếu một giao diện ở chế độ hỗn tạp, nó sẽ kích hoạt cảnh báo. Giao diện mạng ở chế độ hỗn tạp có thể là dấu hiệu cho thấy có phần mềm độc hại.
- Kiểm tra rootkit
- Sử dụng chữ ký từ các file mặc định để quét và phát hiện: rootkit_files.txt, rootkit_trojans.txt và win_malware_rcl.txt.

3.3.5. Command execution

Module này được xuất hiện từ phiên bản 3.1.0, cho phép Server chạy các lệnh, các script trên Agent từ xa để trích xuất các thông tin cần thiết cho việc theo dõi thường xuyên hoạt động của hệ điều hành như dung lượng ổ cứng, bất thường liên quan với việc có các port mới được mở, …

Đầu ra của các command này có thể được định cấu hình như một log file và được xử lý như một log file (xem lại mục ***3.3.2)***. Đây là một chức năng hiệu quả nhưng đem lại một vài rủi ro với việc thực thi câu lệnh từ xa. Đọc thêm tại: https://wazuh.com/blog/scheduling-remote-commands-for-wazuh-agents/ để xem thêm các biện pháp bảo mật nhằm tránh sử dụng sai module này.

Dưới đây là mô hình hoạt động của module này:



Hình 18. Module Command Execution

3.3.6. System inventory

Chứa thông tin về phần cứng và phần mềm trong cơ sở hạ tầng CNTT. Giữ tất cả tài sản giúp các tổ chức tối đa hóa khả năng hiển thị của phần cứng và phần mềm trong môi trường.

Để duy trì kho lưu trữ hệ thống tập trung, các Agent thu thập thông tin hệ thống từ các điểm cuối được giám sát và gửi đến Server. Module Syscollector chịu trách nhiệm thu thập dữ liệu đó từ mỗi Agent. Dữ liệu mà Agent thu thập bao gồm thông tin về phần cứng và hệ điều hành, chi tiết phần mềm đã cài đặt, giao diện mạng, cổng và các quy trình đang chạy. Agent cũng có thể thu thập dữ liệu về các bản cập nhật Windows từ các điểm cuối của Windows. Có thể định cấu hình loại thông tin nào muốn thu thập hoặc bỏ qua.

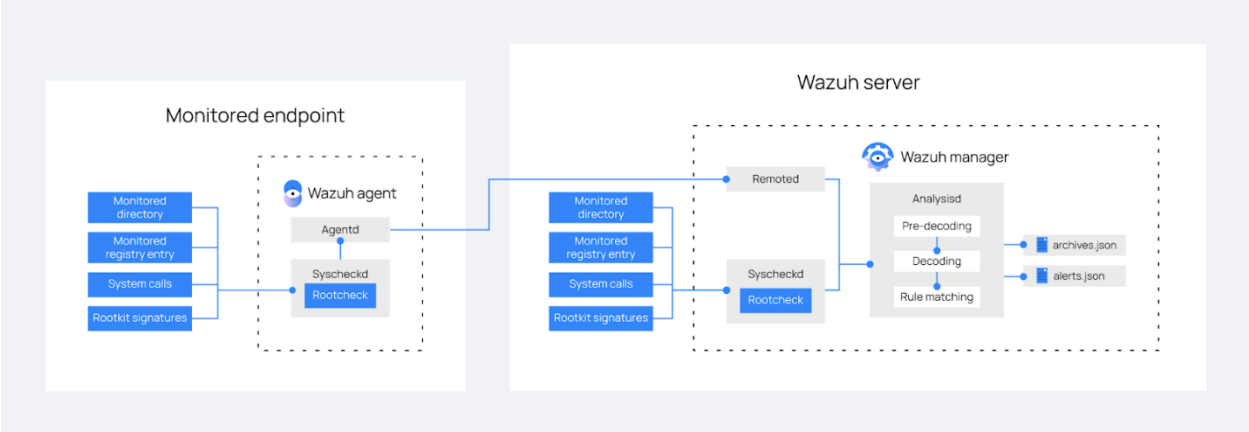
Người dùng có thể tạo báo cáo kiểm kê hệ thống từ bảng Dashboard, đây có thể là tài nguyên hữu dụng trong quá trình tìm kiếm mối đe dọa. Thông tin có trong báo cáo có thể được sử dụng để xác định các ứng dụng, quy trình, dịch vụ không mong muốn và các tạo phẩm độc sử dụng mô-đun Syscollector để thu thập thông tin liên quan từ điểm cuối được giám sát. Khi dịch vụ đại lý bắt đầu trên một điểm cuối được giám sát, mô-đun Syscollector sẽ chạy quét định kỳ và thu thập dữ liệu về các thuộc tính hệ thống được xác định trong cấu hình của bạn. Dữ liệu đầu tiên được lưu trữ trong cơ sở dữ liệu cục bộ tạm thời trên điểm cuối.

Agent chuyển tiếp dữ liệu mới được thu thập từ cơ sở dữ liệu cục bộ của nó đến Server. Mỗi Agent sử dụng một cơ sở dữ liệu riêng trên Server. Server cập nhật các bảng thích hợp của cơ sở dữ liệu inventory trên Server bằng cách sử dụng thông tin mà Agent gửi. Ví dụ: Server lưu trữ thông tin liên quan đến phần cứng trong một bảng có tên là sys_hwinfo.

Dashboard tự động hiển thị dữ liệu được lưu trữ trong cơ sở dữ liệu inventory. Tuy nhiên, có thể truy vấn cơ sở dữ liệu bằng API Wazuh hoặc công cụ SQLite.

3.3.7. Malware detection

Module này không phát hiện các malware dựa trên chữ ký mà sử dụng các hành vi bất thường để làm đầu vào phát hiện các malware tiềm năng mà chữ ký đã bỏ sót. Malware detection sử dụng hai module chính trong việc phát hiện bất thường là FIM và SCA. Do đã trình bày 2 module này ở phía trên, phần này sẽ không trình bày chi tiết lại.

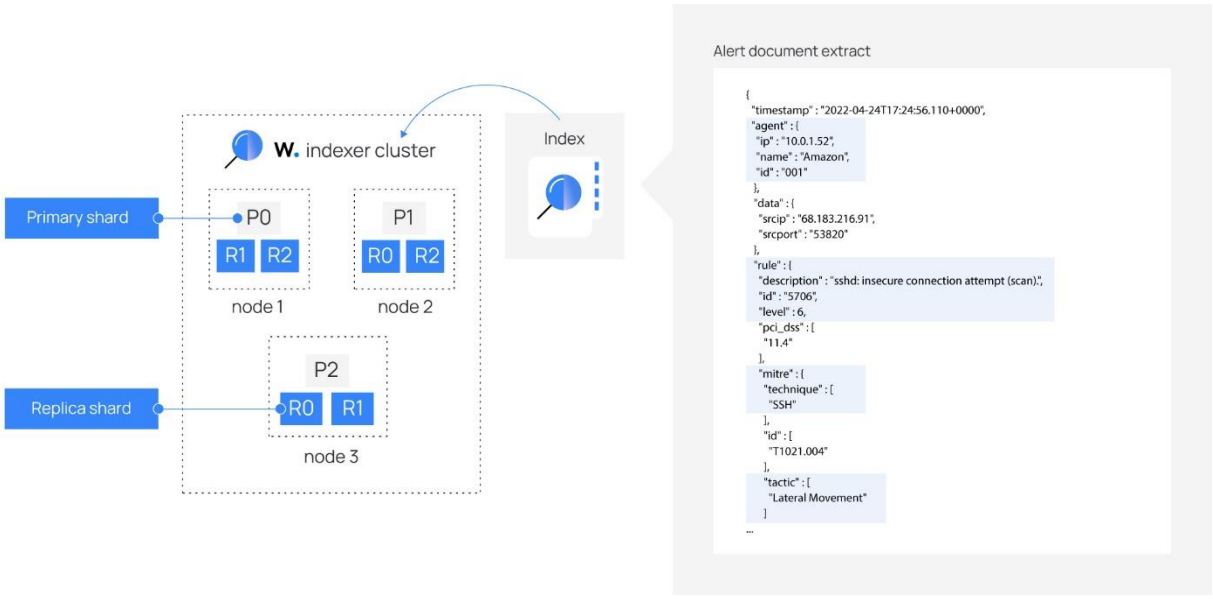


Hình 19. Module Malware Detection

3.4. Wazuh Indexer

Đây là một trong hai thành phần có sự thay đổi đáng kể từ phiên bản Wazuh 4.3. Wazuh Indexer thay thế cho Elasticsearch. Lý do cho sự thay đổi này rất đơn giản: Wazuh là một giải pháp mã nguồn mở còn bộ công cụ ELK stack đã bắt đầu thu phí với những thành phần nâng cao, mở rộng.

Wazuh Indexer là một nhánh phát triển của Opensearch, là một công cụ phân tích và tìm kiếm toàn văn bản có khả năng mở rộng cao. Thành phần này lập chỉ mục và lưu trữ các cảnh báo; phân tích và tìm kiếm dữ liệu theo thời gian thực; đóng vai trò giống như cơ sở dữ liệu trong bộ công cụ này.



Hình 20. Wazuh Indexer

Thành phần này lập chỉ mục, lưu trữ các cảnh báo do Server tạo ra. Dữ liệu được lưu dưới dạng JSON. Mỗi một loại dữ liệu lại đánh một khoá, tên trường hoặc thuộc tính, kiểu dữ liệu, …

Indexer có thể được cài đặt dưới dạng một node đơn hoặc đa node với nhiều điểm trong mô hình cài đặt phân tán, qua đó mang lại khả năng mở rộng và tính sẵn dùng cho hệ thống. Mỗi node đóng vai trò như một server có chức năng hoàn chỉnh với nhiệm vụ: đánh index, search, lưu trữ dữ liệu; node có một unique name không trùng với các node khác.

