VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION SYSTEMS

SIEM - WAZUH

Information Technology Security Methods

Prepared by: Simonas Riška

Checked by: lect. [redacted]

VILNIUS 2024

First of all, I downloaded the Wazuh installation script using curl from the official Wazuh repository. After verifying the script was successfully saved as wazuh-install.sh, I prepared to execute it with administrator privileges by running the command sudo bash wazuh-install.sh - this command is used to install and configure Wazuh components on the system:

```
siemadmin@siemadmin:~$ curl -s0 https://packages.wazuh.com/4.9/wazuh-install.sh -o wazuh-install.sh
siemadmin@siemadmin:~$ ls
wazuh-install.sh
siemadmin@siemadmin:~$ sudo bash wazuh-install.sh -a_
```

After running the Wazuh installation script, I successfully installed all necessary components, including the Wazuh indexer, manager, and dashboard. The installation completed with the credentials provided to access the Wazuh web interface:

```
Wazuh Server [Running] - Oracle VirtualBox
File  Machine  View  Input  Devices  Help
02/12/2024 21:30:19 INFO: Wazuh web interface port will be 443.
02/12/2024 21:30:23 INFO: --- Dependencies ----
02/12/2024 21:30:23 INFO: Installing apt-transport-https.
02/12/2024 21:30:26 INFO: Installing debhelper.
02/12/2024 21:30:41 INFO: Wazuh repository added.
02/12/2024 21:30:41 INFO: --- Configuration files ---
02/12/2024 21:30:41 INFO: Generating configuration files.
02/12/2024 21:30:41 INFO: Generating the root certificate.
02/12/2024 21:30:42 INFO: Generating Admin certificates.
02/12/2024 21:30:42 INFO: Generating Wazuh indexer certificates.
02/12/2024 21:30:42 INFO: Generating Filebeat certificates.
02/12/2024 21:30:42 INFO: Generating Wazuh dashboard certificates.
02/12/2024 21:30:43 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
02/12/2024 21:30:43 INFO: --- Wazuh indexer ---
02/12/2024 21:30:43 INFO: Starting Wazuh indexer installation.
02/12/2024 21:32:05 INFO: Wazuh indexer installation finished.
02/12/2024 21:32:05 INFO: Wazuh indexer post-install configuration finished.
02/12/2024 21:32:05 INFO: Starting service wazuh-indexer.
02/12/2024 21:32:18 INFO: wazuh-indexer service started.
02/12/2024 21:32:18 INFO: Initializing Wazuh indexer cluster security settings.
02/12/2024 21:32:22 INFO: Wazuh indexer cluster security configuration initialized.
02/12/2024 21:32:22 INFO: Wazuh indexer cluster initialized.
02/12/2024 21:32:22 INFO: --- Wazuh server ---
02/12/2024 21:32:22 INFO: Starting the Wazuh manager installation.
02/12/2024 21:33:36 INFO: Wazuh manager installation finished.
02/12/2024 21:33:36 INFO: Wazuh manager vulnerability detection configuration finished.
02/12/2024 21:33:36 INFO: Starting service wazuh-manager.
02/12/2024 21:33:52 INFO: wazuh-manager service started.
02/12/2024 21:33:52 INFO: Starting Filebeat installation.
02/12/2024 21:34:00 INFO: Filebeat installation finished.
02/12/2024 21:34:00 INFO: Filebeat post-install configuration finished.
02/12/2024 21:34:00 INFO: Starting service filebeat.
02/12/2024 21:34:02 INFO: filebeat service started.
02/12/2024 21:34:02 INFO: --- Wazuh dashboard ---
02/12/2024 21:34:02 INFO: Starting Wazuh dashboard installation.
02/12/2024 21:35:10 INFO: Wazuh dashboard installation finished.
02/12/2024 21:35:10 INFO: Wazuh dashboard post-install configuration finished.
02/12/2024 21:35:10 INFO: Starting service wazuh-dashboard.
02/12/2024 21:35:11 INFO: wazuh-dashboard service started.
02/12/2024 21:35:12 INFO: Updating the internal users.
02/12/2024 21:35:17 INFO: A backup of the internal users has been saved in the /etc/wazuh-indexer/internalusers-backup folder.
02/12/2024 21:35:26 INFO: The filebeat.yml file has been updated to use the Filebeat Keystore username and password.
02/12/2024 21:37:07 INFO: Initializing Wazuh dashboard web application.
02/12/2024 21:37:08 INFO: Wazuh dashboard web application initialized.
02/12/2024 21:37:08 INFO: --- Summary ---
02/12/2024 21:37:08 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: akU4?mnyNR7v3gpW5+3wey4qJlio67QL
02/12/2024 21:37:08 INFO: Installation finished.
wazuhserver@wazuhserver:~$
Right Ctrl
```

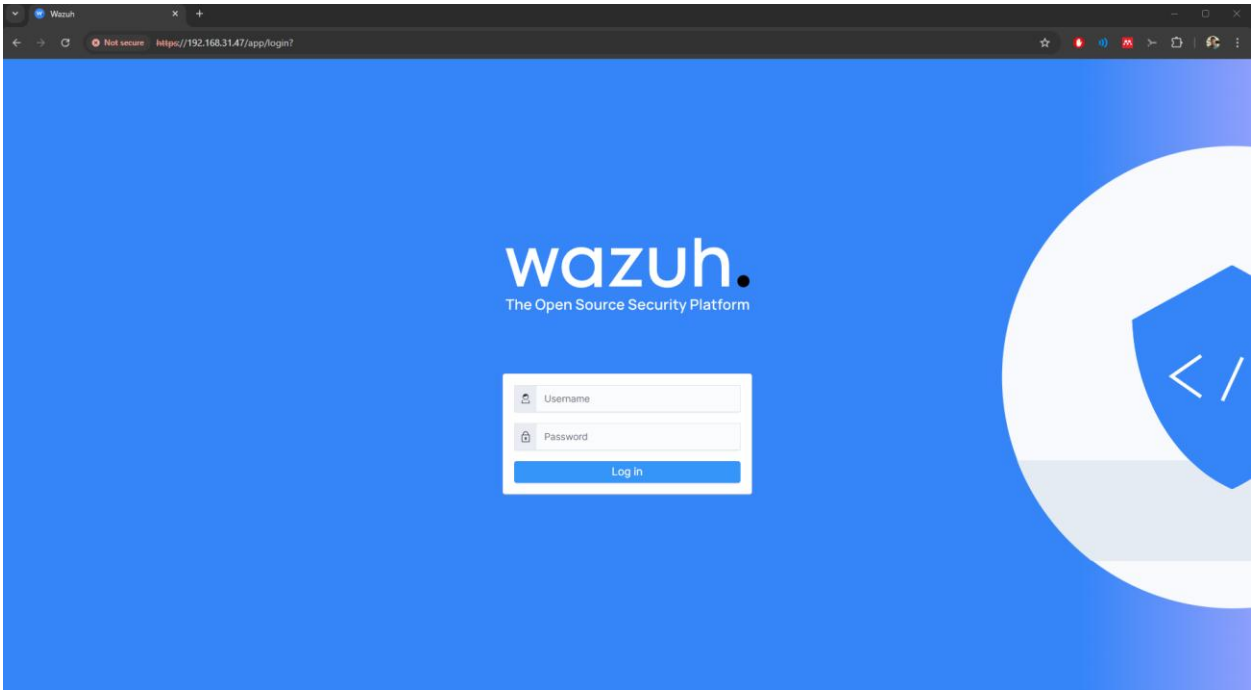
Here it shows how can I access dashboard with user admin and password akU4?mnyNR7v3gpW5+3wey4qJlio67QL:

```
02/12/2024 21:37:08 INFO: --- Summary ---
02/12/2024 21:37:08 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: akU4?mnyNR7v3gpW5+3wey4qJlio67QL
02/12/2024 21:37:08 INFO: Installation finished.
```

