

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



**Nghiên cứu và triển khai Wazuh XDR cho bảo mật
mạng máy tính**

(Research and deploy Wazuh XDR for network security)

GIẢNG VIÊN HƯỚNG DẪN: ThS. NGUYỄN DUY

**SINH VIÊN THỰC HIỆN:
PHẠM HOÀNG PHÚC – 21521295**

TP. HỒ CHÍ MINH, 2025

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU	1
1.1 Bối cảnh	1
1.2 Giải pháp	1
1.2.1 Extended Detection and Response (XDR)	1
1.2.2 Công nghệ Wazuh.....	2
1.3 Mục tiêu và phạm vi nghiên cứu	3
1.3.1 Phạm vi.....	3
1.3.2 Mục tiêu	3
CHƯƠNG 2: CÔNG NGHỆ HỖ TRỢ	4
2.1 Mod Security	4
2.2 DVWA.....	4
2.3 Snort.....	4
2.4 Suricata	4
2.5 ELK stack	4
2.6 ClamAV.....	5
2.7 VirusTotal.....	5
CHƯƠNG 3: KIẾN TRÚC.....	6
CHƯƠNG 4: NGHIÊN CỨU THỰC NGHIỆM.....	8
4.1 Kịch bản 1: Tương tác giữa ModSecurity và Wazuh trong ngăn chặn tấn công mạng8	
4.1.1 Mô tả	8
4.1.2 Thực hiện	8
4.2 Kịch bản 2: Kiểm tra tương tác giữa Snort và Wazuh	14
4.2.1 Mô tả:	14
4.2.2 Thực hiện	14
4.3 Kịch bản 3: Giám sát command thực thi trên Linux	18
4.3.1 Mô tả:	18
4.3.2 Thực hiện	18
4.4 Kịch bản 4: Giám sát tính toàn vẹn tệp tin (FIM)	22
4.4.1 Mô tả:	22
4.4.2 Thực hiện	22
4.5 Kịch bản 5: Tự động xóa file độc hại và xây dựng IOCs cho threat intelligence.....	24
4.5.1 Mô tả:	24

4.6	Kịch bản 6: Ngăn chặn các tác nhân gây hại đã biết trước	31
4.6.1	Mô tả:	31
4.6.2	Thực hiện:	32
4.7	Kịch bản 7: Tích hợp Wazuh và Teler để phát hiện tấn công.....	37
4.7.1	Mô tả:	37
4.7.2	Thực hiện:	38
4.8	Kịch bản 8: Tích hợp Suricata IDS với Wazuh phát hiện hành vi bất thường	40
4.8.1	Mô tả:	40
4.8.2	Thực hiện:	41
CHƯƠNG 5: KẾT LUẬN	44
CHƯƠNG 6: HƯỚNG PHÁT TRIỂN	45
TÀI LIỆU THAM KHẢO	46

DANH MỤC HÌNH

□&□

Hình 3.1 Mô hình thực nghiệm	6
Hình 4.1 Cài đặt DVWA và ModSecurity trên máy endpoint Ubuntu	9
Hình 4.2 Cài đặt Wazuh Agent trên máy endpoint Ubuntu	9
Hình 4.3 Cấu hình rule ModSecurity	9
Hình 4.4 Attacker thực hiện tấn công SQLi.....	12
Hình 4.5 Attacker thực hiện tấn công XSS	12
Hình 4.6 Log ModSecurity ghi nhận sự kiện tấn công.....	12
Hình 4.7 Dashboard ghi nhận chi tiết log tấn công SQLi.....	13
Hình 4.8 Dashboard ghi nhận chi tiết log tấn công XSS	13
Hình 4.9 rule cấu hình chặn kết nối ngoài port chỉ định.....	14
Hình 4.10 Khi này cảnh báo về kết nối trái phép sẽ được gửi đến Wazuh và hiển thị ở dashboard	15
Hình 4.11 Kiểm tra lại trên một trường hợp khác, lần này ta viết rule để ngăn chặn ICMP flood attack.....	16
Hình 4.12 Rule chặn ICMP flood attack.....	16
Hình 4.13 Attacker ping victim.....	16
Hình 4.14 Attack dùng hping3 để tấn công ICMP flood attack	17
Hình 4.15 Log chặn ICMP flood attack từ Wazuh	17
Hình 4.16 Cấu hình thu thập log audit4.164.17c4.18	18
Hình 4.Thực hiện các lệnh đơn giảnThực75Thực h4.17.....	19
Hình 4. Danh sách command thực hiện4.18	19
Hình 4.19 Kết quả hiển thị trên dashboard của Wazuh	23
Hình 4.20 Chi tiết của hành động thay đổi nội dung file.....	23
Hình 4.21 Tải về một mã độc từ database của malware bazaar và dùng ClamAV để scan.....	25
Hình 4.22 Wazuh cảnh báo file là độc hại và xóa file thành công	26
Hình 4.23 Lưu hash của mã độc để làm IOC.....	26
Hình 4.24 Kiểm tra lại hash của file trên VirusTotal.....	27
Hình 4.25 Kiểm tra lại hash của file trên VirusTotal.....	27
Hình 4.26 Rule wazuh để kích hoạt xóa file tự động và lưu hash.....	28
Hình 4.27 Rule wazuh để kích hoạt xóa file tự động và lưu hash.....	29
Hình 4.28 Rule wazuh để kích hoạt xóa file tự động và lưu hash.....	30

Hình 4.29 Hai máy endpoint chạy máy chủ web Apache	32
Hình 4.30 Hai máy endpoint chạy máy chủ web Apache	33
Hình 4.31 Cấu hình rule cho Wazuh nhận diện địa chỉ IP	33
Hình 4.32 Cấu hình rule cho Wazuh nhận diện địa chỉ IP	33
Hình 4.33 Cấu hình active response thực hiện chặn địa chỉ IP đáng ngờ trong 60 giây	34
Hình 4.34 Máy RHEL truy cập vào máy chủ web	35
Hình 4.35 Wazuh server nhận được log từ active response trả về	35
Hình 4.36 Log active response trả về khi Wazuh nhận thấy hành vi đáng ngờ từ IP máy RHEL	36
Hình 4.37 Chi tiết về 1 log được active response trả về trên Wazuh	36
Hình 4.38 Log active response trả về khi nhận thấy hành vi đáng ngờ từ IP máy RHEL đến endpoint Ubuntu.....	37
Hình 4.39 Chi tiết 1 log được active response trả về trên Wazuh	37
Hình 4.40 Cài đặt Teler	38
Hình 4.41 Rule được tích hợp	38
Hình 4.42 Thực hiện mô phỏng tấn công Nikto	39
Hình 4.43 Log Teler trả về khi bị tấn công	39
Hình 4.44 Log từ Teler được đẩy về Wazuh server	40
Hình 4.45 Chi tiết log mà Wazuh nhận được khi bị tấn công	40
Hình 4.46 Cài đặt Suricata.....	41
Hình 4.47 Thêm rule tích hợp trên Wazuh server	42
Hình 4.48 Thực hiện tấn công mô phỏng.....	42
Hình 4.49 Thực hiện tấn công mô phỏng.....	42
Hình 4.50 Log từ Suricata đẩy về Wazuh server	43
Hình 4.51 Chi tiết log từ Wazuh dashboard.....	43

CHƯƠNG 1: GIỚI THIỆU

1.1 Bối cảnh

Trong thời đại công nghệ hiện nay, sự phát triển nhanh chóng của các hệ thống số hóa, điện toán đám mây, Internet of thing (IoT) và các mô hình làm việc từ xa đã tạo ra một môi trường kỹ thuật số ngày càng phức tạp. Các doanh nghiệp không chỉ phải đối mặt với sự tăng về số lượng và mức độ tinh vi của các cuộc tấn công mạng, từ ransomware, phishing đến các cuộc tấn công APT, mà còn phải bảo vệ lượng dữ liệu khổng lồ trên nhiều nền tảng khác nhau.

Thách thức lớn của việc triển khai bảo mật mang doanh nghiệp là thiếu khả năng phát hiện và phản ứng nhanh với các mối đe dọa, sự rời rạc, thiếu đồng bộ giữa các giải pháp bảo mật hiện có, và tình trạng thiếu hụt chuyên gia an ninh mạng. Ngoài ra, số lượng lớn các cảnh báo giả từ các hệ thống bảo mật hiện có cũng gây khó khăn cho việc tập trung vào các mối nguy thực sự.

XDR (Extended Detection and Response) ra đời như một giải pháp tiên tiến để giải quyết những thách thức này. XDR tích hợp và tự động hóa việc thu thập, phân tích và phản ứng với các dữ liệu từ nhiều nguồn khác nhau, bao gồm endpoint, mạng và ứng dụng đám mây. Không chỉ giúp phát hiện các mối đe dọa nhanh hơn, XDR còn cung cấp cái nhìn toàn diện và tập trung, giảm thiểu sự rời rạc giữa các công cụ bảo mật. Bằng cách tối ưu hóa quy trình phản ứng và tăng cường khả năng phối hợp, XDR mang lại lợi ích vượt trội trong việc bảo vệ doanh nghiệp trước các mối nguy ngày càng phức tạp.

1.2 Giải pháp

1.2.1 Extended Detection and Response (XDR)

XDR (Extended Detection and Response) là một giải pháp bảo mật tiên tiến được thiết kế để phát hiện, phân tích và phản ứng với các mối đe dọa an ninh mạng một cách toàn

diện và hiệu quả. XDR tích hợp dữ liệu từ nhiều nguồn khác nhau, bao gồm endpoint, email, mạng, ứng dụng đám mây và những môi trường khác trong hệ sinh thái số của doanh nghiệp.

XDR hoạt động như một nền tảng hợp nhất, thu thập và phân tích dữ liệu từ các nguồn này để cung cấp cái nhìn toàn diện về toàn bộ chuỗi tấn công. XDR có khả năng phát hiện những mối đe dọa phức tạp, khó phát hiện mà các giải pháp riêng lẻ có thể bỏ sót. Đồng thời, nó giúp tự động hóa việc phản hồi và xử lý các sự cố, từ đó tiết kiệm thời gian và giảm thiểu rủi ro cho doanh nghiệp.

Tính năng nổi bật của giải pháp XDR:

- Tích hợp dữ liệu từ nhiều nguồn để cung cấp một cái nhìn đồng bộ về các mối đe dọa trên toàn bộ hệ thống.
- Phát hiện các mối đe dọa phức tạp và tinh vi nhờ tích hợp AI và khả năng phân tích dữ liệu lớn.
- Tự động hóa phản ứng, tăng tốc quá trình ứng phó sự cố, giám phụ thuộc vào can thiệp thủ công
- Giảm sự rò rỉ rắc giữa các giải pháp bảo mật riêng lẻ, giúp đơn giản hóa quản lý an ninh mạng.

1.2.2 Công nghệ Wazuh

Wazuh là một nền tảng bảo mật mã nguồn mở được thiết kế để cung cấp khả năng giám sát, phát hiện mối đe dọa và quản lý tuân thủ. Wazuh được sử dụng rộng rãi như một giải pháp bảo mật hợp nhất chức năng của SIEM và XDR.

Tính năng nổi bật của Wazuh-XDR bao gồm:

- **Phát hiện xâm nhập:** Sử dụng các quy tắc bảo mật để phát hiện các hành vi đáng ngờ trên endpoint, ứng dụng, và network.
- **Giám sát điểm cuối theo thời gian thực:** Giám sát tính toán vẹn tệp tin (FIM), phát hiện các hành vi bất thường trên endpoints.
- **Tích hợp và mở rộng:** Tích hợp với các giải pháp, công cụ bảo mật khác nhau để tăng khả năng phát hiện và phản ứng.
- **Phân tích log tập trung:** Thu thập log từ nhiều nguồn khác nhau và giám sát chúng trên một bảng điều khiển tập trung.
- **Tự động hóa phản ứng sự cố:** Cung cấp khả năng tự động hóa quy trình phản ứng như là cô lập endpoint bị nhiễm mã độc, chặn traffic mạng đáng ngờ, hoặc gửi cảnh báo đến đội ngũ bảo mật.
- **Quản lý tuân thủ:** Hỗ trợ các tiêu chuẩn bảo mật và pháp lý như PCI DSS, GDPR, HIPAA, ISO 27001, giúp doanh nghiệp đảm bảo tuân thủ các quy định.

