



ÉCOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE - MOHAMMEDIA

Department of Information and Mathematics

IT Engineering, Cybersecurity and Digital Trust engineering

MODULE : CLOUD COMPUTING

Wazuh SIEM/EDR Cloud Lab Implementation

Elèves :
Yasser NAMEZ

Enseignant :
Prof. Azeddine KHIAT

Table des matières

1 Architecture VPC + Security Groups + EC2 Instances	4
1.1 Overview	4
1.2 AWS VPC Configuration	4
1.3 EC2 Instances	5
1.3.1 Wazuh Server Instance	6
1.3.2 Linux Client Instance	6
1.3.3 Windows Client Instance	7
1.4 Security Groups Configuration	7
1.4.1 Wazuh Server Security Group	7
1.4.2 Wazuh Clients Security Group	8
1.5 Network Architecture Diagram	8
1.6 Connectivity Testing	9
1.6.1 SSH to Wazuh Server	9
1.6.2 SSH to Linux Client	9
1.6.3 RDP to Windows Client	10
1.7 Infrastructure Summary	10
2 Wazuh Server Setup and Dashboard	11
2.1 Wazuh Services Status	11
2.2 Wazuh Dashboard Access	12
2.3 Agent Enrollment and Status	13
2.3.1 Linux Agent Connection	13
2.3.2 Windows Agent Connection	14
2.4 Active Agents Overview	15
3 Security Events - Linux Client	17
3.1 Alert 1 : SSH Brute Force Attack	17
3.1.1 Attack Simulation	17
3.1.2 Wazuh Detection	18
3.1.3 Event Details	18
3.1.4 Security Implication	18
3.2 Alert 2 : Privilege Escalation (Sudo)	19
3.2.1 Wazuh Detection	19
3.2.2 Event Details	20
3.2.3 Security Implication	20
3.3 Alert 3 : File Integrity Monitoring (FIM)	20
3.3.1 Event Details	21
3.3.2 Security Implication	21
4 Security Events - Windows Client	22
4.1 Alert 1 : Failed Logon Attempts	22
4.1.1 Wazuh Detection	22
4.1.2 Event Details	23
4.1.3 Security Implication	23
4.2 Alert 2 : User Account Creation and Group Modification (IAM)	23
4.2.1 Wazuh Detection	24

4.2.2	Event Details - User Creation (4720)	24
4.2.3	Event Details - Group Modification (4732)	24
4.2.4	Security Implication	25
4.3	Alert 3 : Sysmon Process Creation (EDR)	25
4.3.1	Sysmon Events in Wazuh	25
4.3.2	Event Details	26
4.3.3	EDR Value	26
5	SIEM vs EDR - Comparative Analysis	27
5.1	SIEM (Security Information and Event Management)	27
5.1.1	Key Capabilities Demonstrated in Lab	27
5.1.2	Lab Use Cases	27
5.1.3	Strengths & Limitations	27
5.2	EDR (Endpoint Detection and Response)	27
5.2.1	Key Capabilities Demonstrated in Lab	28
5.2.2	Lab Use Cases	28
5.2.3	Strengths & Limitations	28
5.3	Wazuh : Unified SIEM + EDR Platform	28
5.3.1	Demonstrated Integration Benefits	29
6	Identity and Access Management (IAM) & Privileged Access Management (PAM)	30
6.1	IAM Concepts Demonstrated	30
6.1.1	Authentication Monitoring	30
6.1.2	Account Lifecycle Management	30
6.1.3	Authorization & Privilege Changes	30
6.1.4	Identity-Based Threat Detection	30
6.2	PAM (Privileged Access Management) Relevance	30
6.2.1	Privileged Activity Monitoring	30
6.2.2	Compliance Support	30
6.3	Lab Demonstration Summary	31
7	Threat Hunting - Sample Queries & Analysis	32
7.1	Query 1 : Failed Authentication Analysis	32
7.1.1	Wazuh Query	32
7.1.2	Analysis Approach	32
7.1.3	Detection Indicators	32
7.1.4	Response Actions	32
7.2	Query 2 : Privileged Account Activity	32
7.2.1	Wazuh Query	32
7.2.2	Analysis Approach	33
7.2.3	Detection Indicators	33
7.2.4	Response Actions	33
7.3	Query 3 : Process Anomaly Detection (EDR)	33
7.3.1	Wazuh Query	33
7.3.2	Analysis Approach	33
7.3.3	Detection Indicators	34
7.3.4	Response Actions	34

7.4 Threat Hunting Best Practices	34
8 Conclusion	35
8.1 Technical Achievements	35
8.2 Security Concepts Validated	35
8.3 Operational Value	35

1 Architecture VPC + Security Groups + EC2 Instances

1.1 Overview

This section describes the cloud infrastructure deployed on AWS for the Wazuh SIEM/EDR lab environment. The architecture follows security best practices with network segmentation and least-privilege access controls.

1.2 AWS VPC Configuration

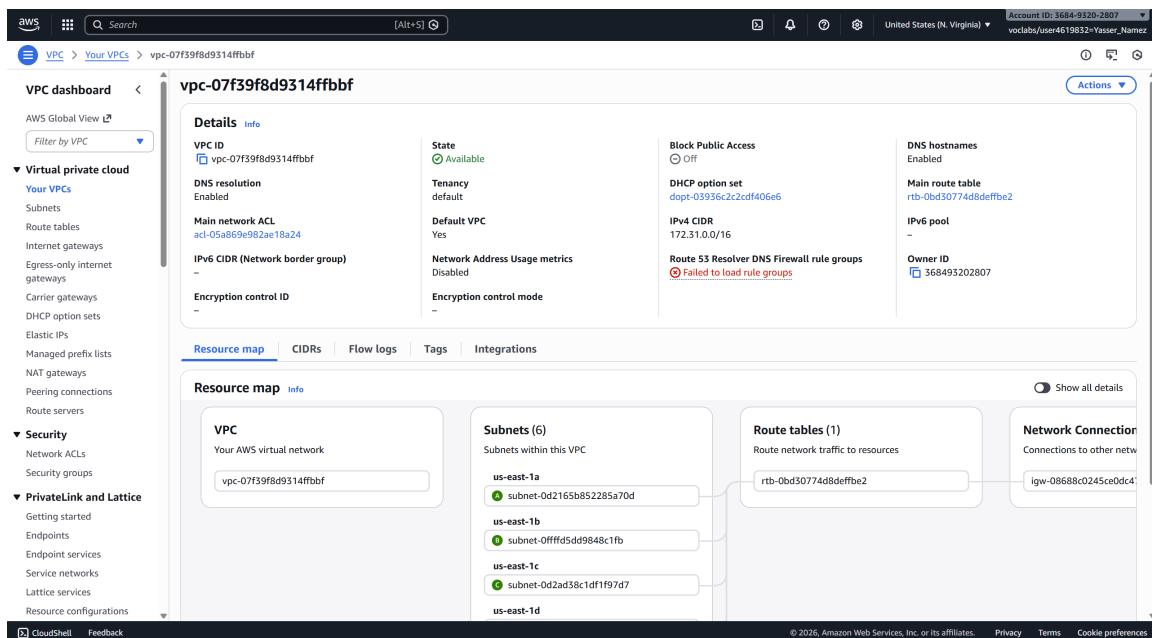
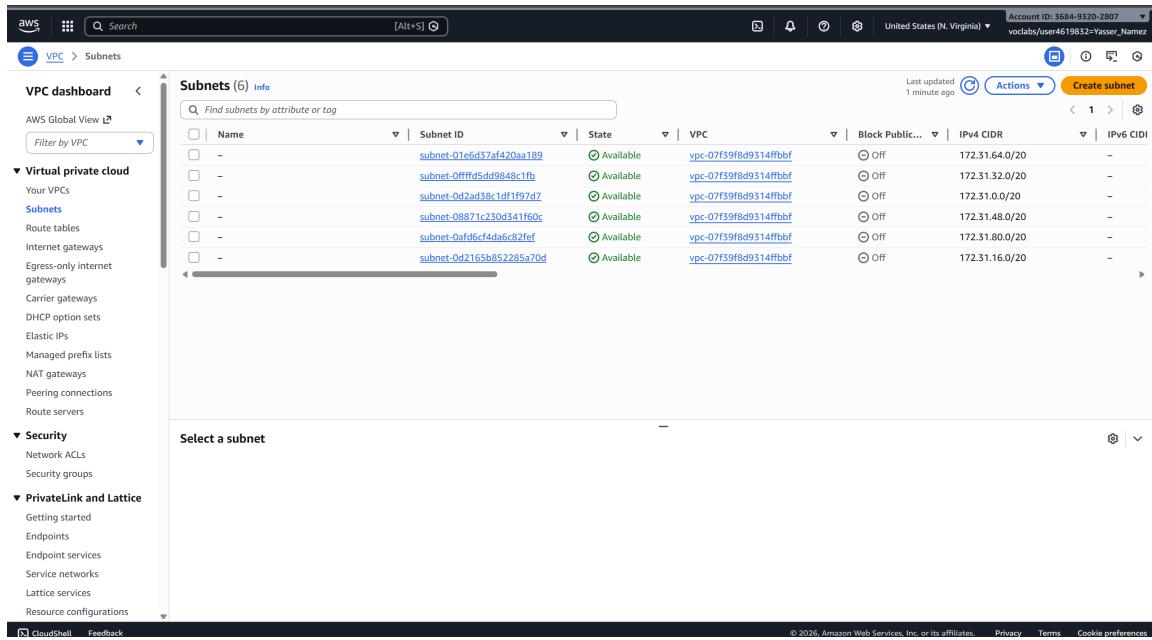


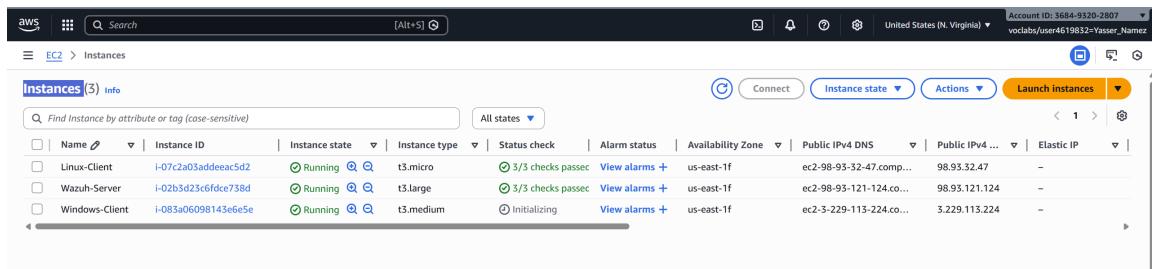
FIGURE 1 – AWS VPC Configuration - 10.0.0.0/16



Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR	IPv6 CIDR
-	subnet-01e6d37af420aa189	Available	vpc-07f39f8d9314ffbbf	Off	172.31.64.0/20	-
-	subnet-0ffffd5dd9848c1fb	Available	vpc-07f39f8d9314ffbbf	Off	172.31.32.0/20	-
-	subnet-0d2ad38c1df1f97d7	Available	vpc-07f39f8d9314ffbbf	Off	172.31.0.0/20	-
-	subnet-08871c230d341f50c	Available	vpc-07f39f8d9314ffbbf	Off	172.31.48.0/20	-
-	subnet-0af6fc4d46c62fe	Available	vpc-07f39f8d9314ffbbf	Off	172.31.80.0/20	-
-	subnet-0d2165b852285a70d	Available	vpc-07f39f8d9314ffbbf	Off	172.31.16.0/20	-

FIGURE 2 – VPC Subnets Configuration

1.3 EC2 Instances



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Linux-Client	i-07ca03addeac5d2	Running	t3.micro	3/3 checks passed	View alarms	us-east-1f	ec2-98-93-32-47.comp...	98.93.32.47	-
Wazuh-Server	i-02b3d23c6fdce738d	Running	t3.large	3/3 checks passed	View alarms	us-east-1f	ec2-98-93-121-124.co...	98.93.121.124	-
Windows-Client	i-083a06098143e6e5e	Running	t3.medium	Initializing	View alarms	us-east-1f	ec2-3-229-113-224.co...	3.229.113.224	-

