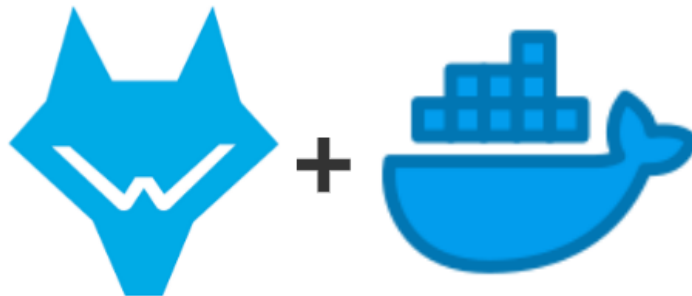


---

# Build Open Source SIEM HA Using Wazuh + Docker Swarm *by : Ardita*

## SUMMARY



Wazuh is a free and open source platform for threat detection, security monitoring, incident response and regulatory compliance. It can be used to monitor endpoints, cloud services and containers, and to aggregate and analyze data from external sources. (source : [wazuh docs](#))

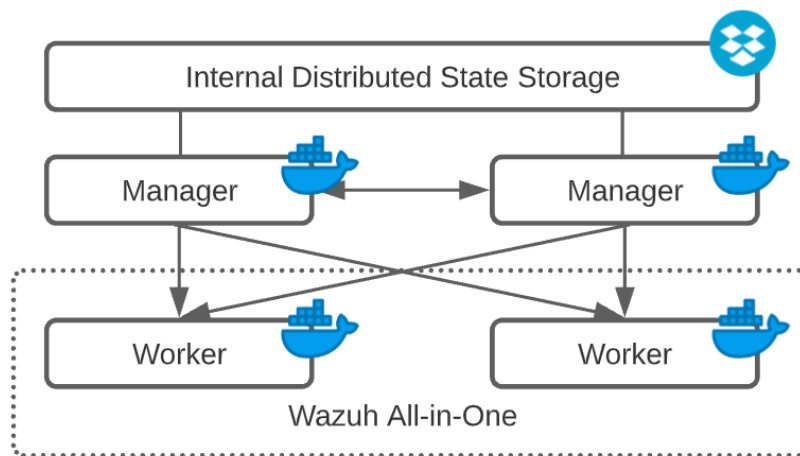
One of the wazuh functions can be used to detect brute-force attacks. Brute forcing SSH (on Linux) or RDP (on Windows) are common attack vectors. Wazuh provides out-of-the-box rules capable of identifying brute-force attacks by correlating multiple authentication failure events.

## PREREQUISITES

For All-in-One Deployment :

1. **Operating system** (Amazon Linux 2, CentOS 7 and later, Debian 8 ELTS and later, Fedora Linux 31 and later, openSUSE Tumbleweed, Leap 15.2 and later, Oracle Linux 6 Extended and later, Red Hat Enterprise Linux 6 ELS and later, Ubuntu 14.04 ESM and later)
2. Min 4GB of **RAM**
3. Min 2 **CPU** cores
4. **Docker**
5. **Portainer** (Optional: for manage docker via dashboards)

## DOCKER SWARM TOPOLOGY



## ACTION

**Note :** Assuming docker and portainer have been installed !

### Install Wazuh All-in-One

1. Clone wazuh repository to your system using git, current wazuh stable version on official documentation is v4.2.5.

```
git clone https://github.com/wazuh/wazuh-docker.git -b v4.2.5 --depth=1
```

