

SIEM'S: HOME LAB

2110040122@klh.edu.in
BOBBA PANDURANGA
CYBER SECURITY

Table of Contents

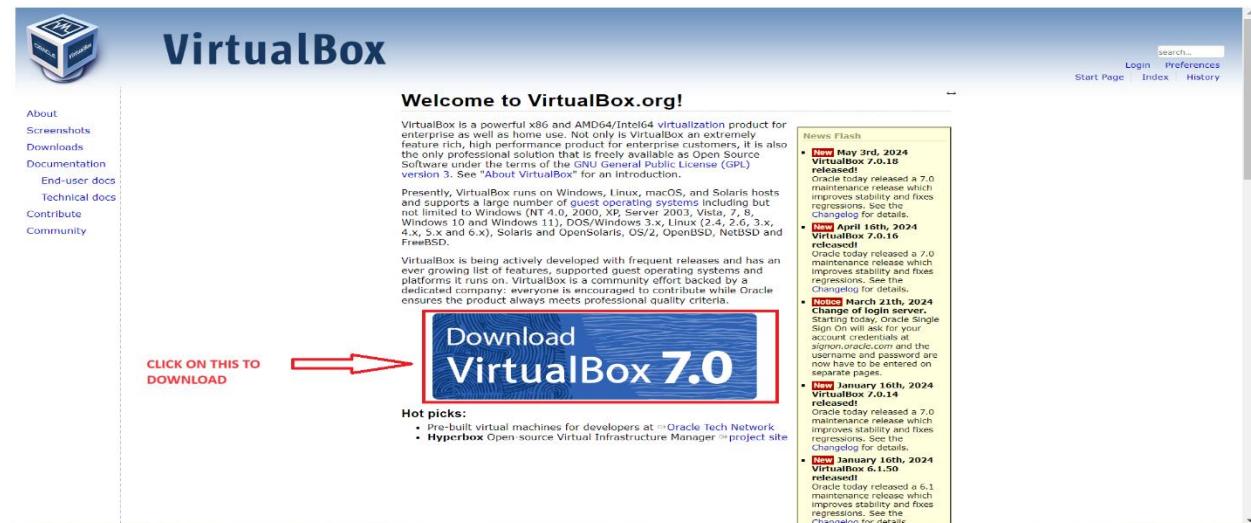
Virtual Machine Installation	1
Installation of Windows 10 Pro in Virtual Machine	2
Type chapter title (level 3).....	3
Malware detection	4
Wazuh for malware detection	5
Detecting malicious activities with threat detection	
.....	6

1. INSTALLING VIRTUAL MACHINE

STEP 1: Go To The Virtual Machine Official Website: <https://www.virtualbox.org/>



STEP 2: Download The Virtual Machine By Clicking On Download Button



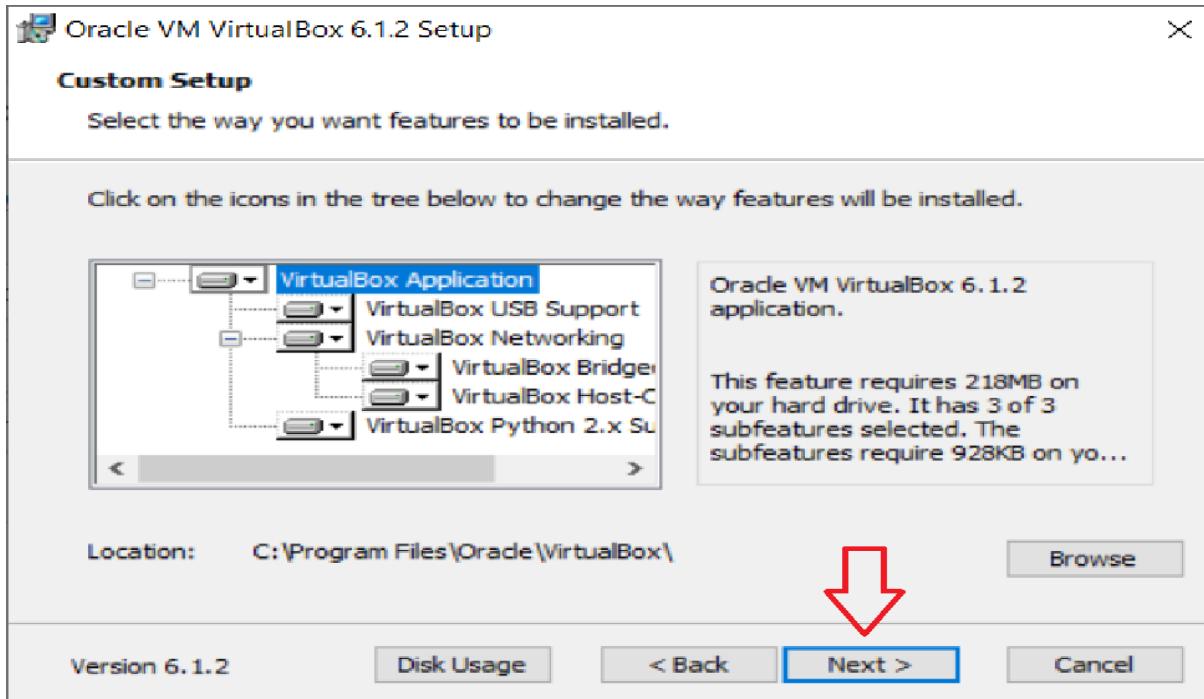
STEP 3: Select Virtualbox As Per Your OS

The screenshot shows the official VirtualBox website at <https://www.virtualbox.org>. The main heading is "VirtualBox". On the left, there's a sidebar with links like "About", "Screenshots", "Downloads", "Documentation", "End-user docs", "Technical docs", "Contribute", and "Community". The main content area is titled "Download VirtualBox" and contains a section for "VirtualBox binaries". A red box highlights the "VirtualBox 7.0.18 platform packages" section, which lists "Windows hosts", "macOS / Intel hosts", "Linux distributions", "Solaris hosts", and "Solaris 11 IPS hosts". A red arrow points from the text "SELECT AS PER YOUR OS" to this list. Below this, there's a note about GPL version 3, a changelog link, and a note about checksums. Further down are sections for "VirtualBox 7.0.18 Oracle VM VirtualBox Extension Pack" and "VirtualBox 7.0.18 Software Developer Kit (SDK)".

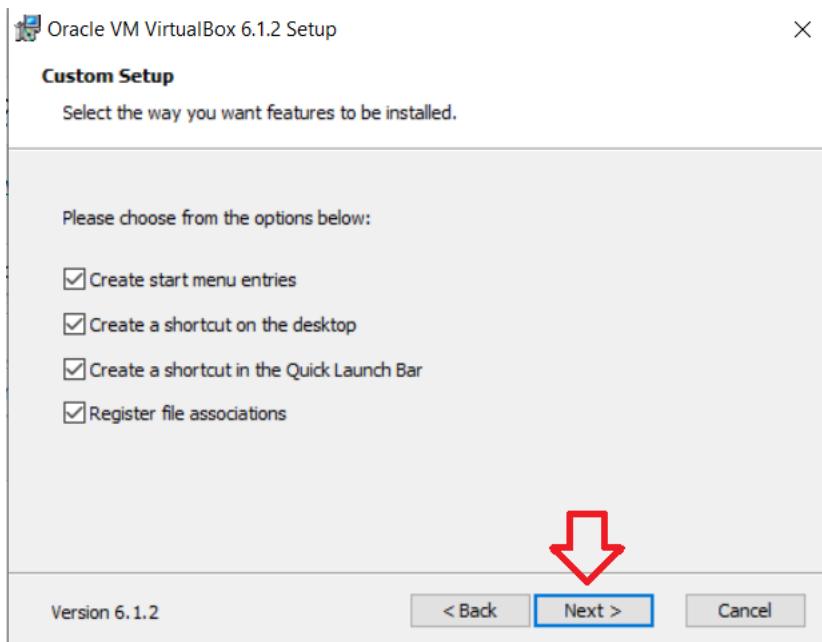
STEP 4: Install The Virtual Box

The screenshot shows the "Oracle VM VirtualBox 6.1.2 Setup" window. It features a large image of a VirtualBox host machine inside a cardboard box. The title bar says "Welcome to the Oracle VM VirtualBox 6.1.2 Setup Wizard". The text in the center states: "The Setup Wizard will install Oracle VM VirtualBox 6.1.2 on your computer. Click Next to continue or Cancel to exit the Setup Wizard." A red arrow points downwards towards the "Next >" button at the bottom right. At the bottom left, it says "Version 6.1.2".

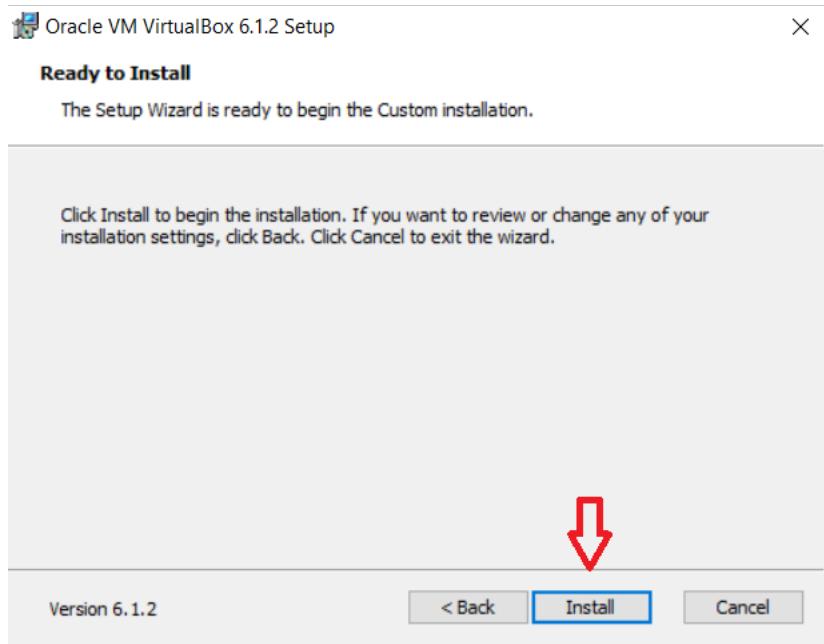
STEP 5: Select Installation Location



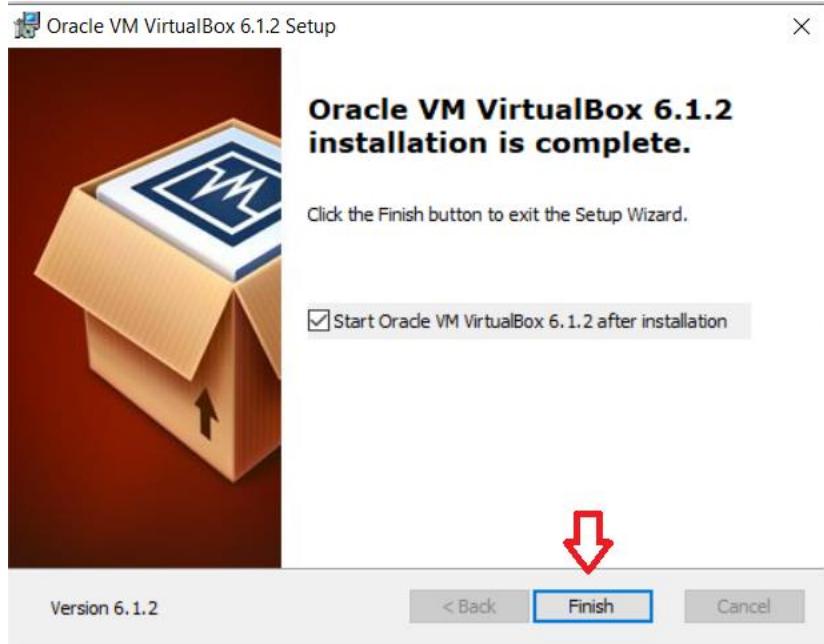
STEP 6: Creating Entries and Shortcuts:



STEP 7: Ready To Install:



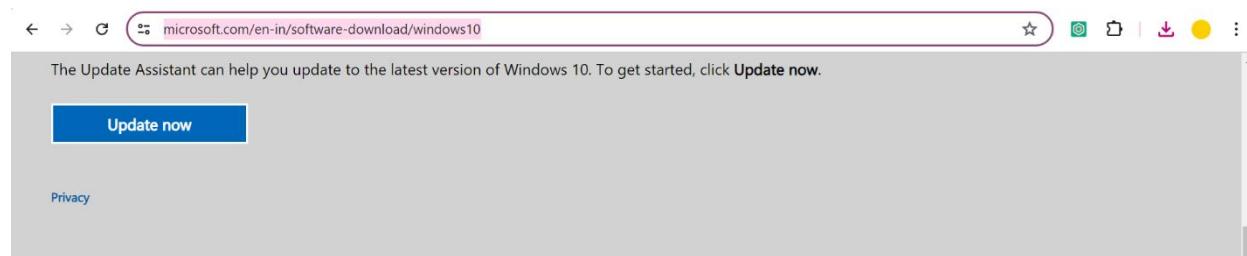
STEP 8: Finished Installation



1.1 INSTALING WIN 10 PRO IN VIRTUAL MACHINE

STEP 1: GO TO WINDOWS OFFICIAL WEBSITE: <https://www.microsoft.com/en-in/software-download/windows10>

STEP 2: DOWNLOAD THE TOOL.

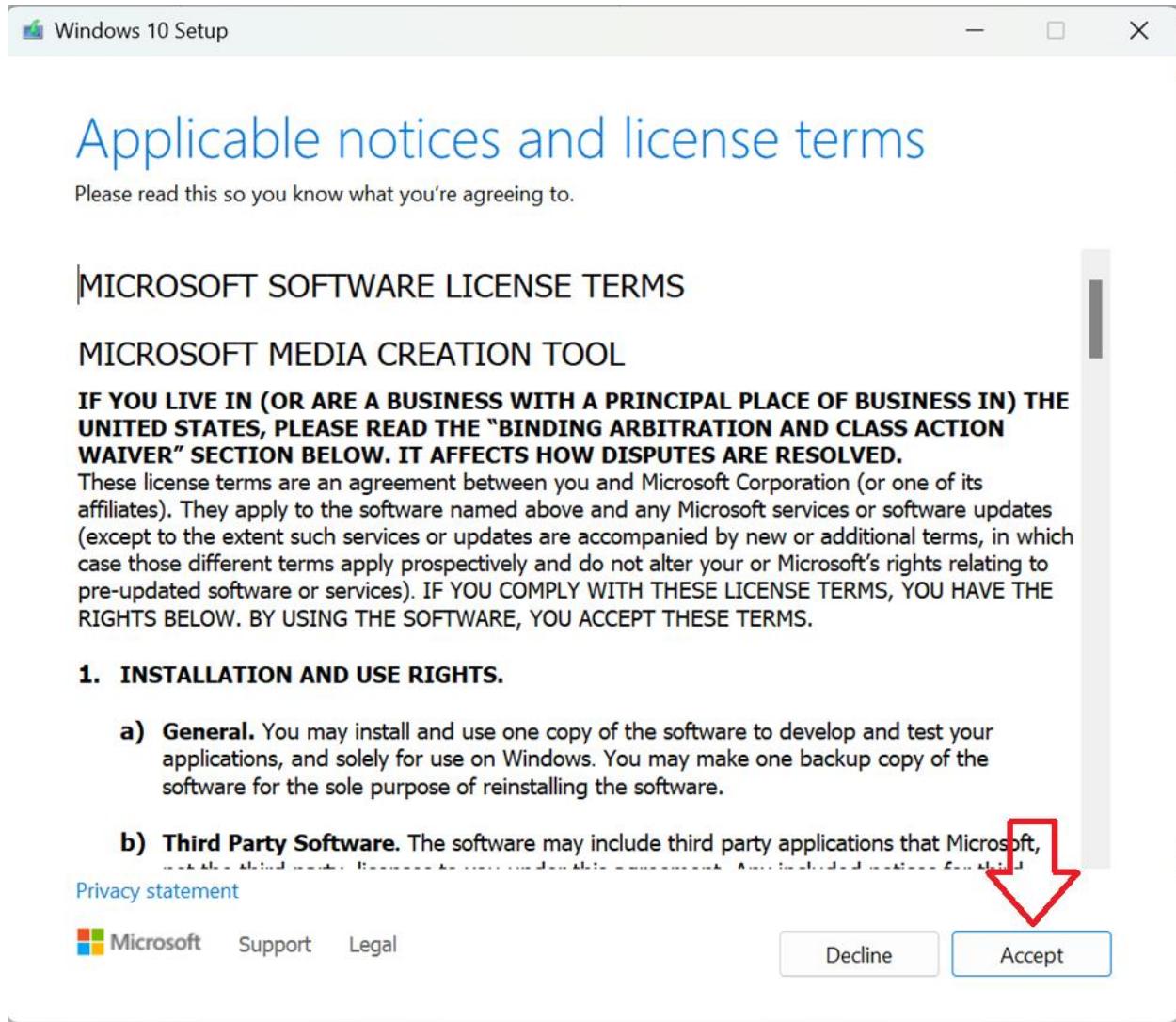


Create Windows 10 installation media

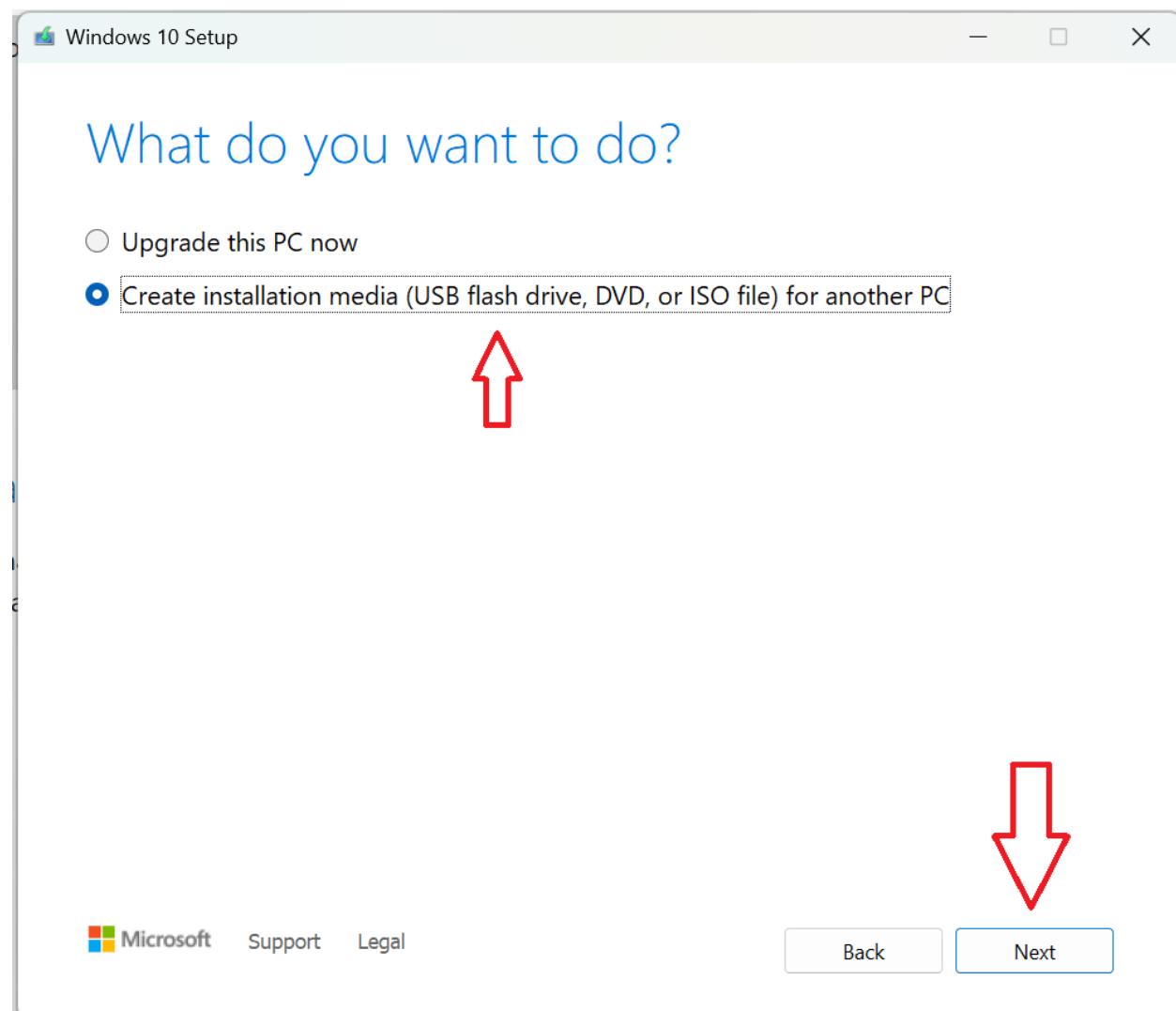
To get started, you will first need to have a licence to install Windows 10. You can then download and run the media creation tool. For more information on how to use the tool, see the instructions below.

A screenshot of a web page for creating Windows 10 installation media. A red box highlights the 'Download tool now' button, which has a red arrow pointing to it from the left. To the right of the button is a small image of a laptop displaying the Windows 10 desktop. Below the button is a 'Privacy' link. At the bottom of the page are two expandable sections: '+ Using the tool to upgrade this PC to Windows 10 (click to show more or less information)' and '+ Using the tool to create installation media (USB flash drive, DVD, or ISO file) to install Windows 10 on a different PC (click to show more or less information)'.

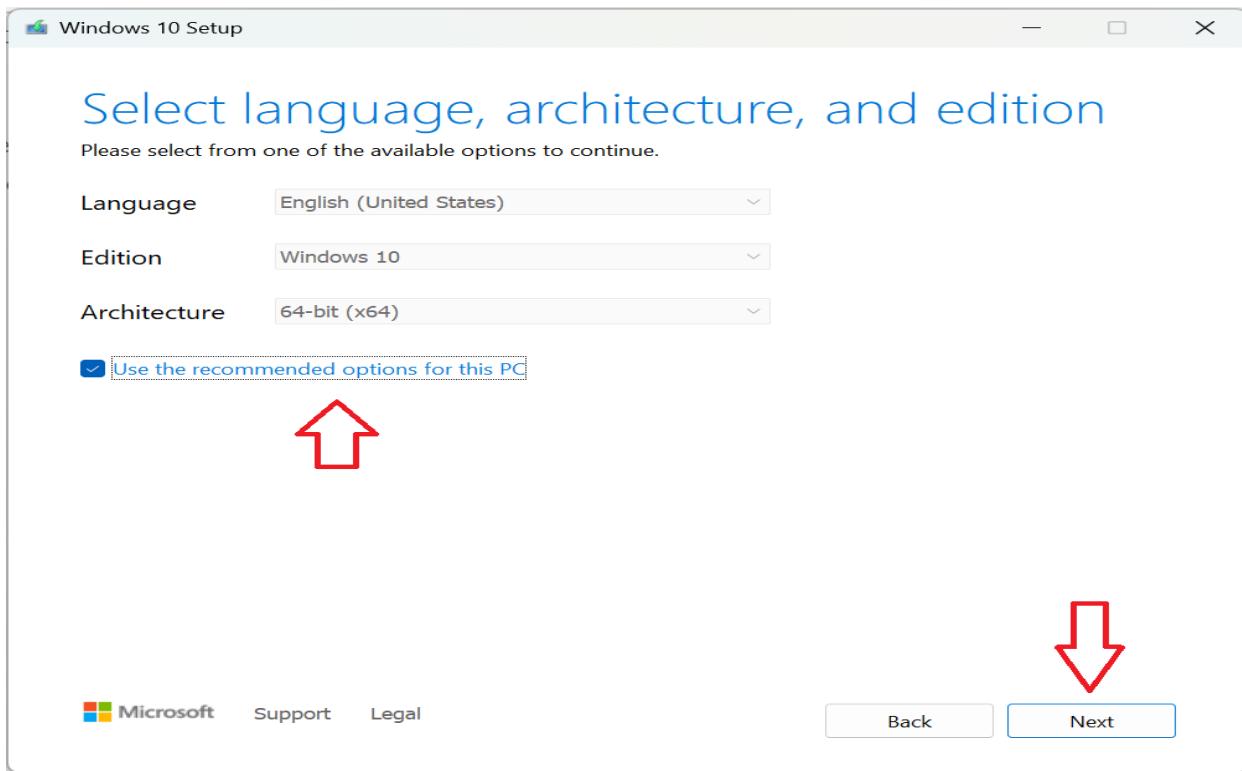
STEP 3: GO TO DOWNLOADS AND FIND THE TOOL AND RUN IT.