1.3 Mục tiêu và phạm vi nghiên cứu

1.3.1 Phạm vi

Mô phỏng mô hình mạng doanh nghiệp. Triển khai giải pháp Wazuh XDR tích hợp với các công nghệ bảo mật khác, bao gồm Snort, Suricata, ClamAV, Mod Security, DVWA, ELK stack và VirusTotal.

1.3.2 Mục tiêu

Mục tiêu của nhóm chúng em là triển khai giải pháp XDR với Wazuh nhằm tạo hệ thống giám sát và theo dõi hoạt động trên các endpoints theo thời gian thực, phát hiện và phản ứng tự động với các sự cố trên mô hình mạng đang được bảo vệ.

CHƯƠNG 2: CÔNG NGHỆ HỖ TRỢ

Ngoài Wazuh đã được giới thiệu ở trên, dưới đây là các công nghệ được tích hợp vào hệ thống để tạo hệ sinh thái bảo mật mạnh mẽ, nâng cao khả năng phát hiện và ứng phó với các mối đe dọa tiềm tàng.

2.1 Mod Security

Mod Security là một công cụ WAF mạnh mẽ, giúp bảo vệ các ứng dụng web khỏi các cuộc tấn công và các mối đe dọa bảo mật, đồng thời cung cấp khả năng giám sát và ghi log hoạt động.

2.2 DVWA

Damn Vulnerable Web Application là một ứng dụng web có chứa lỗ hổng bảo mật, được sử dụng để thực hành mô phỏng các loại tấn công web phổ biến, giúp nâng cao kỹ năng bảo mật ứng dụng web.

2.3 Snort

Snort là một hệ thống phát hiện và ngăn chặn xâm nhập mã nguồn mở, hỗ trợ nhiều giao thức, dễ dàng mở rộng và có cộng đồng người dùng lớn. Snort nhận diện các hoạt động bất thường trên mạng thông qua việc so sánh với các quy tắc đã được định nghĩa.

2.4 Suricata

Suricata là một IDS/IPS được thiết kế để hoạt động hiệu quả trên phần cứng hiện đại và có khả năng xử lý đa luồng. Tương tự như Snort, Suricata sử dụng các quy tắc để phân tích lưu lượng mạng. Ngoài ra, Suricata cũng tích hợp các khả năng như phân tích HTTP, TLS, và các giao thức khác.

2.5 ELK stack

Bộ công cụ bao gồm Elasticsearch, Logstash và Kibana, có khả năng thu thập, chuyển đổi, lưu trữ, tìm kiếm và phân tích dữ liệu log.

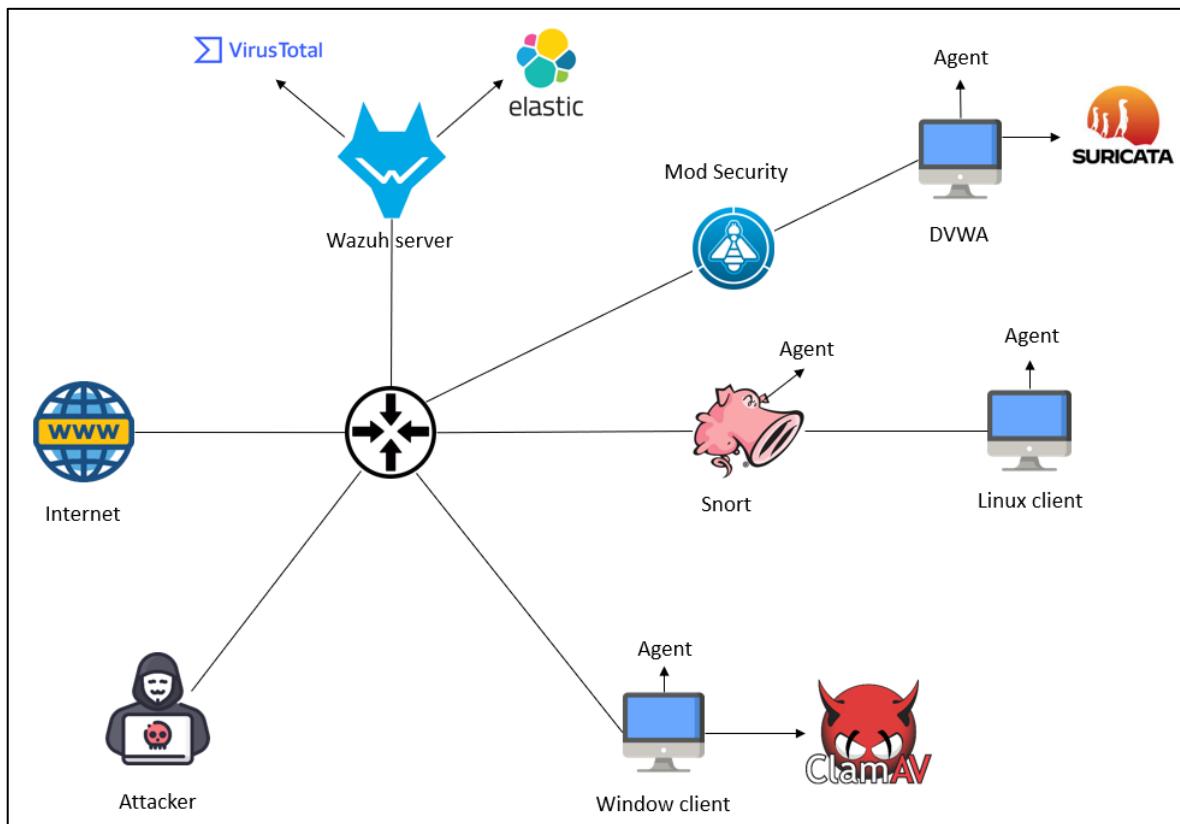
2.6 ClamAV

Phần mềm anti-virus mã nguồn mở chạy đa nền tảng. Mục đích chính của ClamAV là phát hiện các loại phần mềm độc hại khác nhau như là virus, worms, trojans, rootkits.

2.7 VirusTotal

Công cụ trực tuyến phân tích các tệp và URL đáng ngờ để phát hiện các loại phần mềm độc hại và nội dung độc hại bằng cách sử dụng các anti-virus engine và website scanners.

CHƯƠNG 3: KIẾN TRÚC



Hình 3.1 Mô hình thực nghiệm

Mô hình gồm 7 máy ảo mô phỏng môi trường mạng doanh nghiệp, bao gồm:

Ba máy endpoints cần bảo vệ gồm máy window Client (win10), Linux client (Metasploitable2) và máy DVWA chứa web có lỗ hổng. Mỗi máy được cài đặt Wazuh agent để giám sát hoạt động và phát hiện bất thường.

Máy Snort được cấu hình ở mode inline để phát hiện và ngăn chặn tấn công đến Linux client.

Máy Router để NAT các máy trong mạng nội bộ ra Internet

Máy Attacker để mô phỏng tấn công.

ClamAV cài đặt trên máy Window Client để quét virus.

Suricata cài đặt trên máy DVWA hoạt động ở passive mode, giám sát các xâm nhập và cảnh báo khi có tấn công.

Mod Security cài đặt trên máy DVWA để ngăn chặn các cuộc tấn công web.

Máy Wazuh server nhận log được thu thập từ các agent và hiển thị tập trung trên dashboard. Ngoài ra, trên Wazuh server còn tích hợp VirusTotal và ELK stack.

CHƯƠNG 4: NGHIÊN CỨU THỰC NGHIỆM

4.1 Kịch bản 1: Tương tác giữa ModSecurity và Wazuh trong ngăn chặn tấn công mạng

4.1.1 Mô tả

Mô tả chi tiết:

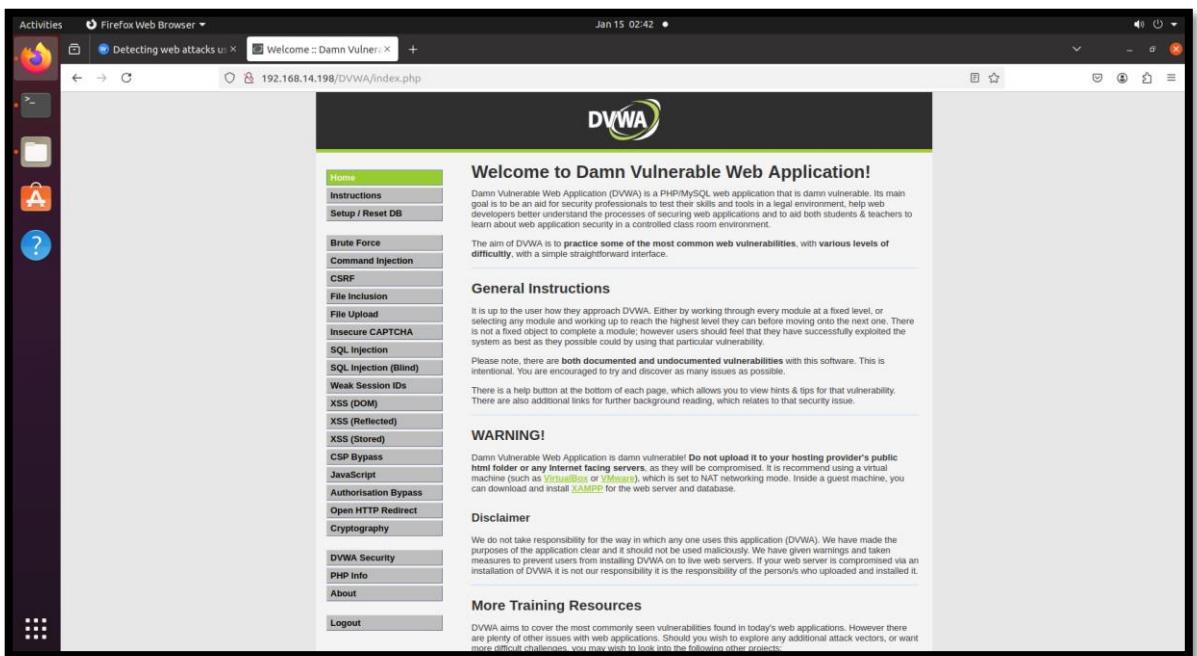
- Cấu hình ModSecurity trên máy chủ web bảo vệ ứng dụng DVWA.
- Thực hiện các cuộc tấn công vào DVWA từ một máy tấn công giả lập.
- ModSecurity ngăn chặn tấn công, ghi log sự kiện và chuyển tiếp log đến Wazuh server.
- Trên Wazuh dashboard, log từ ModSecurity được hiển thị rõ ràng bao gồm thông tin về nguồn gốc và loại tấn công

4.1.2 Thực hiện

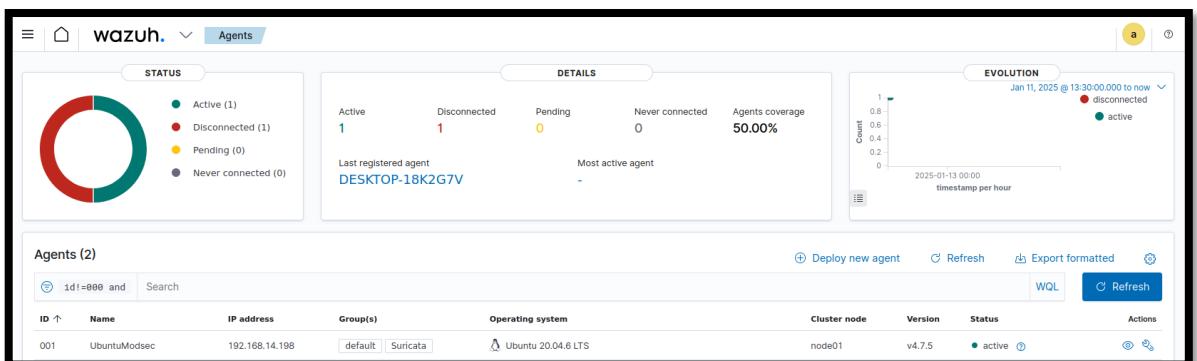
Mục tiêu: Đánh giá khả năng phát hiện và ghi nhận log tấn công từ ModSecurity, cũng như việc gửi log đến Wazuh Server để hiển thị trên dashboard

