ALERT TRIAGE WITH THREAT INTELLIGENCE

Introduction

This document outlines the simulation of alert triage and threat intelligence validation using Wazuh, VirusTotal, and AlienVault OTX. The activities include generating mock alerts for Suspicious PowerShell Execution and SSH remote login attempts, analyzing alerts in Wazuh, and validating indicators of compromise (IOCs) against public threat intelligence sources. The purpose is to demonstrate the workflow for triaging and validating alerts in a Security Operations Center (SOC) environment.

Simulating PowerShell Execution Alert

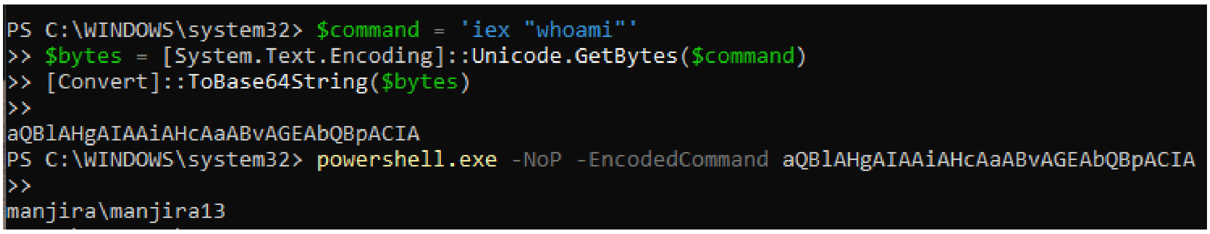
- On Windows 11, a PowerShell command was executed to simulate malicious activity:

$command = 'iex "whoami"'

$bytes = [System.Text.Encoding]::Unicode.GetBytes($command)

[Convert]::ToBase64String($bytes)

powershell.exe -NoP -EncodedCommand <base64_string>



```
PS C:\WINDOWS\system32> $command = 'iex "whoami"'
>> $bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
>> [Convert]::ToBase64String($bytes)
>>
aQBlAHgAIAAiAHcAaABvAGEAbQBpACIA
PS C:\WINDOWS\system32> powershell.exe -NoP -EncodedCommand aQBlAHgAIAAiAHcAaABvAGEAbQBpACIA
>>
manjira\manjira13
```

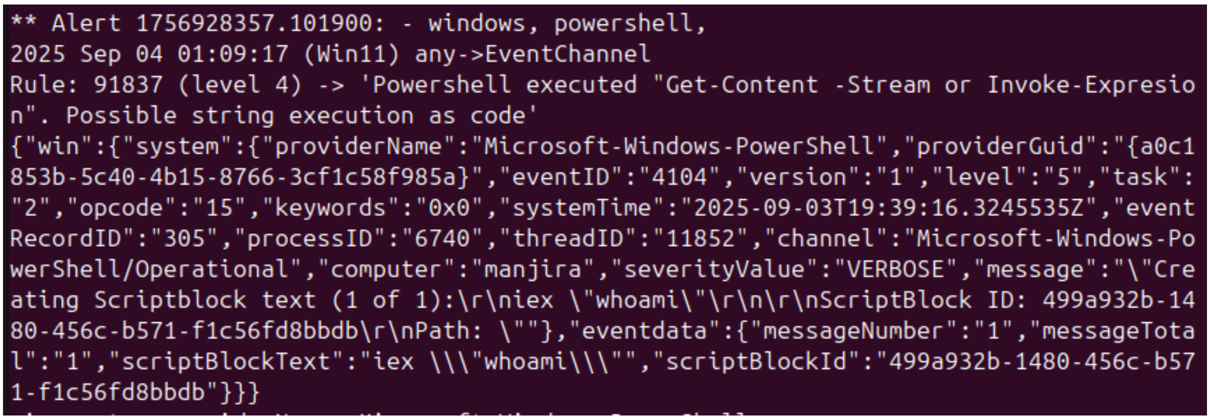
1: Generating Windows Alert

- Wazuh agent on Ubuntu captured the event and generated an alert in alerts.log.