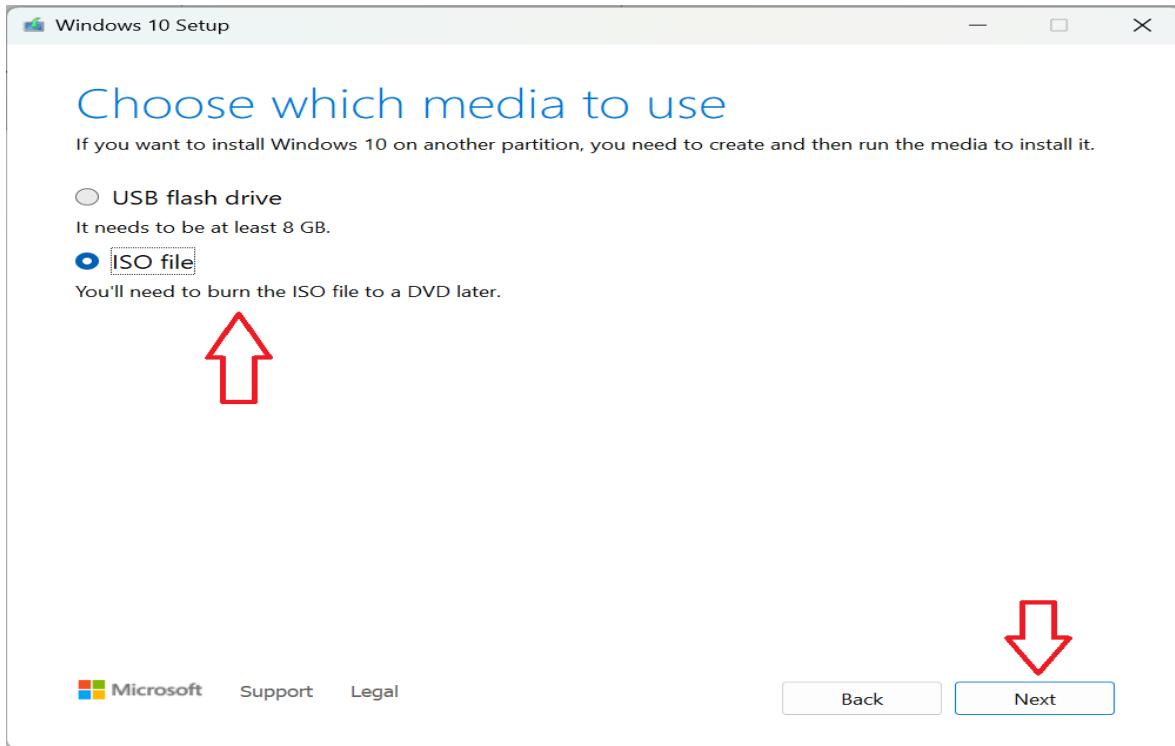
STEP 4: SELECT THE CREATE INSTALLATION MEDIA (ISO) AND THEN CLICK ON NEXT



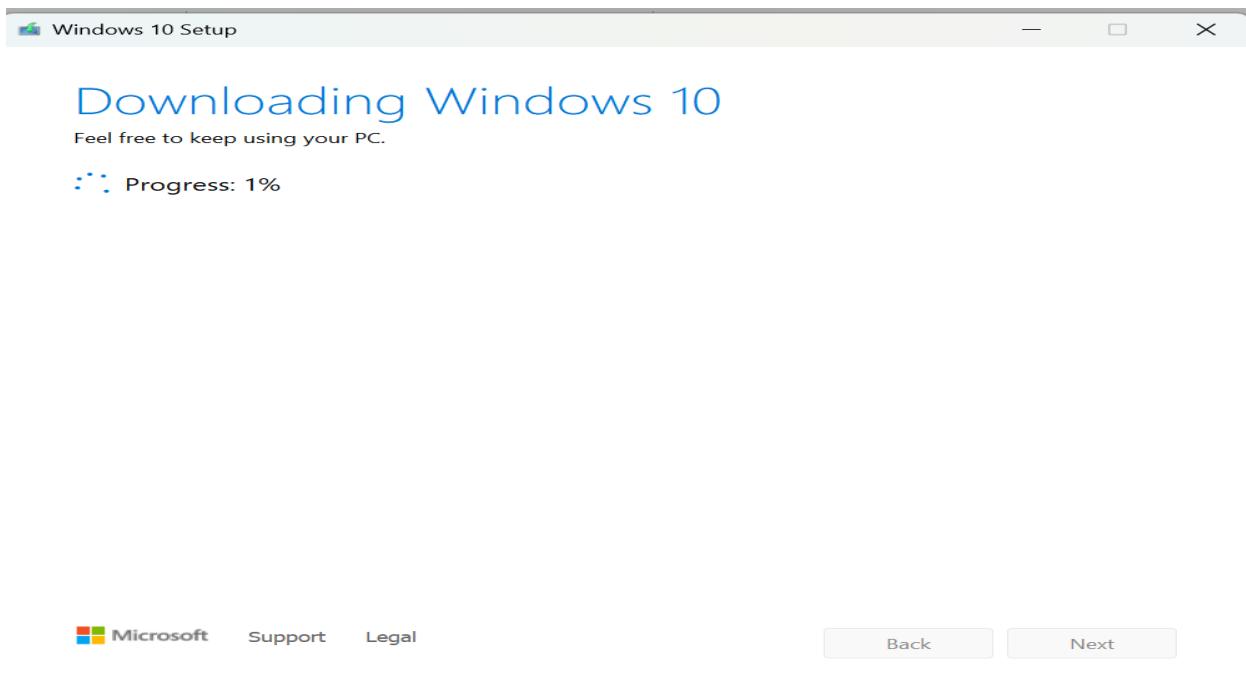
STEP 5: SELECT THE OPTIONS AS SHOWN BELOW



STEP 6: SELECT THE ISO FILE AND CLICK ON NEXT

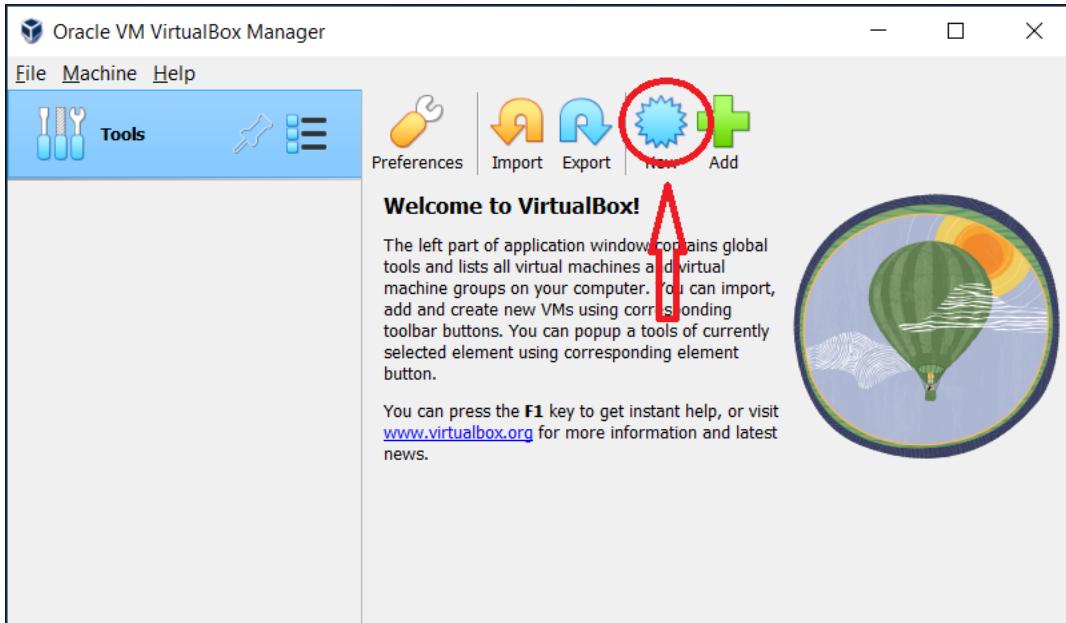


STEP 7: SELECT THE FILE DESTINATION AND CLICK ON NEXT WIN 10 FILE WILL START DOWNLOADING



INSTALLING THE WIN 10 IN VIRTUALBOX

STEP 1: OPEN THE VIRTUALBOX AND SELECT THE NEW OPTION

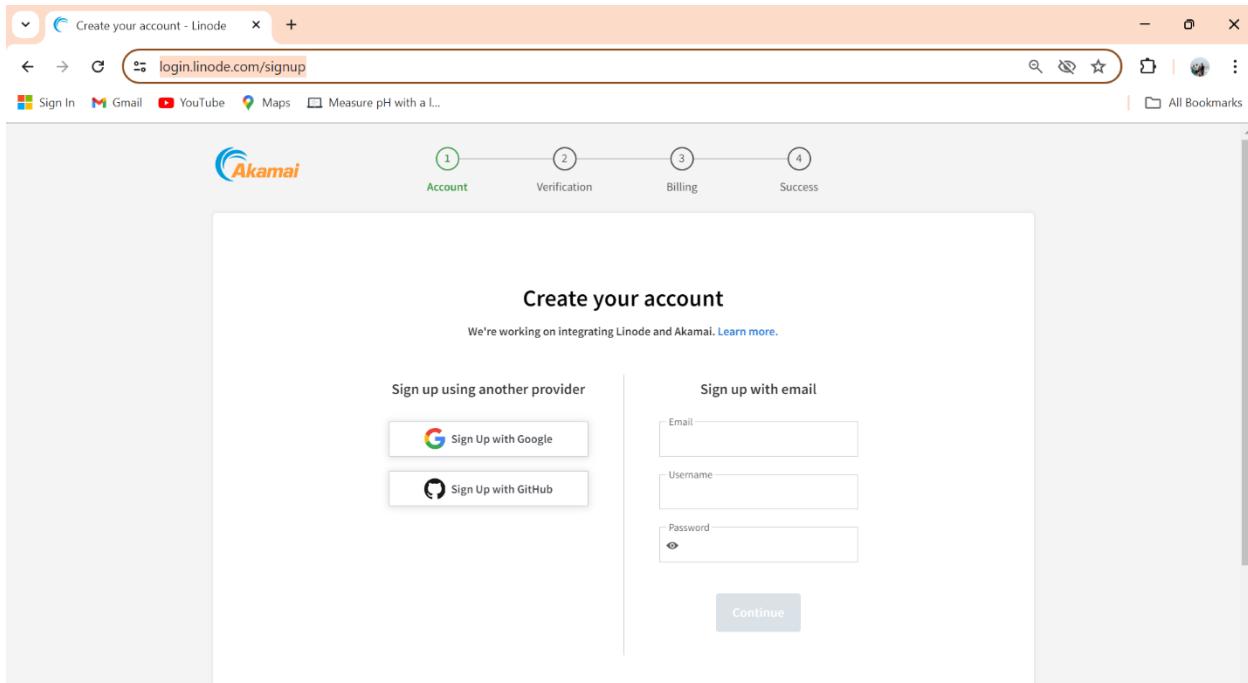


STEP 2: SELECT THE WIN 10 ISO FILE AND CLICK ON FINISH

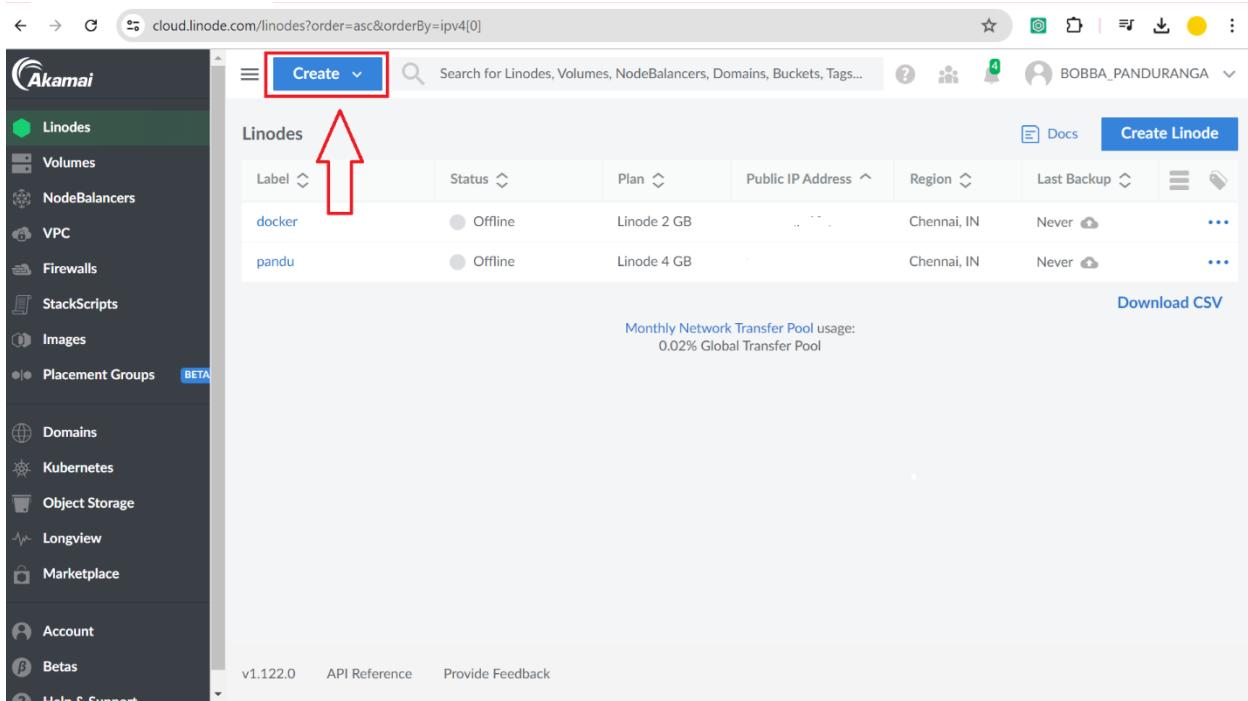
STEP 3: That is Windows 10 has been installed on the Virtual Machine. You can now open the Windows 10 operating system in VM. Set up a new username and password and use Windows 10 in VM for various purposes

STAGE 3 : CREATING ubuntu SERVER AND DOCKER ON akamai cloud manger

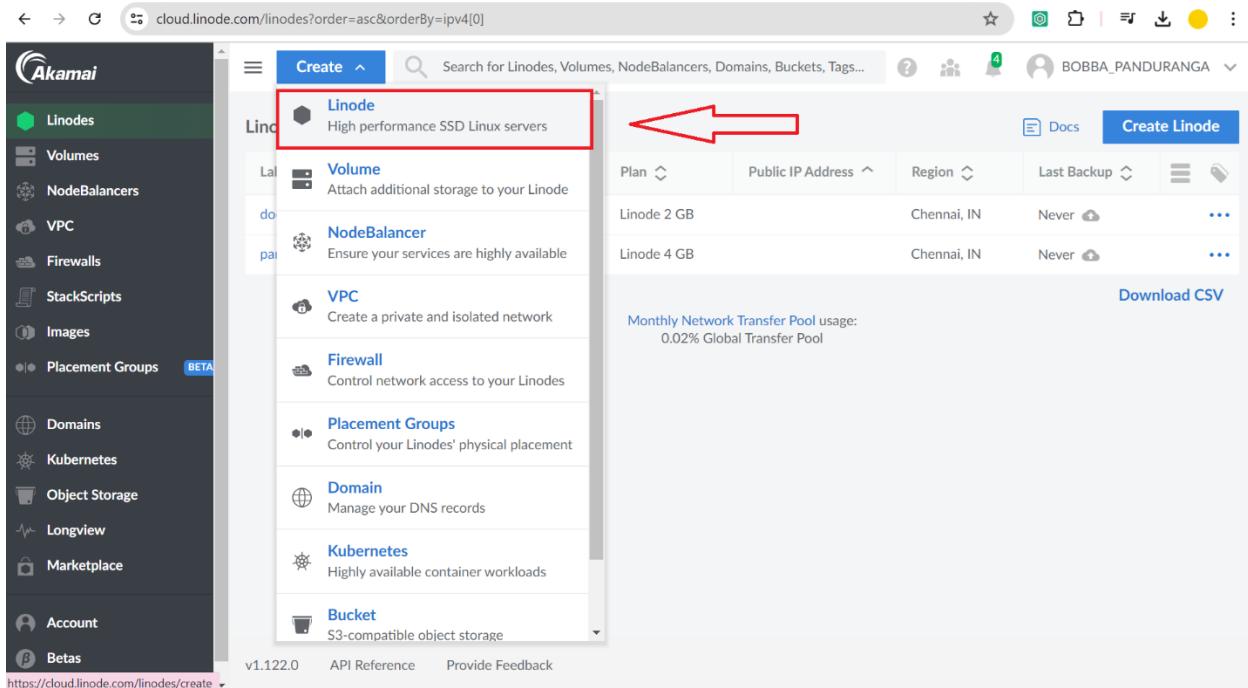
STEP 1: Go to This Website [Https://Login.Linode.Com/Signup](https://Login.Linode.Com/Signup) And Signup First And Create Your Account



STEP 2: Click On Create To Create The Ubuntu Server

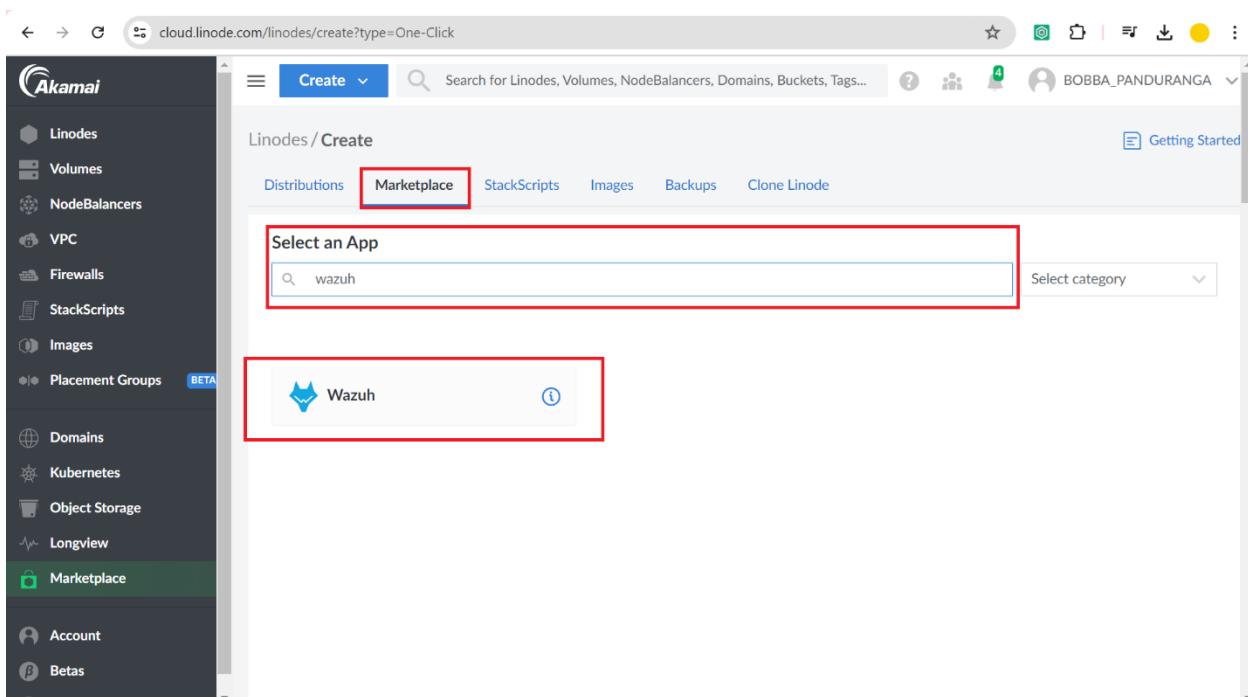


STEP 3: Click On Linode



The screenshot shows the Linode Cloud interface. On the left, there's a sidebar with various services: Linodes, Volumes, NodeBalancers, VPC, Firewalls, StackScripts, Images, Placement Groups (Beta), Domains, Kubernetes, Object Storage, Longview, Marketplace, Account, and Betas. The 'Linodes' option is highlighted with a red box. The main content area has a 'Create' button at the top left. A search bar is followed by a dropdown menu with 'Linode' selected, also highlighted with a red box. Below the search bar, there are links for Volume, NodeBalancer, VPC, Firewall, Placement Groups, Domain, Kubernetes, and Bucket. The main table lists two Linode instances: 'Linode 2 GB' and 'Linode 4 GB', both located in 'Chennai, IN' with 'Never' last backup. A red arrow points from the 'Linode' link in the dropdown to the 'Linode' link in the table header.

STEP 4: Select The Marketplace → Search Wazuh On Select An App →Select Wazuh:



The screenshot shows the 'Linodes / Create' page. The sidebar on the left is identical to the previous one. At the top, there are tabs: Distributions, Marketplace (which is highlighted with a red box), StackScripts, Images, Backups, and Clone Linode. Below these tabs is a search bar with the placeholder 'Search for Linodes, Volumes, NodeBalancers, Domains, Buckets, Tags...'. To the right of the search bar is a 'Getting Started' link. The main area is titled 'Select an App' and contains a search input field with 'wazuh' typed into it. A red box highlights this search input field. Below the search field, the results show a single item: 'Wazuh', which is also highlighted with a red box. The Wazuh entry includes a small icon and a help icon (info symbol) to its right.

STEP 5: Scroll Down And Fill The Details As Shown Below

The screenshot shows the 'Wazuh Setup' configuration page. On the left is a sidebar with various Akamai services: Linodes, Volumes, NodeBalancers, VPC, Firewalls, StackScripts, Images, Placement Groups (BETA), Domains, Kubernetes, Object Storage, Longview, Marketplace (highlighted in green), Account, Betas, and Help & Support.

The main form has several input fields:

- Email address (for the Let's Encrypt SSL certificate) (required): user@domain.tld
- The limited sudo user to be created for the Linode: *No Capital ... (red box)
- Your Linode API token. This is needed to create your Wazuh server's DNS records: Enter your token (red box)
- Subdomain: www (red box)

Below these fields is a note: "These fields are additional configuration options and are not required for creation." It includes a section for "Disable root access over SSH?" with "Yes" and "No" radio buttons, where "No" is selected.

STEP 6: Select The Image File Make Sure That You Should Select The Ubuntu 22.04 LTS Only And Select Your Nearest Region

The screenshot shows the 'Select an Image' page. The sidebar is identical to the previous one.

The main area has three sections:

- Select an Image:** A dropdown menu showing 'Ubuntu 22.04 LTS' (highlighted with a red box and an arrow pointing to it).
- Region:** A dropdown menu labeled 'Region' with 'Select a Region' (highlighted with a red box).
- Linode Plan:** A section with tabs for Dedicated CPU (selected), Shared CPU, High Memory, GPU, and Premium CPU. Below it is a table for selecting a region to view plans and prices.

STEP 7: Select The Shared Cpu And Select The Linode 4 GB Plane As Shown Below

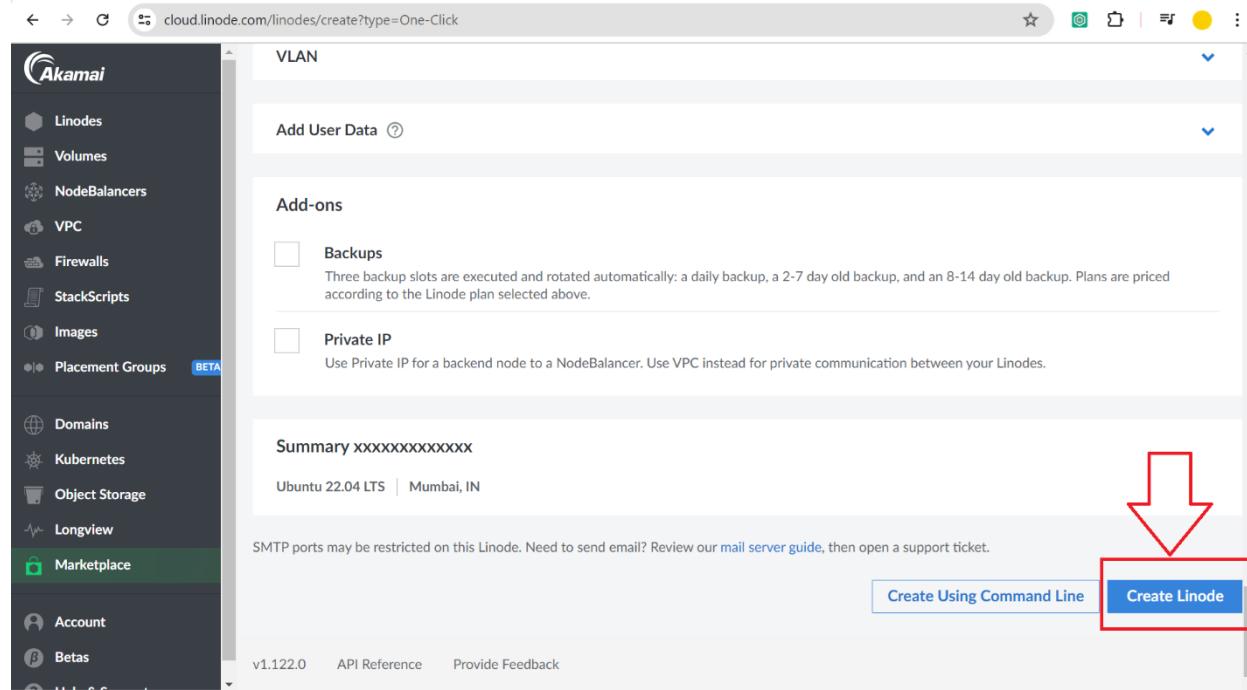
The screenshot shows the Akamai cloud interface for creating a new Linode. The left sidebar includes options like Linodes, Volumes, NodeBalancers, VPC, Firewalls, StackScripts, Images, Placement Groups (BETA), Domains, Kubernetes, Object Storage, Longview, Marketplace (selected), Account, and Betas. The top navigation bar shows the URL cloud.akamai.com/linodes/create?type=One-Click. The main content area has tabs for Dedicated CPU, Shared CPU (selected), High Memory, GPU, and Premium CPU. A sub-section titled "Shared CPU instances are good for medium-duty workloads and are a good mix of performance, resources, and price." provides a brief overview. A table lists various Linode plans with their details: Plan, Monthly cost, Hourly cost, RAM, CPUs, Storage, Transfer, and Network In / Out. The "Linode 4 GB" plan is highlighted with a red box.

Plan	Monthly	Hourly	RAM	CPUs	Storage	Transfer	Network In / Out
Nanode 1 GB	\$5	\$0.0075	1 GB	1	25 GB	1 TB	40 Gbps / 1 Gbps
Linode 2 GB	\$12	\$0.018	2 GB	1	50 GB	2 TB	40 Gbps / 2 Gbps
Linode 4 GB	\$24	\$0.036	4 GB	2	80 GB	4 TB	40 Gbps / 4 Gbps
Linode 8 GB	\$48	\$0.072	8 GB	4	160 GB	5 TB	40 Gbps / 5 Gbps
Linode 16 GB	\$96	\$0.144	16 GB	6	320 GB	8 TB	40 Gbps / 6 Gbps
Linode 32 GB	\$192	\$0.288	32 GB	8	640 GB	16 TB	40 Gbps / 7 Gbps
Linode 64 GB	\$384	\$0.576	64 GB	16	1280 GB	20 TB	40 Gbps / 9 Gbps
Linode 96 GB	\$576	\$0.864	96 GB	20	1920 GB	20 TB	40 Gbps / 10 Gbps
Linode 128 GB	\$768	\$1.152	128 GB	24	2560 GB	20 TB	40 Gbps / 11 Gbps
Linode 192 GB	\$1152	\$1.728	192 GB	32	3840 GB	20 TB	40 Gbps / 12 Gbps

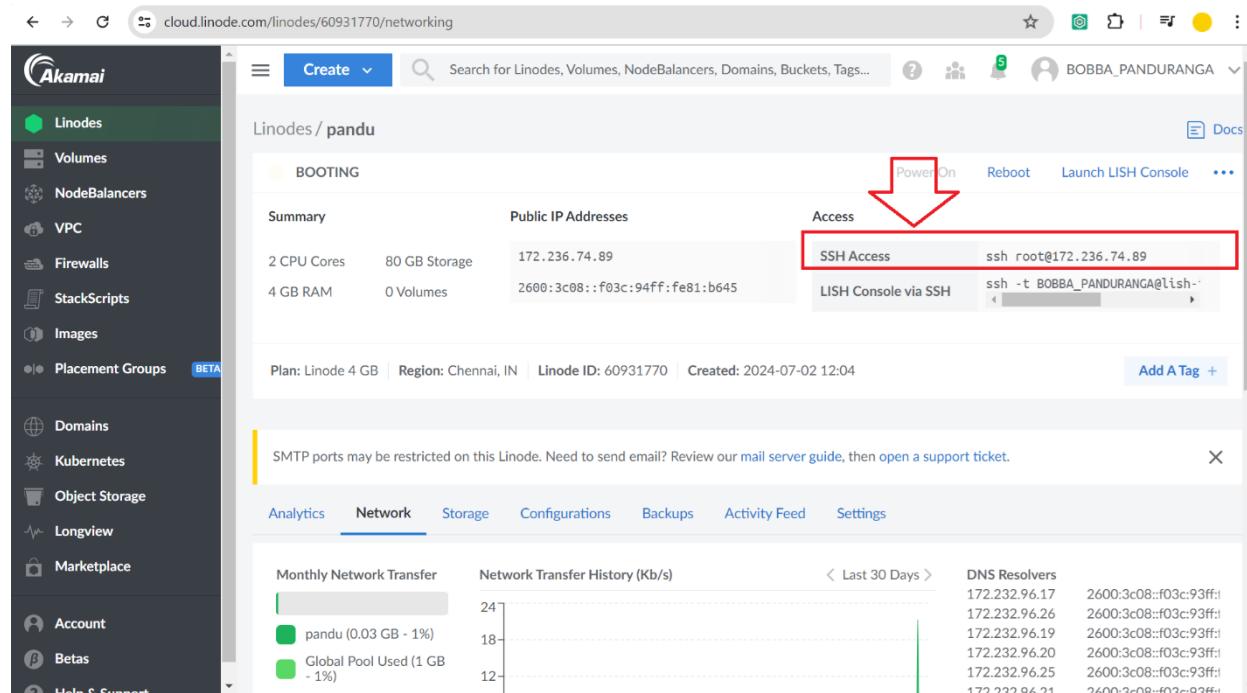
STEP 8: Now Fill The Details Linode Label As You Like And Give The Root Password As You Like Make Sure That You Remember That Root Password Again You Cant Able To Change

The screenshot shows the "Details" section of the Linode creation form. The left sidebar is identical to the previous screenshot. The "Linode Label" field contains the placeholder "xxxxxxxxxxxx" and is highlighted with a red box and an arrow. The "Root Password" field has a placeholder "Enter a password." and a strength indicator showing "Strength: Weak", also highlighted with a red box and an arrow. Other sections visible include "Add Tags", "Placement Groups in Mumbai, IN (ap-west)", and "SSH Keys".