Chi tiết thực hiện:

- Đầu tiên ta cấu hình ModSecurity, DVWA và Wazuh Agent lên máy endpoint Ubuntu



Hình 4.1 Cài đặt DVWA và ModSecurity trên máy endpoint Ubuntu



Hình 4.2 Cài đặt Wazuh Agent trên máy endpoint Ubuntu

- Cài đặt các rule cấu hình trên ModSecurity để ngăn chặn các cuộc tấn công như SQL Injection, XSS..

```
GNU nano 4.8                               /etc/modsecurity/sqlinjection.conf
SecRule ARGS "@rx (?i)(select|s+.*\s+from|union\s+select|insert\s+into|update\s+.*\$set|delete\s+from|or\s+[\'\"]d+=d[\'\"]|[\\'\'].*\['\"]\$s=\$s[\'\'].*\['\"])" \
"id:1000001,"                                \
phase:2, \
t:none, \
block, \
msg:'SQL Injection attack detected.', \
severity:'CRITICAL', \
log, \
tag:'application-multi', \
tag:'attack-sql'" \
SecRule ARGS "@rx (?i)<script>|<img onerror=|javascript:|alert(|<svg|<iframe|document|.cookie)" \
"id:1000003,"                                \
phase:2, \
t:none, \
block, \
msg:'XSS attack detected.', \
severity:'CRITICAL', \
log, \
tag:'application-multi', \
tag:'attack-xss'"
```

Hình 4.3 Cấu hình rule ModSecurity

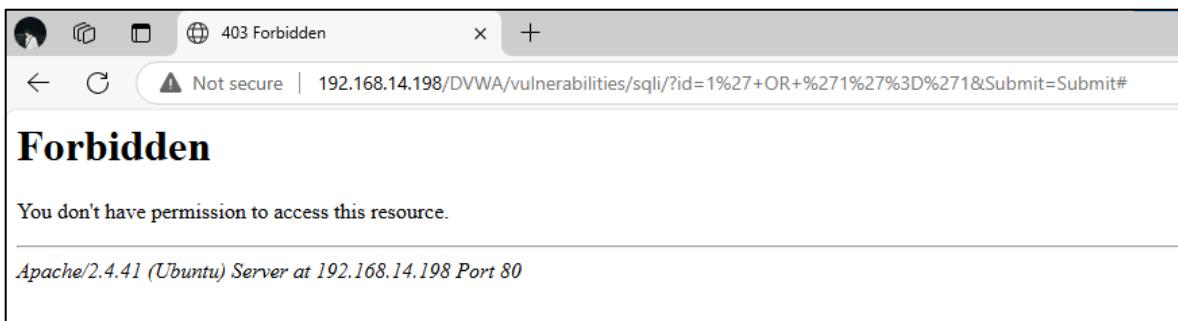
- Rule cấu hình:

```
SecRule ARGS "@rx\n\n(?i)(select|s+.*|s+from|union|s+select|insert|s+into|update|s+.*|s+set|delete|s+from|or|s+["\n"]|d+=|d+["\"]|[\""].*[""]|s*=|s*[""].*[""])" \n\n    "id:1000001,\n    phase:2,\n    t:none,\n    block,\n    msg:'SQL Injection attack detected.',\n    severity:'CRITICAL',\n    log,\n    tag:'application-multi',\n    tag:'attack-sqli'"\n\nSecRule ARGS "@rx\n\n(?i)<script>|<img|onerror=|javascript:|alert(|<svg|<iframe|document|.cookie)" \n\n    "id:1000003,\n    phase:2,\n    t:none,\n    block,\n    msg:'XSS attack detected.',\n    severity:'CRITICAL',\n    log,\n    tag:'application-multi',\n    tag:'attack-xss'"
```

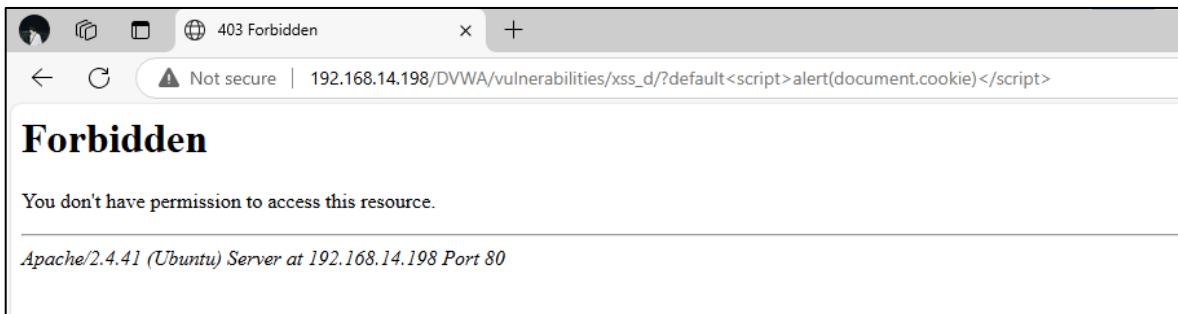
- **Rule 1: Phát hiện SQL Injection**

- SecRule ARGS: Rule này áp dụng cho tất cả các tham số (ARGS) trong request, bao gồm query string, body, hoặc các dữ liệu được gửi tới server.

- @rx: Sử dụng biểu thức chính quy (regular expression) để dò tìm các mẫu ký tự nghi vấn.
 - (?i): Biểu thức chính quy không phân biệt chữ hoa/thường.
 - block: Chặn request nếu khớp với rule.
 - msg: Ghi lại thông báo "SQL Injection attack detected." vào log.
 - severity: Mức độ nghiêm trọng là "CRITICAL".
 - tag: Đánh dấu rule liên quan đến tấn công SQL Injection (attack-sqli).
 - **Rule 2: Phát hiện XSS (Cross-Site Scripting)**
 - SecRule ARGS: Áp dụng cho tất cả các tham số trong request.
 - @rx: Dò tìm các mẫu ký tự nghi vấn thông qua biểu thức chính quy.
 - (?i): Không phân biệt chữ hoa/thường.
 - Các thẻ hoặc thuộc tính HTML/JS thường được dùng trong tấn công XSS:
 - <script>: Thẻ script để nhúng mã JavaScript.
 - : Thẻ hình ảnh, có thể bị lợi dụng để chèn mã độc với thuộc tính onerror.
 - onerror=: Sự kiện lỗi trong HTML/JS, thường được dùng để thực thi mã độc.
 - javascript::: Gọi trực tiếp mã JavaScript.
 - alert(: Hàm JavaScript thường được dùng để kiểm tra tấn công.
 - <svg>: Thẻ SVG có thể bị lợi dụng để nhúng mã độc.
 - <iframe>: Thẻ nhúng nội dung từ trang khác, có thể dùng để đánh cắp thông tin.
 - document.cookie: Dữ liệu cookie của người dùng, mục tiêu phổ biến trong XSS.
 - block: Chặn request nếu khớp với rule.
 - msg: Ghi lại thông báo "XSS attack detected." vào log.
 - severity: Mức độ nghiêm trọng là "CRITICAL".
 - tag: Đánh dấu rule liên quan đến tấn công XSS (attack-xss).
-
- Sử dụng máy attacker để thực hiện cuộc tấn công vào DVWA



Hình 4.4 Attacker thực hiện tấn công SQLi



Hình 4.5 Attacker thực hiện tấn công XSS

- ModSecurity chặn tấn công và ghi nhận log trong tệp nhật ký

```
kenzy@ubuntu:~$ tail -f /var/log/apache2/modsec_audit.log
Apache-Handler: application/x-httdp-php
Stopwatch: 1736940938554746 39123 (- -)
Stopwatch2: 1736940938554746 39123; combined=4646, p1=1343, p2=2019, p3=133, p4=891, p5=251, sr=505, sw=9, l=0, gc=0
Response-Body-Transformed: Dechunked
Producer: ModSecurity for Apache/2.9.3 (http://www.modsecurity.org/); OWASP CRS/3.2.0.
Server: Apache/2.4.41 (Ubuntu)
Engine-Mode: "ENABLED"

--afe9813d-Z--

--44d54a73-A-
[15/Jan/2025:03:38:33 --0800] Z4eeORfrtP8Vf-boCstanQAAAAM 192.168.14.1 60393 192.168.14.198 80
--44d54a73-B-
GET /DVWA/vulnerabilities/sqli/?id=1%27+OR+%271%27%3D%271&Submit=Submit HTTP/1.1
Host: 192.168.14.198
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.14.198/DVWA/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=snttl7ghns0vl9pitpc5t3rrs; security=low

--44d54a73-F-
HTTP/1.1 403 Forbidden
Content-Length: 279
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

--44d54a73-E--
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at 192.168.14.198 Port 80</address>
</body></html>
```

Hình 4.6 Log ModSecurity ghi nhận sự kiện tấn công

- Wazuh Agent trên máy endpoint đẩy log của ModSecurity lên Wazuh Server. Truy cập dashboard Wazuh để xem chi tiết log (loại tấn công, hành động của ModSecurity)

t	agent.ip	192.168.14.198
t	agent.name	UbuntuModsec
t	data.id	403
t	data.protocol	GET
t	data.srcip	192.168.14.198
t	data.url	/DVWA/vulnerabilities/sqli/?id=%27+OR+%271%27%3D%271%27+-&Submit=Submit
t	decoder.name	web-accesslog
t	full_log	192.168.14.198 - - [09/Jan/2025:00:11:39 -0800] "GET /DVWA/vulnerabilities/sqli/?id=%27+OR+%271%27%3D%271%27+-&Submit=Submit HTTP/1.1" 403 496 "http://192.168.14.198/DVWA/vulnerabilities/sqli/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:133.0) Gecko/20100101 Firefox/133.0"
t	id	1736410301.1520420
t	input.type	log
t	location	/var/log/apache2/access.log
t	manager.name	kenzy-virtual-machine
t	rule.description	SQL injection attempt.

Hình 4.7 Dashboard ghi nhận chi tiết log tấn công SQLi

t	agent.ip	192.168.14.198
t	agent.name	UbuntuModsec
t	data.id	403
t	data.protocol	GET
t	data.srcip	192.168.14.1
t	data.url	/DVWA/vulnerabilities/xss_d/?default=%3Cscript%3Ealert(document.cookie)%3C/script%3E
t	decoder.name	web-accesslog
t	full_log	192.168.14.1 - - [10/Jan/2025:07:45:43 -0800] "GET /DVWA/vulnerabilities/xss_d/?default=%3Cscript%3Ealert(document.cookie)%3C/script%3E HTTP/1.1" 403 496 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.4895.152 Safari/537.36"
t	id	1736523945.1776610
t	input.type	log
t	location	/var/log/apache2/access.log
t	manager.name	kenzy-virtual-machine
t	rule.description	XSS (Cross Site Scripting) attempt.

Hình 4.8 Dashboard ghi nhận chi tiết log tấn công XSS

Kết quả mong đợi:

- Tấn công bị chặn thành công

- Log được ghi nhận và hiển thị chi tiết trên Wazuh dashboard

4.2 Kịch bản 2: Kiểm tra tương tác giữa Snort và Wazuh

4.2.1 Mô tả:

- Cấu hình Snort ở chế độ inline để phát hiện và ngăn chặn tấn công đến máy Linux Client
- Máy Attacker mô phỏng tấn công Linux Client
- Viết rule để ngăn chặn tấn công và kiểm tra kết quả thông qua quan sát Snort log gửi đến dashboard của Wazuh.