Trong node lại có nhiều index – nơi chứa các document liên quan tới nhau thông qua việc đánh index, mỗi document sẽ được đánh một hay nhiều index. Việc tìm kiếm các tài liệu này được dựa trên phương thức inverted index, có nghĩa là các keyword sẽ được lưu cùng với số trang thay vì lưu tuần tự các chỉ mục; điều này tạo nên tính nhanh và near-realtime searching cho Wazuh trong việc tìm kiếm (được công bố dưới 1 giây).

Các chỉ số khác nhau để lưu trữ các loại sự kiện khác nhau, mặc định như sau:

Index	Mô tả
wazuh-alerts	Lưu trữ các cảnh báo được tạo bởi Server. Được tạo mỗi khi một sự kiện vượt qua rule có mức độ ưu tiên đủ cao (ngưỡng này có thể định cấu hình).
wazuh-archives	Lưu trữ tất cả các sự kiện (dữ liệu lưu trữ) mà Server nhận được, cho dù chúng có vi phạm rule hay không.
wazuh-monitoring	Lưu trữ dữ liệu liên quan đến trạng thái Agent theo thời gian. Được giao diện web sử dụng để biểu thị thời điểm các Agent riêng lẻ đang hoặc đã hoạt động, ngắt kết nối hoặc chưa bao giờ được kết nối.
wazuh-statistics	Lưu trữ dữ liệu liên quan đến hiệu suất của Server. Được giao diện web sử dụng để biểu thị số liệu thống kê hiệu suất.

Hình 21. Các chỉ số của Indexer

Một index lại được chia thành các shards, có thể hiểu shards là kết quả của việc phân nhỏ các index với các chức năng tương tự index, có thể đứng độc lập và có thể được phân tán trên nhiều node khác nhau. Việc phân nhỏ này giúp việc dữ liệu đảm báo tính sẵn dùng, đồng thời các shards có thể tìm kiếm đồng thời một lúc nhằm tăng tốc độ truy vấn thông tin.

Đơn vị nhỏ nhất trong Indexer là document lưu dưới dangj JSON, mỗi document lại có một kiểu dữ liệu gọi là type, các kiểu dữ liệu này đa dạng như: ngày, giờ, vị trí địa lý, numer, string, double, …

Wazuh Indexer là thành phần quan trọng có tính quan trọng trong việc thể hiện tính năng SIEM của bộ công cụ này. Việc dựa trên một công cụ hướng tài liệu mạnh mẽ như Elasticsearch cung cấp khả năng tìm kiếm near-realtime searching đáp ứng khả năng tìm kiếm tức thì để phân tích và phản ứng sự cố; ngoài ra việc phát triển thành một nhánh của Opensearch (dựa trên Elastichsearch) giúp bộ công cụ này hoàn toàn kiểm soát các chức năng mà vẫn giữ khả năng mở trong ma nguồn của mình.

3.5. Wazuh Dashboard

Đây là thành phần dựa trên Opensearch Dashboard, trước đây là dựa trên Kibana. Cung cấp giao diện người dùng web linh hoạt và trực quan để khai thác, phân tích và trực quan hoá dữ liệu cảnh báo và sự kiện bảo mật. Ngoài ra còn ucng cấp các tính năng cho kiểm soát dựa trên vai trò và đăng nhập một lần. Các thành phần chính của Dashboard bao gồm:

- Phân tích và biểu diễn dữ liệu
- Giám sát và cấu hình Agent
- Quản lý chính Wazuh: Rules, Decoders, …
- Developer tools: bộ test rule và bộ tương tác với Wazuh API

3.6. Luật trong Wazuh

Luật là một phần quan trọng, cốt yếu của Wazuh để tạo nên tính năng của bộ công cụ này. Luật trong Wazuh nhìn chung dựa vào các bộ luật trong Ossec được cải tiến để giám sát các thành phần của hệ điều hành, các dịch vụ, các thành phần quan trọng. Luật được viết và lưu trên Wazuh Manager (server) dưới dạng .xml (Extensible Markup Language) và có cây thư mục như sau:

```
/var/ossec/
        ├─ etc/
        |   ├─ decoders/
        |   |       └─ local_decoder.xml
        |   └─ rules/
        |         └─ local_rules.xml
        └─ ruleset/
              ├─ decoders/
              └─ rules/
```

Hình 22. Cây thư mục quy tắc và bộ giả mã

Thư mục /var/ossec/ruleset/rules/ chứa các file rule mặc định có sẵn của wazuh gồm 4287 luật được đánh ID đến 99,999. (các file luật sẽ được mô tả ở phụ lục)

File /var/ossec/etc/local_rules.xml là tệp luật rỗng, sinh ra với mục đích cho phép người dung tạo và tuỳ chỉnh luật theo ý muốn (custom rule), được đánh ID từ 100,000 đến 120,000.

3.6.1. Cấp bậc luật

Luật trong Wazuh mang 16 cấp độ từ 0 tới 15 tuần tự tăng dần mức độ nguy cơ, được đánh dấu bằng thẻ <level> tới hệ thống được giám sát nêu ra dưới bảng sau:

Level	Tiêu đề	Mô tả
0	Bỏ qua	Không có hành động. Được sử dụng để tránh phát hiện giả. Các quy tắc này được quét trước tất cả các quy tắc khác. Bao gồm các sự kiện không liên quan đến bảo mật.

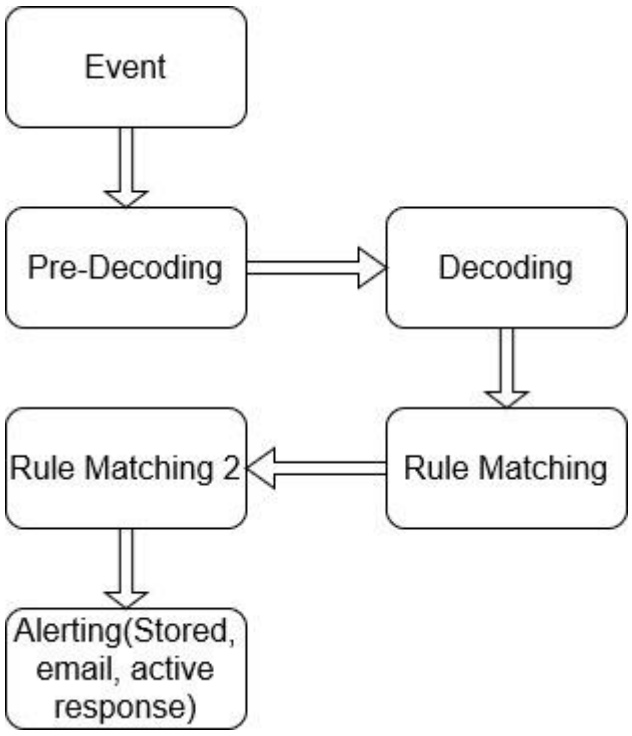
1		
2	Thông báo với ưu tiên thấp	Thông báo hệ thống hoặc thông báo trạng thái. Chúng không liên quan đến bảo mật.
3	Sự kiện xác thực thành công	Chúng bao gồm các lần đăng nhập thành công, các sự kiện cho phép tường lửa, v.v.
4	Lỗi hệ thống với ưu tiên thấp	Lỗi liên quan đến cấu hình không tốt hoặc thiết bị/ứng dụng không được sử dụng. Chúng không liên quan đến bảo mật và thường do cài đặt mặc định hoặc kiểm tra phần mềm gây ra
5	Lỗi do người dung tạo ra	Bao gồm mật khẩu bị bỏ lỡ, hành động bị từ chối, v.v. Không liên quan đến bảo mật
6	Tấn công liên quan thấp	Những điều này chỉ ra một Worm hoặc vi-rút không ảnh hưởng đến hệ thống (như red code cho máy chủ apache, v.v.). Chúng cũng bao gồm các sự kiện IDS thường xuyên và lỗi thường xuyên xảy ra
7	Bắt được các “bad word”	Những sự kiện này thường không được phân loại và có thể có một số liên quan đến bảo mật
8	Lần đầu gặp	Bao gồm các sự kiện nhìn thấy lần đầu tiên. Lần đầu tiên một sự kiện IDS được kích hoạt hoặc lần đầu tiên người dùng đăng nhập. Nó cũng bao gồm các hành động liên quan đến bảo mật (như khởi động trình nghe trộm hoặc hành động tương tự).
9	Lỗi từ nguồn không hợp lệ	Bao gồm các nỗ lực đăng nhập với tư cách là người dùng không xác định hoặc từ một nguồn không hợp lệ. Có thể có liên quan đến bảo mật (đặc biệt nếu được lặp lại). Chúng cũng bao gồm các lỗi liên quan đến tài khoản "admin" (root).

10	Nhiều lỗi do người dùng tạo	Chúng bao gồm nhiều mật khẩu tệ, nhiều lần đăng nhập thất bại, v.v. Những điều này có thể chỉ ra một cuộc tấn công hoặc có thể chỉ là người dùng vừa quên thông tin đăng nhập của mình
11	Cảnh báo kiểm tra toàn vẹn	Chúng bao gồm các thông báo liên quan đến việc sửa đổi các tệp nhị phân hoặc sự hiện diện của rootkit (bởi SCA). Đây có thể chỉ ra một cuộc tấn công đã thành công. Cũng bao gồm các sự kiện IDS sẽ bị bỏ qua (nhưng có số lần lặp lại cao).
12	Sự kiện quan trọng	Chúng bao gồm các thông báo lỗi hoặc cảnh báo từ hệ thống, kernel, v.v. Đây có thể chỉ ra một cuộc tấn công chống lại một ứng dụng cụ thể.
13	Lỗi bất thường (quan trọng cao)	Hầu như nó match với một mẫu tấn công phổ biến
14	Sự kiện an ninh quan trọng	Hầu như được thực hiện với mối tương quan và nó chỉ ra một cuộc tấn công.
15	Tấn công nghiêm trọng	Không có khả năng là giả. Cần được xử lý ngay lập tức.