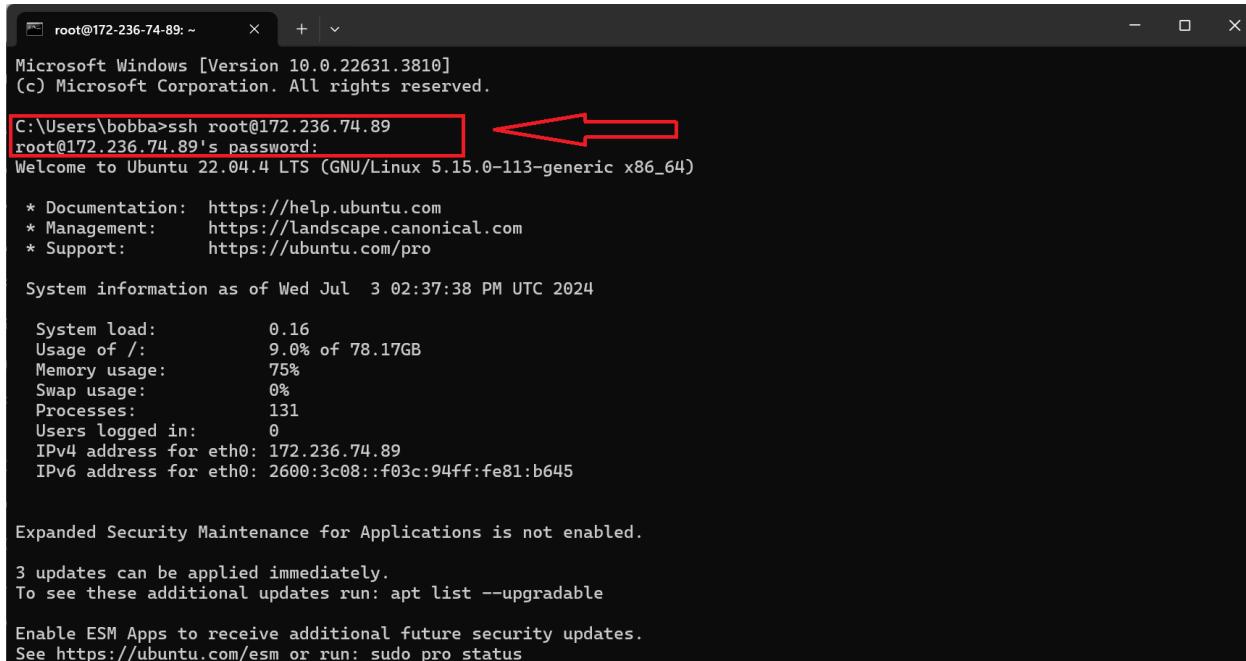
STEP 9: Click On Create The Linode Now You Are Good To Go



STEP 10: After creating the server now copy the SSH ACCESS CODE



STEP 11: GO TO CMD IN WINDOWS PAST THE SSH ACCESS CODE And Enter The Password As You Given Before



```
root@172-236-74-89: ~      + | ~
Microsoft Windows [Version 10.0.22631.3810]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bobba>ssh root@172.236.74.89
root@172.236.74.89's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-113-generic x86_64)

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

System information as of Wed Jul 3 02:37:38 PM UTC 2024

System load:          0.16
Usage of /:           9.0% of 78.17GB
Memory usage:         75%
Swap usage:           0%
Processes:            131
Users logged in:     0
IPv4 address for eth0: 172.236.74.89
IPv6 address for eth0: 2600:3c08::f03c:94ff:fe81:b645

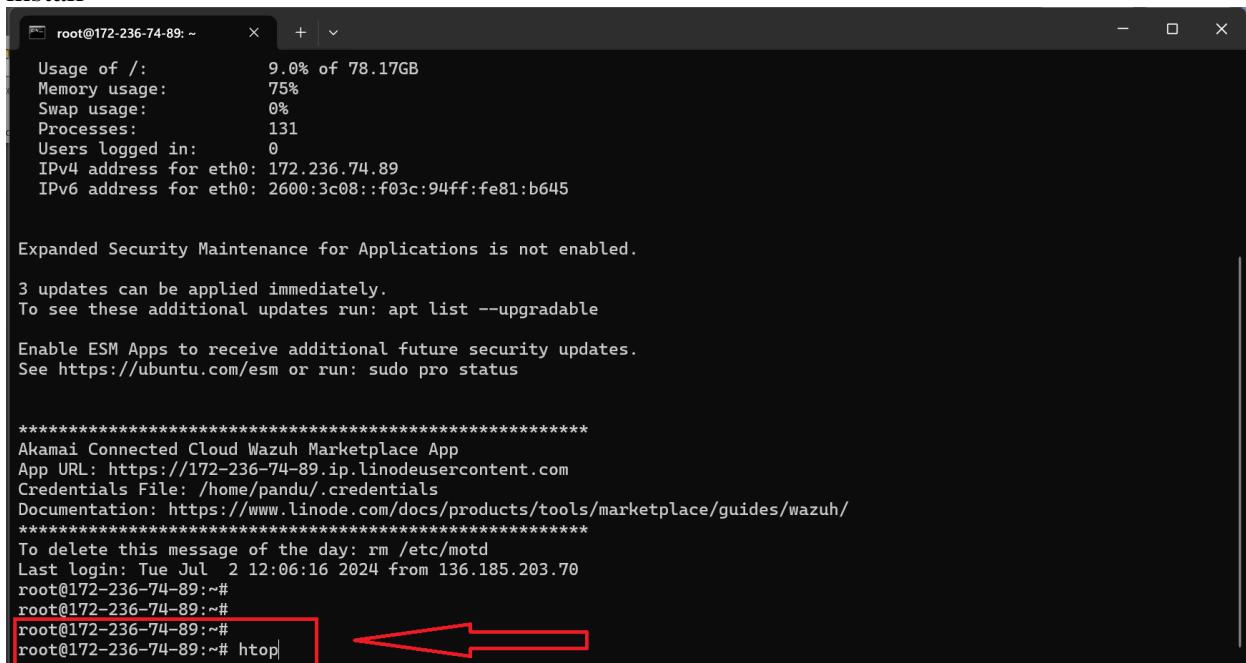
Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

STE:P12 to see the internal process such as

install



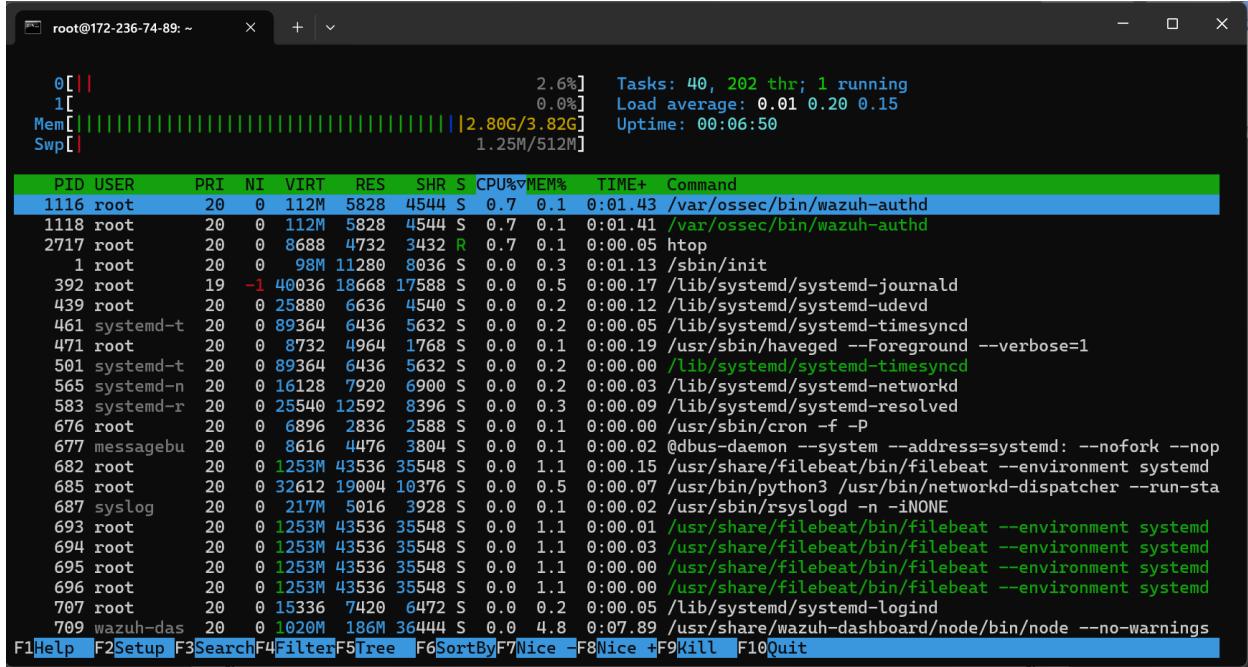
```
root@172-236-74-89: ~      + | ~
Usage of /:           9.0% of 78.17GB
Memory usage:         75%
Swap usage:           0%
Processes:            131
Users logged in:     0
IPv4 address for eth0: 172.236.74.89
IPv6 address for eth0: 2600:3c08::f03c:94ff:fe81:b645

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

*****
Akamai Connected Cloud Wazuh Marketplace App
App URL: https://172-236-74-89.ip.linodeusercontent.com
Credentials File: /home/pandu/.credentials
Documentation: https://www.linode.com/docs/products/tools/marketplace/guides/wazuh/
*****
To delete this message of the day: rm /etc/motd
Last login: Tue Jul 2 12:06:16 2024 from 136.185.203.70
root@172-236-74-89:~#
root@172-236-74-89:~#
root@172-236-74-89:~# htop
```



- Use the htpasswd command to reset the password for the Wazuh web interface. Replace admin with the appropriate username if different.

bash

Copy code

```
htpasswd -c /var/ossec/api/configuration/auth/user admin
```

- You will be prompted to enter and confirm the new password.

□ Restart Wazuh API:

- After resetting the password, restart the Wazuh API service to apply the changes.

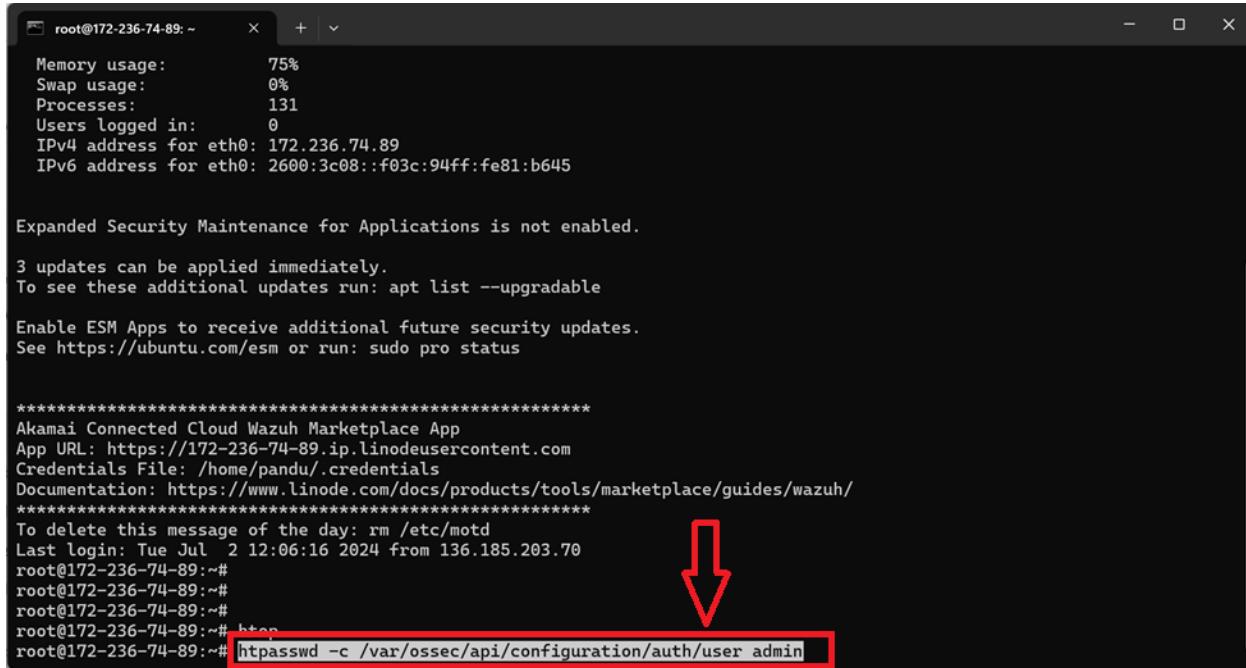
bash

Copy code

```
systemctl restart wazuh-api
```



STEP 13: If Want To Change The Passwaord Of Wazuh Login Tou Can Do By Apply Ing The Following Comand



```
root@172-236-74-89: ~
Memory usage: 75%
Swap usage: 0%
Processes: 131
Users logged in: 0
IPv4 address for eth0: 172.236.74.89
IPv6 address for eth0: 2600:3c08::f03c:94ff:fe81:b645

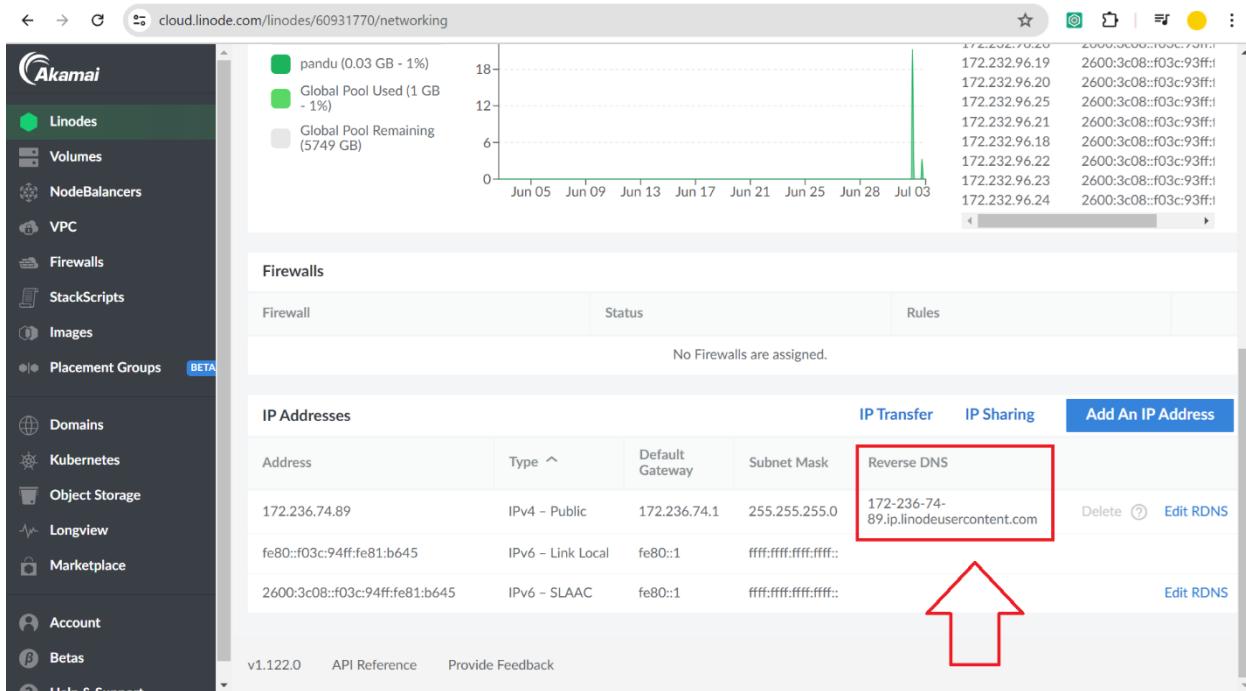
Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

*****
Akamai Connected Cloud Wazuh Marketplace App
App URL: https://172-236-74-89.ip.linodeusercontent.com
Credentials File: /home/pandu/.credentials
Documentation: https://www.linode.com/docs/products/tools/marketplace/guides/wazuh/
*****
To delete this message of the day: rm /etc/motd
Last login: Tue Jul 2 12:06:16 2024 from 136.185.203.70
root@172-236-74-89:~#
root@172-236-74-89:~#
root@172-236-74-89:~# htpasswd -c /var/ossec/api/configuration/auth/user admin
```

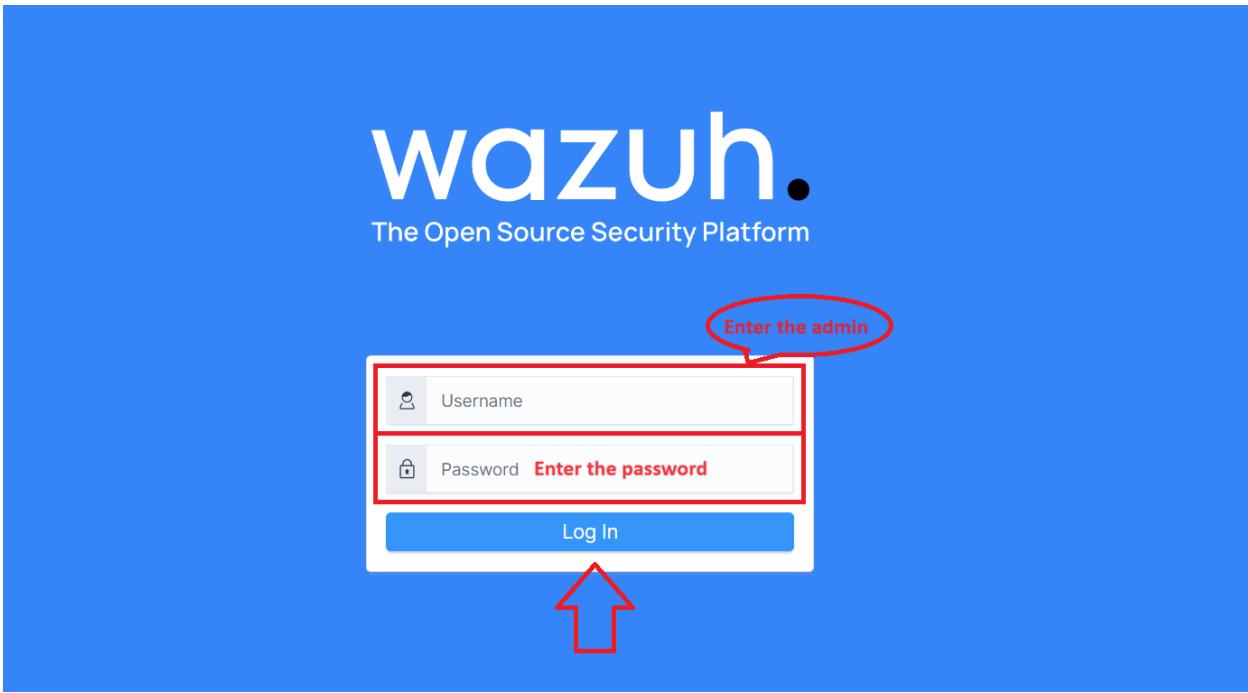
STEP 14: Copy The Ip As I Shown Below And Past It In New Tab Url Box



cloud.linode.com/linodes/60931770/networking

IP Addresses

Address	Type	Default Gateway	Subnet Mask	IP Transfer	IP Sharing	Add An IP Address
172.236.74.89	IPv4 – Public	172.236.74.1	255.255.255.0	Reverse DNS	172-236-74-89.ip.linodeusercontent.com	Delete Edit RDNS
fe80::f03c:94ff:fe81:b645	IPv6 – Link Local	fe80::1	ffff:ffff:ffff:ffff::			Edit RDNS
2600:3c08::f03c:94ff:fe81:b645	IPv6 – SLAAC	fe80::1	ffff:ffff:ffff:ffff::			



The image shows the Wazuh dashboard. The top section features the Wazuh logo and a series of status indicators: "Check Wazuh API connection" (green checkmark), "Check Wazuh API version" (blue circle), "Check alerts index pattern" (green checkmark), "Check monitoring index pattern" (green checkmark), and "Check statistics index pattern" (green checkmark). The main content area of the dashboard is currently empty.

The screenshot shows the Wazuh dashboard with the URL [https://172-236-74-89.ip.linodeusercontent.com/app/wazuh#/overview/?_g=\(filters:!\(\),refreshInterval:\(pause:!t,value:0\),time:\(from:now-24h,to:now\)\)](https://172-236-74-89.ip.linodeusercontent.com/app/wazuh#/overview/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-24h,to:now))). The dashboard displays the following statistics:

Total agents	Active agents	Disconnected agents	Pending agents	Never connected agents
0	0	0	0	0

A message at the top states: "⚠️ No agents were added to this manager. [Add agent](#)". Below this, there are two main sections: "SECURITY INFORMATION MANAGEMENT" and "AUDITING AND POLICY MONITORING".

- SECURITY INFORMATION MANAGEMENT:**
 - Security events:** Browse through your security alerts, identifying issues and threats in your environment.
 - Integrity monitoring:** Alerts related to file changes, including permissions, content, ownership and attributes.
- AUDITING AND POLICY MONITORING:**
 - Policy monitoring:** Verify that your systems are configured according to your security policies baseline.
 - System auditing:** Audit users behavior, monitoring command execution and alerting on access to critical files.
 - Security configuration assessment:** Scan your assets as part of a configuration assessment audit.

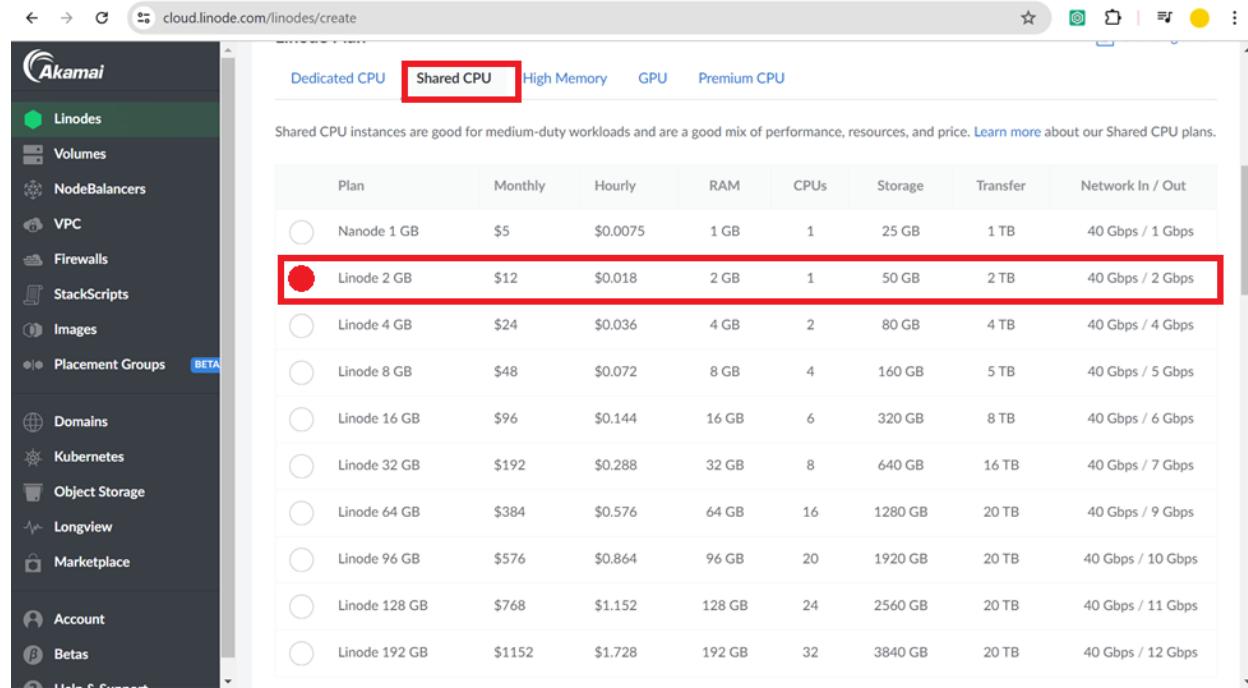
Creating the dock server:

STEP 1: Create the new server by following the below commands

- Click on the create
- Select the distributions option
- Select the os imager as I shown below it must be Ubuntu 24.04.LTS

The screenshot shows the Linode creation interface on the cloud.linode.com website. The URL is <https://cloud.linode.com/linodes/create>. The "Distributions" tab is selected. In the "Choose an OS" section, "Ubuntu 24.04 LTS" is highlighted with a red box. Other options listed include "Ubuntu 22.04 LTS" and "Ubuntu 20.04 LTS".

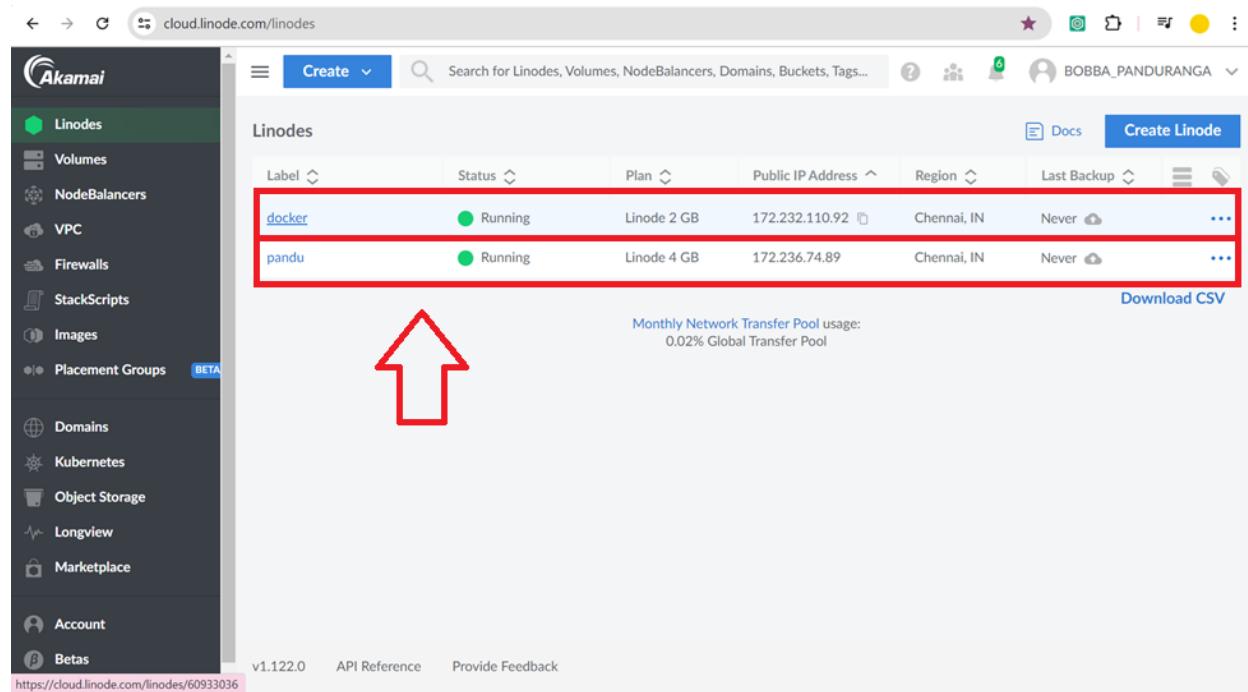
STEP 2: Select The Shared CPU And Select The 2gb Version And Click On Create Linode



The screenshot shows the Linode creation interface. The 'Shared CPU' tab is selected, indicated by a red box. Below it, a table lists various Linode plans. The 'Linode 2 GB' plan is highlighted with a red box. The table columns include Plan, Monthly price, Hourly price, RAM, CPUs, Storage, Transfer, and Network In / Out.

Plan	Monthly	Hourly	RAM	CPUs	Storage	Transfer	Network In / Out
Nanode 1 GB	\$5	\$0.0075	1 GB	1	25 GB	1 TB	40 Gbps / 1 Gbps
Linode 2 GB	\$12	\$0.018	2 GB	1	50 GB	2 TB	40 Gbps / 2 Gbps
Linode 4 GB	\$24	\$0.036	4 GB	2	80 GB	4 TB	40 Gbps / 4 Gbps
Linode 8 GB	\$48	\$0.072	8 GB	4	160 GB	5 TB	40 Gbps / 5 Gbps
Linode 16 GB	\$96	\$0.144	16 GB	6	320 GB	8 TB	40 Gbps / 6 Gbps
Linode 32 GB	\$192	\$0.288	32 GB	8	640 GB	16 TB	40 Gbps / 7 Gbps
Linode 64 GB	\$384	\$0.576	64 GB	16	1280 GB	20 TB	40 Gbps / 9 Gbps
Linode 96 GB	\$576	\$0.864	96 GB	20	1920 GB	20 TB	40 Gbps / 10 Gbps
Linode 128 GB	\$768	\$1.152	128 GB	24	2560 GB	20 TB	40 Gbps / 11 Gbps
Linode 192 GB	\$1152	\$1.728	192 GB	32	3840 GB	20 TB	40 Gbps / 12 Gbps

STEP 3: Now select the docker and open it



The screenshot shows the Linode list interface. Two instances are listed: 'docker' and 'pandu', both in a 'Running' status. A red box highlights both entries. A large red arrow points upwards towards the highlighted instances. The table columns include Label, Status, Plan, Public IP Address, Region, and Last Backup.

Label	Status	Plan	Public IP Address	Region	Last Backup
docker	Running	Linode 2 GB	172.232.110.92	Chennai, IN	Never
pandu	Running	Linode 4 GB	172.236.74.89	Chennai, IN	Never

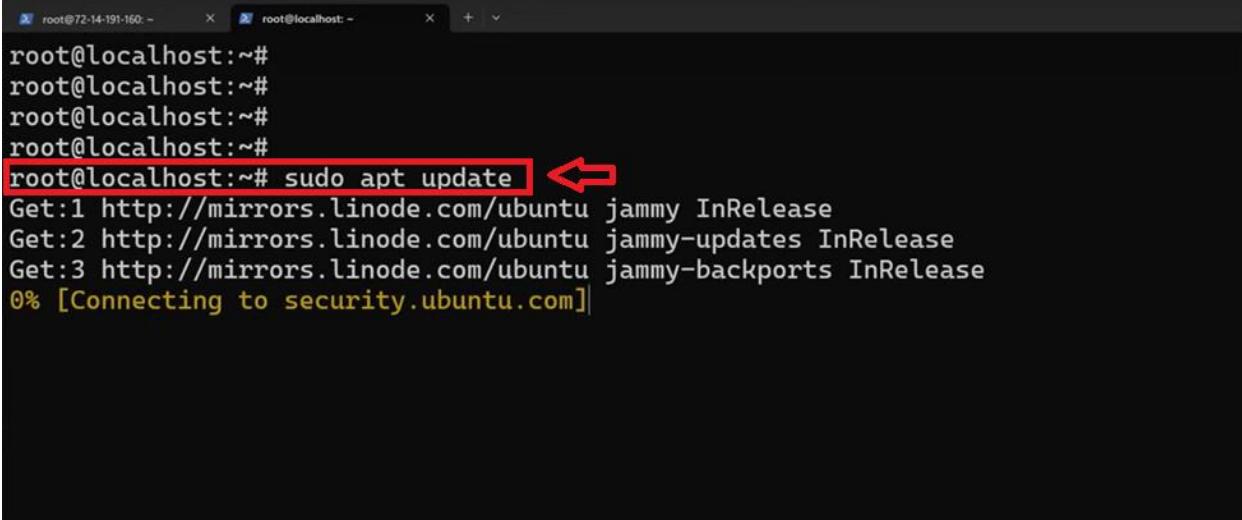
STEP 4: Now Copy The Ssh Access

The screenshot shows the Linode Cloud interface. On the left, there's a sidebar with various services: Linodes, Volumes, NodeBalancers, VPC, Firewalls, StackScripts, Images, Placement Groups (BETA), Domains, Kubernetes, Object Storage, Longview, Marketplace, Account, and Betas. The main area shows a single running Linode named 'docker'. It has 1 CPU Core, 50 GB Storage, 2 GB RAM, and 0 Volumes. Under the 'Access' section, it shows the Public IP Address (172.232.110.92) and the SSH command: ssh root@172.232.110.92. A red box highlights this command, and a red arrow points to the copy icon (a clipboard with a plus sign) next to it.

STEP 5: Past It In Cmd (Regular Mode) Enter The Password As Well

