ALERT TRIAGE PRACTICE

Introduction

This document details the process of performing an alert triage using **Wazuh**, **AlienVault OTX**, and **VirusTotal**. The goal is to simulate mock alerts, analyze them, validate potential threats, and document findings. The task involves generating SSH login failure events, observing them in Wazuh, and performing threat intelligence validation.

Tools Used:

- **Wazuh:** Open-source security monitoring and SIEM tool.
- **AlienVault OTX:** Threat intelligence platform for IOC validation.
- **Windows PowerShell:** Used to generate mock authentication events.

Setting up the Environment

1. **Ensure Wazuh Manager is running** on Ubuntu.
2. **Install and connect the Wazuh Agent** on Windows 11.
3. Confirm **logs are being received** by Wazuh:
    - Go to **Wazuh → Security Events**.
    - Verify that the Windows agent is online and sending events.

Generating Mock Alerts on Windows 11

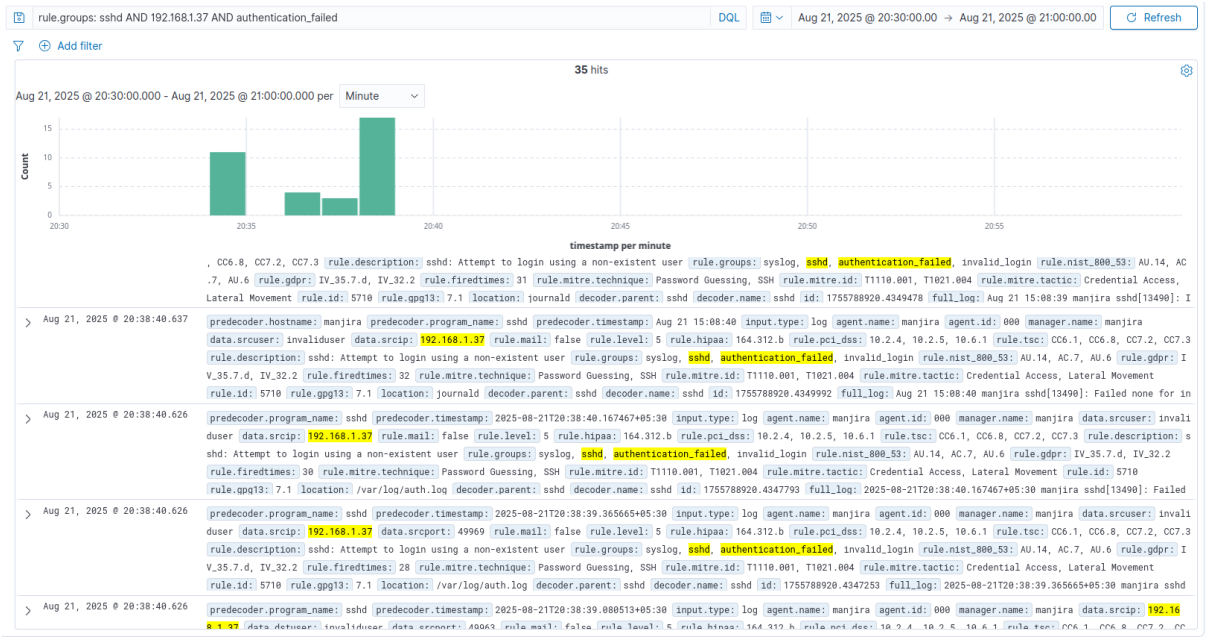
1. Open **PowerShell as Administrator**.
2. Execute the following commands sequentially to simulate failed logins 5 times:
   net user fakeuser1 Password123 /add
   net user fakeuser1 /delete
3. Each command generates a **Windows Security Event**, simulating brute-force or invalid login attempts.
4. Wazuh captures these events automatically.

```
PS C:\WINDOWS\system32> for ($i=1; $i -le 10; $i++) {
>>     ssh invaliduser@192.168.1.35
>> }
>>
The authenticity of host '192.168.1.35 (192.168.1.35)' can't be established.
ED25519 key fingerprint is SHA256:+ou6TPoe8viw1Q6nAhVh3NOGgizTyNqXwXHjGHk9Pjk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.35' (ED25519) to the list of known hosts.
invaliduser@192.168.1.35's password:
Permission denied, please try again.
invaliduser@192.168.1.35's password:
Permission denied, please try again.
invaliduser@192.168.1.35's password:
invaliduser@192.168.1.35: Permission denied (publickey,password).
invaliduser@192.168.1.35's password:
Permission denied, please try again.
invaliduser@192.168.1.35's password:
Connection closed by 192.168.1.35 port 22
invaliduser@192.168.1.35's password:
Permission denied, please try again.
invaliduser@192.168.1.35's password:
Permission denied, please try again.
invaliduser@192.168.1.35's password:
invaliduser@192.168.1.35: Permission denied (publickey,password).
invaliduser@192.168.1.35's password:
Permission denied, please try again.
```