FIGURE 3 – All EC2 Instances Running

1.3.1 Wazuh Server Instance

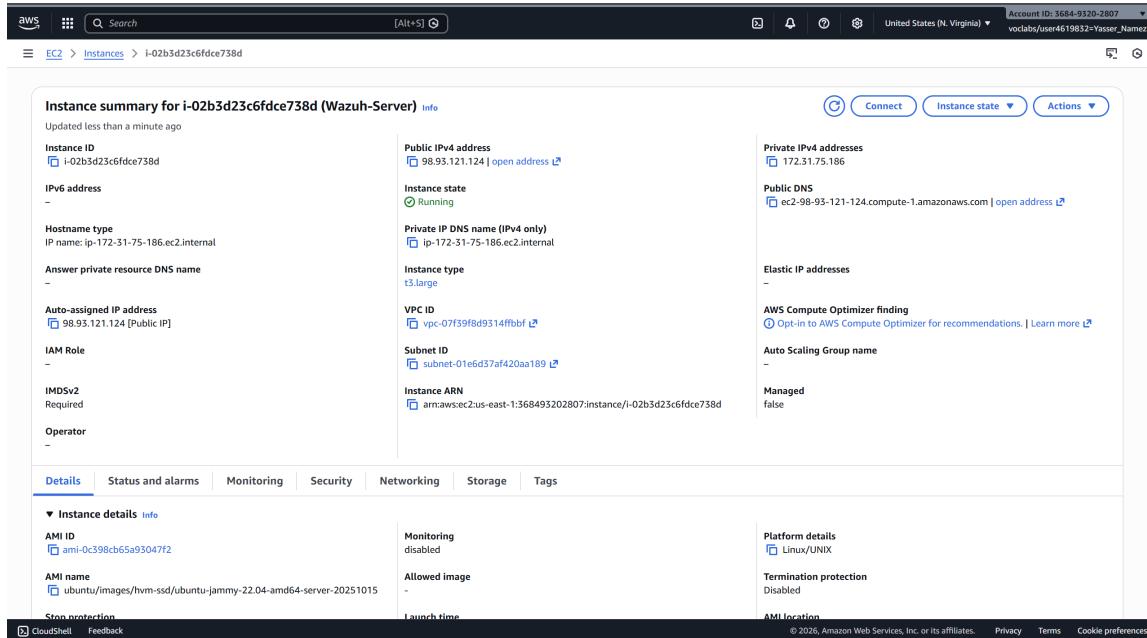


FIGURE 4 – Wazuh-Server Instance Details (Ubuntu 22.04, t3.large)

1.3.2 Linux Client Instance

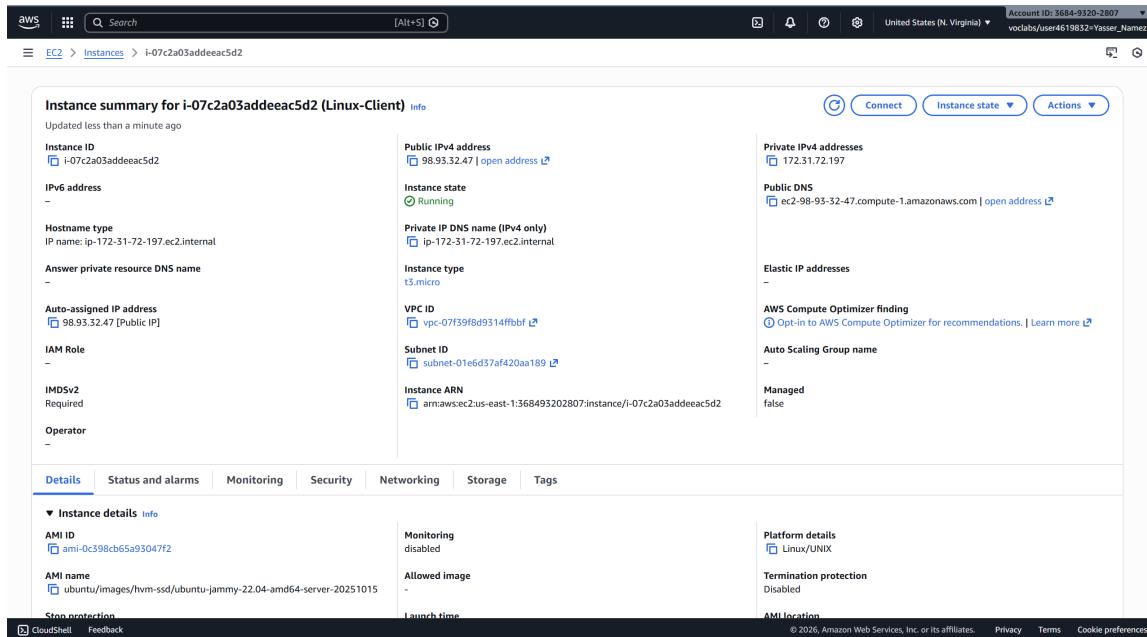


FIGURE 5 – Linux-Client Instance Details (Ubuntu 22.04, t2.micro)

1.3.3 Windows Client Instance

Instance summary for i-083a06098143e6e5e (Windows-Client) [Info](#)

Updated less than a minute ago

Instance ID	i-083a06098143e6e5e	Public IPv4 address	3.229.113.224 open address
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-64-222.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-64-222.ec2.internal
Answer private resource DNS name	-	Instance type	t3.medium
Auto-assigned IP address	3.229.113.224 [Public IP]	VPC ID	vpc-07f39f8d9314ffbf
IAM Role	-	Subnet ID	subnet-01e6d37af420aa189
IMDSv2	Required	Instance ARN	arn:aws:ec2:us-east-1:368493202807:instance/i-083a06098143e6e5e
Operator	-		

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

▼ Instance details [Info](#)

AMI ID	ami-06777e7ef7441def	Monitoring	disabled	Platform details	Windows
AMI name	Windows_Server-2025-English-Full-Base-2025.12.10	Allowed image	-	Termination protection	Disabled
Stop protection	-	Launch time	-	AMI iteration	-

© 2026, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

FIGURE 6 – Windows-Client Instance Details (Windows Server 2022, t2.medium)

1.4 Security Groups Configuration

1.4.1 Wazuh Server Security Group

sg-013b8487893e74ae4 - Wazuh-Server-SG

[Actions](#)

Details	Security group name : Wazuh-Server-SG	Security group ID : sg-013b8487893e74ae4	Description : Security group for Wazuh server	VPC ID : vpc-07f39f8d9314ffbf
Owner : 368493202807	Inbound rules count : 4 Permission entries	Outbound rules count : 1 Permission entry		

[Inbound rules](#) [Outbound rules](#) [Sharing](#) [VPC associations](#) [Tags](#)

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sg-03666d19a960b7635d	IPv4	SSH	TCP	22	196.64.23.110/32	SSH access
-	sgr-09ec4469022dbd737	IPv4	HTTPS	TCP	443	196.64.23.110/32	Wazuh Dashboard
-	sgr-01c348c15e6fcf4c2	-	Custom TCP	TCP	1514	sg-0be305e637154660...	Agent communication
-	sgr-0ca5259e1ea4738b	-	Custom TCP	TCP	1515	sg-0be305e637154660...	Agent enrollment

© 2026, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

FIGURE 7 – Wazuh-Server-SG : Inbound Rules (Ports 22, 443, 1514, 1515)

1.4.2 Wazuh Clients Security Group

The screenshot shows the AWS Cloud Console interface for managing security groups. The security group in question is named "Wazuh-Clients-SG" (sg-0be305e637154660d). The "Inbound rules" tab is selected, showing two entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-027d77ecde547f64c	IPv4	SSH	TCP	22	196.64.23.110/32	SSH for Linux client
-	sgr-0dd9fc06b8956aa9e	IPv4	RDP	TCP	3389	196.64.23.110/32	RDP for Windows client

FIGURE 8 – Wazuh-Clients-SG : Inbound Rules (SSH 22, RDP 3389)

1.5 Network Architecture Diagram

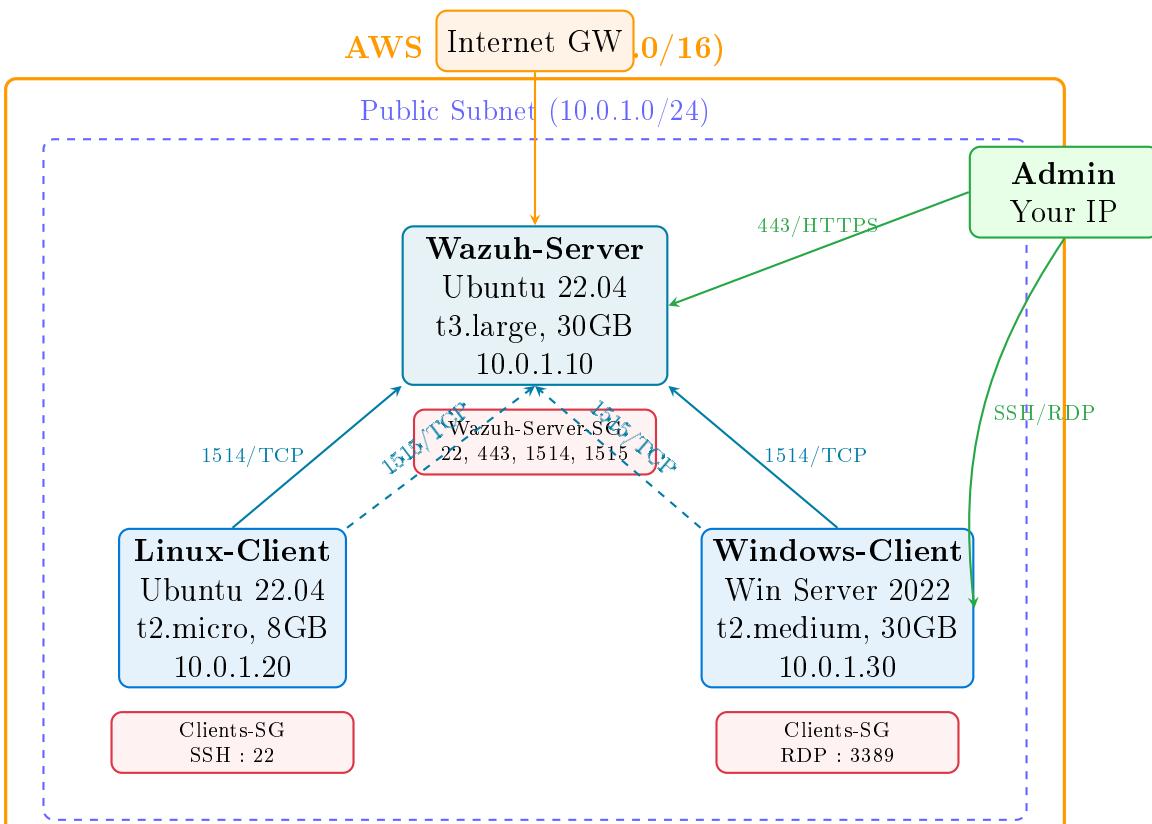
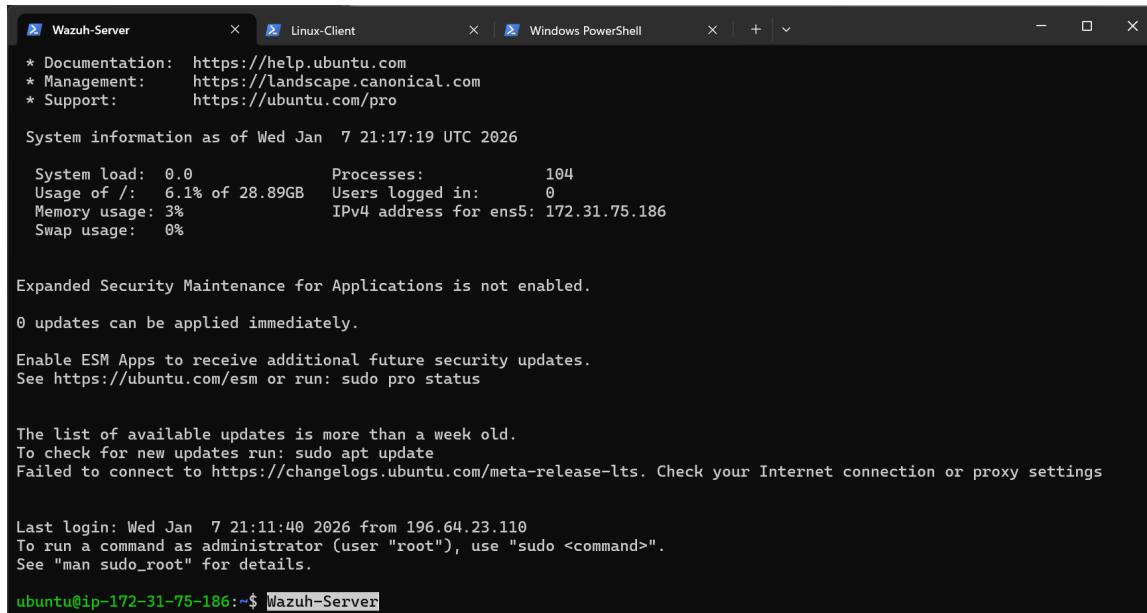