I checked my ip address for wazuh dashboard ip with ip addr command:

```
wazuhserver@wazuhserver:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:0d:40:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.47/24 metric 100 brd 192.168.31.255 scope global dynamic enp0s3
       valid_lft 43165sec preferred_lft 43165sec
    inet6 fe80::a00:27ff:fe0d:405e/64 scope link
       valid_lft forever preferred_lft forever
```

I went to this ip address and I accessed wazuh dashboard:



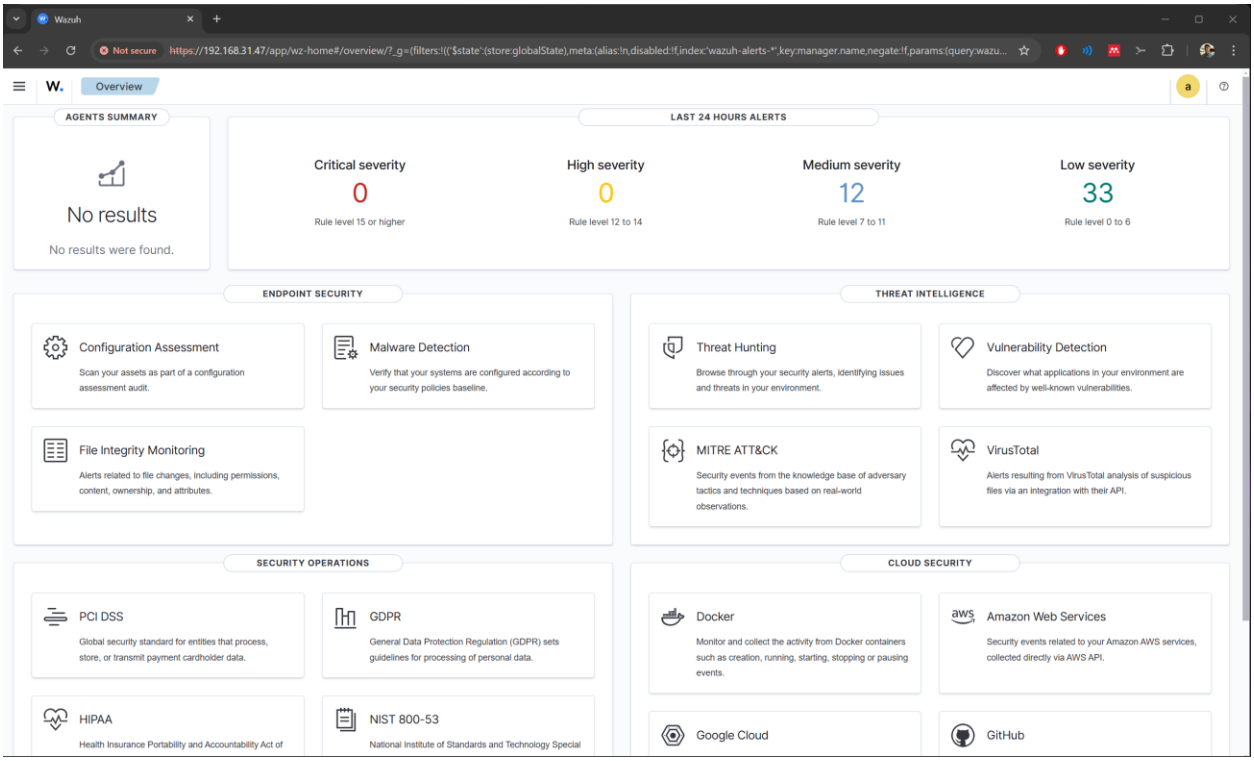
Then I logged in. There was no agents right now as I haven't added any, just a few medium/low severity alerts (like using sudo command and such) coming from host (my server):

Low: Informational events (e.g., successful logins).

Medium: Suspicious activity (e.g., failed login attempts).

High: Potentially malicious (e.g., privilege escalation).

Critical: Confirmed malicious (e.g., malware detection).



Then I used Wazuh documentation to install agent in Ubuntu 24.04 client:

```
client2user@client2user-VirtualBox:~$ curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo gpg --dearmor -o /us
r/share/keyrings/wazuh-archive-keyring.gpg
client2user@client2user-VirtualBox:~$ echo "deb [signed-by=/usr/share/keyrings/wazuh-archive-keyring.gpg] https://pack
ages.wazuh.com/4.x/apt/ stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh-archive-keyring.gpg] https://packages.wazuh.com/4.x/apt/ stable main
client2user@client2user-VirtualBox:~$ sudo apt-get update
```

```
client2user@client2user-VirtualBox:~$ sudo apt-get install wazuh-agent
```

I edited configuration file /var/ossec/etc/ossec.conf to provide the ip address of Wazuh server:

```
client2user@client2user-VirtualBox:~$ sudo nano /var/ossec/etc/ossec.conf
```

There, I edited the file to include server address in <address</address>:

```
client2user@client2user-VirtualBox: ~
  GNU nano 7.2                                  /var/ossec/etc/ossec.conf
<!--
  Wazuh - Agent - Default configuration for ubuntu 24.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <client>
    <server>
      <address>192.168.31.47</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>ubuntu, ubuntu24, ubuntu24.04</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
  </client>
```

And I restarted wazuh-agent:

```
client2user@client2user-VirtualBox:~$ sudo systemctl start wazuh-agent
client2user@client2user-VirtualBox:~$ sudo nano /var/ossec/etc/ossec.conf
client2user@client2user-VirtualBox:~$ sudo systemctl restart wazuh-agent
```

On the Wazuh server, I ran the command sudo /var/ossec/bin/wazuh-authd -a to manually generate an agent authentication key – it allowed me to securely register the Wazuh agent with the manager by providing it with a unique key for identification:

```
wazuhserver@wazuhserver:~$ sudo /var/ossec/bin/wazuh-authd -a
```