2. Move to wazuh-docker directory.

```
cd wazuh-docker/
```

3. Deploy wazuh All-in-One using Docker Swarm stack deploy, it will be distributed on all docker worker nodes.

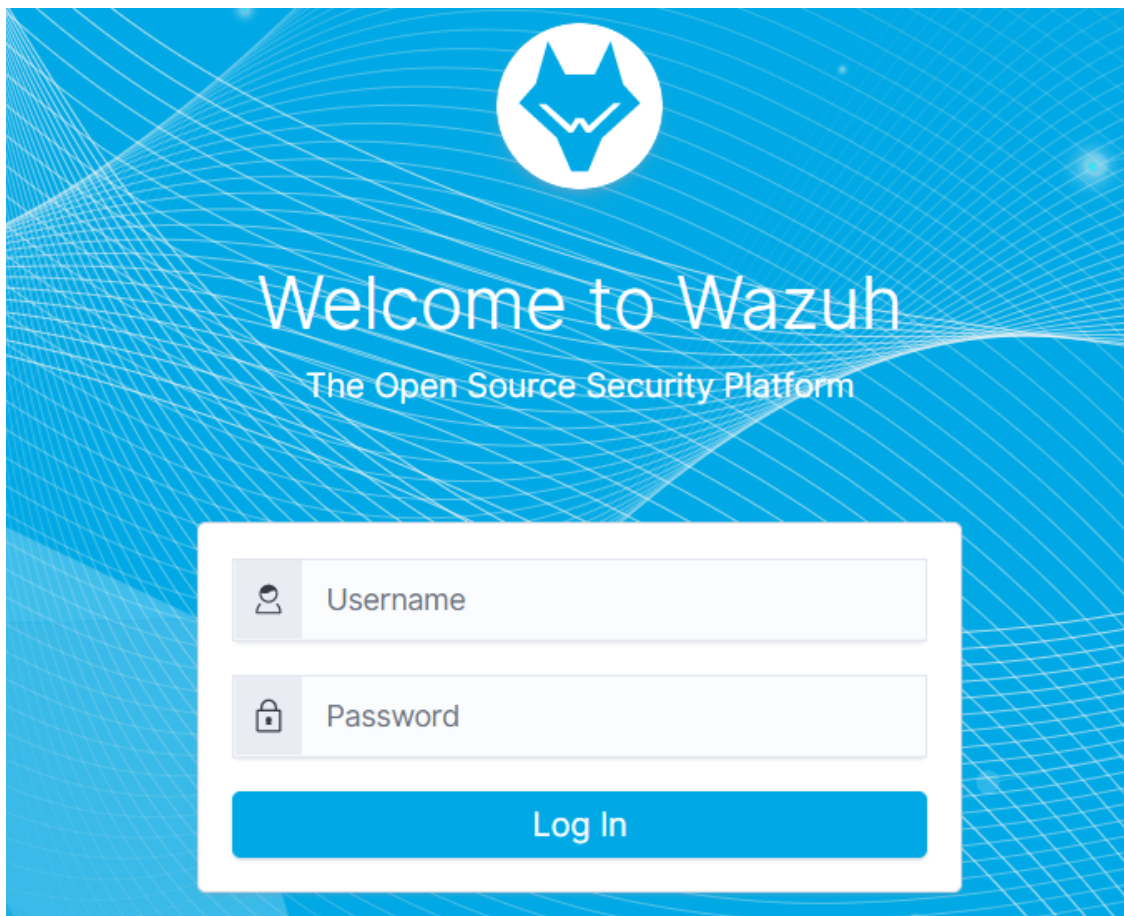
```
docker stack deploy -c docker-compose.yml Wazuh
```

4. Check wazuh service is available on docker swarm via CLI or Portainer.

```
[root@docker01 wazuh-docker]# docker service ls
ID                NAME                MODE                REPLICAS
je73cnm5hf3s     jenkins             replicated          1/1
n93yld65vbe9     portainer_agent     global             2/2
ycdhorwv3jd      portainer_portainer replicated          1/1
u2x3bo17pocj     wazuh_elasticsearch replicated          1/1
nrbdy3ioe90s     wazuh_kibana        replicated          1/1
u765wypkvcao     wazuh_wazuh         replicated          1/1
```

Services			
		<a href="#">Update</a>	<a href="#">Remove</a>
Search...			
<input type="checkbox"/>	Name ↓↑	Image	Scheduling Mode
<input type="checkbox"/>	> wazuh_elasticsearch	amazon/opendistro-for-elasticsearch:1.13.2	replicated 1 / 1 ↓ Scale
<input type="checkbox"/>	> wazuh_kibana	wazuh/wazuh-kibana-odfe:4.2.5	replicated 1 / 1 ↓ Scale
<input type="checkbox"/>	> wazuh_wazuh	wazuh/wazuh-odfe:4.2.5	replicated 1 / 1 ↓ Scale

5. Access wazuh dashboards, the default **user** is **admin** and **password** is **admin**.



6. Install wazuh agent on an example Linux server using script on Wazuh > Agents > Deploy New Agent.

7. Choose the operating system, version, and architecture based on the server that will be monitoring.

## Deploy a new agent

- 1 Choose the Operating system**  
 Red Hat / CentOS  Debian / Ubuntu  Windows  MacOS
- 2 Choose the version**  
 CentOS5  CentOS6 or higher
- 3 Choose the architecture**  
 i386  x86\_64  armhf  aarch64

8. Define wazuh server address (can be predefined on setting) and assign the agent to a group (the default group is default).