```
PS C:\Users\chuck> ssh root@173.255.204.175 ↗
The authenticity of host '173.255.204.175 (173.255.204.175)' can't be established.
ED25519 key fingerprint is SHA256:Uthxd92pFSV5TkLC6pAoGEpW8pVhrDrfeBi4cdIKaSU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '173.255.204.175' (ED25519) to the list of known hosts.
root@173.255.204.175's password:
```

STEP 6: Now create the docker with the following commands



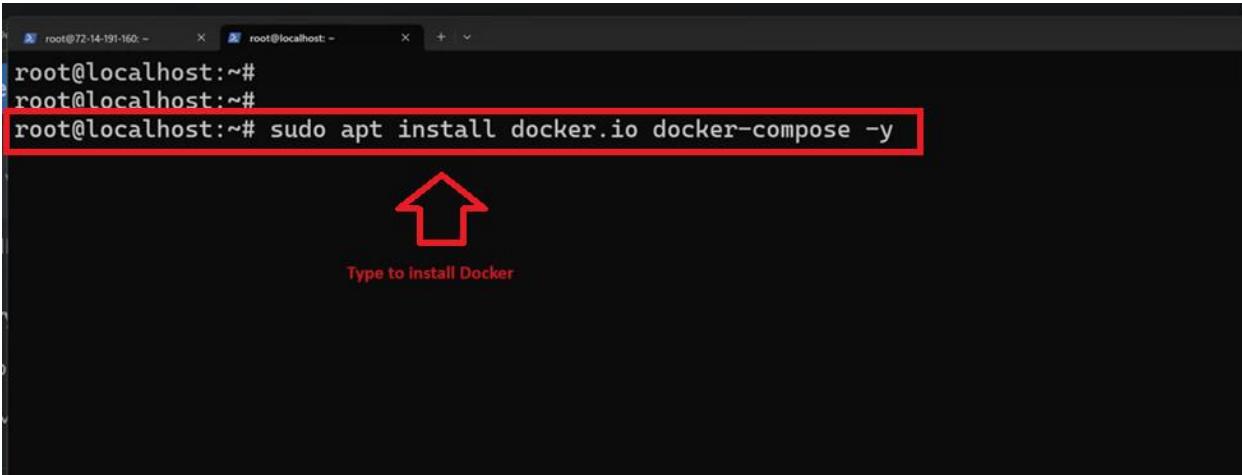
```
root@localhost:~#  
root@localhost:~#  
root@localhost:~#  
root@localhost:~#  
root@localhost:~# sudo apt update
```

Get:1 http://mirrors.linode.com/ubuntu jammy InRelease
Get:2 http://mirrors.linode.com/ubuntu jammy-updates InRelease
Get:3 http://mirrors.linode.com/ubuntu jammy-backports InRelease
0% [Connecting to security.ubuntu.com]

❖ Must visit this link to learn more about :

- <https://documentation.wazuh.com/current/deployment-options/docker/wazuh-container.html>

STEP 7: Type The Following Command To Install Docker In Your Server



```
root@localhost:~#  
root@localhost:~#  
root@localhost:~# sudo apt install docker.io docker-compose -y
```

Type to install Docker

STEP 8 : Clone The Docker By Typing The Clone Command As Shown Below

```
root@localhost:~# git clone https://github.com/wazuh/wazuh-docker.git -b v4.4.5
Cloning into 'wazuh-docker'...
remote: Enumerating objects: 10586, done.
remote: Counting objects: 100% (626/626), done.
remote: Compressing objects: 100% (337/337), done.
remote: Total 10586 (delta 320), reused 547 (delta 263), pack-reused 9960
Receiving objects: 100% (10586/10586), 313.86 MiB | 43.15 MiB/s, done.
Resolving deltas: 100% (5397/5397), done.
Note: switching to 'ce8dd29425e4bc03135b5f2b71d7425b7d865f5d'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

git switch -c <new-branch-name>

Or undo this operation with:

git switch -

Turn off this advice by setting config variable advice.detachedHead to false
```

root@localhost:~# ls
wazuh-docker
root@localhost:~# |

```
root@localhost:~# ls
wazuh-docker
root@localhost:~# cd wazuh-docker/
root@localhost:~/wazuh-docker# ls
build-docker-images CHANGELOG.md indexer-certs-creator LICENSE multi-node README.md single-node VERSION
root@localhost:~/wazuh-docker# cd single-node/
root@localhost:~/wazuh-docker/single-node# docker-compose -f generate-indexer-certs.yml run --rm generator

Digest: sha256:73cd153b2d533c4b7c4a52bb6411a8bb03eaeb95f692da61e6953e4555c1526
Status: Downloaded newer image for wazuh/wazuh-dashboard:4.4.5
Creating single-node_wazuh.indexer_1 ... done
Creating single-node_wazuh.manager_1 ... done
Creating single-node_wazuh.dashboard_1 ... done
root@localhost:~/wazuh-docker/single-node# docker stats
```

```

root@localhost: ~
CONTAINER ID  NAME          CPU %     MEM USAGE / LIMIT   MEM %     NET I/O      BLOCK I/O
PIDS
d345e358352d  single-node_wazuh.dashboard_1  0.20%    155.5MiB / 1.918GiB 7.92%    251kB / 264kB   186MB / 67.4M
B 12
0f1f6ee63ed87  single-node_wazuh.indexer_1    0.86%    1.031GiB / 1.918GiB 53.76%   296kB / 244kB   1.66GB / 730M
B 65
6dc0e36b697c  single-node_wazuh.manager_1    0.80%    257.5MiB / 1.918GiB 13.11%   58.6kB / 85.7kB  421MB / 87.1M
B 155
|

```

STEP 9: Now Done The Docker Is Installed In Server Now Copy The <Ip> In New Tab As <https://<Ip>>

SMTP ports may be restricted on this Linode. Need to send email? Review our [mail server guide](#), then open a support ticket.

Analytics Network Storage Configurations Backups Activity Feed Settings

Monthly Network Transfer Network Transfer History (Kb/s) < Last 30 Days >

Network Transfer History (Kb/s)

100
75
50
25
0

Jun 05 Jun 08 Jun 11 Jun 14 Jun 17 Jun 20 Jun 23 Jun 26 Jun 29 Jul 03

Global Pool Used (1 GB - 1%)
Global Pool Remaining (5749 GB)

DNS Resolvers

172.232.96.17 2600:3c08::f03c:93ff:fe7c:1135
172.232.96.26 2600:3c08::f03c:93ff:fe7c:11f8
172.232.96.19 2600:3c08::f03c:93ff:fe7c:11d2
172.232.96.20 2600:3c08::f03c:93ff:fe7c:11a7
172.232.96.25 2600:3c08::f03c:93ff:fe7c:11ad
172.232.96.21 2600:3c08::f03c:93ff:fe7c:110a
172.232.96.18 2600:3c08::f03c:93ff:fe7c:11f9
172.232.96.22 2600:3c08::f03c:93ff:fe7c:1137
172.232.96.23 2600:3c08::f03c:93ff:fe7c:11db
172.232.96.24 2600:3c08::f03c:93ff:fe7c:1164

Firewalls

Firewall	Status	Rules
No Firewalls are assigned.		

IP Addresses

Address	Type	Default Gateway	Subnet Mask	Reverse DNS	IP Transfer	IP Sharing	Add An IP Address
172.232.110.92	IPv4 - Public	172.232.110.1	255.255.255.0	172-232-110-92.ip.linodeusercontent.com	Delete	Edit RDNS	
fe80::f03c:94ff:fe81:cbcc	IPv6 - Link Local	fe80::1	ffff:ffff:ffff:ffff::				Edit RDNS
2600:3c08::f03c:94ff:fe81:cbcc	IPv6 - SLAAC	fe80::1	ffff:ffff:ffff:ffff::				

copy and paste it in url bar on new window as <https://<id>>

The screenshot shows the Wazuh interface with the title "wazuh." at the top. Below it is a list of connectivity checks:

- Check Wazuh API connection ✓
- Check Wazuh API version ⓘ
- Check alerts index pattern ✓
- Check monitoring index pattern ✓
- Check statistics index pattern ✓

Now We Have To Add The Pc To Our Sever

The screenshot shows the Wazuh interface with the title "wazuh." and a "Modules" tab selected. It displays the following agent counts:

Total agents	Active agents	Disconnected agents	Pending agents	Never connected agents
0	0	0	0	0

A message indicates: "⚠ No agents were added to this manager. [Add agent](#)".

The page is divided into two main sections:

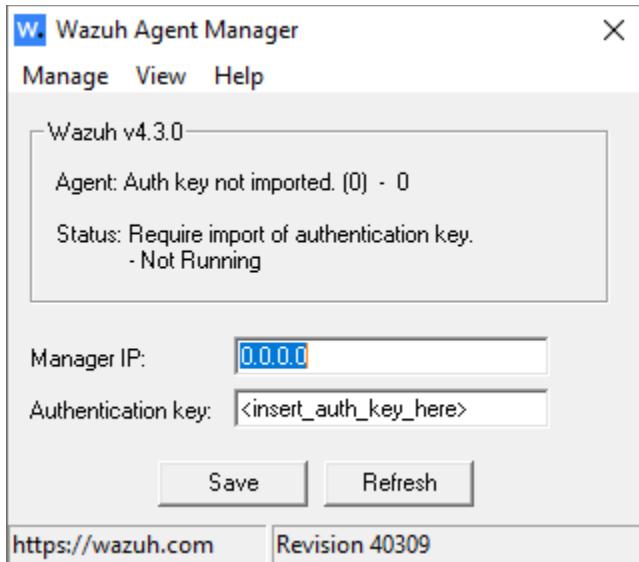
- SECURITY INFORMATION MANAGEMENT**:
 - Security events: Browse through your security alerts, identifying issues and threats in your environment.
 - Integrity monitoring: Alerts related to file changes, including permissions, content, ownership and attributes.
- AUDITING AND POLICY MONITORING**:
 - Policy monitoring: Verify that your systems are configured according to your security policies baseline.
 - System auditing: Audit users behavior, monitoring command execution and alerting on access to critical files.
 - Security configuration assessment: Scan your assets as part of a configuration assessment audit.

STEP 1: Download The Wazuh Agent From Below Link

<https://documentation.wazuh.com/4.7/installation-guide/wazuh-agent/index.html>

STEP 2: Install The Wazuh-Agent And Type The IP That You Have Pasted In URL Bar

STEP 3: After Click On Save And Go To Manage Option And Click On Start Option



STEP 4: Now Click On The Agent Option

A screenshot of the Wazuh web interface. At the top, there's a navigation bar with icons for home, search, and modules. The main content area shows statistics for agents:

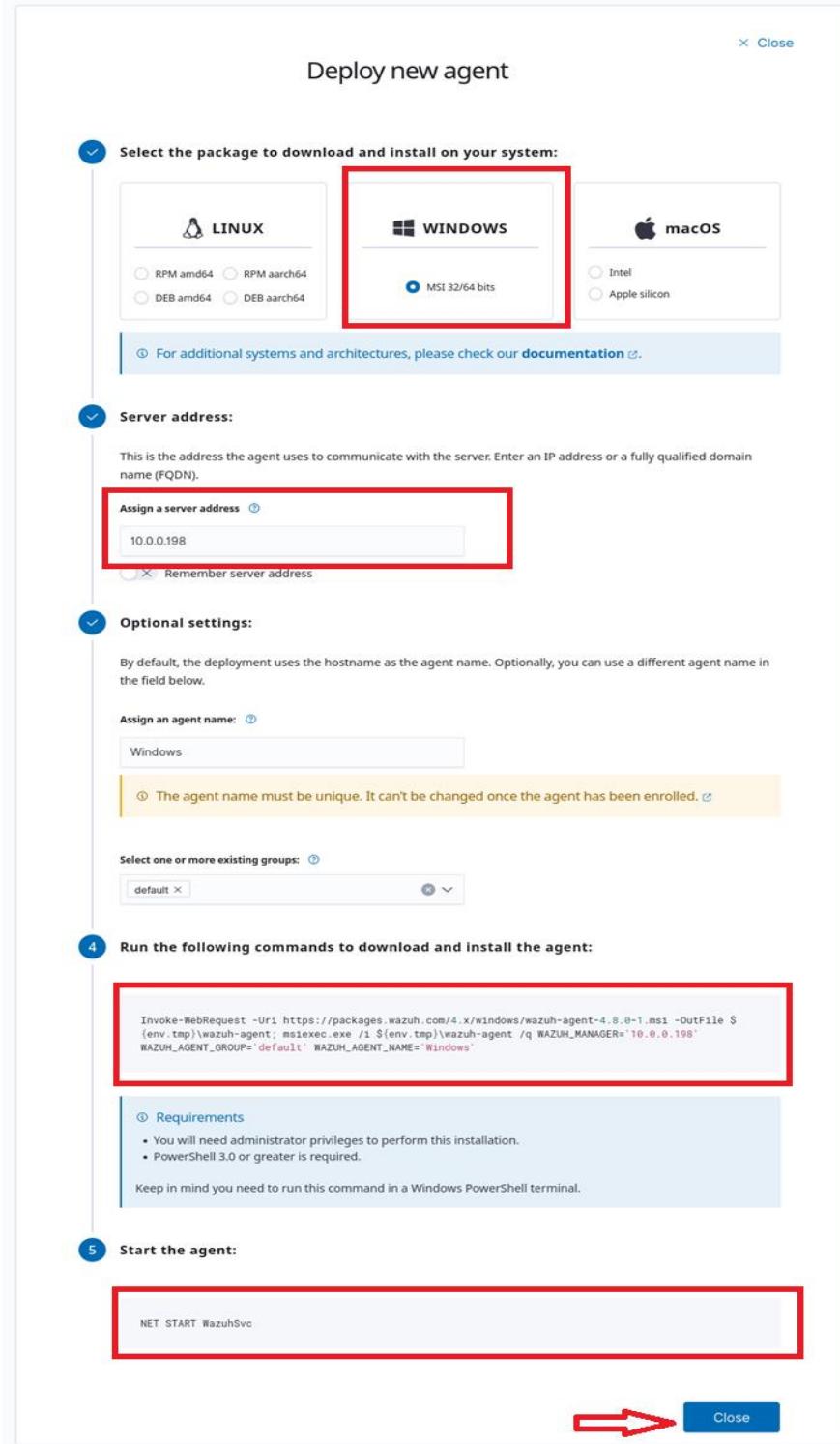
Total agents	Active agents	Disconnected agents	Pending agents	Never connected agents
0	0	0	0	0

Below this, a yellow banner states 'No agents were added to this manager.' with a red-bordered 'Add agent' button. The interface is divided into two main sections: 'SECURITY INFORMATION MANAGEMENT' and 'AUDITING AND POLICY MONITORING', each containing several sub-options with brief descriptions.

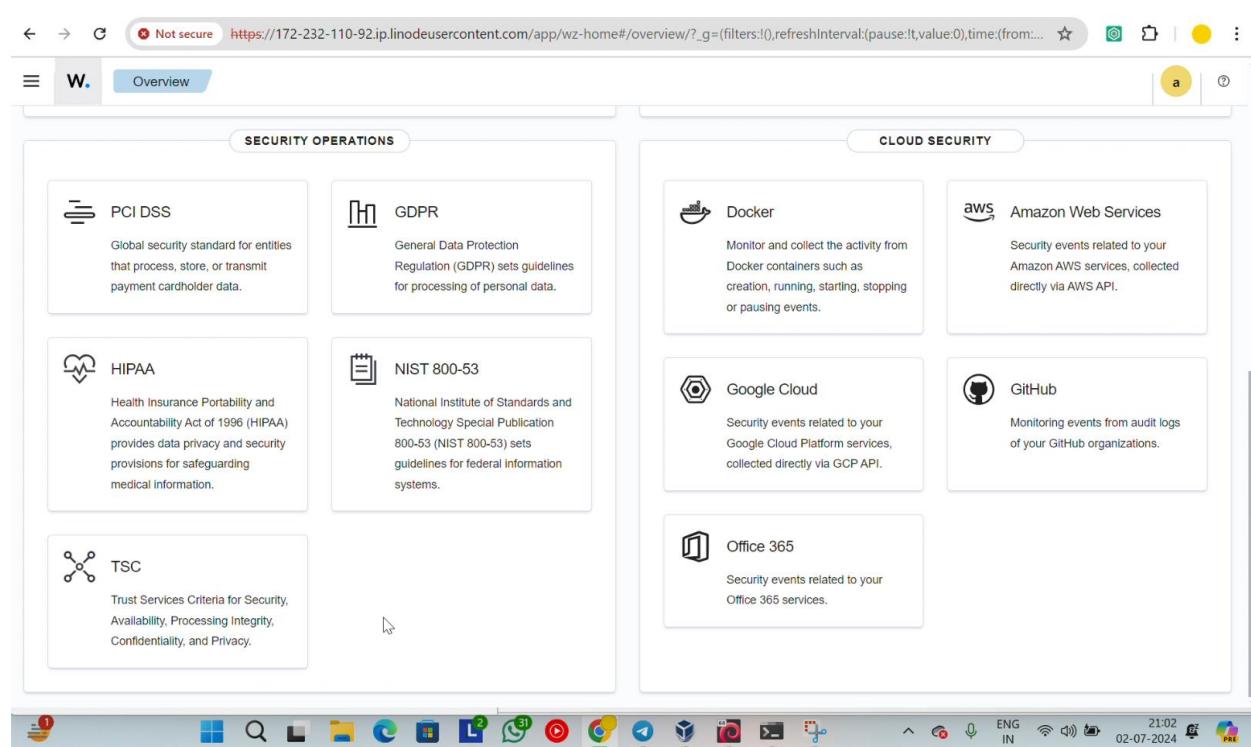
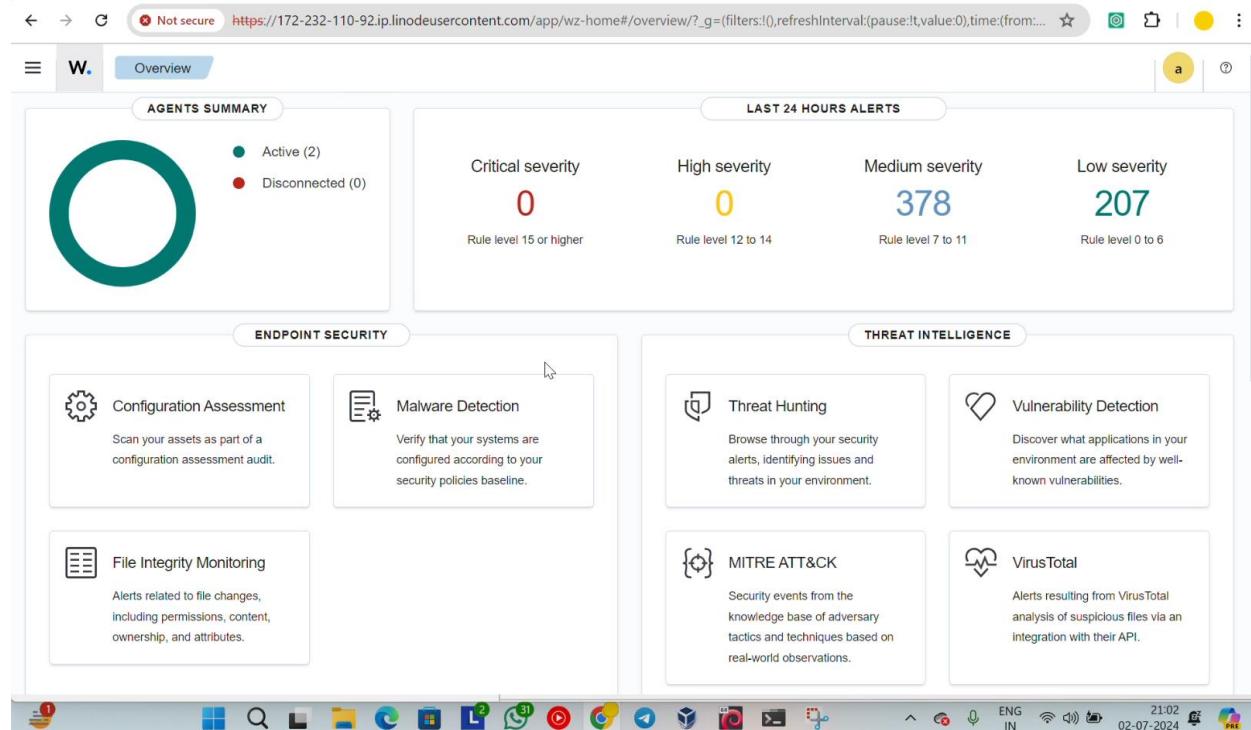
STEP 5: Enter The IP As You Enter In URL Bar

STEP 6: Copy The Command As You Got And Go To Windows Powershell (Admin) And Copy Them

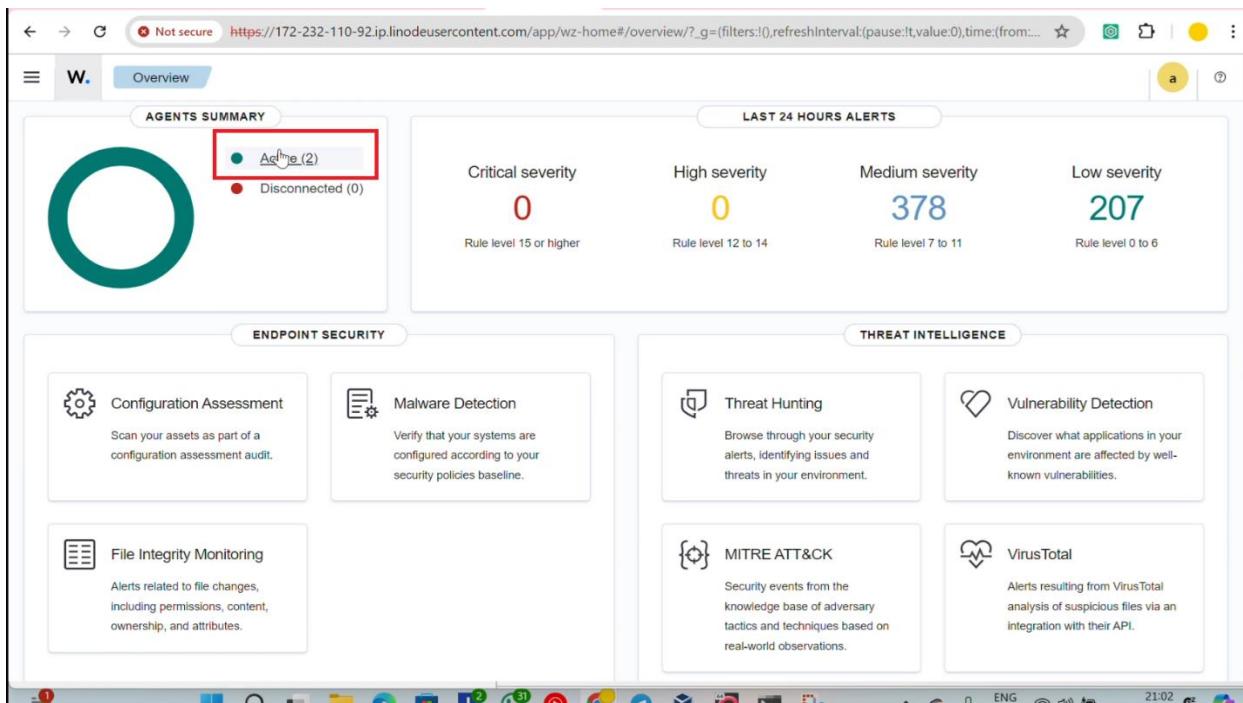
STEP 7: Click On The Close Button As Shown Below



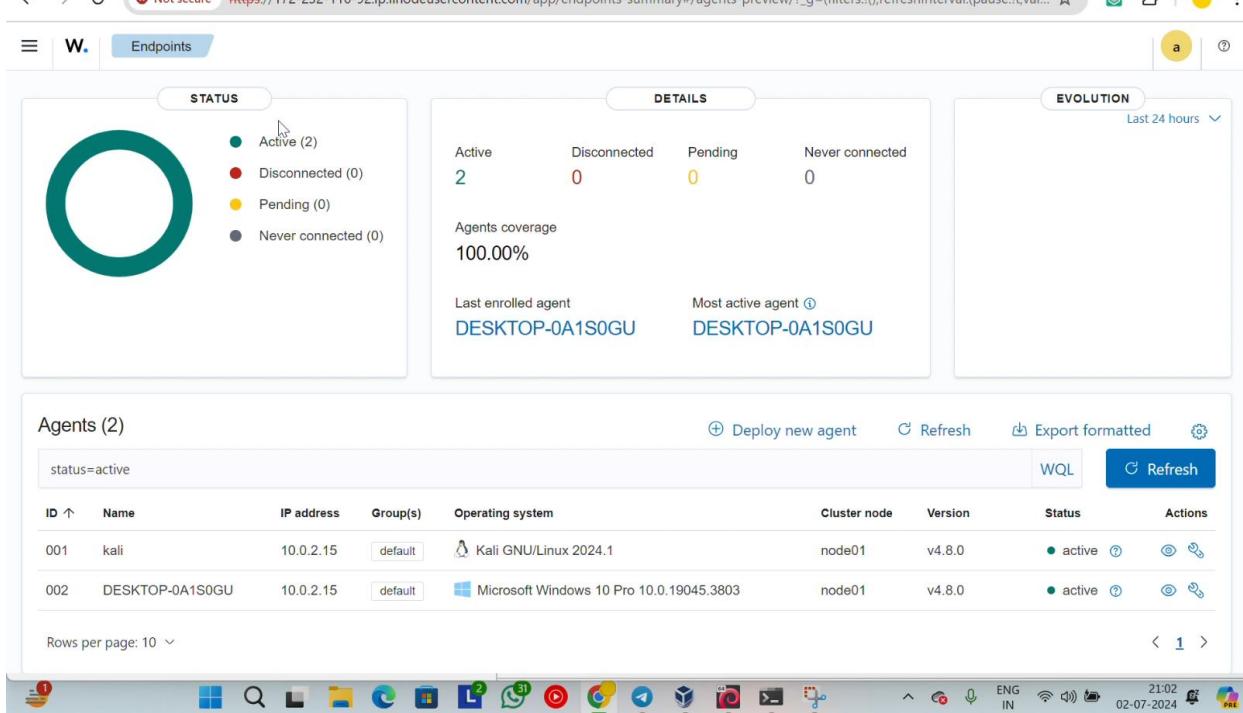
STEP 8: Now You Will See Your Dashboard



STEP 9: click on the Active



The screenshot shows the Wazuh web interface with the URL <https://172-232-110-92.ip.linodeusercontent.com/app/wz-home#/overview>. The top navigation bar includes a 'W.' logo, a search bar, and various icons. The main dashboard features an 'AGENTS SUMMARY' section with a large green circle icon, a red box highlighting 'Active (2)', and a 'Disconnected (0)' status. Below this are sections for 'LAST 24 HOURS ALERTS' (Critical: 0, High: 0, Medium: 378, Low: 207) and 'ENDPOINT SECURITY' (Configuration Assessment, Malware Detection, File Integrity Monitoring). The 'THREAT INTELLIGENCE' section includes Threat Hunting, Vulnerability Detection, MITRE ATT&CK, and VirusTotal.



The screenshot shows the Wazuh web interface with the URL <https://172-232-110-92.ip.linodeusercontent.com/app/endpoints-summary#/agents-preview>. The top navigation bar includes a 'W.' logo, a search bar, and various icons. The main dashboard features a 'STATUS' section with a large green circle icon, a red box highlighting 'Active (2)', and a legend for Disconnected (0), Pending (0), and Never connected (0). Below this are sections for 'DETAILS' (Active: 2, Disconnected: 0, Pending: 0, Never connected: 0) and 'EVOLUTION' (Last 24 hours). The bottom section displays a table titled 'Agents (2)' with columns for ID, Name, IP address, Group(s), Operating system, Cluster node, Version, Status, and Actions. The table lists two agents: 'kali' (IP 10.0.2.15, OS Kali GNU/Linux 2024.1, Cluster node node01, Version v4.8.0, Status active) and 'DESKTOP-0A1S0GU' (IP 10.0.2.15, OS Microsoft Windows 10 Pro 10.0.19045.3803, Cluster node node01, Version v4.8.0, Status active).

2. Malware detection **Permalink to this headline**

Malware, short for malicious software, refers to any software specifically designed to harm or exploit computer systems, networks, or users. It is created with the intention of gaining unauthorized access, causing damage, stealing sensitive information, or performing other malicious activities on a target system. There are various types of malware, each with specific functions and infection methods. Some common types of malware include viruses, worms, ransomware, botnets, spyware, trojans, and rootkits.

Malware detection is crucial for safeguarding computer systems and networks from cyber threats. It helps identify and mitigate malicious software that can cause a data breach, system compromise, and financial loss.

Wazuh for malware detection **Permalink to this headline**

Traditional methods, which rely solely on signature-based detections, have limitations and fail to capture new threats. Signature-based approaches struggle with detecting zero-day attacks, polymorphic malware, and other evasion techniques employed by threat actors. As a result, organizations are at risk of undetected breaches and data exfiltration. Wazuh empowers organizations to detect and respond to sophisticated and evasive threats effectively. Wazuh encompasses different modules that identify malware properties, activities, network connections, and more.

Detecting malicious activities with threat detection rules **Permalink to this headline**

Wazuh has threat detection rules that enable behavior-based malware detection. Instead of relying solely on predefined signatures, Wazuh focuses on monitoring and analyzing abnormal behavior exhibited by malware. This allows Wazuh to detect known and previously unknown threats. This way, Wazuh provides a proactive and adaptable defense

against cyber threats. Wazuh has out-of-the-box rulesets that are specifically designed to trigger alerts for recognized malware patterns, providing a quick response to potential security incidents. For example, the image below shows an alert with rule ID [92213](#) triggered when an executable is dropped in a folder commonly used by malware. This alert prompts security teams to begin the investigation and remediation process.

The screenshot shows the Wazuh Threat Hunting interface. At the top, there's a sidebar with various log fields like agent.id, agent.ip, etc. The main area has tabs for 'Time' and 'rule.description'. Under 'rule.description', it says 'May 3, 2024 @ 15:01:00,082 Executable file dropped in folder commonly used by malware'. Below this is an 'Expanded document' section with 'Table' and 'JSON' tabs. The JSON tab displays the following data:

```

{
  "_index": "wazuh-alerts-4.x-2024.05.03",
  "agent.id": "001",
  "agent.ip": "192.168.0.154",
  "agent.name": "Windows11",
  "data.win.eventdata.creationUtcTime": "2024-05-03 15:00:50.257",
  "data.win.eventdata.image": "C:\Program Files (x86)\Microsoft\EdgeWebView\Application\114.0.1823.82\msedgewebview2.exe",
  "data.win.eventdata.processGuid": "(45cd4aff-0fa2-64c9-3601-000000006000)",
  "data.win.eventdata.processId": "1668",
  "data.win.eventdata.ruleName": "technique_id=T1047,technique_name=File System Permissions Weakness",
  "data.win.eventdata.targetFilename": "C:\Users\Thecotliking\AppData\Local\Temp\chrome_ComponentUnpacker_Begin0szipping1668_1275834351\manifest.json"
}

```

Wazuh allows users to create [custom rules](#) for more flexibility in detection, empowering them to focus on relevant activities, and optimizing malware detection. Wazuh decodes and organizes logs from monitored endpoints into fields, which can then be utilized to create custom rules for alerting when malicious activity is detected.

Wazuh rules use multiple fields that denote indicators of compromise (IOCs) to reduce false positives and detect known malware based on specific behaviors. These rules can connect related malware activities, such as intrusion, privilege escalation, lateral movement, obfuscation, and exfiltration for comprehensive detection.

Below is an example of some Wazuh custom rules created to alert on malicious activities of the LimeRAT malware:

```

<group name="lime_rat,sysmon,">

    <!-- Rogue create netflix.exe creation -->
    <rule id="100024" level="12">
        <if_sid>61613</if_sid>
        <field name="win.eventdata.image" type="pcre2">\.exe</field>
        <field name="win.eventdata.targetFilename" type="pcre2">(?i)[c-
z]:\\Users\\.+\\AppData\\Roaming\\checker netflix\\.exe</field>
        <description>Potential LimeRAT activity detected: checker netflix.exe created at
$(win.eventdata.targetFilename) by $(win.eventdata.image).</description>
        <mitre>
            <id>T1036</id>
        </mitre>
    </rule>

    <!-- Registry key creation for persistence -->
    <rule id="100025" level="12">
        <if_group>sysmon</if_group>
        <field name="win.eventdata.details" type="pcre2">(?i)[c-
z]:\\Users\\.+\\AppData\\Roaming\\checker netflix\\.exe</field>
        <field name="win.eventdata.targetObject" type="pcre2">HKU\\.+\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\checker
netflix\\.exe</field>
        <field name="win.eventdata.eventType" type="pcre2" >^SetValue$</field>
        <description>Potential LimeRAT activity detected: $(win.eventdata.details) added itself to the
Registry as a startup program $(win.eventdata.targetObject) to establish
persistence.</description>
        <mitre>
            <id>T1547.001</id>
        </mitre>
    </rule>

    <!-- Network activity detection -->
    <rule id="100026" level="12">
        <if_sid>61605</if_sid>

```

```

<field name="win.eventdata.image" type="pcre2">>(?i)[c-
z]::\\Users\\\\.+\\\\AppData\\\\Roaming\\\\checker netflix\\.exe</field>
<description>Potential LimeRAT activity detected: Suspicious DNS query made by
$(win.eventdata.image).</description>
<mitre>
<id>T1572</id>
</mitre>
</rule>

<!-- LimeRAT service creation -->
<rule id="100028" level="12">
<if_sid>61614</if_sid>
<field name="win.eventdata.targetObject" type="pcre2"
>HKLM\\\\System\\\\CurrentControlSet\\\\Services\\\\disk</field>
<field name="win.eventdata.eventType" type="pcre2" >^CreateKey$</field>
<description>Potential LimeRAT activity detected: LimeRAT service
$(win.eventdata.targetObject) has been created on $(win.system.computer).</description>
<mitre>
<id>T1543.003</id>
</mitre>
</rule>

</group>

```

These rules create alerts that are visible in the **Threat Hunting** module on the Wazuh dashboard.

The screenshot shows a table titled "Security Alerts" within the "Threat Hunting" tab of the Wazuh interface. The table has columns for Time, Agent, Agent name, Technique(s), Tactic(s), Description, Level, and Rule ID. There are six rows of data, each detailing a specific alert related to LimeRAT activity on agent 001 (Windows11) on May 4, 2024.

Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
May 4, 2024 @ 21:31:34,468	001	Windows11			Potential LimeRAT activity detected: Suspicious DNS query made by C:\Users\...\User\AppData\Roaming\checker.netflix.exe	12	100026
May 4, 2024 @ 21:31:33,462	001	Windows11			Potential LimeRAT activity detected: LimeRAT service HKLM\System\CurrentControlSet\Services\disk has been created on WinDev2210Eval	12	100028
May 4, 2024 @ 21:31:33,453	001	Windows11			Potential LimeRAT activity detected: DLL C:\Windows\Microsoft.NET\Framework\v4.0.30319\nscree.dll is injected on WinDev2210Eval.	12	100027
May 4, 2024 @ 21:31:32,385	001	Windows11			Potential LimeRAT activity detected: C:\Users\...\User\AppData\Roaming\checker.netflix.exe added itself to the Registry HKU\S-1-5-21-1217721-146985683-4281812179-1000\Software\Microsoft\Windows\CurrentVersion\Run\checker.netflix.exe as a startup program to establish persistence.	12	100025
May 4, 2024 @ 21:31:31,158	001	Windows11			Potential LimeRAT activity detected: checker.netflix.exe created at C:\Users\...\User\Downloads\limerat.exe.	12	100024

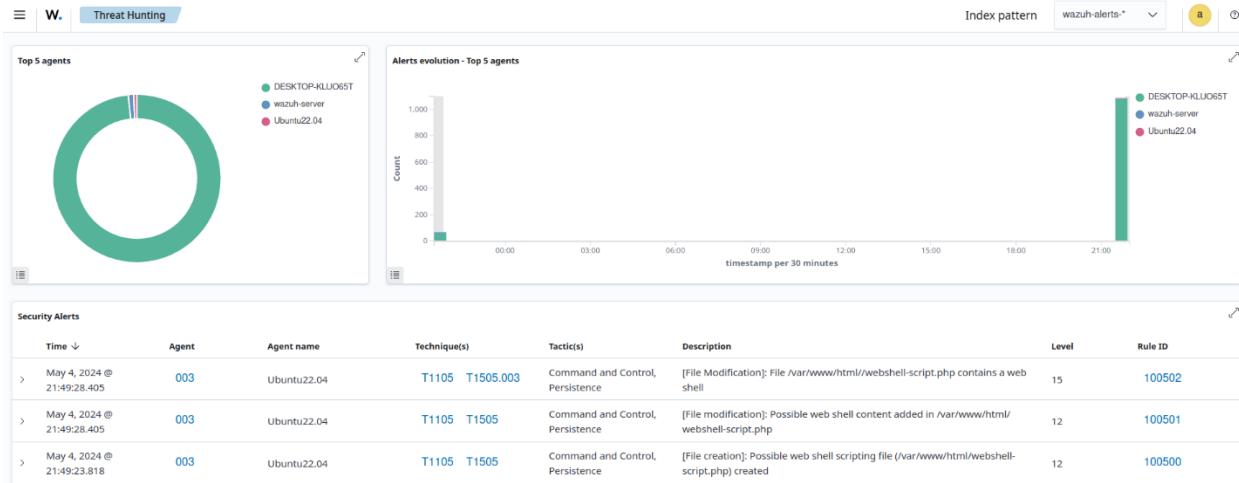
Refer to the blog post on [LimeRAT detection and response with Wazuh](#) for the full configuration.

Wazuh identifies behavior indicative of malware, it generates real-time alerts and notifications, enabling security teams to respond swiftly and mitigate potential risks before they escalate.

Leveraging file integrity monitoring for detecting malware activity[Permalink to this headline](#)

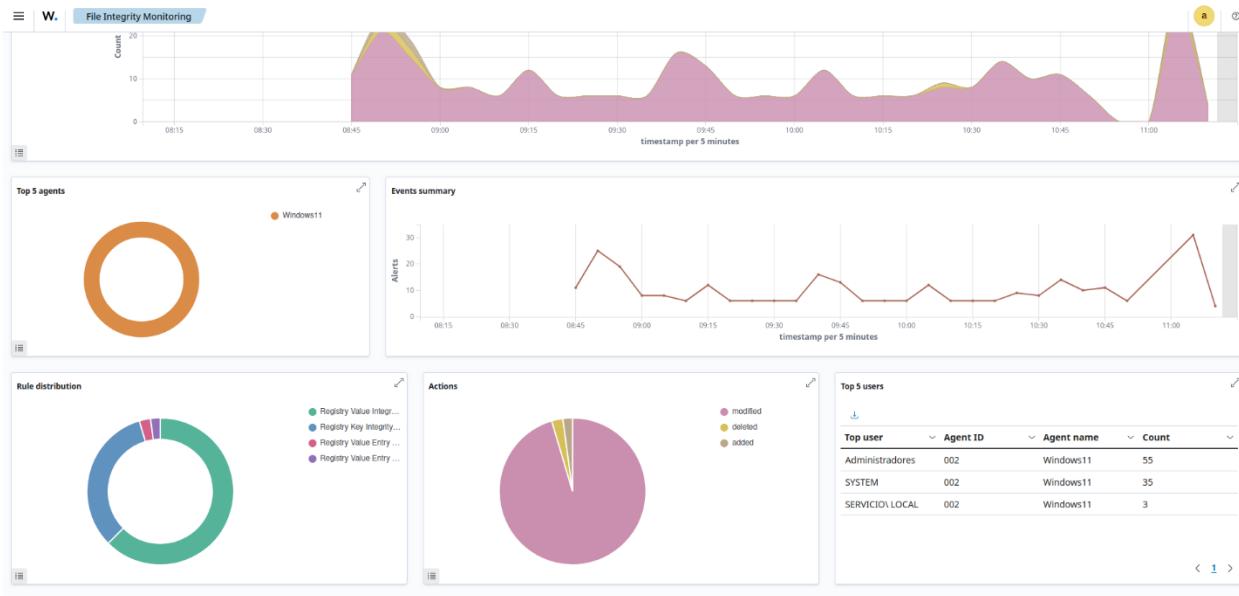
File Integrity Monitoring (FIM) is a valuable component in malware detection. Wazuh provides [FIM capabilities](#) to monitor and detect changes to files and directories on monitored endpoints. These changes include creation, modification, or deletion. While FIM provides essential insights, combining it with other capabilities and integrations further enhances its effectiveness for malware detection. Wazuh allows security teams to create custom rules based on FIM events, enabling targeted malware detection. These customizable rules correlate FIM events with specific indicators of compromises such as suspicious file extensions, code snippets, or known malware signatures.

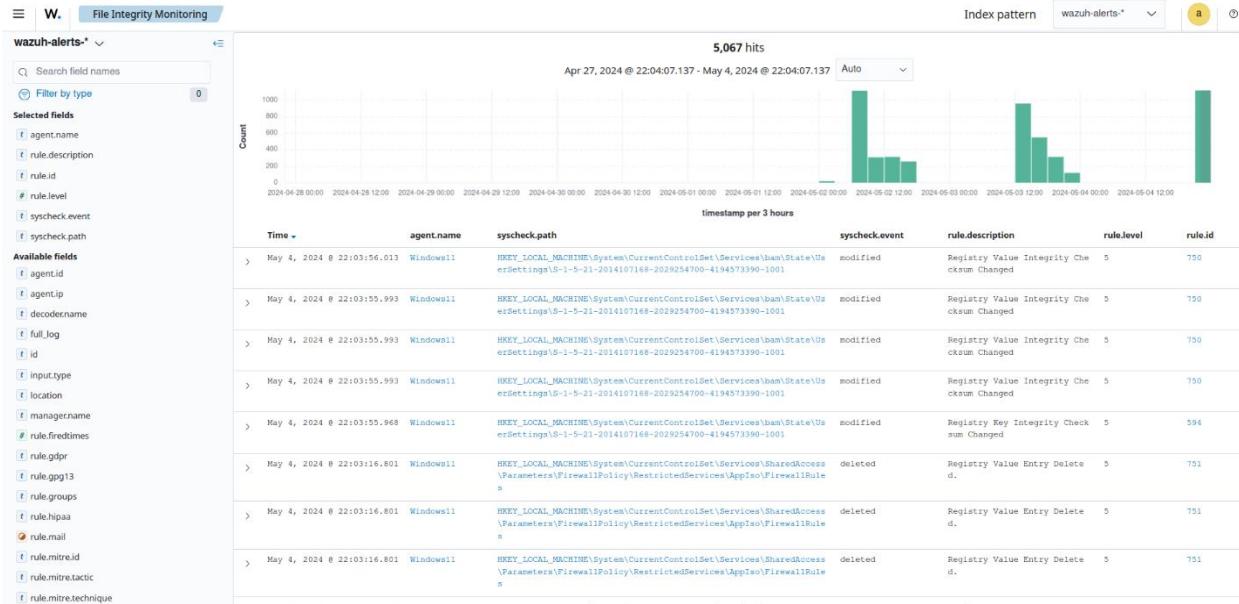
The image below shows an alert when a web shell creates or modifies a file on a web server.



Malware frequently targets Windows Registry to achieve malicious objectives, such as establishing persistence and performing other malicious actions. The Wazuh File Integrity Monitoring (FIM) module includes [Windows Registry monitoring](#) that monitors commonly targeted registry paths to detect modifications. When changes occur, the FIM module triggers real-time alerts, empowering security teams to swiftly identify and respond to suspicious registry key manipulation.

The images below display the Wazuh FIM module dashboard and events of Windows Registry modifications.





Unveiling stealthy threats with rootkit detection[Permalink to this headline](#)

Rootkits are malicious software designed to conceal the presence of malware on an endpoint by manipulating operating system functions such as altering system calls or modifying kernel data structures. Wazuh has a [Rootcheck module](#) that periodically scans the monitored endpoint to detect rootkits both at the kernel and the user space level. The rootcheck identifies and alerts potential rootkit activity. By analyzing system behavior and comparing it to known rootkit patterns, Wazuh promptly detects rootkit-related patterns and raises alerts for further investigation.

Below, we show an example of an alert generated by the Wazuh Rootcheck module when it detects an anomaly in the filesystem:

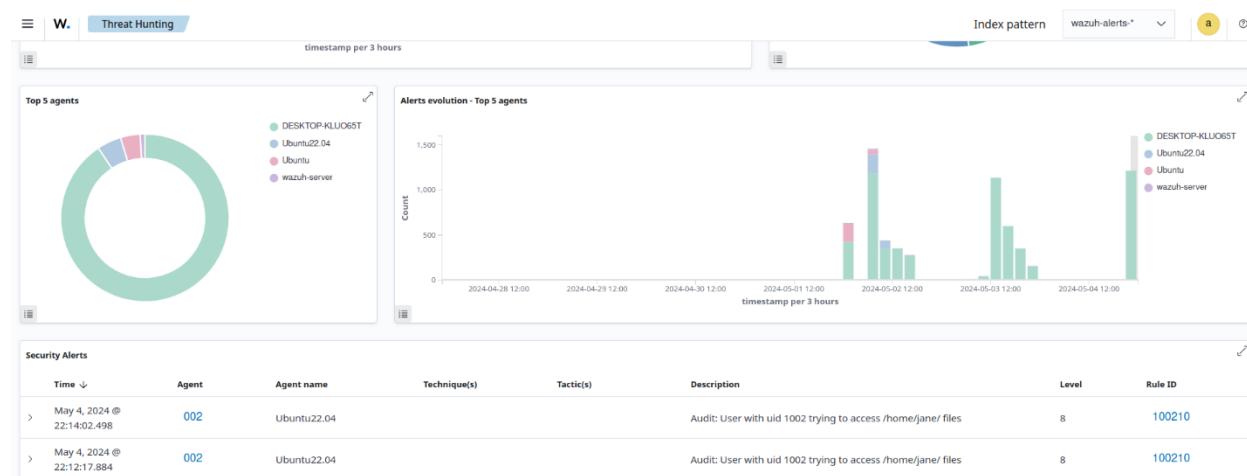
```
** Alert 1668497750.1838326: - ossec,rootcheck,pci_dss_10.6.1,gdpr_IV_35.7.d,
2022 Nov 15 09:35:50 (Ubuntu) any->rootcheck
Rule: 510 (level 7) -> 'Host-based anomaly detection event (rootcheck).'
Rootkit 't0rn' detected by the presence of file '/usr/bin/.t0rn'.
title: Rootkit 't0rn' detected by the presence of file '/usr/bin/.t0rn'.
```

While Wazuh continues to enhance its [rootkit behavior detection capabilities](#), the [Command monitoring module](#) can also be configured to monitor command-line activities across endpoints, enabling the detection of malicious commands and malware activities. This module provides organizations with a comprehensive approach to uncovering hidden threats and safeguarding their systems effectively.

Monitoring system calls for malware and anomaly detection [Permalink to this headline](#)

Wazuh [monitors system calls](#) on Linux endpoints to bolster malware detection and aid in anomaly detection. Wazuh utilizes the Linux Audit system to monitor system calls.

System call monitoring in combination with Wazuh File Integrity Monitoring (FIM) and threat intelligence integration enhances malware detection. It captures security-relevant events like file access, command execution, and privilege escalation, providing real-time insights into potential security incidents. This comprehensive approach strengthens organizations' cybersecurity resilience. In the image below, you can visualize the alerts for privilege abuse on the Wazuh dashboard for Ubuntu Linux 22.04.



Wazuh empowers security teams to leverage the audit rules provided by Audited. Creating custom rules based on system call events enhances malware detection efforts and strengthens overall cybersecurity resilience.

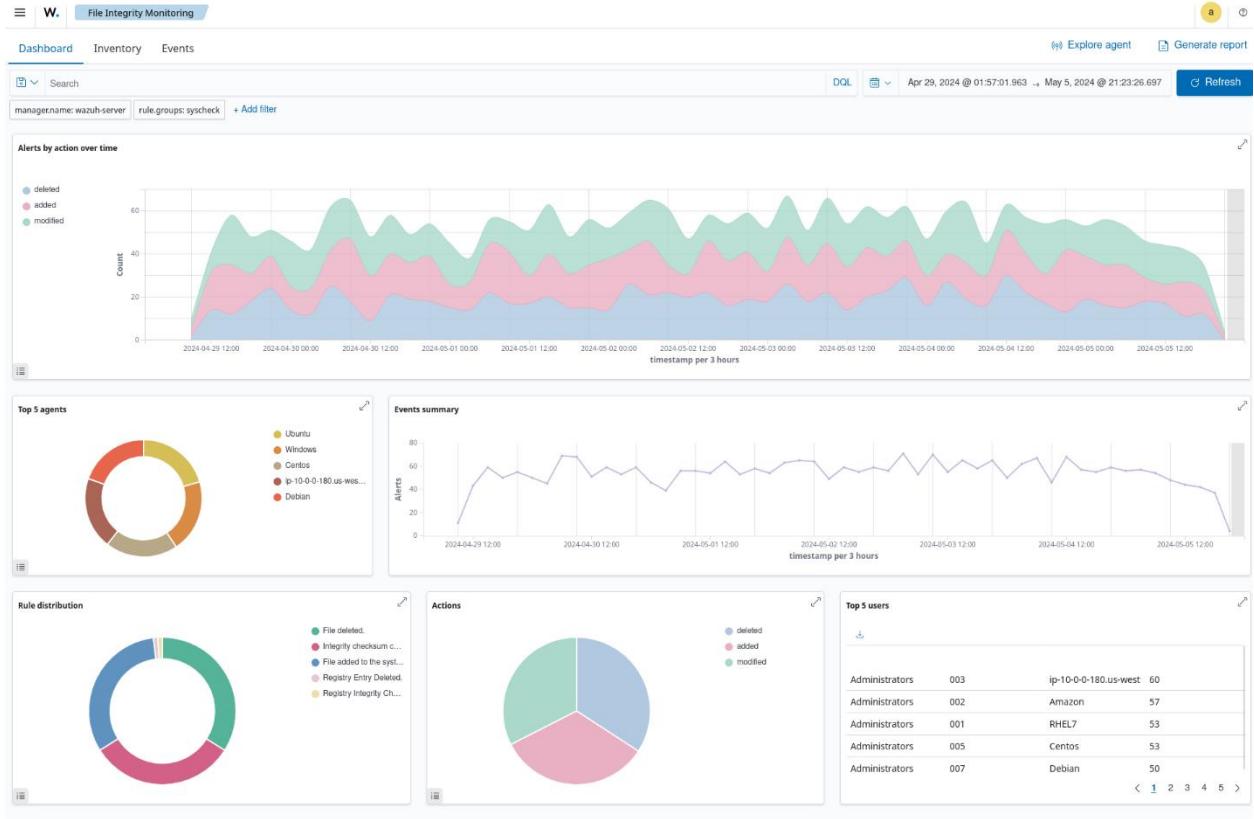
File Integrity Monitoring

You can find a dedicated **File Integrity Monitoring** module in the Wazuh dashboard where all file integrity events triggered from monitored endpoints are reported. This increases visibility as it provides valuable information on the status of monitored directories and their potential impact on the security posture. The [Wazuh FIM dashboard](#) has three different sections to view FIM analysis results; **Inventory, Dashboard, and Events**.

1. The **Inventory** section displays a list of all files that the FIM module has indexed. Each file has entry information including the filename, last modification date, user, user ID, group, and file size.

File ↑	Last Modified	User	User ID	Group	Group ID	Size
/bin	Apr 15, 2024 @ 23:02:50.000	root	0	root	0	7
/etc/.pwd.lock	Apr 15, 2024 @ 23:02:56.000	root	0	root	0	0
/etc/adduser.conf	Apr 15, 2024 @ 23:02:57.000	root	0	root	0	3028
/etc/alternatives/README	Apr 6, 2022 @ 06:40:25.000	root	0	root	0	100
/etc/alternatives/awk	Apr 15, 2024 @ 23:05:39.000	root	0	root	0	13
/etc/alternatives/nawk	Apr 15, 2024 @ 23:05:39.000	root	0	root	0	13
/etc/alternatives/pager	Apr 15, 2024 @ 23:06:02.000	root	0	root	0	9
/etc/alternatives/srm	Apr 15, 2024 @ 23:05:57.000	root	0	root	0	17
/etc/alternatives/which	Apr 15, 2024 @ 23:03:16.000	root	0	root	0	26
/etc/apt/apt.conf.d/01-vendor-ubuntu	Apr 8, 2022 @ 07:22:23.000	root	0	root	0	92
/etc/apt/apt.conf.d/01autoremove	Apr 8, 2022 @ 07:22:23.000	root	0	root	0	630
/etc/apt/apt.conf.d/70debconf	Feb 20, 2022 @ 11:42:49.000	root	0	root	0	182
/etc/apt/apt.conf.d/docker-autoremove-suggests	Apr 15, 2024 @ 23:06:21.000	root	0	root	0	44
/etc/apt/apt.conf.d/docker-clean	Apr 15, 2024 @ 23:06:21.000	root	0	root	0	318
/etc/apt/apt.conf.d/docker-disable-periodic-update	Apr 15, 2024 @ 23:06:21.000	root	0	root	0	27

2. The **Dashboard** section shows an overview of the events triggered by the FIM module for all monitored endpoints. You can also streamline it to show the events for a selected monitored endpoint.



3. The **Events** section shows the alerts triggered by the FIM module. It displays details such as the agent name, the file path of the monitored file, the type of FIM event, a description of the alert, and the rule level of each alert.

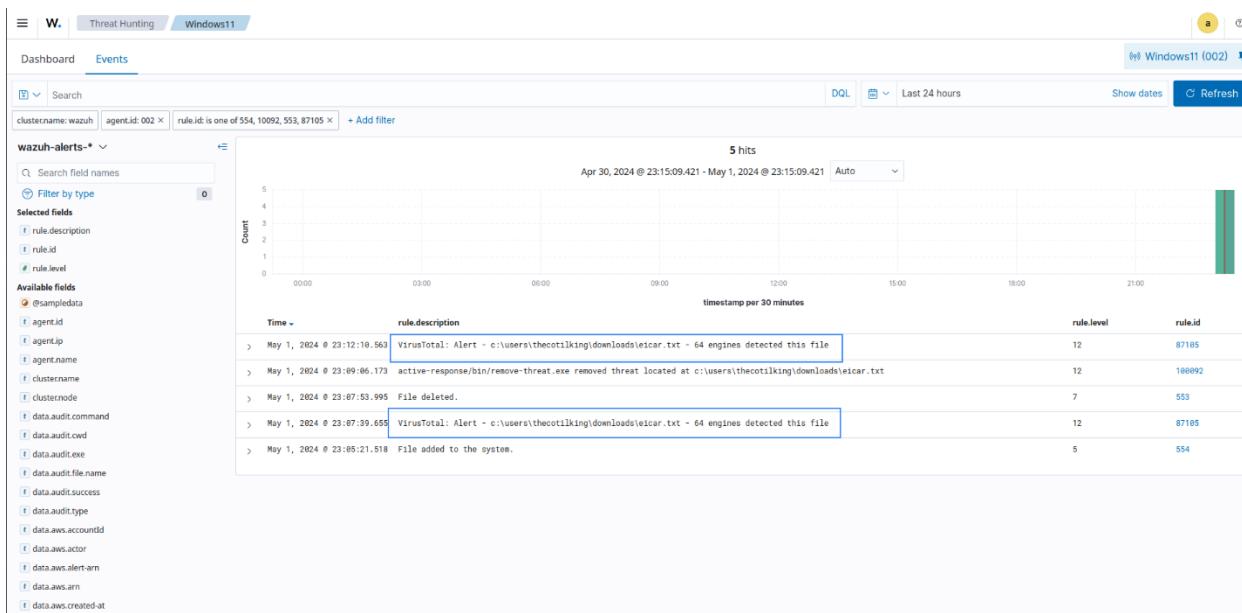
The screenshot shows the Wazuh FIM Events table with the following columns:

Time	agent.name	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
May 5, 2024 @ 21:12:27.423	Debian	/run/utmp	added	File added to the system.	5	554
May 5, 2024 @ 21:02:15.016	Ubuntu	/run/utmp	modified	Integrity checksum changed.	7	550
May 5, 2024 @ 21:01:17.372	Windows	[x32] HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfacer\{54b31d7e-3fbf-4bde-9ab2-106a939cd78c}	modified	Integrity checksum changed.	7	550
May 5, 2024 @ 21:01:16.712	Amazon	/var/osquery/osquery.db/CURRENT	added	File added to the system.	5	554
May 5, 2024 @ 20:58:56.488	Debian	/var/wazuh/queue/fim/db/fim-db	modified	Integrity checksum changed.	7	550
May 5, 2024 @ 20:56:49.917	Debian	/etc/elasticsearch/users	deleted	File deleted.	7	553
May 5, 2024 @ 20:49:27.462	RHEL7	/var/log/lastlog	deleted	File deleted.	7	553
May 5, 2024 @ 20:31:59.443	Debian	/var/wazuh/queue/fim/db/fim-db-journal	added	File added to the system.	5	554
May 5, 2024 @ 20:28:45.580	ip-10-0-0-180.us-west-2.com	/etc/filebeat/fields.yml	deleted	File deleted.	7	553

Below are common use cases the Wazuh FIM module would assist you in monitoring within your environment.

[Enhancing malware detection with threat intelligence integration](#) [Permalink to this headline](#)

Users can boost their malware detection capabilities by [integrating with threat intelligence](#) sources. These intelligence feeds enrich the Wazuh knowledge base with additional up-to-date information on known malicious IP addresses, domains, URLs, and other indicators of compromise. Examples of threat intelligence sources Wazuh can integrate with include VirusTotal, MISP, and more.

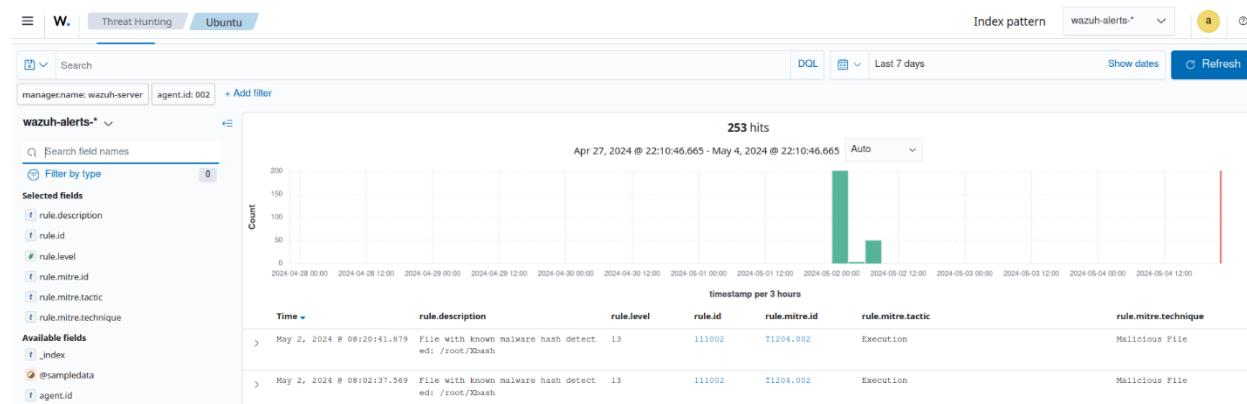


Wazuh proactively identifies malicious files by comparing the identified IOCs with the information stored in the [CDB lists](#) (constant databases). These lists can store known malware indicators of compromise (IOCs) including file hashes, IP addresses, and domain names.

You can customize entries in either `key:value` or `key:` format for tailored detection, an example of such is seen below. A CBD list containing known MD5 malware hashes of the Mirai and Xbash malware is used for detection:

```
e0ec2cd43f71c80d42cd7b0f17802c73:mirai  
55142f1d393c5ba7405239f232a6c059:Xbash
```

Upon detection, these alerts are observed within the **Threat Hunting** module of the Wazuh dashboard, as seen below.



Refer to the [Use case: Detecting malware using file hashes in a CDB list](#) for full configurations.

Monitoring file integrity[Permalink to this headline](#)

Modifications to configuration files and file attributes are frequent occurrences within endpoints in an IT infrastructure. However, if not validated, there may be unauthorized and inadvertent changes that could affect the behavior of the endpoints or the applications running in them. The Wazuh FIM module runs periodic scans on specific files and directories to detect file changes in real time. It scans the designated files to create a baseline of the current state. It checks for file modifications by comparing checksums and attribute values to the baseline, generating alerts if discrepancies are found.

The Wazuh FIM module supports various configuration options that enable effective monitoring of assets:

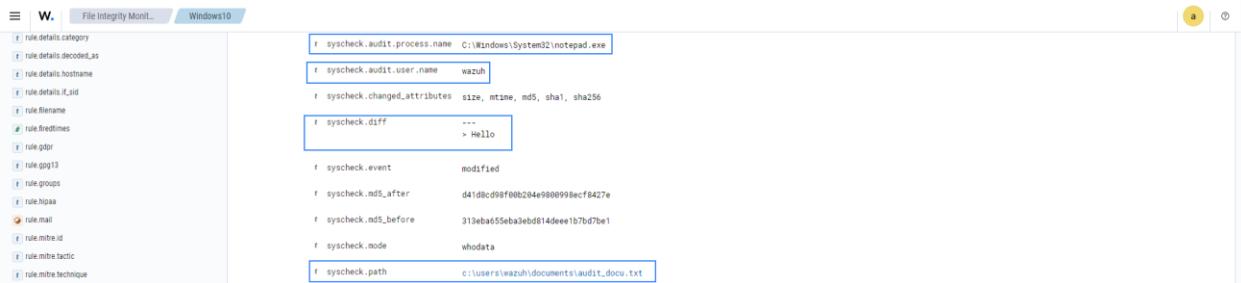
- **Real-time monitoring:** The FIM module provides a `realtime` attribute that enables continuous monitoring of specified directories. This feature is particularly useful for monitoring critical directories and tracking changes immediately after they occur. Wazuh allows you to specify the directories or files in the monitored endpoints that would be reported in real-time if file changes occur.
- **Scheduled monitoring:** The `frequency` option in the Wazuh FIM module allows users to customize the scheduling of each FIM scan performed in your monitored endpoints. The default scan interval for the FIM module is 12 hours (43200 seconds) and can be customized on each endpoint. Alternatively, scans can be scheduled using the `scan_time` and the `scan_day` options. These options help users to set up FIM scans outside business hours or during holidays.
- **Who-data monitoring:** Wazuh captures advanced insights into file changes using the who-data functionality. This functionality uses audit tools like the Linux Audit subsystem and Microsoft Windows SACL to determine important information about the detected file changes. The who-data monitoring functionality allows the FIM module to obtain information on when the change event occurred, who or what made the change, and what content was changed. This is useful in maintaining accountability and validating if changes made to monitored files or directories were authorized and performed using approved processes.

Below is an example of an alert generated when a monitored file is changed on a Windows endpoint.

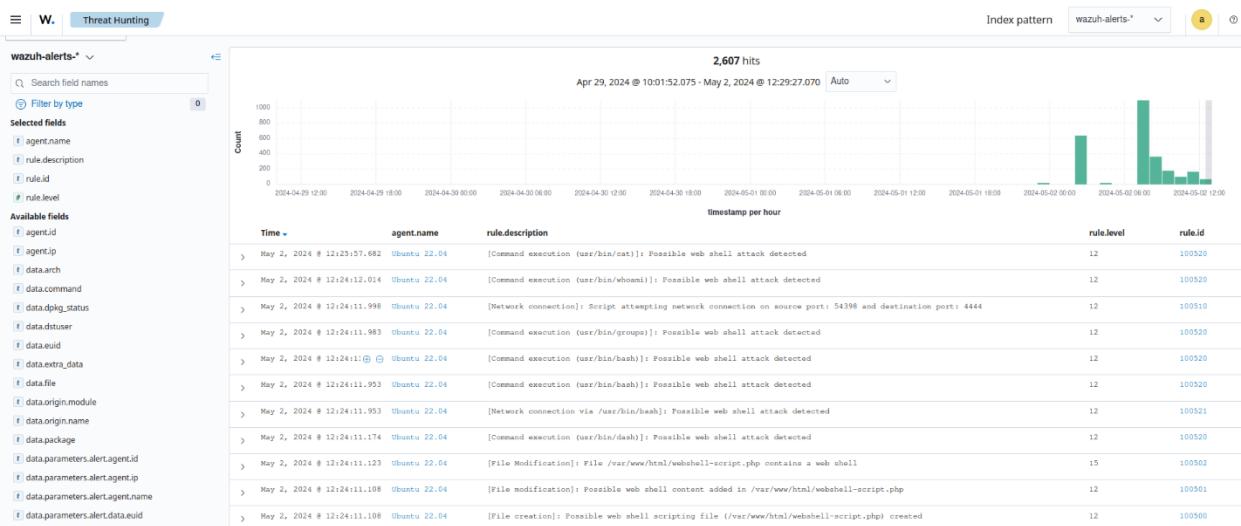


Available fields		Time	syscheck.path	timestamp per 3 hours	rule.description	rule.level	rule.id
syscheck.path	@sampledata	> May 5, 2024 @ 20:22:34.311	c:\users\wazuh\documents\audit_docu.txt	-	Integrity checksum changed.	7	550
agent.id	agent.ip	> May 5, 2024 @ 17:30:40.565	c:\users\wazuh\documents\audit_docu.txt	added	File added to the system.	5	554

In alert fields, the *who-data* metadata shows that the user `wazuh` added the word `Hello` to the `audit_docu.txt` file using the `Notepad.exe` process.



- Reporting changes in file values:** The FIM module provides a `report_changes` attribute that records and reports the exact content changed in a text file to the Wazuh server. The attribute enables the Wazuh agent to make copies of monitored files to a private location on each endpoint for further review. This monitoring option is helpful when users want to initiate specific responses when file changes in monitored directories match the behavior of known malicious activities. For example, the alert below indicates when Wazuh detects the creation of a web shell scripting file `webshell-script.php` in a monitored directory.



- Recording file attributes:** Users can configure the FIM module to record specific attributes of a monitored file. Wazuh supports various file attributes that users can

use to specify the file metadata that the FIM module will record or ignore. For example, this monitoring option would be useful when users want to record only the SHA-256 hash of a configuration file, excluding other hash types.

Detecting and responding to malware [Permalink to this headline](#)

The Wazuh FIM module integrates with other Wazuh capabilities and third-party threat intelligence solutions to create a comprehensive security monitoring environment. This is imperative to enhance malware detection and response capabilities, ensuring robust defense against cyber threats.

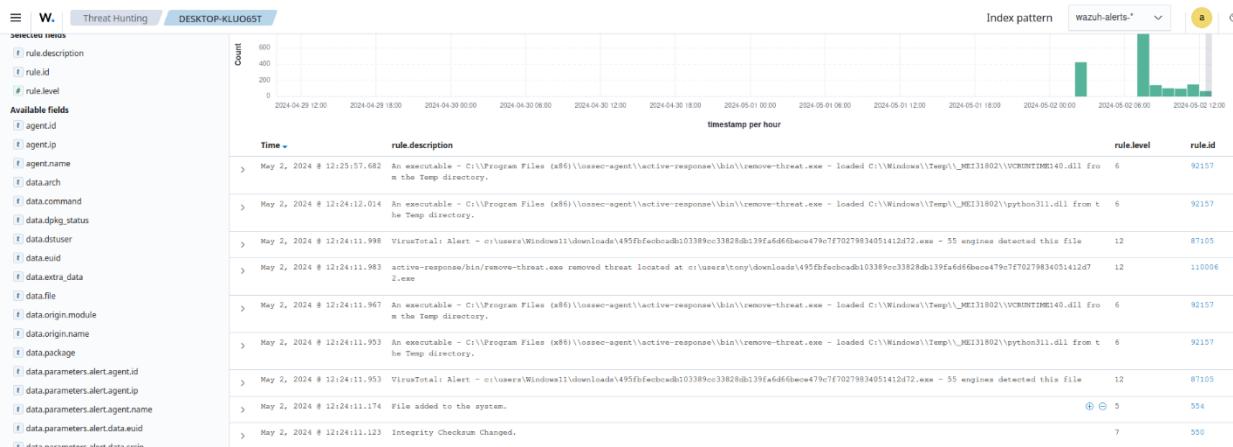
The Wazuh FIM module supports various integrations, including but not limited to:

- **File integrity monitoring and YARA:** By combining the Wazuh FIM module and the YARA tool, it is possible to detect malware when suspicious file additions or modifications are identified. The YARA rule files contain samples of malware indicators that are downloaded to the monitored endpoints. When the FIM module detects a change in the monitored file or directory, it executes a YARA scan using a script to determine if it is malware. If the YARA rule finds a match with a file, it will send the scan results to the Wazuh server for decoding and alerting. This would be reported according to the custom rule and decoder configurations configured on the Wazuh server. Check this documentation for more information on [how to integrate the Wazuh FIM module with YARA](#).
- **File integrity monitoring and VirusTotal:** The Wazuh [Integrator module](#) connects to external APIs and alerting tools such as VirusTotal. The [VirusTotal integration](#) uses the VirusTotal API to detect malicious file hashes within the files and directories monitored by the FIM module. Once enabled, when FIM generates alerts, Wazuh initiates the VirusTotal integration to extract the hash value associated

with the flagged file from the alert. The VirusTotal API is then used to compare these hashes against its scanning engines for potentially malicious content.

- **File integrity monitoring and active response:** The [Wazuh active response](#) module automatically responds to threats identified in a timely manner. This combination enables the FIM module to not only detect but also respond to malicious activities. You can configure active response scripts to execute when the FIM module detects file changes in your monitored environment. Additionally, it also generates alerts for the response performed. This reduces the Mean Time To Respond (MTTR) as malicious changes detected are remediated in a timely manner.

In the image below Wazuh triggers when a file is added to the monitored endpoint. The VirusTotal API scans the file and identifies it as malicious content on 55 engines. Then the Wazuh active response module acts immediately to remove the threat from the monitored endpoint.



- **File integrity monitoring and CDB list:** Wazuh FIM module also detects malicious files by checking the presence of known malware signatures when combined with [CDB lists \(constant database\)](#). CDB lists are used to store known malware indicators of Compromise (IOCs) such as file hashes, IP addresses, and domain names. When CDB lists are created, Wazuh checks if field values from FIM alerts

such as file hash match the keys stored in the CDB lists. If matched, it generates an alert and response based on how you configure your custom rule.



Monitoring Windows Registry [Permalink to this headline](#)

The Wazuh FIM module periodically scans Windows Registry entries, stores its checksums and attributes in a local database, and alerts when changes in registry values are detected. This would keep users informed about registry modifications resulting from user activities or software installations whether malicious or not.

You can configure the Wazuh open source FIM module to monitor [Windows Registry values](#) using various configuration options. The `report_changes` attribute in the `windows_registry` option provides a granular breakdown of modification detected in the monitored Windows Registry value. You can configure which Windows Registry attributes the module would record or ignore. For example, you can choose to record the `check_shasum` attribute and ignore the `check_md5sum` attribute, if your CDB list only contains SHA1 hashes of malicious files.

The image below shows the event of a modified Windows registry value in a monitored endpoint.



The alert when expanded shows the modified field.

The screenshot shows a Wazuh File Integrity Monitor interface for a Windows-10 system. A specific event is highlighted in yellow, indicating a change in file attributes. The event details are as follows:

Field	Value
syscheck.changed_attributes	size, md5, sha1, sha256
syscheck.diff	size, sha1
syscheck.event	modified
syscheck.md5_after	b271050350de1d00d0b531d91e85e7c
syscheck.md5_before	47f43e7f4d15737d0d1ef3e92bf08ec4
syscheck.mode	scheduled
syscheck.path	HKEY_LOCAL_MACHINE\System\Setup\Custom_key
syscheck.sha1_after	7e08ac8708c6ebe7d820c798620ddc2b2bf016e
syscheck.sha1_before	336c907c1df6030a9f3e0eb2c30e3488748688
syscheck.size_after	92141d8e4eff92a0f5cb0ea304e4db0fe8e99b03e64ac9742a80d43990c
syscheck.size_before	5908ecc4b0ff4681fc90a73b70241ae5285ee0f703b2419cbd364fb9f90b22
syscheck.value_name	F1H
syscheck.value_type	REG_SZ
timestamp	May 2, 2024 @ 12:24:12.014

Threat actors maintain persistence by commonly adding programs for their malicious activities to the *Run* and *RunOnce* keys in the Registry. Additionally, Wazuh detects any suspicious programs added to the startup registry keys. This allows you to take appropriate action to remove them before they cause harm to your system.

The screenshot shows a Wazuh File Integrity Monitor interface for a Windows-10 system. It displays a list of registry key changes over time, specifically focusing on the *RunOnce* and *Run* keys under the *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion* path. The events are as follows:

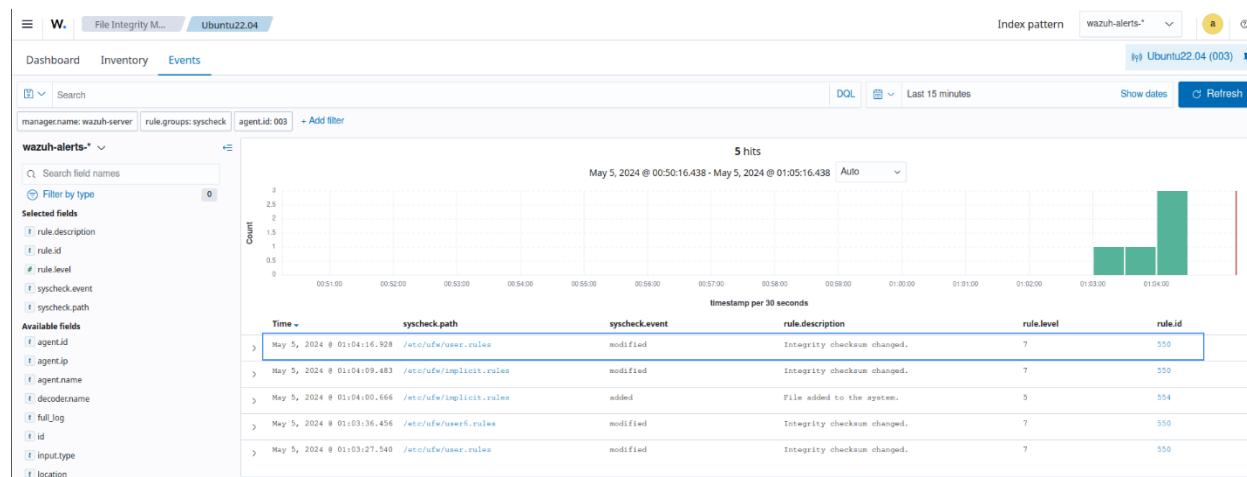
Time	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
May 2, 2024 @ 12:24:11.967	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce	added	Registry Value Entry Added to the System	5	750
May 2, 2024 @ 12:24:11.953	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce	modified	Registry Key Integrity Checksum C changed	5	594
May 2, 2024 @ 12:24:11.078	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run	added	Registry Value Entry Added to the System	5	752
May 2, 2024 @ 12:24:19.321.132	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run	modified	Registry Key Integrity Checksum C changed	5	594

Meeting regulatory compliance[Permalink to this headline](#)

Meeting regulatory compliance requirements is an important consideration for organizations in various industries. File integrity monitoring is a requirement for achieving compliance with regulations such as PCI DSS, SOX, HIPAA, NIST SP 800-53, among others.

You can customize the Wazuh FIM module to monitor specific files and directories where your organization's sensitive and confidential data are stored. Wazuh provides a comprehensive report that outlines the changes made to the files and directories being monitored. This feature is particularly useful for ensuring compliance with various regulatory standards.

For example, organizations can meet the [CM-3 Configuration change control](#) requirement in NIST SP 800-53 standard by using Wazuh. The control requires organizations to protect information at rest and monitor configuration changes in their infrastructure. The image below shows an event generated when the permissions for Uncomplicated Firewall (UFW) rule files are modified on a monitored endpoint.



[Log data analysis](#)[Permalink to this headline](#)

Log data analysis is a crucial process that involves examining and extracting valuable insights from log files created by different systems, applications, or devices. These logs contain records of events that provide useful information for troubleshooting, security analysis and monitoring, and optimizing performance. Log data analysis is an essential practice that contributes to a secure, efficient, and reliable IT ecosystem.

Wazuh collects, analyzes, and stores logs from endpoints, network devices, and applications. The Wazuh agent, running on a monitored endpoint collects and forwards system and application logs to the Wazuh server for analysis. Additionally, you can send log messages to the Wazuh server via syslog or third-party API integrations.

[Log data collection](#)[Permalink to this headline](#)

Wazuh collects logs from a wide range of sources, enabling comprehensive monitoring of various aspects of your IT environment. You can check our documentation on [Log data collection](#) to understand better how Wazuh collects and analyzes logs from monitored endpoints. Some of the common log sources supported by Wazuh include:

- **Operating system logs:** Wazuh collects logs from several operating systems, including [Linux](#), [Windows](#), and [macOS](#).

Wazuh can collect syslog, auditd, application logs, and others from Linux endpoints.

Wazuh collects logs on Windows endpoints using the Windows event channel and Windows event log format. By default, the Wazuh agent monitors the System, Application, and Security Windows event channels on Windows endpoints. The Wazuh agent offers the flexibility to configure and monitor other [Windows event channels](#).

Wazuh utilizes the unified logging system (ULS) to collect logs on macOS endpoints. The macOS ULS centralizes the management and storage of logs across all the system levels.

The image below shows an event collected from the [Microsoft-Windows-Sysmon/Operational](#) event channel on a Windows endpoint.

The screenshot shows a threat hunting search results page for the index pattern "wazuh-alerts-*". A single alert is displayed, triggered on May 5, 2024, at 12:59:02.334. The alert details an executable dropped in the Windows root folder. The expanded document view shows the JSON structure of the event data, including fields like _index, rule.description, rule.level, rule.id, agent.ip, agent.name, data.win.eventdata.authenticationPackageName, data.win.eventdata.creationUtcTime, data.win.eventdata.image, data.win.eventdata.logonProcessName, data.win.eventdata.logonType, data.win.eventdata.param1, data.win.eventdata.param2, data.win.eventdata.param3, data.win.eventdata.param4, data.win.eventdata.procguid, data.win.eventdata.procsessionId, data.win.eventdata.procName, data.win.eventdata.subjectDomainName, data.win.eventdata.subjectLogonId, data.win.eventdata.subjectUserName, data.win.eventdata.subjectUserId, data.win.eventdata.targetDomainName, data.win.eventdata.targetLinkedLogonId, data.win.eventdata.targetLogonId, data.win.eventdata.targetUserName, data.win.eventdata.targetUserId, data.win.eventdata.virtualAccount, data.win.system.channel, data.win.system.computer, data.win.system.eventID, and data.win.system.eventRecordID.

- Syslog events:** Wazuh gathers logs from [syslog-enabled](#) devices, encompassing a wide array of sources including Linux/Unix systems and network devices that do not support agent installation. The image below shows an alert triggered when a new user is created on the Linux endpoint and the log is forwarded to the Wazuh server via rsyslog.

The screenshot shows a threat hunting search results page for the index pattern "wazuh-alerts-*". A single alert is displayed, triggered on May 5, 2024, at 13:09:41.272. The alert details a new user added to the system. The expanded document view shows the JSON structure of the event data, including fields like @timestamp, _id, agent.id, agent.ip, agent.name, data.dsUser (with Stephen highlighted), data.gid, data.home, data.shell, data.uid, and decodername.

- Agentless monitoring:** The Wazuh agentless monitoring module monitors endpoints that don't support agent installation. It requires an SSH connection between the endpoint and the Wazuh server. The Wazuh [agentless monitoring](#) module monitors

files, directories, or configurations and runs commands on the endpoint. The image below is an alert from an agentless device on the Wazuh dashboard.

W. Discover

wazuh-alerts*

Search field names Filter by type

Selected fields

- rule.description
- rule.level
- rule.id
- agentless.host

Available fields

- _index
- agent.ip
- agent.name
- data.wen.eventdata.authentication.hashAlgorithm
- data.wen.eventdata.eventTime
- data.wen.eventdata.eventType
- data.wen.eventdata.eventValue.level
- data.wen.eventdata.eventValue.logs

May 4, 2024

Count

May 4, 2024

Table JSON

↑ _index	wazuh-alerts-4.x-2024.05.05
↑ agent.ephemeral_id	6fed6291-e23d-4a30-ad9b-20f0f172ee7a
↑ agent.hostname	wazuh
↑ agent.id	3e01657-dff0-4c9d-8518-aa556aef110
↑ agent.name	wazuh
↑ agent.type	filebeat
↑ agent.version	7.10.2
↑ agentless.host	192.168.0.137
↑ agentless.script	ssh_integrity_check_linux
↑ agentless.user	agentless
↑ decoder.name	syscheck_integrity_changed
↑ full_log	<pre>File '/var/spool/dri/file1.log' checksum changed. Size changed from '5' to '6'. Old md5sum was: 'b91e090e5fd1a3dc2f60dab8fca65e*'. New md5sum is: '21b8e0d8d180912c7161dd08fb5d*'. Old sha1sum was: 'a2266a9b0da0b452cc0e1284932ac97e*'. New sha1sum is: '0e61d14dc147350ee07e74f70ed6d6a71d*'</pre>
↑ id	1714930301.5300160
↑ input.type	log

View surrounding documents View single document

- **Cloud provider logs:** Wazuh integrates with cloud providers like [AWS](#), [Azure](#), [Google Cloud](#), and [Office 365](#) to collect logs from cloud services such as EC2 instances, S3 buckets, Azure VMs, and more. The image below shows the **CLOUD SECURITY** section in the Wazuh dashboard.

W. Overview

Index pattern wazuh-alerts.* ▾

ENDPOINT SECURITY

-  Configuration Assessment
Scan your assets as part of a configuration assessment audit.
-  Malware Detection
Verify that your systems are configured according to your security policies baseline.

File Integrity Monitoring
Alerts related to file changes, including permissions, content, ownership, and attributes.

THREAT INTELLIGENCE

-  Threat Hunting
Browse through your security alerts, identifying issues and threats in your environment.
-  Vulnerability Detection
Discover what applications in your environment are affected by well-known vulnerabilities.

SECURITY OPERATIONS

-  PCI DSS
Global security standard for entities that process, store, or transmit payment cardholder data.
-  GDPR
General Data Protection Regulation (GDPR) sets guidelines for processing of personal data.

-  HIPAA
Health Insurance Portability and Accountability Act of 1996 (HIPAA) provides data privacy and security provisions for safeguarding medical information.
-  NIST 800-53
National Institute of Standards and Technology Special Publication 800-53 (NIST 800-53) sets guidelines for federal information systems.

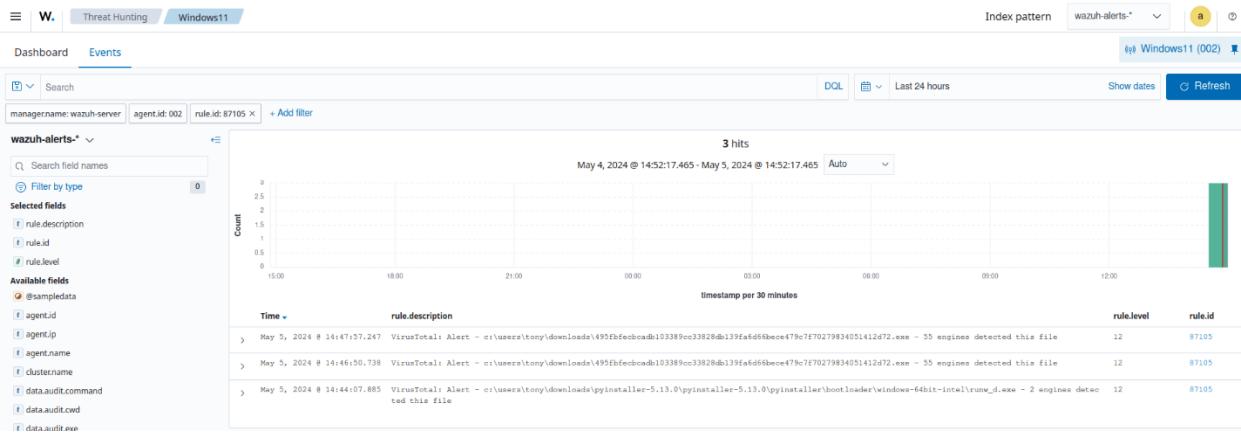
CLOUD SECURITY

-  Docker
Monitor and collect the activity from Docker containers such as creation, running, starting, stopping or pausing events.
-  AWS Amazon Web Services
Security events related to your Amazon AWS services, collected directly via AWS API.

-  Google Cloud
Security events related to your Google Cloud Platform services, collected directly via GCP API.
-  GitHub
Monitoring events from audit logs of your GitHub organizations.

-  Office 365
Security events related to your Office 365 services.

- **Custom logs:** You can configure Wazuh to collect and parse logs from several applications and third-party security tools like [VirusTotal](#), [Windows Defender](#), and [ClamAV](#). The image below shows an alert of a log from VirusTotal processed by the Wazuh server.



Rules and decoders [Permalink to this headline](#)

Wazuh [rules and decoders](#) are core components in log data analysis and threat detection and response. Wazuh provides a powerful platform for log data analysis, allowing organizations to enhance their security posture by promptly detecting and responding to potential security threats.

Wazuh decoders are responsible for parsing and normalizing log data collected from various sources. Decoders are essential for converting the raw log data in several formats into a unified and structured format that Wazuh can process effectively. Wazuh has pre-built decoders for common log formats such as syslog, Windows event channel, macOS ULS, and more. Additionally, Wazuh allows you to define [custom decoders](#) for parsing logs from specific applications or devices with unique log formats. By using decoders, Wazuh can efficiently interpret log data and extract relevant information, such as timestamps, log levels, source IP addresses, user names, and more. As shown below, you can view Wazuh

out-of-the-box and custom decoders on the **Server management > Decoders** of the Wazuh dashboard.

Name	Program name	Order	File	Path
wazuh			0005-wazuh_decoders.xml	ruleset/decoders
agent-buffer		level	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		agent.id, agent.name, status	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		error	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		agent.cur_version	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		agent.new_version	0005-wazuh_decoders.xml	ruleset/decoders
agent-restart		module	0005-wazuh_decoders.xml	ruleset/decoders
fm-state			0005-wazuh_decoders.xml	ruleset/decoders
json			0006-json_decoders.xml	ruleset/decoders
wazuh-api			0007-wazuh-api_decoders.xml	ruleset/decoders

Wazuh ruleset detects security events and anomalies in log data. These rules are written in a specific format and they trigger alerts when certain conditions are met. The rules are defined based on certain criteria like log fields, values, or patterns to match specific log entries that may indicate security threats. Wazuh provides a wide range of pre-built rules covering common security use cases. Additionally, administrators can create [custom rules](#) tailored to their specific environment and security requirements. The **Server management** category of the Wazuh dashboard lets you view the default and custom **Rules**.

ID	Description	Groups	Regulatory compliance	Level	File	Path
1	Generic template for all syslog rules.	syslog		0	0010-rules_config.xml	ruleset/rules
2	Generic template for all firewall rules.	firewall		0	0010-rules_config.xml	ruleset/rules
3	Generic template for all ids rules.	ids		0	0010-rules_config.xml	ruleset/rules
4	Generic template for all web rules.	web-log		0	0010-rules_config.xml	ruleset/rules
5	Generic template for all web proxy rules.	squid		0	0010-rules_config.xml	ruleset/rules
6	Generic template for all windows rules.	windows		0	0010-rules_config.xml	ruleset/rules
7	Generic template for all wazuh rules.	ossec		0	0010-rules_config.xml	ruleset/rules
200	Grouping of wazuh rules.	wazuh		0	0016-wazuh_rules.xml	ruleset/rules
201	Agent event queue rule	agent_flooding, wazuh		0	0016-wazuh_rules.xml	ruleset/rules
202	Agent event queue is level full.	agent_flooding, wazuh	PCI_DSS, GDPR	7	0016-wazuh_rules.xml	ruleset/rules

For example, the rule below includes a `match` field used to define the pattern that the rule looks for. The rule also has a `level` field that specifies the priority of the resulting alert. Additionally, rules enrich events with technique identifiers from the MITRE ATT&CK framework and map them to regulatory compliance controls.

