Network Security Assessment and Monitoring using Nmap and Wazuh

Followed this blog to integrate NMAP and Wazuh for scanning and monitoring of vulnerabilities:  
<https://wazuh.com/blog/nmap-and-chatgpt-security-auditing/#nmap-script>

**NMAP:**  
Nmap port scanning is a technique used to discover open ports on target systems. It involves sending packets to specific ports and analyzing responses to determine whether the port is open, closed, or filtered. Nmap offers various scan types, such as TCP Connect, SYN, and UDP scans, to gather information about the services running on the target system. Additionally, it can detect the operating system and service versions to help assess potential vulnerabilities. The results are provided in different formats for easy analysis and integration with other tools. Overall, Nmap port scanning is a critical component of network reconnaissance and security auditing.

**Wazuh:**

Wazuh is an open-source security monitoring platform that provides intrusion detection, log analysis, and security event correlation capabilities. It is built on top of the ELK (Elasticsearch, Logstash, and Kibana) stack, integrating advanced security functionalities such as host-based intrusion detection system (HIDS), file integrity monitoring (FIM), and vulnerability detection.

Wazuh collects and analyzes security data from multiple sources, including logs, system events, and network traffic, to identify and respond to security threats effectively. It offers real-time alerting, centralized logging, and compliance management features, making it suitable for both small businesses and large enterprises.

The platform's architecture consists of agents installed on endpoints, a central manager for log aggregation and analysis, and a user interface (Kibana) for visualization and reporting. Wazuh is highly customizable, allowing users to define their own rules, policies, and alerts tailored to their specific security requirements.

**Integrating NMAP with Wazuh:**

Integrating Nmap with Wazuh involves leveraging Nmap's network scanning capabilities to identify open ports and services on target systems and then forwarding this information to Wazuh for analysis and response. This integration enhances Wazuh's ability to detect and respond to potential security threats by providing additional context about the services running on monitored endpoints.

The process typically involves:

1. Performing Nmap scans on target systems to identify open ports and services.
2. Configuring Wazuh to ingest Nmap scan results, either by parsing Nmap output files or by directly integrating Nmap scans into Wazuh agents.
3. Analyzing Nmap scan results within Wazuh to correlate them with other security events and detect any anomalies or potential security breaches.
4. Triggering alerts or automated responses based on the Nmap scan findings, such as blocking suspicious ports or services, generating notifications, or initiating further investigation.

Overall, integrating Nmap with Wazuh provides organizations with a more comprehensive security monitoring solution, enabling them to identify and respond to network-based threats more effectively.

**Infrastructure:**

* Wazuh ova 4.7
* Windows 11 endpoint that has wazuh agent setup

**Following is the NMAP python script I used provided by** [**Iseoluwa Titiloye Oyeniyi**](https://wazuh.com/?s=Iseoluwa+Titiloye+Oyeniyi&filter=blog)**:**  
  
#!/var/ossec/framework/python/bin/python3

# Copyright (C) 2015-2023, Wazuh Inc.

import nmap

import time

import json

import platform

# The function to perform network scan on a host endpoint

def scan\_subnet(subnet):

nm = nmap.PortScanner()

nm.scan(subnet)

results = []

for host in nm.all\_hosts():

for proto in nm[host].all\_protocols():

if proto not in ["tcp", "udp"]:

continue

lport = list(nm[host][proto].keys())

lport.sort()

# Iterate over each port for the current host and protocol

for port in lport:

hostname = ""

json\_output = {

'nmap\_host': host,

'nmap\_protocol': proto,

'nmap\_port': port,

'nmap\_hostname': "",

'nmap\_hostname\_type': "",

'nmap\_port\_name': "",

'nmap\_port\_state': "",

'nmap\_port\_service': ""

}

# Get the first hostname and it’s type

if nm[host]["hostnames"]:

hostname = nm[host]["hostnames"][0]["name"]

hostname\_type = nm[host]["hostnames"][0]["type"]

json\_output['nmap\_hostname'] = hostname

json\_output['nmap\_hostname\_type'] = hostname\_type

# Get the port name if available

if 'name' in nm[host][proto][port]:

json\_output['nmap\_port\_name'] = nm[host][proto][port]['name']

# Get the port state if available

if 'state' in nm[host][proto][port]:

json\_output['nmap\_port\_state'] = nm[host][proto][port]['state']

# Get the port service version if available

if 'product' in nm[host][proto][port] and 'version' in nm[host][proto][port]:

service = nm[host][proto][port]['product'] + " " + nm[host][proto][port]['version']

json\_output['nmap\_port\_service'] = service

results.append(json\_output)

return results

# The function to append the scan results to the active response log file

def append\_to\_log(results, log\_file):

with open(log\_file, "a") as active\_response\_log:

for result in results:

active\_response\_log.write(json.dumps(result))

active\_response\_log.write("\n")

# Specify the address(es) to scan

subnets = ['127.0.0.1']