- 4 Wazuh server address**

You can predefined the Wazuh server address with the `enrollment.dns` Wazuh app setting.
- 5 Assign the agent to a group**

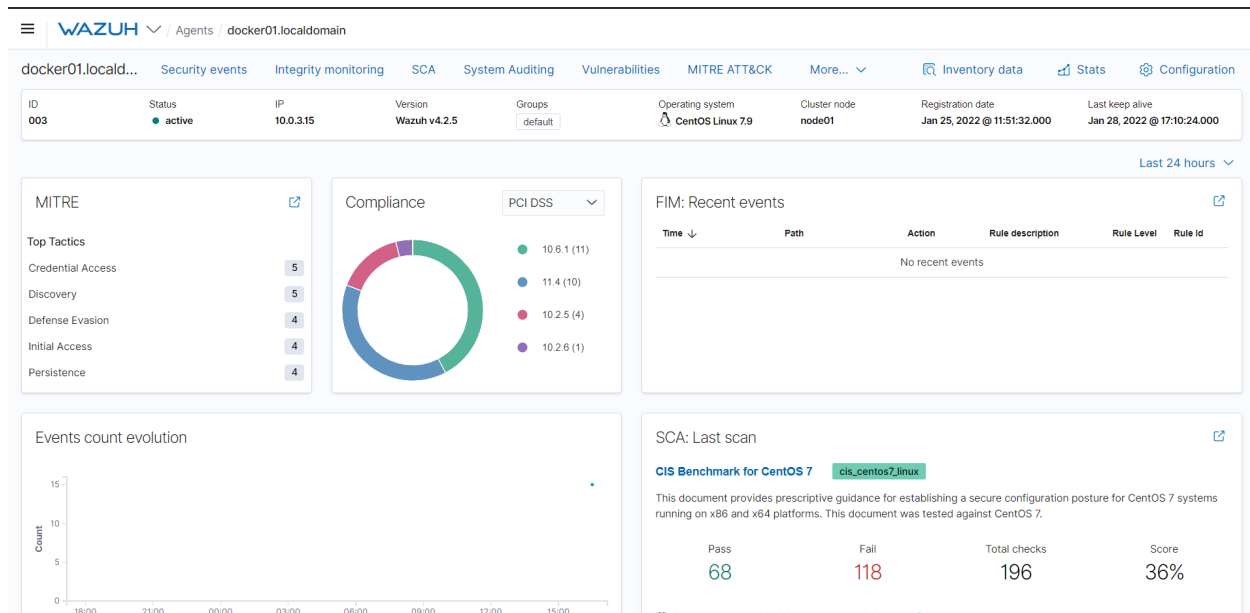
Select one or more existing groups

9. Wazuh will be create a simple script, run on the server that will be monitoring and reload the service.
10. Check on dashboards.

## RESULTS

### Wazuh Dashboards

Wazuh has very complex dashboards, we can use benchmark functions, security events, inventory data and others to create monitoring on the server. One of the functions is to detect brute force attacks.



### How to detect a brute-force ?

We can use brute-force attack tools such as hydra for a demo.

1. Install hydra tools or use pentest operating systems such as Kali Linux.

```
yum install -y hydra
```

2. Run hydra command for brute-force attack.

```
hydra -l exampleuser -p wrong_password 192.168.56.101 ssh
```

3. Check on wazuh dashboards > discover, user exampleuser will try to login and wazuh create log for this action.



The screenshot shows a Wazuh dashboard interface. At the top, a search bar contains the text "sshd: Attempt to login using a non-existent user". Below this, the "Expanded document" section displays a table of log fields. The table has two columns: field names and their corresponding values. The field "data.srcuser" is highlighted in blue, and its value "exampleuser" is also highlighted with a red box. The "full\_log" field contains the full log message: "Jan 28 05:26:54 docker01 sshd[6340]: Failed password for invalid user exampleuser from 192.168.56.105 port 36554 ssh2".

Field	Value
_id	-Rs5oH4B3f1EAXPqWKEY
_index	wazuh-alerts-4.x-2022.01.28
_score	-
_type	_doc
agent.id	003
agent.ip	10.0.3.15
agent.name	docker01.localdomain
data.srcip	192.168.56.105
data.srcuser	exampleuser
decoder.name	sshd
decoder.parent	sshd
full_log	Jan 28 05:26:54 docker01 sshd[6340]: Failed password for invalid user exampleuser from 192.168.56.105 port 36554 ssh2

4. In the next section, wazuh defines this action based on MITRE RULE as a brute-force attack.

rule.mitre.id	T1110
rule.mitre.tactic	Credential Access
rule.mitre.technique	Brute Force
rule.nist_800_53	AU.14, AC.7, AU.6
rule.pci_dss	10.2.4, 10.2.5, 10.6.1
rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
timestamp	Jan 28, 2022 @ 17:26:55.944

5. Next step we will try to make a reporting dashboard, See You...

## NOTES

<https://docs.docker.com/engine/swarm/>

<https://documentation.wazuh.com/current/getting-started/index.html>

<https://documentation.wazuh.com/current/proof-of-concept-guide/detect-brute-force-attack.html>