Bảng 1. Các cấp độ luật trong Wazuh

3.6.2. Cách thức hoạt động

Luật là một phần trong Wazuh, nằm trong luồng hoạt động chung và có mô hình hoạt động như sau:



Hình 23. Mô hình hoạt động của luật trong Wazuh

Các pha bắt đầu với việc nhận các Event từ Remoted – như log từ agent và agentless, các thông tin được thực thi từ command. Thông tin được tiền xử lý bằng bước Pre-Decoding để xác định các thông số: full event, timestamp, hostname, program_name. Ví dụ như sau:

```
**Phase 1: Completed pre-decoding.
        full event: 'Dec 25 20:45:02 MyHost example[12345]: User 'admin'
logged from '192.168.1.100''
        timestamp: 'Dec 25 20:45:02'
        hostname: 'MyHost'
        program_name: 'example'
```

Các thông số sau đó được chuyển tới bước decoding để phân giải chi tiết các sự kiện thành các trường: name(tên sự kiện như sshd), dstuser, srcip, ... Ví dụ về quá trình decoding như sau:

```
**Phase 2: Completed decoding.
        name: 'example'
        dstuser: 'admin'
        srcip: '192.168.1.100'
```

Sau khi nhận được các thông tin từ decoding, wazuh sẽ lọc các luật để tìm luật khớp với thông tin được gửi tới:

```
**Phase 3: Completed filtering (rules).
        id: '100010'
        level: '0'
        description: 'User logged'
        groups: '['custom_rules_example']'
        firedtimes: '1'
        mail: 'False'
```

Từ thông tin trên ta nhận thấy đây là một sự kiện người dùng đăng nhập hệ thống match với luật 100010 có mức cảnh báo là 0 và sẽ không gửi mail đi. Thông tin này sẽ được lưu lại trong hệ thống dù có cảnh báo hay không như đã nêu ở các phần trước.

3.6.3. Nhóm các luật mặc định được sử dụng trong phạm vi báo cáo

0010-rules_config.xml	**Hệ thống**	**Sinh các template cho các rule khác**
0015-ossec_rules.xml	Hệ thống	Hoạt động và thông tin của Agent, các module của Wazuh như rootcheck, audit và các thông tin về ổ cứng, usb, sự thay đổi file log, thông tin về active response.
0016-wazuh_rules.xml	Hệ thống	Thông báo về việc remote của server tới agent; module: syscollector; kiểm tra phần trăm database của module fim.
0020-syslog_rules.xml	Hệ thống	Bắt gặp các bad_word; thông tin về syslog daemon, hệ thống file; network file system; điều khiển truy cập; xinet daemon; openLDAP (Lightweight Directory Access Protocol-Xác thực tập trung); rshd daemon (thực thi dòng lệnh từ xa); hệ thống email; nhân Linux; cron jobs (lên lịch tự động); quyền sudo, su; thêm sửa xoá user; vpn pptpd; syslog fts (full text search); dpkg (Debian package); yum; scsi controller (Small Computer System Interface)

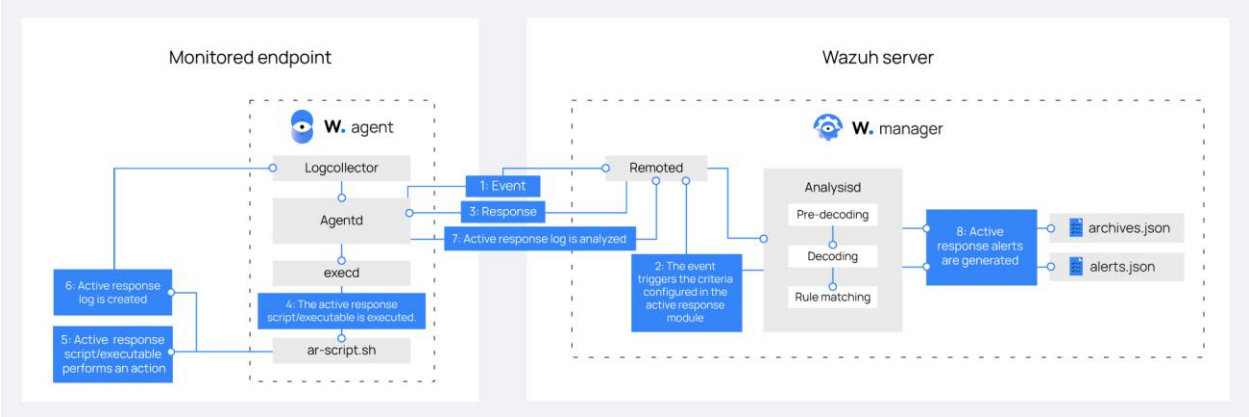
0365-auditd_rules.xml	Audit	Kiểm toán daemon, chế độ promiscuous có đang bật, tiến trình hay file thực thi bất thường; kiểm soát truy cập: DAC, MAC, RBAC; kiểm soát user và group; SElinux; kiểm soát hệ thống file và thư mục
0280-attack_rules.xml	Hệ thống	Các dấu hiệu bị tấn công như: xác thực thành công hoặc thất bại, buffer overflow, user thay đổi thông tin, virus; leo thang đặc quyền; dấu hiệu bị quét từ một ip
0095-sshd_rules.xml và 0305, 0685	Dịch vụ SSH	Liên quan tới dịch vụ ssh cổng 22: xác thực thành công, thất bại; bị tấn công, khai thác, …

Bảng 2. Các file luật mặc định được sử dụng trong phạm vi báo cáo

*Xem thêm tại phụ lục việc phân loại các luật còn lại

3.7. Phản hồi chủ động trong Wazuh

3.7.1. Mô hình hoạt động của module Active Response



Hình 24. Modul Active Response

Active response nằm trong chu trình cuối của luồng hoạt động trong Wazuh; nằm chung hàng với: cảnh báo và lưu trữ.

Mô tả về luồng hoạt động của module này được thể hiện rõ rang trên ảnh. Event được nhận bởi Remoted daemon, so sánh với luật, nếu trùng khớp với cấu hình cài đặt từ trước, Server sẽ gửi yêu cầu với Module này trên Agent để kích hoạt các script/thực thi thông qua ar-scipts.sh. Active response được thực thi, tạo ra log lưu vào actve-response.log và được thu thập bởi module Logcollector và gửi tới Server thông qua module Agentd -> Server tạo cảnh báo và lưu lại.

Có thể sử dụng bất kỳ ngôn ngữ lập trình nào để tuỳ chỉnh và tạo các Active-response.

Có 2 loại Active-respponse là:

- Stateless active response: phản hồi không trạng thái, tức là các phản hồi sẽ chỉ được thực hiện một lần duy nhất và sẽ không đặt lại trạng thái mặc định ban đầu. Ví dụ như một IP cố gắng SSH vào máy tính khi không có quyền sẽ bị block vĩnh viễn.
- Stateful active response: phản hồi có trạng thái, tức là các phản hồi sẽ được thực hiện hoàn nguyên hay dừng lại sau một khoảng thời gian nhất định. Các user-case được trình bày phía dưới trong phần **5.5** đều sử dụng loại phản hồi này để block user hoặc drop một IP trong một khoảng thời gian xác định.

3.7.2. Các thành phần mặc định được cài đặt

Các phản hồi ngược mặc định được lưu trong `/var/ossec/active-response/bin`.

Scrips	**Mô tả**
disable-account	Disables a user account
firewall-drop	Thêm một địa chỉ IP vào bảng deny của iptables
firewalld-drop	Thêm một địa chỉ IP vào drop list. Firewalld cần được cài đặt trên điểm cuối.
host-deny	Thêm một địa chỉ IP vào `/etc/hosts.deny` file.
ip-customblock	Chỉnh sửa Wazuh block, sửa đổi cho một phản hồi dễ dàng hơn.
Ipfw	Firewall-drop phản hồi script được tạo cho IPFW. IPFW cần được cài đặt trên máy endpoint.
Npf	Firewall-drop phản hồi script được tạo cho NPF. NPF cần được cài đặt trên máy endpoint.

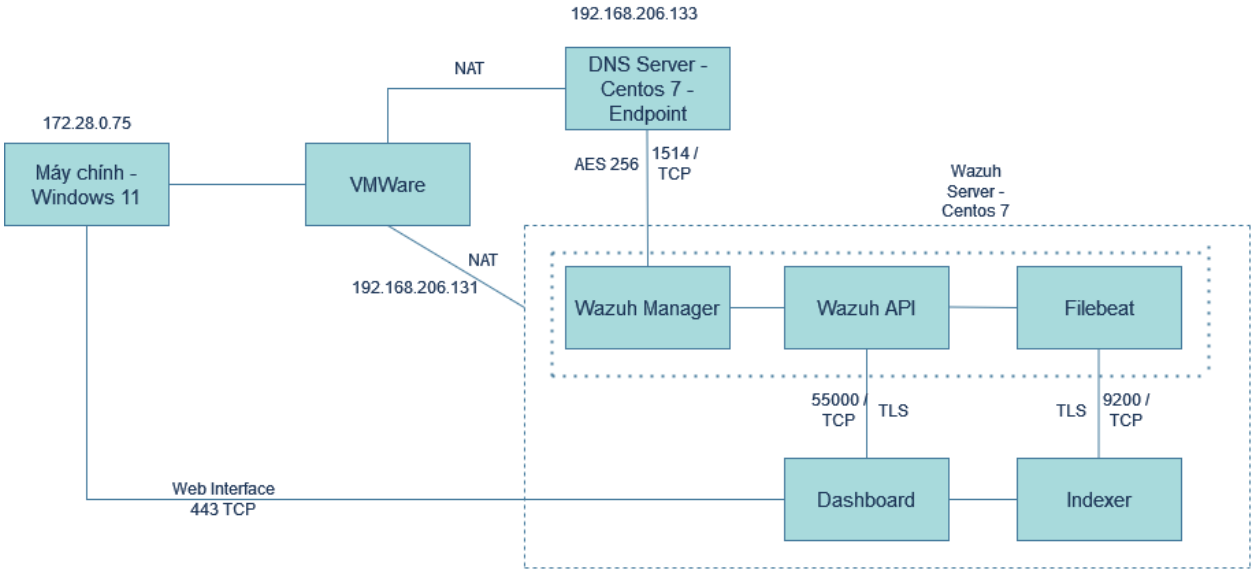
Wazuh-slack	Đăng thông báo lên Slack. Cần một slack hook thông qua một `extra_args`.
Pf	Firewall-drop phản hồi script được tạo cho PF. PF cần được cài đặt trên máy endpoint.
Restart.sh	Restarts the Wazuh Agent or manager.
Restart-wazuh	Restarts the Wazuh Agent or manager.
Route-null	Thêm một địa chỉ IP vào định tuyến null.