Verifying Alerts in Wazuh

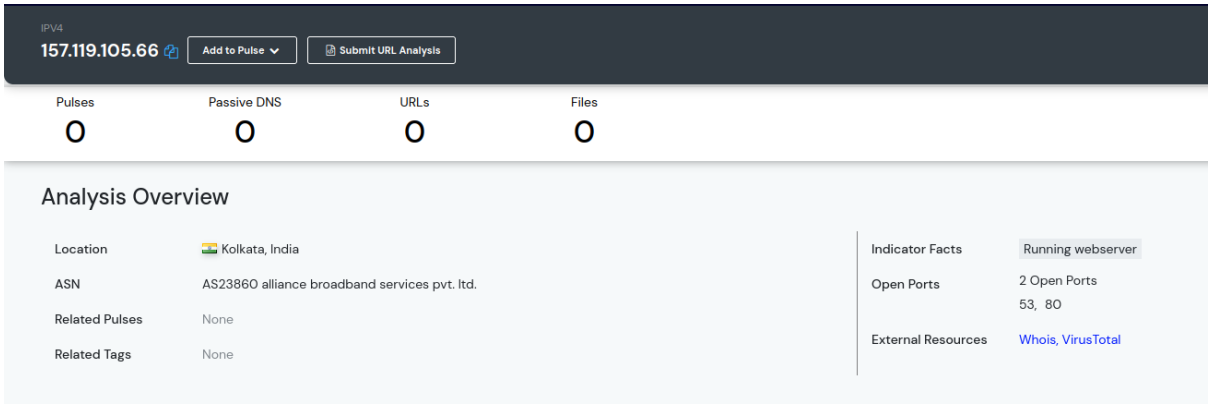
1. Open Wazuh **Security Events** tab.
2. Filter events by:
    - **Program:** sshd or sudo
    - **Username:** fakeuser1 → fakeuser5
    - **Event Type:** Authentication Failure
3. Confirm at least **5 mock alerts** are displayed.
4. Document alert details in a table:

Alert ID	Description	Source IP	Priority	Status
001	Brute-force SSH	157.119.105.66	Medium	Open
002	Brute-force SSH	157.119.105.66	Medium	Open
003	Brute-force SSH	157.119.105.66	Medium	Open
004	Brute-force SSH	157.119.105.66	Medium	Open
005	Brute-force SSH	157.119.105.66	Medium	Open



Threat Intelligence Validation using AlienVault OTX

1. Navigate to AlienVault OTX and log in.
2. Enter the **source IP** from your alerts into the search bar.
3. Analyze the results:
    - If the IP is **listed**, note associated threat categories (e.g., brute-force, botnet).
    - If the IP is **not listed**, mark as **no known threat**.



4. Document validation in a summary:

Example: “The source IP 157.119.105.66 was cross-referenced on AlienVault OTX. No malicious activity was reported, indicating a likely false positive. This confirms the alert is benign and part of mock testing.”

Mock Alerts JSON File

For completeness, here is a JSON representation of the 5 mock SSH login attempts that can be injected into Wazuh:


```
[
  {
    "_index": "wazuh-alerts-4.x-2025.08.21",
    "_id": "GNQszZgBlmAKjlAc2_wV1",
    "_score": 1,
    "_source": {
      "predecoder": { "hostname": "manjira", "program_name": "sshd", "timestamp": "Aug 21 15:08:41" },
      "agent": { "name": "manjira", "id": "000" },
      "manager": { "name": "manjira" },
      "data": { "srcip": "157.119.105.66", "dstuser": "invaliduser", "srcport": "50001" },
      "rule": { "mail": false, "level": 5, "description": "sshd: Attempt to login using a non-existent user", "id": "5710", "firedtimes": 34, "groups": ["syslog","sshd","authentication_failed","invalid_login"] },
      "decoder": { "parent": "sshd", "name": "sshd" },
      "full_log": "Aug 21 15:08:41 manjira sshd[13490]: Connection reset by invalid user invaliduser 157.119.105.66 port 50001 [preauth]",
      "input": { "type": "log" },
      "@timestamp": "2025-08-21T15:08:41.639Z",
      "location": "journald",
      "id": "1755788922.1"
    }
  }
  // Repeat similar structure for 4 more alerts with ports 50002–50005
```

]

This JSON can be used for **Wazuh mock alert injection** if you prefer not to generate events manually.

References

- https://documentation.wazuh.com
- https://otx.alienvault.com
- https://www.virustotal.com
- https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/net-user