FIGURE 9 – Wazuh SIEM/EDR Lab Architecture Diagram

1.6 Connectivity Testing

1.6.1 SSH to Wazuh Server



```

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Wed Jan 7 21:17:19 UTC 2026

System load: 0.0 Processes: 104
Usage of /: 6.1% of 28.89GB Users logged in: 0
Memory usage: 3% IPv4 address for ens5: 172.31.75.186
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

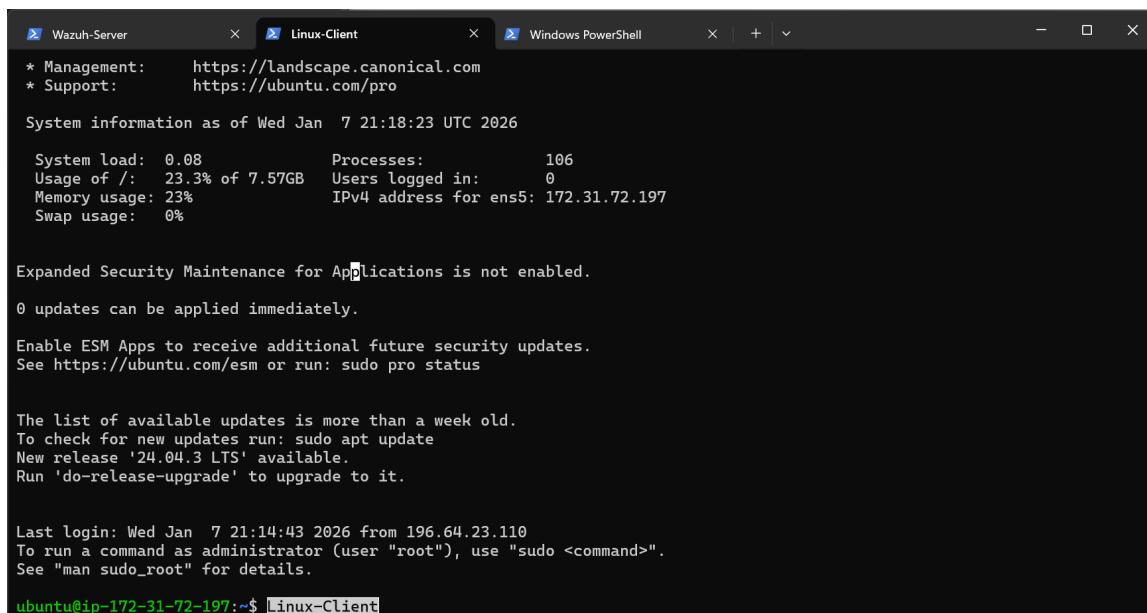
Last login: Wed Jan 7 21:11:40 2026 from 196.64.23.110
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-75-186:~$ Wazuh-Server

```

FIGURE 10 – SSH Connection Test to Wazuh-Server

1.6.2 SSH to Linux Client



```

* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Wed Jan 7 21:18:23 UTC 2026

System load: 0.08 Processes: 106
Usage of /: 23.3% of 7.57GB Users logged in: 0
Memory usage: 23% IPv4 address for ens5: 172.31.72.197
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Jan 7 21:14:43 2026 from 196.64.23.110
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-72-197:~$ Linux-Client

```

FIGURE 11 – SSH Connection Test to Linux-Client

1.6.3 RDP to Windows Client

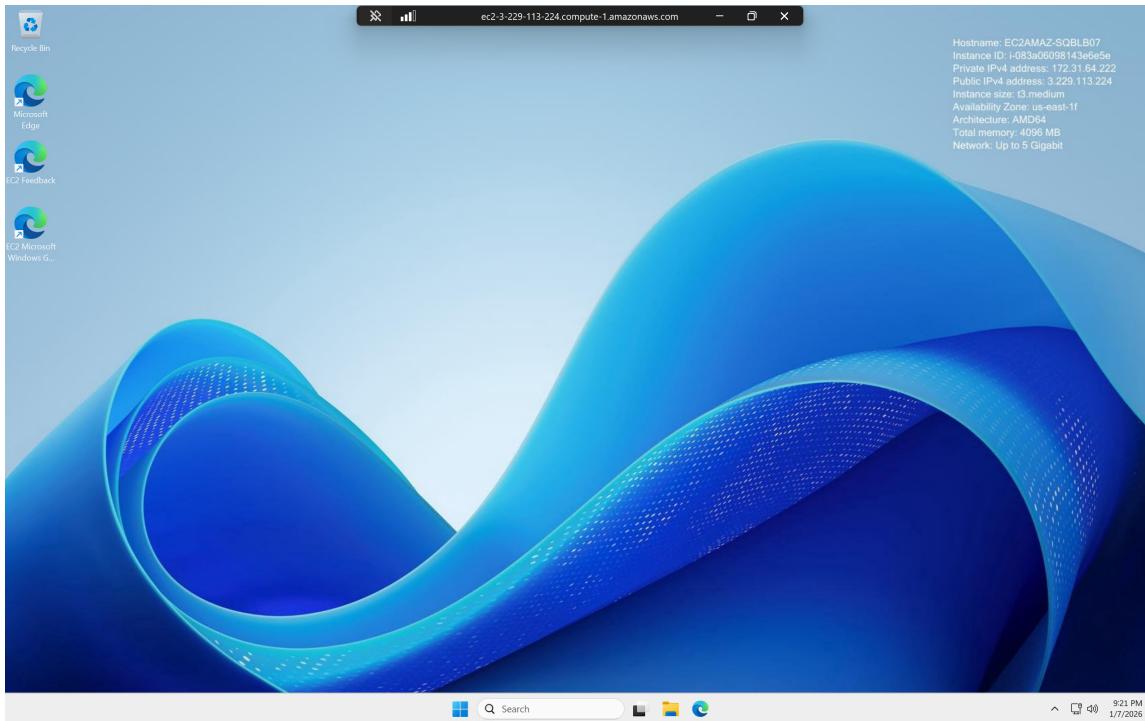


FIGURE 12 – RDP Connection Test to Windows-Client

1.7 Infrastructure Summary

Instance	Operating System	Type	Storage	Private IP
Wazuh-Server	Ubuntu 22.04 LTS	t3.large	30 GB	10.0.1.10
Linux-Client	Ubuntu 22.04 LTS	t2.micro	8 GB	10.0.1.20
Windows-Client	Windows Server 2022/2025	t2.medium	30 GB	10.0.1.30

TABLE 1 – EC2 Instance Specifications

Security Best Practices Implemented

- All agent-to-server communication uses private IP addresses within the VPC
- Management ports are restricted to specific administrator IP addresses
- Security groups follow the principle of least privilege
- Encrypted communication channels for all sensitive traffic
- Network segmentation isolates the monitoring infrastructure

2 Wazuh Server Setup and Dashboard

2.1 Wazuh Services Status

```
  Wazuh-Server x  Linux-Client x + 
ubuntu@ip-172-31-75-186:~$ sudo systemctl status wazuh-manager
sudo systemctl status wazuh-indexer
sudo systemctl status wazuh-dashboard
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2026-01-08 10:20:55 UTC; 1h 0min ago
       Tasks: 121 (limit: 9361)
      Memory: 376.4M
        CPU: 1min 9.567s
     CGroup: /system.slice/wazuh-manager.service
             └─47762 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py

Jan 08 10:20:46 ip-172-31-75-186 env[47293]: Started wazuh-db...
Jan 08 10:20:47 ip-172-31-75-186 env[47293]: Started wazuh-execd...
Jan 08 10:20:48 ip-172-31-75-186 env[47293]: Started wazuh-analysisd...
Jan 08 10:20:48 ip-172-31-75-186 env[47293]: Started wazuh-syscheckd...
Jan 08 10:20:49 ip-172-31-75-186 env[47293]: Started wazuh-remoted...
Jan 08 10:20:51 ip-172-31-75-186 env[47293]: Started wazuh-logcollector...
Jan 08 10:20:52 ip-172-31-75-186 env[47293]: Started Wazuh-monitord...
Jan 08 10:20:53 ip-172-31-75-186 env[47293]: Started wazuh-moduleds...
● wazuh-indexer.service - Wazuh indexer
   Loaded: loaded (/lib/systemd/system/wazuh-indexer.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2026-01-08 10:19:28 UTC; 1h 2min ago
       Docs: https://documentation.wazuh.com
 Main PID: 3892 (java)
    Tasks: 76 (limit: 9361)
   Memory: 4.2G
        CPU: 1min 58.392s
     CGroup: /system.slice/wazuh-indexer.service
             └─3892 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.networkaddress.cache.ttl=60 -Dopensearch.

Jan 08 10:19:06 ip-172-31-75-186 systemd[1]: Starting Wazuh-indexer...
Jan 08 10:19:10 ip-172-31-75-186 systemd--entrypoint[3892]: WARNING: A terminally deprecated method in java.lang.System >
Jan 08 10:19:10 ip-172-31-75-186 systemd--entrypoint[3892]: WARNING: System::setSecurityManager has been called by org.o>
Jan 08 10:19:10 ip-172-31-75-186 systemd--entrypoint[3892]: WARNING: Please consider reporting this to the maintainers >
Jan 08 10:19:10 ip-172-31-75-186 systemd--entrypoint[3892]: WARNING: System::setSecurityManager will be removed in a fut>
Jan 08 10:19:12 ip-172-31-75-186 systemd--entrypoint[3892]: WARNING: A terminally deprecated method in java.lang.System >
Jan 08 10:19:12 ip-172-31-75-186 systemd--entrypoint[3892]: WARNING: System::setSecurityManager has been called by org.o>
```