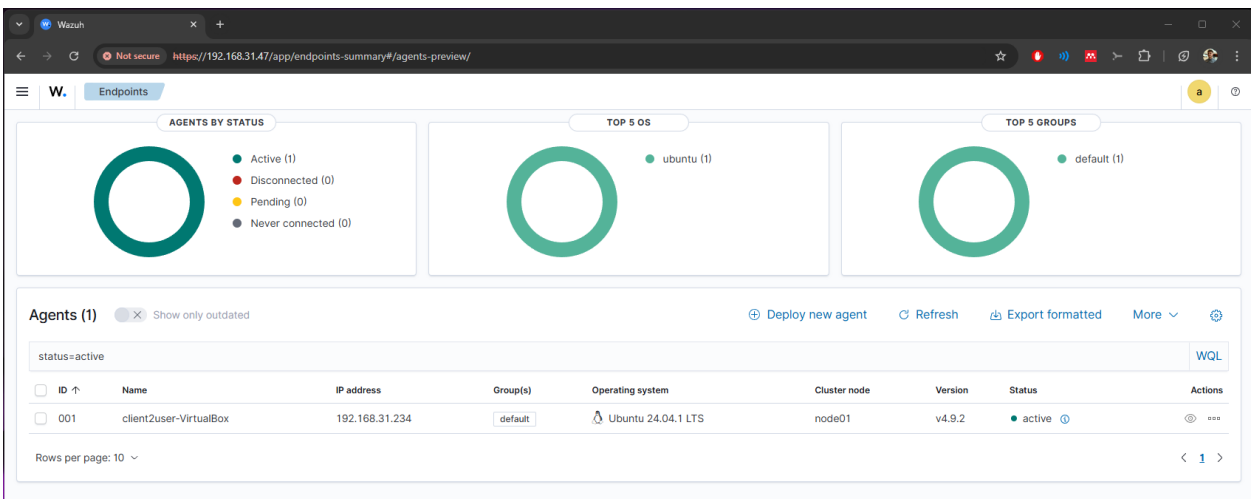
On the Wazuh agent machine, I ran the command sudo /var/ossec/bin/agent-auth -m 192.168.31.47 to register the agent with the Wazuh manager at the specified IP address:

```
client2user@client2user-VirtualBox:~$ sudo /var/ossec/bin/agent-auth -m 192.168.31.47
2024/12/03 00:50:10 agent-auth: INFO: Started (pid: 7629).
2024/12/03 00:50:10 agent-auth: INFO: Requesting a key from server: 192.168.31.47
2024/12/03 00:50:10 agent-auth: INFO: No authentication password provided
2024/12/03 00:50:10 agent-auth: INFO: Using agent name as: client2user-VirtualBox
2024/12/03 00:50:10 agent-auth: INFO: Waiting for server reply
2024/12/03 00:50:10 agent-auth: ERROR: Duplicate agent name: client2user-VirtualBox (from manager)
2024/12/03 00:50:10 agent-auth: ERROR: Unable to add agent (from manager)
client2user@client2user-VirtualBox:~$ sudo systemctl restart wazuh-agent
client2user@client2user-VirtualBox:~$
```

I used the command sudo /var/ossec/bin/agent_control -l on the Wazuh server to verify the list of connected agents. The output confirmed that the agent with ID 001 (named client2user-VirtualBox) is active and successfully connected to the Wazuh manager:

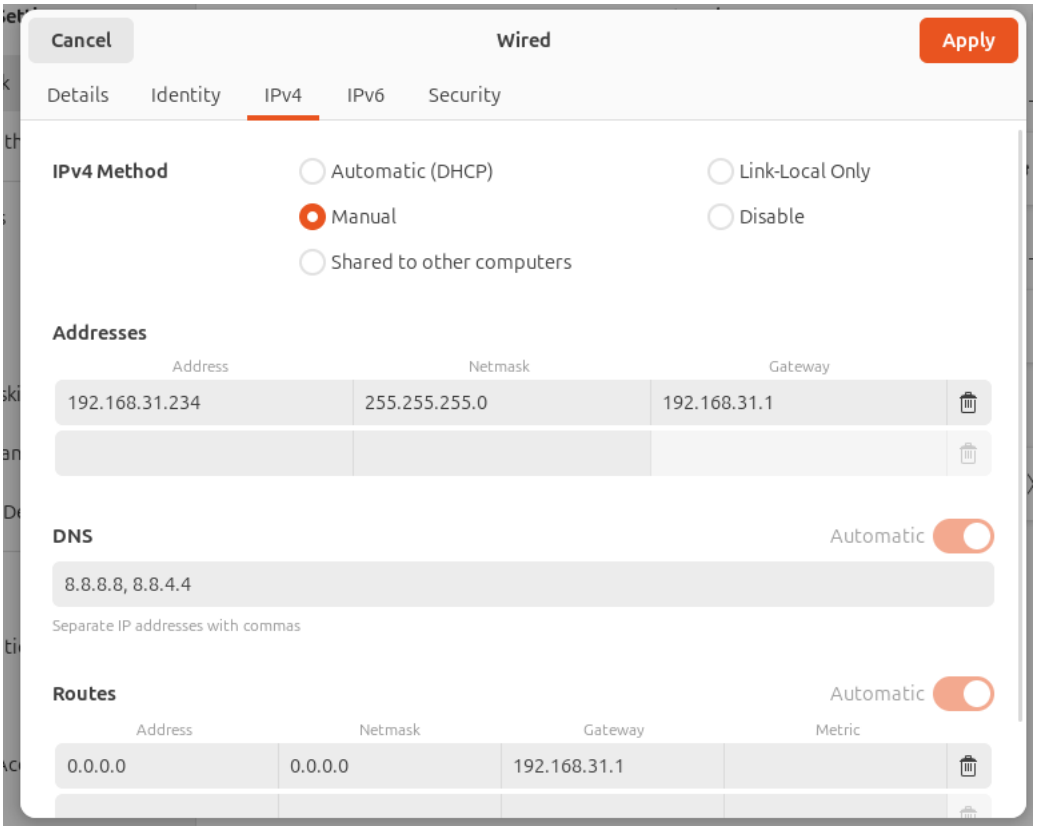
```
wazuhserver@wazuhserver:~$ sudo /var/ossec/bin/agent_control -l

Wazuh agent_control. List of available agents:
   ID: 000, Name: wazuhserver (server), IP: 127.0.0.1, Active/Local
   ID: 001, Name: client2user-VirtualBox, IP: any, Active

List of agentless devices:
```

In the Wazuh Dashboard, the connected agent (client2user-VirtualBox) is now reflected in the Agents Summary section. It shows as active, confirming successful registration and communication between the agent and the Wazuh manager:

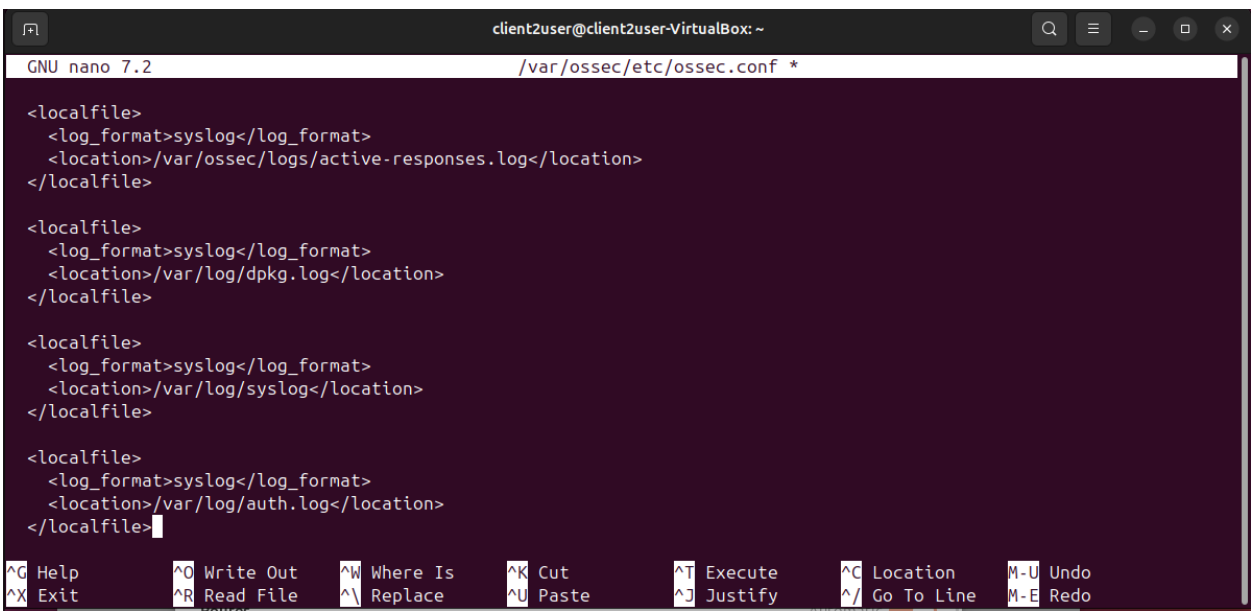


Then, I took client ip address and set up to be static:

```
client2user@client2user-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6e:82:8f brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.234/24 brd 192.168.31.255 scope global dynamic noprefixroute enp0s3
       valid_lft 40344sec preferred_lft 40344sec
    inet6 fe80::a00:27ff:fe6e:828f/64 scope link
       valid_lft forever preferred_lft forever
client2user@client2user-VirtualBox:~$ ip route
default via 192.168.31.1 dev enp0s3 proto dhcp src 192.168.31.234 metric 100
192.168.31.0/24 dev enp0s3 proto kernel scope link src 192.168.31.234 metric 100
client2user@client2user-VirtualBox:~$
```



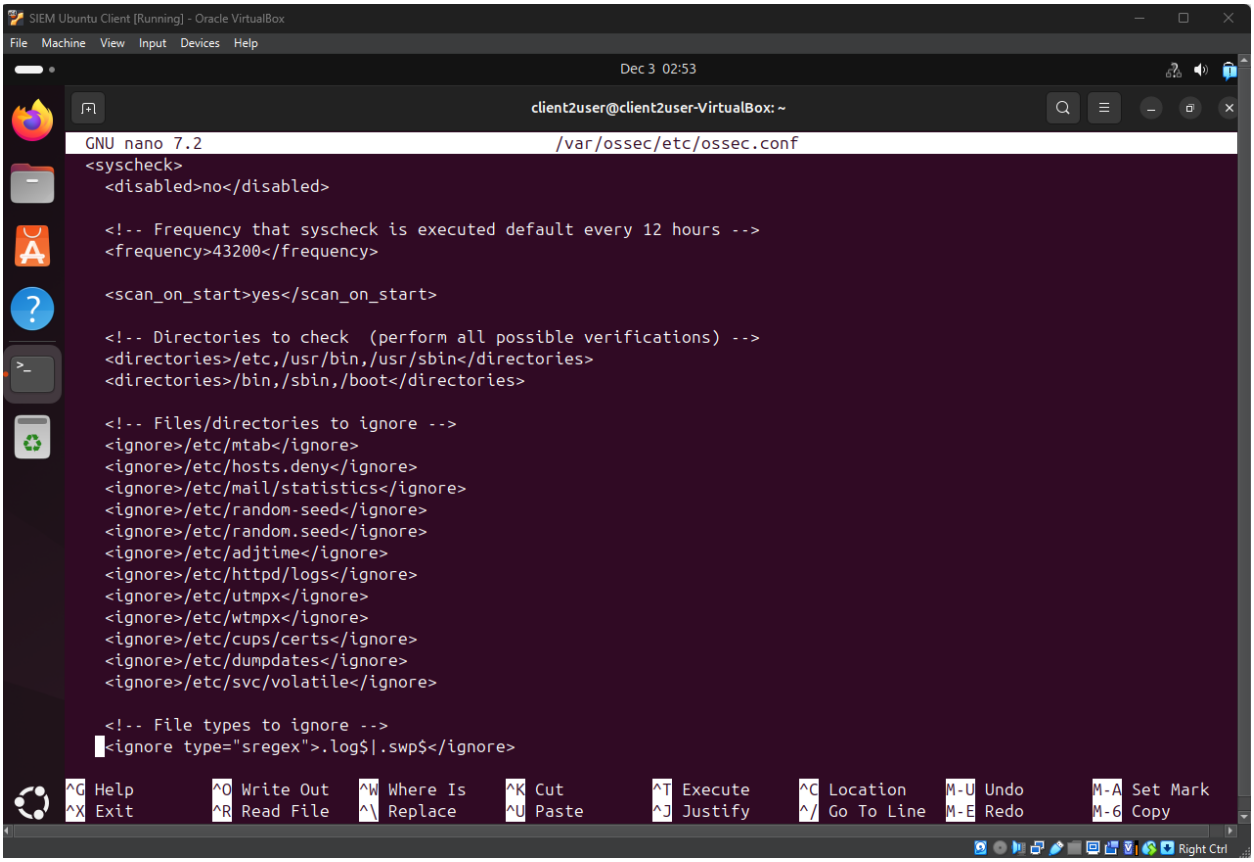
For log collection, I updated the ossec.conf file on the Wazuh agent to monitor additional system logs. Specifically, I added /var/log/syslog and /var/log/auth.log under the <localfile> section. These configurations allow the Wazuh agent to collect system and authentication logs for further analysis by the Wazuh manager:



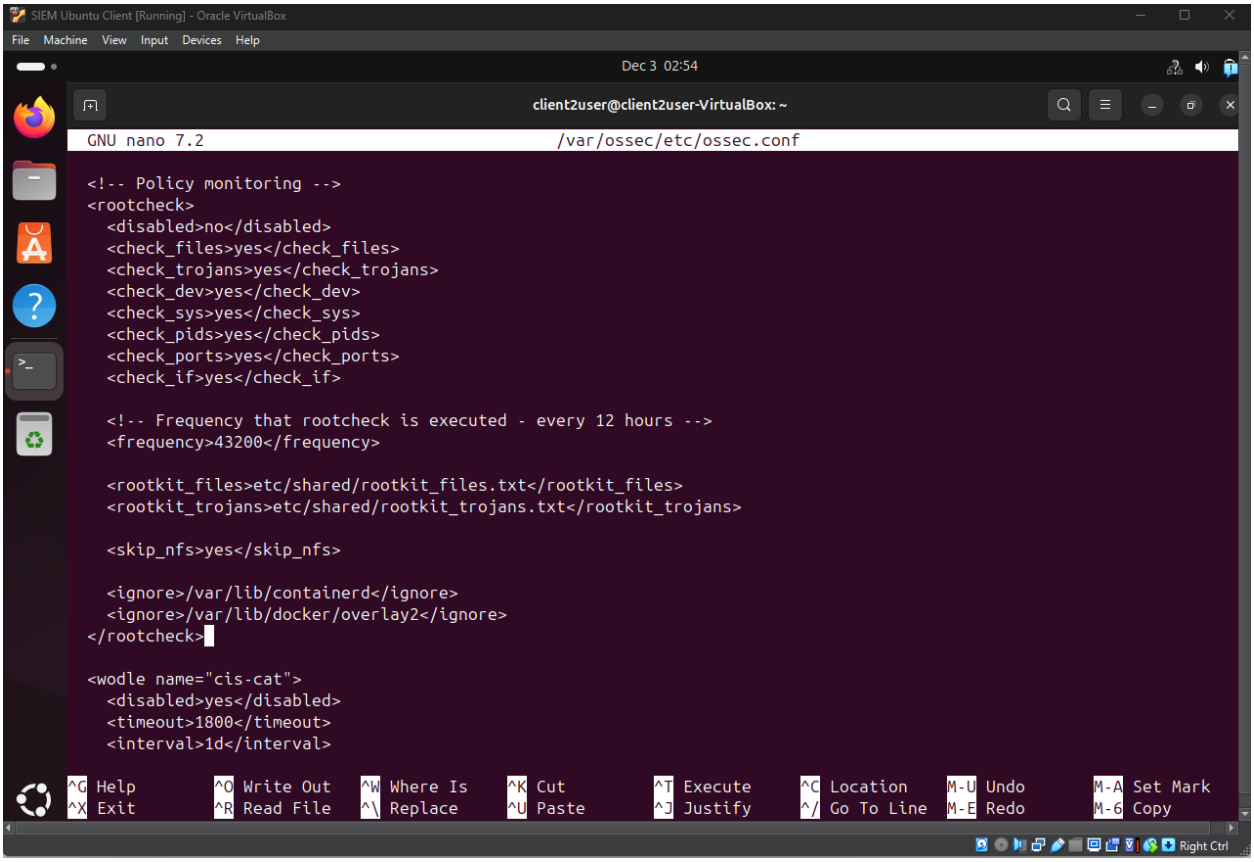
To demonstrate how my SIEM (Wazuh) performs pre-processing and normalization of received data, I configured the system to monitor a variety of log sources, such as system authentication logs and application logs. Pre-processing in Wazuh involves collecting raw logs from monitored endpoints and categorizing them into a unified format using decoders. These decoders extract key attributes (like timestamps, user IDs, and event actions) from different log formats, enabling normalization. Normalized data is then matched against rules to generate actionable alerts. For

this task, I monitored /var/log/auth.log for authentication events and configured Wazuh to trigger alerts for failed login attempts during the initial data processing phase.

Then I configured the <syscheck> module in the ossec.conf file on the agent. I enabled syscheck with <disabled>no</disabled> - this ensures the file integrity monitoring feature is active, directories such as /etc, /usr/bin, /usr/sbin, /bin, and /sbin are explicitly included.



Then, I configured the <rootcheck> module in the ossec.conf file to detect rootkits and other anomalies. I enabled rootcheck with <disabled>no</disabled>. It enables checks for files, trojans, devices, syscalls, ports, and process IDs.



To test the security incident detection functionality I generated a file integrity event, ran the command echo "Test modification" | sudo tee -a /etc/hosts - this added the text "Test modification" to the /etc/hosts file, triggering a file integrity monitoring alert on the Wazuh agent:

```
client2user@client2user-VirtualBox:~$ echo "Test modification" | sudo tee -a /etc/hosts
Test modification
```

The Wazuh manager detected the modification of /etc/hosts and recorded the event with alert:


```
> Dec 3, 2024 @ 02:54:47.466  predecoder.program_name: sudo predecoder.timestamp: 2024-12-03T02:55:04.537220+02:00 input.type: log agent.ip: 192.168.31.234 agent.name: client2user-VirtualBox
agent.id: 001 manager.name: wazuhserver data.srcuser: client2user data.dstuser: root data.tty: pts/0 data.pwd: /home/client2user data.command: /usr/bin/tee -a /
etc/hosts rule.mail: false rule.level: 3 rule.pci_dss: 10.2.5, 10.2.2 rule.hipaa: 164.312.b rule.tsc: CC6.8, CC7.2, CC7.3 rule.description: Successful sudo to RO
OT executed. rule.groups: syslog, sudo rule.nist_800_53: AU.14, AC.7, AC.6 rule.gdpr: IV_32.2 rule.firedtimes: 2 rule.mitre.technique: Sudo and Sudo Caching
rule.mitre.id: T1548.003 rule.mitre.tactic: Privilege Escalation, Defense Evasion rule.id: 5402 rule.pci_dss: 7.6, 7.8, 7.13 location: /var/log/auth.log
```


Then I used sudo su – command:

```
client2user@client2user-VirtualBox:~$ sudo su -
```

The Wazuh agent detected a privilege escalation event where the command sudo su - was executed. This alert highlights a root access attempt:


```
> Dec 3, 2024 @ 03:04:18.075  predecoder.program_name: sudo predecoder.timestamp: 2024-12-03T03:04:34.387890+02:00 input.type: log agent.ip: 192.168.31.234 agent.name: client2user-VirtualBox
agent.id: 001 manager.name: wazuhserver data.srcuser: client2user data.dstuser: root data.tty: pts/0 data.pwd: /home/client2user data.command: /usr/bin/su -
rule.mail: false rule.level: 3 rule.pci_dss: 10.2.5, 10.2.2 rule.hipaa: 164.312.b rule.tsc: CC6.8, CC7.2, CC7.3 rule.description: Successful sudo to ROOT execute
d. rule.groups: syslog, sudo rule.nist_800_53: AU.14, AC.7, AC.6 rule.gdpr: IV_32.2 rule.firedtimes: 1 rule.mitre.technique: Sudo and Sudo Caching
rule.mitre.id: T1548.003 rule.mitre.tactic: Privilege Escalation, Defense Evasion rule.id: 5402 rule.pci_dss: 7.6, 7.8, 7.13 location: /var/log/auth.log
```


I used the command sudo tail -f /var/ossec/logs/alerts/alerts.json on the Wazuh server to monitor alert logs in real time. This file captures all generated alerts, including those triggered by security events and configuration changes:

```
wazuhserver@wazuhserver:~$ sudo tail -f /var/ossec/logs/alerts/alerts.json
```