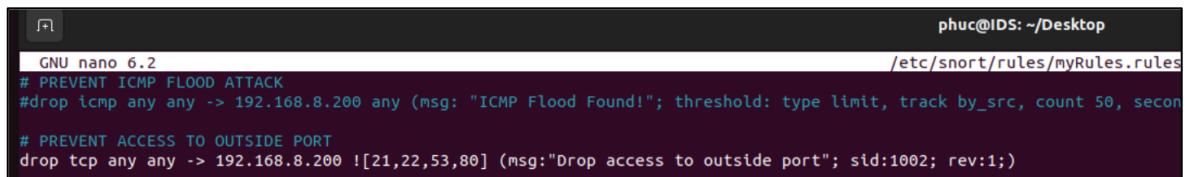
4.2.2 Thực hiện

Mục tiêu:

Đánh giá khả năng phát hiện và ngăn chặn tấn công của Snort, đồng thời kiểm tra hoạt động đầy log về Wazuh Server.

Chi tiết thực hiện:

Viết rule để chỉ cho kết nối đến máy Linux client trong dải port [21,22,53,80]



```

phuc@IDS: ~/Desktop
GNU nano 6.2
/etc/snort/rules/myRules.rules
# PREVENT ICMP FLOOD ATTACK
#drop icmp any any -> 192.168.8.200 any (msg: "ICMP Flood Found!"; threshold: type limit, track by_src, count 50, second 60)

# PREVENT ACCESS TO OUTSIDE PORT
drop tcp any any -> 192.168.8.200 ![21,22,53,80] (msg:"Drop access to outside port"; sid:1002; rev:1;)

```

Hình 4.9 rule cấu hình chặn kết nối ngoài port chỉ định

drop tcp any any -> 192.168.8.200 ![21,22,53,80] (msg:"Drop access to outside port"; sid:1002; rev:1;)

Rule này sẽ drop tất cả các gói tin tcp gửi 192.168.8.200 nếu kết nối đó sử dụng port ngoài dải port chỉ định

Attacker có thể kết nối đến Linux client thông qua port 22 nhưng sẽ bị chặn nếu dùng port 5000

```

└─(phuc@phuc)-[~/Desktop]
└─$ ping 192.168.8.200
PING 192.168.8.200 (192.168.8.200) 56(84) bytes of data.
64 bytes from 192.168.8.200: icmp_seq=1 ttl=63 time=4.75 ms
64 bytes from 192.168.8.200: icmp_seq=2 ttl=63 time=6.95 ms
64 bytes from 192.168.8.200: icmp_seq=3 ttl=63 time=2.94 ms
^C
--- 192.168.8.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.940/4.879/6.953/1.641 ms

└─(phuc@phuc)-[~/Desktop]
└─$ sudo telnet 192.168.8.200 5000
Trying 192.168.8.200 ...
^C

└─(phuc@phuc)-[~/Desktop]
└─$ sudo telnet 192.168.8.200 22
Trying 192.168.8.200 ...
Connected to 192.168.8.200.
Escape character is '^]'.
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
^C
Connection closed by foreign host.

```

Hình 4.10 Khi này cảnh báo về kết nối trái phép sẽ được gửi đến Wazuh và hiển thị ở dashboard

t full_log	[**] [1:1002:1] Drop access to outside port [**]
t id	1736157123.6355
t input.type	log
t location	/var/log/snort/alert
t manager.name	wazuh
t rule.description	IDS event.
# rule.firedtimes	5
t rule.groups	ids

Hình 4.11 Thông tin chi tiết về cảnh cáo lưu tại /var/log/snort/alert cho thông tin về thời gian xảy ra sự cố, địa chỉ attacker và port sử dụng.

```

[**] [1:1002:1] Drop access to outside port [**]
[Priority: 0]
01/06-16:48:07.880273 10.81.7.100:51748 -> 192.168.8.200:5000
TCP TTL:63 TOS:0x0 ID:51341 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xD391ACBB Ack: 0x0 Win: 0x7D78 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1969325585 0 NOP WS: 7

[**] [1:1002:1] Drop access to outside port [**]
[Priority: 0]
01/06-16:48:08.903361 10.81.7.100:51748 -> 192.168.8.200:5000
TCP TTL:63 TOS:0x0 ID:51342 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xD391ACBB Ack: 0x0 Win: 0x7D78 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1969326609 0 NOP WS: 7

```

Hình 4.11 Kiểm tra lại trên một trường hợp khác, lần này ta viết rule để ngăn chặn ICMP flood attack

```

root@192:~/home/phuc/Desktop
GNU nano 6.2
/etc/snort/rules/DACN.rules
# PREVENT ICMP FLOOD ATTACK
drop icmp any any -> 192.168.8.200 any (msg:"ICMP Flood Attack Found!"; detection_filter:track by_src, count 100, seconds 5; sid:1000001; rev:1;)

```

Hình 4.12 Rule chặn ICMP flood attack

```

drop icmp any any -> 192.168.8.200 any (msg: "ICMP Flood Found!";
detection_filter:track by_src, count 100, seconds 5; sid: 1000001; rev:1;)

```

Rule này sẽ drop gói tin ICMP từ bất kỳ nguồn nào nếu nguồn đó gửi gói tin quá ngưỡng 100 gói/5s đến địa chỉ 192.168.8.200

Attacker vẫn có thể ping đến Linux client bình thường

```

root@phuc:~/.home/phuc/Desktop]
# ping 192.168.8.200
PING 192.168.8.200 (192.168.8.200) 56(84) bytes of data.
64 bytes from 192.168.8.200: icmp_seq=1 ttl=63 time=2.34 ms
64 bytes from 192.168.8.200: icmp_seq=2 ttl=63 time=2.34 ms
64 bytes from 192.168.8.200: icmp_seq=3 ttl=63 time=2.01 ms
^C
--- 192.168.8.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.006/2.228/2.340/0.157 ms

```

Hình 4.13 Attacker ping victim

```
(root@phuc)-[~/home/phuc/Desktop]
# sudo hping3 -1 --flood 192.168.8.200 -d 655495
HPING 192.168.8.200 (eth1 192.168.8.200): icmp mode set, 28 headers + 135 data bytes
hping in flood mode, no replies will be shown
^C
-- 192.168.8.200 hping statistic --
111813 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Hình 4.14 Attack dùng hping3 để tấn công ICMP flood attack

Tấn công bị chặn và gửi cảnh cáo về Wazuh dashboard

The screenshot shows a Wazuh dashboard interface with the 'Security events' tab selected. A single event is listed:

Field	Value
_index	wazuh-alerts-4.x-2025.01.10
agent.id	004
agent.ip	10.11.13.110
agent.name	IDS_agent
decoder.name	snort
decoder.parent	snort
full_log	[**] [1:1000001:1] ICMP Flood Attack Found! [**]
id	1736510363.1236669
input.type	log
location	/var/log/snort/alert
manager.name	wazuh
rule.description	IDS event.

Hình 4.15 Log chặn ICMP flood attack từ Wazuh

Xem chi tiết cảnh báo từ Snort, ta tìm được IP kẻ tấn công và kích cỡ payload của gói tin

```
[**] [1:1000001:1] ICMP Flood Attack Found! [**]
[Priority: 0]
01/10-18:57:43.217450 10.81.7.100 -> 192.168.8.200
ICMP TTL:63 TOS:0x0 ID:59175 IpLen:20 DgmLen:163
Type:8 Code:0 ID:44102 Seq:26112 ECHO

[**] [1:1000001:1] ICMP Flood Attack Found! [**]
[Priority: 0]
01/10-18:57:43.219243 10.81.7.100 -> 192.168.8.200
ICMP TTL:63 TOS:0x0 ID:55620 IpLen:20 DgmLen:163
Type:8 Code:0 ID:44102 Seq:26368 ECHO
```

4.3 Kịch bản 3: Giám sát command thực thi trên Linux

4.3.1 Mô tả:

- Sử dụng công cụ auditd thu thập, theo dõi tất cả Linux command được người dùng thực hiện trên máy Snort.
- Cấu hình để Wazuh cảnh báo đặt biệt khi người dùng thực thi các command nguy hiểm.
- Phát hiện leo thang đặt quyền khi khác thác lõi hỏng trên vim để thực thi command với quyền root

4.3.2 Thực hiện

Mục tiêu:

Giám sát toàn bộ Linux command chạy trên endpoint.

Chi tiết thực hiện:

Rule auditd để thu thập Linux command

```
root@IDS:/home/phuc/Desktop# auditctl -l
-a always,exit -F arch=b64 -S execve -F key=audit-wazuh-c
-a always,exit -F arch=b32 -S execve -F key=audit-wazuh-c
```

Câu hình ossec.conf để thu thập log từ auditd

```
#Log auditd
<localfile>
  <log_format>audit</log_format>
  <location>/var/log/audit/audit.log</location>
</localfile>
```

Hình 4.16 Câu hình thu thập log audit4.164.17c4.18

Thực hiện một số command như whoami, ping, cat

```

phuc@IDS:~/Desktop$ whoami
phuc
phuc@IDS:~/Desktop$ ping google.com
PING google.com (142.250.199.206) 56
64 bytes from nchkbg-am-in-f14.1e100
4 ms
64 bytes from nchkbg-am-in-f14.1e100
8 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0%
rtt min/avg/max/mdev = 29.813/30.607/
phuc@IDS:~/Desktop$ cat scripts.txt

```

Hình 4.Thực hiện các lệnh đơn giản4.17

Không chỉ thu thập command do người dùng nhập mà còn thu thập command mà hệ thống chạy ngầm, OS phát sinh hay system healthcheck.

Time	rule.description
Jan 11, 2025 @ 01:03:49.585	Audit: Command: /usr/bin/cat.
Jan 11, 2025 @ 01:03:42.496	Audit: Command: /usr/bin/ping.
Jan 11, 2025 @ 01:03:37.577	Audit: Command: /usr/bin/grep.
Jan 11, 2025 @ 01:03:37.573	Audit: Command: /usr/bin/grep.
Jan 11, 2025 @ 01:03:37.570	Audit: Command: /usr/bin/netstat.
Jan 11, 2025 @ 01:03:37.569	Audit: Command: /usr/bin/dash.
Jan 11, 2025 @ 01:03:37.566	Audit: Command: /usr/bin/whoami.

Hình 4. Danh sách command thực hiện4.18

Tạo danh sách các command nguy hiểm

```
root@wazuh: /home/wazuh/Desktop
GNU nano 6.2      /var/ossec/etc/lists/suspicious-programs *
ncat:yellow
ps aux:yellow
nc:red
rm:red
tcpdump:orange
```

Câu hình rule để Wazuh tự động cảnh cáo khi người dùng thực thi các command trong danh sách trên

```
<!-- Suspicious command -->
<group name="audit">
  <rule id="100210" level="12">
    <if_sid>80792</if_sid>
    <list field="audit.command" lookup="match_key_value" check_value="red">etc/lists/suspicious-programs</list>
    <description>Audit: Highly Suspicious Command executed: ${audit.exe}</description>
    <group>audit_command,</group>
  </rule>
</group>
```

Thử thực hiện lệnh: nc -l 5555

```
phuc@IDS: ~/Desktop
phuc@IDS:~/Desktop$ nc -l 5555
^C
phuc@IDS:~/Desktop$
```

Wazuh dashboard xuất hiện cảnh cáo đặc biệt

Time	rule.description
Jan 11, 2025 @ 15:23:40.218	Audit: Highly Suspicious Command executed: /usr/bin/nc
Jan 11, 2025 @ 15:22:22.129	Audit: Command: /usr/bin/clear.
Jan 11, 2025 @ 15:22:18.126	Audit: Command: /usr/bin/cat.
Jan 11, 2025 @ 15:22:16.153	Audit: Command: /usr/bin/last.

Câu lệnh chi tiết cũng được ghi lại

