SSH Brute Force Attack Detection & Active Response with Wazu

SSH Brute Force Attack Detection & Active Response with Wazuh

Created By: Amir Raza

Follow Me: www.linkedin.com/in/amirsoc

Overview

Wazuh is a powerful open-source security platform for threat detection, monitoring, and automated response. This guide shows how to detect and respond to SSH brute-force attacks using Wazuh's Active Response feature.

What is an SSH Brute Force Attack?

An SSH brute-force attack involves an attacker trying many username and password combinations to gain unauthorized access to a server. If successful, it can lead to a serious security breach.

Lab Setup

Victim Machine IP: 100.119.94.32 (ubuntu - Victim)

```
amir@Ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:02:4c:1c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 65778sec preferred_lft 65778sec
    inet6 fd17:625c:f037:2:4202:73c2:4fee:4ac/64 scope global temporary dynamic
       valid_lft 86241sec preferred_lft 14241sec
    inet6 fd17:625c:f037:2:a00:27ff:fe02:4c1c/64 scope global dynamic mngtmpaddr
       valid_lft 86241sec preferred_lft 14241sec
    inet6 fe80::a00:27ff:fe02:4c1c/64 scope link
       valid_lft forever preferred_lft forever
3: tailscale0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1280 qdisc fq_codel state
 UNKNOWN group default qlen 500
    link/none
    inet 100.119.94.32/32 scope global tailscale0
```

Wazuh Server IP: 100.108.221.35(kali)

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.90.80  netmask 255.255.255.0  broadcast 10.10.90.255
        inet6 fe80::20c:29ff:fed9:85dc  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:d9:85:dc  txqueuelen 1000  (Ethernet)
        RX packets 232744  bytes 153444380 (146.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 112839  bytes 17993134 (17.1 MiB)
        TX errors 0  dropped 222 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 541116  bytes 210181024 (200.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 541116  bytes 210181024 (200.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tailscale0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1280
        inet 100.108.221.35  netmask 255.255.255.255  destination 100.108.221.35
        inet6 fe80::bd0b:c31:6bc2:bb7b  prefixlen 64  scopeid 0x20<link>
        inet6 fd7a:115c:a1e0::501:dd26  prefixlen 128  scopeid 0x0<global>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 3512  bytes 526400 (514.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3202  bytes 455580 (444.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 1: Make Sure SSH Service Is Running on the Victim Machine

Before testing anything, check if the SSH service is active on your victim machine (Ubuntu/Kali). This ensures the system is ready to receive SSH login attempts.

Use this command:

```
sudo systemctl status ssh
```

```
amir@Ubuntu:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enab>
     Active: active (running) since Tue 2025-05-13 04:42:02 UTC; 4s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 6019 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 6021 (sshd)
      Tasks: 1 (limit: 4549)
     Memory: 1.2M (peak: 1.5M)
        CPU: 15ms
     CGroup: /system.slice/ssh.service
             └─6021 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

May 13 04:42:02 Ubuntu systemd[1]: Starting ssh.service - OpenBSD Secure Shell >
May 13 04:42:02 Ubuntu sshd[6021]: Server listening on :: port 22.
May 13 04:42:02 Ubuntu systemd[1]: Started ssh.service - OpenBSD Secure Shell s>
```

If SSH is running correctly, you'll see a message saying something like:

Active: active (running)

Step 2: Simulate a Brute Force Attack Using Hydra

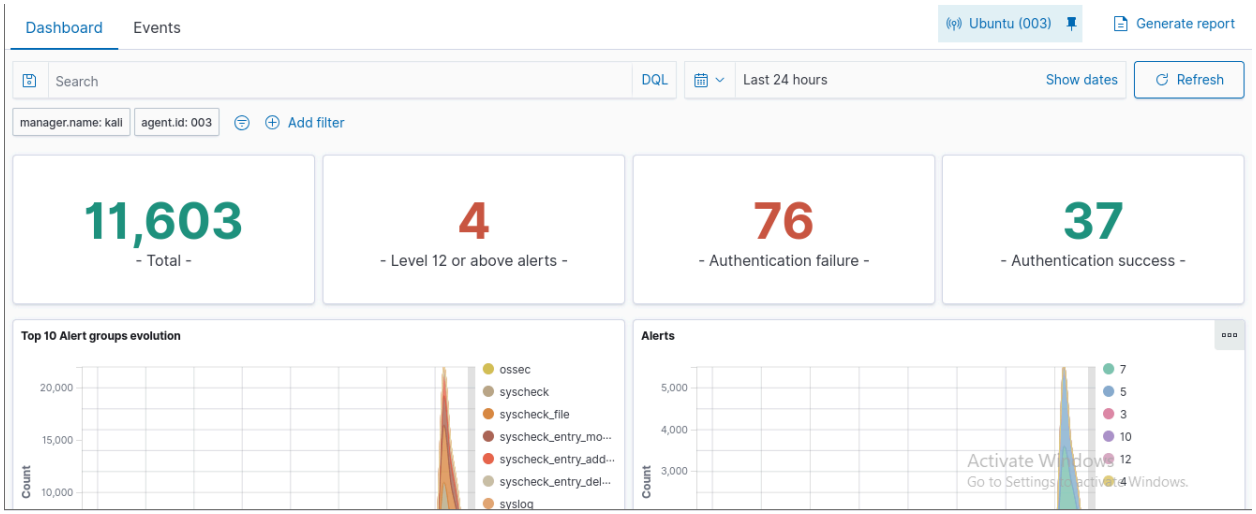
Now, from the attacker machine (like Kali Linux), use Hydra to launch a brute-force attack on the victim's SSH service.