# path of the log file

if platform.system() == 'Windows':

log\_file = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"

elif platform.system() == 'Linux':

log\_file = "/var/ossec/logs/active-responses.log"

else:

log\_file = "/Library/Ossec/logs/active-responses.log"

for subnet in subnets:

results = scan\_subnet(subnet)

append\_to\_log(results, log\_file)

time.sleep(2)

**Windows Endpoint:**

* Python 3
* NMAP v7.94
* [Microsoft Visual C++ 2015 Redistributable](https://www.microsoft.com/en-us/download/confirmation.aspx?id=52685).
* Python setup and permissions

**Configuring the ossec conf file:**  
<!-- Run nmap python script -->

<localfile>

<log\_format>full\_command</log\_format>

<command>C:\Users\<USERNAME>\Documents\nmapscan.exe</command>

<frequency>604800</frequency>

</localfile>

**Adding rule in the Wazuh server:**

<group name="linux,nmap,">

<rule id="100100" level="3">

<decoded\_as>json</decoded\_as>

<field name="nmap\_port">\.+</field>

<field name="nmap\_port\_service">\.+</field>

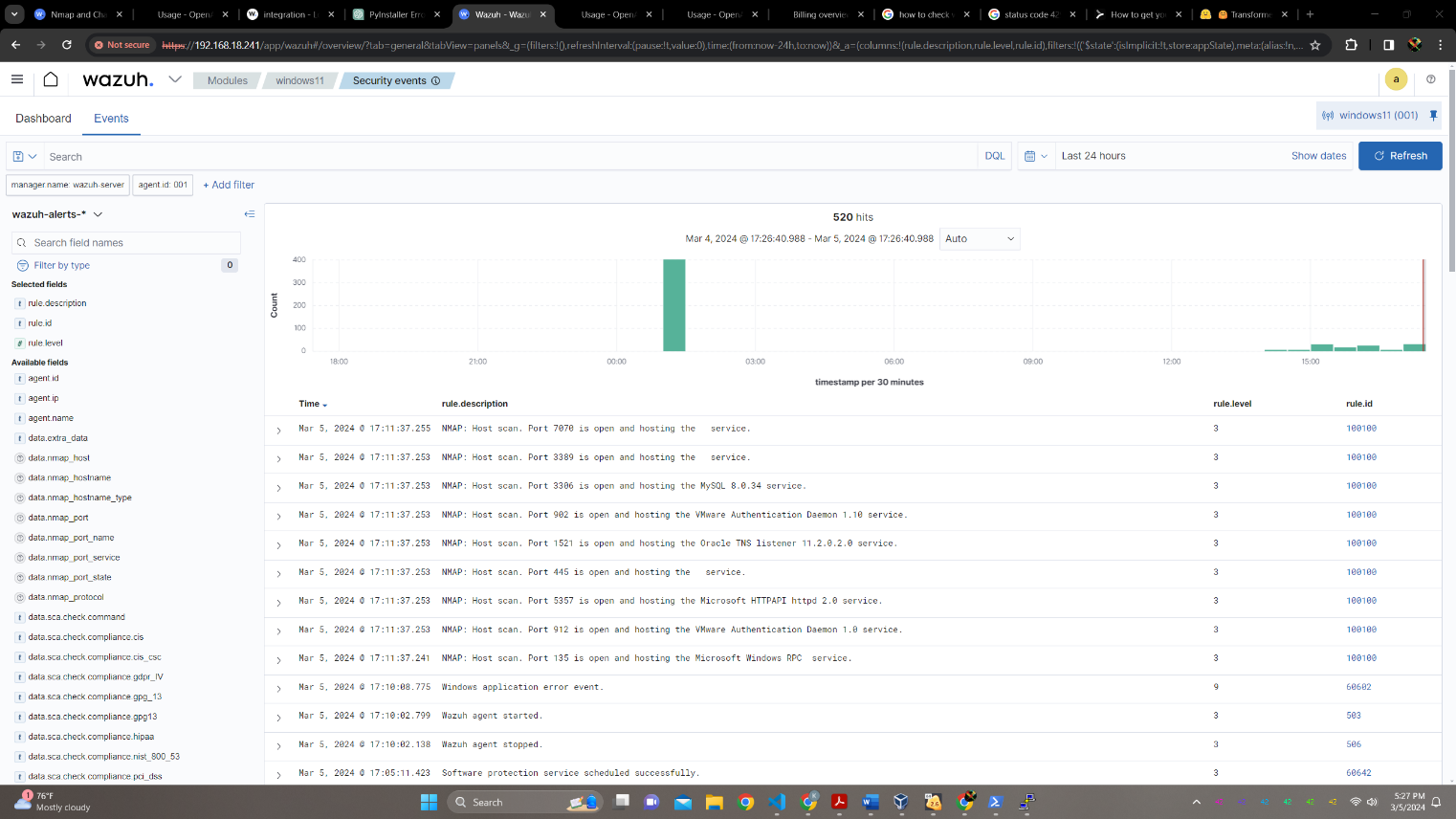
<description>NMAP: Host scan. Port $(nmap\_port) is open and hosting the $(nmap\_port\_service) service.</description>

<options>no\_full\_log</options>

</rule>

</group>

**Scan Results:**

****