The alert details in Wazuh included:

Alert ID: 004

Description: PowerShell Execution



```
** Alert 1756928357.101900: - windows, powershell,
2025 Sep 04 01:09:17 (Win11) any->EventChannel
Rule: 91837 (level 4) -> 'Powershell executed "Get-Content -Stream or Invoke-Expresio
n". Possible string execution as code'
{"win":{"system":{"providerName":"Microsoft-Windows-PowerShell","providerGuid":"{a0c1
853b-5c40-4b15-8766-3cf1c58f985a}","eventID":"4104","version":"1","level":"5","task":
"2","opcode":"15","keywords":"0x0","systemTime":"2025-09-03T19:39:16.3245535Z","event
RecordID":"305","processID":"6740","threadID":"11852","channel":"Microsoft-Windows-Po
werShell/Operational","computer":"manjira","severityValue":"VERBOSE","message":"\"Cre
ating Scriptblock text (1 of 1):\r\niex \"whoami\"\r\n\r\nScriptBlock ID: 499a932b-14
80-456c-b571-f1c56fd8bbdb\r\nPath: \""},"eventdata":{"messageNumber":"1","messageTota
l":"1","scriptBlockText":"iex \\\"whoami\\\"","scriptBlockId":"499a932b-1480-456c-b57
1-f1c56fd8bbdb"}}}
```

2: Wazuh Alerts

Simulating SSH Remote Login Alert

From Windows, SSH attempts were made to the Ubuntu Wazuh agent:

ssh manjira@192.168.1.35

Multiple failed login attempts triggered Wazuh to log alerts for potential brute-force activity.

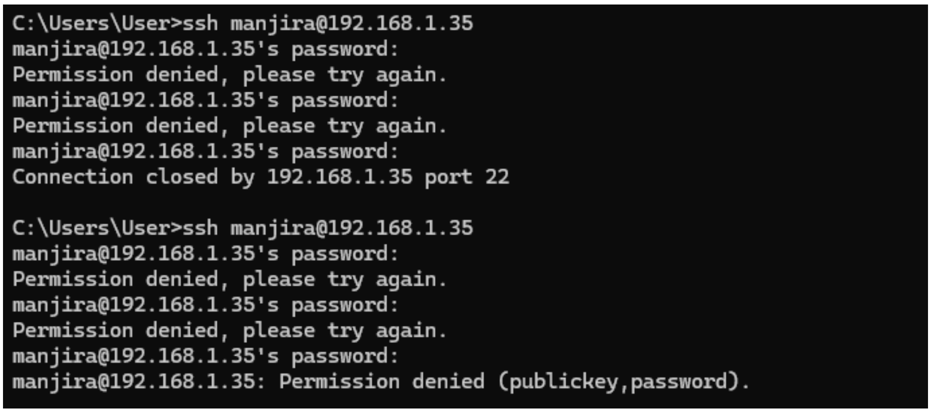
Alerts included information such as the user attempting login, timestamp, and TTY.

Triage in Wazuh

- Alerts were monitored using:

sudo tail -f /var/ossec/logs/alerts/alerts.log | grep powershell

sudo tail -f /var/ossec/logs/alerts/alerts.log | grep ssh



```
C:\Users\User>ssh manjira@192.168.1.35
manjira@192.168.1.35's password:
Permission denied, please try again.
manjira@192.168.1.35's password:
Permission denied, please try again.
manjira@192.168.1.35's password:
Connection closed by 192.168.1.35 port 22

C:\Users\User>ssh manjira@192.168.1.35
manjira@192.168.1.35's password:
Permission denied, please try again.
manjira@192.168.1.35's password:
Permission denied, please try again.
manjira@192.168.1.35's password:
manjira@192.168.1.35: Permission denied (publickey,password).
```

3: SSH Alerts Generation

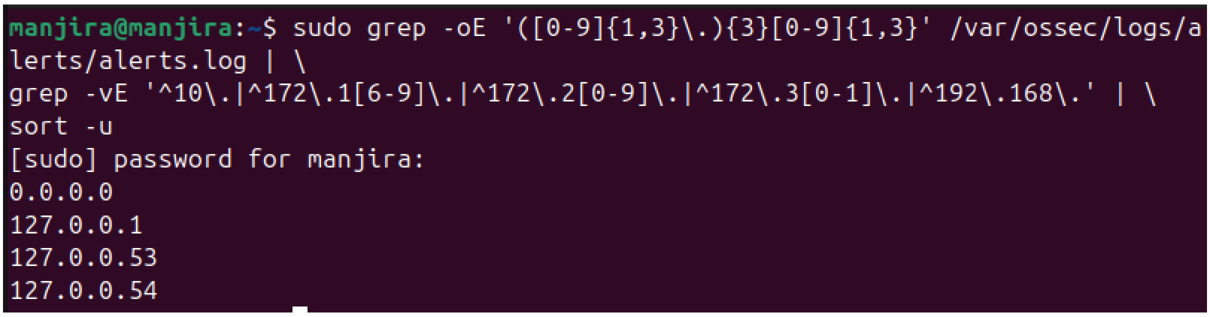
- Alerts were categorized by severity, source, and type of activity.