Bảng 3. Các Active-response script mặc định

Tính năng này của wazuh đại diện cho "Response" trong công nghệ XDR – Extended Detection and Response nhưng nhu đã nói trong phần giới thiệu về IDS/IPS, tính năng phản hồi với tính chất của một IPS cần được sử dụng một cách thận trọng và tuân thủ theo chính sách riêng được công ty đề ra tránh việc ảnh hưởng tới tính khả dụng của hệ thống đối với người sử dụng.

IV. MÔ HÌNH HỆ THỐNG

4.1. Mô hình cài đặt



Hình 25. Mô hình cài đặt

4.2. Thông số cài đặt

STT	Tên thiết bị	Thông số môi trường	Thông số cài đặt	Chú thích
1	Wazuh Server	OS: CentOS 7 IP: 192.168.206.131 Hostname: wazuh-user	CPU: 4 nhân 4 luồng MEM: 8GB SSD: 50 GB Card mạng: NAT	
2	DNS Server	OS: CentOS 7 IP: 192.168.206.133 Hostname: ducthiago	CPU: 1 nhân 1 luồng MEM: 1GB SSD: 20 GB Card mạng: NAT	
3	Máy tính cá nhân	OS: Windows 11 IP: 172.28.0.75 Hostname: vandu	CPU: 6 nhân 12 luồng MEM: 16GB SSD: 500 GB	Đây là tài nguyên khi chưa chia sẻ

			Card mạng: NAT	cho máy ảo

Bảng 4. Thông số cài đặt

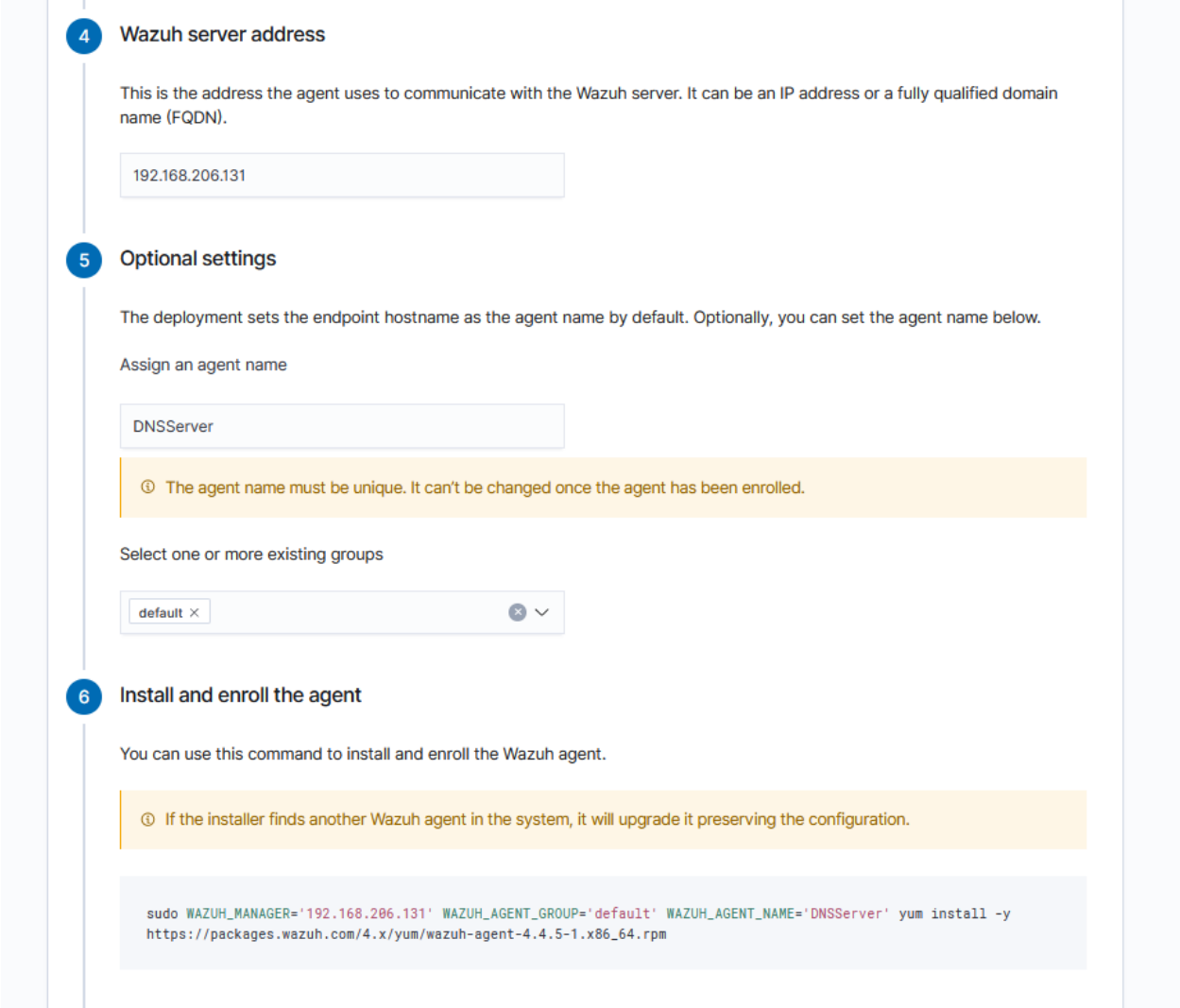
4.3. Cách đăng ký Agent

Có hai phương pháp chính được dùng để đăng ký Agent với Wazuh manager với yêu cầu chung cho cả 2 phương pháp như sau:

- Wazuh manager phải đang được chạy
- Wazuh agent phải đã được cài và chạy trên máy endpoint
- Các cổng: 1514/TCP – 1515/TCP – 55000/TCP phải được mở

4.3.1. Đăng ký thông qua cấu hình Agent

Đây là kiểu đăng ký tự động cho Agent, cũng là kiểu đăng ký được sử dụng trong phần báo cáo này. Sử dụng câu lệnh để đăng ký với cách biến đăng ký. Câu lệnh này có thể tự viết hoặc lấy thông qua các lựa chọn trên Wazuh Dashboard như sau:



Hình 26. Đăng ký Agent

Sử dụng câu lệnh này để cài đặt và đăng ký đồng thời trên endpoint.

4.3.2. Đăng ký thông qua manager API

Trong cách đăng ký này, Agent sau khi được cài đặt sẽ gửi một yêu cầu đăng ký tới máy chủ và máy chủ sẽ trả lại một key, Agent nhận key này và nhập thủ công để kết nối.

V. CÁC THÀNH PHẦN CẦN ĐƯỢC GIÁM SÁT VÀ CÁCH THỰC HIỆN

5.1. Giám sát gói tin và lưu lượng dữ liệu vào, ra tại điểm cuối

Module được sử dụng: Command Execution trên agent

Các thông số được theo dõi: tổng gói tin vào, ra (rx/tx packet); tổng lượng dữ liệu vào/ra (rx/tx bytes); lượng gói tin vào/ra môi giây (rx/tx_packet/s); lượng dữ liệu vào/ra mỗi giây (rx/tx_kb/s); số gói tin vào/ra bị drop (rx/tx_dropped).

Các bước thực hiện:

- Xác định được các thông số cần theo dõi như trên
- Sử dụng các câu lệnh (command) để lấy được các thông số trên:

- Để lấy thông tin về tổng số gói tin, dữ liệu vào ra; bị drop:
  - Cat /sys/class/net/ens33/statics/[thông tin cần lấy]
  - Để lấy thông tin về dữ liệu trên mỗi giây, sử dụng một scipt đơn giản viết bằng bash với công thức: sau – trước / thời gian -> dùng sleep trong bash

```
 GNU nano 2.3.1                          File: netmo.sh

#!/bin/bash

time="1"     # second
int="ens33"   # network interface
                txpkts_old="`cat /sys/class/net/$int/statistics/tx_packets`" # sent packets
                rxpkts_old="`cat /sys/class/net/$int/statistics/rx_packets`" # recv packets
                txbytes_old="`cat /sys/class/net/$int/statistics/tx_bytes`" # sent bytes
                rxbytes_old="`cat /sys/class/net/$int/statistics/rx_bytes`" # recv bytes
                        sleep $time
                txpkts_new="`cat /sys/class/net/$int/statistics/tx_packets`" # sent packets
                rxpkts_new="`cat /sys/class/net/$int/statistics/rx_packets`" # recv packets
                txbytes_new="`cat /sys/class/net/$int/statistics/tx_bytes`" # sent bytes
                rxbytes_new="`cat /sys/class/net/$int/statistics/rx_bytes`" # sent bytes

                txpkts="`expr $txpkts_new - $txpkts_old`"                   # evaluate expressions for sent packets
                rxpkts="`expr $rxpkts_new - $rxpkts_old`"                   # evaluate expressions for recv packets
                txbps0="`expr $txbytes_new - $txbytes_old`"
                rxbps0="`expr $rxbytes_new - $rxbytes_old`"
                txbps="`expr $txbps0 / 1024`"
                rxbps="`expr $rxbps0 / 1024`"
                        echo "tx $txpkts pkts/5s - rx $rxpkts pkts/s on interface $int"
                        echo "tx $txbps kb/5s - rx $rxbps kb/s on interface $int"
```