```
t data.audit.cwd      /home/phuc/Desktop  
t data.audit.egid     1000  
t data.audit.euid     1000  
t data.audit.exe      /usr/bin/nc  
t data.audit.execve.a0 nc  
t data.audit.execve.a1 -l  
t data.audit.execve.a2 5555
```

Thử một trường hợp khác. Ở đây ta có user phuc không được dùng command useradd

```
phuc@IDS:~/Desktop$ useradd  
bash: /usr/sbin/useradd: Permission denied  
phuc@IDS:~/Desktop$ sudo useradd  
Sorry, user phuc is not allowed to execute '/usr/sbin/useradd' as root on IDS.  
phuc@IDS:~/Desktop$
```

Lợi dụng lỗ hổng của trình soạn thảo vim để leo quyền root và thực thi lệnh useradd để tạo user attacker2

```
phuc@IDS:~/Desktop$ whoami  
phuc  
phuc@IDS:~/Desktop$ sudo vim -c ':!/bin/sh'  
[sudo] password for phuc:  
  
^[[I#  
/bin/sh: 1:      : not found  
# whoami  
root  
# useradd attacker2  
# passwd attacker2  
New password:  
BAD PASSWORD: The password is a palindrome  
Retype new password:  
passwd: password updated successfully  
#
```

Thông tin về việc leo quyền này đã được Wazuh ghi log lại và gửi cảnh cáo

Jan 11, 2025 @ 02:12:39.725	T1136	Persistence	New user added to the system.
Jan 11, 2025 @ 02:12:07.672	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.
Jan 11, 2025 @ 02:12:07.670	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.

Lệnh để thực thi leo quyền cũng được ghi log chi tiết

t agent.ip	10.11.13.110
t agent.name	IDS_agent
t data.command	/usr/bin/vim -c :!/bin/sh
t data.dstuser	root
t data.pwd	/home/phuc/Desktop
t data.srcuser	phuc

4.4 Kịch bản 4: Giám sát tính toàn vẹn tệp tin (FIM)

4.4.1 Mô tả:

Sử dụng module FIM của Wazuh để giám sát thư mục download trên máy Windows Client. Tất cả thay đổi của các file trong thư mục này đều sẽ được ghi log lại

4.4.2 Thực hiện

Mục tiêu:

Giám sát toàn bộ thay đổi diễn ra trong tệp tin và thư mục được quản lý

Chi tiết thực hiện:

Cấu hình file ossec.conf để giám sát thư mục Download trên máy Window client

```
</synchronization>  
<directories whodata="yes" report_changes="yes" check_all="yes" realtime="yes">C:/Users/Phuc/Downloads</directories>  
</syscheck>
```

Thực hiện các thay đổi trong thư mục Downloads

```
PS C:\Users\Phuc\Downloads> ls  
  
Directory: C:\Users\Phuc\Downloads  
  
Mode LastWriteTime Length Name  
---- - - - -  
-a--- 1/11/2025 12:07 PM 34 test1.txt  
  
PS C:\Users\Phuc\Downloads> echo AAAAAA > test2.txt  
PS C:\Users\Phuc\Downloads> ls  
  
Directory: C:\Users\Phuc\Downloads  
  
Mode LastWriteTime Length Name  
---- - - - -  
-a--- 1/11/2025 12:07 PM 34 test1.txt  
-a--- 1/11/2025 12:16 PM 16 test2.txt  
  
PS C:\Users\Phuc\Downloads> cat test1.txt  
Pham Hoang Phuc - 1111111111111111  
PS C:\Users\Phuc\Downloads> cat test1.txt  
Pham Hoang Phuc - 21521295  
PS C:\Users\Phuc\Downloads> rm test2.txt  
PS C:\Users\Phuc\Downloads>
```

Hình 4.19 Kết quả hiển thị trên dashboard của Wazuh

syscheck.path	syscheck.event	rule.description
c:\users\phuc\downloads\test2.txt	deleted	File deleted.
c:\users\phuc\downloads\test1.txt	modified	Integrity checksum changed.
c:\users\phuc\downloads\test2.txt	added	File added to the system.

Hình 4.20 Chi tiết của hành động thay đổi nội dung file

```
syscheck_integrity_changed

    ✓ File 'c:\users\phuc\downloads\test1.txt' modified
        Mode: realtime
        Changed attributes: size,mtime,md5,sha1,sha256
        Size changed from '34' to '26'
        Old modification time was: '1736572079', now it is '1736572597'
        Old md5sum was: 'b326c280dc130fd773416a02ff99defc'
        New md5sum is : '243e8b07c0c0f8f0e77e280898e0cb36'
        Old sha1sum was: '4e487cb02aa939f082f808d2e7704575cd8e9034'
        New sha1sum is : '1e6b8bcc088491f348fee75ec5761e8c6e772ca7'
        Old sha256sum was: '9b98d15612957165bdfad398c48c2dd2f132d3f6371d9712df5e4062415069ee'
        New sha256sum is : '65cd7c791da45bbcc14f6ab5fce61eedab4da3b7dad225805ae63e04267eb7c7'
```

4.5 Kịch bản 5: Tự động xóa file độc hại và xây dựng IOCs cho threat intelligence

4.5.1 Mô tả:

Dùng ClamAV để scan file tải về trong thư mục Downloads trên máy Window client. Nếu ClamAV không phát hiện được bất thường thì sẽ đẩy hash lên Wazuh server. Tích hợp VirusTotal trên Wazuh Server sẽ scan hash của file được tải về trong thư mục Downloads. Nếu Wazuh xác nhận đây là file độc hại thì Wazuh agent trên máy Window client sẽ kích hoạt script xóa file tự động để xóa file độc hại. Đồng thời hash của mã độc sẽ được lưu lại để cải thiện database của ClamAV.

4.5.2 Thực hiện:

Mục tiêu:

Phát hiện, xóa file độc hại tự động và lưu hash làm IOCs

Chi tiết thực hiện:

Cấu hình tích hợp VirusTotal trên Wazuh server

```

## VIRUSTOTAL ####
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>8b1dfc59e467369ced6e1798465c608bbddf0d1d9cac028305ea6de38d44eb4e</api_key>
    <group>syscheck</group>
    <rule_id>554,550</rule_id>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>

```

Hình 4.21 Tải về một mã độc từ database của malware bazaar và dùng ClamAV để scan.

Name	Date modified	Type	Size
cr.dll	12/26/2024 11:39 PM	Application exten...	643 KB
ronwod	12/26/2024 11:39 PM	Application	28 KB

ClamAV không phát hiện được mã độc

```

C:\Sample\malware>clamscan.exe -r 694428711a5a7194da04f29d442b2d749c12b771c8c0c4684fcb97a64c9e5ea
Loading: 53s, ETA: 0s [=====] 8.70M/8.70M sigs
Compiling: 9s, ETA: 0s [=====] 41/41 tasks

C:\Sample\malware\694428711a5a7194da04f29d442b2d749c12b771c8c0c4684fcb97a64c9e5ea\cr.dll: OK
C:\Sample\malware\694428711a5a7194da04f29d442b2d749c12b771c8c0c4684fcb97a64c9e5ea\ronwod.exe: OK

----- SCAN SUMMARY -----
Known viruses: 8703433
Engine version: 1.4.1
Scanned directories: 1
Scanned files: 2
Infected files: 0
Data scanned: 0.69 MB
Data read: 0.65 MB (ratio 1.06:1)
Time: 66.375 sec (1 m 6 s)
Start Date: 2024:12:30 19:45:28
End Date: 2024:12:30 19:46:34

```

Đưa file vào thư mục Downloads, khi này file sẽ bị xóa tự động

Script python xóa file tự động

```

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

import os
import sys
import json
import datetime

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

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

OS_SUCCESS = 0
OS_INVALID = -1

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

    def write_debug_file(ar_name, msg):
        with open(LOG_FILE, mode="a") as log_file:
            log_file.write(str(datetime.datetime.now().strftime("%Y/%m/%d %H:%M:%S")) + " " + ar_name + ": " + msg + "\n")

    def setup_and_check_message(argv):
        # get alert from stdin
        input_str = ""
        for line in sys.stdin:
            input_str = line

```

Hình 4.22 Wazuh cảnh báo file là độc hại và xóa file thành công

Dec 30, 2024 @ 19:46:59.027 active-response/bin/remove-threat.exe removed threat located at c:\users\phuc\downloads\ronwod.exe
Dec 30, 2024 @ 19:46:57.517 File deleted.
Dec 30, 2024 @ 19:46:57.295 active-response/bin/remove-threat.exe removed threat located at c:\users\phuc\downloads\cr.dll
Dec 30, 2024 @ 19:46:57.105 VirusTotal: Alert - c:\users\phuc\downloads\ronwod.exe - 41 engines detected this file
Dec 30, 2024 @ 19:46:55.743 File deleted.
Dec 30, 2024 @ 19:46:55.208 VirusTotal: Alert - c:\users\phuc\downloads\cr.dll - 11 engines detected this file
Dec 30, 2024 @ 19:46:52.776 File added to the system.
Dec 30, 2024 @ 19:46:52.691 File added to the system.

Hình 4.23 Lưu hash của mã độc để làm IOC

The screenshot shows the Wazuh interface with the 'Management' tab selected. In the 'CDB lists' section, there is a link to 'mal-md5-list'. Below it, there is a search bar labeled 'Search...' and a 'Key' section containing three hash values: 44d88612fea8a8f36de82e1278abb02f, a1809a9703c98f714bc85ba1a995588c, and 63ff0c8e75aa669f22e79ebf017c0aa8.

Hình 4.24 Kiểm tra lại hash của file trên VirusTotal

The screenshot shows the VirusTotal detection report for the file e8ac8d925f9b53bb66892cbac2f38cf7c1bcc5802a79c74c6d8b54e684b66e6aaronwod.exe. The report indicates a Community Score of 41/67. The main summary states: '41/67 security vendors flagged this file as malicious'. Below this, the file name is listed as e8ac8d925f9b53bb66892cbac2f38cf7c1bcc5802a79c74c6d8b54e684b66e6aaronwod.exe. Underneath the file name, there are three tags: 'peexe', 'spreader', and 'idle'. At the bottom of the report, there are tabs for 'DETECTION', 'DETAILS', 'RELATIONS', 'BEHAVIOR', and 'COMMUNITY' (which has a notification count of 3).

Hình 4.25 Kiểm tra lại hash của file trên VirusTotal

```

        break

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

message.alert = data

command = data.get("command")

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

return message

def send_keys_and_check_message(argv, keys):

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

    write_debug_file(argv[0], keys_msg)

    print(keys_msg)
    sys.stdout.flush()

```

Hình 4.26 Rule wazuh để kích hoạt xóa file tự động và lưu hash

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

    # write_debug_file(argv[0], input_str)

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

    action = data.get("command")

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

    return ret

def main(argv):
    write_debug_file(argv[0], "Started")

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

    if msg.command < 0:
```

Hình 4.27 Rule wazuh để kích hoạt xóa file tự động và lưu hash

```

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

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

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

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

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

    write_debug_file(argv[0], "Ended")

    sys.exit(OS_SUCCESS)

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

```

Hình 4.28 Rule wazuh để kích hoạt xóa file tự động và lưu hash

```

<!-- FIM -->
<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,">
    <rule id="100200" level="7">
        <if_sid>550</if_sid>
        <field name="file">C:/Users/Phuc/Downloads</field>
        <description>File modified in C:\Users\Phuc\Downloads directory.</description>
    </rule>
    <rule id="100201" level="7">
        <if_sid>554</if_sid>
        <field name="file">C:/Users/Phuc/Downloads</field>
        <description>File added to C:\Users\Phuc\Downloads directory.</description>
    </rule>
</group>

<!-- VIRUSTOTAL & Auto-delete Malware -->
<group name="virustotal,">
    <rule id="100092" level="12">
        <if_std>657</if_std>
        <match>Successfully removed threat</match>
        <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>
    </rule>
    <rule id="100093" level="12">
        <if_std>657</if_std>
        <match>Error removing threat</match>
        <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
    </rule>
</group>

<!-- BUILD IOCs -->
<group name="locs,>
    <rule id="111000" level="0">
        <decoded_as>ioc_builder</decoded_as>
        <description>Grouping of IoC rules.</description>
    </rule>

    <rule id="111001" level="5">
        <if_sid>111000</if_sid>
        <field name="ioc_not_found">^True$</field>
        <description>Suspicious IoC "$(ioc)" added to "$(ioc_file)".</description>
    </rule>

    <rule id="111002" level="5" ignore="60">
        <if_sid>111000</if_sid>
        <field name="ioc_not_found">^False$</field>
        <description>Suspicious IoC "$(ioc)" already found in "$(ioc_file)".</description>
    </rule>
</group>