FIGURE 13 – Wazuh Services Running on Server

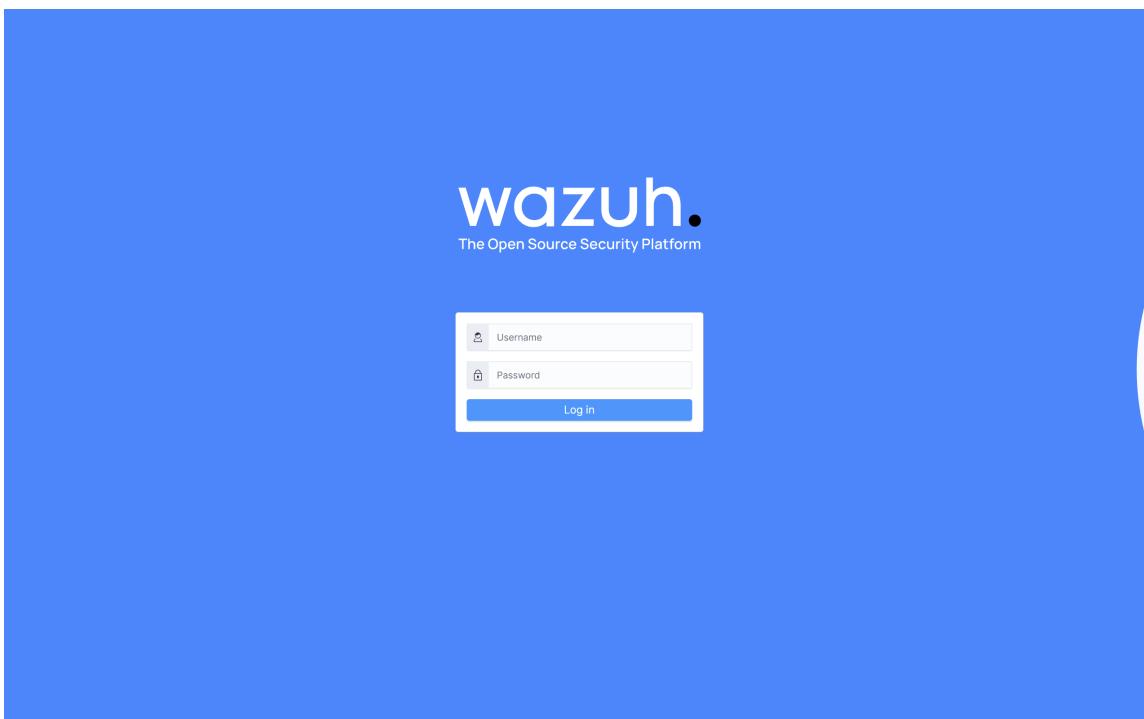


FIGURE 14 – Wazuh Services Status Verification

2.2 Wazuh Dashboard Access

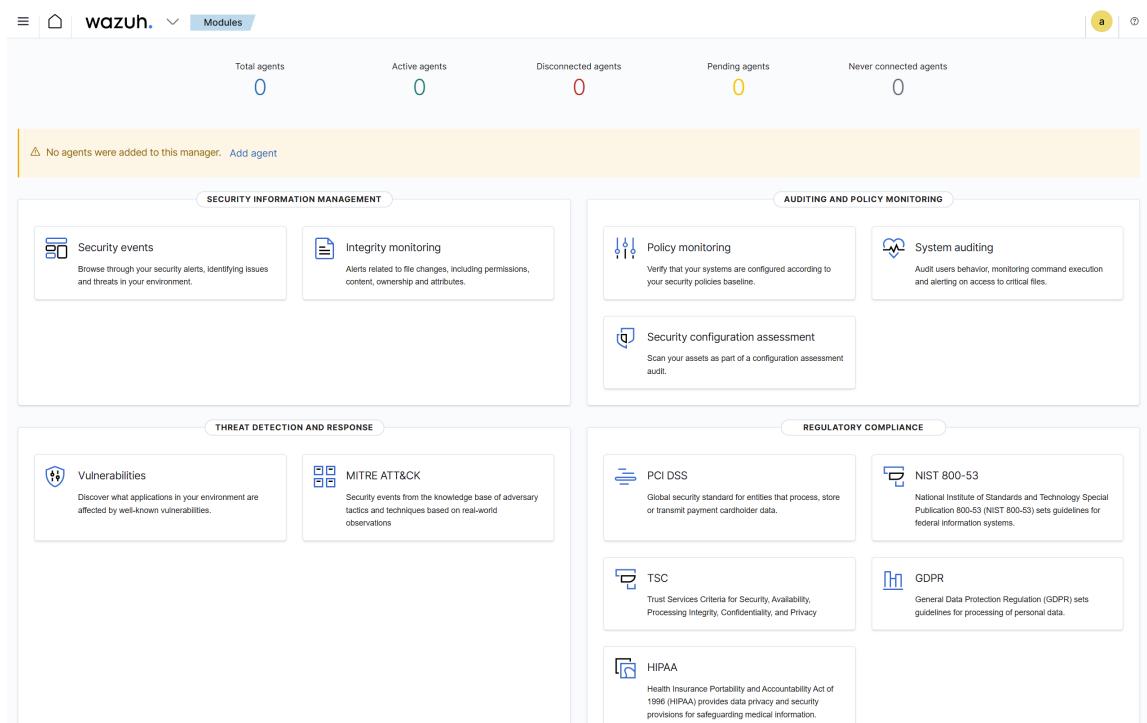


FIGURE 15 – Wazuh Dashboard Login Interface

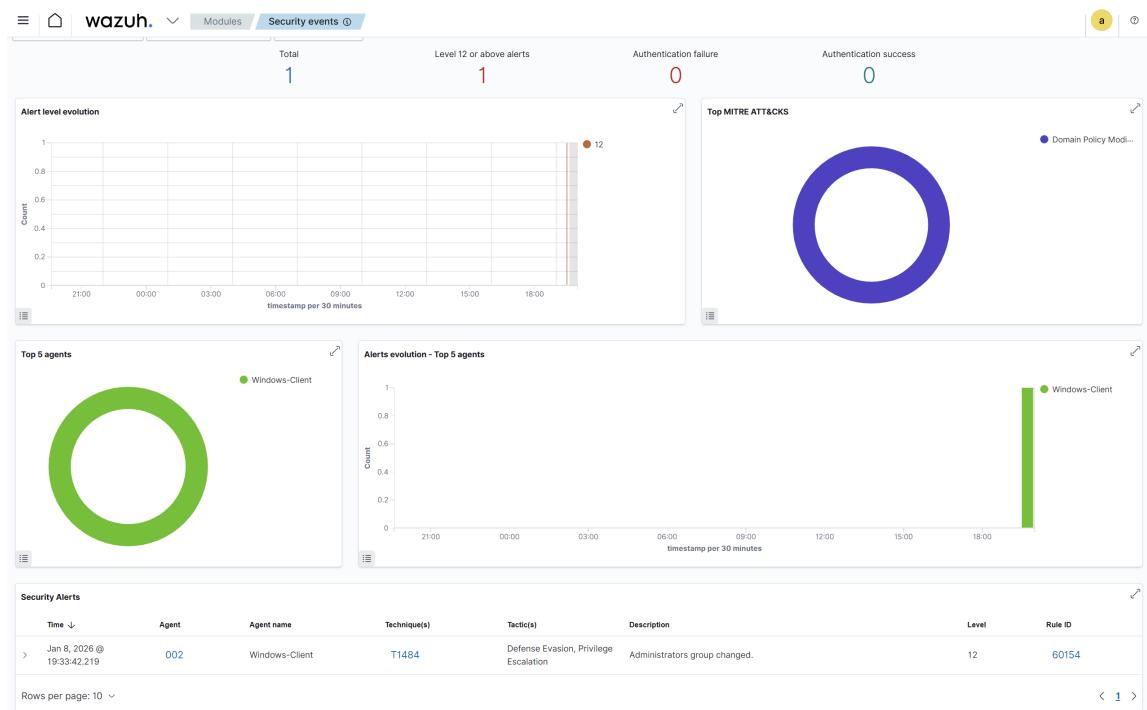


FIGURE 16 – Wazuh Dashboard Main Overview

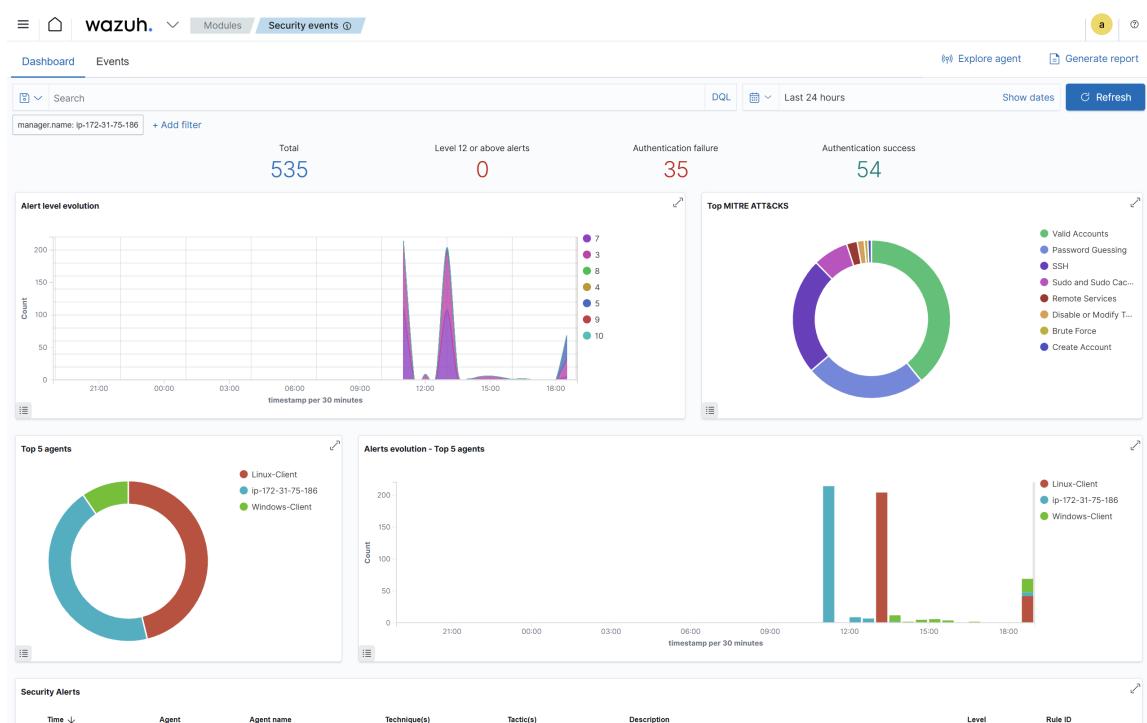
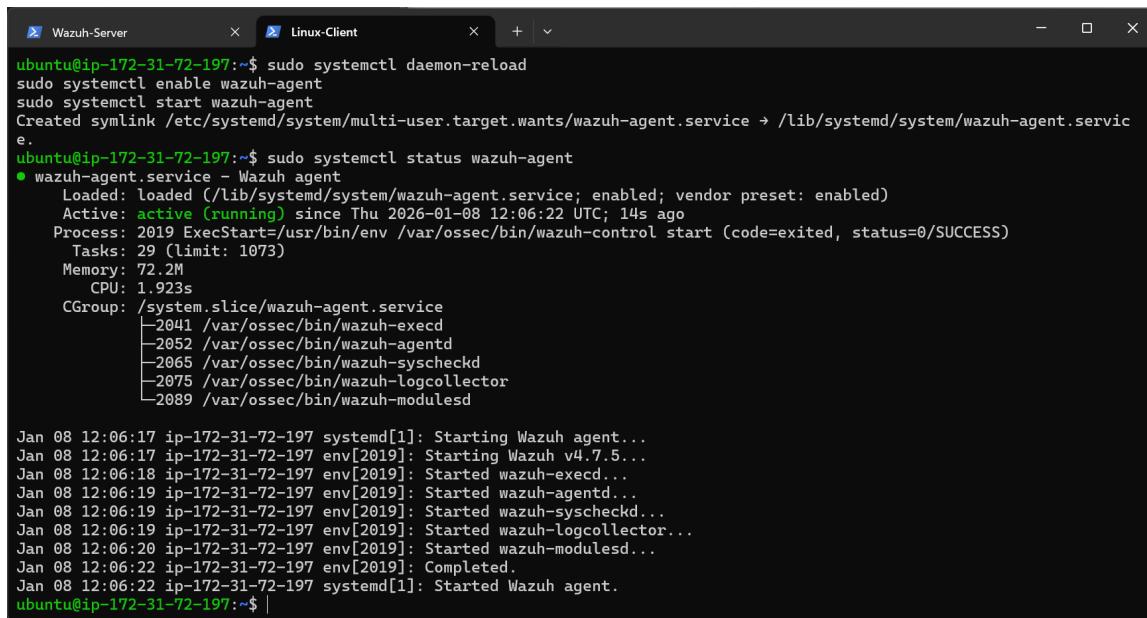


FIGURE 17 – Wazuh Security Events Dashboard

2.3 Agent Enrollment and Status

2.3.1 Linux Agent Connection



```

ubuntu@ip-172-31-72-197:~$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.service.

ubuntu@ip-172-31-72-197:~$ sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
  Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2026-01-08 12:06:22 UTC; 14s ago
    Process: 2019 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   Tasks: 29 (limit: 1073)
  Memory: 72.2M
     CPU: 1.923s
    CGroup: /system.slice/wazuh-agent.service
            └─2041 /var/ossec/bin/wazuh-execd
              ├─2052 /var/ossec/bin/wazuh-agentd
              ├─2065 /var/ossec/bin/wazuh-syscheckd
              ├─2075 /var/ossec/bin/wazuh-logcollector
              └─2089 /var/ossec/bin/wazuh-modulesd

Jan 08 12:06:17 ip-172-31-72-197 systemd[1]: Starting Wazuh agent...
Jan 08 12:06:17 ip-172-31-72-197 env[2019]: Starting Wazuh v4.7.5...
Jan 08 12:06:18 ip-172-31-72-197 env[2019]: Started wazuh-execd...
Jan 08 12:06:19 ip-172-31-72-197 env[2019]: Started wazuh-agentd...
Jan 08 12:06:19 ip-172-31-72-197 env[2019]: Started wazuh-syscheckd...
Jan 08 12:06:19 ip-172-31-72-197 env[2019]: Started wazuh-logcollector...
Jan 08 12:06:20 ip-172-31-72-197 env[2019]: Started wazuh-modulesd...
Jan 08 12:06:22 ip-172-31-72-197 env[2019]: Completed.
Jan 08 12:06:22 ip-172-31-72-197 systemd[1]: Started Wazuh agent.
ubuntu@ip-172-31-72-197:~$ |