The alerts.json file showed detailed logs of security events, including specific incidents such as "session opened for user root" and "successful sudo to ROOT executed." These logs include metadata like rule IDs, severities, descriptions, and timestamps, allowing me to analyze and verify that the alerting system is operational:


```
{"timestamp":"2024-12-03T01:04:24.125+0000","rule":{"level":3,"description":"PAM: Login session closed.","id":"5502","firedtimes":3,"mail":false,"groups":["pam"
,"syslog"],"pci_dss":["10.2.5"],"gpg13":["7.8","7.9"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7"],"tsc":["CC6.8","CC7.2","CC7.3"]},"
agent":{"id":"001","name":"client2user-VirtualBox","ip":"192.168.31.234"},"manager":{"name":"wazuhserver"},"id":"1733187864.91561","full_log":"2024-12-03T03:04:
41.692985+02:00 client2user-VirtualBox sudo: pam_unix(sudo:session): session closed for user root","predecoder":{"program_name":"sudo","timestamp":"2024-12-03T0
3:04:41.692985+02:00"},"decoder":{"parent":"pam","name":"pam"},"data":{"dstuser":"root"},"location":"/var/log/auth.log"}
{"timestamp":"2024-12-03T01:09:18.538+0000","rule":{"level":3,"description":"Successful sudo to ROOT executed.","id":"5402","mitre":{"id":["T1548.003"],"tactic"
:["Privilege Escalation","Defense Evasion"],"technique":["Sudo and Sudo Caching"]},"firedtimes":2,"mail":false,"groups":["syslog","sudo"],"pci_dss":["10.2.5","1
0.2.2"],"gpg13":["7.6","7.8","7.13"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7","AC.6"],"tsc":["CC6.8","CC7.2","CC7.3"]},"agent":{"i
d":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188158.91980","full_log":"Dec 03 01:09:16 wazuhserver sudo[8888]: wazuhserver : TTY=tt
y1 ; PWD=/home/wazuhserver ; USER=root ; COMMAND=/usr/bin/tail -f /var/ossec/logs/alerts/alerts.json","predecoder":{"program_name":"sudo","timestamp":"Dec 03 01
:09:16","hostname":"wazuhserver"},"decoder":{"parent":"sudo","name":"sudo","ftscomment":"First time user executed the sudo command"},"data":{"srcuser":"wazuhser
ver","dstuser":"root","tty":"tty1","pwd":"/home/wazuhserver","command":"/usr/bin/tail -f /var/ossec/logs/alerts/alerts.json"},"location":"journald"}
{"timestamp":"2024-12-03T01:09:18.539+0000","rule":{"level":3,"description":"PAM: Login session opened.","id":"5501","mitre":{"id":["T1078"],"tactic":["Defense
Evasion","Persistence","Privilege Escalation","Initial Access"],"technique":["Valid Accounts"]},"firedtimes":3,"mail":false,"groups":["pam","syslog","authentica
tion_success"],"pci_dss":["10.2.5"],"gpg13":["7.8","7.9"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7"],"tsc":["CC6.8","CC7.2","CC7.3"
]},"agent":{"id":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188158.92564","full_log":"Dec 03 01:09:16 wazuhserver sudo[8888]: pam_un
ix(sudo:session): session opened for user root(uid=0) by wazuhserver(uid=1000)","predecoder":{"program_name":"sudo","timestamp":"Dec 03 01:09:16","hostname":"wa
zuhserver"},"decoder":{"parent":"pam","name":"pam"},"data":{"srcuser":"wazuhserver","dstuser":"root(uid=0)","uid":"1000"},"location":"journald"}
{"timestamp":"2024-12-03T01:09:38.558+0000","rule":{"level":3,"description":"PAM: Login session closed.","id":"5502","firedtimes":4,"mail":false,"groups":["pam"
,"syslog"],"pci_dss":["10.2.5"],"gpg13":["7.8","7.9"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7"],"tsc":["CC6.8","CC7.2","CC7.3"]},"
agent":{"id":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188178.93007","full_log":"Dec 03 01:09:37 wazuhserver sudo[8888]: pam_unix(s
udo:session): session closed for user root","predecoder":{"program_name":"sudo","timestamp":"Dec 03 01:09:37","hostname":"wazuhserver"},"decoder":{"parent":"pam
","name":"pam"},"data":{"dstuser":"root"},"location":"journald"}
{"timestamp":"2024-12-03T01:09:42.562+0000","rule":{"level":3,"description":"Successful sudo to ROOT executed.","id":"5402","mitre":{"id":["T1548.003"],"tactic"
:["Privilege Escalation","Defense Evasion"],"technique":["Sudo and Sudo Caching"]},"firedtimes":3,"mail":false,"groups":["syslog","sudo"],"pci_dss":["10.2.5","1
0.2.2"],"gpg13":["7.6","7.8","7.13"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7","AC.6"],"tsc":["CC6.8","CC7.2","CC7.3"]},"agent":{"i
d":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188182.93378","full_log":"Dec 03 01:09:42 wazuhserver sudo[8892]: wazuhserver : TTY=tt
y1 ; PWD=/home/wazuhserver ; USER=root ; COMMAND=/usr/bin/tail -f 5 /var/ossec/logs/alerts/alerts.json","predecoder":{"program_name":"sudo","timestamp":"Dec 03
01:09:42","hostname":"wazuhserver"},"decoder":{"parent":"sudo","name":"sudo","ftscomment":"First time user executed the sudo command"},"data":{"srcuser":"wazuhs
erver","dstuser":"root","tty":"tty1","pwd":"/home/wazuhserver","command":"/usr/bin/tail -f 5 /var/ossec/logs/alerts/alerts.json"},"location":"journald"}
{"timestamp":"2024-12-03T01:09:42.562+0000","rule":{"level":3,"description":"PAM: Login session opened.","id":"5501","mitre":{"id":["T1078"],"tactic":["Defense
Evasion","Persistence","Privilege Escalation","Initial Access"],"technique":["Valid Accounts"]},"firedtimes":4,"mail":false,"groups":["pam","syslog","authentica
tion_success"],"pci_dss":["10.2.5"],"gpg13":["7.8","7.9"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7"],"tsc":["CC6.8","CC7.2","CC7.3"
]},"agent":{"id":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188182.93966","full_log":"Dec 03 01:09:42 wazuhserver sudo[8892]: pam_un
ix(sudo:session): session opened for user root(uid=0) by wazuhserver(uid=1000)","predecoder":{"program_name":"sudo","timestamp":"Dec 03 01:09:42","hostname":"wa
zuhserver"},"decoder":{"parent":"pam","name":"pam"},"data":{"srcuser":"wazuhserver","dstuser":"root(uid=0)","uid":"1000"},"location":"journald"}
{"timestamp":"2024-12-03T01:09:48.568+0000","rule":{"level":3,"description":"PAM: Login session opened.","id":"5501","mitre":{"id":["T1078"],"tactic":["Defense
Evasion","Persistence","Privilege Escalation","Initial Access"],"technique":["Valid Accounts"]},"firedtimes":5,"mail":false,"groups":["pam","syslog","authentica
tion_success"],"pci_dss":["10.2.5"],"gpg13":["7.8","7.9"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7"],"tsc":["CC6.8","CC7.2","CC7.3"
]},"agent":{"id":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188188.94409","full_log":"Dec 03 01:09:48 wazuhserver sudo[8897]: pam_un
ix(sudo:session): session opened for user root(uid=0) by wazuhserver(uid=1000)","predecoder":{"program_name":"sudo","timestamp":"Dec 03 01:09:48","hostname":"wa
zuhserver"},"decoder":{"parent":"pam","name":"pam"},"data":{"srcuser":"wazuhserver","dstuser":"root(uid=0)","uid":"1000"},"location":"journald"}
{"timestamp":"2024-12-03T01:09:48.568+0000","rule":{"level":3,"description":"Successful sudo to ROOT executed.","id":"5402","mitre":{"id":["T1548.003"],"tactic"
:["Privilege Escalation","Defense Evasion"],"technique":["Sudo and Sudo Caching"]},"firedtimes":4,"mail":false,"groups":["syslog","sudo"],"pci_dss":["10.2.5","1
0.2.2"],"gpg13":["7.6","7.8","7.13"],"gdpr":["IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["AU.14","AC.7","AC.6"],"tsc":["CC6.8","CC7.2","CC7.3"]},"agent":{"i
d":"000","name":"wazuhserver"},"manager":{"name":"wazuhserver"},"id":"1733188188.94852","full_log":"Dec 03 01:09:48 wazuhserver sudo[8897]: wazuhserver : TTY=tt
y1 ; PWD=/home/wazuhserver ; USER=root ; COMMAND=/usr/bin/tail -f 1 /var/ossec/logs/alerts/alerts.json","predecoder":{"program_name":"sudo","timestamp":"Dec 03
01:09:48","hostname":"wazuhserver"},"decoder":{"parent":"sudo","name":"sudo","ftscomment":"First time user executed the sudo command"},"data":{"srcuser":"wazuhs
erver","dstuser":"root","tty":"tty1","pwd":"/home/wazuhserver","command":"/usr/bin/tail -f 1 /var/ossec/logs/alerts/alerts.json"},"location":"journald"}
```