Hình 27. Bash script lấy thông tin

> bash [file bash] | grep ens33 | awk ‘NR==2{print $2,$6}’

- Tiếp theo, cấu hình cho agent tạo file ossec.conf để thực hiện việc gửi các thông tin trên (kèm với timestamp và hostname) tới server. Ví dụ:

```
<ossec_config>
  <localfile>
     <log_format>full_command</log_format>
     <command>bash /usr/netmo.sh | grep ens33 | awk 'NR==1{print $2,$6}'</command>
     <alias>packet_recv_tran</alias>
     <out_format>$(timestamp) $(hostname) packet_recv_tran: $(log)</out_format>
     <frequency>5</frequency>
  </localfile>

  <localfile>
     <log_format>full_command</log_format>
     <command>bash /usr/netmo.sh | grep ens33 | awk 'NR==2{print $2,$6}'</command>
     <alias>kb_recv_tran</alias>
     <out_format>$(timestamp) $(hostname) kb_recv_tran: $(log)</out_format>
     <frequency>5</frequency>
  </localfile>
```


```
<ossec_config>
  <localfile>
     <log_format>full_command</log_format>
     <command>bash /usr/netmo.sh | grep ens33 | awk 'NR==1{print $2,$6}'</command>
     <alias>packet_recv_tran</alias>
     <out_format>$(timestamp) $(hostname) packet_recv_tran: $(log)</out_format>
     <frequency>5</frequency>
  </localfile>

  <localfile>
     <log_format>full_command</log_format>
     <command>bash /usr/netmo.sh | grep ens33 | awk 'NR==2{print $2,$6}'</command>
     <alias>kb_recv_tran</alias>
     <out_format>$(timestamp) $(hostname) kb_recv_tran: $(log)</out_format>
     <frequency>5</frequency>
  </localfile>
```


Hình 28. Cấu hình trên Agent để lấy thông tin mạng

- Thông tin được gửi từ agent sẽ được xử lý như một log tại server. Cấu hình decoder của server tại local_decoder.xml để phân giải và lấy các thông số cần

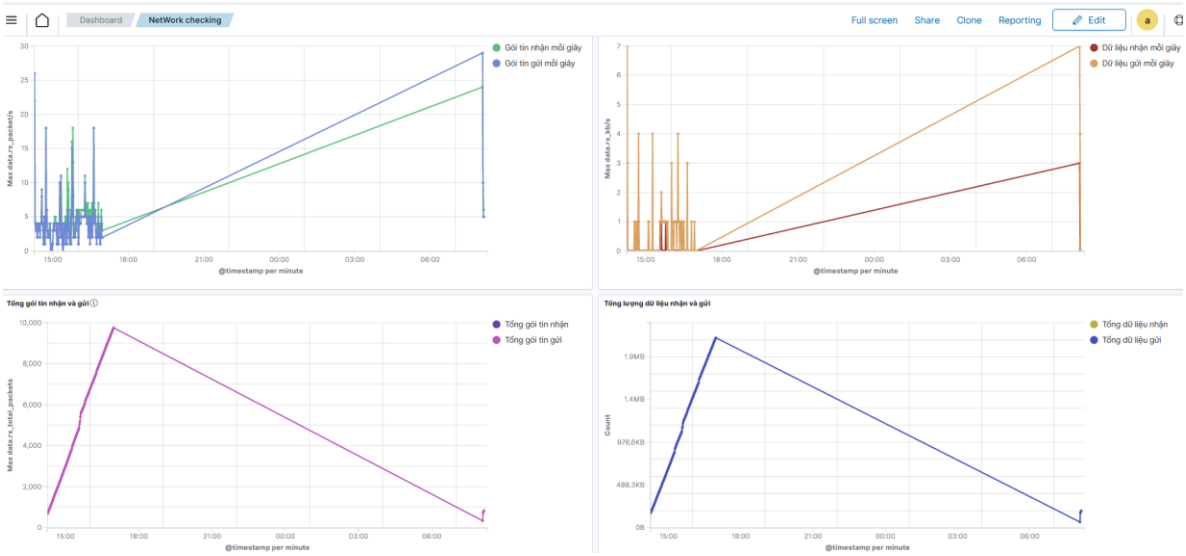
thiết. Ví dụ:

```
</decoder>

  <decoder name="packet_recv_tran">
     <program_name>packet_recv_tran</program_name>
  </decoder>
  <decoder name="packet_recv_tran">
     <parent>packet_recv_tran</parent>
     <prematch>ossec: output: 'packet_recv_tran':\.</prematch>
     <regex offset="after_prematch">(\S+) (\S+)</regex>
     <order>tx_packet/s, rx_packet/s</order>
  </decoder>

   <decoder name="kb_recv_tran">
     <program_name>kb_recv_tran</program_name>
   </decoder>
   <decoder name="kb_recv_tran">
     <parent>kb_recv_tran</parent>
     <prematch>ossec: output: 'kb_recv_tran':\.</prematch>
     <regex offset="after_prematch">(\S+) (\S+)</regex>
     <order>tx_kb/s, rx_kb/s</order>
   </decoder>
```

Hình 29. Bộ giải mã lấy thông tin về mạng trên Server

- Tạo các rule tại file local_rules.xml trên server để tạo cảnh báo khi các thông số trên được gửi thành công tới server
- Server đã nhận được thông tin, các trường thông tin cũng đã được đánh index. Tuy nhiên các thông số trên mặc định sẽ là kiểu string, không thể dùng để tạo bảng, trực quan hoá dữ liệu. Cần thay đổi kiểu từ string sau double tại file /etc/filebeat/wazuh-template.json
- Index lại các trường sử dụng dev-tool trên Dashboard

- Sau khi đổi kiểu dữ liệu thành công thì tạo bảng biểu diễn:



Hình 30. Kết quả hiển thị thông tin về mạng trên Dashboard

5.2. Giám sát mức sử dụng phần cứng hệ thống

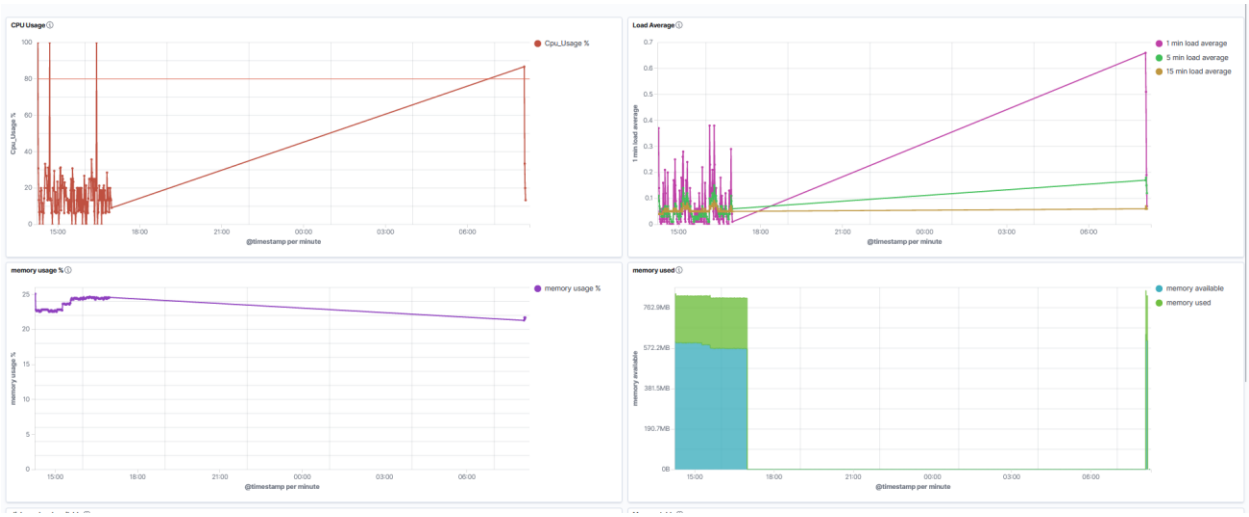
Module được sử dụng: Command Execution

Các thông số cần theo dõi: cpu_usage_%, memory_usage_%, disk_usage_%, 1min_loadAverage, 5mins_loadAverage, 15mins_loadAverage, memory_used_bytes, memory_available_bytes, disk_used_bytes, disk_free_bytes

➢ Theo dõi cpu, ram, disk và load average

Các bước thực hiện: Tương tự như phần trên

Kết quả hiển thị:



Hình 31. Kết quả hiển thị thông tin về phần cứng trên Dashboard

5.3. Giám sát hành vi bất thường có thể là rootkit

Sử dụng các chức năng như đã nêu trong mục 3.3.4 về module SCA. Các chức năng và luật trong module này đều là mặc định và được viết trong file rule mặc định 0015-ossec_rules.xml được mô tả trong mục 1.5.3

5.4. Giám sát toàn vẹn file (các thư mục, file quan trọng gồm cả dịch vụ named.conf)

Có hai cách là sử dụng module SCA được nêu trong phần 3.3.3 và các luật được viết trong file 0015-ossec_rules.xml và được decode bằng các decoder liên quan tới syscheck. Hoặc là sử dụng gói cài đặt mặc định được cài đặt trên Centos 7 là auditd, đây cũng là gói cài đặt được viết các rule mặc định tại file 0365-auditd_rules.xml.

Các thư mục quan trọng đều được mặc định theo dõi bởi module SCA. Tuy nhiên, tuỳ theo mỗi hệ thống sẽ có những thư mục, file cần theo dõi với mức độ khác biệt vì nó ảnh hưởng trực tiếp tới cách mà dịch vụ hay hệ thống vẫn hành.