```

FIGURE 18 – Linux Agent Running and Connected

2.3.2 Windows Agent Connection

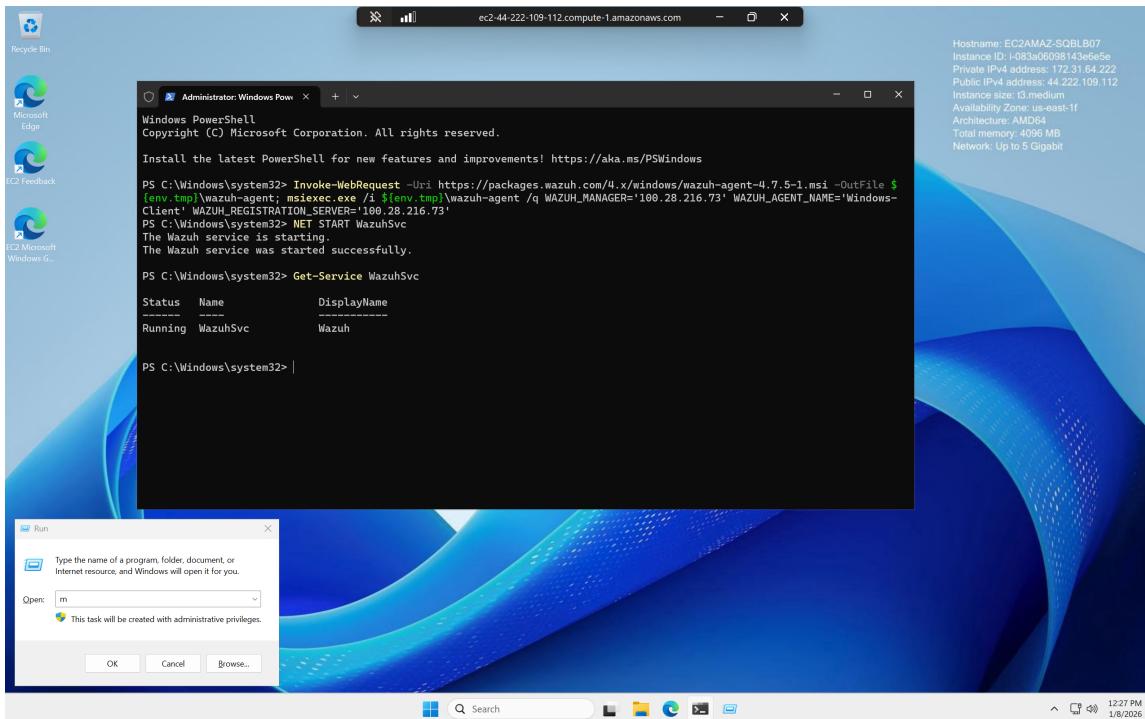


FIGURE 19 – Windows Agent Connecting to Wazuh Server

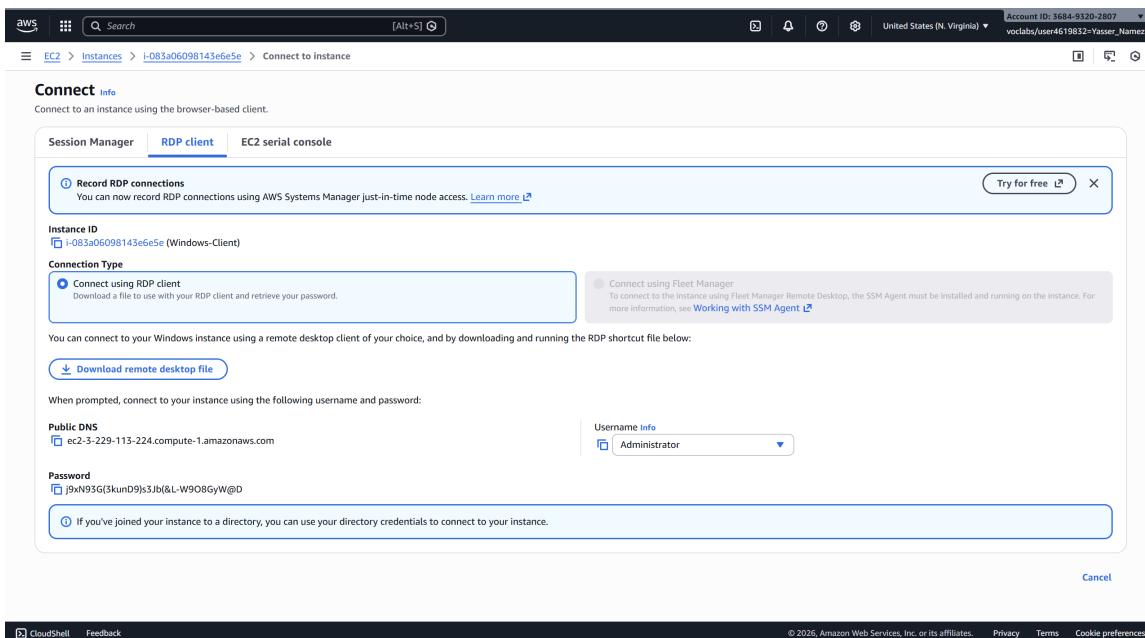


FIGURE 20 – Windows Client Agent Connection Status

2.4 Active Agents Overview

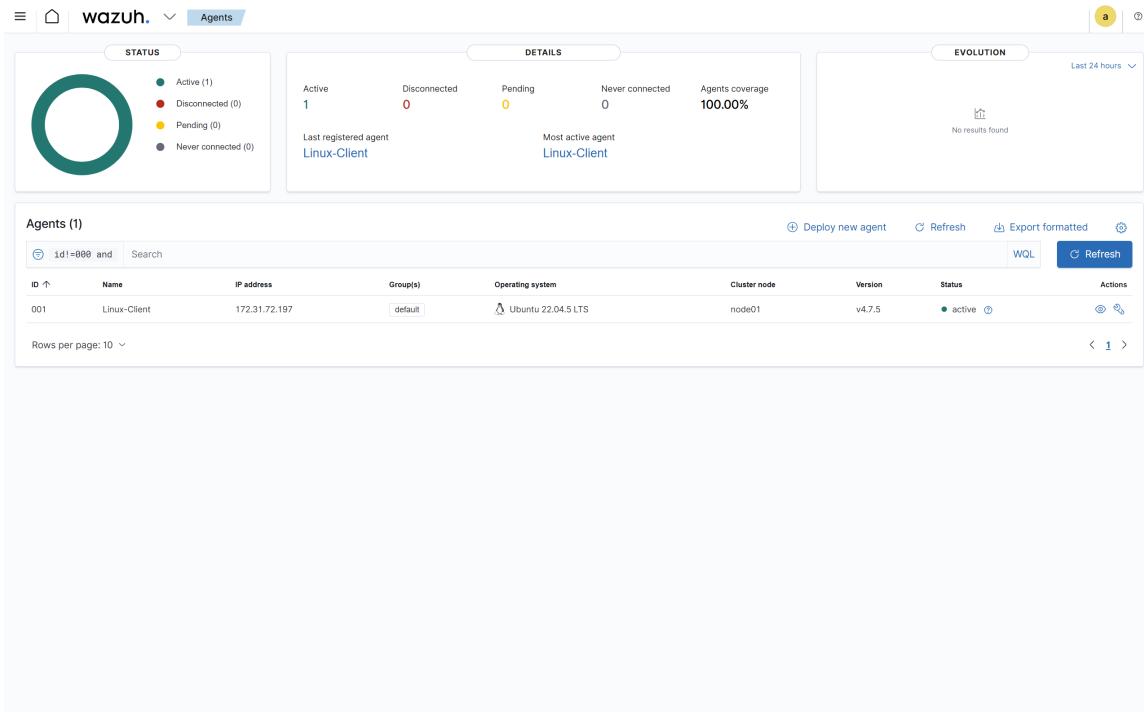


FIGURE 21 – Active Agents Dashboard View 1

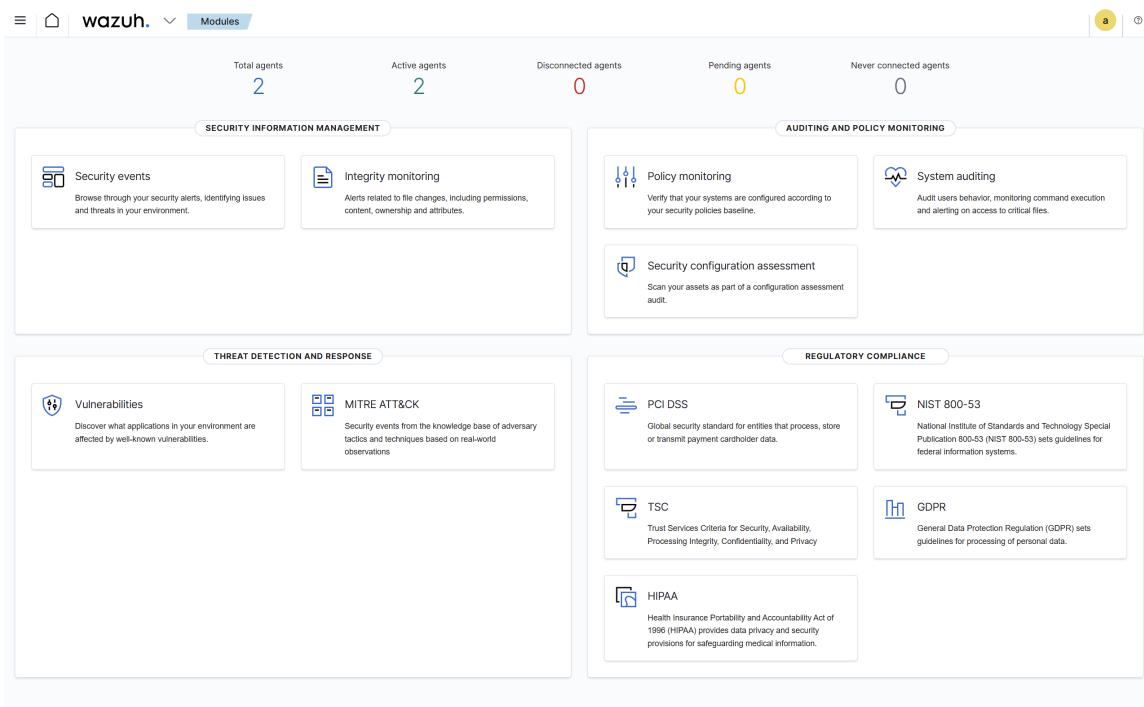


FIGURE 22 – Active Agents Dashboard View 2 - Both Agents Connected

Agent Name	IP Address	OS	Status	Version	
Linux-Client	10.0.1.20	Ubuntu 22.04	Active	4.x	
Windows-Client	10.0.1.30	Windows Server 2022	Active	4.x	

TABLE 2 – Connected Agents Summary

3 Security Events - Linux Client

This section documents the security alerts captured from the Linux client (Ubuntu 22.04) demonstrating Wazuh's detection capabilities for common attack patterns and security events.