To apply correlation rules in Wazuh, I navigated to the directory /var/ossec/ruleset/rules, which contains a wide range of predefined rules provided by Wazuh. These rules are categorized for different use cases, including syslog, SSH, web application monitoring, and more. The screenshot showcases the available rule files in the system, indicating that the correlation rules are in place and ready for use. These rules are designed to analyze collected logs and generate

alerts based on detected security events:

```
wazuhserver@wazuhserver:~$ sudo ls /var/ossec/ruleset/rules
0010-rules_config.xml            0215-policy_rules.xml            0420-freeipa_rules.xml                0625-cisco-asa_rules.xml
0015-ossec_rules.xml             0220-msauth_rules.xml            0425-cisco-estreamer_rules.xml        0625-mcafee_epo_rules.xml
0016-wazuh_rules.xml             0225-mcafee_av_rules.xml         0430-ms_wdefender_rules.xml           0630-nextcloud_rules.xml
0017-wazuh-api_rules.xml         0230-ms-se_rules.xml             0435-ms_logs_rules.xml                0635-owlh-zeek_rules.xml
0020-syslog_rules.xml            0235-vmware_rules.xml            0440-ms_sqlserver_rules.xml           0640-junos_rules.xml
0025-sendmail_rules.xml          0240-ids_rules.xml               0445-identity_guard_rules.xml         0675-panda-paps_rules.xml
0030-postfix_rules.xml           0245-web_rules.xml               0450-mongodb_rules.xml                0680-checkpoint-smart1_rules.xml
0035-spamd_rules.xml             0250-apache_rules.xml            0455-docker_rules.xml                 0690-gcp_rules.xml
0040-imapd_rules.xml             0255-zeus_rules.xml              0460-jenkins_rules.xml                0695-f5_bigip_rules.xml
0045-mailscanner_rules.xml       0260-nginx_rules.xml             0470-vshell_rules.xml                 0700-paloalto_rules.xml
0050-ms-exchange_rules.xml       0265-php_rules.xml               0475-suricata_rules.xml               0705-sophos_fw_rules.xml
0055-courier_rules.xml           0270-web_appsec_rules.xml        0480-qualysguard_rules.xml            0715-freepbx_rules.xml
0065-pix_rules.xml               0275-squid_rules.xml             0485-cylance_rules.xml                0750-github_rules.xml
0070-netscreenfw_rules.xml       0280-attack_rules.xml            0490-virustotal_rules.xml             0755-office365_rules.xml
0075-cisco-ios_rules.xml         0285-systemd_rules.xml           0495-proxmox-ve_rules.xml             0770-gitlab_rules.xml
0080-sonicwall_rules.xml         0290-firewalld_rules.xml         0500-owncloud_rules.xml               0775-arbor_rules.xml
0085-pam_rules.xml               0295-mysql_rules.xml             0505-vuls_rules.xml                   0780-fireeye_rules.xml
0090-telnetd_rules.xml           0300-postgresql_rules.xml        0510-ciscat_rules.xml                 0785-huawei-usg_rules.xml
0095-sshd_rules.xml              0305-dropbear_rules.xml          0515-exim_rules.xml                   0800-sysmon_id_1.xml
0100-solaris_bsm_rules.xml       0310-openbsd_rules.xml           0520-vulnerability-detector_rules.xml 0810-sysmon_id_3.xml
0105-asterisk_rules.xml          0315-apparmor_rules.xml          0525-openvas_rules.xml                0820-sysmon_id_7.xml
0110-ms_dhcp_rules.xml           0320-clam_av_rules.xml           0530-mysql_audit_rules.xml            0830-sysmon_id_11.xml
0115-arpwatch_rules.xml          0325-opensmtpd_rules.xml         0535-mariadb_rules.xml                0840-win_event_channel.xml
0120-symantec-av_rules.xml       0330-sysmon_rules.xml            0540-pfsense_rules.xml                0850-audit_rules.xml
0125-symantec-ws_rules.xml       0335-unbound_rules.xml           0545-osquery_rules.xml                0860-sysmon_id_13.xml
0130-trend-osce_rules.xml        0340-puppet_rules.xml            0550-kaspersky_rules.xml              0870-sysmon_id_8.xml
0135-hordeimp_rules.xml          0345-netscaler_rules.xml         0555-azure_rules.xml                  0900-firewall_rules.xml
0140-roundcube_rules.xml         0350-amazon_rules.xml            0560-docker_integration_rules.xml     0905-cisco-ftd_rules.xml
0145-wordpress_rules.xml         0360-serv-u_rules.xml            0565-ms_ipsec_rules.xml               0910-ms-exchange-proxylogon_rules.xml
0150-cimserver_rules.xml         0365-auditd_rules.xml            0570-sca_rules.xml                    0915-win-powershell_rules.xml
0155-dovecot_rules.xml           0375-usb_rules.xml               0575-win-base_rules.xml               0920-oracledb_rules.xml
0160-vmpop3d_rules.xml           0380-redis_rules.xml             0580-win-security_rules.xml           0925-eset-remote_rules.xml
0165-vpopmail_rules.xml          0385-oscap_rules.xml             0585-win-application_rules.xml        0935-cloudflare-waf_rules.xml
0170-ftpd_rules.xml              0390-fortiddos_rules.xml         0590-win-system_rules.xml             0945-sysmon_id_10.xml
0175-proftpd_rules.xml           0391-fortigate_rules.xml         0595-win-sysmon_rules.xml             0950-sysmon_id_20.xml
0180-pure-ftpd_rules.xml         0392-fortimail_rules.xml         0600-win-wdefender_rules.xml          0955-WEF-baseline_rules.xml
0185-vsftpd_rules.xml            0393-fortiauth_rules.xml         0601-win-vipre_rules.xml              0960-macos_rules.xml
0190-ms_ftpd_rules.xml           0395-hp_rules.xml                0602-win-wfirewall_rules.xml          0990-amazon-security-lake_rules.xml
0195-named_rules.xml             0400-openvpn_rules.xml           0605-win-mcafee_rules.xml             0995-microsoft-graph_rules.xml
0200-smbd_rules.xml              0405-rsa-auth-manager_rules.xml  0610-win-ms_logs_rules.xml            0997-maltiverse_rules.xml
0205-racoon_rules.xml            0410-imperva_rules.xml           0615-win-ms-se_rules.xml              0998-aws-security-hub-rules.xml
0210-vpn_concentrator_rules.xml  0415-sophos_rules.xml            0620-win-generic_rules.xml
wazuhserver@wazuhserver:~$ _
```