Dưới đây là các thư mục, file quan trọng cần được giám sát:

STT	Thư mục/file	Trạng thái đầu
1	etc/named.conf	-rw-r-----. 1 root named 764 Jul 26 16:31 /etc/named.conf
2	etc/passwd	-rwxr--r--. 1 root root 943 Jul 5 11:35 /etc/passwd
3	etc/shadow	----------. 1 root root 736 Jul 5 11:35 /etc/shadow

Bảng 5. Các file, thư mục cần giám sát

Chi tiết thực hiện việc giám sát file /etc/named.conf:

- Thực hiện thêm luật cho dịch vụ kiểm toán audit sử dụng câu lệnh:
  sudo auditctl -w /etc/named.conf -p rwa -F uid!=0 -F uid!=25 -k power_abuse_named

  Trong đó:
  - w: thư mục hoặc file cần được giám sát
  - p: các thuộc tính cần giám sát
  - r: read, đọc
  - w: write, ghi
  - a: append, thêm vào
  - x: execute, thực thi
  - F: tạo một trường có thể sử dụng toán tử
  - uid: user id với != là nếu user 0(root) hoặc user 25(named) có thực thi hay thay đổi file thì cũng không sao.
  - k: đặt một ID không trùng cho rule này
  - l: liệt kê các luật hiện có
- Kiểm tra các luật đã được thêm thành công hay chưa sử dụng câu lệnh:
  [root@localhost home]# auditctl -l

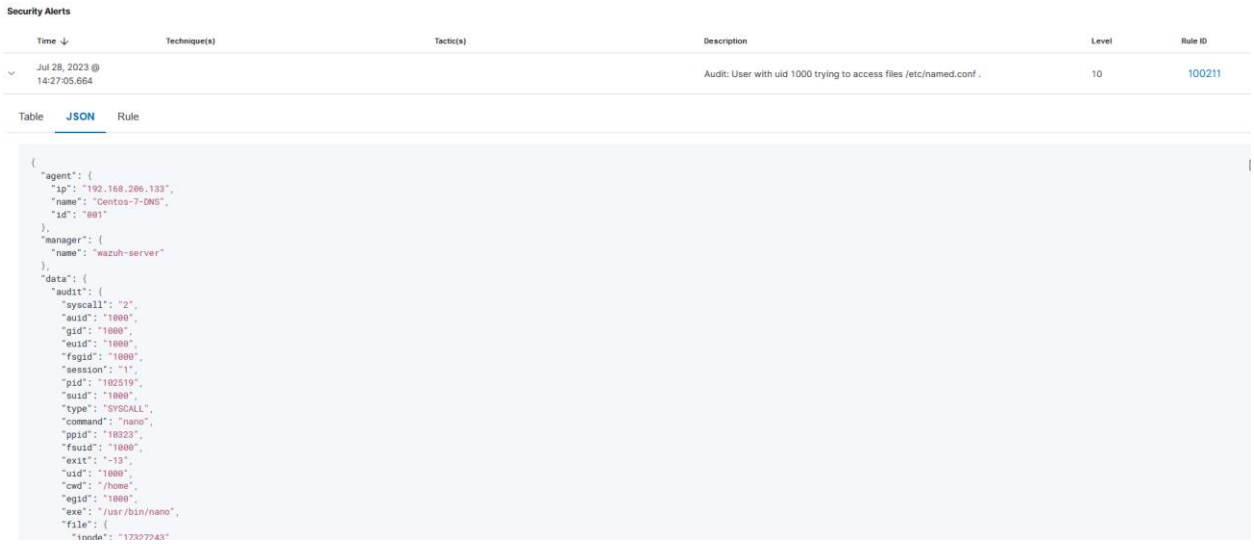
  -a always,exit -S all -F path=/etc/named.conf -F perm=rwa -F uid!=0 -F uid!=25 -F key=power_abuse_named

- Như vậy, việc giám sát file /etc/named.conf đã được thiết lập thành công, mọi thông tin về file này được ghi lại vào file log /var/log/audit/audit.log. Bây giờ cần cài đặt để wazuh xử lý file log này như một file log thông thường.
- Trên wazuh server, tại file /var/ossec/etc/lists/audit-keys, thêm key cho audit:
  power_abuse_named:abuse_named

  Như vậy đã thêm được cặp key:value vào file audit-keys, khi gọi tới value là abuse_named, khoá power_abuse_named sẽ được gọi, tương ứng với luật đã thêm phía trên.
- Tạo luật cho việc giám sát file /etc/named:

```
<group name="audit">
  <rule id="100210" level="10">
    <if_sid>80700</if_sid>
    <list field="audit.key" lookup="match_key_value"
check_value="abuse">etc/lists/audit-keys</list>
    <description>Audit: User with uid $(audit.uid) trying to access $(audit.file.name)
files.</description>
    <group>audit_command,</group>
  </rule>
</group>
```

- Thử sử dụng nano để truy cập vào file /etc/named sử dụng user thường. Kết quả trên Dashboard:



Hình 32. Kết quả cho giám sát file /etc/named.conf

- Tương tự với các file, thư mục khác.

5.5. Phản hồi ngược các cuộc tấn công, hành vi đáng ngờ

- Use-case 1:

  Một user không có quyền kiểm ban đầu với file /etc/named.conf cố gắng thực hiện các hành động đáng ngờ sử dụng các câu lệnh như: nano, vi, chmod, chown, … nhiều lần trong một khoảng thời gian nhất định -> block account trong một khoảng thời gian.

- Trước hết, cần tạo luật ứng với hành động liên tục cố gắng thâm nhập trên trong file /var/ossec/etc/local-rules.xml:

```
  <rule id="100212" level="12" frequency="5" timeframe="30">
   <if_matched_sid>100211</if_matched_sid>
   <description>Bất thường trong việc cố gắng thay đổi file name
d.conf</description>
  </rule>
```

Hình 33. Luật cho use-case 1

- Trong file ossec.conf, cấu hình:

```
<command>
  <name>disable-account</name>
  <executable>disable-account</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>
```

Hình 34. Cấu hình active-response cho use-case 1

- <name>: Đặt tên cho lệnh.
- <executable>: Chỉ định tập lệnh phản hồi tích cực phải chạy sau trình kích hoạt.
- <timeout_allowed>: Cho phép hết thời gian chờ sau một khoảng thời gian. Thẻ này được đặt thành "yes" ở đây, đại diện cho một phản hồi tích cực có trạng thái.

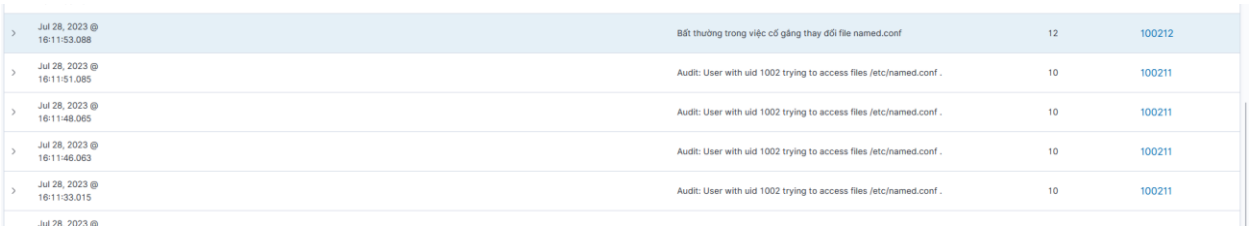
- Trong file ossec.conf này tiếp tục viết vào thẻ <active-response>

```
<active-response>
 <command>disable-account</command>
 <location>local</loation>
 <rules_id>100212</rules_id>
 <timeout>120</timeout>
</active-response>
```

Hình 35. Tạo active-response ứng với rule 100212

Tài khoản này sẽ bị block 2 phút nếu thực hiện đủ 5 lần truy cập khi không đủ quyền trong 30 giây vào file /etc/named.conf

- Kết quả thực hiện:



Jul 28, 2023 @ 16:11:53.088	Bất thường trong việc cố gắng thay đổi file named.conf	12	100212
Jul 28, 2023 @ 16:11:51.085	Audit: User with uid 1002 trying to access files /etc/named.conf .	10	100211
Jul 28, 2023 @ 16:11:48.065	Audit: User with uid 1002 trying to access files /etc/named.conf .	10	100211
Jul 28, 2023 @ 16:11:46.063	Audit: User with uid 1002 trying to access files /etc/named.conf .	10	100211
Jul 28, 2023 @ 16:11:33.015	Audit: User with uid 1002 trying to access files /etc/named.conf .	10	100211

Hình 36. Kết quả cho use-case 1

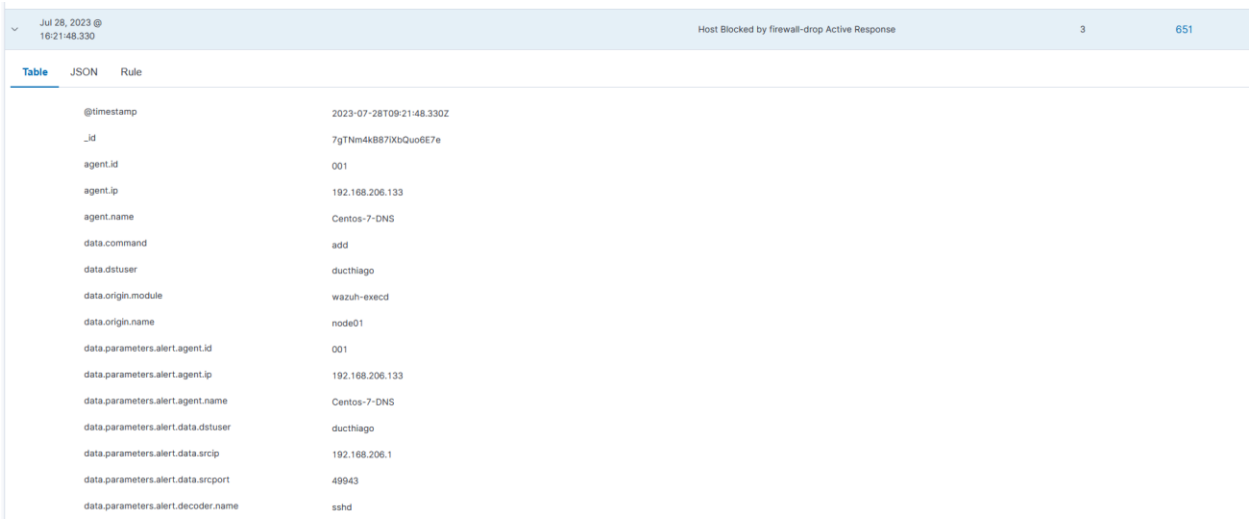
- User-case 2:

  Ngăn chặn một cuộc tấn công vét cạn nhằm xác thực SSH.

```
<active-response>
  <command>firewall-drop</command>
  <location>local</location>
  <rules_id>5763</rules_id>
  <timeout>180</timeout>
</active-response>
```

Hình 37. Cấu hình active-response cho use-case 2 với rule 5763

- Kết quả:



Jul 28, 2023 @ 16:21:48.330 — Host Blocked by firewall-drop Active Response — 3 — 651

Table | JSON | Rule

@timestamp	2023-07-28T09:21:48.330Z
_id	7gTNm4kB87iXbQuo6E7e
agent.id	001
agent.ip	192.168.206.133
agent.name	Centos-7-DNS
data.command	add
data.dstuser	ducthiago
data.origin.module	wazuh-execd
data.origin.name	node01
data.parameters.alert.agent.id	001
data.parameters.alert.agent.ip	192.168.206.133
data.parameters.alert.agent.name	Centos-7-DNS
data.parameters.alert.data.dstuser	ducthiago
data.parameters.alert.data.srcip	192.168.206.1
data.parameters.alert.data.srcport	49943
data.parameters.alert.decoder.name	sshd

Hình 38. Kết quả cho use-case 2

Không thể ping tới máy Centos nữa:

```
C:\Users\vandu>ping 192.168.206.133

Pinging 192.168.206.133 with 32 bytes of data:
Request timed out.
```

Hình 39. Kết quả cho use-case 2

5.6. Cảnh báo thông qua việc gửi email

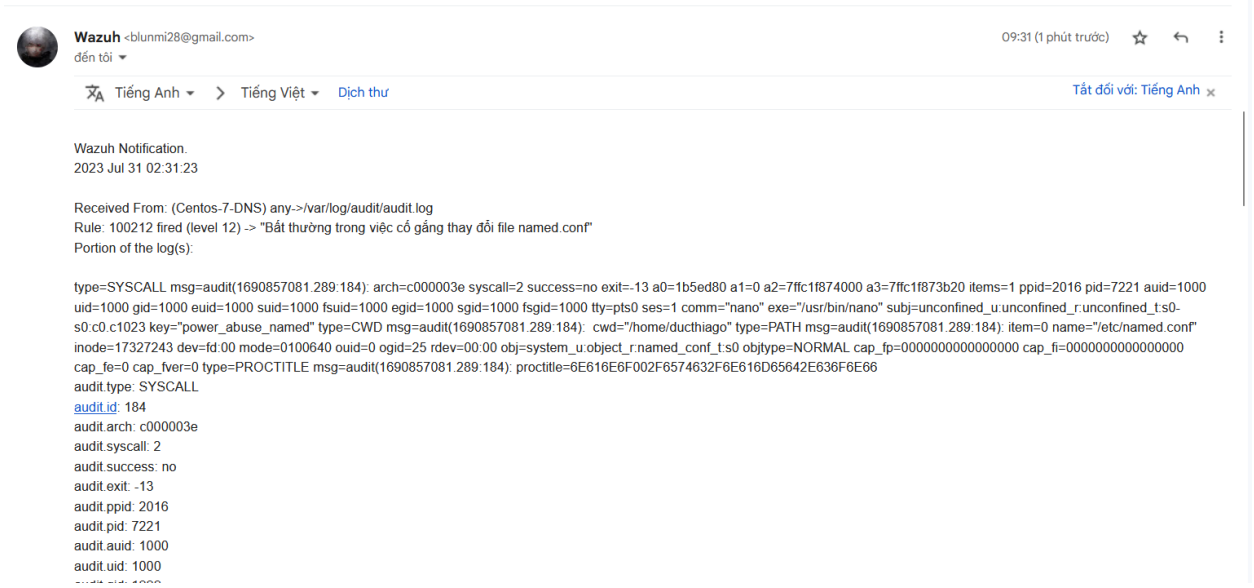
Sau đây là các cài đặt cho việc gửi email:

```
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>yes</email_notification>
    <smtp_server>localhost</smtp_server>
    <email_from>blunmi28@gmail.com</email_from>
    <email_to>ducthiago28@gmail.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
  </global>

  <alerts>
    <log_alert_level>5</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>
```

Hình 40. Cấu hình cho việc gửi email

Dịch vụ mail được sử dụng là SMTP có xác thực và được mã hoá bằng TLS trên cổng 573; tài khoản được sử dụng là gmail [blunmi28@gmail.com] và tài khoản nhận là gmail [ducthiago28@gmail.com]; mỗi giờ sẽ chỉ có tối đa 12 cảnh báo được gửi và chỉ có những cảnh báo lơn hơn hoặc bằng level 12 mới được gửi đi thông qua email.

Kết quả thử nghiệm với rule 100212 – khi một user không có quyền hạn muốn xem, sửa, xoá file named.conf như sau:



Wazuh <blunmi28@gmail.com> 09:31 (1 phút trước)
đến tôi

Tiếng Anh > Tiếng Việt Dịch thư Tắt đối với: Tiếng Anh

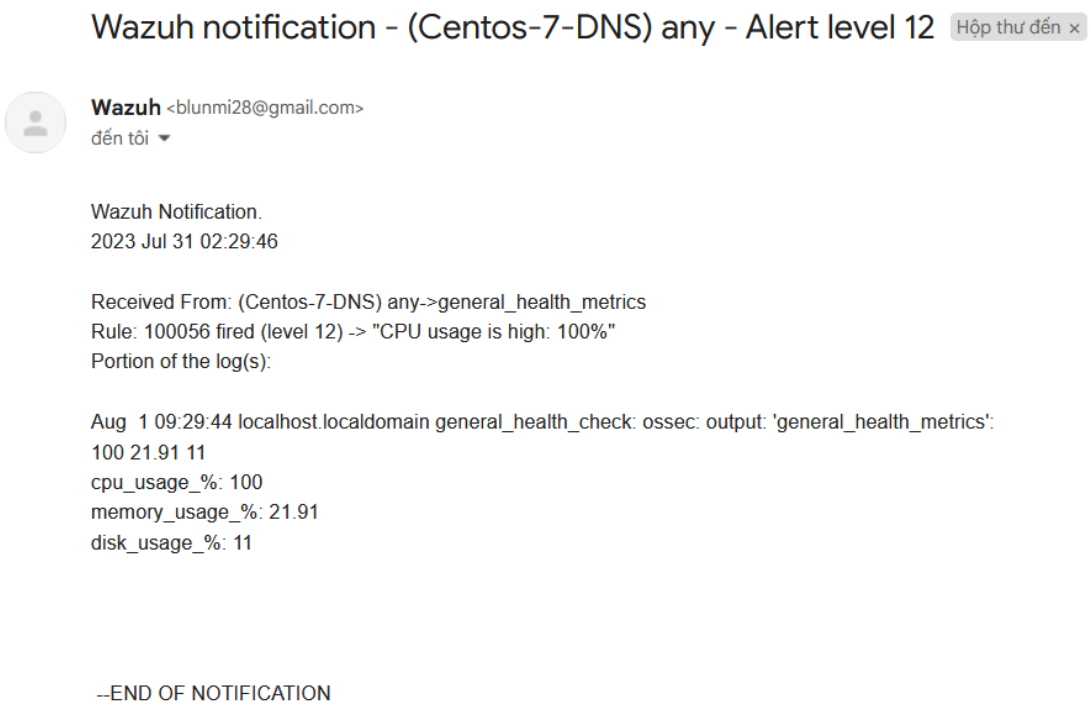
Wazuh Notification.
2023 Jul 31 02:31:23

Received From: (Centos-7-DNS) any->/var/log/audit/audit.log
Rule: 100212 fired (level 12) -> "Bất thường trong việc cố gắng thay đổi file named.conf"
Portion of the log(s):

type=SYSCALL msg=audit(1690857081.289:184): arch=c000003e syscall=2 success=no exit=-13 a0=1b5ed80 a1=0 a2=7ffc1f874000 a3=7ffc1f873b20 items=1 ppid=2016 pid=7221 auid=1000 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=1 comm="nano" exe="/usr/bin/nano" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="power_abuse_named" type=CWD msg=audit(1690857081.289:184): cwd="/home/ducthiago" type=PATH msg=audit(1690857081.289:184): item=0 name="/etc/named.conf" inode=17327243 dev=fd:00 mode=0100640 ouid=0 ogid=25 rdev=00:00 obj=system_u:object_r:named_conf_t:s0 objtype=NORMAL cap_fp=0000000000000000 cap_fi=0000000000000000 cap_fe=0 cap_fver=0 type=PROCTITLE msg=audit(1690857081.289:184): proctitle=6E616E6F002F6574632F6E616D65642E636F6E66
audit.type: SYSCALL
audit.id: 184
audit.arch: c000003e
audit.syscall: 2
audit.success: no
audit.exit: -13
audit.ppid: 2016
audit.pid: 7221
audit.auid: 1000
audit.uid: 1000

Hình 41. Kết quả cho việc gửi cảnh báo email use-case 1

Kết quả thử nghiệm với rule 100056 – khi CPU bị sử dụng tới 100% công suất:



Wazuh notification - (Centos-7-DNS) any - Alert level 12 Hộp thư đến ×

Wazuh <blunmi28@gmail.com>
đến tôi

Wazuh Notification.
2023 Jul 31 02:29:46

Received From: (Centos-7-DNS) any->general_health_metrics
Rule: 100056 fired (level 12) -> "CPU usage is high: 100%"
Portion of the log(s):

Aug 1 09:29:44 localhost.localdomain general_health_check: ossec: output: 'general_health_metrics':
100 21.91 11
cpu_usage_%: 100
memory_usage_%: 21.91
disk_usage_%: 11

--END OF NOTIFICATION

Hình 42. Kết quả cho việc CPU bị dùng tới 100%

TÀI LIỆU THAM KHẢO

[1] P. R. Technology, "linkedin," 27 04 2023. [Online]. Available: https://www.linkedin.com/pulse/hids-vs-nids-pro-resources-consulting. [Accessed 14 07 2023].