3.1 Alert 1 : SSH Brute Force Attack

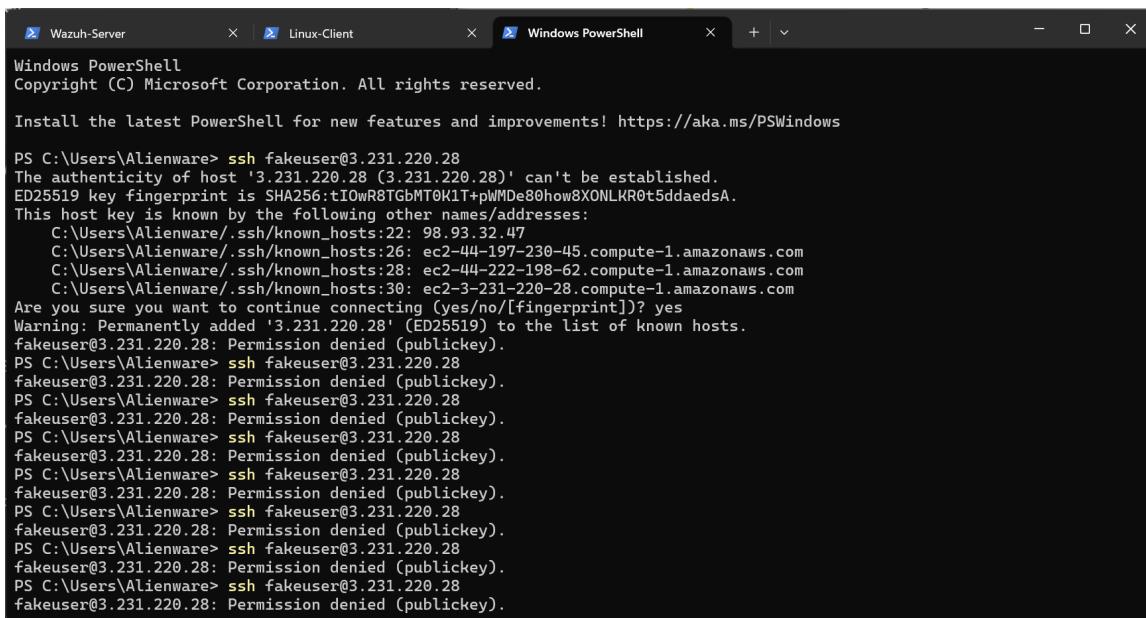
SSH Authentication Failure Detection

Rule IDs : 5710, 5712, 5720 (Progressive Severity)

Severity Level : 5-10 (Medium to High)

MITRE ATT&CK : T1110 - Brute Force

3.1.1 Attack Simulation



```

Wazuh-Server | Linux-Client | Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Alienware> ssh fakeuser@3.231.220.28
The authenticity of host '3.231.220.28 (3.231.220.28)' can't be established.
ED25519 key fingerprint is SHA256:tI0wR8TGBMT0K1T+pWMDe80how8XONLKR0t5ddae
This host key is known by the following other names/addresses:
  C:\Users\Alienware/.ssh/known_hosts:22: 98.93.32.47
  C:\Users\Alienware/.ssh/known_hosts:26: ec2-44-197-230-45.compute-1.amazonaws.com
  C:\Users\Alienware/.ssh/known_hosts:28: ec2-44-222-198-62.compute-1.amazonaws.com
  C:\Users\Alienware/.ssh/known_hosts:30: ec2-3-231-220-28.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.231.220.28' (ED25519) to the list of known hosts.
fakeuser@3.231.220.28: Permission denied (publickey).
PS C:\Users\Alienware> ssh fakeuser@3.231.220.28
fakeuser@3.231.220.28: Permission denied (publickey).

```

FIGURE 23 – SSH Brute Force Attack Simulation in Terminal

3.1.2 Wazuh Detection

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jan 8, 2026 @ 18:53:05.428	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:05.428	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:03.426	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:03.426	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:01.585	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:01.426	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:01.424	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:53:01.424	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 18:52:59.421	001	Linux-Client	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710

FIGURE 24 – SSH Failed Authentication Events Detected by Wazuh

3.1.3 Event Details

Attribute	Value
Log Source	/var/log/auth.log
Event Type	SSH Authentication Failure
Detection Pattern	Multiple consecutive failed login attempts
Source IP	External attacker IP
Target User	root, admin, ubuntu (common usernames)

TABLE 3 – SSH Brute Force Alert Details

3.1.4 Security Implication

- **Risk :** Potential unauthorized access attempt via credential guessing
- **Recommendation :** Implement fail2ban, use SSH key authentication, disable root login
- **Response :** Block source IP at firewall, review successful authentications

3.2 Alert 2 : Privilege Escalation (Sudo)

Privilege Escalation Detection

Rule IDs : 5402, 5401

Severity Level : 3-5 (Low to Medium)

MITRE ATT&CK : T1548 - Abuse Elevation Control Mechanism

3.2.1 Wazuh Detection

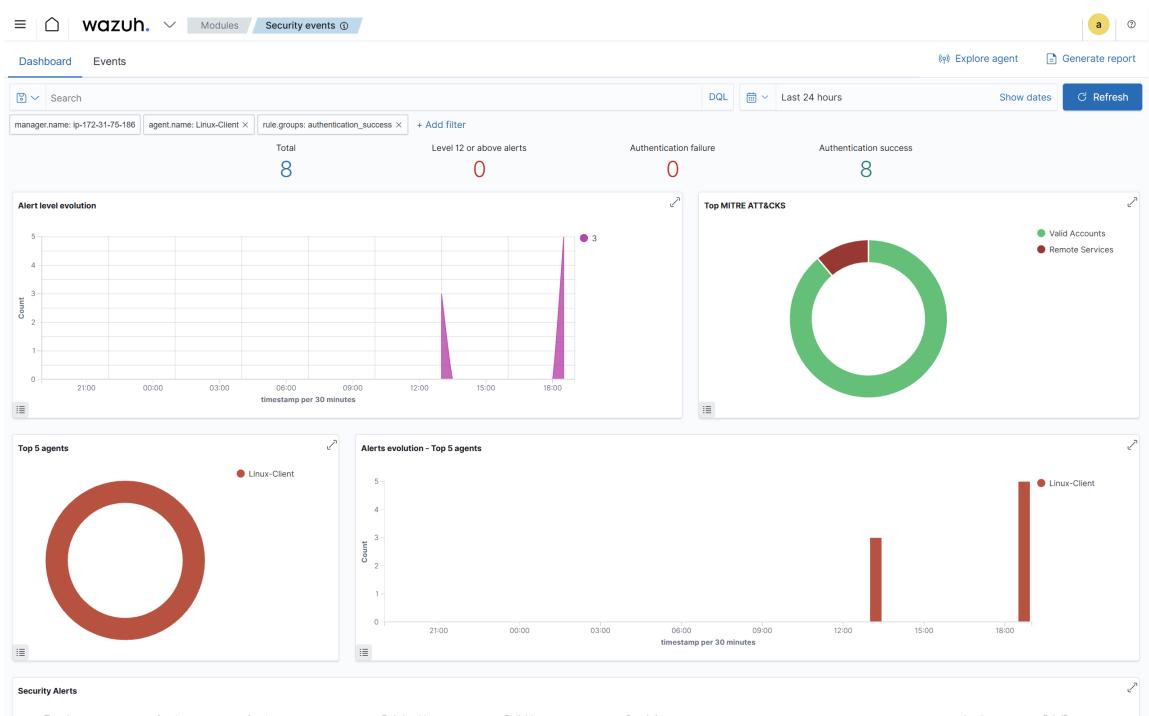
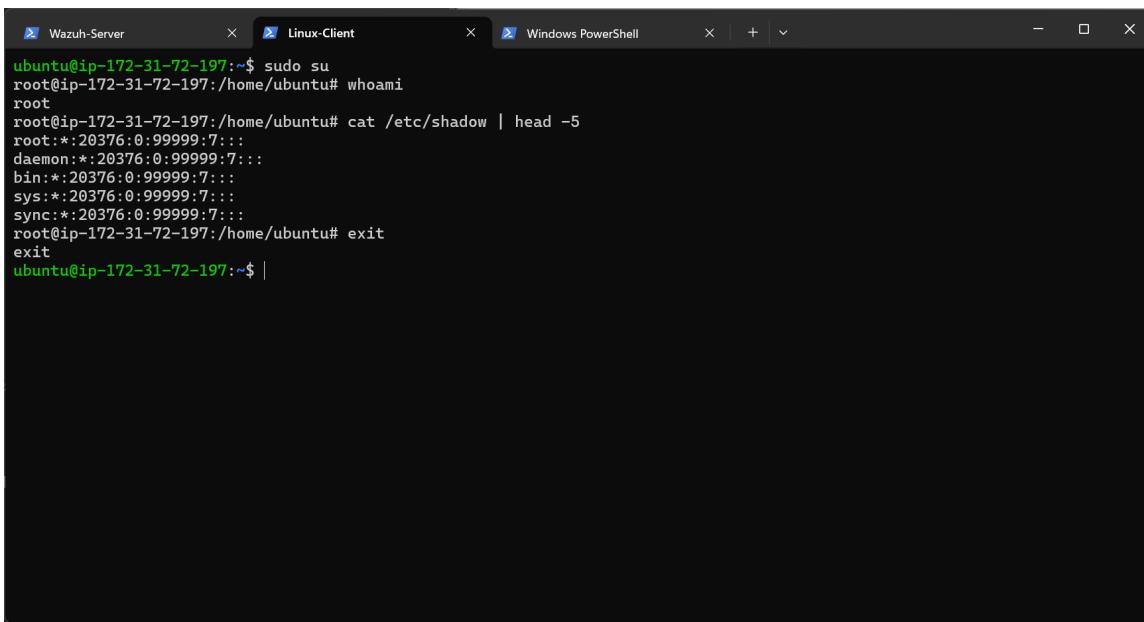


FIGURE 25 – Successful sudo to ROOT Execution Detected



The screenshot shows three terminal windows side-by-side:

- Wazuh-Server**: Shows a command-line interface with the output of a log entry.
- Linux-Client**: Shows a terminal session where the user runs `sudo su`, becomes root, and runs `cat /etc/shadow | head -5` to view password hashes.
- Windows PowerShell**: An empty terminal window.

FIGURE 26 – Privileged Root Access Event Details

3.2.2 Event Details

Attribute	Value
Log Source	/var/log/auth.log
Event Type	Successful sudo execution
User	ubuntu
Command	su (switch user to root)
Target User	ROOT

TABLE 4 – Privilege Escalation Alert Details

3.2.3 Security Implication

- **Context** : May be legitimate administrative activity
- **Monitoring Value** : Critical for detecting unauthorized privilege escalation
- **Best Practice** : Implement just-in-time privilege access, log all sudo commands

3.3 Alert 3 : File Integrity Monitoring (FIM)

File Integrity Change Detection

Rule IDs : 550, 553

Severity Level : 7 (High)

MITRE ATT&CK : T1078 - Valid Accounts, T1136 - Create Account

3.3.1 Event Details

Attribute	Value
Detection Module	Syscheck (File Integrity Monitoring)
Monitored File	/etc/passwd
Change Type	Modified
Detection Method	Checksum comparison (MD5, SHA256)
Previous Hash	[Original file hash]
New Hash	[Modified file hash]

TABLE 5 – FIM Alert Details

3.3.2 Security Implication

- **Risk** : Critical system file modification may indicate :
 - Unauthorized account creation
 - Account manipulation by attacker
 - Potential system compromise
- **Response** : Immediately investigate file changes, compare with known-good baseline
- **Forensics** : Review file modification timestamps, correlate with authentication logs

4 Security Events - Windows Client

This section documents the security alerts captured from the Windows client (Windows Server 2022) demonstrating Wazuh's detection capabilities for Windows-specific security events, including enhanced EDR telemetry from Sysmon integration.