```

4.6 Kịch bản 6: Ngăn chặn các tác nhân gây hại đã biết trước

4.6.1 Mô tả:

Mô tả chi tiết:

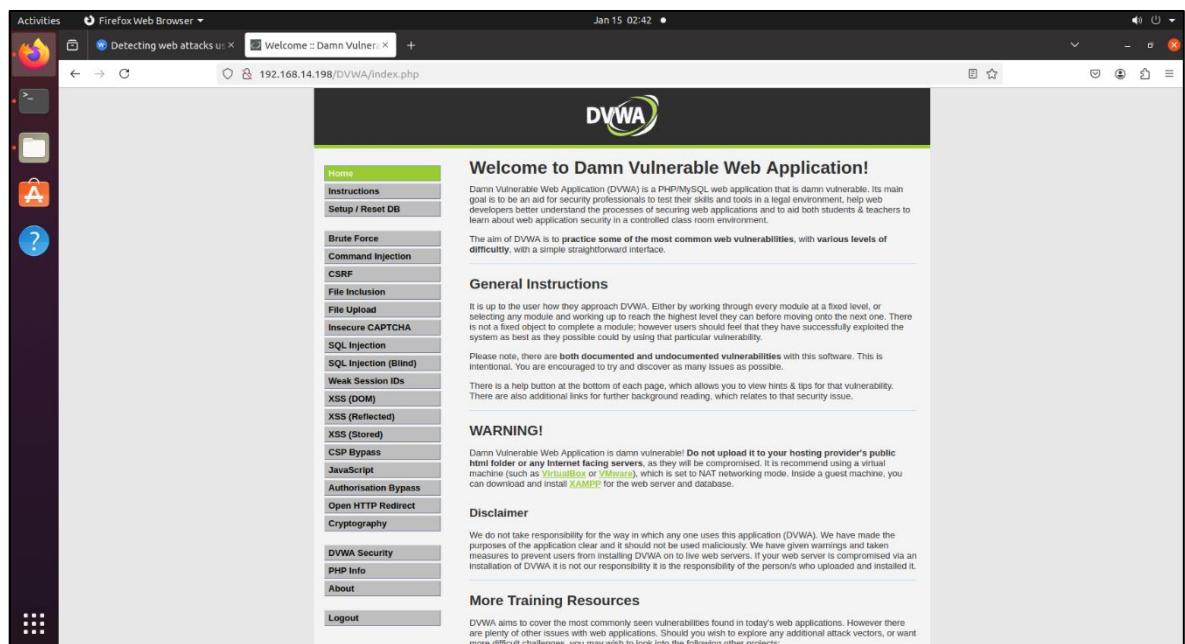
- Cấu hình Active Response trên Wazuh để nhận diện và chặn từ kết nối từ địa chỉ IP tấn công giả lập (RHEL attacker)
- Máy tấn công có gắng truy cập tài nguyên web trên các endpoint (Windows và Ubuntu)
- Active Response tự động thêm IP này và danh sách chặn, từ chối mọi kết nối sau đó trong 60 giây. Log sự kiện sẽ hiển thị trên dashboard

4.6.2 Thực hiện:

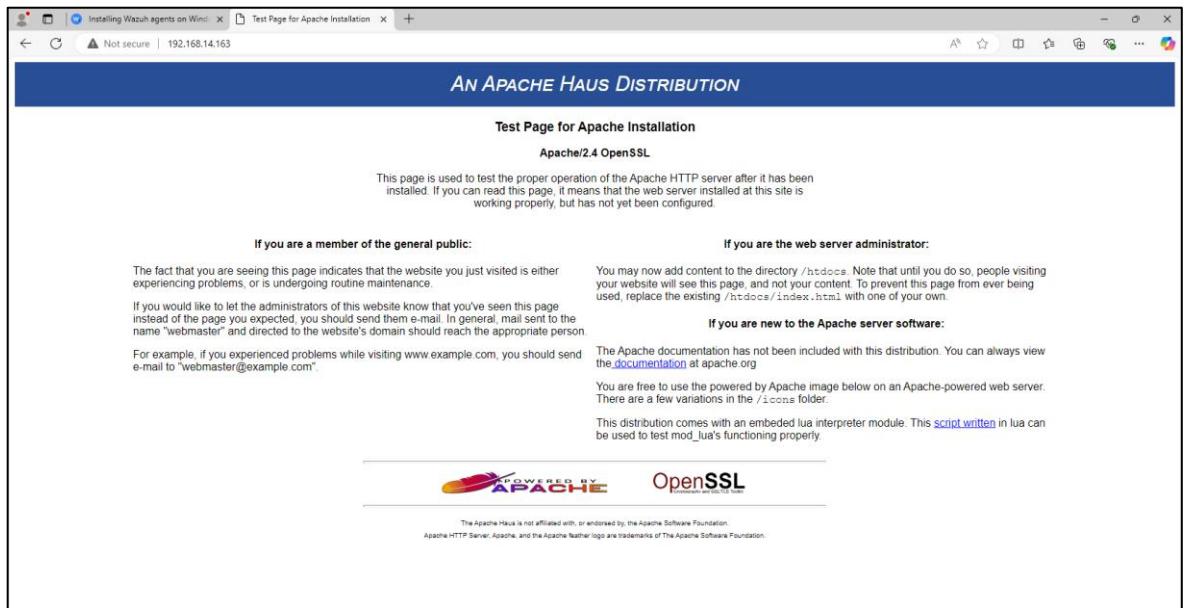
Mục tiêu: Triển khai cơ chế bảo vệ chủ động bằng cách sử dụng Wazuh để giám sát và chặn các địa chỉ IP độc hại, bảo vệ tài nguyên web trên các điểm cuối.

Chi tiết thực hiện:

- Sử dụng máy DVWA từ kịch bản trước và cấu hình thêm 1 máy endpoint Windows chạy máy chủ web Apache (máy endpoint này cũng cài đặt Wazuh Agent)



Hình 4.29 Hai máy endpoint chạy máy chủ web Apache



Hình 4.30 Hai máy endpoint chạy máy chủ web Apache

- Cấu hình cho Wazuh nhận diện và đánh dấu IP máy RHEL attacker là tấn công độc hại. Kích hoạt cơ chế chặn truy cập từ IP độc hại bằng Active Response bảo vệ tài nguyên web

```
kenzy@kenzy-virtual-machine: $ sudo wget https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/allenvault_reputation.ipset -O /var/ossec/etc/lists/allenvault_reputation.ipset
--2025-01-09 18:29:42-- https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/allenvault_reputation.ipset
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9495 (9,3K) [text/plain]
Saving to: '/var/ossec/etc/lists/allenvault_reputation.ipset'

/var/ossec/etc/lists/allenvault_reputation.ipset 100%[=====] 9,27K --.-KB/s in 0,06
1s
2025-01-09 18:29:43 (17,1 MB/s) - '/var/ossec/etc/lists/allenvault_reputation.ipset' saved [9495/9495]
kenzy@kenzy-virtual-machine: $ sudo echo 192.168.14.204 >> /var/ossec/etc/lists/allenvault_reputation.ipset
```

Hình 4.31 Cấu hình rule cho Wazuh nhận diện địa chỉ IP

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

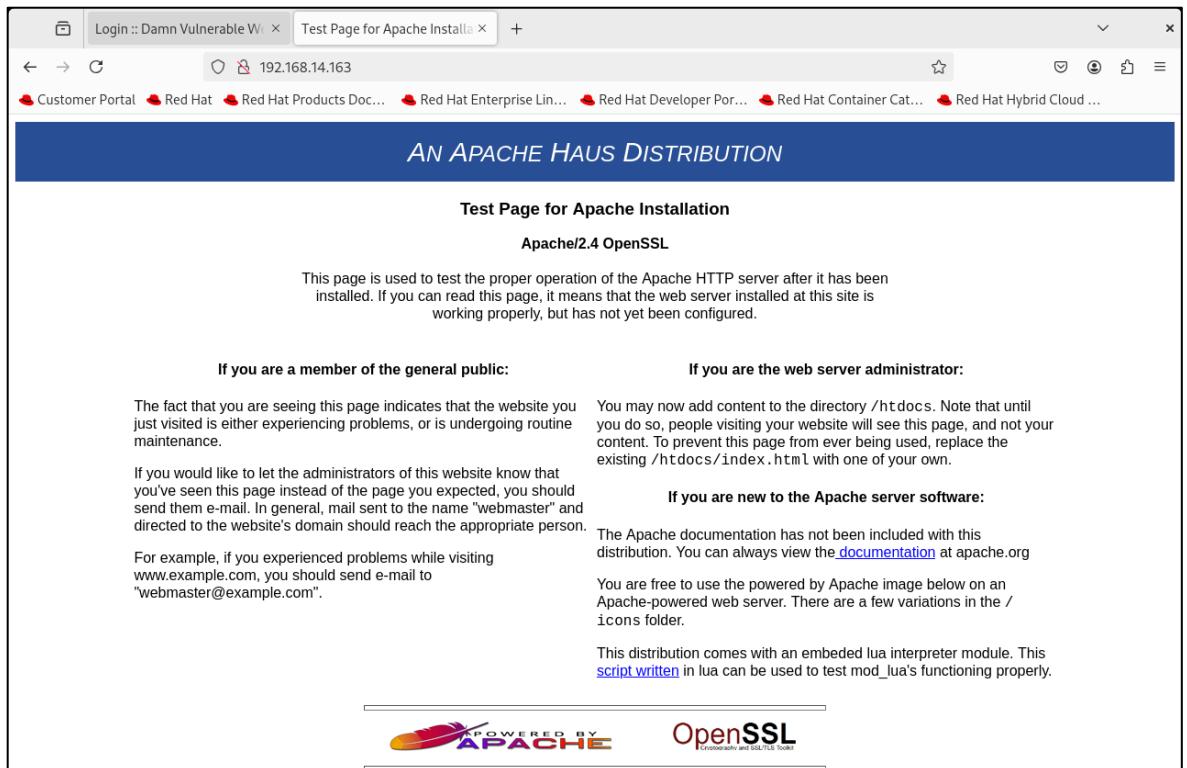
Hình 4.32 Cấu hình rule cho Wazuh nhận diện địa chỉ IP

```
<ossec_config>
  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>60</timeout>
  </active-response>
</ossec_config>

<ossec_config>
  <active-response>
    <command>netsh</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>60</timeout>
  </active-response>
</ossec_config>
```

Hình 4.33 Cấu hình active response thực hiện chặn địa chỉ IP đáng ngờ trong 60 giây

- Ta thử truy cập lại bằng máy RHEL attacker



Hình 4.34 Máy RHEL truy cập vào máy chủ web

- Wazuh phát hiện hành vi đáng ngờ từ IP của máy RHEL

t agent.name	UbuntuModsec	t agent.name	DESKTOP-18K2G7V
t data.id	302	t data.id	200
t data.protocol	GET	t data.protocol	GET
t data.srcip	192.168.14.204	t data.srcip	192.168.14.204
t data.url	/DVWA/	t data.url	/
t decoder.name	web-accesslog	t decoder.name	web-accesslog
t full_log	192.168.14.204 - - [10/Jan/2025:06:49:45 -0800] "GET /DVWA/ HTTP/1.1" 200 11992	t full_log	192.168.14.204 - - [10/Jan/2025:22:04:33 +0700] "GET / HTTP/1.1" 200 11992
t id	1736520585.1460879	t id	1736521480.1607025
t input.type	log	t input.type	log
t location	/var/log/apache2/access.log	t location	C:\Apache24\logs\access.log
t manager.name	kenzy-virtual-machine	t manager.name	kenzy-virtual-machine
t rule.description	IP address found in Blacklist database.	t rule.description	IP address found in Blacklist database.
# rule.firedtimes	16	# rule.firedtimes	15
t rule.groups	attack	t rule.groups	attack