IOC Validation with Threat Intelligence

- **Extracted public IPs from Wazuh alerts (private IPs excluded) using:**

sudo grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' /var/ossec/logs/alerts/alerts.log | \

grep -vE '^10\.|^172\.1[6-9]\.|^172\.2[0-9]\.|^172\.3[0-1]\.|^192\.168\.' | sort –u



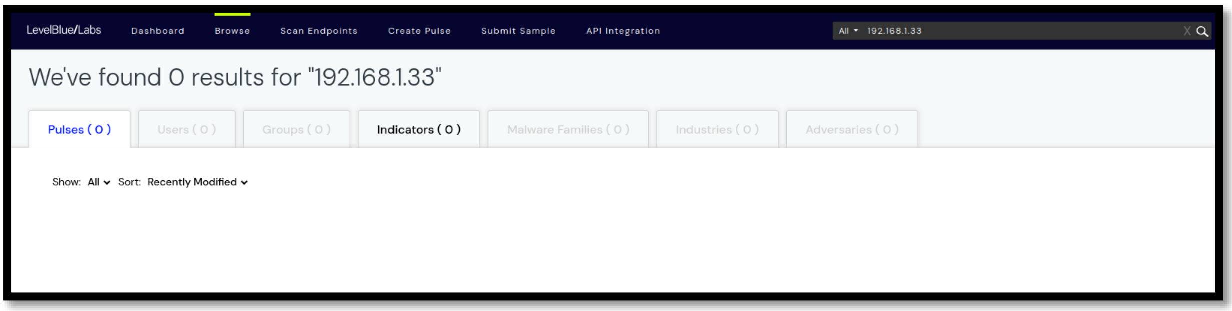
```
manjira@manjira:~$ sudo grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' /var/ossec/logs/a
lerts/alerts.log | \
grep -vE '^10\.|^172\.1[6-9]\.|^172\.2[0-9]\.|^172\.3[0-1]\.|^192\.168\.' | \
sort -u
[sudo] password for manjira:
0.0.0.0
127.0.0.1
127.0.0.53
127.0.0.54
```

4: Extracted IPs

- **Cross-referencing with VirusTotal and AlienVault OTX returned no matches because:**

Alerts were generated in a lab environment using private IP addresses.

No real malware hashes were present.



192.168.1.33

We've found 0 results for "192.168.1.33"

Pulses (0) | Users (0) | Groups (0) | Indicators (0) | Malware Families (0) | Industries (0) | Adversaries (0)

Show: All Sort: Recently Modified

5: OTX Result

Alert Summary Table

Alert ID	Description	Source IP	Priority	Status
004	PowerShell Execution	192.168.1.101	High	Open
005	SSH Remote Login	192.168.1.35	Medium	Open

Troubleshooting

- **Wazuh Manager Not Starting:**
  1. Check syntax of `local_rules.xml`.
  2. Ensure the root element is `<group>` instead of `<ruleset>`.
  3. Restart Wazuh after fixing configuration: `sudo systemctl restart wazuh-manager`.
- **Alerts Not Appearing in Dashboard:**
  1. Verify Wazuh manager is running.
  2. Ensure JSON output to Elasticsearch is enabled in `ossec.conf`.
  3. Confirm Kibana is connected to Elasticsearch and indexes are refreshed.
- **SSH Alerts Delayed:**
  1. Real-time monitoring may lag; use `tail -f` to observe alerts.
  2. Confirm SSH failed attempts are correctly logged in system logs.

References

- Wazuh Documentation: https://documentation.wazuh.com/
- VirusTotal: https://www.virustotal.com/
- AlienVault OTX: https://otx.alienvault.com/
- PowerShell Documentation: https://docs.microsoft.com/en-us/powershell/
- SSH Protocol Overview: https://www.ssh.com/ssh/protocol