```
<rule id="5715" level="3">
  <if_sid>5700</if_sid>
  <match>^Accepted|authenticated.\$</match>
  <description>sshd: authentication success.</description>
  <mitre>
    <id>T1078</id>
    <id>T1021</id>
  </mitre>

  <group>authentication_success,gdpr_IV_32.2,gpg13_7.1,gpg13_7.2,hipaa_164.312.b,nist_800_
53_AU.14,nist_800_53_AC.7,pci_dss_10.2.5,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>
```

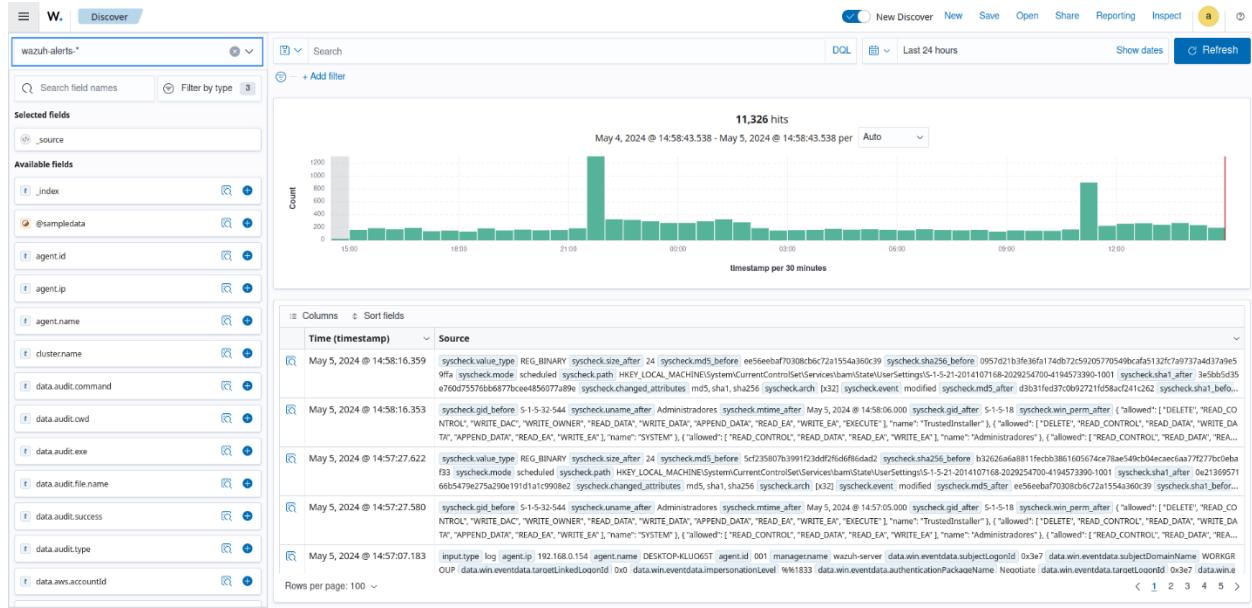
Log data indexing and storage[Permalink to this headline](#)

The [Wazuh indexer](#) is a highly scalable, distributed real-time search and analytics engine. The Wazuh indexer is critical in log analysis as it stores and indexes alerts generated by the Wazuh server. These alerts are stored as JSON documents.

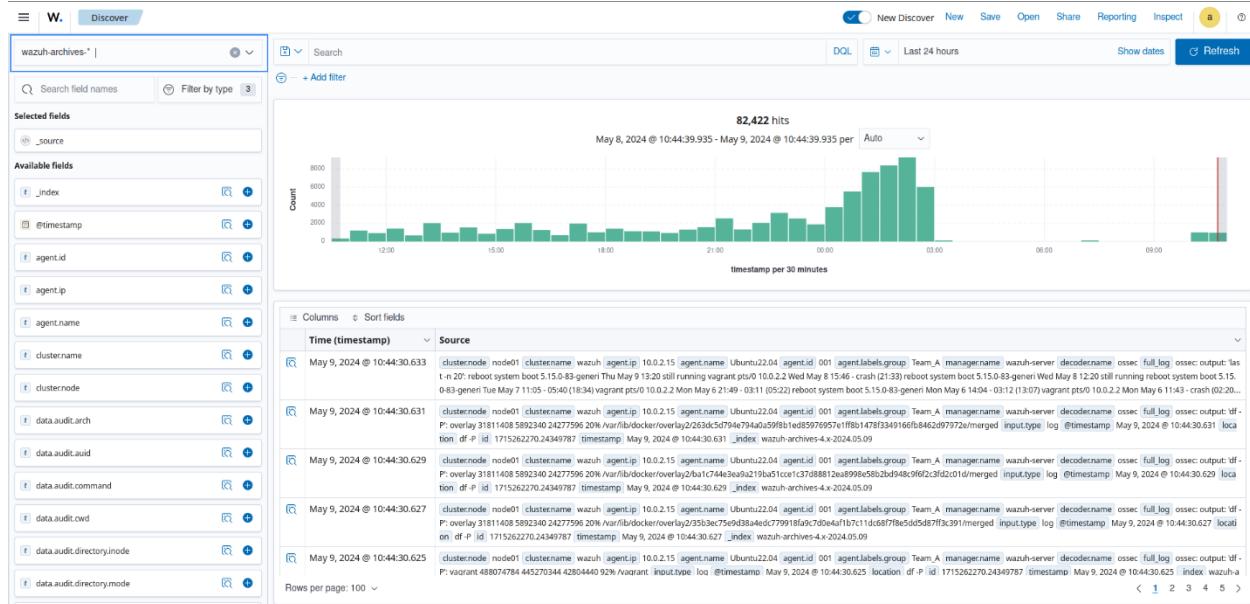
The Wazuh indexer guarantees redundancy by storing the JSON documents across several containers called shards and distributing the shards across multiple nodes. This implementation prevents downtime when hardware failures or cyber-attacks occur and increases query capacity as nodes are added to a cluster.

Wazuh uses four indices to store several event types:

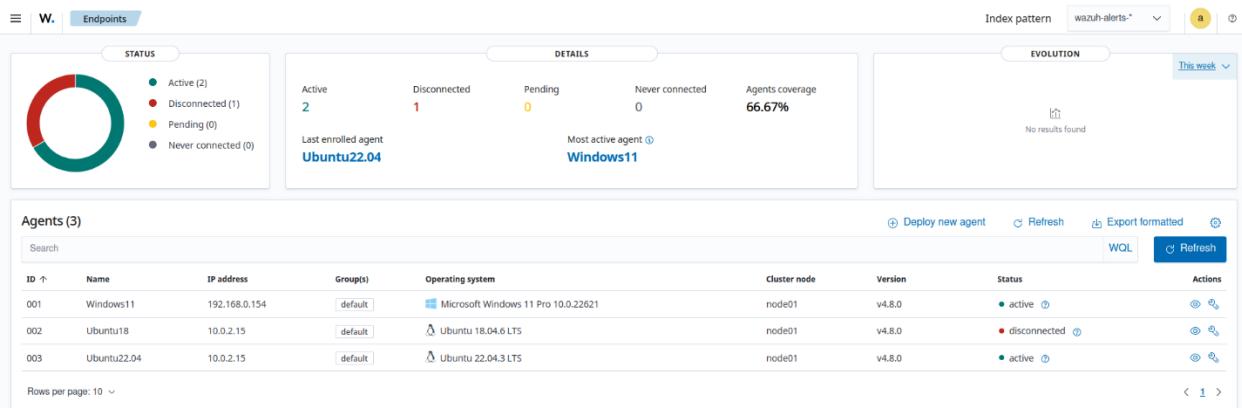
- **wazuh-alerts** stores alerts generated by the Wazuh server when an event triggers a rule with high enough priority. The image below shows alerts in the **Discover** module of the Wazuh dashboard. The index pattern is set to **wazuh-alerts-*** by default.



- **wazuh-archives** index stores all events received from the Wazuh server regardless of whether they trigger an alert. The [Wazuh archives](#) use this index to enable log retention and querying capabilities that offer deeper insight into events happening within monitored endpoints. Wazuh archives are disabled by default because of the huge storage requirements needed to store all the logs. The image below shows archived events in the **Discover** section of Wazuh dashboard with the index pattern set to **wazuh-archives-***.



- **wazuh-monitoring** index stores data about the state of Wazuh agents over a period of time. The state of the agent could be **Active**, **Disconnected**, or **Never connected**. This information is very useful in tracking Wazuh agents that are not reporting to the dashboard for several reasons that need investigation. The image below shows the connection status of the agents on the Wazuh dashboard. The agent information as shown in the image is collected from the **wazuh-monitoring** index.



- **wazuh-statistics** index stores performance data related to the Wazuh server. This information is critical to ensuring the Wazuh server performs optimally with the

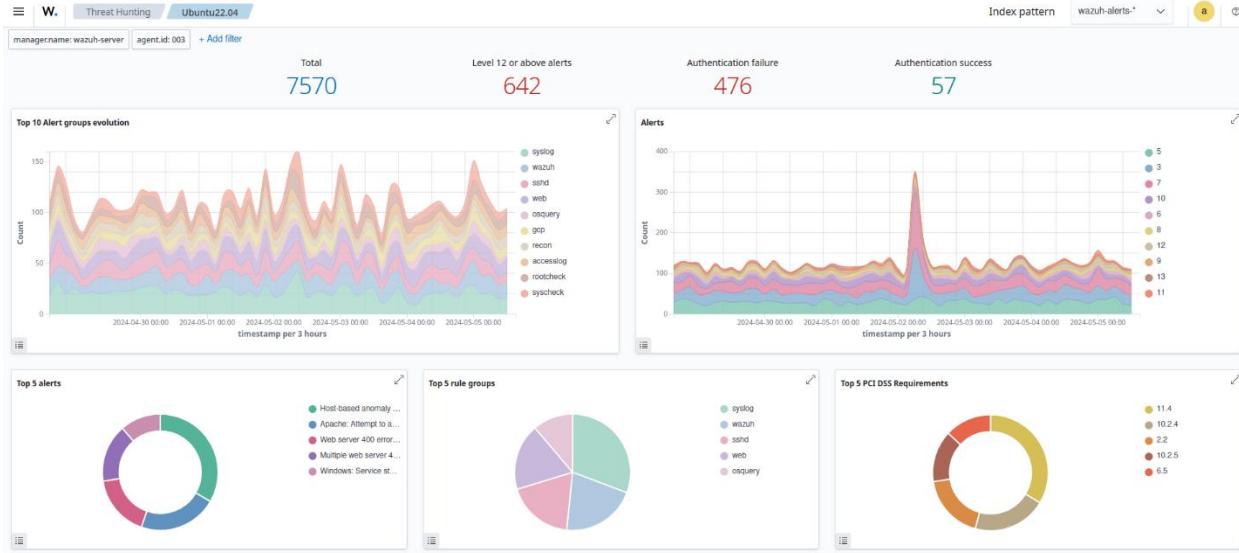
available computing resources. The image below shows performance-related events on the Wazuh dashboard.



Log data querying and visualization[Permalink to this headline](#)

The Wazuh dashboard offers log data querying and visualization capabilities. You can leverage the dashboard's intuitive interface to conduct complex searches and queries to extract meaningful insights from the log data collected by Wazuh.

Wazuh provides a set of predefined dashboards and visualizations out of the box, specifically tailored to security monitoring and compliance use cases. These dashboards provide insight into common security events such as failed logins, malware detection, and system anomalies. You can further customize these dashboards to suit your specific needs and requirements. Below is a sample image of the **Security event** dashboard showing several interesting information like **Top 5 PCI DSS Requirements**, **Top 5 alerts**, and **Alert groups evolution**.



The Wazuh dashboard enables users to explore log entries in real time, apply various filters, and drill down into specific events or time ranges. This flexibility allows security analysts to identify trends, anomalies, and potential security incidents within their environment.

Wazuh allows users to [create customized dashboards](#) that display key performance indicators, security metrics, and real-time monitoring of critical systems and applications. Users can assemble multiple visualizations, such as pie charts, line graphs, and heat maps, onto a single dashboard, providing a holistic view of their infrastructure's security posture. The following blog posts detailed how to query and create custom dashboards:

- [Monitoring macOS resources with Wazuh](#)
- [Monitoring Linux resources with Wazuh](#)
- [Monitoring Windows resources with Performance Counters](#)

Vulnerability detection[Permalink to this headline](#)

Software vulnerabilities are weaknesses in code that can allow attackers to gain access to or manipulate the behavior of an application. Vulnerable software applications are

commonly targeted by attackers to compromise endpoints and gain a persistent presence on targeted networks.

Vulnerability detection is the process of identifying these flaws before they are discovered and exploited by attackers. The goal of vulnerability detection is to identify vulnerabilities so that remediation can be carried out to prevent successful attacks.

The [Wazuh](#) agent uses the [Syscollector](#) module to collect inventory details from the monitored endpoint. It sends the collected data to the Wazuh server. Within the Wazuh server, the [Vulnerability Detection](#) module correlates the software inventory data with vulnerability content documents to detect vulnerable software on the monitored endpoint.

Wazuh detects vulnerable applications, generating risk reports, using our Cyber Threat Intelligence (CTI) platform. In this platform, we aggregate vulnerability data from diverse sources like operating system vendors and vulnerability databases, consolidating it into a unified, reliable repository. The process involves standardizing the varied formats into a common structure. Additionally, we maintain the integrity of our vulnerability data by doing the following.

- Rectifying format inconsistencies like version errors and typos.
- Completing missing information.
- Incorporating new cybersecurity vulnerabilities.

Subsequently, we merge this content, uploading the compiled documents to a cloud server. Finally, we publish these documents to our CTI API.

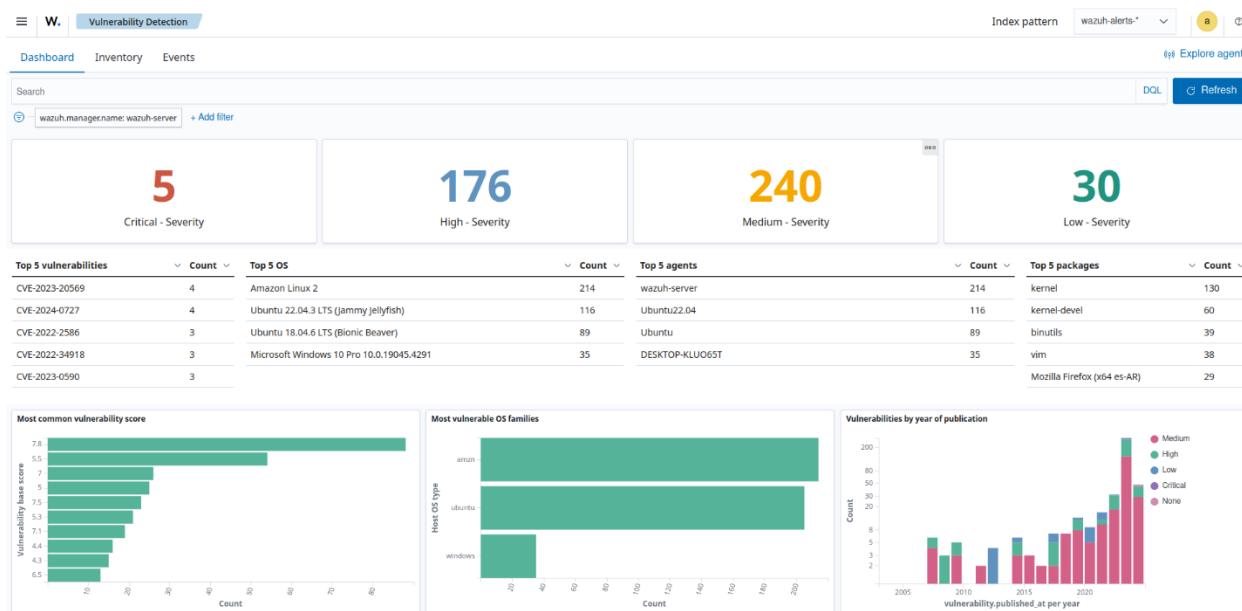
Relying on the Wazuh CTI, the [Vulnerability Detection](#) module supports a variety of operating systems, such as Windows, CentOS, Red Hat Enterprise Linux, Ubuntu, Debian, Amazon Linux, Arch Linux, and macOS operating systems, and applications.

Achieve comprehensive visibility

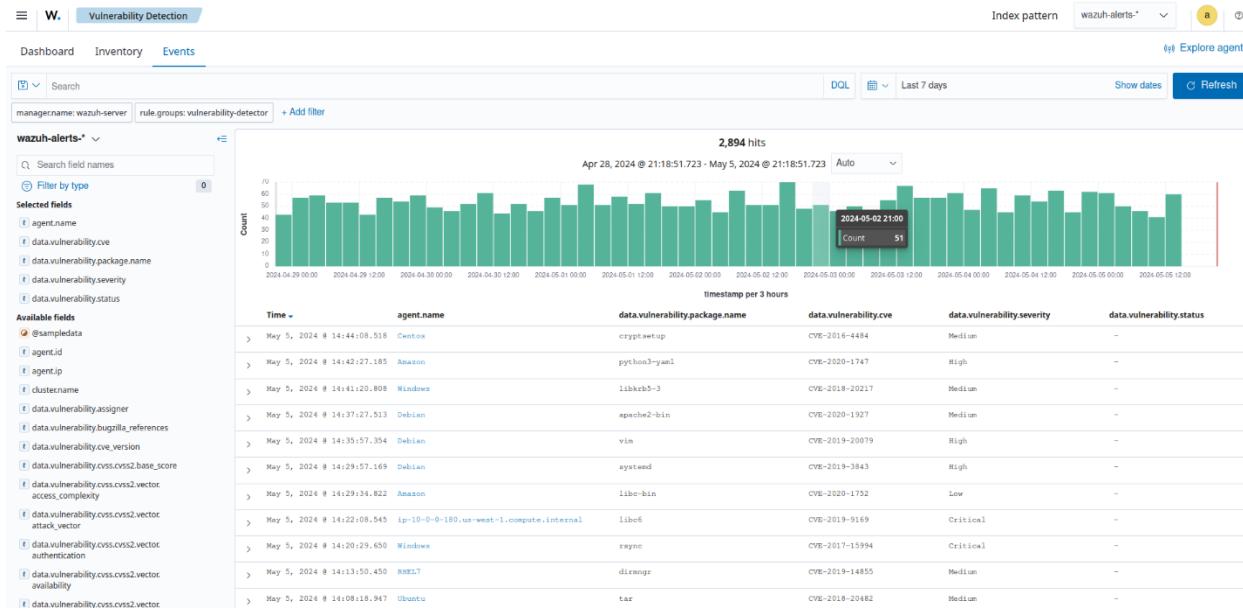
[Permalink to this headline](#)

The Vulnerability Detection module generates alerts for vulnerabilities discovered on the operating system and applications installed on the monitored endpoint. It correlates the software inventory collected by the Wazuh agent with the vulnerability content documents and displays the alert generated on the Wazuh dashboard. This provides a clear and comprehensive view of vulnerabilities identified in all monitored endpoints, allowing you to view, analyze and fix vulnerabilities.

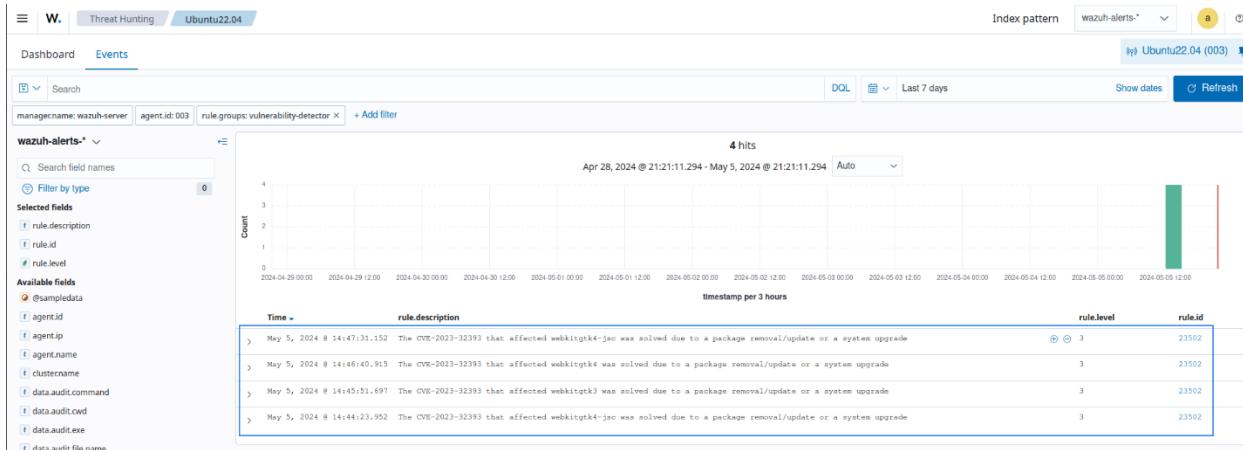
The vulnerability detection dashboard shows the frequency of occurrences in different categories such as package name, operating system, agent name, vulnerability ID, and alert severity. This allows analysts to direct their focus appropriately.



You can view the alerts generated on the dashboard when new vulnerabilities are discovered.



The alerts generated on the dashboard could also be a result of remediation activities. The image below shows alerts generated after an upgrade or an uninstallation of a package resolved a vulnerability.



Obtain actionable intelligence from vulnerability alerts [Permalink to this headline](#)

Wazuh vulnerability alerts contain relevant information about the identified vulnerability which can help users understand and decide on remediation steps. You can see an example of a vulnerability detection alert below:

This screenshot shows the Wazuh Vulnerability Detection interface. The top navigation bar includes 'Dashboard', 'Inventory', 'Events', 'Vulnerability Detection' (selected), 'Index pattern' (set to 'wazuh-alerts-*'), and 'Explore agent'. A search bar at the top left contains 'wazuh-server rule.groups:vulnerability-detector' with an 'Add filter' button. To the right are 'DQL' and 'Last 7 days' buttons, along with 'Show dates' and 'Refresh' buttons.

The main search results for 'wazuh-alerts-*' show a histogram titled '2,892 hits' from April 28, 2024, to May 5, 2024. The x-axis represents time in 3-hour intervals, and the y-axis represents the count of hits, ranging from 0 to 70. The histogram bars are teal.

Below the histogram is a table titled 'Expanded document' showing detailed information for a specific event. The event details are as follows:

Time	data.vulnerability.package.name	data.vulnerability.cve	data.vulnerability.severity	data.vulnerability.status
May 5, 2024 8:14:44:08.518	cryptsetup	CVE-2016-4484	Medium	-

Below the table is a 'Table' view showing a large list of vulnerability details. The table has columns for various metadata fields like 'id', 'rule.name', 'rule.level', etc., and corresponding values. The table is paginated with '1' and '2' at the bottom.

On the right side of the interface, there are buttons for 'View surrounding documents' and 'View single document'.