To configure data retention as part of my SIEM setup, I modified the internal_options.conf file located in /var/ossec/etc/. The screenshot highlights settings related to log retention and rotation. I configured monitord.keep_log_days to 31, ensuring that logs are retained for 31 days before being automatically deleted. Log rotation was enabled by setting monitord.rotate_log to 1, and I specified the maximum size for log files (monitord.size.rotate=512) as well as the maximum number of rotations per day (monitord.daily_rotations=12). Additional parameters, such as delays for real-time notifications (syscheck.rt_delay=5), were adjusted to optimize performance:

```
  GNU nano 7.2                                  /var/ossec/etc/internal_options.conf
maild.grouping=1

# Maild full subject (0=disabled, 1=enabled)
maild.full_subject=0

# Maild display GeoIP data (0=disabled, 1=enabled)
maild.geoip=1


# Monitord day_wait. Amount of seconds to wait before rotating/compressing/signing [0..600]
# the files.
monitord.day_wait=10

# Monitord compress. (0=do not compress, 1=compress)
monitord.compress=1

# Monitord sign. (0=do not sign, 1=sign)
monitord.sign=1

# Monitord monitor_agents. (0=do not monitor, 1=monitor)
monitord.monitor_agents=1

# Rotate plain and JSON logs daily. (0=no, 1=yes)
monitord.rotate_log=1

# Days to keep old ossec.log files [0..500]
monitord.keep_log_days=31

# Size of internal log files to rotate them (Megabytes) [0..4096]
monitord.size_rotate=512

# Maximum number of rotations per day for internal logs [1..256]
monitord.daily_rotations=12

# Number of minutes for deleting a disconnected agent [0..9600]. (0=disabled)
monitord.delete_old_agents=0

# Syscheck perform a delay when dispatching real-time notifications so it avoids
# triggering on some temporary files like vim edits. (ms) [0..1000]
syscheck.rt_delay=5

# Maximum number of directories monitored for realtime on windows [1..1024]
syscheck.max_fd_win_rt=256

# Maximum number of directories monitored for who-data on Linux [1..4096]
syscheck.max_audit_entries=256

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket  M-Q Previous
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was    M-W Next
```

Here's how to perform each practical task using **Ubuntu Server (running Wazuh 4.9)** and **Ubuntu Client 24.04 (with Wazuh agent)**:

---

1. Generate and Analyze Logs

Ubuntu Server (Wazuh Manager):

1. Open the Wazuh dashboard and navigate to the **Discover** section.
2. Search for logs by filtering for the specific system or agent in the query bar.

Ubuntu Client (Wazuh Agent):

1. Simulate a security event by running ssh user@localhost and entering an incorrect password multiple times.
2. Verify logs by checking /var/log/auth.log for Failed password entries.

---

2. Apply Correlation Rules

Ubuntu Server (Wazuh Manager):

1. Navigate to **Rules** > **Create Rule** in the Wazuh dashboard.
2. Create a rule to correlate multiple failed login attempts:
3. <group name="authentication_failures">
4. <rule id="100001" level="8">
5. <decoded_as>syslog</decoded_as>
6. <description>Multiple failed SSH login attempts</description>
7. <match>Failed password</match>
8. <same_ip/>
9. <frequency>5</frequency>
10. <timeframe>300</timeframe>
11. </rule>
12. </group>
13. Save the rule and restart the Wazuh Manager with systemctl restart wazuh-manager.

---

3. Set Up Alerts

Ubuntu Server (Wazuh Manager):

1. Create a rule in /var/ossec/rules/ to monitor access attempts to a sensitive file.
2. Example rule for /etc/passwd access:
3. <group name="file_monitoring">
4. <rule id="100002" level="10">
5. <description>Unauthorized access to /etc/passwd</description>
6. <match>access</match>
7. <location>/etc/passwd</location>
8. </rule>
9. </group>
10. Enable alerts in the Wazuh dashboard and specify email or Slack notifications.

---

4. Monitor Real-Time Events

Ubuntu Server (Wazuh Manager):

1. Navigate to **Monitoring** > **Active Alerts** in the dashboard.
2. Use the search filters to observe port scans or unusual traffic patterns.
3. Test using nmap -sS localhost on the client to simulate a port scan.

---

5. Perform Log Retention and Search

Ubuntu Server (Wazuh Manager):

1. Go to **Discover** in the Wazuh dashboard.
2. Query logs from the past 7 days using:
3. host:<agent-name> AND message:<search-term>
4. Logs are retained as per the /var/ossec/logs configuration.

---

6. Generate Reports

Ubuntu Server (Wazuh Manager):

1. Navigate to **Reporting** > **Generate Report**.
2. Configure a report for all access attempts to sensitive files like /var/log/secure or /etc/passwd.
3. Set the time range to the last 30 days and download the PDF/CSV.

---

7. Configure a Dashboard

Ubuntu Server (Wazuh Manager):

1. Open the Wazuh dashboard and go to **Dashboards**.
2. Create a custom dashboard showing:
    - Login failures (filter: Failed password).
    - Critical system alerts (filter: level:[8 TO 10]).
    - Trends using **Time Series** widgets.

---

8. Test Incident Response

Ubuntu Client (Wazuh Agent):

1. Simulate a malware event by creating a suspicious file:
2. echo "malicious_code" > /tmp/malware.sh
3. chmod +x /tmp/malware.sh
4. The agent will log the activity, and Wazuh Manager will generate an alert if FIM (File Integrity Monitoring) is configured.

Ubuntu Server (Wazuh Manager):

1. Check the **Active Alerts** section for the generated incident.

---

9. Identify Anomalies

Ubuntu Server (Wazuh Manager):

1. Enable the Wazuh **Anomaly Detection** module.
2. Navigate to **Modules** > **Anomalies** in the dashboard.
3. Monitor alerts for unusual user logins:
    - Use a different geographic location to log in to the agent if possible.

---

10. Analyze File Integrity

Ubuntu Server (Wazuh Manager):

1. Configure File Integrity Monitoring (FIM) in /var/ossec/etc/ossec.conf:
2. <fim>

3. <directories check_all="yes">/etc</directories>
4. </fim>
5. Restart the manager with systemctl restart wazuh-manager.

Ubuntu Client (Wazuh Agent):

1. Modify a file in /etc, e.g., nano /etc/passwd.
2. Verify alerts in the manager dashboard under **FIM Events**.

These instructions ensure you can practically complete and test tasks using Wazuh. Let me know if you need more detailed steps for specific tasks!