4.1 Alert 1 : Failed Logon Attempts

Windows Authentication Failure Detection

Windows Event ID : 4625
Wazuh Rule ID : 60122
Severity Level : 5 (Medium)
MITRE ATT&CK : T1110 - Brute Force

4.1.1 Wazuh Detection

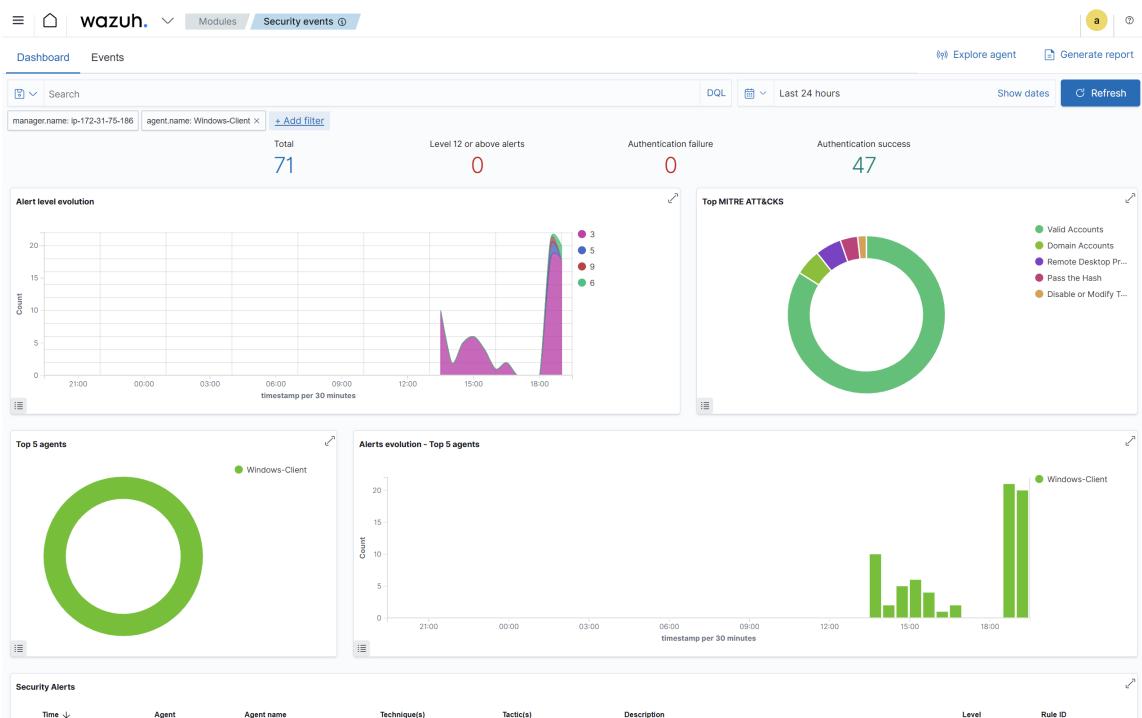


FIGURE 27 – Windows Failed Authentication Events in Wazuh Dashboard

4.1.2 Event Details

Attribute	Value
Event Source	Security Event Log
Event ID	4625 (Failed Logon)
Target Username	FakeUser
Logon Type	10 (RemoteInteractive / RDP)
Failure Reason	Unknown user name or bad password
Source Network Address	Attacker IP

TABLE 6 – Windows Failed Logon Alert Details

4.1.3 Security Implication

- **Risk** : Potential RDP brute force attack targeting the system
- **Recommendation** : Enable account lockout policies, implement Network Level Authentication (NLA)
- **Response** : Block source IP, enable MFA for RDP access

4.2 Alert 2 : User Account Creation and Group Modification (IAM)

User Account and Group Modification Detection

Windows Event IDs : 4720 (User Created), 4732 (Group Modified)

Wazuh Rule IDs : 60103, 60137

Severity Level : 8-10 (High to Critical)

MITRE ATT&CK : T1136 - Create Account, T1078.003 - Local Accounts

4.2.1 Wazuh Detection

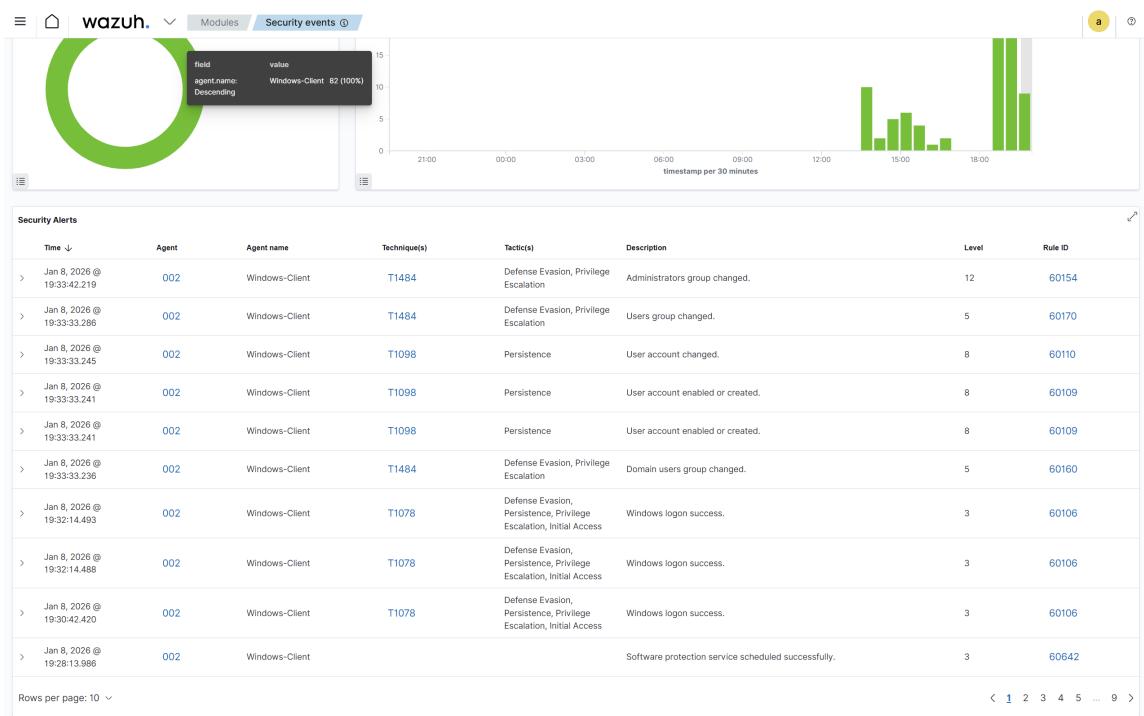


FIGURE 28 – User Account Creation and Security Group Modification Events

4.2.2 Event Details - User Creation (4720)

Attribute	Value
Event Source	Security Event Log
Event ID	4720 (User Account Created)
New Account Name	labuser
Created By	Administrator
Account Domain	WINDOWS-CLIENT

TABLE 7 – User Account Creation Alert Details

4.2.3 Event Details - Group Modification (4732)

Attribute	Value
Event Source	Security Event Log
Event ID	4732 (Member Added to Security-Enabled Local Group)
Target Group	Administrators
Member Added	labuser
Modified By	Administrator

TABLE 8 – Security Group Modification Alert Details

4.2.4 Security Implication

- **Monitoring Value** : Tracks all account creation and privilege changes
- **Risk** : Privilege escalation - new administrator account created
- **Response** : Validate authorization, review admin activity, check for lateral movement
- **Compliance** : Essential for SOX, PCI-DSS, HIPAA audit requirements

4.3 Alert 3 : Sysmon Process Creation (EDR)

Enhanced EDR Telemetry - Process Creation

Sysmon Event ID : 1 (Process Create)
Severity Level : Informational to High (context-dependent)
MITRE ATT&CK : Multiple (T1059, T1203, T1204)

4.3.1 Sysmon Events in Wazuh

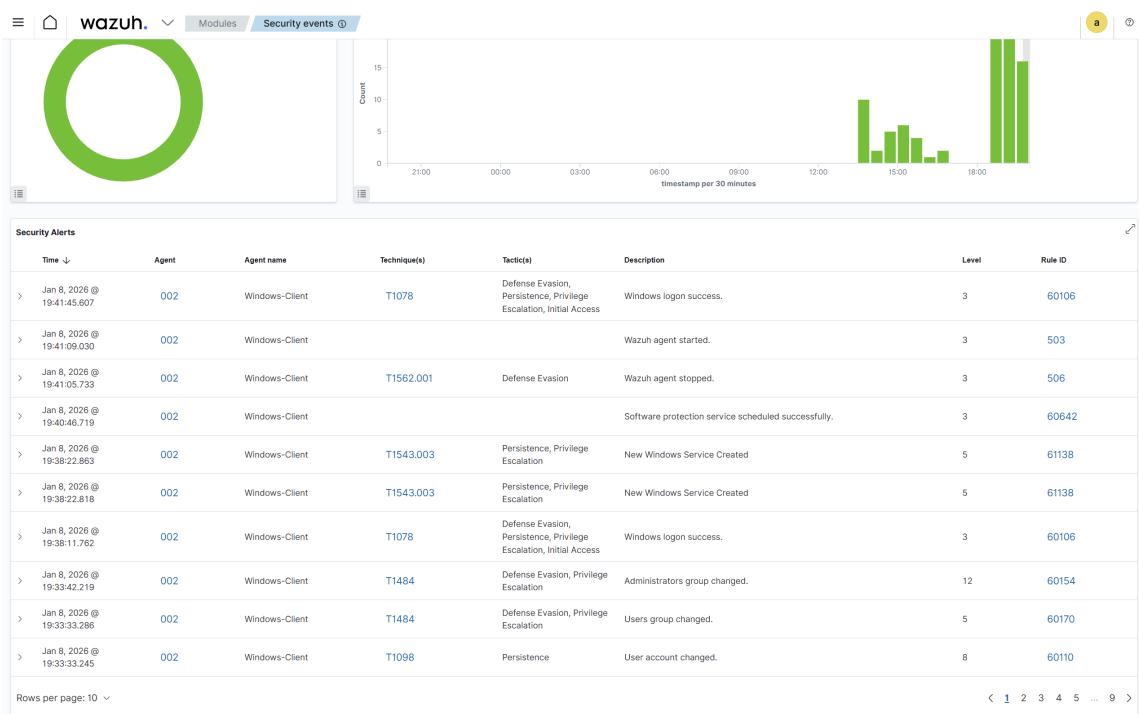


FIGURE 29 – Sysmon Events Overview in Wazuh Dashboard

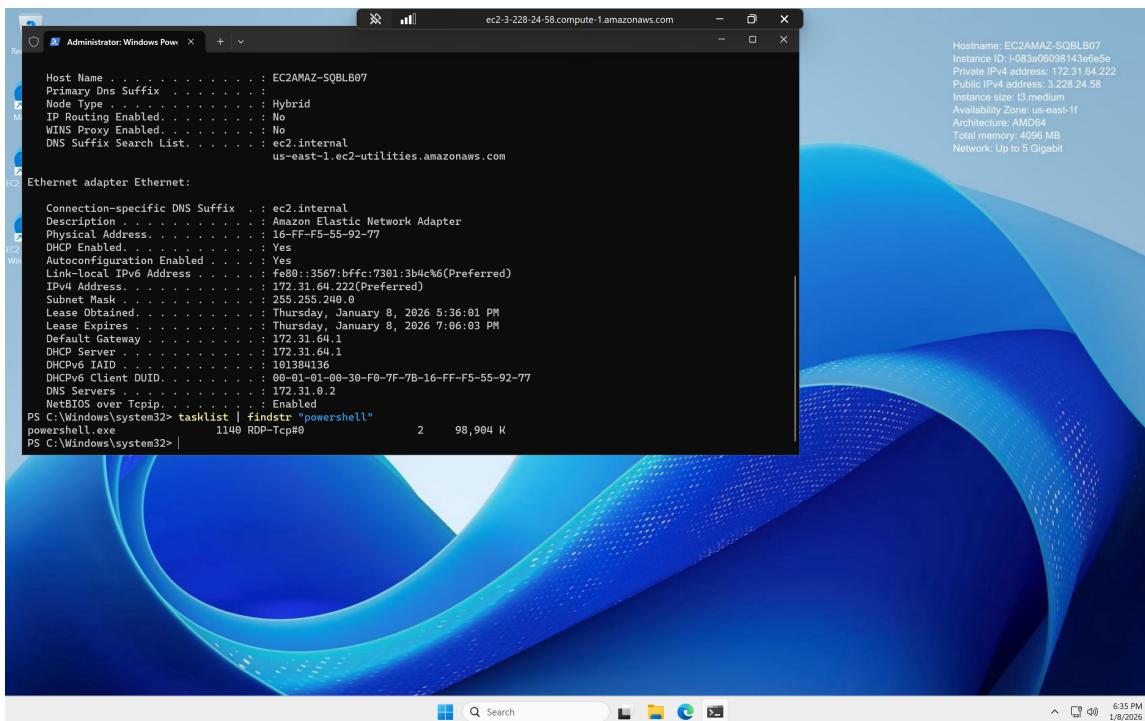


FIGURE 30 – Process Creation Events (Sysmon Event ID 1) Details

4.3.2 Event Details

Attribute	Value
Event Source	Microsoft-Windows-Sysmon/Operational
Event ID	1 (Process Create)
Image	Full executable path captured
CommandLine	Complete command-line arguments
ParentImage	Parent process that spawned this process
User	Execution context
Hashes	MD5, SHA256 file hashes

TABLE 9 – Sysmon Process Creation Event Details

4.3.3 EDR Value

- **Granular Visibility** : Full command-line arguments captured
- **Process Ancestry** : Parent-child process relationships tracked
- **Hash Collection** : File hashes enable threat intelligence correlation
- **Detection Capabilities** :
 - Living-off-the-land binary (LOLBIN) attacks
 - Malicious script execution
 - Lateral movement patterns
 - Fileless malware detection

5 SIEM vs EDR - Comparative Analysis

5.1 SIEM (Security Information and Event Management)

Definition

SIEM is a centralized platform for log aggregation, correlation, and analysis that provides holistic visibility across an organization's IT infrastructure.