Hình 4.35 Wazuh server nhận được log từ active response trả về

Cơ chế Active Response trả về log cho chúng ta khi ngăn chặn hành vi đáng ngờ từ địa chỉ IP độc hại

- + Với máy endpoint Windows



Hình 4.36 Log active response trả về khi Wazuh nhận thấy hành vi đáng ngờ từ IP máy RHEL

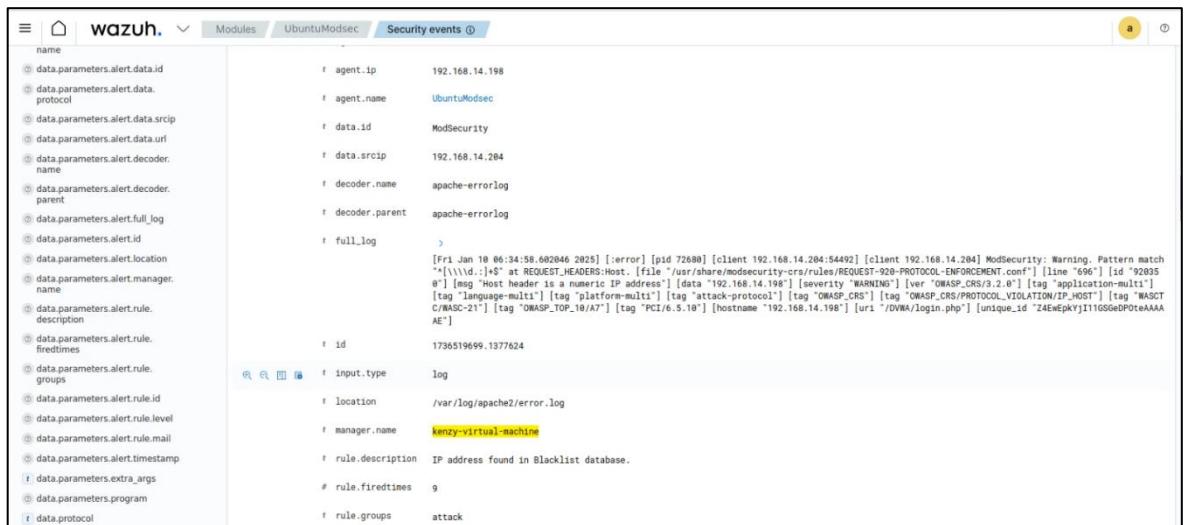
This screenshot shows a detailed log entry from the Wazuh interface. The log is presented in a table with two columns. The left column contains various alert parameters, and the right column contains their corresponding values. The log entry details an active response triggered by a specific alert rule.

t agent.ip	192.168.14.163
t agent.name	DESKTOP-18K267V
t data.command	delete
t data.origin.module	wazuh-execd
t data.origin.name	node01
t data.parameters.alert.agent.id	002
t data.parameters.alert.agent.ip	192.168.14.163
t data.parameters.alert.agent.name	DESKTOP-18K267V
t data.parameters.alert.data.id	200
t data.parameters.alert.data.protocol	GET
t data.parameters.alert.data.srcip	192.168.14.204
t data.parameters.alert.data.url	/
t data.parameters.alert.decoder.name	web-accesslog
t data.parameters.alert.full_log	192.168.14.204 - - [10/Jan/2025:22:04:33 +0700] "GET / HTTP/1.1" 200 11992
t data.parameters.alert.id	1736521488.1607025
t data.parameters.alert.location	C:\Apache24\logs\access.log
t data.parameters.alert.manager.name	kenzy-virtual-machine
t data.parameters.alert.rule.description	IP address found in Blacklist database.

Hình 4.37 Chi tiết về 1 log được active response trả về trên Wazuh
+ Với máy endpoint Ubuntu:



Hình 4.38 Log active response trả về khi nhận thấy hành vi đáng ngờ từ IP máy RHEL đến endpoint Ubuntu



Hình 4.39 Chi tiết 1 log được active response trả về trên Wazuh

Kết quả mong đợi:

- Hành vi đáng ngờ từ địa chỉ IP độc hại được ngăn chặn thành công
- Log sự kiện phản ánh rõ ràng các thông tin như IP, thời gian chặn, lý do chặn

4.7 Kịch bản 7: Tích hợp Wazuh và Teler để phát hiện tấn công

4.7.1 Mô tả:

Mô tả chi tiết

- Cấu hình Wazuh và tích hợp các rule từ công cụ Teler
- Sử dụng Nikto trên máy attacker (kali) để thực hiện cuộc tấn công giả lập

- Teler phát hiện tấn công, gửi log tới Wazuh và hiển thị cảnh báo trên dashboard

4.7.2 Thực hiện:

Mục tiêu: Kết hợp Teler với Wazuh để phát hiện và ghi nhận log từ các cuộc tấn công web giả lập.

Chi tiết thực hiện:

- Cài đặt Teler trên máy endpoint Ubuntu chạy web server DVWA

Hình 4.40 Cài đặt Teler

- Tích hợp các rule của Teler với Wazuh Agent để ghi nhận log

```
<group name="teler">
  <rule id="100012" level="10">
    <decoded_as>json</decoded_as>
    <field name="category" type="pcre2">Common Web Attack(: .*)?|CVE-[0-9]{4}-[0-9]{4,7}</field>
    <field name="request_uri" type="pcre2">\D.+|-</field>
    <field name="remote_addr" type="pcre2">\d+.\d+.\d+.\d+|::1</field>
    <mitre>
      <id>T1210</id>
    </mitre>
    <description>teler detected ${category} against resource ${request_uri} from ${remote_addr}</description>
  </rule>

  <rule id="100013" level="10">
    <decoded_as>json</decoded_as>
    <field name="category" type="pcre2">Bad (IP Address|Referrer|Crawler)</field>
    <field name="request_uri" type="pcre2">\D.+|-</field>
    <field name="remote_addr" type="pcre2">\d+.\d+.\d+.\d+|::1</field>
    <mitre>
      <id>T1590</id>
    </mitre>
    <description>teler detected ${category} against resource ${request_uri} from ${remote_addr}</description>
  </rule>

  <rule id="100014" level="10">
    <decoded_as>json</decoded_as>
    <field name="category" type="pcre2">Directory Bruteforce</field>
    <field name="request_uri" type="pcre2">\D.+|-</field>
    <field name="remote_addr" type="pcre2">\d+.\d+.\d+.\d+|::1</field>
    <mitre>
      <id>T1595</id>
    </mitre>
    <description>teler detected ${category} against resource ${request_uri} from ${remote_addr}</description>
  </rule>
</group>
```

Hình 4.41 Rule được tích hợp

Hình: Rule được tích hợp với Wazuh Agent

- Sử dụng công cụ Nikto trên máy attacker (Kali) để thực hiện giả lập tấn công

```
(kali㉿kali)-[~]
$ nikto -h http://192.168.14.198/DVWA/

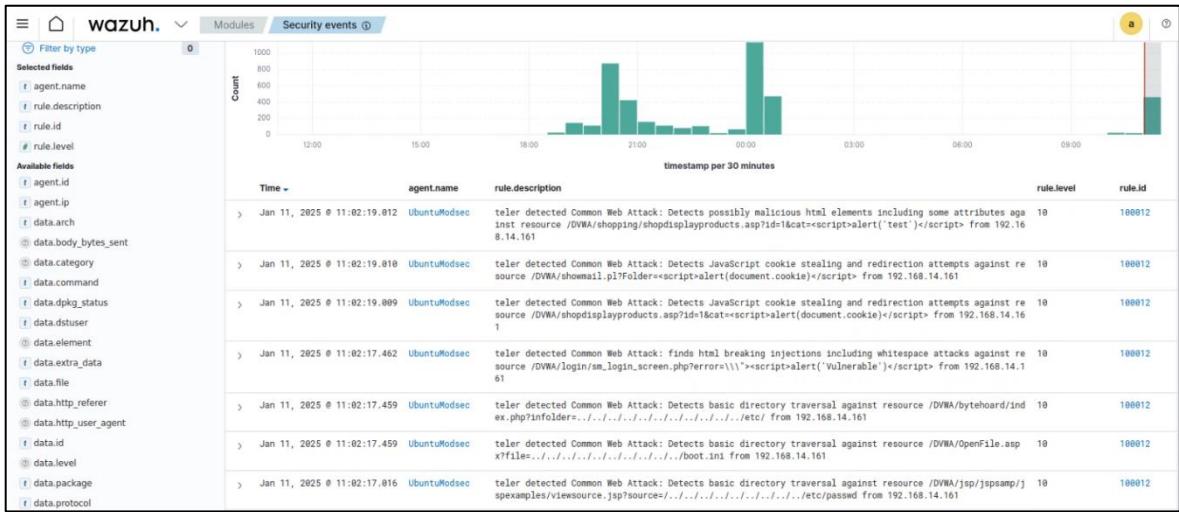
- Nikto v2.5.0
=====
+ Target IP:          192.168.14.198
+ Target Hostname:    192.168.14.198
+ Target Port:        80
+ Start Time:         2025-01-10 12:30:15 (GMT-5)
=====
+ Server: Apache/2.4.41 (Ubuntu)
+ /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page /DVWA redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ /DVWA/login.php: Admin login page/section found.
+ /DVWA/.dockerrigore: .dockerrigore file found. It may be possible to grasp the directory structure and learn more about the site.
+ 8102 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2025-01-10 12:30:48 (GMT-5) (33 seconds)
=====
+ 1 host(s) tested
```

Hình 4.42 Thực hiện mô phỏng tấn công Nikto

- Teler phát hiện các yêu cầu tấn công và ghi log chi tiết

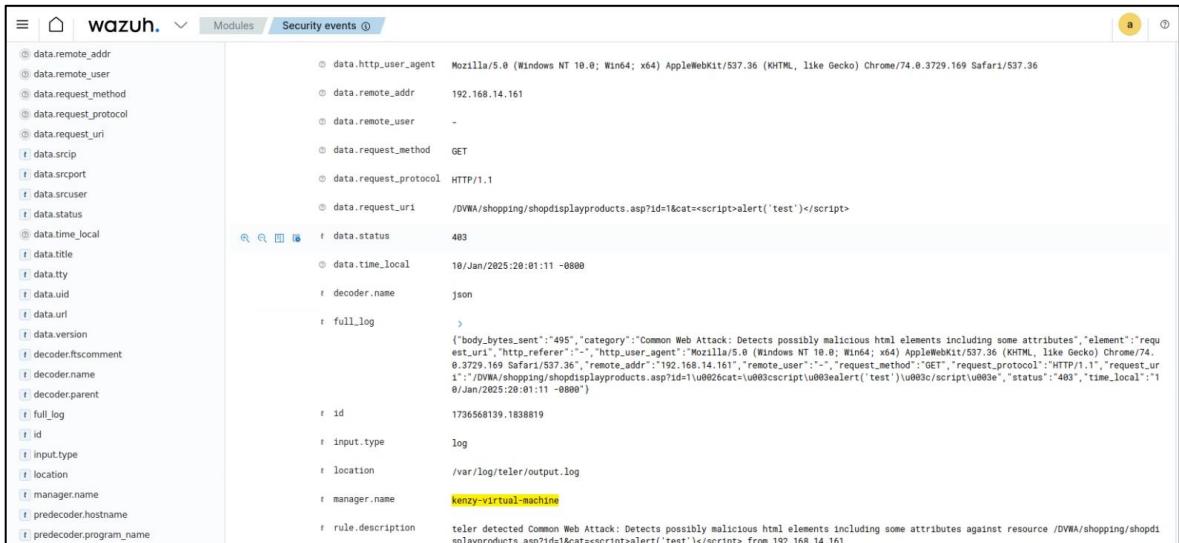
Hình 4.43 Log Teler trả về khi bị tấn công