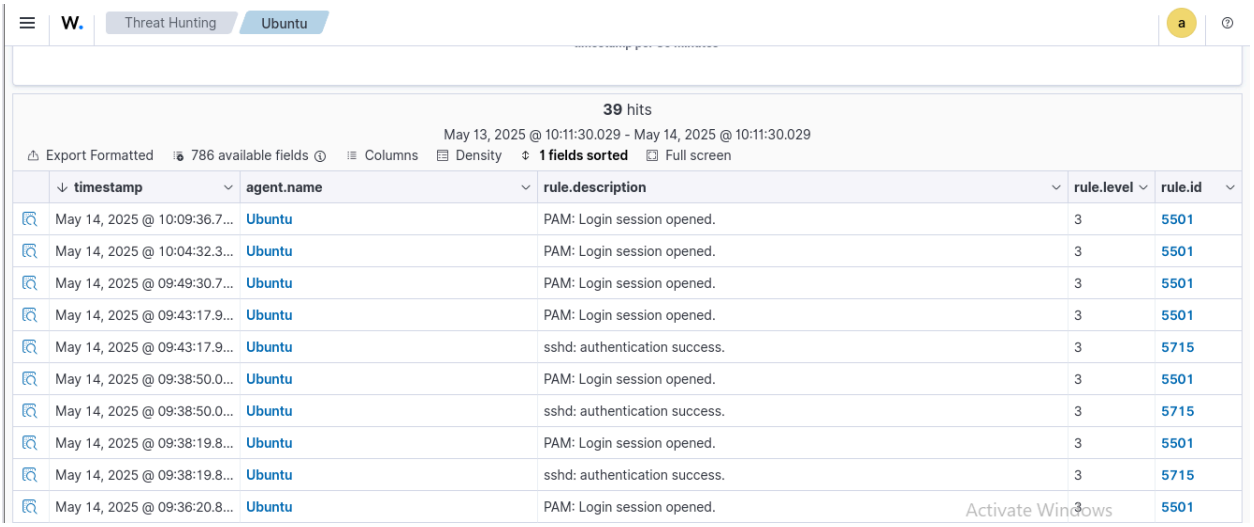
Hydra -l amir -p mypasswords.txt -t 4 -vV 100.119.94.32 ssh

```
┌──(kali㉿kali)-[~]
└─$ hydra -l amir -P mypasswords.txt -t 4 -vV 100.119.94.32 ssh

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding
, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-14 09:38:47
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1/p:4), ~1 try per task
[DATA] attacking ssh://100.119.94.32:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://amir@100.119.94.32:22
[INFO] Successful, password authentication is supported by ssh://100.119.94.32:22
[ATTEMPT] target 100.119.94.32 - login "amir" - pass "ali%987" - 1 of 4 [child 0] (0/0)
[ATTEMPT] target 100.119.94.32 - login "amir" - pass "altaf$345" - 2 of 4 [child 1] (0/0)
[ATTEMPT] target 100.119.94.32 - login "amir" - pass "root123" - 3 of 4 [child 2] (0/0)
[ATTEMPT] target 100.119.94.32 - login "amir" - pass "[redacted]" - 4 of 4 [child 3] (0/0)
[22][ssh] host: [redacted]   login: amir   password: [redacted]
[STATUS] attack finished for 100.119.94.32 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-14 09:38:50
```

After getting the password by using hydra tool I successfully login into another target machine using their password.