5.1.1 Key Capabilities Demonstrated in Lab

- **Log Collection** : Aggregation from multiple sources (Linux auth.log, Windows Event Logs)
- **Real-time Correlation** : Event correlation and alerting across systems
- **Centralized Visibility** : Single dashboard for heterogeneous environments
- **Compliance Support** : Audit trails for regulatory requirements
- **Historical Analysis** : Long-term data retention for forensic investigations

5.1.2 Lab Use Cases

- SSH brute force detection through authentication log analysis
- Centralized monitoring of multi-OS environment
- Rule-based alerting on security events
- Cross-system correlation (e.g., authentication events across Linux and Windows)

5.1.3 Strengths & Limitations

Strengths	Limitations
Broad infrastructure visibility	Relyes on logs (may miss activity)
Compliance and audit support	Less granular endpoint visibility
Pattern recognition across systems	Reactive detection model
Long-term data retention	Requires proper log forwarding setup

TABLE 10 – SIEM Strengths and Limitations

5.2 EDR (Endpoint Detection and Response)

Definition

EDR is an agent-based solution providing deep endpoint visibility, behavioral analysis, and threat response capabilities at the process and file level.

5.2.1 Key Capabilities Demonstrated in Lab

- **Process Monitoring** : Sysmon Event ID 1 - detailed process creation tracking
- **Network Visibility** : Sysmon Event ID 3 - network connection monitoring
- **File Integrity** : FIM module detecting critical file changes
- **Command-line Visibility** : Full command-line argument capture
- **Process Ancestry** : Parent-child process relationship tracking
- **Hash Collection** : File hashes for threat intelligence correlation

5.2.2 Lab Use Cases

- Sysmon providing detailed process creation data with full command lines
- Process hash collection enabling malware analysis
- Behavioral analysis for detecting suspicious patterns
- Network connection monitoring for C2 detection

5.2.3 Strengths & Limitations

Strengths	Limitations
Deep endpoint visibility	Agent required on each endpoint
Behavioral detection capabilities	Higher resource consumption
Proactive threat hunting	Endpoint-focused (may miss network attacks)
Rich forensic data	Can generate high data volume

TABLE 11 – EDR Strengths and Limitations

5.3 Wazuh : Unified SIEM + EDR Platform

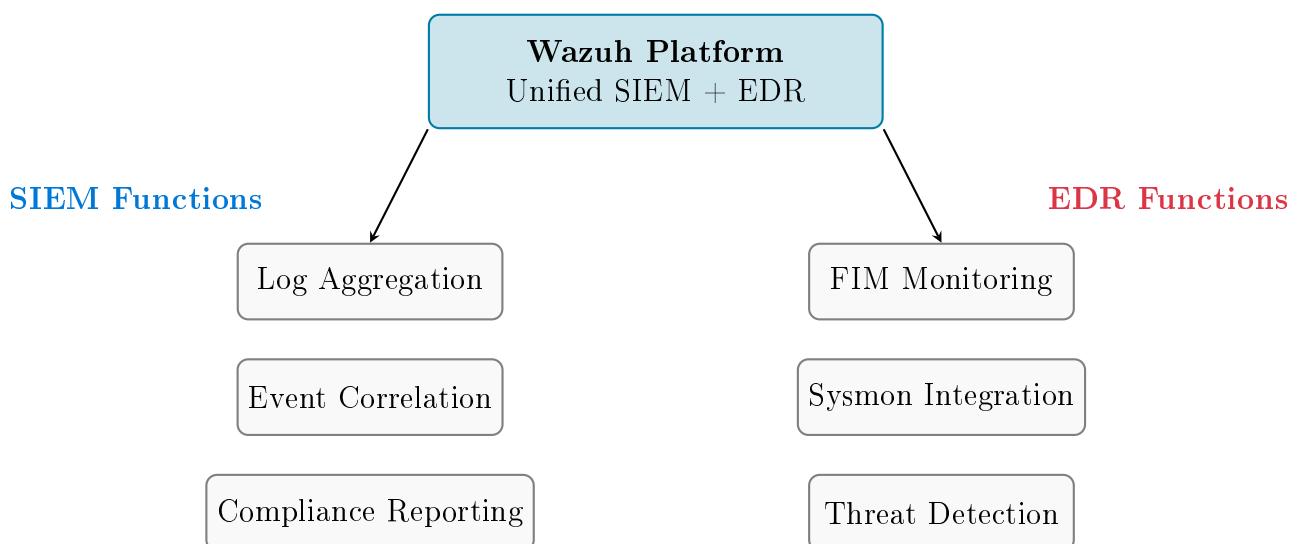


FIGURE 31 – Wazuh Unified SIEM + EDR Architecture

5.3.1 Demonstrated Integration Benefits

- **Single Pane of Glass** : Unified dashboard for all security operations
- **Cross-Domain Correlation** : Link endpoint activity with network/system logs
- **Reduced Tool Sprawl** : One platform instead of multiple point solutions
- **Cost-Effective** : Open-source solution with enterprise capabilities
- **FIM Bridge** : File Integrity Monitoring spans both SIEM and EDR domains
- **Enhanced Telemetry** : Sysmon integration provides deep EDR visibility

6 Identity and Access Management (IAM) & Privileged Access Management (PAM)

6.1 IAM Concepts Demonstrated

6.1.1 Authentication Monitoring

Event	ID	Security Value
Windows Failed Logon	4625	Track unauthorized access attempts
SSH Auth Failure	Rule 5710	Detect brute force attacks
Successful Auth	Various	Establish baseline behavior

TABLE 12 – Authentication Monitoring Events

6.1.2 Account Lifecycle Management

- **Windows Event 4720** : User account creation detected and logged
- **Account Modifications** : All changes tracked through audit logs
- **Compliance Value** : Supports joiner/mover/leaver processes

6.1.3 Authorization & Privilege Changes

- **Windows Event 4732** : Group membership changes monitored
- **Linux Sudo Tracking** : Full command history with user context
- **Detection** : Unauthorized privilege escalation attempts identified

6.1.4 Identity-Based Threat Detection

- Multiple failed logins indicate credential stuffing attempts
- Unusual authentication patterns signal compromised accounts
- Lateral movement detection through authentication correlation

6.2 PAM (Privileged Access Management) Relevance

6.2.1 Privileged Activity Monitoring

Activity	Monitoring Capability
Sudo Executions	Logged with timestamp, user, and full command
Windows Admin Actions	Tracked through security event logs
Root/SYSTEM Access	High-severity alerts generated

TABLE 13 – Privileged Activity Monitoring

6.2.2 Compliance Support

- **Least Privilege** : Visibility into who uses privileged accounts and when

- **Just-In-Time Access** : Logs validate that privileges used were authorized
- **Audit Trail** : Full documentation for SOX, PCI-DSS, HIPAA compliance
- **Anomaly Detection** : After-hours privileged access flagged

6.3 Lab Demonstration Summary

IAM/PAM Lab Activities

1. Created user "labuser" on Windows (Event 4720)
2. Added "labuser" to Administrators group (Event 4732)
3. Monitored sudo usage for privilege escalation (Rule 5402)
4. Tracked authentication attempts and failures across platforms
5. Established complete audit trail for identity-related activities

7 Threat Hunting - Sample Queries & Analysis

7.1 Query 1 : Failed Authentication Analysis

Objective

Identify potential brute force attacks or credential stuffing attempts across the infrastructure.

7.1.1 Wazuh Query

```
rule.groups:"authentication_failed" AND agent.name:"Linux-Client"
"
```

7.1.2 Analysis Approach

1. **Pattern Recognition** : Look for multiple failures from same source IP
2. **Timing Analysis** : Check intervals between attempts (automated vs manual)
3. **Correlation** : Link with successful authentication after failures
4. **User Enumeration** : Investigate which accounts are being targeted

7.1.3 Detection Indicators

- > 5 failures within 5 minutes = likely automated attack
- Failures for multiple usernames = scanning/enumeration
- Geographic anomalies = authentication from unusual locations

7.1.4 Response Actions

1. Block source IP at firewall/security group
2. Implement account lockout policies
3. Enable MFA for targeted accounts
4. Review for any successful authentications from same source

7.2 Query 2 : Privileged Account Activity

Objective

Monitor for unauthorized privilege escalation or suspicious administrative activity.

7.2.1 Wazuh Query

```
(rule.id:5402 OR data.win.eventdata.eventID:4672)
AND NOT user.name:"expected_admin"
```

7.2.2 Analysis Approach

1. **Identify** : All sudo/administrator usage
2. **Filter** : Exclude known legitimate administrators
3. **Timing** : Check for unusual timing (after-hours, weekends)
4. **Correlate** : Link with other suspicious activities

7.2.3 Detection Indicators

- Privilege escalation by non-admin users
- Service accounts used interactively
- Unusual commands with elevated privileges
- Privilege use from unexpected systems/IPs

7.2.4 Response Actions

1. Validate whether privilege use was authorized
2. Review command history for malicious activity
3. Disable compromised accounts immediately
4. Initiate incident response if unauthorized

7.3 Query 3 : Process Anomaly Detection (EDR)

Objective

Identify suspicious process execution patterns indicating malware or living-off-the-land attacks.

7.3.1 Wazuh Query

```
data.win.system.eventID:1 AND
(data.win.eventdata.commandLine:(*powershell* AND *-enc*) OR
 data.win.eventdata.commandLine:(*cmd* AND */c* AND *certutil*))
```

7.3.2 Analysis Approach

1. **Encoded Commands** : Look for encoded PowerShell (common in malware)
2. **Process Relationships** : Identify unusual parent-child patterns
3. **LOLBins** : Check for system binaries used maliciously
4. **Network Correlation** : Link process creation with network connections (Sysmon Event 3)

7.3.3 Detection Indicators

- PowerShell with encoded commands (-enc, -EncodedCommand)
- cmd.exe spawning unusual children (certutil, bitsadmin, wmic)
- Processes running from temporary directories
- Suspicious parent processes (e.g., Excel spawning PowerShell)
- Processes with network connections to unknown IPs

7.3.4 Response Actions

1. Isolate affected endpoint immediately
2. Collect process memory dump for analysis
3. Check file hashes against threat intelligence
4. Review network logs for C2 communication
5. Initiate full forensic investigation

7.4 Threat Hunting Best Practices

Practice	Description
Time Correlation	Look for events within short timeframes
Baseline Normal	Understand typical activity before hunting anomalies
Pivot on Indicators	One suspicious event often leads to others
Document Findings	Maintain a hunting journal with queries and results
Continuous Improvement	Refine queries based on findings and new threats

TABLE 14 – Threat Hunting Best Practices

8 Conclusion

This lab successfully demonstrated the implementation of a cloud-based Wazuh SIEM/EDR platform on AWS infrastructure. Key accomplishments include :

8.1 Technical Achievements

- Deployed a complete monitoring infrastructure with proper network segmentation
- Successfully enrolled and monitored both Linux and Windows agents
- Captured and analyzed real security events including :
 - SSH brute force attacks
 - Privilege escalation activities
 - File integrity changes
 - Windows authentication events
 - Process-level EDR telemetry via Sysmon

8.2 Security Concepts Validated

- **SIEM Capabilities** : Centralized log aggregation, correlation, and alerting
- **EDR Capabilities** : Deep endpoint visibility with process and file monitoring
- **IAM/PAM** : Identity and privilege monitoring for compliance and security
- **Threat Hunting** : Practical queries for proactive threat detection

8.3 Operational Value

Wazuh provides a cost-effective, open-source solution that combines SIEM and EDR capabilities in a unified platform. This lab demonstrates its applicability for :

- Small to medium enterprise security monitoring
- Cloud-native security operations
- Compliance and audit requirements
- Security Operations Center (SOC) implementations

Key Takeaway

The integration of SIEM and EDR capabilities in a single platform like Wazuh provides comprehensive security visibility while reducing operational complexity and cost—essential for modern security operations in cloud environments.