[2] A. M. D. Suman Thapa, "THE ROLE OF INTRUSION DETECTION/PREVENTION SYSTEMS IN MODERN COMPUTER NETWORKS: A REVIEW," Minnesota, 2020.

[3] Wazuh, "Blog Wazuh," [Online]. Available: https://wazuh.com/blog/. [Accessed 7 2023].

[4] Wazuh, "Wazuh Documention," [Online]. Available: https://documentation.wazuh.com/current/index.html. [Accessed 7 2023].

[5] VNNIC, "https://vnnic.vn," [Online]. Available: https://vnnic.vn/gi%E1%BB%9Bi-thi%E1%BB%87u-v%E1%BB%81-vnnic. [Accessed 08 08 2023].

[6] P. V. Anh, Hướng Dẫn Cài Đặt Máy Chủ DNS, TP. HCM: Nhà xuất bản Bưu Điện, 2006.

PHỤ LỤC

Đánh nhãn các luật

File rule	Dịch vụ	Nội dung
0010-rules_config.xml	Hệ thống	Sinh các template cho các rule khác
0015-ossec_rules.xml	Hệ thống	Hoạt động và thông tin của Agent, các module của Wazuh như rootcheck, audit và các thông tin về ổ cứng, usb, sự thay đổi file log, thông tin về active response.
0016-wazuh_rules.xml	Hệ thống	Thông báo về việc remote của server tới agent; module: syscollector; kiểm tra phần trăm database của module fim.
0020-syslog_rules.xml	Hệ thống	Bắt gặp các bad_word; thông tin về syslog daemon, hệ thống file; network file system; điều khiển truy cập; xinet daemon; openLDAP (Lightweight Directory Access Protocol-Xác thực tập trung); rshd daemon (thực thi dòng lệnh từ xa); hệ thống email; nhân Linux; cron jobs (lên lịch tự động); quyền sudo, su; thêm sửa xoá user; vpn pptpd; syslog fts (full text search); dpkg (Debian package); yum; scsi controller (Small Computer System Interface)
0025, 0030, 0035, 0040, 0045, 0050, 0055, 0155, 0160, 0165, 0325, 0135, 0140, 0495	Email	Liên quan tới hệ thống Email
0065, 0070, 0080, 0290, 0390, 0540, 0625, 0680, 0700, 0705, 0900	Tường lửa	Các loại tường lửa
0075, 0310	Hệ điều hành	Hệ điều hành cisco ios (Internetwork Operating System), openbsd,

0090-telnetd_rules.xml	Dịch vụ	Dịch vụ telnet trên cổng 23
0095-sshd_rules.xml và 0305, 0685	Dịch vụ SSH	Liên quan tới dịch vụ ssh cổng 22: xác thực thành công, thất bại; bị tấn công, khai thác, …
0100	Dịch vụ	Auditing cho HĐH Solaris
0110, 0220, 0225, 0230, 0330, 0430, 0435, 0440, 0470, 0515, 0565, 0575, 0580, 0585, 0590, 0595, 0600, 0601, 0602, 0605, 0610, 0615, 0620,	Windows	
0315, 0405, 0420, 0525, 0445, 0525,	audit, xác thực	
0120, 0125, 0320, 0385, 0410, 0415, 0480, 0485, 0490, 0505, 0550, 0510, 0520, 0675	Diệt virus, bảo vệ máy tính bên thứ 3	
0105, 0085, 0130, 0145, 0150, 0200, 0340, 0345, 0395, 0425	Các dịch vụ khác	
0115-arpwatch_rules.xml	Hệ thống	Giám sát hệ thống mạng
0170, 0175, 0180, 0185, 0190, 0360	Dịch vụ FTP	
0205, 0210, 0400, 0275	Dịch vụ VPN	
0240, 0475, 0635, 0640	IDS	
0215-policy_rules.xml	Hệ thống	Luật về policy

0245, 0250, 0255, 0260, 0265, 0270	Dịch vụ WEB	
0280-attack_rules.xml	Hệ thống	Các dấu hiệu bị tấn công như: xác thực thành công hoặc thất bại, buffer overflow, user thay đổi thông tin, virus; leo thang đặc quyền; dấu hiệu bị quét từ một ip
0285, 0375	Hệ thống	
0295, 0300, 0380, 0450, 0530, 0535, 0545,	Database	
0195, 0335	DNS	
0350, 0455, 0460, 0500, 0555, 0560, 0630, 0690	Cloud	
0365-auditd_rules.xml	Audit	Kiểm toán daemon, chế độ promiscuous có đang bật, tiến trình hay file thực thi bất thường; kiểm soát truy cập: DAC, MAC, RBAC; kiểm soát user và group; SElinux; kiểm soát hệ thống file và thư mục
0570-sca_rules.xml	Kiểm tra cấu hình bảo mật	****
0710-wazuh-api-rules.xml	Hệ thống	Nhận request các dạng, thông báo xác thực api, địa chỉ IP đang được nghe.

Bảng 6. Đánh nhãn các quy tắc mặc định

Chi tiết luật về dịch vụ named – Bind9

STT	ID	ID Cha	Level	Nội dung
1	12100		0	Nhóm các luật cùng nội dung với named
2	12101	12100	12	Gói DNS không hợp lệ, có khả năng bị tấn công.

3	12109	12100	12	Named gặp lỗi nghiêm trọng, bị tắt.
4	12108	12100	5	Truy vấn cache bị từ chối, có thể do lỗi cấu hình
5	12149	12108	10	Truy vấn cache bị từ tối 8 lần trong 2 phút từ cùng 1 nguồn ip
6	12102	12100	9	Chuyển zone không thành công sử dụng giao thức AXFR
7	12144	12100	9	Server không đủ bộ nhớ để reload lại cấu hình
8	12110		8	Serial number của Master lại nhỏ hơn của Slave
9	12111		8	Không thể thực hiện chuyển giao zone
10	12103	12100	4	Update DNS thất bại | Cấu hình sai
11	12104	12100	4	Cấu hình sai quyền với file log (Không thể đổi tên log file)
12	12105	12100	4	Lỗi khi phân giải tên miền (sử dụng RCODE)
13	12106	12100	4	Bị từ chối thông báo từ máy non-master (lỗi cấu hình)
14	12112	12100	4	Lỗi chuyển vùng (do zone hết hạn)
15	12128	12100	1	Bắt đầu chuyển dùng sử dụng giao thức AXFR
16	12129	12128	4	Chuyển vùng thất bại do không thể kết nối tới máy chủ
17	12133	12100	4	Cảnh báo khi người có có quyền cố gắng truy cập vào file cấu hình named.conf
18	12134	12100	4	Cảnh báo khi người có có quyền cố gắng truy cập vào file cấu hình named.conf
19	12136	12128	4	Không thể kết nối để chuyển zone do không thể truy cập host – master dường như bị down
20	12119	12100	3	Thông báo bắt đầu dịch vụ

21	12125	12100	3	Lỗi cấu hình (không thể reload dịch vụ)
22	12139	12100	3	Yêu cầu chuyển giao zone không hợp lệ (NOAUTH)
23	12130	12100	2	Không thể nghe trên IPv6 interfaces
24	12131	12100	2	Không thể liên kết với một interface (interface ignored)
25	12140	12100	2	Không thể refresh một domain từ máy master
26	12114	12100	1	Hostname chứa ký tự xấu (sử dụng check-name kiểm tra)
27	12116	12100	1	Lỗi cú pháp trong file named.conf
28	12117	12100	1	Thời gian chuyển giao zone đã hết hạn
29	12118	12100	1	Zone bị nhân đôi (đã tồn tại trước đó)
30	12120	12100	1	Thiếu bản ghi A hoặc AAAA
31	12121	12100	1	Zone bị xoá khỏi máy chủ Master
32	12122	12100	1	$ORIGIN của zone và tên được khai bởi bản ghi SOA không giống nhau
33	12127	12100	1	$ORIGIN của zone và tên được khai bởi bản ghi SOA không giống nhau
34	12141	12100	1	Bản ghi SOA không nằm ở trên cùng của zone
35	12145	12100	1	Chuyển vùng không thành công
36	12107	12100	0	Update bị từ chối (sử dụng giao thức động RFC2136)
37	12113	12100	0	Chuyển zone bị hoãn
38	12115	12100	0	Chuyển giao zone được thực hiện
39	12123	12100	0	Đã tồn tại một zone trước đó giống hệt zone này

40	12126	12100	0	Zone bị xoá khỏi máy chủ Master
41	12132	12128	0	Master không có thẩm quyền với zone (not authoritative)
42	12135	12100	0	Tên miền trong SOA -E
43	12137	12100	0	Tên miền được truy vấn cho một zone đã được chuyển giao
44	12138	12100	0	Bản ghi được sử dụng
45	12142	12100	0	Kênh command của zone đang được nghe
46	12143	12100	0	Named đã tạo một zone trống tự động
47	12146	12100	0	Không thể phản hồi một yêu cầu DNS
48	12147	12100	0	Không thể cập nhật tên miền forwarding
49	12148	12100	0	Phân tích cú pháp tệp cấu hình thất bại

Bảng 7. Các quy tắc trong tệp quy tắc mặc định cho dịch vụ Bind9