- Wazuh Agent đẩy log từ Teler lên Wazuh Server



Hình 4.44 Log từ Teler được đẩy về Wazuh server

- Chi tiết log cảnh báo trên Wazuh dashboard



Hình 4.45 Chi tiết log mà Wazuh nhận được khi bị tấn công

Kết quả mong đợi:

- Các tấn công được phát hiện và log sự kiện được gửi lên Wazuh
- Dashboard hiển thị chi tiết các loại tấn công và IP nguồn

4.8 Kịch bản 8: Tích hợp Suricata IDS với Wazuh phát hiện hành vi bất thường

4.8.1 Mô tả:

Mô tả chi tiết:

- Cài đặt và cấu hình Suricata để giám sát lưu lượng mạng
- Thực hiện các cuộc tấn công mạng giả lập như scan port hay tấn công brute force
- Suricata phát hiện các hành vi tấn công bất thường, gửi log tới Wazuh và cung cấp thông tin chi tiết trên dashboard

4.8.2 Thực hiện:

Mục tiêu: Đánh giá khả năng giám sát lưu lượng mạng và phát hiện các hành vi bất thường khi tích hợp Suricata với Wazuh

Chi tiết thực hiện:

- Cài đặt Suricata trên máy chủ DVWA để giám sát lưu lượng mạng

```
kenzy@ubuntu:~$ sudo add-apt-repository ppa:olsf/suricata-stable
[sudo] password for kenzy:
Sorry, try again.
[sudo] password for kenzy:
Suricata IDS/IPS/NSM stable packages
https://suricata.io/
https://olsf.net/

Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and Prevention System and Network Security
Open Source and owned by a community run non-profit foundation, the Open Information Security Foundation (OISF). Suricata
Community.

This Engine supports:

- Multi-Threading - provides for extremely fast and flexible operation on multicore systems.
- Multi Tenancy - Per vlan/Per interface
- Uses Rust for most protocol detection/parsing
- TLS/SSL certificate matching/logging
- JA3 TLS client fingerprinting
- JA3S TLS server fingerprinting
- IEEE 802.1ad (QinQ) and IEEE 802.1Q (VLAN) support
- VXLAN support
- All JSON output/logging capability
- IDS runmode
- IPS runmode
- IDPS runmode
```

Hình 4.46 Cài đặt Suricata

- Thêm rule vào file localrule của Wazuh server nhằm phát hiện các dạng tấn công

```

<group name="custom_active_response_rules">
  <rule id="100200" level="12">
    <if_sid>86600</if_sid>
    <field name="event_type">^alert$</field>
    <match>ET DOS Inbound GoldenEye DoS attack</match>
    <description>GoldenEye DoS attack has been detected. </description>
    <mitre>
      <id>T1498</id>
    </mitre>
  </rule>

  <rule id="100201" level="12">
    <if_sid>86600</if_sid>
    <field name="event_type">^alert$</field>
    <match>ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)</match>
    <description>Nmap scripting engine detected. </description>
    <mitre>
      <id>T1595</id>
    </mitre>
  </rule>
</group>

```

Hình 4.47 Thêm rule tích hợp trên Wazuh server

- Thực hiện mô phỏng tấn công mạng

```

└─(kali㉿kali)-[~]
$ sudo nmap -sS --script=vuln 192.168.14.198
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-11 03:05 EST
zsh: suspended  sudo nmap -sS --script=vuln 192.168.14.198

```

Hình 4.48 Thực hiện tấn công mô phỏng

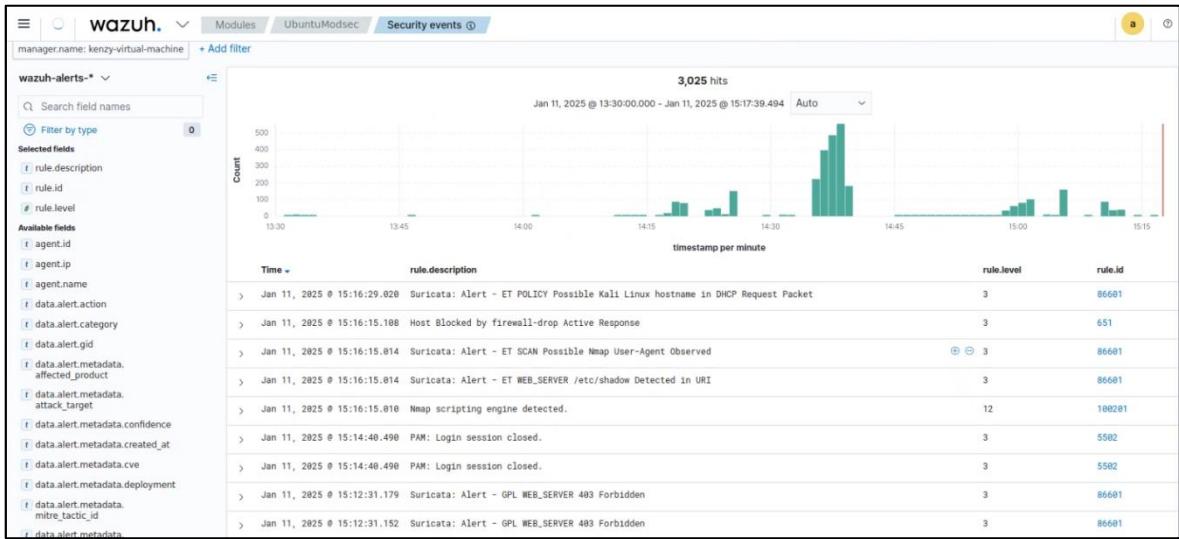
```

└─(kali㉿kali)-[~/GoldenEye]
$ ./goldeneye.py http://192.168.14.198

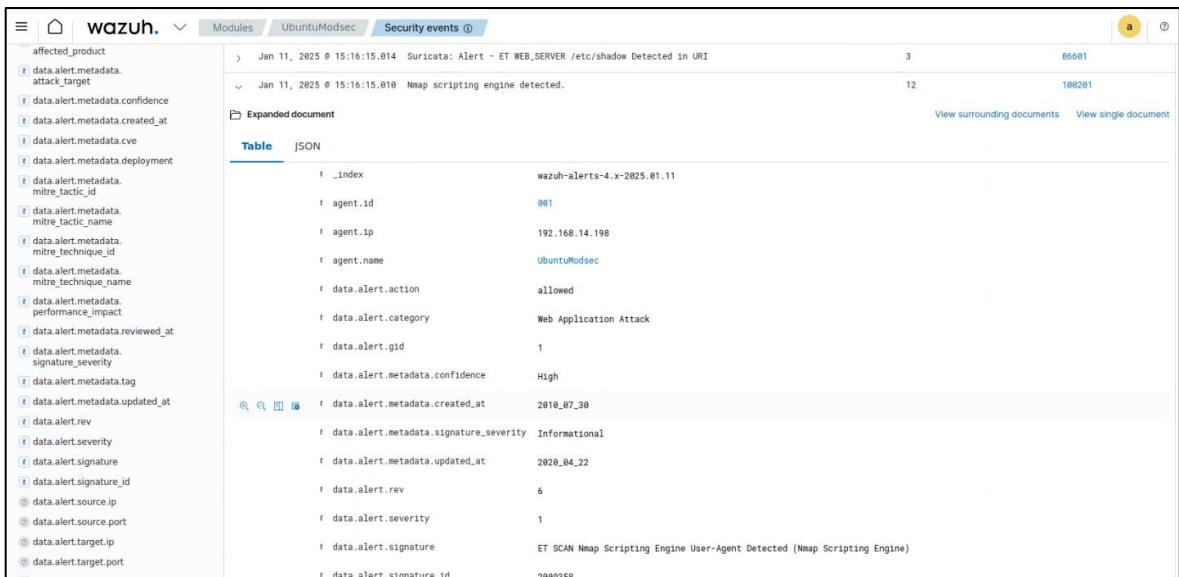
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>
[...]
Hitting webserver in mode 'get' with 10 workers running 500 connections each. Hit CTRL+C to cancel.
[...]
results saved in nmap_scans_results.txt

```

Hình 4.49 Thực hiện tấn công mô phỏng



Hình 4.50 Log từ Suricata đầy về Wazuh server



Hình 4.51 Chi tiết log từ Wazuh dashboard

Kết quả mong đợi:

- Suricata phát hiện hiệu quả các cuộc tấn công giả lập
- Log sự kiện được gửi đến Wazuh và hiển thị chi tiết trên dashboard

LINK VIDEO DEMO:

<https://drive.google.com/drive/folders/1CJj6xX-KhlrlVlqCIMSkM5s87hsXipZi?usp=sharing>

CHƯƠNG 5: KẾT LUẬN

Nghiên cứu đã khẳng định Wazuh XDR là một công cụ mạnh mẽ và hiệu quả trong việc bảo mật mạng máy tính. Việc triển khai hệ thống trong môi trường mô phỏng đã giúp kiểm chứng các tính năng quan trọng như quản lý tập trung log, phát hiện mối đe dọa thời gian thực và khả năng phản ứng nhanh chóng. Kết quả cho thấy Wazuh XDR không chỉ giúp phát hiện mà còn hỗ trợ ngăn chặn các mối đe dọa tiềm tàng thông qua việc tích hợp hiệu quả các công cụ bảo mật như ModSecurity, Snort, Suricata, ClamAV...

Hệ thống đã đạt được các mục tiêu đề ra ban đầu, bao gồm việc giảm thiểu các rủi ro bảo mật và cải thiện khả năng giám sát an ninh mạng một cách toàn diện. Các cảnh báo được tạo ra kịp thời và chính xác, cho phép người quản trị viên xử lý các mối đe dọa một cách nhanh chóng. Đồng thời, khả năng tùy chỉnh và mở rộng của Wazuh cũng giúp nó phù hợp với nhiều mô hình tổ chức khác nhau.

Tuy nhiên vẫn còn một số hạn chế, như việc tối ưu hóa các quy tắc phát hiện để giảm thiểu cảnh báo giả và quản lý hiệu quả dữ liệu nhật ký lớn. Đây là những vấn đề cần được giải quyết trong tương lai để nâng cao hiệu suất và độ tin cậy của hệ thống hơn nữa.

Tóm lại, Wazuh là một giải pháp toàn diện và linh hoạt, xứng đáng được dùng để triển khai trong các hệ thống mạng thực tế nhằm tăng cường khả năng bảo vệ và giám sát an ninh mạng.

CHƯƠNG 6: HƯỚNG PHÁT TRIỂN

Các hướng phát triển trong tương lai gồm:

- **Tăng cường tích hợp:** Kết hợp với các nguồn thông tin tình báo mới đe dọa bổ sung để nâng cao khả năng phát hiện các mối đe dọa tinh vi hơn và công cụ bảo mật khác.
- **Tự động hóa:** Tự động hóa các hành động như chặn IP tấn công, cài đặt thiết bị bị xâm nhập, ứng dụng học máy để tối ưu phát hiện và giảm cảnh báo giả.
- **Mở rộng kịch bản:** Thêm kịch bản tấn công nâng cao (APT, ransomware) và tích hợp thêm thiết bị IoT để kiểm tra khả năng bảo vệ trong môi trường đa dạng hơn.
- **Đào tạo và tài liệu:** Xây dựng tài liệu chi tiết và các bài tập thực hành để hỗ trợ tổ chức khác triển khai Wazuh XDR một cách hiệu quả.

TÀI LIỆU THAM KHẢO

Tài liệu hướng dẫn Wazuh: <https://documentation.wazuh.com/>

Kịch bản tham khảo: <https://igorsec.blog/category/home-lab/>

Tài liệu hướng dẫn sử dụng Snort, ModSecurity, ClamAV, Suricata