```
{  
  "_index": "wazuh-alerts-4.x-sample-threat-detection",  
  "_id": "e2ffSY8Be9PWdpLhA_nt",  
  "_version": 1,  
  "_score": null,  
  "_source": {  
    "predecoder": {},  
    "cluster": {  
      "name": "wazuh"  
    },  
    "agent": {  
      "ip": "197.17.1.4",  
      "name": "Centos",  
      "id": "005"  
    },  
    "manager": {  
      "name": "wazuh-server"  
    },  
    "data": {  
      "vulnerability": {  
        "severity": "Medium",  
        "package": {  
          "condition": "Package less or equal than 2.1.7.3-2",  
          "name": "cryptsetup",  
          "version": "2:1.6.6-5ubuntu2.1",  
          "architecture": "amd64"  
        },  
        "references": [  
          "http://hmarco.org/bugs/CVE-2016-4484/CVE-2016-4484_cryptsetup_initrd_shell.html",  
          "http://www.openwall.com/lists/oss-security/2016/11/14/13",  
          "http://www.openwall.com/lists/oss-security/2016/11/15/1",  
          "http://www.openwall.com/lists/oss-security/2016/11/15/4",  
          "http://www.openwall.com/lists/oss-security/2016/11/16/6",  
          "http://www.securityfocus.com/bid/94315",  
        ]  
      }  
    }  
  }  
}
```

```
"https://gitlab.com/cryptsetup/cryptsetup/commit/ef8a7d82d8d3716ae9b58179590f7908981fa0cb",
"https://nvd.nist.gov/vuln/detail/CVE-2016-4484",
"http://people.canonical.com/~ubuntu-security/cve/2016/CVE-2016-4484.html",
"https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-4484"

],
"cve_version": "4.0",
"assigner": "cve@mitre.org",
"published": "2017-01-23",
"cwe_reference": "CWE-287",
"title": "CVE-2016-4484 on Ubuntu 16.04 LTS (xenial) - low.",
"rationale": "The Debian initrd script for the cryptsetup package 2:1.7.3-2 and earlier allows physically proximate attackers to gain shell access via many log in attempts with an invalid password.",
"cve": "CVE-2016-4484",
"state": "Fixed",
"bugzilla_references": [
  "https://launchpad.net/bugs/1660701"
],
"cvss": {
  "cvss2": {
    "base_score": "7.200000",
    "vector": {
      "integrity_impact": "complete",
      "confidentiality_impact": "complete",
      "availability": "complete",
      "attack_vector": "local",
      "access_complexity": "low",
      "authentication": "none"
    }
  },
  "cvss3": {
    "base_score": "6.800000",
    "vector": {
      "user_interaction": "none",
      "privileges_required": "none",
      "vector_string": "CVSS:3.0/AV:L/AC:L/PR:N/UI:N/P:U/E:H/S:U/D:H/I:H/C:H"
    }
  }
}
```

```
        "integrity_impact": "high",
        "scope": "unchanged",
        "confidentiality_impact": "high",
        "availability": "high",
        "attack_vector": "physical",
        "access_complexity": "low",
        "privileges_required": "none"
    }
}
},
"updated": "2017-01-26"
}
},
"@sampledata": true,
"rule": {
    "firedtimes": 290,
    "mail": false,
    "level": 7,
    "pci_dss": [
        "11.2.1",
        "11.2.3"
    ],
    "tsc": [
        "CC7.1",
        "CC7.2"
    ],
    "description": "CVE-2016-4484 affects cryptsetup",
    "groups": [
        "vulnerability-detector"
    ],
    "id": "23504",
    "gdpr": [
        "IV_35.7.d"
    ]
},
"location": "vulnerability-detector",
```

```
"id": "1580123327.49031",
"decoder": {
  "name": "json"
},
"timestamp": "2024-05-05T17:44:08.518+0000"
},
"fields": {
  "data.vulnerability.published": [
    "2017-01-23T00:00:00.000Z"
  ],
  "data.vulnerability.updated": [
    "2017-01-26T00:00:00.000Z"
  ],
  "timestamp": [
    "2024-05-05T17:44:08.518Z"
  ]
},
"highlight": {
  "manager.name": [
    "@opensearch-dashboards-highlighted-field@wazuh-server@/opensearch-dashboards-highlighted-field@"
  ],
  "rule.groups": [
    "@opensearch-dashboards-highlighted-field@vulnerability-detector@/opensearch-dashboards-highlighted-field@"
  ]
},
"sort": [
  1714931048518
]
}
```

As you can see above, the alert contains key information about the detected vulnerability. This information includes the CVE information, reference links for further research, and a description that provides a concise explanation of the vulnerability.

Track vulnerability remediation[Permalink to this headline](#)

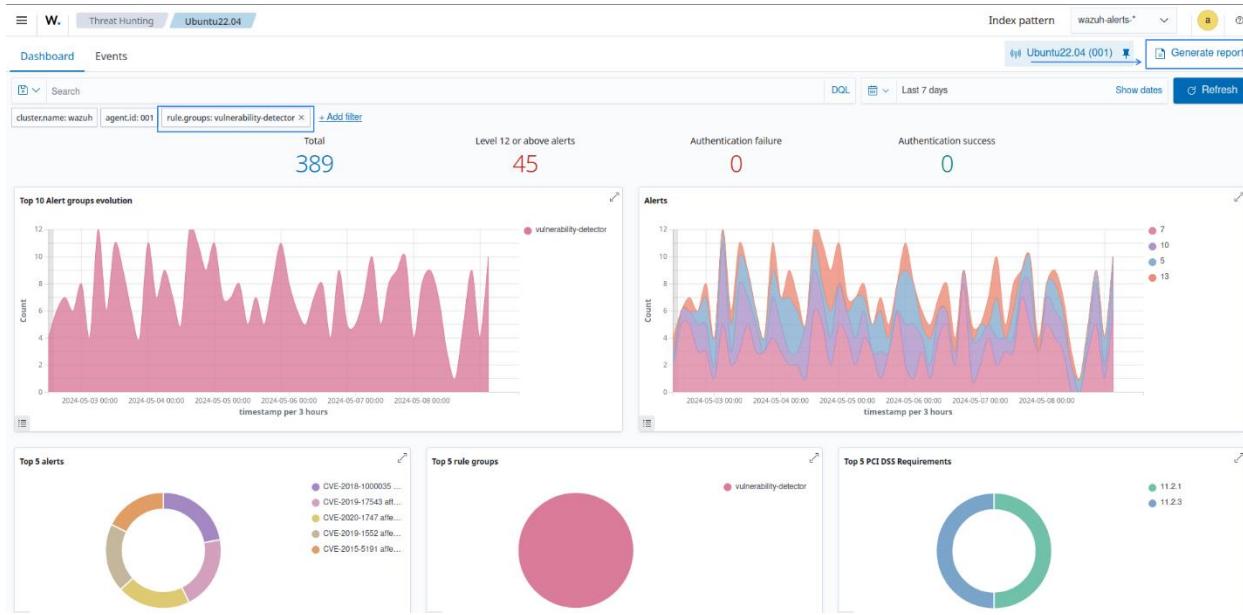
The Wazuh Vulnerability Detection module also allows you to confirm when a vulnerability has been remediated. This feature detects when a patch or software upgrade resolves a previously detected vulnerability. The feature is enabled using the **hotfixes** option and is available for Windows endpoints.

The screenshot shows a Wazuh interface for tracking vulnerabilities. At the top, there are tabs for 'Vulnerability De...' and 'Windows11'. The main area displays a document titled 'May 4, 2024 8:22:47:57.342 Microsoft Edge' with the identifier 'CVE-2021-24113'. The document status is 'Solved'. The document content is presented in a JSON-like table:

Field	Value
_index	wazuh-alerts-4.x-2024.05.05
agent.id	001
agent.ip	192.168.0.154
agent.name	Windows11
data.vulnerability.cve	CVE-2021-24113
data.vulnerability.cvss.cvss2.base_score	5.800000
data.vulnerability.enumeration	CVE
data.vulnerability.package.architecture	i686
data.vulnerability.package.name	Microsoft Edge
data.vulnerability.package.version	124.0.2476.67
data.vulnerability.published	Feb 25, 2021 8:20:15:16.000
data.vulnerability.reference	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2021-24113
data.vulnerability.severity	Medium
data.vulnerability.status	Solved
data.vulnerability.title	CVE-2021-24113 affecting Microsoft Edge was solved
data.vulnerability.type	Packages
data.vulnerability.updated	Dec 29, 2023 8:14:15:59.000

Use vulnerability reports to identify critical security issues[Permalink to this headline](#)

Wazuh provides users with the ability to download a report that contains security events related to discovered and resolved vulnerabilities. This feature allows users to identify endpoints with unresolved vulnerabilities and keep track of remediation activities.



Incident response[Permalink to this headline](#)

A security incident refers to any adverse event or activity that risks or threatens the confidentiality, integrity, or availability of digital assets, networks, data, or resources. Such incidents include unauthorized access, data breaches, malware infections, denial-of-service attacks, and any other activities that compromise the security posture of an organization's information technology environment.

The goal of incident response is to effectively handle a security incident and restore normal business operations as quickly as possible. As organizations' digital assets continuously grow, managing incidents manually becomes increasingly challenging, hence the need for automation.

Automated incident response involves automatic actions taken when responding to security incidents. These actions can include isolating compromised endpoints, blocking malicious IP addresses, quarantining infected devices, or disabling compromised user accounts. By

1 prevent or minimize the impact of incidents, and efficiently handle a large volume of security events.

Wazuh Active Response module[Permalink to this headline](#)

The Wazuh [Active Response](#) module allows users to run automated actions when incidents are detected on endpoints. This improves an organization's incident response processes, enabling security teams to take immediate and automated actions to counter detected threats.

You can also configure the actions to be either stateless or stateful. Stateless active responses are one-time actions while stateful responses revert their actions after some time.

Default active response actions[Permalink to this headline](#)

Out-of-the-box scripts are available on every operating system that runs the Wazuh agents.

Some of the [default active response](#) scripts include:

Name of script	Description
disable-account	Disables a user account
firewall-drop	Adds an IP address to the iptables deny list.
firewalld-drop	Adds an IP address to the firewalld drop list.
restart.sh	Restarts the Wazuh agent or server.
netsh.exe	Blocks an IP address using netsh.

Custom active response actions[Permalink to this headline](#)

One of the benefits of the Wazuh Active Response module is its adaptability. Wazuh allows security teams to create [custom active response](#) actions in any programming language, tailoring them to their specific needs. This ensures that when a threat is detected, the response can be customized to align with the organization's requirements.

Automating incident response with Wazuh[Permalink to this headline](#)

To leverage the Wazuh Active Response module, you need to [configure](#) the action to be carried out when a specific event occurs on a monitored endpoint. For example, you can configure the Wazuh Active Response module to delete a malicious executable from an infected endpoint. In the examples that follow, we show how the Wazuh Active Response module handles different incidents.

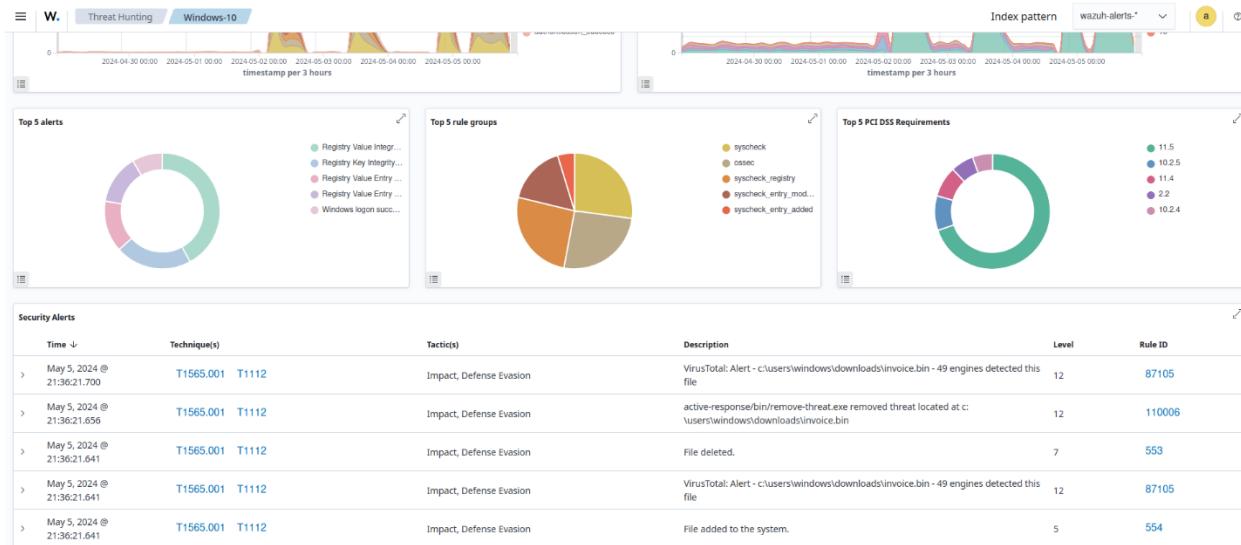
Removing malware[Permalink to this headline](#)

You can use the Wazuh Active Response module in conjunction with the [File Integrity Monitoring](#) module and [VirusTotal integration](#) to detect and remove malicious files from an endpoint.

The image below shows the following activities:

1. Rule ID [554](#) is fired when a file is added to the [Downloads](#) directory which is monitored with the Wazuh File Integrity Monitoring module.
2. Rule ID [87105](#) triggers when Wazuh extracts the file hash, requests data about the file hash from the VirusTotal database via its API, and receives a malicious file response.

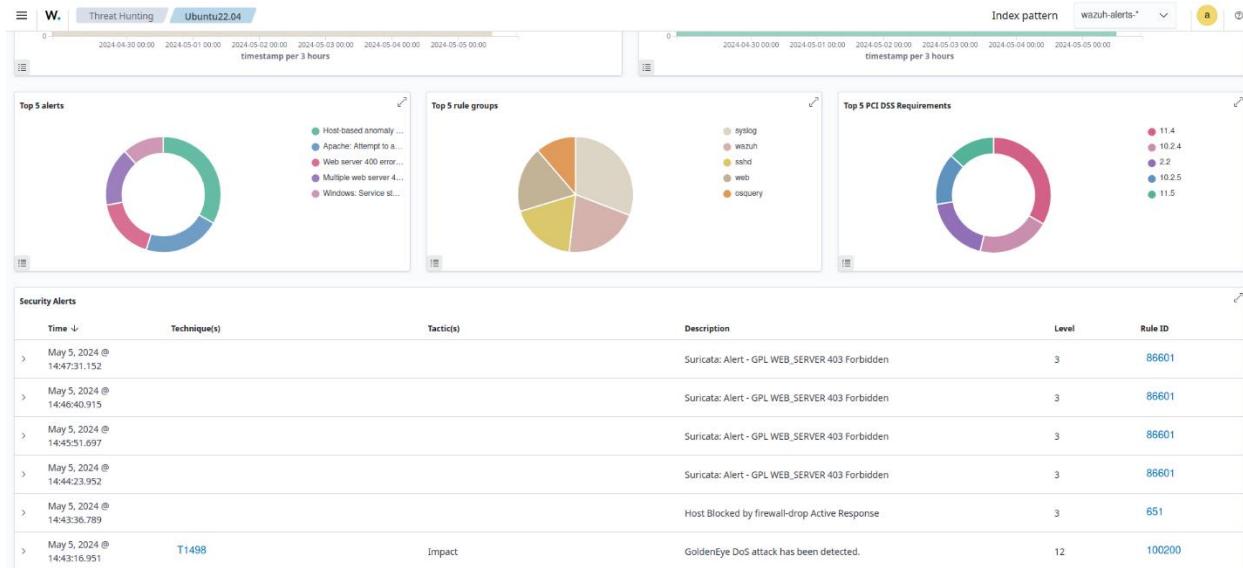
3. Rule ID **553** is fired when a file is deleted from the **Downloads** directory which is monitored with the Wazuh File Integrity Monitoring module.
4. Rule ID **110006** is fired when the Wazuh Active Response module deletes the malicious file from the endpoint.



In this scenario, the Wazuh Active Response module automatically removes the malicious file, reducing the time between threat detection and mitigation.

Responding to DoS attacks [Permalink to this headline](#)

The primary goal of a DoS attack is to render the target inaccessible to legitimate users, causing a denial of service. In the image below, we show how the Wazuh Active Response module blocks malicious IP addresses performing a DoS against a web server on an Ubuntu endpoint.

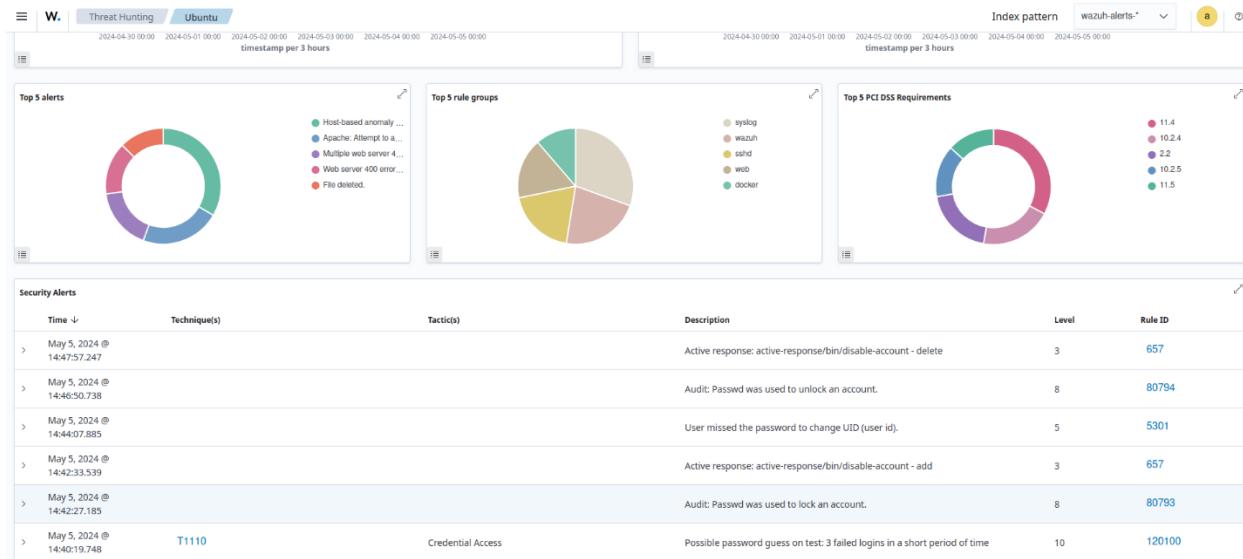


In this case, the Wazuh Active Response module automatically blocks the malicious hosts from causing a DoS attack on the web server. Thereby ensuring the availability of the web server to the authorized users.

Disabling a user account after a brute-force attack [Permalink to this headline](#)

Account lockout is a security measure used to defend against brute force attacks by limiting the number of login attempts a user can make within a specified time. We use the Wazuh Active Response module to disable the user account whose password is being guessed by an attacker.

In the image below, the Wazuh Active Response module disables the account on a Linux endpoint and re-enables it again after 5 minutes.



In this scenario, when an attacker tries to guess a user's password repeatedly and fails, the account becomes temporarily inaccessible. This impedes attackers who rely on brute-force methods to guess user account passwords.

By utilizing the Wazuh Active Response module, security teams can automate responses to different incidents. Thereby ensuring efficient incident response and a more resilient cybersecurity posture.