In the **Wazuh Dashboard**, go to the **Threat Hunting** section. Under the **Events** tab, you'll see multiple logs related to **authentication failures**, indicating SSH brute-force attempts.





timestamp	agent.name	rule.description	rule.level	rule.id
May 14, 2025 @ 10:09:36.7...	Ubuntu	PAM: Login session opened.	3	5501
May 14, 2025 @ 10:04:32.3...	Ubuntu	PAM: Login session opened.	3	5501
May 14, 2025 @ 09:49:30.7...	Ubuntu	PAM: Login session opened.	3	5501
May 14, 2025 @ 09:43:17.9...	Ubuntu	PAM: Login session opened.	3	5501
May 14, 2025 @ 09:43:17.9...	Ubuntu	sshd: authentication success.	3	5715
May 14, 2025 @ 09:38:50.0...	Ubuntu	PAM: Login session opened.	3	5501
May 14, 2025 @ 09:38:50.0...	Ubuntu	sshd: authentication success.	3	5715
May 14, 2025 @ 09:38:19.8...	Ubuntu	PAM: Login session opened.	3	5501
May 14, 2025 @ 09:38:19.8...	Ubuntu	sshd: authentication success.	3	5715
May 14, 2025 @ 09:36:20.8...	Ubuntu	PAM: Login session opened.	3	5501

Table JSON

@timestamp	May 14, 2025 @ 09:43:17.935
_index	wazuh-alerts-4.x-2025.05.14
agent.id	003
agent.ip	10.0.2.15
agent.name	Ubuntu
data.dstuser	amir(uid=1000)
data.srcuser	amir
data.uid	0
decoder.name	pam
decoder.parent	pam
full_log	May 14 13:43:16 Ubuntu sshd[10629]: pam_unix(sshd:session): session opened for user amir(uid=1000) by amir(uid=0)
id	1747230197.14754660
input.type	log
location	journald
manager.name	kali

rule.description	PAM: Login session opened.
rule.firedtimes	31
rule.gdpr	IV_32.2
rule.gpg13	7.8, 7.9
rule.groups	pam, syslog, authentication_success
rule.hipaa	164.312.b
rule.id	5501
rule.level	3
rule.mail	false
rule.mitre.id	T1078
rule.mitre.tactic	Defense Evasion, Persistence, Privilege Escalation, Initial Access
rule.mitre.technique	Valid Accounts
rule.nist_800_53	AU.14, AC.7
rule.pci_dss	10.2.5
rule.tsc	CC6.8, CC7.2, CC7.3
timestamp	May 14, 2025 @ 09:43:17.935

Step 3: Activate Wazuh's Active Response Feature

Open the Wazuh configuration file:

Sudo nano /var/ossec/etc/ossec.conf

Add the following line:

```
</command>
<command>
  <name>firewall-drop</name>
  <executable>firewall-drop</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>

<active-response>
  <command>firewall-drop</command>
  <location>local</location>
  <rules_id>5760</rules_id>
  <timeout>180</timeout>
</active-response>
```

Step 4: Restart the Wazuh Manager and Agent Services

Restart the Wazuh Manager

Run the following command to restart the Wazuh Manager service:

sudo systemctl restart wazuh-manager

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart wazuh-manager
```

Restart the Wazuh Agent

Next, restart the Wazuh Agent with this command:

sudo systemctl restart wazuh-agent

This ensures all recent configuration changes take effect properly.

```
amir@Ubuntu:~$ sudo systemctl restart wazuh-agent
amir@Ubuntu:~$
```

Step 5: Re-run the Brute Force Attack

Now, simulate the SSH brute-force attack again using Hydra:

sudo hydra -L user.txt -P pass.txt ssh://10.10.90.118

This time, Wazuh should detect the attack and **automatically block the attacker's IP address** using active response.

This confirms that Wazuh successfully detected and responded to the brute-force attempt.

```
┌──(kali㉿kali)-[~]
└─$ hydra -l babli  -P mypasswords.txt  ssh://100.119.94.32

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purpos
es (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-17 07:40:11
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 74 login tries (l:1/p:74), ~5 tries per task
[DATA] attacking ssh://100.119.94.32:22/
[STATUS] 58.00 tries/min, 58 tries in 00:01h, 18 to do in 00:01h, 14 active
[STATUS] 38.00 tries/min, 76 tries in 00:02h, 1 to do in 00:01h, 4 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-17 07:42:25
```

Step 6: Confirm Active Response in Wazuh Logs:

Open the **Wazuh Dashboard**, go to the **Threat Hunting** → **Events** tab, and search for a log entry that says something like:

Host blocked by firewall-drop Active Response

Summary

Wazuh's **Active Response** is a powerful feature that helps protect your Linux server from brute-force SSH attacks. Here's what makes it so effective:

- ✅ **Real-time Detection**: It continuously monitors your system for suspicious login attempts, like multiple failed SSH logins.

- 🔐 **Automatic IP Blocking**: As soon as a brute-force attempt is detected, Wazuh can automatically block the attacker's IP using firewall rules — no manual action needed.

- ⚙️ **Customizable Responses**: You can configure how Wazuh responds, such as how long to block the IP or what method to use (e.g., firewalld, iptables, etc.).

- 📊 **Detailed Logging**: All actions are logged and can be viewed in the Wazuh Dashboard, so you always know what's happening.

- 🚀 **Boosts Server Security**: By automating threat response, Wazuh helps you stay a step ahead of attackers and reduces the risk of intrusion.